HOWTO - High Performance Linpack (HPL) on NVIDIA GPUs

This is a step by step procedure on how to run NVIDIA's version of the HPL benchmark on NVIDIA's S1070 and S2050 GPUs. We also compare the hybrid GPU runs with plain CPU runs on Intel's X5570 and X5670 processors to illustrate the performance boost gained with GPUs.

The following hardware/software was used for the first benchmark:

- Node with Intel Quad Core X5570 (dual socket) and Tesla S1070 (node sees 2 GPUs)
- Node has 12GB of RAM
- RedHat Enterprise Linux 4.8 64-bit
- Intel compiler version 11.1.072
- Intel MKL version 10.2.5.035
- Openmpi version 1.2.5
- Cudatoolkit 2.2 (cudatoolkit_2.2_linux_64_rhel4.7.run)
- NVIDIA driver supporting CUDA 2.2 (NVIDIA-Linux-x86_64-185.18.36-pkg2.run)
- Modified version of HPL from NVIDIA (hpl-2.0_CUDA_May_09_02_gt200.tgz)

#First you need to install the NVIDIA driver

```
[root@superbeast078 nvidia_cuda_rhel4.8]# ./NVIDIA-Linux-x86_64-185.18.36-pkg2.run-k `uname -r` -s
Verifying archive integrity... OK
Uncompressing NVIDIA Accelerated Graphics Driver for Linux-x86_64
.....
```

#Load the driver

```
[root@superbeast078 nvidia_cuda_rhel4.8]# modprobe -vvv nvidia
insmod /lib/modules/2.6.9-89.ELlargesmp/kernel/drivers/video/nvidia.ko
[root@superbeast078 nvidia_cuda_rhel4.8]# lsmod | grep nvidia
nvidia              11107116  0
i2c_core               36289  1 nvidia
```

#Test it is working

```
[root@superbeast078 nvidia_cuda_rhel4.8]# nvidia-smi
==============NVSMI LOG==============
Timestamp                       : Sat Jan  8 16:43:21 2011
Unit 0:
    Product Name            : NVIDIA Tesla S1070 -500
    Product ID              : 920-20804-0401
    Serial Number           : 0383609000106
    Firmware Ver            : 3.6
    Intake Temperature      : 21 C
    GPU 0:
        Product Name    : Tesla T10 Processor
        Serial          : 977055089846
        PCI ID          : 5e710de
        Bridge Port     : 0
        Temperature     : 24 C
    GPU 1:
        Product Name    : Tesla T10 Processor
        Serial          : 977055089846
        PCI ID          : 5e710de
        Bridge Port     : 2
        Temperature     : 23 C
    Fan Tachs:
        #00: 3504 Status: NORMAL
        #01: 3294 Status: NORMAL
        #02: 3570 Status: NORMAL
        #03: 3490 Status: NORMAL
        #04: 3654 Status: NORMAL
        #05: 3540 Status: NORMAL
        #06: 3436 Status: NORMAL
        #07: 3402 Status: NORMAL
        #08: 3590 Status: NORMAL
        #09: 3572 Status: NORMAL
        #10: 3574 Status: NORMAL
        #11: 3458 Status: NORMAL
        #12: 3582 Status: NORMAL
        #13: 3570 Status: NORMAL
    PSU:
        Voltage         :  11.99 V
        Current         :  11.95 A
        State           : Normal
    LED:
        State           : GREEN
```

#Install CUDA Toolkit

[root@superbeast078 cuda-2.2]# ./cudatoolkit_2.2_linux_64_rhel4.7.run -- auto

#Install Openmpi

Download from:
http://www.open-mpi.org/software/ompi/v1.2/downloads/openmpi-1.2.5.tar.gz

#Set your environment variables to point to the Intel compilers

[root@superbeast078 openmpi]# cat set-env
export CC=${CC:-/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64/icc}
export CXX=${CXX:-/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64/icpc}
export F77=${F77:-/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64/ifort}
export FC=${FC:-/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64/ifort}
export FC90=${FC90:-/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64/ifort}
[root@superbeast078 openmpi]# source set-env

