

Um framework para agrupar funções com base no comportamento da comunicação de dados para plataformas multiprocessadas

Rafael Ribeiro dos Santos

Orientador: Prof. Dr. Vanderlei Bonato



Conteúdo

- Introdução
- Framework Desenvolvido
- Resultados
- Conclusão
- Trabalhos Futuros

Conteúdo

- **Introdução**
 - **Contextualização e Motivação**
 - **Problema**
 - **Objetivos**
- Framework Desenvolvido
- Resultados
- Conclusão
- Trabalhos Futuros

Contexto do trabalho

- Nos últimos anos, a demanda por poder computacional aumentou devido a algumas aplicações denominadas *Supercomputing*;
- Começa-se a utilizar paralelismo em larga escala, incluindo plataformas heterogêneas compostas por aceleradores customizados.

Motivação

- Explorar todo o paralelismo existente em uma plataforma multiprocessada;
- Mapear cada grupo para um núcleo disponível na plataforma, de modo a alcançar um aumento no desempenho da aplicação;
- Utilizar o meio de comunicação de maneira mais adequado.

Problema

- Como as funções de aplicações se comportam durante sua execução?
- Quais funções devem ser agrupadas?
- Como analisar o comportamento do meio de comunicação para realizar o agrupamento?

DESAFIO

Desenvolver uma ferramenta para auxiliar no agrupamento de funções considerando a comunicação durante a execução da aplicação e o meio de comunicação existente.

Objetivos

- Realizar vários agrupamentos considerando a comunicação existente na aplicação durante sua execução;
- Agrupar as funções da aplicação de modo a explorar as características de uma plataforma multiprocessada.
- Implementação de um *framework* para realizar agrupamentos e escolher o melhor considerando a largura de banda disponível durante a execução da aplicação.

Conteúdo

- Introdução
- **Framework Desenvolvido**
- Resultados
- Conclusão
- Trabalhos Futuros

