NATAL, 2018

Realização: SBC

Organização: UFRN

anais 2018

XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

12º BRESCI – BRAZILIAN E-SCIENCE WORKSHOP

CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018

#COMPUTAÇÃOESUSTENTABILIDADE

CSBC 2018

anais 2018

XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO
CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018
#COMPUTAÇÃOESUSTENTABILIDADE

Coordenador Geral

Francisco Dantas de Medeiros Neto (UERN)

Comissão Organizadora

Bartira Paraguaçu Falcão Dantas Rocha (UERN)
Camila Araújo Sena (UERN)
Everton Ranielly de Sousa Cavalcante (UFRN)
Felipe Torres Leite (UFERSA)
Ilana Albuquerque (UERN)
Isaac de Lima Oliveira Filho (UERN)
Priscila Nogueira Krüger (UERN)

Realização

Sociedade Brasileira de Computação

Organização

Universidade do Estado do Rio Grande do Norte

CSBC 2018
XXXVIII Congresso da
Sociedade Brasileira de Computação

Apresentação

Estes anais registram os trabalhos apresentados durante o XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018), realizado em Natal-RN, de 22 a 26 de julho 2018. O evento teve como tema central a Computação e Sustentabilidade, pois se compreende que o avanço da computação e as questões ambientais devem caminhar lado-a-lado, tendo em vista que as técnicas computacionais necessitam ser usadas para possibilitar o desenvolvimento sustentável, e, desse modo, equilibrar as necessidades ambientais, econômicas e sociais.

Organizar o maior evento acadêmico de Computação da América Latina foi um privilégio e um desafio. Foi enriquecedor promover e incentivar a troca de experiências entre estudantes, professores, profissionais, pesquisadores e entusiastas da área de Computação e Informática de todo o Brasil. Ao mesmo foi desafiador termos que lidar, principalmente, com às dificuldades impostas pelo momento de crise que o nosso Brasil vem enfrentando. Uma crise que afeta diretamente nossas pesquisas e, consequentemente, o desenvolvimento e inovação do nosso amado Brasil.

Por meio de seus 25 eventos, o CSBC 2018 apresentou mais de 300 trabalhos, várias palestras e mesas-redondas. O Congresso ainda abrigou diversas reuniões, que incluem a reunião do Fórum de Pós-Graduação, a reunião do CNPq/CAPES, a reunião dos Secretários Regionais SBC, a reunião das Comissões Especiais e a reunião do Fórum IFIP/SBC.

O sucesso do CSBC 2018 só foi possível devido à dedicação e entusiasmo de muitas pessoas. Gostaríamos de agradecer aos coordenadores dos 25 eventos e aos autores pelo envio de seus trabalhos. Além disso, gostaríamos de expressar nossa gratidão ao Comitê Organizador, por sua grande ajuda em dar forma ao evento; e, em especial, à equipe da Sociedade Brasileira de Computação (SBC), por todo apoio.

Por fim, reconhecemos a importância do apoio financeiro da CAPES, do CNPq, do CGI.br, do Governo do Estado do Rio Grande do Norte, da Prefeitura Municipal do Natal, da Prefeitura Municipal de Parnamirim, da CABO Telecom, da ESIG Software e Consultoria, da DynaVideo e do SENAI.

Natal (RN), 26 de julho de 2018.

Chico Dantas (UERN)
Coordenador Geral do CSBC 2018

Anais do CSBC 2018

12º BRESCI – BRAZILIAN E-SCIENCE WORKSHOP

12º BRESCI
Brazilian e-Science Workshop

Apresentação

Nas últimas décadas, tem havido uma revolução no modo como a ciência e a engenharia têm sido conduzidas, ao se utilizar de forma intensiva as tecnologias de informação e comunicação (TICs). Essa nova forma de realizar a ciência, denominada de *e-Science* ou e-Ciência, desempenha hoje um papel fundamental na metodologia de trabalho adotada por muitos grupos de pesquisa em todo o mundo.

O XII BreSci tem como objetivo colaborar com os esforços de *e-Science* propondo um fórum de discussão sobre temas relevantes dessa área de estudo. Além da trilha principal, que tem um escopo mais amplo e mais relacionado com as TICs, o workshop conta também com uma trilha de aplicações específica para discutir temas relacionados às áreas particulares de aplicação da *e-Science*.

A aproximação com pesquisadores dessas áreas da ciência visa a estreitar o relacionamento entre os participantes das diversas áreas. Ademais, essa aproximação propicia a identificação de demandas relativas à infraestrutura computacional sob o ponto de vista das áreas-fim. De outro lado, essa colaboração também propicia às áreas-fim uma melhor difusão das soluções elaboradas pela comunidade de computação.

O BreSci, em sua décima segunda edição, sendo a nona edição colocada no Congresso da Sociedade Brasileira de Computação (CSBC), recebeu submissões tanto de artigos completos quanto de artigos resumidos em suas duas trilhas. À edição de 2018 do evento, foram submetidos 19 trabalhos no total (6 deles para a trilha de aplicações). Desses, 9 foram aceitos para publicação como artigos completos, sendo 3 deles da trilha de aplicações. Além disso, mais 6 trabalhos foram aceitos para publicação como artigos resumidos, sendo 2 deles da trilha de aplicações. Os artigos aceitos abordam temas relevantes de pesquisa em Computação, como integração de dados, ontologias, reconhecimento de padrões, processamento de alto desempenho, simulação, workflows científicos e outros modelos e técnicas habilitadores da *e-Science*, bem como aplicações em diversas áreas, como astronomia, bioinformática, biodiversidade, geologia, geografia e engenharia ambiental.

Carla Osthoff Ferreira de Barros (LNCC)
Kelly Rosa Braghetto (USP)
Coordenadoras do BreSci 2018

Coordenação Geral

- Carla Osthoff Ferreira de Barros (LNCC)
- Kelly Rosa Braghetto (USP)

Comitê Diretivo

- Carla Osthoff Ferreira de Barros (LNCC, co-chair 2018)
- Kelly Rosa Braghetto (USP, co-chair 2018)
- Rafaelli Coutinho (CEFET/RJ, co-chair 2017)
- Emanuele Santos (UFC, co-chair 2017)
- Kary A. D. C. S. Ocaña (LNCC, co-chair 2016)
- Luiz M. R. Gadelha Jr. (LNCC, co-chair 2015)

Coordenação Local

- Demóstenes Santos de Sena (IFRN)

Comitê de Programa

- Ana Carolina Guimarães (FIOCRUZ)
- André Rodrigues Soares (LNCC)
- Andrey Brito (UFCG)
- Antônio Tadeu Gomes (LNCC)
- Bruno Schulze (LNCC)
- Carla Osthoff (LNCC)
- Cristina Boeres (UFF)
- Daniel Cordeiro (USP)
- Daniel de Oliveira (UFF)
- Debora Drucker (EMBRAPA)
- Diogo Tschoeke (UFRJ)
- Duncan Ruiz (PUC-RS)
- Eduardo Bezerra (CEFET/RJ)
- Eduardo Dalcin (Inst. Pesq. Jardim Botânico)
- Eduardo Ogasawara (CEFET/RJ)
- Emanuele Santos (UFC)
- Fábio Lopes (Mackenzie)
- Fábio Porto (LNCC)
- Fernanda Campos (UFJF)
- Gilberto Pastorello (Berkeley Lab)
- Glauber Wagner (UFSC)
- João Gomes (UFC)
- João Setubal (USP)
- Jonas Dias (DELL EMC)
- José Antonio Macêdo (UFC)
- Kary Ocaña (LNCC)
- Kele Belloze (CEFET/RJ)
- Kelly Braghetto (USP)
- Leonardo Azevedo (IBM/UNIRIO)

- Leonardo Murta (UFF)
- Luciano Digiampietri (USP)
- Luiz Manoel Rocha Gadelha Júnior (LNCC)
- Maicon Alves (UFF)
- Marco Antônio Araújo (UFJF)
- Mariano Silva (LNCC)
- Priscila Goliatt (UFJF)
- Rafaelli Coutinho (CEFET/RJ)
- Regina Braga (UFJF)
- Ricardo Ogando (ON-MCTI & LIneA)
- Roberto Pinto Souto
- Sérgio Lifschitz (PUC-Rio)
- Sérgio Manuel Serra da Cruz (UFRRJ)
- Silvia Olabarriaga (University of Amsterdam)
- Tainá Raiol (FIOCRUZ)
- Ubiratam De Paula (UFRRJ)
- Victor Stroele (UFJF)
- Vinod Rebello (UFF)
- Yuri Nogueira (UFC)

12º BRESCI – BRAZILIAN E-SCIENCE WORKSHOP (BRESCI 2018)

Minicurso

Introdução ao Deep Learning – Classificação de Imagens

João Paulo Navarro[1] - Arquiteto de Soluções, NVIDIA

Resumo:

Deep Learning está dando às máquinas capacidades super-humanas de reconhecimento de objetos, e substituindo o modelo *rule-based* por modelos preditivos aprendidos diretamente dos dados. Este minicurso apresenta uma introdução ao fluxo de trabalho de *machine learning* e oferece uma experiência prática na utilização de redes neurais profundas (DNN – *Deep Neural Networks*) para resolver problemas reais de classificação de imagens. A metodologia apresentada inclui as etapas de preparação dos dados, definição do modelo, treinamento da rede, estratégias de validação e testes para melhoria do modelo. Este treinamento mostra os benefícios da utilização de GPUs no treinamento de redes neurais profundas. Ao final, o aluno terá o conhecimento necessário para utilizar a ferramenta NVIDIA DIGITS para treinar uma DNN em seu próprio conjunto de imagens para classificação.

Palestra

Plataforma NVIDIA para Inteligência Artificial e Deep Learning

João Paulo Navarro[1] - Arquiteto de Soluções, NVIDIA

Resumo:

Deep Learning (DL) é a técnica de *Machine Learning* que vem proporcionando avanços surpreendentes nos mais variados fluxos de trabalho da indústria e academia. A Inteligência Artificial moderna é a 4ª revolução industrial e a plataforma da NVIDIA fornece poder computacional para os mais complexos algoritmos de DL. A nova arquitetura de GPUs Volta, juntamente com o CUDA 9 e os SDKs da NVIDIA, foram aprimorados para incluir algoritmos especializados e altamente otimizados para extrair o máximo potencial das placas de vídeo no treinamento de algoritmos de DL, utilizados nos mais importantes *frameworks* da atualidade (TensorFlow, Caffe, Torch, etc.). Veremos aplicações das técnicas de DL nas mais variadas áreas do conhecimento, como Visão Computacional, Carros Autônomos e Robótica.

[1] Bio: João Paulo Navarro é Cientista da Computação e mestre em Modelagem Computacional (UFJF), tendo dedicado boa parte de sua carreira ao desenvolvimento projetos de computação científica, simulação física e *machine learning*. Possui vasta experiência no desenvolvimento de algoritmos e técnicas de visualização voltadas ao processamento geofísico. Hoje, na NVIDIA, é Arquiteto de Soluções com foco em computação de alto desempenho e *Deep Learning.*

12º BRESCI – BRAZILIAN E-SCIENCE WORKSHOP (BRESCI 2018)

Artigos Completos

Artigos Resumidos

Análise de Eficiência Alimentar de Gado Leiteiro a partir da Integração de Bases Heterogêneas e Ontologias

Heitor Magaldi[1], Regina Braga[1], Wagner Arbex[1,2], Mariana Magalhães Campos[2], Carlos Cristiano Hasenclever Borges[1], José Maria N. David[1], Fernanda Campos[1] Victor Stroele[1]

[1] Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brasil

[2] Embrapa Gado de Leite, Empresa Brasileira de Pesquisa Agropecuária – Embrapa, Juiz de Fora, Minas Gerais, Brasil

heitor.magaldi@gmail.com.br, {regina.braga, wagner.arbex, jose.david, carlos.borges, fernanda.campos, victor.stroele}@ufjf.edu.br, mariana.campos@embrapa.br

Abstract. *With today's increasingly competitive market, dairy farmers need to cut costs and make their herds competitive. In this sense, the computational support has provided alternatives to the identification of more efficient animals and, consequently, providing economic and environmental gains. This paper presents an architecture to support food efficiency research developed at Embrapa Gado de Leite, with the aim of discovering new knowledge and new relationships in a large experiment dataset, using ontologies and data analysis visualization techniques. The preliminary evaluation results showed to be promising. Therefore, we consider that the use of ontologies and visualization techniques can contribute to the advancement of research in feed efficiency.*

Resumo. *Com o mercado atual cada vez mais competitivo, os produtores de leite precisam reduzir custos e tornar seus rebanhos competitivos. Nesse sentido, o suporte computacional tem proporcionado alternativas para a identificação de animais mais eficientes e, consequentemente, proporcionando ganhos econômicos e ambientais. Este artigo apresenta uma arquitetura de apoio à pesquisa de eficiência alimentar desenvolvida na Embrapa Gado de Leite, com o objetivo de descobrir novos conhecimentos e novas relações em um grande conjunto de dados experimentais, utilizando de técnicas de visualização, ontologias e análise de dados. Os resultados preliminares da avaliação mostraram-se promissores. Portanto, consideramos que o uso de ontologias e técnicas de visualização podem contribuir para o avanço da pesquisa em eficiência alimentar.*

1. Introdução

Atualmente, a pecuária leiteira brasileira vem lidando com novos desafios. O aumento dos custos de produção, da preocupação dos consumidores com a qualidade e segurança alimentar, do bem-estar animal e impactos ambientais da agropecuária ocasionam margens de lucro menores aos produtores [Campos et al. 2015]. Campos et al. (2015) destacam que os gastos com a alimentação representam o principal custo da atividade

pecuária. Além disso, com a eficiência alimentar, além de impactos econômicos, impactos ambientais são observados, pois animais eficientes produzem menor desperdício de nutrientes e excreções. Um animal é classificado como eficiente quando atinge o mesmo nível de produção consumindo menos alimento que os demais.

Hoje, os pesquisadores analisam esses dados de forma manual, utilizando de ferramentas estatísticas para seleção dos animais eficientes e, a partir disso, analisarem em que os animais eficientes diferem dos não eficientes. Essas análises são realizadas a partir de planilhas eletrônicas. O cruzamento de dados entre animais, considerando diferentes índices de eficiência alimentar e considerando os diversos experimentos é uma atividade que demanda tempo e expertise computacional e por conta disso, nem sempre é realizada. Além disso, a possibilidade de uso de ferramental adequado para estas análises em diversos contextos, seja *in loco* onde os animais vivem, sejam em laboratórios específicos, é um requisito importante. Dessa forma, uma arquitetura para o apoio das pesquisas em eficiência alimentar, capaz de processar análises sobre os dados considerando descoberta de informações implícitas, permitindo avaliar os dados sobre diferentes perspectivas, é de grande importância para os avanços das pesquisas nessa área. Além disso, o uso de uma arquitetura orientada a serviços, considerando uma abordagem de *Software as a Service*, é importante neste contexto, permitindo que diferentes aplicações façam uso dessas análises e os resultados possam ser reutilizados em diferentes contextos.

O artigo está organizado em quatro seções. Na seção 2, apresenta-se alguns conceitos importantes e trabalhos relacionados. A seção 3 apresenta a arquitetura proposta. A seção 4 traz as discussões sobre os resultados; e, por fim, a seção 5, as conclusões.

2. Trabalhos Relacionados

Gruber (1993) definiu ontologia como uma “especificação explicita de um conceito”. Guarino, em 1998, por sua vez, descreveu que a ontologia pode ser descrita nas mais variadas formas, dependendo de seu campo de aplicação. Ontologias computacionais são meios para modelar formalmente a estrutura de um sistema, isto é, as entidades relevantes e suas relações que emergem a partir da sua observação e que são úteis para os objetivos desejados.

Em 1999, Baker et al. destacam o crescimento do uso de ontologias no domínio de engenharia de software e aplicações web, com o intuito de promover a integração, interoperabilidade e visualização dos dados. Posteriormente, Miah, Gammack e Kerr (2007) desenvolveram um modelo ontológico capaz de centralizar o acesso aos dados de diversas bases, com objetivo de facilitar a consulta e descoberta de informações de forma mais simplificada aos usuários. Em 2015, Verhoosel, Bekkum e Evert propuseram uma abordagem com uso de ontologias com o propósito de unir bases distintas, proporcionando a análise de um grande volume de dados. Os autores desenvolveram uma ontologia composta de 28 conceitos relacionados a eficiência alimentar.

Tomic et al. (2015), com a arquitetura agriOpenLink, utilizam ontologia para centralizar as consultas e acesso aos dados produzidos nos diversos serviços relacionados a agropecuária, tais como dados climáticos, eólicos, pragas, raças, entre outros. Observa-se, pelos autores citados, a utilização da ontologia para fins exclusivos

de mapeamento. Vale frisar, até o presente momento, nas buscas realizadas em bases indexadas da área da computação, não houveram retornos similares a abordagem proposta neste artigo. Dessa forma, o presente trabalho busca apresentar uma arquitetura capaz de apoiar as pesquisas em eficiência alimentar, a partir do uso de ontologias e análise de dados, utilizando bases de experimentos coletados em campo.

3. Arquitetura *FeedEfficiencyService*

Considerando a necessidade de interoperabilidade entre os dados dos diversos experimentos relacionados a eficiência alimentar e a dificuldade na realização de análises precisas acerca dos dados, incluindo a descoberta de informações implícitas que poderiam apoiar a evolução dos experimentos, a arquitetura *FeedEfficiencyService* foi proposta no contexto das pesquisas em eficiência alimentar da EMBRAPA-Gado de Leite.

Inicialmente, devido aos diversos experimentos já conduzidos estarem armazenados em bases de dados heterogêneas, foi desenvolvida uma camada tradutora (*wrapper*) genérica que permitiu a interoperabilidade dos experimentos já conduzidos para o modelo de dados da arquitetura, conforme apresentado na Figura 1. Esta camada foi especializada para considerar dados específicos dos experimentos analisados. No entanto, caso novos experimentos devam ser analisados, a camada genérica pode ser facilmente especializada para um novo contexto.

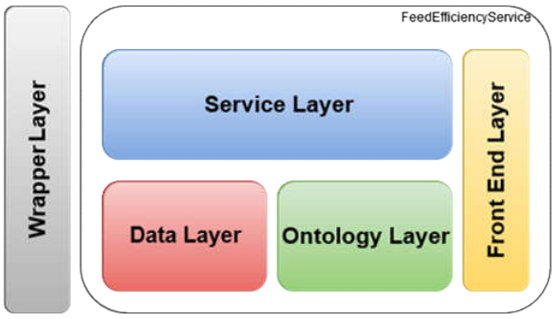

Figura 1. Arquitetura *FeedEfficiencyService*.

Para o armazenamento dos experimentos, um modelo de dados foi especificado, de forma englobar facilmente dados dos novos experimentos. Para apoio a integração e análises dos dados dos experimentos, foi especificada uma ontologia, denominada *Feed Efficiency Ontology* (FEO) (Figura 2). A ontologia permite a integração semântica entre os experimentos relacionados a eficiência alimentar, permitindo aos pesquisadores a classificação dos animais nos experimentos e a interoperabilidade entre os dados, com vistas a realizar análises cruzadas e descoberta de novas conexões entre experimentos.

A linguagem adotada para implementação da ontologia foi o OWL (*Web Ontology Language*), recomendada pelo W3C [Bechhofer et al. 2004; Hitzler, Krotzsch e Rudolph 2009]. Além disso, a ontologia foi especificada para a classificação dos índices de eficiência alimentar. Assim, foram criadas classes específicas e regras específicas para realizar a classificação, considerando os três possíveis níveis: eficiente, intermediário e ineficiente (Figura. 2).

A classificação dos animais se dá através do cálculo de seus respectivos índices, cujas informações foram obtidas junto aos pesquisadores da EMBRAPA Gado de Leite.

Para os cálculos, a arquitetura necessita que o pesquisador insira os dados alimentares do experimento, Ganho de Peso Diário (GPDkg), Ingestão de Matéria Seca (IMS) e Peso Metabólico Médio (PMmediokg) de cada animal participante do experimento. Considerando que a obtenção desses dados é por meio de análises químicas e que essas variam conforme o alimento fornecido, não foi possível a automatização dessa etapa.

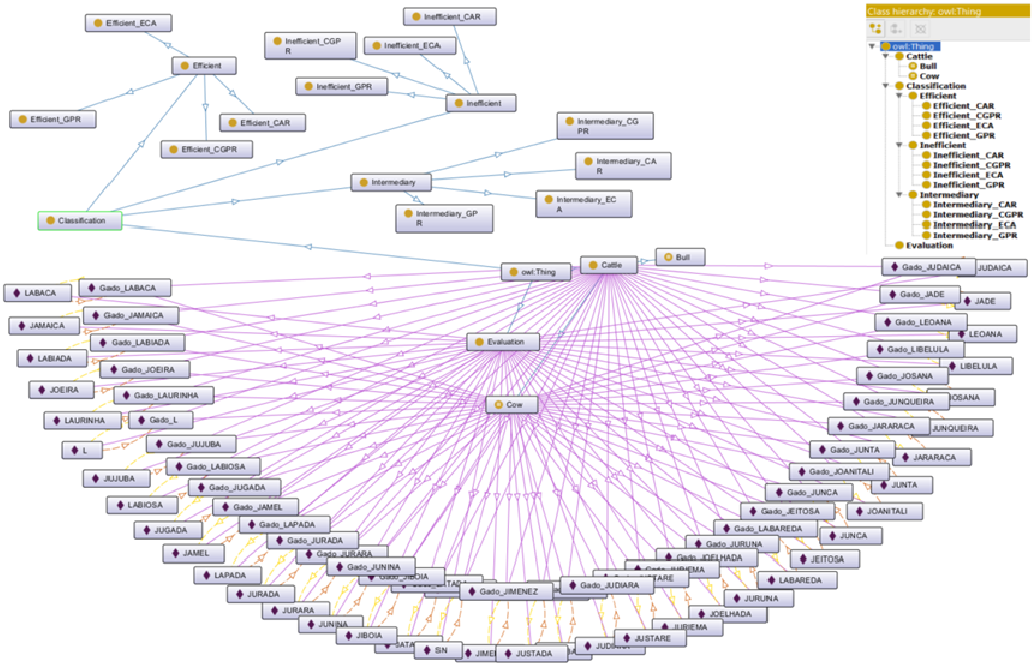

Figura 2. *Feed Efficiency Ontology* (FEO).

Os valores de IMS Observado e GMD Observado representados na Tabela 1 referem-se aos valores reais coletados durante todo o experimento. Os três índices adotados para o experimento, apesar de fórmulas distintas, baseiam-se nos dados alimentares GPDkg, IMS e PMmediokg para suas apurações, apresentado na Tabela 1.

Tabela 1. Fórmulas para apuração dos índices de eficiência alimentar.

Sigla	Descrição	Fórmula
CAR	Consumo Alimentar Residual	IMS Observado – IMS Esperado
ECA	Eficiência de Conversão Alimentar	IMS/GMD
GPR	Ganho de Peso Residual	GMD Observado – GMD Esperado
CGPR	Consumo e Ganho de Peso Residual	GPD + (CAR * (-1))

A apuração dos valores IMS esperado e GMD esperado é obtida através de regressões lineares múltiplas. Para a classificação automática dos animais segundo os índices e posterior análise dos resultados, foram criadas regras lógicas na ontologia, que permitem a classificação e a identificação de novas relações ontológicas. Para isso, regras especificadas em SWRL (*Semantic Web Rule Language*) foram criadas, considerando os índices CAR, GPR e ECA (Figura 3). As regras SWRL (Figura 3) permitem a composição de associações em busca de novo conhecimento, além da redução da complexidade da consulta à ontologia através do SPARQL. Por exemplo, a regra S1 (Figura 4) tem o papel de classificar os animais eficientes sobre o índice CAR.

Para tal, ela utiliza informações previamente conhecidas, tais como: ser uma instância de *Cattle*, possuir uma avaliação no índice CAR e ter uma avaliação inferior a -0.13. Assim, um animal que possua essas combinações é classificado como CAR eficiente.

Name	Rule
S1	Cattle(?cattle) ^ isEvaluationOf(?cattle, ?evaluation) ^ Experiment CAR(?evaluation, ?EvaluationCAR) ^ swrlb:lessThan(?EvaluationCAR, -0.13) -> Efficient CAR(?cattle)
S2	Cattle(?c) ^ isEvaluationOf(?c, ?y) ^ Experiment CAR(?y, ?EvaluationCAR) ^ swrlb:greaterThan(?EvaluationCAR, 0.13) -> Inefficient CAR(?c)
S3	Cattle(?c) ^ isEvaluationOf(?c, ?y) ^ Experiment CAR(?y, ?EvaluationCAR) ^ swrlb:lessThanOrEqual(?EvaluationCAR, 0.13) ^ swrlb:greaterThanOrEqual(?EvaluationCAR, -0.13) -> Intermediary CAR(?c)
S4	Cattle(?cattle) ^ isEvaluationOf(?cattle, ?evaluation) ^ Experiment GPR(?evaluation, ?EvaluationGPR) ^ swrlb:greaterThan(?EvaluationGPR, 0.0422) -> Efficient GPR(?cattle)
S5	Cattle(?cattle) ^ isEvaluationOf(?cattle, ?evaluation) ^ Experiment GPR(?evaluation, ?EvaluationGPR) ^ swrlb:lessThan(?EvaluationGPR, -0.0422) -> Inefficient GPR(?cattle)
S6	Cattle(?c) ^ isEvaluationOf(?c, ?y) ^ Experiment GPR(?y, ?EvaluationGPR) ^ swrlb:lessThanOrEqual(?EvaluationGPR, 0.0422) ^ swrlb:greaterThanOrEqual(?EvaluationGPR, -0.0422) -> Intermediary GPR(?c)
S7	Cattle(?cattle) ^ isEvaluationOf(?cattle, ?evaluation) ^ Experiment ECA(?evaluation, ?EvaluationECA) ^ swrlb:lessThan(?EvaluationECA, -0.3685) -> Efficient ECA(?cattle)
S8	Cattle(?c) ^ isEvaluationOf(?c, ?y) ^ Experiment ECA(?y, ?EvaluationECA) ^ swrlb:greaterThan(?EvaluationECA, 0.3685) -> Inefficient ECA(?c)
S9	Cattle(?c) ^ isEvaluationOf(?c, ?y) ^ Experiment ECA(?y, ?EvaluationECA) ^ swrlb:lessThanOrEqual(?EvaluationECA, 0.3685) ^ swrlb:greaterThanOrEqual(?EvaluationECA, -0.3685) -> Intermediary ECA(?c)

Figura 3. Regras SWRL.

```
Cattle(?cattle) ^ isEvaluationOf(?cattle, ?evaluation) ^ Experiment CAR(?evaluation, ?EvaluationCAR)
^ swrlb:lessThan(?EvaluationCAR, -0.13) -> Efficient CAR(?cattle)
```

Figura 4. Regra SWRL – S1 : CAR Eficiente .

Foi implementado um serviço web RESTful em JAVA, responsável por disponibilizar serviços para o armazenamento, gerência e consulta aos dados. E, através desse permitir a interoperabilidade com outras aplicações e serviços. Com objetivo de ilustrar a interação entre as tecnologias, um cenário de uso foi elaborado para um melhor entendimento dessa. Assim, a Figura 5 ilustra os passos de uma requisição do pesquisador até o retorno do serviço web com a visualização desejada.

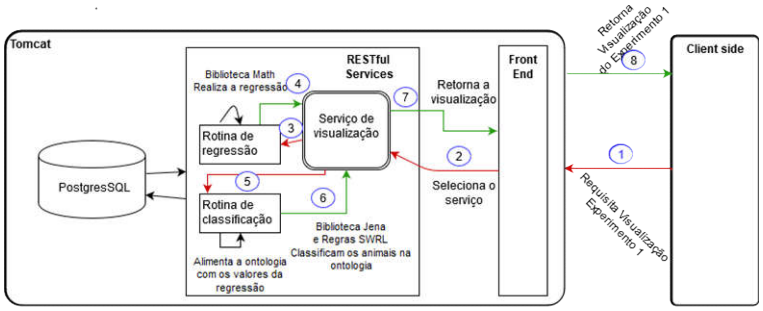

Figura 5. Exemplo de interação entre tecnologias, a fim de retornar a classificação dos animais no experimento.

Foi também elaborado um diagrama de sequência com os passos anteriores (sem uso da arquitetura *FeedEfficiencyService*) (Figura 6) e com o uso da arquitetura (Figura 7), para a classificação dos animais em um experimento. A constante necessidade da intervenção do pesquisador e a inexistência de automação para a transcrição e a análise dos dados produzidos estão presentes no modelo anterior. Nesse sentido, um comparativo entre os experimentos, os animais e os índices era uma tarefa complexa, pois os comparativos ocorriam em planilhas e as análises ficavam a cargo do pesquisador. Assim, quanto maior o volume dos experimentos, animais e índices, mais inviável ficavam as análises.

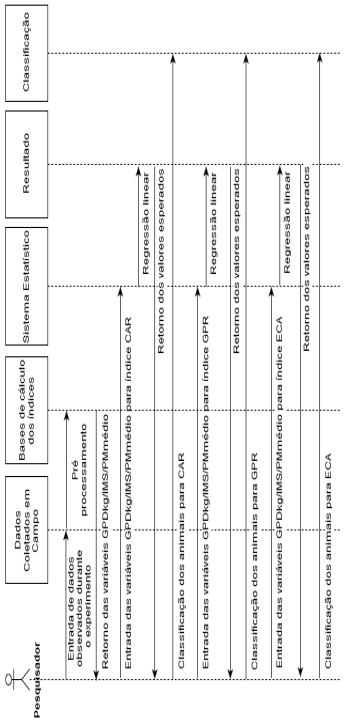

Figura 6. Diagrama de sequência para classificação dos animais, anterior ao uso da arquitetura.

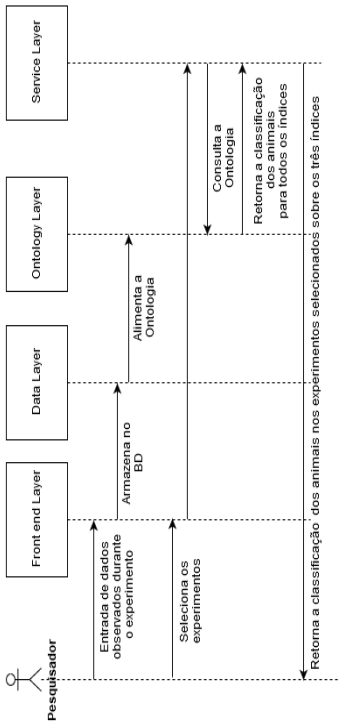

Figura 7. Diagrama de sequência para classificação dos animais utilizando a arquitetura.

Através da utilização da arquitetura, o pesquisador deixa de lado a utilização de inúmeras ferramentas de terceiros e passa a utilizar apenas a arquitetura. Essa tem o papel de centralizar todas as informações dos experimentos e permitir o acesso a todos os pesquisadores da instituição. Para o acesso às classificações, o pesquisador escolhe apenas o experimento ou os experimentos desejados e repassa à arquitetura, ela irá classificar os animais e trazer as análises dos dados

Na próxima seção, são apresentados alguns resultados gerados a partir do uso da arquitetura, considerando um conjunto de dados reais, utilizados em experimentos na Embrapa Gado de Leite.

5. Resultados e Discussões

Como prova de conceito da viabilidade da arquitetura, utilizou-se dados de experimentos relacionados com eficiência alimentar. Mais especificamente, dados do experimento fase 1 – Aleitamento com 30 dias, 56 dias e 80 dias, com animais entre 0 e 3 meses, conforme especificado na Tabela 1. O experimento foi conduzido com 37 novilhas F1 Girolando. Para isso, os serviços da *FeedEfficiencyService* foram utilizados a partir do *Front End Layer*. Os dados do experimento foram carregados, segundo o modelo de dados, e instanciados na ontologia. A partir do uso do *reasoner* e regras lógicas definidas, estes dados relativos a animais do experimento foram classificados

como: eficientes, intermediárias e ineficientes, considerando os índices de eficiência alimentar. A partir dos dados classificados, foi possível produzir saídas de acordo com as necessidades da pesquisa. Por exemplo, a figura 8 apresenta dados do experimento 1, sob o ponto de vista de cada um dos índices de eficiência alimentar. Isso se dá através da utilização visualizações interativas, que quando adotadas permitem a observação de um grande volume de dados. Outro ponto que deve ser destacado é a inclusão na visualização de uma paleta de três cores, sendo elas: verde para eficiente, amarelo para intermediário e vermelho para ineficiente. As cores foram utilizadas na classificação dos animais, bem como na coloração das arestas que relacionavam os animais aos experimentos, como pode ser visto na Figura 8.

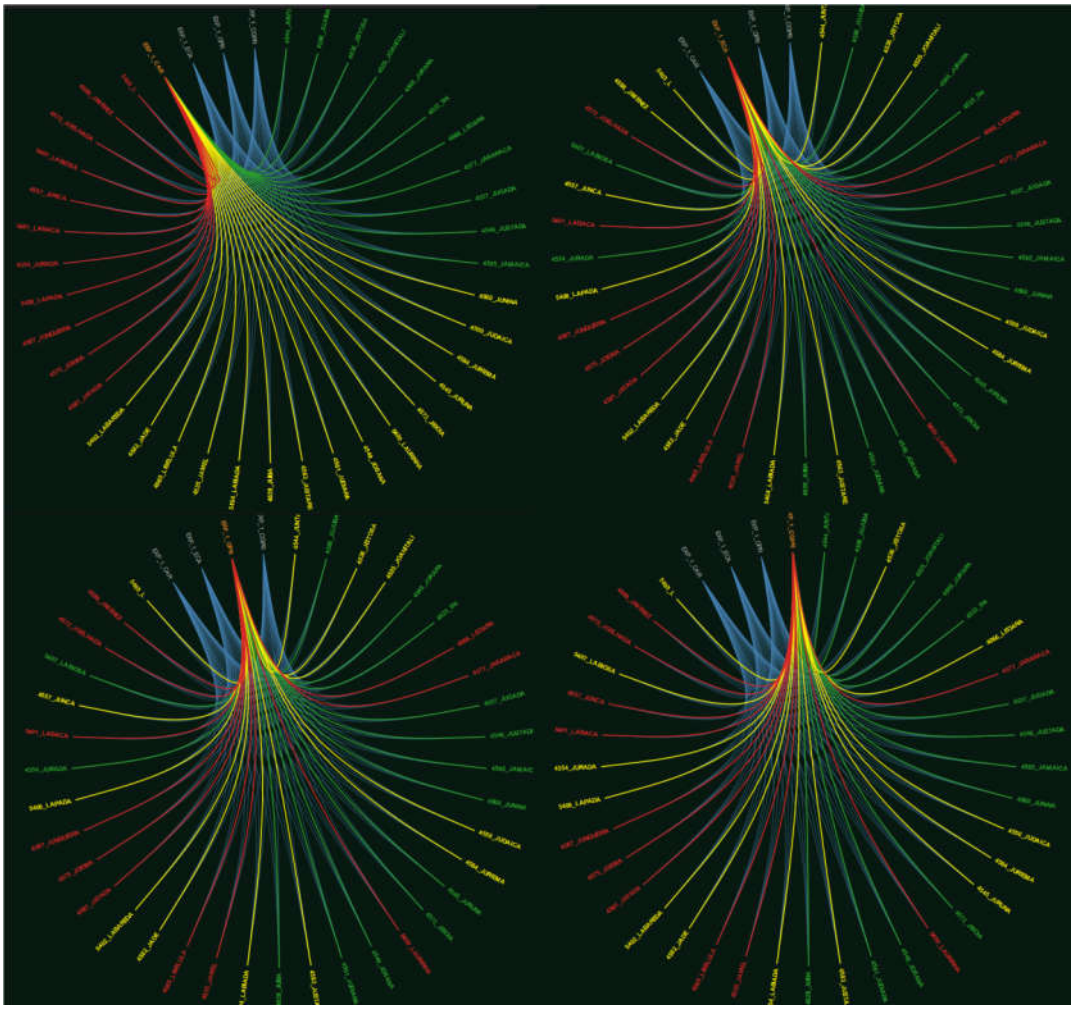

Figura 8. Visualização de classificação sob o ponto de vista do experimento 1 e dos índices de CAR, ECA, GPR e CGPR.

5 Conclusões

A grande competitividade do mercado atual exige que os produtores leiteiros busquem meios para reduzir seus custos e tornar seus rebanhos competitivos. Nesse sentido, o

apoio computacional vem fornecendo alternativas a identificação de animais mais eficientes e, por consequência, proporcionando ganhos econômicos e ambientais.

Observa-se, pelos autores citados, a utilização da ontologia para fins exclusivos de mapeamento. Vale frisar, até o presente momento, nas buscas realizadas em bases indexadas da área da computação, não houveram retornos similares a abordagem descrita. Dessa forma, o presente trabalho buscou apresentar uma arquitetura capaz de apoiar as pesquisas em eficiência alimentar, a partir do uso de ontologias e análise de dados, utilizando bases de experimentos coletados em campo.

A arquitetura visa promover maior poder de análise, de acesso aos experimentos, aumentando a confiabilidade e o reuso desses dados por outros pesquisadores. A partir de uma prova de conceito, pode-se observar indícios de que o uso de ontologias podem contribuir para o avanço das pesquisas em eficiência alimentar na Embrapa Gado de Leite. No entanto, experimentos formais devem ser conduzidos, de forma a comprovar estes indícios.

Como trabalho futuro, pretende-se desenvolver uma rede de ontologias, relacionando todos os experimentos, com o objetivo de observar a manutenção das classificações de eficiência nas fases de avaliação. Outro ponto a ser pesquisado é a busca por padrões nos dados brutos, através de algoritmos de classificação e agrupamentos.

Referências

Baker, Patricia G.. et al. An ontology for bioinformatics applications. Bioinformatics, v. 15, n. 6, p. 510-520, 1999.

Bechhofer, Sean. OWL: Web ontology language. In: Encyclopedia of Database Systems. Springer US p. 2008-2009 (2009)

Campos, Mariana .M., Leao, Juliana .M., Lima, Juliana .A.M., Machado, Fernanda .S. Tecnologias de precisão na avaliação de eficiência alimentar. Cadernos Técnicos de Veterinária e Zootecnia. n79 p. 73-85 (2015)

Gruber, Thomas R. et al. A translation approach to portable ontology specifications. Knowledge acquisition, v. 5, n. 2, p. 199-220 (1993)

Guarino, Nicola et al. Formal ontology and information systems. In: Proceedings of FOIS. p. 81-97 (1998)

Hitzler, Pascal; Krotzsch, Markus; Rudolph, Sebastian. Foundations of semantic web technologies. CRC Press (2009)

Miah, Shah J.; Gammack, John; Kerr, Don. Ontology development for context-sensitive decision support. In: Semantics, Knowledge and Grid, Third International Conference on. IEEE. p. 475-478 (2007).

