

2010

- 1) Evolução das velocidades de processamento, de acesso a memória e ao disco e das interfaces de rede - Um apanhado histórico até os dias de hoje
- 2) Green Computing - Qual a perspectiva em organização de computadores ?
- 3) "The 1000 core microprocessor: Will we be ready for it?", Yale Patt (University of Texas at Austin)
SBAC 2009
- 4) "GPUs in High-Performance Computing architectures, software stack, education, and applications", Wen-Mei Hwu (University of Illinois at Urbana-Champaign)
SBAC 2009
- 5) Tilera pushes to 100 cores with mesh processor
http://www.theregister.co.uk/2009/10/26/tilera_third_gen_mesh_chips/
- 6) The new era in genomics: Opportunities and challenges for high performance computing
Srinivas Aluru, Iowa State University
IPDPS 2010
- 7) "Computing at the Crossroads". Dan Reed
Scalable and Multicore Computing Strategist, Microsoft
HIPC2009
- 8) "The End of Denial Architecture and the Rise of Throughput Computing "
Bill Dally - Chief Scientist and VP of Research, NVIDIA
Bell Professor of Engineering, Stanford University
HIPC2009
- 9) "Bringing Supercomputing to the Masses"
Justin R. Rattner
Senior Fellow and Vice President
Intel Chief Technology Officer
HIPC2009
- 10) Virtualização - Uma visão geral
- 11) "Impact of Architecture and Technology for Extreme Scale on Software and Algorithm Design"
Prof. Jack Dongarra - University of Tennessee, Oak Ridge National Laboratory, University of Manchester
EUROPAR 2010
- 12) "Computational Epidemiology: a New Paradigm in the Fight against Infectious Diseases"
Dr. Vittoria Colizza - ISI Foundation
EUROPAR 2010
- 13) "Innovation in Cloud Computing Architectures"
Prof. Ignacio M. Llorente - Universidad Complutense de Madrid
EUROPAR 2010
- 14) Exaflop/s, Seriously!

Prof. David Keyes
Dean, Mathematical and Computer Science & Engineering, KAUST
Fu Foundation Professor of Applied Mathematics, Columbia Univ.
ISPDC 2010

15) When Core Multiplicity Doesn't Add Up
Assistant Prof. Nikos Hardavellas
Department of Electrical Engineering and Computer Science
McCormick School of Engineering and Applied Sciences
Northwestern University
ISPDC 2010

Mais ideias:
<http://techresearch.intel.com/ProjectHome.aspx>
<http://www.research.ibm.com/cell/>

2008

Estão disponíveis os seguintes temas para apresentação de trabalhos:

1) Finding speedup in parallel processors - Michael J. Flynn
Keynote speaker do ISPDC 2008
Abstract: The emphasis on multi core architectures and multi node parallel processors comes about, in part, from the failure of frequency scaling not because of breakthroughs in parallel programming or architecture.

Progress in automatic compilation of serial programs into multi tasked ones has been slow. The standard approach to programming HPC is to implement an application on as many multi core processors as possible, up to a point of memory saturation; after which partitioning continues over multiple such nodes. Now the inter node communications reduces the computational efficiency and scales up cost, power, cooling requirements and reliability concerns.

We'll consider an alternative model which stresses maximizing the node speedup as far as possible before considering multi node partitioning. Node speedup starts with the use of an accelerator (FPGA based, so far) adjunct to the computational node and then uses a cylindrical rather than layered programming model to insure application speedup.

2) Holistic Design of Multicore Architectures - Dean Tullsen
Keynote speaker do ISPDC 2008
Abstract: In recent years, the processor industry has moved from a uniprocessor focus to increasing numbers of cores on chip. But we cannot view those cores in the same way we did when we lived in a uniprocessor world. Previously, we expected each core to provide good performance on virtually any application, with energy efficiency, and without error. But now the level of interface with the user and the system is the entire multicore chip, and those requirements need only be met at the chip level - no single core need meet them. This provides the opportunity to think about processor architecture in whole new ways.

