

Linux User Environment

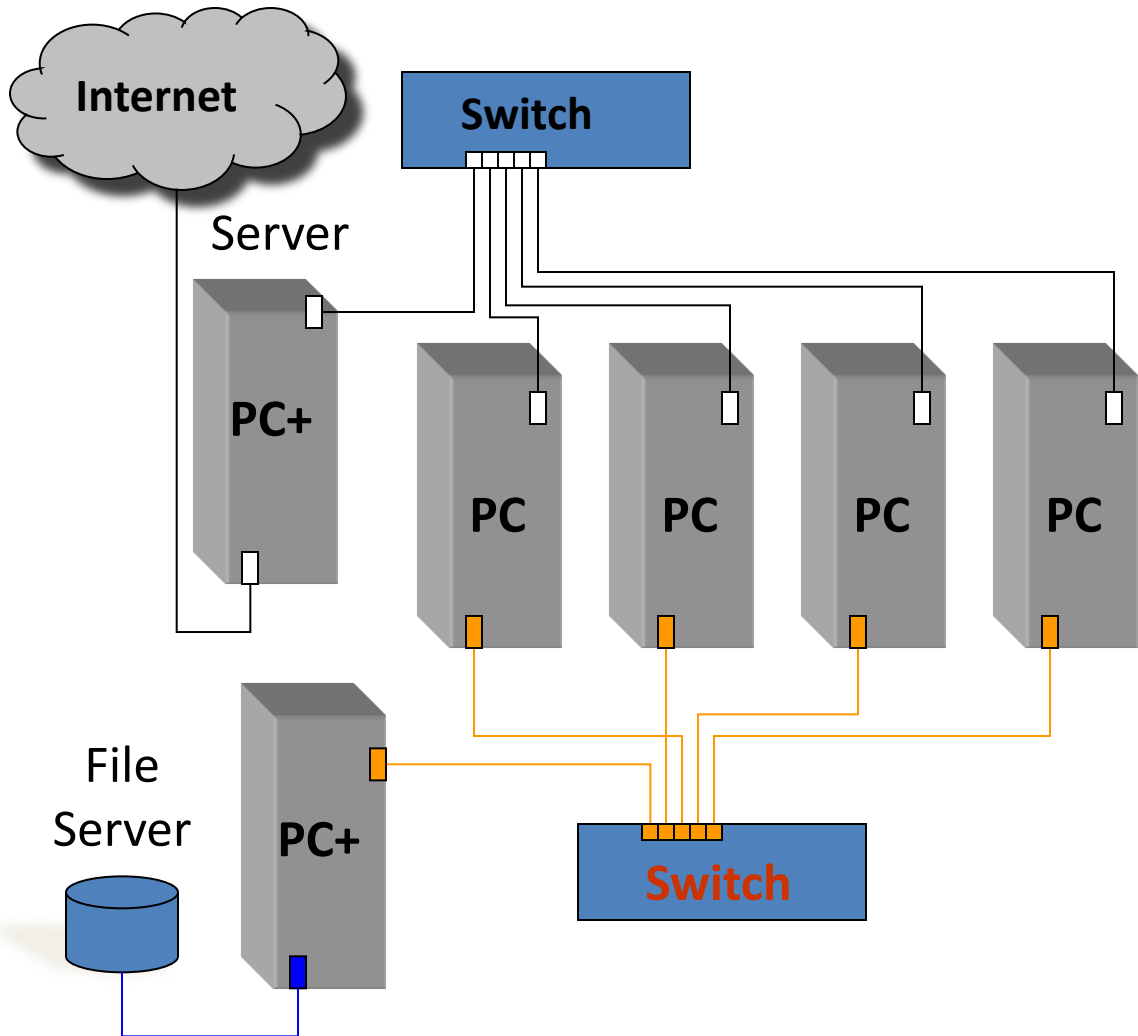
Robert McLay

Outline

- Cluster Architecture and File Systems
- Initial Login and Software Modules
- Job Submission
- Additional Software

