# Running Jobs on Comet
## (a practical guide)

Mary Thomas (mthomas@sdsc.edu)

April 19, 2019
Part 2

# Outline

- **Getting Started/Comet System Environment**
- **Comet Overview**
- **Compiling and Linking Code**
- **Running Parallel Jobs**
  - Running OpenMP Jobs
  - Running MPI Jobs
  - Running Hybrid MPI-OpenMP Jobs
  - Running GPU/CUDA Jobs
- **Final Comments**

# Hands-on Examples

# General Steps: Compiling/Running Jobs

- **Change to working directory**
  cd /home/$USER/comet-examples/MPI

- **Verify modules loaded:**
  module list
  Currently Loaded Modulefiles:
  1) intel/2013_sp1.2.144  2) mvapich2_ib/2.1  3) gnutools/2.69

- **Compile the MPI hello world code:**
  mpif90 -o hello_mpi hello_mpi.f90

- **Verify executable has been created (check that date):**
  ls -lt hello_mpi
  -rwxr-xr-x 1 user sdsc 721912 Mar 25 14:53 **hello_mpi**

- **Submit job from IBRUN directory (not required but helps with organization):**
  **cd /home/$USER/comet-examples/MPI/IBRUN**
  **sbatch --res=comet-examplesDAY1 hellompi-slurm.sb**

# Hands On Examples

- **Examples for :**
  - MPI
  - OpenMP
  - HYBRID
  - Local scratch

- **Running on Comet Compute Nodes**
  - 2-Socket (Total 24 cores)
  - Intel Haswell Processors

# Getting Set up

- Create a test directory (`e.g. comet-examples`)
- Copy the `/shared/apps/PHYS244` codebase to your test directory.
- Change to the test examples directory:

```
[comet-ln2:~] mkdir comet-examples
[comet-ln2:~/comet-examples/PHYS244] cd MPI
[comet-ln2:~/comet-examples/PHYS244/MPI] ll
total 872
drwxr-xr-x  4 user use300      7 Aug  6 09:55 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rwxr-xr-x  1 user use300 721944 Aug  6 09:55 hello_mpi
-rwxr-xr-x  1 user use300 721912 Aug  5 19:11 hello_mpi.bak
-rw-r--r--  1 user use300    357 Aug  5 19:22 hello_mpi.f90
drwxr-xr-x  2 user use300      6 Aug  6 10:04 IBRUN
drwxr-xr-x  2 user use300      3 Aug  5 19:02 MPIRUN_RSH
[comet-ln2:~/comet-examples/PHYS244/MPI] cat hello_mpi.f90
!  Fortran example
   program hello
   include 'mpif.h'
   integer rank, size, ierror, tag, status(MPI_STATUS_SIZE)

   call MPI_INIT(ierror)
   call MPI_COMM_SIZE(MPI_COMM_WORLD, size, ierror)
   call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierror)
   print*, 'node', rank, ': Hello and Welcome to Webinar Participants!'
   call MPI_FINALIZE(ierror)
   end
```

# Running MPI Jobs

# MPI Hello World

- Change to the MPI examples directory:

```
[comet-ln2:~/comet-examples/PHYS244] cd MPI
[comet-ln2:~/comet-examples/PHYS244/MPI] ll
total 872
drwxr-xr-x  4 user use300      7 Aug  6 09:55 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rwxr-xr-x  1 user use300 721944 Aug  6 09:55 hello_mpi
-rwxr-xr-x  1 user use300 721912 Aug  5 19:11 hello_mpi.bak
-rw-r--r--  1 user use300    357 Aug  5 19:22 hello_mpi.f90
drwxr-xr-x  2 user use300      6 Aug  6 10:04 IBRUN
drwxr-xr-x  2 user use300      3 Aug  5 19:02 MPIRUN_RSH
[comet-ln2:~/comet-examples/PHYS244/MPI] cat hello_mpi.f90
!  Fortran example
   program hello
   include 'mpif.h'
   integer rank, size, ierror, tag, status(MPI_STATUS_SIZE)

   call MPI_INIT(ierror)
   call MPI_COMM_SIZE(MPI_COMM_WORLD, size, ierror)
   call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierror)
   print*, 'node', rank, ': Hello and Welcome to Webinar Participants!'
   call MPI_FINALIZE(ierror)
   end
```

