

Performance Evaluation with Industrial Applications

Rudolf Eigenmann
Purdue University
eigenman@ecn.purdue.edu

Greg Gaertner
Compaq Computer Corporation
ggg@zko.dec.com

Faisal Saied Mark Straka
NCSA
{fsaied,straka}@ncsa.uiuc.edu

October 1, 1998

1 Motivation and Goals

It is being increasingly understood that the evaluation of computer systems performance must be based on realistic, full applications. Kernel, algorithm, and small-application benchmarks are important for measuring and discussing the performance of individual components of a computer system. However, the complex performance behavior of these components in concert and in the context of a large computational problem can only be realistically understood if we observe the problem as a whole. These needs are further increased by the fact that computer systems are able to solve ever larger and more complex problems. Doing performance evaluation at this large scope is a grand challenge and exceeds the resources of most research projects.

To address this challenge, a joint industrial-academic effort has been initiated with the mission to facilitate performance evaluation and benchmarking with large, industrial applications. There are two complementary goals. The industrial participants pursue the development of systems benchmarks that provide meaningful and relevant applications for measuring and comparing high-performance computer systems across diverse platforms. The academic participants seek realistic program suites for evaluating and guiding the research of new concepts and prototypes. Both performance evaluation efforts have the need for large-scope, industrially relevant applications.

2 History: the SPEC High-Performance Group

In 1994 the High-Performance Group of the Standard Performance Evaluation Corporation (SPEC/HPG) was founded with the mission to establish, maintain, and endorse a suite of benchmarks representative of real-world, high-performance computing applications [EH96]. SPEC/HPG members represent Hewlett-Packard, Compaq Computer Corp., Fujitsu America, International Supercomputing Technology Institute (ISTI, France), Kuck & Associates, Sun Microsystems, the Parkbench organization (represented by the University of Tennessee), the University of Illinois, the University of Minnesota, and Purdue University.

Several efforts joined forces to form SPEC/HPG and to initiate a new benchmarking venture that is supported broadly. The two driving forces came from the Perfect Benchmarks [BCK⁺89, CKPK90] effort and the Standard Performance Evaluation Corporation (SPEC). The SPEC organization, since its foundation in the late 1980's, has become the accepted leader in evaluating the performance of workstations. SPEC's mission is to provide

tools that help both manufacturers and users gain more insight into the performance that can be expected from their computer systems. SPEC is broadly industry-based. Its membership has grown to more than 50 organizations, including computer vendors, systems integrators, research firms and academic institutions. Benchmark test results, such as performance numbers of the SPEC95 suite, are reported on the Web at www.spec.org. With the formation of the High-Performance Group, SPEC extends its activities to address performance evaluation needs across the spectrum of today's high-performance systems. Further efforts have been initiated by SPEC, such as the new benchmarks for Java and Web technology.

The Perfect Benchmarks effort was initiated in 1988 to provide a balanced set of realistic scientific and engineering application programs and to use them for evaluating high-performance computers. In coordination with the vendors of such systems, performance results were obtained and documented in several reports. This effort led to the first comprehensive evaluation of the sustainable application performance of supercomputers at that time. While the Perfect Benchmarks are still used by many researchers, the need for an updated set of programs with larger data sets and the objective to maintain the benchmarking effort over a long time period led to the search for new benchmarking partners, which resulted in the foundation of the SPEC High-Performance Group.

A notable, complementing benchmarking effort that has influenced SPEC/HPG is the *Parkbench* organization (PARallel Kernels and BENCHmarks). Parkbench was founded in 1992 by a group of interested parties from universities, laboratories and industry. Members from both Europe and the USA [HE94] are participating in this effort with the goal of establishing a comprehensive set of parallel benchmarks and to set standards for benchmarking methodologies. Maintaining a repository for the benchmarks and results is part of the objective. A representative of the Parkbench activity is a member of SPEC/HPG and the two efforts attempt to complement each other.

SPEC/HPG had to overcome several challenges to reach the current status, which offers a benchmark suite for high-performance computers and a published set of reviewed results. These challenges include:

- The search for industrially relevant candidate benchmark applications. We have found that the range of computational applications that meet the selection criteria is rather limited. These criteria are that the codes
 - are being widely-used to solve realistic problems,
 - can be distributed by SPEC (basically with a not-for-profit agreement),
 - are available in a serial and a parallel code variant,
 - have a sponsor that is able to support the development of the code into a benchmark.