Tomic, Dana, et al. "Experiences with Creating a Precision Dairy Farming Ontology (DFO) and a Knowledge Graph for the Data Integration Platform in agriOpenLink." Journal of Agricultural Informatics 6.4 (2015).

Verhoosel, Jack PC; Van Bekkum, Michael; Van Evert, Frits. Ontology matching for big data applications in the smart dairy farming domain. In: OM. 2015. p. 55-59.

A volunteer computing system implemented with peer-to-peer communication optimized for small and limited environments

Caio Santiago[1], **Luciano Antonio Digiampietri**[2]

[1]Bioinformática – Universidade de São Paulo (USP)
05.508-090 – São Paulo – SP – Brazil

[2]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
03.828-000 – São Paulo – SP – Brazil

{caio.santiago,digiampietri}@usp.br

Abstract

The computational needs of scientific experiments often require powerful computers. One alternative way to obtain this processing power is taking advantage of the idle processing of personal computers as volunteers. This technique is known as volunteer computing and has great potential in helping scientists. However, there are several issues which can reduce the efficiency of this approach when applied to complex scientific experiments, such as, the ones with long processing time, very large input or output data, etc. In order to face these challenges, we designed a volunteer computing system based on peer-to-peer communication. When compared with the local execution of activities and traditional volunteer computing, the execution time was improved and, in some cases, there was also a reduction of the server upload bandwidth use.

1. Introduction

The computational requirements of scientific experiments often demand powerful computers, which are usually expensive and, probably, they will be idle part of the time [Acharya et al. 1997]. On the other hand, the advance of the personal computers, with multi-core CPUs and GPUs, usually take care easily of the users' needs. Thus, there is a scenario with personal computers idle at part of the time and with scientific computers overloaded in specific moments, during the execution of scientific experiments.

Scientific experiments, in several cases, are organized as bag-of-tasks [Kwan and Jogesh 2010] or scientific workflows [Medeiros et al. 1996]. Bag-of-tasks are composed of a set of completely independent tasks, what is very different from workflows where a task needs to wait for the conclusion of another task. Both, typically, require huge computational power and a way to obtain it is the use of several personal computers, for example, desktop grids [Kondo et al. 2007, Anderson 2004] or volunteer computing [Anderson and Fedak 2006].

Volunteer computing (VC) projects take idle resources from donors: the tasks are sent to volunteers (in general using the Internet), and they send the results back to a server. This approach may provide a lot of computational power [Anderson 2004], but in scientific experiments, there are many issues which can turn this approach inefficient, such as long processing tasks [Dethier et al. 2008], great volumes of data to be transferred [Duan et al. 2012] or instability in volunteer computers [Dias et al. 2010]. The majority of these issues are related to the low-speed communication with donors across the Internet.

In order to solve these issues, some works proposed the use of peer-to-peer (P2P) concepts [Majithia et al. 2004, Zhao et al. 2009] in workflow execution using volunteer computing. This is a dynamic approach and it is able to deal, for example, with heterogeneous environments and faults. However, the majority of the research using P2P in this field aims to create huge and scalable networks. In this work we propose a different approach, dealing with small networks, composed of heterogeneous participants, and with bandwidth limited to speeds similar to the average speed on the Internet. However, it is worth highlighting that P2P communication makes more sense when the data are more frequently reused. Thus, this approach is preferable for experiments with dependent tasks, different from many approaches [Majithia et al. 2004, Gentzsch et al. 2013] in this area that work with bag-of-tasks (sets of independent tasks).

This paper is organized as follows. Section 2 presents related work. Section 3 describes the proposed solution. Section 4 contains the framework evaluation. The results are presented in Section 5. Section 6 presents the final remarks and future work.

2. Related Work

Some works are based only on a server (acting like the source of the tasks) and clients (the volunteers of the system), such as BOINC [Anderson 2004]. However, there are some systems that have others participants, they are not servers or volunteers, that are responsible for the communication among the participants.

Murata et al. 2008 present a work based on BOINC, where the volunteers are clustered in small groups (all of them are disjoint sets). The BOINC continues as the unique server, but each group is responsible to balance by itself. The same was proposed by Wen Dou et al. 2003, using instead of groups, random neighborhood relations.

The works that use intermediate participants in the communication can be divided into two main approaches: trees and directed acyclic graphs (DAG). In the tree-based approaches, the server is the root, the volunteers are, typically, the leaves, and the others are known as supertrees. The tasks travel from the root to the leaves, using decisions taken in each step. The decision could be based on probabilistic models [Kwan and Jogesh 2010] or other factors, such as reputation [Rius et al. 2012].

Mastroianni [Mastroianni et al. 2009] adopted a simpler strategy (from the scheduler algorithm point of view), but with more elements. The volunteers (workers) request tasks to the super-peers and the super-peers request them to the Data Cachers. The Data Cachers receive tasks from the Job Manager. The workers also have to request the input data directly from the Data Source, which receives data from the Job Manager. At last, workers send their results to Job Manager.

The works cited do not consider the data transfer costs because they deal with high-speed communication grids or small tasks. In the other cases, the transfer cost is very relevant.

3. Developed Solution

The development of this work was based on an extension [Digiampietri et al. 2014b] of the Workflow Management System (WfMS) called WOODSS (*A Spatial Decision Support System based on Workflows*) [Seffino et al. 1999], an open source system written in Java extended in this project to deal with P2P communication.

Based on the initial structure of the WfMS and on the extensions developed, a scheduling algorithm (Algorithm 1) that applies P2P techniques was specified and implemented. This algorithm runs on volunteer computers. The aim is to make the volunteers more proactive, i.e., they prepare themselves (downloading inputs) to the next tasks and send inputs to neighbors while they are processing a task. The main role of the server is to respond the volunteers' requests. Algorithm 1 works as follows. Each volunteer establishes a connection with the server in order to obtain information about others volunteers and tasks. After this, the volunteer requests a list of neighbors. With this information, the volunteer will send and receive data. Then the volunteer requests a list of "schematic objects" which are a simplified representation of the workflow that contains only tasks' relationships and inputs' identifications.

```
Connect to the server;
objs ← Request schematic objects;
Priority queue q ← Create priority queue with objs;
while q is not empty do
    Object obj ← Choose best task(q);
    Download inputs not downloaded yet of obj;
    Alert serve about the execution start of obj;
    if obj is able to execute then
        Task a ← Download task(obj);
        Start background thread;
        Execute a;
        Stop background thread;
        Send result;
        Mark result as a parameter not confirmed;
```

Algorithm 1: Scheduling algorithm that apply P2P techniques – Working in the volunteer computers

Then the algorithm decides what task will be processed and what task will be downloaded, with the criteria defined in the rules of the priority queue, based on: Select first tasks that do not depend on other not yet executed ones; the number of parameters; and the number of parameters already downloaded. When a volunteer chooses an input he asks the neighbors if they have it. If none of them have the desired input, it will be downloaded entirely from the server, in the other case, the data is downloaded from the server and the neighbors. Once there is a task ready, the processing is started and a background thread is also started. The background thread is similar to the process to choose a task, with the difference of do not download the task by itself, downloading only parameters of future tasks. This algorithm was designed specifically for small networks. For larger networks, a more complex algorithm with a more robust coordination strategy is necessary.

4. Evaluation

An infrastructure was implemented to measure the performance of the developed solution, which contains two test cases to evaluate three scheduling approaches. In order to do this, four computers were used. The computers' hardware and operating system specifications are presented in Table 1.

Table 1. Features of the computers used.

#	Processor	Clock	Memory	O.S.
1	Intel®Core i5-3230M	4x3.20 GHz	8 GB	Ubuntu 14.04
2	Intel®Core i3-350M	4x2.26 GHz	4 GB	Ubuntu 14.04
3	Intel®Pentium T3400	2x2.26 GHz	3 GB	Ubuntu 12.04
4	Intel®Core 2 Duo T5750	2x2.00 GHz	2 GB	Ubuntu 12.04

In order to simulate the real world connections, all bandwidths were limited via software. We used the average bandwidth connection, measured in Brazil (download at 2.4 Mbps) [Akamai 2013]. The upload bandwidth was considered as 10% of the download bandwidth.

All test cases were compared using two different systems (the traditional VC system and the proposed solution using P2P), based on the developed extension of WOODSS (with three different scheduling approaches). Both used the Computer 4 as the server and the others as volunteers. Also was tested the scenario with and without redundancy required, i.e., all results are confirmed for more than a volunteer to be accepted. The next subsection describes the two test cases used. Each case was performed three times and all presented results correspond to average values.

4.1. Test Case of Social Network Analysis

The first test case was planned to show the performance of the algorithm in a real area of knowledge. We chose a test case based on a workflow of social network analysis [Digiampietri et al. 2014a, Digiampietri et al. 2015]. This test case has approximately 20 high dependent tasks, alternating between light and heavy loads. The data size alternated too, between 20KB and 5MB.

4.2. Test Case of Toy Examples

We create another test case based on structures described by Bharathi et al. 2008. These examples allow a detailed analysis of the strengths and weaknesses of the proposed approach. Bharathi defined that there are four basic structures for workflows, and the combinations of these structures allow the creation of any complex workflows. The structures are: *Data Aggregation* (DA), *Data Distribution* (DD), *Data Redistribution* (DR), and *Pipeline* (P).

Each test case was limited to a single type of structure, with the same shape and the same number of tasks (four tasks DA and DD, seven tasks for DR and three for P), but independent from each other. The tests are composed of 10 structures (i.e., a workflow with ten copies of the same structure, but each copy was able to run independently from the others). The intention is to verify the impact when the volunteer has, or not, many tasks to be chosen.

The tasks from this case are all of them toy examples. All the tasks process the classic problem $3n + 1$to the interval from 1 to 5.10^8. Therefore, there will be a runtime variation in the execution of each individual task caused only by differences in the performance of the computers. Furthermore, the input and output data were defined with constant size (it is important to highlight the selected problem need only a number as input and a boolean as output, but, to serve the purposes of this evaluation, we introduced an input and output parameter with constant size).

5. Results

This section describes the results from each test case.

5.1. Test Case of Social Network Analysis

The results from the application of P2P technique in the Social Network Analysis test case were very positive. The developed solution (P2P) was faster than the others (up to 1,11 of speedup) and the time spent only transferring date was smaller than the traditional VC approach. Table 2 presents these results.

Table 2. Execution and transfer time from the test case of social network analysis

Execution		Run Time	Speedup	Transfer Time
Without Redundancy	VC	4:57:00	1.11	0:28:19
	P2P	4:27:45		0:19:48
With Redundancy	VC	8:33:30	1.04	1:05:28
	P2P	8:15:02		0:29:14

There are two main reasons to explain the improvement in the speedup. The first one is that the execution of the background thread did not increase runtime. It was observed that the volunteers have less variation in processing since the time between one task and the next was smaller than the traditional VC system. The second reason is the proactive behavior of the volunteers in preparing tasks to be processed before they are really necessary and, therefore, reducing the time expended exclusively with the inputs download. On average, the time spent with transfer corresponds to 9.5% (and 12.7% with redundancy) of the total execution time in the traditional volunteer computing system, and in the developed solution corresponds to 7.9% (and 5.9% with redundancy).

The volunteers get more parameters than they really need (Table 3). It happens because, when a volunteer is choosing the next parameter to be downloaded, it does not know yet if he will really execute the respective task. Between the moment of the download of the parameters and the execution, the task could be concluded by other volunteer or just could have enough volunteers processing the task.

Table 3. Data transferred by each volunteer in MBs on the social network analysis (*Download* - D.; *Upload* - U.)

Execution		Server		Volunt. 1		Volunt. 2		Volunt. 3	
		U.	D.	U.	D.	U.	D.	U.	D.
With Redundancy	VC	75.54	15.01	4.14	31.63	5.59	25.04	5.66	23.21
	P2P	115.74	17.92	6.30	38.29	6.74	37.70	5.00	38.82
Without Redundancy	VC	153.31	30.18	9.41	64.15	11.26	54.65	10.22	38.38
	P2P	99.39	31.33	7.12	23.71	14.76	42.62	10.17	29.85

5.2. Test Case of Toy Examples

In the cases with the toy examples structures (equals and independents), the results were more promising. This kind of test help to illustrate the behavior of the system imagining the execution of complex workflows.

The volunteer computing with P2P was faster than the traditional VC (Table 4). The improvements in execution time were significant, in both cases: with or without redundancy. The processing time of the two solutions proves that the P2P approach has a speed increase between 1.14 and 1.33 (and with redundancy between 1.22 and 1.33).

Table 4. Time of runtime/transfer of the test case with toy examples

Execution		Without Redundancy			With Redundancy		
		Run Time	Speedup	Transfer Time	Run Time	Speedup	Transfer Time
Data Aggregation	VC	3:09:50	1.33	0:57:55	7:17:01	1.23	1:18:55
	P2P	2:22:37		0:21:53	5:56:38		0:36:51
Data Distribution	VC	2:51:00	1.14	0:41:37	6:58:02	1.33	1:55:24
	P2P	2:29:13		0:20:04	5:13:22		0:37:58
Data Redistribution	VC	5:04:07	1.20	1:25:21	11:14:16	1.28	2:54:57
	P2P	4:12:43		0:31:49	8:44:02		1:01:52
Pipeline	VC	2:20:37	1.30	0:33:24	5:13:39	1.22	1:08:03
	P2P	1:48:28		0:15:09	4:15:22		0:28:53

The time spent exclusively with data transfers was reduced at least in the half in almost all cases. The volunteers (from the volunteer computing with P2P) expended most of the transfer time sending results, because the majority of the tasks did not spend extra time downloading parameters (they were downloaded in the background thread), but the upload of results is an activity without background threads (in order to optimize the total execution time).

Taking into consideration the transfer issues, the results were very positive, which is different from the ones achieved in the social network analysis cases. Table 5 shows the application of P2P obtained results better than the traditional volunteer computing, with or without redundancy. In some cases, the server did not download more data than in the traditional computing approach. It creates a situation in which the server downloaded more data than uploaded. Receiving more data than sending is typically considered a very positive thing in many real scenarios (for example, most of the Internet providers provide a much higher download rate than the upload one), favoring the proposed approach again.

The difference of this test case with the social network analysis one is due to the difference of output size of the tasks. In this case, the inputs and outputs have the same size, but in social network analysis case, the size of the first task input was bigger than the output of any task.

Table 5. Data transferred by each volunteer in MBs on test case with toy examples (*Download* - D.; *Upload* - U.)

Execution		Without Redundancy								With Redundancy							
		Server		Volunt. 1		Volunt. 2		Volunt. 3		Server		Volunt. 1		Volunt. 2		Volunt. 3	
		U.	D.	U.	D.	U.	D.	U.	D.	U.	D.	U.	D.	U.	D.	U.	D.
Data Aggregation	VC	120.68	82.38	37.27	60.17	25.66	30.57	20.74	34.10	243.28	167.59	75.72	113.37	49.73	70.50	43.52	66.35
	P2P	75.92	85.04	40.20	18.73	22.98	26.14	16.49	26.22	80.29	165.71	65.30	20.80	44.60	23.73	44.99	31.59
Data Distribution	VC	84.64	84.28	37.48	37.57	25.72	26.63	18.37	19.14	168.12	165.18	76.21	75.33	46.86	47.34	41.01	44.19
	P2P	33.97	83.37	38.37	11.16	25.41	13.00	18.31	8.66	35.77	164.07	69.75	10.32	46.21	11.90	42.00	11.75
Data Redistribution	VC	185.21	147.01	62.49	92.37	44.27	43.99	36.20	45.41	372.88	293.27	132.15	168.38	81.50	98.83	71.88	101.33
	P2P	71.02	143.72	63.14	19.98	42.48	24.51	34.86	27.28	86.46	288.25	127.12	28.60	80.16	23.84	70.74	31.54
Pipeline	VC	61.79	61.29	26.85	26.75	18.91	19.41	14.21	15.25	121.99	124.28	55.21	54.73	33.55	33.93	32.48	34.27
	P2P	28.13	59.52	30.32	8.36	14.46	8.57	14.35	10.51	26.21	123.21	58.20	7.97	35.74	8.24	26.36	9.40

6. Final Remarks and Future Work

In this work, we presented an alternative solution to the problem of executing scientific workflows in a distributed way, in small networks. It combines the use of volunteer computing with peer-to-peer techniques. The current approach is not designed to deal with big networks because the communication process is not scalable. The comparison of the proposed approach with local execution of tasks and the traditional volunteer computing

showed promising results. The developed solution was tested with two cases: the first is a real case of social network analysis and the second is a test case with toy examples (created considering the basic workflow structures). In both cases, the developed solution obtained positive results to reduce the total run time and the bandwidth needs of the server (except in one specific case). The results are more expressive when considering the scenarios which require redundancy of task execution. Among the main contributions, we highlight the algorithm presented in Section 3. It corresponds to improvements in the execution time of scientific experiments for small networks and with limited bandwidth.Another contribution of this approach is that a volunteer can help the execution of the workflows not only running tasks but also helping in the sharing of input data.

As future work, we intend to improve the volunteer communication. We also intend to make the scheduling algorithm able to work with massive quantities of volunteers. Finally, we will improve the algorithm with a robust fault tolerance mechanism.

Acknowledgment

The work presented in this paper was partially funded by CAPES, CNPq, and FAPESP.

References

Acharya, A., Edjlali, G., and Saltz, J. (1997). The utility of exploiting idle workstations for parallel computation. *ACM SIGMETRICS Performance Evaluation Review*, 25(1):225–234.

Akamai (2013). The state of the internet. Technical Report Vol 6, Num 2, Akamai Faster Forward.

Anderson, D. (2004). BOINC: a system for public-resource computing and storage. In *Fifth IEEE/ACM International Workshop on Grid Computing*, pages 4–10.

Anderson, D. P. and Fedak, G. (2006). The Computational and Storage Potential of Volunteer Computing. In *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, volume 1, pages 73–80.

Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M.-H., and Vahi, K. (2008). Characterization of scientific workflows. In *2008 Third Workshop on Workflows in Support of Large-Scale Science*, pages 1–10. IEEE.

Dethier, G., Briquet, C., Marchot, P., and de Marneffe, P.-A. (2008). LBG-SQUARE Fault-Tolerant, Locality-Aware Co-Allocation in P2P Grids. In *2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 252–258. IEEE.

Dias, J., Ogasawara, E., de Oliveira, D., Pacitti, E., and Mattoso, M. (2010). Improving Many-Task computing in scientific workflows using P2P techniques. In *2010 3rd Workshop on Many-Task Computing on Grids and Supercomputers*, pages 1–10. IEEE.

Digiampietri, L., Alves, C., Trucolo, C., and Oliveira, R. (2014a). Análise da rede dos doutores que atuam em computação no brasil. In *CSBC-BraSNAM 2014*.

Digiampietri, L., de Jsus Prez-Alczar, J., Santiago, C., Oliveira, G., Khouri, A., and Arajo, J. (2014b). A framework for automatic composition of scientific experiments: Achievements, lessons learned and challenges. In *VIII Brazilian e-Science Workshop*.

Digiampietri, L. A., Maruyama, W. T., Santiago, C. R. N., and da Silva Lima, J. J. (2015). Um Sistema de Predio de Relacionamentos em Redes Sociais. In *Simpsio Brasileiro de Sistemas de Informao (SBSI 2015)*, pages 139–146.

Duan, K., Padget, J., Kim, H. A., and Hosobe, H. (2012). Composition of engineering web services with universal distributed data-flows framework based on ROA. In *Proceedings of the Third International Workshop on RESTful Design - WS-REST '12*, page 41, New York, New York, USA. ACM Press.

Gentzsch, W., Grandinetti, L., Joubert, G., Ricci, L., Baraglia, R., Ghafarian, T., Deldari, H., Javadi, B., Yaghmaee, M. H., and Buyya, R. (2013). CycloidGrid: A proximity-aware P2P-based resource discovery architecture in volunteer computing systems. *Future Generation Computer Systems*, 29(6):1583–1595.

Kondo, D., Fedak, G., Cappello, F., Chien, A. A., and Casanova, H. (2007). Characterizing resource availability in enterprise desktop grids. *Future Generation Computer Systems*, 23(7):888–903.

Kwan, S. K. and Jogesh, K. M. (2010). Bag-of-Tasks applications scheduling on volunteer desktop grids with adaptive information dissemination. In *IEEE Local Computer Network Conference*, pages 544–551. IEEE.

Majithia, S., Shields, M., Taylor, I., and Wang, I. (2004). Triana: a graphical Web service composition and execution toolkit. In *Proceedings. IEEE International Conference on Web Services, 2004.*, pages 514–521. IEEE.

Mastroianni, C., Cozza, P., Talia, D., Kelley, I., and Taylor, I. (2009). A scalable super-peer approach for public scientific computation. *Future Generation Computer Systems*, 25(3):213–223.

Medeiros, J. W., Weske, M., Vossen, G., and Bauzer, C. (1996). Scientific workflow systems. *NSF Workshop on Workflow and Process Automation: State-of-the-art and Future Directions*.

Murata, Y., Inaba, T., Takizawa, H., and Kobayashi, H. (2008). Implementation and evaluation of a distributed and cooperative load-balancing mechanism for dependable volunteer computing. In *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, pages 316–325. IEEE.

Rius, J., Estrada, S., Cores, F., and Solsona, F. (2012). Incentive mechanism for scheduling jobs in a peer-to-peer computing system. *Simulation Modelling Practice and Theory*, 25:36–55.

Seffino, L., Medeiros, C., Rocha, J., and Yi, B. (1999). WOODSS - A Spatial Decision Support System based on Workflows. *Decision Support Systems*, 27(1-2):105–123.

Wen Dou, Yan Jia, Huai Ming Wang, Wen Qiang Song, and Peng Zou (2003). A P2P approach for global computing. In *Proceedings International Parallel and Distributed Processing Symposium*, page 6. IEEE Comput. Soc.

Zhao, Z., Yang, F., and Xu, Y. (2009). PPVC: A P2P volunteer computing system. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 51–55. IEEE.

Hydric-Agent: Ferramenta de Simulação Baseada em Agentes para Gestão da Água em Áreas Residenciais

Fernando L. Alencar[1], Diana J. Monsalve-Herrera[2], Carolina G. Abreu[1], Cassio G. C. Coelho[1], Conceição de Maria A. Alves[2], Célia G. Ralha[1]

[1]Programa de Pós-graduação em Informática, Departamento de Ciência da Computação, Universidade de Brasília (UnB), DF, Brasil

[2]Programa de Pós-graduação em Tecnologia Ambiental e Recursos Hídricos, Departamento de Engenharia Civil e Ambiental, UnB , DF, Brasil

{carolabreu,calves,ghedini}@unb.br

Abstract. *In the implementation of water management plans, it is necessary to understand the behavior of water consuming agents in relation to decisions, plans and projects of water system management. This integration allows decision making by managers in accordance with the reality of the water system. In the present work we present the Hydric-Agent tool developed in JADEX that allows the simulation of water consumers under different water management scenarios, providing the user of the platform with a view of the cooperativism of the water consumer agent and the degree of adhesion to the action of management.*

Resumo. *Na implementação de planos de gerenciamento hídrico, se faz necessário compreender o comportamento dos agentes consumidores de água diante de decisões, planos e projetos de gerenciamento do sistema hídrico. Essa integração possibilita tomadas de decisão por parte dos gestores em concordância com a realidade do sistema hídrico. No presente trabalho se apresenta a ferramenta Hydric-Agent desenvolvida em JADEX que permite a simulação de consumidores de água sob diferentes cenários de gestão hídrica, fornecendo ao usuário da plataforma uma visão do cooperativismo do agente consumidor de água e o grau de adesão à ação de gestão estabelecida.*

1. Introdução

A gestão de sistemas hídricos urbanos se desenvolve com ênfase em processos hidrológicos, econômicos, políticos e sociais. A complexidade das leis que governam esses processos e as ações que resultam das interações entre eles proporcionam propriedades aos sistemas hídricos urbanos que permitem sua caracterização como sistemas complexos adaptativos [Mitchell 2009, Sichman 2015]. A operação de um sistema hídrico urbano envolve a participação de atores como consumidores de água, companhias de saneamento básico e agências reguladoras desse serviço público. A dinâmica e a possibilidade de aprendizado, de evolução e de adaptação de atores envolvidos no funcionamento desses sistemas resultam em comportamentos não determinísticos que caracterizam sistemas complexos. Considerar o comportamento desses atores na simulação de sistemas hídricos urbanos pode contribuir para definição de programas (medidas e ações) de gestão mais efetivos e adaptados à realidade local [Giacomoni et al. 2013, Kanta and Zechman 2013].

Os sistemas hídricos abordados como sistemas complexos adaptativos(SCA) podem ser simulados por meio de modelos baseados em agentes (MBA) [Holland 1995], uma vez que esses são considerados como uma metáfora natural para a representação de sistemas complexos. Os MBAs, por sua vez, podem ser representados por meio de Sistemas Multiagentes (SMA), área de Inteligência Artificial Distribuída. SMA permite a implementação de simulações com ênfase nas ações e interações de agentes em um ambiente computacional.

Um dos usos de SMA é representar o raciocínio e o conhecimento de agentes heterogêneos que interagem uns com os outros, cooperativamente, em busca de um objetivo global. O Hydric Agent é uma ferramenta de SMA, baseada na simulação de modelos baseados em agentes (MBA), que visa a auxiliar o gestor do setor de recursos hídricos a selecionar ações e medidas de gestão de sistema hídrico de acordo com o perfil comportamental da comunidade ou de consumidores de água. Por meio da aplicação do Hydric Agent é possível integrar ou inserir o comportamento do usuário em modelos de sistemas hídricos que geralmente representam apenas processos hidráulicos e hidrológicos sem levar em consideração a influência do componente social no comportamento desses sistemas. O objetivo deste trabalho é apresentar a ferramenta Hydric Agent e seu processo de construção e implementação. Complementarmente, serão ilustradas algumas possibilidades de análises que resultaram de sua aplicação a um estudo de caso na comunidade de Brazlândia, Distrito Federal.

2. Hydric-Agent

Nessa sessão serão apresentados alguns fundamentos e métodos que foram utilizados na construção da proposta de ferramenta do Hydric-Agent. Uma detalhada pesquisa de campo foi realizada para que os agentes representassem de forma mais realista o comportamento de cidadãos da região de estudo. Por fim, os detalhes da modelagem conceitual e arquitetural do Hydric-Agent são apresentados.

2.1. Conceitos

Sob o ponto de vista da computação, um agente é uma entidade autônoma inteligente dotada de sensores e de atuadores, que a permitem acessar o ambiente em que se encontra e atuar sobre ele. Outras características inerentes aos agentes, como proatividade, mobilidade e comunicação proporcionam a eles a capacidade de adaptação a mudanças ambientais, considerando não apenas as percepções do meio em que se encontram, mas também objetivos próprios que eles desejam alcançar [Russel and Norvig 2010]. O Projeto de construção de um agente pode ser descrito como PAGE (*Perceptions, Actions, Goals e Environment*). O agente percebe e interage com o ambiente. O nível de complexidade de um SMA é dado principalmente pela variedade e completude do ambiente.

A cognição dos agentes pode ser projetada de diferentes meios. Uma das opções é o modelo mentalista baseado em Crença-Desejo-Intenção ou *Belief-Desire-Intention* (BDI). O modelo é composto por uma arquitetura deliberativa no qual o estado interno de um agente pode ser descrito por conjuntos de estados mentais [Bratman 1987]. As crenças representam aquilo que o agente pode perceber e registrar do ambiente além de percepções internas a seu próprio respeito. Os desejos são o conjunto de atitudes mentais que motivam o planejamento do agente a realizar suas metas. As intenções definem os planos que serão executados para a realização de um objetivo [Wooldridge 2009].

Os planos são conjuntos ordenados de ações resultantes de um processo de raciocínio prático constituído de duas etapas: deliberação e planejamento (ou raciocínio meios-fim). Na etapa de deliberação, o agente acessa os estados atuais de suas crenças e de outras variáveis internas, como mensagens, para decidir qual o objetivo deve ser admitido como corrente no momento. Na etapa subsequente, planejamento, o agente toma o resultado da etapa anterior para analisar que ações são possíveis de serem tomadas em uma ordem lógica para alcançar o objetivo atual [Wooldridge 2009].

2.2. Levantamento de Dados

Para viabilizar a definição do comportamento dos agentes no simulador Hydric-Agent, uma pesquisa de campo foi conduzida. O levantamento das crenças e atitudes de cidadãos reais foi realizada a partir de 320 questionários aplicados na região de interesse [Monsalve-Herrera 2018]. Esse material permitiu a definição do modelo conceitual dos agentes, com base no framework $i*$ [Yu 1995]. A caracterização das percepções, ações, objetivos e o ambiente foram extraídas dos questionários, com o auxílio de especialistas, resultando na configuração do comportamento dos agentes BDI-cognitivos.

Os agentes foram considerados como domicílios consumidores de água, que podem comportar-se de forma cooperativa ou não cooperativa. Os domicílios foram categorizados em três classes de rendas: renda alta (mais que 10 salários mínimos); renda média (2 a 4 salários mínimos); e renda baixa (menos que 2 salários mínimos). Cada domicílio também foi caracterizado conforme a escolaridade: nível fundamental, médio e superior completos e incompletos.

Agentes não cooperativos caracterizam-se por não se adaptar às regras de consumo de água estabelecidas pelo gestores e não diminuirem a quantidade de água consumida, podendo ainda, aumentar seu consumo de água. Já os agentes do tipo cooperativo adaptam-se às regras de consumo de água estabelecidas pelo gestor e diminuem a quantidade de água consumida como resposta a medidas e ações de gestão impostas pelo gestor. As possíveis medidas de gestão de demanda de água implantadas são: campanhas educativas e implantação de tarifa de contingência.

2.3. Arquitetura e Implementação

A partir do levantamento de dados e definição do modelo conceitual foi construída a ferramenta Hydric-Agent[1] utilizando como *middleware* o framework JADEX [Braubach and Pokahr 2012] e a linguagem de programação Java. Foi proposta uma arquitetura em três camadas com hierarquia híbrida, ou seja, as mudanças de comportamento podem ser acionadas diretamente pela alteração de uma medida do gestor (hierarquia *top-down*) ou pela tomada de decisão individual de cada agente consumidor (*bottom-up*). Os requisitos do funcionamento da ferramenta Hydric-Agent também são derivados do modelo conceitual e a implementação baseou-se na modelagem do SMA por meio da metodologia Tropos [Bresciani et al. 2004]. A visão geral pode ser observada na Figura 1. Percebe-se a definição de um agente gestor e de agentes consumidores, de diferentes tipologias. Cada tipologia determina como os agentes consumidores irão interagir com o recurso água conforme as suas crenças (renda, taxa de escolaridade, etc). A implementação dos agentes segue o conceito PAGE, conforme apresentado na Tabela 1.

[1]Código-fonte disponível em: `https://github.com/MASE-UnB/Hydric-Agents-BDI`

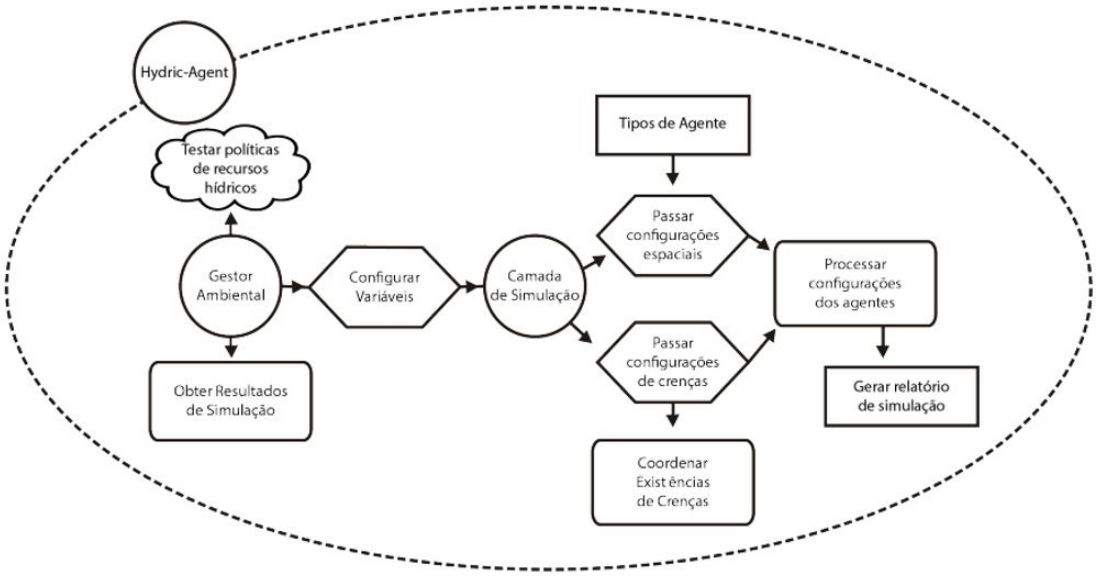

Figura 1. Diagrama Tropos que representa o Projeto Arquitetural Hydric-Agent

Tabela 1. PAGE utilizado para desenvolvimento da plataforma Hydric-Agent

Percepções	Campanhas educativas Tarifa Estação de chuva ou seca
Ações	Economia de Água Desperdício de Água
Objetivos	Satisfazer as necessidades mensais de uso de água
Ambiente	Área residencial urbana Grid de simulação

A Tabela 2 apresenta a classificação do ambiente real esperado de uma área residencial urbana em contraste com o ambiente computacional modelado na ferramenta Hydric-Agent. Esse ambiente modelado é o que será percebido pelos agentes computacionais que possuem crenças baseadas no estudo de campo e apresentam mecanismo deliberativo-cognitivo construído conforme o modelo BDI.

No Hydric-Agent foram construídas três camadas: interface, controle e física. A camada de interface fornece ao usuário a visualização do cenário em forma de grid e as opções de configurações do sistema (Figura 2). Por meio dessa camada o usuário pode configurar o tamanho do grid (*Configure Grid*), adicionar ou excluir agentes (através de cliques nas células do grid), alterar o ambiente com a inserção ou remoção das crenças (*Toggle Tax e Toggle Education*), controlar os steps da simulação (*Next Step*), iniciar e pausar a execução do software (*Start/Stop*) e gerar o relatório da simulação (*Generate Report*). Na camada de controle, as configurações definidas através da Camada de Interface são utilizadas para execução da simulação. A organização e composição hierárquica dos agentes, assim como os mecanismos de comunicação deles e seus respectivos PAGE são descritas nessa camada. Na camada física atuam os agentes consumidores de água o

modelo de raciocínio BDI específico para cada agente. Essa camada é a responsável por gerar todos os dados de consumo dos agentes do Sistema multiagente e enviá-los para a camada de controle para que possam ser transcritos em formato de relatório. A ferramenta produz dados de consumo de água dos agentes e lista o número de agentes cooperativos por renda e por escolaridade. Esses dados podem ser utilizados como cenário inicial de ferramentas de suporte à decisão de recursos hídricos, oferecendo uma visão holística do sistema a ser avaliado pelo gestor da água.

Tabela 2. Classificação do Ambiente

Real	Modelado
Parcialmente observável	Parcialmente observável
Estocástico	Determinístico
Sequencial	Episódico
Dinâmico	Estático
Contínuo	Discreto
Multiagente	Multiagente
Híbrido	Híbrido

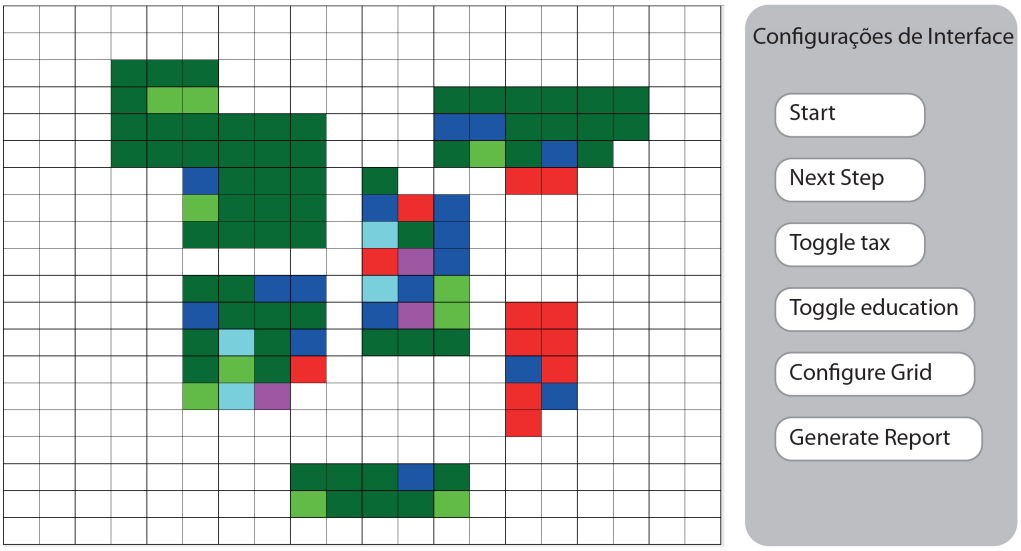

Figura 2. Interface e Grid de Simulação Hydric-Agent

A simulação foi configurada para que cada passo represente um mês de consumo de água, uma vez que há diferença nas políticas aplicadas durante os meses secos e chuvosos. Os agentes são distribuídos pelo grid sendo que suas cores e tonalidade mostram o cooperativismo de cada agente por tipo de renda como apresentado na Figura 3. Nessa versão do Hydric-Agent, considerou-se não haver influência de vizinhança nos padrões de consumo de água residencial urbana, ou seja, residentes não tomam decisão de consumir por influência de vizinhos.

3. Estudo de Caso: Consumo de Água Residencial Urbano

A presente situação de crise hídrica no DF resultou na adoção de medidas de racionamento de água para a maioria das regiões administrativas. Os níveis dos principais reservatórios de abastecimento da região, Descoberto e Santa Maria, atingiram em 2017 os

menores índices do registro histórico. Como estudo de caso foi escolhida a área urbana da Região Administrativa IV do DF, Brazlândia. A represa do Rio Descoberto e a formação do Lago Descoberto, que antes faziam parte de fazendas da região de Brazlândia, hoje são responsáveis pelo abastecimento de mais de 60% da água de todo o DF. A Figura 4 apresenta a visão geral da área urbana, utilizada como guia para construção do grid de simulação Hydric-Agent.