Framework desenvolvido

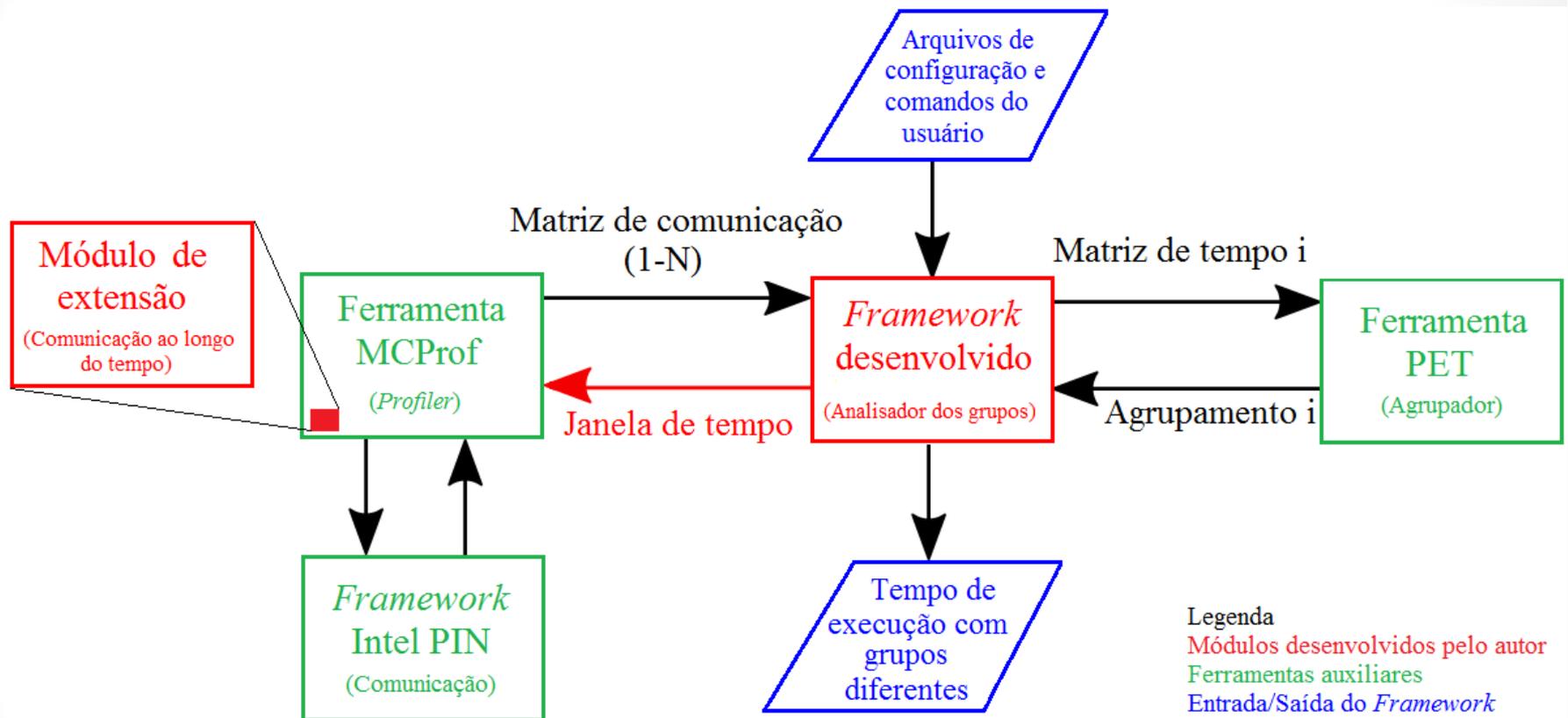


Figura 2 – Esquema geral do *framework* desenvolvido.