# MPI Hello World: Compile

Set the environment and then compile the code

```
[comet-ln2:~/comet-examples/PHYS244/MPI] module purge
[comet-ln2:~/comet-examples/PHYS244/MPI] module load gnutools
[comet-ln2:~/comet-examples/PHYS244/MPI] module load intel mvapich2_ib
[comet-ln2:~/comet-examples/PHYS244/MPI] module list
Currently Loaded Modulefiles:
  1) gnutools/2.69          2) intel/2013_sp1.2.144   3) mvapich2_ib/2.1

[comet-ln2:~/comet-examples/PHYS244/MPI] which mpif90
/opt/mvapich2/intel/ib/bin/mpif90

[comet-ln2:~/comet-examples/PHYS244/MPI] mpif90 -o hello_mpi hello_mpi.f90
[comet-ln2:~/comet-examples/PHYS244/MPI]
```

Try to run from command line:  it works, but it is not recommended.

```
[comet-ln2:~/comet-examples/PHYS244/MPI] mpirun -np 4 ./hello_mpi
 node           0 : Hello and Welcome Webinar Participants!
 node           1 : Hello and Welcome Webinar Participants!
 node           2 : Hello and Welcome Webinar Participants!
 node           3 : Hello and Welcome Webinar Participants!
```

# Using Interactive mode

Move to the IBRUN directory, and request nodes:

```
[comet-ln2:~/comet-examples/PHYS244/MPI/IBRUN] date
Tue Jan  8 00:22:42 PST 2019
[comet-ln2:~] hostname
comet-ln2.sdsc.edu
[comet-ln2:~/comet-examples/PHYS244/MPI/IBRUN] srun --pty --nodes=1 --ntasks-per-node=24 -p debug -t
00:30:00 --wait 0 /bin/bash
srun: job 20912306 queued and waiting for resources
srun: job 20912306 has been allocated resources
[comet-14-01:~/comet-examples/PHYS244/MPI/IBRUN] hostname
comet-14-01.sdsc.edu
[comet-14-01:~/comet-examples/PHYS244/MPI/IBRUN] mpirun -np 4 ../hello_mpi
 node         0 : Hello and Welcome Webinar Participants!
 node         1 : Hello and Welcome Webinar Participants!
 node         2 : Hello and Welcome Webinar Participants!
 node         3 : Hello and Welcome Webinar Participants!
[comet-14-01:~/comet-examples/PHYS244/MPI/IBRUN] exit
exit
[comet-ln2:~/comet-examples/PHYS244/MPI/IBRUN]
```

- Exit interactive session when work is done or you will be charged CPU time.
- Beware of oversubscribing your job: asking for more cores than you have. Intel compiler allows this, but your performance will be degraded.

# MPI Hello World: Batch Script

Move to the IBRUN directory, where the SLURM batch script is located:

```
[comet-ln2:~/comet-examples/PHYS244/MPI] cd IBRUN/
[comet-ln2:~/comet-examples/PHYS244/MPI/IBRUN] cat hellompi-slurm.sb
#!/bin/bash
#SBATCH --job-name="hellompi"
#SBATCH --output="hellompi.%j.%N.out"
#SBATCH --partition=compute
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=24
#SBATCH --export=ALL
#SBATCH -t 01:30:00

#This job runs with 2 nodes, 24 cores per node for a total of 48 cores.
#ibrun in verbose mode will give binding detail

ibrun -v ../hello_mpi
```

# MPI Hello World: submit job & monitor

- To run the job, use the batch script submission command.
- Monitor the job until it is finished using the squeue command.

```
[comet-ln3:~/comet-examples/PHYS244/MPI/IBRUN] sbatch hellompi-slurm.sb
Submitted batch job 20918244
[comet-ln3:~/comet-examples/PHYS244/MPI/IBRUN] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20918244   compute hellompi  user PD       0:00      2 (None)
[comet-ln3:~/comet-examples/PHYS244/MPI/IBRUN] squeue -u user JOBID PARTITION     NAME
USER ST       TIME  NODES NODELIST(REASON)
          20918244   compute hellompi  user  R       0:01      2 comet-11-[01,58]
[comet-ln3:~/comet-examples/PHYS244/MPI/IBRUN] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20918244   compute hellompi  user CG       0:02      1 comet-11-01
[comet-ln3:~/comet-examples/PHYS244/MPI/IBRUN] ll
total 67
drwxr-xr-x 2 user use300    5 Jan  8 13:25 .
drwxr-xr-x 4 user use300    8 Jan  8 13:12 ..
-rw-r--r-- 1 user use300 9218 Jan  8 13:25 hellompi.20918244.comet-11-01.out
-rw-r--r-- 1 user use300  342 Aug  5 19:34 hellompi-slurm.sb
```