3) Novel distributed processing paradigms: computing with condensed graphs

- John Morrison

Keynote speaker do ISPDC 2008

Abstract: Condensed Graphs provide a simple expression of complex dependencies in a program task graph or a work flow. In these graphs, nodes represent tasks and edges represent the sequencing constraints associated with those tasks. The sequence of task execution can be altered by altering the relationship between various nodes. These simple topological changes do not, in general, alter the meaning of the task graph or work flow (although they can affect program termination). Rather, they result in a change in execution order, reflecting either an imperative, data-driven or demand-driven computation. In fact, any desired combination of all three paradigms can be represented within the same task graph or work flow. This flexibility leads to many advantages both in the expression task graphs and in their implementation.

This talk will introduce the concept of Condensed Graphs and discuss various implementation platforms already developed for their execution. In particular, an overview of the WebCom Abstract Machine will be presented. A "Grid Enabled" version of this system, known as WebCom-G is currently being developed as a candidate operating system for Grid-Ireland. The mission of this project is to hide the complexities of computational platform from computational scientists - thus allowing them to concentrate on expressions of solutions to problems rather than on the implementation of those solutions. The status of this project will also be reported.

4) Reinventing Computing - Burton Smith

Keynote speaker IPDPS 2007

Abstract:

The many-core inflection point presents a new challenge for our industry, namely general-purpose parallel computing. Unless this challenge is met, the continued growth and importance of computing itself and of the businesses engaged in it are at risk. We must make parallel programming easier and more generally applicable than it is now, and build hardware and software that will execute arbitrary parallel programs on whatever scale of system the user has. The changes needed to accomplish this are significant and affect computer architecture, the entire software development tool chain, and the army of application developers that will rely on those tools to develop parallel applications. This talk will point out a few of the hard problems that face us and some prospects for addressing them.

5) Avoiding the Memory Bottleneck through Structured Arrays - Michael J. Flynn

Keynote speaker IPDPS 2007

Abstract:

Basic to parallel program speedup is dealing with memory bandwidth requirements. One solution is an architectural arrangement to stream data across multiple processing elements before storing the result in memory. This MISD type of configuration provides multiple operations per data item fetched from memory. One realization of this streamed approach uses FPGAs. We'll discuss both the general memory problem and

some results based on work at Maxeler using FPGAs for acceleration.

6) Quantum Physics and the Nature of Computation - Professor Umesh Vazirani

Keynote speaker IPDPS 2007

Abstract:

Quantum physics is a fascinating area from a computational viewpoint. The features that make quantum systems prohibitively hard to simulate classically are precisely the aspects exploited by quantum computation to obtain exponential speedups over classical computers. In this talk I will survey our current understanding of the power (and limits) of quantum computers, and prospects for experimentally realizing them in the near future. I will also touch upon insights from quantum computation that have resulted in new classical algorithms for efficient simulation of certain important quantum systems.

7) Programming Models for Petascale to Exascale - Katherine Yelick

keynote speaker IPDPS 2008

Abstract:

Multiple petascale systems will soon be available to the computational science community and will represent a variety of architectural models. These high-end systems, like all computing platforms, will have an increasing reliance on software-managed on-chip parallelism. These architectural trends bring into question the message-passing programming model that has dominated high-end programming for the past decade. In this talk I will describe some of the technology challenges that will drive the design of future systems and their implications for software tools, algorithm design, and application programming. In particular, I will show a need to consider models other than message passing as we move towards massive on-chip parallelism. I will talk about a class of partitioned global address space (PGAS) languages, which are an alternative to both message passing models like MPI and shared memory models like OpenMP. PGAS languages offer the possibility of a programming model that will work well across a wide range of shared memory, distributed memory, and hybrid platforms. Some of these languages, including UPC, CAF and Titanium, are based on a static model of parallelism, which gives programmers direct control over the underlying processor resources. The restricted nature of the static parallelism model in these languages has advantages in terms of implementation simplicity, analyzability, and performance transparency, but some applications demand a more dynamic execution model, similar to that of Charm++ or the recently developed HPCS languages (X10, Chapel, and Fortress). I will describe some of our experience working with both static and dynamically managed applications and some of the research challenges that I believe will be critical in developing viable programming techniques for future systems.

