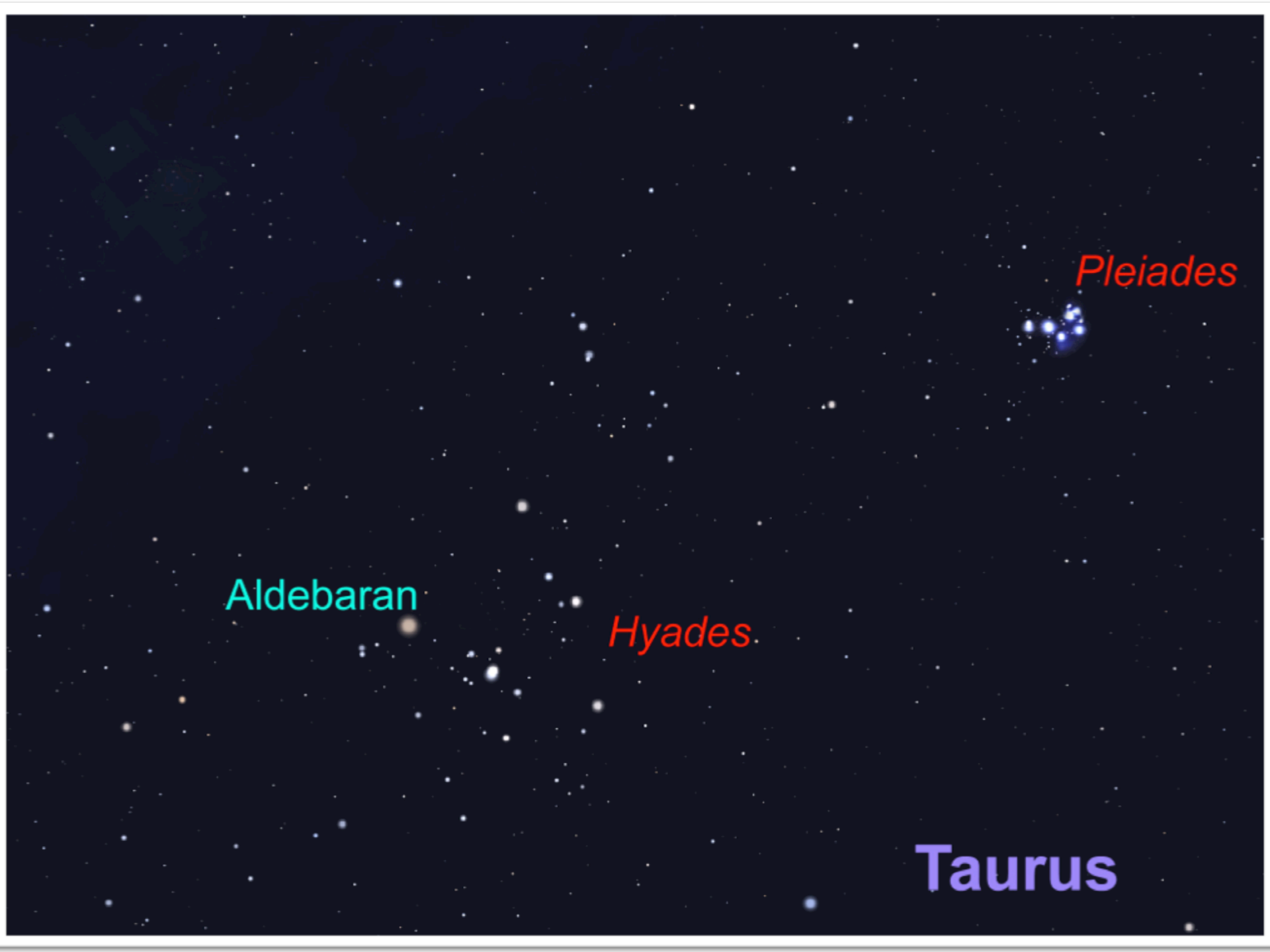
Introduction to Hyades

Shawfeng Dong

Department of Astronomy &
Astrophysics, UCSSC

Hyades

① Hardware Architecture
② Accessing Hyades
③ Computing Environment
④ Compiling Codes
⑤ Running Jobs
⑥ Visualization and Analysis
⑦ Q&A

