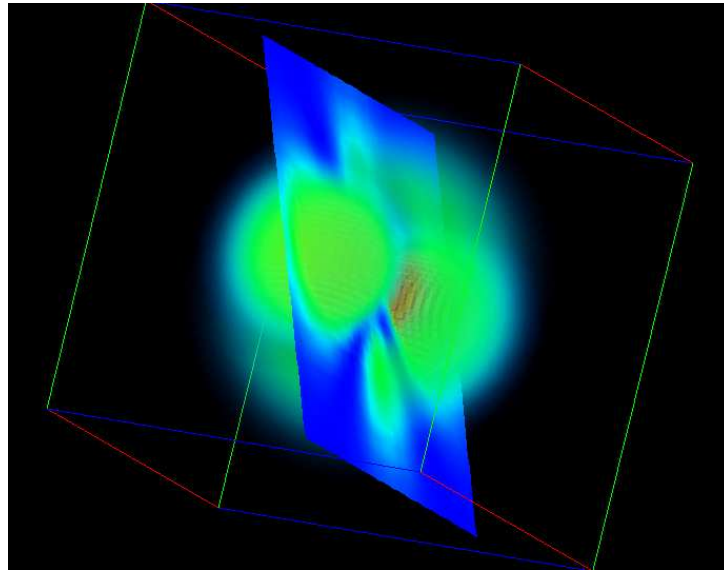


# High Performance Computing at ac3

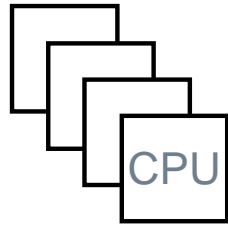


# HPC Platforms available to ac3 users

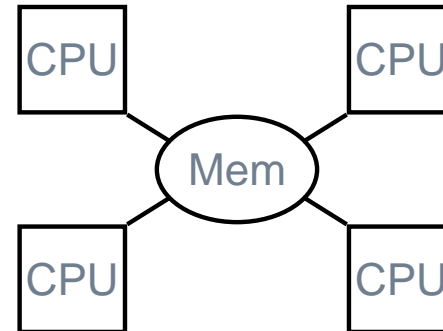
- Machines at *ac3*, a company owned jointly by a number of NSW universities and NSW State Government.
- Machines at *APAC*, a national partnership of state based organisations such as *ac3*.

# Types of HPC

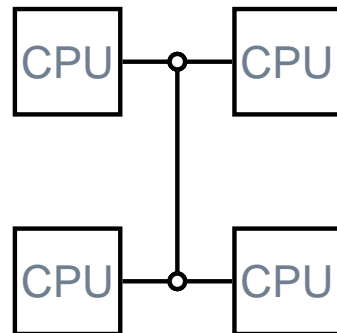
1. Vector



2. Shared Memory (SMP)



3. Clusters



The **Earth Simulator** (currently the third fastest machine in the world) is a cluster of vector SMPs.

SMP clusters are very popular.

# @ac3: SGI SMP systems



**SGI Origin** (Clare) 64 processors with 32GB shared memory. 50GFlops peak speed - decommissioned.

**SGI Power Challenge** (Napier) 28 processors with 4GB shared memory. 11GFlops peak. Used for interactive code development and educational purposes - decommissioned.

# @ac3: NEC Vector system



**NEC SX5** (Hunter) 2 vector processors sharing  
12GB memory. 16GFlops peak

# @ac3: Dell cluster system



**Dell Beowulf cluster** (Barossa) 155 dual processor Pentium 4 nodes at 3GHz. 1.7TFlops peak, 290 GB memory in total.

Benchmarked at 1.096 TFlops (Linpack), this is Australia's most powerful academic computer, and is currently 330th on the Top 500 list. It entered the list at position 108.

# APAC National Facility



**HP (Compaq) SC 1 x 16 processor SMP node (32GF peak) with 16GB memory, and 120 quad processor nodes (960GF peak, 700GB memory).** It is replaced by an Altix cluster consisting of approx. 1000 CPU's.

