Third Party Software: How do I run the HPL test on a Bright Cluster?

How do I run the HPL test on a Bright Cluster?

What is HPL?

Slightly rewording from http://www.netlib.org/benchmark/hpl/:

***HPL** is a software package that solves a **random, dense linear** system of equations with N unknowns, in double-precision arithmetic on distributed-memory computers. It then prints the estimated performance of the distributed-memory computer as the ratio between the number of effective floating point operations divided by the amount of time in seconds it took to perform those operations.*

It can thus be regarded as a portable as well as freely available implementation of the High Performance Computing LINPACK Benchmark.

The algorithm

- Two-dimensional block-cyclic data distribution
- Right-looking variant of the LU factorization with row partial pivoting featuring multiple look-ahead depths
- Recursive panel factorization with pivot search and column broadcast combined
- Various virtual panel broadcast topologies
    - bandwidth reducing swap-broadcast algorithm
    - backward substitution with look-ahead of depth 1

The HPL algorithm is changeable and can be tuned via 17 parameters:

N	Problem size	**Pmap**	Process mapping
NB	Blocking factor	**threshold**	for matrix validity test
P	Rows in process grid	**Ndiv**	Panels in recursion
Q	Columns in process grid	**Nbmin**	Recursion stopping criteria
Depth	Lookahead depth	**Swap**	Swap algorithm
Bcasts	Panel broadcasting method	**L1, U**	to store triangle of panel
Pfacts	Panel factorization method	**Align**	Memory alignment
Rfacts	Recursive factorization method	**Equilibration**	

The parameters in the red cells are the most important tuning parameters, while the ones in the green cells are of somewhat lesser importance. The parameter space is huge and exploring the full parameter space is inefficient as there are more than a billion different combinations. If one considers only combination of the most important parameters the parameter space reduces to 50,000 combinations.

Third Party Software: How do I run the HPL test on a Bright Cluster?

There are several algorithms that aim at cutting the amount of time required for tuning, but they are beyond the scope of this article. However, it is worth mentioning that one does not have to use the maximal problem size in tuning. Early termination can be used to cut down the time, and the fact that tuning algorithms perform well and seem to be robust suggest that for a given architecture the amount of important parameters might be smaller than suggested above.

The tuning parameters need to be stored in a file named HPL.dat.

Setting the values of HPL.dat parameters

The problem size N should be set to a large value but not too large. The amount of computations scales as the cube of that number which means that doubling N requires eight times more computations. Values of the order of 1000-10,000 are reasonable for modern systems. Make sure that there is no paging occurring while executing the problem by monitoring the `vmstat` command. A useful formula to estimate the problem size is:

N = sqrt (0.75 * Number of Nodes * Minimum memory of any node / 8)

HPL uses the block size NB for the data distribution as well as for the computational granularity. Small values are better from a load balancing point of view, but too small values might limit performance. A value in the range [32,...,256] is good for modern systems with large values being more efficient for larger problem sizes.

The process decomposition should be roughly square with P greater than or equal to Q. The product PxQ should obviously match the number of processors/cores that are going to be used for HPL. The most important factor that affects the selection of this parameter is the type of physical network with which the nodes of the cluster are interconnected. Flat grids like 4x1, 8x1, 4x2, etc. are good for ethernet-based networks.

The Rfact and Pfact values need experimentation, but for very large clusters both parameters can be set to the right-looking part of the LU decomposition.

Once the panel factorization has been computed, this panel of columns is broadcast to the other process columns. There are many possible broadcast algorithms, and the software currently offers 6 variants to choose from. A brief note on some of these:

- **Algorithm 2**: Increasing-ring (modified) is the usually the best choice
- **Algorithm 4**: Increasing-2-ring (modified) is usually a second choice.
- **Algorithm 5**: Long (bandwidth reducing) - this can be a good choice for systems with very fast nodes, but a very slow network.

