pawsey

Supercomputing User Training

Module 7: Accounting Model Overview

Pawsey Training Series

Supercomputing User Training

1. Supercomputing Introduction
2. Logging In
3. Filesystems Overview
4. Moving Data In and Out
5. Using Software Modules
6. Using Software Containers
7. Accounting Model Overview
8. Job Scheduling Overview
9. Running Jobs
10. Testing Job Runs
11. Managing Project Data

Outcomes for this Module

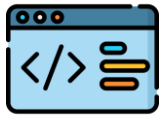
- Describe what an accounting model and a compute allocation are
- Describe the accounting model that is implemented at Pawsey
- Monitor your compute allocation

✓ Prerequisite knowledge:

- ✓ **Bash shell basics**
- ✓ **User Training 02: Logging In**

Watch for These Signs!

Definition of new concepts



Hands-on coding (demo)

Best practices

Exercises and solutions

Warnings (bad practices)

Links to user documentation

Icons made by Freepik from www.flaticon.com

Pawsey

Australian Government

NCRIS National Research Infrastructure for Australia

CSIRO

Curtin University

Murdoch University

Government of Western Australia

ECU Edith Cowan University

The University of Western Australia

Accounting Model and Compute Allocations at Pawsey

Supercomputers are Shared Infrastructure

- Supercomputers are expensive, get replaced every few years and consume electricity whether used or not
- Hence, we want them running at maximum capacity including night and weekends

To manage such shared infrastructure, we make use of:

- The Accounting Model: to define and measure compute resources utilised by user base
- Compute Allocations: to partition total available compute resources across users over each calendar year
- The Job Scheduler: to orchestrate workflows received by user base

NOTE: discussion of merit allocation schemes is out of scope for this module

**** More information on the job scheduler in User Training 08: Job Scheduling Overview***

What is an Accounting Model?

Accounting Model
A method to define and measure the usage of resources at a compute infrastructure.

Service Unit (SU)
The unit of measure for consumable compute resources.

- Typically, a SU relates to the *hourly usage of CPU cores* of a supercomputer
- At Pawsey, 1 SU is equivalent to 1 hour use of 1 physical CPU core.

- Cost of a job (SU): number of CPU cores requested (CPU) × wall time (h)
- Total SUs available in a year: total available CPU cores (CPU) x hours in a year (h)