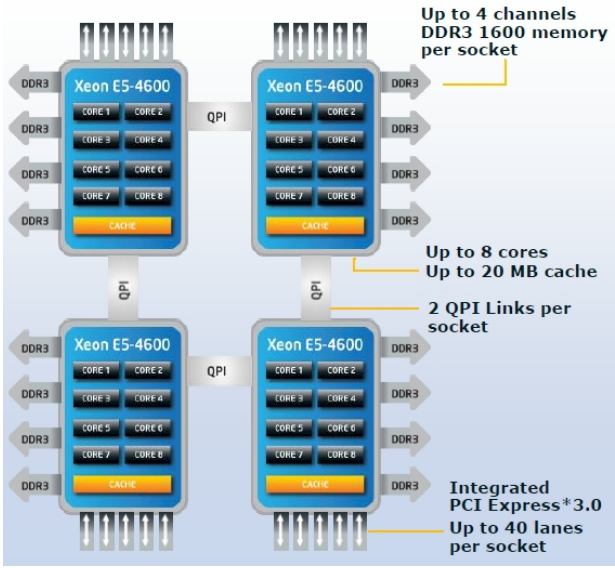

Pleiades
Aldebaran
Hyades
Taurus

Hardware Summary

Component	QTY	Description
Master Node	1	PowerEdge R820, 4x 8-core Intel Xeon E5-4620, 128GB of memory, 8x 1TB HDD
Analysis Node	1	PowerEdge R820, 4x 8-core Intel Xeon E5-4640, 512GB of memory, 2x 600GB SSD
Type I Compute Nodes	180	PowerEdge R620, 2x 8-core Intel Xeon E5-2650, 64GB of memory, 1TB HDD
Type IIa Compute Nodes	8	PowerEdge C8220x, 2x 8-core Intel Xeon E5-2650, 64GB of memory, 2x 500GB HDD, 1x Nvidia K20
Type IIb Compute Node	1	PowerEdge R720, 2x 6-core Intel Xeon E5-2630L, 64GB of memory, 500GB HDD, 2x Xeon Phi 5110P
Lustre Scratch Storage	1	Dell/Terascala solution, 146TB of usable storage.
Huawei UDS Storage	1	1PB private cloud storage
InfiniBand		Mellanox QDR (40 Gb/s) non-blocking Infiniband solution

Master Node

- Dell PowerEdge R820
- 4x Intel Xeon E5-4620 (8-core, 2.2GHz)
- 128 GB memory
- 8x 1TB HDD in RAID-6
  - hyades.ucsc.edu
  - plumbing for the cluster
  - Compile and debug codes
  - Submit & monitor jobs, etc



Analysis Node

- Dell PowerEdge R820
- 4x Intel Xeon E5-4640 (8-core, 2.4GHz)
- 512 GB memory (16GB/core)!
- 2x 600GB SSD in RAID-0
  - eudora.ucsc.edu
  - Visualization/analysis & big-memory jobs

Type I Compute Nodes

- Dell PowerEdge R620
- 2x Intel Xeon E5-2650 (8-core, 2.0GHz)
- 64GB memory (4GB/core, 1.6GHz RDIMMs)
- 1TB HDD
- 180 nodes

