Session 1: Applications and Algorithms Using Hardware Accelerators

- Towards Seismic Wave Modeling on Heterogeneous Many-core Architectures using Task-based Runtime System.
  Victor Martinez, David Michea, Fabrice Dupros, Olivier Aumage, Samuel Thibault, Hideo Aochi and Philippe Navaux
  Abstract: Understanding three-dimensional seismic wave propagation in complex media is still one of the main challenges of quantitative seismology. Because of its simplicity and numerical efficiency, the finite-differences method is one of the standard techniques implemented to consider the elastodynamics equation. Additionally, this class of modeling heavily relies on parallel architectures in order to tackle large scale geometries including a detailed description of the physics. Last decade, significant efforts have been devoted towards efficient implementation of the finitedifferences methods on emerging architectures. These contributions have demonstrated their efficiency leading to robust industrial applications. The growing representation of heterogeneous architectures combining general purpose multicore platforms and accelerators leads to re-design current parallel application. In this paper, we consider StarPU task-based runtime system in order to harness the power of heterogeneous CPU+GPU computing nodes. We detail our implementation and compare the performance obtained with the classical CPU or GPU only versions. Preliminary results demonstrate significant speedups in comparison with the best implementation suitable for homogeneous cores.

- Optimized Parallel Label Propagation based Community Detection on the Intel®Xeon Phi ™Architecture.
  Andrei Khlopotine
  Abstract: Complex systems such as social, biological and information networks, characterized by millions to billions of sub-entities and relationships between them are best represented by graphs. A distinguishing feature of such complex systems is the self-organization into dense clusters called communities. Detecting such communities in massive graphs is critical to the understanding of such complex systems. However community detection is non-trivial and is expensive due to a vast number of computations needed to fully realize the underlying relationships. While a number of near-exact and heuristic algorithms is available for community detection, parallelizing such algorithms to fully leverage the advantages of parallel hardware is still a challenging problem. Most presently available approaches attempt to optimize run-time complexities by scaling original serial community detection algorithms. Graph algorithms in general and existing community detection algorithms in particular are known for not being commensurate with linear scalability on parallel systems. Therefore, there is a need for a combination of high-performance software and a hardware platform that would support efficient parallel

graph processing with respect to community detection. We present an Intel®Xeon Phi$^{\mathrm{TM}}$Label Propagation algorithm (PLPA) variant of community detection algorithm based on label propagation (LP). Our algorithm was tuned for the Intel®Xeon Phi$^{\mathrm{TM}}$platform a novel architecture that provides 50+ physical cores with simultaneous multithreading. We outline Phi$^{\mathrm{TM}}$architecture advantages and limitations for PLPA and massive graph processing in general. We present test results of running our algorithm on large real-world networks while achieving near linear speedups and improving the quality of the detected communities. We also analyze possibilities of processing massive networks that cannot fully fit in a Phi$^{\mathrm{TM}}$memory and hence we extend our initial solution to a modified PLPA (PLPA-M).

Session 2: Applications and Algorithms Using Hardware Accelerators

- GPU-accelerated High-speed Eye Pupil Tracking System.
  Juan Mompeán, Juan L. Aragón, Pablo Artal and Pedro Prieto
  Abstract: Pupil tracking under infrared illumination is an important tool for many researchers in physiological visual optics and ophthalmology. It is also a relevant topic for gaze tracking which is used in psychological and medical research, marketing, human-computer interaction, virtual reality and other areas. A typical setup can be either a low-cost webcam with some infrared LEDs or glasses with mounted cameras and infrared illumination. In this work, we evaluate and parallelize several pupil tracking algorithms with the aim of estimating the pupil's position and size with high accuracy in order to develop a high-speed pupil tracking system. To achieve high processing speed the original non-parallel algorithms have been parallelized by using CUDA and OpenMP. Graphics cards are designed to process images at very high frequencies and resolutions, and CUDA enables them to be used for general purpose computing. Our experimental results show that pupil tracking can be efficiently performed at high speeds with high-resolution images (up to 530 Hz with images of 1280x1024 pixels) using a state-of-the-art GP-GPU.