# MPI Hello World: Output

Monitor the job until it is finished

```
[comet-ln2:~/comet-examples/PHYS244/MPI/IBRUN] cat hellompi.20912353.comet-20-06.out
IBRUN: Command is ../hello_mpi
IBRUN: Command is /home/user/comet-examples/PHYS244/MPI/hello_mpi
IBRUN: no hostfile mod needed
IBRUN: Nodefile is /tmp/AaTm2VFWKx
IBRUN: MPI binding policy: compact/core for 1 threads per rank (12 cores per socket)
IBRUN: Adding MV2_USE_OLD_BCAST=1 to the environment
IBRUN: Adding MV2_CPU_BINDING_LEVEL=core to the environment
IBRUN: Adding MV2_ENABLE_AFFINITY=1 to the environment
IBRUN: Adding MV2_DEFAULT_TIME_OUT=23 to the environment
IBRUN: Adding MV2_CPU_BINDING_POLICY=bunch to the environment
IBRUN: Adding MV2_USE_HUGEPAGES=0 to the environment
IBRUN: Adding MV2_HOMOGENEOUS_CLUSTER=0 to the environment
IBRUN: Adding MV2_USE_UD_HYBRID=0 to the environment
IBRUN: Added 8 new environment variables to the execution environment
IBRUN: Command string is [mpirun_rsh -np 48 -hostfile /tmp/AaTm2VFWKx -export-all
/home/user/comet-examples/PHYS244/MPI/hello_mpi]
 node           15 : Hello and Welcome Webinar Participants!
 node           16 : Hello and Welcome Webinar Participants!
 node           19 : Hello and Welcome Webinar Participants!
 node            9 : Hello and Welcome Webinar Participants!
……..
node           25 : Hello and Welcome Webinar Participants!
 node           30 : Hello and Welcome Webinar Participants!
 node           29 : Hello and Welcome Webinar Participants!
 node           33 : Hello and Welcome Webinar Participants!
 node           31 : Hello and Welcome Webinar Participants!
IBRUN: Job ended with value 0
```

# Running OpenMP Jobs

# OpenMP Hello World

Change to the OPENMP examples directory:

```
[comet-ln2:~/comet-examples/PHYS244] cd OPENMP
[comet-ln2:~/comet-examples/PHYS244/OPENMP] ls -al
total 498
drwxr-xr-x  2 user use300      8 Aug  5 23:25 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rw-r--r--  1 user use300    267 Aug  5 22:19 hello_openmp.f90
-rw-r--r--  1 user use300    311 Aug  5 23:25 openmp-slurm.sb
-rw-r--r--  1 user use300    347 Aug  5 19:02 openmp-slurm-shared.sb

[comet-ln2:~/comet-examples/PHYS244/OPENMP] cat hello_openmp.f90
      PROGRAM OMPHELLO
      INTEGER TNUMBER
      INTEGER OMP_GET_THREAD_NUM

!$OMP PARALLEL DEFAULT(PRIVATE)
      TNUMBER = OMP_GET_THREAD_NUM()
      PRINT *, 'Hello from Thread Number[',TNUMBER,'] and Welcome Webinar!'
!$OMP END PARALLEL

      STOP
      END
```

# MPI Hello World: Compile

Check the environment and then compile the code

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP] module list
Currently Loaded Modulefiles:
  1) gnutools/2.69          2) intel/2013_sp1.2.144   3) mvapich2_ib/2.1
[comet-ln2:~/comet-examples/PHYS244/OPENMP] ifort -o hello_openmp -openmp hello_openmp.f90
```

Compile using the ifort command

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP] ifort -o hello_openmp -openmp hello_openmp.f90
[comet-ln2:~/comet-examples/PHYS244/OPENMP]
```

# OpenMP Hello World: Controlling #Threads

A key issue when running OpenMP code is controlling thread behavior.
If you run from command line, it will work, but it is not recommended because you will be using Pthreads, which automatically picks the number of threads - in this case 24.

