AHPCRC
Army High Performance Computing Research Center

AHPCRC Bulletin

Volume 2 Issue 1

Distribution Statement A: Approved for public release; distribution is unlimited.

HPC Enabling Technologies, Advanced Algorithms

INSIDE THIS ISSUE

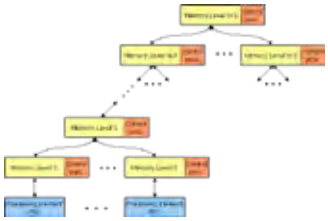
The Army High Performance Computing Research Center, a collaboration between the U.S. Army and a consortium of university and industry partners, develops and applies high performance computing capabilities to address the Army's most difficult scientific and engineering challenges.

AHPCRC also fosters the education of the next generation of scientists and engineers—including those from racially and economically disadvantaged backgrounds—in the fundamental theories and best practices of simulation-based engineering sciences and high performance computing.

AHPCRC consortium members are: Stanford University, High Performance Technologies Inc., Morgan State University, New Mexico State University at Las Cruces, the University of Texas at El Paso, and the NASA Ames Research Center.

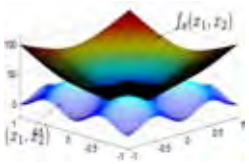
http://www.ahpcrc.org

High performance computing (HPC), once the sole province of room-sized supercomputers, now also includes clusters built from commercially available components. Multiprocessor personal computers are now commonplace, and personal multicore parallel-processing machines are likely in the foreseeable future.

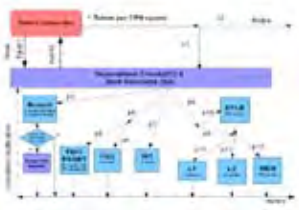


At present, the parallel codes required by HPC machines are largely custom-built and optimized for each cluster configuration or supercomputer on which they run. Researchers in AHPCRC Technical Area 4 focus on improving processes for developing scalable, accurate parallel programs that are easily ported from one machine to another and that can be optimized for resource-efficient performance in a variety of computing environments.

Clockwise from top: Memory hierarchy, mathematical surrogate function, processor modeling, flexible machine architecture. Graphics provided by AHPCRC researchers.



They do this by analyzing the performance of programs as they execute; developing hardware and software capabilities in tandem; and developing algorithms that work well for modeling, simulation, and problem-solving in a variety of HPC environments. ✯



Note:
The AHPCRC website address is now www.ahpcrc.org. Your current bookmark for http://me.stanford.edu/research/centers/ahpcrc will also work, no changes necessary.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
AHPCRC (Army High Performance Computing Research Center) Bulletin. Volume 2, Issue 1, 2011	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army High Performance Computing Research Center,c/o High Performance Technologies, Inc,11955 Freedom Drive, Suite 1100,Reston,VA,20190-5673**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**24**	

Stream Programming for High Performance Computing

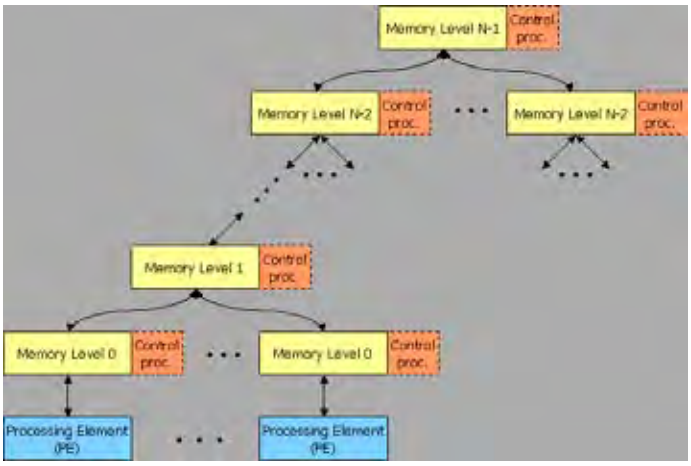
Parallel programming is an intrinsic part of high performance computing. Whether a programmer is adapting existing software or implementing new functionality, codes must be designed to run accurately, reliably, and efficiently on systems with tens to thousands of processors working cooperatively. Alex Aiken, Professor of Computer Science at Stanford University, says, "Programmers tend to think of parallel programming as a problem of dividing up computation, but in fact the most difficult part of parallelism is often communication, simply moving the data to where it needs to be."

Aiken, along with Stanford computer science professors William Dally and Patrick Hanrahan, is working to develop the Sequoia programming language. Their efforts, supported in part by AHPCRC, will provide Army researchers with the ability to port parallel programs to many types of computing systems and architectures without sacrificing performance.

To write a program that achieves the best performance on a specific system, a programmer must understand the system and design the code to fit the specific characteristics of the system. Code that works especially well on one architecture may not achieve anywhere near the same level of performance on a system with a different size or structure. Conversely, programs written to be highly portable may not perform optimally on any system. The Sequoia language seeks to address this problem by allowing programmers to write code that is functionally correct on any system, then tune the performance to the characteristics of a specific system, using the underlying Sequoia interface.

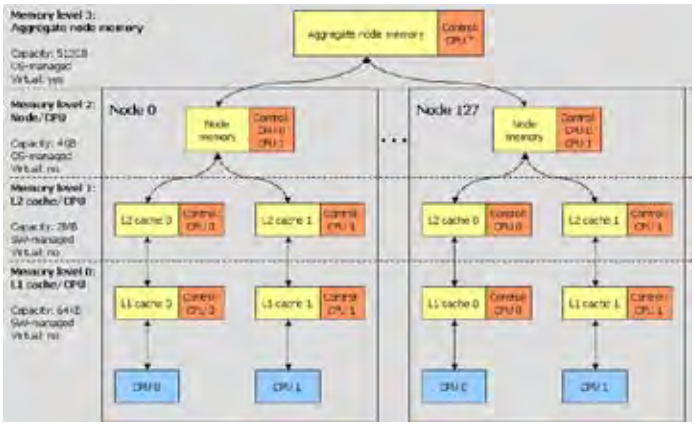
Memory Hierarchies

Traditional desktop computers use random access memory (RAM) to hold data for access by one or two processors. Programming applications for these computers does not require explicit mechanisms for getting data into and out of memory, since this is handled transparently by the hardware. As technological and programming advances place greater demands on memory access, hardware-managed data caches have been implemented to bridge the gap between the rate at which a processor requests data and the rate at which the computer's memory can provide it.



Schematic of Sequoia's hierarchical memory design. Graphics courtesy of Alex Aiken, Stanford University.

High-performance parallel architectures, including IBM's Cell (high-throughput) and NVIDIA and ATI's GPUs (graphics processing unit, *see "Terms and Abbreviations," page 5*) processors, increase performance and efficiency by allowing software to manage a hierarchy of memories. *(One example is shown above.)* Such systems are highly parallel—they consist of many processing elements (PEs) operating in isolation, drawing data only from their own small, fast local memory devices—the "leaves" of the memory "tree." Individual PEs do not have access to the entire memory hierarchy, and there are no data caches. A conventional high-latency, low-bandwidth external memory device serves as the "root." Between the root and leaves are various memory structures such as on-die storage, local DRAM (dynamic random access memory), or remote memory with high-speed interconnects. Data and code move between levels in the hierarchy as asynchronous block transfers explicitly orchestrated by the software. This "exposed-communication architecture" requires programmers to build into the software the directives to move data between nodes at adjacent levels of the memory hierarchy. Explicit management of the memory hierarchy gives the



Virtual levels in Sequoia represent an abstract memory hierarchy without specifying data transfer mechanisms, giving the programmer the ability to adapt data flow to a variety of machine architectures.

programmer direct control over locality, allowing the programmer to write locality-aware programs and thus improving performance. The node-level orchestration aspect bears emphasizing, because parallel codes have typically addressed the horizontal communication of data among nodes of a machine. Newer architectures also require managing the data as it moves vertically between levels of a memory hierarchy.

The Sequoia language places data movement and placement explicitly under the control of the programmer. Machine architecture is represented in the language as abstracted memory hierarchy trees *(schematic, above)*. Self-contained computations called "tasks" are used as the basic units of computation. Tasks are functions, and thus, free of side effects. When a task is invoked, it is normally run at the next lower level of the memory hierarchy; a parallel loop with a task as its body launches the task calls in parallel on "children" of the current node. Task parameter passing is copy-in-copy-out, so a task will copy its argument data from the "parent" memory, run the task locally one level lower in the memory hierarchy, and then copy the results back to the parent. Tasks provide for the expression of explicit communication and locality, isolation and parallelism, algorithmic variants, and parameterization. These properties allow Sequoia programs to be portable across machines without sacrificing the ability to tune for performance.

Mapping to Machines

Tasks are arranged in a hierarchy, and they can call other tasks. The novel problem in compiling a Sequoia program is to map the task hierarchy on to the memory hierarchy of a given machine. The programmer only works with an abstract memory hierarchy, one which does not depend on the specific memory sizes, or number of computer nodes, or depth of a particular memory hierarchy. This allows a programmer a high degree of control over both the data and the parallel computation without tying a program to a particular machine architecture.

Sequoia programs may have multiple mappings, one for each target architecture. These mappings are, of course, different. For instance, the best block size for one machine will be very different from the best block size for another machine. The portions of the program that deal with the higher-level code common to all machines are kept separate from the part that handles machine-specific mapping and optimization. Programmers can control all details of mapping an algorithm to a specific machine, including defining and selecting values for the tunable parameters. "Tunables" allow the programmer to specify the initial sizes of arrays, blocks, and data structures. Tunables help keep programs machine-independent, but suitable tunable values must be found for each given architecture. Because tunables interact, the space of possible tunable values can be very large and complex.

