Advanced HPC course

.... research begins with you.

Agenda

1) Quick Overview of HPC and Hex (15 mins)
2) HPC Job Submission (60 min)

break (30 min)

3) Software Compile / Installs / Misc (60min)

Module 1:
Quick Overview of HPC and Hex

What is HPC ?

- HPC, or high-performance computing, refers to the application of supercomputers or clusters of computers to computational problems that typically arise through scientific inquiry.
- HPC is useful when a computational problem:
  - **Is too large** to solve on a conventional laptop or workstation (because it requires too much memory or disk space) or ...
  - **Would take too long** (because the algorithm is complex, the dataset is large, or data access is slow) or ...
  - **Are too many** – High Throughput Computing

Reasons to use UCT HPC ?

- You have a program that can be recompiled or reconfigured to use optimized numerical libraries that are available on HPC systems but not on your own system.

- You have a "parallel" problem, e.g. you have a single application that needs to be rerun many times with different parameters.
- You have an application that has already been designed with parallelism
- To make use of the large memory available
- Our facilities are reliable and regularly backed up

When not to use HPC ?

- When applications require databases. Databases which run on single nodes.
- GUI applications (depends on the build type of the application). DepthMapX

Parallelism on HPC

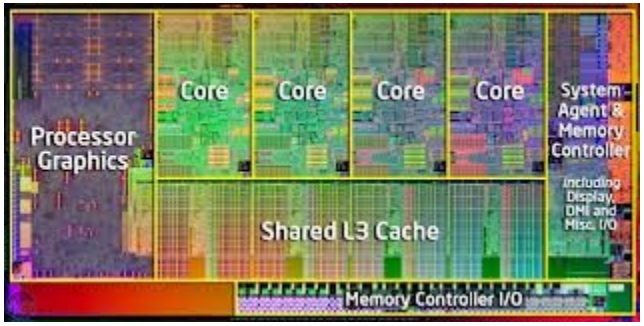
- Programs for HPC systems must be split up into many smaller “sub-programs“ which can be executed in parallel on different processors

- Writing parallel software can be challenging, and many existing software packages do not support parallelism & may require development.

NOTE: Many tasks cannot be parallelised

What does HPC consist of ?

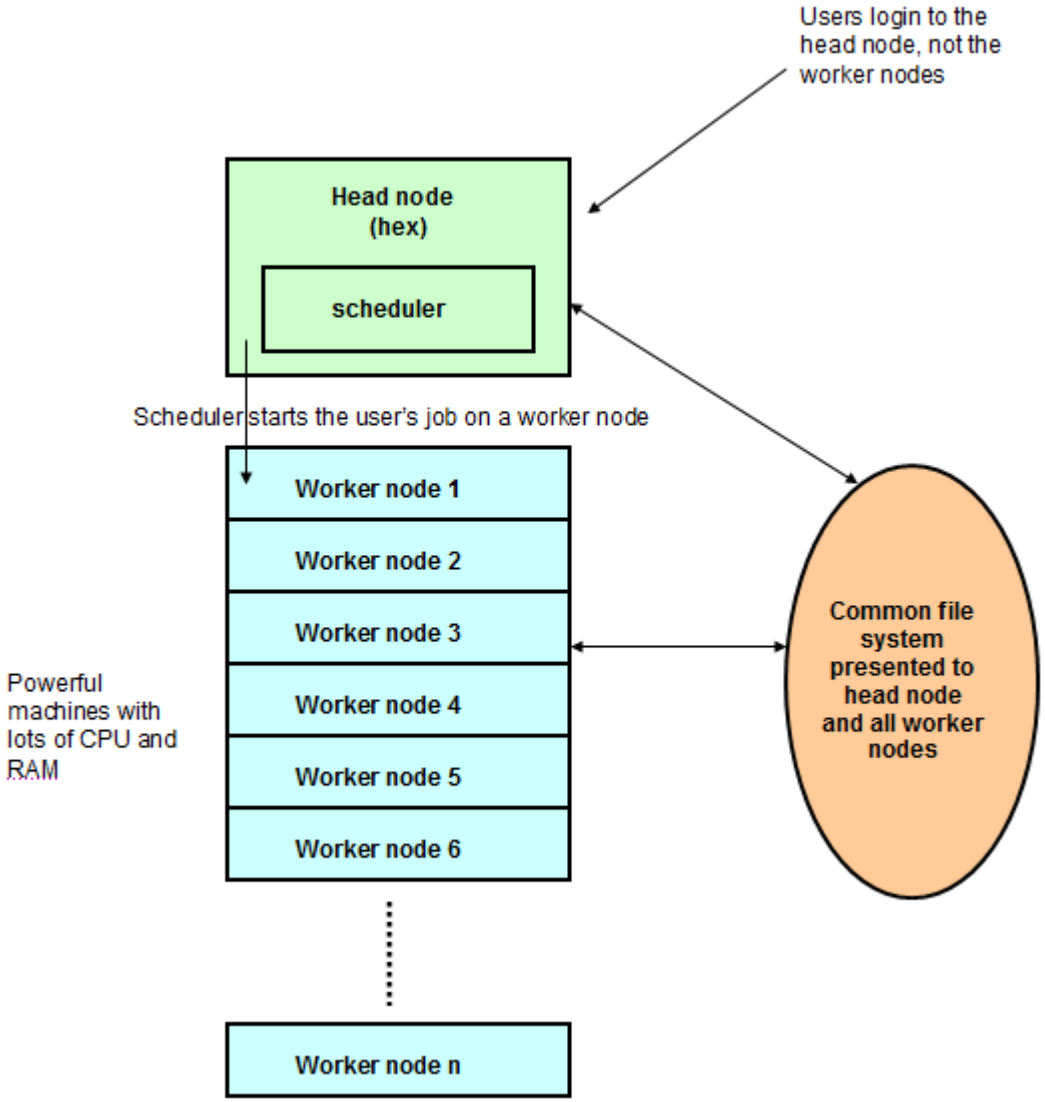
- HPC is the aggregation of computing resources.
  - Cores (cpus / sockets)
  - RAM
  - Disk
  - Interconnect



