Performance Engineering Research Institute
SciDAC-2 Enabling Technologies Institute:
Final Report for the University of North Carolina
DE-FC02-06ER25764

Robert Fowler

30 June 2014

I. Introduction

This document is the final report for cooperative agreement DE-FC02-06ER25764, the University of North Carolina component of Performance Engineering Research Institute (PERI), an Enabling Technologies Institute of the Scientific Discovery through Advanced Computing (SciDAC-2) program of the Department of Energy's Office of Science (DOE SC) Advanced Scientific Computing Research (ASCR) program.

Enhancing the performance of SciDAC applications on petascale systems had high priority within DOE SC at the start of the second phase of the SciDAC program, SciDAC-2, and it continues to do so today. Achieving expected levels of performance on high-end computing (HEC) systems is growing ever more challenging due to enormous scale, increasing architectural complexity, and increasing application complexity. To address these challenges, PERI implemented a unified, tripartite research plan encompassing: (1) performance modeling and prediction; (2) automatic performance tuning; and (3) performance engineering of high profile applications. The PERI performance modeling and prediction activity developed and refined performance models, significantly reducing the cost of collecting the data upon which the models are based, and increasing model fidelity, speed and generality. PERI's primary research activity was automatic tuning (autotuning) of scientific software. This activity was spurred by the strong user preference for automatic tools and was based on previous successful activities such as ATLAS, which automatically tuned components of the LAPACK linear algebra library, and other recent work on autotuning domain-specific libraries. Our third major component was application engagement, to which we devoted approximately 30% of our effort to work directly with SciDAC-2 applications. This last activity not only helped DOE scientists meet their near-term performance goals, but also helped keep PERI research focused on the real challenges facing DOE computational scientists as they entered the petascale era.

In the body of this document, we report on the technical activities conducted by researchers at the University of North Carolina.

Research Thrusts.

In this section we summarize the research activities and results under PERI by UNC researchers.

Management of Performance Data

As a part of PERI efforts, the PERI performance database was joint project among PERI researchers and outside collaborators at the University of Oregon, Portland State University, and Texas A&M University. The database served as a repository for storing performance data and environmental metadata, including compilation parameters and run time conditions collected by performance tools such as TAU, PerfTrack, Prophesy and SvPablo. It provided a common web interface for users to browse performance data collected by specific tools and performance data analysis with the tool. This is living on as the TAU performance database.

UNC researchers at RENCI worked on interoperability issues, including the database XML scheme and data dictionary. Using the Lattice QCD code MILC-7.4.0 and the SciDAC LQCD libraries (QMP, QIO, QDP, QLA, and QOPQDP) information for the database was collected on the NERSC Opteron cluster, the Cray XT3 systems at ORNL, the IBM BlueGene/L (BGL) at ANL, and the BGL system at RENCI. Data was collected using a variety of sources including TAU, PerfTrack, and Prophesy. Some of the results appear ed on a poster at SciDAC2007 meeting.

The PERI performance database was hosted at the University of North Carolina throughout the duration of the project.

Methods and Tools for Scalable Performance Diagnosis

It is possible to keep the local cost of performance measurement very low on small, single node systems by using event-based sampling of hardware performance counters using tools such as HPCToolkit and Oprofile. However, for large parallel systems, collecting and analyzing performance data at a central location is not economical due to the communication costs incurred, either explicitly or within an I/O system. PERI researchers at UNC therefore led efforts to develop methods and tools for scalable performance analysis.

The first effort was the development and extension AMPL (Adaptive Monitoring and Profiling Library) to use stratified sampling techniques to limit the number of nodes where performance data must be captured and analyzed.

During FY2007, we implemented a portable version of AMPL with a focus its application to the IBM BlueGene architecture. This version of AMPL wasintegrated with the TAU toolkit from the University of Oregon, and it could be launched from Eclipse/PTP. We published a paper “Scalable Methods for Monitoring and Detecting Behavioral Classes in Scientific Codes” (T. Gamblin, R. Fowler, and D. Reed). In this paper, we assessed the scalability of AMPL as run on the two rack (2K nodes, 4K processors) BlueGene/L at RENCI. The benchmark codes included the ASCI sPPM benchmark, ADCIRC, and a Chombo (LBL and UCB) test program.

Extreme scale systems have hundreds of thousands of processor cores, so their applications must be massively parallel. Effective use of these systems requires efficient interprocess communication through memory hierarchies and complex network topologies. Tools to collect and analyze detailed data about this communication are needed to facilitate its optimization. However, several factors complicate tool design. First, the performance problems unique to extreme scale systems are those that emerge only when moving to this scale; otherwise, one could debug and tune on smaller, more manageable systems. Second, large-scale runs on large systems will be a precious commodity; dedicated performance runs at scale will be rare, so scalable tools must have low enough overhead that they can be used routinely in production. Third, the volume of performance data from extreme systems stress communication and storage resources. Finally, analyzing this volume of data could require a large parallel computer itself.

In response to these problems, in FY08 research and development of scalable tools became the primary thrust of PERI research at RENCI. The focus at RENCI was primarily on the use of statistical clustering methods to reduce overhead and data volume [Gamblin2008b]. Under PERI funding we joined forces with researchers at LLNL by taking a common approach and using a common infrastructure [deSupinski2008a,Gamblin2008c]. The focus was to identify the nature of application load imbalances and to capture detailed load balance statistics in a scalable fashion. The data reduction strategy is based on using wavelet transforms.