Inicialização do *framework*

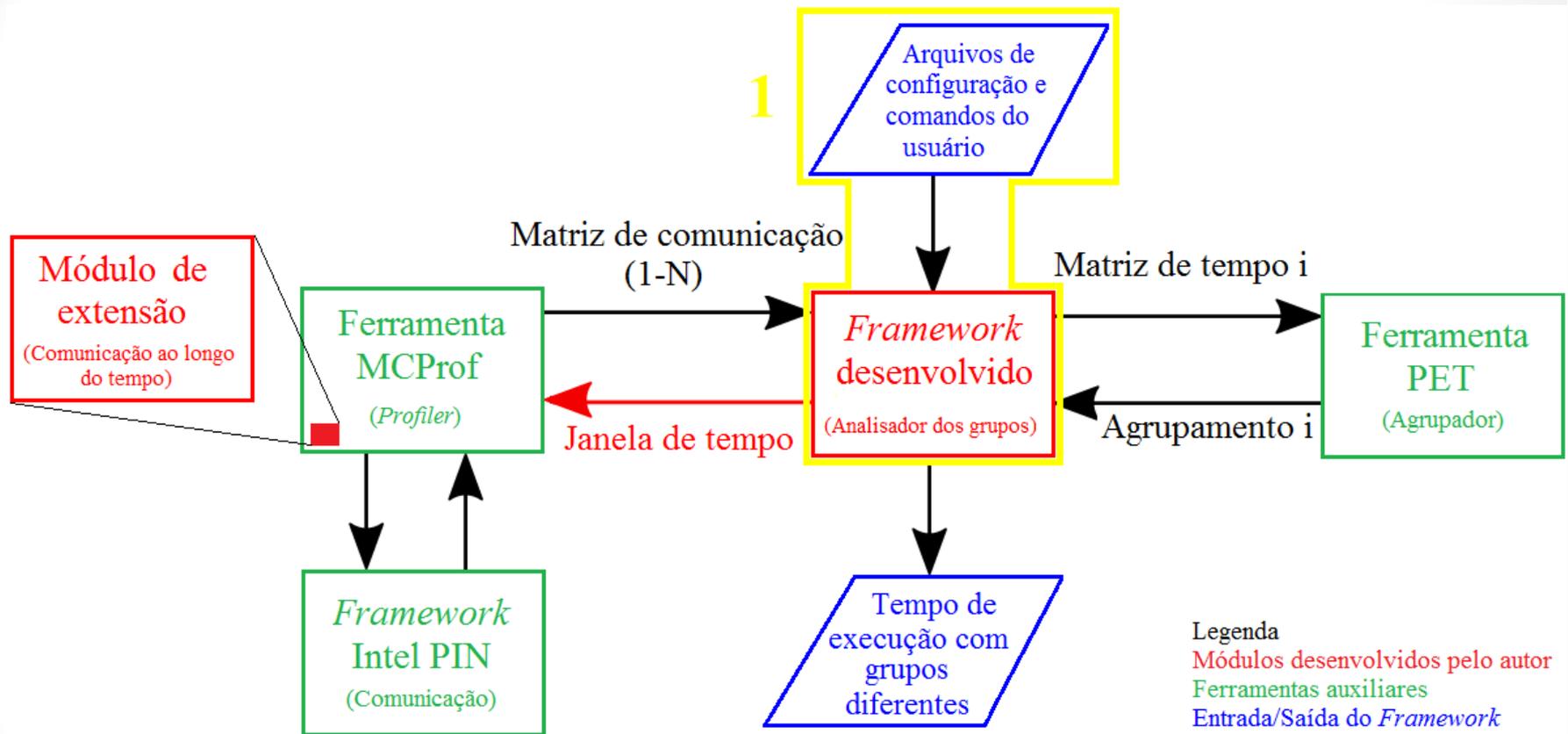


Figura 3 – Esquema geral do *framework* desenvolvido.

Inicialização do *framework*

- O framework utiliza dois arquivos de configuração para inicializar sua execução.
 - 1º Arquivo:
 - Janela de tempo;
 - Número de funções;