Hex Cluster Architecture

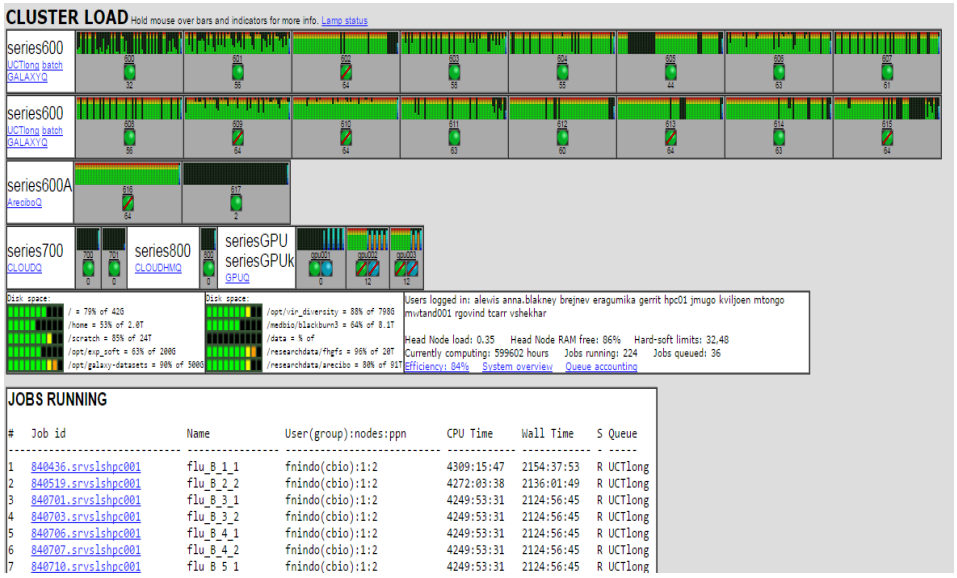
- Operating system: SLES 11sp3
- X86_64
- HPC server WLM: Torque PBS
- Scheduler: Maui
- Worker nodes:
  - 9x Dell C6145 – Many Core / dense array
  - 3x SuperMicro GPU servers (Tesla M2090 / K40)
  - 2x High Memory Machines – Dell R820 – 1TB RAM (TBD)
  - 2x High Memory Virtual Machines (HMVM)
- FhGFS Storage nodes:
  - 4x Dell R620s
  - 4x Dell MD1220 – 28.8 TB each RAID6 (1HS), 92TB usable

