The *CSEM* Multi-processor Computing Facility

J. Knap

California Institute of Technology

Hardware Configuration

Front-end

Machine name: **alicante.aero.caltech.edu**
COMPAQ AlphaServer **DS10** 617 MHz
2GB memory and 56GB disk space

Compute nodes

Machine names: **n01-n34**
34 *COMPAQ AlphaServer*s **DS10L** 617 MHz
512MB memory and 10GB disk space

Machine names: **n35-n62**
28 *COMPAQ AlphaServer*s **DS10L** 617 MHz
1GB memory and 20GB disk space

Interconnect

100 Mbit/s Fast Ethernet

Software Environment

Operating System

RedHat Linux 7.0 for Alpha

Tools & Utilities

GNU tools – editors, compilers, interpreters, debuggers, version control, *etc.*

LinuxAlpha PowerTools – suite of compilers, libraries and code development tools

MPICH – portable implementation of the *Message Passing Interface* (MPI) Standard

Portable Batch System – queue management and job scheduling

LinuxAlpha PowerTools – Compilers

Compaq FORTRAN Compiler for Linux Alpha, V1.1 – fort

Fortran 95 implementation

```
% fort -o a.out main.f
% man fort
```

Compaq C++ Version 6.3 for Linux Alpha – cxx

C++ Standard Library

```
% cxx -o a.out main.cc
% man cxx
```

Compaq C Version 6.2 for Linux Alpha – ccc

```
% ccc -o a.out main.c
% man ccc
```

LinuxAlpha PowerTools – Libraries

Compaq Portable Math Library (CPML)

Drop-in replacement for the standard math library `/usr/lib/libm.a`
Performance optimized for the *Compaq Alpha* architecture

```
% ccc -o a.out main.cc -lcpml
```

Compaq Extended Math Library (CXML)

BLAS – Basic Linear Algebra Subroutines
LAPACK – An extensive library of Linear System and Eigenproblem Solvers
Sparse Linear System Solvers – A library of direct and iterative sparse solvers
Signal Processing – FFTs, sine/cosine transforms, convolution, correlation and filters

```
% ccc -o a.out main.cc -lxml
```

Documentation

`http://h18000.www1.hp.com/math/documentation/index.html`

LinuxAlpha PowerTools - Debugger

Ladebug Symbolic Debugger

- Symbolic, source-level debugger
- Command-line interface
- Extensive support for debugging programs written in C, C++, and FORTRAN
- No support for third-party (e.g. GNU) compilers
- `% ladebug a.out`

Documentation

`http://h21007.www2.hp.com/dspp/files/unprotected/ladebug/ladebug_man_eng.htm`

Message Passing Interface (MPI)

MPICH – freely available, portable implementation of *MPI*, the Standard for message-passing libraries. *MPI* – the software framework for the development of distributed multi-processor applications.

Location

```
/usr/local/mpich
```

MPI-aware Compilers

```
/usr/local/bin/mpicc
/usr/local/bin/mpiCC
/usr/local/bin/mpif77
```

Job launcher

```
/usr/local/bin/mpirun
```

Documentation

http://www-unix.mcs.anl.gov/mpi/mpich

Portable Batch System (PBS)

PBS – ''a networked subsystem for submitting, monitoring, and controlling a work load of batch jobs on one or more systems''

