Goal: Establish TACC as a leading center for *ENABLING* computational biology research.

- Attract Life Science investigators to use TACC computing resources
- Establish collaborative efforts with investigators wanting to develop/implement HPC for their work
- Develop new/novel technologies to further enable computational biology research
Computational Biology @ TACC

– Dynamic core group of 5 members with complimentary backgrounds in structural biology, genomics, mathematics, cell biology, computer science

– Utilization of TACC systems for Biological research has greatly increased from ~18% to over 25%

– Support a growing software stack to support broad range of life science research
Variety of Advanced Computing Resources

High Performance Computing

Data Storage / Archive Systems

Software & Tools
High Performance Computing

Data Storage / Archive Systems

Software & Tools
Ranger
Ranger

- **Compute power - 504 Teraflops**
  - 3,936 four socket nodes
  - 62,976 cores, 2.0 GHz AMD Opteron

- **Memory - 125 Terabytes**
  - 2GB/core, 32 GB/node

- **Disk subsystem**
  - 1.7 PB Storage (Lustre Parallel File System)
  - 1 PB in /work filesystem

- **Interconnect**
  - 8 Gb/s InfiniBand
Lonestar

• **Main Compute Nodes - 302 Teraflops**
  – 1,888 compute nodes, 22,656 cores, 3.33 GHz Intel “Westmere”
  – 24 GB/node (44 TB total)

• **Large Memory Compute Nodes**
  – 5 nodes, 120 cores, 3.33 GHz Intel “Westmere”
  – 1TB RAM/node (5TB total)

• **Disk subsystem**
  – 1 PB global storage (Lustre Parallel File System)
  – 276 TB local storage (146GB/node)

• **Interconnect**
  – 40 Gbps QDR InfiniBand
Longhorn

- **256-Node Viz and Compute System**
  - 240 Dell R610 Compute Nodes, each with:
    - 2 X Intel Nehalem quad-core processors (8 cores) @ 2.53 GHz
    - 48GB RAM
    - 73GB local disk
  - 16 Dell R710 Compute Nodes, each with
    - 2 X Intel Nehalem quad-core processors (8 cores) @ 2.53 GHz
    - 144GB RAM
    - 73GB local disk
  - 128 NVIDIA Quadro Plex S4s, each with 4 NVIDIA FX 5800 GPUs
  - Mellanox QDR InfiniBand Interconnect
  - 210TB global file system, managed by the Lustre Parallel File System
Advanced Visualization Systems

- **ACES Visualization Laboratory**
  - 307 Mpixel Tiled-Display System (Stallion)
  - Sony 9M Pixel Projection System (Bronco)
  - 3D Capable 82” DLP Display (Mustang)
High Performance Computing

Data Storage / Archive Systems

Software & Tools
Corral

- High Performance Data Applications Facility
  - 1.2 Petabytes raw capacity
  - Lustre file system accessible from Ranger, Lonestar and Longhorn head nodes only
  - Support for MySQL and Postgres databases
Ranch Archival System
Ranch Archival System

- **STK SL8500 Modular Library System**
  - 5,000 Cartridge Capacity, 500 GB per Cartridge (Native, uncompressed)
  - 2.5 PB Storage Capacity, upgrading to 10PB
  - Accessed through $ARCHIVE environment variables
Resource Allocations - How Do I Get Time on TACC Systems?

- **NO COST** for using TACC systems for academic research

- Time on TACC systems is acquired by submitting an allocation request

- Open to all US academic researchers through the TeraGrid allocations process

- Resources are allocated on a project basis to a single principal investigator (PI)
TACC Allocations

• **UT system researchers**
  – Apply directly through TACC portal (portal.tacc.utexas.edu)
  – Up to 500,000 SUs on Ranger and Lonestar
  – Above 500K, apply through TeraGrid

• **All other US academic researchers**
  – Apply through TeraGrid (www.teragrid.org/web/user-support/getting_started)
  – Startup allocations up to 250k on Ranger and Lonestar approved quickly, can apply anytime
  – Allocations >250k must go through research proposal process, quarterly review, more rigorous
Support Services - How do I get Help?

