High Performance Compute Cluster

Overview for Researchers and Users

Copyrights, Licenses and Acknowledgements

Agenda

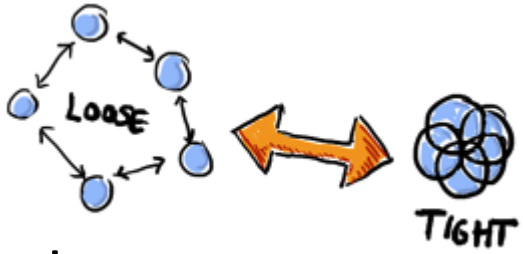
- HPC cluster technology overview
- Getting started
  - Cluster access methods (CLI, graphical, web-portal)
  - Data storage (access, retrieval, quotas, performance)
- Software development environment
  - Using compilers, libraries and MPIs
  - Software application services
- Job scheduler system
  - Principles and benefits
  - Creating job scripts, submitting jobs and getting results
  - Special cases, tuning
- Documentation review
  - Where to get help

HPC Cluster Technology

An overview for Researchers and Users

Beowulf Architecture

- Loosely-coupled Linux Supercomputer
- Efficient for a number of use-cases
  - Embarrassingly parallel / single-threaded jobs
  - SMP / multi-threaded, single-node jobs
  - MPI / parallel multi-node jobs
- Very cost effective HPC solution
  - Commodity X86_64 server hardware and storage
  - Linux based operating system
  - Specialist high-performance interconnect and software



Beowulf Architecture

- Scalable architecture
  - Dedicated management/storage node
    - Logically separate login node instance
  - Multiple compute nodes for different jobs
    - Standard memory jobs (6GB/core)
    - High memory jobs (25GB/core)
    - GPU/visualisation node
  - Multiple storage tiers for user data
    - 1TB local scratch disk on every node
    - 450TB Lustre scratch filesystem
    - 12TB user home-directory space
    - 500TB file-based storage system
    - 1PB object storage system

Storage services

- File-based storage
  - Several different tiers available
    - Single-node scratch disk
    - Shared scratch filesystem
    - User home-dir and tier1 data storage
  - POSIX compatible with wide range of applications
  - POSIX permissions for sharing (e.g. `drwxr-xr-x`)
- Object-based storage
  - Objects are replicated asynchronously
  - S3 protocol with access+secret key access
  - HTTPS URL for public data

Cluster facilities

- Cluster service nodes
  - Onboard storage, dedicated networking
- 2 x cluster login nodes
- High bandwidth QDR Infiniband network
- Separate management LAN
- Compute nodes
  - 6 x 20-core nodes with 256GB RAM
  - 8 x 20-core ndoes with 128GB RAM
  - 2 x 40-core nodes with 1TB RAM
  - 1 x GPU nodes with Nvidia K4200

User facilities

- Modern 64-bit Linux operating system
  - Compatible with a wide range of software
- Pre-installed with tuned HPC applications
  - Compiled for the latest CPU architecture
- Comprehensive software development environment
  - C, C++, Fortran, Java, Ruby, Python, Perl, R
  - Modules environment management

User facilities

- High throughput job-scheduler for increased utilisation
  - Resource request limits to protect running jobs
  - Fair-share for cluster load-balancing
  - Resource reservation with job backfilling
- Multiple user-interfaces supporting all abilities
  - Command-line access
  - Graphical desktop access

HPC Cluster Hardware

An overview for Researchers and Users

System inventory

- All hardware has a unique serial number
- Recorded on asset documents
- Soft-copy available on cluster
  - */opt/service/docs/* directory
- Used to open service request with HP

Service node configuration

- Dual HP DL380 gen9 servers

Compute node configuration

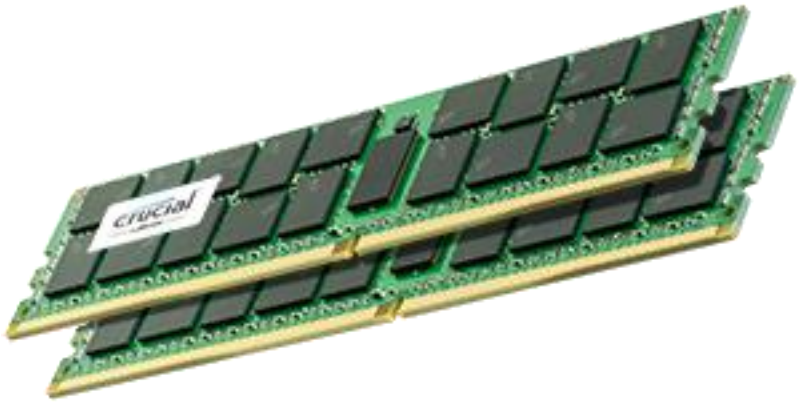
- Compute nodes are HP Apollo 2000 servers

Compute node configuration

- 14 x HP Apollo 2000 compute nodes with:
  - 2 x Intel Xeon *Haswell* E5-2660v3 10-core CPUs per compute node
  - 1 x 1000GB 7.2K RPM disk per compute node
  - Standard memory and high-memory configurations include:
    - 8 x compute nodes with 128GB memory (8 x 16GB DDR4 1866MHz memory)
    - 6 x compute nodes with 256GB memory (16 x 16GB DDR4 1866MHz memory)

Compute node configuration

- 2 x HP DL560 gen9 compute nodes with:
  - 4 x Intel Xeon *Haswell* E5-4660v3 10-core CPUs per compute node
  - 2 x compute nodes each with 1024GB memory (32x32GB DIMMs)
  - 1 x 1000GB 7.2K RPM disk per compute node



Infiniband fabric

- High-performance 40Gbps Infiniband fabric
  - 32Gbps effective bandwidth per link
  - 1.23us latency for small messages (measured)

Storage system configuration

- Lustre storage using HP MSA2040 12Gb SAS arrays
  - Scratch filesystem with 16 x 28TB object storage devices
  - Tier1 filesystem with 12 x 45TB object storage devices
  - RAID6 protected storage with remote backup

Storage system configuration

- Object storage using HP SL4540 storage servers
  - 4 x 265TB object storage servers with 2X replication
  - S3 gateway for user access

Accessing the cluster

An overview for Researchers and Users

Command-line access

- Login via SSH
  - Use SSH client on Linux, UNIX and Mac hosts
  - Use PuTTY, Exceed, OpenSSH on Windows
- Connect to the login node

```
ssh username@kelvin.qub.ac.uk
ssh username@kelvin1.qub.ac.uk
ssh username@kelvin2.qub.ac.uk
```