Type IIa Compute Nodes

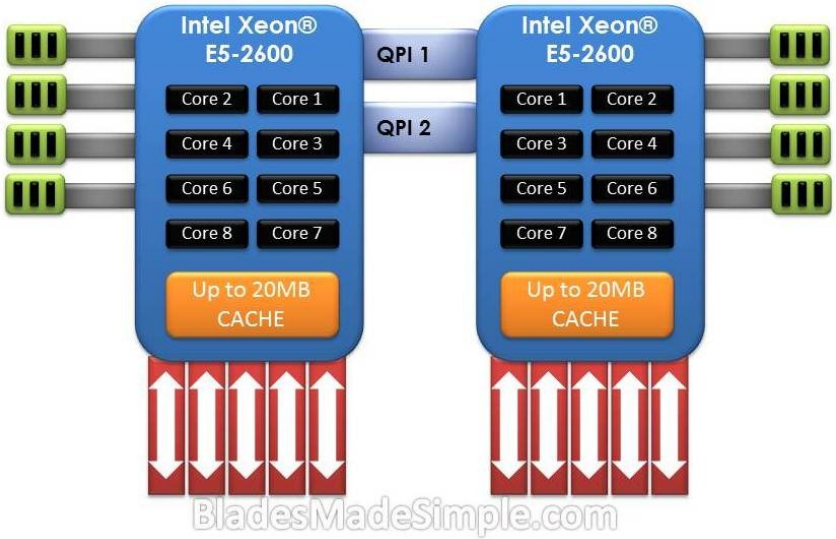
- Dell PowerEdge C8220x
- 2x Intel Xeon E5-2650 (8-core, 2.0GHz)
- 64GB memory (4GB/core, 1.6GHz RDIMMs)
- 2x 500GB HDDs
- 1x Nvidia Tesla K20 GPU accelerator
- 8 nodes

Type IIb Compute Node

- Dell PowerEdge R720
- 2x Intel Xeon E5-2650 (6-core, 2.0GHz)
- 64GB memory (5.3 GB/core, 1.6GHz RDIMMs)
- 500GB SAS HDD
- 1x Intel Xeon Phi 5110P coprocessor

Intel Sandy Bridge Xeon E5-2650

- 8 cores / 16 threads
- 2GH / 2.8GHz max Turbo
- 20MB shared Level 3 cache
  - 8x 256KB Level 2 cache
  - 8x 64KB (32+32) Level 1 cache
- 2 QPI links @ 8 GT/s
  - 8 x 20/8 x 64/80 = 16 GB/s
- 51.2 GB/s memory bandwidth @DDR3-1600
  - 1.6 (GT/s) x 8 (Byte) x 4 (channels) = 51.2 GB/s
- AVX (Advanced Vector Extensions)
  - 256-bit, 8 double-precision FLOP per cycle
  - SSE ➔ SSE2 ➔ SSE3 ➔ SSSE3 ➔ SSE4 ➔ AVX