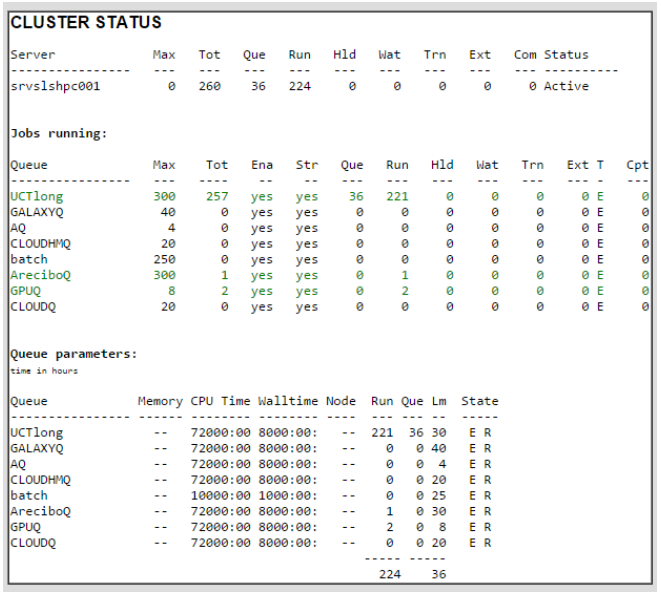
Architecture



The dashboard

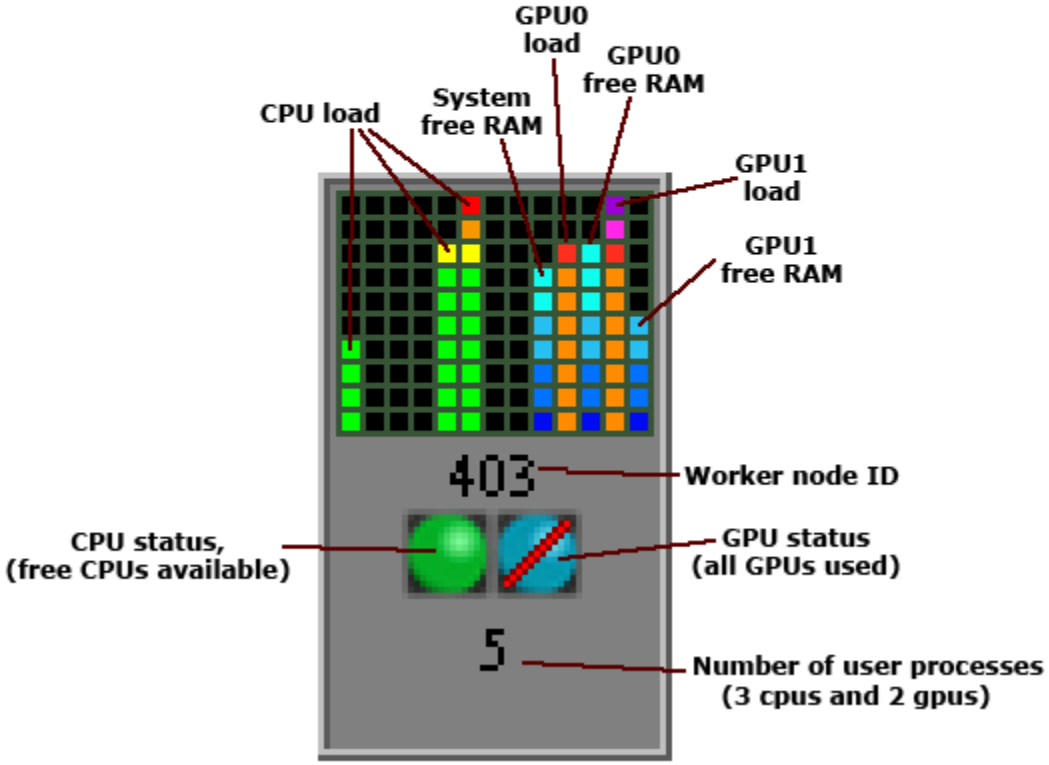
- To keep track of the cluster's status, workload and the jobs that are running go to: http://hex.uct.ac.za