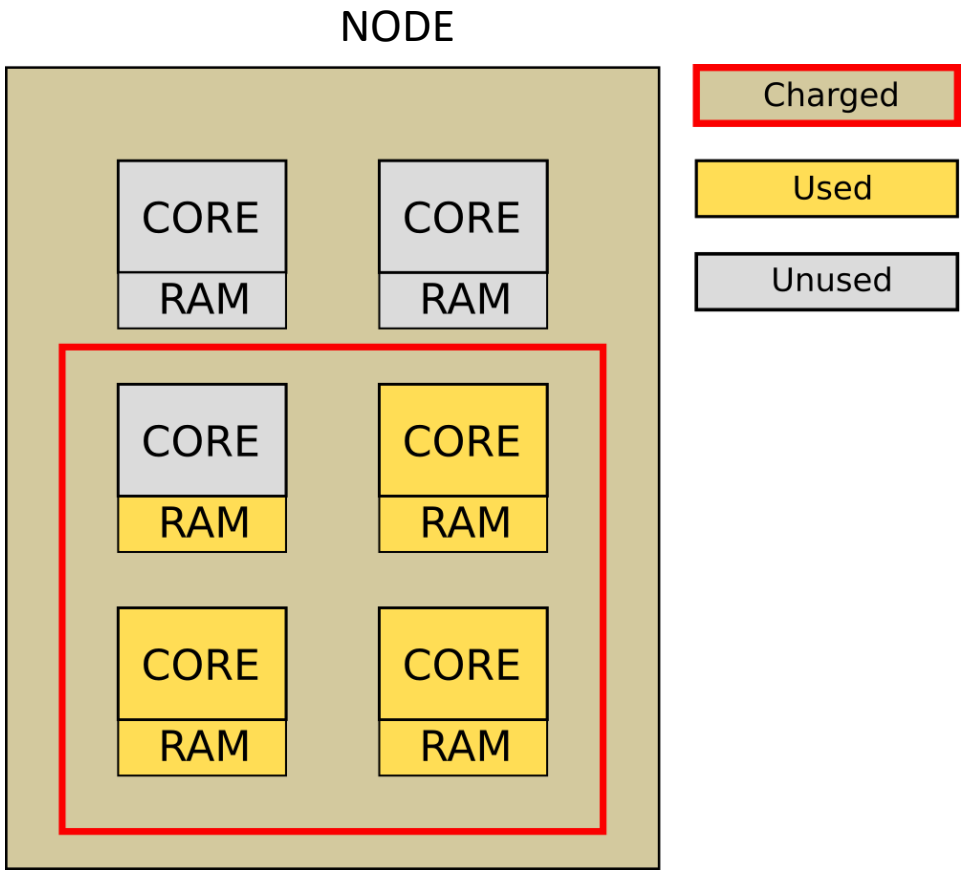
Examples

1 SU = 1 hour use of 1 CPU core
= ½ hour use of 2 CPU cores

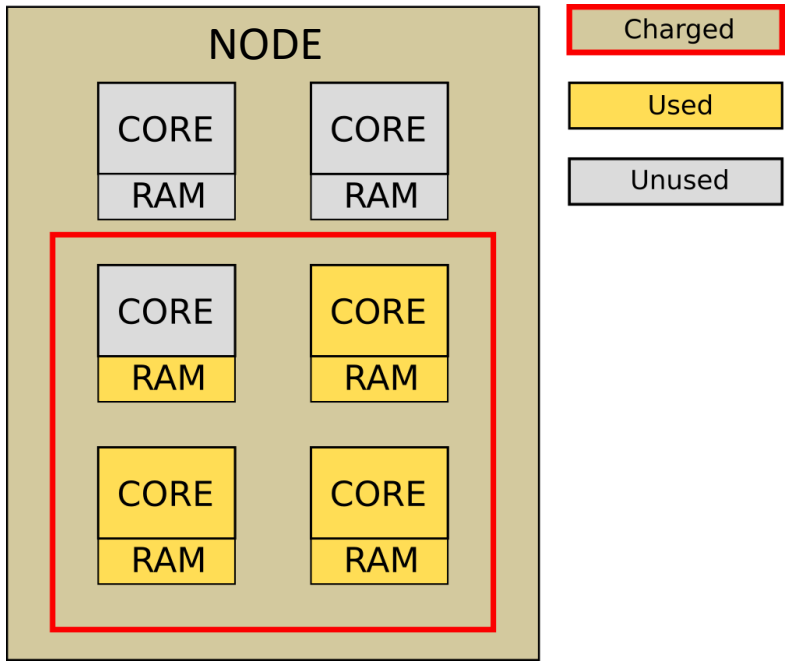
576 SU = 24 hours on 1 Magnus node (24 CPU cores)
= 4.5 hours on 1 Setonix node (128 CPU cores)

Accounting Model on Setonix Phase 1

- Consumable resources in each CPU compute node
  - 128 CPU cores
  - Total Random Access Memory (RAM) memory (varies by type of node)
- Proportional allocation of fraction of a node is allowed
- 1 Service Unit
  - 1 hour use of 1 CPU core
  - 1 hour use of 1/128th of the total node RAM
  - A job is charged based on the largest fraction of the two consumables
- Accounting (cost) of a job (SU)
  - Largest fraction × nodes × wall time
  - Minimum: 1 SU per hour per node
  - Maximum: 128 SU per hour per node
  - NOTE: if a job finishes before its scheduled end time, it is accounted for the actual (shorter) usage, not for the scheduled (longer) one