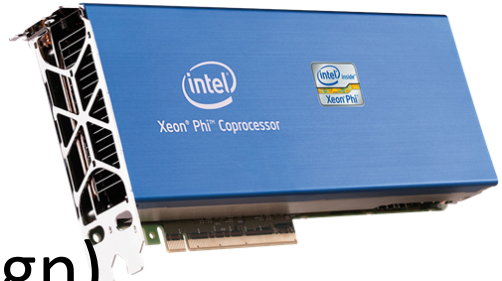


Nvdia Tesla K20 Kepler GPU

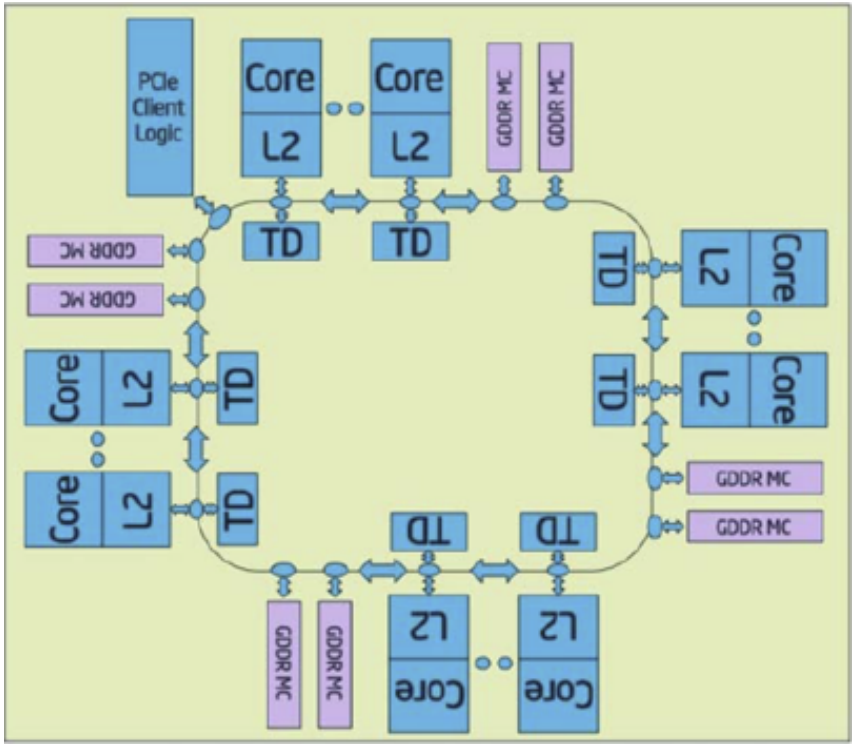
- 2496 CUDA cores / 832 DP units / 13 SMX
- Core speed: 705 MHz
- Double precision performance: 1.173 TFLOPS
  = 0.705 (GHz) x 832 (DP units) x 2 (FMA)
- Single precision performance: 4.577 TFLOPS
  = 0.705 (GHz) x 2496 (CUDA cores) x 2 (FMA)
- Memory: 5.2GHz, 320-bit wide, 5GB GDDR5
  5.2 (GHz) x 320 / 8 = 208 GB/s
- PCI express 2.0 x16
  500 (MB/s) x 8/10 x 16 = 8 GB/s (16 GB/s duplex)

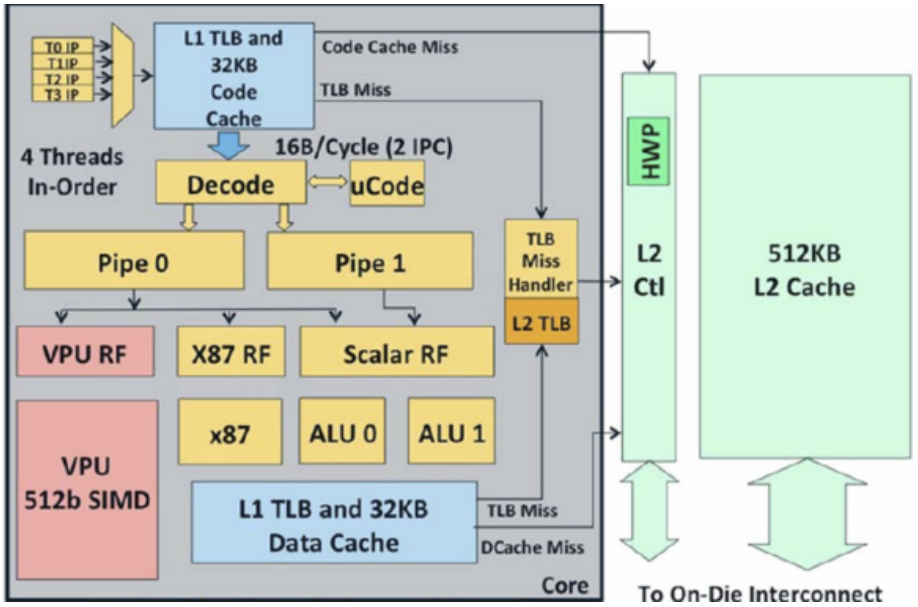

SMX
Instruction Cache
Warp Scheduler
Warp Scheduler
Warp Scheduler
Warp Scheduler
Dispatch
Register File (65,536 x 32-bit)
Core
DP Unit
LD/ST
SFU
Interconnect Network
64 KB Shared Memory / L1 Cache
48 KB Read-Only Data Cache
Tex

