IPDPS 2006 Technical Program

April 25-29, 2006 – Rhodes Island, Greece

Tuesday, April 25, 2006

Workshops 1-11

Wednesday, April 26, 2006

Keynote Speaker
Manish Gupta, IBM T.J. Watson Research Center
Massively Parallel Systems: Ready or Not, Here They Come

Rising power dissipation in microprocessor chips is driving computer architects towards a variety of solutions, all of which require exploiting greater degrees of parallelism. Hence, even though Moore's Law is alive, relying primarily on frequency scaling is no longer a viable path for meeting the growing computational needs of applications. There are several hurdles to exploiting greater levels of parallelism, such as, programming complexity, communication bottlenecks, interference from operating system services, and system management costs. We describe our experiences with the IBM Blue Gene project on pushing the limits of scalability in all aspects of system design. We will present our successes as well as outstanding challenges in programming and managing massively parallel systems. We will argue that we need another revolution in software to help achieve scientific breakthroughs and truly deliver on the promise offered by the next generation of high performance computing systems.

Plenary Session: Best Papers

Session 1: Scheduling
Session 2: P2P and Grid Computing, 1
Session 3: Memory Systems and Caches
Session 4: Consistency In Grids

Session 5: Hashing
Session 6: Parallel and Distributed Algorithms
Session 7: P2P and Grid Computing, 2
Session 8: Processor Designs

Thursday, April 27, 2006

Keynote Speaker
Yves Robert, LIP Laboratory - CNRS/ENS Lyon
Static Scheduling for Large-Scale Platforms: Can One Hope for Efficiency?

We discuss the potential and limitations of static scheduling techniques for heterogeneous clusters, grids, and large-scale decentralized platforms. We start with platform/application models and review several trade-offs between "tractability" and "accuracy". The traditional scheduling objective, namely, predicting and achieving optimal execution time (or, makespan), must be abandoned. However, we show how to approach this objective by using the power of divisible, steady-state, and flow-based approaches. For very large-scale platforms, a centralized scheduling mechanism is not realistic — but how can one even hope for decentralized yet provably efficient schedulers? We present sophisticated algorithmic approaches to this goal, illustrated by simple, yet significant applications, such as the problems of scheduling independent tasks and of implementing collective communications (e.g., broadcast, multicast, etc.).

Session 9: Load Balancing

Session 10: Computational Science: Biology, Chemistry, and Physics

Session 11: Performance Evaluation and Models

Session 12: Input/Output

Session 13: Scheduling, 2

Session 14: Data-Intensive Applications

Session 15: Energy Considerations

Session 16: Compilers and Optimization

Session 17: Memory Sharing

Session 18: Communication and Coordination

Session 19: Fault and Failure Tolerance

Session 20: MPI

Banquet
Invited Speaker: Bill Dally, Stanford University
Challenges and Opportunities for Parallel Computing

The next several years promise to be a golden age for parallel computing research. Parallel computing is becoming mainstream with even desktop computers having multiple processors. Key research problems stand in the way of efficiently using this emerging mainstream capability. Exploiting locality is a central problem. Bandwidth is the critical resource that dominates cost and performance of modern computing systems - arithmetic is almost free - and the cost of bandwidth increases greatly with distance. Exploiting locality requires an architecture that exposes data location at all levels – so it can be optimized by the programmer and compiler. Efficiently mapping to such an exposed-communication architecture requires new programming languages and compilation techniques. This talk will discuss the challenges and opportunities in parallel computing research with particular attention to the challenge of locality. Examples will be drawn from the Imagine and Merrimac stream processor projects.

Friday, April 28, 2006

Keynote Speaker
Horst D. Simon, Lawrence Berkeley National Laboratory
Progress in Supercomputing: The Top Three Breakthroughs of the Last 20 Years and the Top Three Challenges for the Next 20 Years

• As a community we have almost forgotten, what supercomputing was like twenty years ago in 1985. The state of the art system then was a 2 Gflop/s peak Cray-2, with at that time phenomenal 2 GBytes memory. It was the era of custom built vector mainframes, where anything beyond 100 Mflop/s sustained was considered excellent performance. The software environment was Fortran with vectorizing compilers (at best), and a proprietary operating system. There was hand tuning only, no tools, no visualization, and dumb terminals with remote batch. If one was lucky and had an account, remote access via 9600 baud was state-of-the-art. Usually a single code developer developed and coded everything from scratch.

• What a long way have we come in the last twenty years! Teraflop/s level performance on inexpensive, highly parallel commodity clusters, open source software, community codes, grid access via 10 Gbit/s, powerful visualization systems, and a productive development environment on a desktop system that is more powerful than the Cray-2 from 20 years ago – these are the characteristics of high performance computing in 2005.

• Of course a significant contribution to this progress is due to the continued increase of computing power following Moore's law. But what I want to argue here is, that progress was not just simply quantitative alone. We did not just get more of the same at a cheaper price. There were several powerful ideas and concepts that were shaped in the last 20 years, that made supercomputing the vibrant field that it is today. As an active researcher in the field for the last 25 years, I will offer my subjective opinion, what were the real top breakthrough ideas that led to qualitative change and significant progress in our field.

• Retrospection leads to extrapolation: in the last part of the lecture, I will envision, what supercomputing will be like 20 years from now in the year 2025. Can we expect similar performance increases? How will supercomputing change qualitatively? And what are the top challenges that we will have to overcome to reach that vision of supercomputing in 2025?

Session 21: Routing

Session 22: Image Processing And Visualization

Session 23: Reconfigurable And Multiple-Width Systems

Session 24: Programming Abstractions

Session 25: Resource Allocation

Session 26: Partitioning And Refinement

Session 27: Collective Communication

Session 28: Distributed Coordination

Session 29: Symbolic Computing Applications

Session 30: Multithreading

Session 31: Runtime Optimizations

Session 32: Distributed Systems

Saturday, April 29, 2006

Workshops 12-19