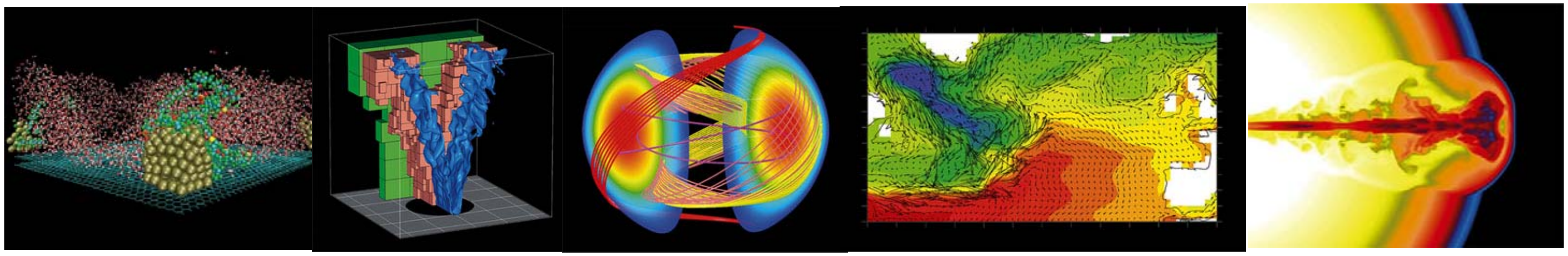


Franklin: NERSC Cray XT4

Helen He
NERSC User Services
yhe@lbl.gov

NERSC User Group Meeting
September 17-20, 2007

Training Topics

- Overview
- XT4 Architecture
- PGI Compiler
- Single Node and MPI Optimizations
- Performance and Profiling Tools
- Running Jobs
- Third-party Softwares
- ACTS Tools
- DDT Debugger
- I/O
- Grid Services
- Benchmark Performance