#Compile Openmpi

[root@superbeast078 openmpi]# tar -xzvf openmpi-1.2.5.tar.gz
[root@superbeast078 openmpi]# cd openmpi-1.2.5
[root@superbeast078 openmpi-1.2.5]#./configure --prefix=/usr/local/mpi/openmpi-1.2.5/intel11
[root@superbeast078 openmpi-1.2.5]# make
[root@superbeast078 openmpi-1.2.5]# make install

#Install HPL from NVIDIA

[chewbacca@superbeast078 S1070]$ tar -xzvf hpl-2.0_CUDA_May_09_02_gt200.tgz
[chewbacca@superbeast078 S1070]$ cd hpl-2.0_CUDA
[chewbacca@superbeast078 hpl-2.0_CUDA]$ vi Make.CUDA_pinned

######Edit the below lines with your settings###########
TOPdir = /home/chewbacca/hpl-gpu/S1070/hpl-2.0_CUDA

MPdir = /usr/local/mpi/openmpi-1.2.5/intel11
MPinc = -I$(MPdir)/include
MPlib = $(MPdir)/lib/libmpi.so

LAdir = /usr/local/intel/11.1.072/mkl/10.2.5.035/lib/em64t
LAlib = -L $(TOPdir)/src/cuda -ldgemm -L/usr/local/cuda/lib -lcublas -L$(LAdir) -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5

CC = /usr/local/mpi/openmpi-1.2.5/intel11/bin/mpicc
LINKER = $(CC)
CCFLAGS = $(HPL_DEFS) -fomit-frame-pointer -O3 -funroll-loops -w -Wall
##

#Make sure your environment variables are set correct, you can set them in your .cshrc file

[chewbacca@superbeast078 hpl-2.0_CUDA]$ env | grep -w LD_LIBRARY_PATH
LD_LIBRARY_PATH=/usr/local/intel/11.1.072/Compiler/11.1/072/lib/intel64:/usr/local/intel/11.1.072/Compiler/11.1/072/lib/intel64:/usr/local/cuda/lib:/usr/local/cuda/lib64:/usr/local/mpi/openmpi-1.2.5/intel11/lib:/usr/X11R6/lib64:/usr/local/lib64:/usr/lib:/home/chewbacca/hpl-gpu/S1070/hpl-2.0_CUDA/src/cuda:/usr/local/intel/11.1.072/mkl/10.2.5.035/lib/em64t

[chewbacca@superbeast078 hpl-2.0_CUDA]$ env | grep -w PATH
PATH=/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64:/usr/local/intel/11.1.072/Compiler/11.1/072/bin/intel64:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/usr/local/cuda/bin:/usr/local/NVIDIA_CUDA_SDK/C/bin/linux/release:/usr/local/NVIDIA_CUDA_SDK/bin/linux/release:/usr/local/uxcat/bin:/usr/local/mpi/openmpi-1.2.5/intel11/bin

#Compile HPL

```
[chewbacca@superbeast078 hpl-2.0_CUDA]$ make arch=CUDA_pinned clean_arch_all
[chewbacca@superbeast078 hpl-2.0_CUDA]$ make arch=CUDA_pinned
[chewbacca@superbeast078 hpl-2.0_CUDA]$ cd bin/CUDA_pinned/
[chewbacca@superbeast078 CUDA_pinned]$ ls
HPL.dat  xhpl
```

#Create the below script to launch

```
[chewbacca@superbeast078 CUDA_pinned]$ cat run_linpack
#!/bin/bash
export HPL_DIR=/home/chewbacca/hpl-gpu/S1070/hpl-2.0_CUDA
export OMP_NUM_THREADS=4       #number of cpu cores per process
export MKL_NUM_THREADS=4       #number of cpu cores per GPU used
export MKL_DYNAMIC=FALSE
export CUDA_DGEMM_SPLIT=0.66  #how much work to offload to GPU for DGEMM
export CUDA_DTRSM_SPLIT=0.63    #how much work to offload to GPU for DTRSM
export LD_LIBRARY_PATH=$HPL_DIR/src/cuda:$LD_LIBRARY_PATH
$HPL_DIR/bin/CUDA_pinned/xhpl
```