CLUSTER ARCHITECTURE AND FILE SYSTEMS

Generic Cluster Architecture

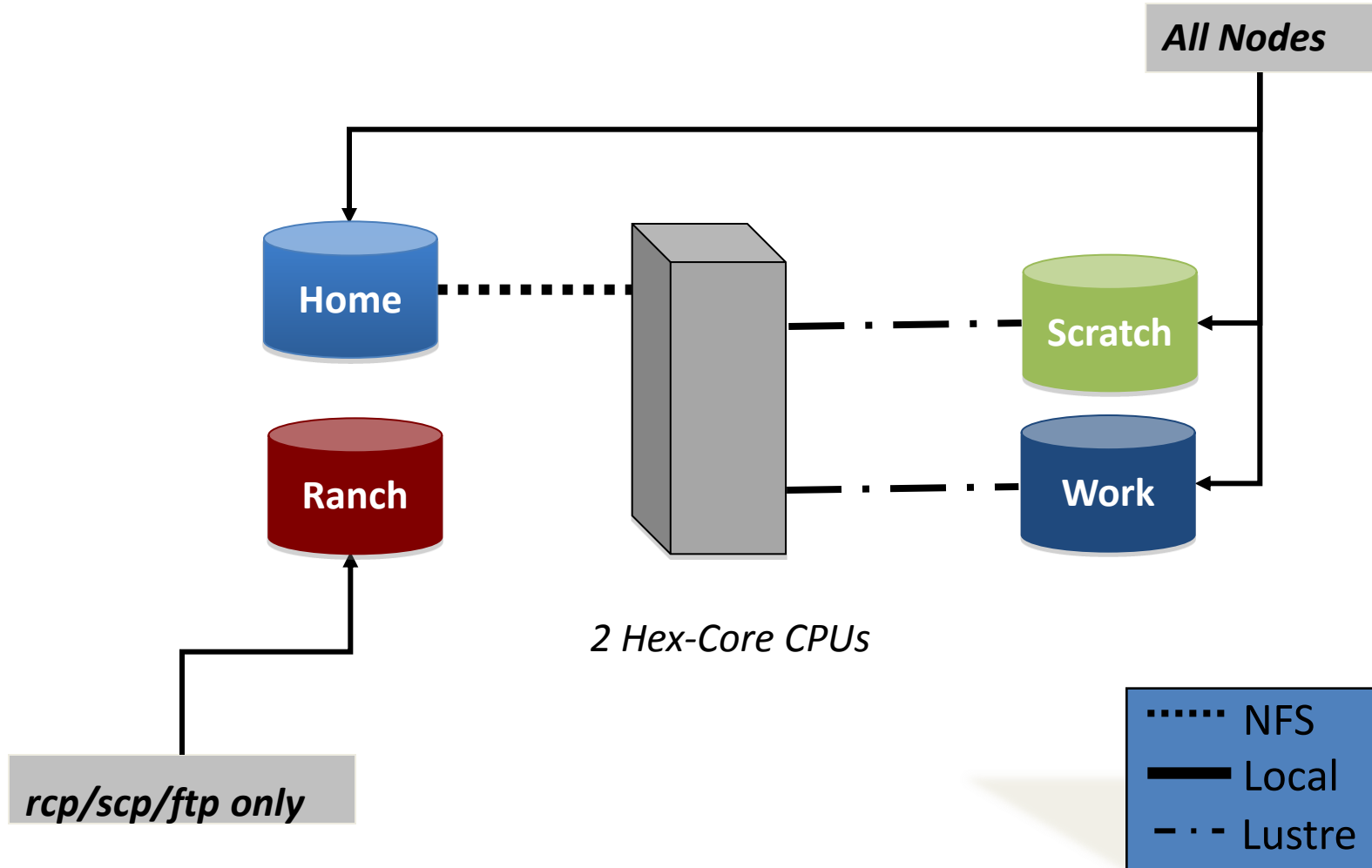


- Ethernet
- Myrinet, IB, Quadrics
- FCAL, SCSI,...

Lonestar



Available File Systems (Lonestar)

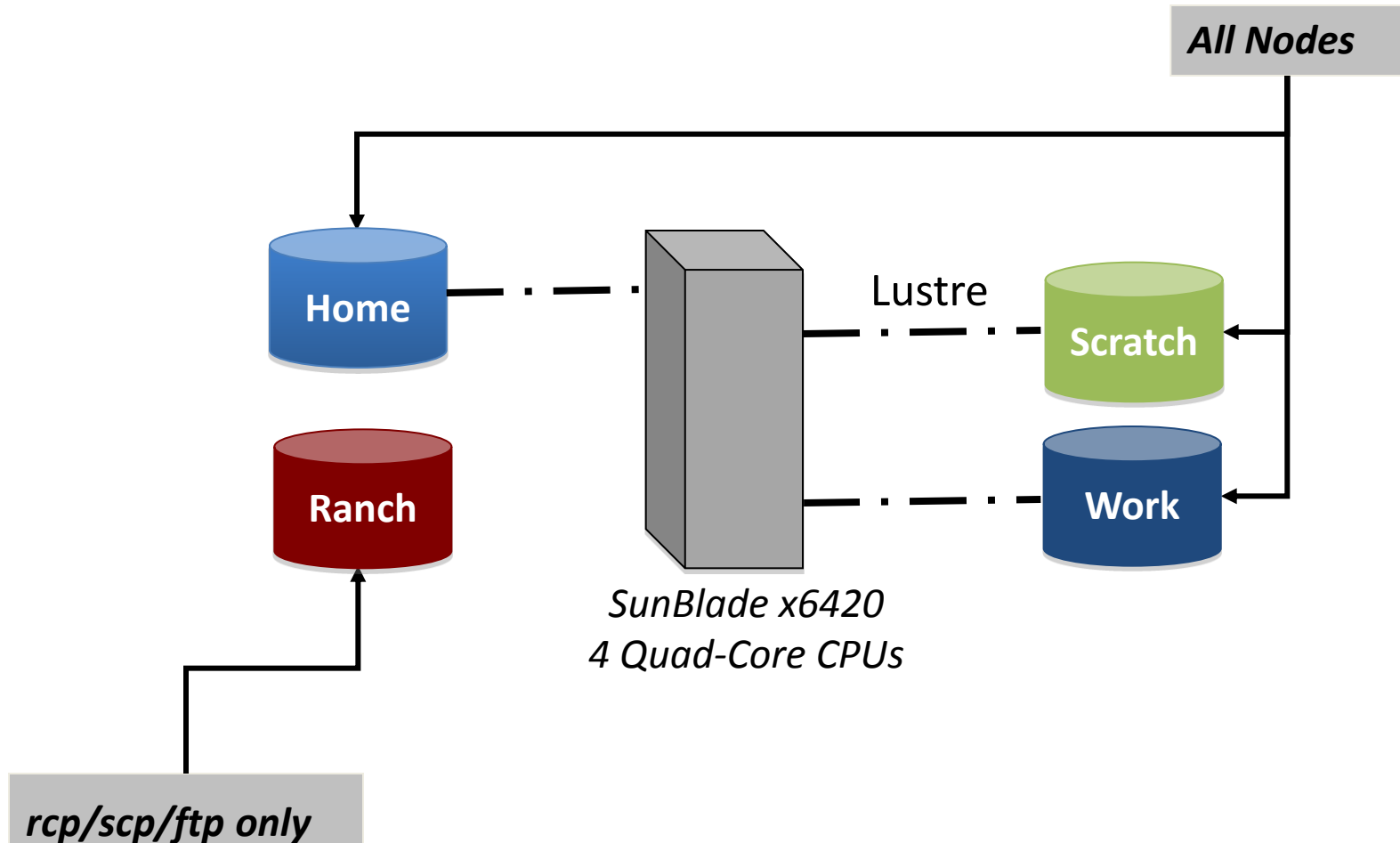


File System Access & Lifetime Table (Lonestar)

Environmental Variable	User Access Limits	Lifetime
\$HOME	1.0 GB	Project
\$WORK	421 TB (total) 250 GB quota	Not Purged/ Not Backed- Up
/tmp (local)	~62 GB	Job Duration
\$SCRATCH	842 TB (Total)	10 days
\$ARCHIVER:\$ARCHIVE	“Unlimited”	Project+

Use the aliases **cd**, **cdw** and **cds** to change directory to \$HOME, \$WORK and \$SCRATCH respectively.

Available File Systems (Ranger)



File System Access & Lifetime Table (Ranger)

Environmental Variable	User Access Limits	Lifetime
\$HOME	6 GB quota	Project
\$WORK	350 GB quota	Project
\$SCRATCH	~400 TB	10 Days
\$ARCHIVER:\$ARCHIVE	“Unlimited”	Project+

- Use the aliases **cd**, **cdw** and **cds** to change directory to \$HOME, \$WORK and \$SCRATCH respectively.