• **Technical documentation**
  – Extensive documentation available
    • www.tacc.utexas.edu/user-services/user-guides/

• **Consulting**
  – Users submit issues via the TACC User Portal
    • portal.tacc.utexas.edu/

• **Training**
  – Taught on-site, sign up at TACC User Portal
    • www.tacc.utexas.edu/services/training/
Key Websites

• **TACC**
  - **User portal**  portal.tacc.utexas.edu
  - **New user info**  www.tacc.utexas.edu/general/newusers/
  - **User guides**  www.tacc.utexas.edu/services/userguides/

• **TeraGrid**
  - **Allocations**  www.teragrid.org/userinfo/access/allocations.php
  - **Proposals**  pops-submit.teragrid.org/
  - **Support & Docs**  www.teragrid.org/userinfo/index.php
System Architecture and Filesystems
Filesystem Overview

• **HOME**
  – Store your source code and build your executables here
  – Use $HOME to reference your home directory in scripts

• **WORK**
  – Store large files here
  – This file system is NOT backed up, use $ARCHIVE for important files!
  – Use $WORK to reference this directory in scripts

• **SCRATCH**
  – Store large input or output files here - TEMPORARILY
  – This file system is NOT backed up, use $ARCHIVE for important files!
  – Use $SCRATCH to reference this directory in scripts

• **ARCHIVE**
  – Massive, long-term storage and archive system
## Ranger Filesystem

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Quota</th>
<th>Backed-up?</th>
<th>Purged?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HOME</td>
<td>6GB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>$WORK</td>
<td>350GB</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>$SCRATCH</td>
<td>400TB</td>
<td>No</td>
<td>Yes - 10 days</td>
</tr>
</tbody>
</table>
# Lonestar Filesystem

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Quota</th>
<th>Backed-up?</th>
<th>Purged?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HOME</td>
<td>1GB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>$WORK</td>
<td>250GB</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>$SCRATCH</td>
<td>4TB</td>
<td>No</td>
<td>Yes - 10 days</td>
</tr>
<tr>
<td>/tmp</td>
<td>~65GB</td>
<td>No</td>
<td>Yes - Immediately</td>
</tr>
</tbody>
</table>

Wednesday, August 10, 11
Ranch Filesystem

- At some point, you may (will!) generate enough data to run into system disk quotas

- Ranch is TACC’s long-term file storage system (ranch.tacc.utexas.edu)

- Users can backup and store all data to Ranch

- Data will be maintained for AT LEAST 1yr after end of allocation

*Ranch is a tape system and accessing files is relatively slow...*
Filesystems Summary

• Ranger and Lonestar have similar but distinct filesystem architectures

• Ranger has no local storage, Lonestar has 146GB/node local disk (/tmp)

• Only your HOME directory is backed up automatically, use $ARCHIVE for important files

TACC staff may delete files from WORK and SCRATCH if the file system becomes full, even if files are less than 10 days old. The use of programs or scripts to actively circumvent the file purge policy will not be tolerated.
User Environment, Login and Software Modules
Overview

• Login and interaction with TACC systems occurs primarily through a terminal shell

• Applications, libraries, compilers and other environment variables are controlled with the “modules” utility. Module commands set up a basic environment for the default compilers, tools, and libraries.

• These environment variables are automatically kept up-to-date when system and application software is upgraded.
Initial Login

• Login with SSH

  % ssh trainXXX@lonestar.tacc.utexas.edu

• Connects you to one of the login nodes on Lonestar

• Default modules and environment variables are set upon login
Initial Login

```
Disk          Usage (GB)  Limit  %Used  File Usage  Limit  %Used
/share       0.0          6      0.43  1516      100000  1.52
/work        282.7       350    80.77 175807     2000000 8.79

login3.ranger.tacc.utexas.edu~: module list
Currently Loaded Modules:
1) TACC-paths
2) Linux
3) cluster-paths
4) pg/7.2-5
5) mvapich/1.0.1
6) binutils-amd/0.70220
7) TERAGRID-paths
8) gx-map/0.5.3.3
9) srb-client/3.4.1
10) tg-policy/0.2
11) tgproxy/0.9.1
12) tgresid/2.3.4
13) tgusage/3.0
14) uberftp/2.6
15) tginfo/1.0.1
16) TERAGRID-BASIC
17) globus/4.0.8
18) GLOBUS-4.0
19) TERAGRID-DEV
20) CTSSV4
21) gzip/1.3.12
22) tar/1.22
23) cluster
24) TACC

login3.ranger.tacc.utexas.edu~:
```
Initial Login