A final recommendation is to ensure that HPL is not going to use more than 80% of the memory available to that system

There are also a couple of web-based HPL.dat generators like the following:

Third Party Software: How do I run the HPL test on a Bright Cluster?

http://www.advancedclustering.com/faq/how-do-i-tune-my-hpldat-file.html

A sample HPL.dat file

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
58464          Ns
1            # of NBs
168            NBs
0            PMAP process mapping (0=Row-,1=Column-major)
1            # of process grids (P x Q)
4            Ps
8            Qs
16.0         threshold
1            # of panel fact
2            PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
4            NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
1            RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
1            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

Prerequisites

HPL requires a distributed-memory system with a tuned implementation of the BLAS library and a working implementation of the MPI standard. The *Bright Cluster Manager* provides several modules of the OpenBLAS library tuned for different architectures as well as a general-purpose module (openblas/dynamic) that attempts to auto-detect the relevant architecture and use routines optimized for that particular system.

Third Party Software: How do I run the HPL test on a Bright Cluster?

The relevant modules can be loaded with the command below:

```
[user@myHeadNode ~]$ module load shared openmpi/gcc openblas/dynamic  hpl
```

The other available tuned versions of OpenBLAS are:

- openblas/bulldozer

- openblas/istanbul

- openblas/nehalem

- openblas/sandybridge

Note: OpenBLAS is available in Bright 6.1. Users of Bright 5.2 and 6.0 can choose between gotoBLAS and BLAS. In the latter case the HPL binary is called xhpl.blas.

Running HPL on a Bright cluster

To run HPL after loading the modules, the user needs to issue the command:

```
mpiexec -machinefile nodes  xhpl | tee HPL.out
```

If your cluster has InfiniBand, and you want to see the performance difference between InfiniBand and Ethernet, you can do a run over Ethernet by issuing the following command:

```
mpiexec -machinefile nodes  --mca btl ^openib  xhpl | tee HPL.out
```

Sample output:

```
================================================================================

HPLinpack 2.1  --  High-Performance Linpack benchmark  --   October 26, 2012

Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK

Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK

Modified by Julien Langou, University of Colorado Denver

================================================================================



An explanation of the input/output parameters follows:

T/V    : Wall time / encoded variant.

N      : The order of the coefficient matrix A.

NB     : The partitioning blocking factor.

P      : The number of process rows.

Q      : The number of process columns.

Time   : Time in seconds to solve the linear system.

Gflops : Rate of execution for solving the linear system.



The following parameter values will be used:



N      :   79232

NB     :     128

PMAP   : Row-major process mapping
```

```
P     :       4

Q     :       4

PFACT  :   Right

NBMIN  :       4

NDIV   :       2

RFACT  :   Crout

BCAST  :  1ringM

DEPTH  :       1

SWAP   : Mix (threshold = 64)

L1     : transposed form

U      : transposed form

EQUIL  : yes

ALIGN  : 8 double precision words



--------------------------------------------------------------------------------



- The matrix A is randomly generated for each test.

- The following scaled residual check will be computed:

      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )

- The relative machine precision (eps) is taken to be               1.110223e-16

- Computational tests pass if scaled residuals are less than                16.0



================================================================================

T/V                N    NB     P     Q               Time                 Gflops

--------------------------------------------------------------------------------
```

Third Party Software: How do I run the HPL test on a Bright Cluster?

```
WR11C2R4       79232   128     4     4            2311.44              1.435e+02

HPL_pdgesv() start time Wed Jan  2 21:53:09 2013



HPL_pdgesv() end time   Wed Jan  2 22:31:40 2013



--------------------------------------------------------------------------------

||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0034949 ..... PASSED

================================================================================



Finished      1 tests with the following results:

              1 tests completed and passed residual checks,

              0 tests completed and failed residual checks,

              0 tests skipped because of illegal input values.

--------------------------------------------------------------------------------



End of Tests.

================================================================================
```

Unique solution ID: #1124
Author: Panos Labropoulos
Last update: 2015-07-23 09:56