



HPC Scheduling and Resource Management

This paper provides an overview of scheduling and resource management considerations in implementing a High Performance Compute cluster from Dell. It also lists and describes the benefits of a range of commercial and open source scheduling and resource management tools currently available.

INTRODUCTION TO HPC CLUSTER RESOURCE MANAGERS AND SCHEDULERS	3
SO YOU'RE GETTING A CLUSTER! CONSIDERATIONS FOR RESOURCE MANAGERS AND SCHEDULERS	3
WORKGROUP CLUSTER VS. MULTI-DEPARTMENTAL CLUSTER.....	5
HETEROGENEOUS VS. HOMOGENEOUS RESOURCES	6
HETEROGENEOUS VS. HOMOGENEOUS USERS	7
HOW TO MANAGE ACCESS TO AND CONTROL OF CLUSTER RESOURCES	8
OVERVIEW OF DISTRIBUTED RESOURCE MANAGERS AND SCHEDULER PACKAGES	8
STANDARDS: DRMAA	10
I WANT IT ALL, AND I WANT IT NOW – SCHEDULER POLICIES.....	10
HOW WILL USERS ACCESS IT?	12
WHAT ABOUT THE CLOUD?	13
OVERVIEW OF CLOUD COMPUTING	13
DRM & SCHEDULER PRODUCT MATRIX.....	14
CONCLUSION	14
ABOUT X-ISS	15

Introduction to HPC Cluster Resource Managers and Schedulers

So You're Getting a Cluster! Considerations for Resource Managers and Schedulers

So you're getting a shiny new High Performance Compute (HPC) cluster. Now what? You've been doing your research and want to know how to best provide access to the system for your HPC users. Whether you have a small group of HPC users or dozens throughout multiple departments, selecting the right software to handle job submissions can make or break the success of your HPC implementation.

In order to select the best system for your cluster, it is important to understand how systems work and interact with the various components of your HPC cluster.

A Quick Overview of HPC Cluster Components

Resource Manager

Commonly referred to as a DRM (Distributed Resource Manager) or DRMS (Distributed Resource Management System).

A software tool that:

- Matches available compute resources to compute demand.
- Provides an administrative interface to compute resources.
- Provides a common user interface to compute resources.
- Responsible for tracking and maintaining the state of all resources under its control:
 - Compute node availability – allows for taking nodes offline for maintenance.
 - Resource availability – such as CPU, memory, disk, architecture, and license information.
- Accepts jobs and places them in a queue.
- Communicates with an agent running on the compute nodes to obtain status and control.

Scheduler

In HPC terms, a scheduler is a program responsible for assigning jobs and tasks to resources according to predetermined policies and resource availability. A job is comprised of one or more tasks along

with other information (such as license requirements, required resources, etc.) that is used by the scheduler. Jobs are submitted to a queue mechanism for proper batch processing and optimization of resource utilization. There may be one or more queues, each with policies controlling priorities, permissions, and access to available resources.

The scheduler communicates with the resource manager to obtain information about queues, loads on compute nodes, and resource availability to make scheduling decisions.

Queue

A process that maintains the current state of jobs submitted but not yet completed.

Some queues can have priority and the ability to pre-empt other queues.

Compute Resources

Considered to be the muscle of the cluster, it may be one or more servers, workstations, or SMP/NUMA machines.

Runs a DRM agent that accepts jobs and communicates with the management machine.

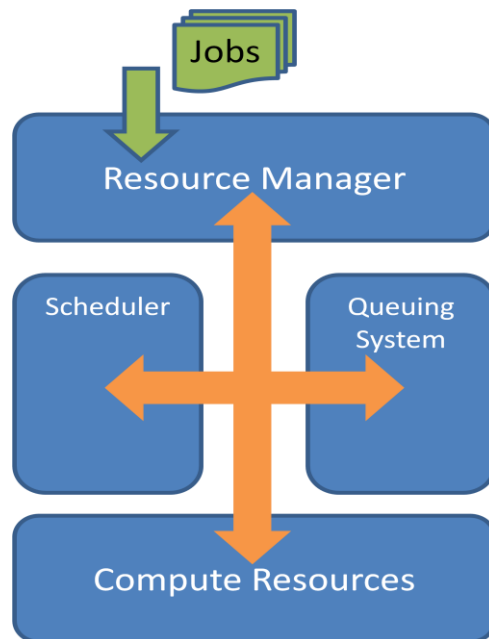


Figure 1: Overview of DRM Components

Why are Resource Managers and Schedulers Important?

