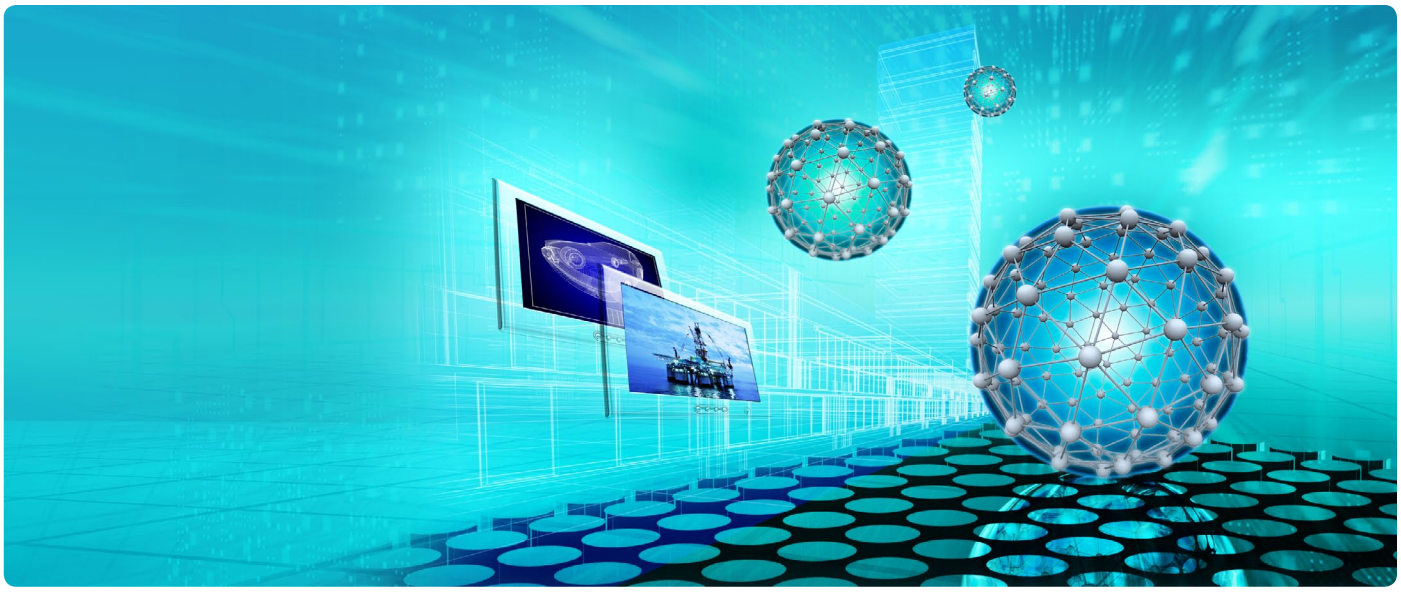


HPC Scheduling & Job Prioritization

Adaptive COMPUTING

Table of contents

Introduction

When children begin formal schooling, one of the first lessons they learn is the importance of forming an orderly line to reach their desired destination (usually during recess or lunch). High-performance computing clusters use schedulers to create a similar queue that ensure the right workloads are completed in the right priority order at the optimal times.

Today, modern schedulers are capable of achieving utilization rates up to 99 percent because of the policies and efficiencies they introduce in HPC cluster environments. The majority of HPC systems are outfitted with a scheduler, whether homegrown or open-source solutions for smaller or startup implementations, or robust commercial solutions for large-scale HPC systems, such as solutions offered by Adaptive Computing.

Adaptive Computing is the largest supplier of HPC workload management software and offers two HPC workload management products with the powerful Moab intelligence engine[1]. The Moab HPC Suite of products leverage Moab's multi-dimensional policies—including priority systems, future reservations, placement strategies and scheduling cycles—to continually schedule and monitor workloads, resources, SLAs and priorities.

Both Moab HPC Suite - Basic Edition and Moab HPC Suite - Enterprise Edition optimize workload output across complex, heterogeneous cluster environments or even HPC cloud environments. Adaptive Computing holds many key patents in large-scale scheduling, data management and reservations that play a critical role in making the Moab HPC solution possible.

Moab's chief task and number-one priority at all times is to schedule workloads. In order to do this, Moab passes through several different phases as part of the scheduling cycle. For the purpose of this white paper, this cycle will be divided into five steps.

The Scheduling Cycle

The scheduling cycle, also known as the scheduling iteration, constantly repeats itself, and is triggered by two types of events:

- **Lifecycle Events** – A new cycle is triggered by certain lifecycle state changes, such as the starting, ending and modifying of jobs.
- **Polling Events** – A new cycle is also triggered upon reaching a timeout, known as a polling interval. Moab regularly polls the cluster to capture the current state of nodes and virtual machines. By default, the polling interval is set to 30 seconds.