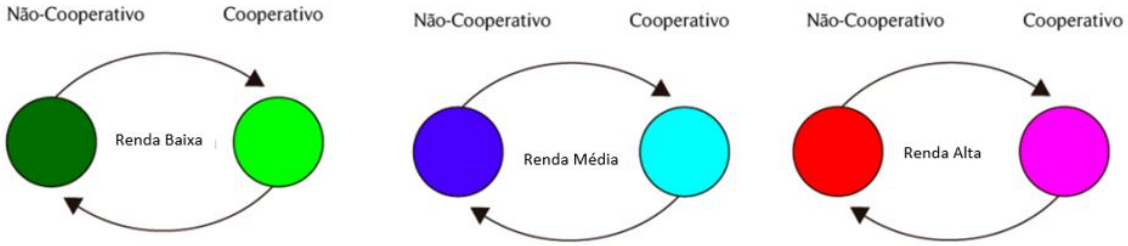

Figura 3. Representação de categoria e comportamento dos agentes

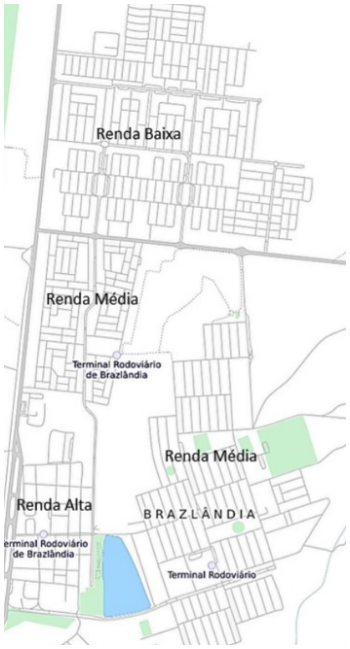

Figura 4. Área residencial urbana da Região Administrativa de Brazlândia, DF

A simulação do Hydric-Agent em Brazlândia permite avaliar a adesão dos consumidores da região às ações da gestão por meio da representação do comportamento cooperativo ou não cooperativo. A área urbana e a mudança de comportamento dos agentes pode ser visualizada quando o usuário acompanha as células do grid, que correspondem às residências, e as cores, que correspondem o tipo de renda e o perfil de cooperação (Figura 3).

3.1. Experimentos e Resultados

As simulações e relatórios produzidos pelo Hydric-Agent permitem observar o comportamento cooperativo de agentes classificados por renda e por escolaridade mensalmente, bem como os respectivos consumos de água. Um exemplo de relatório do Hydric-Agent

está apresentado na Figura 5 que ilustra a porcentagem de agentes de renda baixa (de maior representatividade em Brazlândia), por escolaridade, que apresentaram ação cooperativa após campanhas educativas ao longo do ano (linhas coloridas). Os resultados podem ser comparados com a porcentagem de agentes de renda baixa, por escolaridade, que apresentaram ação cooperativa quando não havia campanhas educativas (sem gestão-SG) (barras coloridas).

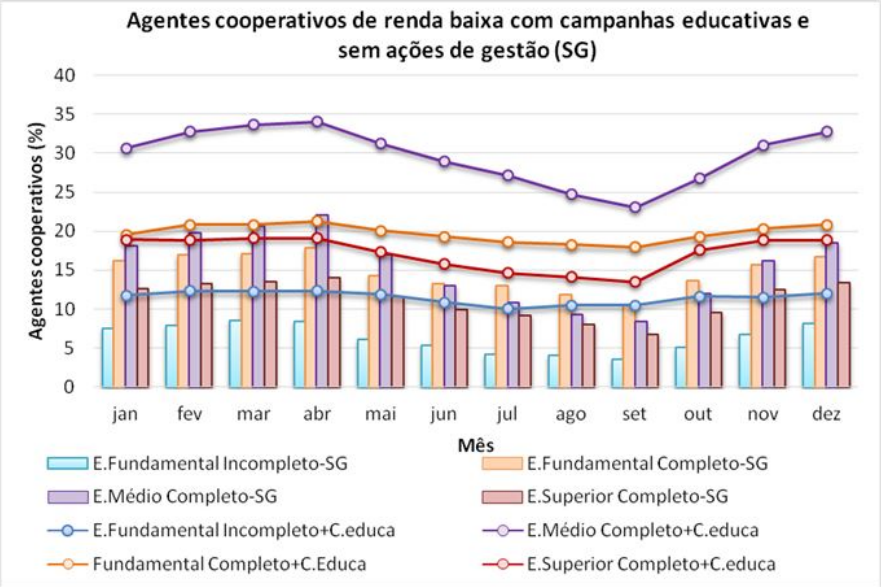

Figura 5. Porcentagem de agentes cooperativos de renda baixa sem ações de gestão (SG) e implementando campanhas educativas simulado em Hydric-Agent

A Figura 5 mostra que consumidores de renda baixa são receptivos a campanhas educativas pois as porcentagens de agentes cooperativos são maiores para todas escolaridades ao longo do ano, quando comparadas com porcentagens da simulação sem ação de gestão (SG). A observação da variação do cooperativismo ao longo do ano permite também avaliar a percepção do consumidor em relação à disponibilidade de água (representada pela ocorrência do período de chuva, outubro a abril). No período de estiagem, maio a setembro, em média há redução de cooperativismo em todas as categorias de renda, mas para a renda baixa essa redução não é acentuada devido a característica de domicílios dessa categoria, por exemplo, a inexistência de grandes áreas de jardins que requeiram irrigação. A funcionalidade do Hydric-Agent também auxilia no entendimento da influência da escolaridade na efetividade de diferentes ações de gestão de água. Análises similares podem ser feitas para as demais categorias de renda, que não foram aqui apresentadas por restrição de espaço. A ferramenta Hydric Agent mostrou-se útil para apoio à seleção de medidas e ações de gestão da água permitindo adaptar as políticas à realidade local.

4. Conclusão

Com a finalidade de propor uma ferramenta que permita auxiliar o gestor recursos hídricos na tomada de decisões relacionadas à gestão de recursos hídricos, o Hydric-Agent propicia uma melhor visualização dos cenários de uso de água urbano. A ferramenta considera

o comportamento do usuário de recursos hídricos para avaliar diferentes estratégias de gestão hídrica adaptativa. As tecnologias utilizadas para implementação foram adequadas por prover bibliotecas e interfaces que facilitam o processo de comunicação entres as entidades do programa, possibilitando uma representação explícita do ambiente e dos objetivos que os agentes podem alcançar. Os relatórios gerados a partir das simulações dos agentes, permitiram a plotagem de gráficos e a análise do comportamento do usuários de recursos hídricos do cenário de estudo. A aplicação Hydric-Agent foi capaz de provar-se dinâmica e útil para a utilização por gestores ambientais ou pesquisadores de áreas interdisciplinares. É ferramenta de apoio à tomada de decisão capaz de promover o entendimento da dinâmica do sistema hídrico e de seus agentes, ao criar um ambiente computacional capaz de simular as necessidades ambientais, econômicas e sociais.

Referências

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. Harvard University Press, Cambidge, MA.

Braubach, L. and Pokahr, A. (2012). Jadex active components framework-bdi agents for disaster rescue coordination. *Software agents, agent systems and their applications*, 32:57–84.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and MultiAgent Systems*, 8(3):203–236.

Giacomoni, M., Kanta, L., and Zechman, E. (2013). Complex Adaptive Systems Approach to Simulate the Sustainability of Water Resources and Urbanization. *Journal of Water Resources Planning and Management*, 139(June):554–564.

Holland, J. H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Kanta, L. and Zechman, E. (2013). Complex Adaptive Systems Framework to Assess Supply-Side and Demand-Side Management for Urban Water Resources. *Journal of Water Resources Planning and Management*, 140(January):75–85.

Mitchell, M. (2009). *Complexity: A Guided Tour*. Oxford University Press, USA.

Monsalve-Herrera, D. J. (2018). *Modelo comportamental com base em agentes para gestão adaptativa de água: caso de estudo de consumo de água residencial urbana em Brazlândia/DF*. PhD thesis, Departamento de Eng. Civil e Ambiental, UnB.

Russel, S. J. and Norvig, P. (2010). *Artificial Intelligence: a modern approach*. Prentice Hall, 2nd edition.

Sichman, J. S. (2015). Operacionalizacao de sistemas complexos. In Furtado, B. A., Sakowski, P., and Tovolli, M., editors, *Modelagem de sistemas complexos para políticas públicas*, page 436. Instituto de Pesquisa Economica Aplicada - IPEA.

Wooldridge, M. (2009). *Introduction to Multiagent Systems*. John Wiley & Sons Ltd, West Sussex, UK, 2nd edition.

Yu, E. S.-K. (1995). *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, Department of Computer Science, University of Toronto.

Functional Requirements for Developing ERAS – A Portal to Support Collaborative Geomechanical Simulations*

Maria Julia Lima, Melissa Lemos, Fernanda Pereira, Rodnei Couto, Deane Roehl

Tecgraf/ PUC-Rio Institute
PO Box 38097 – Rio de Janeiro – RJ – Brazil

{mjulia,melissa,nandalgp,rodnei,deane}@tecgraf.puc-rio.br

Abstract. *One of the most important tasks in geomechanical research is executing analytical and numerical simulations to understand geomechanical phenomena. In order to attain this objective, researchers have to prepare data to perform the simulations, build the models that define the appropriate physical representation and the mathematical modeling of the problem, run a computer system capable of simulating the phenomenon, and visualize and interpret the results. This paper presents the main functional requirements to support the development of solutions that encompass the simulation of geomechanical problems, taking into account a collaborative environment with access to an efficient computer infrastructure. The paper also describes ERAS, a portal developed according to these requirements, highlighting the advantages it brings to researchers in this area.*

Resumo. *Uma das tarefas mais importantes na pesquisa de geomecânica é a execução de simulação numérica e analítica para compreensão dos fenômenos geomecânicos. Com este objetivo, os pesquisadores preparam os dados para realizar as simulações do problema em questão; constroem o modelo que define a representação física apropriada e a modelagem matemática do problema; executam a computação capaz de simular o fenômeno estudado; e visualizam e interpretam os resultados. Este artigo apresenta os principais requisitos funcionais para apoio ao desenvolvimento de soluções que envolvam simulação de problemas geomecânicos, considerando um ambiente colaborativo com acesso a uma infraestrutura computacional eficiente. O artigo também apresenta o ERAS, um portal que está sendo desenvolvido de acordo com estes requisitos, destacando as vantagens que ele traz aos pesquisadores da área.*

1. Introduction

The application of analytical and numerical simulations to geomechanical problems is strategic to the Oil and Gas Industry. Research in this area provides a fundamental comprehension of subsoil geomechanical phenomena, which can optimize the industry's expenditure, while reducing geomechanical risks associated to drilling, completion, and development plan. Reservoir simulators, for example, have a fundamental role in the planning and optimization of new production field developments. Operating a deep-water well entails high costs, and thus minimizing the perforation risks of a dry well or

* This work received funds from Shell Brazil and ANP "Commitment to Investments in Research and Development", in the scope of the "BG- 47 Coupled Geomechanical Modeling" research project.

decreasing the number of wells necessary for the development of a new field can result in substantial savings.

Overall, during a project, the researchers' work in understanding subsoil geomechanical problems entails: (a) collecting, processing and interpreting data; (b) modeling, which defines the appropriate mathematical description of the problem; (c) processing, which executes the computer system capable of simulating the model; (d) analysis and visualization (post-processing), in which results are verified and studies are conducted; and finally (e) report generation and result dissemination. We note that this workflow is not always performed sequentially. In some cases, during analysis and visualization, the researchers may decide to go back to previous stages to prepare new data, revise the model or execute the simulation with new parameters, for example.

The teams involved in these projects are usually multidisciplinary, with professionals working in collaboration in different knowledge areas. In many cases, the simulations demand high computational processing power, producing a huge volume of data. Without the support of tools and an adequate computational infrastructure, researchers may spend a lot of energy and time collecting, organizing, processing and visualizing this data. Furthermore, the experience and the results obtained in those projects are a valuable knowledge source for future project enquiries and decision-making. Thus, the need for organizing, sharing, and reusing data, models, simulators, results, processes and knowledge is of great importance.

The importance of collaboration among researchers has led the development of social environments that support the whole lifecycle of experiments in scientific scenarios. E-ScienceNet [Classe at al. 2017] aims to provide support for geographically distributed researchers in the processes of creating, implementing and sharing experiments. myExperiment [Goble and De Roure, 2007] is an online environment that creates a social network of scientists who collaborate and share Taverna workflows [Wolstencroft et al. 2013]. The number of registered users and workflows available in myExperiment shows the potential of this kind of environment for scientific communities. An alternative is CrowdLabs [Mates at al. 2011], which adopts the model of social web sites to provide another rich collaborative environment for scientists.

This paper presents the main functional requirements to support research projects that involve simulation of geomechanical problems, taking into account a collaborative environment with access to an efficient computational infrastructure. These requirements stem from many years of experience of researchers at Institute Tecgraf/PUC-Rio working in this area. Furthermore, the paper describes ERAS, a portal developed by Tecgraf/PUC-Rio Institute, based on these requirements, as part of a research program in Geomechanics, in which Cambridge University and Berkeley University are also partners, as well as an O&G company.

The paper is organized as follows. Section 2 presents the functional requirements to support the development of solutions that involve simulations of geomechanical problems. Section 3 describes the main functionalities in ERAS Portal. Section 4 shows the services available in ERAS Portal to develop applications that use geomechanical simulations. We present an application for slope stability problems as a use case of the Portal. Finally, Section 5 presents comparisons with available related systems, final comments and future work.

2. Functional Requirements

2.1. Collaborative Project

Research that applies simulation in geomechanical problems is usually conducted by multidisciplinary teams, with professionals working collaboratively in different knowledge areas such as physics, engineering, mathematics, computer science, geology and geophysics. Therefore, it is crucial to have a work environment that promotes the exchange of knowledge and experience among researchers during an on-going project, as well in future projects.

In a geomechanical collaborative environment, researchers not only share data and simulators but also all the stages of the work in progress. In this context, data provenance is essential as it registers input data, models, simulators and results, providing a historical record of the data and its origins. This scenario allows for the reproducibility of results, which can be used to validate simulation results and improve future researchers.

2.2. Collaborative Simulation

The design and development of simulators is shifting towards collaborative paradigm. Therefore, simulation environments for such a paradigm need to take into account the cooperation between design teams in a distributed environment. Supporting the creation of a simulation can happen at different levels, all of which can benefit from working in collaboration. In the basic level, the researcher will use a pre-existing simulation that was carried out by another professional. Furthermore, he will need to know the application domain to create a model with the correct data (such as meshes, boundary conditions, etc.). In the advanced level, the researcher will develop new simulators, based on the available knowledge of the appropriate physical representation and the mathematical model of the problem.

Therefore, it is necessary to have an infrastructure in which expert researchers (from the advanced level) share simulators, models and existing data in such a way that they can be used by the other researchers from the basic level. Likewise, it is suitable to have a more advanced infrastructure, such as tools to build new simulators, to meet the demands of the advanced level.

2.3. The Infrastructure to Execute Simulations

Geomechanical simulations usually include complex calculations, which can be very time consuming in terms of execution. It is essential to have an infrastructure that allows for executing simulations applying distribution and parallelization techniques, taking advantage of heterogeneous resources such as clusters with GPU, fast I/O and CPUs with high speed cores and interconnectivity. Current resources such as cloud services, internet protocols and virtual machines offer accessible and appealing infrastructure that reduces cost with IT operations, without losing performance, availability and stability.

Furthermore, in a large scale geomechanical project, simulations often generate data at high speed and usually process a large volume of data of different types, which basically determines the characteristics of a big data scenario. Considering the complexity of the calculations and possibly re-executions, generating even more data, it is fundamental to have data intensive computing infrastructure that ensures run-time data

access efficiency, taking into account strategies to minimize data transfer and network latency.

2.4. Pre- and Post-Processing Applications

Data preparation to perform a simulation and the visualizations of the results are stages of research projects that use numerical tools. Geomechanics projects share some common characteristics such as complex geometry, mesh generation, definition of boundary conditions and result visualization. Computer graphics technologies are normally applied in the development of tools that support researchers in these stages. In geomechanical projects, the integration of a computational infrastructure necessary for simulation processing with tools that perform the pre-processing (preparation of input data) and post-processing (visualization of results) still represents a challenge. In many cases, researchers execute these stages independently and perform the integration manually. One example is data transfer and format conversion between an HPC cluster remote environment, which executes simulations, and desktops where users prepare and visualize the results.

Lately, high speed network services and advances in WebGL technologies for 2D and 3D drawing on browsers have allowed using the internet to build pre- and post-processing applications. Therefore, these new applications require a set of services in public or private clouds, that aims at integrating functionalities typical of a collaborative environment and hiding the complexity underlying the development stack of those applications.

3. The ERAS Portal

The ERAS Portal is developed by Institute Tecgraf/PUC-Rio according to the requirements presented in the previous section. It is currently used by researchers in geomechanical projects. Through the portal, researchers have access to software artifacts, such as models, documents and computational resources to access, execute and monitor their simulators, while working in a collaborative environment.

The conceptual architecture of ERAS presents the following modules: (a) ***Dashboard Panel***: a front-end that allows researchers to define, execute and monitor their simulation, and also visualize the results. This module also provides tools for publishing and sharing data among researchers who take part in the same project. (b) ***Execution Environment***: provides access to a data sharing and simulator execution infrastructure in a heterogeneous and distributed environment. (c) ***Simulation Environment***: offers a catalogue of existing simulators that can be used by researchers by inserting an input data set. In its advance mode, this module supports the building of new simulators. (d) ***Knowledge Base***: stores data and information on the research progress shared in the Portal, and also data provenance from simulations that were executed. (e) ***Access Control***: provides control regarding access restrictions according to user profile and the projects they take part in. Next, we present the main features of ERAS. A detail discussion on building new simulators can be found in [Mendes 2016].

3.1. Collaborative Projects

Users organize their work area in the Portal by creating projects. A project is a space for collaboration, storing and sharing of resources used in the simulations, such as input files

with simulation model data, output files with simulation results. The project owner is the user who created it, and he can choose to share it with specific users. The type of sharing can be read-only or read and write. Besides the file storage area used in simulations, a database maintains the project's meta-information, its recent activities, and simulation execution history. This track record is an important register of the parameters used in the simulation and the provenance of the results.

Every project has a forum for discussions among its participants. The forum allows the user to reference the project files and the simulations submitted to execution, facilitating the discussion about the progress of the on-going research. Statistical information, such as project space usage and simulation history, are available to its participants.

3.2. Execution of Simulations

The Portal is responsible for managing and monitoring the execution of simulations in a distributed and heterogeneous environment. Users that take part in the same project can monitor in real-time the processing results. The Portal offers a monitoring dashboard, shown in Figure 1(a), which presents the execution output logs in text and chart format, thus facilitating execution monitoring and the convergence analysis (or not) of the simulation results.

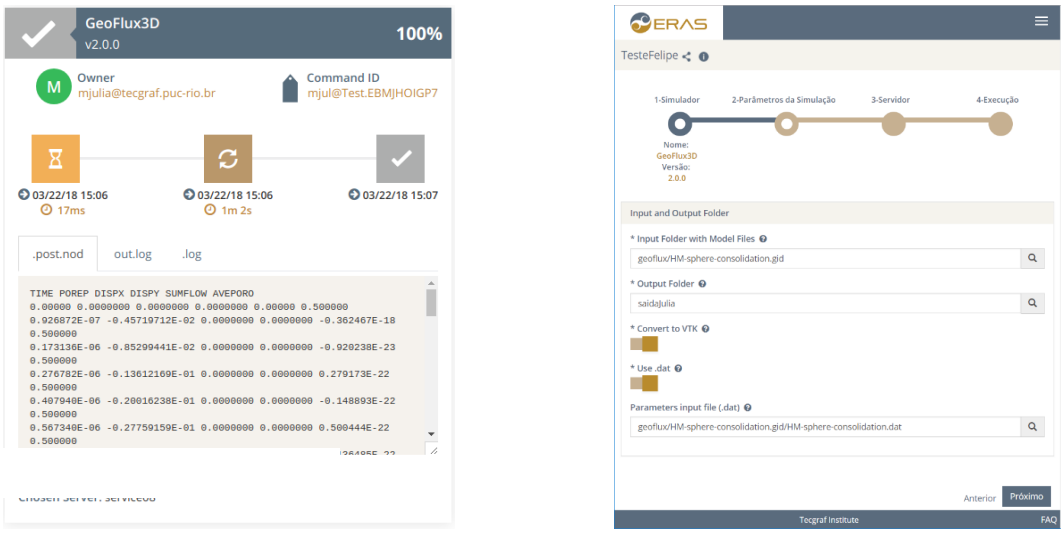

Figure 1. (a) Monitoring Dashboard (b) Form generated from a simulator configurator.

The execution of the simulations can be done in machines available in the Portal infrastructure. This infrastructure is comprised of physical and virtual machines with heterogenic resources and shared access to a data storage area. Monitor agents, which execute in the environment machines, provide information about computational resources, such as CPU, memory, disk and other specific properties set by the ERAS' administrator in each agent. In this way, the available infrastructure is deemed as a virtual cluster of computational resources. The scheduler adopted by Eras Portal's execution model submits the simulations to the cluster's machines that have the necessary requirements for the execution. Before submitting the simulation for execution on the selected machine, ERAS orchestrates the file staging needed. The Portal maintains an access control to its infrastructure that determines which user profiles can execute

simulations in designated machines in the environment. The access control can also limit which simulators and machines can be used by a group of users, establishing resource reserves for certain projects.

ERAS uses the middleware CSGrid [Lima, 2005] as backend. It mediates access to the computational resources of the ERAS environment. The CSGrid has been used in different projects. One of them is the mc2 tool, which supports the prototyping of scientific portals at SINAPAD [Gomes et al. 2015]. While already allocated resources (such as HPC and VM) are the main target hosts used by CSGrid, it can be integrated to other middlewares that manages the components of multiple virtualized infrastructures, including private and public clouds.

3.3. Sharing of Simulators

The Portal has a simulator repository that lists available simulators to which authenticated users are granted access. Each simulator has a configuration file that defines the input and output parameters which ERAS uses to create the simulation execution command in a machine of the environment. The dashboard client uses this same configurator to dynamically generate a web form to enter the parameters during submission. The simulator configurator specification language provides different types of parameters such as file, integer, float, list and table. This configuration feature allows new simulators installed in the portal to have a ready-made interface for parametrization and execution submission. Figure 1(b) shows a form generated from a simulator configurator that uses different types of input parameters for execution.

Simulators can be combined to build workflows, allowing users to create new orchestration models that can be saved in the project area. Therefore, existing simulators are also handled as re-usable and modular components that can be used across many different workflows.

4. Use Case: An application for Slope Stability Analysis

Besides the simulators, the Portal also presents a list of available applications in its dashboard. The applications offer a richer interface for the input data preparation stage (pre-processing) and the analysis and visualization of results (post-processing) stage. These applications employ ERAS Portal's API REST services to access the functionalities described in the previous section. An application for slope stability analysis is an example of a tool developed entirely with ERAS's API REST services.

Slope stability analysis is a procedure commonly used for verifying the safety of natural and artificial slopes (barriers, embankments, etc.). This assessment is done through the Safety Factor (SF), which provides a quantitative indicator of how far the slope is from rupture. Different approaches and methodologies are applied to assess the risks associated to engineering projects. Hybrid methods that combine the Shear Strength Reduction Method (SSRM) with probabilistic analysis and artificial neural networks are some of the techniques found in the literature [Shu and Gong, 2016]. The development of a probabilistic accumulated distribution curve as well as a neural network response surface need several simulation results to achieve a higher accuracy response. Even the SSRM needs a series of results to provide the SF. Therefore, the Slope Web application developed over the ERAS Portal services aims at providing support to this type of project.

In the Slope Web application, the slope geometry is built in a fast and simple manner. When filling in the input data, such as lithology and slope angle, a pre-visualization of the slope is generated, providing the user with a visual notion of the model being created. The user provides the specific weight, elastic properties (Young's module and Poisson's coefficient), and resistance parameters (cohesion and friction angle) of the layers. Finally, the user provides the position of the phreatic line through coordinates of points on the line, or through the "Pen" tool that allows drawing a line. Figure 2 illustrates the Slope Web application, which uses the ERAS Portal services.

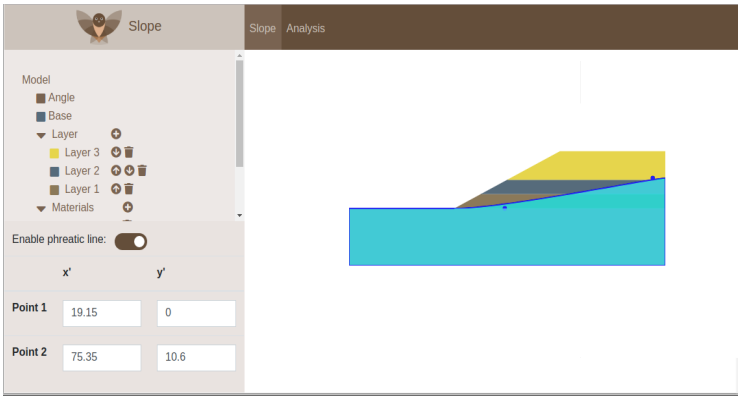

Figure 2. ERAS Portal Slope Web application

To obtain the SF, the user can define a search interval and precision of interest. Under user command, the application submits the simulation in ERAS Portal. The simulator uses the SSRM to calculate the SF, adjusting constantly the resistance parameters and assessing the slope instability through a non-linear finite element analysis. By the end of the execution, the application shows the 2D model generated by the simulation on the ParaViewWeb [Jourdain, 2010] application, integrated to the ERAS Portal for visualization of post-processing results. All the slope models used in the analysis are saved in the user project area, which can be shared with other researchers.

5. Conclusions

In this paper, we presented the main requirements to support the development of solutions that encompass the simulation of geomechanical problems, taking into account a collaborative environment with access to an efficient computer infrastructure. We also introduced the ERAS Portal, developed based on these set of requirements. The Slope Web is an example of an application oriented towards pre- and post-processing and developed over ERAS services.

ERAS is somewhat similar to many other related works in this area, with different approaches though. Different from E-ScienceNet peer-to-peer network architecture, ERAS is based on a client/server model. Currently, ERAS is being evolved to adopt a keyword-based query processing engine [Garcia et al. 2017]. Similar to E-ScienceNet assistance based on domain ontologies, this new capability will help users to search resources. The objectives of ^{my}Experiment and CrowdLabs are very related with the collaboration requirements (sections 2.1 and 2.2) that have driven our development in ERAS for Geomechanical users. However, ERAS project does not focus on Taverna and VisTrails workflows as the main shareable resources.

At present, the first version of ERAS Portal is operational with approximately 50 users, 10 simulators, and 10 machines for simulation execution. Besides the SlopeWeb,

new applications, such as well modeling, are being developed using the ERAS library. Future work includes integrating the framework GeMA [Mendes et al. 2016] to ERAS, which will provide support to the building of new simulators in the Portal. GeMA implements some important concepts of extensibility using of plugins and abstract interfaces, configurable orchestration and fast prototyping. Other future improvements also include the advanced searching mechanism that is being added to ERAS.

References

Classe, T., Braga, R., David, J.M., Campos, F. and Arbex, W. (2017) "A Distributed Infrastructure to Support Scientific Experiments", J. Grid Comput. 15, 4.

García, G.M., Izquierdo, Y.T., Menendez, E., Dartayre, F. and Casanova, M.A. (2017) "RDF Keyword-based Query Technology Meets a Real-World Dataset", Proc. 20th International Conference on Extending Database Technology (EDBT), March 21-24.

Goble, C.A. and De Roure, D. (2007) "^{my}Experiment: social networking for workflow-using e-scientists", WORKS '07: Proceedings of the 2nd workshop on Workflows in support of large-scale science.

Gomes, A. T. A., Bastos, B. F., Medeiros, V. and Moreira, V. M. (2015) "Experiences of the Brazilian national high-performance computing network on the rapid prototyping of science gateways", Concurrency and Computation: Practice and Experience, v. 27 (2), p. 271–289.

Griffiths, D. V. and Lane, P. A. (1999) "Slope stability analysis by finite elements", Géotechnique", v. 49, n. 3, p. 387-403.

Jourdain, S., Ayachit, U. and Geveci, B. (2011) "ParaViewWeb: A Web Framework for 3D Visualization and Data Processing". International Journal of Computer Information Systems and Industrial Management Applications, v. 3, p. 870–877.

Lima, M. J., Melcop, T., Cerqueira, R., Cassino, C., Silvestre, B., Nery, M. and Ururahy, C. (2005) "CSGrid: um Sistema para Integração de Aplicações em Grades Computacionais", In: Salão de Ferramentas do 23o. Simpósio Brasileiro de Redes de Computadores, Fortaleza, v. 2. p. 1207-1214.

Mates, P., Santos, E., Freire, J. and Silva, C.T. (2011) "CrowdLabs: Social Analysis and Visualization for the Sciences", International Conference on Scientific and Statistical Database Management SSDBM 2011, p. 555-564.

Mendes, C.A.T., Gattass, M. and Roehl, D. (2016) "The Gema Framework: An Innovative Framework for the Development of Multiphysics and Multiscale Simulations", Proceedings of the ECCOMAS Congress 2016.

Shu, S. X. and Gong, W. H. (2016) "An artificial neural network-based response surface method for reliability abalyses of c-slopes with spatially variable soil", China Ocean Engineering, v. 30, n. 1, p. 113-122.

Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalga, A., Balcazar Vargas, M., Sufi, S., Goble, C. (2013): The Taverna workflow suite: designing and Acids Research, 41(W1): W557–W561. https://doi.org/10.1093/nar/gkt328

Integração de Dados na Detecção de Alvos para Fármacos de *Schistosoma mansoni*

Francimary P. Garcia[1], **Kele Teixeira Belloze**[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – CEFET/RJ

francigarciaoliveira@gmail.com, kele.belloze@cefet-rj.br

Abstract. *Schistosomiasis mansoni caused by Schistosoma mansoni organism is a major neglected disease occurring in the world. However, there is a single drug recommended by the World Health Organization for its treatment. Therefore, searching for alternative drug targets in the fight against the disease is important. This work aims to identify possible new drug targets for S. mansoni. The methodology adopts an approach based on orthology and homology making use of essential proteins of model organisms and proteins already known as drug targets, and integration of these data. As a preliminary result, a list of 91 candidate proteins for drug targets was found.*

Resumo. *A esquistossomose causada pelo organismo Schistosoma mansoni é uma doença negligenciada importante pela ocorrência no mundo. Contudo, existe um único medicamento recomendado pela Organização Mundial de Saúde para o seu tratamento. Logo, pesquisas por alvos para fármacos alternativos no combate à doença são importantes. Este trabalho tem como objetivo identificar possíveis novos alvos para fármacos de S. mansoni. A metodologia adota uma abordagem baseada em ortologia e homologia fazendo uso de proteínas essenciais de organismos modelo e proteínas já conhecidas como alvos de fármacos, e integração destes dados. Resultados preliminares apontam uma lista de 91 proteínas candidatas a alvos para fármaco.*

1. Introdução

A esquistossomose é uma doença negligenciada causada por organismos helmintos (vermes parasitários) do gênero *Schistosoma*. De acordo com a Organização Mundial de Saúde (OMS), a doença afeta quase 240 milhões de pessoas em todo o mundo [WHO 2018]. A ocorrência da doença prevalece em áreas tropicais e subtropicais, em comunidades pobres sem água potável e saneamento adequado. Mais de 700 milhões de pessoas vivem em áreas com essas características no mundo. O *S. mansoni*, causador da maioria das infecções em humanos, é a única espécie do gênero descrita no Brasil [Souza et al. 2011]. De acordo com dados do Ministério da Saúde, o número de casos da doença na área endêmica do Brasil, a qual engloba principalmente a região Nordeste, é de quase 20 mil [MS 2017].

Para o combate à doença é utilizado o medicamento praziquantel (PZQ) recomendado pela OMS, que tem sido usado na prática clínica há quase quatro décadas [Neves et al. 2016]. No entanto, devido à alta incidência de reinfecção, o uso generalizado e repetido deste medicamento em áreas endêmicas, suscita preocupações sobre o

desenvolvimento de resistência ao medicamento pelo helminto. Este problema é enfatizado ainda mais pela falta de eficácia do PZQ contra os vermes juvenis, o que é uma causa potencial de falha no tratamento em áreas endêmicas [Caffrey et al. 2009]. Por esta razão, pesquisas por alvos para fármacos alternativos no combate à esquistossomose são urgentes.

Diante do cenário apresentado, o objetivo deste trabalho é identificar possíveis novos alvos (proteínas) para fármacos de *S. mansoni* no combate à doença, tendo como foco os atributos de essencialidade e drogabilidade das proteínas. Para a metodologia desse trabalho, é feita uma integração de dados obtidos por meio da aplicação de uma abordagem baseada em ortologia e homologia na qual são utilizados: i) dados sobre as proteínas essenciais de organismos modelo, para trabalhar o atributo da essencialidade e, ii) dados sobre proteínas já classificadas como alvos de fármacos desenvolvidos e comercializados, para trabalhar o atributo da drogabilidade. Como resultado foi encontrada uma lista de 91 proteínas de *S. mansoni* candidatas a alvos para fármacos.

Além dessa introdução, este artigo está organizado nas seguintes seções: a seção 2 descreve os conceitos que embasam este trabalho; a seção 3 apresenta os trabalhos relacionados; a seção 4 detalha a metodologia usada na condução da pesquisa; a seção 5 apresenta os resultados obtidos e a seção 6 descreve as considerações finais sobre o artigo.

2. Alvos para Fármacos e Homologia

A descoberta de fármacos baseada em alvo é uma técnica comumente utilizada, porque pode reduzir os custos de algumas experiências laboratoriais necessárias para o processo inicial de desenvolvimento de fármacos [Guido et al. 2010]. Contudo, essa é uma tarefa não trivial a partir de dados experimentais. Sendo assim, as análises *in silico* apoiam essa descoberta levantando características consideradas desejáveis em um alvo para fármaco, como a essencialidade (se ausente, causa a morte da célula biológica), a drogabilidade (se as moléculas semelhantes a fármacos são suscetíveis de interagir com o alvo), a especificidade/seletividade (potencial para inibir o patógeno sem prejudicar o hospedeiro) e a importância das fases do ciclo de vida do patógeno relevantes para a saúde humana [Crowther et al. 2010].

Entre as estratégias existentes para a descoberta de fármacos para tratar doenças tropicais negligenciadas, podemos citar a abordagem baseada em homologia. Essa abordagem é usada para inferir relações biológicas e características da evolução entre os organismos que estão sendo comparados [Morrison et al. 2015].

Homologia é a relação de ancestralidade entre duas ou mais entidades (e.g. genes ou proteínas), ou seja, significa dizer que as mesmas compartilham um ancestral comum [Koonin 2005]. Sendo assim, a homologia é um termo qualitativo [Moreira 2015]. A similaridade, por sua vez, corresponde ao grau de proximidade entre duas ou mais sequências moleculares, geralmente expresso em porcentagem (%). Portanto, a similaridade é um termo quantitativo. Sequências ou estruturas similares podem ou não compartilhar de um ancestral comum. Por exemplo, podemos dizer que dois genes homólogos são 90% similares no nível da sequência de nucleotídeos. Porém, estes genes não podem ser referidos como 90% homólogos [Moreira 2015].

Em um conceito mais específico, a ortologia é definida como um tipo de homologia onde genes de diferentes espécies descendem de um único gene no último ancestral

comum, a partir de um processo de especiação [Fitch 1970]. Os genes ortólogos são importantes para a compreensão da genômica e da biologia molecular. Isso se deve ao fato que conhecida a função de um determinado gene ortólogo A que se apresenta ortólogo a um gene B recém-sequenciado ou com função desconhecida, é possível inferir, ao menos de forma provisória, a função do gene B por meio de sua alta similaridade e conservação com esse gene bem conhecido (gene A) [Moreira 2015].

3. Trabalhos Relacionados

TDR Targets [Agüero et al. 2008] é um trabalho de destaque quando o assunto é a identificação de alvos para fármacos em patógenos de doenças negligenciadas. TDR Targets é um banco de dados que foi criado para facilitar as análises focadas em alvo para esses patógenos, os quais são priorizados pelo Programa Especial de Pesquisa e Treinamento em Doenças Tropicais (TDR) da Organização Mundial de Saúde. O banco de dados pode ser usado para duas tarefas científicas gerais: i) análise de proteínas individuais, encontrando informações relacionadas ao seu potencial como alvo para fármacos e; ii) triagem e classificação de múltiplas proteínas como candidatas alvo para fármacos de acordo com os critérios especificados pelo usuário.

Os trabalhos de [Crowther et al. 2010] e [Caffrey et al. 2009] utilizam análise *in silico* para identificar alvos para fármacos. Crowther e colaboradores (2010) apresentaram uma abordagem para priorizar as proteínas dos patógenos observando se as mesmas atendem aos critérios considerados desejáveis em um alvo para fármacos. Esses critérios são baseados em ambas as informações derivadas da sequência (por exemplo, massa molecular) e dos dados funcionais sobre a expressão, essencialidade, fenótipos, vias metabólicas e drogabilidade. Esta abordagem também destaca o fato que para muitos critérios relevantes faltam dados em patógenos menos estudados (por exemplo, helmintos), sendo demonstrado como essa questão pode ser parcialmente vencida utilizando-se do mapeamento de dados de genes homólogos em organismos bem estudados.

Caffrey e colaboradores (2009) empregaram uma abordagem de química comparativa utilizando o genoma do *S. mansoni* de forma a identificar genes essenciais putativos com base em semelhança com genes/proteínas essenciais identificados por determinação experimental em dois organismos modelo, o nematoide *Caenorhabditis elegans* e a mosca da fruta *Drosophila melanogaster*. Em seguida, definiram um subconjunto de possíveis alvos para fármacos para os quais as informações estruturais de proteínas alvo são conhecidas, incluindo ligantes.

O presente trabalho adotou, de maneira similar aos trabalhos apresentados, os atributos de essencialidade e drogabilidade para o levantamento de proteínas candidatas a alvo para fármaco de *S. mansoni*. Por meio de uma abordagem baseada em ortologia e homologia, utiliza os dados de proteínas bem anotadas e conhecidas como as proteínas dos organismos modelo e de um banco de dados de alvos para fármacos para levantar as proteínas do *S. mansoni* que podem conter os atributos de essencialidade e drogabilidade. O diferencial em relação aos demais trabalhos é a integração dos dados realizada, na qual encontra inicialmente as proteínas candidatas essenciais do *S. mansoni* e a partir destas, encontra as proteínas com características de drogabilidade, resultando assim em conjunto de proteínas com características de essencialidade e drogabilidade ao mesmo tempo.

4. Metodologia

A metodologia aplicada na condução desta pesquisa baseou-se no trabalho de [Belloze 2013] que propôs a utilização dos conceitos de homologia e atributos de essencialidade e drogabilidade da proteína para apoiar a priorização de alvos no combate a doenças tropicais negligenciadas causadas por protozoários. O presente trabalho se diferencia no organismo de estudo, na ferramenta adotada para a busca de proteínas ortólogas (Atividade 1 descrita a seguir) e na integração de dados realizada. Assim, a metodologia aplicada considerando suas modificações em relação ao trabalho supracitado é apresentada na Figura 1 e detalhada nas atividades 1 e 2 descritas a seguir.

