PAWSEY supercomputing centre

Advancing Research with Pawsey

Dr. Charlene Yang
Supercomputing Application Specialist

Pawsey Supercomputing Centre

- A joint venture of UWA, Curtin, ECU, Murdoch and CSIRO
- Funded by state, federal governments and five partners

PAWSEY supercomputing centre

Pawsey Services

Compute

- Magnus
- Galaxy
- Zeus
  - Zythos
- NeCTAR

Data

- RDSI
- Pawsey HSM
- Data tools
  - LiveARC
  - hadoop
  - compression

Visualization

- Remote vis nodes
- Immersive/tiled displays
- Image capture
- User interface

Training

Internship

Expertise

PAWSEY supercomputing centre

What can we do for you…

DATA

- Storage
  - RDSI – 6.5PB SAS/SATA, GPFS file system
  - HSM – SGI, 6PB disk + 35PB tape
  - LiveARC/Hadoop …
- Transfer
  - data mover nodes: magnus/galaxy-data.pawsey.org.au
- File systems
  - /home, /group and /scratch
  - Lustre, highly optimized for IO

VISUALIZATION

- Remote vis nodes, workstations
  - iDome, tile displays
  - Avizo, Drishti, Visit, Paraview …

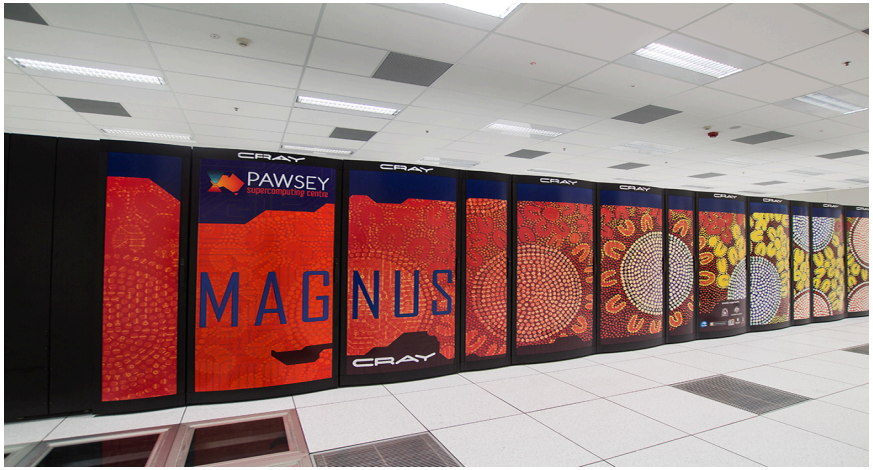
PAWSEY supercomputing centre

What can we do for you…

SUPERCOMPUTING

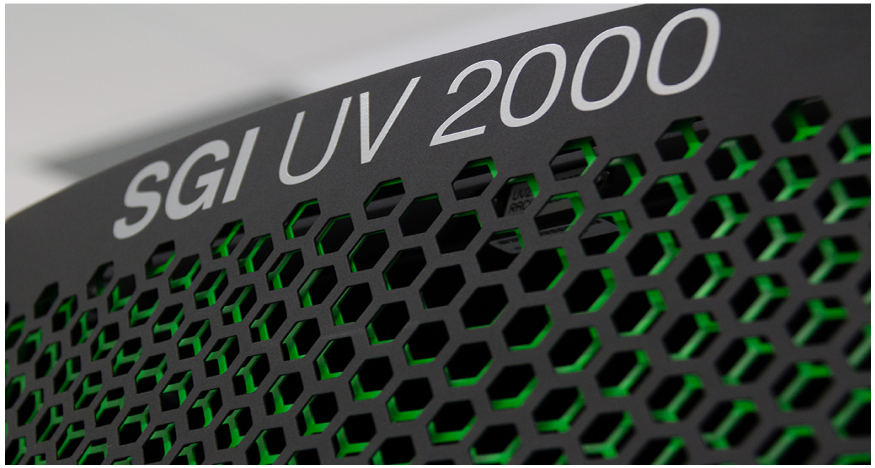
- Magnus
  - Cray XC40, Intel Haswell, 1.1PFLOPS, No.1 in southern hemi
  - general purposed, massively parallel workflow
  - 24 cores per node, 64GB RAM
- Zeus/Zythos
  - SGI, heterogeneous, large memory, GPU, pre/post processing
  - Zeus: Ivy Bridge, 16/20 cores per node, 40+ nodes
    128-512GB RAM, K5000/K20 GPUs
  - Zythos: UV2000, 6TB shared memory, 264 Sandy Bridge cores
    4x K20's, 96 hours walltime
- NeCTAR
  - Research Cloud, pilot studies, long-running jobs, instances up to 16 cores, 64GB RAM, months(!!)