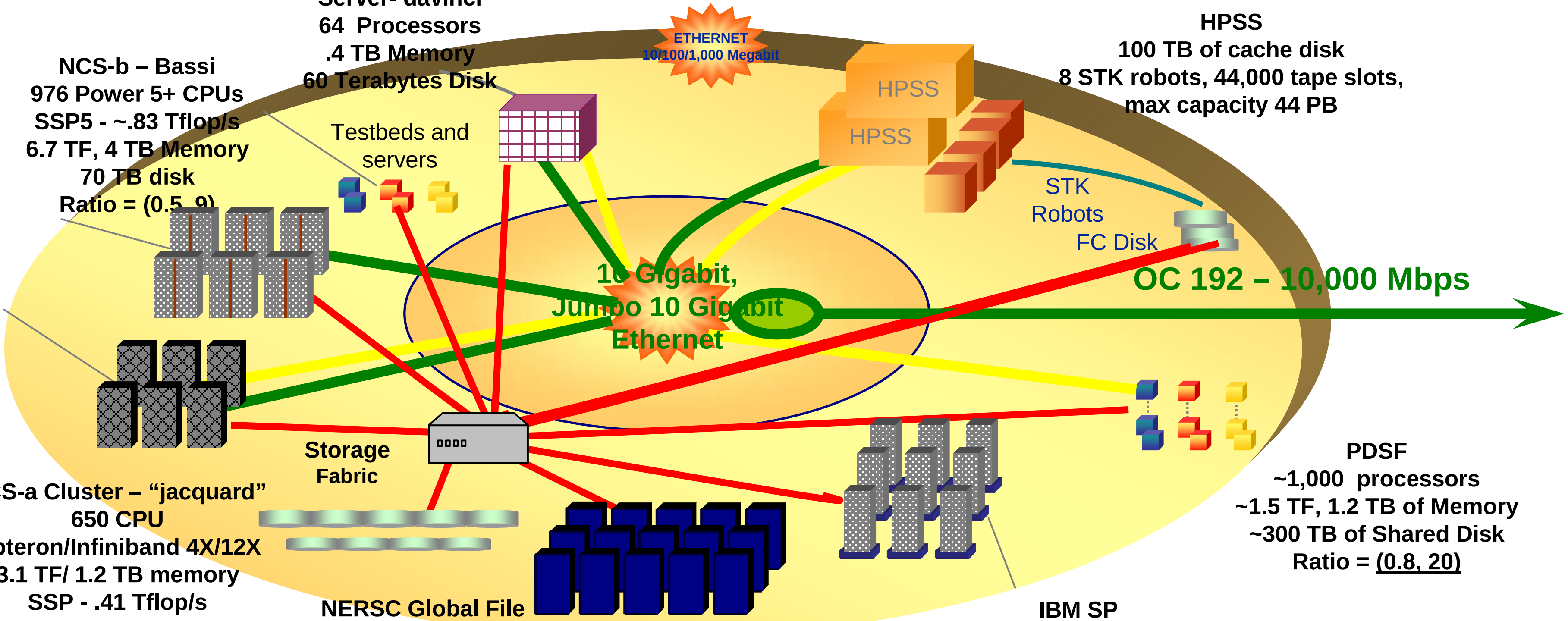
Franklin

Benjamin Franklin,
one of America's first scientists, performed ground breaking work in energy efficiency, electricity, materials, climate, ocean currents, transportation, health, medicine, acoustics and heat transfer.

About Franklin

- Largest Cray XT-4
- 9,740 nodes with 19,480 CPU (cores)
- dual-core AMD Opteron 2.6 GHz, 5.2 GFlops/sec peak
- 102 node cabinets
- 16 KWs per cabinet (~1.7 MWs total)
- 39.5 TBs aggregate memory
- 16.1+ Tflop/s Sustained System Performance (SSP)
  (Seaborg - ~0.98, Bassi - ~0.83)
- 101.5 Tflop/s theoretical system peak performance
- Cray SeaStar2 / 3D Torus interconnect (17x24x24)
- ~366 TBs of usable shared disk

NERSC Systems 2007



Visualization and Post Processing Server- davinci
64 Processors
.4 TB Memory
60 Terabytes Disk

NCS-b – Bassi
976 Power 5+ CPUs
SSP5 - ~.83 Tflop/s
6.7 TF, 4 TB Memory
70 TB disk
Ratio = (0.5, 9)

HPSS
100 TB of cache disk
8 STK robots, 44,000 tape slots,
max capacity 44 PB

NCS-a Cluster – "jacquard"
650 CPU
Opteron/Infiniband 4X/12X
3.1 TF/ 1.2 TB memory
SSP - .41 Tflop/s
30 TB Disk
Ratio = (.4,10)

NERSC Global File System
300 TB shared usable disk

Cray XT4
NERSC-5 - "Franklin"
SSP ~16 Tflop/s

PDSF
~1,000 processors
~1.5 TF, 1.2 TB of Memory
~300 TB of Shared Disk
Ratio = (0.8, 20)

IBM SP
NERSC-3 – "Seaborg"
6,656 Processors (Peak 10 TFlop/s)
SSP5 – .98 Tflop/s
7.8 Terabyte Memory
55 Terabytes of Shared Disk
Ratio = (0.8,4.8)

Ratio = (RAM Bytes per Flop, Disk Bytes per Flop)

Franklin’s Role at NERSC

- Next “flagship” system at NERSC after Seaborg.
- Serves the needs for most NERSC users from capacity jobs to capability jobs.
- Expected to significantly increase computational time for NERSC users.
- DOE expects significant percentage of time to be used for capability jobs on Franklin.

