EuroCC@Greece

HPC for the Greek Health & Life Sciences Sector

Introduction to Workload Management

Eleni Kanellou
Institute of Computer Science
Foundation of Research and Technology, Hellas
kanellou@ics.forth.gr

EURO
Greece

FORTH
INSTITUTE OF COMPUTER SCIENCE

What is workload management?

◎ Workload management
  ○ Coordinate the distribution of jobs to resources
  ○ Queue tasks
  ○ Data management in a heterogeneous computing environment
  ○ Access to data files
  ○ Etc.

◎ Resource management

No clear distinction: Several enterprise-level systems implement several of these aspects.

  ○ Access and manage resources, be they computational, memory, storage, or other

  ○ book-keep allocation of jobs to resources

◎ Scheduler
  ○ Evaluate job needs and assign jobs to suitable nodes, taking into consideration aspects such as duration, priorities, etc.,
  ○ enforce duration limits
  ○ Determine which job to run next on some resource

Some basic concepts
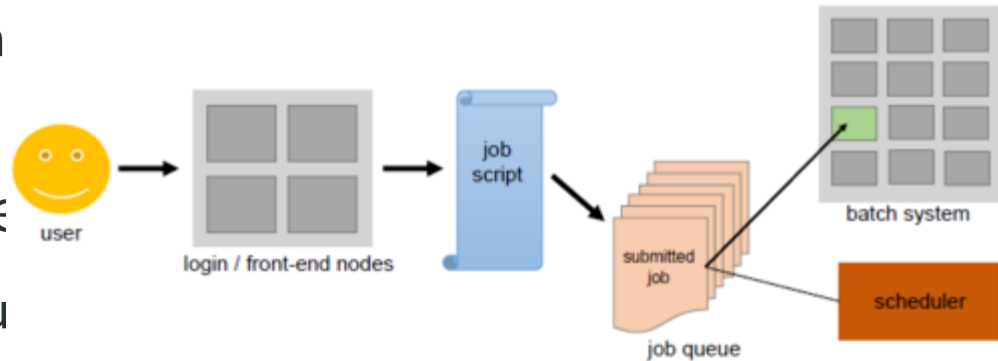
◎ Core: smallest computing unit (CPU, GPU, ...)

◎ Socket: A processor i.e. multiple cores, sharing the same memory

◎ Node: one or more sockets

◎ Task: a unit of computation

◎ Job: a set of tasks

◎ Workload: a set of jobs

  ◎ Sequential

  ◎ Parallelizable

◎ Queue: Incoming jobs are ordered in queues until they can be processed

Some scheduling policies

◎ FIFO: First come, first serve

◎ Smallest job first

◎ Higher priority job first

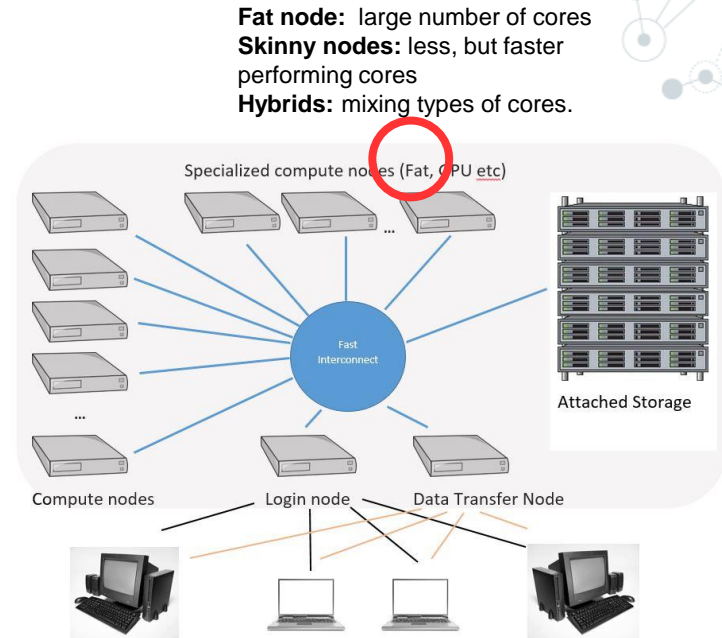◎ Round robin: alternate scheduling jobs out of a set of queues

◎ Fullest queue first

The concept of batch processing

◎ Jobs submitted in sets or "batches"

  – Identical tasks ca
    setup times

◎ Job executed whe

  – Idle time of resou

user

login / front-end nodes

job script

submitted job

job queue

batch system

scheduler

So what is an HPC cluster?

◎ Collection of nodes

– Login nodes

– Data transfer nodes

– Computation nodes

◎ Fast interconnect

◎ Large storage

Fat node: large number of cores
Skinny nodes: less, but faster performing cores
Hybrids: mixing types of cores.

Specialized compute nodes (Fat, GPU etc)

Fast Interconnect

Attached Storage

Compute nodes   Login node   Data Transfer Node

Why manage workload/resources when using HPC?

◎ "At home": Intuitive and manual resource allocation, workload management, and planning!

> ➢ _Interactivity_ is key!

◎ In HPC cluster: not possible.
- ○ Limited access
- ○ Resources not exposed to simple user
- ○ Resources coveted by several users concurrently!

◎ Automation necessary.
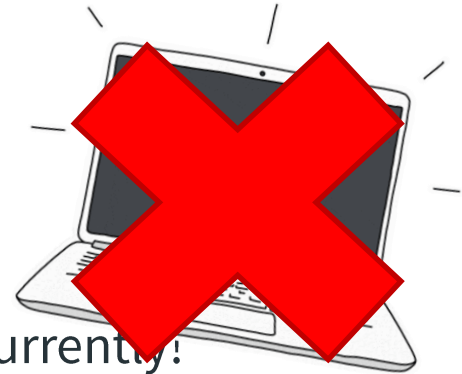
Know when to request fat or skinny nodes:

– Do I run simulations for longer than weeks?

– Skinny more suited to sequential workloads, e.g. BLAST run

– Is my computation composed of many small tasks?

– Fat more suitable for highly parallel workloads, e.g. finite element simulation

– Do I need large storage space for my output?

So why be aware of workload management issues?

Know when not to use it

Know what duration class to request

– Do I am to visualize output?

– Scheduler might "kick you out" of a resource if constraints are violated

– Do my tasks require interaction?

– Am I still designing my code?

So why be aware of workload management issues?

◎ Know how to divide your workload.

- Data parallel workload?

- Task parallel workload?

- Dependencies between tasks?

◎ Know which parts of a job to "speed up".

◎ Take highest advantage possible out of resources.

Thank you!

Any questions?

You can find me at:

kanellou@ics.forth.gr