- Efficient Irregular Wavefront Propagation Algorithms on Intel Xeon Phi.
  Jeremias Moreira, George Teodoro, Alba Melo, Jun Kong, Tahsin Kurc and Joel Saltz
  Abstract: We investigate the execution of the Irregular Wavefront Propagation Pattern (IWPP), a fundamental computing structure used in several image analysis operations, on the Intel®Xeon Phi$^{\mathrm{TM}}$co-processor. An efficient implementation of IWPP on the Xeon Phi is a challenging problem because of IWPP's irregularity and the use of atomic instructions in the original IWPP algorithm to resolve race conditions. On the Xeon Phi, the use of SIMD and vectorization instructions is critical to attain high performance. However, SIMD atomic instructions are not supported.

Therefore, we propose a new IWPP algorithm that can take advantage of the supported SIMD instruction set. We also evaluate an alternate storage container (priority queue) to track active elements in the wavefront in an effort to improve the parallel algorithm efficiency. The new IWPP algorithm is evaluated with Morphological Reconstruction and Imfill operations as use cases. Our results show performance improvements of up to 5.63x on top of the original IWPP due to vectorization. Moreover, the new IWPP achieves speedups of 45.7x and 1.62x, respectively, as compared to efficient CPU and GPU implementations.

Session 3: New Architectures and Hardware Mechanisms to Improve Performance

- Performance and Energy Efficient Hardware-based Scheduler for Symmetric/Asymmetric CMPs.
  Nikola Markovic, Daniel Nemirovsky, Osman Unsal, Mateo Valero and Adrian Cristal
  Abstract: As thread level parallelism in applications has continued to expand, so has research in chip multi-core processors. Since more and more applications become multi-threaded we expect to find a growing number of threads executing on a machine. Consequently, the operating system will require increasingly larger amounts of CPU time to schedule these threads efficiently. Instead of perpetuating the trend of performing more complex thread scheduling in the operating system, we propose a hardware implementation of the Thread Lock Section-aware Scheduling (TLSS) scheduling mechanism. This lightweight mechanism helps to identify multi-threaded application bottlenecks such as thread synchronization sections and complements the Fairness-aware Scheduler method. It is, to our knowledge, the first hardware based lock section-aware scheduling that is energy attentive and can be applied to both asymmetric and symmetric CMPs. It achieves an average performance gains of 10.9 percent (geometric mean) compared to the state-of-the-art Linux OS Scheduler when applied on the Symmetrical Chip Multi-Processor (SCMP). At the same time, it is 81 percent more EDP (energy-delay product) efficient when applied on an Asymmetrical Chip Multi-Processor (ACMP) and compared to the Linux OS Scheduler on an SCMP, where ACMP and SCMP take relatively the same chip area.

- Analysis and Optimization of Engines for Dynamically Typed Languages.
  Gem Dot, Alejandro Martínez and Antonio González
  Abstract: Dynamically typed programming languages have become very popular in the recent years. These languages ease the task of the programmer but introduce significant overheads since assumptions about the types of variables have to be constantly validated at run time. JavaScript is a widely used dynamically typed language that has gained significant popularity in recent years. In this paper, we provide a detailed analysis of

the two main sources of overhead in the JavaScript execution. The first one is the runtime overhead needed for dynamic compilation and housekeeping activities (i.e. garbage collector, compilation, etc.). The second one is the additional checks and guards introduced by the dynamic nature of JavaScript. Then, we propose three new HW/SW optimizations that reduce this latter type of overhead. We show that these two types of overhead represent 35% and 25% respectively of the total execution time on average for a representative workload, and the proposed optimizations provide a 6% average speedup.

- Memory Centric Computation (mc2) for Large-scale Graph Processin
  Kattamuri Ekanadham and Guojing Cong
  Abstract: Large-scale graph processing is an increasingly important workload in modern systems. Conventional systems are usually optimized for locality of memory references, using caches and parallelization techniques to cover long memory latencies. However since graphs are distributed over memory in unpredictable manner, their processing does not exhibit great locality. While graph algorithms have plenty of parallelism, they are not easily amenable for effective vectorization, as the memory references are scattered all over. What is needed is a paradigm to specify a number of parallel tasks, each of which performs a short computation near the memory and a mechanism to efficiently execute them. In this paper, we propose a novel computational model that is memory-centric: the computation is organized as a collection of functions, each of which operates on a specific piece of data and is executed close to the memory where it resides. Basic primitives are provided to orchestrate the flow, synchronization and execution of the functions at their respective data points to accomplish a global task. We propose a scalable architecture to execute this computational model. We simulate an implementation of this architecture to compare the performance of running some graph algorithms on it with observed performance when the same algorithms were run on conventional systems. Preliminary results for a few graph algorithms show our approach is very promising in improving the performance of graph algorithms.

