EDWARD WALKER

benchmarking Amazon EC2 for high-performance scientific computing

Edward Walker is a Research Scientist with the Texas Advanced Computing Center at the University of Texas at Austin. He received his PhD from the University of York (UK) in 1994, and his research interests include designing fault-tolerant distributed systems, HPC programming languages, and user-centric operating/run-time systems.

ewalker544@gmail.com

Benchmark results can be downloaded from http://www.usenix.org/publications/login/2008-10/benchmark_results.tgz.

HOW EFFECTIVE ARE COMMERCIAL cloud computers for high-performance scientific computing compared to currently available alternatives? I aim to answer a specific instance of this question by examining the performance of Amazon EC2 for high-performance scientific applications. I used macro and micro benchmarks to study the performance of a cluster composed of EC2 high-CPU compute nodes and compared this against the performance of a cluster composed of equivalent processors available to the open scientific research community. My results show a significant performance gap in the examined clusters that system builders, computational scientists, and commercial cloud computing vendors need to be aware of.

The computer industry is at the cusp of an important breakthrough in high-performance computing (HPC) services. Commercial vendors such as IBM, Google, Sun, and Amazon have discovered the monetizing potential of leasing compute time on nodes managed by their global datacenters to customers on the Internet. In particular, since August 2006, Amazon has allowed anyone with a credit card to lease CPUs with their Elastic Compute Cloud (EC2) service. Amazon provides the user with a suite of Web-services tools to request, monitor, and manage any number of virtual machine instances running on physical compute nodes in their datacenters. The leased virtual machine instances provide to the user a highly customizable Linux operating system environment, allowing applications such as Web hosting, distributed data analysis, and scientific simulations to be run. Recently, some large physics experiments such as STAR [1] have also experimented with building virtual-machine-based clusters using Amazon EC2 for scientific computation. However, there is a significant absence of quantitative studies on the suitability of these cloud computers for HPC applications.

It is important to note what this article is not about. This is not an article on the benefits of virtualization or a measurement of its overhead, as this is extensively covered elsewhere [2]. This is also not an article evaluating the counterpart online storage service Amazon S3, although a quantitative study of this is also critical. Finally, this is not an article examining the cost benefits of using cloud computing in IT organizations, as this is amplified elsewhere by its more eloquent advocates [3].

Instead, this article describes my results in using macro and micro benchmarks to examine the "delta" between clusters composed of currently available state-of-the-art CPUs from Amazon EC2 versus clusters available to the HPC scientific community circa 2008. My results were obtained by using the NAS Parallel Benchmarks to measure the performance of these clusters for frequently occurring scientific calculations. Also, since the Message-Passing Interface (MPI) library is an important programming tool used widely in scientific computing, my results demonstrate the MPI performance in these clusters by using the mpptest micro benchmark. The article provides a measurement-based yardstick to complement the often hand-waving nature of expositions concerning cloud computing. As such, I hope it will be of value to system builders and computational scientists across a broad range of disciplines to guide their computational choices, as well as to commercial cloud computing vendors to guide future upgrade opportunities.

Hardware Specifications

In our performance evaluation, we compare the performance of a cluster composed of EC2 compute nodes against an HPC cluster at the National Center for Supercomputing Applications (NCSA) called Abe. For this benchmark study we use the high-CPU extra large instances provided by the EC2 service. A comparison of the hardware specifications of the high-CPU extra large instances and the NCSA cluster used in this study is shown in Table 1. We verified from information in /proc/cpuinfo in the Linux kernel on both clusters that the same processor chip sets were used in our comparison study: dual-socket, quad-core 2.33-GHz Intel Xeon processors.

	EC2 High-CPU Cluster	**NCSA Cluster**
Compute Node	7 GB memory, 4 CPU cores per processor (2.33-GHz Xeon), 8 CPU per node, 64 bits, 1690 GB storage	8 GB memory, 4 CPU cores per processor (2.33-GHz Xeon), 8 CPU per node, 64 bits, 73 GB storage
Network Interconnect	High I/O performance (specific interconnect technology unknown)	Infiniband switch

TABLE 1. HARDWARE SPECIFICATIONS OF EC2 HIGH-CPU INSTANCES AND NCSA ABE CLUSTER.

NAS Parallel Benchmark

The NAS Parallel Benchmarks (NPB) [4] comprise a widely used set of programs designed to evaluate the performance of HPC systems. The core benchmark consists of eight programs: five parallel kernels and three simulated applications. In aggregate, the benchmark suite mimics the critical computation and data movement involved in computational fluid dynamics and other "typical" scientific computation. A summary of the characteristics of the programs for the Class B version of NPB used in this study is shown in Table 2.

The benchmark suite comes in a variety of versions, each using different parallelizing technologies: OpenMP, MPI, HPF, and Java. In this study we use the OpenMP [5] version to measure the performance of the eight-CPU single compute node. We also use the MPI [7] version to characterize the distributed-memory performance of our clusters.

