Creating a Framework for Systematic Benchmarking of High Performance Computing Systems

Michael M. Pozulp
The College of William & Mary, Williamsburg, Virginia

This technical report reviews the author's work from May 28th, 2013 through August 9th, 2013 during his internship at NASA Ames Research Center in the NASA Advanced Supercomputing (NAS) Division under the mentorship of Division Chief, Dr. Piyush Mehrotra. The work focuses on providing a general solution to workflow-related problems inherent in the benchmarking work being performed within the Division. A test-framework concept is developed and implemented. The data collected from its use is analyzed for performance regressions.

I. Results: NAS Parallel Benchmarks[1]

Rather than describe the framework, here I highlight the benchmarking that it fostered.

A. NPB Performance Scaling

The constructed framework was used to investigate parallel performance scaling across the four generations of Intel Xeon processors that compose the Pleiades supercomputer system.[2] At the time of this writing, those four generations are (from newest to oldest) Sandybridge, Westmere, Nehalem, and Harpertown. It was hypothesized that as the number of processors increased toward infinity, applications with relatively high communication-intensity would hit a network-imposed performance ceiling and nullify any performance gains resultant from hardware improvements in the newer generations of Intel Xeon processors. If this were true, then there is the opportunity to exploit arbitrage in processor pricing. The purchase of cheaper hardware that performs just as well as more expensive hardware can increase aggregate performance per dollar or allow for portions of the budget to be re-allocated to network technologies focused on raising the performance ceiling. As expected, stalling of performance increases was observed simultaneously for all four generations at 512 processes, but the performance did not converge to a common value. Rather, using data collected on the NPB conjugate-gradient (CG) benchmark, it is apparent that performance remained better for newer generations, and worse for older (Figure 1). Although the results contradict the hypothesis, there are many more applications and parameters to investigate. Future studies should also seek to quantify communication intensity in a more rigorous manner.

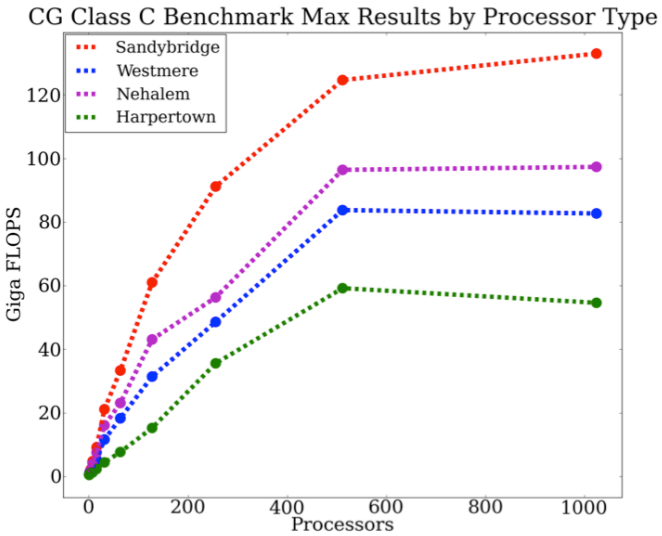

Figure 1. CG benchmark Class C scaling results by processor type.

A. NPB Performance Variation

In addition to performance scaling, the framework has been used to analyze substantial performance variation in the NPB applications run on Pleiades. Data collected during the investigation of NPB performance scaling was leveraged by the framework's plotting capabilities to display scatterplots of the results and visually reveal the variation. The effects of the variation were enough that looking at the mean or median observation for processor scaling changed the ordinal ranking of generation performance when compared to plots using the maximum observed performance. The variability is of particular interest for the CG benchmark considering the recent attention it has received in the HPC community as a result of its consideration as an alternative to HPL for benchmarking the Top 500 Supercomputers.[3]

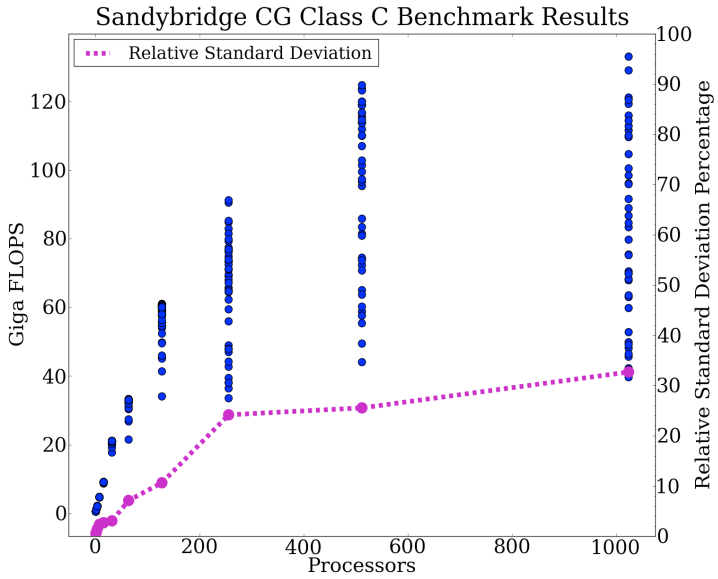

Figure 2. CG benchmark Class C results on Sandybridge processors

It is hypothesized that for each particular NPB, the extent of the variation corresponds to the communication intensity of the application kernel. The corollary following this hypothesis is that there must be a quantifiable correlation between performance degradation and the logical network distance that communicated data must traverse. From data collected within the framework, one may observe the relative standard deviation, increasing along with the number of processors (Figure 2). Ultimately, the hope is that in cases of communication-intensive applications we may optimize performance by taking care to allocate contiguous nodes when scheduling the job. This would mitigate, but not solve the problem. We need a dynamic job scheduler or some tool that can freeze jobs running on a system and reorganize them into more clustered and less-fragmented node groupings. HTCondor[4], a shared-memory solution, is a successful implementation of the latter that has yet to be implemented for distributed-memory clusters. Chant[5] is an experimental solution affecting the distributed-memory case that needs more research.

Data from the framework is being used to map the logical distances of nodes running jobs to performance results in hopes of quantifying the degradation effects.

Acknowledgments

Work performed under the expert direction of Mentor and Division Chief Dr. Piyush Mehrotra, APP Group members Henry Jin, Bob Hood, Johnny Chang and help from many others in the NASA Advanced Supercomputing (NAS) Division at NASA Ames Research Center.

References

[1]Bailey, *et al.*, "The NAS Parallel Benchmarks" *NAS Technical Report Database* [online database], URL: **http://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-007.pdf** [cited 2 August 2013].

[2]Pleiades Supercomputer "**http://www.nas.nasa.gov/hecc/resources/pleiades.html**" [cited 2 August 2013].

[3]Heroux, M. A., and Dongarra, J., "Toward a New Metric for Ranking High Performance Computing Systems" *Netlib* [online database], URL: **http://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-007.pdf** [cited 2 August 2013].

[4]Thain, D., Tannenbaum, T., and Livny, M., "Distributed Computing in Practice: The Condor Experience" *University of Winsconsin* [online database], URL: **http://research.cs.wisc.edu/htcondor/doc/condor-practice.pdf** [cited 2 August 2013].

[5]Haines, M., Cronk, D., and Mehrotra, P., "On the Design of Chant: A Talking Threads Package" *IEEE Xplore* [online database], URL: **http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=344298** [cited 2 August 2013].