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP]./hello_openmp
 Hello from Thread Number[           0 ] and Welcome Webinar!
 Hello from Thread Number[           2 ] and Welcome Webinar!
  .   .   .   .
 Hello from Thread Number[          22 ] and Welcome Webinar!
 Hello from Thread Number[          11 ] and Welcome Webinar!
 Hello from Thread Number[          23 ] and Welcome Webinar!
```

To control thread behavior, there are several key environment variables:
OMP_NUM_THREADS controls the number of threads allowed, and OMP_PROC_BIND binds threads to “places” (e.g. cores) and keeps them from moving around (between cores).

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP] export OMP_NUM_THREADS=4; ./hello_openmp
HELLO FROM THREAD NUMBER = 3
HELLO FROM THREAD NUMBER = 1
HELLO FROM THREAD NUMBER = 2
HELLO FROM THREAD NUMBER = 0
```

See: https://www.ibm.com/support/knowledgecenter/SSGH2K_13.1.3/com.ibm.xlc1313.aix.doc/compiler_ref/ruomprun.html

# OpenMP Hello World: Batch Script

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP] cat openmp-slurm.sb
#!/bin/bash
#SBATCH --job-name="hello_openmp"
#SBATCH --output="hello_openmp.%j.%N.out"
#SBATCH --partition=compute
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --export=ALL
#SBATCH -t 01:30:00

#SET the number of openmp threads
export OMP_NUM_THREADS=24

#Run the job using mpirun_rsh
./hello_openmp
```

- Comet supports shared-node jobs (more than one job on a single node).
- Many applications are serial or can only scale to a few cores.
- Shared nodes improve job throughput, provide higher overall system utilization, and allow more users to run on jobs.

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP] cat openmp-slurm-shared.sb
#!/bin/bash
#SBATCH --job-name="hell_openmp_shared"
#SBATCH --output="hello_openmp_shared.%j.%N.out"
#SBATCH --partition=shared
#SBATCH --share
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --mem=80G
#SBATCH --export=ALL
#SBATCH -t 01:30:00

#SET the number of openmp threads
export OMP_NUM_THREADS=16

#Run the openmp job
./hello_openmp
```

# OpenMP Hello World: submit job & monitor

To run the job, type the batch script submission command:

```
[comet-ln2:~/comet-examples/PHYS244/OPENMP] sbatch openmp-slurm.sb
Submitted batch job 20912556
[comet-ln2:~/comet-examples/PHYS244/OPENMP] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20912556   compute hello_op  user PD       0:00      1 (None)
[comet-ln2:~/comet-examples/PHYS244/OPENMP] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20912556   compute hello_op  user  R       0:00      1 comet-10-45
[comet-ln2:~/comet-examples/PHYS244/OPENMP] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20912556   compute hello_op  user CG       0:03      1 comet-10-45
[comet-ln2:~/comet-examples/PHYS244/OPENMP] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[comet-ln2:~/comet-examples/PHYS244/OPENMP] cat  hello_openmp.20912556.comet-10-45.out
 Hello from Thread Number[           0 ] and Welcome Webinar Participants!
 Hello from Thread Number[          18 ] and Welcome Webinar Participants!
 Hello from Thread Number[           4 ] and Welcome Webinar Participants!
 Hello from Thread Number[          15 ] and Welcome Webinar Participants!
 Hello from Thread Number[          21 ] and Welcome Webinar Participants!
 Hello from Thread Number[          11 ] and Welcome Webinar Participants!
 Hello from Thread Number[          16 ] and Welcome Webinar Participants!
….
```

# Running Hybrid MPI-OpenMP Jobs

# Hybrid MPI + OpenMP Jobs

- Several HPC codes use a hybrid MPI, OpenMP approach.
- **ibrun** wrapper developed to handle hybrid use cases.
  - Automatically senses the MPI build (*mvapich2, openmpi*) and binds tasks correctly.
- **ibrun** -help gives detailed usage info.

# Hybrid MPI + OpenMP Hello World

```
[comet-ln2:~/comet-examples/PHYS244] cd HYBRID
[comet-ln2:~/comet-examples/PHYS244/HYBRID] ls -al
total 94
drwxr-xr-x  2 user use300      5 Jan  8 01:53 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rw-r--r--  1 user use300    636 Aug  5 19:02 hello_hybrid.c
-rw-r--r--  1 user use300    390 Aug  5 19:02 hybrid-slurm.sb
[comet-ln2:~/comet-examples/PHYS244/HYBRID] cat hello_hybrid.c
#include <stdio.h>
#include "mpi.h"
#include <omp.h>

int main(int argc, char *argv[]) {
  int numprocs, rank, namelen;
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int iam = 0, np = 1;

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Get_processor_name(processor_name, &namelen);

  #pragma omp parallel default(shared) private(iam, np)
  {
    np = omp_get_num_threads();
    iam = omp_get_thread_num();
    printf("Hello Webinar particpants from thread %d out of %d from process %d out of %d on %s\n",
           iam, np, rank, numprocs, processor_name);
  }

  MPI_Finalize();
}
```

