Performance Evaluation Research Center (PERC)
Executive Summary
04 September 2001

PERC is an Integrated Software Infrastructure Center (ISIC) for enabling improved of high-end computer performance. In particular, the Center is developing a *science* for understanding performance of scientific applications on high-end computer systems, and is also developing *engineering* strategies for improving performance on these systems. The project integrates several active efforts in the high performance computing (HPC) community and is forging alliances with application scientists working on DOE Office of Science (SC) missions to ensure that the resulting techniques and tools are truly useful to end users.

The project focuses on how one can best execute a specific application on a given platform. In particular, we seek to

1. Understand the key factors in applications that affect performance.
2. Understand the key factors in computer systems that affect performance.
3. Develop models that accurately predict performance of applications on systems.
4. Develop an enabling infrastructure of tools for performance monitoring, modeling and optimization.
5. Validate these ideas and infrastructure via close collaboration with DOE SC and other application owners.
6. Transfer the technology to end-users.

This activity focuses on high performance computing systems (i.e., large distributed memory parallel systems; large shared memory systems; and large cluster systems). We also focus on representative scientific applications and problems of interest to the SC, initially those areas emphasized in the SciDAC application projects. Our objective is to characterize the realistic performance bounds of HPC applications and systems, understand why these limits exist, determine what can be done to more closely approach these limits, and predict what can be expected on future applications and systems. To that end, we will develop benchmarks, models, analytical techniques, and tools.

A *science* of performance predicts what levels of performance an application can achieve on a particular platform. Performance *engineering* develops practical strategies to maximize achieved performance. Both objectives require understanding how application codes exploit, or fail to exploit, the available resources on a particular platform. The PIs represented in this activity bring diverse and complementary expertise, experience, tools, and research plans. However, they share a common vision on how to achieve this understanding of performance.

The fundamental premise of this activity is that overall application performance (namely wall-clock execution time) is dominated by how well the application exploits the entire memory hierarchy of a machine. Hence, a science of performance must develop abstractions of performance phenomena or *models* that capture memory system performance accurately. These models require some performance data as input and, just as importantly, require significant volumes of performance data to validate the models. To this end, we will adopt existing *benchmarks* when possible, but generate new benchmarks when required. In addition, we must also make significant advances in *performance monitoring and collection tools*. Further, we must gather the data for relevant applications. Finally, performance engineering requires that we use our models to guide the development of new *performance optimizers* that enhance the exploitation of the memory hierarchy of end-user codes. Thus, we focus on four areas: benchmarking; performance tools; modeling and analysis; and performance optimizers.

We require understandable, clear benchmarks that make extensive use of the memory hierarchy as a baseline for our model and tool development. Furthermore, several key DOE SC disciplines are not well represented by existing benchmarks. The Oak Ridge National Laboratory (ORNL) and the Lawrence Berkeley National Laboratory (LBNL) will produce several discipline-specific benchmarks, with an early emphasis on SciDAC applications such as climate, chemistry and high-energy nuclear physics. These benchmarks will include full applications, kernels, and low-level benchmarks, with established linkages among levels. There will also be an effort to modify benchmarks to determine performance sensitivity to tuning, and to explore opportunities for building performance portability into application codes.

For performance monitoring tools, we intend to develop a software infrastructure for monitoring and collecting performance data. In most cases, prototypes of these tools already exist. Some of these include the SvPablo toolkit, under development at the University of Illinois Urbana-Champaign (UIUC), the Sigma tool for cache measurement from the University of Maryland (UMD), and the PAPI infrastructure from the University of Tennessee Knoxville (UTK). Our work will focus on adding necessary features to these tools, porting them to all platforms of interest (so that our modeling and analysis work will be based on fully comparable data), and enhancing the reliability and usability of these tools for application scientists. We will also develop an infrastructure for collecting and analyzing the data that these tools produce.

Performance modeling and analysis represent the scientific heart of this project. We intend to develop a hierarchy of models that reasonably and accurately describe and predict performance. These efforts will build on activities at UMD, UIUC, the Lawrence Livermore National Laboratory (LLNL), the Argonne National Laboratory (ANL), ORNL, the San Diego Supercomputing Center (SDSC), and LBNL. Our research plan is to develop each of these approaches and evaluate their predictive capability against a common set of test cases, disclosing which modeling and analysis techniques are the most effective and easily performed. All of the above-mentioned institutions will participate in the testing and analysis of results. LBNL and UTK will provide a common repository for data, analyses, and tools.

Finally, for performance optimizers, we intend to develop software tools that will either optimize codes automatically or assist users in optimizing codes. Some of these tools will require data provided by the performance monitoring tools and modeling techniques mentioned above. Others, such as the Self Adapting Numerical Software (SANS) tool being developed at UTK, will perform the necessary measurements themselves. As before, each of these activities will build on some existing research projects. ORNL, UMD, LLNL, ANL, UTK and LBNL are participating in this activity.

Dr. David H. Bailey of LBNL is leading the laboratory activities in PERC, while Prof. Jack Dongarra of the University of Tennessee, Knoxville, is leading the university activities. The PERC web site is: http://perc.nersc.gov.

We have made some initial contacts for collaboration with other SciDAC projects, including the following:

High Energy and Nuclear Physics
1. Shedding New Light on Exploding Stars: Terascale Simulations of Neutrino-Driven SuperNovae and Their NucleoSynthesis
http://www.osti.gov/scidac/henp/projects/mezzacappa.html

2. Advanced Computing for 21st Century Accelerator Science and Technology
http://www.osti.gov/scidac/henp/projects/ko_ryne.html

Biology and Environmental Research
3. Collaborative Design and Development of the Community Climate System Model for Terascale Computers
http://www.osti.gov/scidac/ber/projects/malone.html

Fusion Energy Sciences
4. Numerical Computation of Wave-Plasma Interactions in Multi-dimensional Systems
http://www.osti.gov/scidac/fes/projects/batchelor.html

Advanced Scientific Computing
5. Terascale Optimal PDE Solvers (TOPS)
http://www.osti.gov/scidac/computing/projects/keyes.html

6. Applied Partial Differential Equations Center (APDEC)

7. Scientific Data Management (SDM)
http://sdm.lbl.gov/sdmcenter

Chemical Sciences
8. Accurate Properties for Open-Shell States of Large Molecules
http://www.osti.gov/scidac/bes/projects/taylor.html