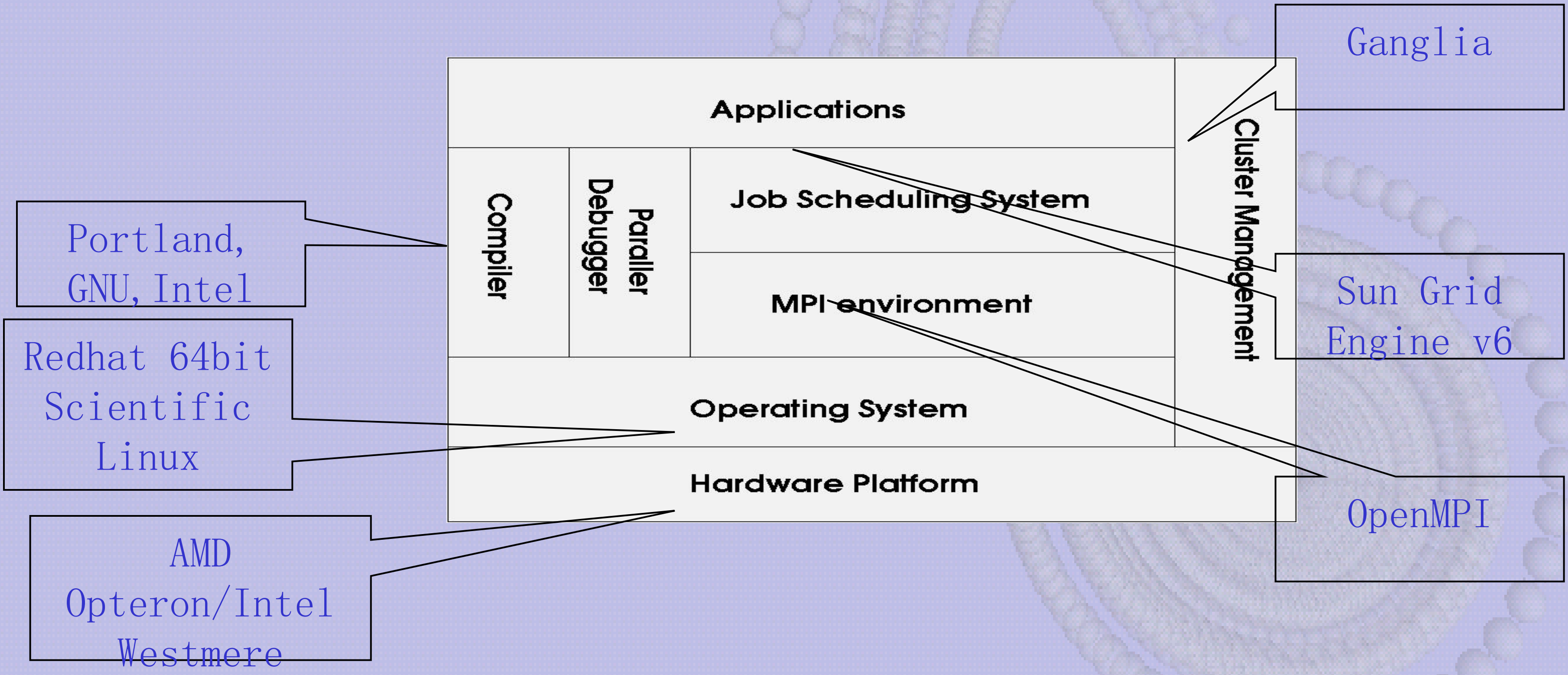
Software stack on the worker nodes



Applications

Compiler

Paraller Debugger

Job Scheduling System

MPI environment

Operating System

Hardware Platform

Cluster Management

Ganglia

Sun Grid Engine v6

OpenMPI

Portland, GNU, Intel

Redhat 64bit Scientific Linux

AMD Opteron/Intel Westmere

Keeping up-to-date with application packages

From time to time the application packages get updated. When this happens;

- news articles inform us of the changes via iceberg news. To read the news- type news on iceberg or check the URL: http://www.wrgrid.group.shef.ac.uk/icebergdocs/news.dat
- The previous versions or the new test versions of software are normally accessed via the version number. For example;

abaqus69 , nagexample23 , matlab2011a

Running application packages in batch queues

icberg have locally provided commands for running some of the popular applications in batch mode. These are;

runfluent , runansys , runmatlab , runabaqus

To find out more just type the name of the command on its own while on iceberg.