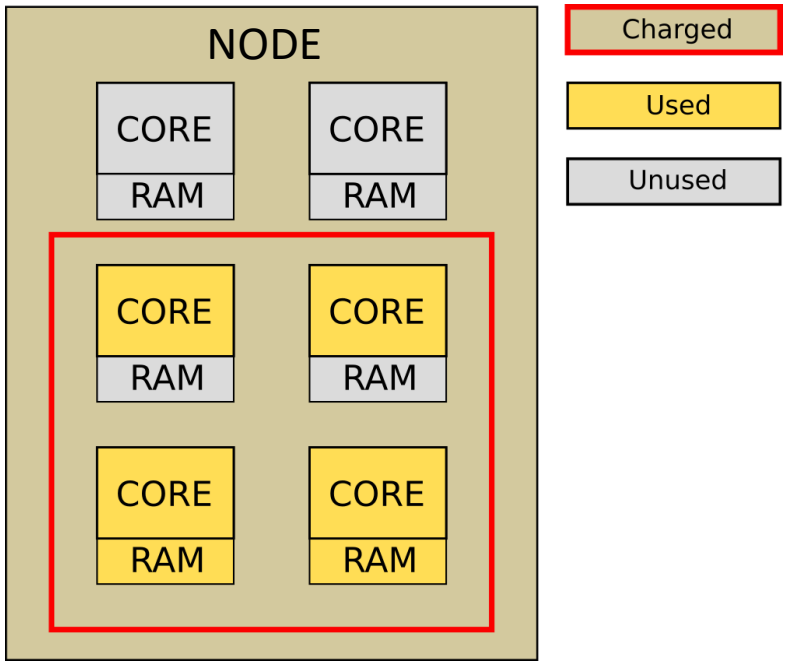


Examples of Resource Accounting on Setonix



Example 1

- CPU cores proportion = 1/2
- RAM proportion = 2/3
- Resource Charging = 2/3



Example 2

- CPU cores proportion = 2/3
- RAM proportion = 1/3
- Resource Charging = 2/3

Examples of Estimating Service Units on Setonix

- Running for one day on 8 nodes
  - 24 hours × 8 nodes × 128 cores/node = 24,576 Service Units
- Running for 10 hours on 32 cores within a node, shared access
  - 10 hours × 32 cores = 320 Service Units
- Running for 10 hours on 32 cores within a node, exclusive access
  - 10 hours × 128 cores = 1,280 Service Units
- Running for 6 hours on 4 cores within a node, requesting 115 GB of RAM (i.e. half the node total)
  - 4 cores / 128 cores = 0.03125
  - 115 GB / 230 GB = 0.5 → largest fraction
  - 6 hours × 0.5 × 128 cores = 384 Service Units

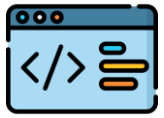
Compute Allocations at Pawsey

Compute Allocation
Amount of Service Units awarded to a project that was successful in a merit allocation call.

- The total service units available on Setonix are partitioned across the successful projects
- Each project has their own compute allocation
- Compute allocations are evenly divided in quarters along the calendar year
- Resources accounting is setup around the principle that all projects should get their fair share of usage, relative to the awarded allocation (via the job scheduler, see later)
- Projects are encouraged to proactively monitor their allocation through each quarter, to optimise usage

More details @
- Allocation Results and Administration

Monitoring Use of Your Project Allocation



Demo on Setonix – let's do this together

- Allocation usage can be checked from the command line:
  - Use the command `pawseyAccountBalance`
    - Option to select specific project: `-p project-id`
    - Option for breakdown of each user: `-u`

Proactively monitor your project allocation
Aim for a full usage of your quarterly allocations, without significant under- or over-utilisations.

More details @
- Job Scheduling # Project accounting

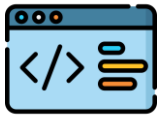
OUTPUTS: Monitoring Use of Your Project Allocation

```
$ pawseyAccountBalance
Compute Information
-------------------
          Project ID     Allocation          Usage     % used
          ----------     ----------          -----     ------
          pawsey0001          25000          19595       78.4
$ pawseyAccountBalance -p pawsey0012
Compute Information
-------------------
          Project ID     Allocation          Usage     % used
          ----------     ----------          -----     ------
          pawsey0012         575000         157605       27.4
```

```
$ pawseyAccountBalance -p pawsey0012 -u
Compute Information
-------------------
            Project ID     Allocation          Usage     % used
            ----------     ----------          -----     ------
            pawsey0012         575000         157605       27.4
            --espinosa                        128725       22.4
          --mdelapierre                        15541        2.7
              --pelahi                          8883        1.5
..
```

Summary

- Terms we learnt
  - Accounting Model
  - Service Unit
  - Compute Allocation



- Tasks we learnt
  - Monitor compute allocation: `pawseyAccountBalance`

- Proactively monitor your project allocation

Pawsey

Australian Government

NCRIS National Research Infrastructure for Australia

CSIRO

Curtin University

Murdoch University

Government of Western Australia

ECU Edith Cowan University

The University of Western Australia

Getting Help

Getting Help

https://support.pawsey.org.au

Pawsey has extensive User Support Documentation.

Areas covered include:

- System user guides
- Knowledge Base
- Pawsey-supported software list
- Maintenance logs
- Policies and terms of use

For further assistance, contact the help desk, via User Support Portal.

Help us to help you by providing details, such as:

- Which resource
- Error messages
- Location of files
- SLURM job id
- Your username if having login issues
- Never tell us (or anyone) your password!

Become a Pawsey Friend and receive our Newsletter:
https://pawsey.org.au/pawsey-friends/

Q & A Session