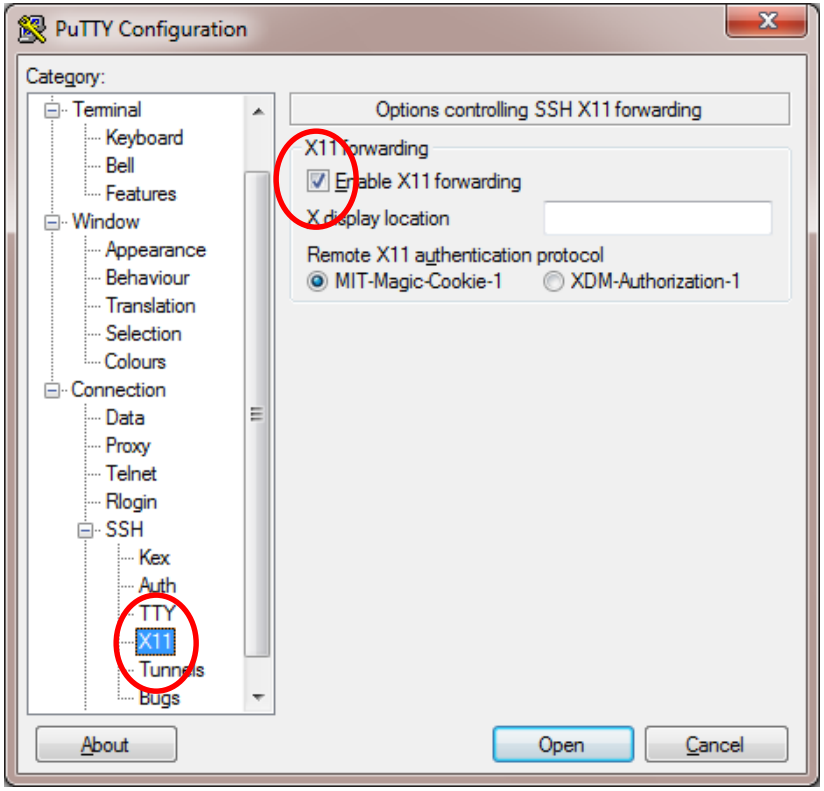
- Use your AD username and password

Admin command-line access

- Commands requiring special permissions require "**`sudo`**" access
  - Only enabled for admin accounts
  - Privileged access to shared filesystems

Graphical Access

- X-forwarding is supported for graphical apps
  - Use the "**`ssh -X`**" command to enable
  - Available via PuTTY



Copying files to the cluster

- Use your login credentials (AD user + password)
- Copy to the login node
- Use client for your operating system
  - From Linux/Mac:

    `scp <file> username@kelvin.qub.ac.uk`
  - From Windows (Putty command-line client):

    `pscp <file> username@kelvin.qub.ac.uk`
  - Windows GUI

    https://winscp.net/

Storing data objects

- Users each have a private storage area for objects
- Access is provided via the S3 object storage protocol
  - Users have an access-key and a secret-key
  - Access details are available from the cluster home-dir
    - `~/.s3cfg` file
- Shared account credentials are also available for teams of users to collaborate

Storing data objects from the cluster

- Use the s3cmd command
- Available on login and compute nodes
- Uses the .s3cfg config file in home-dir
- Objects are stored in named buckets
- Example:
  - Create a new bucket
  - Upload a new object and download again
  - Remove objects and buckets

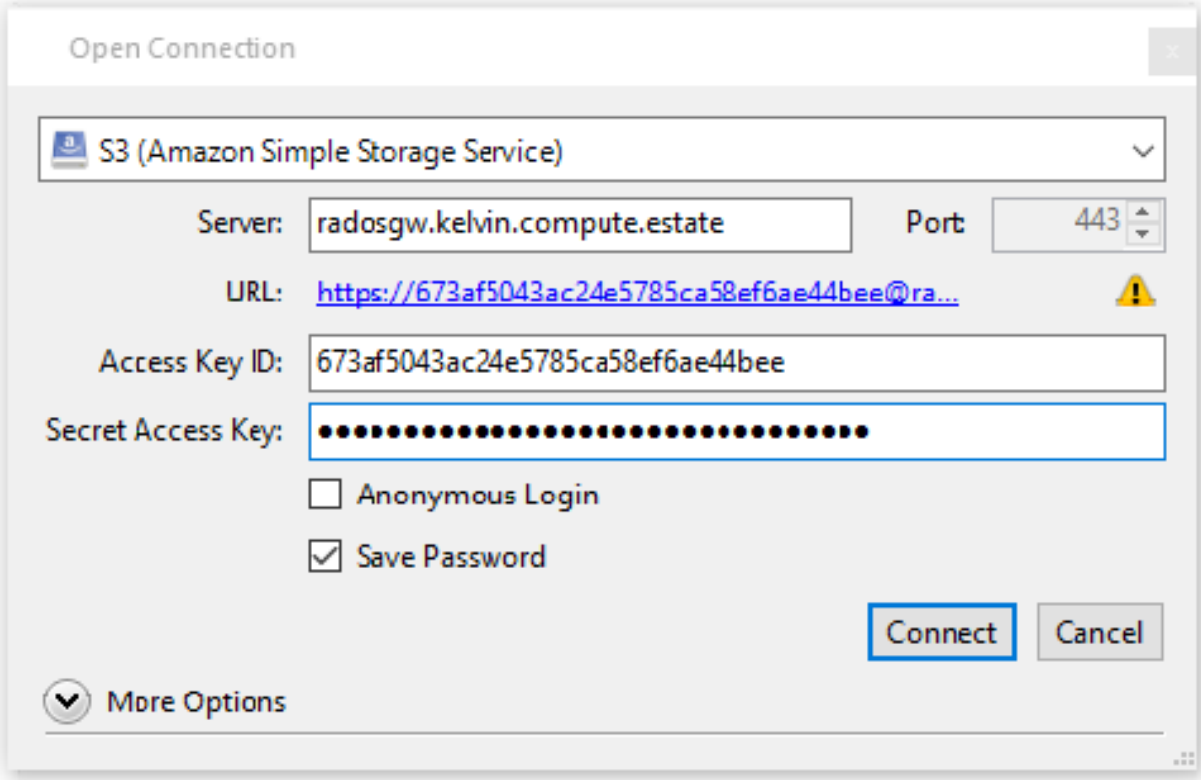
Storing data objects from the cluster

```
[alces-cluster@login1(kelvin) ~]$ s3cmd mb s3://testset123
Bucket 's3://testset123/' created

[alces-cluster@login1(kelvin) ~]$ s3cmd put /mnt/scratch/users/alces-
cluster/mydatafile s3://testset123/object123
/mnt/scratch/users/alces-cluster/mydatafile ->
           s3://testset123/object123  [1 of 1]
 10 of 10   100% in    0s   126.82 B/s  done

[alces-cluster@login1(kelvin) ~]$ s3cmd get s3://testset123/object123 newfile
s3://testset123/object123 -> newfile  [1 of 1]
10 of 10   100% in    0s    24.15 B/s  done

[alces-cluster@login1(kelvin) ~]$ s3cmd rm s3://testset123/object123
File s3://testset123/object123 deleted

[alces-cluster@login1(kelvin) ~]$ s3cmd rb s3://testset123
Bucket 's3://testset123/' removed
```