 - Número de grupos;
 - Largura de banda máxima;
 - Largura de banda mínima;
 - Função de transferência;
 - 2º Arquivo:
 - Caminhos para executar as ferramentas auxiliares.

Obtenção das matrizes de comunicação

2

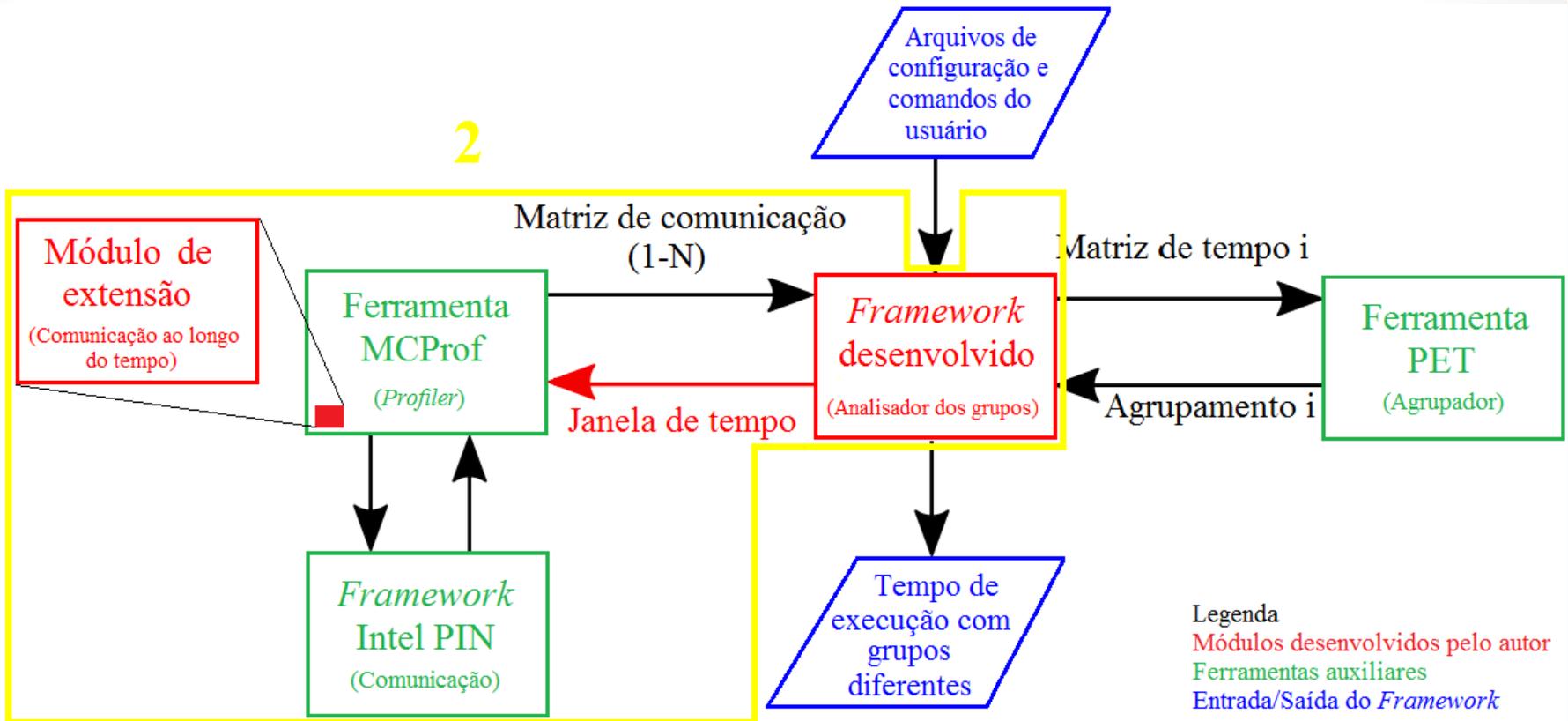


Figura 4 – Esquema geral do *framework* desenvolvido.

