HPCS HPCchallenge Benchmark Suite

David Koester
The MITRE Corporation
Email Address: dkoester@mitre.org

Jack Dongarra and ***Piotr Luszczek***
ICL/UT
Email Addresses: {dongarra, luszczek}@cs.utk.edu

Abstract
The Defense Advanced Research Projects Agency (DARPA) High Productivity Computing Systems (HPCS) HPCchallenge Benchmarks examine the performance of High Performance Computing (HPC) architectures using kernels with more challenging memory access patterns than just the High Performance LINPACK (HPL) benchmark used in the Top500 list. The HPCchallenge Benchmarks build on the HPL framework and augment the Top500 list by providing benchmarks that bound the performance of many real applications as a function of memory access locality characteristics. The real utility of the HPCchallenge benchmarks are that architectures can be described with a wider range of metrics than just Flop/s from HPL. Even a small percentage of random memory accesses in real applications can significantly affect the overall performance of that application on architectures not designed to minimize or hide memory latency. The HPCchallenge Benchmarks includes a new metric — Giga UPdates per Second — and a new benchmark — RandomAccess — to measure the ability of an architecture to access memory randomly, i.e., with no locality. When looking only at HPL performance and the Top500 List, inexpensive build-your-own clusters appear to be much more cost effective than more sophisticated HPC architectures. HPCchallenge Benchmarks provide users with additional information to justify policy and purchasing decisions. We will compare the measured HPCchallenge Benchmark performance on various HPC architectures — from Cray X1s to Beowulf clusters — in the presentation and paper. Additional information on the HPCchallenge Benchmarks can be found at http://icl.cs.utk.edu/hpcc/

Introduction
At SC2003 in Phoenix (15-21 November 2003), Jack Dongarra (ICL/UT) announced the release of a new benchmark suite — the HPCchallenge Benchmarks — that examine the performance of HPC architectures using kernels with more *challenging* memory access patterns than High Performance Linpack (HPL) used in the Top500 list. The HPCchallenge Benchmarks are being designed to *complement* the Top500 list and provide benchmarks that *bound* the performance of many real applications as a function of memory access characteristics — e.g., spatial and temporal locality. Development of the HPCchallenge Benchmarks is being funded by the Defense Advanced Research Projects Agency (DARPA) High Productivity Computing Systems (HPCS) Program.

The HPCchallenge Benchmark Kernels

Local	Global
DGEMM (matrix x matrix multiply)	High Performance LINPACK (HPL)
STREAM • COPY • SCALE • ADD • TRIADD	PTRANS — parallel matrix transpose
RandomAccess	(MPI)RandomAccess
1D FFT	1D FFT
I/O b_eff — effective bandwidth benchmark	

Additional information on the HPCchallenge Benchmarks can be found at http://icl.cs.utk.edu/hpcc/.

Flop/s

The Flop/s metric from HPL has been the de facto standard for comparing High Performance Computers for many years. HPL works well on all architectures — even cache-based, distributed memory multiprocessors — and the measured performance may not be representative of a wide range of real user applications like adaptive multi-physics simulations used in weapons and vehicle design and weather, climate models, and defense applications. HPL is more compute friendly than these applications because it has more extensive memory reuse in the Level 3 BLAS-based calculations. .

Memory Performance

There is a need for benchmarks that test memory performance. When looking only at HPL performance and the Top500 List, inexpensive build-your-own clusters appear to be much more cost effective than more sophisticated HPC architectures. HPL has high spatial and temporal locality — characteristics shared by few real user applications. HPCchallenge benchmarks provide users with additional information to justify policy and purchasing decisions

Not only does the Japanese Earth Simulator outperform the top American systems on the HPL benchmark (Tflop/s), the differences in bandwidth performance on John McCalpin's STREAM TRIAD benchmark (Level 1 BLAS) shows even greater performance disparity. The Earth Simulator outperforms the ASCI Q by a factor of 4.64 on HPL. Meanwhile, the higher bandwidth memory and interconnect systems of the Earth Simulator are clearly evident as it outperforms ASCI Q by a factor of 36.25 on STREAM TRIAD. In the presentation and paper, we will compare the measured HPCchallenge Benchmark performance on various HPC architectures — from Cray X1s to Beowulf clusters — using the updated results at http://icl.cs.utk.edu/hpcc/hpcc_results.cgi

Even a small percentage of random memory accesses in real applications can significantly affect the overall performance of that application on architectures not designed to minimize or hide memory latency. Memory latency has not kept up with Moore's Law. Moore's Law hypothesizes a 60% compound growth rate per year for microprocessor "performance", while memory latency has been improving at a compound rate of only 7% per year. The memory-processor performance gap has been growing at a rate of over 50% per year since 1980. The HPCchallenge Benchmarks includes a new metric — Giga UPdates per Second — and a new benchmark — RandomAccess — to measure the ability of an architecture to access memory randomly, i.e., with no locality.

GUPS is calculated by identifying the number of memory locations that can be randomly updated in one second, divided by 1 billion (1e9). The term "randomly" means that there is little relationship between one address to be updated and the next, except that they occur in the space of ½ the total system memory. An update is a read-modify-write operation on a table of 64-bit words. An address is generated, the value at that address read from memory, modified by an integer operation (add, and, or, xor) with a literal value, and that new value is written back to memory