**Dell Beowulf cluster LC 150 Pentium 4 nodes at 2.66GHz. 800GFlops peak, 150GB memory.**



# Deciding on a Machine

To decide which machine is best for your task, you need to consider:

# Deciding on a Machine

To decide which machine is best for your task, you need to consider:

- How much memory you need.

# Deciding on a Machine

To decide which machine is best for your task, you need to consider:

- How much memory you need.
- How much independent parallelism you have.

# Deciding on a Machine

To decide which machine is best for your task, you need to consider:

- How much memory you need.
- How much independent parallelism you have.
- Is your code parallelised in a shared memory or distributed memory way.

# Deciding on a Machine

To decide which machine is best for your task, you need to consider:

- How much memory you need.
- How much independent parallelism you have.
- Is your code parallelised in a shared memory or distributed memory way.
- Is your application compute bound, communication bound or I/O bound.

# Barossa is good for...

- Lots of independent tasks needing less than 1GB memory

# Barossa is good for...

- Lots of independent tasks needing less than 1GB memory
- SMP tasks up to 2GB memory

# Barossa is good for...

- Lots of independent tasks needing less than 1GB memory
- SMP tasks up to 2GB memory
- Compute bound distributed memory tasks. (Communication bound tasks may be better served on the SC or Swan).

# Barossa is good for...

- Lots of independent tasks needing less than 1GB memory
- SMP tasks up to 2GB memory
- Compute bound distributed memory tasks. (Communication bound tasks may be better served on the SC or Swan).
- Barossa is not particularly good for I/O bound tasks.

# Getting Access

- Access to ac3 systems is obtained through your *Campus Coordinator*. This will enable minimal usage, i.e. no project
  - See <http://www.ac3.edu.au> for details
  - Signed paper form to ac3
  - Online researcher database entry (providing information about your work)
- Access to APAC system is through applying for a research grant at <http://nf.apac.edu.au>:
  - Startup Grant (1000SUs) - 1 SU  $\approx$  1 hr on a 1GHz CPU
  - APAC Merit Allocation Grant
  - ac3 Research resource allocation (Partner

# ac3 Partner share procedure

- Fill out online form at <http://nf.apac.edu.au> (if new account)
  - Fill out ac3 resource allocation application at <http://www.ac3.edu.au>
  - Fill out researcher database entry
- Apply for an APAC grant through ac3 or do a private allocation

# Introductory User Guides

These are a *must read!*

Topics covered:

- Logging in and setting up your account
- Using compilers and numerical libraries
- Parallelising code
- Using batch system
- Using scratch files systems

For ac3's systems: see <http://www.ac3.edu.au>

For APAC NF: see <http://nf.apac.edu.au>

# Getting Help

- Scientists should do *science*, not *computer science*!
- Seek professional help, don't bang your head on a brick wall!
- First port of call: [help@ac3.com.au](mailto:help@ac3.com.au)
- HPCSU (UNSW) part of ac3 user support team; USyd Vislab has an HPC expert.

# Using batch queues

PBS used on Barossa, Swan and APAC machines. Hunter uses NQS variant:

**qsub** submit a job

**qstat** query job or queue status

**qdel** delete a job from queue

qsub.pl script provides a PBS compatible interface for Hunter

# qsub options

- lwalltime=, -lcpus= request walltime or CPU time
- lncpus=, -lnodes= request a certain number of CPUs or nodes (parallel jobs)
- lmem= request a certain amount of memory
- lnodes= $x$ :ppn=2 Request  $2x$  CPUs, spread over  $x$  nodes.
- q *queuename* Specify a queue
- A *projectname* Specify a project you are joining in

# qstat output

```
[rks@barossa gen-random]$ qstat -u rks
```

```
barossa.ac3.com.au:
```

Job ID	Username	Queue	Jobname	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
83991	rks	workq	pt0.91	1	1	512mb	72:00	R	04:05
83992	rks	workq	pt0.92	1	1	512mb	72:00	R	03:53
84056	rks	workq	gen-ran	1	1	512mb	72:00	R	00:01