The criterion of unrestricted distribution by SPEC has been especially difficult to reconcile with the requirement for industrial relevance. The most relevant codes typically have commercial value and are not freely available. Because of this, a recent initiative by SPEC/HPG has been to include new benchmarks and benchmarking rules for commercial codes by independent software vendors (ISVs).

The SPEChpc benchmark suite			
Code	Area	Programming language	#lines
SPECchem	Molecular modeling	Fortran 77 and C / PVM,MPI,OpenMP	110'000
SPECseis	Seismic processing	Fortran 77 and C / PVM,MPI,OpenMP	20'000
SPECclimate	Weather modelling	Fortran 77 and C / PVM,MPI,OpenMP	50'000

Table 1: SPEChpc benchmarks

- The preparation of the codes for benchmarking purposes. To this end we have defined appropriate data sets, inserted validation procedures in the codes, run the codes on the various platforms to verify portability, and resolved a number of portability and correctness problems. This process was time-consuming due to the large size of the codes and the required system resources for executing the benchmarks.
- The definition of run rules. High-performance machines are expensive, justifying higher investments in software tuning than for workstations. While the benchmark run rules had to reflect this fact, they also had to limit time investment in tuning a benchmark. The current rules allow for most source-level code improvements, but disallow excessive tuning, such as assembly-level optimizations. Another issue was the tradeoff between protecting company interests in keeping code optimizations confidential and disclosing all code modifications for reproducibility of the submitted results. The current run rules require the full disclosure of all code improvements, except that system-specific libraries may be used (given they are generally available) without documentation of their implementation.

3 Current Status of the SPEChpc benchmarks

A first release of the a SPEChpc96 benchmarks included two codes, SPECseis [MH93], and SPECchem [SBB⁺93], which are representative of applications used in the seismic, and the chemical and pharmaceutical industries, respectively. Both codes are actual suites of applications, with the specific application being selected through input parameters to a program run. A third code, SPECclimate has been included in 1998. It is an application used for weather forecasting, developed by the National Center for Atmospheric Research (NCAR) and collaborators. The codes are listed in Table 1. All benchmarks are available in a serial and a parallel code variant. The parallel codes are available in both message passing and in parallel directive form. The SPEChpc96 release included only message passing (PVM) variants, while a new release (to be available in Q4/1998) will include MPI and OpenMP code versions.

SPEChpc results are published on SPEC’s official web page (www.spec.org/hpg/). These publications include performance results, details of the system configurations on which the measurements were taken, and disclosure notes. The disclosure notes give the information necessary to reproduce the results, such as code modifications, compiler flags, and special make procedures. Figure 1 shows several of these results. The performance metric is computed as $\frac{\text{elapsed time (in seconds)}}{86400s}$. While this metric can be interpreted as “how many times the benchmark can be run in a day”, it is *not* intended to be a throughput measure. The rationale behind this metric is that only elapsed (i.e., wallclock) time is a reasonable measure, but that

the metric should increase as performance increases. The normalization by the “number of seconds in a day” is an arbitrary decision.