- Progressive Codesign of an Architecture and Compiler using a Proxy Application.
  Arpith Jacob, Tong Chen, Zehra Sura, Changhoan Kim, Carlo Bertolli, Samuel Antao, Kevin O'Brien and Ravi Nair
  Abstract: The Active Memory Cube (AMC) is a novel near memory processor that exploits high memory bandwidth and low latency close to DRAM to execute scientific applications in an energy-efficient manner. Its energy efficiency is derived from a combination of its novel scalar-vector data-flow path combined with its simple control-flow path that required the development of a sophisticated compiler, co-designed with the architecture. Such co-design is commonly done using hand-tuned codes for simple kernels that typically do not capture the nuances of real world applications or reveal the complexities of programming a heterogeneous

system. At the same time, an entire application is intractable to an early-stage compiler. In this work we describe a progressive, iterative methodology to the co-design of the compiler and architecture for the AMC using LULESH, a real-world hydrodynamics proxy application. We focus on a procedure that calculates the kinematic variables for domain elements. During the concept phase we looked at simpler kernels directly derived from the procedure, and progressively moved to the entire procedure as the compiler and simulation environment matured. We found this progression from the simpler extracted kernels to the entire procedure useful in gradually exposing new issues in the compiler and architecture. Directly applying optimizations developed for the simpler kernels resulted in poor performance on the proxy application, but after developing new compiler passes, the former optimizations could be applied profitably. Co-design on a proxy application revealed opportunities to refine the micro architecture that were not identified with simple micro benchmarks alone. Development of future accelerators and their programming environments can benefit from a similar iterative co-design through component kernels to entire proxy applications.

Session 4: Memory Systems and Optimizations

- Tidy Cache: Improving Data Placement in Die-stacked DRAM Caches.
  Adrià Armejach, Adrian Cristal and Osman S. Unsal
  Abstract: Die-stacked DRAM caches are likely to become available in mainstream chips in the near future. DRAM caches are typically used as a last level shared cache behind the traditional hierarchy of on-chip SRAM caches. However, its internal organization differs from traditional caches as it is based on DRAM technology that provides significantly diverse access latencies depending on the state of its internal structures. Accesses that hit in the row-buffer require only one DRAM command and are significantly faster than those that require closing the row-buffer to load a new row to read from. Prior work has focused on maximizing row-buffer locality while maintaining high cache hit ratios. However, past designs do not consider performance problems that may arise due to interleaved accesses from different applications that compete for the shared DRAM resources, nor the different access patterns and locality characteristics that each of these applications may have. In this paper, we first identify performance pathologies that are specific to DRAM caches which arise due to the interference caused by interleaved accesses from multiple cores. We then propose Tidy Cache, a novel DRAM cache design that is able to ameliorate these performance pathologies by dynamically adapting the replacement policy for demanded data. Our performance evaluation results show that our design outperforms the state-of-the-art by 9.2% for multi-programmed SPEC workloads and by 16.7% for a set of TPC-H queries, mainly due to significantly better cache miss ratios.

- Unifying Router Power Gating with Data Placement for Energy-Efficient NoC.
  Yuho Jin
  Abstract: Network-on-Chip (NoC) is a critical hardware supporting on-chip data movement in multicore and manycore processors. Data movement is predicted to become a major power-consuming operation compared to computation as the technology scales. This paper shows that unifying router power gating with data placement significantly reduces dynamic and static power needed for moving data in NoC for memory hierarchy. Region-based data placement enables to localize private data traffic and to concentrate shared data traffic in one region of NoC, which shapes traffic in a wellbehaved way and increases power gating opportunities. In this regard, a dimensionally power-gated router with a regionbased routing algorithm is proposed to reduce router static power and performance/energy overheads in power gating. Full-system evaluation using SPEComp benchmarks shows that the dynamic power gating management achieves NoC power savings by 46improves energy-efficiency by
  20

- i-MIRROR: A Software Managed Die-Stacked DRAM-Based Memory Subsystem.
  Jee Ho Ryoo, Karthik Ganesan, Yao-Min Chen and Lizy John

  Abstract: This paper presents an operating system managed die-stacked DRAM called i-MIRROR that mirrors high locality pages from off-chip DRAM. Optimizing the problems of reducing cache tag area, reducing transfer bandwidth and improving hit latency altogether while using die-stacked DRAM as hardware cache is extremely challenging. In this paper, we show that performance and energy efficiency can be obtained by software management of die-stacked DRAM, which eliminates the need for tags, the source of aforementioned problems.