The Stanford team has completed the design and implementation of a Sequoia "autotuner" that automatically searches the space of tunable parameters for the highest performance combinations, relieving the programmer of specifying all but the most critical tunables. In fact, to date they have yet to find an example of a program where any choices made by a programmers are superior to the program tunable values set by the autotuner. In at least one program, programmers were never able to find suitable tunable settings by hand, but the autotuner was able to find a high performance combination of tunable values.

continued on page 4

Stream Programming
continued from page 3

Language for Location
The syntax that Sequoia uses is an extension of the C++ programming language, but Sequoia introduces language constructs that produce a very different programming model. Unlike C++, which provides no information on the location in the machine where a computation is performed, Sequoia makes it easier to develop a parallel program that is "aware" of the memory hierarchy configuration in the machine on which it is running. Computations are localized to specific memory locations, and the language mechanisms describe communications among these locations.

Dynamic Data
For many irregular applications, details of the data transfers that dominate performance are not known until run-time. For instance, the structure of a sparse matrix or a graph being traversed is not known until the data structure is loaded or generated. These structures may also be dynamic, changing periodically during program execution. To achieve high performance on such applications requires that the problem subdivision at the core of Sequoia be extended to use run-time information.

Subdividing graph structures requires balancing parallelism and locality considerations. That is, connected nodes of the graph are grouped to enhance re-use and to ensure that processors that produce information are located near those that use this information. At the same time, nodes along an activity front should be distributed to enable parallelism.

One way to strike the right balance between these two goals is to optimize the program's execution using run-time libraries in Sequoia that perform graph partitioning and distribution. Another approach is to invoke a portion of the compiler at run time to reoptimize the partitioning periodically as the data structures evolve.

Irregular Access
Initial experiments during the development of Sequoia focused on applications, such as matrix multiplication and three-dimensional fast Fourier transforms, with relatively regular data access patterns. The Stanford team has also completed the design of Sequoia extensions that cleanly allow the expression of computations with irregular access patterns, such as graph traversal algorithms. The redesigned language has been implemented, and a new compiler is being completed. The new design preserves the original Sequoia language as a subset of the new language. The compiler has been upgraded to provide reasonable error messages, robustness, and ease of use in preparation for releasing the Sequoia system outside of the research group. An Army radar application that was previously compiled for GPUs is being ported to the new version of the language in the context of the new Sequoia compiler, with the intention of making this a much easier program to write, and making it immediately usable on other architectures.

IBM Roadrunner supercomputer, Los Alamos National Laboratory. (Photograph courtesy of the U.S. Department of Energy.)

Putting Sequoia to the Test
A complete Sequoia programming system has been implemented, including a compiler and runtime system for both GPUs and distributed memory clusters that delivers efficient performance running Sequoia programs on both of these platforms. An alpha version of this programming system will soon be made public.

A major system of tests of Sequoia programs was completed on Cerillos, the open version of the Roadrunner supercomputer at Los Alamos National Laboratory *(see photo)*. Roadrunner, the world's first petaflop computer, and until recently the top supercomputer in the

world, has a very deep memory hierarchy, making it an ideal testbed for Sequoia applications. Several scaling issues have been identified as a result of this implementation, and these have been corrected. This work has also pointed to a new research direction in the layout of data; specifically, understanding in a language with explicit locality how a programmer may specify sophisticated mappings of data to processors, such as the partially replicated data that arises in programs with "ghost" cells.

The Road Ahead

For applications where the time spent in recompiling would be dwarfed by the actual computation, the Stanford group plans to investigate much later binding of compilation decisions, perhaps even providing a way to invoke the full Sequoia compiler at run-time.

One of the largest remaining obstacles to programmer productivity is writing high-performance "node code," sequential kernels for the complex instruction sets of contemporary processors. The group plans to provide more semi-automatic support for the performance-oriented kernel programmer, including taking advantage of vector hardware (which, in the Sequoia view of the world, is just another level of the memory hierarchy).

The Stanford group is initiating work with Center collaborators on additional, larger applications in early 2010. In addition, release of the existing compiler to Army and other users is projected for early 2010. ✯

References:

https://zebra.llnl.gov/seminar/view.php?id=351 (Alex Aiken talk abstract, used in intro)

http://www.stanford.edu/group/sequoia/cgi-bin (Sequoia home page)

Fatahalian, K., Knight, T. J., Houston, M., Erez, M., Horn, D. R., Leem, L., Park, J. Y., Ren, M., Aiken, A., Dally, W. J., and Hanrahan, P. 2006. Sequoia: Programming the memory hierarchy. In Proceedings of the 2006 ACM/IEEE Conference on Supercomputing.

Knight, T. J., Park, J. Y., Ren, M., Houston, M., Erez, M., Fatahalian, K., Aiken, A., Dally, W. J., and Hanrahan, P. Compilation for Explicitly Managed Memory Hierarchies PPoPP 2007

Terms and Abbreviations

Blade server is a stripped-down server computer with a modular design optimized to minimize the use of physical space and minimize power consumption, while retaining all the functional components of a computer

Cell processors, of the type used in Sony PlayStations, use a novel memory coherence architecture that places priority on efficiency, bandwidth, and peak computational throughput over simplicity of program code. Cell presents a challenging environment for software development.

CPU Central Processing Unit, the part of a computer that fetches, decodes, executes, and writes back the sequence of instructions making up a computer program

DIMM Dual In-line Memory Module, a series of DRAM integrated circuits amenable to the 64-bit data paths now commonly in use in PCs

DMA Direct Memory Access allows computer hardware subsystems to access system memory for reading or writing, independently of the CPU

DRAM Dynamic Random Access Memory, a form of temporary high-density data storage

FPGA Field-Programmable Gate Array, an integrated circuit that functions as a hardware accelerator, designed to be configured post-manufacture by the customer or designer to implement logical functions

GPU Graphics Processing Unit, a specialized processor, designed to be efficient at manipulating computer graphics, that functions as a hardware accelerator

LUT Lookup Table, a data structure that reduces processing time by precalculating certain values for use in later calculations

Opteron is a line of server and workstation processors manufactured by AMD

PCI Peripheral Component Interconnect, an industry-standard bus for attaching peripheral devices to computers

RAMP Research Accelerator for Multiple Processors, a shared, supported, FPGA-based platform for multi-core architectures that supports research on parallel architectures and software

RDMA Remote Direct Memory Access, permits direct access between the memory areas of two computers without the intervention of either computer's operating system, which facilitates rapid throughput in massively parallel clusters

SIMD Single instruction–multiple data stream

Tflop teraflop, or one trillion floating point operations per second, a measure of the speed at which a computer carries out calculations

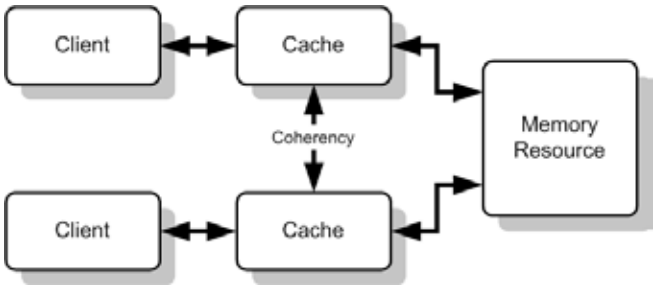
Transactional memory simplifies parallel programming by supporting regions of the code that can be executed independently

Flexible Architecture Research Machine (FARM)

As heterogeneous systems that combine CPUs, GPUs and FPGAs (central processing units, graphics processing units, and field-programmable gate arrays, *see "Terms and Abbreviations, page 5*) become more common, it is necessary to develop and customize software and hardware in tandem to ensure that both achieve optimum performance. A more accurate picture of parallel software performance emerges when this software can be tested at full scale and full speed, but the ability to perform such tests is limited by the availability of large-scale computing resources. A readily available, reconfigurable testbed could facilitate algorithm and software development and provide a means of testing new architectures.

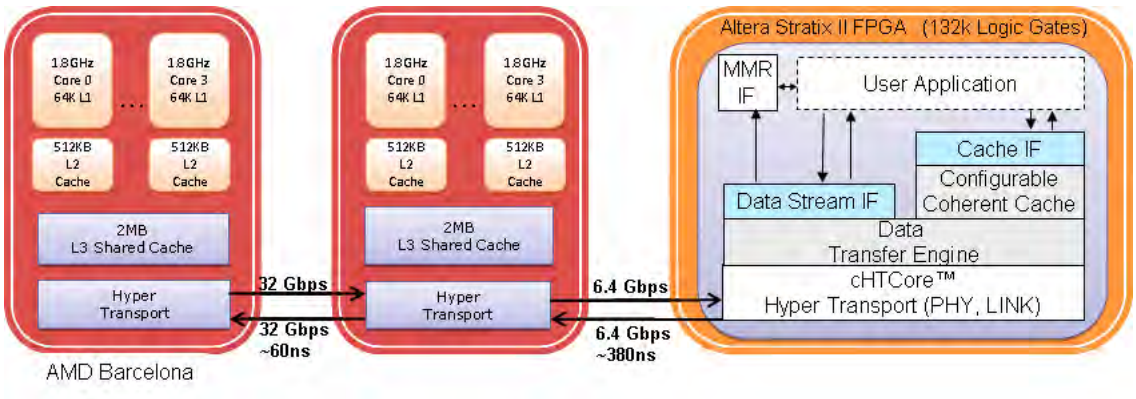
Stanford University Electrical Engineering and Computer Science professors Kunle Olukotun and Christos Kozyrakis are developing the Flexible Architecture Research Machine (FARM), a vehicle for hardware/software codesign, intended to accelerate architecture and algorithmic research on novel parallel models. FARM facilitates realistic application development environments for tightly-coupled heterogeneous systems, combining commercial server technology with FPGAs to provide a flexible and scalable high-performance parallel machine that can run full-sized applications at full hardware speeds.