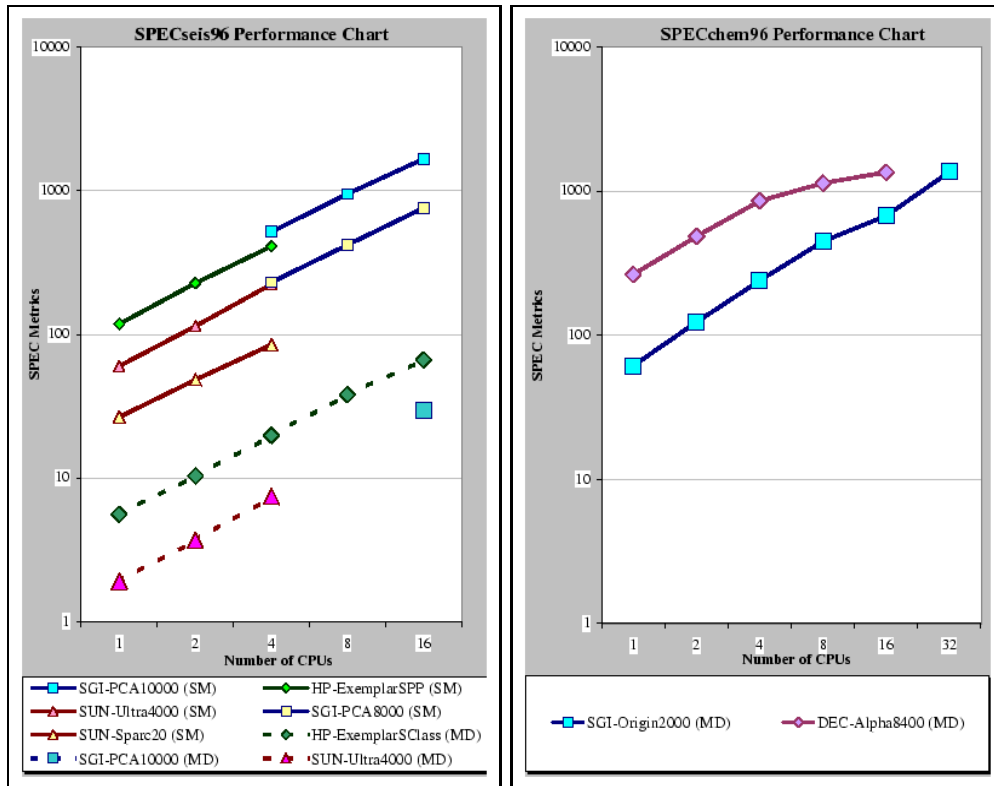


Figure 1: SPECChpc96 results. For full disclosure see www.spec.org/hpg/results.html

4 Ongoing Efforts

Advancing SPECChpc benchmarks

SPECChpc results. The most important current goal of the SPEC/HPG committee is to increase the result sets of the SPECChpc suite. While it is relatively easy to generate results for the small and medium data sets, it takes substantial resources to run the large data sets. For example, the SPECseis_XL extra large set, requiring 4 TeraBytes of disk space, is difficult to run because of the expense of allocating the necessary resources.

Including new benchmarks Another important goal is the inclusion of additional benchmarks. We are currently considering a number of candidate codes from NCSA users. Several research codes have been used as benchmarks at NCSA over the past several years. Two of these, ZEUS-MP and PUPI, have been developed by NCSA research groups. PUPI is a path integral monte carlo particle code. ZEUS-MP is a computational fluid dynamics code under development at the Laboratory for Computational Astrophysics (University of Illinois at Urbana-Champaign) for the simulation of astrophysical phenomena. Another code, UHBD (University of Houston

Brownian Dynamics), solves the linearized and non-linear Poisson-Boltzmann equation using a finite-difference method. A further candidate application, UHGROMOS, is a parallel molecular dynamics program developed at the University of Groningen (the Netherlands) and the University of Houston.

In addition, in a joint effort with a benchmarking effort for “the Climate, Weather, and Ocean Modeling Community”, led by NCAR and the University of Minnesota, we are including new applications in climate and ocean modeling. Among those, NCAR’s CCM3 code is considered next. CCM3 is an atmospheric general circulation model designed for climate research on high-end computers.

Coordination with related benchmarking efforts. Establishing and maintaining working relationships with other benchmarking organizations [ea91, BBDS92, Don90, vdS91, Gin95, SWG92] is a continuous SPEC/HPG mission. The joint effort with a benchmarking group at NCAR was mentioned above. SPEC/HPG intends to complement rather than compete with other benchmarking activities and it will provide services where other benchmarking efforts lack resources. Several coordination efforts are underway. For example, the Parkbench benchmarking model envisions three layers of measurements, at the kernel, algorithm, and compact application level, respectively. Kernel benchmarks are quick to execute and can give insights into basic system characteristics, whereas application benchmarks measure the overall behavior at the price of longer benchmarking times. A fourth layer in the Parkbench model would include full applications, such as those considered in the SPEC/HPG suite. Further discussions have been initiated by SPEC/HPG to include benchmarks used by the DOD modernization program and by the ASCI benchmarking groups in the SPECchpc suite.

While many of these activities complement each other naturally, there are also important differences, relevant for benchmark users. For example, the DOD benchmarking effort uses several defense codes that are not available to the public. SPEC/HPG, by contrast, attempts to include industrial applications, which may be commercially or publicly available. The Parkbench effort, also by contrast, considers only freely available codes. Further differences are in the process of reporting results. Parkbench results will become public if they satisfy basic reporting requirements, whereas SPEC/HPG’s official results are subject to a strict review process with the intent of increasing accuracy and repeatability of the reported data.