  In the proposed scheme, the operating system loads pages from disks to die-stacked DRAM on a page fault at the same time as they are loaded to off-chip DRAM. Our scheme maintains the pages in off-chip and die-stacked DRAM in a synchronized/mirrored state by exploiting the parallel loading capability to die-stacked and off-chip DRAM from the disk. This eliminates the need for physical page movement to the slower off-chip DRAM upon eviction from die-stacked DRAM. Requests for pages that got evicted from die-stacked DRAM are simply serviced by the slower off-chip DRAM to prevent frequent data movements of large pages and thrashing between conflicting pages. The operating system periodically monitors the usage of the pages in off-chip DRAM and promotes high locality pages to die-stacked DRAM. Our evaluations show that the proposed hardware-assisted software-managed i-MIRROR scheme achieves an IPC improvement of 13% while consuming 6% less energy than prior state-of-the-art die-stacked caching schemes and 79% improvement in terms of

IPC and 72% in terms of energy savings over systems without die-stacked DRAM support.

Session 5: Code Optimization

- Fusion of calling sites.
  Douglas Teixeira, Sylvain Collange and Fernando Pereira.
  Abstract: The increasing popularity of Graphics Processing Units (GPUs), has brought renewed attention to old problems related to the Single Instruction, Multiple Data execution model. One of these problems is the reconvergence of divergent threads. A divergence happens at a conditional branch when different threads disagree on the path to follow upon reaching this split point. Divergences may impose a heavy burden on the performance of parallel programs. In this paper we propose a compiler level optimization to mitigate this performance loss. This optimization consists in merging function call sites located at different paths that sprout from the same branch. We show that our optimization adds negligible overhead on the compiler. It does not slowdown programs in which it is not applicable, and accelerates substantially those in which it is. As an example, we have been able to speed up the well known SPLASH Fast Fourier Transform benchmark by 11%.

- OpenCL Kernel Fusion for GPU, Xeon Phi and CPU.
  Jiří Filipovič and Siegfried Benkner
  Abstract: Kernel fusion is an optimization method, in which the code from several kernels is composed to create a new, fused kernel. It can push the performance of kernels beyond limits given for their isolated, unfused form. In this paper, we introduce a classification of different types of kernel fusion for both data dependent and data independent kernels. We study kernel fusion on three types of OpenCL devices: GPU, Xeon Phi and CPU. Those hardware platforms have quite different properties, thus, kernel fusion often affects performance in quite different ways. We analyze the impact of kernel fusion on those hardware platforms and show how it can be used to improve performance. Based on our study we also introduce a basic transformation method for generating fused kernels, which has good potential to be automatized.

Session 6: System Characterization and Performance Evaluation

- WattWatcher: Fine-Grained Power Estimation For Emerging Workloads.
  Michael Lebeane, Jee Ho Ryoo, Reena Panda and Lizy John
  Abstract: Extensive research has focused on estimating power to guide advances in power management schemes, thermal hot spots, and voltage noise. However, simulated power models are slow and struggle with deep software stacks, while direct measurements are typically coarse-grained.

This paper introduces WattWatcher, a multicore power measurement framework that offers fine-grained functional unit breakdowns. WattWatcher operates by passing event counts and a hardware descriptor file into configurable back-end power models based on McPAT. Researchers and vendors can add other processors to our tool by mapping to the WattWatcher interface. We show that WattWatcher, when calibrated, has a MAPE (mean absolute percentage error) of 2.67% aggregated over all benchmarks when compared to measured power consumption on SPEC CPU 2006 and multithreaded PARSEC benchmarks across three different machines of various form factors and manufacturing processes. We present two use cases showing how WattWatcher can derive insights that are difficult to obtain through other measurement infrastructures. Additionally, we illustrate how WattWatcher can be used to provide insights into challenging big data and cloud workloads on a server CPU. Through the use ofWattWatcher, it is possible to obtain a detailed power breakdown on real hardware without vendor proprietary models or hardware instrumentation.