S3 storage access

- Other s3cmd options:
  - Recursively get/put (-r)
  - Enable (-e) / disable (-d) encryption
    - Requires a passphrase to be set in config file
    - Stores files in encrypted format
  - Make object pubic (-P)
    - N.B. remember to use **HTTPS://** when sharing URLs
- GUI access also available
  - *https://**cyberduck**.io*

Cyberduck S3 GUI



HPC Cluster Software

An overview for Researchers and Users

Operating system

- Unified 64-bit Linux installation
  - RedHat Enterprise Linux 6.6 64-bit
  - Automatic Transparent HugePage support
- Centralised software deployment system
  - Headnode manages cluster software repositories
  - Compute and login nodes deployed from masters
    - Personality-less software simplifies management
    - Automatic software deployment for system regeneration
    - Centralised software patching and update system

Application software

- Supporting software provided by RHEL distro
  - Compilers, interpreters, drivers
- Specific software repositories also included
  - Fedora/EPEL: RedHat compatible libraries and tools
  - Open Grid Scheduler: Job scheduler software
  - Alces Software: Symphony HPC Toolkit
- Application software

Cluster data storage

- Multiple filesystems provided for users
  - **`/opt/apps`** software repository:
    - Location for site-installed software packages
    - Contains shared, user-facing software
    - Read-only for users
  - **`/opt/gridware`** software repository:
    - Location for Alces-installed software packages
    - Managed by Alces Gridware packager
    - Contains shared, user-facing software
    - Read-only for users

Cluster data storage

- Multiple filesystems provided for users
  - Home-directories (**`/users`**) – 12TB
    - Default place that users login to
    - 50GB/100K file quota
    - Contains help files and links to available scratch areas
  - Parallel filesystem (**`/mnt/scratch`**) – 450TB
    - Large, shared storage area
    - High performance and expandable
    - Orphan data is automatically deleted
  - Local scratch (**`/tmp`**) – 880GB/node
    - Local scratch disk on nodes = fastest available storage
    - Orphan data is automatically deleted

Cluster data storage

- Multiple filesystems provided for users
  - Tier1 data storage (**`/mnt/tier1`**) – 500TB
    - Results data, available from cluster login nodes
    - Group quotas enabled
    - Backed up to secondary site
  - Object storage – 2PB raw / 1PB replicated
    - Available at URL:
      `s3://radosgw.kelvin.compute.estate`
    - Login with your access and secret key
    - Automatically replicated to secondary site
    - Object URL in the format:

      `s3://<bucket-name>.radosgw.kelvin.compute.estate/<object-name>`

Cluster data storage summary

Storage type	Location	Access	Persistence	Write Performance
Application store	`/opt/apps` `/opt/gridware`	Read-only	Shared, Permanent	N/A
Object store	`S3://radosgw.kelvin.compute.estate`	Read-write	Shared, Permanent	Medium
Tier-1	`/mnt/tier1` `(login nodes only)`	Group quotas	Shared, Permanent	Medium
Home directory	`~` `(/users/<username>)`	Quota enabled	Shared, Permanent	Medium
Shared scratch	`/mnt/scratch`	Full	Shared, auto-deletion	Fast
Local scratch	`/tmp`	Full	Local to node, auto-deletion	Fastest

Efficient space usage

- Data stored in scratch areas may be removed
  - Compute node disks are persistent during jobs only
  - Old data automatically removed from scratch areas
  - Data stored in /mnt/scratch is not subject to quota
- Consider compressing data stored in home-dir
  - Standard Linux utilities installed on login nodes
    - gzip, bzip2, zip/unzip, TAR

File permissions and sharing

- All files and directories have default permissions
  - Contents of your home directory are private

```
[alces-cluster@login1(cluster) users]$ ls -ald $HOME
drwx------ 17 alces-cluster users 24576 Feb 18 17:57 /users/alces-cluster
```

  - To give colleagues access to your files, use the *chmod* command:

```
[alces-cluster@login1(cluster) users]$ chmod g+rx $HOME
[alces-cluster@login1(cluster) users]$ ls -ald $HOME
drwxr-x--- 17 alces-cluster users 24576 Feb 18 17:57 /users/alces-cluster
```

- Scratch directories are more open by default

Software Development Environment

- *modules* environment management
  - The primary method for users to access software
  - Enables central, shared software library
  - Provides separation between software packages
    - Support multiple incompatible versions
    - Automatic dependency analysis and module loading
  - Available for all users to
    - Use centralised application repository
    - Build their own applications and modules in home dirs
    - Ignore modules and setup user account manually

Modules environment management

- Loading a module does the following:
  - Puts binaries in your $PATH for your current session
  - Puts libraries in your $LD_LIBRARY_PATH
  - Enables manual pages and help files
  - Sets variables to allow you to find and use packages

```
[alces-cluster@login1(cluster) ~]$ module load apps/breakdancer
apps/breakdancer/1.3.5.1/gcc-4.4.6+samtools-0.1.18+boost-1.51.0
 |
 OK
[alces-cluster@login1(cluster) ~]$ echo $BREAKDANCERDIR
/opt/gridware/pkg/apps/breakdancer/1.3.5.1/gcc-4.4.6+samtools-0.1.18+boost-1.51.0
[alces-cluster@login1(cluster) ~]$ echo $BREAKDANCERBIN
/opt/gridware/pkg/apps/breakdancer/1.3.5.1/gcc-4.4.6+samtools-0.1.18+boost-1.51.0/bin
[alces-cluster@login1(cluster) ~]$ which breakdancer-max
/opt/gridware/pkg/apps/breakdancer/1.3.5.1/gcc-4.4.6+samtools-0.1.18+boost-
1.51.0/bin/breakdancer-max
```