Connecting to Franklin

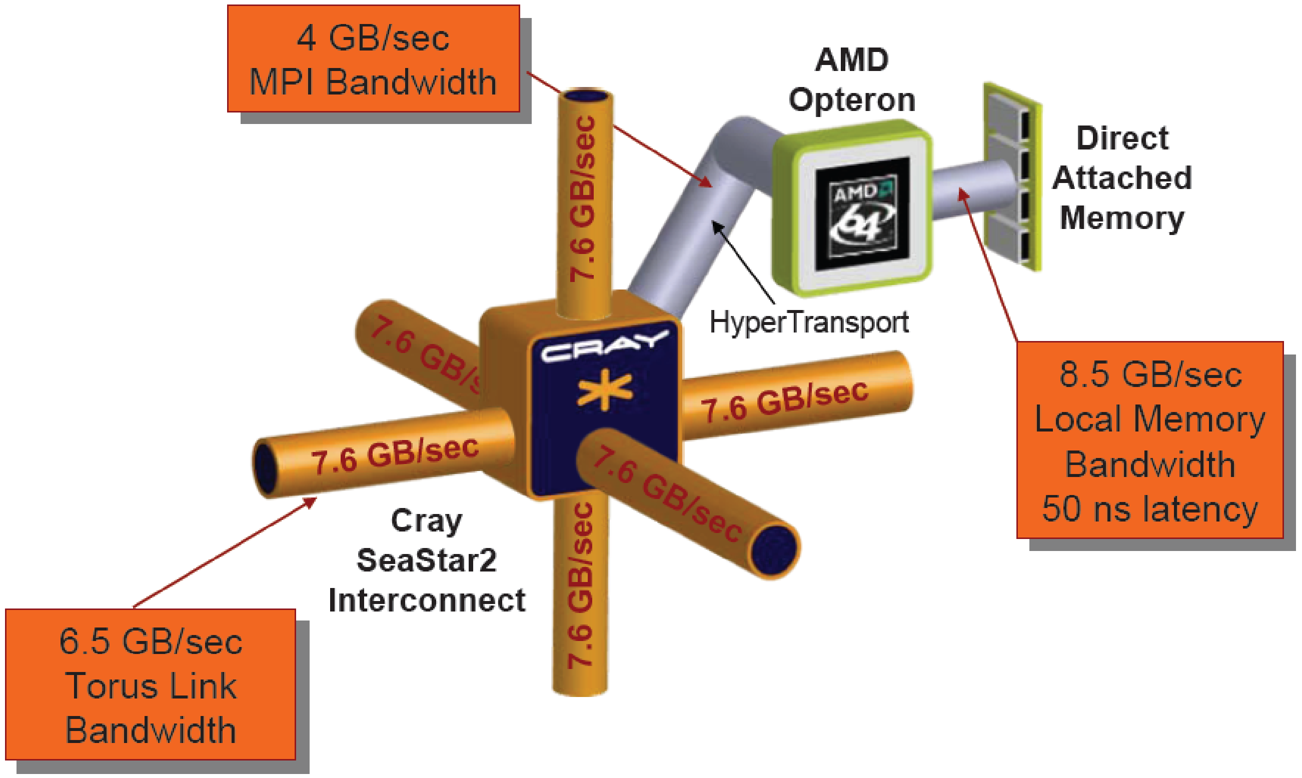
- Interactive shell access is via SSH.
- *% ssh [–l login_name] franklin.nersc.gov*
- Use your NIM password for Franklin password.
- Default shell is csh, use NIM interface to change.
- Dot startup files are read only
  - use *.<name>.ext* to add customizations
  - *fixdots* command will repair files

Quota and File Backup

- Franklin quota limit:
  - $HOME: 15 GB space, 25,000 inodes.
  - $SCRATCH: 500 GB space, 50,000 inodes.
- NERSC policy for $SCRATCH
  - Files older than 7 days may be killed when more space is needed on the system.
- Please backup your files frequently to HPSS
  - Both *hsi* and *htar* work on Franklin.

XT4 Architecture

The Cray XT4 Processing Element: Providing a bandwidth-rich environment



Interconnect

- Cray SeaStar2
  - 7.6 GB/s peak bi-directional bandwidth per link
  - 50 nanosecond per link latency
  - 6.3 TB/s bi-section bandwidth
  - 3D Torus: 17x24x24
- Ensures high performance, low-latency communication for MPI and SHMEM jobs.
- MPI latency ~ 8 us.

Node Allocations

- Service nodes (52)
  - 8 GBs memory
  - Login nodes (16): dual-port, gigE
  - Network nodes (4): 10 gigE
  - IO nodes (28): Lustre
  - System nodes (4): boot, SDB + backups
  - 2 SMWs
- Compute nodes (9,688)
  - 4 GBs memory
  - 3.66 GBs Usable memory for MPI jobs

Office of Science
U.S. DEPARTMENT OF ENERGY

Lustre File Systems

- Both $HOME and $SCRATCH are Lustre file systems.
- DDN S2A9500.