Intel Xeon Phi 5110P



- 60 cores (in-order, dual-issue x86 design)
- 4 threads per core
- Core speed: 1.053 GHz
- 512-bit AVX
- Double precision performance: 1.01 TFLOPS
  = 1.053 (GHz) x 60 (cores) x 512/64 x 2 (FMA)
- Memory: 8GB GDDR5
  5 (GT/s) x 16 (channels) x 4 (B) = 320 GB/s
- PCI express 2.0 x16
  500 (MB/s) x 8/10 x 16 = 8 GB/s (16 GB/s duplex)





http://www.tomshardware.com/reviews/xeon-phi-larrabee-stampede-hpc,3342-3.html

Key to performance: vectorization

Peak performance of Intel Xeon E5-2650

= 8 (AVX) x 8 (cores) x 2.0 (GHz) = 128 GFLOPS

Peak Performance of all Intel Xeon E5-2650s

= 128 GFLOPS x 2 x 188 (nodes) = 48 TFLOPS

Peak Performance of Nvidia K20 = 1.17 TFLOPS

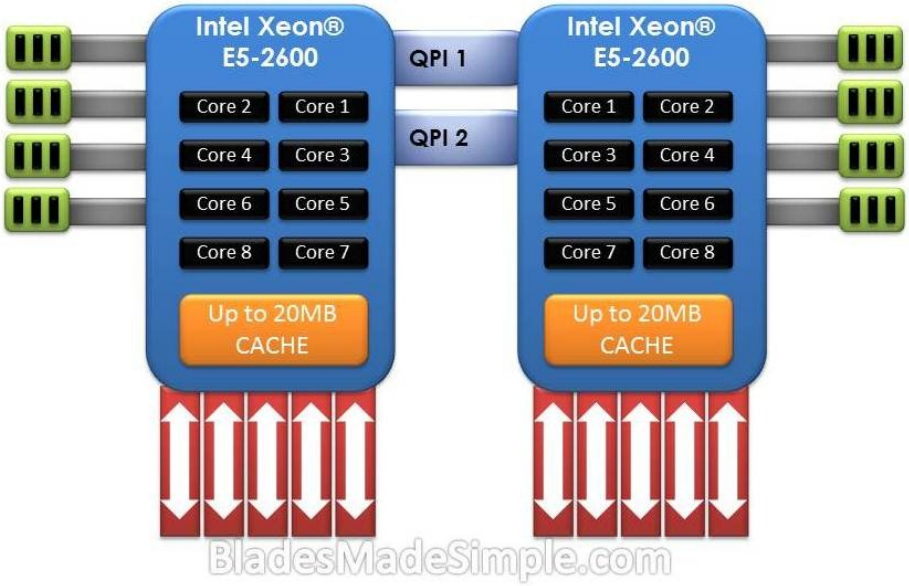
Peak Performance of Xeon Phi 5110P = 1.01 TFLOPS

Total Peak Performance of Hyades

= 48 + 8x1.17 + 1.01

= 58 TFLOPS

Key to performance: Data Locality / Communication Avoidance



	Bandwidth	Ratio
Intel Xeon E5-2650	128 x 4 x 2 x 8 = 8192 GB/s	1
Memory	51.2 GB/s	160
QPI	16 GB/s	512
PCIe 2.0 x16	8 GB/s	1024
QDR InfiniBand	4 GB/s	2048
SSD	500 MB/s	16384
HDD	100 MB/s	10000

Applications

- Massively parallel jobs
  - Pure MPI jobs
  - Hybrid (MPI + OpenMP) jobs
- Multithreaded jobs
- Embarrassingly parallel jobs
- Serial jobs
- Visualization / Analysis
- Big data jobs
  - Hadoop
- GPU computing
- MIC computing

Accessing Hyades

Only SSH access is allowed!