Obtenção das matrizes de comunicação

- Realização de uma chamada de sistema para a ferramenta MCProf (I. Ashraf and V.M. Sima and K.L.M. Bertels, 2015)
 - Obter as matrizes de comunicação;
 - Representam a quantidade de bytes trocados entre cada função.

	F1	F2	F3
F1	0	11	8
F2	0	0	16
F3	0	0	0

Figura 5 – Exemplo de uma matriz de comunicação.

- São obtidas N matrizes de comunicação.
 - Representam a comunicação existente durante a execução da aplicação.
 - É utilizado o módulo de extensão desenvolvido

Módulo de extensão

- Módulo desenvolvido para obter as matrizes de comunicação através do tempo.
- Funciona da seguinte forma:
 - Sempre que uma rotina é retornada da tabela de símbolo, é realizada uma checagem;
 - Se a checagem for positiva, é retornada uma matriz de comunicação para aquela janela de tempo;
 - A matriz de comunicação do MCProf é zerada para acumular novamente a comunicação.

Realização do agrupamento

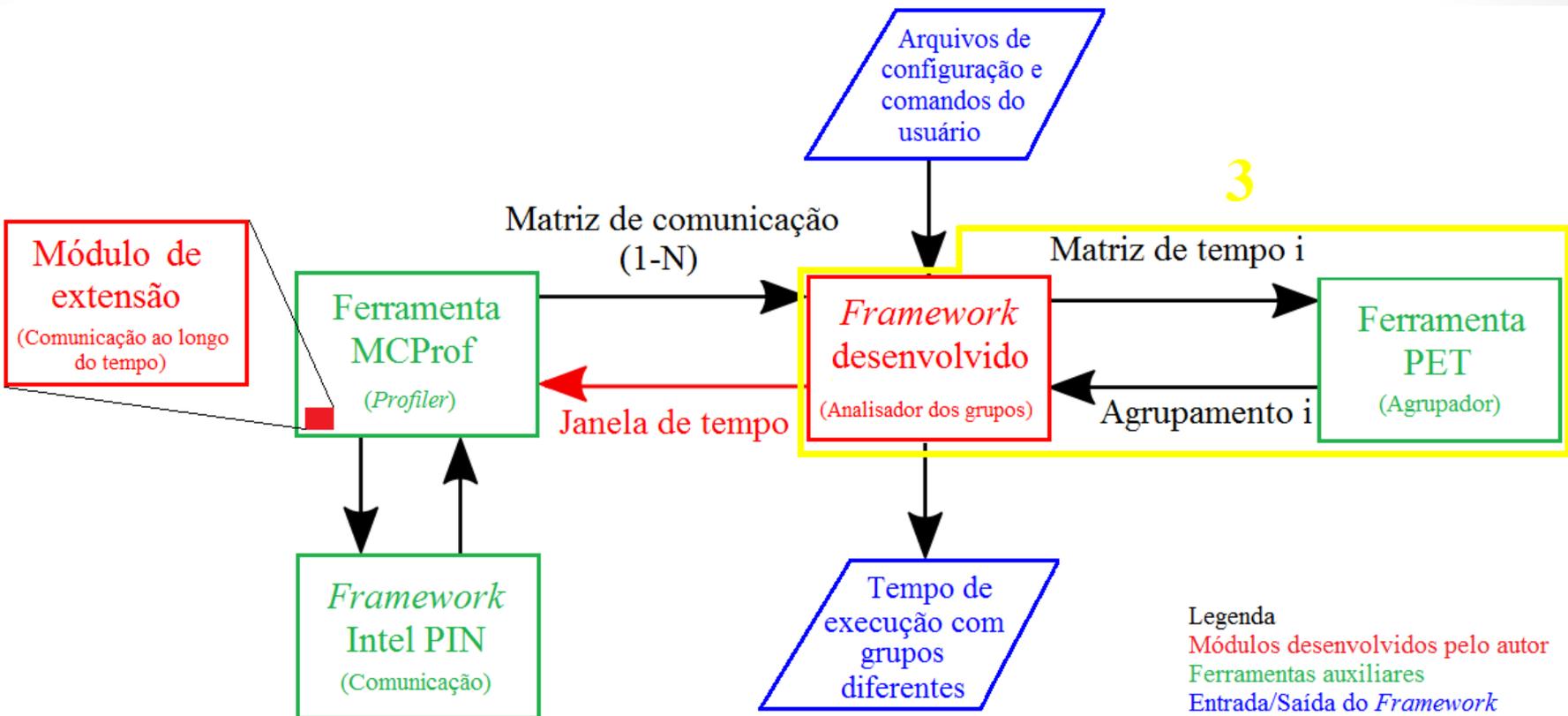


Figura 5 – Esquema geral do *framework* desenvolvido.

Realização do agrupamento

- O framework converte as matrizes de comunicação para matrizes de tempo
 - Utiliza a taxa de transferência para isso.
- Em seguida, envia cada matriz para a ferramenta PET (ASHRAF et al., 2013).
 - Somente uma matriz é enviada por vez;
- Após todas as matrizes de tempo serem agrupadas, é enviada a matriz de comunicação acumulada para encontrar um agrupamento.

