Automated Performance Comparison  
Joachim Worringen  
C&C Research Labs, NEC Europe Ltd.*

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 multidimensional parameter space quickly becomes hard to manage, and the relevant performance differences hard to localize. We wanted to create an environment that makes the execution of the benchmarks and the result data processing simple an automated routine job. Additionally, it should support and accelerate the analysis by presenting only relevant data points, while at the same time present all gathered data in clear, well laid out charts.

Framework for Automation of Performance Comparison

We created a framework for the automation of performance comparison which is based on shell scripts and perfbase XML experiment, input and query descriptions:
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 database.
4. Run the analysis script which issues perfbase queries to perform the comparison.

Changing parameters in the analysis script allows to modify the comparison result.

The details of the scripts need to adapt to the benchmark and environment that is used:
• job creation script needs to consider the benchmark and job submission parameters
• job submission script has to be adapted to the used queuing system
• import script needs to use a matching XML input description
• Analysis script needs to use a specific XML query description to show the desired information

This means, the basic XML descriptions need to be adapted once to the benchmark, and the scripts once for every job submission environment. Using the provided examples and templates, this is a relatively simple task. Typically, the benchmarks and environments don’t change frequently and don’t differ significantly.

Intel Message Passing Benchmark

The Intel MPI Benchmark (IMB) is a well-known and widely used MPI micro benchmark which measures the performance of individual MPI point-to-point communication patterns and collective communication operations. A single run of this benchmark with 64 processes will perform 80 tests with 32 data sizes each. For each data size, between 1 and 3 latencies are reported, resulting in more than 5000 data points. This amount of data can hardly be analyzed manually. Instead, we define a threshold for results that are considered as differing. Only for these cases, we report a single line with the key information like percentage of data points being different, the average difference and the standard deviation. The full range plots showing absolute and relative performance is generated as well and can be analyzed based upon the summary report.

NAS Parallel Benchmarks

The NAS Parallel Benchmarks 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.

perfbase Toolkit for Experiment Management and Analysis

Perfbase is toolkit for experiment management and analysis, which helps to maintain control of the output data of arbitrary benchmarks, applications or test suites. It is set of Python scripts (about 15k lines of code) that interact with the user as a frontend. The user passes information via command line parameters and XML description files. At the backend, a PostgreSQL database server is used for data storage and retrieval.

Perfbase employs a workflow that closely adheres to the typical tasks that a researcher, developer or engineer has to perform whenever he evaluates a system using (a set of) benchmarks or applications:
• Define an experiment that defines the parameters and result values for the evaluation.
• Perform (run) the experiments and gather the output data in files
• Import the data into the perfbase database using an input description, creating ‘runs’.
• Define and perform queries upon the runs to retrieve relevant information.
• If necessary, add or change parameter or result value definitions, import more data etc.

For data import, arbitrarily formatted text files can be used, including tables, XML or anything else. Query output can be text files with data columns, gnuplot input files, or XML.