WORKSHOP

ScalPerf'05:
SCALABLE APPROACHES to HIGH PERFORMANCE and HIGH PRODUCTIVITY COMPUTING

October 09-13, 2005

Bertinoro International Center for Informatics
`http://www.cs.unibo.it/bici/`

ABSTRACTS

Toward a Stochastic Analysis of Replacement Policies

Gianfranco Bilardi
University of Padova

The performance of replacement policies for managing different levels of the memory hierarchy have been widely investigated for over forty years. However, relatively few analytical results are available, so that most comparative studies are based on simulations.

In this paper, we consider a framework for stochastic analysis. The address trace is modeled as a stochastic process, assuming that the LRU stack distances form a sequence of independent, identically distributed random variables. Intuitively, the probability that a given address A is accessed at a given time t is a function $s(d)$ of the number d of different addresses appeared in the trace since A occurred last before t. A wide class of workloads can be modeled by varying the distribution $s(\cdot)$.

We show that the probability of a miss can be precisely obtained for some known policies while for other policies it can be reasonably estimated. We also formulate a distribution-dependent policy that can be shown to be (on-line) optimal for significant classes of stack-distance distributions. We also state a number of interesting open problems.

(The work reported in this talk is joint with K. Ekanadham and P. Pattnaik.)

Dealing with Memory Hierarchy when Generating High Performance Libraries

Francois Bodin
INRIA/IRISA

Advanced microprocessors rely on complex memory hierarchies to reduce memory access time. In average, they bring high performance but they also may generate many performance instabilities: a small change in the input data may produce a large increase in execution time. It is expected from high performance libraries such as BLAS, etc. that they exhibit few performance instabilities on a large range of inputs. To achieve this the libraries are built using a set of computation kernels, each kernel being tuned for some of the input parameters (vector size, data alignment). Currently, the libraries are mostly built by hand.

In this talk, we first address the performance instabilities due to memory accesses in the case of the Itanium2 architecture. Then we show how to automatically construct high performance libraries with limited performance instabilities while controlling the code size. The technical core is first a load/store instruction scheduling technique. The second part of the construction process is an algorithm to assemble a library routine from various computation kernels. The proposed method produces codes that outperformed high quality libraries such as Intel MKL.

Performance Analysis of Blue Gene/L Hierarchy Configurations

José Castanos
IBM Thomas J. Watson Research Center

The full 64K node Blue Gene/L system has been successfully installed and accepted at LLNL, and approximately 20 customers are already using smaller Blue Gene/L systems.

As part of the final acceptance test for LLNL, in the last few weeks we investigated the effects of transient errors on the system reliability of very large systems. In this presentation we will describe the different configurations of the memory hierarchy that we studied, and we will analyze their performance implications.

Perfomance Analysis Challanges in Current and Future Supercomputers

Luiz Derose
Cray Inc.

In order to achieve high-performance on future supercomputers, users need a new generation of performance tools that can handle the challenges associated with multiple levels of parallelism, hundreds of thousands of computing elements, and novel programming paradigms. Current performance tools can barely support the needs of existing architectures and will certainly not be adequate for future high-performance computing systems. A new generation of performance tools will have to be able to automate the process of performance analysis, and will require a hardware-monitoring infrastructure able to handle massive volume of performance data, with low overhead. In order to accomplish these goals, the design of this new generation of performance analysis tools must be integrated into the early stages of the architecture, operating system, and programming environment design. Otherwise, the productivity of these systems will be severely compromised. In this talk I will discuss the state of the art in performance tools and will describe the challenges associated with peta-scale systems.

Issues in Operating System Virtualization

Hubertus Franke
IBM Thomas J. Watson Research Center

In operating system virtualization multiple operating system (OS) images are collocated and concurrently running on the same server platform. Besides fault isolation, this allows for a higher utilization of the server itself. However this higher utilization also contributes to more resource contention in the memory hierarchy. In this talk we will give an overview of various operating system

virtualization technologies that are currently deployed in the industry. We then describe the issues that arise out of the various virtualization techniques with respect to the memory subsystem and the its management.

Cache Oblivious Stencil Computations

Matteo Frigo
IBM Austin Research Laboratory

A stencil describes the computation of a grid point at time $T+1$ as a function of neighboring grid points at time T. This computational pattern arises frequently in scientific computing, for example in explicit finite-difference methods for solving differential equations.

In this talk, we discuss a novel algorithm for stencil computations that applies to arbitrary stencils in n-dimensional spaces. On a machine with an "ideal cache" of size Z, for sufficiently large problems, the algorithm computes P grid elements while incurring $O(P/Z^{1/n})$ cache misses, which matches the lower bound of [Hong and Kung, 1981]. The algorithm is "cache oblivious": it does not contain the cache size Z as a parameter.