- Performance Characterization of Modern Databases on Out-of-order CPUs. *Reena Panda, Christopher Erb, Michael Lebeane, Jeeho Ryoo and Lizy Kurian John*
  Abstract: Big data revolution has created an unprecedented demand for intelligent data management solutions on a large scale. While data management has traditionally been used as a synonym for relational data processing, in recent years a new group popularly known as NoSQL databases have emerged as a competitive alternative. There is a pressing need to gain greater understanding of the characteristics of modern databases to architect targeted computers. In this paper, we investigate four popular NoSQL/SQL-style databases and evaluate their hardware performance on modern computer systems. Based on data collected from real hardware, we evaluate how efficiently modern databases utilize the underlying systems and make several recommendations to improve their performance efficiency. We observe that performance of modern databases is severely limited by poor cache/memory performance. Nonetheless, we demonstrate that dynamic execution techniques are still effective in hiding a significant fraction of the stalls, thereby improving performance. We further show that NoSQL databases suffer from greater performance inefficiencies than their SQL counterparts. SQL databases outperform NoSQL databases for most operations and are beaten by NoSQL databases only in a few cases. NoSQL databases provide a promising competitive alternative to SQL-style databases, however, they are yet to be optimized to fully reach the performance of contemporary SQL systems. We also show that significant diversity exists among different database implementations and big-data benchmark designers can leverage our analysis to incorporate representative workloads to encapsulate the full spectrum of data-serving applications. In this paper, we also compare data-serving applications with other popular benchmarks such as SPEC CPU2006 and

SPECjbb2005.

- Cloud Services Evaluation through QoE: A Methodological Approach.
Frederico Guilherme Irigoyen Da Costa, Maria Cristina Felippetto de Castro, Candice Muller and Fernando C. C. De Castro
Abstract: Cloud computing has been touted as a revolutionary concept in computing in the Information Age, since it enhances the quality of communication and it is highly cost effective. Cloud computing market has attracted the interest of several providers and corporations, creating an environment in which the userÂ´s Quality of Experience (QoE) becomes a competitive advantage. Cloud services are often available as Web applications, since Web browsers may provide a more user friendly interface. Thus, Web Application Response plays a critical role in the perceptions of cloud service users. This article proposes a methodology to evaluate the user Quality of Experience of cloud services, with focus on web applications, using the MOS score in a user-centered approach. This methodology estimates the QoE from the end-to-end response time and adjusts the estimated score according to the evaluation context, thought maximum session time. Estimation of QoE is a differentiating factor in choosing cloud service providers and defining the form of implementing cloud applications (e.g., through programming language, page type or application server). The proposed methodology has been applied to three cloud service servers, located in Brazil, Europe and USA and several case studies in business contexts have been evaluated, comparing different clients and server applications in a monitored environment. The results point to the crucial role that the evaluation period plays in the comparison of solutions.

- Non-stationary Simulation of Computer Systems and Dynamic Performance Evaluation: a Concern-based Approach and Case Study on Cloud Computing.
Lourenço Alves Pereira Júnior, Edwin Luis Choquehuanca Mamani, Marcos José Santana, Regina Helena Carlucci Santana, Pedro Northon Nobile and Francisco José Monaco
Abstract: This paper introduces an approach to the design of discrete event simulation experiments aimed at transient performance analysis. Specially in complex, multi-tier applications, the net effects of small delays introduced by buffers, IO operations, communication latency and averaged measurements, may result in significant inertia along the input-output path. In order to bring out these dynamic properties, the simulation experiment should excite the system with non-stationary workload under controlled conditions. The work discusses on the dynamic properties of large-scale distributed computer systems and how these may impact delivered performance. These rationales are explored to motivate a concern-based architecture which captures the elicited requirements. The design approach is systematic formulated and illustrated by a case study on extending a well-known cloud computing simulation framework to meet the aimed features. Experimental results of ongoing work are also addressed.

- Serialization Management for Best-Effort Hardware Transactional Memory: A key for performance.
  Matthew Gaudet, Jose Nelson Amaral and Guido Araujo
  Abstract: Most studies of Best-Effort HTM (BE-HTM) performance use a single serialization manager and a single parameter value across all benchmarks, inputs and thread counts. The experimental study in this paper indicates that the values chosen for serialization-manager parameters have a significant effect on performance in the Blue Gene/Q's (BG/Q) BE-HTM system. Moreover, for a given serialization manager, different benchmarks typically require different parameter values to achieve the best performance. BG/Q features two TM settings that represent two different HTM designs. A study of these two settings indicate that serialization-management decisions are also sensitive to changes in the HTM design. Therefore the choice of serialization management, including the tuning parameters, should be reevaluated for each new platform because effectiveness is affected even by relatively small changes to the HTM design.