The dashboard

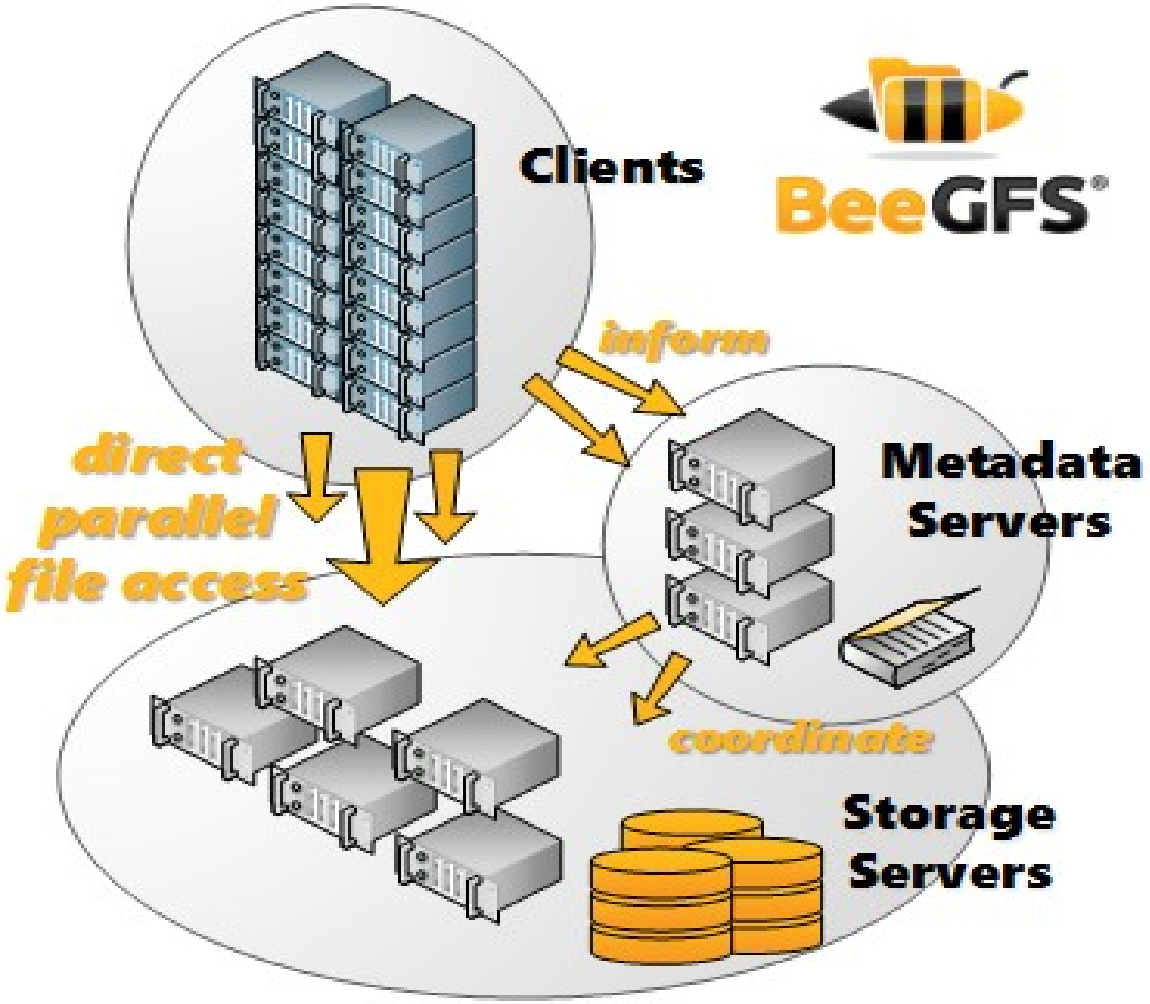
Icon	Value	Description
	Free CPUs	There are free CPUs, jobs may be submitted to this node.
	Job-exclusive	All CPUs are busy, the node is running but no further jobs may be submitted.
	Busy	Torque mom daemon or CPUs too busy to respond to further requests. Jobs are running but may be degraded.
	Down	Node down or PBS mom daemon offline or not responding, no jobs may be submitted.
	Free GPUs	There are free GPUs, jobs may be submitted.
	Busy	All GPUs are busy, the node is running but no further jobs may be submitted.