Now that you understand what resource managers and schedulers are and how they interact, knowing their benefits will help you in your selection process. Each DRM and scheduling package has its own strengths and weaknesses so the “right” choice depends on the unique requirements of your HPC environment and the relative importance you place on them.

Increased throughput / utilization

- Accomplished with various queuing algorithms (e.g. backfilling)

Decrease administrative cost

- Less time spent in managing resources through use of abstraction and cluster-oriented tools
- Reduced effort managing jobs and queues

Improved job administration

- Global access to queues, submission rules, etc.

Improved job submission

- Abstracts resources making it simpler for users to utilize
- Provides common interface – less training, lower learning curve

Enforce business rules

- Job priority
- Resource allocation
- Submission filters to verify correctness of job data

Workgroup Cluster vs. Multi-Departmental Cluster

Not all HPC clusters are designed to meet the same goals. Some organizations may purchase a HPC cluster to provide compute resources to a small group of engineers or scientists, while others may be deploying a cluster designed to be used by multiple departments using various applications. Generally speaking, these clusters can be organized into two categories: workgroup and multi-departmental clusters.

Workgroup Clusters

Workgroup clusters are often utilized by a single department or small group of users and may be utilized in different ways than clusters serving the needs of multiple departments.

Workgroup clusters tend to be homogeneous in respect to both hardware and users.

It may not be important for workgroup clusters to be fully utilized all of the time. They tend to be used heavily for project work and then may sit idle for periods of time.

Queues tend to have fewer jobs waiting.

Primary goal is to run the job as quickly as possible. The scheduler needs to focus on optimizing available resources to process running jobs for quicker turnaround.

Multi-Departmental Clusters

These clusters are used by more than one department, often simultaneously.

Each department or user may have different applications and differing agendas which result in different resource requirements.

These tend to be heterogeneous in some way – if not hardware, then by user.

Queues tend to have more jobs waiting to be processed.

High utilization is usually an important objective.

Heterogeneous vs. Homogeneous Resources

If HPC clusters are designed for different requirements, it stands to reason that the compute resources that make up the cluster are often customized as well. Sometimes these customized resources mean that you may have different hardware within the cluster, or perhaps differing operating systems. For example, you may have 16 compute nodes running on Dell PowerEdge R610 servers augmented by a few Dell Precision T7500 workstations running powerful graphic processors for specialized graphics rendering applications. This mix of hardware resources presents a challenge for your DRM and scheduler. Deciding where to run a process is easier if all the resources are the same (homogenous) – simply pick the resources that aren't being used!

So what makes a cluster heterogeneous? By classical definition, if any of the resources have differing computation units (general purpose CPU vs. graphics processing unit, or GPU) or instruction set architectures (IA64, RISC, x86), then they are heterogeneous. The term is also commonly used to describe clusters with different operating systems (e.g. Linux vs. Windows).

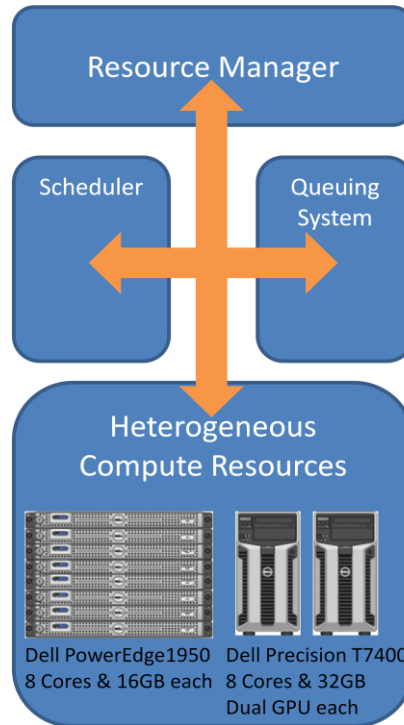


Figure 2: DRM with Heterogeneous Resources

Heterogeneous vs. Homogeneous Users

The term heterogeneous or homogeneous can be used to describe users as well. This could mean that your HPC user base has a single use-case scenario for the cluster, like a small team of engineers using a computational fluid dynamics package like ANSYS FLUENT™, making them homogeneous. It could mean that you have multiple teams in multiple departments, some using complex financial models based on Microsoft Excel™ spreadsheets, others performing oil and gas reservoir simulations. Add to this the fact that many times the client workstation may be running heterogeneous operating systems (e.g. Windows for the financial analysts and Linux for reservoir engineers).

