https://ulhpc-tutorials.readthedocs.io/en/latest/parallel/mpi/HPL/

Wget <https://netlib.org/benchmark/hpl/hpl-2.3.tar.gz>

wget https://github.com/xianyi/OpenBLAS/archive/refs/tags/v0.3.23.tar.gz

tar –xvf v0.3.23.tar.gz

cd OpenBLAS-0.3.23/

make -j32

make -j32 PREFIX=/app/openBLAS install

>>>check the openBLAS

cd /app/openBLAS

vim /root/.bashrc

export LAdir = /app/openBLAS

export LAinc =$(LAdir)/include/

export LAlib =$(LAdir)/lib/

cp -r Make.UNKNOWN.in /opt/hpl-2.3/

mv Make.UNKNOWN.in Make.Linux-AMD

vim Make.Linux-AMD

SHELL = /bin/sh

CD = cd

CP = cp

LN\_S = ln -s

MKDIR = mkdir

RM = /bin/rm -f

TOUCH = touch

# - Platform identifier ------------------------------------------------

ARCH = Linux-AMD

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

TOPdir = /app/hpl-2.3

INCdir = $(TOPdir)/include

BINdir = $(TOPdir)/bin/$(ARCH)

LIBdir = $(TOPdir)/lib/$(ARCH)

HPLlib = $(LIBdir)/libhpl.a

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

MPdir = /usr/mpi/gcc/openmpi-4.1.5rc2

MPinc = $(MPdir)/include

#MPlib = $(MPdir)/lib64/libmpich.a

MPlib = $(MPdir)/lib64

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

LAdir = /app/openBLAS

LAinc =$(LAdir)/include/

LAlib =$(LAdir)/lib/

# - F77 / C interface --------------------------------------------------

F2CDEFS = -DAdd\_\_ -DF77\_INTEGER=int –DstringSunStyle

# - HPL includes / libraries / specifics -------------------------------

HPL\_INCLUDES = -I$(INCdir) -I$(INCdir)/$(ARCH) $(LAinc) $(MPinc)

HPL\_LIBS = $(HPLlib) $(LAlib) $(MPlib)

# - Compile time options -----------------------------------------------

HPL\_OPTS = -DHPL\_CALL\_CBLAS -DHPL\_DETAILED\_TIMING -DHPL\_PROGRESS\_REPORT

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

HPL\_DEFS = $(F2CDEFS) $(HPL\_OPTS) $(HPL\_INCLUDES)

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

CC = mpicc

CCNOOPT = $(HPL\_DEFS)

OMP\_DEFS = -fopenmp

CCFLAGS = $(HPL\_DEFS) -O3 -w -z noexecstack -z relro -z now -Wall

#

LINKER = $(CC)

LINKFLAGS = $(CCFLAGS) $(OMP\_DEFS)

ARCHIVER = ar

ARFLAGS = r

RANLIB = echo

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

Vim Make.top

./configure LIBS="-I /app/openBLAS/include -L /app/openBLAS/lib"

make clean arch=Linux-AMD

make arch=Linux-AMD

make -j32 install

export C\_INCLUDE\_PATH=/usr/include/mpi:$C\_INCLUDE\_PATH

ls /opt/hpl-2.3/testing/ptest/HPL.dat

cp –r /opt/hpl-2.3/testing/ptest/HPL.dat /app/hpl/bin

vim /opt/hpl-2.4/TUNING

ln -s /app/openBLAS/lib/libopenblas.so.0 /usr/lib/libopenblas.so.0

export PATH="/app/openBLAS/include:$PATH"

export LD\_LIBRARY\_PATH="/app/openBLAS/lib:$LD\_LIBRARY\_PATH"

/usr/lib64

[root@node lib64] ln -s libibumad.so.1.0.3 libibumad.so

[root@node lib64] ln -s libibverbs.so.1.0.0 libibverbs.so

HPL ERROR from process # 0, on line 610 of cuntion HPL\_pdinfo:

>>> Illegal input in file HPL.dat. Exiting ...<<<

Explanation: Topspin?supports the 1.0a release of xhpl. The error occurs when your HPL.dat file is version 1.0.

Solution: Edit the HPL.dat file and add the following line. The line must be the 9th line of the file:

0 PMAP process mapping (0=Row-, 1=Column-major)

# mpirun --allow-run-as-root -np 32 ./xhpl -f hpl.out ##it working 100%

# mpirun --allow-run-as-root -np 64 --oversubscribe ./xhpl ## 50%

#mpirun --allow-run-as-root -np 4 ./xhpl < HPL.dat |tee xhpl\_output.txt

Vim HPL.dat

HPL.out output file name (if any)

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

1 # of problems sizes (N)

286848 Ns

1 # of NBs

192 NBs

0 PMAP process mapping (0=Row-,1=Column-major)

1 # of process grids (P x Q)

4 Ps

4 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)

##### This line (no. 32) is ignored (it serves as a separator). ######

0 Number of additional problem sizes for PTRANS

1200 10000 30000 values of N

0 number of additional blocking sizes for PTRANS

40 9 8 13 13 20 16 32 64 values of NB

:wq!

(P \* Q) grid SIZE=equal to the number of processors in cluster. 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). Q slightly larger than P. P and Q for our 512 node cluster

P \* Q

1 \* 4096

2 \* 2048

4 \* 1024

8 \* 512

16 \* 256

32 \* 128

64 \* 64

“N”, which is the size of your problem, and usually the goal is to find the largest size problem =would fit in your system’s memory. “N” = close to your total memory size. don’t make it equal to 100%. some of the memory is consumed by the system. so choose something like 90% of the size of the total memory size. If you choose a small value for “N”, this will result in not enough work performed on each CPU and will give you bad results and low efficiency. If you choose a value of “N” exceeding your memory size, swapping will take place and the performance will go down.

“NB”, which is the block size in the grid. Usually block sizes giving good results are within the [96,104,112,120,128, …, 256] range.

sqrt((Memory Size in Gbytes \* 1024 \* 1024 \* 1024 \* Number of Nodes) /8) \* 0.90 Let’s do it step by step, so we have 512 nodes each with 12GB of memory:

(12 \* 1024 \* 1024 \* 1024 \* 512) /8 = 824633720832

Then the square root of that figure is ~ 908093 (Which is equivalent to 100% of total memory) Then that figure times 0.9 is ~ 817283