Program	Description	Size	Memory (Mw)
EP	Embarrassingly parallel Monte Carlo kernel to compute the solution of an integral.	230	18
MG	Multigrid kernel to compute the solution of the 3-D Poisson equation.	2563	59
CG	Kernel to compute the smallest eigenvalue of a symmetric positive definite matrix.	75000	97
FT	Kernel to solve a 3-D partial differential equation using an FFT-based method.	512 × 2562	162
IS	Parallel sort kernel based on bucket sort.	225	114
LU	Computational fluid dynamics application using symmetric successive over-relaxation (SSOR).	1023	122
SP	Computational fluid dynamics application using the Beam-Warming approximate factorization method.	1023	22
BT	Computational fluid dynamics application using an implicit solution method.	1023	96

TABLE 2. NPB CLASS B PROGRAM CHARACTERISTICS

NPB-OMP VERSION

We ran the OpenMP version of NPB (NPB3.3-OMP) Class B on a high-CPU extra large instance and on a compute node on the NCSA cluster. Each compute node provides eight CPU cores (from the dual sockets), so we allowed the benchmark to schedule up to eight parallel threads for each benchmark program. On the NCSA cluster and EC2, we compiled the benchmarks using the Intel compiler with the option flags "-openmp -O3."

Figure 1 shows the runtimes of each of the programs in the benchmark. In general we see a performance degradation of approximately 7%–21% for the programs running on the EC2 nodes compared to running them on the NCSA cluster compute node. This percentage degradation is shown in the overlaid line-chart in Figure 1. This is a surprising result; we expected the performance of the compute nodes to be equivalent.

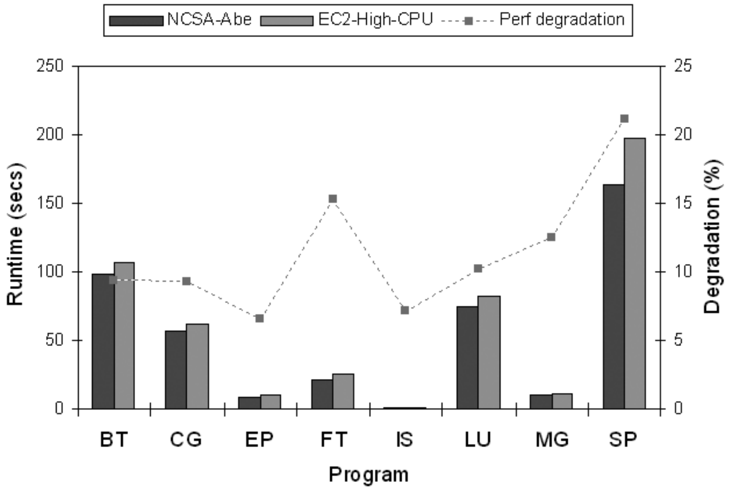

FIGURE 1. NPB-OMP (CLASS B) RUNTIMES ON 8 CPUS ON EC2 AND NCSA CLUSTER COMPUTE NODES. OVERLAID IS THE PERCENTAGE PERFORMANCE DEGRADATION IN THE EC2 RUNS.

NPB-MPI VERSION

We ran the MPI version of NPB (NPB3.3-MPI) Class B on multiple compute nodes on the EC2 provisioned cluster and on the NCSA cluster. For the EC2 provisioned cluster, we requested 4 high-CPU extra large instances, of 8 CPUs each, for each run. On both the EC2 and NCSA cluster compute nodes, the benchmarks were compiled with the Intel compiler with option flag -O3. For the EC2 MPI runs we used the MPICH2 MPI library (1.0.7), and for the NCSA MPI runs we used the MVAPICH2 MPI library (0.9.8p2). All the programs were run with 32 CPUs, except BT and SP, which were run with 16 CPUs.

Figure 2 shows the run times of the benchmark programs. From the results, we see approximately 40%–1000% performance degradation in the EC2 runs compared to the NCSA runs. Greater then 200% performance degradation is seen in the programs CG, FT, IS, IU, and MG. Surprisingly, even EP (embarrassingly parallel), where no message-passing communication is performed during the computation and only a global reduction is performed at the end, exhibits approximately 50% performance degradation in the EC2 run.

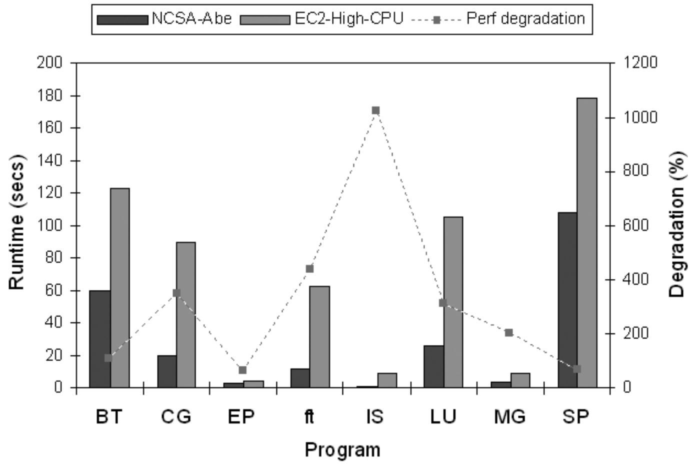

FIGURE 2. NPB-MPI (CLASS B) RUNTIMES ON 32 CPUS ON THE NCSA AND EC2 CLUSTER. BT AND SP WERE RUN WITH 16 CPUS ONLY. OVERLAID IS THE PERCENTAGE DEGRADATION IN THE EC2 RUNS.