In some longer-range modeling research work, RENCI and LLNL researchers developed a framework for scalable performance measurement and analysis based on wavelet transformations and LLNL's scalable MPI tracing mechanism. A tool called Libra was been implemented to capture detailed load balance statistics and to identify the nature and sources of application load imbalances. The mechanism provides an ability to understand how load differences evolve across all tasks and across different code regions over a series of application time steps [Gamblin2008a; Gramblin2008b]. This work is a major part of Gamblin's Ph.D. dissertation [Gamblin2009].

Researchers at UNC/RENCI, in collaboration with researchers at LLNL, applied Libre and other tools to address performance issues related to systems with 10,000 to 100,000+ nodes [deSupinski2008a; deSupinski2008b; Gamblin2008a; Gamblin2008b; Porterfield2008b].

Libre interacts with LLNL's scalable MPI tracing mechanism. It captures detailed load balance statistics and to identify the nature and sources of application load imbalances. In the past year, we have ported Libra to a variety of platforms including X86_64/Infiniband Linux clusters, Cray XT5s (Jaguar and Kraken at ORNL), and to the NSF TACC Ranger system at the University of Texas. A consequence of this effort has been the ability to do cross-platform load imbalance studies. The effort has also simplified and rationalized the structure of Libra and it's build process. The software was made available to users of the Ranger system at the University of Texas. The effort has also simplified and rationalized the structure of Libra and it's build process. The software is now available at http://github.com/tgamblin/libra.
The scalable tools work has continued with work on the Muster scalable clustering library. This work is described in [Gamblin et al, SC2010] and the software is available at http://github.com/tgamblin/muster.
As a synergistic activity under NSF funding, RENCI did initial work to deploy these technologies to the IBM version of the NSF Blue Waters system, originally planned to go on-line in 2011. We this synergistic work when the IBM system was canceled.

Performance Tools for Modern Programming Practices.

One of the keys to achieving acceptable performance on recent generations of processors is inlining small routines and code fragments and then aggressively applying compiler optimization. This is especially important in C++ codes. Modern programming styles in C++ are productive, given the power of abstraction, but the resulting codes involve "towers" of template methods with code fragments inlined within code fragments. As a consequence, it is difficult attribute execution costs to program constructs using conventional performance tools that rely on inserting instrumentation at the source code level. While it would also be possible to insert instrumentation in executable binaries, the highly optimized object code has been transformed to bear little resemblance to the original source, thus making attribution of performance measures to source code constructs meaningles. Without such attribution, one cannot readily optimize the code. The number of C++ codes and libraries is increasiong. Codes of interest to the Office of Science that we used to drive this work include the LQCD codes QDP++ and Chroma, Chombo (SciDAC), Rosetta (INCITE), and MADNESS.

Of particular interest, and difficulty, is the Chroma lattice QCD code (SciDAC and INCITE), which uses templates extensively, including the use of expression templates. As part of our engagement with the Chroma developers, we worked with our PERI collaborators at Rice and LLNL (Quinlan) to measure and diagnose Chroma performance issues. Nathan Tallent completed a Masters Thesis "Binary Analysis for Attribution and Interpretation of Performance Measurements on Fully-Optimized Code" in May 2007 that addresses the use of available compiler-generated debugging information for performance attribution. Chroma was one of the examples used in this thesis, but enough information for diagnosis (cache behavior, stall cycles, etc) was not collected. We are presented diagnosis information for Chroma in a poster at the SciDAC meeting in June 2007.

New programming languages and methods hold the promise of improving the productivity of computational scientists and programmers, while multi-core architectures are posing new challenges to achieving high performance. Chroma uses C++ "template meta-programming" to implement a high-level application-specific language, while simultaneously achieving high performance. Working with Jefferson Laboratory researchers we identified several performance issues previously "buried" under the template framework. Using HPCToolkit and pfmon, RENCI researchers have helped to diagnose parallelization strategies for the current generation of multi-core chips [Fowler2008b]. Using similar methods with the enhanced HPCToolkit, RENCI and Rice researchers engaged in a highly-interactive process with Robert Harrison of ORNL to successively identify and fix performance issues in the MADNESS chemistry code. Thus far this effort has used HPCToolkit's ability to analyze C++ template meta-programming to identify and improve performance issues in core template methods for small matrix operations. It has also identified an issue in common with Chroma in which blocking synchronization of threads on Linux can put processor cores into a power saving state, thus incurring severe performance penalties. The effort has also identified inter-thread synchronization dependence issues that have driven PERI tool development. A joint Rice and RENCI paper to appear in PPoPP2010 [Tallent2010a] describes this work. This problem drove PERI researchers to further investigations of improved methods for analyzing on-box parallelism for multi-core, multi-socket systems.

In the 2011 year, the SciDAC USQCD project shifted its emphasis to new methods appropriate for emerging multi- and many-core architectures. This is a step towards future co-design of exascale systems. One major part of this effort has been the implementation of LQCD libraries and codes for GPGPUs. To support this activity, PERI researchers at RENCI worked on performance analysis tools and methods that provide a unified view of performance across both CPUs and GPUs. The other major component of the effort was to get good performance on conventional multi-core processors. In this domain, the challenge is that the per-core memory concurrency and bandwidth are declining. Hence the challenge is to partition and schedule computations to improve data reuse in shared cache while decreasing offered load to the memory system. We are working with researchers at Jefferson Lab to explore program structuring options and are applying our Resource Centric Reflection (RCR) tool (See below.) to this problem.

Performance Measurement and Analysis of Shared Resources on Multi-core Chips.