8) High Performance Computing in the Multi-core Area - Arndt Bode

Keynote speaker ISPD 2007

abstract:

Multi-core technologies, application specific accelerators and fault tolerance requirements are defining a new hardware basis for high performance computing systems. Multi-core will make evolve parallelism from a niche product in HPC to the standard programming model and, therefore, trigger new developments in languages, environments and tools. Accelerators and dynamic system behaviour from fault tolerance

will make virtualization techniques necessary to support programmability. Energy efficiency as a new optimization target for HPCN systems will force future systems to bring all of these techniques together.

9) The Excitement in Parallel Computing - Laxmikant (Sanjay) Kale
Keynote speaker HIPC 2008

Abstract The almost simultaneous emergence of multicore chips and petascale computers presents multidimensional challenges and opportunities for parallel programming. Machines with hundreds of TeraFLOP/S exist now, with at least one having crossed the 1 PetaFLOP/s rubicon. Many machines have over 100,000 processors. The largest planned machine by NSF will be at University of Illinois at Urbana-Champaign by early 2011. At the same time, there are already hundreds of supercomputers with over 1,000 processors each. Adding breadth, multicore processors are starting to get into most desktop computers, and this trend is expected to continue. This era of parallel computing will have a significant impact on the society. Science and engineering will make breakthroughs based on computational modeling, while the broader desktop use has the potential to directly enhance individual productivity and quality of life for everyone. I will review the current state in parallel computing, and then discuss some of the challenges. In particular, I will focus on questions such as: What kind of programming models will prevail? What are some of the required and desired characteristics of such model/s? My answers are based, in part, on my experience with several applications ranging from quantum chemistry, biomolecular simulations, simulation of solid propellant rockets, and computational astronomy.

10) The future is parallel but it may not be easy - Michael J. Flynn
Keynote speaker HIPC 2007

Abstract
Processor performance scaling by improving clock frequency has now hit power limits. The new emphasis on multi core architectures comes about from the failure of frequency scaling not because of breakthroughs in parallel programming or architecture. Progress in automatic compilation of serial programs into multi tasked ones has been slow. A look at parallel projects of the past illustrates problems in performance and programmability. Solving these problems requires both an understanding of underlying issues such as parallelizing control structures and dealing with the memory bottleneck. For many applications performance comes at the price of programmability and reliability comes at the price of performance.

11) Petaflop/s, Seriously - David Keyes
Keynote speaker HIPC 2007

Abstract:
Sustained floating-point rates on real applications, as tracked by the Gordon Bell Prize, have increased by over five orders of magnitude from 1988, when 1 Gigaflop/s was reported on a structural simulation, to 2006, when 200 Teraflop/s were reported on a molecular dynamics simulation. Various versions of Moore's Law over the same interval provide only two to three orders of magnitude of improvement for an individual processor; the remaining factor comes from concurrency, which is of order 100,000 for the BlueGene/L computer, the platform of choice for the majority of recent Bell Prize finalists. As the semiconductor industry begins to slip relative to its own roadmap for

silicon-based logic and memory, concurrency will play an increasing role in attaining the next order of magnitude, to arrive at the long-awaited milestone of 1 Petaflop/s sustained on a practical application, which should occur around 2009. Simulations based on Eulerian formulations of partial differential equations can be among the first applications to take advantage of petascale capabilities, but not the way most are presently being pursued. Only weak scaling can get around the fundamental limitation expressed in Amdahl's Law and only optimal implicit formulations can get around another limitation on scaling that is an immediate consequence of Courant-Friedrichs-Lewy stability theory under weak scaling of a PDE. Many PDE-based applications and other lattice-based applications with petascale roadmaps, such as quantum chromodynamics, will likely be forced to adopt optimal implicit solvers. However, even this narrow path to petascale simulation is made treacherous by the imperative of dynamic adaptivity, which drives us to consider algorithms and queueing policies that are less synchronous than those in common use today. Drawing on the ScaLES report (2003-04), the latest ITRS roadmap, some back-of-the-envelope estimates, and numerical experiences with PDE-based codes on recently available platforms, we will attempt to project the pathway to Petaflop/s for representative applications.