NDS Number of nodes TSK Task (interesting for parallel jobs)

# Maui scheduler

Used on Clare and Barossa  
showq (or “pbs showq” on Barossa) can be used  
to obtain additional queue information:

```
[rks@barossa gen-random]$ pbs showq -i
```

JobName	Priority	XFactor	User	Procs	WCLimit	SystemQueueT
84057*	10003	1.8	alexg	20	1:00:00	Mon Jun 28 14:33:
84059*	8721	1.0	houska	8	6:06:00:00	Mon Jun 28 15:07:
84040*	6749	1.0	ahmadj	1	3:00:00:00	Mon Jun 28 12:54:
84041*	6749	1.0	ahmadj	1	3:00:00:00	

$$\text{XFactor} = 1 + \frac{\text{queued time}}{\text{requested walltime}}$$

→ larger XFactor causes larger likelihood to get started

# Priority

- requested walltime -
- length of time queued +
- fare share FS (number of submitted jobs) -
- resource allocation (memory) -
- submitted queue +/-

```
$ diagnose -p
```

```
Job      PRIORITY*   Cred( User:Acct:Class)   FS( User:Acct)   Serv(QTi
```

Job	PRIORITY*	Cred( User:Acct:Class)	FS( User:Acct)	Serv(QTi
444531	20990345	100.0( 0.0:41962: 10.0)	0.0(525.1: 0.0)	0.0(382

# Preemption

- When another job is suspended so that a high priority job can start.
- Automatic preemption is available on Swan (next) and the APAC SC, but not on Barossa. Running jobs will run to completion or their requested limit.
- A large parallel job will cause queues to drain. Jobs with small wallclock requests will “backfill” (Barossa’s alternative to preemption).
- Because Barossa does not have preemption, long running jobs (in the xlong queue) are

# Estimating when your job will start

```
$ showstart 444531
job 444531 requires 16 procs for 1:06:25:00
Earliest start in      1:15:28:23 on Sun Jan 16 08:44:26
Earliest completion in 2:21:53:23 on Mon Jan 17 15:09:26
Best Partition: DEFAULT
```

- This estimate is **conservative**. It assumes all jobs will run for their requested time. Resources may become available sooner.
- It cannot be used for jobs submitted in the future which get higher priority (hence the caveat “earliest”).

# Understanding the machine's state

```
$ qview
barossa015 . 1 441911      carlm (1200mb, 512mb)
barossa025 . 1 444043      bsoule (1200mb, 512mb)
barossa037 . 1 444292  lambui01 ( 128mb, 512mb)
barossa050 . 0      free
barossa055 0 0      free
barossa064 . 1 444534  sinavafi ( 512mb, 1700mb)
barossa073 0 0      free
barossa098 0 0      free
barossa115 0 0      free
barossa117 0 0      free
barossa137 . 1 444539  sinavafi ( 512mb, 1700mb)
```

- The 0 status indicates a node is offline for some reason
- Column 3 indicates number of CPUs in use on that node.

# Queues on Barossa

**priority** Highest priority, charged at  $3\times$  → SU is affected stronger

**xlarge** For jobs with more than 32 nodes

**xlong** For jobs longer than 3 days

**checkable** For jobs that can be checkpointed

**single** Single CPU jobs

**stampfl,robinson** private queues

# Notes on queues

- xlong and xlarge require you to email [help@ac3.com.au](mailto:help@ac3.com.au) before jobs are scheduled
- checkable jobs can be killed at any time to make room for xlarge jobs → restart opportunity
- default — if no queue specified, PBS will route job to most appropriate
- workq — a “catchall” queue, low priority, with reducing share (ac3 will reduce that queue further)

# Checkpointing

Checkable jobs should

# Checkpointing

Checkable jobs should

- at minimum specify `-r`. Job will be requeued if killed.