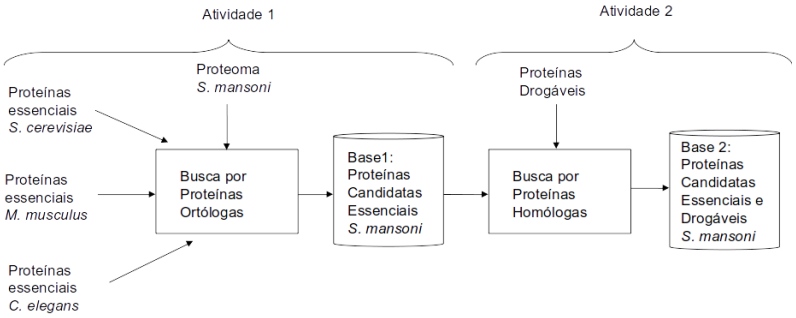

Figura 1. Busca de proteínas ortólogas, entre as proteínas do *S. mansoni* e as proteínas essenciais dos três organismos modelo apresentados e com este resultado, a busca de proteínas homólogas, contra proteínas drogáveis, resultando na base de proteínas candidatas essenciais e drogáveis do *S. mansoni*.

Atividade 1: inicialmente foi realizada a identificação de proteínas ortólogas entre as proteínas do *S. mansoni* e as proteínas essenciais de três organismos modelo eucarióticos: *Caenorhabditis elegans* (nematódeo), *Saccharomyces cerevisiae* (levedura) e *Mus musculus* (camundongo), baseando-se no conceito de essencialidade. De acordo com o conceito de ortologia, duas proteínas ortólogas podem ter a mesma função. Sendo a proteína do organismo modelo uma proteína essencial, foi pretendido então, por ortologia, sugerir o atributo de essencialidade às proteínas do *S. mansoni*. As proteínas ortólogas obtidas foram submetidas a um critério de corte considerando a ocorrência repetida nos quatro organismos. Desta maneira, foi obtida a base intermediária (Base 1) de proteínas candidatas a essenciais do *S. mansoni*.

O proteoma do *S. mansoni* foi obtido a partir da base *Ensembl Metazoa* [Kersey et al. 2017], enquanto as proteínas essenciais dos organismos modelo foram obtidas a partir da base de genes essenciais DEG (Database of Essential Gene) [Zhang et al. 2004].

Para a busca de ortologia foi utilizada a ferramenta *Orthofinder* [Emms and Kelly 2015], selecionada por aplicar um método que infere grupos de ortólogos (ortogrupos) de genes codificadores de proteínas. Na busca por sequências ortólogas, o *Orthofinder* executa as seguintes atividades:

1- Realiza busca BLAST [Altschul et al. 1990] *all-versus-all* (utilizando *e-value* padrão 1e-3); 2- Compara comprimento do gene e realiza normalização filogenética da distância do parâmetro *Score bit* (mede a similaridade de sequência, independente do tamanho da sequência de consulta e do tamanho do banco de dados) do BLAST, para que

os melhores resultados entre todas as espécies alcancem as mesmas pontuações, independentemente do comprimento da sequência ou da distância filogenética; 3- Delimita limiares de similaridade de sequência de ortogrupos usando RBNHs (*Reciprocal Best Length-Normalised Hit*); 4- Constrói um gráfico de ortogrupos para entrada no MCL (*Markov Cluster Algorithm*) [Enright et al. 2002]; 5- Agrupa genes em ortogrupos usando o MCL.

Atividade 2: em seguida, foi conduzido o processo de identificação de proteínas homólogas entre as proteínas candidatas a essenciais do *S. mansoni* levantadas na atividade 1 (Base1) e proteínas drogáveis (alvos para fármacos) disponibilizadas publicamente no banco de dados *DrugBank* [Wishart et al. 2017]. Nesta atividade, foi considerada apenas a homologia entre as sequências, pois duas proteínas homólogas possuem alta similaridade. Logo, se uma proteína que já é um alvo para fármaco e, portanto, possui características de drogabilidade, for altamente similar a uma proteína do *S. mansoni*, podemos sugerir que esta última pode conter características de drogabilidade também, não importando a função. As proteínas já consideradas alvos para fármacos foram obtidas dos conjuntos de dados das categorias *Approved* e *Small Molecule* do banco de dados *DrugBank*. A ferramenta BLAST foi utilizada para identificação das proteínas homólogas. Esta atividade resultou em uma base de dados (Base 2) composta por proteínas candidatas essenciais e drogáveis do organismo estudado, representadas por sequências primárias.

5. Resultados

Para a identificação das proteínas ortólogas foram utilizadas as sequências de proteínas dos quatro organismos (*S. mansoni* e os três organismos modelo) na mesma execução, possibilitando desta forma, a construção da árvore filogenética dos organismos e posterior identificação de ortólogos. O número de proteínas ortólogas encontradas é mostrado na Figura 2, na qual, por meio de uma representação de conjuntos, estão destacadas as quantidades de proteínas ortólogas entre o *S. mansoni* e cada um dos organismos modelo e as proteínas em comum a cada dois e três organismos modelo. A ortologia entre o *S. mansoni* e o *C. elegans*, por exemplo, resultou em 169 proteínas que só ocorreram nessa combinação, 118 proteínas que ocorreram também na ortologia entre o *S. mansoni* e o *S. cerevisae*, 111 proteínas que ocorreram também na ortologia entre o *S. mansoni* e o *M. musculus* e 138 proteínas que ocorreram nos três processos de ortologia.

Para continuidade da pesquisa, foram utilizadas as sequências de proteínas que representaram a interseção do resultado da ortologia entre as proteínas de *S. mansoni* e as proteínas essenciais dos três organismos modelo, ou seja, 138 proteínas do *S. mansoni* também encontradas na base de proteínas essenciais dos três organismos modelo, representando assim uma maior chance de se caracterizarem como essenciais.

Para a identificação das proteínas homólogas, a base de proteínas drogáveis foi composta de 7.172 sequências, das quais 2.683 sequências pertencentes à categoria *Approved* e 4.489 sequências pertencentes à categoria *Small Molecule*. Foi executada a ferramenta BLAST, na sua modalidade BLASTp (comparação entre sequências de proteínas) utilizando como entrada (proteína *query*), o arquivo com 138 sequências do *S. mansoni* candidatas essenciais e a base de proteínas drogáveis citada anteriormente.

Algumas execuções do BLASTp foram realizadas a fim de identificar os melhores valores para os parâmetros *evalue, best_hit_score_edge e best_hit_overhang*, além de consultas à literatura. Os parâmetros usados realizam as seguintes restrições às combinações

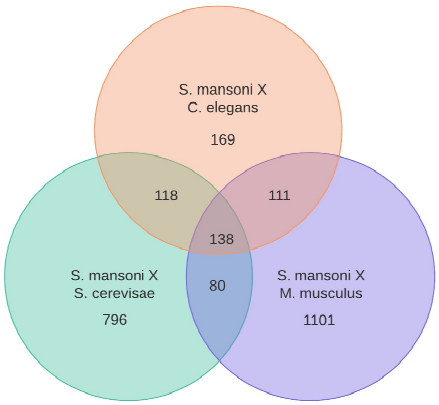

Figura 2. Análise quantitativa da ortologia realizada entre o proteoma do *S. mansoni* e cada conjunto de proteínas essenciais dos três organismos modelo e suas interseções.

obtidas: *e-value* indica o número de alinhamentos que seriam esperados apresentando valores de escore iguais ou melhores que o encontrado por acaso, dado o tamanho do banco de dados; *best_hit_score_edge* restringe os resultados aos melhores hits encontrados para cada *query* dentro do valor de e-value escolhido e *best_hit_overhang* controla quando um HSP (High-scoring Segment Pair) é considerado suficientemente curto para ser filtrado devido à presença de outro HSP.

Os valores usados foram: *e-value* = 1e-10 , *best_hit_score_edge* = 0.05 e *best_hit_overhang* = 0.25. Como resultado, foi obtida uma lista com 91 proteínas candidatas à alvos para fármacos. Uma lista com as 10 proteínas que apresentaram maiores percentuais (%) de identidade é mostrada na Tabela 1, na qual são apresentadas as seguintes informações: identificador da proteína do *S. mansoni*, nome da proteína, identificador da proteína homóloga do *DrugBank*, os valores de *e-value*, *bit score* e percentual (%) de identidade (Ident), obtidos no processo de homologia do BLAST. É apresentada também a informação sobre qual a categoria do *DrugBank* ocorreu a homologia, se *Approved* ou *Small Molecule*.

6. Conclusão

A pesquisa por alvos para fármacos que combatam a esquistossomose tem caráter importante devido ao elevado número de pessoas expostas a condições de pobreza que favorecem a ocorrência da doença nos países subdesenvolvidos. A dificuldade de investimentos da iniciativa privada neste setor e a existência de apenas um medicamento e que ainda pode vir a desenvolver resistência pelo parasita, representam preocupações que confirmam a necessidade de pesquisas por fármacos alternativos ao *S. mansoni*.

A metodologia proposta neste trabalho foca nos atributos de essencialidade e drogabilidade das proteínas na busca por candidatas a alvos que possam ser utilizados em pesquisa por novos fármacos, reduzindo o tempo e o custo envolvidos no processo de desenvolvimento de fármacos. Após as etapas realizadas nas atividades 1 e 2 da metodologia, foi obtida uma lista com 91 proteínas do organismo estudado, consideradas candidatas essenciais e drogáveis para acompanhamento experimental em testes de bancada a fim de verificar possíveis novos alvos para fármacos.

Tabela 1. Da lista de 91 proteínas candidatas essenciais e drogáveis do *S. mansoni*, são apresentadas as dez proteínas com maiores percentuais de identidade. A=Approved, S=Small.

S. mansoni	Nome da Proteína	ID Drugbank	Evalue	Bit Score	%Ident	A /S
Smp_026560.1	Putative calmodulin	P0DP25	4E-072	214	99.07	A
Smp_026560.2	Putative calmodulin	P0DP25	3E-103	295	97.99	A
Smp_203130.1	Putative uncharacterized protein	P63261	2E-133	380	96.72	A
Smp_183710.1	Putative actin	P63261	0.0	762	95.99	A
Smp_046600.1	Actin-1	P63261	0.0	762	95.99	A
Smp_161920.1	Putative actin	P63261	0.0	731	93.58	A
Smp_202970.1	Putative uncharacterized protein	P63261	0.0	727	91.15	A
Smp_018240.2	Cell division control protein 48 aaa family protein	P55072	0.0	1092	85.85	S
Smp_018240.1	Cell division control protein 48 aaa family protein	P55072	0.0	1021	84.63	S
Smp_067980.1	Ubiquitin conjugating enzyme E2, putative	P62837	6E-043	137	83.12	S

Este é um trabalho em andamento e como próximos passos desta pesquisa, estão previstas mais três etapas. Primeiro, o cruzamento da lista de proteínas final obtida com uma lista de proteínas essenciais do *Homo sapiens*, de modo a garantir que não incluiremos nesta nenhuma proteína essencial ao ser humano. Em seguida será realizada a identificação de informações sobre as estruturas secundárias destas proteínas, de forma a enriquecer a base de dados concebida. Finalmente, será feita a obtenção de um índice de drogabilidade para a lista de proteínas obtida, utilizando-se de modelos de padrões frequentes como Apriori, de modo a identificar comportamentos consistentes entre as proteínas candidatas e pesos obtidos por meio da análise de características da lista de proteínas.

Referências

Agüero, F., Al-Lazikani, B., Aslett, M., Berriman, M., Buckner, F. S., Campbell, R. K., Carmona, S., Carruthers, I. M., Chan, A. E., Chen, F., et al. (2008). Genomic-scale prioritization of drug targets: the tdr targets database. *Nature reviews Drug discovery*, 7(11):900.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.

Belloze, K. T. (2013). *Priorização de alvos para fármacos no combate a doenças tropicais negligenciadas causadas por protozoários.* Doutorado em biologia computacional e sistemas, Instituto Oswaldo Cruz, Fiocruz, Rio de Janeiro.

Caffrey, C. R., Rohwer, A., Oellien, F., Marhöfer, R. J., Braschi, S., Oliveira, G., McKerrow, J. H., and Selzer, P. M. (2009). A comparative chemogenomics strategy to predict potential drug targets in the metazoan pathogen, schistosoma mansoni. *PloS one*, 4(2):e4413.

Crowther, G. J., Shanmugam, D., Carmona, S. J., Doyle, M. A., Hertz-Fowler, C., Berriman, M., Nwaka, S., Ralph, S. A., Roos, D. S., Van Voorhis, W. C., et al. (2010).

Identification of attractive drug targets in neglected-disease pathogens using an in silico approach. *PLoS neglected tropical diseases*, 4(8):e804.

Emms, D. M. and Kelly, S. (2015). Orthofinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome biology*, 16(1):157.

Enright, A. J., Van Dongen, S., and Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584.

Fitch, W. M. (1970). Further improvements in the method of testing for evolutionary homology among proteins. *Journal of molecular biology*, 49(1):1–14.

Guido, R. V. C., Andricopulo, A. D., and Oliva, G. (2010). Planejamento de fármacos, biotecnologia e química medicinal: aplicações em doenças infecciosas. *Estudos Avançados*, 24:81 – 98.

Kersey, P. J., Allen, J. E., Allot, A., Barba, M., Boddu, S., Bolt, B. J., Carvalho-Silva, D., Christensen, M., Davis, P., Grabmueller, C., et al. (2017). Ensembl genomes 2018: an integrated omics infrastructure for non-vertebrate species. *Nucleic acids research*, 46(D1):D802–D808.

Koonin, E. V. (2005). Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, 39:309–338.

Moreira, L. (2015). Ciências genômicas: fundamentos e aplicações. *Ribeirão Preto: Sociedade Brasileira de Genética*.

Morrison, D. A., Morgan, M. J., and Kelchner, S. A. (2015). Molecular homology and multiple-sequence alignment: an analysis of concepts and practice. *Australian Systematic Botany*, 28(1):46–62.

MS (2017). Ministério da saúde. Disponível em: `http://portalms.saude.gov.br/saude-de-a-z/esquistossomose/situacao-epidemiologica`. Data do Acesso: 21 de Março de 2018.

Neves, B. J., Dantas, R. F., Senger, M. R., Melo-Filho, C. C., Valente, W. C., De Almeida, A. C., Rezende-Neto, J. M., Lima, E. F., Paveley, R., Furnham, N., et al. (2016). Discovery of new anti-schistosomal hits by integration of qsar-based virtual screening and high content screening. *Journal of medicinal chemistry*, 59(15):7075–7088.

Souza, F., Vitorino, R. R., Costa, A., Faria Jr, F., Santana, L., and Gomes, A. P. (2011). Esquistossomose mansônica: aspectos gerais, imunologia, patogênese e história natural. *Rev Bras Clin Med*, 9(4):300–7.

WHO (2018). Shistosomiasis. Disponível em: `http://www.who.int/schistosomiasis/en/`. Data do Acesso: 01 de Março de 2018.

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., et al. (2017). Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082.

Zhang, R., Ou, H.-Y., and Zhang, C.-T. (2004). Deg: a database of essential genes. *Nucleic acids research*, 32(suppl_1):D271–D272.

Modelagem de um *Data Mart* para Leituras do Fluxo de Múons Captadas pelos Telescópios *New*-Tupi*

Lucas Bertelli[1], Marcel N. de Oliveira[2], Nívia Ferreira[2], Carlos E. Navia[2], Daniel de Oliveira[1]

[1]Instituto de Computação – Universidade Federal Fluminense (IC/UFF)

[2]Instituto de Física – Universidade Federal Fluminense (IF/UFF)

lucasbm@id.uff.br, {paulista,nivia}@fisica.if.uff.br

navia@if.uff.br, danielcmo@ic.uff.br

Abstract. *The muon is the most abundant charged particle of cosmic radiation secondary at sea level. By reading the flow of muons, physicists are able to analyze and identify transient solar events that can impact our planet. New-Tupi telescopes are capable of reading the flow of muons that comes to Earth. These telescopes generate a large amount of data that needs to be queried by physicists. However, currently these telescopes store all the readings performed in binary files, which makes it difficult to elaborate queries on the data and its subsequent analysis. The goal of this paper is to propose a Data Mart for the New-Tupi telescope data, allowing for physicists to perform more complex queries in an easy way and with acceptable performance without having to rely on scripts or third-party programs to implement queries about the files.*

Resumo. *O múon é a partícula carregada mais abundante da radiação cósmica secundária ao nível do mar. Por meio da leitura do fluxo de múons, físicos são capazes de analisar e identificar eventos solares transientes, que podem gerar impactos em nosso planeta. Os telescópios New-Tupi são telescópios capazes de efetuar a leitura do fluxo de múons que chega à Terra. Esses telescópios geram um grande volume de dados que precisa ser consultado pelos físicos. Entretanto, atualmente tais telescópios armazenam todas as leituras realizadas em arquivos binários, o que dificulta a elaboração de consultas sobre os dados e sua posterior análise. O objetivo deste artigo é propor um Data Mart para os dados do telescópio New-Tupi, possibilitando aos físicos realizarem consultas mais complexas de forma fácil e com desempenho aceitável sem ter que recorrer à scripts ou programas de terceiros para implementar as consultas sobre os arquivos.*

1. Introdução

Os raios cósmicos são partículas que atingem a atmosfera da Terra a todo instante. Eles podem ser de origem galáctica ou frutos da radiação solar. Os raios cósmicos podem ser divididos entre primários e secundários. Os primários possuem sua origem em fontes astrofísicas e os secundários são fruto da interação dos primários com o topo de nossa atmosfera, gerando assim um chuveiro de partículas. Nesses eventos, os píons [Lattes et al. 1947] decaem produzindo múons, neutrinos e raios gama através do seguintes processos: $\pi^{\pm} \rightarrow \mu^{\pm} + \overline{\nu}_{\mu}$ e $\pi^{0} \rightarrow \gamma + \gamma$.

O múon é a partícula carregada mais abundante da radiação cósmica secundária ao nível do mar e a única partícula com carga elétrica capaz de penetrar profundamente no subsolo

*Este artigo foi financiado parcialmente pelo CNPq, CAPES e FAPERJ

terrestre [Zavattini 1975]. A medição do fluxo dessa partícula permite que cientistas estudem eventos solares transientes, tais como: erupções solares, ejeções de massa coronal (*Coronal Mass Ejection* - CME), radiações e tempestades geomagnéticas. Alguns desses eventos podem causar consequências para os modernos meios de comunicação e clima da Terra [Augusto et al. 2017]. Existem diversos detectores de partículas que são capazes de detectar raios cósmicos secundários, como os compõem a rede mundial de monitores de nêutrons (*Network Neutron Monitor Database* - NMDB[1]). A Universidade Federal Fluminense possui os telescópios de múons *New*-Tupi[Augusto et al. 2017]. O projeto *New*-tupi é um *upgrade* do projeto Tupi [Augusto et al. 2012a].

Os telescópios *New*-Tupi geram um grande volume de dados diário, aproximadamente 48.000 leituras do fluxo de múons (*i.e.,* uma leitura a cada dois segundos aproximadamente). Atualmente, tais dados são armazenados em arquivos binários (*.DAT), que são gerados automaticamente pelos telescópios e gravados em um repositório na nuvem. De forma a extrair conhecimento útil de tais leituras (*e.g.,* analisar a ocorrência de eventos solares transientes, por exemplo), os físicos necessitam realizar uma série de consultas complexas, muitas delas com agregações sobre tais dados (somatórios, médias, *etc.*) utilizando diferentes *bins* (intervalos de tempo como dias, semanas, meses), o que torna o trabalho tedioso e propenso a erros se realizado de forma manual ou por meio do uso de *scripts*.

Diante desse cenário, podemos observar diversos problemas e/ou limitações: (i) Duplicidade de arquivos: dois aquivos podem ser referentes a mesma leitura diária; (ii) os físicos precisam implementar os cálculos necessários em uma ferramenta de análise, ainda que sejam análises simples e recorrentes; (iii) cada físico pode implementar cálculos diferentes, o que pode gerar inconsistência na análise, e (iv) as ferramentas de análise muitas vezes não oferecem um desempenho aceitável ou não são capazes de trabalhar com a quantidade de dados envolvida. Dessa forma, é fundamental que se possa fornecer capacidade analítica de um banco de dados para os físicos.

Nas últimas décadas, uma abordagem capaz de prover tal capacidade analítica, chamada de *Data Warehouse* (DW), tem sido amplamente utilizadas em diversos domínios [Golfarelli and Rizzi 2009], sejam eles acadêmicos ou comerciais [Inmon 1992]. DWs são bases de dados multidimensionais que integram informações de diversas fontes a fim de facilitar a análise de dados. Um DW reúne e consolida informações de diversos *Data Marts* (DM) [Inmon 1992], que são um subconjunto dos DWs que possuem um objetivo específico. Um DM é uma coleção de dados orientada por assuntos, variante no tempo, e não volátil. Uma das maiores vantagens dos DMs frente aos bancos de dados transacionais é que eles possuem dados previamente sumarizados, *e.g.,* dados agregados por mês ou ano [Inmon 1992], o que acelera e facilita o processo de análise dos dados.

Dessa forma, o objetivo do presente artigo é desenvolver um DM, chamado TupiDM, para representar as leituras do fluxo de múons dos telescópios *New*-Tupi. Assim, a ideia é fornecer uma maneira estruturada e simples de os físicos processarem e analisarem um grande conjunto de dados científicos e apoiar a sua tomada de decisão.

Esse artigo se encontra estruturado em 5 seções. A Seção 2 apresenta o referencial teórico. A Seção 3 apresenta a abordagem proposta chamada TupiDM. A Seção 4 apresenta avaliação experimental. A Seção 5 discute trabalhos relacionados, e, finalmente, a Seção 6 conclui o artigo e discute trabalhos futuros.

[1]http://www.nmdb.eu/

2. Referencial Teórico

Essa seção apresenta alguns dos conceitos importantes para a compreensão desse artigo como modelagem dimensional e os telescópios *New*-tupi.

2.1. Modelagem Dimensional

Segundo [Kimball and Ross 2002], a modelagem dimensional é uma técnica de projeto de bancos de dados que visa apoiar consultas analíticas. Faz-se uso de redundâncias planejadas dos dados para melhorar o desempenho das consultas [Kimball and Ross 2002, Inmon 1992]. O modelo dimensional de um banco de dados é composto pelas tabelas *Fato* com suas respectivas *Dimensões*. As dimensões podem ser compartilhadas por tabelas fato diferentes. Existem dois modelos de implementação e um banco de dados dimensional: o Modelo Estrela [Kimball and Ross 2002] e o Modelo Floco de Neve [Inmon 1992]. O Modelo Estrela possui a tabela fato centralizada com as suas respectivas dimensões no seu entorno. Nesse modelo, a tabela fato possui chaves estrangeiras para todas as suas dimensões, sendo um modelo desnormalizado. O Modelo Floco de Neve é uma variação do Modelo Estrela, no qual todas as dimensões são normalizadas, fazendo com que sejam geradas quebras na tabela original ao longo de hierarquias existentes em seus atributos.

Um DW é constituído pela união dos DMs. Assim, como nos DMs, um DW preferencialmente deve ser modelado de forma dimensional, pois em comparação com um banco de dados transacional e normalizado, a modelagem dimensional produz modelos mais previsíveis e compreensíveis, facilitando a utilização e assimilação pelos usuários finais (no contexto desse artigo, os físicos), além de possibilitar consultas com alto desempenho [Kimball and Ross 2002]. Portanto, a modelagem dimensional possui uma estrutura simplificada, mais próxima da visão que o físico tem do seu domínio, facilitando assim a compreensão, de forma que os próprios físicos possam criar suas consultas. Apesar de terem uma estrutura diferente de bancos de dados transacionais, os bancos de dados dimensionais podem ser modelados sobre Sistemas de Gerência de Bancos de Dados (SGBDs) relacionais como o mySQL ou o PostsgreSQL.

2.2. Os Telescópios *New*-Tupi

Os telescópios de múons *New*-Tupi estão localizados no Instituto de Física da Universidade Federal Fluminense (22.9° S, 43.2° W; 5 m acima do nível do mar). Os telescópios são constituídos por dois detectores fixos e outros dois que podem ser orientados de modo a detectar partículas provenientes de uma determinada direção [Augusto et al. 2017]. Os telescópios *New*-Tupi são construídos a partir de quatro detectores de partículas idênticos. Cada detector é construído com base em um cintilador plástico (Eljen EJ-208) de tamanho (150 cm $\times$ 75 cm $\times$ 5 cm) e uma fotomultiplicadora (Hamamatsu R877) de 127 mm de diâmetro, colocado dentro de uma caixa de formato piramidal truncada com quatro inclinações. O conjunto do detector é conectado à base fotomultiplicadora Ortec ScintiPack^{TM} (Modelo 296).

Quando o múon atravessa o cintilador, este emite luz fluorescente que é captada pela fotomultiplicadora. A fotomultiplicadora converte a luz de baixa intensidade em um sinal elétrico, que é pré-amplificado até uma amplitude suficiente para facilitar uma posterior análise. A Figura 1 (esquerda) mostra uma fotografia dos telescópios *New*-Tupi. Os quatro detectores são colocados em pares, com os detectores T1 e T2 no topo, e B1 e B2 no fundo, como mostrado na Figura 1 (direita). Este *layout* permite medir o fluxo de múons a partir de três direções, a vertical (zênite), oeste e leste (com uma inclinação de 45 graus). Os telescópios registram a taxa de coincidência para a incidência vertical usando os pares de

detectores (T1, B1) e (T2, B2), bem como as coincidências cruzadas entre T1 e B2 (incidência oeste) e os T2 e B1 (incidência leste). A separação entre os detectores (vertical e horizontal) é 2,83 m.

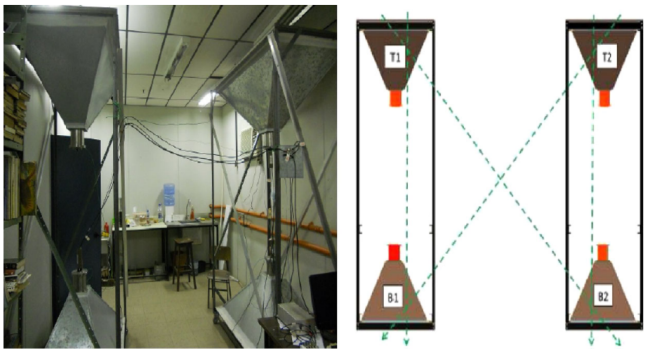

Figura 1. *Esquerda*: Fotografia dos telescópios New-Tupi. *Direita*: esquema geral do telescópio New-Tupi.

Os telescópios são automatizados e funcionam continuamente. Seus resultados ajudam a fomentar uma área emergente de estudos conhecida como clima espacial. Eles trabalham de forma sincronizada para medir continuamente o fluxo de partículas derivadas da radiação do Sol, investigando as possíveis relações entre os ciclos solares e as variações climáticas da Terra [Augusto et al. 2017]. Um exemplo do funcionamento do *New*-Tupi pode ser visto na Figura 2. A Figura 2 mostra o efeito do nível do solo da segunda maior tempestade geomagnética do atual ciclo solar (ciclo 24), conhecido como "tempestade do solstício de 2015". Esta tempestade geomagnética está associada a um período em que a Terra foi atingida por 4 CMEs. O painel direito da Figura 2 mostra que esta tempestade geomagnética atingiu a condição de Kp=8 (grave). O índice Kp é baseado na média da componente horizontal do campo magnético da Terra realizada por 13 estações de magnetômetros que estão situadas ao longo do equador terrestre. O painel esquerdo da Figura 2 mostra uma queda na contagem do telescópio *New*-Tupi vertical de forma bem relacionadas com dados do monitor de nêutrons situado no polo Sul e também com a sequência dos impactos provocados pelas CMEs. A queda na taxa de contagem do telescópio é esperada, conhecida como decréscimo Forbush [Augusto et al. 2012b] e está associada às tempestades geomagnéticas. Em geral, quanto mais intensa é a tempestade geomagnética, maior a queda na taxa de contagem de um detector de partículas ao nível do solo.

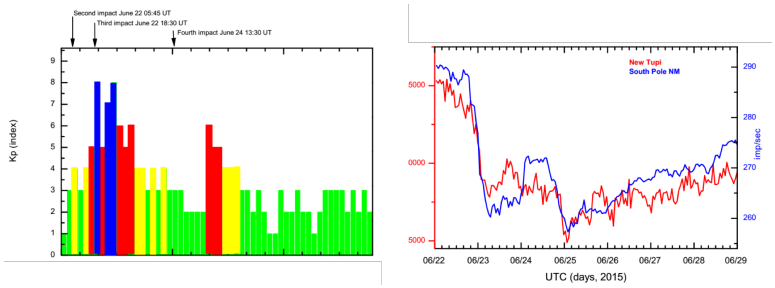

Figura 2. *Painel esquerdo*: Estimativa do índice planetário Kp (dados de 3 horas) por sete dias consecutivos, com início em 22 de junho de 2015 às 00:00 UT. *Painel direito*: Perfis temporais da taxa de contagem, para o mesmo período

3. Abordagem Proposta: TupiDM

Nessa seção apresentamos a abordagem proposta nesse artigo chamada TupiDM.

3.1. Cenário Atual

Atualmente, os dados capturados pelos telescópios *New*-Tupi são armazenados em formato binário em um repositório nuvem. As medições são fragmentadas em arquivos diários, com extensões .DAT. Os arquivos se encontram organizados por ano, mês e dia, com a nomenclatura DB_Tupi_"Ano"_"Mes"_"Dia". O conteúdo do arquivo é separado por tabulações, sendo a primeira coluna correspondente ao momento da medição representado no formato de tempo universal (número de segundos decorridos a partir de 01/01/1900). A segunda coluna corresponde às contagens para o telescópio vertical e a última coluna corresponde a soma das contagens dos telescópios inclinados (escaler). Atualmente, cabe aos físicos, localizar os arquivos desejados, realizar o *download* e calcular as agregações necessárias, de acordo com o período de tempo que se deseja analisar. Esse agrupamento hoje deve ser feito via *script* ou planilhas, o que não é escalável.

3.2. Modelo de Dados do TupiDM

A Figura 3 apresenta o Modelo do TupiDM. Foi utilizada a modelagem dimensional estrela [Kimball and Ross 2002], sendo a tabela fato representada pela tabela *FAT_SINAIS*, responsável por armazenar os fatos, que no contexto desse artigo são as contagens de sinais coincidentes detectados no detector vertical e no escaler, armazenados, respectivamente, nos atributos *valorVertical* e *valorEscaler*. Ainda na tabela *FAT_SINAIS*, os atributos *idTempo* e *idTelescopio* são chaves estrangeiras para as tabelas que representam as dimensões tempo e telescópio, respectivamente.

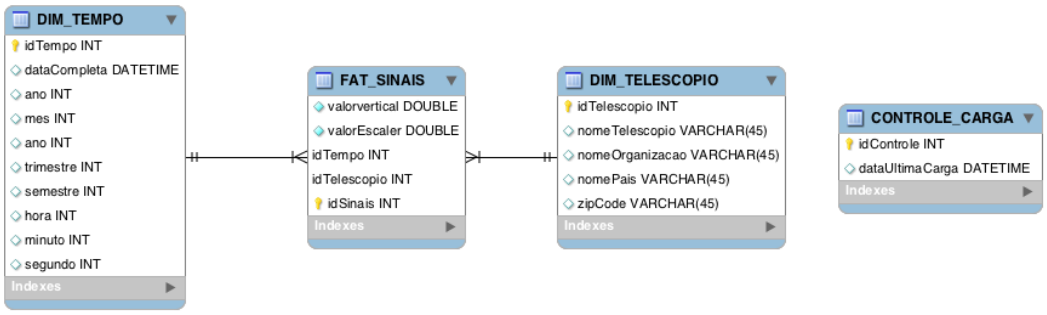

Figura 3. Esquema Estrela do TupiDM

A tabela *DIM_TEMPO* representa a dimensão tempo, sendo responsável por armazenar todas as possíveis granularidades de tempo para um determinado fato. O atributo *dataCompleta* representa a data completa (até milissegundos). Os atributos *ano, mes, dia, trimestre, semestre, hora, minuto* e *segundo* são as representações numéricas de partes da data completa. O atributo *idTempo* é a chave primária da tabela. A tabela *DIM_TELESCOPIO* representa a dimensão telescópio, sendo responsável por armazenar todas os possíveis grupos de telescópios para um fato. Ela é capaz de armazenar dados de outros telescópios de múons além dos *New*-Tupi, de forma a permitir comparações entre medições de diferentes telescópios de múons. Porém, no momento contém apenas o *New*-Tupi. Ela possui os atributos *idTelescopio, nomeTelescopio, nomeOrganizacao, nomePais* e *zipCode*, que armazenam, respectivamente, chave primária da tabela, nome do grupo de telescópio, nome da organização ao qual o telescópio pertence, o país sede e a zona de informação postal.

3.3. Arquitetura Proposta

A Figura 4 apresenta a arquitetura proposta, desde o processo de captação dos dados pelos telescópios até a carga no DM proposto. A arquitetura proposta segue o tradicional ciclo

de carga de um DW [Inmon 1992]. Os arquivos com as leituras vertical e escaler gerados pelos telescópios *New*-Tupi são armazenados no repositório remoto do Google Drive, que chamaremos somente de Drive a partir desse momento. Diariamente o componente ETL (*Extract, Transform, Load*), implementado por meio de um *script* Python, é executado. Esse *script* verifica se existem atualizações ou novos arquivos no Drive, e, caso haja, a mesma realiza o processo de carga no TupiDM. O TupiDM foi implementado no Sistema de Gerência de Banco de Dados PostgreSQL versão 9.6, instalado em um servidor com o sistema operacional Linux, distribuição Mint. O *script* python responsável pelo ETL foi agendado para ser diariamente executado com o *cron* do Linux.

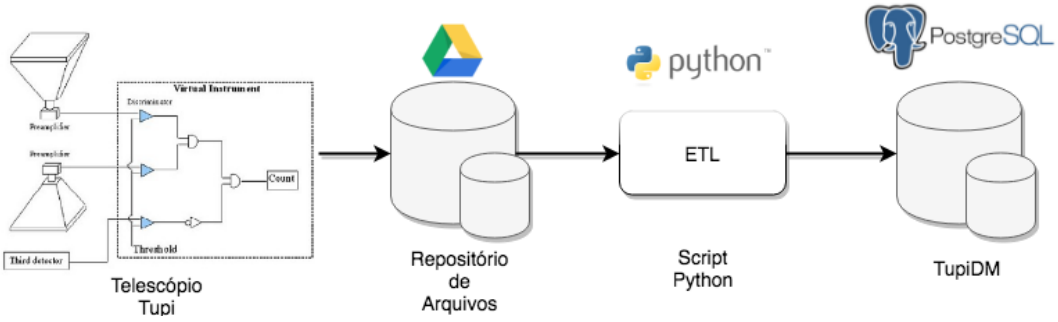

Figura 4. Arquitetura proposta para carga do TupiDM

Algorithm 1 Pseudo-Código do componente ETL

```
1:  novoArquivo: Array[];
2:  dataUltimaCarga: ZonedDateTime;
3:  dataUltimaCarga := consultaUltimaCarga();
4:  novoArquivo = checaNovoArquivoTupi (dataUltimaCarga);
5:  se (novoArquivo ≠ ∅) então
6:      novoArquivo := ordenaDataCrescente(novoArquivo);
7:      desabilitaIndiceBanco();
8:      para i ← 0 até tamanho(novoArquivo)-1 faça
9:          arquivo := novoArquivo[i];
10:         converteTempoUniversal(arquivo);
11:         carregaDM(arquivo);
12:     fim para
13:     insereDataControleCarga(date());
14:     habilitaIndicesBanco();
15: fim se
```

No Algoritmo 1 observamos os principais procedimentos realizados pelo componente ETL. Primeiramente, o componente consulta na tabela *CONTROLE_CARGA* do TupiDM quando foi feita a última carga de dados. Utilizando-se da data retornada pela consulta, o *script* busca no Drive por novos arquivos ou atualizações nos arquivos gerados pelo telescópio *New*-Tupi. Se houver resultados, os arquivos novos/atualizados são ordenados de forma crescente a partir de sua data de atualização, para que ao serem carregados na máquina, os arquivos mais atuais sobrescrevam os mais antigos na pasta, eliminando assim duplicatas. Os índices criados no TupiDM são desabilitados para agilizar o processo de carga. Para cada arquivo é verificado se já existem medições para aquele dia no TupiDM, e, caso haja, todos os registros correspondentes ao dia que está sendo inserido são excluídos, dessa forma garantimos que não há duplicidade de dados. Posteriormente, para cada linha contida nos arquivos a serem carregados é realizada a transformação do valor correspondente ao "tempo universal" para um formato de data, hora, minutos e segundos. Depois da transformação

Tabela 1. Desempenho das consultas no TupiDM

Consulta	S/ agregações	C/ agregações
Consultar o somatório dos valores das leituras verticais e escaler durante o ano de 2017	297.467 ms (aprox. 5 min)	352 ms
Consultar o somatório dos valores das leituras verticais e escalar no dia 02/06/2016	192.390 ms (aprox. 3,5 min)	121 ms
Consultar todas as leituras realizadas no dia 02/06/2016	2.432 ms	2.567 ms

a linha é inserida no TupiDM de acordo com a modelagem apresentada na Seção 3.2 e os agregados afetados têm seus valores atualizados. Ao término da carga, os índices são habilitados novamente e é inserida a data e hora, atuais na tabela *CONTROLE_CARGA*.

4. Avaliação Experimental

De forma a avaliar a abordagem proposta nesse artigo, realizamos uma avaliação experimental com uma amostra de 754 arquivos de leituras do telescópio *New*-Tupi. Esses arquivos correspondem às medições no período compreendido entre os anos 2014 e 2017. Após a execução do componente ETL, foram carregados apenas 750 arquivos no TupiDM, pois o algoritmo de carga descartou 4 arquivos duplicados. A tabela de fatos *FAT_SINAIS* possui um total de 33.311.791 de registros e a tabela da dimensão tempo *DIM_TEMPO*, também possui um total de 33.311.791 de registros, uma vez que existe uma leitura do telescópio para cada unidade de tempo representada. O TupiDM sem índices criados ocupou um espaço em disco de aproximadamente 4 GB, e com índices criados, um espaço aproximado de 12 GB.

A Tabela 1 exibe três consultas e seus tempos de execução associados (por limitações de espaço apresentamos apenas três consultas). Para cada consulta apresentamos o tempo de execução com agregações pré-calculadas e sem agregações pré-calculadas. Na primeira consulta foi explorada a agregação dos valores dos campos de leitura vertical e escaler durante todo o ano de 2017. A segunda consulta explora agregação dos valores de leitura vertical e escaler para um único dia. Ambas consultas se beneficiam de agregações pré-calculadas presentes no TupiDM e, por isso, apresentam grande variação no tempo de execução com o uso e sem o uso de agregações pré-calculadas. A terceira consulta lista todas as leituras em um dia. Como essa consulta não possui agregação, as agregações pré-calculadas não influenciam seu desempenho. É importante ressaltar que a amostra contém apenas 750 arquivos, e os tempos de execução tendem a aumentar quando todos os dados estiverem carregados no TupiDM. Este experimento reforça a necessidade de se utilizar agregações pré-calculadas.