Obtenção do melhor agrupamento

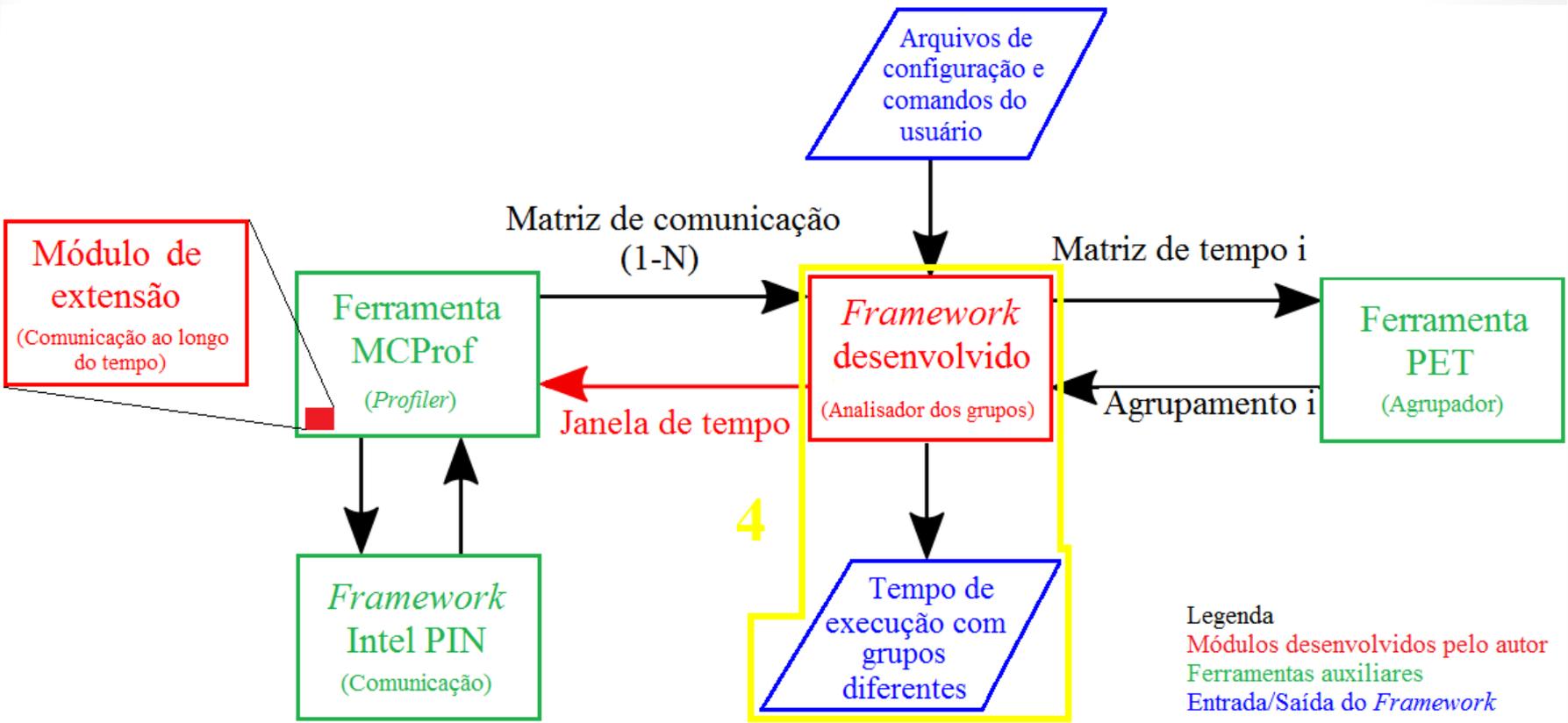


Figura 6 – Esquema geral do *framework* desenvolvido.

Obtenção do melhor agrupamento

- O framework analisa todos os grupos encontrados para escolher qual possui o menor tempo de comunicação.
 - Analisa inicialmente o agrupamento encontrado com a matriz de comunicação acumulada;
 - Em seguida, o processo é repetido com cada agrupamento encontrado para cada matriz de tempo;
 - Cada agrupamento será testado em cada matriz de tempo para obter o tempo;
 - Os resultados encontrados serão armazenados em uma lista;
 - Será escolhido o menor tempo encontrado nessa lista.
- Por fim, será apresentado o menor tempo obtido utilizando as matrizes de tempo e utilizando o método da literatura.

Conteúdo

- Introdução
- Framework Desenvolvido
- **Resultados**
- Conclusão
- Trabalhos Futuros

Resultados

- Foram utilizadas 3 funções para representar a taxa de transferência do barramento.

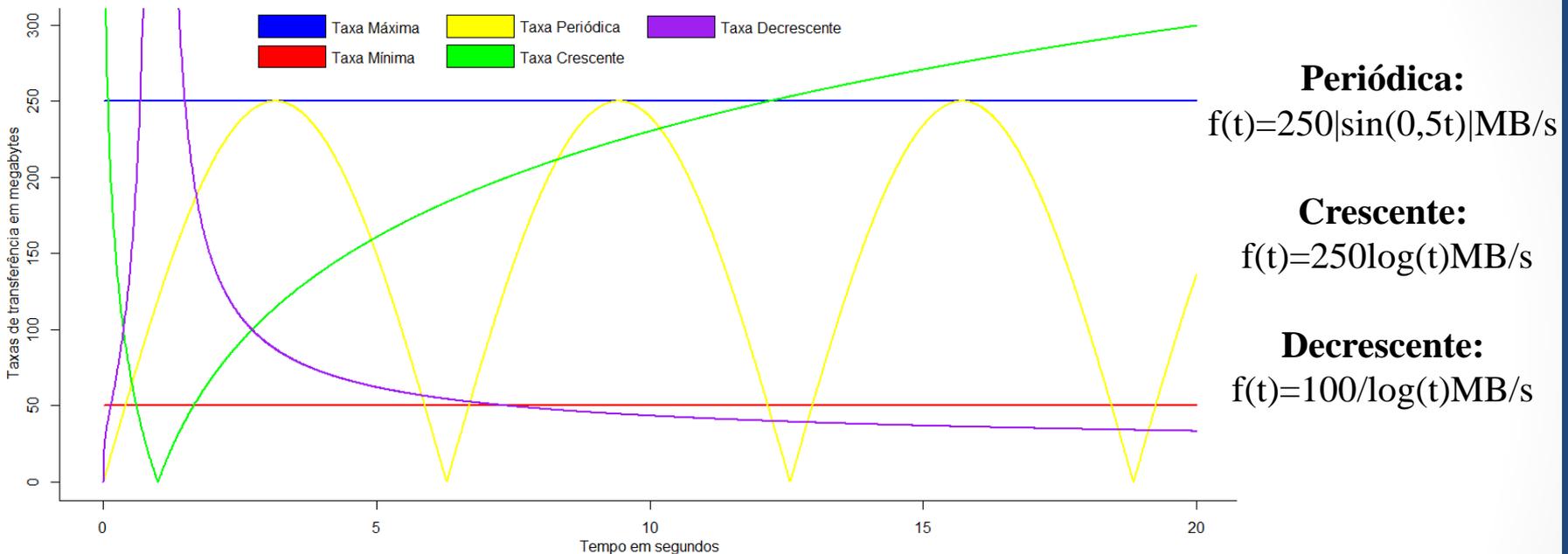


Figura 7 – Representação das taxas de transferência

- Foram realizados 30 testes com cada aplicação para cada função de transferência.
- Foram realizados os testes de hipótese **Teste t de Student** (HAYNES, 2013) e **Teste de Wilcoxon** (WILCOXON, 1945) para validar os resultados.