- To check usage:

```
%login3 lfs quota -u <username> <directory>
```

Moving Files From Site to Site

- Collect files together:
 - *tar*
- Compress files:
 - *gzip* → transfer → *gunzip* (60-70%)
- Transfer files:
 - *ftp* (insecure; naked ftp connections not allowed)
 - *scp/sftp* (10-15x slower than ftp)
 - *bbcp* (faster, supports striped writes; must install at your site)

Using *bbcp*

- Download the *bbcp* source (for your local machine):
<http://www.slac.stanford.edu/~abh/bbcp/bbcp.tar.Z>
- Builds easily under linux (tested) and a few other UNIXes, Darwin requires a few tweaks to the *bbcp* Makefile.
- To copy a file to ranger, use login3 or login4 directly
`% bbcp filename login3.ranger.tacc.utexas.edu:`
- More info at Ranger user guide:
<http://www.tacc.utexas.edu/services/userguides/ranger/#highspeedtransfers>

INITIAL LOGIN AND SOFTWARE MODULES

Initial Login

- Login with SSH

```
% ssh ranger.tacc.utexas.edu
```

- Connects you to login3.ranger.tacc.utexas.edu or login4.ranger.tacc.utexas.edu
- **Do not** overwrite ~/.ssh/authorized_keys
 - Feel free to add to it if you know what it's for
 - SSH used for job start-up on the compute nodes, mistakes ~/.ssh can prevent jobs from running

Startup Scripts & Modules

- Login shell is set with “**chsh -s <login shell>**”
 - Takes some time to propagate (~1 hour)
- “**chsh -l**” will list available login shells.
- Each shell reads a set of configuration scripts.
- Bourne-type shells (Bourne, Korn, and Bash Shells)

System-wide config scripts:

Bash: /etc/tacc/profile
 /etc/tacc/bashrc
 /etc/profile.d/<xxx>.sh
Tcsh: /etc/tacc/csh.cshrc
 /etc/tacc/csh.login
 /etc/profile.d/<xxx>.csh

User-customizable config script:

Bash: ~/.bashrc, ~/.profile
Tcsh: ~/.cshrc, ~/.login

User Startup Files: Bash

~/profile:

If [-f ~/.bashrc]; then

 . ~/.bashrc

fi

~/bashrc:

If [-z "\$_READ" -a -z "\$ENVIRONMENT"]; then

 export _READ=1

 # Put any module commands here:

 # module load git

fi

User Startup Scripts: Tcsh

```
~/.cshrc:  
if ( ! $_READ && ! $?ENVIRONMENT ) then  
  setenv _READ "read"  
  #  
  # Place any module commands here:  
  # module load git  
  #  
endif
```

Modules

- Modules are used to set up your PATH and other environment variables

```
% module help                {lists options}
% module list                 {lists loaded modules}
% module avail               {lists available modules}
% module load <module>        {add a module}
% module unload <module>     {remove a module}
% module swap <mod1> <mod2>  {swap two modules}
% module help <mod1>         {module-specific help}
% module spider             {lists all modules}
% module spider petsc       {list all version of
petsc}
```

Modules

- Available modules depend on:
 - The compiler (eg. PGI, Intel) and
 - The MPI stack selected
- On ranger, default is PGI/Mvapich1
- On lonestar, default is Intel/Mvapich

- To unload all grid-related modules:

```
login3% module unload CTSSV4
```

- To return to default modules:

```
login3% module purge; module load TACC
```

Modules

- Modules available before selecting compiler:

```
login3(1)$ module purge; module avail
```

```
----- /opt/apps/modulefiles -----  
beta  
binutils-amd/070220  
ddt/2.3.1  
gcc/4.2.0  
gcc/4.3.2  
gcc/4.4.0  
gcc/4.4.1 (default)  
git/1.6.3.1  
gmake/3.81  
gmp/4.2.4  
gotoblas/1.23 (default)  
gsl/1.11  
gzip/1.3.12  
intel/10.1 (default)  
intel/9.1  
irods/2.1  
launcher/1.3  
lua/5.1.4  
mkl/10.0 (default)  
mpfr/2.3.2  
mysql/5.1.32  
ncl_ncarg/5.1.1  
numpy/1.2.1  
papi/3.5.0  
papi/3.6.0 (default)  
pgi/7.1  
pgi/7.2-5 (default)  
pgi/8.0-6  
postgres/8.3.6  
python/2.5.2  
star-ccm/4.04.011  
subversion/1.5.1 (default)  
subversion/1.6.1  
sun/12  
tar/1.22  
vis/1.0
```

Modules

- Additional modules become available after choosing a compiler...

```
login3(2)$ module load intel; module avail
```

```
----- /opt/apps/intel10_1/modulefiles -----  
acml/4.1.0                hecura-debug/1.4r3392    mvapich/1.0  
autodock/4.0.1           hecura/0.1               mvapich/1.0.1 (default)  
boost/1.37.0             hecura/1.4rc2           mvapich2-new/1.2  
boost/1.39.0 (default)   hecura/trunk_2009_09_20 (default) mvapich2/1.2  
fftw3/3.1.2              hmmer/2.3.2             nco/3.9.5  
gotoblas/1.26 (default)  metis/4.0               netcdf/3.6.2  
gotoblas/1.30            mvapich-devel/1.0       openmpi/1.2.4  
gotoblas2/1.00          mvapich-old/1.0.1      openmpi/1.2.6  
hdf5/1.6.5              mvapich-ud/1.0         openmpi/1.3 (default)
```