# Hybrid Hello World: Compile, batch script

- To compile the hybrid MPI + OpenMPI code, we need to refer to the table of compilers listed above (and listed in the user guide).
- We will use the command **mpicc -openmp**

```
[comet-ln2:~/comet-examples/PHYS244/HYBRID] mpicc -openmp -o hello_hybrid hello_hybrid.c
[comet-ln2:~/comet-examples/PHYS244/HYBRID] ls -al
total 94
drwxr-xr-x  2 user use300      5 Jan  8 02:00 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rwxr-xr-x  1 user use300 103032 Jan  8 02:00 hello_hybrid
-rw-r--r--  1 user use300    636 Aug  5 19:02 hello_hybrid.c
-rw-r--r--  1 user use300    390 Aug  5 19:02 hybrid-slurm.sb
[comet-ln2:~/comet-examples/PHYS244/HYBRID]
```

```
[comet-ln2:~/comet-examples/PHYS244/HYBRID] cat hybrid-slurm.sb
#!/bin/bash
#SBATCH --job-name="hellohybrid"
#SBATCH --output="hellohybrid.%j.%N.out"
#SBATCH --partition=compute
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=24
#SBATCH --export=ALL
#SBATCH -t 01:30:00

#This job runs with 2 nodes, 24 cores per node for a total of 48 cores.
# We use 8 MPI tasks and 6 OpenMP threads per MPI task

export OMP_NUM_THREADS=6
ibrun --npernode 4 ./hello_hybrid
```

# Hybrid Hello World: submit job & monitor

To run the job, type the batch script submission command:

```
[comet-ln2:~/comet-examples/PHYS244/HYBRID] sbatch hybrid-slurm.sb
Submitted batch job 20912643
[comet-ln2:~/comet-examples/PHYS244/HYBRID] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20912643   compute hellohyb  user PD       0:00      2 (None)
[comet-ln2:~/comet-examples/PHYS244/HYBRID] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20912643   compute hellohyb  user  R       0:01      2 comet-06-[48,64]
[comet-ln2:~/comet-examples/PHYS244/HYBRID] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          20912643   compute hellohyb  user CG       0:06      2 comet-06-[48,64]
[comet-ln2:~/comet-examples/PHYS244/HYBRID] squeue -u user
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[comet-ln2:~/comet-examples/PHYS244/HYBRID] ll
total 132
drwxr-xr-x  2 user use300      7 Jan  8 02:12 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rwxr-xr-x  1 user use300 103032 Jan  8 02:00 hello_hybrid
-rw-r--r--  1 user use300   3771 Jan  8 02:12 hellohybrid.20912643.comet-06-48.out
-rw-r--r--  1 user use300    636 Aug  5 19:02 hello_hybrid.c
-rw-r--r--  1 user use300    390 Aug  5 19:02 hybrid-slurm.sb ….
```

# Hybrid Hello World: Output

Code ran on:

- 2 nodes,
- 4 cores per node,
- 6 threads per core

```
[comet-ln2:~/comet-examples/PHYS244/HYBRID] cat hellohybrid.20912643.comet-06-48.out | sort
Hello from thread 0 out of 6 from process 0 out of 8 on comet-06-48.sdsc.edu
Hello from thread 0 out of 6 from process 1 out of 8 on comet-06-48.sdsc.edu
Hello from thread 0 out of 6 from process 2 out of 8 on comet-06-48.sdsc.edu
Hello from thread 0 out of 6 from process 3 out of 8 on comet-06-48.sdsc.edu
Hello from thread 0 out of 6 from process 4 out of 8 on comet-06-64.sdsc.edu
Hello from thread 0 out of 6 from process 5 out of 8 on comet-06-64.sdsc.edu
Hello from thread 0 out of 6 from process 6 out of 8 on comet-06-64.sdsc.edu
Hello from thread 0 out of 6 from process 7 out of 8 on comet-06-64.sdsc.edu
Hello from thread 1 out of 6 from process 0 out of 8 on comet-06-48.sdsc.edu
Hello from thread 1 out of 6 from process 1 out of 8 on comet-06-48.sdsc.edu
Hello from thread 1 out of 6 from process 2 out of 8 on comet-06-48.sdsc.edu
Hello from thread 1 out of 6 from process 3 out of 8 on comet-06-48.sdsc.edu
Hello from thread 1 out of 6 from process 4 out of 8 on comet-06-64.sdsc.edu
Hello from thread 1 out of 6 from process 5 out of 8 on comet-06-64.sdsc.edu
Hello from thread 1 out of 6 from process 6 out of 8 on comet-06-64.sdsc.edu
Hello from thread 1 out of 6 from process 7 out of 8 on comet-06-64.sdsc.edu
Hello from thread 2 out of 6 from process 0 out of 8 on comet-06-48.sdsc.edu
Hello from thread 2 out of 6 from process 1 out of 8 on comet-06-48.sdsc.edu
Hello from thread 2 out of 6 from process 2 out of 8 on comet-06-48.sdsc.edu
Hello from thread 2 out of 6 from process 3 out of 8 on comet-06-48.sdsc.edu
Hello from thread 2 out of 6 from process 4 out of 8 on comet-06-64.sdsc.edu
Hello from thread 2 out of 6 from process 5 out of 8 on comet-06-64.sdsc.edu
Hello from thread 2 out of 6 from process 6 out of 8 on comet-06-64.sdsc.edu
Hello from thread 2 out of 6 from process 7 out of 8 on comet-06-64.sdsc.edu
Hello from thread 3 out of 6 from process 0 out of 8 on comet-06-48.sdsc.edu
Hello from thread 3 out of 6 from process 1 out of 8 on comet-06-48.sdsc.edu
Hello from thread 3 out of 6 from process 2 out of 8 on comet-06-48.sdsc.edu
Hello from thread 3 out of 6 from process 3 out of 8 on comet-06-48.sdsc.edu
Hello from thread 3 out of 6 from process 4 out of 8 on comet-06-64.sdsc.edu
Hello from thread 3 out of 6 from process 5 out of 8 on comet-06-64.sdsc.edu
Hello from thread 3 out of 6 from process 6 out of 8 on comet-06-64.sdsc.edu
Hello from thread 3 out of 6 from process 7 out of 8 on comet-06-64.sdsc.edu
Hello from thread 4 out of 6 from process 0 out of 8 on comet-06-48.sdsc.edu
Hello from thread 4 out of 6 from process 1 out of 8 on comet-06-48.sdsc.edu
Hello from thread 4 out of 6 from process 2 out of 8 on comet-06-48.sdsc.edu
Hello from thread 4 out of 6 from process 3 out of 8 on comet-06-48.sdsc.edu
Hello from thread 4 out of 6 from process 4 out of 8 on comet-06-64.sdsc.edu
Hello from thread 4 out of 6 from process 5 out of 8 on comet-06-64.sdsc.edu
Hello from thread 4 out of 6 from process 6 out of 8 on comet-06-64.sdsc.edu
Hello from thread 4 out of 6 from process 7 out of 8 on comet-06-64.sdsc.edu
Hello from thread 5 out of 6 from process 0 out of 8 on comet-06-48.sdsc.edu
Hello from thread 5 out of 6 from process 1 out of 8 on comet-06-48.sdsc.edu
Hello from thread 5 out of 6 from process 2 out of 8 on comet-06-48.sdsc.edu
Hello from thread 5 out of 6 from process 3 out of 8 on comet-06-48.sdsc.edu
Hello from thread 5 out of 6 from process 4 out of 8 on comet-06-64.sdsc.edu
Hello from thread 5 out of 6 from process 5 out of 8 on comet-06-64.sdsc.edu
Hello from thread 5 out of 6 from process 6 out of 8 on comet-06-64.sdsc.edu
Hello from thread 5 out of 6 from process 7 out of 8 on comet-06-64.sdsc.edu
```

# Compiling and Running GPU/CUDA

# Comet GPU Hardware

| ***NVIDIA Kepler K80 GPU Nodes*** | |
|---|---|
| Node count | 36 |
| CPU cores:GPUs/node | 24:4 |
| CPU:GPU DRAM/node | 128 GB:48 GB |
| ***NVIDIA Pascal P100 GPU Nodes*** | |
| Node count | 36 |
| CPU cores:GPUs/node | 28:4 |
| CPU:GPU DRAM/node | 128 GB:64 GB |

# GPU/CUDA: check node for GPU card

Note: you will be able to compile GPU code on the login nodes, but they will not run. To see if your node has GPU hardware, run *lspci*. Comet login nodes do not have GPU.

```
[comet-ln2:~/comet-examples/PHYS244/CUDA] lspci | grep VGA
09:00.0 VGA compatible controller: ASPEED Technology, Inc. ASPEED Graphics Family
(rev 30)
```

If the node does have a GPU card, you will see output similar to the following (example from a different system):

```
[user@host.sdsu.edu]$ ssh node9 "/sbin/lspci | grep VGA"
01:00.0 VGA compatible controller: NVIDIA Corp.. NV44 [GeForce 6200 LE] (rev a1)
02:00.0 VGA compatible controller: NVIDIA Corp.. GF100 [GeForce GTX 480] (rev a3)
03:00.0 VGA compatible controller: NVIDIA Corp.. GF100 [GeForce GTX 480] (rev a3)
```