- Magnus



- Zeus

- Zythos



- NeCTAR

PAWSEY supercomputing centre

Reservation for today…

- cyang@**magnus**-1:~> scontrol show reservation

  ReservationName=courseq StartTime=09:00:00 EndTime=17:00:00 Duration=08:00:00

  Nodes=nid000[16-23] **NodeCnt=8** CoreCnt=96 Features=(null) PartitionName=workq Flags=DAILY

  Users=(null) **Accounts=courses01** Licenses=(null) State=ACTIVE

- Login: ssh *couxxx*@magnus.pawsey.org.au
- Mac/Linux – terminal; Window – MobaXterm

PAWSEY supercomputing centre

User Environment

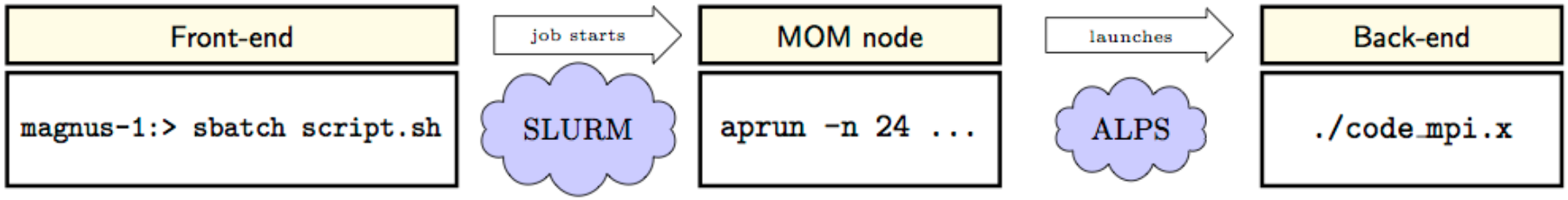
- df
- /home, /group and /scratch

- module list/ module avail/ module help

- PrgEnv-cray, PrgEnv-intel and PrgEnv-gnu

- https://portal.pawsey.org.au/docs/Supercomputers/Magnus/Magnus_User_Guide_New

Scheduling System



- Resource reservation - SLURM
  - SLURM directives e.g. #SBATCH --nodes=2
- Application placement - ALPS
  - aprun options e.g. -n, -N, and -d

- Job submission - SLURM
  - Batch jobs: sbatch *job_script*
  - Interactive jobs: salloc

SLURM directives

Option	Purpose
--account=*account*	Set the account to which the job is to be charged
--nodes=*nnodes*	Request a number of 24-core nodes
--time=*hh:mm:ss*	Set the wall-clock time limit for the job
--job-name=*name*	Set the job name (as it appears under squeue)
--output=*filename*	Set the (standard) output file name Use the token "%j" to include jobid
--error=*filename*	Set the (standard) error file name
--partition=*partition*	Specify a partition (either workq or debugq)
--array=*list*	Specify an array job
--mail-type=*list*	Request e-mail notification for events in *list*. The list may include *BEGIN*, *END*, *FAIL*, or a comma separated combination of valid tokens.
--mail-user=*address*	Specify an e-mail address for notifications

aprun options

Option	Purpose
-n	Set the total number of tasks The default is -n 1
-N	Set the number of tasks per node Should be specified for MPI jobs
-S	Set the number of tasks per NUMA region (or socket)
-d	Set the number of threads per task Coordinate with OMP_NUM_THREADS for OpenMP applications
-j	Set number of task/threads per physical core Default is -j 1 For hyper-threading use -j 2
-cc	Bind tasks to cores
-b	Bypass transfer of executable to compute nodes Not set by default

SLURM commands

Command	Purpose
• **sbatch *[options]* script.sh**	Submit the batch script script.sh
• **scancel *jobid***	Terminate a previously submitted job
• **squeue *[options]***	Show contents of queue
squeue -u user-id	limits the output to userid
squeue -p debugq	limits the output to, e.g., debugq
• **scontrol hold *jobid***	Put job jobid into (user) hold state
• **scontrol release *jobid***	Release job jobid from hold state
• **scontrol show job *jobid***	Show details for a job jobid
• **salloc *[options]***	Request an interactive job

salloc --nodes=1 --time=01:00:00 --account=[your-account]