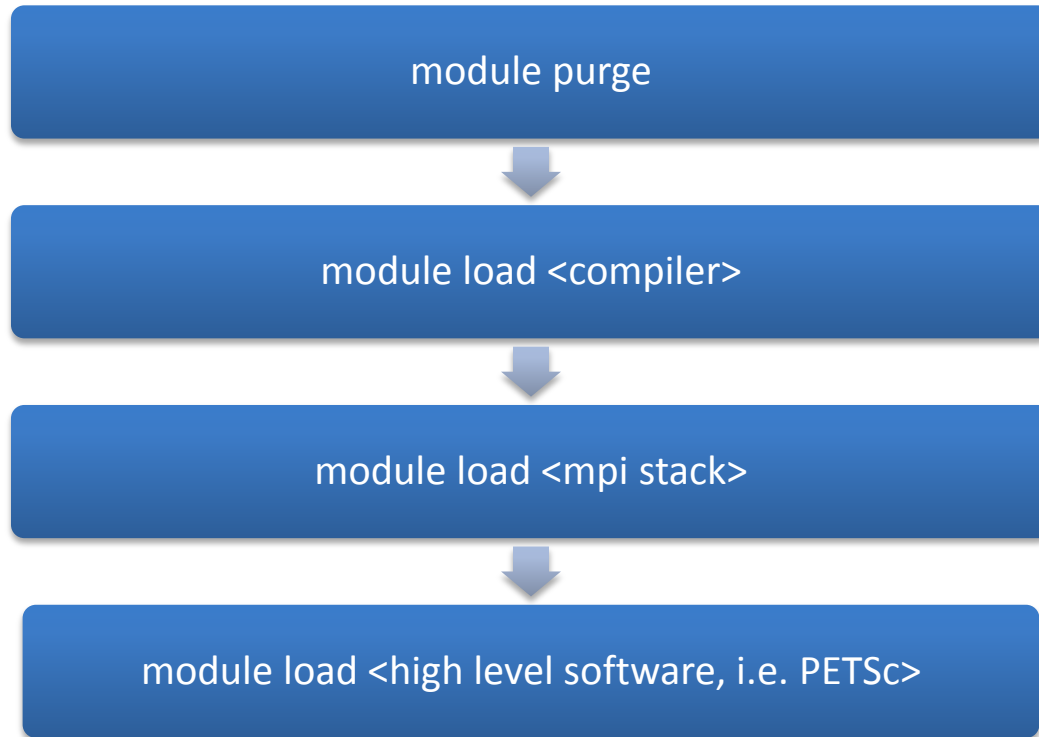
Modules

- And even more after selecting an MPI stack.

```
login3(3)$ module load mvapich; module avail
```

```
----- /opt/apps/intel10_1/mvapich1_1_0_1/modulefiles -----  
amber/9.0  
arpack/2.1  
charm++/6.0  
desmond/2.0.4  
espresso/4.1-scalapack  
fftw2-test/2.1.5  
fftw2/2.1.5  
gamess/03_2007  
gromacs/4.0.5 (default)  
hypre/2.0.0-SmallScale (default)  
ipm/0.922  
kojak/2.2 (default)  
lammps/25Aug09 (default)  
mpiP/3.1.1  
mpiblast/1.5.0  
namd/2.6  
nwchem/5.0  
nwchem/5.1.1 (default)  
pdtoolkit/3.12 (default)  
petsc/2.3.3 (default)  
petsc/2.3.3-complex  
petsc/2.3.3-complexdebug  
petsc/2.3.3-complexdebugdynamic  
petsc/2.3.3-complexdynamic  
petsc/2.3.3-debug  
petsc/2.3.3-dynamic  
petsc/3.0.0-complexdebug  
petsc/3.0.0-complexdebugdynamic  
petsc/3.0.0-complexdynamic  
petsc/3.0.0-cxx  
petsc/3.0.0-cxxdebug  
petsc/3.0.0-debug  
petsc/3.0.0-debugdynamic  
petsc/3.0.0-dynamic  
petsc/3.0.0-uni  
petsc/3.0.0-unidebug  
phdf5/1.8.2  
plapack/3.2 (default)  
pmetis/3.1  
scalapack/1.8.0  
slepc/2.3.3 (default)  
slepc/2.3.3-complex  
slepc/2.3.3-dynamic  
sprng/2.0  
tao/1.9 (default)  
tao/1.9-debug  
tau/2.17 (default)  
trilinos/9.0.0
```

Modules



- The default modules are suitable for most users.

Module Hierarchy

- On Ranger we have Three Compilers: PGI, Intel, GCC
- On Ranger we have Three MPI implementations: Mvapich, Mvapich2, Openmpi
- Packages like Petsc or FFTW2 could have 9 versions (not all combinations exists)

Module Hierarchy (2)

- \$ module avail # tells what packages are avail.
with current compiler/mpi
pairings.
- \$ module spider # tells what packages are avail.
across all compiler/mpi
pairings.

Module Hierarchy (3)

\$ module purge; module load TACC

\$ module swap pgi intel

Due to MODULEPATH changes the follow
modules have been reloaded:

- 1) mvapich

Module Feature: Family

- We support two Families: Compilers and MPI implementations. You can only have one member of the family.
- Only one compiler, One MPI Stack.
- Env. Var: TACC_FAMILY_COMPILER: intel, pgi, gcc
- Env. Var: TACC_FAMILY_MPI: mvapich, mvapich2, openmpi
- Can be used in Makefiles, and Scripts.

Module Feature: User default Modules

- Users can create their own default list of modules
- Example:
- `$ module purge; module load TACC;`
- `$ module rm CTSSV4; module swap pgi intel`
- `$ module setdefault`
- Users now will load intel and no Teragrid modules every time they login.