Session 7: Fault Tolerance and Cloud Storage

- Exploring Energy-Consistency Trade-off in Cassandra Cloud Storage System.
  Houssem-Eddine Chihoub, Shadi Ibrahim, Yue Li, Gabriel Antoniu, Maria S. Perez and Luc Bougé
  Abstract: Apache Cassandra is an open-source cloud storage system that offers multiple types of operation-level consistency including eventual consistency with multiple levels of guarantees and strong consistency. It is being used by many datacenter applications (e.g., Facebook and AppScale). Most existing research efforts have been dedicated to exploring trade-offs such as: consistency vs. performance, consistency vs. latency and consistency vs. monetary cost. In contrast, a little work is focusing on the consistency vs. energy trade-off. As power bills have become a substantial part of the monetary cost for operating a data-center, this paper aims to provide a clearer understanding of the interplay between consistency and energy consumption. Accordingly, a series of experiments have been conducted to explore the implication of different factors on the energy consumption in Cassandra. Our experiments have revealed a noticeable variation in the energy consumption depending on the consistency level. Furthermore, for a given consistency level, the energy consumption of Cassandra varies with the access pattern and the load exhibited by the application. This further analysis indicates that the uneven distribution of the load amongst different nodes also impacts the energy consumption in Cassandra. Finally, we experimentally compare the impact of four storage configuration and data partitioning policies on the energy consumption in Cassandra: interestingly, we achieve 23% energy saving when assigning 50% of the nodes to the hot pool for the applications with moderate ratio of reads and writes, while applying eventual (quorum) consistency. This

study points to opportunities for future research on consistency energy trade-offs and offers useful insight into designing energy efficient techniques for cloud storage systems.

- COMET: Client-Oriented Metadata Service for Highly Available Distributed File Systems.
  Ruini Xue, Lixiang Ao and Zhongyang Guan
  Abstract: Highly available metadata services of distributed file systems are essential to cloud applications. However, existing highly available metadata designs lack client-oriented features that treat metadata discriminately, leading to a single metadata fault domain and low availability. After investigating the workload characteristics of Hadoop, we propose Client-Oriented METadata (COMET), a novel highly available metadata service design that divides and distributes metadata into independent regions in terms of clients. These regions are isolated fault domains inherently, and failures in one region will not break file operations in other regions. A prototype of COMET was implemented based on HDFS, and the experimental results show that COMET can significantly improve metadata availability of HDFS without obvious performance degradation. It can also deliver scalable performance and faster metadata recovery due to its decentralized architecture.

- A Fault-Tolerance Protocol for Parallel Applications with Communication Imbalance.
  Esteban Meneses and Laxmikant Kale
  Abstract: The predicted failure rates of future supercomputers loom the groundbreaking research large machines are expected to foster. Therefore, resilient extreme-scale applications are an absolute necessity to effectively use the new generation of supercomputers. Rollback-recovery techniques have been traditionally used in HPC to provide resilience. Among those techniques, message logging provides the appealing features of saving energy, accelerating recovery, and having low performance penalty. Its increased memory consumption is, however, an important downside. This paper introduces memory-constrained message logging (MCML), a general framework for decreasing the memory footprint of message-logging protocols. In particular, we demonstrate the effectiveness of MCML in maintaining message logging feasible for applications with substantial communication imbalance. This type of applications appear in many scientific fields. We present experimental results with several parallel codes running on up to 4,096 cores. Using those results and an analytical model, we predict MCML can reduce execution time up to 25% and energy consumption up to 15%, at extreme scale.

Session 8: Scheduling and Virtual Machines

- Comparison of Static and Runtime Resource Allocation Strategies for Matrix Multiplication.

Olivier Beaumont, Lionel Eyraud-Dubois, Abdou Guermouche and Thomas Lambert
Abstract: The tremendous increase in the size and heterogeneity of supercomputers makes it very difficult to predict the performance of a scheduling algorithm. In this context, relying on purely static scheduling and resource allocation strategies, that make scheduling and allocation decisions based on the dependency graph and the platform description, is expected to lead to large and unpredictable makespans whenever the behavior of the platform does not match the predictions. For this reason, the common practice in most runtime libraries is to rely on purely dynamic scheduling strategies, that make short-sighted scheduling decisions at runtime based on the estimations of the duration of the different tasks on the different available resources and on the state of the machine. In this paper, we consider the special case of Matrix Multiplication, for which a number of static allocation algorithms to minimize the amount of communications have been proposed. Through a set of extensive simulations, we analyze the behavior of static, dynamic, and hybrid strategies, and we assess the possible benefits of introducing more static knowledge and allocation decisions in runtime libraries.

