A Tool for Cluster Performance Tuning and Optimization: Beowulf Performance Suite

Douglas Eadline, Ph.D.

17th March 2003

deadline@basement-supercomputing.com

Abstract

A collection of cluster performance tools has been integrated into a single package. The test suite measure various aspects of Beowulf Clusters. A description of the test suite as well as installation and usage information is provided. Results for tests can be displayed in HTML.

Background

The Beowulf Performance Suite (BPS) was designed to provide a comprehensive and comparative way of measuring cluster performance. Although BPS contains many benchmarking programs, BPS is not designed to benchmark clusters. BPS is designed as an analysis tool to measure differences due to hardware or software changes on the same cluster. In addition, successfully running all the tests provides some assurances that the cluster is configured properly. The suite provides a graphical user interface for running the programs as well as generating HTML output files. The use of HTML make it trivial to share your results with others on the web. The following tests are available:

- bonnie - hard drive performance
- stream - memory performance
- netperf - general network performance
- netpipe - detailed network performance
- nas - nas NASA parallel tests
- unixbench - general Unix benchmarks
- lmbench - micro Linux benchmarks

In addition, all the tests are are "open source" and the tar files are included in the binary RPM. Some of the tests are are already compiled, while others are built when the test runs. It should be emphasized that the intent of BPS is not to try an obtain optimum numbers for your cluster, but rather generate a baseline on which to measure the effects of changes on your cluster.

Some users have questioned why the LINPACK benchmark is not included in the suite. While the LINPACK is a good measure of performance, my main concern was that the test suite would be used as a "my cluster is better than your cluster" set of tests due to the fact that LINPACK is used to rate the top supercomputers. My goal was to provide a set of tools to measure performance increases (or decreases) when things are changed.

Obtaining/Installing BPS

You may download the bps rpm or tar file from http://www.plogic.com/bps. BPS does require pygtk, python perl, gnuplot, and expect. These packages are normally found in most distributions (i.e. Red-Hat 7.3) or otherwise easily added. These specific versions required to build and run this version of the package are (other versions may work):

```
pygtk-0.6.9-3.i386.rpm
perl-5.6.1-34.99.6.i386.rpm
python-1.5.2-38.i386.rpm
gnuplot-3.7.1-17.i386.rpm
expect-5.32.2-67.i386.rpm
```

Running in X windows

The BPS provides an X windows interface. The interface can be started by executing the following command:

```
# xbps
```

The desired set of benchmark(s) can then be selected from the list. Note, Netpipe and Netperf require information before they will run (see settings). The selected benchmarks can then be run by pressing the `Run selected benchmarks` button. In addition, the NAS suite will only run on one node unless other settings are provided under settings. Unless otherwise specified, results are placed in `~/bps-logs directory`. After your benches have run, an HTML file of the results can be produced by clicking on the `Create html file of results` button. This will create a file called index.html in the logs directory which you can open with your browser.

Settings

Using the settings is the better way to run the benchmarks, and in the case of Netpipe and Netperf is necessary. The settings, however, are not much more complex than running the benches.

- Log Directory - Sets the directory your bench logs will be placed into. Also, when creating an html file from the results, you must create the html file in the same directory as your log directory. Otherwise the program will not be able to find the files it needs to create the page.
- Netperf and Netpipe - The settings for both of these look the same but are distinct, as Netpipe is different than Netperf. You must provide a valid receiver and sender node in both cases. The receiver node setting is the node that will be running the bench as a receiver, and the sender node is the one that will be running it as a sender. Since these benches run as benchmarks for the network communications between the two nodes, it is necessary to have both nodes specified. The interface is determined by the names of the hosts.
- NAS - The default NAS settings are shown when you choose this option. The default settings will run the NAS suite on a single node for the S class size (i.e. workstation). However, if you have two or more nodes/processors, you can add them in the machine list. Because compilation is a part of the tests NAS runs, you can specify which compiler you want it to use. You should also select the MPI version and number of processors you have.
- Machine Info - This is standard information about your motherboard,memory, type of network interface, and the distribution of Linux you are using.
- Prompting - This turns window prompts on/off.

Finding out more about the benchmarks

In the Help menu, you can find a one sentence synopsis of what each benchmark does by selecting Benchmark Info.

Running in Text Mode

To run bps in text mode enter:

```
# bps
```

The following command line options are available:

```
# bps -h
Usage: /usr/bps/bin/bps <OPTIONS>
Options:
-b bonnie++
-s stream
-f <send node>,<receive node> netperf to remote node
-p <send node>,<receive node> netpipe to remote node
-n <compiler>,<#processors), NAS parallel benchmarks
<test size>,<MPI>, compiler={gnu,pgi,Intel}
<machine1,machine2,...> test size={A,B,C,dummy}
MPI={mpich,lam,mpipro}
-k keep NAS directory when finished
-u unixbench
-m lmbench
-l <log_dir> benchmark log directory
-w preserve existing log directory
-i <mboard manufacturer>, machine information
   <mboard model>,<memory>,
   <interconnect>,<linux ver>
-v show version
-h show this help
```

If you are running the NAS tests from the bps script, you may tell it to keep "`-k`" the NAS directory (npb) in your log directory. This can be useful in tracking down compile errors (see the ~npb/make.log). HTML output files can be produced from the results files in <log directory> by executing:

```
# bps-html <log directory>
```

Important Notes

- All tests are archived in the src directory.
- The bps suite is best run as a normal user, not root. Some of the tests (i.e. NAS parallel) will not run as root.
- Not all features of the command line interface are possible with the GUI.
- When using Netpipe and Netperf Benchmarks, rsh with no password must be permitted between the nodes.
- Under normal operation, xbps will always preserve the existing log directory. This feature is to ensure previous results will not be overwritten. You can copy previous log files (from log directories) into the current log directory for bps-html conversion.