Like the Cray XD1 supercomputer, FARM integrates CPUs and FPGAs; but FARM goes further, with the inclusion of GPUs. Moreover, FARM connects the FPGAs directly to the CPUs through cache-coherent links to maintain the consistency of data stored in local caches of shared resources *(illustration above)*, which provides for faster and finer-grained FPGA–CPU communication and allows researchers to use the FPGAs to enhance the memory system with transactions or streams.



Cache coherency maintains the consistency of data stored in local caches of shared resources. (Wikimedia Commons, public domain.)

For algorithm development using existing architectures, the FARM can be used as a high-density, high-bandwidth supercomputer. For architecture and software research on novel architectures, the FPGAs can be programmed to introduce new functionality into the memory system. Unlike commercial CPU–FPGA systems, the FARM CPU and FPGA communicate using cache coherent hypertransport links (bidirectional high-bandwidth, low-latency point-to-point links). The application hardware block is defined by the application. Coherency support makes it possible for the CPU and FPGA to communicate in a fine-grained manner with very low latency (delay between the executable instruction commanding an



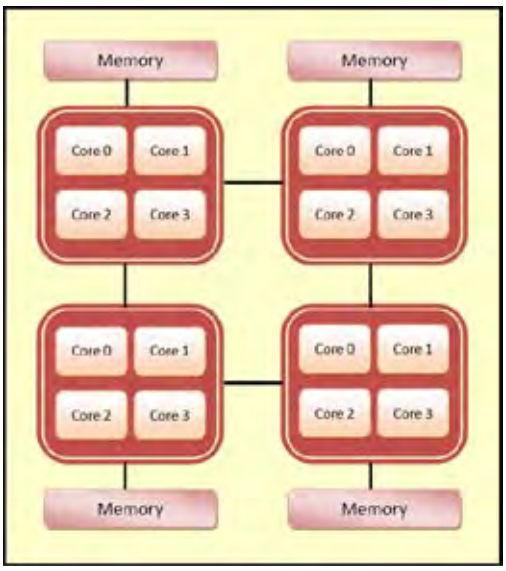
action and the hardware performing the action), and it allows the FPGA to "cache" shared data inside the configurable coherent cache. This capability makes it possible to implement protocols that interact directly with the memory system.

The Stanford group has a fully operational FARM system *(diagram, previous page)* consisting of 16 AMD Opteron CPU cores and one Altera FPGA. The completed FARM prototype system has been used to prototype a hybrid hardware–software transactional memory (hybrid-TM) system that can run full-sized TM applications—an example of the good performance achieved through the careful interplay and codesign of the TM software and hardware. This codesign capability was only possible because the Stanford group was able to change hardware and software at the same time and still experiment with realistic full-size applications using the FARM environment. Transactional memory promises to reduce substantially the difficulty of writing correct, efficient, and scalable concurrent programs.

The Stanford group has implemented two versions of the hybrid-TM system, one optimized for large transactions and one for small transactions. Both versions achieve substantial performance improvements over a software TM system for their target transaction sizes.

In the course of developing the hybrid-TM system, the Stanford group created a generic cache coherent interface inside of the FPGA that makes it much simpler to prototype other application accelerators. The working high-speed (200 MHz) cache-coherent interface between the multi-core CPUs and FPGA chips uses coherent hypertransport. This is one of a few systems in the world that has this capability. The base prototype was purchased from A&D Technology. Considerable engineering effort was expended in developing and improving the FPGA design to get the system working reliably and at high speed. Drivers have been developed for FARM using both the Open Solaris and Linux operating systems.

continued on page 8



Above and previous page: two possible configurations of FARM. (IF=interface) Graphics courtesy of Kunle Olukotun and Christos Kozyrakis, Stanford University.

More on Transactional Memory

Transactional memory (TM) promises to substantially reduce the difficulty of writing correct, efficient, and scalable concurrent programs. "Bounded" and "best-effort" hardware TM proposals impose unreasonable constraints on programmers, while more flexible software TM implementations are considered too slow. Proposals for supporting "unbounded" transactions in hardware entail significantly higher complexity and risk than best-effort designs. Hybrid Transactional Memory is an approach to implementing TM in software so that it can use best-effort hardware TM (HTM) to boost performance, but does not depend on HTM. Programmers can develop and test transactional programs in existing systems today, and can enjoy the performance benefits of HTM support as it becomes available.

Adapted from: P. Damron et al. Proceedings of the 12th international conference on Architectural support for programming languages and operating systems table of contents. San Jose, California, USA
http://portal.acm.org/citation.cfm?id=1168857.1168900

See also:
An Effective Hybrid Transactional Memory System with Strong Isolation Guarantees. Chi Cao Minh, Martin Trautmann, JaeWoong Chung, Austen McDonald, Nathan Bronson, Jared Casper, Christos Kozyrakis, Kunle Olukotun. ISCA'07, June 9–13, 2007, San Diego, California, USA. ACM 978-1-59593-706-3/07/0006
http://tcc.stanford.edu/publications/tcc_isca2007.pdf

FARM

continued from page 7

Two techniques have been developed for tolerating the latency of fine-grained asynchronous communication with an out-of-core accelerator. These techniques are applicable to any accelerator, but only work with a cache-coherent coupling between the FPGA and the CPU. A system for Transactional Memory Acceleration using Commodity Cores (TMACC) has been designed that uses general-purpose out-of-core Bloom filters to accelerate the detection of conflicts between transactions. A complete hardware implementation of TMACC using the FARM is the only hardware implementation that the Stanford group is aware of that handles large transactions. The potential of TMACC has been demonstrated by evaluating the implementation using a custom micro-benchmark and the full STAMP benchmark suite. For all but short transactions, it is not necessary to modify the processor to obtain a substantial improvement in TM performance. For medium to large transactions, TMACC outperforms a software-only TM system by 2–5 times, showing maximum speedup within 8% of an upper bound on TM acceleration.

Eventually, the FARM system will be scaled beyond a single node and the software infrastructure will be developed to make heterogeneous systems easier to program. Ideally, the system will include enough flexibility to satisfy programmers, without sacrificing an excessive amount of speed or introducing undue complexity into the system. In addition, the system must be amenable to adaptation as newer technologies and capabilities evolve. Discussions are in progress with ARL/CISD about how FARM might be used to accelerate applications of interest to the Army, including work in the machine learning area. ✯

FARM Specs

FARM combines commercial server technology with FPGAs to provide a flexible high-performance parallel machine. The basis for FARM is a conventional blade server that accommodates multiple 64-bit Opteron blades, each with a multi-core chip, DRAM DIMMs, and a PCI-Express connection for high-end GPU board.

FPGAs are introduced by removing one of the Opteron blades and introducing in its place a commercially available blade with a high-density FPGA chip. The FPGA blade is directly connected to the Opteron blades through a cache-coherent Hyper-Transport link.

The FPGA blade can access the DRAM, GPU, and network resources available in other blades without interrupting the CPUs. The high-speed network interfaces (e.g. Infiniband or 10G-Ethernet) and appropriate logic in the FPGAs makes it possible to extend communication protocols and memory models across multiple blade chassis in on a standard server rack.

Overall, a single FARM rack will include up to 126 Opteron chips (504 cores, 32 TFLOPS – double precision), 72 GPUs (144 TFLOPS – single precision), 72 FPGAs (~21 million LUTs) and 1 Tbytes of DRAM. The exact balance of depends on the mix of boards and components used in the specific machine configuration.

FARM runs on OpenSolaris, an open-source, Unix-based operating system based on Sun Microsystems' Solaris.

Simulation & Modeling to Enhance the Performance of Systems of Multicore Processors

Resource-intensive applications, including large-scale simulations, can take weeks to execute, even on the most powerful computing systems. Thus, it is critical to design and tune software to use computing resources efficiently, and to incorporate effective mechanisms for error recovery. On the hardware side, computing systems incorporate an ever-increasing variety of processors, memory devices, and I/O (input–output) subsystems. The challenge is to build software architectures that are capable of functioning on a variety of configurations without sacrificing performance and accuracy.

Patricia Teller and Sarala Arunagiri (University of Texas at El Paso), Jeanine Cook (New Mexico State University), and their co-workers and students are using a three-pronged approach to optimizing and tuning application performance on heterogeneous computer nodes: measurement, acceleration, and modeling. They are testing their concepts on Chimera, a research computing cluster with a variety of processor architectures and hardware accelerators, that was installed at UTEP in 2008. Chimera is equipped with Opteron, Niagara 2, and Cell/PS3 processor architectures, as well as hardware accelerators. *(See "Terms and abbreviations," page 5.)*

Initial efforts to enable accurate application-to-architecture mapping have shown good progress. Optimizing the performance of an application requires knowledge of the characteristics of the hardware system on which the application runs. It also requires knowing the characteristic resource needs of the application itself. Dynamic profiling and monitoring tools, analytical models, and simulation are used to analyze application behavior in terms of resource needs, such as CPU and memory hierarchy characteristics. This permits identification of poorly performing or frequently executed parts of the code—and, when possible, modifying the code or system software to decrease overall execution time.

Measurement

The Chimera cluster at UTEP. Photographs and graphics courtesy of Pat Teller, UTEP.

System performance is analyzed by porting applications to various computing platforms—a challenging task at present. One way to do this is to port an application to a platform without initially requiring the application to take full advantage of the platform's processor architecture. For example, it is fairly easy to port an application to a Cell Broadband Engine on Chimera if it executes solely on the host PowerPC, not taking advantage of the eight available SPUs (synergistic processing units, a feature of the Cell processor). The next step requires a thorough processor, memory, and I/O subsystem performance analysis and characterization, and optimization of the base code.