5. Trabalhos Relacionados

Algumas soluções existentes já exploram a aplicação de bancos de dados para representar o fluxo de múons [Verducci 2007]. Entretanto, esses bancos de dados são comumente disponibilizados em arquivos .CSV para os físicos, o que limita a capacidade analítica dos mesmos. Nenhuma das abordagens existentes provê acesso a um banco de dados ”consultável” e nem que possua capacidades analíticas de um DM. [Verducci 2007] propõe um banco de dados para o fluxo de múons do CERN (Organização Europeia para a Pesquisa Nuclear). [Verducci 2007] afirma que o uso de agregações pré-calculadas pode acelerar a pesquisa, principalmente para dados em larga-escala. Similarmente, o *Cosmic Ray Muon Database*[2] também provê acesso a um *dataset* de leituras de múons em formato .CSV, o que também limita a capacidade analítica.

[2]http://cosray.shinshu-u.ac.jp/crest

6. Conclusões e Trabalhos Futuros

Os telescópios *New*-Tupi realizam leituras do fluxo de múons que chega ao nosso planeta. A análise desse fluxo pode identificar erupções solares, que são eventos que podem causar suspensões de atividades eletromagnéticas. Atualmente, todas as leituras realizadas pelos telescópios são armazenadas em arquivos binários. Dado o grande volume de dados que são gerados, consultar e analisar tais dados em arquivos se tornou uma tarefa tediosa e propensa a erros.

Nesse artigo, propomos o uso de um *Data Mart* para os dados capturados pelos telescópios *New*-Tupi chamado TupiDM. O TupiDM segue uma modelagem dimensional do tipo estrela [Kimball and Ross 2002] que permite que dados sejam armazenados de forma pré-calculada no TupiDM, acelerando assim consultas que eram bastante lentas anteriormente. Foi realizada uma avaliação experimental do TupiDM utilizando-se um sub-conjunto dos dados gerados (33.311.791 registros de leituras) e constatou-se que em algumas consultas a redução no tempo da consulta chegou a ordens de grandeza.

Como sugestões para trabalhos futuros, pode-se apontar a elaboração de uma aplicação *Web* que visa oferecer visualizações sobre os dados do TupiDM, que apresente gráficos de series temporais dinâmicos, de acordo com os filtros de tempo oferecidos. Além disso, uma aplicação *Web* também permitirá aos pesquisadores visualizar e analisar esses dados de qualquer lugar. Outro trabalho futuro interessante seria o estudo de padrões nos dados do TupiDM, utilizando-se de técnicas de mineração de dados, o que pode gerar novas descobertas na área.

Referências

Augusto, C. R. A., Kopenkin, V., Navia, C. E., Tsui, K. H., Shigueoka, H., Fauth, A. C., Kemp, E., Manganote, E. J. T., de Oliveira, M. A. L., Miranda, P., Ticona, R., and Velarde, A. (2012a). Variations of the muon flux at sea level associated with interplanetary icmes and corotating interaction regions. *The Astrophysical Journal*, 759(2):143.

Augusto, C. R. A., Kopenkin, V., Navia, C. E., Tsui, K. H., and Sinzi, T. (2012b). Search for a simultaneous signal from small transient events in the pierre auger observatory and the tupi muon telescopes. *Phys. Rev. D*, 86:022001.

Augusto, C. R. A., Navia, C. E., de Oliveira, M. N., Nepomuceno, A. A., Kopenkin, V., and Sinzi, T. (2017). Muon excess at sea level during the progress of a geomagnetic storm and high-speed stream impact near the time of earth's heliospheric sheet crossing. *Solar Physics*, 292(8):107.

Golfarelli, M. and Rizzi, S. (2009). *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Inmon, W. H. (1992). *Building the Data Warehouse*. John Wiley & Sons, Inc., New York, NY, USA.

Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition.

Lattes, C. M. G., Occhialini, G. P. S., and Powell, C. F. (1947). Observations on the Tracks of Slow Mesons in Photographic Emulsions. 2. *Nature*, 160:486–492. [,103(1947)].

Verducci, M. (2007). Atlas conditions database and calibration streams. *Nucl. Phys. B (Proc.Suppl.)*, 172:250.

Zavattini, E. (1975). Section 5 - muon capture. In Hughes, V. W. and Wu, C., editors, *Muon Physics*, pages 219 – 261. Academic Press.

Towards an e-infrastructure for Open Science in Soils Security

Sérgio Manuel Serra da Cruz[1,2,3], Marcos Bacis Ceddia[1], Eber Assis Schmitz[3], Gabriel S. Rizzo[2], Renan C. T. Miranda[2], Sabrina O. Cruz[2], Ana Clara Correa[2], Felipe Klinger[2], Elton Marinho[3], Pedro Vieira Cruz[2]

[1] Universidade Federal Rural do Rio de Janeiro – PPGMMC/UFRRJ
[2] Programa de Educação Tutorial - PET-SI/UFRRJ
[3] Universidade Federal do Rio de Janeiro – PPGI/UFRJ

serra@ufrrj.br, ceddia@ufrrj.br

Abstract. *Soils Security is a critical and growing global concern. The OpenSoils´ objective is to host, connect and share large amounts of curated soil data and knowledge at the Brazilian and South America level. The e-infrastructure consists of several layers of services, a database of soil profiles, a cloud-based computational framework to compute and share soil data integrated with a map visualization tools. OpenSoils is open, elastic, provenance-oriented and lightweight computational e-infrastructure that collects, stores, describes, curates, harmonizes and directs to various soil resource types: large datasets of soils profiles, services/applications, documents, projects and external links. OpenSoils is the first open science-based computational framework of soils security in the literature.*

1. Introduction

Agriculture consists of a complex science from a data-centric point of view, with different disciplines (from genomics to soil sciences) and, different scales (from genes to geolocalisation). The ability to explore this complex dataset is a crucial issue to tackle new agricultural and societal challenges like food and soils security (WOLFERT *et al.*, 2017). To Koch *et al.* (2013), soils are probably the most important natural resource and biosystem that support the human and terrestrial life. It is a primary, finite natural resource which derives other resources, goods, and services.

Soils security is an emerging chief concept of soil sciences motivated by sustainable development and precision agriculture. It is related to the maintenance and improvement of the global soil resource to produce food, fibers and fresh water, human health, carbon sequestration, contribute to energy and climate sustainability, and to maintain the biodiversity and the overall protection of the ecosystem (KOCH et al., 2013). Soils security, like food security, has several dimensions (*e.g.*, capability, condition, capital, connectivity, and codification) that interact with environmental, social, and economic components (MCBRATNEY, FIELD & KOCH, 2014). Soils security is a data-intensive research domain which life-cycle starts at the harvest of new soils data in the field and finish at scientist's visualization workstation or decision maker´s desk (Figure 1). It is important to highlight that Figure 1 did not capture the complexity of soils security, once it does not encompass the interconnection of the five dimensions and the political, economic and sociological aspects of soil use and management. Figure 1 summarizes the life cycle of soil information at the research and academic level, which is the primary focus of this research.

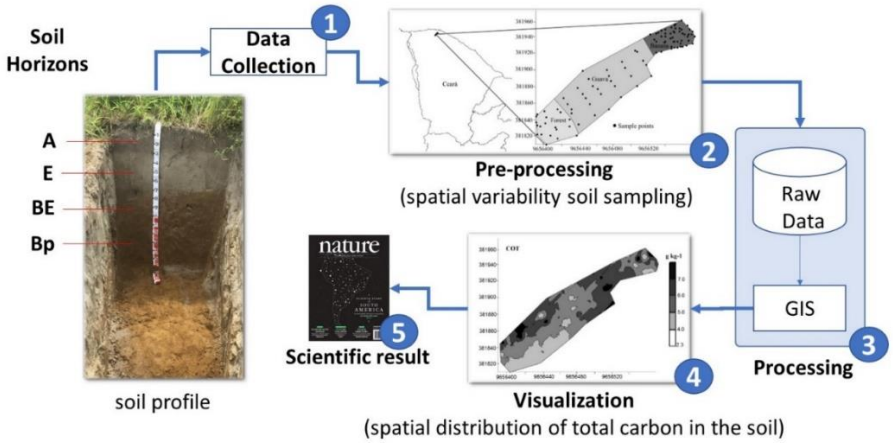

Figure 1 – Example of soil horizons and the main phases of the life-cycle of soils investigations (maps adapted from MELO *et al*., 2016).

Soils and food security investigations are in a rapid transformation. However, these disciplines did not draw the same degree of attention of other e-science subjects like bioinformatics, astronomy, computational chemistry. We advocate the utter necessity to do interdisciplinary research considering the roles of computer science, data governance, supply chain data integration and mathematical modeling in soils security to face the challenges. We foresee that several open data, semantic web, open science, big data, and data science approaches may aid the soils community to make wider investigations, do more accurate predictions in precision agriculture and deliver more knowledge to the society.

The goal of this paper is to present the big picture of OpenSoils. It was conceived to guide Brazilian policies by designing and laying the groundwork for a long-term effort aiming at achieving an e-infrastructure for open science in soils security that would position Brazil as a major global player at the forefront of research and innovation in this area. This paper is organized as follows. Section 2 presents the background. Section 3 presents OpenSoils conceptual architecture and uses. Section 4 the related work and Section 5 concluding remarks and future work.

2. Soil, Soils Data, and Open Science

The development of soil from inorganic and organic materials is a complex natural process. The soil is defined as the layer(s) of generally loose mineral and organic material that is affected by physical, chemical, and/or biological processes at or near the planetary surface and usually hold liquids, gases, and biota and support plants (VAN ES, 2017). The soil is considered an open system that interacts with other components of the geologic cycle. The characteristics of a soil are a function of Parent material, Climate, Relief, Organisms and Time. (PANSU & GAUTHEYROU, 2006). Soils are evaluated in the field through soil profiles, which is defined as a two-dimensional section composed of a vertical succession of horizons, commonly named O, A, B, C (beginning at the surface), that have been subjected to soil-forming processes (Figure 1). Each soil profile has very specific mineralogical, morphological, chemical, physical, biological and environmental properties. Soil investigations require actions in the field and wet scientific laboratories because soils properties are diverse and are hard to be collected, mapped, analyzed, stored and shared as soils data in databases.

Soils investigations, like any other scientific domain, has a life cycle and characteristics that deserves efforts to improve the long-term data management and use of

strategic the data assets (YAMSON *et al.,* 2016, ARROUAYS *et al*., 2017). Soil data has key features, for instance, there are lots of legacies unanalyzed raw data. However, either new or existing soils data are heterogeneous in its values and semi-structured in its formats.

Currently, there are many isolated data silos which store legacy soils data as (*e.g.,* scientific papers, spreadsheets, text, pdf files or web pages), having poor semantics and lacking metadata descriptors. Additionally, several soil databases are either inaccessible to structured queries or are presented as simple spreadsheets or text files, being hardly shared and reused by farmers and policymakers (ARROUAYS *et al.,* 2017). Lots of soil data and knowledge are still currently fragmented and at risk of getting lost in digital data silos or even in simple tables in scientific papers. Consequently, reproducing the results from scratch from several soils experiments is both time-consuming and error-prone at best, and sometimes impossible.

Recent evidence from meta-research studies suggests that problems with research integrity and reproducibility in several scientific domains (BAKER, 2016; NEVES *et al.,* 2017; FANELLI, 2018 & HUTSON, 2018; FREIRE & CHIRIGATI, 2018). Many scientists, journals, and funders are concerned about the biased, low reproducible and irreproducible scientific findings in soils security as well. Thus, one approach that may serve to expand the reliability and robustness of soils security investigations is the adoption of open science (MUNAFÒ, 2016), e-science (HEY *et al*., 2009) and data provenance (BUNEMAN *et al.,* 2000 & FREIRE *et al.,* 2008).

Open science is an umbrella term encompassing a multitude of assumptions about the future of knowledge construction (FECHER & FRIESIKE, 2013). It is a global movement to make scientific research, data, and dissemination accessible at all levels of an inquiring society. Nowadays, there are some open science infrastructures (*e.g.,* OpenAIRE, OSF, EOSC, among others) not experienced with features of soils security challenges. E-infrastructure is a computational tool that promotes open, centralized workflows by enabling capture of different aspects and products of the research life-cycle, including developing a research idea, designing an investigation, storing and analyzing collected data, and writing and publishing reports or papers. The e-infrastructures support a variety of scientific tools and services to assist in the research process (FOSTER & DEARDORFF, 2017).

3. OpenSoils e-infrastructure

It is useful to start from a theoretical e-infrastructure framing the complexity of challenges and demystifying the role of big data in soils security. OpenSoils is an open, elastic, provenance-oriented and lightweight computational open science e-infrastructure which rely on four overarching layers. Figure 2 illustrates the e-infrastructure, the layers and summarizes the data life-cycle of soil data (showed as arrows) (DEELMAN *et al.,* 2009; CRUZ, CAMPOS & MATTOSO, 2009; MATTOSO *et al.,* 2010).

(i) The end-users layer (*e.g.,* soil specialists, data managers, policy makers) uses on the web portal and mobile applications. They are used to collect and ingest new soil data directly from the fields into OpenSoilsDB using OpenSoils app or query data through the web portal aiding policy-makers to make decisions (DSS), and urban planners do envision new soils usage (PSS).

The specialists and researchers use this layer to handle data. The first can use mobile, IoT and web applications (*e.g.,* OpenSoils App and Wet Lab tools) to collect the data directly in the fields and trace the route of each soil sample collected and sent to the chemistry and

physics laboratories (*i.e.,* wet labs) to be further analyzed. Usually, each soil sample is submitted *in situ* by the specialists to morphological analyses. Thus, OpenSoils app sends raw data to the database. After that, each soil sample is tagged and shipped to laboratories where the scientist does (*in vitro*) wet experiments and further execute (*in silico)* computational scientific experiments with SisGExp (CRUZ & NASCIMENTO, 2016) which evaluate specific physic-chemical properties of each soil horizon.

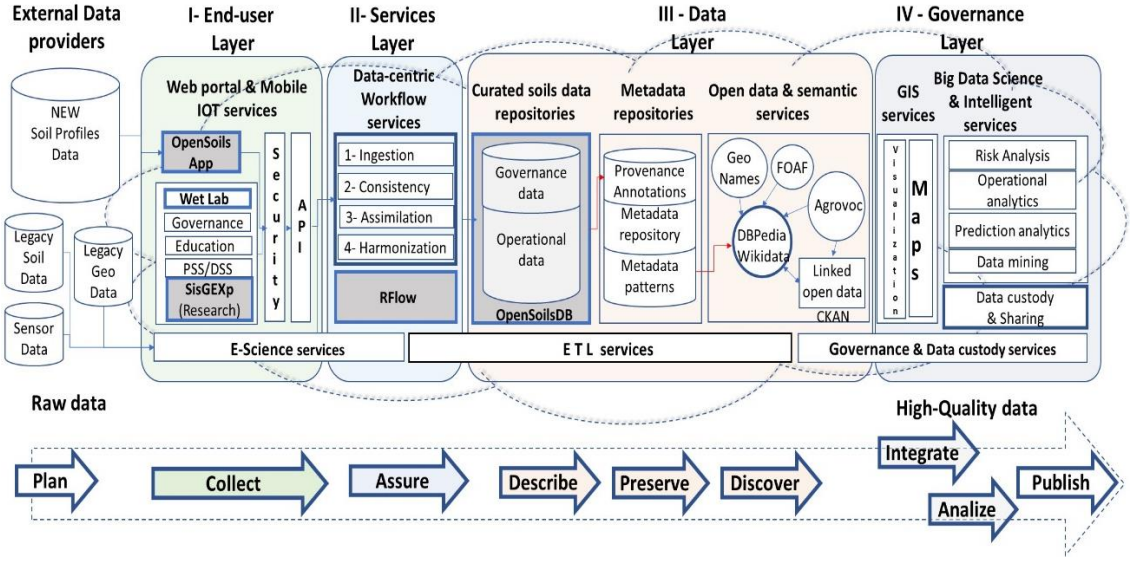

Figure 2 – Overview of the conceptual architecture of OpenSoils (the arrows describe data operations within the phases of life-cycle of soils investigations).

(ii) The services layer uses scientific and business models to generate curated data; they are composed of set data-centric scientific workflows (which ingest and analyses the consistency of the incoming of legacy soils data). RFlow is part of the layers (NASCIMENTO, 2015). It is a provenance-based approach that aid researchers to reproduce scientific experiments based on R scripts. RFlow manages, shares, and enacts the computational scientific workflows that encapsulate legacy R scripts it transparently captures provenance of R scripts and endows experiments reproducibility.

(iii) The data layer stores in the core of OpenSoils, it stores, describes, curates, various soils data sets, and metadata descriptors. The internal structure supports a diversified degree of data granularity and uses a relational database named OpenSoilsDB (former InfoSoilsBR, (RIZZO, CEDDIA & CRUZ, 2017). It can store new curated soils data annotated with provenance.

Much of the information needed to assure the data quality and to allow researchers to reproduce soils security experiments can be obtained by systematically capturing its provenance. Provenance refers to the record trail that accounts for the origin of a piece of data (FREIRE *et al.,* 2008). OpenSoilsDB can store workflow and scripts provenance. Workflow provenance consists of the record of the derivation of a result (*e.g.,* a soil profile, an image, a map) by a computational process represented as scientific workflows. Script provenance is obtained by analyzing the source code of soils security experiments represented as R scripts (PIMENTEL *et al.,* 2017). OpenSoilsDB uses W3C PROV-DM recommendation to store prospective and retrospective provenance for workflows and scripts (MOREAU & MISSIER, 2013). Besides, OpenSoilsDB supports FAIR guidelines (Findable, Accessible, Interoperable, and Reusable) for scientific data management and sharing (WILKINSON *et al.,* 2016).

The database also supports the ingestion of legacy soils data imported through ETL tools (*e.g.,* Pentaho/Kettle). The layer can store operational and governance data. Besides, to support open data we use CKAN (http://ckan.org/) which stores curated open data sets. Besides, CKAN is an international open data standard provides a streamlined way to make curated soils data publishable, usable, discoverable and interoperable by third-part soils applications. CKAN support data annotation with thesaurus ensuring semantic interoperability between computer systems, research teams or community users to exchange data with unambiguous meaning.

The thesaurus is used to semantically annotate soils data, allowing us to link it as RDF triples in DBpedia (2018), as depicted in Figure 2. The thesaurus used in the e-infrastructure is Agrovoc (CARACCIOLO *et al.,* 2013). Currently, Agrovoc is a SKOS-XL concept scheme published as Linked Open Data which covers all areas of interest of the Food and Agriculture Organization (FAO), including food, agriculture, environment. FAO publishes it; it is edited by a community of experts and consists of over 34,000 concepts available in 29 languages. It is used by researchers, librarians and information managers for indexing, retrieving and organizing data in agricultural information systems.

OpenSoilsDB database has two abstraction layers (*e.g.,* operational and governance). The lower operational layer aims to serve high quality-assessed, georeferenced soils profiles database to the Brazilian and international communities upon their standardization and harmonization. Each soil profile description recorded in the database has more than 40 entities, and 250 attributes to stores the soil properties and soil experiments (*e.g.,* mineralogical, morphological, chemical, physical and environmental data). Furthermore, the database support data versioning, data provenance, and stores georeferenced soil data as text and images about physic-chemical analytical data from each horizon and soil samples analyzed in wet laboratories.

The upper layer of the OpenSoilsDB improves the accessibility and reuse of soil data and knowledge. Data governance and data literacy are two important building blocks in the knowledge base of information professionals involved in supporting data-intensive research, and both address data quality and research data management. Adopting data governance in OpenSoils is advantageous because it is a service based on standardized, repeatable processes and is designed to enable the transparency of data-related processes and cost reduction. It refers to rules, policies, standards; decision rights; accountabilities and methods of enforcement.

(iv) The governance layers are composed by data management, data license, analytical and visualization tools and map generation services that can be connected to other software (*e.g.,* QGIS, ArcGIS, R, Tableau or sci-kit-learn) to generate analytical reports, soils prediction, raster maps to name a few.

Although received little attention in soils research communities, this layer is foundational for soils security. The prime function of the layer is to improve and maintain the quality of the soils dataset; thus, to be successful at governance, quality must be continuously measured, and the results continuously fed back by the data and services layers. We stress that this layer has roles of individuals. For instance, these individuals are the application owners, data custodians and application data architect, they are responsible for compliance with data standards, resolve data-related issues, share the soil datasets, and support enforcement of data/soil standards.

3.1 Daily uses of OpenSoils

OpenSoils was conceived as an e-infrastructure because refers to a combination and interworking of digitally-based software technologies, resources (data, services, digital repositories), communications (protocols and data access rights), and the people and organizational structures needed to support modern and collaborative research in soils security. OpenSoils has three primary uses:

(i) Offer diverse, integrated, timely and trustworthy digital repositories to researchers (*e.g.,* statistical studies of the quality of soils, soils mapping, evaluation of contamination by heavy metals and organic waste management system).

(ii) Offer tools to city planners, agronomists, farmers to make better decisions using high-quality harmonized open data (*e.g.,* studies to erosion, risk of landslides, risk of flooding, potential for agricultural use of soils; environmental and economic and ecological zoning, insurance of agronomic entreprises, land classification for irrigation; support in the recommendation of fertilizers and limestone).

(iii) Help students to increase their knowledge and skills about soils, the e-infrastructure is connected to the Brazilian Soils Museum at UFRRJ, where users can explore the collection of soil monoliths, soil artifacts, pictures and browse the data.

4. Related Work

Traditionally, soils security has operated along disciplinary lines in using and applying its data and analytical tools. Soils data management, curation, and governance is an issue that is still underestimated in soils sciences, with data being analyzed for isolated applications and with small groups of researchers working with isolated data silos on their personal computers and not properly sharing them (LOKERS et al., 2016). Today, there are no open science software platforms to support the full cycle of research in soils security. Thus, we conceived OpenSoils as an open e-infrastructure that than be used by the researcher, decision maker, data curator, city planner, farmer and students.

The investigations of soils security in Brazil and Latin America are still beginning. They are depicted as several isolated investigations and data silos about legacy soils data. For instance, BDSolos (BDSOLOS, 2018) is a relational database developed by EMBRAPA Solos that stores about 9.000 soils profiles. The database has no provenance nor metadata descriptors, besides there are no public interfaces to allow researchers to insert new soils data. Furthermore, the interfaces to query data are hard to be used even by soil specialists. Last but not least, there are no concerns about soils security nor map visualization facilities. We can point out the same limitations are shared by Fe.BR (FEBR, 2018). It is a single HTML website that stores the same type of data of BDSolos. The dataset is presented as a set of google docs resting in a virtual drive on the Web; their authors claim that it is open data. However, we stress that it fails to fulfill the eight Open Data principles (OGP, 2018), has no governance policies and unfortunately does not commit with the best web semantic practices (GYRARD *et al.,* 2015).

Fortunately, OpenSoils is entirely different from related works; it was conceived to adopt the open science, e-science, open data and data provenance emerging trends. First, it is a multi-disciplinary, community and data integrative e-infrastructure. Second, it supports the movement to make scientific experiments more reproducible and the publications and scientific data available as open access. Third, it can handle large amounts of data of soils security investigations. Fourth, it based on web, workflows services, and clouds infrastructure

which offer access to elastic and abundant resources that can be provisioned and de-provisioned on-demand.

5. Concluding Remarks

Conditions are now ripe for a comparable step change in the interplay between soils science and computer science, a change that will not only spur economic growth and competitive advantage, but also will help scientists to develop solutions to our societal challenges, understand climate change, and explore new frontiers of knowledge.

The soil has an integral part to play in the global environmental sustainability challenges. Nevertheless, there is still a lot of computational work needed to be fully developed in soil sciences. The growth of open science and the curated open soils databases may aid scientist to increase the reliability, robustness, and reproducibility of soils security experiments.

In this paper we presented OpenSoils, a novel e-infrastructure which provide knowledge about soils security to different kinds of users and not only researchers. The infrastructure enhances reproducibility and delivers high-quality soils datasets, knowledge and maps based on curated open data. OpenSoils is being developed; the mobile apps can be found at PET-SI Google Play and the further information about Wet Labs applications, the scientific workflows or ETL components can be found at http://www.opensoils.org.

As future work, we plan to finish the implementation of the e-infrastructure and investigate the alternative semantic relationships between soils data, digital objects and related domains to enhance solutions and improve data sharing, data curation, and long-term data stewardship policies.

Acknowledgments

This work was supported in part by the Brazilian funding agencies FNDE, PIBIC/CNPq and Petrobras. The author's thanks, PET-SI/UFRRJ, MEC/SESU, Reds CYTED – BigDSSAgro and SmartLogistcs@IB.

References

Arrouays, D. et al., Soil legacy data rescue via GlobalSoilMap and other international and national initiatives. GeoResJ 14, pages 1-19, 2017.

Baker, M., 1,500 scientists lift the lid on reproducibility. Nature. 533:7604, 2016.

Buneman, P., Khanna, S., Tan, W-C. Data Provenance: Some Basic Issues. In: Kapoor S., Prasad S. (eds) FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science. FSTTCS 2000. Lecture Notes in Computer Science, vol 1974. Springer, Berlin, Heidelberg.

BDSolos, Banco de Dados de Solos. https://www.bdsolos.cnptia.embrapa.br/consulta_publica.html, (acessado em 9.3.2018).

Caracciolo, C. et al. The AGROVOC Linked Dataset. Semantic Web, 4, 3, pages. 341-348. 2013.

Cruz, S.M.S, Campos, M. L. M and Mattoso, M. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. In: SERVICES I, pages. 259-266, USA, 2009.

Cruz, S.M.S, Nascimento, J.A.P. SisGExp: Rethinking Long-Tail Agronomic Experiments. IPAW 2016.

DBPedia, http://wiki.dbpedia.org/ (acessado em 24.3.2018).

Deelman, E. et al., Workflows and e-Science: An overview of workflow system features and capabilities. Future Generation Computer Systems, 25:5, pages. 528–540, 2009.

Fanelli, D. Opinion: Is science really facing a reproducibility crisis, and do we need it to? Proceedings of the National Academy of Sciences of the USA, March 2018.

FeBR, Repositório de dados de solos. http://coral.ufsm.br/febr/, (acessado em 9.3.2018).

Freire, J., Koop, D., Santos, E., Silva C.T. Provenance for Computational Tasks: A Survey. Computing in Science and Engineering, 10:3, pages 11–21, 2008.

Freire, J. Chirigati, F. Provenance and the Different Flavors of Computational Reproducibility. IEEE Data Engineering Bulletin, 41(1), pages. 15-26, 2018.

Fecher, B. and Friesike, S. Open Science: One Term, Five Schools of Thought. Opening Science, pages. 17-47, 2013.

Foster, E. D., Deardorff, A. Open Science Framework (OSF). J Med Libr Assoc. 105:2, pages. 203–206. 2017.

Gyrard, A., Serrano, M., Atemezing G. A. Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things. 2nd IEEE World Internet of Things, 2015.

Hey, T., Tansley, S., Tolle, K. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.

Hutson, M., Artificial intelligence faces reproducibility crisis. Science, 359: 6377, pp. 725-726, 2018.

Koch, A. et al., Soil Security: Solving the Global Soil Crisis. Global Policy, 4:4 ages 434-441. 2013.

Lockers, R. et al., Analysis of Big Data technologies for use in agro-environmental science. Environmental Modelling & Software, 84, pp. 494-504, 2016.

Mattoso, M. et al., Towards supporting the life cycle of large-scale scientific experiments. International Journal of Business Process Integration and Management, 5:1, pages 79-92, 2010.

McBratney, A., Field, D. J and Koch, A. The dimensions of soil security. Geoderma. 213, pages 203-213, 2014.

Melo, A. A. B. et al., Spatial distribution of organic carbon and humic substances in irrigated soils under different management systems in a semi- Arid zone in Ceará, Brazil. SEMINA: CIENCIAS AGRARIAS, 37:4, pages 1845-1856, 2016.

Moreau. L., Missier, P. PROV-DM: The PROV Data Model. https://www.w3.org/TR/prov-dm/ (acessado em 24.3.2018).

Munafò, M. Open Science and Research Reproducibility. Ecancer medical science. 10, ed56. 2016.

Nascimento, J. A. P. RFLOW: uma arquitetura para execução e coleta de proveniência de workflows estatísticos. Dissertação de Mestrado, UFRRJ, 2015.

Neves V. C. et al., Managing Provenance of Implicit Data Flows in Scientific Experiments. ACM ACM Transactions on Internet Technology. Volume 17 Issue 4, Article No. 36, 2017.

OGP, Open Government Partnership, 2017. https://www.opengovpartnership.org/countries/brazil (acessado em 9.3.2018).

Pansu, M., Gautheyrou, J., Handbook of Soil Analysis, Springer, 2006.

Pimentel, J. F et al., noWorkflow: a Tool for Collecting, Analyzing, and Managing Provenance from Python Scripts 2017. Proceedings of the VLDB. vol 10:12, pages 1841-1844, 2017.

Rizzo, G.S.C, Ceddia, M. B., Cruz, S. M. S. Banco de Dados Pedológico: Primeiros Estudos. V RAIC – UFRRJ, 2017.

Wilkinson, M. D. et al., The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3, Article number: 160018, 2016.

Worlfert, S. et al., Big Data in Smart Farming – A review. Agricultural Systems, v. 153, pages 69-80, 2017.

Yamson, D. O., et al., Putting Soils Security on the Policy Agenda: Need for a Familiar Framework. Challenges. 4:2 15 pages. 2016.

van Es, H., A New Definition of Soil CSANews 62:20-21, 2017.

Uma análise sobre as bulas de medicamentos no Brasil

Alexandre Martins da Cunha[1,2], **Gabriel Nascimento**[1], **Gustavo Paiva Guedes**[1]

[1]CEFET/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
Av. Maracanã, 229 - Rio de Janeiro - RJ - Brasil.

[2]UFF - Universidade Federal Fluminense
Rua Professor Marcos Waldemar de Freitas Reis, s/n - Niterói - RJ - Brasil

{alexandre.cunha,gabriel.nascimento}@eic.cefet-rj.br,
gustavo.guedes@cefet-rj.br

Abstract. *The World Health Organization states that a certain level of self-medication may be acceptable, as long as it occurs responsibly. In this aspect, the package insert may aid in clarifying the patient's illness and treatment. However, several studies highlight problems presented in the package inserts. The present study aims to contribute by analyzing the current scenario of drug inserts in Brazil. The results show that the leaflets present on the Internet are poorly homogeneous and omit side effects. The analysis performed on a couple of package inserts on the ANVISA website indicates the existence of very technical terms and some omissions.*

Resumo. *A Organização Mundial de Saúde destaca que um certo nível de automedicação pode ser aceitável, desde que ocorra de maneira responsável. Nesse aspecto, a bula do medicamento pode auxiliar no esclarecimento quanto à enfermidade e tratamento do paciente. Entretanto, diversos estudos destacam problemas apresentados nas bulas de medicamentos. O presente estudo tem o objetivo de contribuir analisando o atual cenário das bulas de medicamentos no Brasil. Os resultados indicam que as bulas presentes na internet são pouco homogêneas e omitem eventos adversos. A análise efetuada em um par de bulas do site da ANVISA indica a existência de termos muito técnicos e algumas omissões.*

1. Introdução

O Brasil é o quinto país do mundo em número de buscas por orientação médica pela internet com o objetivo de diagnóstico e automedicação [de Oliveira et al. 2013]. A busca por informações sobre medicamentos e tratamentos médicos em sites de busca e redes sociais cresce em larga escala, o que causa preocupação para os profissionais da área de saúde [Silva and Castro 2008]. Essa preocupação é decorrente da utilização de medicamentos sem orientação médica, o que é um perigo iminente à saúde, sendo considerado um problema mundial [de Aquino et al. 2010]. Alguns problemas oriundos da automedicação são: aumento do erro no diagnóstico da doença, aparecimento de efeitos indesejáveis graves, reações alérgicas e utilização de dosagem insuficiente ou excessiva [Organization et al. 2000]. Além disso, pode mascarar os verdadeiros sintomas, comprometendo o tratamento adequado [Stimmel 1983].

O cenário apresentado indica que o médico deve ser a principal fonte de informação ao paciente [Silva et al. 2000]. No entanto, a Organização Mundial de Saúde (OMS) destaca que um certo nível de automedicação pode ser aceitável, desde que ocorra de forma responsável [Organization et al. 2000]. Nesse aspecto, a bula do medicamento pode auxiliar no esclarecimento quanto à enfermidade e tratamento do paciente, e portanto, é a principal fonte de informação depois do médico [Silva et al. 2000]. A presença da bula é obrigatória nos medicamentos, no entanto, apenas a inclusão da bula não é suficiente, o importante é que os pacientes sejam capazes de compreender as informações lá descritas [Da Silva et al. 2006].

Na tentativa de fornecer informações mais homogêneas aos pacientes, a Agência Nacional de Vigilância Sanitária (ANVISA) criou uma regulamentação pela portaria 110/97[1]. Assim, as bulas de medicamentos devem conter obrigatoriamente informações de prazo de validade, cuidados de armazenamento, posologia, eventos adversos, dentre outras. No entanto, alguns estudos destacam que os indivíduos ainda possuem dificuldades na leitura das bulas, fazendo parte das principais reclamações "a linguagem muito científica" e a "excessiva quantidade de informações" [Da Silva et al. 2006, Paula et al. 2009]. Além disso, as informações nas bulas não são homogêneas, sendo notadas omissões de interações medicamentosas, erros de nomenclatura e omissão de eventos adversos [Korolkovas et al. 2006].

O cenário acima apresentado destaca diversos problemas existentes em bulas provenientes do site da ANVISA e de bulários na internet. Os bulários da internet recebem ênfase por, de maneira geral, apresentarem notória baixa qualidade [Eysenbach et al. 2002]. Embora os estudos apresentados até aqui relatem problemas já existentes entre os anos de 2000 e 2010, cabe a esse estudo uma investigação inicial sobre o atual cenário referente aos bulários na internet e às bulas existentes no site da ANVISA.

Nesse panorama, as contribuições desse estudo são: (*i*) construção de um conjunto de dados proveniente de um de bulário na internet, visto que não foram encontrados conjuntos de dados públicos de sites de bulários em português do Brasil; (*ii*) análise os eventos adversos de diferentes bulas do conjunto de dados proposto em *(i)*, de forma a verificar sua homogeneidade (e.g., omissões de eventos adversos e excesso de termos científicos); (*iii*) análise manualmente um par de bulas extraído do site da ANVISA, também no que se refere a homogeneidade.

O restante desse artigo está organizado da seguinte forma: a Seção 2 descreve trabalhos relacionados ao contexto de qualidade de bulários; a Seção 3 detalha o conjunto de dados produzido; a Seção 4 apresenta a metodologia utilizada para analisar os eventos adversos das bulas; a Seção 5 descreve os resultados alcançados no presente estudo e por fim, a Seção 6 conclui e apresenta alguns cenários de trabalhos futuros.

2. Trabalhos Relacionados

O estudo produzido em [Silva et al. 2000] se preocupa em identificar a adequação do conteúdo e da forma das bulas de medicamentos com relação a seção de *Informação ao paciente*. Dada a relevância da bula como referência escrita para pacientes, foram selecionadas as bulas dos medicamentos mais prescritos no ambulatório do Hospital de Clínicas

[1]Portaria nº 110, de 10 de março de 1997 – disponível em http://www.anvisa.gov.br

de Porto Alegre. Em seguida, foi realizada uma pesquisa para avaliar a existência de frases padronizadas exigidas pela portaria 110/97[2]. Também foi apresentado um formulário a alguns pacientes para a avaliação da compreensão do conteúdo das bulas, considerando paciente todo indivíduo sem formação técnico-científica. A avaliação contou com 48 bulas de 26 laboratórios, disponíveis em junho de 1998 em três farmácias de redes distintas, localizadas em Porto Alegre. Os resultados do estudo evidenciam que nenhuma das bulas atendeu, em sua totalidade, às informações exigidas por lei.

O estudo realizado em [Eysenbach et al. 2002] realiza um revisão sistemática para avaliar a dificuldade de se encontrar informações de qualidade em saúde na internet. Essa revisão utiliza como base de dados os *websites MEDLINE*, *PREMEDLINE*, *Science Citation Index*, *Social Sciences Citation Index*, *Arts and Humanities Citation Index*, *LISA*, *CINAHL*, *PsychINFO*, *EMBASE*, *SIGLE*, além de buscas pela internet, acervo pessoal e manuais. A revisão sistemática totalizou 79 estudos, 5941 sites de saúde e 1329 páginas da Web. O objetivo principal consistiu em analisar o processo de avaliação da "qualidade" de informações na área de saúde na internet. Os resultados alcançados indicam ser necessária uma padronização de critérios de qualidade, dada as diferentes abordagens adotadas nos diferentes estudos. No entanto, destacam que o risco que os indivíduos possuem de encontrar sites inadequados é uma função da proporção de informação inadequada na internet sobre a inabilidade dos indivíduos filtrarem os sites relevantes.

O estudo desenvolvido por [Segura-Bedmar and Martínez 2017] apresenta uma proposta para melhorar o vocabulário contido nas bulas de maneira automática. Para isso, foram utilizados recursos da área de PLN (Processamento de Linguagem Natural). O estudo utiliza *word embeddings* (em português, vetores de palavras), que são representações vetoriais capazes de capturar o valor semântico das palavras. O objetivo é encontrar o sinônimo mais simples para cada termo, utilizando como conjunto de dados o EasyDPL (Easy Drug Package Leaflets), que conta com 306 bulas e 1400 eventos adversos e sinônimos [Segura-Bedmar et al. 2016]. Os experimentos apresentam precisão de $38,5\%$ na detecção de sinônimos.

O estudo desenvolvido neste trabalho se assemelha aos supra-citados por avaliar as informações presentes nas bulas de medicamentos. No entanto, se concentra na avaliação da homogeneidade dos eventos adversos tanto em *websites* quanto no site da ANVISA. Mais especificamente, se difere dos dois últimos por ser realizado em bulas na língua portuguesa.

3. Construção do conjunto de dados

A construção do conjunto de dados de bulas da internet foi conduzida da seguinte maneira. Inicialmente, foi efetuada uma busca no motor de busca do Google®[3] utilizando a *string* "bulas *online*". Essa busca retornou milhares de resultados. No entanto, nos concentramos no primeiro resultado, ou seja, no site *bulário.com*.

Dessa maneira, foram extraídos 1.564 textos de bulas do site *bulario.com*. Os textos compreendem um total de 1.375.684 palavras e o número de termos únicos corresponde a 32.180. O conjunto de dados se encontra nos formatos .xlsx e .json e pode ser

[2]Portaria nº 110, de 10 de março de 1997 – disponível em http://www.anvisa.gov.br
[3]Buscador - Google® - www.google.com.br

encontrado em *https://github.com/LaCAfe/Bulario2018PT-br*. Esse conjunto de dados é denominado `Bulario2018PT-br`.