# GPU/CUDA MatMul

- Change to the CUDA examples directory:

```
[comet-ln2:~/comet-examples/PHYS244] cd CUDA
[comet-ln2:~/comet-examples/PHYS244/CUDA] ll -al
total 474
drwxr-xr-x  2 user use300     16 Jan  8 09:47 .
drwxr-xr-x 16 user use300     16 Aug  5 19:02 ..
-rw-r--r--  1 user use300    503 Jan  8 09:31 CUDA.20915480.comet-31-11.out
-rw-r--r--  1 user use300    253 Aug  5 19:02 cuda.sb
-rw-r--r--  1 user use300   5106 Aug  5 19:02 exception.h
-rw-r--r--  1 user use300   1168 Aug  5 19:02 helper_functions.h
-rw-r--r--  1 user use300  29011 Aug  5 19:02 helper_image.h
-rw-r--r--  1 user use300  23960 Aug  5 19:02 helper_string.h
-rw-r--r--  1 user use300  15414 Aug  5 19:02 helper_timer.h
-rwxr-xr-x  1 user use300 535634 Jan  8 09:28 matmul
-rw-r--r--  1 user use300  13556 Aug  6 00:54 matrixMul.cu
```

# GPU/CUDA: Compile

- Set the environment
- Then compile the code

```
[comet-ln2:~/cuda/gpu_enum] module purge
[comet-ln2:~/cuda/gpu_enum] which nvcc
/usr/bin/which: no nvcc in (/usr/lib64/qt-
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/sdsc/bin:/o
pt/sdsc/sbin:/opt/ibutils/bin:/usr/java/latest/bin:/opt/pdsh/bin:/opt/rocks/bin:/opt/
rocks/sbin:/home/user/bin)
[comet-ln2:~/cuda/gpu_enum] module load cuda
[comet-ln2:~/cuda/gpu_enum] which nvcc
/usr/local/cuda-7.0/bin/nvcc
[comet-ln2:~/cuda/gpu_enum] nvcc -o gpu_enum -I.  gpu_enum.cu
[comet-ln2:~/cuda/gpu_enum] ll gpu_enum
-rwxr-xr-x 1 mthomas use300 517632 Apr 10 18:39 gpu_enum
[comet-ln2:~/cuda/gpu_enum]
```

# GPU/CUDA: Interactive Node

- Set the environment
- Then compile the code

```
[comet-ln2:~/comet-examples/PHYS244/CUDA] module load cuda
[comet-ln2:~/comet-examples/PHYS244/CUDA] srun --partition=gpu-shared --nodes=1 --
ntasks-per-node=7 --gres=gpu:p100:1 -t 00:10:00  --pty --wait=0 --export=ALL
/bin/bash
srun: job 22527658 queued and waiting for resources
….
35 MINUTES LATER!!!!

[mthomas@comet-33-09:~]
```

# GPU/CUDA: Interactive Node

Check node configuration:

```
[mthomas@comet-33-09:~/cuda/gpu_enum] nvidia-smi
Wed Apr 10 20:38:51 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 396.26                 Driver Version: 396.26                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  On   | 00000000:04:00.0 Off |                    0 |
| N/A   62C    P0   150W / 250W |   6484MiB / 16280MiB |     87%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P100-PCIE...  On   | 00000000:05:00.0 Off |                    0 |
| N/A   50C    P0   148W / 250W |    527MiB / 16280MiB |     54%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla P100-PCIE...  On   | 00000000:85:00.0 Off |                    0 |
| N/A   32C    P0    29W / 250W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla P100-PCIE...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   32C    P0    29W / 250W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0    111621      C   java                                       6474MiB |
|    1     93567      C   python3                                     517MiB |
+-----------------------------------------------------------------------------+
```

# GPU/CUDA: Batch Script Config

- GPU nodes can be accessed via either the "gpu" or the "gpu-shared" partitions.

```
#SBATCH -p gpu
   or
#SBATCH -p gpu-shared
```

- In addition to the partition name(required), the type of gpu(optional) and the individual GPUs are scheduled as a resource.

```
#SBATCH --gres=gpu[:type]:n
```

- GPUs will be allocated on a first available, first schedule basis, unless specified with the [type] option, where type can be k80 or p100 (type is case sensitive)

```
#SBATCH --gres=gpu:4      #first available gpu node
#SBATCH --gres=gpu:k80:4 #only k80 nodes
#SBATCH --gres=gpu:p100:4 #only p100 nodes
```