HOME

- 4 OSS / 4 OSTs
- ~4.3 TB LUNs
- ~18 TBs total
- Stripe size 1 MB
- Stripe count 1
- Blocksize 4096
- 1 Couplet: 16 Tiers (shared with test system)

SCRATCH (now)

- 20 OSS / 80 OSTs
- ~4.3 TB LUNs
- ~348 TBs total
- Stripe size 1 MB
- Stripe count 4
- Blocksize 4096
- 5 Couplets: 160 Tiers

SCRATCH (in production)

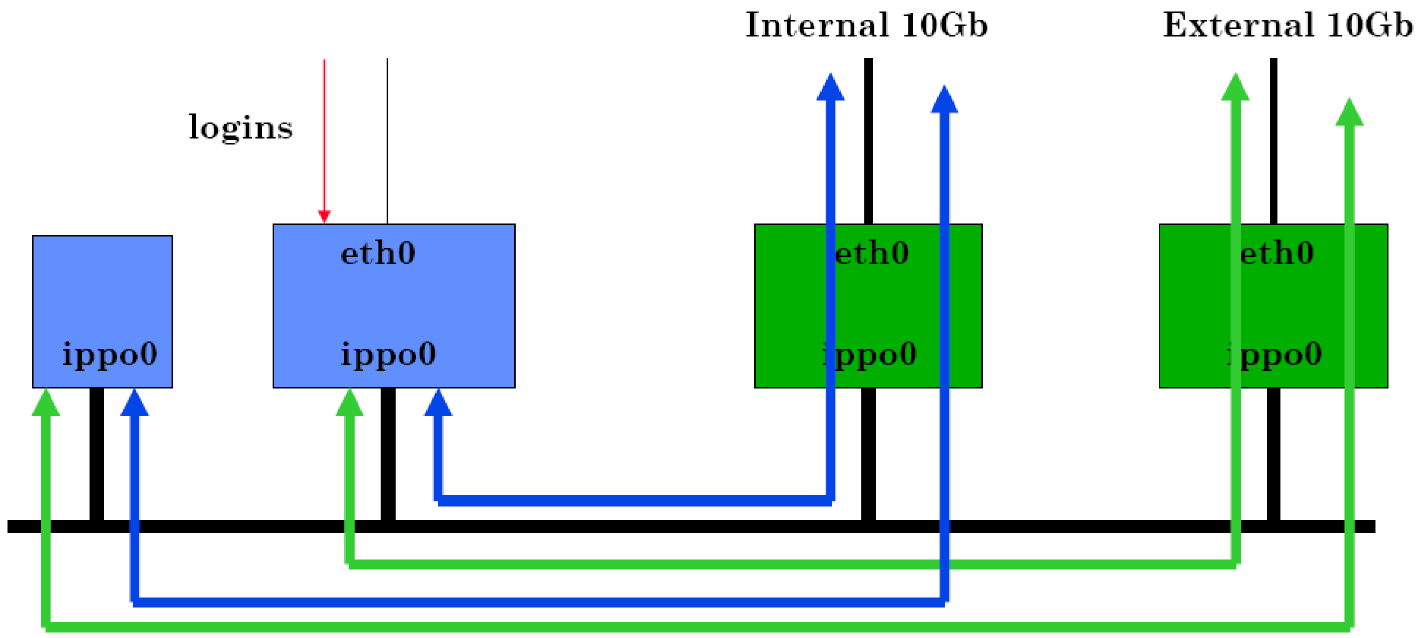
- 24 OSS / 96 OSTs
- ~4.3 TB LUNs
- ~348 TBs total
- Stripe size 1 MB
- Stripe count 4
- Blocksize 4096
- 6 Couplets: 192 Tiers

Office of Science
U.S. DEPARTMENT OF ENERGY

I/O

- IO Libraries
  - NetCDF 3.6.1
  - HDF5 1.6.5 serial and parallel versions
- IO Striping
  - You can change the striping pattern across the OSTs on a *per directory* basis yourself.
  - Default stripe count is 4 on $SCRATCH.
- See I/O talk tomorrow for more details.

Networking



Sixty 4 Gbps Fibre Channel Data Connections
Four 10 Gbps Ethernet Network Connections
Sixteen 1 Gbps Ethernet Network Connections

Software Configuration

- SuSE SLES 9.0 Linux on service nodes
- Compute Node Linux (CNL) for all compute nodes
  - Cray’s light weight Linux kernel
- Portals communication layer
  - MPI, Shmem, OpenMP
- Lustre Parallel File System
- Torque resource management system with the Moab scheduler
  - Most expected functions including backfill, fairshare, advanced reservation
- ALPS utility to launch compute node applications