In the past few years, many programming models, languages, and platforms have been developed to aid programmers in porting legacy codes to new multicore, multithreaded architectures. Cilk, OpenCL, and Sequoia *(article on page 2)* are among the many proposed languages and platforms designed for this purpose. To facilitate performance analysis studies on Chimera, CUDA and OpenCL have been installed, along with the library CUBLAS (Compute Unified Basic Linear Algebra Subprograms), the OpenCL profiler, pyCUDA, and the PAPI patch, which facilitates use of hardware performance counters. User support documents are being developed for all of the new software installed on Chimera, and are available to users via a user-only wiki. These include a comparison of OpenCL and CUDA, which is being developed from a programmer's perspective.

Acceleration

Performance optimization through acceleration involves modifying application codes and algorithms to adapt to the underlying architecture, scheduling

continued on page 10

Performance Simulation and Modeling

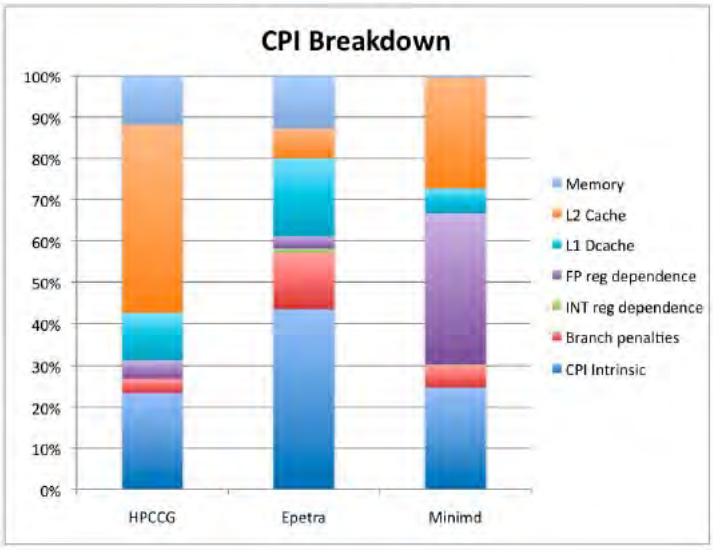
continued from page 9

appropriate threads to multi-threaded cores, and using hardware accelerators such as GPUs and FPGAs (graphics processing units and field-programmable gate arrays) to decrease the execution time of key portions of the application code.

Hardware accelerators: GPUs and FPGAs are generally used to accelerate portions of an application code that can be executed in parallel with other code tasks and are amenable to being implemented on these architectures. Programming languages and development environments such as NVIDIA's CUDA make it easier to port applications so that they execute efficiently on GPUs. Much effort has been focused on porting kernels to Chimera's NVIDIA Tesla GPUs. Some simple matrix manipulation codes have been ported, as well as an image processing code that extracts dozens of features from an image. The UTEP/NMSU group is studying Army applications and benchmarks related to science and surveillance, with the intent of choosing two of them for GPU implementation. They are identifying candidate applications in collaboration with Army researchers.

Simultaneous multithreading (SMT) is a technique for improving the overall resource utilization and throughput of superscalar CPUs with hardware multithreading. Replication of key CPU hardware (e.g., the register file and instruction buffer, one per hardware thread) permits multiple independent tasks (threads of execution) to execute concurrently on the hardware threads and share other SMT CPU resources. SMT CPU throughput depends on the amount of interference among the concurrently executing tasks that compete for SMT shared resources. Tasks with different shared resource needs do not interfere with one another and, thus, do not inhibit each other's execution.

Recent work at UTEP showed that the aggregate performance experienced by a given pair of applications scheduled to execute concurrently on a POWER5 SMT core with two hardware threads can vary significantly depending on the hardware thread priority settings. The research group developed a methodology, based on application signature sets, that, given a co-schedule, predicts priorities that will minimize application interference and deliver best throughput (IPC, instructions per cycle). The default priority settings assign equal opportunity to use the shared SMT CPU resources. An initial implementation of the methodology for an IBM POWER5 processor produced throughput gains over the default priorities between 0% (11 co-schedules) and 16.42% (9 co-schedules). For the eight co-schedules with floating-point unit usage that exceeds that of the fixed-point unit by 10% or more, the methodology yields a throughput improvement of 3.56–16.42%.



Computing resource allocation in a Niagara 2 single-core processor. (CPI is cycles per instruction.)

This research also showed that 17 of the possible 10,000 POWER5 application signatures (application characteristics related to use of shared SMT CPU resources) sufficed to characterize 95.6% of the execution time of the applications studied: 20 SPEC CPU2006 benchmarks, 3 NAS serial benchmarks, and 10 PETSc KSP solvers. The group is currently investigating anomalies in POWER5 hardware thread scheduling.

The group has documented and catalogued the set of scripts used for developing application signature sets

and SMT execution of application pairs, and they have documented the design of the scripts themselves. This has produced a report that serves as a user manual for this research.

I/O subsystems of high-performance computer systems generally include RAID (redundant array of independent disks) storage as a building block at levels of the memory hierarchy that experience high I/O contention. Under such conditions, I/O schedulers must provide performance isolation and differentiated service to concurrently-active clients. A performance isolation strategy is successful when each workload's I/O performance is similar to that achieved with a dedicated storage utility of a certain fixed capacity—it guarantees each competing workload a share of storage performance. When shares are proportional to workload priorities, the storage system is said to provide differentiated service as well. Existing scheduling algorithms that isolate I/O performance and provide differentiated service are limited to single-disk systems.

Recent I/O scheduling work at UTEP has developed a new I/O scheduling algorithm, called FAIRIO, which enables RAID storage systems to provide both performance isolation and differentiated service. Through detailed simulation, FAIRIO has been shown to provide isolated and differentiated service for idealized and real I/O workloads. When performance is tuned, the experienced disk-time utilization is within 4% (for idealized workloads) and 11% (for real workloads) of being perfectly proportional. Throughput is not degraded; in fact, it is marginally improved. Future work aims to demonstrate that FAIRIO can be adapted to provide proportional sharing, i.e., differentiated service, for a variety of resources.

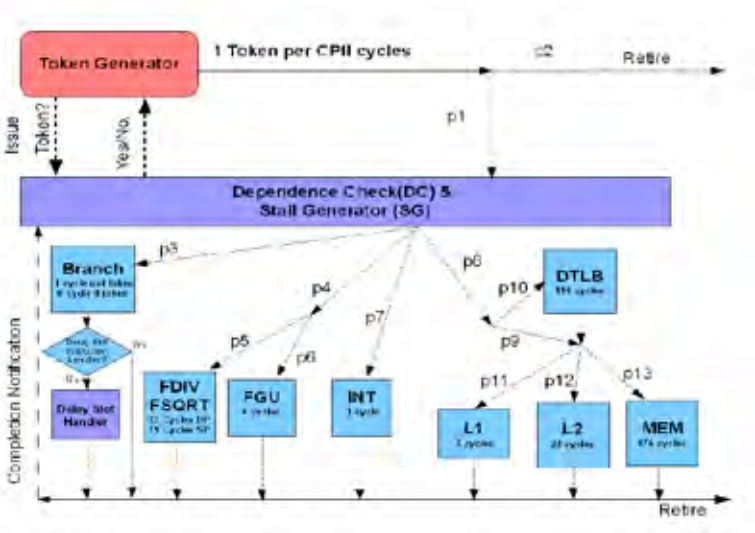
Modeling

System modeling not only enables fast and accurate performance prediction and analysis, it provides a means to define an architecture that will produce optimal performance. This information could be used to drive processor design, to make procurement decisions, and to define an FPGA architecture implementation. In addition, system modeling can be used to gain insights that can lead to the development of techniques that can enhance system performance.

Monte Carlo Modeling: In an effort to model, in a time-efficient manner, the performance of Army applications executed on next-generation systems, the group has adopted a Monte Carlo modeling methodology to model, predict, and analyze the performance of contemporary multi-core architectures: the Sun Niagara, the IBM Cell Broadband Engine, the Intel Itanium 2, and the Opteron processors. During 2009, the existing Monte Carlo methodology was enhanced at NMSU with a technique to model out-of-order instruction execution, and work began to enhance this methodology with power models. A modeling framework was implemented that enables users to develop Monte Carlo models of contemporary and future multicore architectures.

Performance characteristics predicted by the models are validated against the performance of their real-world counterparts. Validation of the Niagara 2 single-core model has been completed; all model predictions are now within 7% of measured values. After analyzing validation results, the Niagara 2 single-core model was adapted to include the latency load for load–load instruction sequences and data forwarding within the

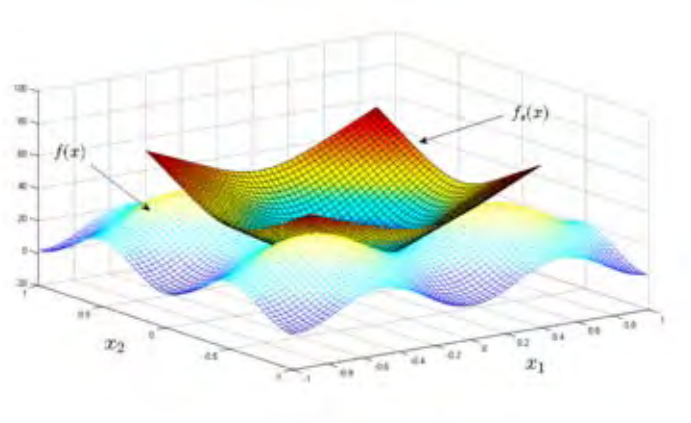
continued on page 20



Niagara 2 Monte Carlo modeling results.

Hybrid Optimization Schemes for Parameter Estimation Problems

Effective use of the recent innovations in computer architecture can be limited by difficulties in writing functionally correct parallel applications that also achieve high performance. Hybrid algorithms combine the advantages of more than one computing method to optimize the solution of mathematical problems with large numbers of parameters and many solutions that are "best" only for limited ranges of a particular parameter or set of parameters (local minima).