Modules environment management

- Using the *module avail* command

```
[alces-cluster@login1(cluster) ~]$ module avail
--------------------------- /opt/gridware/etc/modules -------------------------------
apps/bowtie/1.0.0/gcc-4.4.7
apps/bowtie2/2.1.0/gcc-4.4.7
apps/cmake/2.8.10.2/gcc-4.4.7
apps/cpanminus/1.5017/noarch
apps/cufflinks/2.1.1/gcc-4.4.7+boost-1.49.0+samtools-0.1.19+eigen-3.0.5
apps/gbrowse/2.55/gcc-4.4.7+perl-5.18.0
apps/gromacs/4.6.5/gcc-4.4.7+openmpi-1.6.5+fftw3_float-3.3.3+fftw3_double-3.3.3
apps/gromacs_double/4.6.5/gcc-4.4.7+openmpi-1.6.5+fftw3_double-3.3.3
apps/gromacs_float/4.6.5/gcc-4.4.7+openmpi-1.6.5+fftw3_float-3.3.3
apps/iprscan/5.3.46.0/bin
apps/muscle/3.8.31/gcc-4.4.7
apps/ncbiblast/2.2.29/gcc-4.4.7
apps/paml/4.7a/gcc-4.4.7
apps/perl/5.18.0/gcc-4.4.7
apps/psipred/3.4/gcc-4.4.7
apps/python/2.7.5/gcc-4.4.7
apps/samtools/0.1.19/gcc-4.4.7
apps/satsuma/3.0/gcc-4.4.7
apps/tophat/2.0.10/gcc-4.4.7+samtools-0.1.19+boost-1.49.0
```

Modules environment management

- Other modules commands available
  - `# module unload <module>`
    - Removes the module from the current environment
  - `# module list`
    - Shows currently loaded modules
  - `# module display <module>`
    `# module whatis <module>`
    - Shows information about the software
  - `# module keyword <search term>`
    - Searches for the supplied term in the available modules *whatis* entries

Modules environment management

- By default, modules are loaded for your session
- Loading modules automatically (all sessions):
  - `# module initadd <module>`
    - Loads the named module every time a user logs in
  - `# module initlist`
    - Shows which modules are automatically loaded at login
  - `# module initrm <module>`
    - Stops a module being loaded on login
  - `# module use <new module dir>`
    - Allows users to supply their own modules branches

Modules environment management

- Using modules
  - Shared applications are installed with a module
    - New requests are assessed for suitability to be a shared application
  - Available from /opt/gridware NFS share
    - No applications installed on compute nodes
    - Same application set available throughout the cluster
  - Modules always available to all users
    - Can be loaded on the command line
    - Can be included to load automatically at login
    - Can be included in scheduler job scripts

HPC Applications

- Many popular applications available
- Centrally hosted to avoid replication
- Multiple versions supported via modules
- Interactive applications also supported
  - Please run on compute nodes via qrsh
  - Users should not run applications on login node
- Open-source and commercial apps supported
  - Commercial application support provided by license provider

Installing new applications

- Site-installed applications:
  - Install new applications in /opt/apps
  - Create module files as required
  - Shared filesystem available on all compute nodes
  - Users can also install their own packages in home dirs
- Alces-installed applications:
  - Make application requests via your site admin
  - Commercial applications require appropriate licenses
    - Site license may be required
    - License server available on cluster service nodes

Cluster job scheduler

- Open Grid Scheduler
  - Developed from Sun Grid Engine codebase
    - Virtually identical to original syntax
  - Open-source with wide range of contributors
  - Free to download and use
    - Please consider acknowledging copyrights
  - Very stable and established application
  - Many command-line utilities and tools
    - Admin GUI also available but rarely used

Cluster job scheduler

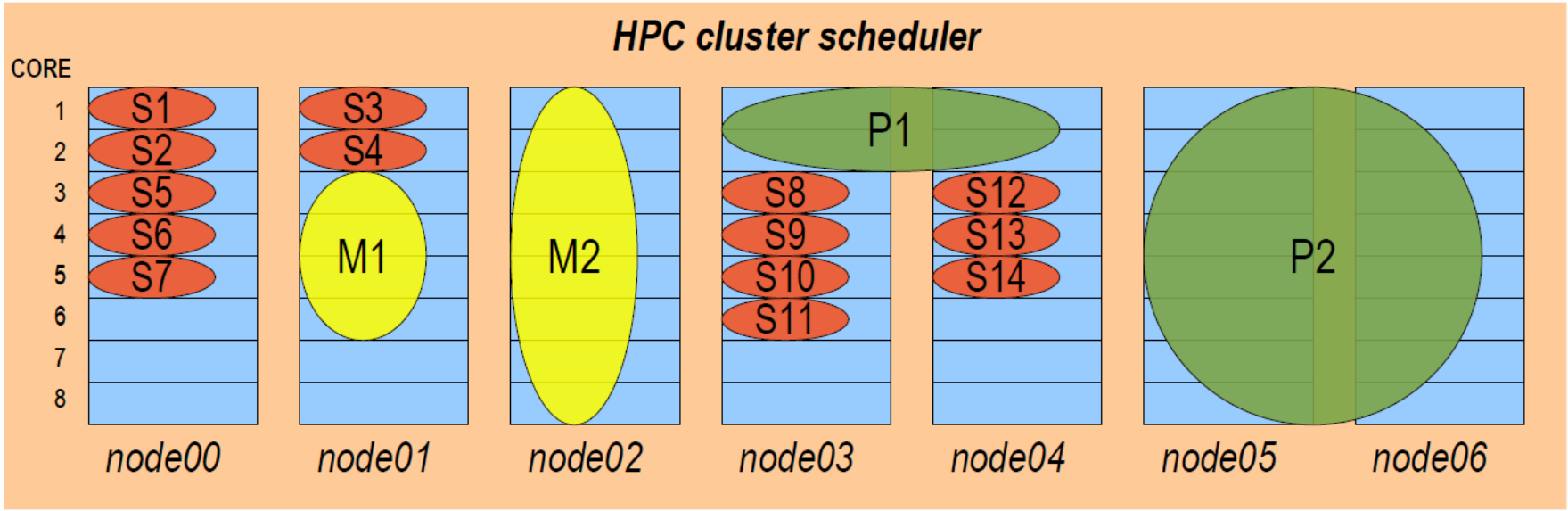
- Why do we need a job scheduler?
  - Need to allocate compute resources to users
  - Need to prevent user applications from overloading compute nodes
  - Want to queue work to run overnight and out of hours
  - Want to ensure that users each get a fair share of the available resources