# Checkpointing

Checkable jobs should

- at minimum specify `-r`. Job will be requeued if killed.
- ideally write a checkpoint file, so no work is lost

# Checkpointing

Checkable jobs should

- at minimum specify `-r`. Job will be requeued if killed.
- ideally write a checkpoint file, so no work is lost
- could also use a log file to determine where calculation left off

# Writing Checkpoint files

- **SIGUSR1** and **SIGTERM** is sent by batch system prior to job being killed.

# Writing Checkpoint files

- **SIGUSR1** and **SIGTERM** is sent by batch system prior to job being killed.
- Trapping these signals may not give sufficient time to reach a checkpointable part of the program.

# Writing Checkpoint files

- **SIGUSR1** and **SIGTERM** is sent by batch system prior to job being killed.
- Trapping these signals may not give sufficient time to reach a checkpointable part of the program.
- Alternatively, write a checkpoint whenever possible (within reasonable I/O demands)

# Writing Checkpoint files

- **SIGUSR1** and **SIGTERM** is sent by batch system prior to job being killed.
- Trapping these signals may not give sufficient time to reach a checkpointable part of the program.
- Alternatively, write a checkpoint whenever possible (within reasonable I/O demands)
- Trap **SIGTERM** and record its arrival. If the signal was received before the checkpoint write, exit immediately, if received during the checkpoint, exit immediately after checkpoint.

# Checkpointing options

No system support for checkpointing is provided on Barossa, applications must provide their own! **SIGUSR1** and **SIGTERM** are sent from the system but no software is provided for trapping the signal.

Use Classdesc<sup>a</sup> for C++ or FClassdesc<sup>b</sup> for Fortran90 to write a binary file representing the relevant state data of your program.

Traditional (non-classdesc) checkpointing techniques are also possible<sup>c</sup>, but code is harder to maintain.

---

<sup>a</sup><http://parallel.hpc.unsw.edu.au/classdesc>

<sup>b</sup><http://parallel.hpc.unsw.edu.au/fclassdesc>

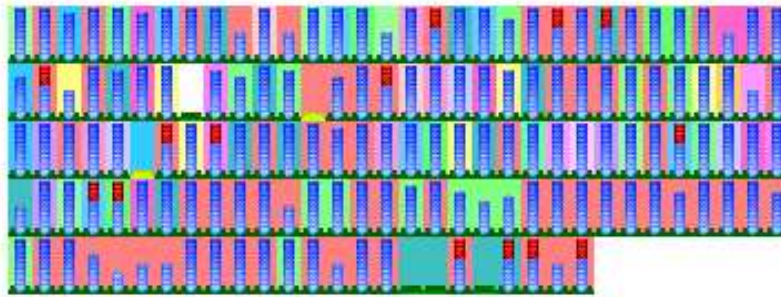
# clumon

<http://barossa.ac3.com.au/clumon>

## BAROSSA.AC3.COM.AU Cluster Monitor

[Main](#)  
[Hosts](#)  
[Resources](#)  
[Queues](#)  
[Jobs](#)  
[Alerts](#)  
[Adm](#)  
[Notes](#)  
[Help](#)

Nodes: 152



Job	Owner	Job Name	Queue	State	Nodes	Time Used
<a href="#">79374</a>	kutteh	v9.0-h9.0	workq	RUNNING	1	77:46:27
<a href="#">79414</a>	kutteh	v12.0-h7.0	workq	RUNNING	1	41:59:40
<a href="#">79438</a>	kutteh	v14.0-h10.0	workq	RUNNING	1	36:34:17
<a href="#">79439</a>	kutteh	v14.0-h12.0	workq	RUNNING	1	36:34:09
<a href="#">79469</a>	kutteh	v16.0-h52.0	workq	RUNNING	1	19:45:58
<a href="#">79470</a>	kutteh	v18.0-h1.5	workq	RUNNING	1	18:47:50