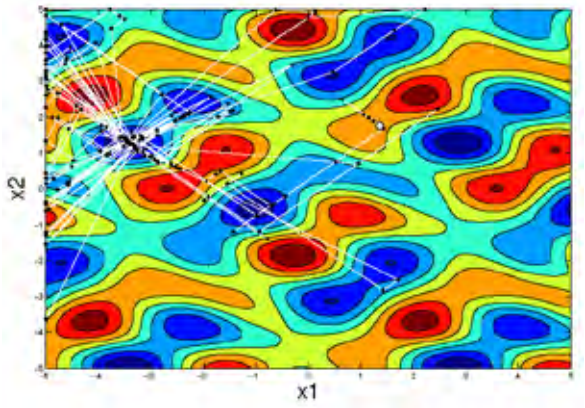


A Gaussian function (upper surface) serves as a surrogate for the more computationally expensive Rastringin function (bottom surface). Graphics courtesy of Leticia Velázquez, UTEP.

The University of Texas at El Paso (UTEP) associate professors Miguel Argáez and Leticia Velázquez (mathematical sciences) and computational science graduate students Miguel Hernandez IV, Carlos Ramirez, and Reinaldo Sánchez are developing mathematical and computational tools that facilitate the implementation of problem-solving applications on highly parallel systems. They are also demonstrating a practical migration path from current programming approaches to a transaction-based model.

Step by Step

The mathematical side of this project focuses on a hybrid algorithmic approach for solving general optimization problems, including automated parameter estimation problems *(see schematic, next page)*. In particular, efforts are focused on global optimization problems having many local minima—that is, finding a set of parameters that works best over the entire region of interest from a large group of locally viable candidates.

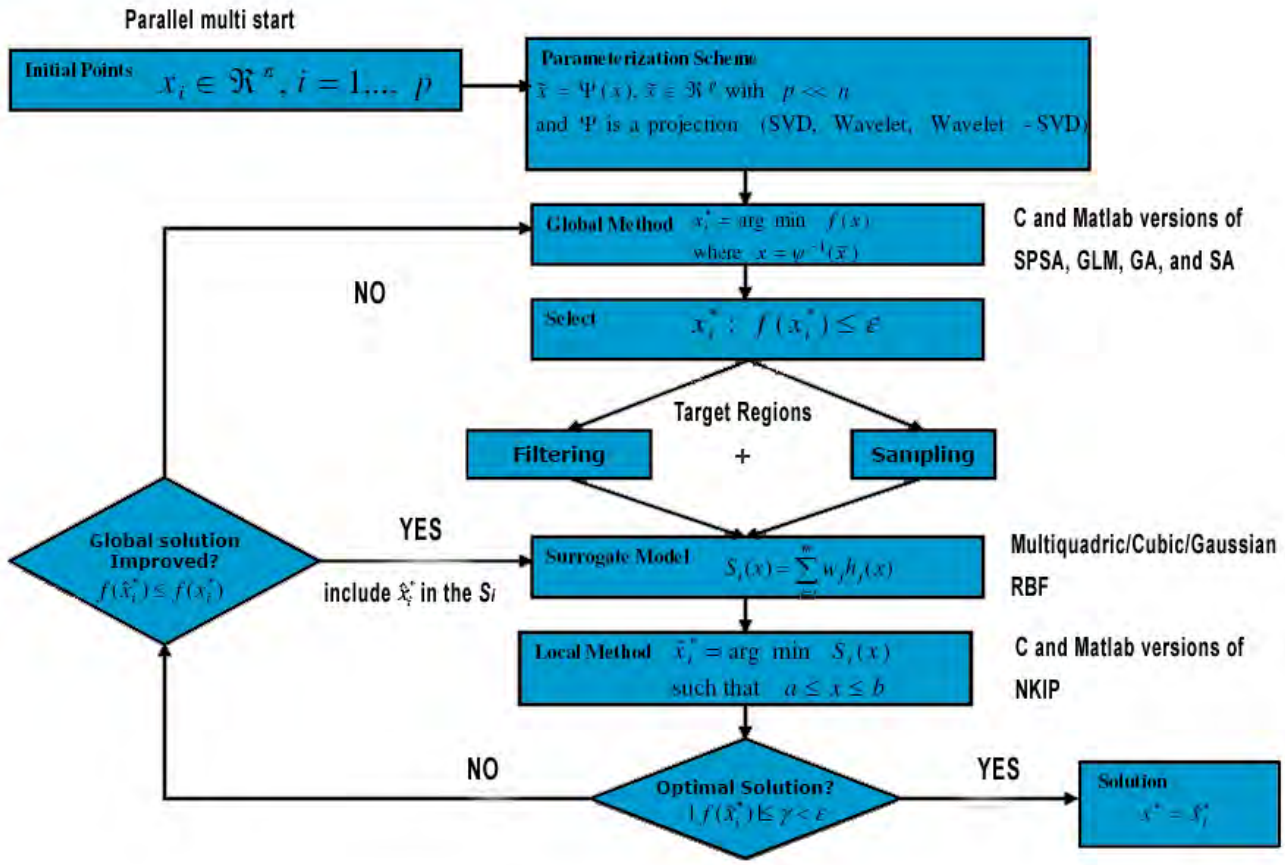


SPSA search across a parameter surface.

The UTEP team has developed a method that begins with one of several global stochastic techniques: Simultaneous Perturbation Stochastic Approximation (SPSA), Global Levenberg–Marquardt (GLM), Genetic Algorithms (GA), or Simulated Annealing (SA). These techniques are sampling methods that perform a global search of the parameter space *(illustration, below left)*. This search may start from multiple initial guesses (parallel multi-start). In many real applications, it is difficult or impossible to compute derivatives of the function being optimized (gradient directions in which the function is increasing or decreasing). Most of these global methods do not use derivatives, and thus do not require this information in order to work.

The stochastic search produces target regions where the global optimal solution may lie. A surrogate model is constructed by filtering data points from these regions. The surrogate model behaves in a mathematically similar fashion to the function of interest while being less demanding in terms of computational resources. In particular, multiquadric, cubic and Gaussian radial basis functions are selected to build the surrogate model.

The surrogate model is used to perform local searches, using the Newton–Krylov Interior Point (NKIP)



algorithm developed by the UTEP team. This method calculates gradients and evaluates whether a given gradient is steep enough to lead to a global minimum. Moreover, the algorithm allows the inclusion of physical bounds for obtaining feasible approximate solutions to the problem.

Making it Work

Implementing these algorithms for a parallel computing environment requires several novel approaches. For example, introducing transactions (individual small operations) as the key abstraction for expressing parallelism facilitates maintaining a computer system in a known, consistent state by ensuring that interdependent operations are either all completed successfully or all canceled successfully.

The UTEP group is developing a simple distributed-memory programming model that can scale to systems with thousands of processors. Their C version software framework (also available in Matlab) is being tested on two high performance machines: the Mana Dell Tera-Scale HPC system (Maui High Performance Computing Center, Air Force Research Laboratory) and the Lonestar Dell Linux cluster (Texas Advanced Computing Center, University of Texas).

The first version of a semigraphical ncurses-type software interface for the hybrid optimization algorithm has been completed. (ncurses is a library of functions written for Unix applications.) This interface enables the user to call the HPC hybrid optimization C code with a choice of global methods: SPSA, GLM, GA, or SA. It also enables the use of a surrogate model and an option to continue with NKIP. Documentation for this interface is in progress, along with a manual for the hybrid algorithm in general.

Applications

One area of particular interest is estimation theory—a branch of statistics that is often used to assist in inter-

continued on page 14

Hybrid Optimization Schemes

continued from page 13

preting the results of scientific experiments. Mathematical calculations use observable information as input to produce an approximation to the parameter of interest when an exact solution is not possible. Such techniques are especially useful for signal processing and telecommunications problems, as well as for scientific applications with irregular and adaptive behavior.

Hybrid optimization codes developed as a result of this work are being applied to Stanford's AERO-F computational fluid dynamics code. This code is used by AHPCRC researchers in Technical Area 1 to develop flapping and twisting wing models for micro-aerial vehicles, hummingbird-sized airborne vehicles that can be used for sensing and surveillance. Large-scale simulations using this code are planned for 2010.

The hybrid algorithm is also being implemented in the PyADH simulator's adaptive hydraulics modeling modules from the Engineer Research and Development Center, U.S. Army Corps of Engineers, for use on the Lonestar cluster. Large-scale applications are being prepared to run on this simulator in 2010.

Moving forward

Uncertainty issues and regularization techniques for the hybrid algorithm are currently being evaluated. Additional functionalities are being developed, for inclusion in the hybrid package. For example, users will have the option to approximate derivatives using random forward and backward differences. They can compare the results with the well-known difference approximations used in most algorithms. The goal of this effort is to reduce the number of function evaluations per iteration.