FhGFS / BeeGFS Parallel Storage

- Pure software solution for scale-out parallel network-storage.
- Each HPC node is connected with IB cables to the IB switch. The FhGFS store is connected to the same switch.
  - /researchdata/fhgfs/ (will soon change to /scratch)
- Advantages : Very very very fast storage
- Disadvantages: No backups, “volatile” area, cleanup required.

FhGFS / BeeGFS Architecture



FhGFS / BeeGFS connected to HEX

- Parallel storage is connected via Infiniband (RDMA only). The only TCP connection which exists is for Admon / MGMT services.
- TCP is the backup protocol should RDMA (IB switch) fail.
- Headnode maps the FhGFS store as TCP over 1gb/sec unfortunately.
- Once your job executes on a worker node, traffic to the storage service is 56gb/sec

Module 2:
Various Job Submission Methods – Interactive

Software Required

Use your web browser to download Putty and PuttySCP from: http://www.putty.org

- Click on the “Download Putty” link and download:
  - putty.exe (a Telnet and SSH client)
  - WinSCP (GUI-BASED SCP)

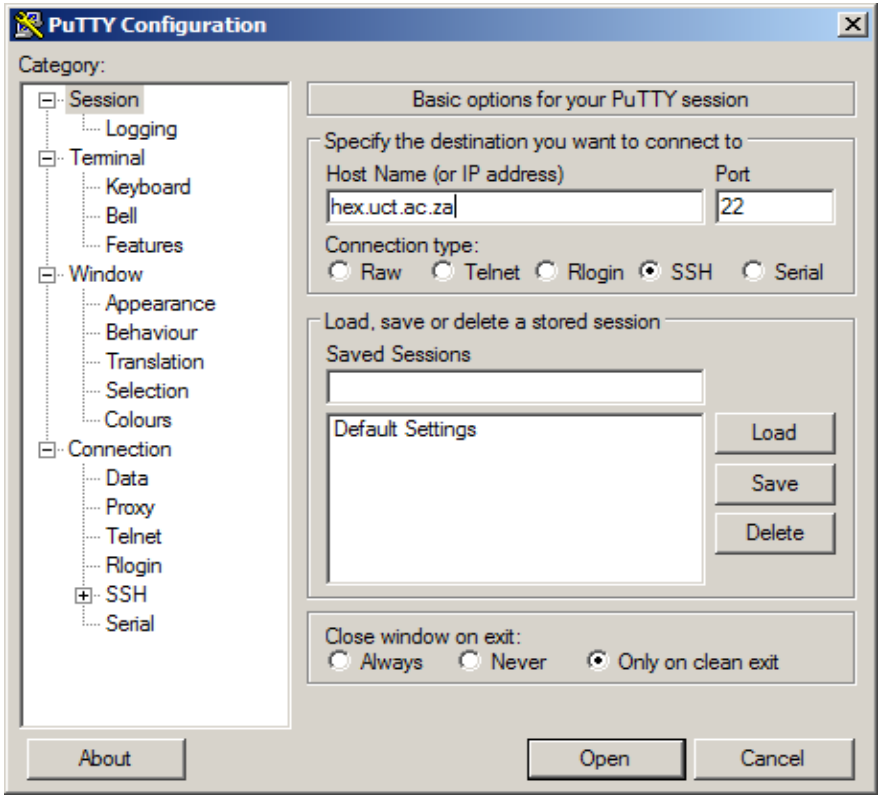
- Double click to install on your PC.
- MacOS users may launch a terminal