KLT

- KLT é uma técnica utilizada para encontrar pontos de similaridades entre duas imagens. A aplicação foi agrupada em 4 grupos.

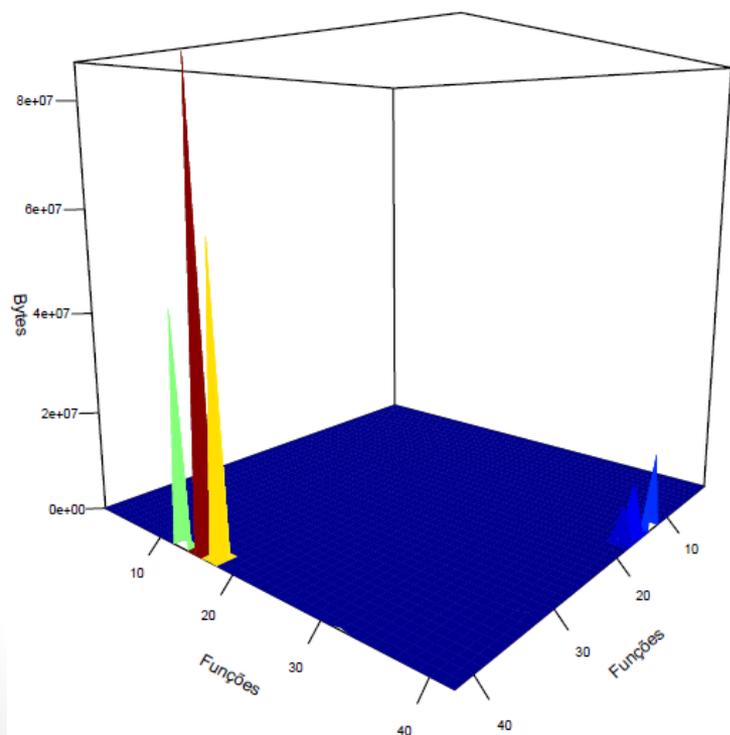


Figura 8 – Comunicação Acumulada

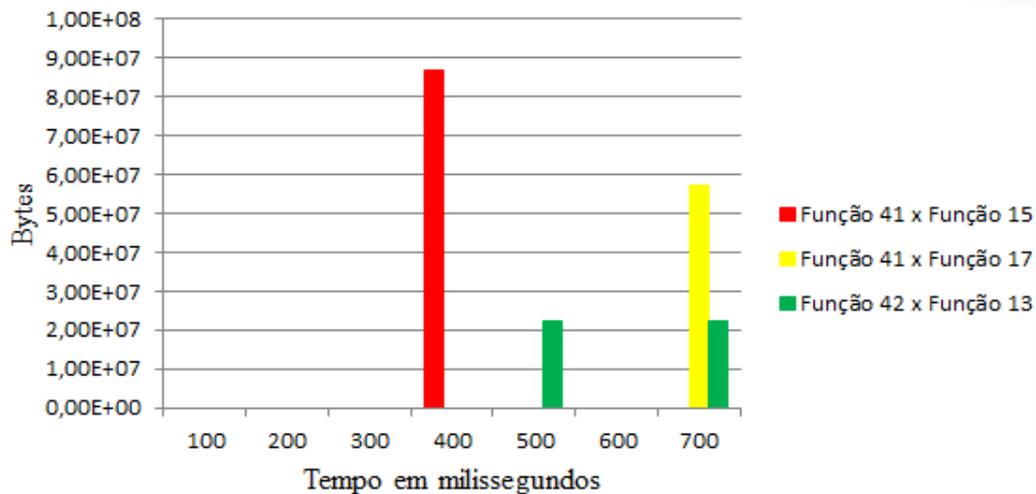


Figura 9 – Comunicação Distribuída

KLT

	Função Periódica		Função Crescente		Função Decrescente	
	Comunicação Acumulada	Comunicação Distribuída	Comunicação Acumulada	Comunicação Distribuída	Comunicação Acumulada	Comunicação Distribuída
Média	1,129	0,959	0,642	0,542	2,794	2,182
Ganho	1,177		1,184		1,281	