---

Project balances for user mbgonzo

<table>
<thead>
<tr>
<th>Name</th>
<th>Avail SU</th>
<th>Expires</th>
<th>Name</th>
<th>Avail SU</th>
<th>Expires</th>
</tr>
</thead>
<tbody>
<tr>
<td>TG-MCB110022</td>
<td>50000</td>
<td></td>
<td>CompBioApps</td>
<td>100000</td>
<td></td>
</tr>
<tr>
<td>TG-STA060015N</td>
<td>655362</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Disk quotas for user mbgonzo

<table>
<thead>
<tr>
<th>Disk</th>
<th>Usage (GB)</th>
<th>Limit</th>
<th>%Used</th>
<th>File Usage</th>
<th>Limit</th>
<th>%Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home1</td>
<td>0.0</td>
<td>1.1</td>
<td>0.03</td>
<td>95</td>
<td>101000</td>
<td>0.09</td>
</tr>
<tr>
<td>/work</td>
<td>6.7</td>
<td>250.0</td>
<td>2.67</td>
<td>36692</td>
<td>2000000</td>
<td>1.83</td>
</tr>
</tbody>
</table>

login1.ls4.tacc.utexas.edu: module list
Currently Loaded Modules:
1) TACC-paths
2) Linux
3) cluster-paths
4) intel/11.1
5) mvapich2/1.6
6) gzip/1.3.12
7) tar/1.22
8) cluster
9) TACC
Customizing Your Environment

• Default login shell is “bash”

• Login shell can be set
  – `%chsh -s <login shell>`
  – `%chsh -l` (lists available login shells)
  – Takes some time to propagate (~1 hour)

• Each shell reads a set of configuration scripts
## Startup Scripts

User-customizable config scripts for C-shell type shells (csh, tcsh)

<table>
<thead>
<tr>
<th>Ranger</th>
<th>Lonestar</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HOME/.cshrc_user</td>
<td>$HOME/.cshrc</td>
</tr>
<tr>
<td>$HOME/.login_user</td>
<td>$HOME/.login</td>
</tr>
</tbody>
</table>

User-customizable config scripts for bash shell

<table>
<thead>
<tr>
<th>Ranger</th>
<th>Lonestar</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HOME/.profile_user</td>
<td>$HOME/.profile</td>
</tr>
<tr>
<td>$HOME/.bashrc_user</td>
<td>$HOME/.bashrc</td>
</tr>
</tbody>
</table>
Applications, libraries, compilers and other environment variables are controlled with the “modules” utility. Module commands set up a basic environment for the default compilers, tools, and libraries.

Available modules depend on:
- The compiler (eg. PGI, Intel)
- The communication/MPI stack selected

To unload all optional modules:
- %module purge

To return to default modules:
- %module purge; module load TACC
The default modules are suitable for many users.

If you change compilers, be sure to also reload MPI stacks and other high-level software.

```
module purge
module load <compiler>
module load <mpi stack>
module load <high-level software, eg. mpiBLAST>
```
# Exploring Modules

## Useful module commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>% module</code></td>
<td>{lists options}</td>
</tr>
<tr>
<td><code>% module list</code></td>
<td>{lists loaded modules}</td>
</tr>
<tr>
<td><code>% module avail</code></td>
<td>{lists available modules}</td>
</tr>
<tr>
<td><code>% module spider</code></td>
<td>{lists ALL modules}</td>
</tr>
<tr>
<td><code>% module load &lt;module&gt;</code></td>
<td>{add a module}</td>
</tr>
<tr>
<td><code>% module unload &lt;module&gt;</code></td>
<td>{remove a module}</td>
</tr>
<tr>
<td><code>% module swap &lt;mod1&gt; &lt;mod2&gt;</code></td>
<td>{swap two modules}</td>
</tr>
<tr>
<td><code>% module help &lt;mod1&gt;</code></td>
<td>{module-specific help}</td>
</tr>
</tbody>
</table>
Exploring Modules