- Xming for Windows (**Tick NoACL**) / MacOSX may use Quartz

Course Credentials

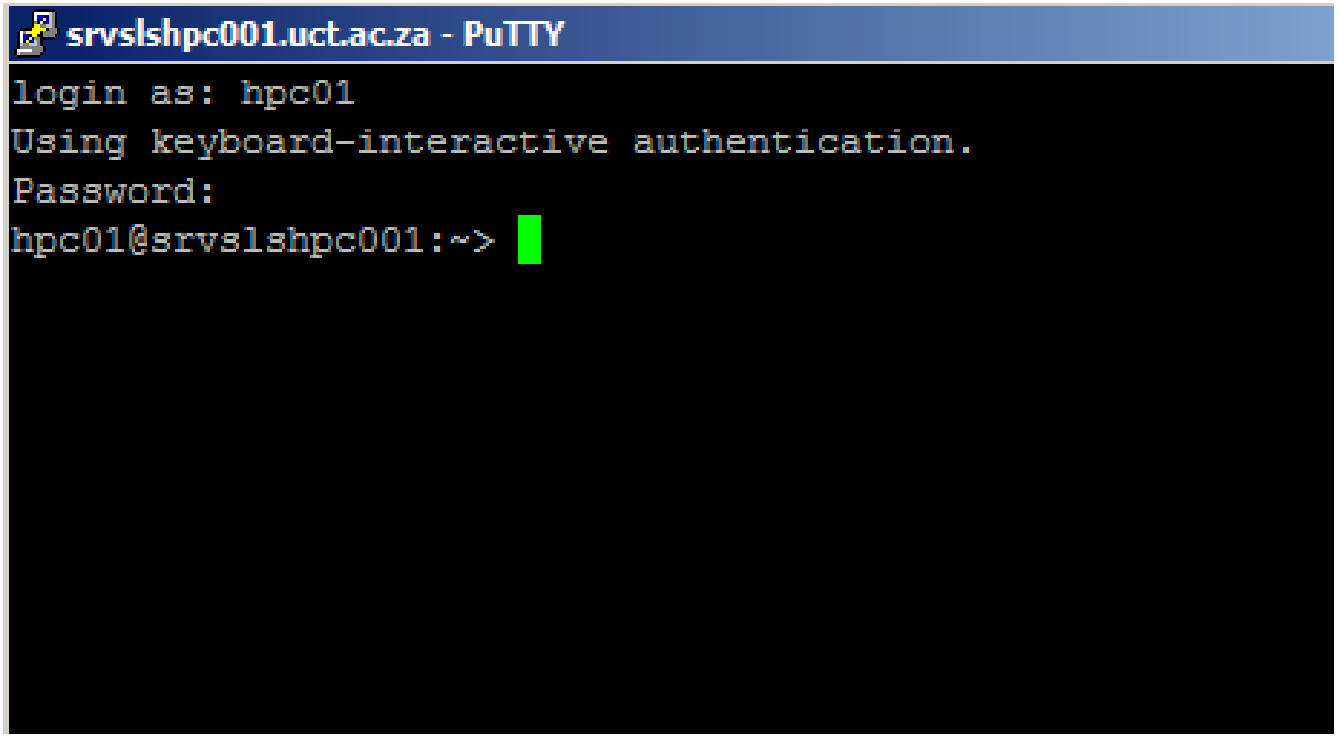
- Start the putty telnet/ssh client by double clicking on putty.exe and connect to the HPC Machine
  - Host: **`hex.uct.ac.za`**
  - Connection Type: **`ssh`**
  - Port: **22**

1. Select SSH, X11, Enable X11
2. Click on session, top left.
3. Save the session: Hex,
4. Click the **save** button.



Course Credentials

- Log into the training HPC system using the Test Account allocated to you, e.g.
  - **Account Name:** hpc0(n)
  - **Password:** train0(n)



Git clone eResearchUCT

- Git clone the example scripts from bitbucket.org

“git clone https://bitbucket.org/eresearchUCT/training-material.git”

- Change directory into “ training-material/pbs-sample-scripts “

Standard Job Submission

```
#PBS -N Standard-job
#PBS -l nodes=1:series600:ppn=1
#PBS -q UCTlong
#PBS -l walltime=1:00:00
#PBS -V

hostname -f
sleep 30
```

Exercise 1 (a)

Command	Description
• **Add the #PBS -M e-mail-addy** • **Add #PBS abe directives** • **qsub standard-job.pbs**	Submit Standard Job

Array Jobs

- Use Case: Lots of input files, not possible to submit manually.