Often, in cases of HPC clusters used by multiple departments, one department may have purchased the cluster from their budget and will “lend” or sell time on it to other departments to defray the cost. This scenario benefits from advanced DRM systems that allow for enforcing a lower priority to the jobs submitted by the non-owner. Charge-back accounting and reporting are other features to look for in these situations.

How to Manage Access to and Control of Cluster Resources

Overview of Distributed Resource Managers and Scheduler Packages

There are numerous resource managers and schedulers available on the market. Some are Open Source and freely available (be sure to verify the licensing requirements before installing them on your cluster). Others are either commercial packages or commercially supported Open Source packages. Commercial packages sold by Dell include Moab from Clustercorp, LSF from Platform Computing, and HPC Server 2008 from Microsoft. Some of the popular free Open Source packages (not explicitly supported by Dell) are Sun Grid Engine (SGE) from Sun Microsystems, Torque/Maui from Adaptive Computing (f.k.a. Cluster Resources), and Lava from Platform Computing.

Commercial Resource Managers and Schedulers

For organizations that require commercial support and may be looking for advanced features not available in open source packages, there are several solutions sold through Dell that should be considered:

Moab Workload Manager is a cluster workload management package from Adaptive Computing that integrates the scheduling, managing, monitoring, and reporting of cluster workloads. Moab Workload Manager is part of the Moab Cluster Suite, which simplifies and unifies management across one or multiple hardware, operating system, storage, network, license, and resource manager environments. Moab's development was based on the Open Source Maui job scheduling package (see below).

Platform LSF allows you to manage and accelerate batch workload processing for mission-critical compute- or data-intensive application workload. With Platform LSF you can intelligently schedule and guarantee the completion of the batch workload across your distributed, virtualized, High Performance Computing (HPC) environment. The benefits include maximum resource utilization and a commercially supported product.

Microsoft HPC Server 2008 is more than a resource manager or scheduler; it is a complete cluster management system. Resource provisioning, identity management, comprehensive reporting, and supporting network services are included. In addition to batch job scheduling, HPC08 supports new Service Oriented Application (SOA) workloads. HPC Server 2008 represents the second generation of Microsoft's HPC product (formerly CCS – Compute Cluster Server).

Open Source Resource Managers and Schedulers

There are many Open Source DRM and scheduler packages available, ranging from comprehensive to focused and from robust to buggy. Although not an exhaustive list, the packages below are some of the most commonly used in the industry.

SGE (Sun Grid Engine) is a Distributed Resource Management software system. Almost identical to the commercial version (Sun N1 Grid Engine) offered by Sun Microsystems, SGE aggregates compute power and delivers it as a network service. Grid Engine software is used with a set of computers to create powerful compute farms and clusters which are used in a wide range of technical computing applications. SGE also provides a Service Domain Manager (SDM) Cloud Adapter and initial Power Saving support. It provides an interface to manage Amazon Elastic Compute Cloud (EC2) Amazon Machine Images (AMI). It supports cloud-only and hybrid (local+cloud) Sun Grid Engine environments. SGE presents a seamless, integrated computing capability where users can start interactive, batch, and parametric jobs.

TORQUE is an open source Distributed Resource Management system providing control over batch jobs and distributed computing resources. It is an advanced open-source product based on the original PBS (Portable Batch System) project and incorporates community and professional development. It incorporates significant advances in the areas of scalability, reliability, and functionality and is currently in use at tens of thousands of leading government, academic, and commercial sites throughout the world. TORQUE may be freely used, modified, and distributed and is designed to work with the Maui Scheduler (below).

Maui Scheduler is an open source job scheduler for clusters. It is an optimized, configurable tool capable of supporting an array of scheduling policies, fair share capabilities, dynamically determined priorities, and exclusive reservations. A detailed system diagnostics, extensive resource utilization tracking, statistics, and reporting engine provides insight into how the cluster is being used. It also includes a unique advanced built-in HPC simulator for analyzing workload, resource, and policy changes. Maui must be used with a resource manager such as TORQUE (above).