The algorithm was applied to LBMHD, a hydrodynamics program from the HPCC suite of DARPA benchmarks, obtaining a speedup of 7x with respect to the original code.

Joint work with Volker Strumpen.

System Software Design for A Mani-Core Chip Based Supercomputer System: A Case Study on Performance Issues

Guang R. Gao
University of Delaware

In this talk, we present a case study - illustrating the problems facing the design of petascale parallel computing systems based on a mani-core-on-a-chip architecture. We discuss several fundamental performance issues facing system software designers for such architectures including: execution models that support dynamic and adaptive multithreading, fine-grain synchronization, global name-space and memory consistency. A design study derived from the experience on IBM Cyclops-64 architecture will be discussed.

System Architecture Design at Extreme-Scale Using Predictive Performance Models

Adolfy Hoisie
Los Alamos National Laboratory

We will begin by briefly describing novel methodologies for performance analysis, modeling and prediction applicable to extreme-scale parallel architectures and applications developed by the Performance and Architecture Lab (PAL) at Los Alamos.

Of the various uses of performance modeling that will be mentioned, system architecture design will be showcased. We will describe the process of capturing the workload represented by entire applications into accurate predictive models. We will walk through the process of predicting system performance using these models. We will attempt to answer "what if" type if scenarios through examples of point-design studies for specific architectures.

Performance Characteristics of eCommerce Workloads

Joefon Jann
IBM Thomas J. Watson Research Center

Since the late 90s, eCommerce has been the major driving force for the growth in the world economy (eCommerce being commerce that uses the Internet as a key part of its commercial transactions). Hence, for the past decade, the growth in the IT (Information Technology)infrastructure has been shaped by the needs of eCommerce. Extensive use of componentization, object-oriented programming, and the separation of roles using container technologies have been the characteristics of the eCommerce software stack, which is somewhat different from the traditional HPC (High Performance Computing) application stack.

In this talk, I will review the characteristics of this eCommerce software stack as viewed by extensive and detailed performance measurements carried out by our team. In particular, I will present the observed variability in macroscopic performance measurements and the techniques we have developed to extract meaningful information from them. This talk will also present and discuss the observed cycle per instructions (CPI) bottlenecks in terms of CPU architectural features, such as branch prediction, cache miss, and translation miss.

Predictable Cache Performance - A Review and Challenges

Ben Juurlink
University of Delft

Because cache performance is difficult to predict, caches are not often used in real-time systems. Instead, software-controlled scratchpad memories are employed to mitigate the memory latency. Although scratchpad memories have predictable performance, they are inflexible because the mapping of data to the memory has to be reconsidered each time the code is changed. Furthermore, it is difficult to employ scratchpad memories in multiprocessor environments, because it implies that the designer also has to take care of process scheduling.

In this talk we will first review existing work. Predictable cache performance is already a problem when the cache is private to a processor but the processor is time-shared between several processes. However, when the cache is shared between several processors, the problem is even more severe. Then we will discuss the limitations of current approaches and present some research challenges. In addition, a general framework is presented to map a process network to a multiprocessor platform while meeting non-functional requirements.

Exploiting Predictibaility of Message-Driven Objects to Scale the Memory Hierarchy

Laxmikant Kale
University of Illinois Urbana-Champaing

The "memory wall" is growing increasingly taller: processors today are much faster than the DRAM memory, in terms of latency. The 4-decades old idea of virtual memory and memory hierarchy was designed in an old context, and must be re-examined. In particular, the "principle of locality" - a heuristic observation about behavior of programs - was used to justify organization of data into pages and cache-lines, along with block-fetching of entire lines (pages) when one item is referenced.

We argue that the principle of locality is a weak predictive principle, and stronger predictive principles exist. In particular, we argue for (what we call) a "principle of persistence": Expressing a computation in terms of message-driven objects allows us to predict accesses at runtime, based on peeks at the message-queue. This suggests a memory organization that uses small fast memories (SRAMs) as scratchpad memory, and permits a prefetch strategy with much better performance than a cache can provide. In particular, there is potential for reducing the "hit penalty" in addition to the reduction in miss ratio. Although developed in the context of parallel applications, these ideas are also useful for sequential programs. I will explain my proposals using examples in our research in parallel applications.

Optimization of the FD-TD Integration Scheme for the Maxwell Equations

Paolo Palazzari
ENEA and Ylichron Srl

In the talk, after reviewing the basics of the FD-TD (Finite Difference in the Time Domain) integration scheme for the Maxwell equations, I will discuss its memory and computational requirements, showing both its classical implementation and a new, memory aware, optimized implementation. Some performance results, obtained executing the optimized algorithm on different parallel systems, will be given.