Recently, the UTEP team proposed a path-following fixed-point method for large-scale l_1 underdetermined problems and their applications in compressed sensing (a technique for acquiring and reconstructing signals). Also, the team has developed a projected conjugate gradient algorithm for solving overdetermined systems in l_q quasinorms. ✯

AHPCRC Tech Area 4 Principal Investigators

Tech Area Lead: Pat Hanrahan

Stream Programming for HPC

Alex Aiken
Computer Science
Stanford University
aiken@cs.stanford.edu
(650) 725-3359

William Dally
Willard R. and Inez Kerr
Bell Professor of Computer
Science, Chairman of
the Computer Science
Department
Stanford University
billd@csl.stanford.edu
(650) 725-8945

Patrick Hanrahan
Canon Professor in the School of
Engineering,
Stanford Computer Graphics
Laboratory
Stanford University
hanrahan@cs.stanford.edu
(650) 723-8530

Flexible Architecture Research Machine

Oyekunle Olukotun
Electrical Engineering and Computer Science,
Director, Pervasive Parallelism Laboratory
Stanford University
kunle@stanford.edu
(650) 725-3713

Christoforos Kozyrakis
Electrical Engineering and Computer Science
Stanford University
christos@ee.stanford.edu
(650) 725-3716

Simulation & Modeling to Enhance the Performance of Systems of Multicore Processors

Patricia Teller
Computer Science
The University of Texas at El Paso
pteller@utep.edu
(915) 747-5939

Sarala Arunagiri
Computer Science
The University of Texas at El Paso
sarunagiri@utep.edu
(915) 747-8819

Jeanine Cook
Klipsch School of Electrical and Computer Engineering
New Mexico State University
jecook@nmsu.edu
(575) 646-3153

Hybrid Optimization Schemes for Parameter Estimation Problems

Miguel Argáez
Mathematical Sciences
The University of Texas at El Paso
margaez@utep.edu
(915) 747-6775

Leticia Velázquez
Mathematical Sciences
The University of Texas at El Paso
leti@math.utep.edu
(915) 747-6768

Pre-Freshman Engineering Program at NMSU

Summer 2009 marked the 13th year of New Mexico State University's Pre-Freshman Engineering Program (PREP), an intensive mathematics-based pre-college summer program that provides educational enrichment for achieving local middle- and high-school students. Although PREP is open to everyone, the program focus is on female and minority populations traditionally underrepresented in science, technology, engineering, and mathematics.

AHPCRC is a key funding agency for this program, as a part of its educational outreach mandate, and NMSU's College of Engineering provides instructors, curriculum, and facilities. Students attend tuition-free (there are no other fees, and lunch is provided), and they may attend PREP for four years before entering college. More than 90% of PREP participants go on to pursue higher education.

This year, 163 students successfully completed the six-week program. 144 students received high school credits, and 19 students received 6 college credits and 2 high school credits. Participants came to the NMSU campus to study logic, algebraic structures, technical writing, engineering, computer science, and physics. Off-campus field trips provided an opportunity for hands-on learning.

PREP is administered through WERC: A Consortium for Environmental Education and Technology Development, as part of the Institute for Energy and the Environment for the NMSU College of Engineering. (WERC, originally the Waste-management, Education and Research Consortium, was renamed to reflect its broader mission and accomplishments.) This program recruits students from the school districts in Doña Ana County NM (Las Cruces, Gadsden, and Hatch). It prepares these students for careers in science, technology, engineering, and mathematics by stimulating students' interest in higher mathematics and science and providing problem-solving sessions to equip them with the necessary tools and the desire to complete pre-calculus and calculus during high school.

Friday field trips and Career Awareness Seminars provide the students with opportunities to meet and interact with professionals, who instill in the students the vision and passion to become the scientific leaders of tomorrow.

This year marked the inception of PREP 4, a pilot test program for fourth-year PREP participants, who were awarded six college credits upon successful completion of the program. This expansion of the program came about in response to requests from the PREP students, and 19 students completed PREP 4. Students were accepted to PREP 4 based on successful completion of the first three years of PREP with at least an 85% average, a high school GPA of 90%, and not yet having graduated from high school.

The PREP 4 curriculum, *Special Topics for Engineering IV and Mathematics for Technicians,* included an introduction to the various disciplines in engineering and technology, experimental methods, technical reports, and presentation methods. These topics, along with teamwork strategies, ethics, globalization, and lifelong learning concepts, were discussed and applied in project-oriented structured design processes to solve a variety of engineering and technical problems. Students were made aware of a variety of relevant professional organizations, and they were introduced to many of the available campus resources, including student organizations, Student Employment Services, Financial Aid, Cooperative Education, libraries, and the offices and resources offered by NMSU Engineer-

ing and the Dona Ana Community College (DACC) Technical & Industrial Studies Department.

As a part of the enrichment and partnership with the U.S. Army High Performance Computing Laboratory, researchers Dr. Jing He and Dr. Hong Huang conducted classes in the Visual C++ programming language, analyzing Internet connection and performance using Visualroute, diagnosing connectivity problems, and analyzing network protocols using Wireshark.

Dr. Erica Voges recently assumed the directorship of the PREP program. Karen Mikel, program manager during the 2009 summer session, stated, "Through AHPCRC's commitment to excellence and generous financial support over the next few years, PREP will continue to grow." ✯

All the PREP faculty and staff attended a workshop conducted by Dr. Robert Panoff, founder of the Shodor Educational Foundation, Inc. All photos courtesy Karen Mikel, NMSU.

continued on page 18

Hands-On Learning Highlights

Through the support of AHPCRC, 2009 PREP students got hands-on experience in science and technology:

- PREP 1 students visited the International Space Museum (Alamogordo, NM), which focuses on educating visitors from around the world on the history, science, and technology of space. During their visit, the students observed NASA technology and multiple rocket launches.
- On the campus of NMSU, students viewed the large wind-tunnel that is used for research by the mechanical and aeronautical engineering departments, bio-fueled concept vehicles, and the water laboratory.
- PREP 1 & 4 students received briefings on the Unmanned Aerial Systems Technical Analysis and Applications Center, designed to promote safe integration of the unmanned systems in the National Airspace System.
- PREP 2 & 4 students toured and received briefings at Holloman Air Force Base (near Tularosa, NM). While there, the students worked with the Explosive Ordinance Devices Division, the High Speed Track, the T-38 Aircraft Training Facility, and Heritage Park.
- PREP 3 students visited the Army Research Laboratory at White Sands Missile Range. During the visit, students received presentations from the Survivability/Lethality Analysis Directorate, Computational and Information Science Directorate, and the Electromagnetic Vulnerability Assessment Facility. The group performed hands-on activities that included learning the importance of meteorology for battlefield conditions while participating in demonstrations of battlefield communications, and they had full hands-on access to a fully equipped High Mobility Multipurpose Wheeled Vehicle
- PREP 4 students visited the NASA Johnson Space Center White Sands Test Facility for approach and landing training for astronauts.
- PREP 4 students toured the White Sands Test Facility new state-of-the-art Range Launch Complex control room and control tower
- PREP 1, 2, and 3 students interacted with guest lecturers during the Career Awareness component for PREP. The lecturers included: Mr. Ed Creegan, ARL Military Battlefield Engineering division; NMSU physics professor Stephen Kanim; NMSU Electrical Engineering department head Paul Furth; and NMSU Civil Engineering department head Ricardo Jacquez.
- PREP 1, 2, 3 and 4 students worked with computers to learn about basic hardware and software components, development of algorithms through flowcharts, BASIC programming, Visual C++, Web Design, Microsoft Office, MatLab, and other topics.
- PREP 1, 2, and 3 students designed, built, and launched multiple rockets, from single-stage to multi-stage, with recovery packages.

The PREP students learned about solar panels and other renewable energy sources.

NMSU PREP Program

continued from page 17

Projects and field trips from the inaugural season of PREP 4

PREP 4 students learned about "suiting up" and manipulating the explosives robot at the Holloman AFB Explosive Ordnance Devices Division.

Project 1: Test Engineer

Groups of students tested Estes Rocket engines, as part of periodic test plan to ensure nominal performance. The students set up a force transducer and a Pasco hand-held controller/data acquisition unit to measure force, in Newtons, generated by the rocket engines. Ambient test conditions were measured. Students provided a summary of their rocket engine performance and compared this to historic nominal test data provided by the customer. Students plotted engine performance in Excel and Matlab as part of their report.

Project 2: Boe-Bot Challenge

This project covered mechanical engineering, electrical engineering, and programming. Students learned what a continuous servo-motor is and how to zero and control the servo-motor using programming scripts in PBasic. The students then assembled their wheeled robot, called the boe-bot. Using servo-control scripts, they wrote the program that allowed the robot to navigate a simple maze and return to the start point. The second challenge was to breadboard mechanical switches that would be used to sense an obstacle based on switching logic. The students used IF/THEN-type logic programs statements to determine if a switch of the 2-switch configuration was closed, and if so, which one, and then jump into a subroutine to perform a collision avoidance maneuver. The robot was instructed to locate a specific stopping point on the floor following the second collision avoidance maneuver. Challenge 3 was to breadboard a light-sensing voltage divider circuit that was used to determine the logic needed to follow a black line on the floor. No remote control was used in any of the challenges. Each student wrote a report on his or her experience with a statement about how they could outfit a boe-bot to operate on the Moon.

Project 3: Composite Fabrication and Testing (Materials – Chemical, Mechanical Engineering)

Students created foam-core composites representing their "company's" offerings. These composites were made of 2 × 12 inch DOW Corning blue foam board cores, with either carbon fibers or glass fibers on their surfaces. A two-part epoxy was used as the matrix. The students demonstrated the importance of proper mixing of the matrix, proper wetting of the fibers during layup, proper orientation of unidirectional fibers, and a vacuum bagging method for applying the necessary pressure on a non-flat object so that the composite follows curved contours as it cures. After the composites were cured, the students selected the three strongest samples for further testing to determine the Young's Modulus and the strength to weight ratio for the composite. These properties were compared to aluminum, steel, and titanium.

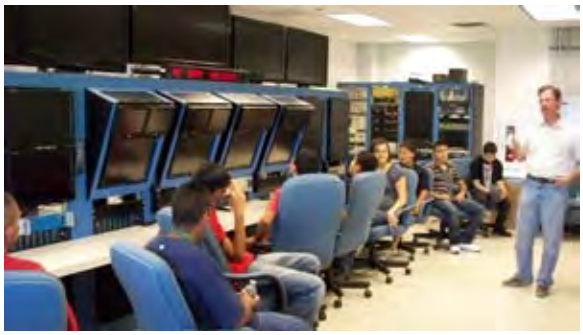
Project 4: Airfoil Modeling

Students were instructed in the use of FoilSim to design an airfoil. They plotted lift vs. angle of attack and air speed for their designs. The students were then instructed in the use of Inventor to create a virtual solid model of their airfoil, using the scaled geometry values provided by FoilSim. The airfoil model was printed on the 3D printer and tested in the JetStream 500 Wind-tunnel. The students then compared experimental results to the results predicted by the model, and they explained any discrepancies. The students were instructed to determine the sources of uncertainty in their experiment.

