

Using the CITRIS Cluster and the NERSC Seaborg System

CS 267 Spring 2005

Yozo Hida

yozo@cs.berkeley.edu

February 4, 2005

Outline

CITRIS Cluster

Hardware

Storage

Running Jobs

Tools

Reporting Problems

Measuring Performance

NERSC Seaborg

Hardware

Storage

Running Jobs

Tools

Reporting Problems

x86 Millennium Cluster

Outline

CITRIS Cluster

Hardware

Storage

Running Jobs

Tools

Reporting Problems

Measuring Performance

NERSC Seaborg

Hardware

Storage

Running Jobs

Tools

Reporting Problems

x86 Millennium Cluster

CITRIS Cluster – Itanium 2

- ▶ 64 Dual Itanium 2 nodes (61 compute, 3 frontend).
- ▶ 900 MHz Slow Nodes (22) – McKinley
- ▶ 1.3 GHz Fast Nodes (42) – Madison
- ▶ 32 KB L1, 256 KB L2, and 1.5 / 3.0 MB L3 Cache
- ▶ 4 GB memory per node.
- ▶ Nodes c16-c32 are on fast Myrinet network
≈ 800 MB/s - 1 GB/s.

Itanium 2 (1.3 GHz)

- ▶ Two FMA (fused multiply-add) units, peak of 5.2 GFlop/s.
- ▶ Also has vector instructions (two single-precision entries).
- ▶ 128 integer (64-bit) and floating-point (82-bit) registers.
- ▶ Caches: 32 KB, 256 KB, 3 MB
- ▶ Line sizes: 64 b, 128 b, 128 b.
- ▶ Cache bandwidth: 32 GB/s.
- ▶ Memory bandwidth: 6.2 GB/s.
- ▶ Details in *Intel Itanium 2 Processor Reference Manual for Software Development and Optimization*

Storage

- ▶ Home directory – NFS, slow.
 - Semi-permanent space. Use for keeping long term files.
- ▶ Shared workspace `/work`.
 - ▶ Shared by all the nodes.
 - ▶ `mkdir /work/username`
 - ▶ 30-day deletion policy.
 - ▶ Not backed up, meant for staging runs.
- ▶ Local scratch space `/scratch`
 - ▶ Fast, but each node has its own `/scratch`.
 - ▶ High-speed RAID0 storage.
 - ▶ 10-day deletion policy.
 - ▶ Not backed up, for use with program checkpointing.

Shared Interactive Use

- ▶ 45 nodes (both 900 MHz and 1.3 GHz) for shared use.
- ▶ Frontend login nodes
`{lemon, lime}.millennium.berkeley.edu`
- ▶ Shared use: immediately starts, but speed depends on the load.
- ▶ Don't run heavy jobs on login nodes.
- ▶ Use `gexec` to run jobs.
 - ▶ Set environment variable `GEXEC_SVRS`.
`export GEXEC_SVRS="c1 c2 c3 c4"`
 - ▶ Specify number of processors to `gexec`
`gexec -n 4 /path/to/my/program`
 - ▶ Can use `screen` program to detach from terminal session to reattach later.
- ▶ Use `gstat` to show the load on each node.

Batch System

- ▶ 16 nodes (all 1.3 GHz).
- ▶ Frontend login node:
`grapefruit.millennium.berkeley.edu`
- ▶ Queue system:
 - ▶ gets exclusive use of nodes requested, but must wait in queue.
 - ▶ Jobs requesting many nodes may spend long time in the queue.
- ▶ PBS (Portable Batch System)
 - ▶ Installed in `/usr/pbs/bin`, man pages in `/usr/pbs/man`.
 - ▶ Use `qsub` `myscript` to submit jobs.
Still need `gexec` or `mpirun` in the script.
 - ▶ Use `qstat` for queue status.
 - ▶ Use `qdel` to cancel a submitted job.
- ▶ When a node is allocated, you get both processors.

Example Script

```
#!/bin/sh
#PBS -l nodes=7:ppn=2
#PBS -l mem=400mb
#PBS -l walltime=1:00:00
gexec -n 0 /path/to/my/program
```

- ▶ Lines starting with #PBS are PBS directives.
 - ▶ Requests 7 nodes, 2 processors each (total of 14 processors).
 - ▶ Will use maximum of 400 MB of memory.
 - ▶ Will run at most 1 hour.
 - ▶ Always specify these to guard against program bugs.
 - ▶ Many other directives described in *PBS Pro Users Guide*.
- ▶ Submit with `qsub myscript`.
- ▶ After execution, output (stdout and stderr) saved in files `myscript.o<job_id>` and `myscript.e<job_id>`

Tools on CITRIS

- ▶ Located in `/usr/mill`, `/usr/mill/pkg`.
 - ▶ Add `/usr/mill/bin` to your `$PATH`.
 - ▶ Add `/usr/mill/lib` to your `$LD_LIBRARY_PATH`.
 - ▶ Add `/usr/mill/man` to your `$MANPATH`.
- ▶ Compilers
 - ▶ GNU `gcc` version 3.3.5 in `/usr/bin`.
 - ▶ Intel C++/Fortran 90 compiler `icc` and `ifort` (version 8.1) in `/usr/mill/bin`.
- ▶ Debuggers: `gdb`, `ddd`, `idb` (Intel).
- ▶ Libraries: BLAS, ATLAS, LAPACK in `/usr/lib`.
- ▶ PAPI: `/usr/mill/lib`.