EDITING FILES

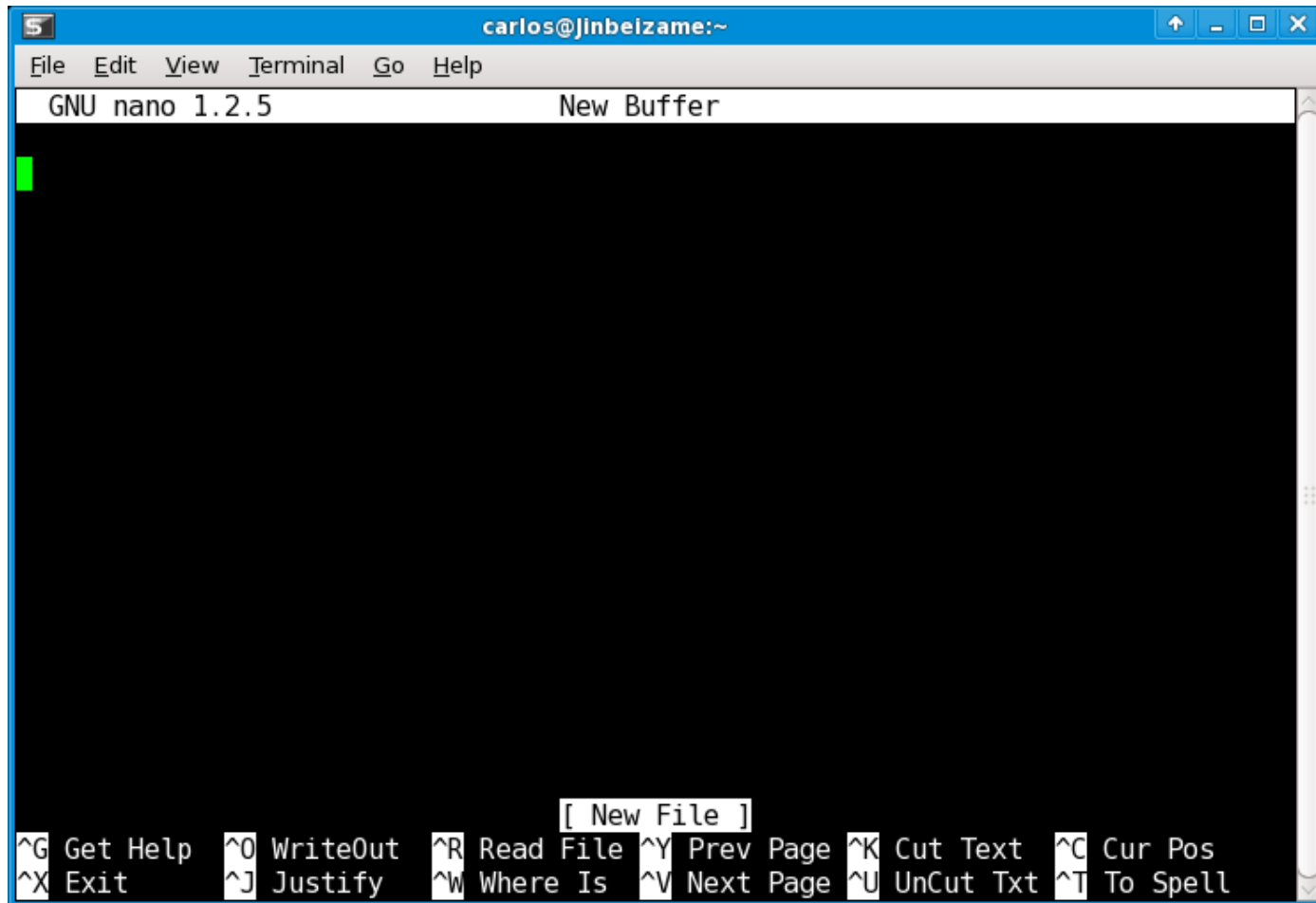
TEXT EDITORS

- Most Linux systems have two or three commonly installed text editors:
 - nano
 - vi
 - emacs
- Editors are a matter of preference, there is no right or wrong editor.

nano

- All operations commands are preceded by the Control key:
 - ^G Get Help
 - ^O WriteOut
 - ^X Exit
 -
- If you have modified the file and try to exit (^X) without writing those changes (^O) you will be warned.
- Makes text editing simple, but it has less powerful options than vi (search with regular expressions, etc..)

nano default screen

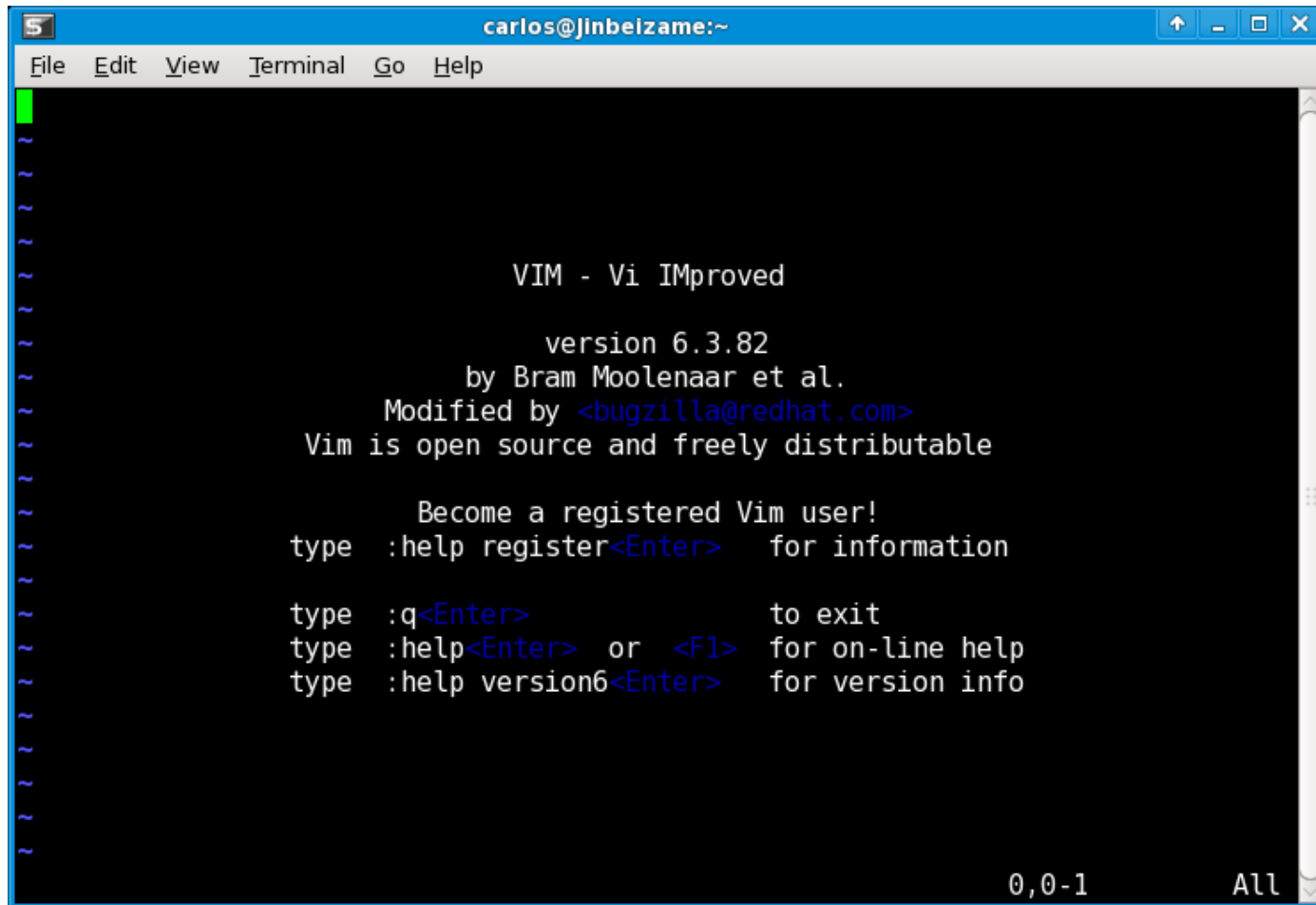


```
carlos@jinbelzame:~
File Edit View Terminal Go Help
GNU nano 1.2.5 New Buffer
[ New File ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Txt ^T To Spell
```