Programming Environment

- PGI compilers: assembler, Fortran, C, and C++
- GNU compilers: C, C++, and Fortran F77
- Parallel Programming Models: Cray MPICH2 MPI, Cray SHMEM, and OpenMP
- AMD Core Math Library (ACML): BLAS, LAPACK, FFT, Math transcendental libraries, Random Number generators, GNU Fortran libraries
- LibSci scientific library: ScaLAPACK, BLACS, SuperLU
- A special port of the glibc GNU C library routines for compute node applications
- Craypat and Cray Apprentice2
- Performance API (PAPI)
- Modules
- Distributed Debugging Tool (DDT)

Current System Software Levels

- Service Nodes OS: Linux 2.6.16
- Compute Nodes OS: CNL 2.0.14+
- PGI 7.0.7 and many earlier versions
- GCC 4.2.1 and many earlier versions
- XT-MPT 2.0.14
- Torque 2.2.0
- Moab 5.1.0
- Lustre 2.0.14
- Craypat /Apprentice2 3.2.3
- PAPI 3.5.0C
- Module 3.1.6
- ACML 3.6.1
- LibSci 10.0.1
- FFTW 2.1.5, 3.1.1

CNL Advantages *vs.* Catamount for NERSC Users

- **Easier to port from other platforms**
- **Easier transition for users, better for users not to have to go through additional step of CVN, then to CNL**
- **Quicker compiles (at least in some cases)**
- **More functionality (OpenMP, system calls, C/R, sockets)**
- **Requirement for possible quad-core upgrade**
- **More options for debugging tools**
- **Potential for Franklin to be on NGF sooner**

Modules Environment

- *% module list*
- *% module avail*
- *% module load*
- *% module unload*
- *% module swap*
- *% module display*
- *% module show*
- *% module help*

Compiling on Franklin

- PGI compilers are the default.
- GNU compilers are available
  - % *module swap PrgEnv-pgi PrgEnv-gnu*
- Use compiler wrappers
  - ftn for Fortran, cc for C and CC for C++ codes.
  - Compiler wrappers take care of automatically linking system and scientific libraries.
- Compile times are longer and executable file sizes are bigger than with full Linux.

Sample PGI Compiles

- Compile MPI code
  - *% ftn mpi_program.f*
  - *% cc mpi_program.c*
  - *% CC mpi_program.C*
- Compile Fortran Shmem code
  - *% ftn shmem_program.f*
- Compile Fortran OpenMP code
  - *% ftn -mp=nonuma openmp_program.f*
- Compile Fortran Mixed MPI/OpenMP code
  - *% ftn -mp=nonuma mixed_program.f*

PGI Compiler Options

- -fast (= -O2 -Munroll=c:1 -Mnoframe –Mlre)
  - -Munroll=c specifies completely unroll loops with this loop count or less
  - -Mnoframe does not set up a stack frame `
  - -Mlre is loop-carried redundancy elimination
- -fastsse (NERSC recommends)

  (= -fast -Mvect=sse, -Mscalarsse, -Mcache_align –Mflushz)
  - -Mvect=sse,-Mscalarsse optimized for SSE hardware and vectorization
  - -Mcache_align aligns the top level arrays and objects on cache-line boundaries
  - -Mflushz flushes SSE denormal numbers to zero
- Include -Mipa=fast,inline on the link line as well if Interprocedural Analysis (IPA) is used.
- See PGI Compiler talk today for other compiler options.

Using ACML and LibSci

- The AMD Core Math Library (ACML) provides a highly optimized library of BLAS, LAPACK, FFT routines, and a Random Number Generator Suite.
- The Cray Scientific Libraries package (LibSci) includes ScaLAPACK, BLACS, and SuperLU routines.
- The ACML and LibSci modules are automatically loaded upon login. The compiler wrappers (ftn, CC, cc) will automatically load and link to the ACML and LibSci libraries.

Limited glibc Functions on CNL

- Provides limited support of the process control functions such as popen(), fork() and exec(); the resulting processes execute in the limited RAM disk environment on each compute node.
- Supports cpuinfio and meminfo /proc files that contain information about your login node.
- Does not support getgrgid(), getgrnam(), getpwnam(), and getpwuid() functions.
- Does not support custom functions that require a daemon.