4. Metodologia de análise

A metodologia de análise do presente estudo foi dividida em duas fase. A primeira fase consiste em analisar as bulas do conjunto de dados `Bulario2018PT-br` para evidenciar a homogeneidade das bulas presentes na internet. Dessa maneira, foi selecionada aleatoriamente uma bula desse conjunto de dados. Em seguida, foi selecionada aleatoriamente uma outra bula com o mesmo princípio ativo da primeira. Posteriormente, foi realizada uma análise manual dos eventos adversos em cada uma das bulas.

Ainda na primeira fase, foram selecionados aleatoriamente seis pares de bulas do `Bulario2018PT-br` para análise com o coeficiente de Jaccard. A Eq. 1 apresenta o coeficiente de Jaccard, em que $0 \leq J(doc_1, doc_2) \leq 1$. Essa medida corresponde à proporção do número de palavras em comum entre os documentos e o número de palavras únicas em ambos os documentos. Primeiramente, foi realizado um pre-processamento utilizando um *script* em Python 3, removendo a acentuação e convertendo as duas sequências de texto (*strings*) para letras minúsculas. Em seguida, o *script* calcula o coeficiente de Jaccard para essas duas *strings*.

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2} \tag{1}$$

A segunda fase consiste em analisar os eventos adversos em um par de bulas do site da ANVISA. Para isso, foi selecionada aleatoriamente uma bula de medicamento e, em seguida, uma outra bula com o mesmo princípio ativo. Posteriormente, foi efetuada uma análise manual dos termos em cada uma das bulas.

5. Resultados

5.1. Análise de bulas provenientes da internet

A análise das bulas provenientes da internet foi efetuada com base no conjunto de dados `Bulario2018PT-br`. Foi selecionado aleatoriamente o medicamento Alprazolam da Medley e em seguida, o medicamento Apraz® do laboratório Hypermarcas. A Tabela 1 exibe os eventos adversos de cada um desses medicamentos. Para facilitar a compreensão, a tabela foi subdividida em três partes. Na primeira parte, são exibidos os termos em comum em ambas às bulas e, para isso, foram considerados os eventos adversos sinônimos (e.g., "perda de apetite" na bula do Alprazolam e "diminuição do apetite" na bula do Apraz). Na segunda parte, são exibidos os eventos adversos que existem na bula do Alprazolam e não existem na bula do Apraz. Por fim, na terceira parte, são apresentados os eventos adversos que existem na bula do Apraz e não existem na bula do Alprazolam.

Tabela 1. "Eventos Adversos"dos medicamentos Alprazolam e Apraz®

Alprazolam [Bulário.com 2011a]	**Apraz® [Bulário.com 2011b]**
depressão, sonolência, perda de memória, dificuldade em controlar os movimentos do corpo, dificuldade em falar, tontura ou dor de cabeça, prisão de ventre, perda de apetite, diminuição da libido, sensação de cabeça vazia	depressão, sonolência, alterações na memória, falta de coordenação motora, fala lentificada e difícil de compreender, tontura, dor de cabeça, prisão de ventre, diminuição do apetite, diminuição da libido, sensação de cabeça vazia
diarreia, aumento da líbido, problemas de fígado, alterações do equilíbrio, aumento do batimento cardíaco, problemas de atenção	- - -
- - - - - - -	sedação, secura na boca, cansaço extremo, irritabilidade, confusão, confusão mental, ansiedade, dificuldade para dormir, nervosismo, alterações no do equilíbrio, problemas de atenção, aumento do sono, lentidão, tremor, visão embaçada, náusea, inflamação na pele, impotência sexual, diminuição do peso ou aumento do peso

É possível constatar que diversos eventos adversos são omitidos em ambas as bulas. Além disso, pode-se notar a não-homogeneização dos termos, visto que sinônimos são utilizados para descrever os mesmos eventos. Para apresentar uma análise mais precisa, foram analisados seis pares de bulas, selecionadas aleatoriamente, utilizando o coeficiente de Jaccard, conforme apresentado na Tabela 2. A média do coeficiente de Jaccard é indicada na última linha e corresponde a $0,27$.

Medicamento 1	Medicamento 2	Jaccard (J)
Dipirona	Novalgina	$0,13$
Flanax	Naproxeno	$0,31$
Histadin	Loratadina	$0,21$
Doralgina	Neosaldina	$0,32$
Clonazepam	Rivotril	$0,31$
Exodus	Espran	$0,35$
Média		$0,27$

Tabela 2. Análise dos eventos adversos em seis bulas do *bulario.com*.

5.2. Análise de bulas provenientes do site da ANVISA

A análise das bulas provenientes do site da ANVISA foi efetuada com base na seleção aleatória de um nome de medicamento a partir do `Bulario2018PT-br`. O medicamento selecionado foi o Citalopram (Medley). Em seguida, foi selecionado um outro medicamento com o mesmo princípio ativo (i.e., Bromidrato de Citalopram, do laboratório Nova Química). Os eventos adversos de cada um dos medicamentos é exibido na Tabela 3, que foi subdividida em quatro áreas, conforme a categoria das reações nas bulas:

reações muito comuns (RMC), reações comuns (RC), reações incomuns (RI) e reações raras (RR). Os eventos adversos sinônimos foram exibidos em negrito. A ausência de um efeito colateral em qualquer uma das bulas foi representada como "—".

Tabela 3. Eventos adversos dos medicamentos: Citalopram® [ANVISA 2011a] e Bromidrato de Citalopram (Nova Química). [ANVISA 2011b]

Tipos de Reações	Citalopram (Medley)	Bromidrato de Citalopram (Nova Química)
RMC	***náusea***, boca seca, ***insônia***, sonolência, ***aumento da sudorese***.	***náusea (enjoo)***, boca seca, ***dificuldades para dormir***, sonolência; ***aumento do suor***.
RC	diminuição do apetite, agitação, ***diminuição da libido***, ansiedade, nervosismo, ***confusão***, sonhos anormais, tremores, fadiga, tontura, ***parestesia***, diarreia, vômitos, ***diminuição do peso***, ***tinitus***, ***alterações da ejaculação, impotência, falha da ejaculação***, ***orgasmo anormal em mulheres***, distúrbio de atenção, constipação, ***prurido***, ***artralgias***, ***mialgias***, bocejo, **—**.	diminuição do apetite, agitação, ***diminuição do desejo sexual***, ansiedade, nervosismo, ***sentir-se confuso***, sonhos anormais, tremores, fadiga, tonturas, ***formigamento na pele (parestesia)***, diarreia, vômitos, ***perda de peso***, ***tinitus (zumbido no ouvido)***, ***homens podem apresentar problemas de ereção e ejaculação***, e ***mulheres podem apresentar dificuldade para chegar ao orgasmo***, distúrbios de atenção, constipação, ***coceira (prurido)***, ***dores musculares e nas juntas***, bocejos, formigamento ou dormência nas mãos ou nos pés.
RI	aumento do apetite, ***fotossensibilidade***, ***taquicardia***, ***bradicardia***, agressividade, alucinações, ***mulheres: menorragia***, ***edema***, ***eritema (rash)***, despersonalização, ***síncope***, ***retenção urinária***, ***alopecia***, ***púrpura***, urticária, ***midríase (que pode levar ao glaucoma agudo de ângulo fechado)***, mania, aumento do peso.	aumento do apetite; ***sensibilidade à luz***, ***alteração (aumento ou diminuição) dos batimentos cardíacos***, agressividade, alucinação, ***sangramento menstrual excessivo***, ***inchaços nos braços ou pernas (edema)***, ***erupções cutâneas (rash)***, despersonalização, ***desmaio***, ***dificuldade para urinar***, ***perda de cabelo***, ***manchas roxas (púrpura)***, urticária, ***pupilas aumentadas (midríase)***, mania, **—**.
RR	***convulsão de grande mal***, alterações no paladar, hepatite, ***discinesia***, ***hemorragia***, hiponatremia, pirexia.	***convulsões***, alterações do paladar, hepatite, ***movimentos involuntários dos músculos***, ***sangramento***, hiponatremia, **—**.

Embora a Tabela 3 indique que alguns eventos adversos foram omitidos em ambos os medicamentos, a bula do Bromidrato de Citalopram (Nova Química) traz um aviso em negrito, informando ao paciente que se o mesmo apresentar algum dos eventos adversos abaixo, deve parar o tratamento e procurar o médico imediatamente: (a) febre alta;

(b) agitação; (c) confusão; (d) movimentos involuntários dos músculos; (e) pele, língua, lábios ou face inchadas; (f) dificuldades em respirar ou engolir; (g) sangramentos não usuais, incluindo hemorragias gastrointestinais.

Ainda analisando a Tabela 3, pode-se notar que são utilizados sinônimos para se referir ao mesmo evento adverso (e.g., síncope e desmaio). Além disso, foram observados diversos termos técnicos (e.g., tinitus, parestesia) no Citalopram (Medley) sem a devida associação com a sua descrição. Também foi evidenciada a ausência de alguns eventos adversos (e.g., pirexia, aumento do peso). Por fim, foram encontrados diversos eventos adversos descritos de maneira diferente nas duas bulas. Por exemplo, no Bromidrato de Citalopram (Nova Química) encontramos a descrição "mulheres podem apresentar dificuldade para chegar ao orgasmo" que se difere de "orgasmo anormal em mulheres", conforme descrito na bula do Citalopram (Medley).

6. Conclusão

Neste estudo buscou-se investigar a situação atual da seção "eventos adversos" das bulas encontradas em *websites*. Para isso, foi desenvolvido um conjunto de dados contendo 1.564 bulas do site *www.bulario.com*. As análises realizadas com esse conjunto de dados indicam problemas na nomenclatura e na homogeneidade das bulas, mesmo com as restrições impostas pela portaria 47/2009[4], que rege, entre outros itens, a homogeneização das bulas. Quase uma década após a vigência da portaria, ainda é possível encontrar bulários defasados. Uma análise minuciosa foi efetuada nos medicamentos Alprazolam e o Apraz, encontrados no *bulário.com*. Esses medicamentos exibem diversas omissões e falta de homogeneização na descrição dos eventos adversos. Isso exige bastante atenção, dado que esses medicamentos são rotulados como *tarja preta*. O *bulario.com* informa que, entre outras fontes, utiliza o site da ANVISA como fonte de informação.

A análise efetuada em duas bulas de medicamentos provenientes do bulário da ANVISA evidenciaram a não-homogeneização das bulas, sendo por omissões de eventos adversos ou por eventos descritos ora no singular, ora no plural. Também foram encontrados diversos termos técnicos sem suas descrições, como por exemplo, *hiponatremia*, *artralgias* e *mialgias*. Também é importante destacar que apenas uma das bulas trouxe o aviso sobre a interrupção do tratamento caso fossem observados alguns eventos adversos.

Esse estudo abre margem para novas pesquisas que avaliem outros bulários e outros pares de bulas provenientes do site da ANVISA. Também evidencia a necessidade de uma maior homogeneização dos termos presentes nas bulas. Essa homogeneização pode ser efetuada com a criação de uma ontologia para os eventos adversos na língua portuguesa, similar à ontologia encontrada para a língua inglesa [He et al. 2014]. Esse cenário pode trazer benefícios aos pacientes e, consequentemente, à saúde pública.

Referências

ANVISA (2011a). Bula do medicamento citalopram∗®. Disponível em: `http://www.anvisa.gov.br/datavisa/fila_bula/frmVisualizarBula.asp?pNuTransacao=11062252015&pIdAnexo=3010130`. Data do Acesso: 08 Mar. 2018.

[4]Portaria nº 47, de 8 de setembro de 2009 – essa portaria se encontra disponível no endereço http://portal.anvisa.gov.br/documents/33836/2814380/RDC+47+09.pdf

ANVISA (2011b). Bula do medicamentoprocimax∗®. Disponível em: http://www.anvisa.gov.br/datavisa/fila_bula/frmVisualizarBula.asp?pNuTransacao=9983572015&pIdAnexo=2948521. Data do Acesso: 08 Mar. 2018.

Bulário.com (2011a). Bula do medicamento alprazolam. Disponível em: https://www.bulario.com/alprazolam. Data do Acesso: 08 Mar. 2018.

Bulário.com (2011b). Bula do medicamento apraz®. Disponível em: https://www.bulario.com/apraz_comprimidos. Data do Acesso: 08 Mar. 2018.

Da Silva, M., Almeida, A. E. d., Oliveira, A., Correia, C., Benzatti, F., Fernandes, J., Barbosa, G., Pimenta, C., Costa, T., and Doneida, V. (2006). Estudo da bula de medicamentos: uma análise da situação. *Revista de Ciências Farmacêuticas Básica e Aplicada*, pages 229–236.

de Aquino, D. S., de Barros, J. A. C., and da Silva, M. D. P. (2010). A automedicação e os acadêmicos da área de saúde. *Revista Ciência & Saúde Coletiva*, 15(5).

de Oliveira, F., Goloni-Bertollo, E. M., and Pavarino, É. C. (2013). A internet como fonte de informação em saúde. *Journal of Health Informatics*, 5(3).

Eysenbach, G., Powell, J., Kuss, O., and Sa, E.-R. (2002). Empirical studies assessing the quality of health information for consumers on the world wide web: a systematic review. *Jama*, 287(20):2691–2700.

He, Y., Sarntivijai, S., Lin, Y., Xiang, Z., Guo, A., Zhang, S., Jagannathan, D., Toldo, L., Tao, C., and Smith, B. (2014). Oae: the ontology of adverse events. *Journal of biomedical semantics*, 5(1):29.

Korolkovas, A., França, F. F. d. A. C., et al. (2006). Dicionário terapêutico guanabara. In *Dicionário terapêutico guanabara*.

Organization, W. H. et al. (2000). Guidelines for the regulatory assessment of medicinal products for use in self-medication.

Paula, C. d. S., Costa, C. K., Miguel, M. D., Zanin, S. M., and Spinillo, C. G. (2009). Análise crítica de bulas sob a perspectiva do usuário de medicamentos. *Visão Acadêmica*, 10(2).

Segura-Bedmar, I. and Martínez, P. (2017). Simplifying drug package leaflets written in spanish by using word embedding. *Journal of biomedical semantics*, 8(1):45.

Segura-Bedmar, I., Núnez-Gómez, L., Fernández, P. M., and Quiroz, M. (2016). Simplifying drug package leaflets. In *SMBM*, pages 20–28.

Silva, E. V. d. and Castro, L. L. C. d. (2008). A internet como forma interativa de busca de informação sobre saúde pelo paciente.

Silva, T. d., Dal-Pizzol, F., Bello, C. M., Mengue, S. S., and Schenkel, E. P. (2000). Bulas de medicamentos e a informação adequada ao paciente. *Revista de Saúde Pública*, 34:184–189.

Stimmel, G. (1983). Political and legal aspects of pharmacist prescribing. *American Journal of Health-System Pharmacy*, 40(8):1343–1344.

Uma Plataforma Computacional para a Construção de Bancos de Dados para Experimentos de Neurociência*

Kelly Rosa Braghetto[1,2], Evandro Santos Rocha[1], Carlos Eduardo Ribas[1], Cassiano Reinert Novais dos Santos[1], Sueli dos Santos Rabaça[1], Margarita Ruiz Olazar[1]

[1]Centro de Pesquisa, Inovação e Difusão em Neuromatemática

[2]Departamento de Ciência da Computação - Instituto de Matemática e Estatística
Universidade de São Paulo – SP – Brazil

{kellyrb,erocha,ribas,cacorns,suelisr,mrolazar}@ime.usp.br

Resumo. *Dados científicos abertos são fundamentais para se ter uma ciência de melhor qualidade e de maior impacto. A criação de bancos de dados científicos abertos envolve vários desafios, como a criação de representações padronizadas para os dados e metadados de diferentes domínios do conhecimento e o desenvolvimento de recursos computacionais para auxiliar os cientistas na coleta e manutenção de dados de qualidade. Este artigo apresenta uma plataforma de software livre para o gerenciamento e compartilhamento de dados de experimentos em Neurociência. Essa plataforma permite registrar os dados e metadados de experimentos de forma segura e amigável, integrando registros de dados de diferentes tipos, como clínico, eletrofisiológico e comportamental.*

Abstract. *Open scientific data is fundamental to support better quality and higher impact reproducible science. The creation of open scientific databases involves a number of challenges, such as the creation of standardized representations of data and metadata from different domains of knowledge, as well as the development of computational resources to assist scientists in collecting and maintaining high-quality data. This paper presents a free software computational platform for the management and sharing of data from Neuroscience experiments. This platform allows to register data and metadata of experiments in a safe and user-friendly way, integrating data records of different types, such as clinical, electrophysiological and behavioral.*

1. Introdução

Com o aumento do uso de computação nas mais variadas áreas da ciência, os dados têm assumido um papel cada vez mais importante no processo de descoberta científica. Muitos dos resultados científicos que são divulgados hoje são embasados por dados digitais coletados ou gerados em experimentos científicos. Logo, é imprescindível que esses dados sejam confiáveis e estejam publicamente acessíveis, para que os resultados possam ser validados e reproduzidos. Apesar disso, em muitos domínios da ciência, como é o

*Esta plataforma foi desenvolvida como parte das atividades do Centro de Pesquisa, Inovação e Disseminação em Neuromatemática, financiado pela FAPESP (número do processo: 2013/07699-0). O trabalho também recebeu financiamento do CNPq (número do processo: 426579/2016-0).

caso da Neurociência, a disponibilização pública de dados de experimentos científicos ainda não é a regra, mas sim a exceção. A coleta de dados em experimentos é um trabalho difícil e dispendioso e ainda pouco reconhecido pela comunidade científica.

A representação e o armazenamento digital de dados científicos envolve diversos desafios. O projeto e a execução de um experimento científico inclui várias etapas, nas quais os seus parâmetros e a sua estrutura são definidos. No domínio da Neurociência, mais particularmente, caracterizar um experimento não é uma tarefa trivial. Existem diferentes tipos de experimentos (e.g., comportamentais, cognitivos, eletrofisiológicos e de neuroimagens), uma grande variabilidade na estrutura dos processos experimentais e uma alta heterogeneidade de formatos de dados coletados.

Para que um cientista possa fazer uma análise correta dos dados de um experimento em Neurociência ou seja capaz de reproduzi-lo, ele precisa conhecer informações sobre o processo experimental completo, ou seja, sobre como os dados foram coletados ou gerados. Além disso, há outras informações "ortogonais" ao processo experimental que também são indicadores importantes da qualidade dos dados coletados. Como exemplo, pode-se citar informações sobre o laboratório onde o experimento foi realizado, sobre os profissionais responsáveis pelo experimento e pelas coletas de dados e até mesmo publicações ou outros resultados decorrentes do experimento. Os dados sobre o processo experimental mais as informações ortogonais à realização de um experimento podem ser entendidos como *metadados* ou *dados de proveniência* dos dados experimentais.

1.1. Bancos de Dados na Neurociência

A construção, manutenção e curadoria de bancos de dados públicos em Neurociência é hoje considerada fundamental para um avanço mais efetivo na compreensão do funcionamento do cérebro e no tratamento de suas patologias. Esse movimento surgiu de maneira mais sistemática na área a partir da década de 90 [Chicurel 2000, Koslow 2000, Koslow 2002], quando se deu a primeira grande iniciativa de compartilhamento de dados coletados a partir de medidas de ressonância magnética funcional – o *International Consortium for Brain Mapping*[1]. Várias iniciativas de compartilhamento de dados de diferentes tipos têm sido implementadas desde então, principalmente em consórcios e grandes projetos como *Human Brain Project*[2], *International Neuroinformatics Coordinating Facility* (INCF)[3] e *Brain Research and Integrative Neuroscience Network* (BRAINnet)[4].

Apesar do crescente interesse na área, muitos dos bancos de dados de Neurociência disponíveis na atualidade possuem deficiências que dificultam o reúso dos dados e inviabilizam a aplicação de procedimentos computacionais para a descoberta automática de novos conhecimentos. Dentre essas deficiências, é possível destacar [Kötter 2001]: (i) dados de baixa qualidade, inconsistentes ou incompletos; (ii) bancos de dados que são "federações" de conjuntos heterogêneos de dados (com qualidades e estruturas diferentes e sem uma visão unificada dos dados); (iii) bancos de dados que são públicos mas não completamente abertos, ou seja, que impõem restrições ao acesso e ao reúso dos dados.

[1]International Consortium for Brain Mapping – http://www.loni.usc.edu/ICBM/
[2]Human Brain Project – https://www.humanbrainproject.eu
[3]International Neuroinformatics Coordinating Facility – https://www.incf.org/
[4]BRAINnet – http://www.brainnet.net/

1.2. Representação e Armazenamento de Dados em Neurociência

Muitos cientistas armazenam digitalmente os dados de seus experimentos como arquivos comuns, mantidos no sistema de arquivos de um computador. Os dados de proveniência, quando digitalizados, acabam virando arquivos texto (com dados não estruturados) ou planilhas sem uma estrutura padronizada. Essa forma de armazenamento dificulta a manutenção, a recuperação, o compartilhamento e o reúso dos dados.

Apesar da ausência de padrões, já existem iniciativas relacionadas à criação de diretrizes que definem quais são os dados que um pesquisador precisa reportar quando publica os resultados de um experimento em Neurociência. Exemplos de diretrizes desse tipo são a MINI (*Minimum Information about a Neuroscience Investigation*) [Gibson et al. 2009], a MINEMO (*Minimal Information for Neural Electromagnetic Ontologies*) [Frishkoff et al. 2011] e a *Guidelines for reporting an fMRI study* [Poldrack et al. 2008]. Essas *check lists* em geral apontam as informações que são consideradas importantes para a análise dos dados coletados e para a compreensão do experimento realizado. Entretanto, essas informações podem não ser suficientes para apoiar a reprodução do experimento ou o reúso dos seus dados.

Quanto à representação padronizada de dados de Neuroimagem, existem propostas tais como o *Neuroimaging Data Model* (NIDM)[5] e XCEDE-DM [Ghosh et al. 2012], modelos que capturam detalhes da aquisição e análise das imagens. Esses modelos de dados derivam do W3C PROV [Moreau et al. 2008], um modelo padrão para a troca de informações de proveniência. Para dados de eletrofisiologia, o modelo *Neurodata Without Borders* (NWB) [Teeters et al. 2015] se destaca. Olazar et al. [Ruiz-Olazar et al. 2016] caracterizaram e comparam algumas dessas iniciativas. Como nenhuma dessas propostas de padronização foi amplamente adotada na comunidade de Neurociência até agora, as ferramentas computacionais ainda são o principal recurso de suporte para cientistas interessados no intercâmbio de informações de Neurociência.

1.3. Nova Plataforma para a Construção de Bancos de Dados de Experimentos

Para que se tenha bancos de dados de experimentos que atendam requisitos mínimos de qualidade, é preciso que os cientistas tenham à sua disposição ferramentas computacionais que os amparem nas suas tarefas rotineiras de condução dos experimentos e de coleta e análise dos dados relacionados. Apesar de já existirem repositórios públicos para o armazenamento e compartilhamento de dados na área de Neurociência, ainda há uma forte carência por ferramentas de software que mantenham, de forma integrada e categorizada, todos os dados coletados em um experimento e a suas informações de proveniência.

Com o objetivo de propor uma solução para esse problema, este artigo apresenta uma plataforma de software de apoio à criação e à disponibilização pública de bancos de dados de experimentos em Neurociência. Essa plataforma é composta por duas ferramentas de software – o *Neuroscience Experiments System* (NES) e o *NeuroMat Open Database* (NeuroMat DB). Essas ferramentas são uma iniciativa do Centro de Pesquisa e Difusão em Neuromatemática (CEPID NeuroMat), financiado pela FAPESP.

O NES é um software livre para o gerenciamento de dados de experimentos de Neurociência. O NES garante que todo dado de experimento registrado no banco de

[5]Neuroimaging Data Model – http://nidm.nidash.org/

dados mantido por ele esteja devidamente acompanhado de suas informações de proveniência, impondo um formato comum para a representação e armazenamento dos dados de experimentos de um mesmo laboratório ou grupo de pesquisa. Com isso, os dados armazenados têm melhor qualidade e ficam mais fáceis de serem compartilhados publicamente e reusados. O NES permite que dados anonimizados de um experimento sejam enviados para o NeuroMat DB, que disponibiliza dados publicamente por meio de um portal Web.

2. Experimentos em Neurociência

Experimentos em Neurociência podem ser de diferentes tipos, como, por exemplo, comportamentais, de eletrofisiologia ou de neuroimagem. Experimentos de eletrofisiologia geralmente envolvem Eletroencefalografia (EEG), Estimulação Magnética Transcraniana (EMT) ou Eletromiografia (EMG), enquanto que os de neuroimagens costumam gerar imagens por Ressonância Magnética (RMI) ou Ressonância Magnética Funcional (fMRI).

Em um experimento em Neurociência, os sujeitos (humanos ou animais) geralmente são separados em grupos. Cada grupo pode ser submetido a um conjunto de condições experimentais específicas. Cada tipo de experimento envolve uma preparação específica para sua realização, como, por exemplo, configurações no equipamento de coleta de sinais eletrofisiológicos e a colocação de eletrodos em locais previamente definidos do corpo dos sujeitos do experimento. Todas as definições sobre um experimento, incluindo definições sobre os objetivos, a descrição dos grupos de sujeitos que serão testados, as condições experimentais às quais os grupos serão submetidos e os tipos de coletas de dados que serão realizados, são chamadas pelos cientistas de *protocolo experimental*.

O processo experimental geralmente é composto por três etapas: (i) *planejamento do experimento*, (ii) *coleta e armazenamento de dados* e (iii) *análise dos dados*. Na fase de *planejamento do experimento*, é realizada a preparação para realização do experimento, onde devem ser identificados os tipos de dados que serão coletados e a definição do protocolo experimental. Depois, um grupo de sujeitos é selecionado para participar do experimento e a coleta dos dados começa a ser realizada. Utilizando o protocolo experimental como guia, na fase de *coleta e armazenamento de dados* é realizado o registro dos dados primários do experimento para cada participante. Dados primários são os que estão em sua forma bruta (i.e., como foram recebidos das suas fontes). A coleta de dados primários pode ser realizada de várias formas, como, por exemplo: o preenchimento de questionários projetados para o experimento; o uso de equipamentos para a captura de dados, como EEG e MRI; observação do comportamento do participante em resposta às condições experimentais às quais ele é exposto. Na última fase, que é a de *análise de dados*, os dados primários são processados e analisados e podem gerar novos dados, chamados de *dados derivados*. Dados derivados e seus processos de geração também devem ser armazenados e documentados para serem posteriormente reusados ou reproduzidos.

3. O *Neuroscience Experiments System* (NES)

O *Neuroscience Experiments System* (NES) é uma ferramenta de software livre para auxiliar neurocientistas no gerenciamento dos dados de seus experimentos. O NES coleta, estrutura e organiza os dados de todas as etapas do processo experimental de um experimento em Neurociência. A Figura 1 ilustra como o NES se integra ao processo experimental. Na fase do desenho experimental, o NES permite registrar os metadados do experimento, como a descrição do protocolo experimental, a configuração dos equipamentos

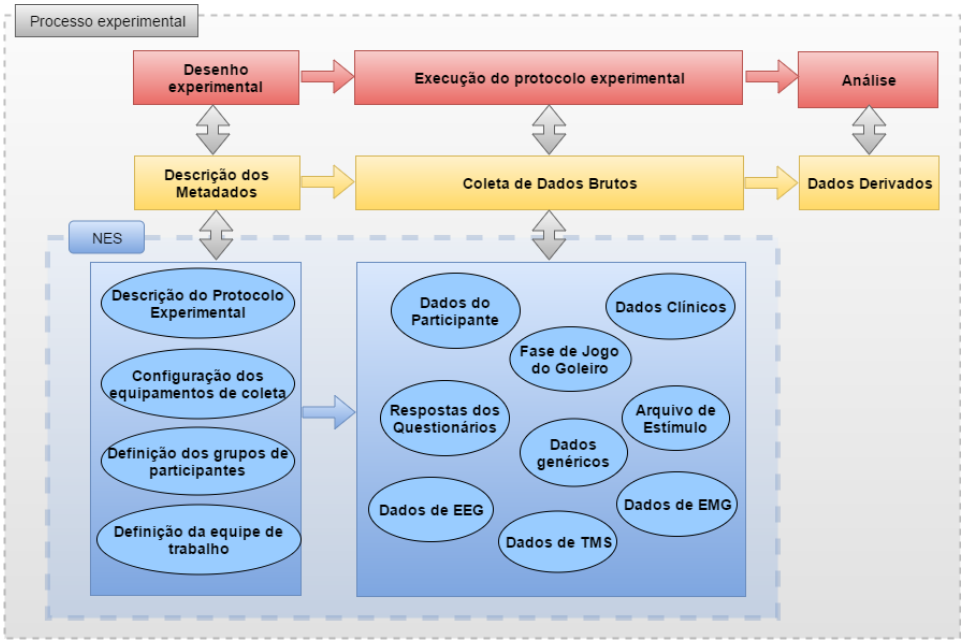

Figura 1. Diagrama da integração do NES no processo experimental

de coleta dos dados, a definição dos grupos de participantes e da equipe de trabalho, entre outras informações que descrevem o contexto do experimento.

Para caracterizar dados experimentais e metadados de vários tipos de experimentos, o NES foi desenvolvido para contemplar a coleta de dados de diversas naturezas, como dados de EEG, EMG, EMT, dados de estímulos (i.e., auditivos, visuais, etc.), dados genéricos (i.e., estabilometria, cinemática, comportamentais, etc.), dados de respostas de questionários, dados vindos da integração com o Jogo do Goleiro[6], dados sobre os participantes (sujeitos) dos experimentos, assim como outras informações que podem ser inseridas de forma textual ou como arquivos anexados ao experimento. Assim, o NES fornece uma plataforma estruturada, robusta e abrangente, com suporte a dados de proveniência, para possibilitar a rastreabilidade dos dados e a reprodução dos experimentos.

O modelo de dados do NES está alinhado com diretrizes para reportar experimentos em eletrofisiologia, como as já citadas MINI e MINEMO. O NES também trabalha com vários formatos de arquivos usados pela comunidade de Neurociência. Em particular, é interoperável com o formato *Neuroscience Without Border* (NWB), uma das iniciativas mais promissoras para a representação padronizada de dados de eletrofisiologia. O modelo de armazenamento usado no NES permite que dados provenientes de fontes de informação variadas sejam convertidos em formatos interoperáveis (e.g., CSV, JSON, PDF, etc.) que podem ser facilmente exportados para uso em diferentes plataformas computacionais.

Por ter sido desenvolvido como um sistema Web, o NES possui uma interface amigável e pode ser executado em vários tipos de dispositivos, como computadores *desktop*, *tablets*, *smartphones*, etc., com a mesma qualidade de apresentação, provendo maior fa-

[6]O Jogo do Goleiro é um jogo desenvolvido pelo CEPID NeuroMat para estudar a forma como o cérebro humano reconhece padrões: `http://game.numec.prp.usp.br/`

cilidade de uso. Os dados gerenciados pelo NES são armazenados em um banco de dados relacional, usando o *PostgreSQL*, de forma a garantir facilidade de manutenção e segurança dos dados. O armazenamento dos dados coletados por sistemas externos é feito por meio de arquivos nos formatos padronizados existentes. Os dados de proveniência desses últimos também são armazenados, facilitando a recuperação posterior.

3.1. Funcionalidades do NES

Experimentos em Neurociência envolvem uma grande heterogeneidade de formatos de dados e metadados complexos. Para esses requisitos, o NES fornece as funcionalidades:

Registro de Participantes: para ajudar no controle dos participantes, o NES mantém o cadastro de informações pessoais, dados sociodemográficos, história social e avaliações médicas. Às avaliações médicas, é possível associar diagnósticos com o uso do Código Internacional de Doenças (CID) e anexar possíveis exames realizados.

Gerenciamento de Questionários: o NES está integrado o LimeSurvey[7] para gerenciar a administração de questionários eletrônicos usados na coleta de dados em experimentos. Com o uso de questionários eletrônicos, outras dados mais específicos ou avaliações longitudinais podem ser coletadas de forma estruturada.

Gerenciamento de Experimento: essa funcionalidade envolve o registro e a configuração do experimento, assim como a descrição do protocolo experimental. Um protocolo experimental é descrito no NES como um *workflow* não-automatizado. Dessa forma, o NES consegue representar todas as condições experimentais como passos de um processo, onde cada passo pode ser algo como: uma instrução para o participante, uma aplicação de questionário, a apresentação de um estímulo, uma tarefa para o participante, uma tarefa para o experimentador, um conjunto de passos, etc. Um conjunto de passos é uma forma de organizar sub-passos, que podem ser realizados de forma paralela ou sequencial. Uma vez que o protocolo experimental tenha sido criado, é possível realizar coletas de dados referentes aos participantes do experimento. O NES é capaz de lidar com dados eletrofisiológicos (e.g., EEG e EMG) coletados em vários formatos usados pela comunidade de Neurociência. No NES, cada dado coletado em um experimento está sempre associado ao sujeito do qual foi coletado e a uma etapa específica do protocolo experimental.

Exportação de Dados: o NES permite exportar todos os dados e metadados dos experimentos que ele armazena. A exportação inclui os dados dos sujeitos do experimento (i.e., respostas dos questionários, dados clínicos, dados primários, etc.) e metadados sobre o protocolo experimental (i.e., descrição do experimento e das etapas do protocolo, configuração de equipamentos e anotações feitas por cientistas). Além disso, é possível realizar filtros pelos dados dos participantes, como sexo, diagnóstico e idade. NES exporta os dados textuais e numéricos organizados em arquivos em formatos textuais puros (e.g., CSV e JSON). Dados de EEG podem ser exportados no formato NWB.

3.2. Integração com o NeuroMat Open Database (NeuroMat DB)

O NES é um software que deve ser instalado em um servidor Web para gerenciar e manter os dados de experimentos de um laboratório ou grupo de pesquisa em Neurociência em particular. Para manter a segurança dos dados, o NES usa criptografia de dados e mecanismos de controle de acesso baseado em perfis de usuários.

[7]LimeSurvey – https://www.limesurvey.org/

Para compartilhar dados e metadados de experimentos de forma pública, o CEPID NeuroMat disponibiliza o *Neuromat Open Database* (NeuroMat DB). O portal Web do NeuroMat DB[8] é uma plataforma de acesso aberto para compartilhamento e busca de dados e metadados de experimentos de Neurociência. Através do NES, um pesquisador pode enviar dados e metadados de seus experimentos para o serviço Web do NeuroMat DB. Os dados dos participantes são anonimizados antes de serem enviados; nenhum dado sensível é enviado do NES para o banco de dados aberto. Quando um novo conjunto de dados de um experimento chega ao Neuromat DB, um comitê de curadoria analisa se os dados são apropriados para publicação. Os conjunto de dados aprovados pelo comitê são disponibilizados publicamente no portal Web do Neuromat DB.

3.3. Licença de Uso e Outros Softwares Livres

Tanto o NES quanto o NeuroMat DB têm uma arquitetura completamente aberta e foram desenvolvidos a partir de ferramentas de software livre, como a linguagem de programação Python, o arcabouço Web Django, o arcabouço de *front-end* Bootstrap e o sistema gerenciador de banco de dados PostgreSQL. A licença do NES e do Neuromat DB é a Mozilla Public License versão 2.0[9]), que dá total liberdade para uso e alterações dos softwares. O código fonte, a documentação e o status do desenvolvimento do NES e do NeuroMat DB podem ser vistos nos seguintes endereços, respectivamente: `https://github.com/neuromat/nes` e `https://github.com/neuromat/portal`.

4. O Uso do NES na Construção de Bancos de Dados

O Laboratório de Pesquisa em Neurociências e Reabilitação (LNR) do Instituto de Neurologia Deolindo Couto (INDC) da UFRJ, em colaboração com o CEPID NeuroMat, está trabalhando na investigação dos mecanismos de plasticidade cerebral e na avaliação de preditores da resposta ao tratamento de reabilitação em pacientes com lesões do plexo braquial – um conjunto de nervos que inervam os membros superiores. O LNR está usando o NES para armazenar e documentar os dados coletados em seus estudos, que são constituídos principalmente por registros eletrofisiológicos, respostas a questionários clínicos e dados comportamentais. Uma versão parcial do conjunto de dados desses estudos já está disponível publicamente no portal Web do NeuroMat DB.

O NES também está sendo usado no gerenciamento de dados dos estudos conduzidos na Rede AMPARO[10], uma iniciativa do CEPID NeuroMat para promover a melhora na qualidade de vida de pessoas vivendo com Doença de Parkinson no Brasil e de seus familiares. Esses estudos envolvem, principalmente, dados clínicos, respostas de questionários e dados comportamentais capturados com o Jogo do Goleiro.

5. Considerações Finais

Existem vários desafios relacionados à criação de bancos de dados abertos na área de Neurociência, como a ausência de padrões para a representação de dados de experimentos, consequência da complexidade e variabilidade da estrutura dos protocolos experimentais. Mas para que se possa disponibilizar publicamente dados experimentais, é preciso garantir que eles sejam registrados de forma estruturada, com consistência e completude,

[8]NeuroMat DB – `http://neuromatdb.numec.prp.usp.br/`
[9]Mozilla Public License versão 2.0 – `https://www.mozilla.org/en-US/MPL/2.0/`
[10]Rede AMPARO – `https://amparo.numec.prp.usp.br/`

conjuntamente com suas informações de proveniência. Isso é essencial para que dados desse tipo possam ser entendidos e reusados.

Nesse contexto, o *Neuroscience Experiments System* (NES) e o *NeuroMat Open Database* (NeuroMat DB) são importantes contribuições para a comunidade de Neurociência, por constituírem um plataforma de software livre amigável e, ao mesmo tempo, poderosa para o registro e compartilhamento de dados experimentais e suas informações fundamentais de proveniência. Essas ferramentas já estão sendo usadas na criação de bancos de dados que visam amparar progressos na compreensão do funcionamento cerebral e no diagnóstico e tratamento de doenças neurológicas.

Atualmente, estamos estendendo as duas ferramentas a fim de habilitá-las a gerenciar dados derivados (ou seja, dados resultantes de um processamento computacional aplicado sobre os dados primários). Também desejamos incorporar às ferramentas a capacidade de registrar protocolos e coletas de experimentos envolvendo neuroimagens.

Referências

[Chicurel 2000] Chicurel, M. (2000). Databasing the brain. *Nature*, 406(6798):822–825.

[Frishkoff et al. 2011] Frishkoff, G., Sydes, J., Mueller, K., Frank, R., Curran, T., Connolly, J., Kilborn, K., Molfese, D., Perfetti, C., and Malony, A. (2011). Minimal information for neural electromagnetic ontologies (MINEMO): A standards-compliant method for analysis and integration of event-related potentials (ERP) data. *Standards in Genomic Sciences*, 5(2):211–223.

[Ghosh et al. 2012] Ghosh, S., Nichols, N., Gadde, S., Steffener, J., and Keator, D. (2012). Xcede-dm: A neuroimaging extension to the W3C provenance data model. In *Front. Neuroinform. Conference Abstract: 5th INCF Congress of Neuroinformatics*.