Setting up your software development environment

- Excepting the scratch areas on worker nodes, the view of the filestore is identical on every worker.
- You can setup your software environment for each job by means of the **module** commands.
- All the available software environments can be listed by using the **module avail** command.
- Having discovered what software is available on iceberg, you can then select the ones you wish to use by using-
  module add or **module load** commands
- You can load as many non-clashing modules as you need by consecutive module add commands.
- You can use the **module list** command to check the list of currently loaded modules.

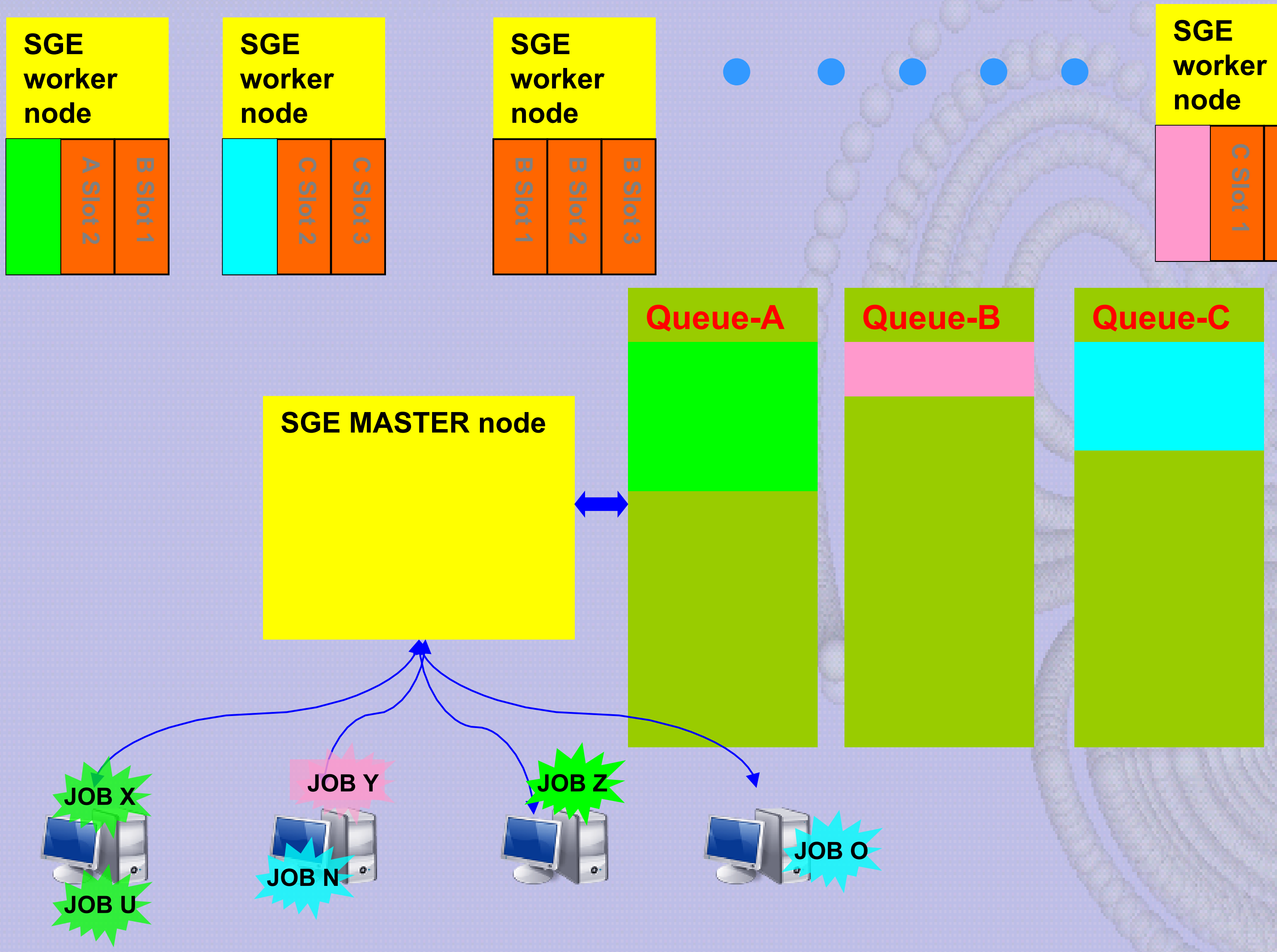
Software development environment

- Compilers
  - PGI compilers
  - Intel Compilers
  - GNU compilers
- Libraries
  - NAG Fortran libraries (MK22 , MK23)
  - NAG C libraries (MK 8)
- Parallel programming related
  - OpenMP
  - MPI (OpenMPI , mpich2 , mvapich)