vi

- Powerful search options but more complicated to use
- Two modes:
 - Insertion mode - allows you to write
 - Command mode - allows you to move with the arrow keys, search, etc...
- Start insertion mode with several commands:
 - a: append after cursor
 - i: insert after cursor
 - R: replace several characters
- Get back to command mode with the Esc key
- See the vi cheat sheet for details

vi default screen



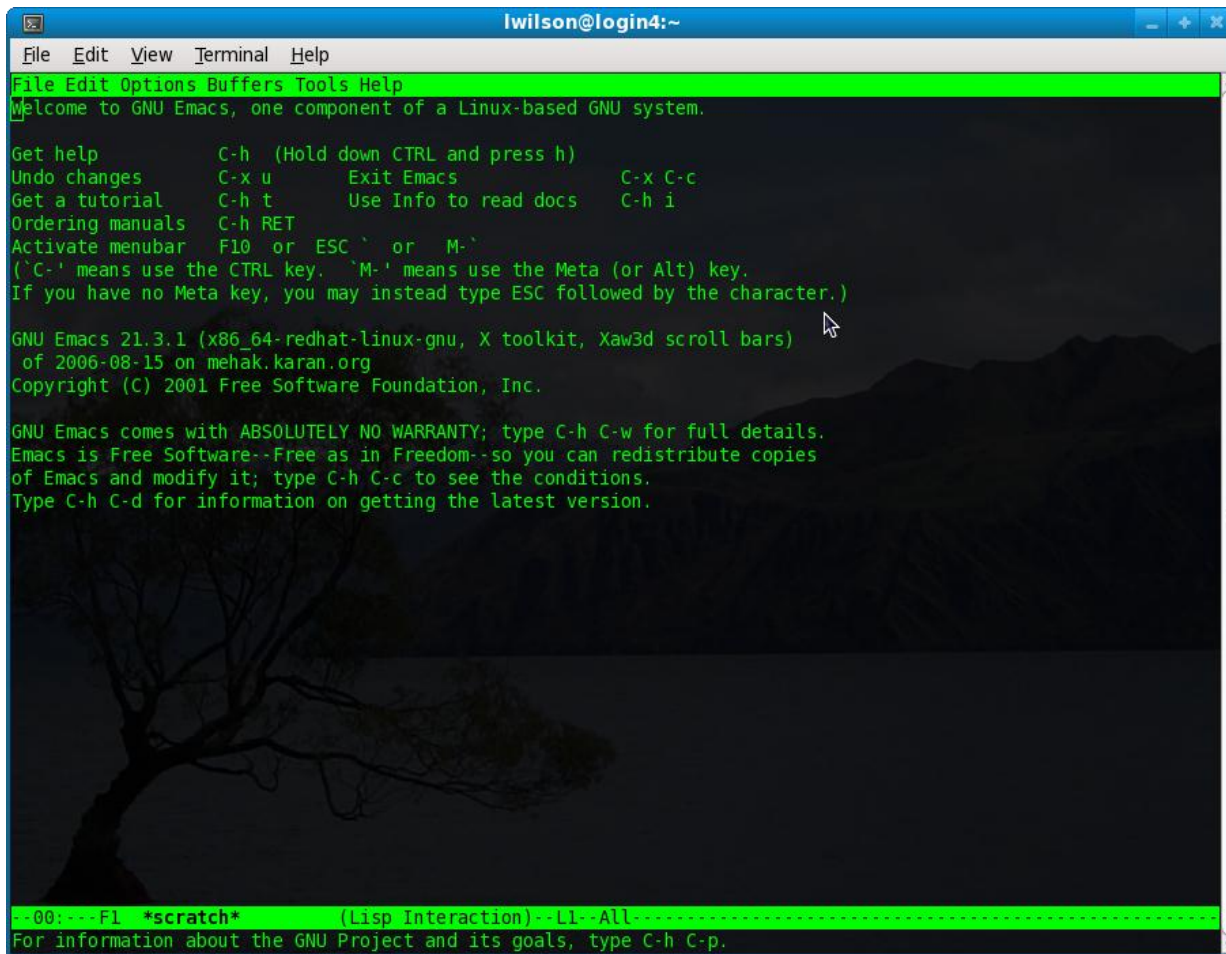
The image shows a terminal window titled "carlos@JInbelzame:~" with a menu bar containing "File", "Edit", "View", "Terminal", "Go", and "Help". The terminal content displays the Vim startup screen, which includes the text "VIM - Vi IMproved", "version 6.3.82", "by Bram Moolenaar et al.", "Modified by <bugzilla@redhat.com>", and "Vim is open source and freely distributable". It also provides instructions on how to become a registered user, exit, and access help. The status bar at the bottom right shows "0,0-1" and "All".

```
carlos@JInbelzame:~  
File Edit View Terminal Go Help  
VIM - Vi IMproved  
version 6.3.82  
by Bram Moolenaar et al.  
Modified by <bugzilla@redhat.com>  
Vim is open source and freely distributable  
  
Become a registered Vim user!  
type :help register<Enter> for information  
  
type :q<Enter> to exit  
type :help<Enter> or <F1> for on-line help  
type :help version6<Enter> for version info  
  
0,0-1 All
```

emacs

- emacs is actually a lisp interpreter with extensions to use it as a text editor
- Can perform the same operations as in vi
- Uses series of multiple keystroke combinations to execute commands
- “Hard to learn, easy to use”