RENCI also embarked on research into resource-centric tools for monitoring and analysis [Feng2008]. As a step in this direction, they have modified the Perfmon2 user-level tool "pfmon" to directly generate multicore profiles in "system-wide mode" that are compatible with Rice's HPCToolkit in "flat profile" mode. Since this approach does not virtualize performance counters as in PAPI, it maintains high reliability while imposing less overhead. Two performance issues being addressed are thread scheduling/synchronization problems and the interaction between SSE4 intensive computational loops and contention for shared memory [Fowler2008b].

During the performance period for PERI we constructed a prototype version of our Resource Centric Reflection tool (RCRTool) [Porterfield, TR-10-01] for AMD multi-core chips. This tool monitors performance events for resources such as shared caches, memory controllers, and interconnect across all of the activities on a chip. We have models for detecting performance bottleneck and conflict phenomena such as bandwidth/concurrency utilization, and cache thrashing due to the aggregate cache footprints of a number of threads exceeding the size of shared cache. In addition to being useful for offline performance tuning, this tool provides real time feedback usable by the operating system and by threading layers to adjust concurrency. this work is now continuing under other funding.

Characterizing Memory Concurrency and Bandwidth on Multi-core chips.

As part of the effort to understand node-wide performance, we began a program in FY09 of using the pChase memory characterization benchmark to characterize successive generations of multi-core processor chips with respect to their ability to support concurrent memory operations at the levels of the hardware thread, core, socket, and board [Porterfield2008b; Porterfield2009] . This work directly extends the Berkeley “roofline” model by adding explicit detail that explains the effects of concurrency in such systems. In particular, it identifies potential bottlenecks at all system levels that can limit multi-threaded performance.

In FY10, we have continued our work on using the pChase memory characterization benchmark to characterize the multi-core processor chips with respect to their ability to support concurrent memory operations at the levels of the hardware thread, core, socket, and board. We quantify the reduction in available bandwidth per core in the current generation of multi-core processor chips [Mandal et al, ISPASS2010]. We added the IBM Power 7, Intel Nehalem EX and Westmere, and AMD Istanbul and Magny Cours processors to the set of systems we have analyzed for codes of interest to DOE Office of Science. Our work identifies potential bottlenecks at all system levels that can limit multi-threaded performance. In addition to tighter limits on available bandwidth per core, the method also characterizes a reduced level of memory concurrency per core. An implication of this is that aggressive compiler loop optimizations to expose memory concurrency are becoming less effective on the new generations of chips [Mandal et al, LCPC2010].

Health Measurements

In the original RENCI/UNC our original SOW, there were several goals regarding monitoring environmental conditions (temperature, power, etc.) and developing open source tools and models to predict impending failures for the use of both application developers concerned with fault resilience, or by system operators concerned with faults, preventative maintenance, and economic operation of their facilities. This was and continues to be an important area, but we discontinued work in this area for several reasons. (1) There were a couple of papers [Pinhiero07,Schroeder07] published that indicate that much of the conventional wisdom regarding the ability to predict disk failures is either wrong or not as compelling as originally thought. Hence, it is not clear how one, especially a user, would use this information. (2) The problem is important enough that recent systems have extensive integrated RAS facilities that monitor and log environmental and fault events. Furthermore, these the sensors tend to have either proprietary interfaces, or they are already being handled out of band by vendor-supplied management hardware and software. In at least one case (IBM's xCAT), this has been open-sourced, but it still requires administrative privileges to use all features. Thus, the problem is important enough that it has mostly been solved by vendors and it is no longer a research issue. (3) No “customers” in national high performance computing facilities expressed a demand for another such set of tools. We therefore suspended our work on the "Health API" (HAPI). (3) The problem of "power triage" at the data center is best dealt with using the vendor supplied tools and by groups that are charged with ata center management. In the commercial domain, emerging Cloud Computing facilities monitor load and power consumption and can power on only that fraction of the system that are needed to keep utilization high without degrading response times.

Power Measurement and Adaptation

On-node power management remains an important research problem. The problem has received wide attention, but truly effective strategies have not emerged. RENCI therefore addressed this issue under PERI. In 2008 we performed experiments that confirm in the HPC domain studies [Hsu01a,Hsu01b,Hsu03] to the effect that in many cases the most effective approach to energy conservation is to use the best performing version of a program (best algorithm, most aggressive compiler optimization) to solve a problem as fast as possible and to then either move on to the next problem or to shut the hardware off. This is the same strategy used by commercial organizations to manage their clouds. The reason behind this result is that idle power of the entire compute node can be significantly larger than the variability in CPU power induced by application level power adjustment.

There were, however, been numerous papers published that study the use of dynamic voltage and frequency scaling (primarily on single core processors) to lower power consumption which, if execution time is not unduely increased, has the potential to decrease energy usage. We therefore hypothesized that there may be potential for energy savings in HPC with acceptable performance penalties for memory-bound applications. Because past studies were done on single core chips, we believe that the problem needs to be revisited In particular, imbalances between multi-core processors and memory systems can be such that some applications remain memory bound even if core clock rates and voltages are reduced. Furthermore, if the imbalance is such that using C cores has minimal contention whereas using C+1 cores causes contention in the form of greatly reduced cache effectiveness at any level, then it may be the case that performance can be increased while simultaneously reducing energy consumption. As discussed below, we have initial confirmation of these hypotheses.