Master Node (hyades.ucsc.edu):

ssh -l username hyades.ucsc.edu

ssh username@hyades.ucsc.edu

Analysis Node (eudora.ucsc.edu):

ssh -l username eudora.ucsc.edu

ssh username@eudora.ucsc.edu

Using SSH key for authentication

Generate authentication key on your client computer:

```
cd $HOME/.ssh
ssh-keygen -b 2048 -t rsa -f hyades
```

Once the public key (hyades.pub) is uploaded to Hyades

```
ssh -i ~/.ssh/hyades -l username hyades.ucsc.edu
```

Add the following to $HOME/.ssh/config

```
host h
  HostName hyades.ucsc.edu
  User username
  IdentityFile ~/.ssh/hyades
```

Then a lot of keystrokes will be saved:

```
ssh h
```

Computing Environment

- 2 storage spaces are mounted on every node
- User home directories
  - NFS
  - /home/username
  - 3.6 TB total capacity
  - For source codes and scripts
- User work directories
  - Lustre file system
  - /pfs/username & symbolic link $HOME/pfs
  - 146 TB total capacity
  - For running simulations

Modules

We use the Modules utility to manage software environment

To list the loaded modules:

```
module list
```

To list the available modules:

```
module avail
```

To load a module (e.g., python 2.7)

```
module load python/2.7
```

To switch modules (e.g., to use a different MPI implementation):

```
module switch intel_mpi/4.1.0.024 rocks-openmpi_ib_intel
```

To learn about a module (e.g., IDL)

```
module help IDL
module display IDL
```

Compilers

- Intel (default and recommended)
- PGI
- GNU

MPI Implementations

- Intel MPI (default and recommended)
- Open MPI
- Mvapich
- Mpich

Compiling serial codes

- C

  icc [compiler_options] hello.c

- C++

  icpc [compiler_options] hello.cpp

- Fortran

  ifort [compiler_options] hello.f

  ifort [compiler_options] hello.f90

  Please consult Hyades wiki for Intel Compiler options.

Compiling OpenMP codes

- C

  icc -openmp -o omp_hello.x omp_hello.c

- C++

  icpc -openmp -o omp_hello.x omp_hello.cpp

- Fortran

  ifort -openmp -o omp_hello.x omp_hello.f

  ifort -openmp -o omp_hello.x omp_hello.f90

Compiling pure MPI codes

- C

  ```
  mpiicc -o mpi_hello.x mpi_hello.c
  ```

- C++

  ```
  mpiicpc -o mpi_hello.x mpi_hello.cpp
  ```

- Fortran

  ```
  mpiifort -o mpi_hello.x mpi_hello.f
  mpiifort -o mpi_hello.x mpi_hello.f90
  ```

Compiling hybrid (MPI + OpenMP) codes

- C

  mpiicc -mt_mpi -openmp -o mpi_hello.x mpi_hello.c

- C++

  mpiicpc -mt_mpi -openmp -o mpi_hello.x mpi_hello.cpp

- Fortran

  mpiifort -mt_mpi -openmp -o mpi_hello.x mpi_hello.f

  mpiifort -mt_mpi -openmp -o mpi_hello.x mpi_hello.f90

Running jobs

- Torque/PBS is the queuing system on Hyades
- Torque utilities
  - qsub ### job submission
  - qstat ### monitoring status of jobs
  - qdel ### terminating jobs prior to completion
- Maui is the job scheduler
- Maui utilities
  - showq ### list both active and idle jobs
  - showbf ### show what resources are immediately available
  - checkjob ### view details of a job in the queue