- Common PBS Environment Variables
  - $PBS_ARRAYID
  - $PBS_JOBID

Exercise 1 (b)

Command	Description
• **qsub -t 1-10 array-job.pbs**	Submit Array Job

Interactive Jobs

- Edit the file - “**interactive-job.pbs “**
- Use Case: Compiling, Debug Application / Testing,
- Advantage: Work directly on a worker node
- Disadvantage: CPU expensive. Get done and exit

Exercise 1 (c)

Command	Description
• **qsub interactive-job.pbs**	Submit Interactive Job. Note: #PBS -I

Interactive Jobs with X support

- Edit the file "**interactive-X-job.pbs**"
- Use Case: Compiling, Debug Application / Testing,
- Advantage: Work directly on a worker node
- Disadvantage: CPU expensive. Get done and exit

Exercise 1 (d)

Command	Description
• **qsub interactive-X-job.pbs**	Submit Interactive X Job. Note: #PBS -I -X

MPI Jobs

- Message Passing Interface (MPI) is used for communication among the nodes running a parallel program on a distributed memory system.
- Compile mpitest.c - “mpicc -o mpitest mpitest.c”
- “qsub mpi-job.pbs“
- Important to use mpicc and mpirun from the same openmpi version.

Modules

- Switching between multiple versions of the same application.
- Use Case: Single job requires functionality from one version of a application and functionality from another version of the same application.
- Sets up Library / Include / Bin / Custom Paths
- “module avail “ - Lists all modules available
- “module load <module>” - Loads a specific module
- Available on headnode only. Available on worker node including the -V #PBS directive.

Modules Exercise 1(f)

`module avail`	*Shows all modules available*
`module load python/anaconda-python-2.7`	*Environment modified for application*
`which python`	*Location for which binary*
`module unload python/anaconda-python-2.7`	*Unload the module*

#PBS -N Tea Time
#PBS walltime=00:30:00

Module 3:
Software Compile / Installs / Misc

The Hex software repository does not contain my software

- All software resides in /opt/exp_soft and shared between the HPC worker nodes using NFS. Please do not store on /scratch.
- Problem: I have a RPM file but cannot install because I do not have root priviledges. Solution: rpm –prefix=/home/username/install-dir -i app.rpm
- Problem: I have the source but its such a mission to compile. Solution: (1)Make a list of dependencies, (2)download install, (3) Compile and view logs
- Roadmap: UCT HPC Continious Intergration environment for keeping software up-to-date

Establish a HPC interactive session

- Update the interactive-job.pbs PPN value from 1 to 2
- “qsub interactive-job.pbs”

PEAR - Paired-End reAd mergeR

- Software for merging raw illumina paired-end reads
- One of many Open Source tools in the Bio-Informatics software catalogue.
- It is one of the most popular tools currently being used on our HPC.
- Quick and simple to compile.
- .. however a lot of people are put off by how long it takes to compile an application, GCC being one of them.

Lets compile some software

- mkdir ~/pear-install
- Change directory into ~/training-material/software-src/
- Uncompress with “ tar xfvz pear-0.9.6-src.tar.gz “
- Change directory into pear-0.9.6-src
- “./configure -- help “ for a list of features and tuning parameters
- “./configure --prefix=/home/username/pear-install/”
- “make -j 2” - Compile the application. “ -j2 “ ??
- “make install” - Install the compiled binary / lib / include

Working remotely with screen

- Allows you create additional virtual terminals inside a single process called “ Screen “
- Use Cases:
  - Works great for unrealible internet connections
  - Long running compilations / file copies
- Execute the command called “ screen “
- “ctrl + a +c “ - Create additional terminals
- “ctrl +a + n or p” - Move back / forward between tty
- “ctrl +a +d “ - Detach from a screen session
- “screen -r -d “ - detach and re-attach
- “screen -x “ - reattach but keep my remote sys active

Being put off from screen because it doesn't scroll

- “ termcapinfo xterm|xterms|xs|rxvt ti@:te@ “

Road Map for UCT HPC 2015

- High Memory Machines (1TB)
- New Cluster , Intel based , more cores
- New scheduler and workload manager (SLURM)

support for UCT Active Directory – authentication

- Expand FhGFS
- Implementation of a CI system
  - Automatically build / apply regretion tests / deploy to hex software repository
  - Automatically build the revelant “module load “ scripts
- Quotas for home directories / scratch
- Better visualization support VirtualGL / TurboVNC

Thank You
Questions ?

Apply for a HPC account
http://srvslnhpc001.uct.ac.za/eresearch/