In 2008-2009 we initiated an effort to rigorously evaluate the opportunities for compiler optimizations, including insertion of power control directives, in the power vs. performance space. A problem with past studies has been that they have relied on the use of external power monitoring devices. This approach has the three problems that results may be confounded power supplies that smooth the load and that have variable efficiencies; that the results are usually available only with a very coarse temporal granularity; and that the power monitor measurements are not correlated with detailed performance monitoring information. We therefore designed and constructed several instances of a prototype RENCI PowerMon, which is an internal device that sits between the power supply and system devices and that can provide frequent, detailed power measurements viaa USB interface to either the system being measured or to an external machine. See [Bedard]. We built built a revised version that fits within a 3.5 in. hard disk slot. The initial version used a software USB modem so it would supply 10 sets of 12 measurements a second. A minor revision to the prototype increases this number to approximately 800 measurements per second. A full revision of the design enabled it to monitor additional devices. Experiments with fine-grain PowerMon and hardware performance counter measurements on a multicore chip (Barcelona) indicate that adaptive runtime reduction of core voltage and frequency during phases of high memory latencies are effective for reducing energy consumption with negligible impact on performance. This runtime adaptation is a form of automatic performance tuning that will inter-operate with the techniques being developed by other PERI researchers. Wrote two papers on the power monitor, one that describes the monitor and one that covers our results.

In 2008-2009 we integrated the stream of power measurements with the output of the Perfmon2 hardware performance counter driver and tools to provide power vs. performance information to discover and guide tuning opportunities. Using RCR Toolkit, we made this information available in a shared memory segment where it is visible to other tools. such as HPCToolkit [Mellor-Crummey2002,Tallent,Adhianto]

The UNC team has also performed experiments on the value of dynamic voltage and frequency scaling (primarily on single core processors) to lower power consumption. This work is the basis for several chapters in Min Yeol Lim's dissertation at NCSU, who graduated during the year and worked for a year at RENCI as a post-doc.

While the PowerMon2 device is cost effective at a parts cost of approximate $60, it is not practical to instrument and analyze and entire large cluster. We therefore used PowerMon2 measurements to derived and validate a surrogate power model based on using the hardware performance event counters built into CPUs. The surrogate model is practical for use across an entire parallel system. Since it is based on program events, it is suitable for use in simulators, emulators, and in static models used for compiler optimization and autotuning. It is a significant step towards unified approach to performancee and power optimization. This work is described in [Lim et al, HPCA2010], which won a "Best Short Paper" award.

Reliable, Distributed Ensembles for Auto-Tuning

Large scale computational science is not just about single large runs on capability systems. Capacity is important for handling complex computational campaigns structured as workflows and ensembles. Such applications may have to run on heterogeneous hardware [Fowler2008a]. In order to perform performance experiments to tune an application to run on diverse architectures, we have performed experiments with using a Grid-enabled workflow architecture to run the experiments as a potentially large ensemble of workflows [Tilson2008]. This results in an exhaustive enumeration of the experiments. While this is more expensive than the intelligent experiment design underlying other PERI based autotuning search methods, the exhaustive search revealed non-intuitive results, mostly stemming from a divergence between vendor compiler documentation and actual implementations.

The Grid workflow framework for autotuningto help automate and parallelize the ensembles of experiments needed to do cross-platform performance analysis studies. Such studies include autotuning experiments as well as scaling studies, evaluation of compiler optimization options, comparisons of libraries, and cross-architecture studies. The framework uses the Taverna workflow engine to reliably and concurrently run a campaign of experiments across multiple systems. Each experiment is a complex workflow that includes application build steps. Results are collected in a performance database that is configured as a web service. The database is currently used in production to do regression tests on codes that are run on multiple Grid systems.

Improving Tool Usability

Two of the major impediments to the widespread use of performance tools are the shallow learning curve in getting started and the labor involved in the cycle of measurement, analysis and tuning for performance improvement. Coupling performance tools with an integrated development and execution environment is one way of lessening these problems. We worked to integrate our current and future tools with the Eclipse/PTP environment being developed at IBM, LANL, and ORNL.

During the FY07, Fowler jointly supervised Adam Bordelon, a Masters student at Rice, in porting a version of HPC viewer and a scalable cluster analysis framework to Eclipse. Bordelon's thesis, "Developing a Scalable, Portable Parallel Performance Analysis Toolkit", was completed in May. We have also used AMPL under Eclipse/PTP.

Characterization of Input/Output Behavior

Altiongh I/O was to one of the core issues addressed by PERI, we discovered that for some important applications scalability is limited because a significant fraction of the wall clock time is in the form of idle time waiting for I/O operations to be completed. We identified these issues and in our application engagement activities, we modified I/O strategies to introduce adequate buffering and asynchronous I/O operations to allow computation and I/O to overlap.

I

Application Engagement Activities.

One third of the PERI budget was directed not towards research, rather towards engagement with application teams funded through the Office of Science (SciDAC, INCITE awardees, Joule applications, and other groups.). In this section we summarize UNC application engagement activities.

Applications Survey

One of the first PERI activities was to implement a web-based application survey to gather information about code characteristics and performance goals and issues for SciDAC applications. The survey submission and some results are at: http://www.peri-scidac.org/perci/survey/. This survey has been an important tool for identifying which computational science projects are good candidates for PERI engagement. An analysis of survey results is available [deSupinski2007].

Liaisons

PERI has also established long-term relationships with projects having clearly identified performance needs and a desire to work with us. These interactions were defined and maintained by individual PERI personnel, who are assigned to be the liaisons between the science application project and PERI. The initial assignments were motivated by the information collected in the PERI Application Survey. The nature and number of liaison interactions have since evolved based on updates to the application surveys and requests from science application personnel, from DOE computing centers, and from DOE headquarters.