# Batch scripts

- Typically bourne shell programs, but can be any interpreted language (eg Perl) where # is a comment character.
- qsub options can be specified on a line beginning with #PBS. These are overridden by command line.
- PATH usually doesn't include "." e.g set appropriate path with `PATH=/opt/mpich-1.2.5.10-ch_p4-gcc/bin:$PATH` or in `.bashrc` set, e.g.:  
`MPICH=/opt/mpich-1.2.5.10-ch_p4-gcc/bin;`  
`export MPICH`

# Batch scripts ...

- Environment contains extra information:
  - \$PBS\_O\_WORKDIR** Place from where job was submitted
  - \$PBS\_NODEFILE** list of node names attached to your job. eg `wc -l $PBS_NODEFILE` will return the number of CPUs your job is running on.
  - \$PBS\_JOBID** is your current job's *jobid*. eg `/scratch/$PBS_JOBID` is the name of a temporary directory on a node's local disk for intensive I/O applications.

# (Trivially) Parallel Batch scripts

```
#!/bin/sh
n=0
while [ $n -lt 10 ]; do
    echo $n >parm${n}.dat
    cat >scr$n <<EOF
#PBS -l cput=03:00:00 -l mem=128MB
a.out parm${n}.dat >out${n}.dat
expr `cat parm${n}.dat` + 10 >parm${n}.dat
qsub scr$n
EOF
    qsub scr$n
    n=`expr $n + 1`
done
```

# OpenMP jobs

- Compiler can autoparallelise, or make use of manual OpenMP compiler directives. See intro guides for specific compiler flags.
- Specify number of threads via `OMP_NUM_THREADS` environment variable. On Barossa, job can make use of 2 threads, SC can make use of 4 and Swan up to 16.
- Use `-lnode=1:ppn=2` option on Barossa. Use `-lnode=1:ppn=16` to select APAC's big SMP node on the SC

# MPI jobs

- Swan (the new Altix) has SGI MPI and MPICH
- Barossa has LAM and MPICH
- LC has LAM
- SC has an elan version, SC will be replaced by a large Altix

In terms of network performance, the order is Swan, SC, Barossa, LC.

In terms of processor performance, the order is Barossa, LC, SC, Swan.

Are you CPU bound or network bound?

# Other parallel jobs (eg PVM, CFX, ...)

On Barossa, you have ssh priveleges to any node running your job. If you haven't a job running you don't.

This means that any parallel transport layer built (like pvm) using ssh will work on Barossa. This is not true of the APAC systems, where you have to use MPI. (PVM is available on the SC).

# Self-submitting batch jobs

```
#!/bin/sh
#PBS -l cput=3:0:0 -l mem=128MB -r y
if [ ! -z "$PBS_O_WORKDIR" ]; then
    cd $PBS_O_WORKDIR
fi
if [ -f stop ]; then exit; fi
if [ -f checkpoint.dat ]; then
    a.out restart >>output
else
    a.out init >>output
fi
qsub $PBS_JOBNAME
```

PBS\_JOBNAME name of the script above

# Scratch I/O

- On a cluster, a system wide file system is provided via NFS to access home and short term data directories.
- NFS cannot cope with lots of small read/write requests
- Each node has a local disk, which can be accessed for the duration of the job. Use `scp` to copy data to/from the local disk (`scfscp` on the SC). On Barossa, this scratch directory is called `/scratch/$PBS_JOBID`

# Resource Allocation

- Application to ac3 resource allocation committee every six months. Grants are assessed on merit (a well composed application is a condition). Proposals are generally around 3 pages long.
- Resources are granted in terms of *system units*. 1SU corresponds to roughly 1 hour on a 1GHz processor.

Machine	SU avail per 6 months
Barossa	3.3 million
APAC NF	132,000
Hunter	114,000
Clare	94,000

# Looking at your resource usage

<http://barossa.ac3.edu.au/pbs>

ProjectID	Grant	Used	Remaining
acnoise	400000	109913	290087
agero	30000	39356	-9356
ahmadj	0	63633	-63633
alexg	32000	23248	8752
apitman	1500	0	1500
ayang	0	134171	-134171
...			