[Gibson et al. 2009] Gibson, F., Overton, P. G., Smulders, T. V., Schultz, S. R., Eglen, S. J., Ingram, D., Panzeri, S., Bream, P., Sernagor, E., Cunningham, M., Echtermeyer, C., Simonotto, J., Kaiser, M., Swan, D. C., and Lord, P. (2009). Minimum information about a neuroscience investigation (MINI): electrophysiology. *Nature Precedings*.

[Koslow 2000] Koslow, S. H. (2000). Should the neuroscience community make a paradigm shift to sharing primary data? *Nature Neuroscience*, 3:863–866.

[Koslow 2002] Koslow, S. H. (2002). Sharing primary data: a threat or asset to discovery? *Nature Reviews Neuroscience*, 3(4):311–313.

[Kötter 2001] Kötter, R. (2001). Neuroscience databases: tools for exploring brain structure–function relationships. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 356(1412):1111–1120.

[Moreau et al. 2008] Moreau, L., Groth, P., Miles, S., Vazquez-Salceda, J., Ibbotson, J., Jiang, S., Munroe, S., Rana, O., Schreiber, A., Tan, V., and Varga, L. (2008). The provenance of electronic data. *Communications of the ACM*, 51(4):52–58.

[Poldrack et al. 2008] Poldrack, R. A., Fletcher, P. C., Henson, R. N., Worsley, K. J., Brett, M., and Nichols, T. E. (2008). Guidelines for reporting an fMRI study. *Neuroimage*, 40(2):409–414.

[Ruiz-Olazar et al. 2016] Ruiz-Olazar, M., Rocha, E. S., Rabaça, S. S., Ribas, C. E., Nascimento, A. S., and Braghetto, K. R. (2016). A review of guidelines and models for representation of provenance information from neuroscience experiments. In *International Provenance and Annotation Workshop*, pages 222–225. Springer.

[Teeters et al. 2015] Teeters, J. L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., Asari, H., Peron, S., Li, N., Peyrache, A., et al. (2015). Neurodata without borders: Creating a common data format for neurophysiology. *Neuron*, 88(4):629–634.

A Search Space Exploration Framework for e-Science Applications

Eric B. Gauch, Bruno E. C. Milanesi, Bruno Silva, Renato L. F. Cunha, Marco A. S. Netto

[1]IBM Research
Rua Tutóia, 1157 - São Paulo - SP - Brasil

{ebgauch,milanesi,sbruno,renatoc,mstelmar}@br.ibm.com

Abstract. *High Performance Computing (HPC) has always been a fundamental component to conduct scientific experiments. Model calibrations/simulations often require several executions of scientific applications by changing their input parameters. This process is a common practice in research even though it represents a tedious and error-prone task. In this paper we propose Copper framework which employs a black-box strategy and contains a set of plugins to accelerate user experiments for exploring search spaces in HPC parametric applications. Copper has been used to conduct scientific experiments in different areas including, agriculture, oil & gas, flood simulation, and bioinformatics.*

1. Introduction

High Performance Computing (HPC) is crucial for the execution of e-Science applications in various fields including agronomy, health, aerospace, circuit design, astronomy, and oil & gas exploration. Usually, researchers have to execute the same applications several times with different parameter values to calibrate models or evaluate what-if scenarios. Without a proper tool, this process is often cumbersome, error-prone, and a time consuming activity. Additionally, e-Science users are highly specialized in their respective research areas but they may not have the necessary expertise to execute their applications in HPC dedicated infrastructure (e.g., cluster).

To fill this gap, we propose Copper, a search space exploration tool for e-Science applications. The tool can execute any application that has a command line interface and makes transparent the access and monitoring of HPC infrastructures such as clusters and clouds. Copper presents a set of plugins to improve the search space exploration by providing hints to the user during the experiment set. In this paper, we present a brief overview of Copper and describe how to connect Copper to user applications. The proposed tool has been used to explore search spaces in e-Science applications from different areas such as agriculture, oil & gas, flood simulations, and bioinformatics [Silva et al. 2018, Silva et al. 2016].

2. Related Work

The execution of large number of independent jobs may become a complex task to users, mainly when distributed resources are used to execute these jobs. Resources may have different computational power and availability and may have different mechanisms to be accessed. Therefore, over the years, several tools have been created such as Condor

[Litzkow et al. 1988], XtremWeb [Fedak et al. 2001], BOINC [Anderson 2004], Nimrod [Abramson et al. 2000], and OurGrid [Andrade et al. 2003] to facilitate user access to these computational platforms. Gil *et al.* [Gil et al. 2011] introduced Wings, a system to assist scientists in designing experiments by tracking constraints and ruling out invalid designs. These tools mainly focus on managing users' jobs looking into the computational infrastructure. The main difference between our tool and the previous ones is that ours provides useful plugins to accelerate search space explorations [Silva et al. 2018]. For instance, Copper presents JobPruner which is a tool that looks into the user's workload and finds patterns from past experiments, which allow users to drastically reduce their search spaces when performing experiments of similar nature.

3. Copper

This section presents an overview of Copper's modules including the backend and frontend interfaces. The main idea is to demonstrate how general applications can be plugged into Copper and accessed via a high level user interface.

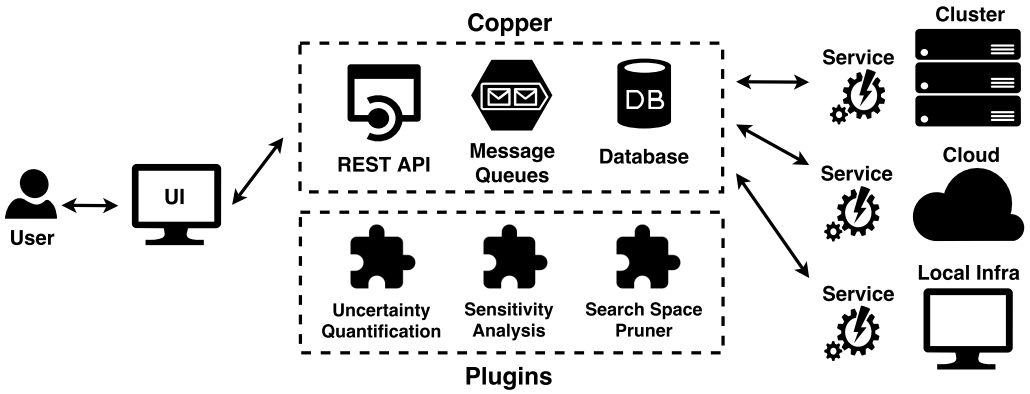

Figure 1. Overview of Copper's workflow

3.1. Backend

As illustrated in Figure 1, the user interacts with the user interface selecting one or more parameters with a single value or a range of values for each of them. These parameters and their respective values are then validated in Copper's backend where a corresponding application with its parameters definition must have already been registered. If a range of values is chosen for a parameter, Copper generates a set of jobs for each combination of parameters and values, and store them in a database. These jobs are sent through a message queue for local or remote processing. Copper can utilize techniques like uncertainty quantification, sensitivity analysis, and search space pruning available as plugins to achieve the user's goal [Silva et al. 2018]. The subsystem that consumes jobs from the message queue, executes user's software, and returns a result to Copper consists of two main components:

- A *service* that pulls messages from the message queue, parses it and calls the User Application, passing as argument the parameters and values from the message.
- The *User Application* must have a command line interface where we can specify the arguments for that application.

In order to execute user's jobs, both the User Application and the service itself have to be installed in the user's computational infrastructure (e.g., cluster, cloud or local machine). Having that satisfied, a configuration file is used to point where the User Application's executable is and some other options like the message queue's address. With all that set, the service can start pulling messages from the message queue. These messages sent from Copper to the message queue and acquired by the service are represented as a JSON (*JavaScript Object Notation*) file and contains all the parameters for the job execution. The service calls the executable specified in the configuration file passing the parameters from the message as command line arguments and will then, retrieve the result and send a message back to Copper. For the User Application, any software that takes a group of parameters as input and outputs a result can take advantage of Copper. The only requirement is that this application has to offer a command line interface. In some cases, this application takes a file with all the parameters as input, so a parser may be needed in order to create this file with the information that came from the message queue.

3.2. Frontend

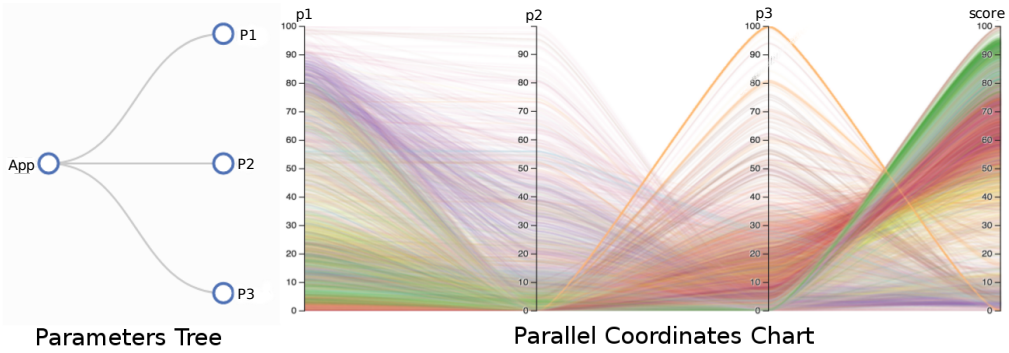

Figure 2. Copper's parameters tree and parallel coordinates chart

Copper's frontend was designed to assist the user with the application's parameters selection, tracking/analysing infrastructure usage, and display of experiments results. To achieve these goals, the interface was subdivided into three main parts: parameter tree (model), performance and infrastructure usage dashboard, and results visualizations. In Figure 2, we present the results visualization tab, in which each job corresponds to a line in the parallel coordinates graph. In this particular experiment, the application has three parameters (p1, p2, and p3) and a single value is provided as a result.

Copper accepts any application parameter to be used by the user, thus allowing a black box strategy. By using a JSON parameter description file, the user interface is automatically created with either the input parameters three (Figure 2) or a group of forms. Therefore, users can easily specify the experiment's input parameters and intuitively visualize the respective results. Users can monitor the execution of ongoing experiments via a infrastructure dashboard, which presents the computational resources (VMs, containers, or dedicated servers) allocated for each set of jobs. Real-time performance metrics can also be tracked in the infrastructure dashboard. These metrics include: job execution throughput and number of waiting/running/completed jobs. The results are then displayed on a parallel coordinates chart, also in real-time, making it easy to visualize all the variables (dimensions) and understand their impact on the model and associated scores.

4. Conclusion

Several areas in science and engineering require many executions of a software system with different values for each supported parameter. These executions are responsible for evaluating complex models using diverse scenarios. Executing all possible values for all supported parameters are usually not feasible, especially under cost and deadline constraints.

To fill this gap, this work introduced a tool to help users in executing their software systems by exploring parameter search spaces. Our main lesson, while developing the tool and evaluating User Applications was related to the particular characteristics of each User Application. Instead of focusing on each possible feature of any application, we concentrated our efforts on creating a general framework that addresses the main characteristics of parametric applications (execution with a variety of input parameters). Advanced users can also take benefit from our framework by creating user interfaces customized to their needs and employ Copper REST API to execute their parametric applications.

References

Abramson, D., Giddy, J., and Kotler, L. (2000). High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid. In *Proceedings of the 14th International Parallel & Distributed Processing Symposium*. IEEE.

Anderson, D. P. (2004). BOINC: A system for public-resource computing and storage. In *Proceedings of the 5th International Workshop on Grid Computing*. IEEE.

Andrade, N., Cirne, W., Brasileiro, F., and Roisenberg, P. (2003). OurGrid: An approach to easily assemble grids with equitable resource sharing. In *Proceeding of the Workshop on Job Scheduling Strategies for Parallel Processing*. Springer.

Fedak, G., Germain, C., Néri, V., and Cappello, F. (2001). Xtremweb: A generic global computing system. In *Proceedings of the 1st International Symposium on Cluster Computing and the Grid*. IEEE.

Gil, Y., Ratnakar, V., Kim, J., Gonzalez-Calero, P., Groth, P., Moody, J., and Deelman, E. (2011). Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems*, 26(1):62–72.

Litzkow, M. J., Livny, M., and Mutka, M. W. (1988). Condor - A hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems*. IEEE.

Silva, B., Netto, M. A., and Cunha, R. L. (2018). JobPruner: A machine learning assistant for exploring parameter spaces in HPC applications. *Future Generation Computer Systems*, 83:144 – 157.

Silva, B., Netto, M. A. S., and Cunha, R. L. F. (2016). SLA-aware Interactive Workflow Assistant for HPC Parameter Sweeping Experiments. In *Proceedings of the 11th Workshop on Workflows in Support of Large-Scale Science with The International Conference for High Performance Computing, Networking, Storage and Analysis*.

Assessing the Impact of Supporting Information on the Scheduling of Scientific Workflows on Clouds

Eduardo Cotrin Teixeira[1], **Daniel Cordeiro**[2], **Kelly Rosa Braghetto**[3]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Cornélio Procópio – PR – Brazil

[2]Escola de Artes, Ciências e Humanidades
[3]Departamento de Ciência da Computação - Instituto de Matemática e Estatística
Universidade de São Paulo (USP) – SP – Brazil

cotrin@utfpr.edu.br, daniel.cordeiro@usp.br, kellyrb@ime.usp.br

Abstract. *Executing scientific workflows in high-performance cloud computing platforms requires the use of scheduling algorithms that allow workflows execution as fast as possible, while minimizing the monetary cost of such executions. In this work we study how the use of supporting information can offer guidance to scheduling algorithms, helping them to devise more efficient execution plans in terms of the total execution time (makespan) and the total monetary cost. Using two large-scale scientific workflows, our experiments showed that simple modifications on a classical scheduling algorithm (HEFT), in conjunction with the appropriate supporting information, could reduce the monetary cost of an execution in up to 59% and reduce the makespan in up to 8.6%.*

1. Introduction

Cloud computing platforms are a viable alternative for running scientific workflows. However, the scheduling on this type of platform generally must consider specific constraints such as a limited budget and the type of computational resources required for the execution. To design effective scheduling algorithms under such constraints that will ensure efficient workflow executions, one must rely on *supporting information*, such as estimated duration of workflow activities or execution time and cost constraints.

This work explores the use of supporting information that can be added to scientific workflow models to support their scheduling and execution on clouds. To assess the impact of use of supporting information, we performed experiments with large-scale, real scientific workflows executed through the workflow management system (WfMS) Pegasus [Deelman et al. 2009]. To be able to consider the supporting information associated to the workflows in the scheduling phase, we modified a classical scheduling algorithm (HEFT) [Topcuouglu et al. 2002] existent in Pegasus to implement two new ones. The first algorithm minimizes the workflow total execution time (makespan), while the second minimizes the total monetary cost to execute the workflow in the cloud computing under a deadline constraint. In the experiments, we observed a reduction in the monetary cost of the workflow execution in the cloud of up to 59% and a reduction in the makespan of up to 8.6%, when compared to the scheduling with no supporting information available.

2. Basic Definitions

A scientific workflow is the automation of a scientific experiment or process, expressed in terms of the activities and their interdependencies [Cuevas-Vicenttin et al. 2012]. Sci-

entific workflow models can be expressed as Directed Acyclic Graphs (DAGs), where each node represents an activity and each directed edge represents a precedence relation between two activities. A *scheduling algorithm* uses this information to choose, for each activity, the resource and execution time that optimizes a specific objective function.

The essential information required to schedule and execute a workflow is: description of the computational resources available for the execution, scheduling objective to be optimized and specification of the activities with their precedence relations (i.e., the workflow DAG). The specification of the activities includes the procedure to be performed (program, script, service, etc.), and the parameters required for the execution (including input and output data). Any additional information related to the workflow or to the execution environment can be considered a *supporting information* for the scheduling.

3. Experimental Evaluation of the Supporting Information Impact

Support information impact on scheduling and execution of scientific workflows on cloud was evaluated using InterNuvem[1], a pay-per-use academic IaaS cloud platform. Ten virtual machines with Ubuntu 12.04 operating system were used, namely, 4 Standard VMs (2 CPUs and 4GB RAM - R$ 0.112/h), 3 Advanced VMs (4 CPUs and 8GB RAM - R$ 0.202/h) and 3 High Performance VMs (8 CPUs and 32 GB RAM - R$ 0.542/h). We used the WfMS Pegasus version 4.5.0 to manage the execution of two large-scale scientific workflows — Montage (four instances, with degrees 0.5, 1.0, 2.0, and 4.0) and Epigenomics (two instances, using TAQ and HEP datasets) [Juve et al. 2013].

We consider the scheduling of scientific workflows as a bicriteria optimization problem, where the performance of the workflow execution is related to two different (and somehow contradictory) performance objectives: the makespan (finishing time of the last activity) and the monetary cost of execution. To schedule and execute the workflow instances considering supporting information, we developed and evaluated two scheduling algorithms: HEFTData and HEFTDeadline.

3.1. HEFTData

HEFT (Heterogeneous Earliest-Finish-Time) [Topcuouglu et al. 2002] is an algorithm for scheduling dependent activities into heterogeneous machines that minimizes makespan. HEFT assigns to each activity a rank (i.e., the expected distance from the end of the execution) defined as $rank(n_i) = \overline{w_i} + \max_{n_j \in succ(n_i)}(\overline{c_{i,j}} + rank(n_j))$, where n_i represents the i^{th} activity, $\overline{w_i}$ is an average computation cost of activity i among all the resource types, $succ(n_i)$ is the set of all activities that immediately depend on activity n_i, and $\overline{c_{i,j}}$ is the average communication cost from activity n_i to n_j considering all pairs of machines types. After that, HEFT assigns the prioritized activities (in descending order of rank) to the machines. Pegasus' HEFT implementation considers only the activity's expected execution time, ignoring communication time. For workflows with large volumes of data, data transfer times may actually exceed the activity durations.

We have implemented in Pegasus a new version of the HEFT algorithm, called HEFTData, which considers supporting information about *input data volume*, besides the *activity estimated duration*. HEFTData works similarly to HEFT, except that tasks' rank

[1]InterNuvem: `https://internuvem.usp.br/`

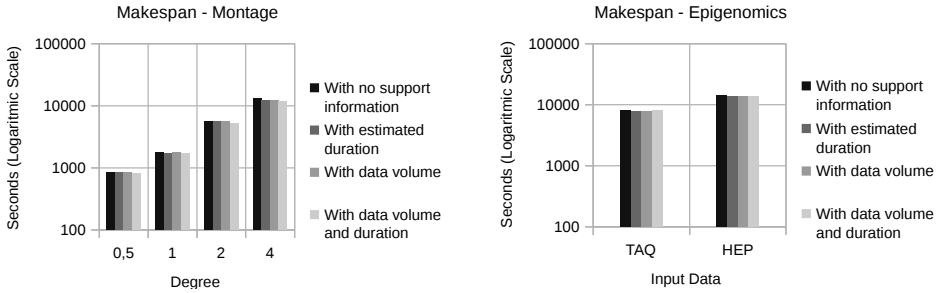

Figure 1. Average makespan for Montage and Epigenomics.

and expected completion time also consider data transfer times. To evaluate HEFTData, additional supporting information was needed: the *estimated average duration of each activity on each type of VM* used in the experiments (measured with at least 20 executions in each type of VM) and the *volume of data to be transferred between the activities*, to calculate the data transfer times. All supporting information was defined as part of the workflow model itself and was determined for each edge in the workflow DAG.

3.2. HEFTDeadline

Monetary cost is an important issue to execute workflows on clouds. Virtual machines may have different prices depending on their configuration, so the choices the scheduler makes have a direct impact on the cost of the execution. On the one hand, faster machines tend to be more expensive, and algorithms like HEFT tend to use more of this type of machine to minimize the makespan. On the other hand, if the scheduler uses cheaper machines, the workflow execution can take more time than the desired by the scientist. HEFTDeadline was designed to prioritize the execution of workflow activities in machines that provide the lowest monetary cost, considering as a constraint the supporting information *deadline*, that must be defined by the scientist for the workflow.

Activities and ranks are computed as in HEFTData. However, when mapping the activities to virtual machines, HEFTDeadline considers only suitable machines resulting on the cheapest execution costs, considering estimated duration of the activity execution multiplied by the price per use time of the machine. In this way, a more expensive machine can be selected if the cost of the activity execution becomes lower due to a shorter estimated duration in this machine. The execution cost also considers that a machine can have multiple execution nodes (e.g., multiple cores or hyper-threading capabilities). If a VM is allocated to an activity, it will be fully billed regardless of how many of its nodes are actually used. Therefore, an activity mapped to an execution node only has a cost if its completion time is greater than the latest completion time of the activities mapped to the other execution nodes of the same VM. The scheduler will not allocate a new VM while execution nodes on active VMs are still available and are more cost effective.

4. Results and Conclusion

Each instance was executed at least 8 times for each scheduler. The results were evaluated in terms of makespan (total duration in seconds) and monetary cost. Figure 1 shows the average makespan obtained using HEFTData and four different configurations of supporting information: (i) using no information, (ii) using only the *estimated activity durations*,

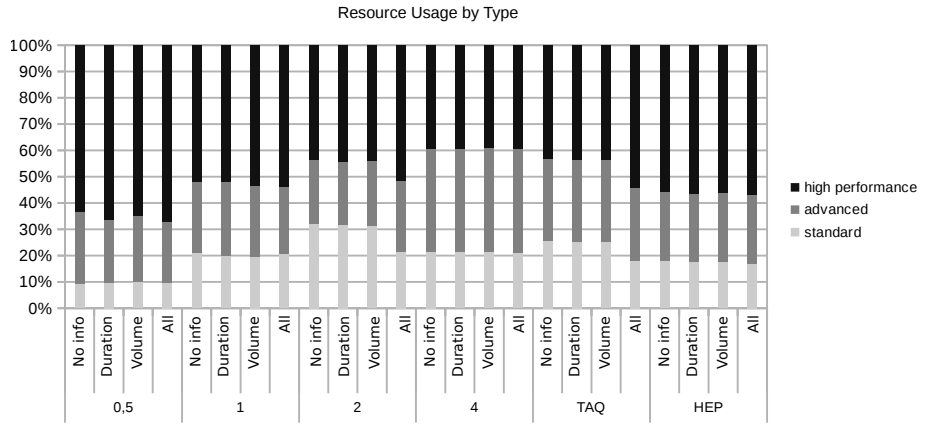

Figure 2. Average distribution of the virtual machines by type.

Table 1. Average monetary costs (in R$) using HEFTDeadline.

	HEFTData		HEFTDeadline			
Workflow	**Makespan**	**Cost**	**Deadline**	**Makespan**	**Cost**	**Cost Reduction**
Montage 0.5	819	0.17	983	906	0.07	59.1%
Montage 1.0	1711	0.39	2053	1924	0.16	59.2%
Montage 2.0	5139	1.22	6167	5678	0.55	55.4%
Montage 4.0	11818	2.56	14182	13543	1.34	47.7%
Epigenomics TAQ	7897	2.83	9476	9187	2.45	13.4%
Epigenomics HEP	13602	5.82	16322	16138	2.37	59.2%

(iii) using only the *input data volumes*, and (iv) using both duration and data volume. In all instances, the use of some supporting information reduced the average makespan. The greatest reduction (8.6%) was observed in the executions of Montage with degree 2.0.

Figure 2 shows the distribution of the average makespan by VM types used in the execution. High-performance VMs tended to be used more often when supporting information was used. Increased parallelism (machines with more processing nodes) and execution speed of the workflow activities provided improvement in the makespan.

Table 1 shows the average monetary costs of the executions with the HEFTDeadline. The deadline used for each instance was 120% of the average makespan obtained for the same instance with HEFTData using the support information *estimated activity durations* and *input data volumes*. The table shows that, with an increase of 20% in the makespan, the HEFTDeadline reduced the workflow average execution cost up to 59.2%.

References

Cuevas-Vicenttin, V. et al. (2012). Scientific workflows and provenance: Introduction and research opportunities. *Datenbank-Spektrum*, 12(3):193–203.

Deelman, E., Gannon, D., Shields, M., and Taylor, I. (2009). Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540.

Juve, G., Chervenak, A., Deelman, E., Bharathi, S., Mehta, G., and Vahi, K. (2013). Characterizing and profiling scientific workflows. *Future Generation Computer Systems*, 29(3):682–692.

Topcuouglu, H., Hariri, S., and Wu, M.-Y. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Trans. Parallel Distrib. Syst.*, 13(3):260–274.

Avaliação do uso eficiente de algoritmos paralelos de filogenia em supercomputadores

Kary Ocaña, Joice Alves, Micaella Coelho, Marcelo Galheigo, Carla Osthoff

Laboratório Nacional de Computação Científica (LNCC), Brasil

{karyann, joice, micaella, galheigo, osthoff}@lncc.br

Resumo. *Ambientes de processamento de alto desempenho (PAD) são requeridos pela bioinformática, em especial nas áreas de filogenia e evolução de organismos. O presente trabalho apresenta uma análise comportamental da ferramenta de filogenia RAxML em ambientes de PAD. O RAxML implementa paralelismo e distribuição de tarefas pelo uso de threads e bibliotecas MPI. As execuções foram realizadas nos clusters do supercomputador Santos Dumont (SDumont). Os resultados em termos de eficiência evidenciam que o impacto é gerado pela variação da configuração, acoplamento e uso do RAxML e do SDumont. Essa informação permitirá um melhor entendimento e uso eficiente desses ambientes especializados.*

Abstract. *Bioinformatics, especially the areas of phylogeny and evolution of organisms, requires high performance computing (HPC) environments. The present work presents a behavioral analysis of the RAxML phylogeny tool in HPC environments. RAxML implements parallelism and task distribution by using threads and MPI libraries. Executions were performed in the Santos Dumont (SDumont) supercomputer. Efficiency results show the impact is generated by the variation in configuration, coupling and use of RAxML and SDumont. This information will allow a better understanding and efficient use of these specialized environments.*

1. Introdução

A bioinformática é uma ciência multidisciplinar que apoia à análise e interpretação da grande quantidade de dados gerados pela biologia molecular, médica e da saúde, por meio do desenvolvimento de algoritmos e abordagens computacionais. A filogenia aplica algoritmos probabilísticos na construção de árvores filogenéticas, no apoio a análise sobre a vida evolutiva dos organismos, genes ou genomas. Alguns métodos de análise filogenética de estado de caractere, como a máxima verossimilhança (MV) e inferência Bayesiana (IB), implementam modelos probabilísticos complexos e estão atualmente dentre dos mais utilizados [Felsenstein 1996; Ronquist and Huelsenbeck 2003; Stamatakis 2014].

O método de MV busca a árvore mais verossímil dentre um conjunto de árvores geradas, tomando em conta informações como os dados genéticos, modelos evolutivos, taxas de mutação, entre outros. O programa de MV RAxML [Stamatakis 2014] foi escolhido por usufruir de características computacionais como paralelismo, distribuição e escalabilidade, o que o faz um candidato interessante para ser acoplado efetivamente em ambientes de processamento de alto desempenho (PAD) como *clusters*, grades e nuvens.

O RAxML, além da versão sequencial, apresenta as versões paralelas PThread, MPI e híbrida, o que possibilita a avaliação de diferentes algoritmos computacionais em ambientes de PAD [Stamatakis 2014]. A versão PThread emprega o paralelismo interno, *i.e.*, aquele que ocorre dentro de um nó do agregado, onde os processadores compartilham espaço de memória e se comunicam por instruções de leitura e escrita (arquitetura de memória compartilhada). Essa versão apresenta a paralelização em grãos finos, onde

pequenas quantidades de tarefas computacionais são realizadas entre os eventos de comunicação. A versão MPI emprega o paralelismo externo, *i.e.*, aquele que ocorre entre os nós agregados, com a comunicação por troca de mensagens através de uma rede que interconecta os nós (arquitetura de memória distribuída). Essa versão apresenta a paralelização em grãos grossos, onde grandes quantidades de tarefas computacionais são realizadas entre os eventos de comunicação e sincronização. Já a versão híbrida explora ambos os métodos PThread e MPI [Abramson *et al*. 2011; Aho *et al*. 2006; Pfeiffer e Stamatakis 2010].

O presente trabalho tem como objetivo avaliar o desempenho das versões paralelas do RAxML nos *clusters* do supercomputador Santos Dumont (SDumont). Os experimentos foram desenhados a fim de explorar algumas das configurações mais representativas, com variações nos dados de entrada (tamanho de dados biológicos) e da parametrização (número de replicações ou *bootstrap*) e das versões do RAxML. Os resultados fornecerão informações importantes de modo a dar suporte aos usuários externos e administradores do SDumont, para o uso eficiente de ferramentas em ambientes de supercomputadores.

Por exemplo, atualmente o LNCC hospeda portais científicos das mais diversas áreas da ciência, dentre eles o BioinfoPortal de bioinformática (http://bioinfo.lncc.br/) e o DockThor de ancoragem molecular (https://dockthor.lncc.br/). Esses portais estão alocados nos *clusters* do SDumont no SINAPAD e são usados intensivamente pela comunidade de pesquisadores das áreas respectivas, no Brasil e no exterior. Por tanto, é de interesse para a comunidade o uso otimizado dessas ferramentas em ambientes de PAD, devido a que pode diminuir o tempo de espera na execução de tarefas nesses portais.

O artigo está organizado em quatro seções, além dessa introdução. A Seção 2 apresenta a configuração do experimento, a Seção 3 discute sobre os resultados de desempenho e finalmente, a Seção 4 conclui este artigo.

2. Ambiente Computacional

O SDumont possui uma arquitetura de configuração híbrida com capacidade de processamento paralelo na ordem de 1,1 Petaflops. Ele é composto por 18.144 núcleos de CPU, distribuídos em 756 nós computacionais e 24 núcleos por nó. Também possui o nó especial MESCA2, de 240 núcleos e memória compartilhada de 6 Terabytes. Todos os nós do SDumont então interconectados por uma rede de alta velocidade, a Infiniband FDR, que integra o sistema de arquivos paralelos. O gerenciador de filas do SDumont é o Slurm.

2.1 Configuração do Ambiente com o RAxML

O RAxML é um programa de código aberto do GNU GPL, distribuído através do repositório GitHub em https://github.com/stamatak/standard-RAxML. A compilação do código é feita atendendo os requisitos dos recursos de CPU de maneira a melhor aproveitar suas características computacionais. Enquanto aos recursos do processador, o RAxML suporta o SSE3 (*Streaming SIMD Extensions 3*), AVX (*Advanced Vector Extensions*) e AVX2 (extensão mais rápida do SSE). Essas instruções são usadas pelo RAxML para acelerar substancialmente os cálculos de probabilidade e parcimônia que conduz. O RAxML pode ser executado com as versões sequencial (HPC) ou paralelas usando MPI, PThreads ou híbrida (MPI + *Threads*). O RAxML foi compilado no SDumont com o AVX.

2.2 Dados Experimentais

Os *datasets* usados são superalinhamentos de genes ortólogos universais de genomas de protozoários. A Tabela 1 mostra as características dos *datasets* usados como número de táxons, tamanho e número de caracteres do alinhamento. Para a execução do RAxML é necessário informar o arquivo no formato PHYLIP [Felsenstein 1985] (dado de entrada), a versão sequencial ou paralela do RAxML (tipo de processamento), o valor de *bootstrap* e o modelo de substituição. Para a análise de eficiência foram realizados dois experimentos: (i) variando o tamanho/caractere do dado de entrada (D0, D5, D3), fixando o *bootstrap* em 100 e (ii) variando os valores de *bootstrap* (100, 1.000, 2.000), fixando o dado de entrada em D5. Os experimentos foram executados três vezes e os resultados apresentados como a média dos valores obtidos.

As versões sequencial, PThread, MPI e híbrida do RAxML foram acopladas no SDumont. Sobre o uso dessas versões, a versão sequencial é geralmente selecionada para tratar conjuntos de dados pequenos a médios (filogenia de genes). As versões paralelas são indicadas para a execução de alinhamentos muito longos ou superalinhamentos (filogenia de genomas) e o seu desempenho depende do tipo *hardware* usado.

Tabela 1. Principais características dos *datasets* usados nos experimentos

Dataset	**Táxons**	**Tamanho**	**Caracteres**
D0	10	3,2 Kilobytes	233
D5	12	79 Kilobytes	4.941
D3	26	792 Kilobytes	22.906

3. Análise dos Resultados de Desempenho dos Experimentos

A Figura 1 apresenta a escalabilidade em termos de eficiência para o experimento com *datasets* de caracteres diferentes (“menor” D0 de 233, “médio” D5 de 4.941, “maior” D3 de 22.906), fixando o valor de *bootstrap* em 100 (considerado o mínimo ideal para a execução no RAxML) em 6 nós (144 *cores*). Pode ser observado que a melhor eficiência (0.9) foi atingida em 5 nós para ambos os *datasets* D5 e D3.

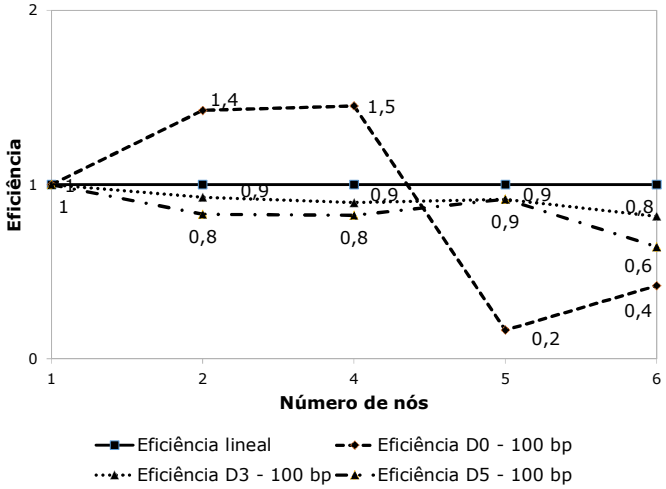

Figura 1. Eficiência do Experimento 1: variando o número de caracteres (D0, D5, D3) e fixando o *bootstrap* (100).

A Figura 2 apresenta a escalabilidade em termos de eficiência para o experimento com valores diferentes no *bootstrap* (100, 1.000, 2.000), fixando o *dataset* em D5 (4.941caracteres). Na Figura 2, o *bootstrap* 100 possui uma melhor eficiência para 5 nós, o *bootstrap* 1.000 apresenta melhor eficiência para 2 nós, e o *bootstrap* 2.000 para 6 nós.

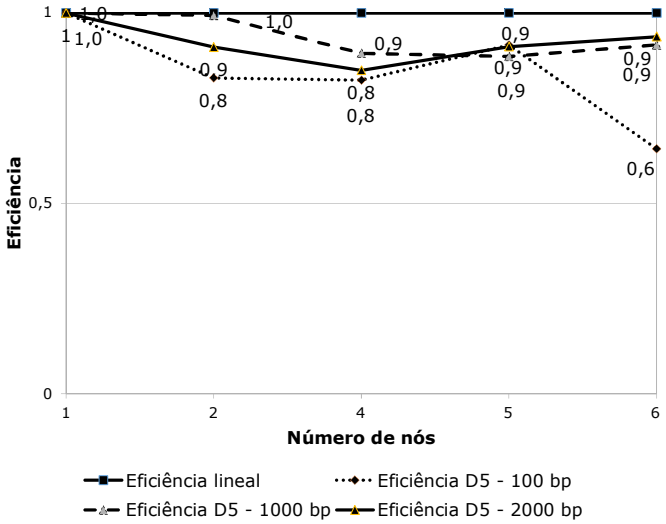

Figura 2. Eficiência do Experimento 2: variando o *bootstrap* (100, 1.000, 2.000), fixando os caracteres com D5.

4. Conclusão

O presente trabalho visa analisar o comportamento da ferramenta de bioinformática RAxML em ambientes de PAD. Os resultados das análises de filogenia indicam que o impacto na eficiência das execuções pode ser afetado por fatores relacionados às características biológicas (número de caracteres), parâmetros usados do RAxML (*bootstrap*) ou configuração dos *clusters* (bibliotecas de compilação ou natureza do processador). Desta maneira, é preciso uma análise exploratória das ferramentas, seus algoritmos e parâmetros, em especial quando experimentos em larga escala são alocados em ambientes especializados como o SDumont.

Como trabalhos futuros serão realizados estudos exaustivos das outras ferramentas acopladas no SDumont. Visa-se investir no uso de perfiladores que levem a desenvolver algoritmos que identifiquem gargalhos e erros de execução e ajudem na toma de decisões do melhor parâmetro, ferramenta ou ambiente de PAD. Finalmente, este trabalho retornará um *feedback* para os administradores do SINAPAD com o intuito de apoiar no uso eficiente de processadores.

Agradecimentos. A pesquisa foi desenvolvida no Laboratório de Bioinformática (LABINFO) do LNCC (https://www.labinfo.lncc.br/), as horas de execução no SDumont fornecidas pelo SINAPAD (http://sdumont.lncc.br/). Ela foi financiada pelo Programa Universal MCTI/CNPq 01/2016, Processo: 429328/2016-8, pelo CNPq, CAPES e FAPERJ

Referências Bibliográficas

Abramson, D., et al. (2011). Parameter Exploration in Science and Engineering Using Many-Task Computing. *IEEE Trans. Parallel Distrib. Syst.*, v. 22, n. 6, p. 960–973.

Aho, A. et al. (2006). *Compilers: Principles, Techniques, and Tools*. 2. ed. Addison Wesley.

Felsenstein, J. (1985). Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution*, v. 39, n. 4, p. 783–791.

Felsenstein, J. (1996). Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Methods in Enzymology*, v. 266, p. 418–427.

Pfeiffer, W. e Stamatakis, A. (2010). Hybrid MPI/Pthreads parallelization of the RAxML phylogenetics code. IEEE. http://ieeexplore.ieee.org/document/5470900/.

Ronquist, F. e Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics (Oxford, England)*, v. 19, n. 12, p. 1572–1574.

Stamatakis, A. (2014). RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, v. 30, n. 9, p. 1312–1313.