The nature of the interaction between PERI and a Science Application project is unique to each project and varies over time. Some of the PERI liaisons were engaged actively, helping enhance the performance of their colleague's codes, bringing in other PERI resources and expertise as needed, and educating PERI researchers as to the important performance issues. Other interactions were more passive, with the liaison simply staying in touch, tracking performance needs, advising on performance issues, and looking for opportunities for PERI to contribute, but not directly engaged for the moment in tuning. Laisons responsibilities for UNC are as follows.

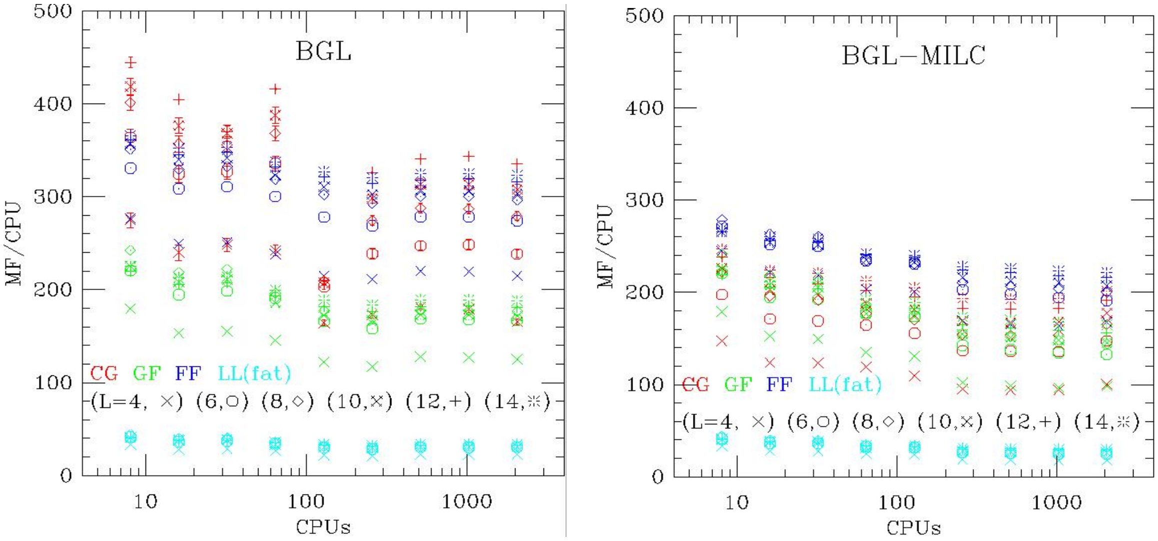
Active liaisons (close collaboration on well-defined performance issues):

National Computational Infrastructure for Lattice Gauge Theory

Application PI: Sugar (UC-Santa Barbara)
PERI liaison: Fowler (UNC)

Passive liaison (monitoring performance needs, providing advice or assistance as needed):
Multidimensional Simulations of Core-Collapse Supernovas
Application PI: Mezzacappa (ORNL)
PERI Liaison: Fowler (UNC)

In addition to our activities with Chroma discussed elsewhere, UNC researchers collectied data for the tuning of the LQCD MILC code and the SciDAC LQCD libraries, both as part of PERI and as part of the SciDAC QCD collaboration. The results of these studies were kept in the PERI performance database.



(a) MILC with SciDAC packages

(b) MILC without SciDAC packages

Figure 1: MFLOPS/CPU vs. number of processors for lattice sizes L = 4^4, 6^4, 8^4, 10^4, 12^4, 14^4

Figure 1 compares the performance MILC with and without the use of the SciDAC libraries. The plots show the computation rates for the CG (Conjugate Gradient), GF (Gauge Force), FF (Fermion Force), and LL (Long Link) phases for both the SciDAC version of MILC and the standard MILC. Figure 1 (a) presents MFLOPS versus the number of processors for varying lattice sizes; Figure 1 (b) presents MFLOPS versus the lattice sizes on one full rack – 2048 processors. The SciDAC MILC variant demonstrates significant performance improvement over the standard MILC, with the CG phase achieving the largest performance gain - 40% to 50% speedup, as shown in both figures. The fermion force phase also displayed 30% to 40% performance gain.

The PERI team at RENCI also engaged the developers of the Chimera Supernova simulation code. Throughout the first three quarters of FY07, this code was is a state of reorganization, so performance measurement and tuning was not a priority for the developers. We did a performance study of the reorganized code in Q4FY07 and shared the results with the Chimera team.

We continuied our liaison relationship with Mezzacappa's astrophysics group working on Simulations of Core-Collapse Supernovas . We produced an HPCToolkit dataset from one example; no glaring problems emerged. They have restructured the Chimera code, so spending the effort to do a more detailed study was not regarded as worthwhile at that time. Eventually, multicore issue for the Chimera and GenASIS codes dominated the discussions. GenASIS is written in C++ and our improvements in HPCToolkit's ability to deal with coding styles that heavily use templates and inlining will be invaluable for this effort.

The Chroma LQCD code is written in C++ using very advanced coding techniques such as *template metaprogramming*. This strategy improves programmer productivity and produces very good, but not the highest, performing code. Joint efforts among RENCI, Jefferson Lab, and Rice University under SciDAC funding have driven the improvement of the binary analysis parts of HPCToolkit to handle these codes.

As mentionsed above, we are applying technologies all or partially developed under PERI (RENCI resource-centric performance analysis, HPCToolkit) to the issue of tuning LQCD codes to effectively use the multi-core processors in today's leadership class systems. Current floating point performance is a little more than half of what is desired. "MPI everywhere" and MPI+OpenMP" models are inadequate to the task. Our preliminary studies are reported in [Fowler2008b] which also and the material was also presented as a poster at SciDAC2008. The results of this work will be applied to the "SciDAC libraries" which are used by the major (MILC, Chroma, CPS) LQCD codes.