Modules available before selecting compiler

<table>
<thead>
<tr>
<th>Module</th>
<th>Version</th>
<th>Module</th>
<th>Version</th>
<th>Module</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBUS-4.0</td>
<td></td>
<td>condor/7.4.4</td>
<td></td>
<td>gx-map/0.5.3</td>
<td></td>
</tr>
<tr>
<td>GLOBUS-5.0</td>
<td></td>
<td>cctsv4</td>
<td></td>
<td>gx-map/0.5.3</td>
<td></td>
</tr>
<tr>
<td>TERAGRID-BASIC</td>
<td></td>
<td>cue-build</td>
<td></td>
<td>koomie/test.1.0.1b</td>
<td></td>
</tr>
<tr>
<td>TERAGRID-DEV</td>
<td></td>
<td>cee-comm</td>
<td></td>
<td>mycluster/2.1.0</td>
<td></td>
</tr>
<tr>
<td>TERAGRID-prods</td>
<td></td>
<td>cue-login-env</td>
<td></td>
<td>pacman/3.20</td>
<td></td>
</tr>
<tr>
<td>apache-ant/1.6.5</td>
<td></td>
<td>cee-tg</td>
<td></td>
<td>pacman/3.26</td>
<td></td>
</tr>
<tr>
<td>condor-g/6.7.18</td>
<td></td>
<td>cee-math</td>
<td></td>
<td>pacman/3.29 (default)</td>
<td></td>
</tr>
<tr>
<td>condor-g/7.2.1</td>
<td></td>
<td>cee-tg</td>
<td></td>
<td>pacman/3.29 (default)</td>
<td></td>
</tr>
<tr>
<td>condor-g/7.4.4 (default)</td>
<td></td>
<td>globus/4.0</td>
<td></td>
<td>pacman/3.29 (default)</td>
<td></td>
</tr>
<tr>
<td>condor/6.7.18</td>
<td></td>
<td>globus/4.0</td>
<td></td>
<td>srb-client/3.4.1</td>
<td></td>
</tr>
<tr>
<td>condor/6.9.3</td>
<td></td>
<td>globus/4.0</td>
<td></td>
<td>srb-client/3.4.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>globus/5.0</td>
<td></td>
<td>teragrid-basic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>globus/5.0</td>
<td></td>
<td>teragrid-dev</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>globus/5.0.2</td>
<td></td>
<td>tg-policy/0.2</td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Module</th>
<th>Version</th>
<th>Module</th>
<th>Version</th>
<th>Module</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/2.10.1</td>
<td></td>
<td>git/1.6.3.1</td>
<td></td>
<td>launch/1.3</td>
<td></td>
</tr>
<tr>
<td>R/2.12.1 (default)</td>
<td></td>
<td>git/1.7.4 (default)</td>
<td></td>
<td>postgres/8.3.6</td>
<td></td>
</tr>
<tr>
<td>atlas/3.9.32</td>
<td></td>
<td>gmake/3.81</td>
<td></td>
<td>python/2.5.2</td>
<td></td>
</tr>
<tr>
<td>autotools/1.0</td>
<td></td>
<td>gmp/4.2.4</td>
<td></td>
<td>python/2.6.4</td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td></td>
<td>gnuplot/4.2.6</td>
<td></td>
<td>python/2.6.5</td>
<td></td>
</tr>
<tr>
<td>bintils-omd/070220</td>
<td></td>
<td>gotoblas/1.23 (default)</td>
<td></td>
<td>python/2.7.1 (default)</td>
<td></td>
</tr>
<tr>
<td>cmake/2.8.1</td>
<td></td>
<td>gsl/1.11</td>
<td></td>
<td>scons/2.0.1</td>
<td></td>
</tr>
<tr>
<td>ddt/2.3.1</td>
<td></td>
<td>gsl/1.13 (default)</td>
<td></td>
<td>star-cm/4.0.011</td>
<td></td>
</tr>
<tr>
<td>ddt/2.6 (default)</td>
<td></td>
<td>gzip/1.3.12</td>
<td></td>
<td>star-cm/4.0.011 (default)</td>
<td></td>
</tr>
<tr>
<td>gcc/4.2.0</td>
<td></td>
<td>intel/10.1 (default)</td>
<td></td>
<td>subversion/1.6.1</td>
<td></td>
</tr>
<tr>
<td>gcc/4.3.2</td>
<td></td>
<td>intel/11.1</td>
<td></td>
<td>sun/12</td>
<td></td>
</tr>
<tr>
<td>gcc/4.4.0</td>
<td></td>
<td>intel/9.1</td>
<td></td>
<td>tar/1.22</td>
<td></td>
</tr>
<tr>
<td>gcc/4.4.1</td>
<td></td>
<td>irods/2.1</td>
<td></td>
<td>vis/1.0</td>
<td></td>
</tr>
<tr>
<td>gcc/4.4.3</td>
<td></td>
<td>jdk32/1.6.0</td>
<td></td>
<td>zlib/1.2.3</td>
<td></td>
</tr>
<tr>
<td>gcc/4.4.5 (default)</td>
<td></td>
<td>jdk64/1.6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Module</th>
<th>Version</th>
<th>Module</th>
<th>Version</th>
<th>Module</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td></td>
<td>TACC-paths</td>
<td></td>
<td>java/1.5.0</td>
<td></td>
</tr>
<tr>
<td>TACC</td>
<td></td>
<td>cluster-paths</td>
<td></td>
<td>java/1.4.2</td>
<td></td>
</tr>
</tbody>
</table>