#The current NVIDIA version of HPL requires having a 1:1 mapping between HPL #processes and GPUs. Since our node has 2 GPUs, we launch 2 HPL processes. MKL/OMP #threads will distribute the work on the 8 CPU cores. Our problem size N is around 80% of #the available 12GB memory. Below is the HPL.dat input file that was used

```
[chewbacca@superbeast078 CUDA_pinned]$ cat HPL.dat
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)
32032        Ns
1            # of NBs
1152         NBs
0            PMAP process mapping (0=Row-,1=Column-major)
1            # of process grids (P x Q)
1            Ps
2            Qs
16.0         threshold
1            # of panel fact
0            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.
0            RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
0            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
128          swapping threshold
1            L1 in (0=transposed,1=no-transposed) form
1            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

#Prepare nodes file

```
[chewbacca@superbeast078 CUDA_pinned]$ cat nodes
superbeast078
superbeast078
```

#Launch Tesla GPU version of HPL

```
[chewbacca@superbeast078 CUDA_pinned]$ which mpirun
/usr/local/mpi/openmpi-1.2.5/intel11/bin/mpirun
[chewbacca@superbeast078 CUDA_pinned]$ mpirun -np 2 -hostfile nodes ./run_linpack
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
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      :   32032
NB     :    1152
PMAP   : Row-major process mapping
P      :       1
Q      :       2
PFACT  :    Left
NBMIN  :       4
NDIV   :       2
RFACT  :    Left
BCAST  :   1ring
DEPTH  :       0
SWAP   : Mix (threshold = 128)
L1     : no-transposed form
U      : no-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

Assigning device 0  to process on node superbeast078 rank 0
Assigning device 1  to process on node superbeast078 rank 1
DTRSM split from environment variable 0.630000
DGEMM split from environment variable 0.660000
DTRSM split from environment variable 0.630000
DGEMM split from environment variable 0.660000
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR00L2L4       32032  1152     1     2             122.00              1.796e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0044480 ..... 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.
================================================================================
```

Rpeak (theoretical) = 93 GFLOPS (8 CPU cores) + 154 GFLOPS (2 GPUs) = 247 GFLOPS
Rmax (actual) = 179.6 GFLOPS (73% efficiency)

#For comparison, we did the same run but without GPUs using the standard HPL #benchmark, 2 HPL processes and same problem size N. MKL threads will distribute the #work on the 8 CPU cores. Notice that NB is set to 224 which is more suitable for CPU #runs. For the GPU run we had to increase NB to 1152

```
[chewbacca@superbeast078 Linux]$ mpirun -np 2 -hostfile nodes ./run_linpack
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
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      :   32032
NB     :     224
PMAP   : Row-major process mapping
P      :       1
Q      :       2
PFACT  :    Left
NBMIN  :       4
NDIV   :       2
RFACT  :    Left
BCAST  :   1ring
DEPTH  :       0
SWAP   : Mix (threshold = 128)
L1     : no-transposed form
U      : no-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               2.220446e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR00L2L4       32032   224     1     2             257.18             8.520e+01
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0021171 ..... 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.
================================================================================
```

Rpeak (theoretical) = 93 GFLOPS (8 CPU cores)
Rmax (actual) = 85.20 GFLOPS (92% efficiency)

The following hardware/software was used for the second benchmark on Fermi:

- Node with Intel Hex Core X5670 (dual socket) and Tesla S2050 (node sees 2 GPUs)
- Node has 48GB of RAM.
- RedHat Enterprise Linux 5.4 64-bit
- Intel compiler version 11.1.069
- Intel MKL version 10.2.5
- Openmpi version 1.4.1
- Cudatoolkit 3.1
- NVIDIA driver supporting CUDA 3.1 (NVIDIA-Linux-x86_64-256.53.run)
- Modified version of HPL from NVIDIA (hpl-2.0_FERMI_v09.tgz)

#Create the below script to launch