Types of job

S — Serial jobs (single core, single node)

M — Multi-threaded jobs (many cores, single node)

P — Parallel jobs (many cores, many nodes)



Types of job

- Serial jobs
  - Use a single CPU core
  - May be submitted as a task array (many single jobs)
- Multi-threaded jobs
  - Use two or more CPU cores on the same node
  - User must request the number of cores required
- Parallel jobs
  - Use two or more CPU cores on one or more nodes
  - User must request the number of cores required
  - User may optionally request a number of nodes to use

Class of job

- Interactive jobs
  - Started with the "`qrsh`" command
    - Will start immediately if resources are available
    - Will exit immediately if there are no resources available
  - May be serial, multi-threaded or parallel type
  - Users must request a maximum runtime
  - Users should specify resources required to run
  - Input data is the shell session or application started
  - Output data is shown on screen

Interactive Jobs

- "*qrsh*" allows users to run interactive applications directly on compute node
- Do not run demanding applications on login nodes
  - Per-user CPU and memory limits are in place to discourage users from running on login nodes
  - Login node CPU time is not included in scheduler accounting
- Users are prevented from logging in directly to compute nodes; use qrsh for access

```
[alces-cluster@login1(cluster) ~]$ qrsh

[alces-cluster@node12(cluster) ~]$ uptime
 17:01:07 up 7 days,  5:12,  1 user,  load average: 0.00, 0.00, 0.00
[alces-cluster@node12(cluster) ~]$
```

Non-interactive jobs

- Jobs are submitted via a *job-script*
  - Contains the commands to run your job
  - Can contain instructions for the job-scheduler
- Submission of a simple job-script
  - Echo commands to stdout
  - Single-core batch job
  - Runs with default resource requests
  - Assigned a unique job number (e.g. 2573)
  - Output sent to home directory by default

Simple batch job

- Job-script:

  ~/simple_jobscript.sh

```
#!/bin/bash

echo "Starting new job"
sleep 120
echo "Finished my job”
```

```
[alces-cluster@login1(cluster) myjob]$ qsub simple_jobscript.sh
Your job 2573 ("simple_jobscript.sh") has been submitted

[alces-cluster@login1(cluster) myjob]$ qstat
job-ID  prior   name       user         state  queue           slots ja-task-ID
-----------------------------------------------------------------------------
  2573  2.02734 simple_job alces-cluster r     byslot.q@node21   1

[alces-cluster@login1(cluster) myjob]$ cat ~/simple_jobscript.sh.o2573
Starting new job
Finished my job

[alces-cluster@login1(cluster) myjob]$
```

Viewing the queue status

- Use the *qstat* command to view queue status:

```
[alces-cluster@login1(cluster) myjob]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots
-----------------------------------------------------------------------------------------------------
   2575 2.06725 imb-any.sh alces-cluste r     02/14/2014 17:15:38 byslot.q@node08.cluster.local     32
   2576 2.06725 imb-any.sh alces-cluste r     02/14/2014 17:15:43 byslot.q@node10.cluster.local     32
   2577 2.15234 imb-any.sh alces-cluste r     02/14/2014 17:15:43 byslot.q@node15.cluster.local     48
   2578 1.90234 simple_job alces-cluste r     02/14/2014 17:15:58 byslot.q@node11.cluster.local      1
   2579 0.00000 simple_job alces-cluste qw    02/14/2014 17:16:25                                    1
```

- Running jobs (**`r`**) are shown with queues used
- Queuing jobs (**`qw`**) are waiting to run
  - Use "**`qstat -j <jobid>`**" for more information on why queuing jobs have not started yet

Removing Jobs from the queue

- Use the *qdel* command to remove a job:

```
[alces-cluster@login1(cluster) myjob]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots
-----------------------------------------------------------------------------------------------------
   2577 1.90234 imb-any.sh alces-cluste r    02/14/2014 17:15:43 byslot.q@node15.cluster.local     48
   2579 1.27734 simple_job alces-cluste qw   02/14/2014 17:16:25                                    1

[alces-cluster@login1(cluster) myjob]$ qdel 2577
alces-cluster has registered the job 2577 for deletion

[alces-cluster@login1(cluster) myjob]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots
-----------------------------------------------------------------------------------------------------
2579 1.27734 simple_job alces-cluste qw    02/14/2014 17:16:25                                    1
```

- Nodes are automatically cleared up
- Users can delete their own jobs only
  - Operator can delete any jobs

Job-scheduler instructions

- qsub and qrsh can receive instructions from users
- Common instructions include:
  - Provide a name for your job
  - Control how output files are written
  - Request email notification of job status
  - Request additional resources
    - More CPU cores
    - More memory
    - A longer run-time
    - Access to special purpose compute nodes

Job-scheduler instructions

- Job-scheduler instructions may be given as parameters to your *qsub* or *qrsh* command
  - Use “-N <name>” to set a job name
  - Job name is visible via a *qstat* command

```
[alces-cluster@login1(cluster) ~]$ qrsh -N hello
[alces-cluster@node06(cluster) ~]$
[alces-cluster@node06(cluster) ~]$ qstat

job-ID  prior   name       user         state submit/start at     queue                slots
-----------------------------------------------------------------------------------------
   2580 1.90234 hello      alces-cluste r     02/24/2014 17:29:41 byslot.q@node06      1
```

Job-scheduler instructions

- Non-interactive jobs can provide instructions as part of their job-script
  - Begin instructions with “#$” identifier
  - Multiple lines can be processed

```
[alces-cluster@login1(cluster) myjob]$ cat simple_jobscript.sh
#!/bin/bash
#$ -N my_job

echo "Starting new job"
sleep 120
echo "Finished my job"

[alces-cluster@login1(cluster) myjob]$ qsub simple_jobscript.sh
Your job 2581 ("my_job") has been submitted
[alces-cluster@login1(cluster) myjob]$ qstat
job-ID  prior   name       user         state submit/start at     queue                 slots
-----------------------------------------------------------------------------------------
   2581 1.40234 my_job     alces-cluste    r   02/14/2014 17:36:26 byslot.q@node03          1
```

Setting output file location

- Provide a location to store the output of a job
  - Use the "**-o <filename>**" option
  - Supply full path and filename
  - The variable **$JOB_ID** is set automatically to your job ID number

```
[alces-cluster@login1(cluster) myjob]$ cat simple_jobscript.sh
#!/bin/bash
#$ -o ~/outputfiles/myjob.$JOB_ID

echo "Starting new job"
sleep 120
echo "Finished my job"
```