The five-step scheduling cycle proceeds as follows:

1. Update Information from Resource Managers

Moab first attempts to obtain a coherent understanding of the cluster because the more accurate information it has, the better scheduling decisions it can make. In order to do this, Moab contacts the resource manager(s) (e.g., TORQUE, SLURM, Nagios, Ganglia) for information on the full cluster and its resources (i.e., nodes, memory). Moab will do nothing else until it finishes contacting the resource manager(s) or a predefined timeout is reached. The information provided by the resource manager is then transferred to Moab's internal data structure to make a new round of scheduling decisions.

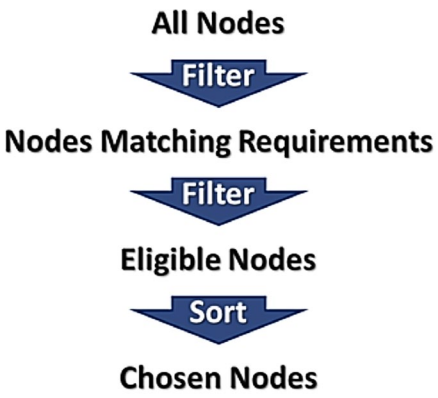
[1] IDC HPC End-user Study of System Software and Middleware in Technical Computing, 2013

2. Manage Workload

In this step, Moab decides which jobs to start, in what order and where they are placed. Policies are used to determine the exact method used, but in all cases jobs are ordered in the queue and then Moab attempts to place (i.e., schedule) them in the cluster.

Job Ordering

Every scheduling iteration Moab calculates a numeric priority for each job in consideration, and then orders them accordingly. Administrators are able to determine which factors are taken into account when this priority is calculated. Additionally, different weights can be applied to these factors, which allows administrators to target specific factors as being more important than others.



Each time through the scheduling cycle, Moab calculates a numeric priority value for each job using the values of the selected components, sub-components and weights. Components can include job credentials, fairshare usage, requested job resources, current and target service levels, consumed resources and job attributes.

Moab's SLA and priority policies ensure the highest priority workloads are considered first. Moab then attempts to start the jobs, beginning at the top and moving down the list. Once it gets to a job that cannot currently be started, a reservation is created for that job. Moab then switches to backfill mode to find jobs in the queue that it will be able to start.

For jobs Moab decides to start, it then needs to decide where to place them.

Job Placement

Job placement is determined through a simple process:

1. Start with all nodes in the cluster.

2. **Filter 1 – Geometry Check:** All nodes that cannot physically run the job are removed from consideration (i.e., too few cores or memory).

3. **Filter 2 – Policy Check:** All nodes that cannot run the job because of policy are removed from consideration (i.e., reservations or other running jobs).

4. **Sort:** Using the Node Allocation Policy, administrators specify which available resources should be allocated to each job. The nodes are sorted and the top results are chosen.

5. The job is sent to selected nodes.

This process is then repeated for each job that can be started. To efficiently use all available resources, Moab packs workloads and backfills around priority jobs and reservations.

3. Refresh Reservations

Following the workload management, Moab evaluates all reservations and makes needed changes as necessary, such as removing expired reservations, adding new reservations and repositioning movable reservations as needed.

4. Update Statistics

Moab updates system statistics from the cycle as well as statistical profiling, if enabled (e.g., users, groups, accounting, classes, quality of service).

5. Handle User Requests

In the final step of the scheduling iteration, Moab waits for and handles administrative requests and user requests (e.g., job submission, job information, blocking commandline requests) and portal requests (e.g., interactions with other external systems such as Moab Viewpoint).

Moab will ensure the first four steps are always completed, while this last step is optional. For instance, if Moab does not have enough time to accomplish everything within the specified polling interval, this final step will be sacrificed. As a result, user commands can begin timing out. As an upside, this means administrators can often diagnose and rectify the problem before it starts to affect Moab's number one priority: the scheduling of workload.

Monitoring The Scheduling Cycle

Administrators can monitor the different stages of the scheduling cycle through the following command: **mdiag –S –v –v**

This command provides in-depth diagnostic information for the scheduler to help administrators identify and troubleshoot issues, such as a slow-running system. The two –v switches are optional, but will provide additional information about the system.

Usage Information

Moab keeps track of the amount of time it spends in each one of the five phases. They are represented in the output of **mdiag –S,** with some of the phases split into multiple stages, or sub-phases, for display purposes. These include:

- **Sched** – This sub-phase relates to the amount of time Moab devotes to scheduling (i.e., how much time is being used to determine which jobs are going to run where). Spikes here are uncommon, though they can occur when overall system policies are changed or there is a major change in the workload patterns coming into the system.