A Bird's Eye View of the Cell Processor Architecture

Pratap Pattnaik
IBM Thomas J. Watson Research Center

Recent years have seen major advances in the console based game platforms, particularly in the story line sophistication and in the delivery of immersive experience. These have been possible by the deployment of significant amount of computational power in the game boxes. With the continued increase in the popularity of the game consoles, the computational requirements for them continue to increase. To address these needs, Sony, Toshiba, and IBM, by combining their respective knowledge in the area of processor and game design, developed an extremely efficient and high performance **PowerPC** based processor, known as Cell Broadband Engine, popularly referred to as Cell processor. Through its 8 Synergetic Processing Elements (SPE), this processor is capable of delivering an impressive amounf of computational power (256GFlops), in a very small chip size ($\approx 230\ sq.\ mm$). This talk will describe the architecture and the design of Cell in some detail and highlight the programming model required to harness its power.

On the Space and Access Complexity of Computation DAGs

Andrea Pietracaprina
University of Padova

We study the space and the access complexity of computations represented by Directed Acyclic Graphs (DAGs) in hierarchical memory systems. First, we present a unifying framework for proving lower bounds on the space complexity, which captures most of the bounds known in the

literature for relevant DAGs, previously proved through ad-hoc arguments. Then, we expose a close relationship between the notions of space and access complexity, where the latter represents the minimum number of accesses performed by any schedule of a DAG at a given level of the memory hierarchy. Specifically, we present two general techniques to derive bounds on the access complexity of a DAG based on the space complexity of certain subgraphs. One technique, simpler to apply, provides only lower bounds, while the other provides (almost) matching lower and upper bounds and improves upon previous well-known result by Hong and Kung.

(Joint work with G. Bilardi and P. D'Alberto)

The Price of Obliviousness

Keshav Pingali
Cornell University

Cache-oblivious algorithms provide a solution to the problem of writing programs that adapt automatically to memory hierarchies to optimize their performance. These algorithms, which are based on the divide-and-conquer paradigm, enjoy certain important theoretical properties such as I/O optimality, but there are few head-to-head comparisons of the experimental performance of cache-oblivious and cache-aware programs. This talk describes such a study for matrix multiplication. Starting from code that is completely oblivious to machine architecture, we successively add "awareness" to different architectural features by optimizing the code to take advantage of those features until we get a completely cache/architecture-aware code. Our experiments show that obliviousness has a significant penalty on current architectures, and that it will not be easy to eliminate this penalty.

The Potential of On-Chip Multiprocessing for QCD Machines

Geppino Pucci
University of Padova

We explore the opportunities offered by current and forthcoming VLSI technologies to on-chip multiprocessing for Quantum Chromo Dynamics (QCD), a computational grand challenge for which over half a dozen specialized machines have been developed over the last two decades. Based on a careful study of the information exchange requirements of QCD both across the network and within the memory system, we derive the optimal partition of die area between storage and functional units. We show that a scalable chip organization holds the promise to deliver from hundreds to thousands flop per cycle as VLSI feature size scales down from 90 nm to 20 nm, over the next dozen years.

Joint work with: G. Bilardi, A. Pietracaprina, F. Schifano and R. Tripiccione

Optimizing Numerical Simulation Kernels for Current Instruction and Memory Architectures

Ulrich Rüde
University of Erlangen-Nuernberg

Many scientific algorithms are limited by main memory access. While cache blocking techniques can overcome this issue, often they do not work as efficient as expected. We will show that this is often caused by limitations of the compiler to generate efficient code for more complicated loop structures, especially in 3D. Furthermore on many modern CPUs it is crucial to use dedicated instructions (e.g. SIMD) to reach full memory and arithmetic performance. We will show these issues on the example of an optimized Red-Black Gauss-Seidel iterative smoother. While these optimizations work well with structured data for many problems in scientific computing it is necessary to use unstructured grids. Two properties that are problematic in dealing with very large systems are the memory requirements of explicitly storing the discretization matrix, and the inherent indirection involved in accessing the matrix entries. We developed a framework that attempts to remove limitations on the size of the problem that can be solved, using finite element discretizations of partial differential equations by using a process of regular refinement of an unstructured input grid, to generate nested hierarchy of patch-wise structured grids that are suitable for use with geometric multigrid. The use of stencil-based component algorithms in the multigrid implementation makes it possible to achieve extremly high performance which leads to an efficient solver that is scalable to thousands of processors.