Project 5: Kelvin Bridge Project
The students built a truss bridge from 1/8" × 1/8" balsa sticks and glue based on the Kelvin Bridge #2 design (named for the educational supply company that provided the parts and design). Students were given instructions on how to use a truss simulation program located on the Johns Hopkins University website, to predict the bridge results. Students were shown how to create free body diagrams for the bridge to look at the reaction forces. The bridges were then load tested to see if their design would hold a 17-pound steel weight.

PREP 4 students at White Sands Missile Range.

Project 6: Glider Modeling
Students learned to use a glider modeling program to design an optimized glider, which could be cut out using the classroom laser cutter. Students were instructed in the use of the Aery software program to create their gliders, and they learned how to print their designs and create the AutoCAD drawing necessary to cut their glider wings using the laser cutter. Because of time limitations, this was presented as a demonstration of classroom technology for use in an engineering class as well as a "fun project." The end result was an actual balsa wood glider made from the customized design.



PREP 4 students toured and received instruction on the new state-of-the-art range launch complex control room and control tower at White Sands Missile Range.

Project 7: Rube Goldberg Device
The object of the Rube Goldberg project (right), modeled after this coming school year's competition, was to build a device to squeeze hand sanitizer on a hand using a minimum of 5 steps.

PREP 4 students visited the ARL White Sands Test Facility HOT/COLD room.

Systems of Multicore Processors

continued from page 11

memory and floating-point pipelines. The Niagara 2 multi-core model has been completed, and validation data has been collected. Data have been collected for the initial multithreaded Niagara 2 model.

The initial Monte Carlo Opteron model has been completed. The methodology to implement out-of-order execution is done. The model predicts very accurately, and full validation is in progress. The initial design of the methodology for the Opteron multi-core model has also been completed. The researchers are actively integrating the existing Opteron and the Niagara 2 models into the SST (Structural Simulation Toolkit) exascale system simulator at Sandia National Laboratories. The SST was released under gnu license in 2009.

Power modeling tools and techniques available for emerging architectures are another area of study, with a special focus on architectures on Chimera. Methods used for indirectly measuring CPU power using performance counters and contemporary methods for measuring GPU power are being investigated. At present, power for GPUs and FPGAs is often estimated, but for many applications, these estimates are known to be inaccurate. Direct power measurements are being studied, along with methods for validating these measurements. A report is available on the user wiki.

Modeling for Fault Tolerance: Checkpoint/restart is a common technique that provides fault tolerance for applications executing on massively parallel processing systems. Checkpointing reduces the amount of time and effort wasted when a long software process is interrupted by a hardware or software failure. Checkpoints store data to persistent media such as a file system to enable a process to be restarted from the latest checkpoint rather than starting again at the beginning. The time interval between checkpoints must balance two competing priorities: frequent checkpoints minimize computational losses in the event of a failure, but too many checkpoints can significantly slow the execution of the program.

Mathematical modeling has been used at UTEP to gain insights into the performance impact, in terms of both application execution time and I/O system service, of the choice of time between checkpoints (checkpoint interval length). Through this modeling, the AHPCRC researchers at UTEP have found that, in addition to having a significant impact on the execution time of applications that perform periodic checkpointing, the choice can significantly affect the number of checkpoint I/O operations performed during the application's execution and, thus, its demand on the I/O bandwidth of the computer system.

Existing models determine the checkpoint interval that minimizes wall-clock execution time of an application. UTEP researchers have developed another model that identifies a checkpoint interval that minimizes the aggregate number of checkpoint I/O operations. The UTEP group illustrated the existence of such propitious checkpoint intervals using parameters of four massively parallel processing systems: Red Storm, Jaguar, Blue Gene/L and a theoretical PetaFLOPs system. Using both of these models provides application programmers with a basis for finding a checkpoint interval that balances application execution time and the frequency at which an application performs checkpoint operations. Future work will investigate the use of these models to schedule the checkpoint I/O, called defensive I/O, and productive I/O of multiple concurrently-executing applications.

Applications

Chimera is being used to study applications of interest to the Army in a heterogeneous parallel computing environment. An initial performance analysis has been completed of the Stereo Matching Code from ARL, and the acceleration of computer vision algorithms using GPUs is being explored. Related research focuses on helping computer vision software developers accelerate their algorithms without having to learn the details of how parallel architectures work. Backprojection, or tracing a detected photon back to its source, is another area of interest to Army image processing researchers. Several backprojection algorithms and methods are under investigation. ✯

Publications and Presentations

AHPCRC Publications and Presentations July 2009–February 2010

A complete list of publications and presentations is available at http://www.ahpcrc.org/publications.html

Project 1–1: Ballistic Impact and Optimization of Composite Shields

- High-speed impact with electromagnetically sensitive fabric and induced projectile spin. Zohdi, T. I. *Computational Mechanics,* accepted for publication 2009.
- Modeling and simulation of multiphysical processes in particulate media. Zohdi, T. University of Southern California, Department of Civil Engineering (invited lecture, October 2009).
- Modeling and simulation of multiphysical processes in particulate media: electromagnetic sprays and solids. Zohdi, T. Workshop on Mesoscale Mechanics of Complex Materials. Vancouver, Canada (invited lecture, November 2009)
- Modeling and simulation of multiphysical processes in particulate media: electromagnetic sprays and solids. Zohdi, T. Workshop on Mesoscale Mechanics of Complex Materials. Lawrence Berkeley National Labs (invited lecture, December 2009).

Project 1–4: Flapping and Twisting Aeroelastic Wings for Propulsion

- Global Model Reduction for Fluid-Structure Interaction in Flapping Flexible Wings. M. Wei and T. Yang. *Bulletin of the American Physical Society, 54*(19), 62nd Annual Meeting of the APS Division of Fluid Dynamics, Minneapolis, MN, November, 2009.
- Aerodynamics of a singledegree- of-freedom toy ornithopter. R. Chavez Alarcón, B. J. Balakumar, J. J. Allen. *Bulletin of the American Physical Society, 54*(19), 62nd Annual Meeting of the APS Division of Fluid Dynamics, Minneapolis, MN, November, 2009.
- Optimization Study for Hovering Flapping Flight. H. Bocanegra Evans , J. J. Allen, and B. J. Balakumar. *Bulletin of the American Physical Society, 54*(19), 62nd Annual Meeting of the APS Division of Fluid Dynamics, Minneapolis, MN, November, 2009.

Project 2–1: Dispersion of BWAs in Attack Zones

- A new mass, energy, vorticity, and potential enstrophy conserving scheme for complex boundaries in 3D nonhydrostatic stretched or nested grid models. Ketefian, G.S., Jacobson, M.Z. American Geophysical Union, Fall Meeting, San Francisco, California, December 14–18, 2009.
- A piecewise-linear boundary scheme for the shallow water equations that conserves mass, energy, vorticity, and potential enstrophy. Ketefian, G.S., Jacobson, M.Z. *J. Comp. Phys.,* in review, 2010.
- A numerical study of scalar dispersion downstream of a wall-mounted cube using direct simulations and algebraic flux models. Rossi, R. Phillips, D., Iaccarino, G. Submitted to *International Journal of Heat and Fluid Flows,* 2010.
- A piecewise-linear boundary scheme for the shallow water equations that conserves mass, energy, vorticity, and potential enstrophy. G.S. Ketefian, M.Z. Jacobson, *Journal of Computational Physics,* in review, 2009.
- The global-through-urban nested 3-D simulation of air pollution with a 13,600-reaction photochemical mechanism. Jacobson, M.Z., Ginnebaugh, D. L. *J. Geophys. Res.,* in press, 2010. (preprint available at www.stanford.edu/group/efmh/jacobson/MCM0909.pdf)
- Numerical simulation of scalar dispersion downstream of a square obstacle using gradient-transport type models. R. Rossi, G. Iaccarino. *Atmospheric Environment, 43*(16), 2518–2531, 2009. doi:10.1016/j.atmosenv.2009.02.044

Project 2–2: Micro- and Nano-fluidic Devices for Sorting and Sensing BWAs

- Simulation of Brownian, drug delivery particles in microchannel flows. Saibaba, A., Shaqfeh, E.S.G., Darve, E. Oral Presentation, 62nd Annual American Physical Society Division of Fluid Dynamics Meeting, Minneapolis, MN, Nov. 22–24, 2009.
- Numerical simulation of platelet margination in microcirculation. Zhao, H., Shaqfeh, E.S.G., Darve, E. Oral Presentation, 62nd Annual American Physical Society Division of Fluid Dynamics Meeting, Minneapolis, MN, Nov. 22–24, 2009.
- The Microfluidics of NonSpherical Colloidal Particles and Vesicles with Application to Blood Additives. Shaqfeh, E.S.G. Invited presentation at Small-Scale Hydrodynamics: Microfluidics and Thin Films, Banff International Research Station Workshop, February 7–12, 2009 (Banff, Alberta, Canada).
- On the Taylor Dispersion and Heterogeneous Wall Binding of Particle Flow through Channels of Arbitrary Cross Section and Reaction Profile. Fitzgibbon, S., Shaqfeh, E.S.G. *J. Fluid Mechanics,* submitted February 2010.
- The Microfluidics of Nonspherical Colloidal Particles and Vesicles with Application to Blood Additives. E.S.G. Shaqfeh, H. Zhao, A. Saibaba, E.Darve. AVS Inkjet Technology Topical Conference Sponsorship Form, San Jose, CA, November 8–13, 2009.
- The Microfluidics of Colloidal Particle–Vesicle–Capsule Mixtures with Application to Blood Additives. E.S.G. Shaqfeh, H. Zhao, A. Saibaba, E.Darve. IMA Microfluidics: Electrokinetic and Interfacial Phenomena Workshop, Minneapolis, MN, Dec. 7–11, 2009.