login3.ranger.tacc.utexas.edu:
Exploring Modules

More modules available after selecting compiler
Exploring Modules

Even more modules after selecting MPI stack

%module load mvapich
%module avail

Exploring Modules

Wednesday, August 10, 11
## Computational Biology Modules

<table>
<thead>
<tr>
<th>Application</th>
<th>Ranger/ Lonestar</th>
<th>Module</th>
<th>Compiler</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpiBLAST</td>
<td>R</td>
<td>mpiblast</td>
<td>intel</td>
<td>mvapich</td>
</tr>
<tr>
<td>BLAST++</td>
<td>R</td>
<td>blast</td>
<td>gcc</td>
<td>n/a</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>BioPerl</td>
<td>R</td>
<td>BioPerl</td>
<td>gcc</td>
<td>n/a</td>
</tr>
<tr>
<td>HMMER</td>
<td>R</td>
<td>hmmer</td>
<td>intel</td>
<td>openmpi</td>
</tr>
<tr>
<td>MrBayes</td>
<td>R</td>
<td>mrbayes</td>
<td>intel</td>
<td>mvapich</td>
</tr>
<tr>
<td>SHRiMP</td>
<td>R/L</td>
<td>shrimp</td>
<td>gcc</td>
<td>n/a</td>
</tr>
<tr>
<td>abyss</td>
<td>R/L</td>
<td>abyss</td>
<td>gcc (R), intel (L)</td>
<td>openmpi</td>
</tr>
<tr>
<td>Velvet</td>
<td>L</td>
<td>velvet</td>
<td>intel</td>
<td>n/a</td>
</tr>
<tr>
<td>Trinity</td>
<td>L</td>
<td>trinityrnaseq</td>
<td>gcc</td>
<td>n/a</td>
</tr>
</tbody>
</table>
## Computational Biology Modules

<table>
<thead>
<tr>
<th>Application</th>
<th>Ranger/Lonestar</th>
<th>Module</th>
<th>Compiler</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAMD</td>
<td>R/L</td>
<td>namd</td>
<td>intel</td>
<td>mvapich</td>
</tr>
<tr>
<td>GROMACS</td>
<td>R/L</td>
<td>gromacs</td>
<td>intel, pgi</td>
<td>mvapich, mvapich2</td>
</tr>
<tr>
<td>Desmond</td>
<td>R</td>
<td>desmond</td>
<td>intel</td>
<td>mvapich</td>
</tr>
<tr>
<td>Amber</td>
<td>R</td>
<td>amber</td>
<td>intel, pgi</td>
<td>mvapich, mvapich2</td>
</tr>
<tr>
<td>AutoDock 4</td>
<td>R</td>
<td>autodock</td>
<td>intel, pgi</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Additional Software
Additional Software