Performance Issues in a Historical Perspective

Einar Rustad
Dolphin Interconnect Solutions Inc.

In this talk I would like to take a look at the challenges of future high performance computing in a historical perspective. For some time now, performance of HPC systems has been driven by the continuous development of microprocessor semiconductor technology. Going back 20 years, the microprocessors were not competitive at all in this space; even in price performance a Cray came out better because of the lousy (or missing) floating point performance of early microprocessors. At that time (in the 1980s) several special purpose architectures were developed to handle performance critical tasks, many of them for military applications. Many were done as attached processors to general-purpose front-ends. As we now see the single processor performance improvements diminish, what will be the way forward for HPC - just plain parallelism, emergence of novel architectures or just simply re-emergence of partly forgotten architectures from the 80s?

A Research Program Proposal: Fault Equations for the Memory Hierarchy

Y.C. Tay
National University of Singapore

A *fault* is a reference to one level of the memory hierarchy that results in the retrieval of an object from the slower next level. Fault rates depend on the interaction between memory references and management (e.g. replacement) policy. This makes their analysis mathematically intractable, so analysts have had to impose simplifying assumptions (hypothetical reference patterns, idealized replacement policies, non-interacting processes, etc.).

Moreover, such bottom-up performance modeling, repeated at each level of the hierarchy, is not scalable: it is labor-intensive, and the results cannot keep pace with the changes in hardware design, evolution in software versions and escalation in system complexity.

This presentation proposes a research program to address this issue. Specifically, the program calls for a concerted research effort to find a set of parameterized fault equations, one for each level of the hierarchy, with the parameters calibrated adaptively and automatically. Each equation can then be used online for memory allocation, power management, policy choice, quality-of-service control, etc.

Such a paradigm shift to a uniform, top-down approach for fault analysis and management may have a better chance of coping with the myriad changes and variations in commercial systems, and the infinite variety of application workloads.

Montecalro Simulations of Spin-Glass: a Physicist's Nightmare and a Computer Architect's Paradise

Raffaele Tripiccione
University of Ferrara

Spin glasses are theoretical models of some magnetic systems in which the interactions between magnetic moments are "in conflict" with each other, due to some frozen-in structural disorder.

Analytical techniques are not able to capture any significant physical information about these systems, so Monte Carlo simulation techniques are heavily used in the field. Simulations are made extremely difficult by the irregular "corrugated" structure of the energy landscape in the phase space of the system configuration, so the onset of thermal equilibrium is extremely slow: the state-of-the-art is that the largest system for which thermalization has been achieved has a size of only 20^3 lattice sites.

Spin glass systems have been heavily studied with dedicated computer systems, starting in the late seventies at Caltech, followed by a large system (SUE) developed at Zaragoza a few years ago.

In this talk, I review the algorithmic structure of a Monte Carlo simulation of a a spin-glass system, focusing on the specific features suggesting that an application-specific architecture may be extremely effective, namely: - explicit and large parallelism - very regular algorithmic structure - unconventional arithmetics - very small size of the computational data base.

I then describe the architecture of yet a new spin-glass specific architecture under developments at Ferrara and Zaragozza (known as IANUS), entirely based on FPGA systems. The FPGA approach brings obvious benefits in terms of development time. At a more fundamental level, however, recent FPGA architectures offer large "embedded" memory structures able to implement

the memory hierarchies appropriate for the system and to satisfy the huge bandwidth needed to sustain processing.

A relatively small IANUS system, with 256 FPGA-based processors that we expect to bring on-line in fall 2006, should deliver an application-specific computing power equivalent to roughly 20000 high-end PC's.

CODACS Architecture: a Development Tool for Embedded System Processor Prototyping

Lorenzo Verdoscia
ICAR-CNR

In several real-time applications, the advent of FPGAs and Intellectual Property core availability allow great freedom in the customization of processors for embedded systems. One of the new challenges that such technologies present is how to implement a high performance application on devices with hundreds coarse-grained computing units running at 200 MHz, rather than on one processor running at 20 GHz, which makes the prototyping phase crucial in the development of these systems.

Consequently, to profit by spatial parallelism that such devices offer becomes a non marginal issue. From the architectural point of view, at least two questions arise: how to exploit such spatial parallelism; how to program such platforms. The first one brings us to seriously reconsider the dataflow paradigm, given the fine grain nature of its operations. The second one brings us to seriously reconsider the functional programming style, given its inherent simplicity in writing parallel programs. The talk will discuss our experience in combining these two approaches inside CODACS (COnfigurable DAtaflow Computing System) demonstrator. The resulting architecture offers interesting properties not only as stand-alone computing system but also as development tool for special-purpose processor prototyping activities.