Requesting email notification

- Provide an email address for job status
  - Use the "**-M <email address>**" option
  - Specify conditions to report using "**-m <b|e|a>**"
    - Send email when job **B**egins, **E**nds or **A**borts

```
[alces-cluster@login1(cluster) myjob]$ cat simple_jobscript.sh
#!/bin/bash
#$ -o ~/outputfiles/myjob.$JOB_ID
#$ -M user@work.com -m bea

echo "Starting new job"
sleep 120
echo "Finished my job"
```

Running task arrays

- Used to run an array of serial jobs
  - Input and output files can be separated
  - Individual tasks are independent; may run anywhere
  - Use the "**-t <start>-<end>**" option

```
[alces-cluster@login1(cluster) myjob]$ cat simple_taskarray.sh
#!/bin/bash
#$ -t 1-100 -cwd -o ~/results/taskjob.$JOB_ID

module load apps/fastapp

Fastapp -i ~/inputdata/intput.$SGE_TASK_ID -o \
~/results/out.$SGE_TASK_ID
```

Requesting additional resources

- If no additional resources are requested, jobs are automatically assigned default resources
  - One CPU core
  - Up to 6GB RAM
  - Max of 72-hour runtime
- These limits are automatically enforced
- Jobs must request different limits if required

Requesting more CPU cores

- Multi-threaded and parallel jobs require users to request the number of CPU cores needed
- The scheduler uses a *Parallel Environment (PE)*
  - Must be requested by name
  - Number of slots (CPU cores) must be requested
  - Enables scheduler to prepare nodes for the job
- Three PE available on cluster
  - smp / smp-verbose
  - mpislots / mpislots-verbose
  - mpinodes / mpinodes-verbose

smp Parallel Environment

- Designed for SMP / multi-threaded jobs
  - Job must be contained within a single node
  - Requested with number of slots / CPU-cores
  - Verbose variant shows setup information in job output
  - Memory limit is requested per slot
- Request with "`-pe smp-verbose <slots>`"

smp Parallel Environment

```
[alces-cluster@login1(cluster) smptest]$ cat runsmp.sh
#!/bin/bash
#$ -pe smp-verbose 7  -o ~/smptest/results/smptest.out.$JOB_ID
~/smptest/hello

[alces-cluster@login1(cluster) smptest]$ cat ~/smptest/results/smptest.out.2583
==========================================================
SGE job submitted on Mon Feb 14 14:01:31 GMT 2014
JOB ID: 2583
JOB NAME: runsmp.sh
PE: smp-verbose
QUEUE: byslot.q
MASTER node17.cluster.local
==========================================================
2: Hello World!
5: Hello World!
6: Hello World!
1: Hello World!
4: Hello World!
0: Hello World!
3: Hello World!
==========================================================
[alces-cluster@login1(cluster) smptest]$
```

mpislots Parallel Environment

- Designed for multi-core MPI jobs
  - Job may use slots on many nodes
  - MPI hostfile is generated by scheduler
  - Requested with number of slots / CPU-cores
  - Verbose variant shows setup information in job output
  - Memory limit is requested per slot
    - Default is 6GB per slot (CPU core)
- Request with "`-pe mpislots <slots>`"

mpislots Parallel Environment

```
[alces-cluster@login1(cluster) ~]$ cat mpijob.sh
#!/bin/bash
#$ -pe mpislots-verbose 48 -cwd -N imb-job -o ~/imb/results/imb-job.$JOB_ID -V
module load apps/imb
mpirun IMB-MPI1

[alces-cluster@login1(cluster) ~]$ cat ~/imb/results/imb-job.2584
============================================================
SGE job submitted on Mon Feb 14 14:11:31 GMT 2014
3 hosts used
JOB ID: 2584
JOB NAME: imb-job
PE: mpislots-verbose
QUEUE: byslot.q
MASTER node01.cluster.local
Nodes used:
node01 node02 node03
============================================================
** A machine file has been written to /tmp/sge.machines.2584 on
node01.cluster.local **
============================================================
If an output file was specified on job submission Job Output Follows:
============================================================
```

mpinodes Parallel Environment

- Designed for MPI jobs that use whole nodes
  - Job has a number of entire nodes dedicated to it
  - MPI hostfile is generated by scheduler
  - Requested with number of nodes (20-cores each)
  - Verbose variant shows setup information in job output
  - Memory limit is requested per slot
    - Default is 120GB per slot (complete node)
- Request with "**`-pe mpinodes <slots>`**"

mpinodes Parallel Environment

```
[alces-cluster@login1(cluster) ~]$ cat mpinodesjob.sh
#!/bin/bash
#$ -pe mpinodes-verbose 3 -cwd -N imb-job -o ~/imb/results/imb-job.$JOB_ID -V
module load apps/imb
mpirun IMB-MPI1

[alces-cluster@login1(cluster) ~]$ cat ~/imb/results/imb-job.2585
============================================================
SGE job submitted on Mon Feb 14 15:20:41 GMT 2014
3 hosts used
JOB ID: 2585
JOB NAME: imb-job
PE: mpinodes-verbose
QUEUE: bynode.q
MASTER node03.cluster.local
Nodes used:
node03 node01 node02
============================================================
** A machine file has been written to /tmp/sge.machines.2585 on
node03.cluster.local **
============================================================
If an output file was specified on job submission Job Output Follows:
============================================================
```

Requesting more memory

- Default memory request is 6GB per core
- Request more using "**`-l h_vmem=<amount>`**"

```
[alces@login1(cluster) ~]$ qsub -l h_vmem=256G jobscript.sh
```

- Jobs requesting more memory are likely to queue for longer
  - Need to wait for resources to be available to run
- Jobs requesting less memory are likely to run sooner
  - Scheduler will use all available nodes and memory
- Memory limits are enforced by the scheduler
  - Jobs exceeding their limit will be automatically stopped

How much memory should I request?

- Run the job with a large memory request
- When complete, use "**`qacct -j <jobid>`**"

```
[alces@login1(cluster) ~]$ qsub -l h_vmem=64G simple_jobscript.sh

[alces@login1(cluster) ~]$ qacct -j 2586
==============================================================
qname        byslot.q
hostname     node13.cluster.local
group        users
owner        alces-cluster
project      default.prj
department   alces.ul
jobname      my_job
jobnumber    2586
……
cpu          0.003
maxvmem      204.180M
```

Requesting a longer runtime

- Default runtime is 72-hours
- Time is measured for slot occupancy when job starts
  - Request more using "**`-l h_rt=<hours:mins:secs>`**"

```
[alces@login1(cluster) ~]$ qsub -l h_rt=04:30:00 jobscript.sh
```

- This is a maximum runtime
  - Your job will be accounted for if you finish early
- Job time limits are enforced by the scheduler
  - Jobs exceeding their limit will be automatically stopped

Why not always request large runtimes?