emacs default screen



The screenshot shows a terminal window titled "lwilson@login4:~". The Emacs interface is displayed with a menu bar at the top: "File Edit Options Buffers Tools Help". The main area contains the following text:

```
Welcome to GNU Emacs, one component of a Linux-based GNU system.

Get help          C-h (Hold down CTRL and press h)
Undo changes     C-x u      Exit Emacs          C-x C-c
Get a tutorial   C-h t      Use Info to read docs  C-h i
Ordering manuals C-h RET
Activate menubar F10 or ESC ` or M-`
(`C-' means use the CTRL key. `M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

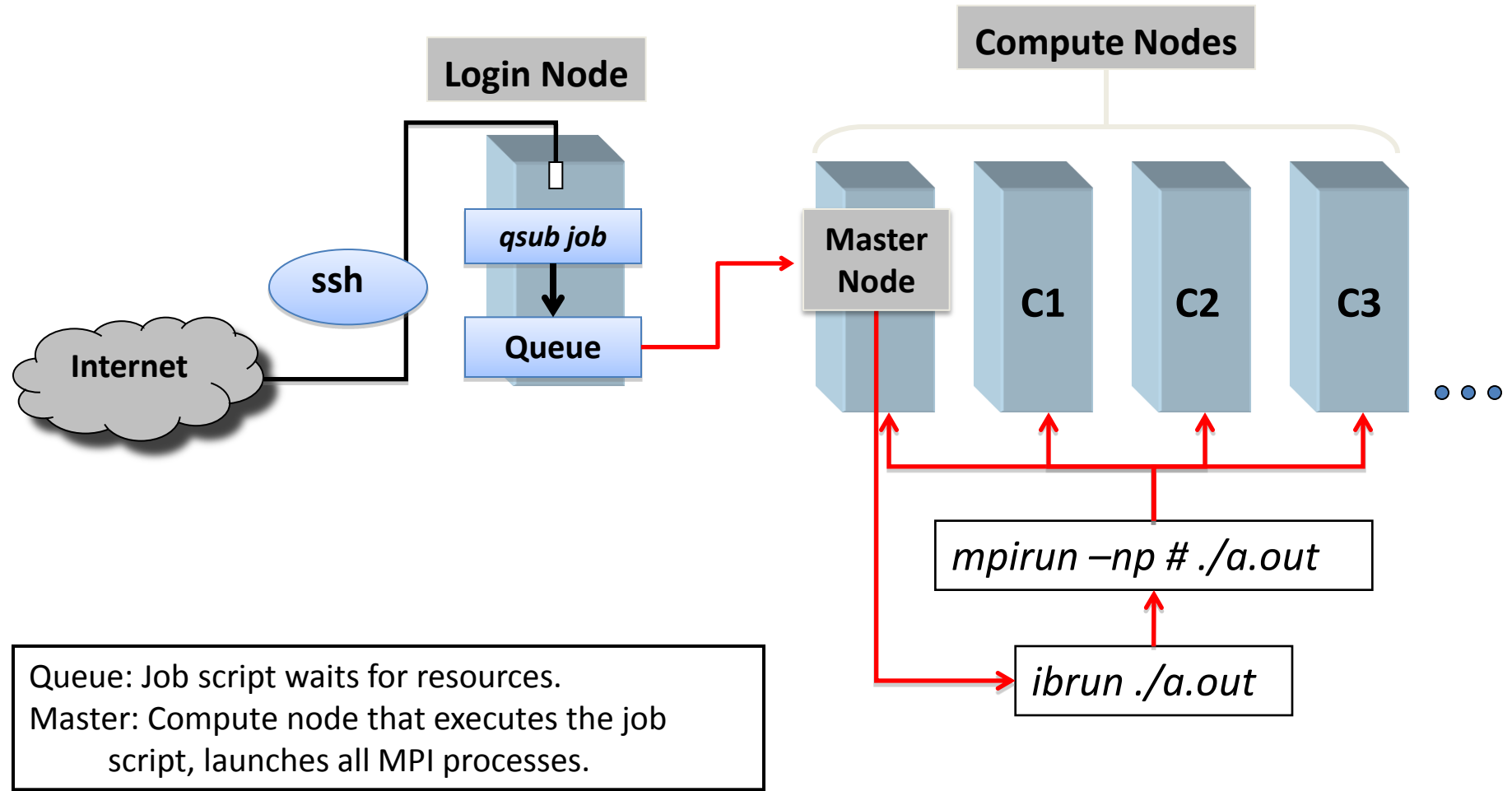
GNU Emacs 21.3.1 (x86_64-redhat-linux-gnu, X toolkit, Xaw3d scroll bars)
  of 2006-08-15 on mehak.karan.org
Copyright (C) 2001 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.
```

At the bottom, a status line shows "--00:-- F1 *scratch* (Lisp Interaction) --LL-- All-----". Below this, a prompt indicates: "For information about the GNU Project and its goals, type C-h C-p."

JOB SUBMISSION

Batch Submission Process



Batch Systems

- *Lonestar, Ranger and Longhorn* uses Sun GridEngine (SGE).
- Order of job execution depends on a variety of parameters:
 - Submission Time
 - Queue Priority
 - Backfill Opportunities
 - Fairshare Priority
 - Advanced Reservations
 - Number of Actively Scheduled Jobs per User

Lonestar Queue Definitions

Queue	Max Runtime	Max Cores	SU Charge Rate	Purpose
normal	24 hours	2052	1.0	Normal usage
development	1 hour	264	1.0	Debugging, allows interactive jobs
serial	12 hours	1	1.0	Uniprocessor jobs

Ranger Queue Definitions

Queue	Max Runtime	Max Nodes (Cores)	SU Charge Rate	Purpose
normal	48 hours	256 (4,096)	1.0	Normal usage
large	48 hours	768 (12,288)	1.0	Large job submission (by permission)
development	2 hours	16 (256)	1.0	Debugging and development
serial	2 hours	1 (16)	1.0	Uniprocessor jobs

SGE: Basic MPI Job Script

```
#!/bin/bash

#$ -pe 16way 32      Wayness and total core number

#$ -N hello         Job name

#$ -o $JOB_ID.out   stdout file name (%J = jobID)

#$ -e $JOB_ID.err   stderr file name

#$ -q normal        Submission queue

#$ -A A-ccsc        Your Project Name

#$ -l h_rt=00:15:00 Max Run Time (15 minutes)

ibrun ./hello       Execution command
```

Way Ness

No. Processes	Ranger (16 way)	Lonestar (12 way)	Longhorn (8 way)
1	1way 16	1way 12	1way 8
4	4way 16	4way 12	4way 8
8	8way 16	8way 12	8way 8
12	12way 16	12way 12	6way 16
16	16way 16	8way 24	8way 16
32	16way 32	8way 48	8way 32
48	16way 48	12way 48	8way 48
64	16way 64	8way 96	8way 64

Job Sizing with SGE on Ranger

- You must always put a multiple of 16 next to the name of the parallel environment

```
#$ -pe 16way 64 {64 tasks, 4 nodes}
```

```
#$ -pe 8way 64 {32 tasks, 4 nodes}
```

- SGE doesn't automatically handle the case where the number of tasks you want is not a multiple of 16
- If you want a non-multiple of 16, you may set

```
#$ -pe 16way 32
```

```
...
```

```
export MY_NSLOTS=23
```

```
...
```

```
ibrun ./mycode
```

SGE: Memory Limits

- Default parallel job submission allocates all 16 compute cores per node.
- If you need more memory per MPI task, you can request fewer cores per node by using one of the 'Nway' environments below.
- Even if you only launch 1 task/node, you will still be charged for all 16!