Rosetta is a community code for modeling protein folding. David Baker of University of Washington has had a 2007 and 2008 INCITE awards to use the BlueGenes at ANL. PERI researchers at UNC/RENCI have been working with members of Professor Brian Kuhlman's lab at UNC, which has exchanged visits and post-docs with Baker's lab, on the tuning of new algorithms for of the Rosetta kernels. These algorithms have a theoretical advantage (fewer operations) over older methods; we are working on improving their memory locality. We have also given this team an account on the two-rack BG/L at RENCI to allow them to submit jobs at low priority, thus acting as a cycle scavenger.

We worked with Professor Gary Lackmann of NCSU to get his "hurricane WRF" running and performant on a BlueGene/L machine. Lackmann's work is supported by DOE Office of Science. This variant on WRF uses a "moving, nested mesh" AMR strategy to resolve fine grain vortices within a hurricane while still be able to support a wide area coarse grain model. A video of a visualization of this work won an "OASCR" award at SciDAC2008 in Seattle. During the 2008 Atlantic hurricane season, RENCI ran this model for the North Atlantic region. Although there has been no statistical validation, the subjective impression has been that the model out-performs standard computer models used by forcasting (A version of hurricane WRF is part of the standard ensemble.) agencies. In particular, its predictions for the landfall of TS Hanna on the North Carolina coast were stable and accurate for at least two days before the event.

Computer System Access for Engagement

The Performance Engineering and Analysis Consortium End Station (PEAC) INCITE project provided the PERI project with access to the Leadership Class Systems at ANL and at ORNL. Access to these systems was critical to achieving PERI's goals in both research and engagement. PEAC also provides access to the larger performance engineering and research community, and provides a mechanism for this larger community to contribute to the success of the DOE Leadership Computing Facilities (LCF) and DOE's goals in computational science at scale. Pat Worley led the effort to submit the PEAC proposals, requesting in 2010 40M CPU hours on the Cray XT systems at ORNL and 20M CPU hours on the IBM BG/P system at ANL. A separate but more modest allocation was also obtained by Bailey on the NERSC facility for PERI project usage.

UNC/RENCI provided and maintained the server for the performance data base as well as for a PERI code repository.

Education and Training

To foster collaboration on its PERI activities, UNC supported two Ph.D. students from Virginia Tech and NCSU for the Summer 0f 2008. In addition to advancing our core PERI research activities, these collaborations are the basis of new collaborative initiatives for which we are seeking other funding streams. Song Huang, summer of '08. (VTech Ph.D. candidate) worked on experimental prototypes of resource-centric performance monitoring for multi-core chips. Xiao Bao, was anNCSU Ph.D. candidate who worked on kernel issues related to multi-core performance. (She returned to China due to family illness)

Todd Gamblin's Ph.D. research was funded partially by PERI. He defended his Ph.D. dissertation in May 2009 and submitted the final version of his dissertation in October 20009. He is currently a member of the technical staff at Lawrence Livermore national laboratory.

Min Yeol Lim of NCSU defended his dissertation of power/energy measurement and management in August 2009. Much of the work was done at RENCI using PERI resources. In October 2009, he began a postdoctoral year at RENCI partially supported by PERI. He left this postdoc after ten months to accept a position with Intel Corporation.

Ardavan Kanani, a computer science graduate student at UNC Chapel Hill, was supported by PERI during the Summer of 2010.

Stephen Olivier was supported during this period on a DoD Graduate Fellowship. Mentoring and supervising of his research at RENCI was funded through PERI and he use PERI equipment and Office of Science applications as part of his research. He is currently on the technical staff of Sandia National Laboratory.

Other Synergistic Activities.

PI Fowler is one of the co-organizers of the third workshop on Functionality of Hardware Performance Monitors to be held in conjunction with Micro-43 to be (was) held December 4 in Atlanta Georgia. This meeting brings together chip designers from the major hardware vendors (Intel, AMD, IBM, Nvidia, Melanox, etc.) with performance tool implementers and major users from industry (Google, Amazon), and government (DOE National Laboratories.

Fowler was also the Industry Chair for PACT09, held in Raleigh, NC.

UNC Publications acknowledging PERI

[Bailey] David H. Bailey, Robert Lucas, Paul Hovland, Boyana Norris, Kathy Yelick, Dan Gunter, Bronis de Supinski, Dan Quinlan, Pat Worley, Jeff Vetter, Phil Roth, John Mellor-Crummey, Allan Snavely, Jeff Hollingsworth, Dan Reed, Rob Fowler, Ying Zhang, Mary Hall, Jacque Chame, Jack Dongarra, Shirley Moore, “Performance Engineering: Understanding and Improving the Performance of Large-Scale Codes,” *CT Watch Quarterly*, vol. 3, no. 4, pg. 18–23, Nov 2007.

[Bedard2010a] Daniel Bedard, Min Yeol Lim, Robert Fowler, and Allan Porterfield. PowerMon: Fine-grained and integrated power monitoring for commodity computer systems. In Proceedings Southeastcon 2010, Charlotte, NC, March 2010. IEEE.

[Bedard2010b] Daniel Bedard, Allan Porterfield, Rob Fowler, Min Yeol Lim. *PowerMon 2: Fine-grained, Integrated Power Measurement,* Technical Report TR-09-04, RENCI, North Carolina, October 2009

[deSupinski2008a] Bronis R. de Supinski, Robert J. Fowler, Todd Gamblin, Frank Mueller, Prasun Ratn and Martin Schulz, “An Open Infrastructure for Scalable, Reconfigurable Analysis,” *In-*

ternational Workshop on Scalable Tools for High-End Computing (STHEC), Kos, Greece, Jun 7, 2008.