CNL Programming Considerations

- **I/O support**
  - stdin, stdout, and stderr are handled by aprun utility.
  - Only processing element (PE 0) could read stdin,
    all PEs could write to stdout and stderr.
- **Timing support**
  - Recommend to use MPI_Wtime(), or a Fortran intrinsic routine such as cpu_time(). Use time difference between two calls for elapsed time.
  - dclock() and etimes() are not supported on the compute nodes.
- **Fortran STOP message**
  - Every process write “FORTRAN STOP” message not scalable
  - Stop this by: setenv NO_STOP_MESSAGE (for csh or tcsh)
    export NO_STOP_MESSAGE (for bash or ksh)

CNL Programming Considerations (cont’d)

- Shared libraries (not supported)
  - No dynamic loading of executable code or shared libraries.
  - Expect in CNL 2.1 release.
- Little-endian support
  - Franklin (and Jacquard) are little-endian,

    Seaborg and Bassi are big-endian.
  - PGI fortran compiler option "-Mbyteswapio" on Franklin could convert between big endian and little endian binary files.

Running jobs

- All Franklin compute nodes are configured as batch nodes.
- Batch jobs could only be run from a Lustre file system: $HOME or $SCRATCH.
- Interactice batch jobs
  - *% qsub –I –V –q interactive -l mppwidth=4* …
  - Wait for a session to start
  - % aprun –n 4 a.out
- /project is mounted via NFS on login nodes only, Franklin compute nodes have no access to /project.
- Small pages only.

Batch Queue Classes

Submit Class	Exec Class	Nodes	Wallclock Limit	Priority	Run Limit	Idle Limit	Queue Limit
interactive	interactive	1-128	30 min	1	2	1	--
debug	debug	1-256	30 min	2	2	1	--
premium	premium	1-4096	12 hrs	4	2	2	--
regular	reg_1	1-127					
	reg_128	128_255					
	reg_256	256-511					
	reg_512	512-1023	12 hrs	5	6	4	--
	reg_1024	1024-2047					
	reg_2048	2048-4095					
	reg_4096	4096-6143	12 hrs	3	1	1	2
	reg_6144	6144-all	arrange	arng	arng	arng	arng
low	low	1-2048	12 hrs	6	--	--	--
special	special	arrange	arrange	arng	arng	arng	arng

Batch Queue Policy

- **Reservation: 128 nodes for interactive/debug, M-F, 5am-6pm.**
- **Jobs use 4096+ nodes are highly favored.**
- **Per user running limit:**
  - **maximum 8 jobs running for all queues combined.**
  - **maximum 2 jobs each running for interactive, debug and premium queues.**
  - **maximum 6 jobs running for reg_1 through reg_2048 execution classes.**
  - **maximum 1 job running for reg_4096 execution class.**
- **Per user idle limit**
  - **maximum 1 job each idle for interactive and debug queues, and maximum 2 jobs idle for premium queue.**
  - **maximum 4 jobs idle for reg_1 through reg_2048 execution classes.**
  - **maximum 1 job idle for reg_4096 execution class.**
- **Disclaimer:** Not fully there yet, still subject to change for fairness and overall throughput. Please check web page for current classes and policy.

ALPS *vs.* PBS Options

aprun option	#PBS -l option	Description
-n 4	-l mppwidth=4	width (number of PEs)
-N 1	-l mppnppn=1	number of PEs per node
-d 2	-l mppdepth=2	number of threads per MPI task

Both “#PBS –l” options and aprun options must exist and match in order to correctly launch a batch job.

To launch a batch job:

```
% qsub batch_script
```

Sample Job Scripts

- Default is dual core mode.

```
#PBS -q debug
#PBS -l mppwidth=4
#PBS -l walltime=00:10:00
#PBS -j eo
#PBS -V
cd $PBS_O_WORKDIR
aprun -n 4 ./a.out
```

- To run in single core mode:

```
#PBS -q debug
#PBS -l mppwidth=4
#PBS –l mppnppn=1
#PBS -l walltime=00:10:00
#PBS -j eo
#PBS -V
cd $PBS_O_WORKDIR
aprun -n 4 –N 1 ./a.out
```

Managing and Monitoring Jobs

- qsub, qdel, qhold
- showq
- qs
- qstat –a
- showstart
- checkjob
- apstat
- xtshowcabs

More Running Modes

- OpenMP jobs
- MPMD jobs
- Multiple sequential jobs
- …
- See Running Jobs talk tomorrow for details.