Distribution Statement A: Approved for public release; distribution is unlimited.

continued on page 22

Publications and Presentations

continued from page 21

Project 2–4 (was Project 2–3 for Years 1 and 2): Protein Structure Prediction for Virus Particles

- Effect of side chain anisotropy on residue contact determination. Sun, W., He, J. *Proceedings of the 2009 IEEE International Conference in Bioinformatics and Biomedicine Workshop,* pp. 181–188, Washington DC, November 2009.
- Structure prediction for the helical skeletons detected from the low resolution protein density map. Al Nasr, K., Sun, W., He., J. *BMC Bioinformatics 11* (Suppl 1) S44, 2010. Also submitted to Asia Pacific Bioinformatics Conference, India, January 2010.

Project 3–1: Information Aggregation and Diffusion in Networks

- Sensing Mobile Objects and Applications. Guibas, L. Presentation at the NTT Laboratories in Kyoto, Japan.

Project 4–3: Simulation & Modeling to Enhance the Performance of Systems of Multicore Processors

- HEC I/O Based on Judicious Checkpointing and I/O QoS. Teller, P. Invited lecture, Laboratory for Computer Systems seminar series. Oak Ridge National Laboratories,October 20, 2009.
- Improving the Throughput of Simultaneous Multithreaded (SMT) Processors using Application Signatures and Hardware Thread Priorities. Mitesh Meswani, doctoral dissertation, UTEP, defended December 4, 2009.
- Extending the Monte Carlo Processor Modeling Technique: A Statistical Performance Model of the Niagara 2 Processor. Submitted to ISPASS 2010 (IEEE International Symposium on Performance Analysis of Systems and Software).

Project 4–4: High Performance Optimization Library

- Stochastic Binormalization of Symmetric Matrices. A. Bradley, W. Murray. SIAM Conference on Applied Linear Algebra, Monterey Bay–Seaside CA, October 2009.
- Optimality principles in nonequilibrium biochemical networks. R. M. T. Fleming, C. Maes, M. A. Saunders, Y. Ye, and B. O. Palsson. Physical Review Letters, , undergoing revisions prior to publication. (preprint available at www.stanford.edu/~yyye/entropyFBA4.pdf).
- An active-set convex QP solver based on regularized KKT systems. C. Maes and M. A. Saunders. Presented at BIRS Workshop 09w5101, Advances and Perspectives on Numerical Methods for Saddle Point Problems, Banff, Alberta, Canada, April 12–17, 2009. http://www.stanford.edu/group/SOL/talks.html. Also presented at SIAM Conference on Applied Linear Algebra, Monterey Bay–Seaside CA, October 2009.

  also: An Active-Set Convex QP Solver Based on Regularized KKT Systems. C.-M. Fransson, T. Wik, B. Lennartson, M. A. Saunders, P.-O. Gutman. *XIEEE Trans. Contr. Sys. Tech. 17* (2), 298–308, 2009.

Project 4–6: Hybrid Optimization Schemes for Parameter Estimation Problems

- A Path Following Method for Large-Scale l_1 underdetermined problems. Poster, The International Conference for High Performance Computing (SC09), Portland, OR, November 20th 2009. Also submitted to *IEEE Transactions on Signal Processing.*
- Ramirez, C., Sanchez, R. Two talks at the 6th Joint UTEP/NMSU Workshop on Mathematics, Computer Science and Computational Sciences, University of Texas at El Paso. El Paso, Texas, November 7th, 2009
- A Hybrid Algorithm for Global Optimization. Accepted for Special SCAN'08 issue of *Reliable Computing.* Also presented at the Multiphysics Conference, Lille, France, December 2009.
- Solving Overdetermined Systems in l_q Quasinorms. Accepted for Special SCAN'08 issue of *Reliable Computing.*
- Hybrid optimization schemes for parameter estimation problems. M. Argáez, L. Velazquez, C. Quintero, F. Zapata. XVII Mathematical Congress, Cali, Colombia, August 2–6, 2009.
- A projected conjugate gradient algorithm for KKT systems. M. Argáez. XVII Mathematical Congress, Cali, Colombia, August 2–6, 2009.
- The Notion of the Quasicentral Path in Linear Programming. M. Argáez, C. Ramirez, L. Velazquez, O. Mendez. XVII Mathematical Congress, Cali, Colombia, August 2–6, 2009.
- A Path Following Method for Large-Scale and Dense–Underdetermined Problems. M. Argáez, C. Ramirez, R. Sanchez, C. Quintero. XVII Mathematical Congress, Cali, Colombia, August 2–6, 2009.
- Comparison of Global Parameterization Schemes for Parameter Estimation Problems. L. Velazquez (presenter), M. Argáez, C. Quintero, C. Ramirez, R. Sanchez. International Symposium in Mathematical Programming, Chicago, IL, August 18–24, 2009.
- The Notion of the Quasicentral Path in Linear Programming. M. Argáez, O. Mendez, L. Velazquez. Submitted to SIAM Journal in Optimization.

Supplemental Task 7: Multiscale Reactive Modeling Institute Support

- Progress report on fitting the Reax force field to a DoD community of CCM users interested in reactive potentials. A. Yau. PET workshop, Memphis, TN, August 6, 2009..

Publications and Presentations

AHPCRC Undergraduate Summer Institute in Computational Science and Engineering (Project title / Student(s) / Mentor(s))

- Flutter prediction for the F-16 Block 40 / Alex Sabatini (Stanford) / David Amsallem, Charbel Farhat
- Micro Air Vehicle Modeling / Ricardo Medina (NMSU), / Charbel Bou-Mosleh, Charbel Farhat
- The Aerodynamic Analysis of a Damaged Wing / Samir Patel (Harvard) / Charbel Bou-Mosleh, Charbel Farhat
- Higher Order Scattering on Submerged Objects / Kalesanmi Kalesanwo (Morgan State) / Paolo Massimi, Charbel Farhat
- Modeling Differing Structural Fabric Designs for Use in Ballistic Shields / Brandon Moultrie (Morgan State), Caraline Murphy (NMSU) / David Powell, Charbel Farhat
- Sparse Matrix Solvers for Multi-Core and Parallel Platforms / Emilio Lopez and Andres Morales (Stanford) / Cris Cecka, Eric Darve
- Automated Calibration of Camera Networks / Daniel Shaffer (Stanford) / Branislav Kusy, Martin Wicke, Leonidas Guibas
- Full Cache Coherency on an FPGA-based Accelerator / Kevin Thompson (NMSU) / Sungpack Hong, Kunle Olukotun
- Mesh Visualization Tool / Richard Gutierrez (NMSU), Edgar Padilla and Essau Ramirez (UTEP) / Zach Devito, Pat Hanrahan, Eric Darve
- Characterization of High-Strength Nano-scale Gold and Aluminum / Michael Hammersley (Stanford) / Chris Weinberger, Sylvie Aubry, Wei Cai
- Thermal Conductivity of GaN Nanowires / Abraham Chukwuka (Morgan State) / Sylvie Aubry, Wei Cai
- Highly Anisotropic Iron in Fusion and Nuclear Power Plants / Stacey Oriaifo (Morgan State) / Sylvie Aubry, Wei Cai

Summer undergraduate program: Introduction to Computational Methods, Morgan State University.
J. Nithianandam, L. Walker, G. Wilkins, organizers
Student presentations:

- Solution of a Differential Equation Using the Finite-Difference Method / Donzell Dunston and Izaiah Wallace
- Parallel Computing: Compilation of C Programs and IMSL in Eclipse / Gary Francis
- Networking Linux Computers for High-Performance Parallel Computing / Rekab Ogunbiyi
- Creation of a Wiki for AHPCRC Documentation / Eric Mfomen
- Configuration of a Computer Cluster for Simulating Complex Antenna Systems / Drew Branch
- Open Message Passage Interface (MPI) and its Use in Parallel Programming / Chandroutie Sankar
- Hardware/Software Setup for Parallel Computing Using Eclipse and PI / Nehemya Cohen
- The Simulation of a Thin-Film Resistor in HFSS as a Means of Understanding and Evaluating the Software / Jan-Paul Alleyne
- Parallelization of Boundary Value Problems Solvers Using the Iterative Substructuring Method / Victor Epee
- Parallelization of Electromagnetic Solvers Using Bivariant Gaussian Elimination (BGE) / Chukwuemeka Obiaka
- Parallel Programming in FORTRAN: Numerical Matrix Analysis and Compilation for a Traditional Language / Chukwuemeka Igwilo
- Training and Evaluation of Mesh Generation Software Using Basic Electromagnetic Structures / Ezekiel Maina

⋆ ⋆ ⋆

AHPCRC Consortium Members

Stanford University
High Performance Technologies, Inc.
Morgan State University
New Mexico State University at Las Cruces
University of Texas at El Paso
NASA Ames Research Center

Distribution Statement A: Approved for public release; distribution is unlimited.

c/o High Performance Technologies, Inc.
11955 Freedom Drive, Suite 1100
Reston, VA 20190-5673
ahpcrc@hpti.com
703-682-5368
http://www.ahpcrc.org

Note:
The AHPCRC website address is now www.ahpcrc.org. Your current bookmark for http://me.stanford.edu/research/centers/ahpcrc will also work, no changes necessary.

INSIDE THIS ISSUE

This work was made possible through funding provided by the U.S. Army Research Laboratory (ARL) through High Performance Technologies, Inc. (HPTi) under contract No. W911NF-07-2-0027, and by computer and software support provided by the ARL DSRC.