12) The Transformation Hierarchy in the Era of Multi-Core - Yale Patt
Keynote speaker HIPC 2007

Abstract The transformation hierarchy is the name I have given to the mechanism that converts problems stated in natural language (English, Spanish, Hindi, Japanese, etc.) to the electronic circuits of the computer that actually does the work of producing a solution. The problem is first transformed from a natural language description into an algorithm, and then to a program in some mechanical language, then compiled to the ISA of the particular processor, which is implemented in a microarchitecture, built out of circuits. At each step of the transformation hierarchy, there are choices. These choices enable one to optimize the process to accommodate some optimization criterion. Usually, that criterion is microprocessor performance. Up to now, optimizations have been done mostly within each of the layers, with artificial barriers in place between the layers. It has not been the case (with a few exceptions) that knowledge at one layer has been leveraged to impact optimization of other layers. I submit, that with the current growth rate of semiconductor technology, this luxury of operating within a transformation layer will no longer be the common case. This growth rate (now more than a billion transistors on a chip is possible) has ushered in the era of the chip multiprocessor. That is, we are entering Phase II of Microprocessor Performance Improvement, where improvements will come from breaking the barriers that separate the transformation layers. In this talk, I will suggest some of the ways in which this will be done.

13) Elastic Parallel Architectures - Prof. Antonio González
Keynote Europar 2008

Abstract

Multicore processors are more power-area-effective and more reliable than single-core processors. Because of that, they have become mainstream in all market segments, from high-end servers to desktop and mobile PCs, and industry's roadmap is heading towards an

increasing degree of threading in all segments. However, single-thread performance still matters a lot, and will continue to be a very important differentiating factor of future highly-threaded processors. Some workloads are tough to parallelize, and Amdahl's law points out the importance of improving performance in all sections of a given application, including parts that have little thread-level parallelism. Given the general-purpose nature of processors, they are expected to provide good performance for all types of workloads, despite of their very different characteristics in terms of parallelism. Many users nowadays are willing to have processors with more thread-level capabilities, but at the same time, the majority of them are also willing to have high performance for lightly threaded applications. The ideal solution is to have a processor where resources can dynamically be devoted to exploit either thread-level or instruction-level parallelism, trying to find the best tradeoff between both of them depending on the particular code that is being running. This approach is what we call an Elastic Parallel Architecture. How to build an effective elastic parallel architecture is an open research question. This talk will discuss the benefits of this type of architecture, and will describe several approaches that are being investigated for implementing it, highlighting the main strengths and weaknesses of each of them.

14) Fault Tolerance for PetaScale Systems: Current Knowledge, Challenges and Opportunities - Prof. Franck Cappello

Abstract:

The emergence of PetaScale systems reinvigorates the community interest about how to manage failures in such systems and ensure that large applications successfully complete. This talk starts by addressing the question of failure rate and trend in large systems, like the ones we find at the top of the top500. Where the failures come from and why we should pay more attention to them than in the past? A review of existing techniques for fault tolerance will be presented: rollback-recovery, failure prediction and proactive migration. We observe that rollback recovery has been deeply studied in the past years, resulting in a lot of optimizations; but is this enough to solve the challenge of fault tolerance raised by Petascale systems? What is the actual knowledge about failure prediction? Could we use it for proactive process migration and if so what benefit could we expect? Unfortunately, despite their high degree of optimization, existing approaches do not fit well with the challenging evolutions of large-scale systems. Thus, through this review of existing solutions and the presentation of the latest research results, we will list a set of open issues. Most of existing key mechanisms for fault tolerance come from the distributed system theory and the Chandy-Lamport algorithm for the determination of consistent global states. We should probably continue to optimize them, like by adding hardware dedicated to fault tolerance. Beside, there is room and even a need for new approaches. Opportunities may come from different origins, such as: 1) other fault tolerance approaches that consider failures as normal events in the system and 2) new algorithmic approaches, inherently fault tolerant. We will sketch some of these opportunities and their associated limitations.

15) 15mm x 15 mm: the new frontier of parallel computing - André Seznec

Apresentação: http://europar2007.irisa.fr/Files/15*15_Europar2007.ppt

16) Democratizing Parallel Software Development - Kunle Olukotun
Keynote SBAC 2007

Abstract: Now that we are firmly entrenched in the multicore era, to increase software functionality without decreasing performance many application developers will have to become parallel programmers. Today, parallel programming is so difficult the that it is only practiced by a few elite programmers. Thus, a key research question is what set of hardware and software technologies will make parallel computation accessible to average programmers. In this talk I will describe a set of architecture and programming language techniques that have the potential to dramatically simplify the task of writing a parallel program.