Managing Your Jobs
Sun Grid Engine Overview

SGE is the resource management system, job scheduling and batch control system. (*Others available such as PBS, Torque/Maui, Platform LSF)*

- Starts up interactive jobs on available workers
- Schedules all batch orientated ‘i.e. non-interactive’ jobs
- Attempts to create a fair-share environment
- Optimizes resource utilization



- Queues
- Policies
- Priorities
- Share/Tickets
- Resources
- Users/Projects

Job scheduling on the cluster

Submitting your job

There are two SGE commands submitting jobs;

- **qsh** or **qrsh** : To start an interactive job
- **qsub** : To submit a batch job

There are also a list of home produced commands for submitting some of the popular applications to the batch system. They all make use of the qsub command.

These are;

runfluent , runansys , runmatlab , runabaqus

Managing Jobs
monitoring and controlling your jobs

- There are a number of commands for querying and modifying the status of a job running or waiting to run. These are;
  - **qstat** or **Qstat** (query job status)
  - **qdel** (delete a job)

  - **qmon** (a GUI interface for SGE)

Running Jobs
Example: Submitting a serial batch job

Use editor to create a job script in a file (e.g. example.sh):

```
          #!/bin/bash
          # Scalar benchmark
          echo ‘This code is running on’     `hostname`
          date
   ./linpack
```

Submit the job:

```
qsub example.sh
```

Running Jobs
qsub and qsh options

-l h_rt=hh:mm:ss	The wall clock time. This parameter must be specified, failure to include this parameter will result in the error message: “Error: no suitable queues”. ***Current default is 8 hours .***
-l arch=intel* -l arch=amd*	Force SGE to select either Intel or AMD architecture nodes. ***No need to use this parameter unless the code has processor dependency***.
-l mem=memory	sets the virtual-memory limit e.g. –l mem=10G (for parallel jobs this is per processor and not total). ***Current default if not specified is 6 GB*** .
-l rmem=memory	Sets the limit of real-memory required ***Current default is 2 GB.*** ***Note: rmem parameter must always be less than mem.***
-help	Prints a list of options
-pe ompigige np -pe openmpi-ib np -pe openmp np	Specifies the parallel environment to be used. **np** is the number of processors required for the parallel job.

Running Jobs
qsub and qsh options (continued)

-N jobname	By default a job's name is constructed from the job-script-file-name and the job-id that is allocated to the job by SGE. This options defines the jobname. ***Make sure it is unique because the job output files are constructed from the jobname*.**
-o output_file	Output is directed to a named file. ***Make sure not to overwrite important files by accident.***
-j y	Join the standard output and standard error output streams ***recommended***
-m [bea] -M email-address	Sends emails about the progress of the job to the specified email address. If used, both –m and –M must be specified. Select any or all of the ***b,e*** and ***a*** to imply emailing when the job ***begins***, ***ends*** or ***aborts***.
-P project_name	Runs a job using the specified projects allocation of resources.
-S shell	Use the specified shell to interpret the script rather than the default bash shell. Use with care. A better option is to specify the shell in the first line of the job script. E.g. ***#!/bin/bash***
-V	Export all environment variables currently in effect to the job.

Running Jobs
batch job example

qsub example:

```
qsub –l h_rt=10:00:00 –o myoutputfile –j y   myjob
```

OR alternatively … the first few lines of the submit script myjob contains -

```
$!/bin/bash
$# -l h_rt=10:00:00
$# -o myoutputfile
$# -j y
```

and you simply type; **qsub myjob**

Running Jobs
Interactive jobs

qsh , qrsh

- These two commands, find a free worker-node and start an interactive session for you on that worker-node.
- This ensures good response as the worker node will be dedicated to your job.
- The only difference between qsh and qrsh is that ;
  qsh starts a session in a new command window where as qrsh uses the existing window.
  Therefore, if your terminal connection does not support graphics (i.e. XWindows) than qrsh will continue to work where as qsh will fail to start.

Running Jobs
A note on interactive jobs

- Software that requires intensive computing should be run on the worker nodes and not the head node.
- You should run compute intensive interactive jobs on the worker nodes by using the qsh or qrsh command.
- Maximum (and also default) time limit for interactive jobs is 8 hours.