Reporting Problems

- ▶ Mail `support@millennium.berkeley.edu`.
- ▶ Visit 505 Soda Hall if above doesn't solve the problem (also inform me as well).
- ▶ Semi-Frequent issues
 - ▶ Home directory gone: file server problem.
 - ▶ `gexec` hangs. meanwhile try other nodes in `GEXEC_SVRS`.
 - ▶ Installed program doesn't work. Inform `support@millennium`, and/or install them yourself.
 - ▶ Batch system doesn't work. Inform `support@millennium`, perhaps visit 505 Soda.

Outline

CITRIS Cluster

Hardware

Storage

Running Jobs

Tools

Reporting Problems

Measuring Performance

NERSC Seaborg

Hardware

Storage

Running Jobs

Tools

Reporting Problems

x86 Millennium Cluster

Things that Affect Performance

- ▶ Parallel use of functional units:
 - ▶ Floating point vs. integer operations.
 - ▶ Multiple functional units
 - ▶ Fused multiply-accumulate units.
- ▶ Cache effects: cache hit / miss numbers.
- ▶ Paging effects: TLB miss numbers.
- ▶ Algorithm (e.g., basic matrix multiply vs. Strassen)

Timing and Counting

- ▶ Modern systems generally have good timing routines.
 - ▶ High resolution (micro- or nano-seconds).
 - ▶ Overhead much higher than resolution.
 - ⇒ Need enough work to measure time spent.
 - ▶ Wall clock (real time): actual time elapsed.
 - ▶ Processor time: CPU time spent by that process.
- ▶ Most processors have hardware counters for various events.
 - ▶ Cache / TLB misses, Floating point operations, etc.
 - ▶ Number of events that can be counted may be limited.

Timing Routines

- ▶ `clock()` (res 1 ms, overhead 0.4 μ s)
 - ▶ Measures time used by the process
 - ▶ Resolution OS/Hardware dependent
 - ▶ `CLOCKS_PER_SEC` does not indicate resolution.
- ▶ `gettimeofday()` (res 1 μ s, overhead 0.4 μ s).
 - ▶ Wall clock
 - ▶ Usually in microsecond resolution.
- ▶ `clock_gettime()` (res 1 μ s, overhead 0.4 μ s).
 - ▶ `CLOCK_REALTIME` measures wall clock.
 - ▶ `CLOCK_PROCESS_CPUTIME_ID` measures process time.
 - ▶ May require linking with `-lrt`.
- ▶ MPI: `MPI_Wtime()`
- ▶ PAPI: `PAPI_get_real_cyc()`, `PAPI_get_real_usec()`

Profilers and Counters

- ▶ `gprof` – general profilers, gives a general idea of where the bottleneck is.
- ▶ `perfctr` – Linux x86
- ▶ `hpm` – IBM Power series
- ▶ PAPI – common API for many platforms (including CITRIS and Seaborg).
 - ▶ Various events: number of cycles, cache misses, flops, etc.
 - ▶ Note: not all events on all platforms.
 - ▶ Different incompatible versions (3.0 on Seaborg, 2.3.x on CITRIS).
 - ▶ See <http://icl.cs.utk.edu/papi/>

Using PAPI

```
#include <stdio.h>
#include <unistd.h>
#include <papi.h>

int main() {
    long_long tm1, tm2;
    PAPI_library_init(PAPI_VER_CURRENT);
    tm1 = PAPI_get_real_cyc();
    sleep(1);
    tm2 = PAPI_get_real_cyc();
    printf("%lld\n", tm2 - tm1);
    return 0;
}
```

Using PAPI

```
% gcc -o papi -O2 -Wall \  
  -I/usr/mill/pkg/papi/include/papi-2.3.4 \  
  papi.c -L/usr/mill/lib -lpapi  
% ./papi  
1300064584  
%
```

- ▶ Gives 1.3×10^9 cycles,
which is what we expect on 1.3 GHz system.

Outline

CITRIS Cluster

Hardware

Storage

Running Jobs

Tools

Reporting Problems

Measuring Performance

NERSC Seaborg

Hardware

Storage

Running Jobs

Tools

Reporting Problems

x86 Millennium Cluster

NERSC Seaborg

- ▶ 416 (380 compute) 16-way SMP nodes (peak of 10 TFlop/s).
- ▶ Total of 6656 (6080 compute) 375 MHz IBM Power 3+ processors.
- ▶ Total of 7.3 TB memory.
- ▶ 44 TB of disk space in GPFS.
- ▶ Additional storage in HPSS: 8.8 PB, 50 TB disk cache, 3.2 GB/s theoretical bandwidth.
- ▶ Two high speed network card per node.
- ▶ Rest of nodes for GPFS, login, network, spare.