Memory Considerations

- Each Franklin compute node has 4GB of memory.
- CNL kernel, uses ~250 MB of memory.
- Lustre uses about 17 MB of memory
- Default MPI buffer size is about 72 MB.
- Single core MPI jobs have ~3.66 GB/task.
- Dual core MPI jobs have ~1.83 GB/task.
- Change MPI buffer sizes by setting certain MPICH environment variables.

Current Known Problems

- Node failure with run time error message: “received node failed or halted event for nid xxxx".
- Use bash shell in batch script with IO redirection causes application fail with “exit codes: 127” error message.
- Running multiple independent parallel jobs in background in a single batch script with aprun does not work.
- Job fails intermittently with wall clock limit exceeded.
- GAMESS (Shmem version) and NWCHEM applications cause system crash

Check updates at Franklin “Known Problems” web page at: https://www.nersc.gov/nusers/resources/franklin/problems.php

Charging Factor

- Free during early user period.
- Charge factor for production is to be determined.
  - Early users report 3.6~14.5 relative speedup from Seaborg.
  - Selected Franklin benchmarks (with minimum source code modification) are 4.9 to 7.3 times faster than on Seaborg.

Craypat

- CrayPat tools can be used without source code modification
- *% module load craypat*
- Build application as usual
- *% pat_build -g mpi,io,heap -u myexecutable*
- For instrumentation of a certain group (mpi) functions only:

  % pat_build -g mpi myexecutable
- For instrumentation of user functions only:
  % pat_build -u myexecutable
- Set the environment variable PAT_RT_HWPC to values [1-9].
- Run the instrumented executable *myExecutable+pat*. Generate a file ending in *.xf*.
- *% pat_report -f ap2 [options] <.xf file>*
- Generate a file ending in *.ap2*, for analysis with Apprentice2.
- See Performance Tools talk tomorrow for more details.

Early User Program

- **Solicited applications and enabled users in different batches.**
- **Currently ~250 early users.**
- **Provided lots of useful feedbacks, helped to identify problems, worked with NERSC/Cray with fixes and workarounds.**
- **Warning: Don't publish any results before Franklin is officially accepted by NERSC. Rerun performance results before submission.**
- **Selected early users feedback:**
  - **Victor Ovchinnikov:** Franklin has been easy to use in both programming (porting) and running codes. I am very pleased and impressed with the quality of this machine. I believe it is an exceptional asset to the computational physics community in the US.
  - **Ned Patten:** The friendly user period on Franklin has significantly impacted our science by allowing us to test the capabilities of our code and to establish that such high resolution simulations will be useful and constructive in understanding within-canopy turbulent transport of carbon dioxide.
  - **Terry Ligocki:** I have been able to compile and run large scaling studies with very few problems. The queuing system has worked well and I have not had any problems with libraries, etc.
  - **Kevin Driver:** Overall, I am impressed with the performance and reliability of Franklin during the testing stage.

Franklin Benchmarks

- Application benchmarks
  - CAM3 - Climate model, NCAR
  - GAMESS - Computational chemistry, Iowa State, Ames Lab
  - GTC - Fusion, PPPL
  - MADbench - Astrophysics (CMB analysis), LBL
  - Milc - QCD, multi-site collaboration
  - Paratec - Materials science, LBL and UC Berkeley
  - PMEMD – Life Science, Univ. North Carolina-Chapel Hill

- Micro benchmarks test specific system features
  - Processor, Memory, Interconnect, I/O, Networking
  - Streams, NPB, IOR, mdsrate, Netperf

- Composite benchmarks
  - Sustained System Performance Test (SSP)
  - Effective System Performance Test (ESP)
  - Full Configuration Test
  - Throughput Test

Benchmark Comments

- Application benchmarks represent over 85% of the NERSC workload by discipline area.
- Cover most frequently used programming libraries and programming languages.
- See Benchmark Performance talk tomorrow for more details.

Hands-on Session

- Accounts enabled on Franklin for training registrants.
- Problems login?
  - See me (Helen He) or email: yhe@lbl.gov
- 128 nodes reserved for training class from Sept 18-20, 9am to 7pm:
  - #PBS –q training
  - Wall clock limit is 1 hour.
  - Max cores limit is 256.
- Along with the ongoing reservation of 128 nodes for interactive and debug queues: Mon-Fri, 5am to 6pm, wall clock limit is 30 min.
- Bring your own codes

More information

- Please see Franklin web page at: https://www.nersc.gov/nusers/resources/franklin/
- Early users use NIM password to access Franklin web page.