```
[chewbacca @superbeast069 CUDA_pinned]$ cat run_linpack
#!/bin/bash
export HPL_DIR=/home/chewbacca/hpl-2.0_FERMI_v09
export MKL_NUM_THREADS=6
export OMP_NUM_THREADS=6
export MKL_DYNAMIC=FALSE
export CUDA_DGEMM_SPLIT=0.836
export CUDA_DTRSM_SPLIT=0.806
export LD_LIBRARY_PATH=$HPL_DIR/src/cuda:$LD_LIBRARY_PATH
$HPL_DIR/bin/CUDA_pinned/xhpl
```

#The current NVIDIA version of HPL requires having a 1:1 mapping between HPL #processes and GPUs. Since our node has 2 GPUs, we launch 2 HPL processes. MKL/OMP #threads will distribute the work on the 12 CPU cores. Our problem size N is around 97% #of the available 48GB memory. Below is the HPL.dat input file that was used

```
 [chewbacca @superbeast069 CUDA_pinned]$ cat HPL.dat
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)
77212    Ns
1          # of NBs
512        NBs
0          PMAP process mapping (0=Row-,1=Column-major)
1          # of process grids (P x Q)
1          Ps
2          Qs
16.0         threshold
1          # of panel fact
0        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.
0        RFACTs (0=left, 1=Crout, 2=Right)
1          # of broadcast
0          BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1          # of lookahead depth
0        DEPTHs (>=0)
2          SWAP (0=bin-exch,1=long,2=mix)
128          swapping threshold
1          L1 in (0=transposed,1=no-transposed) form
1          U  in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)
```

#Launch Fermi GPU version of HPL

```
[chewbacca @superbeast069 CUDA_pinned]$ mpirun -np 2  --mca btl_openib_flags 1  ./run_linpack
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
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      :   77212
NB     :     512
PMAP   : Row-major process mapping
P      :       1
Q      :       2
PFACT  :    Left
NBMIN  :       4
NDIV   :       2
RFACT  :    Left
BCAST  :   1ring
DEPTH  :       0
SWAP   : Mix (threshold = 128)
L1     : no-transposed form
U      : no-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
--------------------------------------------------------------------------------
WR00L2L4       77212   512     1     2             421.96             7.273e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0041960 ..... 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.
================================================================================
```

Rpeak (theoretical) = 140 GFLOPS (12 CPU cores) + 1030 GFLOPS (2 GPUs) = 1170 GFLOPS
Rmax (actual) = 727.3 GFLOPS (62% efficiency)

#For comparison, we did the same run but without GPUs using the standard HPL #benchmark, 2 HPL processes and same problem size N. MKL threads will distribute the #work on the 12 CPU cores. Notice that NB is set to 224 which is more suitable for CPU #runs. For the Fermi GPU run we had to increase NB to 512

```
[chewbacca @superbeast069 Linux]$ mpirun -np 2  --mca btl_openib_flags 1  ./run_linpack
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
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      :   77212
NB     :     224
PMAP   : Row-major process mapping
P      :       1
Q      :       2
PFACT  :    Left
NBMIN  :       4
NDIV   :       2
RFACT  :    Left
BCAST  :   1ring
DEPTH  :       0
SWAP   : Mix (threshold = 128)
L1     : no-transposed form
U      : no-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               2.220446e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR00L2L4       77212   224     1     2            2211.29              1.388e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0020294 ..... 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.
================================================================================
```

Rpeak (theoretical) = 140 GFLOPS (12 CPU cores)
Rmax (actual) = 138.8 GFLOPS (99% efficiency)

Results summary of HPL runs:

Work done by	Rmax (GFLOPS)	Rpeak (GFLOPS)	Efficiency (%)	Problem Size N
Intel Quad Core X5570 (dual socket)	85.2	93	92%	32032
Intel Quad Core X5570 (dual socket) + 2 GPUs from NVIDIA S1070	179.6	247	73%	32032
Intel Hex Core X5670 (dual socket)	138.8	140	99%	77212
Intel Hex Core X5670 (dual socket) + 2 GPUs from NVIDIA S2050	727.3	1170	62%	77212

From the results obtained, the HPL performance gets accelerated by around 2X when aided by 2 GPUs from an S1070 compared to a plain run on an X5570 (8 CPU cores). The performance also gets accelerated by around 5X when aided by 2 GPUS from an S2050 compared to a plain run on an X5670 (12 CPU cores).