Automated Performance Comparison

Joachim Worringen

C&C Research Laboratories, NEC Europe Ltd.
http://www.ccrl-nece.de, worringen@ccrl-nece.de

Keywords: *benchmark, performance comparison, perfbase, test automation*

1 Motivation

Comparing the performance of different HPC platforms with different hardware, MPI libraries, compilers or sets of runtime options is a frequent task for implementors and users as well. Comparisons based on just a few numbers gained from a single execution of one benchmark or application are of very limited value as soon as the system is to run not only this software in exactly this configuration. However, the amount of data produced for thorough comparisons across a multi-dimensional parameter space quickly becomes hard to manage, and the relevant performance differences hard to locate. We deployed *perfbase* [3] as a system to perform performance comparisons based on a large number of test results yet being able to immediately recognize relevant performance differences.

2 Automation of Performance Comparison

perfbase is a toolkit that allows to import, manage, process, analyze and visualize arbitrary benchmark or application output for performance analysis or other purposes like correctness verification. Its core concept is based on defining an *experiment* with *parameter and result values*, import data for different *runs* of the experiment from arbitrarily formatted text files, and perform queries to process, analyze and visualize the data. We have created a set of shell scripts and *perfbase* experiment, input and query descriptions that make up the framework for automated performance comparison. With this framework, only four simple steps are required to produce a thorough comparison:

1. Define the range of parameters for execution (i.e. number or nodes or processes) in the *job creation script*.
2. Execute the *job creation script*, then the *job submission script*. Wait for completion of the jobs.
3. Run the *import script* which uses *perfbase* to extract relevant data from the result files and store it in the *perfbase* experiment.
4. Run the *analysis script* which issues *perfbase* queries to produce the performance comparison. Changing parameters in the analysis script allows to modify the comparison result.

Two examples will illustrate the application of this framework to a single micro-benchmark (*Intel MPI Benchmark*) or a suite of application kernel benchmarks (*NAS Parallel Benchmarks*).

2.1 Intel MPI Benchmark

The *Intel MPI Benchmark* [2] is a well-known and widely used MPI micro benchmark which measures the performance of individual MPI collective communication operations and point-to-point communication patterns.

A single run of this benchmark with 64 processes will perform 80 tests with 24 data sizes each. For each data size, between 1 and 3 latencies are reported, resulting in more than 5000 data points. We report a single line like `Exchange: A better than B for 62% of the chunk sizes: avg 120%, stddev 9` for a test where variant A is on average (over the data sizes) 120% faster than B with a standard deviation of 9. Only differences beyond a definable threshold are listed. Next to this compact report, a large number of plots showing absolute and relative performance is generated for reference.

2.2 NAS Parallel Benchmarks

The *NAS Parallel Benchmarks* [1] are an established set of application kernels often used for performance evaluation. The execution of the NPB can be varied across the kernel type, data size and number of processes. Together with the variation of the component to be evaluated and recommended multiple executions, a large number of result data (performance in MFLOPS) is generated. From this data, we generate a report consisting of a table for each kernel with rows like `C 64 4 6.78`. In this case, the 64 process, 4 processes per node execution of the corresponding kernel for data size `C` delivered 6.79% more performance with variant A than with variant B. The data presented in the tables is also visualized using bar charts.

3 Conclusion

The application of the *perfbase* toolkit allows to thorougly but still conveniently compare benchmark runs performed in two different environments. The important features are the management of a large number of test runs combined with the filtering of non-relevant differences. This allows to actually do in-depth comparisons based on a large variety of tests. The framework can easily be applied to other benchmarks. The *perfbase* toolkit is open-source software available at `http://perfbase.tigris.org` and includes the scripts and experiments described in this paper.

References

1. D. H. Bailey et al. The nas parallel benchmarks. *The International Journal of Supercomputer Applications*, 5(3):63–73, Fall 1991.
2. N.N. *Intel MPI Benchmarks: Users Guide and Methodolgy Description*. Intel GmbH, `http://www.intel.com`, 2004.
3. J. Worringen. Experiment management and analysis with perfbase. In *IEEE Cluster 2005*, `http://www.ccrl-nece.de` (Publication Database), 2005. IEEE Computer Society.