We hypothesize that the Infiniband switch fabric in the NCSA cluster is enabling much higher performance for NPB-MPI. However, we want to quantitatively understand the message-passing performance difference between using a scientific cluster with a high-performance networking fabric and a cluster simply composed of Amazon EC2 compute nodes. The following results use the mpptest benchmark [5] to characterize the message-passing performance in the two clusters.

The representative results shown in this article are from the bisection test. In the bisection test, the complete system is divided into two subsystems, and the aggregate latency and bandwidth are measured for different message sizes sent between the two subsystems. In the cases shown, we conducted the bisection test using 32-CPU MPI jobs.

Figures 3 and 4 show the bisection bandwidth and latency, respectively, for MPI message sizes from 0 to 1024 bytes. It is clearly seen that message-passing latencies and bandwidth are an order of magnitude inferior between EC2 compute nodes compared to between compute nodes on the NCSA cluster. Consequently, substantial improvements can be provided to the HPC scientific community if a high-performance network provisioning solution can be devised for this problem.

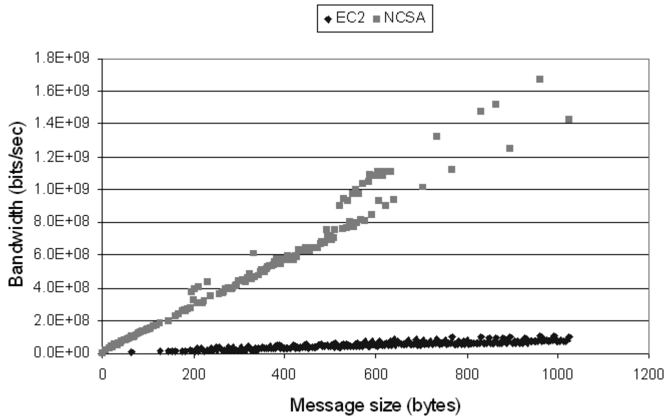

FIGURE 3. MPI BANDWIDTH PERFORMANCE IN THE MPPTEST BENCHMARK ON THE NCSA AND EC2 CLUSTERS

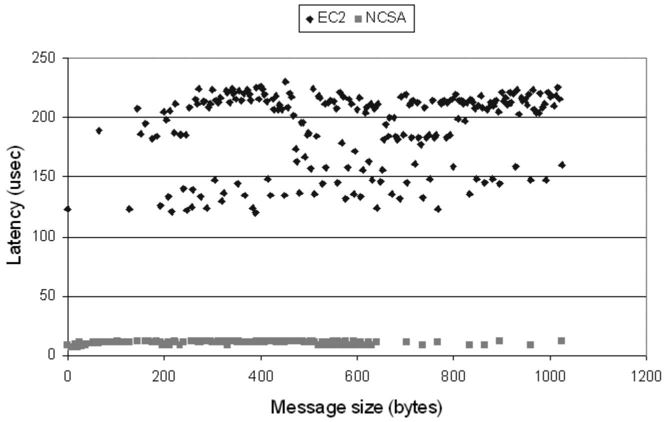

FIGURE 4. MPI LATENCY PERFORMANCE IN THE MPPTEST BENCHMARK ON THE NCSA AND EC2 CLUSTERS

Conclusion

The opportunity of using commercial cloud computing services for HPC is compelling. It unburdens the large majority of computational scientists from maintaining permanent cluster fixtures, and it encourages free open-market competition, allowing researchers to pick the best service based on the price they are willing to pay. However, the delivery of HPC performance with commercial cloud computing services such as Amazon EC2 is not yet mature. This article has shown that a performance gap exists between performing HPC computations on a traditional scientific cluster and on an EC2 provisioned scientific cluster. This performance gap is seen not only in the MPI performance of distributed-memory parallel programs but also in the single compute node OpenMP performance for shared-memory parallel programs. For cloud computing to be a viable alternative for the computational science community, vendors will need to upgrade their service offerings, especially in the area of high-performance network provisioning, to cater to this unique class of users.

REFERENCES

[1] The STAR experiment: http://www.star.bnl.gov/.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the Art of Virtualization,” *ACM Symposium on Operating System Principles*, 2003.

[3] Simson Garfinkel, “Commodity Grid Computing with Amazon’s S3 and EC2”, *;login:*, Feb. 2007.

[4] NAS Parallel Benchmarks: http://www.nas.nasa.gov/Resources/Software/npb.html.

[5] OpenMP specification: http://openmp.org.

[6] Message-Passing Interface (MPI) specification: http://www.mpi-forum.org/.

[7] mpptest—Measuring MPI Performance: http://www-unix.mcs.anl.gov/mpi/mpptest/.