- **RMLoad** – This sub-phase is the amount of time Moab spends reading information from the resource manager(s). If Moab is having a problem contacting the resource manager(s) (a blocking operation), the value for this sub-phase will spike. In that case, the troubleshooting investigation should examine the resource manager(s) and their communication channels with Moab. The other instance where this value may spike is if there is a significant increase in the number of resources being reported by a resource manager. However, this is extremely uncommon in an established system.

- **RMAction** – This sub-phase is the amount of time Moab spends sending its scheduling decisions to the resource managers (e.g., "Start Job X"). Spikes in this metric are generally caused by issues with the resource manager(s) or the communication channel between them and Moab. Generally, a spike seen here will correlate with a spike in **RMLoad.**

- **RMProcess** – This sub-phase records the amount of time it takes Moab to move the information received from the resource manager(s) into Moab's internal data structures. It is very uncommon for this metric to spike, and only occurs in an instance where the resource manager(s) starts reporting a significantly larger number of resources.

- **Trigger** – While triggers are handled by a separate scheduling cycle, their use does have some overhead, which is recorded with this sub-phase. Generally, most HPC systems do not have a significant number of triggers, and this number remains fairly small and steady. Only in specialized cases where a large number of triggers (i.e., hundreds or thousands) are being constantly added and deleted is it likely for this metric to spike. If it does, trigger debugging is the next step.

- **User** – This sub-phase commonly encounters problems, as it represents the amount of time Moab spends answering user requests, including those coming through web portals and MWS. Spikes can be caused by users having their scripts or jobs constantly issuing commands to Moab, such as continual requests to discover a job's status. If this metric spikes, one should consult the logs to determine the source of the unusually high number of requests.

- **Idle** – This sub-phase signifies the amount of time Moab spends waiting for user requests. When Moab cannot fulfill everything it needs to do in a scheduling iteration, it first sacrifices the **Idle** time and then the **User** time if the former was not sufficient. Administrators want the **Idle** metric to always be positive and comfortably but not too far from zero.

Generally, **mdiag –S** shows three different sets of the above stages. For debugging purposes, the five-minute average tends to be the best to use. Problems are often spotted and explored before the 24-hour average will be sufficiently changed to reveal where the play lies. The percentage column tends to be a little harder to use.

Scheduling Iteration Tuning

The first several scheduling iterations after Moab initially starts will generally be extraordinarily long, as Moab is conducting additional "start-up" activities. After these are completed, one should see the average time for each iteration approach the configured time for the scheduling iteration. On stable, long-running systems, the average value should not be more than the configured value.

If this does occur, one needs to determine which of the other sub-phases are creating problems. Alternatively, in some cases the configured scheduling iteration time simply needs to be increased, as there might not be enough time for Moab to complete all of its tasks within the allotted timeslot.

Note: To know when one of the sub-phases is at a non-optimal value, administrators must know the stable baseline for each metric. Adaptive Computing recommends administrators occasionally run **mdiag –S –v –v** to obtain these figures. This is particularly important after making major policies changes. Additionally, as mentioned above, one must make sure the system has "stabilized" before taking the reading.

Because every system possesses different workload patterns and types of resources, number values will be unique to each system. This will help diagnose slowdowns by allowing administrators to compare current values with what is normal for their system.

Conclusion

Moab HPC Suite is the most used scheduling/workload management software at HPC sites and Adaptive Computing is the No. 1 vendor used across unique HPC sites[2]. Administrators rely on Moab HPC Suite more than any other HPC solution because of its demonstrated capabilities in optimizing scheduling and accelerating workload productivity of their systems. Its rich policy set allows administrators to not only deal with the technical issues they face, but also the political and organizational issues before them.

Availability

For questions or to inquire about having Moab manage your system, contact Adaptive Computing.

[2]Intersect360 Research 2012 HPC Site Census Survey: Middleware, December 2012

Let's talk...Set up a Demonstration...and Test in your Environment

An Adaptive Computing solutions advisor can guide you to the products and services that will best meet your needsand will work with you to set up a live, online demonstration designed specifically for your organization.

Contact a solutions advisor by phone or email, or visit our Web site today

North America, Latin America +1 (801) 717.3700
Europe, Middle East, Africa +44 (0) 1483 243578
Asia, Pacific, Japan, India +65 6597-7053
Email: solutions@adaptivecomputing.com
www.adaptivecomputing.com

Corporate Headquarters
1712 S. East Bay Blvd.
Suite 300
Provo, Utah 84606

Adaptive COMPUTING