• TACC cannot support all bio applications

• You may have custom/modified software that you want to run

• Two primary options:
  1. Submit ticket requesting TACC install your desired software package
  2. Build the software package in your HOME or WORK directory
Compilers

• TACC provides compilers supporting all major languages
  – Fortran77, Fortran90, C, C++, Perl, Python

• TACC supports three major compiler families:
  – Portland Group (%module load pgi)
    • Ranger only!
    • C (pgcc), C++ (pgCC), Fortran 77 (pgf77), Fortran 90 (pgf90)
  – Intel (%module load intel)
    • C (icc), C++ (icpc), Fortran (ifort)
  – GCC (%module load gcc)
    • C (gcc), C++ (g++), Fortran
MPI Compilation

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Language</th>
<th>Type Suffix</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpicc</td>
<td>c</td>
<td>.c</td>
<td>mpicc prog.c</td>
</tr>
<tr>
<td>mpiCC</td>
<td>C++</td>
<td>.C, .cc, .cpp, .cxx</td>
<td>mpiCC prog.cc</td>
</tr>
<tr>
<td>mpif77</td>
<td>F77</td>
<td>.f, .for, .ftn</td>
<td>mpif77 -Vaxlib prog.f</td>
</tr>
<tr>
<td>mpif90</td>
<td>F90</td>
<td>.f90, .fpp</td>
<td>mpif90 -Vaxlib prog.f90</td>
</tr>
</tbody>
</table>

- The mpiXXX commands are shell scripts
- They call the underlying C/C++/Fortran compiler
- This depends on the currently-loaded compiler module
Building Apps from Source

• **Considerations:**
  – When possible, build your applications from source rather than running pre-compiled binaries

  – If you choose to use “%make Install”, you will need to modify the “configure” script
    • `./configure --prefix=/my/favorite/directory`

  – For best performance, use the Intel compilers

  – For best compatibility, use the gcc compilers
Job Submission
Batch Systems

- Batch schedulers track, prioritize and submit jobs into a shared system queue

- Ranger and Lonestar use GridEngine (GE) for scheduling jobs

- How quickly your job begins executing depends on a number of factors:
  - Submission Time
  - Queue Priority
  - Backfill Opportunities
  - Fairshare Priority
  - Advanced Reservations
  - Number of Actively Scheduled Jobs per User
Batch Submission Process

- **Queue**: Job script waits for resources.
- **Master**: Compute node that executes the job script, launches all MPI processes.

**Diagram**:
- Login Node
- Compute Nodes (C1, C2, C3)
- Master Node
- `ssh`
- `qsub job`
- `mpirun -np # ./a.out`
- `ibrn ./a.out`
## Basic Job Submission Script

```csh
#!/bin/csh

#$ -A A-ccsc           }----------> Your Project Name
#$ -pe 16way 32         }----------> Parallel environment and # of cores
#$ -l h_rt=00:15:00     }----------> Max Run Time (hh:mm:ss)
#$ -q development       }----------> Submission queue
#$ -N hello             }----------> Job name
#$ -o $JOB_ID.out       }----------> Stdout Output file name
#$ -e $JOB_ID.err       }----------> Stderr Output file name

ibrun ./hello           }----------> Execution Statement
```

**mpirun wrapper script**

**executable**
Job Sizing on Ranger/Lonestar

• Number of cores and nodes to use is set with:
  #$ -pe Nway 16*M

• \( N \) represents the number of cores to utilize *per node*
  – Ranger: \( 1 \leq N \leq 16 \)
  – Lonestar: \( 1 \leq N \leq 12 \)

• \( M \) is the number of *nodes* to utilize

• The TOTAL number of *cores* used is thus: \( N*M \)
Example - Job Sizing on Ranger

• **Example 1:**
  
  ```bash
  #$ -pe 16way 64
  ```

  Requests N=16 “way” and M=4 (4*16=64) nodes. The total number of processes is N*M=64.