Use of SPECchpc in scientific performance evaluation

The goal of the academic participants in SPEC/HPG has been to create a suite of realistic programs for use in performance evaluation efforts of computer systems research projects. To this end, we have engaged in a large number of projects in architecture design, compiler development, and algorithm research that use the SPECchpc suite as a working ground. As part of these projects, extensive characterization data of the SPECchpc programs and their performance behavior is becoming available. This information goes significantly beyond the benchmarking objective of SPEC/HPG and is not subject to formal run rules. The group of scientists interested in using SPEC/HPG benchmarks is informally called the SPEC Academic Advisory Board (SAAB). A Web repository for reporting results of SAAB projects is under development. The following descriptions give an overview of research projects that were performed with SPECchpc benchmarks or are related to these activities.

At Compaq Computer Corporation we have modified the MPI/OpenMP version of SPECchem to allow both static and dynamic allocation of work to processors. The dynamic allocation was controlled by a daemon, which assigned the next available iteration to a processor requesting work. The static allocation was table driven, the table containing processor assignments for the iterations. In order to determine an optimum static allocation scheme, a genetic algorithm was applied using run time as an evolution criteria. The static assignment thus obtained outperformed the dynamic allocation method by nearly 20%.

At Purdue University we have characterized the computation, communication and I/O behavior of the SPECseis application on a number of computer architectures [AE98]. Based on this work we have developed a “performance forecasting” methodology and tool, which can extract this behavior from a given application program and express it mathematically in function of the program’s input data and architectural parameters (such as the number of processors and network characteristics). These expressions can then be used to investigate the application’s behavior on future machines and with very large data sets. Figure 2 shows the result of such an analysis.

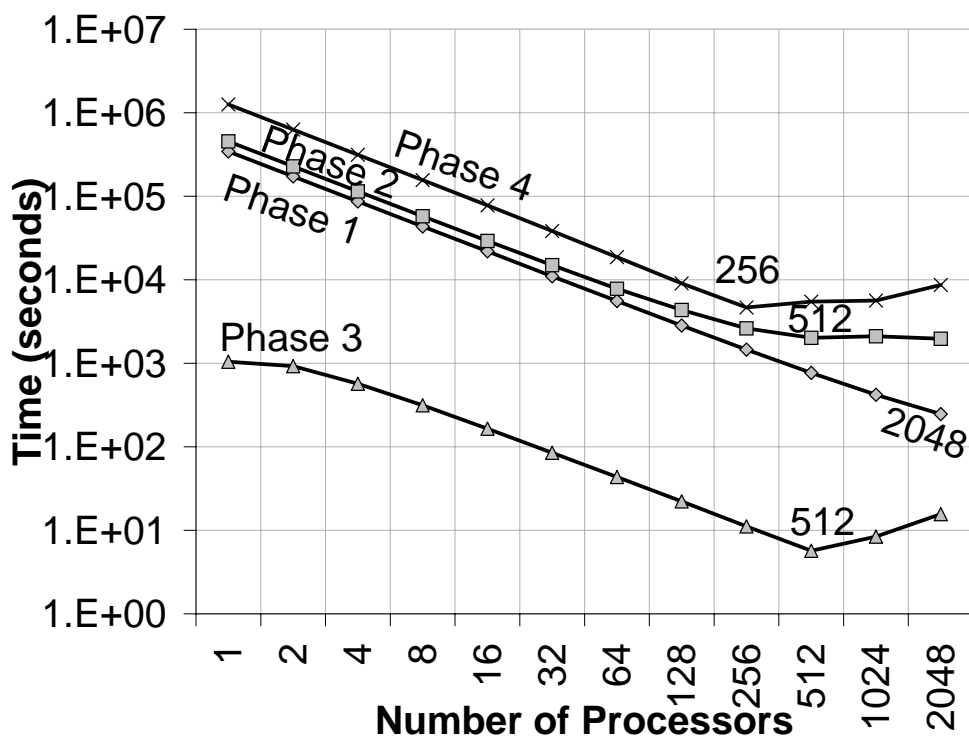


Figure 2: Performance “forecast” of the SPECseis benchmark. The four “phases” of the application scale well up to a certain, phase-specific, best-architecture point. Increasing the number of processors beyond this point degrades performance. These measurements assume a computation-to-communication ratio that is 100 times higher than that of the SGI Origin machine, hence simulating a future computer system with high-performance processors but only moderately increased network speeds.