All jobs must be executed via PBS

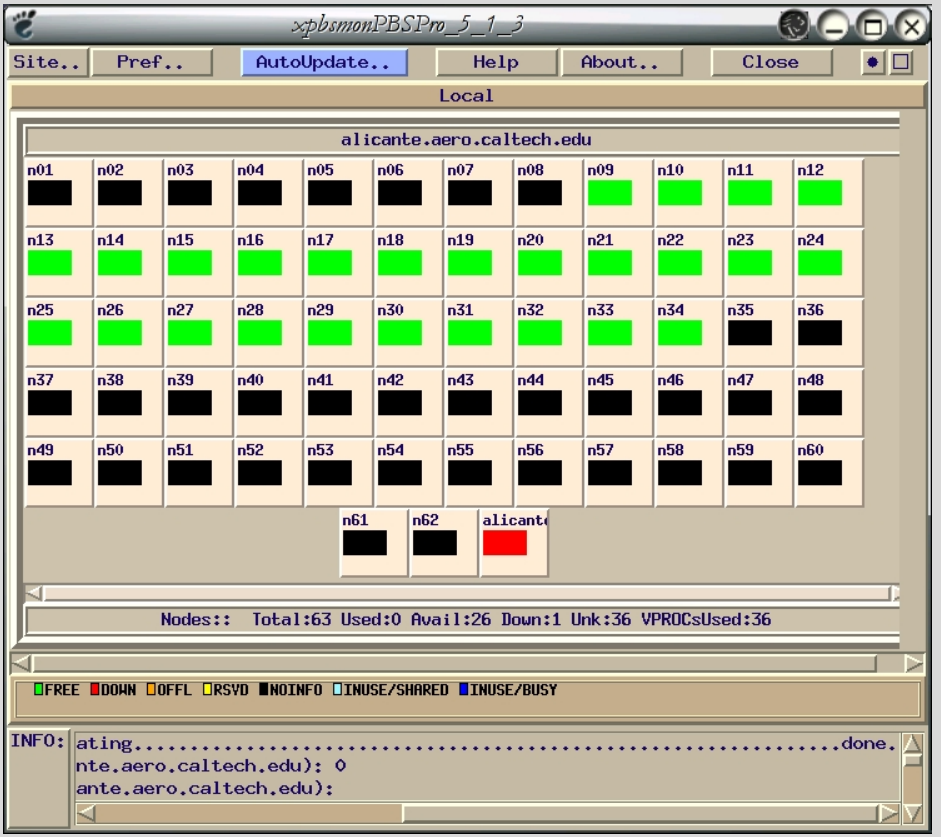
Location

`/usr/pbs`

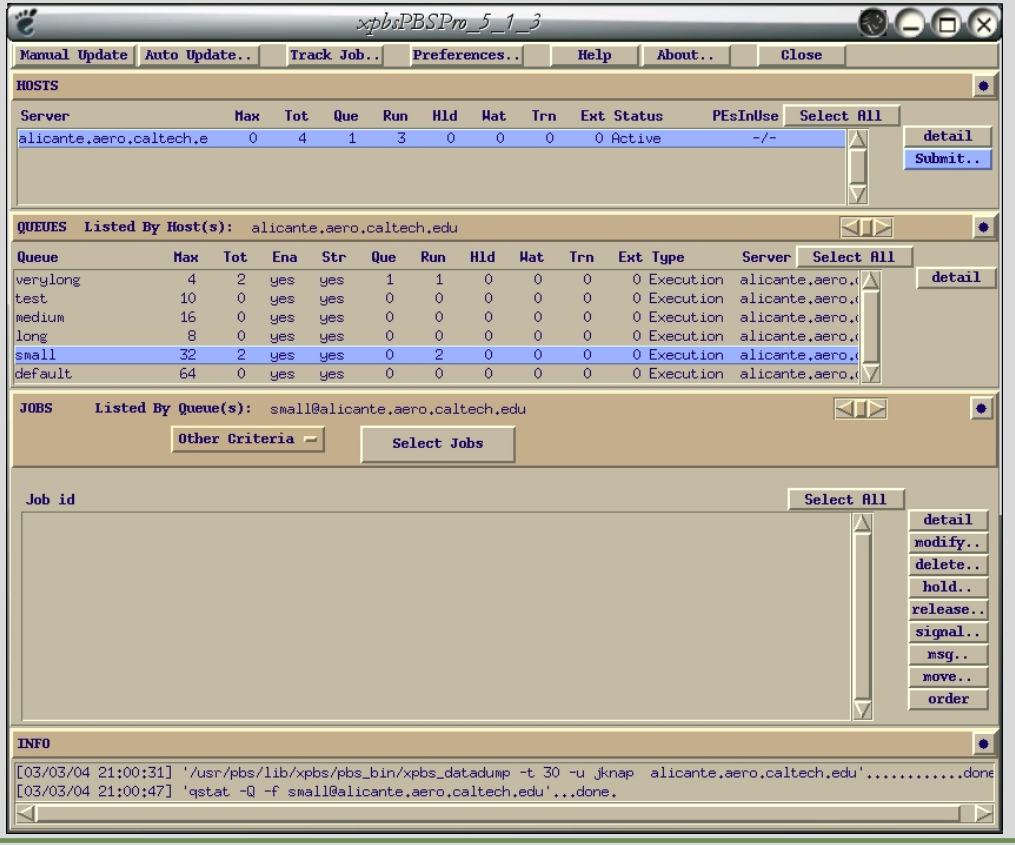
Documentation

- `% man pbs`
- `http://www.solids.caltech.edu/webpage/pbs.html`

Portable Batch System (PBS) – xpbsmon



Portable Batch System (PBS) – xpbs



Portable Batch System (PBS) – Queue Limits

Queue	CPU Time [h]	Max. CPUs	Max. No. Jobs
default	2	62	62
small	12	62	32
medium	24	62	16
long	48	32	8
verylong	120	32	4

User Home Directories

- Home directories ***shared*** across the entire cluster
- The uniform location of the home directory: /home/<username>
- ***Avoid*** launching jobs that frequently output data in their home directories–use the scratch filesystem
- User quotas: 400 MB per-user disk space
- Automatic daily tape-backups