Platform LAVA is an open source scheduler solution based on the workload management product LSF and designed to meet a range of workload scheduling needs for clusters with up to 512 nodes. Lava was designed not to lose a job once submitted to the system. Lava will reliably continue operation even under conditions of very heavy load without losing any pending or active jobs. In addition, as long as one host in the Lava cluster is operational, the system will continue to run. Lava supports multiple candidate master hosts and fails back seamlessly when the master has recovered. This gives Lava tremendous fault tolerance compared to traditional master-slave setup.

Condor is an open source workload manager, developed at the University of Wisconsin – Madison (UW-Madison), with a grid component. Condor performs the traditional batch job queuing and scheduling roles while the grid component allows the system to scavenge wasted CPU resources from idle desktop workstations. This feature implies improved ROI for budget-minded organizations trying to do more with less. Red Hat has based its MRG Grid product (part of Red Hat Enterprise MRG) on Condor and formed a partnership with UW-Madison for further development and enhancement of Condor.

Standards: DRMAA

With HPC clusters becoming more heterogeneous in nature and organizations using the various DRM systems and schedulers available, the challenge of finding a common method of interaction between systems has been a growing concern. One project that has met this challenge head on is the Open Grid Forum specification Distributed Resource Management Application API (DRMAA, pronounced “drama”). This goal of the specification was to produce an API that would be easy to learn, easy to implement, and that would enable useful application integrations with DRMS in a standard way.

The DRMAA specification is language, platform, and DRMS agnostic. A wide variety of systems should be able to implement the DRMAA specification. To provide additional guidance for DRMAA implementation in specific languages, the DRMAA Working Group also produced several DRMAA language binding specifications. These specifications define what a DRMAA implementation should resemble in a given language. Bindings are available for C, Java, Perl, Ruby, and Python languages.

DRMAA is purported to work with SGE, Condor, PBS/TORQUE, LSF, MOAB, and Mathematica.

I Want it All, and I Want it Now – Scheduler Policies

Product support for various policies can range from basic first-in-first-out (FIFO) to extensible, custom policies. The sections below describe some of the more prevalent scheduling policies used today.

FIFO – First In, First Out

The simplest scheduling algorithm, jobs are processed in the order received as resources are made available.

Pre-Emption

Designed to reduce wait times for high-priority jobs, the scheduler may kill lower priority running jobs to free up resources for higher priority jobs.

Advance Reservation

An advance reservation is the mechanism by which the scheduler guarantees the availability of a set of resources at a particular time. It is the job of the scheduler to make certain that the access control list is not violated during the reservation's timeframe on the resources listed. For example, a reservation may specify that node COMPUTE001 is reserved for user Sally on Monday. The scheduler will then be constrained to make certain that only Sally's jobs can use COMPUTE001 at any time on Monday.

Backfill

The scheduler looks ahead in the queue to see if a smaller, shorter job can be run out of order without preventing the next (larger) job from running. For example, if job 1 is running on 96 of 128 nodes and job 2 requires 124 nodes, the scheduler can backfill using job 3 that only requires 4 nodes, thus improving overall utilization. This is contradictory to strict ordering FIFO.

Fair Share

By tracking of historical resource utilization for each user, the scheduler has the ability to modify job priority as required. Users can be given usage targets, floors, and ceilings that can be configured to reflect the budgeted allocation request.

Grow & Shrink

The scheduler allows jobs with multiple tasks to resize over time. It can allocate all available resources to run tasks in parallel and, as tasks complete, it frees resources for other jobs without waiting for the entire job to finish.

How will users access it?

How do users actually submit jobs to the HPC cluster for processing? Depending on the DRM and scheduler, there are one or more ways to accomplish this.

Command Line Interface (CLI)

Historically the most common method of submitting jobs to a HPC cluster, users submit jobs by invoking an application and passing the proper arguments to it to supply resource requests, input data, and other job metadata. Submitting jobs via the CLI is prone to mistakes; an incorrect parameter could cause job failure or even loss of data. To help mitigate this risk, many cluster administrators find themselves building custom scripts for users to ensure correctness of job submission. Using the CLI means providing the users with either telnet or secure shell access to the cluster.

Graphical User Interface (GUI)

Some resource managers provide a GUI for job submission in an effort to make the process more user-friendly. Providing a GUI can help simplify the numerous attributes a user can specify when submitting a job. GUIs tend to be challenging to support in a heterogeneous environment, often requiring special libraries or applications to run them on different systems. For example, some may require X Window and OpenMotif libraries, while others may require Microsoft Windows.