Enriquecimento de Dados de Proveniência de Análises Filogenéticas com Dados do NCBI: uma Abordagem Prática *

Lucas S. Tito[1], Kary A. C. S. Ocaña[2], Daniel de Oliveira[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF) – Niterói – RJ – Brasil

[2]Laboratório Nacional de Computação Científica (LNCC) Petrópolis – RJ – Brasil

ltito@id.uff.br, karyann@lncc.br, danielcmo@ic.uff.br

Abstract. *This paper proposes an approach called BioIntegrator, that aims at integrating and enriching provenance databases from phylogenetic analyzes using metadata present in external sources. Such approach aims at providing more analytical skills to scientists in their daily duties. Although it is a work in progress, the proposed approach has a clear potential regarding the analysis and evaluation of the results of experiments.*

Resumo. *Esse artigo apresenta uma proposta de abordagem, chamada BioIntegrator, para integração e enriquecimento de bases de dados de proveniência de análises filogenéticas com metadados presentes em fontes externas. Tal abordagem visa oferecer maior capacidade analítica aos cientistas em suas tarefas diárias. Apesar de ser um trabalho em andamento, a abordagem proposta tem um potencial claro no que tange a análise e validação de resultados dos experimentos.*

1. Introdução

O volume de dados genômicos passível de análise pela comunidade científica cresce em um ritmo acelerado, devido às recentes tecnologias tanto na área biológica quanto na computação. *e.g.*, as tecnologias de sequenciamento de nova geração (SNG) e processamento de alto desempenho (PAD). Uma das áreas da bioinformática que mais se beneficia dessas tecnologias é a análise evolutiva filogenética. Essa área de pesquisa tem como objetivo gerar conhecimento sobre processos evolutivos ou relações filogenéticas entre espécies. Experimentos filogenéticos recebem como entrada sequências ou até mesmo genomas inteiros, produzindo árvores e diversas estatísticas utilizadas para inferir a história evolutiva ou a filodinâmica e a filogeografia de processos infecciosos entre espécies [Felsenstein 1996] [Ocaña et al. 2011]. Diversos experimentos filogenéticos já foram propostos na literatura, muitos deles modelados como *workflows* científicos e executados em Sistemas de *Workflows* (SWf) em ambientes de PAD, como o SciPhy [Ocaña et al. 2011]. Os *workflows* geram dados de proveniência que representam o conjunto de informações relacionadas à execução de um experimento [Freire et al. 2008]. Tais informações auxiliam os pesquisadores a relacionar a sequência das etapas do experimento, interpretar resultados, estudar a derivação dos dados envolvidos e *etc.*. Apesar de representarem um fator fundamental nos experimentos, as bases de proveniência isoladas, nem sempre fornecem todo o conhecimento necessário para os cientistas analisarem os resultados das suas pesquisas. O poder analítico de dados de proveniência depende muito se os mesmos se encontram associados a dados de domínio [de Oliveira et al. 2015]. Por exemplo, no caso do SciPhy, o *workflow* é composto por programas que alinham sequências, além de gerar árvores filogenéticas. Cada uma dessas sequências representa um gene/genoma de um organismo de interesse, e requere-se que as

*Os autores agradecem à CAPES, CNPq e FAPERJ por financiarem parcialmente este trabalho

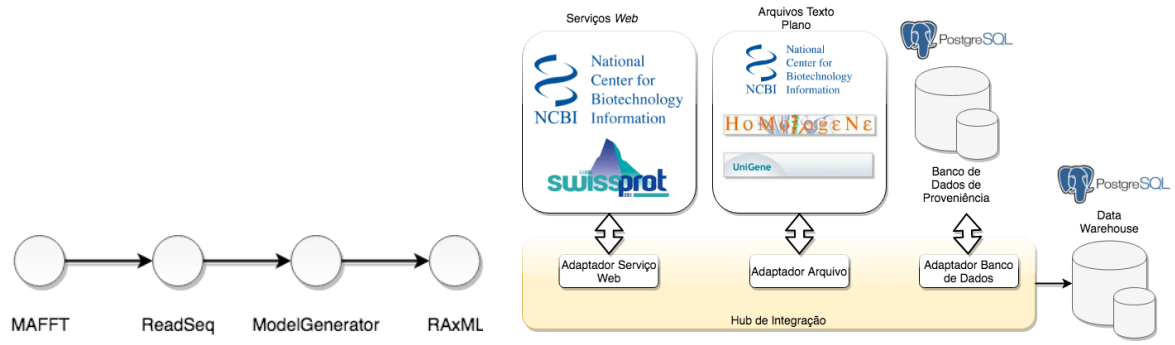

Figura 1. (a) O *Workflow* SciPhy (b) Arquitetura da Abordagem Proposta

informações e metadados sobre tal organismo estejam integrados, para produzir e analisar resultados respaldados por informações científicas relevantes. Muitas vezes, esses dados de domínio necessários encontram-se desassociados dos dados de proveniência do *workflow*, o que requer a manipulação do especialista, que é laboriosa, podendo levar a erros. Enriquecer uma base de proveniência com dados de domínio não é uma tarefa simples, mas é uma abordagem usada atualmente de maneira manual pelos cientistas (biólogos e geneticistas). Abordagens existentes já propuseram essa integração [de Oliveira et al. 2017], porém ou elas assumem que os dados de domínio a serem integrados são definidos *a priori* ou assumem que o acesso é sempre realizado por *Web Services* e nem sempre essas abordagens são possíveis. Além disso, a granularidade da informação pode ser específica para uma área (determinar uma doença genética) que pode ser necessário a manipulação de diferentes dados o que leva a integrar muitos bancos de dados. Este artigo propõe uma abordagem, chamada $BioIntegrator$, que visa a integração entre dados de domínio de diferentes fontes e dados de proveniência gerados por experimentos científicos modelados como *workflows*. Tal abordagem pode ser executada *a priori* ou *a posteriori*, dependendo da necessidade do cientista. Além disso, ela é adaptativa no que tange o acesso aos dados, podendo ser via *Web Services*, programas próprios, extratores de dados, *etc.*. Atualmente, essa integração é focada para análises filogenéticas, mas a mesma tecnologia pode ser extrapolada para outras áreas biológicas.

2. Motivação: o *Workflow* SciPhy

O SciPhy é um *workflow* de bioinformática, que gerencia de forma distribuída e paralela, sequências genéticas e constrói árvores filogenéticas evolutivas entre organismos [Ocaña et al. 2011]. O SciPhy (Figura 1(a)) é composto de quatro atividades: (I) alinhamento de sequências (MAFFT), (II) conversão de alinhamento (ReadSeq), (III) eleição do modelo evolutivo (ModelGenerator) e (IV) geração de árvores (RAxML).

Sendo assim, o SciPhy pode ser executado para múltiplos objetivos, como por exemplo comparar diversas árvores de parasitas, identificar drogas que sejam efetivas contra eles ou realizar estudos de filogeografia e filodinâmica para estudar a propagação entre continentes (*e.g.* Ebola e Zika). O SciPhy foi implementado no SciCumulus [de Oliveira et al. 2010]. A base de proveniência do SciCumulus contém informações do *Workflow*, suas atividades, ativações (execuções de atividades), arquivos produzidos e parâmetros consumidos [de Oliveira et al. 2017]. Entretanto, os dados de domínio não se encontram integrados de forma natural a essa base, pois depende do cientista a escolha de quais bases de dados serão as mais informativas dependendo da pesquisa. Desta forma, a proposta consiste na importação de tais informações para a base de proveniência para enriquecê-la e tornar as análises dos cientistas mais completas. A seguir, apresentamos

a proposta de um arcabouço genérico para enriquecimento de bases de proveniência de *workflows* filogenéticos, que pode ser utilizado por diferentes SWfs.

3. Abordagem Proposta: $BioIntegrator$

De acordo com o que foi apresentado anteriormente, podemos perceber que o cientista necessita de um acesso integrado a múltiplas fontes de dados, desde bancos de dados tradicionais (bancos de dados de proveniência) até dados semiestruturados ou não estruturados, como domínio biológico. Nesse sentido, uma abordagem de enriquecimento de bancos de dados de proveniência visa oferecer um acesso uniforme a fontes de dados distribuídas e heterogêneas. Essa abordagem pode se basear em duas diferentes arquiteturas: (I) Abordagem virtual [Chawathe et al.], onde os dados de proveniência e de domínio permanecem isolados e são integrados via consultas, e (II) Abordagem materializada, onde todos os dados são acessados, limpos, integrados e armazenados em um *Data Warehouse* [Widom 1995] e as consultas analíticas são submetidas ao mesmo *Data Warehouse*.

A Figura 1(b) apresenta a arquitetura da abordagem proposta composta de quatro componentes principais: (I) Fontes externas de dados que podem ser serviços *Web*, bancos de dados ou arquivos, (II) Base de dados de proveniência, que contém todo o histórico de execução do *workflow*, (III) Hub de integração, que contém componentes adaptadores que extraem informações das mais diversas fontes e as integram no (IV) *Data Warehouse* de proveniência que contém dados históricos e de domínio. Nesse contexto, diferentes fontes de dados externas podem ser importadas para o banco de dados de proveniência para o enriquecimento de informações relevantes durante a análise dos dados. *A priori*, utilizaremos três diferentes fontes de dados em nossa abordagem: (I) a base de dados de proveniência do SciPhy no SciCumulus, (II) o NCBI *Taxonomy database*[1], e (III) o *HomoloGene database*[2]. O NCBI *Taxonomy Database* é uma base de dados de informações taxonômicas e filogenéticas, entre outras fontes [Federhen 2011]. A base *HomoloGene* contém informações do gene, proteína, etc. e, juntamente com o NCBI *Taxonomy Database*, pode ser uma fonte rica de informações para a base de proveniência. Para integrar tais bases, optou-se por utilizar a abordagem *Global-As-View* (GAV) [Halevy 2000] que requer que cada objeto do esquema global (o *Data Warehouse*) seja expressado como uma visão de banco de dados a partir das fontes externas. Apesar de ser um trabalho em andamento, a integração foi realizada com sucesso e a abordagem está em processo de validação e mostra-se promissora no que tange oferecer capacidade analítica aos cientistas.

Para exemplificar o apoio oferecido pela abordagem aos cientistas em suas análise e levando em conta as bases de dados supramencionadas, uma possível consulta que seria facilitada é: quais são as categorias, proteínas e nucleotídeos de sequências homólogas, cuja árvore filogenética foi gerada pelo *Sciphy*? Sem a abordagem proposta, os cientistas deveriam buscar no *Homologeny Database* sequências homólogas (que não se encontram alinhadas), as usaria como *input* para o *Sciphy* que por sua vez as alinharia e geraria uma árvore filogenética, e para cada sequência os pesquisadores em questão buscariam no *Taxonomy Database* as categorias, as proteínas e os nucleotídeos das sequências contidas na árvore filogenética já alinhadas. Como mencionado, estas etapas manuais são trabalhosas e podem levar a erros.

4. Trabalhos Relacionados

O problema de integração entre bases de dados biológicas não é novo [Thiam Yui et al. 2011]. Em [Thiam Yui et al. 2011], três soluções são apresentadas: um banco de dados federado, a

[1]https://www.ncbi.nlm.nih.gov/taxonomy
[2]https://www.ncbi.nlm.nih.gov/homologene

abordagem de *data warehousing* (utilizada nesse artigo) e a abordagem baseada em links. [Hernandez and Kambhampati 2004] também discutem diversas abordagens de integração e definem que *data warehousing* é a que oferece mais vantagens.

5. Discussões e Trabalhos Futuros

Neste artigo apresentamos a proposta do $BioIntegrator$, uma abordagem para integração de dados de proveniência com dados de domínio que podem ser armazenados e consultados de diferentes maneiras. O objetivo do $BioIntegrator$ é integrar diferentes bases de dados científicas de domínio com bases de dados de proveniência de SWf, para que juntas aumentem e facilitem a extração do conhecimento de uma determinada área científica. Inicialmente foram incorporadas as bases do NCBI *Taxonomy Database* e do *HomoloGene*, porém planejamos integrar outras bases de dados como o UniProt *database*[3] e de vias metabólicas como o KEGG textitdatabase[4]. Planejamos também exportar o *Data Warehouse* para um banco de dados noSQL de forma a aumentar a escalabilidade da abordagem proposta e desta maneira aprimorar a arquitetura de integração apresentada na Seção 3.

Referências

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J. The tsimmis project: Integration of heterogeneous information sources.

de Oliveira, D., Ogasawara, E., Baião, F., and Mattoso, M. (2010). Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In *2010 IEEE CLOUD*, pages 378–385.

de Oliveira, D., Silva, V., and Mattoso, M. (2015). How much domain data should be in provenance databases? In *TaPP 15*, Scotland.

de Oliveira, W. M., Ocaña, K. A. C. S., de Oliveira, D., and Braganholo, V. (2017). Querying provenance along with external domain data using prolog. *JIDM*, 8(1):3–18.

Federhen, S. (2011). The ncbi taxonomy database. *Nucleic acids research*, 40(D1):D136–D143.

Felsenstein, J. (1996). [24] inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. In *M. in enzym.*, volume 266, pages 418–427.

Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for computational tasks: A survey. *Computing in Science and Engg.*, 10(3):11–21.

Halevy, A. Y. (2000). Theory of answering queries using views. *SIGMOD Rec.*, 29(4):40–47.

Hernandez, T. and Kambhampati, S. (2004). Integration of biological sources: Current systems and challenges ahead. *SIGMOD Rec.*, 33(3):51–60.

Ocaña, K. A. C. S., de Oliveira, D., Ogasawara, E., Dávila, A. M. R., Lima, A. A. B., and Mattoso, M. (2011). Sciphy: A cloud-based workflow for phylogenetic analysis of drug targets in protozoan genomes. In *BSB*, pages 66–70.

Thiam Yui, C., Liang, L. J., Jik Soon, W., and Husain, W. (2011). A survey on data integration in bioinformatics. In *Inf. Eng. and Inf. Sci.*, pages 16–28.

Widom, J. (1995). Research problems in data warehousing. In *CIKM'95*, CIKM '95, pages 25–30, New York, NY, USA. ACM.

[3]http://www.uniprot.org/statistics/Swiss-Prot

[4]http://www.genome.jp/kegg/pathway.html

Rumo à Otimização de Operadores sobre UDF no Spark*

João Antonio Ferreira[1], **Fábio Porto**[2], **Rafaelli Coutinho**[1], **Eduardo Ogasawara**[1]

[1]CEFET/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

[2]LNCC - Laboratório Nacional de Computação Científica

joao.parana@acm.org, fporto@lncc.br

rafaelli.coutinho@cefet-rj.br, eogasawara@ieee.org

Abstract. *Large-scale data analysis has gained much importance in the scientific community due to the Big Data phenomenon. In this context, user-defined functions (UDFs) are commonly implemented in frameworks such as Apache Spark to enable large-scale data analysis. However, the use of UDF brings challenges in optimization of execution as they are opaque. This work proposes a method of optimizing data analysis workflows supported by UDF on Apache Spark. This method is based on SparkSQL's Catalyst API and Scala language macros.*

Resumo. *A análise de dados em larga escala tem ganhado muita importância na comunidade científica devido ao fenômeno do Big Data. Neste contexto, funções definidas pelo usuário (UDF) são, comumente, implementadas em frameworks como Apache Spark para viabilizar a análise de dados em larga escala. No entanto, o uso de UDF traz desafios no processo de otimização de execução pois são opacas. Este trabalho propõe um método de otimização de workflows de análise de dados apoiadas em UDF sobre o Apache Spark. Tal método é baseado na API Catalyst do SparkSQL e em macros da linguagem Scala.*

1. Introdução

A resolução de diversos problemas científicos é computacionalmente intensiva e otimizações voltadas às plataformas de hardware específicas (GPU e FPGA) e algoritmos e métodos matemáticos utilizados não são suficientes devido o aumento da quantidade de fontes de dados. Para se conduzir as análises de dados neste contexto, surgiram soluções para processamento de grandes volumes de dados escaláveis horizontalmente, tal como o modelo MapReduce, implementado em *frameworks* como Apache Spark [Zaharia et al., 2016].

Funções definidas pelo usuário (UDF) são comumente usadas nesses *frameworks* de modo a viabilizar a análise de dados em larga escala em experimentos científicos. Apesar de sua importância, o uso de UDF traz desafios no processo de otimização de execução, devido a dificuldade de se estabelecer semântica clara sobre seus comportamentos que possuem códigos relacionados ao domínio. Com a falta de conhecimento sobre a

*Os autores agradecem à FAPERJ, à CAPES e ao CNPq pelo financiamento do trabalho.

semântica da operação, o Spark é incapaz de otimizar a execução [Armbrust et al., 2015]. Este trabalho propõe um método para otimização das execuções de *workflows* de análise de dados, que possam ser escritas como UDF sobre o Apache Spark. Como primeiro estudo, avaliou-se a viabilidade do processo de otimização da execução de mapeamentos (operador *map*) operando sobre UDF usando a API *Catalyst* do Spark para abordar o problema da execução de atividades restritas (*Constrained Activity*)[1] [Ogasawara et al., 2011]. Uma avaliação preliminar aponta a viabilidade da abordagem.

Além desta introdução, o trabalho está organizado em mais três seções. Na seção 2, são apresentados os conceitos gerais necessários para o entendimento do problema e a abordagem adotada sob a forma de prova de conceito. A seção 3 traz a avaliação preliminar que corrobora com a hipótese. Por fim, a seção 4 apresenta as conclusões.

2. Prova de conceito

O Spark é um *framework* que possibilita a execução das tarefas paralelizáveis de forma distribuída em máquinas *multi-core* ou *clusters* YARN/Mesos/Kubernetes, com ênfase no processamento em *pipeline* das atividades que compõem um *dataflow* [Zaharia et al., 2016]. Ele foi adotado para apoiar os usuários na execução de *workflow* de análise de dados com UDF em ambientes de processamento distribuído em larga escala [Ferreira et al., 2017]. A vantagem desta abordagem é deixar a responsabilidade sobre a complexidade do modelo de execução para o Spark. Isso diferencia a proposta de Ferreira et al. [2017] dos outros sistemas de gerenciamento de *workflow* (SGW). Desta forma, o foco se restringe a anotação semântica de UDF que tanto estabelece a relação de consumo e produção de ativações [Ogasawara et al., 2011] quanto na associação com a proveniência para prover informação necessária a otimização do *workflow*.

Nesta prova de conceito foi realizada uma análise exploratória das APIs do *Catalyst* pertencente ao módulo *SparkSQL* e dos seus pontos de extensão. O *SparkSQL* usa o componente *Catalyst* para otimizar a geração de código em tempo de execução à partir de expressões e comandos SQL, e os pontos de extensão do Spark possibilitam a personalização do ambiente com o uso de novas regras para otimização de *dataflow* de acordo com o domínio do problema.

Na álgebra de *workflow*, o operador *wMap* é definido: $T \leftarrow wMap(A, R)$, onde A é a atividade que produz uma única tupla na relação de saída T para cada tupla consumida na relação de entrada R. Os esquemas de R e T podem ser diferentes num caso mais geral. Portanto, é possível observar que a semântica do operador *wMap* é compatível com seu análogo *map* do *SparkSQL* [Ogasawara et al., 2011; Armbrust et al., 2015]. Considere o seguinte exemplo especificado pela álgebra de *workflow*: $T \leftarrow wMap(B, wMap(A, R))$. Ao implementar este *dataflow* no Spark, o *Catalyst* do *SparkSQL* usará a regra *CollapseProject* na fase de otimização. Esta regra junta as duas expressões dos dois *map* em uma, executando A e B em *pipeline*. Isso pode ser indesejado no caso em que estas atividades forem escritas em código nativo e usarem muitos recursos do hardware. Isto caracteriza, segundo Ogasawara et al. [2011], a existência de uma atividade restrita (*Constrained Ac-*

[1]Nas aplicações científicas é muito comum invocar programas com implementações paralelas que usam todos os núcleos (*cores*) disponíveis em um nó do *cluster*. Devido a essa característica, a ativação de uma atividade restrita bloqueia todos os núcleos, inibindo a utilização por outras ativações. Ogasawara et al. [2011] chama este fenômeno de atividade restrita (*Constrained Activity*).

tivity), onde uma atividade, regida por um operador *wMap* e executada em um *dataflow*, pode consumir muitos recursos computacionais impedindo que outras atividades também regidas por um operador *wMap* possam ser executadas em *pipeline*. Para solucionar isso, Ogasawara et al. [2011] propõem a criação de uma barreira forçando a materialização do resultado intermediário entre a execução das atividades.

Com o objetivo de otimizar este tipo de *workflow*, foi criada uma classe[2] escrita em linguagem Scala para funcionar como ponto de extensão e otimizar operadores *map*. Além dos pontos de extensão do *SparkSQL*, esta classe usa as funcionalidades *quasi-quotes* e macros da linguagem Scala para atuar no *dataflow* antes mesmo de passar o plano lógico da *query* ao *Catalyst*. Ela consiste de uma *case class* do Scala e é responsável por transformar planos lógicos por meio de otimizações baseadas em transformações algébricas. O objetivo desta classe é incluir uma barreira de materialização entre dois operadores *map* adjacentes. Neste caso, isso só é possível se os operadores atuarem sobre UDF anotadas com semântica de utilização excessiva de recursos em tempo de execução, informações estas obtidas da proveniência.

3. Avaliação Preliminar

Nesta seção, procurou-se avaliar a capacidade do Spark em otimizar um *dataflow* com UDF anotada com informações sobre uso excessivo de recursos usando o operador $wMap$, como ilustrado abaixo em código Scala:

```
myDataset.map(heavyUDF1(some, parameters)).map(heavyUDF2(param, ...))
```

O Spark combina, por padrão, os processamentos dessas duas UDF em *pipeline*, pois desconhece a semântica da operação executada e os recursos computacionais consumidos. Caso estas UDF sejam escritas em código nativo, cada uma delas pode consumir todos os recursos de *thread* ou memória do processador de forma que não possam ser invocadas em *pipeline*. Conforme já mencionado, isto caracteriza uma atividade restrita (*Constrained Activity*) e a solução é a criação de uma barreira forçando a materialização do resultado entre a avaliação das duas UDF.

A anotação semântica pode ser feita via sufixo indicando a quantidade de memória RAM utilizada em média ou o número de *threads*/*cores* do processador, ou ambos. Esta anotação é usada na fase inicial para dividir o *workflow* original em dois segmentos e ordená-los antes de passar ao *Catalyst* para que seja feita as outras otimizações possíveis em cada um dos segmentos. As materializações no Spark podem ser provocadas por operadores *collect* aplicados no *dataset*. Esta abordagem foi escolhida, por modificar apenas a representação do dado, preservando o esquema. O importante é que a criação da barreira não modifica a semântica do *dataflow* original.

Para uma avaliação preliminar, a criação de barreira foi aplicada sobre o *workflow* mostrado na Figura 1(a). Nele, tem-se uma relação `R0` onde são aplicados dois operadores *map* em sequência. O primeiro operador rege a udfA que após anotação semântica se transformou em udfA_M3G. O segundo operador rege a udfB que anotado se transformou em udfB_M3G. O sufixo M3G significa que o processo externo invocado pela UDF usa

[2]O código fonte Scala pode ser acessado em https://github.com/joao-parana/wff-catalyst

em média 3GB de memória RAM para cada tupla consumida. Supondo que o subsistema de proveniência tenha informado que o sistema sofrerá com problemas de desempenho ou confiabilidade nestas condições, o *dataflow* pôde ser modificado para materializar o resultado do primeiro *map*. A Figura 1(b) mostra esta situação. Uma relação `R1` foi criada com o propósito de materializar o resultado intermediário para que o Spark não realize a otimização padrão para dois operadores *map* adjacentes. Com isso, o *dataflow* deixou de ser executado em *pipeline* permitindo sua finalização, o que não seria possível na versão não original, pois o programa terminaria logo no início por falta de recursos computacionais disponíveis.

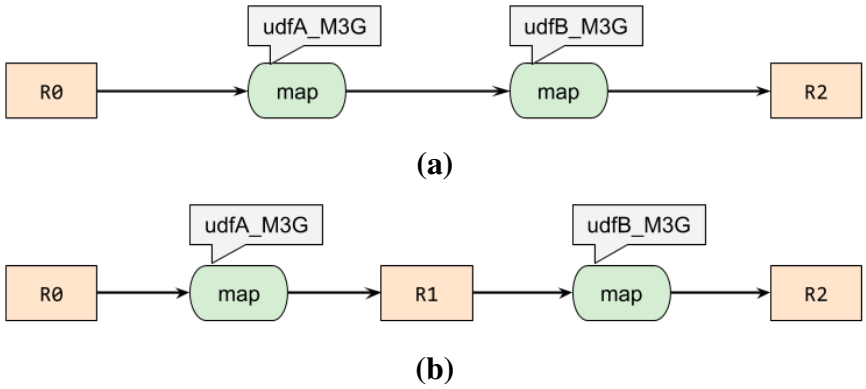

Figura 1. *workflow* antes (a) e depois (b) da inclusão da barreira.

4. Considerações finais

Em Ferreira et al. [2017] foi abordado a possibilidade de usar o Spark como base para a implementação de um *framework* de análise de dados usando a álgebra de *workflow* para facilitar o trabalho dos cientistas. Este presente trabalho aponta que esta abordagem é viável e que é possível criar implementações de otimizações arbitrárias incluindo UDF com o uso dos pontos de extensão disponíveis no SparkSQL. Isso inclui o caso da atividade restrita (*Constrained Activity*) com atividades legadas escritas em código nativo (C++, Fortran, Python, Cython, R, etc.) [Ogasawara et al., 2011].

Referências

Armbrust, M., Xin, R., Lian, C., Huai, Y., Liu, D., Bradley, J., Meng, X., Kaftan, T., Frankliny, M., Ghodsi, A., and Zaharia, M. (2015). Spark SQL: Relational data processing in spark. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 2015-May, pages 1383–1394.

Ferreira, J., Gaspar, D., Monteiro, B., Silva, A. B., Porto, F., and Ogasawara, E. (2017). Uma Proposta de Implementação de Álgebra de Workflows em Apache Spark no Apoio a Processos de Análise de Dados. In *Brazilian e-Science Workshop*.

Ogasawara, E., de Oliveira, D., Valduriez, P., Dias, J., Porto, F., and Mattoso, M. (2011). An algebraic approach for data-centric scientific workflows. In *Proceedings of the VLDB Endowment*, volume 4, pages 1328–1339.

Zaharia, M., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., Stoica, I., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., and Venkataraman, S. (2016). Apache spark: A unified engine for big data processing. *Communications of the ACM*, 59(11):56–65.

Uma Abordagem para Identificação de Padrões de Ocorrência de Eventos Solares Transientes Baseada no Fluxo de Múons*

Mariana Teixeira[1], **Daniel de Oliveira**[1]

[1]Instituto de Computação – Universidade Federal Fluminense (IC/UFF)

teixeiramariana@id.uff.br, danielcmo@ic.uff.br

Resumo. *O múon é uma partícula elementar com carga negativa, e é a componente carregada mais abundante da radiação cósmica que penetra profundamente no solo da Terra. Analisar o fluxo de múons que chega ao nosso planeta é importante para se identificar eventos solares transientes que podem causar prejuízos à população. Esse artigo apresenta uma abordagem baseada em algoritmos de mineração de sequências para automatizar o processo de identificação de padrões frequentes no fluxo de múons capturado. Experimentos realizados mostram que a abordagem é capaz de identificar eventos solares transientes por meio dos padrões frequentes identificados.*

Abstract. *The muon is an elementary particle with a negative charge, and is the most abundant charged component of cosmic radiation that penetrates deep into the Earth's soil. Analyzing the flow of muons that reaches our planet is important to identify solar eruptions that can cause harm to the population. This paper presents an approach based on sequence mining algorithms to automate the process of identifying frequent patterns in the captured muon stream. Experiments carried out show that the approach is capable of identifying eruptions by means of the identified frequent patterns.*

1. Introdução

O Múon é a única partícula com carga elétrica capaz de penetrar profundamente no subsolo terrestre. Tal partícula se torna muito importante nas pesquisas científicas uma vez que a medição do fluxo de múons permite estudar eventos solares transientes Os telescópios *New*-Tupi [Augusto et al. 2012] fazem a detecção de múons (aproximadamente 45.000 leituras por dia) e geram um grande volume de dados, que são atualmente armazenados em arquivos binários. O uso de arquivos é um importante limitador no que tange a execução de consultas e análises, que são fundamentais, uma vez que tais eventos podem desencadear a suspensão de diversas atividades eletromagnéticas, como por exemplo suspender as transmissões das estações de rádio.

O objetivo desse artigo é automatizar o processo de identificação de leituras do fluxo de múons [Augusto et al. 2017], por meio da aplicação de algoritmos de mineração de dados, mais especificamente de mineração de sequências [Han et al. 2011], para identificar padrões de eventos solares transientes. Para avaliar a abordagem proposta utilizamos o *dataset* dos telescópios *New*-Tupi para identificar os padrões frequentes e um segundo *dataset* fornecido pela NASA *(National Aeronautics and Space Administration)*, que contém notícias relacionadas à eventos solares transientes captados (mais especificamente erupções solares), como gabarito para avaliar se os padrões identificados pelos dados dos telescópios *New*-Tupi correspondem de fato a ocorrência de eventos solares transientes.

*O trabalho aqui apresentado foi parcialmente financiado por CNPq, CAPES e FAPERJ

O artigo está organizado da seguinte forma: a Seção 2 apresenta o referencial teórico. A Seção 3 apresenta a abordagem proposta. A Seção 4 apresenta a avaliação experimental e por fim, a Seção 5 conclui esse artigo e apresenta trabalhos futuros.

2. Referencial Teórico

Esta seção tem como objetivo apresentar conceitos importantes para a compreensão da abordagem apresentada nesse artigo.

2.1. Mineração de Sequências

A Mineração de Sequências é um método de Mineração de Dados, que consiste em analisar um determinado *dataset* de sequências, onde cada sequência é composta por conjuntos de itens. Formalmente, uma sequência ∂ é representada por $\{e_1, e_2, e_3, e_n\}$, onde cada e_j, $1 \leq n$ é um evento de a sequência ∂ e e_1 ocorre antes de e_2, que ocorre antes de e_3 e assim por diante. Existem diversos algoritmos para a mineração de sequências, como o *SPAM* [Ayres et al. 2002].

2.2. Os Telescópios *New*-Tupi

Os telescópios *New*-Tupi fazem parte de uma classe de telescópios que visa detectar múons carregados. Eles trabalham de forma sincronizada para medir continuamente o fluxo de partículas derivadas da radiação do Sol, investigando as possíveis relações entre os ciclos solares e as variações climáticas da Terra [Augusto et al. 2017].

Os telescópios *New*-Tupi são construídos a partir de quatro detectores construídos com base em um cintilador plástico (Eljen EJ-208) e uma fotomultiplicadora (Hamamatsu R877). Quando uma partícula carregada rápida (*e.g.* um múon) atravessa o cintilador, este emite uma luz fluorescente que é captada por uma fotomultiplicadora, que são sensores ópticos extremamente sensíveis a luz. A partir disso, a fotomultiplicadora converte a luz de baixa intensidade em um sinal elétrico, que é pré-amplificado até uma amplitude suficiente para posterior análise [Augusto et al. 2017]. Na lógica implementada na aquisição de dados de cada telescópio, os sinais analógicos dos detectores são digitalizados utilizando a técnica de instrumentos virtuais e as ferramentas do *software* Lab-VIEW. Ademais, cada telescópio usa um sistema anti-coincidência.

3. Abordagem Proposta

A abordagem proposta nesse artigo interage com os telescópios *New*-Tupi para identificar padrões frequentes no fluxo de múons a fim de identificar eventos solares transientes. A Figura 1 apresenta a arquitetura conceitual da abordagem proposta. O processo se inicia com o telescópio realizando as leituras do fluxo de múons e armazenando tais resultados em um arquivo binário. Ao final do dia, um *script* copia os dados para um repositório na nuvem. O componente **ETL** (do inglês *Extract Transform Load* - Figura 1 - Passo 1) acessa os arquivos produzidos e converte os dados para o formato CSV. A partir do arquivo CSV gerado, o componente de **Discretização** é ativado (Figura 1 - Passo 2). Esse componente invoca o Orange [1]. que discretiza os valores de leituras do telescópio. Uma vez que os dados se encontram discretizados o componente de **Mineração** é ativado (Figura 1 - Passo 3). Esse componente executa o algoritmo *SPAM* por meio da biblioteca SPMF [2]. O *SPAM* é um algoritmo que foi proposto por [Ayres et al. 2002] para minerar padrões de sequências, e que é eficiente quando os padrões de sequências são muito grandes. Uma vez que a atividade de **Mineração** é concluída, o modelo gerado já pode ser utilizado.

[1](https://orange.biolab.si/)

[2](www.philippe-fournier-viger.com/spmf/)

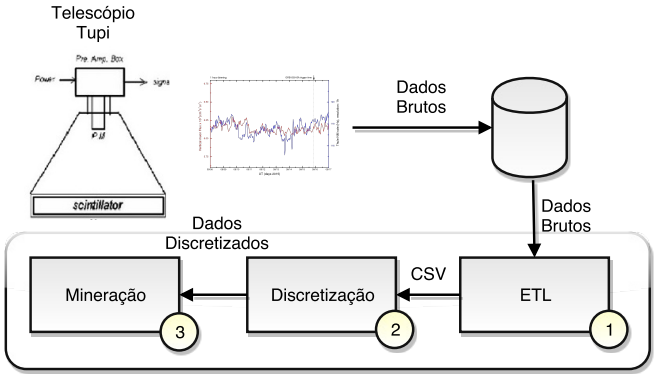

Figura 1. Arquitetura Conceitual da Abordagem Proposta

4. Avaliação Experimental

Para avaliar a abordagem proposta, uma amostra com 3.000.000 de leituras realizadas pelo telescópio *New*-Tupi foi utilizada. O *dataset* contém dados do fluxo de múons captados entre 4 de setembro e 12 de Novembro de 2014. Inicialmente, foram extraídas as *features* do repositório para serem adicionadas a um arquivo CSV, a saber: data e horário da leitura, valor *vertical* e valor *escaler*. Essa transformação foi realizada para que o Orange pudesse discretizar os dados. Os valores foram discretizados em 5 intervalos utilizando o método *Equal Frequency Discretization*. Parte do arquivo de saída é apresentado na Figura 2.

valor vertical	valor escaler	data
$\geq$ 14.5	$\geq$ 105.5	2014-09-04T21:00
$<$ 8.5	94.5 - 99.5	2014-09-04T21:00
10.5 - 12.5	$\geq$ 105.5	2014-09-04T21:00

Figura 2. Parte do arquivo de saída gerado pela discretização do *Orange*

Conforme apresentado na Figura 2, os valores contidos no arquivo são compostos por intervalos. Assim, houve a necessidade de transformar tais intervalos em valores únicos, pois uma estrutura sequencial é melhor minerada quando composta por valores únicos. Foi implementado um *script Python (Orange2SPMF)* que transforma o arquivo de saída gerado pelo *Orange* em um arquivo que contenha sequências que possam ser aceitas como entrada pelo *SPMF*. Para isso, os valores discretizados dos itens das sequências devem ser separados por -1, e o fim de uma sequência deve ser indicada por -2. Um fragmento do arquivo produzido pelo *script* pode ser observado na Figura 3.

```
2014-09-04T21:00,9 -1 100 -1 11 -1 95 -1 15 -1 95 -1 9 -1 89 -1 15 -1 95 -1 11 -1 100 -1 13 -1 100
2014-09-04T21:01,13 -1 89 -1 9 -1 95 -1 9 -1 106 -1 11 -1 106 -1 15 -1 100 -1 11 -1 89 -1 11 -1 106 (...) -2
```

Figura 3. Fragmento do arquivo gerado pela execução do *script Python Orange2SPMF*

O algoritmo *SPAM* foi executado consumindo o arquivo anteriormente citado, explorando diversas combinações dos parâmetros *minsup* (suporte) e *Min Pattern Length* (quantidade mínima de elementos nos padrões encontrados) porém com *Max Gap* com valor fixo 1, que não permite que haja intervalos entre os elementos. Os diversos arquivos de saída do SPMF com os padrões encontrados foram integrados em um único arquivo.

Após a mineração, foi necessário encontrar em quais datas ocorreram cada padrão descrito no arquivo com as saídas agregadas. Essas novas informações obtidas foram consolidadas em um novo arquivo, que é composto por uma coluna relacionada ao padrão

encontrado, uma coluna relacionada ao suporte do padrão e outra coluna relacionada às datas em que esse padrão ocorreu.

Para avaliar a abordagem proposta e validar os resultados obtidos com a mineração de sequências, verificamos, utilizando um *dataset* fornecido pela NASA (*National Aeronautics and Space Administration*), se nas datas em que os eventos frequentes foram identificados de fato ocorreram erupções solares. A *NASA* possui um site[3] que contém o arquivo referente às notícias relacionadas à eventos solares que ocorreram entre 2010 e 2015. De posse das datas e dos padrões frequentes, foi possível analisar no site da *NASA* os dias e classificações dos eventos solares transientes que ocorreram na mesma faixa de tempo dos registros coletados pelo telescópio *New*-Tupi. Com isso, foram cruzados os dados do arquivo de padrões com as datas de eventos solares identificados pela NASA, sendo assim possível criar um novo arquivo que contém os padrões, contidos nas análises de múons feitas pelo *New*-Tupi nas datas de eventos solares noticiados pela *NASA*. Como muitas sequências foram encontradas, foi necessário minerar novamente o este último arquivo visando descobrir qual o padrão de fluxo de múons que pode determinar os eventos solares noticiados pela *NASA*. Novamente, o *SPMF* foi utilizado para minerar tais sequências. O arquivo de saída pode ser observado na Figura 4. Foram identificados eventos solares em 10/09/2014, 19/10/2014 e 30/10/2014, o que coincide com o *dataset* fornecido pela NASA.

```
9 -1 89 -1 9 -1 89 -1 #SUP: 32
89 -1 9 -1 89 -1 9 -1 #SUP: 29
```

Figura 4. Saída gerada pelo *SPMF*

5. Conclusão e Trabalhos Futuros

Os telescópios *New*-Tupi realizam leituras dos fluxos de Múons que chegam ao nosso planeta. A abordagem proposta nesse artigo visa encontrar padrões nos valores referentes ao fluxo de múons capturados pelos telescópios *New*-Tupi, baseando-se no algoritmo de mineração de sequências *SPAM*. De forma a validar os resultados experimentais obtidos, cada padrão frequente identificado foi comparado com um *dataset* da NASA que descreve eventos solares identificadas pela instituição. Todas as sequências identificadas coincidem com eventos identificados pela NASA, mostrando que a aplicação da abordagem proposta é válida e promissora. Apesar de representar um avanço, novos experimentos de maior escala se fazem necessários com o *dataset* completo dos telescópios *New*-Tupi.

Referências

Augusto, C. R. A., Kopenkin, V., Navia, C. E., Tsui, K. H., and Sinzi, T. (2012). Search for a simultaneous signal from small transient events in the pierre auger observatory and the tupi muon telescopes. *Phys. Rev. D*, 86:022001.

Augusto, C. R. A., Navia, C. E., de Oliveira, M. N., Nepomuceno, A. A., Kopenkin, V., and Sinzi, T. (2017). Muon excess at sea level during the progress of a geomagnetic storm and high-speed stream impact near the time of earth's heliospheric sheet crossing. *Solar Physics*, 292(8):107.

Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435. ACM.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

[3]https://www.nasa.gov/mission_pages/sunearth/news/solar-event-2010-2015

Patrocinador Diamante

GOVERNO DO RIO GRANDE DO NORTE

Patrocinadores Bronze

ESIG

dynavideo

CABOTELECOM

Sistema FIERN

NATAL

PARNAMIRIM

Apoio Financeiro

CAPES

MINISTÉRIO DA EDUCAÇÃO

BRASIL

CNPq

Conselho Nacional de Desenvolvimento Científico e Tecnológico

nic.br

cgi.br