Queues on Hyades

- normal (default) queue
  - 16 cores per node (hyper-threading disabled)
  - 120 nodes
  - -q normal
  - -l nodes=4:ppn=16
- hyper queue
  - 32 cores per node (hyper-threading enabled)
  - 60 nodes
  - -q hyper
  - -l nodes=2:ppn=32

Sample PBS script for MPI jobs

```
#PBS -S /bin/bash                    ### shell
#PBS -N hello                        ### job name
#PBS -q normal                       ### queue name
#PBS -l nodes=4:ppn=16               ### 64 cores
#PBS -l walltime=4:00:00             ### 4 hours walltime
#PBS -M shaw@ucsc.edu                ### ask Torque to
#PBS -m abe                          send emails when
                                     job aborts, starts and
                                     ends
cd $PBS_O_WORKDIR
mpirun  -n 64 -env I_MPI_FABRICS shm:ofa ./mpi_hello.x
```

qsub mpi.pbs

Sample PBS script for OpenMP jobs

```
#PBS -S /bin/bash
#PBS -N hello
#PBS -q normal
#PBS -l nodes=1:ppn=16
#PBS -l walltime=4:00:00
#PBS -M shaw@ucsc.edu
#PBS -m abe

cd $PBS_O_WORKDIR
export OMP_NUM_THREADS=16
./omp_hello.x
```

```
### shell
### job name
### queue name
or #PBS -l ncpus=16
### 4 hours walltime
### ask Torque to
send emails when
job aborts, starts and
ends
```

qsub mpi.pbs

Sample PBS script for hybrid jobs

```
#PBS -S /bin/bash                    ### shell
#PBS -N hybrid                       ### job name
#PBS -q normal                       ### queue name
#PBS -l nodes=4:ppn=16               ### 64 cores
#PBS -l walltime=4:00:00             ### 4 hours walltime
#PBS -M shaw@ucsc.edu                ### ask Torque to send
#PBS -m abe                          emails when job aborts,
                                     starts and ends

cd $PBS_O_WORKDIR
export OMP_NUM_THREADS=8
export I_MPI_PIN_DOMAIN=omp
export KMP_AFFINITY=compact
mpirun  -env I_MPI_FABRICS shm:ofa -n 8 -ppn 2 ./hybrid.x
```

qsub mpi.pbs

Sample PBS script for serial jobs

```
#PBS -S /bin/bash
#PBS -N hello
#PBS -q normal
#PBS –l ncpus=1
#PBS -l walltime=4:00:00
#PBS -M shaw@ucsc.edu
#PBS -m abe

cd $PBS_O_WORKDIR
./hello.x
```

```
### shell
### job name
### queue name
### only 1 core
### 4 hours walltime
### ask Torque to
send emails when
job aborts, starts and
ends
```

qsub serial.pbs

Sample script for embarrassingly parallel jobs

```
#PBS -S /bin/bash
#PBS -N jobarray
#PBS -q normal
#PBS -l ncpus=1
#PBS -t 0-31
#PBS -j oe
#PBS -l walltime=4:00:00

cd $PBS_O_WORKDIR
./hello.x $PBS_ARRAYID
```

```
./hello.x 0
./hello.x 1
…
…
./hello.x 31
```

qsub jobarray.pbs

Visualization & Analysis

Eudora is the dedicated viz node

```
ssh -Y -i ~/.ssh/hyades -l username eudora.ucsc.edu
```

Add the following to $HOME/.ssh/config

```
host e
  HostName eudora.ucsc.edu
  User username
  IdentityFile ~/.ssh/hyades
  ForwardX11 yes
  ForwardX11Trusted yes
```

Then a lot of keystrokes will be saved:

```
ssh e
```

Viz tools

- VisIt

  module load VisIt

- Yt

  module load yt

- IDL

  module load IDL

- Python (matplotlib, SciPy, mpi4py, etc)

  module load python

Q&A

Wiki:

http://pleiades.ucsc.edu/hyades/