**HPL Benchmarking**

Install mpich on all the machines to be benchmarked

1. Install ATLAS

**#yum install atlas**

2. Download the hpl-xx.tar.gz file

Untar it

3. **cd hpl-xx/setup/**

In setup there is list of arch file available

Make.FreeBSD\_PIV\_CBLAS Make.IRIX\_FBLAS Make.Linux\_Intel64 Make.Linux\_PII\_FBLAS\_gm Make.PWR2\_FBLAS

Make.SUN4SOL2-g\_FBLAS Make.Tru64\_FBLAS\_elan make\_generic Make.Linux\_ATHLON\_CBLAS Make.Linux\_PII\_CBLAS Make.Linux\_PII\_VSIPL Make.PWR3\_FBLAS Make.SUN4SOL2-g\_VSIPL Make.UNKNOWN.in Make.HPUX\_FBLAS Make.Linux\_ATHLON\_FBLAS Make.Linux\_PII\_CBLAS\_gm Make.Linux\_PII\_VSIPL\_gm Make.PWRPC\_FBLAS Make.T3E\_FBLAS Make.I860\_FBLAS Make.Linux\_ATHLON\_VSIPL Make.Linux\_PII\_FBLAS Make.MacOSX\_Accelerate Make.SUN4SOL2\_FBLAS Make.Tru64\_FBLAS

4. cp the Make.Linux\_PII\_CBLAS to hpl home

5. Edit the make.Linux\_PII\_CBLAS

# ----------------------------------------------------------------------

# - HPL Directory Structure / HPL library ------------------------------

# ----------------------------------------------------------------------

TOPdir = path-where-the-hpl-is-present (ex: /root/hpl-2.3)

# ----------------------------------------------------------------------

# - Message Passing library (MPI) --------------------------------------

# ----------------------------------------------------------------------

MPdir = path-of-mpich (ex:/usr/lib64/mpich)

mpinc = -I$(MPdir)/include

MPlib = $(MPdir)/lib/libmpi.so.12

# ----------------------------------------------------------------------

# - Linear Algebra library (BLAS or VSIPL) -----------------------------

# ----------------------------------------------------------------------

LAdir = /usr/lib64/atlas

LAinc =

LAlib = $(LAdir)/libsatlas.so.3 $(LAdir)/libtatlas.so.3

# ----------------------------------------------------------------------

# - Compilers / linkers - Optimization flags ---------------------------

#----------------------------------------------------------------------

CC = /root/mpich/bin/mpicc

LINKER = /root/mpich/bin/mpicc

6. make the Arch file edited

**make arch=Linux.PII\_BLAS**

After having built the executable hpl/bin/<arch>/xhpl, modify the input data file HPL.dat.

This file should reside in the same directory as the executable hpl/bin/<arch>/xhpl.

This will create hpl binary file(xhpl) and HPL.dat file in /hpl-2.3/bin/Linux\_PII\_BLAS/

7. Tuning of the HPL.dat file

**vi /hpl-2.3/bin/Linux\_PII\_BLAS/HPL.dat**

This file contains information about the problem sizes, machine configuration, and algorithm features to be used by the executable

Description of the HPL.dat

File

**Line 1:** (unused) Typically one would use this line for its own good. For example, it could be used to summarize the content of the input file. By default this line reads: HPL Linpack benchmark input file

**Line 2:** (unused) same as line 1. By default this line reads: Innovative Computing Laboratory, University of Tennessee

**Line 3:** the user can choose where the output should be redirected to. In the case of a file, a name is necessary, and this is the line where one wants to specify it. Only the first name on this line is significant. By default, the line reads:HPL.out output file name (if any). This means that if one chooses to redirect the output to a file, the file will be called "HPL.out". The rest of the line is unused, and this space to put some informative comment on the meaning of this line.

**Line 4:** This line specifies where the output should go. The line is formatted, it must begin with a positive integer, the rest is insignificant. 3 choices are possible for the positive integer, 6 means that the output will go the standard output, 7 means that the output will go to the standard error. Any other integer means that the output should be redirected to a file, which name has been specified in the line above.

This line by default reads:

6 device out (6=stdout,7=stderr,file)

which means that the output generated by the executable should be redirected to the standard output.

**Line 5:** This line specifies the number of problem sizes to be executed. This number should be less than or equal to 20. The first integer is significant, the rest is ignored.

If the line reads:3 # of problems sizes (N) this means that the user is willing to run 3 problem sizes that will be specified in the next line.

**Line 6:** This line specifies the problem sizes one wants to run. Assuming the line above started with 3, the 3 first positive integers are significant, the rest is ignored.

For example:3000 6000 10000 Ns means that one wants xhpl to run 3 (specified in line 5) problem sizes, namely 3000, 6000 and 10000.

**Line 7:**

This line specifies the number of block sizes to be run. This number should be less than or equal to 20. The first integer is significant, the rest is ignored. If the line reads:

5 # of NBs this means that the user is willing to use 5 block sizes that will be specified in the next line.

**Line 8:**

This line specifies the block sizes one wants to run.

Assuming the line above started with 5, the 5 first positive integers are significant, the rest is ignored.

For example:80 100 120 140 160 NBs means that one wants xhpl to use 5 (specified in line 7) block sizes, namely 80, 100, 120, 140 and 160.

"good" block sizes are almost always in the [32 .. 256] interval

subset of N to distribute across nodes

“N”, which is the size of your problem (double precision 8 bytes)

**N = sqrt((Memory Size in Gbytes \* 1024 \* 1024 \* 1024 \* Number of Nodes) /8) \* 0.80**

“NB”, which is the block size in the grid

Note : Usually block sizes giving good results are within the [96,104,112,120,128, …, 256] range

N should be NB aligned

**Line 10:** This line specifies the number of process grid to be runned.

This number should be less than or equal to 20.

The first integer is significant, the rest is ignored.

If the line reads:2 # of process grids (P x Q) this means that you are willing to try 2 process grid sizes that will be specified in the next line.

**Line 11-12:**

2 Ps

8 Qs

(P \* Q) is the size of your grid which is equal to the number of processors your cluster has. In the 512 nodes cluster input file, the P and Q product is 4096 which is basically the number of processors we have (512\*8).

When picking these two numbers, try to make your grid as “square” as possible in shape. The HPL website mentions that best practice is to have it close to being a “square”, thus P and Q should be approximately equal, with Q slightly larger than P.

8. Running the benchmark

**mpirun -np 10 ./xhpl**

Sample of HPL.dat file