- Also, the tests have been designed so that the BPS rpm only needs to be installed on the master node. For this to work, the BPS log directory must be mounted on all nodes (e.g. under /home).
- NAS Parallel Benchmarks have been tested with Paralogic's versions of LAM and MPICH. Please see the NAS documentation for more information. Rather than limit potential BPS users, these are not made a part of the required packages list. The benchmark scripts have been written to rely on the three environment variables (`LAM_HOME` for LAM-MPI, `MPICH_HOME` for MPICH, and `MPIPRO_HOME` for MPI-PRO, and MPICH). If you are having problems with the NAS benchmarks, extract the npb.tar.gz archive in the/usr/bps/src directory and try running the scripts manually (see below). Consult the README.plogic file for more information. Also, if you wish to use the Portland Group or the Intel Compilers make sure they are properly configured. Note: on Paralogic clusters, these tests are designed to use modules for easy transition from one MPI/Compiler version to another.

In Case of Problems

The BPS suite is a collection of many tests. You should have minimal or no problems with the single machine tests. As more machines become involved with the tests, there is room for more configuration errors to arise. If a test does not run, check the test_name.log file in the log directory. In the case of the NAS tests, the results are in the form npb.COMPILER.MPI.CLASS.PROCESSORS.

In general, if you have problems with a test it may be best to run it from the command line. **In the case of the NAS suite, the -k option will keep the npb directory in the log directory so you can run the tests more directly by using the run_suite script in the npb directory.** Also the `README.plogic` file in the npb directly should provide more information on how the tests are run and how to resolve possible problems.

NASA Benchmark Suite (details)

The NAS suite will probably produce the most problems for end users. The main script `run_suite` is designed to "wrap" around and hide the various issues with the different MPI's and compilers. While `run_suite` does an adequate job, it certainly can not predict the potential software environments on a cluster. The NAS suite can be run from the command line. The following options are required. You also need to list the machine names in npb/cluster/machines file (one per line). This file is used by MPI to start your programs. In the npb directory, run

```
# run_suite -h
Usage: ./run_suite <OPTIONS>
```

```
Options:
run_suite,v 0.3 2/12/2002
Usage: ./run_suite <OPTIONS>
Options:
-v verbose output from make stage (default=make.log)
-c <compiler> compiler (gnu/pgi/intel)
-n <processors> number of processors
-t <test> test size (A,B,C,S)
-m <mpi> mpi version(lam,mpich,mpipro,dummy)
-o only build programs
-h show this help
To run on a single CPU use: '-c gnu -n 1 -t S -m dummy'
```

If you have problems producing the binary files, consult the `make.log` file for a complete listing for the make process. Currently the test suite has been tested on Paralogic clusters using GNU, Portland Group, and Intel compilers. It has been tested and works with LAM-MPI, MPICH, and MPI/PRO.

Description of the NAS tests.

BT is a simulated CFD application that uses an implicit algorithm to solve 3dimensional (3D) compressible NavierStokes equations. The finite differences solution to the problem is based on an Alternating Direction Implicit (ADI) approximate factorization that decouples the x, y, and z dimensions. The resulting systems are BlockTridiagonal of 5x5 blocks and are solved sequentially along each dimension.

SP is a simulated CFD application that has a similar structure to BT. The finite differences solution to the problem is based on a BeamWarming approximate factorization that decouples the x, y, and z dimensions. The resulting system has scalar Pentadiagonal bands of linear equations that are solved sequentially along each dimension.

LU is a simulated CFD application that uses symmetric successive overrelaxation (SSOR) method to solve a seven block diagonal system resulting from finite difference discretization of the NavierStokes equations in 3D by splitting to into block Lower and Upper triangular systems.

FT contains the computational kernel of a 3D fast Fourier Transform (FFT)based spectral method. FT performs three one dimensional (1D) FFT's, one for each dimension.

MG uses a Vcycle MultiGrid method to compute the solution of the 3D scalar Poisson equation. The algorithm works continuously on a set of grids that are made between coarse and fine. It tests both short and long distance data movement.

CG uses a Conjugate Gradient method to compute an approximation to the smallest eigenvalue of a large, sparse, unstructured matrix. This kernel tests unstructured grid computations and communications by using a matrix with randomly generated locations of entries.

EP is an Embarrassingly Parallel benchmark. It generates pairs of Gaussian random deviates according to a specific scheme. The goal is to establish the reference point for peak performance of a given platform.

Additional benchmark information:

- General Information http://www.plogic.com/bps
- bonnie++ hard drive performance - http://www.coker.com.au/bonnie++
- stream memory performance - http://www.cs.virginia.edu/stream
- netperf general network performance http://www.netperf.org/netperf/NetperfPage.html
- netpipe detailed network performance http://www.scl.ameslab.gov/Projects/ClusterCookbook/nprun.html
- unixbench general Unix benchmarks http://www.linuxdoc.org/HOWTO/Benchmarking-HOWTO.html
- LMbench low level benchmarks http://www.bitmover.com/lmbench
- NAS Parallel tests http://www.nas.nasa.gov/Software/NPB

Acknowledgments

I wish to acknowledge all the authors of the tests suites used in this package.

Copyright

Copyright (c) 2003, Douglas Eadline, All rights Reserved. This document maybe distributed under the GPL Free Documentation License http://www.gnu.org/licenses/licenses.html#FDL.