Scratch Directories

- Scratch directories ***local*** to all compute nodes
- 4 or 8 GB of disk space available
- `/scratch/<username>` e.g. `/scratch/jknap`
- Files in `/scratch` purged in regular intervals based on size

	size	< 1MB	7 days
1MB <	*size*	<= 10MB	5 days
	size	> 10MB	3 days

- E-mail notifications sent out before any file removal
- Scratch directories can be accessed directly on the front-end as:

  `/scratch/<nodename>/<username>`

  where: nodename = **n01** ... **n62**, and username is the UID, e.g

  `/scratch/n13/jknap`

Getting Started

Getting an account: send an e-mail message to

cluster@aero.caltech.edu

Accessing the cluster

```
% ssh <username>@alicante.aero.caltech.edu
```

Getting help: send an e-mail message to

cluster@aero.caltech.edu

Website:

http://www.solids.caltech.edu/cluster.html

Example

Problem: compute the value of π calculating

$$s = \int_0^1 \frac{4}{1+x^2} dx.$$

Solution: uniformly divide the interval $[0, 1]$ into n subintervals and invoke the midpoint rule of numerical integration

$$\int_a^b f(x)dx \approx f(\frac{a+b}{2})(b-a).$$

Take advantage of the intrinsic ***parallelism*** – evaluations of the midpoint rule a each subinterval independent.

Original file: `/usr/local/mpich/examples/pi3.f`.

Example – MPI Calls

MPI initialization

```
      call MPI_INIT( ierr )
      call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
      call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
```

Brodcast the number of intervals to all tasks

```
      call MPI_BCAST(n,1,MPI_INTEGER,0,MPI_COMM_WORLD,ierr)
```

Collect contributions from tasks

```
      call MPI_REDUCE(mypi,pi,1,MPI_DOUBLE_PRECISION,MPI_SUM,0,
     $     MPI_COMM_WORLD,ierr)
```

Example – Compilation

Invoke MPI-aware compiler: `mpif77`

```
[jknap@alicante ~/tutorial]$ mpif77 pi3.f
```

Make sure shell path contains: `/usr/local/bin`

Alternative compilation:

```
[jknap@alicante ~/tutorial]$ /usr/local/bin/mpif77 pi3.f
```

... or:

```
[jknap@alicante ~/tutorial]$ f77 -I/usr/local/mpich/include pi3.f
-L /usr/local/mpich/lib -lmpich
```

Executable: `a.out`

Example – Job Submission

Batch PBS script: pi.pbs

```
#PBS -l nodes=4
#PBS -q test@alicante.aero.caltech.edu
/usr/local/bin/mpirun -np 4 a.out < input.txt
```

Submit a batch job

```
[jknap@alicante ~/tutorial]$ qsub pi.pbs
11945.alicante.aero.caltech.edu
```

Query job status

```
[jknap@alicante ~/tutorial]$ qstat
Job id           Name             User             Time Use S Queue
---------------- ---------------- ---------------- -------- - -----
11945.alicante   pi.pbs           jknap            00:00:00 R test
```

Output files: pi.pbs.o<job id>, pi.pbs.e<job id>

Example – Job Submission (2)

Change directory before invoking the executable

```
#PBS -l nodes=4
#PBS -q test@alicante.aero.caltech.edu

cd /home/jknap/tutorial

/usr/local/bin/mpirun -np 4 a.out < input.txt
```

Can use xpbs to compose the PBS script

```
[jknap@alicante ~/tutorial]$ /usr/pbs/bin/xpbs
```

Example – Job Submission (3)

Output node names to STDOUT

```
#! /bin/sh
#PBS -l nodes=4
#PBS -q test@alicante.aero.caltech.edu

echo "Running on nodes "
cat ${PBS_NODEFILE}
number_nodes=‘cat ${PBS_NODEFILE} | wc -l‘

cd /home/jknap/tutorial
/usr/local/bin/mpirun -np ${number_nodes} a.out < input.txt
```

Example – Job Submission (4)

STDOUT/STDERR redirection

```
#! /bin/sh
#PBS -l nodes=4
#PBS -q test@alicante.aero.caltech.edu

#PBS -o output
#PBS -e error

cd /home/jknap/tutorial
/usr/local/bin/mpirun -np 4 a.out < input.txt
```

Example – Interactive Job

```
[jknap@alicante ~/tutorial]$ qsub -I -l nodes=4 -q test
qsub: waiting for job 11968.alicante.aero.caltech.edu to start
qsub: job 11968.alicante.aero.caltech.edu ready

[jknap@n60 ~]$ cd tutorial
[jknap@n60 ~/tutorial]$ /usr/local/bin/mpirun -np 4 a.out < input.txt
```

Summary

Getting an account: send an e-mail message to

cluster@aero.caltech.edu

Website:

http://www.solids.caltech.edu/cluster.html