[deSupinski2008b] Bronis R. de Supinski, Rob Fowler, Todd Gamblin, Frank Mueller, Prasun Ratn and Martin Schultz, "An Open Infrastructure for Scalable, Reconfigurable Analysis," *International Workshop on Scalable Tools for High-End Computing (STHEC 2008)*, ACM/SIGARCH, Jul 2008.

[deSupinski2009] B. R. de Supinski, S. Alam, D. H. Bailey, L. Carrington, C. Daley, A. Dubey, T. Gamblin, D. Gunter, P. D. Hovland, H. Jagode, K. Karavanic, G. Marin, J. Mellor-Crummey, S. Moore, B. Norris, L. Oliker, C. Olschanowsky, P. C. Roth, M. Schulz, S. Shende, A. Snavely, W. Spear, M. Tikir, J. Vetter, P. Worley, and N. Wright, "Modeling the Office of Science Ten Year Facilities Plan: The PERI Architecture Tiger Team," *Journal of Physics: Conference Series*, vol. 180 (2009) 012039.

[Feng2008] Wu Feng, Robert J. Fowler, Mark K. Gardner, Song Huang, Allan Porterfield, "Multi-Source Event Generation And Analysis For Performance Understanding In Large-Scale Environments", *ORNL Fall Creek Falls Conference*, Sep 2008, (poster).

[Fowler2008a] Robert J. Fowler, Todd Gamblin, Gopi Kandaswamy, Anirban Mandal, Allan K. Porterfield, Lavanya Ramakrishnan and Daniel A. Reed, "Challenges of Scale: When All Computing Becomes Grid Computing," in Lucio Grandinetti, editor, *High Performance Computing and Grids in Action, Advances in Parallel Computing*, IOS Press, Amsterdam, Mar 2008.

[Fowler2008b] Robert J. Fowler, Todd Gamblin, Allan K. Porterfield, Patrick Dreher, Song Huang and Balint Joo, "Performance Engineering Challenges: The View from RENCI," *J. Phys.: Conf. Ser.* **125** 012065 (6pp) doi: 10.1088/1742-6596/125/1/012065, August 2008. (Also a poster at SciDAC 2008)

[Fowler2009] R. Fowler, L. Adhianto, B. R. de Supinski, M. Fagan, T. Gamblin, M. Krentel, J. Mellor-Crummey, M. Schulz and N. Tallent, "Frontiers of Performance Analysis on Leadership Class Systems," *SciDAC 2009*, San Diego, California, Jun 14 -18, 2009.

[Gamblin2008a] Todd Gamblin, Bronis R. de Supinski, Martin Schulz, Robert J. Fowler and Daniel A. Reed, "Scalable Load Balance Measurement for SPMD Codes," *SC2008*, Austin, Texas, November 15–21, 2008.

[Gamblin2008b] Todd Gamblin, Rob Fowler, and Daniel A. Reed, "Scalable Methods for Monitoring and Detecting Behavioral Classes in Scientific Codes," in *Proceedings of the International Parallel and Distributed Processing Symposium 2008*, Miami, FL, Apr 2008.

[Gamblin2008c] Todd Gamblin, Prasun Ratn, Bronis R. de Supinkski, Martin Schulz, Frank Mueller, Robert J. Fowler, and Daniel A. Reed. An open framework for scalable, reconfigurable performance analysis. SC07, Reno, NV, November 2007. (Poster).

[Gamblin2009] Todd Gamblin, "Scalable Performance Measurement and Analysis," Ph.D. Dissertation, Department of Computer Science, University of North Carolina, 2009.

[Gamblin2010] Todd Gamblin, Bronis de Supinski, Martin Schulz, Rob Fowler, and Daniel Reed. Efficiently clustering performance data at massive scales. In Proceedings of the International Conference on Supercomputing 2010 (ICS2010), Tsukuba, Japan, June 2010. ACM.

[Lim2009] Min Yeol Lim, "Improving Power and Performance Efficiency in Parallel and Distributed Computing Systems," Ph.D. dissertation, Department of Computer Science, North Carolina State University, 2009.

[Lim2010] Min Yeol Lim, Allan Porterfield, and Robert Fowler. SoftPower: Fine-Grain Power Estimations Using Performance Counters. In The ACM International Symposium on High Performance Distributed Computing (HPDC), Chicago, July 2010. ACM. Best short paper award.

[Mandal2010a] Anirban Mandal, Rob Fowler, and Allan Porterfield. Modeling memory concurrency for multi-socket multi-core systems. In Proceedings of the 2010 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS2010), pages 56–75, White Plains, NY, March 2010. IEEE.

Mandal2010b] Anirban Mandal, Min Yeol Lim, Allan Porterfield, and Robert Fowler, Implications for Applications and Compilers of Multi-core Memory Concurrency, International Workshop on Languages and Compilers for Parallel Computing (LCPC2010), Houston, TX, Oct. 2010. (Poster). Full version released as RENCI TR 10-03 (http://www.renci.org/publications/technical-reports).

[Porterfield2008a] Allan Porterfield, Robert Fowler and Mark Neyer, "MAESTRO: Dynamic Runtime Power Control," in *Workshop on Managed Multicore Systems (MMCS)*, Boston, MA, Jun 2008.