IBM Power 3+

- ▶ Clock speed: 375 MHz.
- ▶ Two FMA units: peak performance of 1.5 Gflop/s.
- ▶ Caches: L1 inst. 32 KB, L1 data 64 KB, L2 8 MB (6.4 GB/s).
- ▶ L1 line size: 128 b.
- ▶ Memory per node: 16-64 GB (4 have 64 GB; 64 have 32 GB; 312 have 16 GB).
- ▶ Memory bandwidth: 1.6 GB/s (1.3 GB/s daxpy).

Seaborg Storage

- ▶ Home directory ($\$HOME$)
 - ▶ 15 GB (GPFS), available from every node.
 - ▶ Not backed up.
- ▶ Scratch space ($\$SCRATCH$)
 - ▶ 33 TB (GPFS), shared by everyone, user quota of 250 GB.
 - ▶ Available from every node.
 - ▶ Nominal 7-day deletion policy, but files may be deleted anytime after the job finishes.
 - ▶ Not backed up.
- ▶ User quota can be checked by `myquota`.
- ▶ **Do not use `/tmp` or `/var/tmp`.**

Interactive Jobs

- ▶ Login nodes: `seaborg.nersc.gov`
- ▶ Debug on login nodes and special debug nodes.
- ▶ Interactive jobs limited to 8 nodes, 30 minutes.

Batch System

- ▶ LoadLeveler queue system
 - ▶ Batch jobs use commented shell script (as in CITRIS).
 - ▶ `llqs` – lists full queue.
 - ▶ `llqs -u username` – lists your jobs.
 - ▶ `llsubmit myscript` – submit a job in myscript.
 - ▶ `llcancel <job_id>` – cancel a job in queue.
- ▶ More information at
http://www.nersc.gov/nusers/resources/SP/running_jobs/

Example Script

```
#@ job_name = myjob
#@ account_no = mp309
#@ output = myjob.out
#@ error = myjob.err
#@ job_type = parallel
#@ notification = complete
#@ network.MPI = csss,not_shared,us
#@ node_usage = not_shared
#@ class = regular
#@ tasks_per_node = 16
#@ node = 1
#@ wall_clock_limit = 01:00:00
#@ queue
./my_program
```

See http://www.nersc.gov/nusers/resources/SP/running_jobs/batch.php for details.

Class Repository

- ▶ Supercomputer time is a limited resource!
- ▶ Class repository: mp309.
- ▶ Class allocation: 20,000 processor-hours.
- ▶ Each person has been allocated $\approx 6\%$ of this.
- ▶ Please conserve allocated processor time by
 - ▶ Debug thoroughly before submitting large job.
 - ▶ Using as many of the 16 processors in each node. (If you use a node, you will be charged for all 16 processors on it.)
 - ▶ Setting memory and time limits to your jobs.
 - ▶ Use CITRIS cluster.
- ▶ If you run out of your allocation:
 - ▶ Use your project partners' allocation.
 - ▶ Contact me, I'll see what I can do.

Tools on Seaborg

▶ Modules

- ▶ See all modules: `module avail`.
- ▶ To use a particular module: `module use <modulename>`.
- ▶ IBM C compiler: `module xlc`.

▶ Compilers

- ▶ `gcc/g++/g77` 3.4.1, 3.3, 3.2.1.
- ▶ IBM compilers `xlc` (C), `xlc` (C++), `xlf` (F77), `xlf90` (F90).
- ▶ Details at <http://www.nersc.gov/nusers/resources/SP/programming.php>.

▶ Debuggers: `gdb`, `totalview`, `ddd`, `dbx`.

▶ GNU Tools: `module use gnu`.

Using PAPI on Seaborg

```
% module load papi
% xlc -o papi papi.c $PAPI
% ./papi
375064125
```

- ▶ PAPI version 3.0 on Seaborg (2.3.2 on CITRIS).

Reporting Problems

- ▶ Check status
 - ▶ <http://www.nersc.gov/nusers/status/>
 - ▶ <http://www.nersc.gov/nusers/status/motd.php>
- ▶ Contacts
 - ▶ Seaborg Docs:
<http://www.nersc.gov/nusers/resources/SP/>
 - ▶ Help page: <http://www.nersc.gov/nusers/help/>
- ▶ Password issues
 - ▶ Log into sadmin.nersc.gov to set/change password.
 - ▶ Wait an hour after password changes.

Outline

CITRIS Cluster

Hardware

Storage

Running Jobs

Tools

Reporting Problems

Measuring Performance

NERSC Seaborg

Hardware

Storage

Running Jobs

Tools

Reporting Problems

x86 Millennium Cluster

x86 Millennium Cluster

- ▶ Pentium IIIs, some dual, some quad.
- ▶ Mix of 500, 550, 700 MHz nodes.
- ▶ 512 KB, 1 MB L2 caches.
- ▶ Use for testing / fun.
- ▶ Login nodes {napa, sonoma}.millennium.berkeley.edu.
- ▶ Use gexec with nodes mm1, mm2, etc.