Parallel environment	Description
16way	16 tasks/node, 1.9GB/task
8way	8 tasks/node, 3.8GB/task
4way	4 tasks/node, 7.6GB/task
2way	2 tasks/node, 15.2 GB/task
1way	1 task/node, 30.4 GB/task

ADDITIONAL SOFTWARE

(COMPILERS, MATH LIBRARIES, PERFORMANCE EVALUATION)

Compilers

- Portland Group 7.2-5
 - C (pgcc), C++ (pgCC), Fortran 77 (pgf77), Fortran 90 (pgf90)
 - load: pgi
- Intel 11.1 & 10.1 & 9.1
 - C (icc), C++ (icpc), Fortran (ifort)
 - load: intel or intel/9.1

\$ module swap pgi intel

MPI Compilation

Compiler	Language	Type Suffix	Example
mpicc	c	.c	mpicc prog.c
mpicxx	C++	.C, .cc, .cpp, .cxx	mpicxx prog.cc
mpif77	F77	.f, .for, .ftn	mpif77 -Vaxlib prog.f
mpif90	F90	.f90, .fpp	mpif90 -Vaxlib prog.f90

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

Architecture-Dependent Compiler Options (SSE)

Machine	PGI	Intel 9	Intel 10	Intel 11
Lonestar	n/a	n/a	n/a	-xSSE4.2
Ranger	-tp barcelona-64	-xW	-xO	-msse3

- Available Streaming SIMD Extensions (SSE) depend on the underlying architecture
 - Additional hardware registers allow for Single Instruction over Multiple Data
 - Different levels (SSE/SSE2/SSE3/AMD 3DNow!) for different processors
 - 70+ new instructions that facilitate pipelining and partial vectorization

AMD Math Libraries

- ACML (AMD Core Math Library)
 - LAPACK, BLAS, and extended BLAS (sparse), FFTs (single- and double-precision, real and complex data types).
 - APIs for both Fortran and C
 - <http://developer.amd.com/acml.jsp>
 - `% module load acml`

Intel Math Libraries

- MKL (Math Kernel Library)
 - LAPACK, BLAS, and extended BLAS (sparse), FFTs (single- and double-precision, real and complex data types).
 - APIs for both Fortran and C
 - www.intel.com/software/products/mkl/
 - `% module load mkl`
- VML (Vector Math Library) [equivalent to libmfastv]
 - Vectorized transcendental functions.
 - Optimized for Pentium III, 4, Xeon, and Itanium processors.

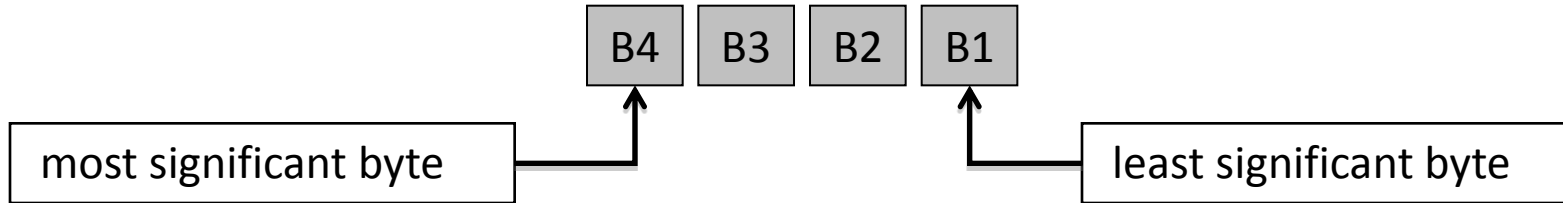
Performance Tools

- PAPI (Performance API)
 - Events, floats, instruction, data access, cache access, TLB misses
 - <http://www.ncsa.uiuc.edu/UserInfo/Resources/Software/Tools/PAPI>
 - `% module load papi`
 - **Will be available soon on lonestar**
- TAU (Tuning and Analysis Utilities)
 - Portable profiling and tracing toolkit for performance analysis of parallel programs
 - www.cs.uoregon.edu/research/paracomp/tau/
 - Fortran 77/90, C, C++, Java
 - OpenMP, pthreads, MPI, mixed mode
 - `% module load pdtoolkit; module load tau`

Little vs. Big Endian

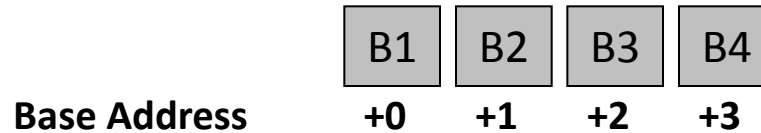
- A byte is the lowest addressable storage unit on many machines.
- A “word” often refers to a group of bytes.
- There are two different ways to store a word on disk and memory: Big Endian and Little Endian.
- Intel Pentium machines (x86 & x86_64) are Little Endian Machines.
- Most “big iron” machines are Big Endian: Crays(Unicos), IBMs(AIX), SGIs(IRIX), & Macs (Motorola processors) are Big Endian machines.
- **Use standard libraries to save binary data**

Little vs. Big Endian Storage



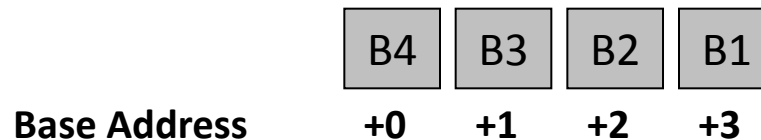
Little Endian = “Little End First”

Bytes are stored from the *least* significant to *most* significant.



Big Endian = “Big End First”

Bytes are stored from the *most* significant to the *least* significant.



Platform Independent Binary Files

- XDR
 - Developed by SUN as part of NFS
 - Using XDR API gives platform independent binary files
 - [/usr/include/rpc/xdr.h](#)
- NETCDF – Unidata Network Common Data Form
 - Common format used in Climate/Weather/Ocean applications
 - <http://my.unidata.ucar.edu/content/software/netcdf/docs.html>
 - `% module load netcdf; man netcdf`
- HDF - Hierarchical Data Format developed by NCSA
 - <http://hdf.ncsa.uiuc.edu/>

Additional Information

<http://www.tacc.utexas.edu>

(Click on the “User Services” drop-down menu and select “User Guides”)