[Porterfield2008b] Allan Porterfield, Robert J. Fowler, Anirban Mandal, and Min Yeol Lim, "Performance Consistency on Multi-Socket AMD Opteron Systems, *UNC/RENCI Technical Report TR-08-07*, RENCI, North Carolina, Dec 2008.

[Porterfield2009] Allan Porterfield, Rob Fowler, Anirban Mandal, and Min Yeol Lim, "Empirical Evaluation of Multi-Core Memory Concurrency," *UNC/RENCI Technical Report TR-09-01*, RENCI, Chapel Hill, North Carolina, January 2009.

[Porterfield2010] Allan Porterfield, Rob Fowler, Min Yeol Lim. *RCRTool: Design Document; Version 0.1,* Technical Report TR-10-01, RENCI, North Carolina, February 2010.

[Ramakrishnan2009] L. Ramakrishnan, D. Nurmi, A. Mandal, C. Koelbel, D. Gannon, T. M. Huang, Y. S. Kee, G. Obertelli, K. Thyagaraja, R. Wolski, A. Yarkhan, D. Zagorodnov, *"VGrADS: Enabling e-Science Workflows on Grids and Clouds with Fault Tolerance",* in Proceedings of the IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC) '09, Portland, OR.

[Tallent2010a] Nathan R. Tallent, John M. Mellor-Crummey, Alan Porterfield. *Analyzing Lock Contention in Multithreaded Applications,* PPoPP 2010, January 2010.

[Tilson2008] Jeffery L. Tilson, Mark S. C. Reed and Robert J. Fowler, "Workflows for Performance Evaluation and Tuning," in *2008 IEEE International Conference on Cluster Computing (Cluster 2008)*, pg. 8, Tsukuba, Japan, Sep 2008, IEEE.

[Tilson2010] Jeffrey L Tilson, Gloria Rendon, Eric Jakobsson. *Using high performance computing and domain-based functional annotation of proteins to enhance discovery of novel proteins, identify functional homology, and characterize phylogenetic relatedness,* Technical Report TR-10-02, RENCI, North Carolina, June 2010.

[Zhang07] Y. Zhang, R. Fowler, K. Huck, A. Malony, A. Porterfield, D. Reed, S. Shende, V. Taylor, and X. Wu. US QCD computational performance studies with PERI. *J. Phys: Conf. Ser, 78(012083):5pp*, August 2007. (Also a poster at SciDAC 2007)

Other References

[Adhianto] Laksono Adhianto, Sinchan Banerjee, Michael Fagan, Mark Krentel, Gabriel Marin, John Mellor-Crummey, Nathan Tallent, “HPCToolkit: Tools for performance analysis of optimized parallel programs,” submitted to *Concurrency and Computation: Practice and Experience*, submitted, Aug 2008.

[McCurdy] C. McCurdy, A. Cox, and J.S. Vetter, “Investigating the TLB Behavior of High-end Scientific Applications on Commodity Microprocessors,” *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Austin, TX, 2008.

[Mellor-Crummey2002] John Mellor-Crummey, Robert Fowler, Gabriel Marin, and Nathan Tallent. HPCView: a tool for top-down analysis of node performance. *The Journal of Supercomputing*, 23:81-104, 2002.

[Tallent] Nathan Tallent, John Mellor-Crummey, Laksono Adhianto, Michael Fagan, and Mark Krentel 2008. “HPCToolkit: Performance Tools for Scientific Computing,” *SciDAC 2008, Journal of Physics Conference Series* 125 012088.

[Pinhiero07] Eduardo Pinhiero, Wolf-ietrich Weber and Luiz Andre Barroso, "Failure Trends in a Large Disk Drive Population." *5th USENIX Conference on File and Storage Technologies (FAST '07*), February 2007

[Schroeder07] Bianca Schroeder and Garth Gibson, "Disk failures in the real world: What does a MTTF of 1,000,000 hours mean to you?", *5th USENIX Conference on File and Storage Technologies (FAST '07*), February 2007

[Hsu01a] C.-H. Hsu, U. Kremer, M. Hsiao, "Compiler- Directed Dynamic Voltage/Frequency Scheduling for Energy Reduction in Microprocessors", "Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED'01)", August, 2001.

[Hsu01b] C.-H. Hsu,U. Kremer, "Dynamic Voltage and Frequency Scaling for Scientific Applications", *Proceedings of the 14th annual workshop on Languages and Compilers for Parallel Computing (LCPC 2001)*, August 2001.

[Hsu03] C.-H. Hsu and Uli Kremer, "The Design, Implementation, and Evaluation of a Compiler Algorithm for CPU Energy Reduction", *Proceedings of the ACM SIGPLAN Conference on Programming Languages Design and Implementation {PLDI)*", June, 2003.

[Williams2007] S. Williams, L. Oliker, R. Vuduc, J. Shalf, K. Yelick and J. Demmel, “Optimization of Sparse Matrix-Vector Multiplication on Emerging Multicore Platforms,” *SC07*, ACM/IEEE, Nov 2007.

[Williams2008a] S. Williams, J. Carter, L. Oliker, J. Shalf and K. Yelick, “Lattice Boltzmann Simulation Optimization on Leading Multicore Platforms,” *International Parallel & Distributed Processing Symposium (IPDPS)*, to appear, 2008. WINNER: Best paper, applications track.

[Williams2008b] Samuel Williams, Kaushik Datta, Jonathan Carter, Leonid Oliker, John Shalf, Katherine Yelick, David Bailey, “PERI - Auto-tuning Memory Intensive Kernels for Multicore,” *SciDAC 2008*, available at http://crd.lbl.gov/~dhbailey/dhbpapers/scidac08_peri.pdf.