# GPU/CUDA: Batch Script

SLURM batch script contents:

```
[comet-ln2: ~/cuda/gpu_enum] cat gpu_enum.sb
#!/bin/bash
#SBATCH --job-name="gpu_enum"
#SBATCH --output="gpu_enum.%j.%N.out"
#SBATCH --partition=gpu-shared          # define GPU partition
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=6
#SBATCH --gres=gpu:1         # define type of GPU
#SBATCH -t 00:10:00

#Load the cuda module
module load cuda

#Run the job
./gpu_enum
```

# GPU/CUDA: submit job & monitor

- To run the job, type the batch script submission command:

```
[comet-ln2:~/cuda/gpu_enum] sbatch gpu_enum.sb
Submitted batch job 22527745
```

- Monitor the job until it is finished

```
[user@comet-ln2:~/cuda/gpu_enum] sbatch gpu_enum.sb
Submitted batch job 22527745
[user@comet-ln2:~/cuda/gpu_enum] squeue -u mthomas
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          22527658 gpu-share     bash  mthomas PD       0:00      1 (Resources)
          22527745 gpu-share gpu_enum  mthomas PD       0:00      1 (None)
```

```
[user@comet-ln2:~/cuda/gpu_enum] cat gpu_enum.22527745.comet-31-10.out
 --- Obtaining General Information for CUDA devices  ---
 --- General Information for device 0 ---
Name: Tesla K80
Compute capability: 3.7
Clock rate: 823500
Device copy overlap: Enabled
Kernel execution timeout : Disabled
 --- Memory Information for device 0 ---
Total global mem: 11996954624
Total constant Mem: 65536
Max mem pitch: 2147483647
Texture Alignment: 512
 --- MP Information for device 0 ---
Multiprocessor count: 13
Shared mem per mp: 49152
Registers per mp: 65536
Threads in warp: 32
Max threads per block: 1024
Max thread dimensions: (1024, 1024, 64)
Max grid dimensions: (2147483647, 65535, 65535)
```

# Wrapping it up

# Yes, You are Correct: Running jobs on HPC Systems is Complex

- **Multiple layers of hardware and software affect job performance**
- **Learn to develop and test in a modular fashion**
- **Build up a suite of test cases:**
  - When things go wrong, make sure you can run simple test cases (HelloWorld).
  - This can eliminate questions about your environment.
- **Consider using a code repository**
  - When things go wrong, you can get back to a working version
- **If you need help/have questions, contact XSEDE help desk:**
  - **They are very helpful and respond quickly**
  - Support users around the world, so they are truly a 7/24 service
  - Avoid wasting your time.

# When Things Go Wrong, Check Your User Environment

- **Do you have the right modules loaded?**
- **What software versions do you need?**
- **Is your code compiled and updated (or did you compile it last year?)**
- **Are you running your job from the right location?**
  - $HOME versus $WORK?

# Run jobs from the right location

- **Lustre scratch filesystem:**
  - /oasis/scratch/comet/$USER/temp_project
  - Preferred: Scalable large block I/O)
- **Compute/GPU node local SSD storage:**
  - /scratch/$USER/$SLURM_JOBID
  - Meta-data intensive jobs, high IOPs)
- **Lustre projects filesystem:**
  - /oasis/projects/nsf
- **/home/$USER:**
  - Only for source files, libraries, binaries.
  - *Do not* use for I/O intensive jobs.

# For Fun:

- **Join the UCSD Supercomputing Club:**
  - http://supercomputingclub.ucsd.edu/
  - https://training.sdsc.edu/scc-training-schedule
  - Rasbery PI^3 event Friday, 4/12/19 @ 3pm
    - Free pie….
- **Check out the Student Cluster Competition Activity @ SDSC:**
  - https://training.sdsc.edu/scc (
  - Training sessions: kickoff on 4/12/19 @1pm
  - Working with the new ARM architecture (RISC)
  - Seeking a few grad students interested in mentoring ☺
- **Take a tour of SDSC!**
  - Supercomputing Club on 4/19/19

# References

- **Comet User Guide**
  - https://www.sdsc.edu/support/user_guides/comet.html#compiling
- **SDSC Training Resources**
  - https://www.sdsc.edu/education_and_training/training.html
  - https://github.com/sdsc-training/webinars
  - Comet shared apps/examples; can be found in
    - /share/apps
- **XSEDE Training Resources**
  - https://www.xsede.org/for-users/training
  - https://cvw.cac.cornell.edu/comet/