- Device-Sensitive Framework for Handling Heterogeneous Asymmetric Clusters Efficiently.
  Valon Raca and Eduard Mehofer
  Abstract: Heterogeneous systems with different types of compute devices are common nowadays in the field of High Performance Computing (HPC). This heterogeneity is not limited to compute devices, but also includes cluster nodes with different hardware configurations leading to asymmetric cluster architectures. In such a hierarchical system OpenCL is not sufficient any more. Support is required to distribute the work efficiently onto the non-identical cluster nodes. Different behavior of the individual compute devices with respect to execution time and energy consumption has to be taken into account to meet the demands of the user. Our framework provides a transparent view on the different compute devices alleviating the programmer to deal with the hardware architecture and device execution behavior explicitly. Besides efficiency considerations, the device sensitive feature includes in addition handling of device failures and appropriate recovery actions. Experiments show that our framework succeeds in distributing the work onto compute devices efficiently.

Session 9: Scheduling and Virtual Machines

- Evaluating the Impact of Memory Allocation and Swap for Vertical Memory Elasticity in VMs.
  Roberto Sawamura, Cristina Boeres and Vinod Rebello
  Abstract: Typically, virtual machine (VM) allocation is based on the host server's ability to meet the VM's maximum CPU, I/O and memory

requirements. However, given that the requirements of applications within the VM may vary during execution, it might be more efficient to also vary over time the amount of resources dedicated to the VM. In cloud systems, vertical elasticity is the dynamic adjustment of the amount of a physical resource, such as memory, CPU cores, etc., that is allocated to a VM. With technology pushing up core counts and speeds of modern servers, and given the growing trend towards server consolidation, making the most of the available memory is crucial for good application performance. This paper investigates the impact of memory allocation and swap usage on VM performance. Through an experimental evaluation, hypervisor independent metrics and policies are identified for consideration by tools that claim to offer vertical memory elasticity. Based on the conclusions, the paper goes on to present a framework of a tool to dynamically manage memory allocations of VMs. Preliminary results with the proposed vertical Memory Elasticity Controller, MEC, highlight some of the benefits to both resource providers and applications through improved efficiency, throughput and performance. Ongoing work will continue to expand on the current evaluation and refine the scheduling policies to further improve the tool.

- Quantum Virtual Machine: a Scalable Model to Optimize Energy Savings and Resource Management.
  Andre Felipe Monteiro and Orlando Loques
  Abstract: This work presents a model for managing Virtual Application Servers (VAS) on cluster-based web servers. In addition to providing energy savings, our model has linear scalability and defines a default processing virtual web server, known as Quantum Virtual Machine (QVM). A set of QVM performs a Logical Web Server (LWS), which operates in a flexible manner, changing its performance and power consumption depending on the workload of the applications. Concepts of agile VAS clone, co-allocation of VAS in the same core, and Dynamic Voltage and Frequency Scaling (DVFS) are used in the model, enabling rapid configuration actions and a fine-grained QoS control. The experiments evaluate the effectiveness of the proposed model by means of power consumption reduction and QoS violation as compared to the Linux CPU governors and state-of-the-art approaches based on optimization. The results show our model conserves up to 51.8% of the energy required by a cluster designed for peak workload scenario, with a negligible impact on the applications performance.

- A Programming Interface for Overload Control in Staged Event-based Architectures.
  Breno Cruz, Noemi Rodriguez and Ana Lúcia Moura
  Abstract: Overload control requires different scheduling policies in different application scenarios. In this work, we propose an API that allows the programmer to choose and program scheduling policies in the specific setting of a staged event driven architecture. Through the study of

various scheduling policies, we have extracted common requirements and built an interface that allows developers to build and combine scheduling controllers. The implementation is specific to the Leda event driven architecture, but the same model could be applied in other systems with similar needs. Using this interface, we have built controllers for a set of scheduling policies, and we report the results of experiments with these policies in two Leda applications with different load profiles.