Tabela 1 – Média dos tempos de comunicação obtidos

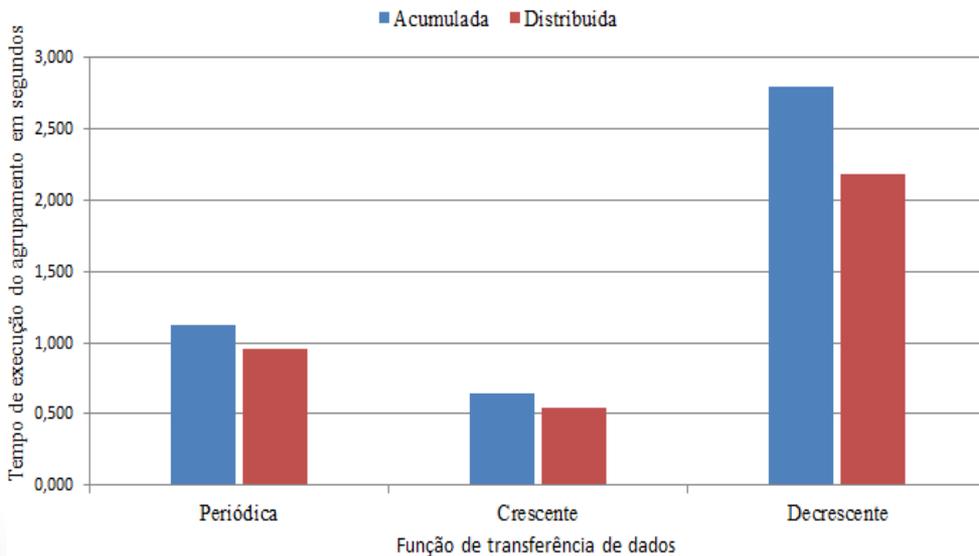


Figura 10 – Comparação dos tempos obtidos

Cluster 0 has 18 functions (12 14 40 41 ...)
 Cluster 1 has 7 functions (...)
 Cluster 2 has 6 functions (...)
 Cluster 3 has 11 functions (16 ...)

Distribuída

Cluster 0 has 11 functions (16 41 ...)
 Cluster 1 has 8 functions (12 40 ...)
 Cluster 2 has 14 functions (...)
 Cluster 3 has 9 functions (14 ...)

Acumulada

Figura 11 – Comparação dos grupos

Demais Aplicações

Aplicação	Função de transferência	Ganho
Sparse 1.3 ^a	Periódica	1,294
Barnes	Periódica	1,858
FMM	Decrescente	2,621
Ocean	Crescente	1,265

Tabela 2 – Ganho obtido por cada aplicação utilizando uma função de transferência.

Conteúdo

- Introdução
- Framework Desenvolvido
- Resultados
- **Conclusão**
- Trabalhos Futuros

Conclusão

- Utilizando a comunicação distribuída ao longo do tempo e as funções de transferência, é possível obter um agrupamento mais eficiente;
- Foi possível obter um ganho de aproximadamente 2,6x em relação aos agrupamentos apresentados na literatura;
- Utilizando somente a comunicação acumulada, não é explorado todo o problema de executar uma aplicação em uma plataforma heterogênea.

Conteúdo

- Introdução
- Framework Desenvolvido
- Resultados
- Conclusão
- **Trabalhos Futuros**

Trabalhos Futuros

- Realizar uma análise detalhada sobre a variação da banda do meio de comunicação;
- Desenvolver uma técnica de agrupamento que considere mais informações além da comunicação.

Agradecimento

- Obrigado a todos pela presença.

Rafael Ribeiro dos Santos

Referências

- I. Ashraf and V.M. Sima and K.L.M. Bertels. Intra-application data-communication characterization. In: Proc. 1st International Workshop on Communication Architectures at Extreme Scale. Frankfurt, Germany: [s.n.], 2015.
- ASHRAF, I.; OSTADZADEH, S. A.; MEEUWS, R.; BERTELS, K. Evaluation Methodology for data communication-aware application partitioning. In: SPRINGER. Euro-Par 2013: Parallel Processing Workshops. [S.l.], 2013. p. 739–748.
- HAYNES, W. Student's t-test. In: Encyclopedia of Systems Biology. [S.l.]: Springer, 2013. p. 2023–2025.
- WILCOXON, F. Individual comparisons by ranking methods. Biometrics bulletin, JSTOR, v. 1, n. 6, p. 80–83, 1945.