The command given on the previous line reflects the default request. To submit a request to the debugq partition, use, e.g.,

salloc -p debugq

Example Jobscripts

```
#!/bin/bash -l
#SBATCH --job-name=myjob
#SBATCH --account=courses01
#SBATCH --nodes=2
#SBATCH --time=00:05:00
#SBATCH --export=NONE
#SBATCH --reservation=courseq

#This is run on the service node
hostname

#This is launched onto the compute node
aprun -n 48 -N 24 /bin/hostname
```


Example Jobscripts

MPI: 48 MPI processes across 2 nodes

```
#!/bin/bash -l
#SBATCH --job-name=mpijob
#SBATCH --account=courses01
#SBATCH --nodes=2
#SBATCH --time=00:05:00
#SBATCH --export=NONE

aprun -n 48 -N 24 ./myprogram
```


Example Jobscripts

OpenMP: 1 process with 24 threads

```
#!/bin/bash -l
#SBATCH --job-name=openmpjob
#SBATCH --account=courses01
#SBATCH --nodes=1
#SBATCH --time=00:05:00
#SBATCH --export=NONE

export OMP_NUM_THREADS=24
aprun -n 1 -N 1 -d 24 ./myprogram
```

Example Jobscripts

Hybrid OpenMP/MPI: 2 processes with 24 threads each

```
#!/bin/bash -l
#SBATCH --job-name=hybridjob
#SBATCH --account=courses01
#SBATCH --nodes=2
#SBATCH --time=00:05:00
#SBATCH --export=NONE

export OMP_NUM_THREADS=24
aprun -n 2 -N 1 -d 24 ./myprogram
```

Job Packing

- **Cray environment variable ALPS_APP_PE:**

- #!/bin/bash --login
- #SBATCH --nodes=1
- #SBATCH --time=00:20:00
- #SBATCH --account=[your-project]
- #SBATCH --export=NONE
- module swap PrgEnv-cray PrgEnv-gnu
- module load r
- aprun -n 24 -N 24 ./**r-wrapper.sh**

- #!/bin/bash
- R --no-save "--args $ALPS_APP_PE "
  < **my-script.R**
  > r-$SLURM_JOBID-$ALPS_APP_PE.log

- args <- commandArgs(TRUE)
- print ("This R script instance has input ")
- print (args)

PAWSEY supercomputing centre

Job Packing

- **Using mpibash:**

- #!/bin/bash --login
- #SBATCH --nodes=2
- #SBATCH --time=00:02:00
- #SBATCH --account=[your-project]
- #SBATCH --export=none
- module load mpibash
- aprun -n 48 -N 24 ./**mpi-bash.sh**

- #!/usr/bin/env mpibash
- enable -f mpibash.so mpi_init
- mpi_init
- mpi_comm_rank rank
- mpi_comm_size size
- echo "Hello from bash mpi rank $rank of $size"
- mpi_finalize

Job Packing

- **Using Python and mpi4py:**

- #!/bin/bash --login
- #SBATCH --nodes=2
- #SBATCH --time=00:10:00
- #SBATCH --account=[your-account]
- module load mpi4py
- aprun -n 48 -N 24 python ./**mpi-python.py**

- # If you have an executable python script with a "bang path",
- # make sure the path is of the form #!/usr/bin/env python
- # aprun -n 48 -N 24 ./**mpi-python-x.py**

Common issues

- module: command not found
- Claim exceeds reservation's resources
- illegal instruction
- OOM killer
- JOB CANCELLED
- https://portal.pawsey.org.au/docs/Template:Supercomputers/Running_on_a_Cray

- Documentation: https://portal.pawsey.org.au
- Helpdesk – help@pawsey.org.au

PAWSEY supercomputing centre

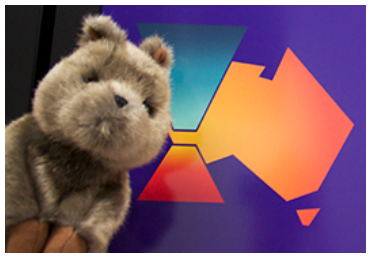












PAWSEY
supercomputing centre

Thank you!

charlene.yang@pawsey.org.au