- The scheduler performs backfilling
  - When resources are being reserved for a large job, the scheduler will automatically run short jobs on idle CPU cores
  - The scheduler needs to know how long your job may run for to allow it to jump the queue
- Users queuing large jobs can request reservation
  - Use the "`-R y`" option to request that resources are reserved for large parallel jobs
  - Reserved resources are included in your usage

Requesting high-memory nodes

- Use of 1TB nodes requires a special request
- Request access using "**`-l himem=true`**"

```
[alces@login1(cluster) ~]$ qsub -l himem=true \
   -l h_vmem=1000G jobscript.sh
```

- Jobs will be scheduled to run on high-mem nodes only
- Remember to request more memory per slot too
- Memory limits are enforced by the scheduler
  - Jobs exceeding their limit will be automatically stopped

Requesting node exclusivity

- Ensures that your job has a complete node
- Access is restricted to particular users and groups
  - Request exclusivity with **"-l exclusive=true"**
  - Your job may queue for significantly longer

```
[alces-cluster@login1(cluster) myjob]$ cat simple_jobscript.sh
#!/bin/bash
#$ -N excjob -h h_rt=2:0:0
#$ -l exclusive=true -l h_vmem=2G

module load apps/veryBusyApp
VeryBusyApp -all
```

Sharing resources

- Job priority affects queuing jobs only
  - Functional shares
    - Static definitions of how much resource a user/group can use
  - Urgency
    - The amount of time a job has been waiting to run
    - Priority of queuing jobs automatically increases over time
  - Fair-share policy
    - Takes past usage of the cluster into account
    - Half-life of 14 days for resource usage
    - Ensures that cluster is not unfairly monopolised
- Use "`qstat`" to show the priority of your job

Sharing resources

- Resource quotas
  - Static limits set by administrators
  - Allows the maximum resource available to a user or group to be controlled
  - Default quotas include:
    - Access to exclusivity flag
    - Limit the maximum number of CPU slots
- Use "**`qquota`**" to show your resource quotas

Queue administration

- Administrator can perform queue management
  - Delete jobs (*qdel*)
  - Enable and disable queues (*qmod*)
    - Disabled queues finish running jobs but do not start more

```
[admin@headnode1(cluster) ~]$ qstat -u \*

job-ID  prior   name       user         state submit/start at     queue                          slots
-----------------------------------------------------------------------------------------------------
   2587 2.03927 imb-any.sh alces-cluste r     02/20/2014 12:11:43 byslot.q@node20.cluster.local     24
   2588 2.03927 imb-any.sh alces-cluste r     02/21/2014 11:11:21 byslot.q@node19.cluster.local     24
   2589 2.03927 imb-any.sh alces-cluste r     02/05/2014 12:41:44 byslot.q@node05.cluster.local     24
   2590 2.15234 imb-any.sh alces-cluste r     02/08/2014 08:21:51 byslot.q@node12.cluster.local     43
   2591 1.10234 my_job     alces-cluste r     02/14/2014 02:42:06 byslot.q@node11.cluster.local      1

[admin@headnode1(cluster) ~]$ qdel 2587
admin has registered the job 2587 for deletion

[admin@headnode1(cluster) ~]$ qmod -d byslot.q@node05
admin@service.cluster.local changed state of "byslot.q@node05.cluster.local" (disabled)

[admin@headnode1(cluster) ~]$
```

Execution host status

- Use the "**qhost**" command to view host status

```
[alces-cluster@login1(cluster) ~]$ qhost
HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
global                  -               -     -       -       -       -       -
node01                  linux-x64      16  0.00   62.9G    1.1G   16.0G     0.0
node02                  linux-x64      16  0.00   62.9G    1.2G   16.0G     0.0
node03                  linux-x64      16  0.00   62.9G    1.2G   16.0G     0.0
node04                  linux-x64      16  1.59   62.9G    4.2G   16.0G     0.0
node05                  linux-x64      16  3.03   62.9G    1.2G   16.0G     0.0
node06                  linux-x64      16  4.01   62.9G    1.2G   16.0G     0.0
node07                  linux-x64      16 16.00   62.9G    8.2G   16.0G     0.0
node08                  linux-x64      16 16.00   62.9G    5.2G   16.0G     0.0
node09                  linux-x64      32 32.00   62.9G   56.2G   16.0G     0.0
node10                  linux-x64      32 32.00   62.9G   56.2G   16.0G     0.0
node11                  linux-x64      32 33.10   62.9G   56.7G   16.0G     0.0
node12                  linux-x64      32 32.94   62.9G   32.2G   16.0G     4.0
node13                  linux-x64      64  0.00   62.9G    1.5G   16.0G     0.0
node14                  linux-x64      64  0.00   62.9G    1.5G   16.0G     0.0
node15                  linux-x64      64 12.40  252.9G   31.2G   16.0G     0.0
node16                  linux-x64      64 64.00  252.9G  250.1G   16.0G     2.0
```

Visualisation support

- Authorized users can use the ***qdesktop*** command
  - Creates an interactive graphical desktop session
  - Server-side 3D rendering available via VirtualGL
- Requires setup and connection via VPN
- Session URL and one-time password in file:

  `~/alcesdesktop.<jobid>`
- Run graphical applications:

  `vglrun <`*`application name`*`>`

  `e.g. vglrun glxgears`
- End your job using "**`qdel`**" or logout of desktop

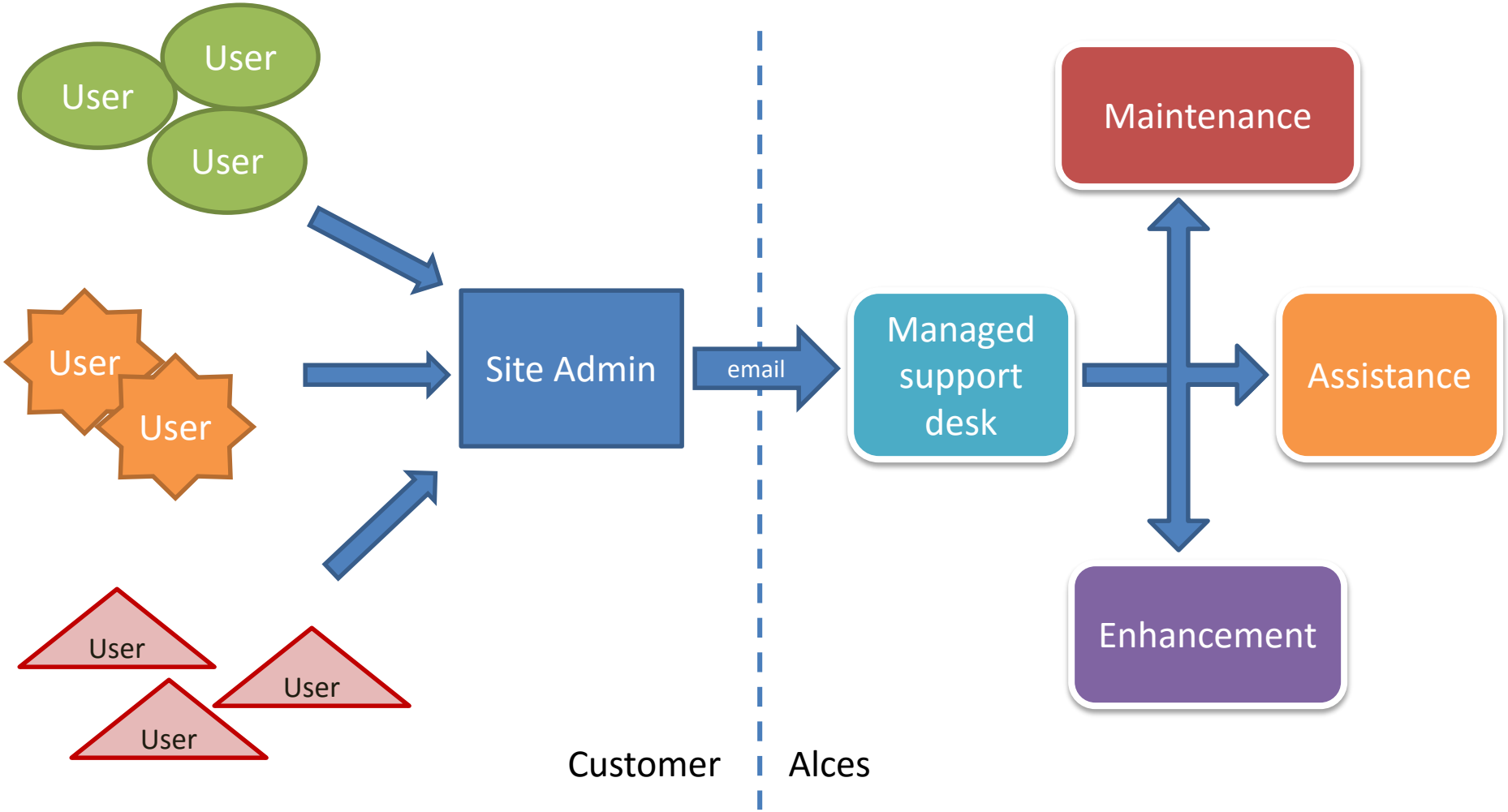
Requesting Support

An administrator overview

Support services

- Site administrator
  - On-site point of contact for all users
    - Service delivery lead for site
  - Organises, priorities and referees user requests
    - Determines usage policy, time sharing, etc.
  - Responsible for user data management
    - Ability to organise data files stored on the cluster
    - Can set user quotas, change file ownerships and groups
  - Requests vendor service via support tickets
    - Changes to the service must be authorized by site admin
    - Delegated responsibility during absence to another admin

Support process



Remote managed HPC service

- System maintenance tasks
  - *"It used to work and now it doesn't"*
  - General system maintenance
    - Generally non-intrusive to user jobs
    - System monitoring, reporting and tuning
    - Redundant hardware to enhance availability
  - Scheduled preventative maintenance
    - Generally requires a pre-arranged outage of HPC service
    - Patching, firmware upgrades, software updates
    - Nominally 1 day of scheduled maintenance every calendar quarter
  - Emergency maintenance (priority 1 requests)
    - Restoration of service after serious failure
    - User jobs may need to be resubmitted after outage

Remote managed HPC service

- HPC usage assistance
  - *"Something isn't working the way I expected"*
  - Primary contact for users is the site admin
  - Further assistance provided by remote service
  - Example user queries include:
    - My job runs more slowly than it did last week
    - How do I write a job-script for my application?
    - My MPI job is using the wrong interconnect
    - Performance of my job is not scaling as expected
    - Where should I store my files?
    - Why is my job still queuing to run?

Remote managed HPC service

- Enhancement requests
  - *"I need help to do a new thing"*
  - Changes to the existing system
  - May require scheduled maintenance period
  - Example enhancement requests may include:
    - Alternative job queue configuration
    - New storage system pool or resource
    - New or upgraded system software
    - New or upgraded application software

Requesting assistance

- Support request email sent by site admin
  - Must contain a description of the problem
  - Username or group with the problem
  - Example job output files that show the issue
  - Node number or system component with a fault
  - Method for replicating the problem
  - Request priority and business case justification
  - support@alces-software.com

Visiting site

- Site engineer visits
  - Hot-swap components typically ship direct to site
  - HP engineers may need to visit site for fixes
  - 4-hour response for critical components
    - Headnodes, service nodes, switches, storage
  - Next-business-day attendance for compute nodes
  - Scheduled by Alces for normal service
  - May be scheduled by customer in an emergency

Service review meetings

- Regular account meeting
  - Review maintenance requests for cluster
    - Plan preventative maintenance sessions
    - Root-cause analysis for unscheduled outages
  - Review assistance requests from users
    - Identify common requests and feedback to documentation
    - Determine future training requirements
  - Review enhancement requests
    - Consider cluster utilisation and update policies as required
    - Prioritise outstanding requests for new software packages
    - Rationalise installed software base

Managing Users

An administrator overview

User authentication

- Users must be authorized for access to cluster
  - Site admin can request user accounts
  - Group information and contact detail recorded
  - New User application form template
- Passwords managed by site AD service
  - Only one password to manage for cluster access
  - Single-sign-on for cluster nodes
    - Queued jobs will not prompt for password when launched
  - Object storage access keys

Privileged user accounts

- Site admin has privileged user account
  - Provided for user administration tasks
  - Full access to manipulate user data
  - Configure user priorities, quotas and permissions

Site admin tasks

- Point of contact for HPC users
- User data manipulation
- Storage quota management
- Final authorization for service changes
- Arbitration for resource conflicts
- Point of contact for regular service reviews
- Request reporting to managed service support

Next Steps

- Login via SSH
- View available software packages
- Create initial job script
- Submit test jobs