Web Portal

A common method of providing a type of GUI using a model that most all users are familiar with is to access a special web portal via a browser. The functionality available in HPC web portals ranges from extremely simple to more advanced and dynamic. There are a few important things to keep in mind when reviewing systems that provide a web portal for job submission:

Does the system provide access to the most commonly used features and functions?

Does the system simplify the process and help the users, or does it hinder the submission process?

Application

Many specialized applications in the fields of computational fluid dynamics, 3D modeling, and bio-informatics have the ability to leverage HPC clusters for computation processing. Either directly (through a proprietary system) or using hooks into common DRMs, the user submits the job without leaving the application they are using.

What about the Cloud?

You may be familiar with “cloud” computing (named for the object most used to represent the Internet on network diagrams), but wonder how it related to high performance computing.

Overview of Cloud Computing

Computing as a service has reached the point of being considered a utility. Now mainstream, you can rent all or portions (through virtualization technologies) of physical machines. Vendors typically charge by the compute hour, although different plans exist. Cloud computing supports a very dynamic model: when workload increases, you can scale out additional resources on demand.

This represents numerous advantages for an organization. In contrast to physical, owned (or leased) systems, there is no upfront investment in infrastructure or people, and ongoing expenses are simplified. Total cost can be close to zero when resources are not in use. The cloud user can pay costs directly proportional to need rather than allocating resources according to average or peak load. However, most clouds have not been designed for the unique requirements of HPC. Specifically, high-speed, low-latency networks interconnecting the compute nodes, which are an important component of HPC clusters, are not found in most cloud infrastructures. However, cloud services may fit where applications do not require this high-speed networking between nodes.

DRM & Scheduler Product Matrix

Product Name	Vendor	Type	License	Cost?	Platforms Supported
Moab Workload Manager	Adaptive Computing	DRM	Commercial	Purchase	Linux, Mac OS X, Windows, AIX, OSF/Tru-64, Solaris, HP-UX, IRIX, FreeBSD & other UNIX platforms
Platform LSF	Platform Computing	DRM	Commercial	Purchase	Linux, Mac OS X, Windows, AIX, Solaris, HP-UX, and IRIX
HPC Server 2008	Microsoft	DRM & Scheduler	Commercial	Purchase	Windows
SGE	Sun Microsystems	DRM & Scheduler	Open Source (SISSL)	Free	Linux, Mac OS X, Windows, AIX, Solaris, and HP-UX
TORQUE	Cluster Resources	DRM	Open Source (OpenPBS)	Free	Linux, AIX, Tru-64, Solaris, HP-UX, IRIX, FreeBSD & other UNIX platforms
Maui	Cluster Resources	Scheduler	Open Source	Free	Linux, AIX, OSF/Tru-64, Solaris, HP-UX, IRIX, FreeBSD & other UNIX platforms
Platform Lava	Platform Computing	Scheduler	Open Source (Apache 2.0)	Free	Linux
Condor	UW-Madison	DRM & Scheduler	Open Source (Apache 2.0)	Free	Linux, Mac OS X, Windows, AIX, Solaris, and HP-UX

Conclusion

Resource managers and schedulers are core components of a functional HPC cluster. They not only allow easy use of the cluster, but also allow you to achieve better utilization and return on your investment. Understanding how the cluster will be used will help you select the best DRM and scheduler to meet your needs. With the many packages available today, you have more choices and a better probability of success than ever before!

About X-ISS

eXcellence in IS Solutions, or X-ISS, is a professional services firm located in Houston, Texas, that specializes in delivering high-performance computing solutions. X-ISS expertise includes designing, deploying, managing, and benchmarking clusters of all types and sizes, including various interconnects, cluster management suites, distributed resource managers, and popular HPC applications.

X-ISS manages and designs HPC Linux and Windows Compute Cluster Server based-clusters for oil and gas, government, education, life sciences, and numerous other vertical markets. X-ISS is vendor-agnostic and works with any hardware vendor or independent software vendor (ISV) to deliver cluster solutions.

X-ISS can help lower total cost of ownership, reduce administrative costs, optimize performance of both hardware and applications, provide independent benchmarks of ISV applications, and manage HPC clusters from small work group clusters through 12,000 core high throughput compute facilities.

If you have any questions regarding this white paper, please contact hpc@x-iss.com.