• **Example 2:**
  
  ```bash
  #$ -pe 8way 64
  ```

  Requests N=8 “way” and M=4 (4*16=64) nodes. The total number of processes is N*M=32.
Memory Limits

- If you need more memory per process, you can request fewer cores per node by using one of the ‘Nway’ environments below.

- **NOTE:** Even if you only launch 1 task/node, you will still be charged for all 16 (Ranger) or 12 (Lonestar)!

<table>
<thead>
<tr>
<th>N</th>
<th>Ranger RAM/process</th>
</tr>
</thead>
<tbody>
<tr>
<td>16way</td>
<td>2GB</td>
</tr>
<tr>
<td>8way</td>
<td>4GB</td>
</tr>
<tr>
<td>4way</td>
<td>8GB</td>
</tr>
<tr>
<td>2way</td>
<td>16 GB</td>
</tr>
<tr>
<td>1way</td>
<td>32 GB</td>
</tr>
</tbody>
</table>
Ranger Queue Definitions

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Runtime</th>
<th>Max Cores</th>
<th>SU Rate</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal</td>
<td>24 hours</td>
<td>4,096</td>
<td>1.0</td>
<td>Normal usage</td>
</tr>
<tr>
<td>development</td>
<td>2 hours</td>
<td>256</td>
<td>1.0</td>
<td>Debugging, testing</td>
</tr>
<tr>
<td>serial</td>
<td>2 hours</td>
<td>16</td>
<td>1.0</td>
<td>Uniprocessor jobs</td>
</tr>
<tr>
<td>long</td>
<td>48 hours</td>
<td>1024</td>
<td>1.0</td>
<td>Long run jobs</td>
</tr>
<tr>
<td>large</td>
<td>24 hours</td>
<td>12,288</td>
<td>1.0</td>
<td>Large job submission (by permission)</td>
</tr>
</tbody>
</table>
# Lonestar Queue Definitions

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Runtime</th>
<th>Max Cores</th>
<th>SU Rate</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal</td>
<td>24 hours</td>
<td>2052</td>
<td>1.0</td>
<td>Normal usage</td>
</tr>
<tr>
<td>development</td>
<td>1 hour</td>
<td>264</td>
<td>1.0</td>
<td>Debugging, testing</td>
</tr>
<tr>
<td>serial</td>
<td>12 hours</td>
<td>12</td>
<td>1.0</td>
<td>Uniprocessor jobs, large memory</td>
</tr>
<tr>
<td>largemem</td>
<td>24 hours</td>
<td>120</td>
<td>4.0</td>
<td>Very large memory jobs</td>
</tr>
</tbody>
</table>
# Job Submission and Monitoring

## Useful SGE commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qsub job</td>
<td>Submits ‘job’ into queue</td>
</tr>
<tr>
<td>showq</td>
<td>Shows all jobs in the queue and their status</td>
</tr>
<tr>
<td>qstat</td>
<td>Shows all user’s active jobs and their status</td>
</tr>
<tr>
<td>qdel</td>
<td>Removes pending or running job</td>
</tr>
<tr>
<td>qhold</td>
<td>Stops user’s current running job</td>
</tr>
</tbody>
</table>
Launcher

- Parametric Job Launcher: a simple utility for submitting multiple serial applications simultaneously.

  % module load launcher

- 2 key components:
  1. paramlist: execution command
  2. launcher.sge: job submission script
Sample Launcher Jobscript

#!/bin/csh

#$ -N Parametric
#$ -pe 12way 12
#$ -q development
#$ -e $JOB_NAME.$JOB_ID.err
#$ -o $JOB_NAME.$JOB_ID.out
#$ -l h_rt=00:05:00
#$ -V
#$ -cwd
#$ -A ProjectName
#------------------------------------------------------

setenv EXECUTABLE $TACC_LAUNCHER_DIR/launcher
setenv CONTROL_FILE my_paramlist
setenv WORKDIR .
Sample Launcher Paramlist

./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
./hello_world.pl
Q&A