At NCSA, a Performance Database and its related web-based infrastructure is being created

and maintained by the Performance Engineering and Computational Methods (PECM) group (www.ncsa.uiuc.edu/SCD/Perf/). It is designed to make accessible to the public the achievements of NCSA and partners in the realm of applications software development and tuning. Database users can find the effects on software performance of changes in compiler technology, numerical libraries, parallel algorithms, operating system upgrades, hardware upgrades, etc. The performance database presents summary graphs of such critical code characteristics as scalability, cache performance, peak Mflops, and algorithmic and architectural cross-comparisons. Eventually, this system is planned to support automated data input through user submission forms, dynamic plotting, and flexible comparison across arbitrary fields in the database (e.g. scaling of MPI on SGI and HP machines).

5 Conclusions

SPEC/HPG is working toward a comprehensive evaluation of all available high-performance computer systems, using applications that are industrially significant and representative of compute-intensive disciplines. Thanks to a broad support by major computer manufacturers and thanks to the active involvement of research organizations, this goal is within reach. We expect the achievement of these goals to have a significant impact on both the computer industry and the research community. For the first time it will be possible to compare systems from very different manufacturers in a quantitative way with applications that are relevant to the end users. Customers will find performance numbers that reflect their use of the system. For the research community, having available a well documented and characterized suite of realistic applications is of great value, because it will enable us to evaluate computer systems concepts and prototypes under most realistic workloads. In this way, we will be able to foresee the behavior of new computer systems and address important performance issues before the systems are delivered.

References

- [AE98] Brian Armstrong and Rudolf Eigenmann. Performance forecasting: Towards a methodology for characterizing large computational applications. In *Proc. of the Int'l Conf. on Parallel Processing*, pages 518–525, August 1998.
- [BBDS92] D. H. Bailey, E. Barszcz, L. Dagum, and H. Simon. NAS parallel benchmark results. In *Proc. Supercomputing '92*, pages 386–393. IEEE Computer Society Press, 1992.
- [BCK⁺89] M. Berry, D. Chen, P. Koss, D. Kuck, L. Pointer, S. Lo, Y. Pang, R. Roloff, A. Sameh, E. Clementi, S. Chin, D. Schneider, G. Fox, P. Messina, D. Walker, C. Hsiung, J. Schwarzmeier, K. Lue, S. Orszag, F. Seidl, O. Johnson, G. Swanson, R. Goodrum, and J. Martin. The Perfect Club Benchmarks: Effective Performance Evaluation of Supercomputers. *Int'l. Journal of Supercomputer Applications*, 3(3):5–40, Fall 1989.
- [CKPK90] George Cybenko, Lyle Kipp, Lynn Pointer, and David Kuck. Supercomputer Performance Evaluation and the Perfect BenchmarksTM. *Proceedings of ICS, Amsterdam, Netherlands*, pages 254–266, March 1990.

- [Don90] J. J. Dongarra. The Linpack benchmark: An explanation. In A. J. van der Steen, editor, *Evaluating Supercomputers*, pages 1–21. Chapman and Hall, London, 1990.
- [ea91] C. A. Addison et. al. The GENESIS distributed-memory benchmarks. In J. J. Dongarra and W. Gentzsch, editors, *Computer Benchmarks*, pages 257–271. Elsevier Science Publishers, Amsterdam, 1991.
- [EH96] Rudolf Eigenmann and Siamak Hassanzadeh. Benchmarking with real industrial applications: The SPEC High-Performance Group. *IEEE Computational Science & Engineering*, 3(1):18–23, Spring 1996.
- [Gin95] Myron Ginsberg. Creating an automotive industry benchmark suite for assessing the effectiveness of high-performance computers. In *Proc. Ninth Int'l. Conf. on Vehicle Structural Mechanics and CAE*, pages 381–390. Society of Automotive Engineers, Inc. Warrendale, PA, 1995.
- [HE94] R. W. Hockney and M. Berry (Editors). PARKBENCH report: Public international benchmarking for parallel computers. *Scientific Programming*, 3(2):101–146, 1994.
- [MH93] C. C. Mosher and S. Hassanzadeh. ARCO seismic processing performance evaluation suite, user's guide. Technical report, ARCO, Plano, TX., 1993.
- [SBB⁺93] M. W. Schmidt, K. K. Baldrige, J. A. Boatz, S. T. Elber, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, and J. A. Montgonery. The general atomic and molecular electronics structure systems. *Journal of Computational Chemistry*, 14(11):1347–1363, 1993.
- [SWG92] Jaswinder Pal Singh, Wolf-Dietrich Weber, and Anoop Gupta. Splash: Stanford parallel applications for shared-memory. *Computer Architecture News*, 20(1):5–44, 1992.
- [vdS91] A. J. van der Steen. The benchmark of the EuroBen group. *Parallel Computing*, 17:1211–1221, 1991.