



Technical Report

Abstract: This document is intended to provide a comparison of three very popular batch systems (SGE, SLURM and Torque/Maui) to manage the allocation of compute cluster resources.

Document Id.: **CESGA-2014-002**


Date: **Frebruary 2014**

Responsible: **Javier Cacheiro**

Status: **FINAL**

Analysis of Batch Systems

SGE, SLURM, and TORQUE/Maui

Document identifier:	DO_SIS_Batch_System_Comparison_Technical_Report_V4.odt
Date:	13/02/2014
Document status:	DRAFT
Document link:	
License:	

Abstract: This document is intended to provide a comparison of three very popular batch systems (SGE, SLURM and Torque/Maui) to manage the allocation of compute cluster resources.

Copyright notice:

Copyright © CESGA, 2014.

See www.cesga.es for details on the copyright holder.

You are permitted to copy, modify and distribute copies of this document under the terms of the CC BY-SA 3.0 license described under <http://creativecommons.org/licenses/by-sa/3.0/>

Using this document in a way and/or for purposes not foreseen in the previous license, requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EGEE-III COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks:

SLURM is a registered *trademark* of SchedMD LLC

Moab is a registered *trademark* of Cluster Resources, Inc. (now Adaptive Computing)

The icons used in this document were obtained from:

<http://www.iconarchive.com>

<http://www.iconarchive.com/show/icloud-icons-by-ahdesign91.html>

<http://www.archlinux.org/packages/extra/any/oxygen-icons/download>

<http://www.iconarchive.com/show/vista-hardware-devices-icons-by-icons-land.html>

<http://hortonworks.com/blog/a-set-of-hadoop-related-icons/>

Document Log

Version	Date	Comment	Author
0	30/01/2014	Definition of the document structure	Javier Cacheiro
1	31/01/2014	Initial version	Javier Cacheiro
2	06/02/2014	Updated specific information about each batch system	Javier Cacheiro
3	07/02/2014	First internal release	Javier Cacheiro
4	13/02/2014	Review	Pablo Rey, Alvaro Simón

Content

1 Introduction.....	6
1.1 Purpose of the document.....	6
1.2 Application Area.....	6
1.3 References.....	6
1.4 Document Amendment Procedure.....	6
1.5 Terminology.....	7
1.6 Conventions.....	7
2 Executive Summary.....	9
3 Structure of this document.....	10
4 Introduction.....	11
5 SGE.....	12
5.1 Introduction.....	12
5.2 Main Functionalities.....	14
5.3 Usage across Supercomputing Sites.....	15
6 SLURM.....	16
6.1 Introduction.....	16
6.2 Main Functionalities.....	17
6.3 Usage across Supercomputing Sites.....	18
7 TORQUE/Maui.....	19
7.1 Introduction.....	19
7.2 Main Functionalities.....	21
7.3 Usage across Supercomputing Sites.....	22
8 Comparison.....	23
9 Conclusions.....	25

1 Introduction

1.1 Purpose of the document

This document provides a comparison of the main batch system solutions available as January 2014. Focus has been made in the solutions that are open-source or offer free versions so batch systems like LSF are excluded of the comparison.

1.2 Application Area

This document is intended for system administrators interested in batch systems; with its aim being to summarise the advantages and disadvantages of each of the job scheduling solutions available as January 2014. To aid readers with specific interests, there is a section devoted to each batch system analysed, and we recommend the readers to refer to *Section 3 Structure of this document* if they are interested in a specific topic.

1.3 References

Table 1: Table of references

R1	Univa Grid Engine: http://www.univa.com/products/grid-engine.php
R2	Son of Grid Engine: https://arc.liv.ac.uk/trac/SGE
R3	Open Grid Scheduler: http://gridscheduler.sourceforge.net/
R4	SLURM: http://slurm.schedmd.com/
R5	Torque: http://www.adaptivecomputing.com/products/open-source/torque/
R6	Moab HPC Suite Enterprise Edition: http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-suite-enterprise-edition/
R7	Altair PBS Works: http://www.pbsworks.com
R8	Torque-Maui Overview and experiences , G. Donvito, INFN Bari
R9	Running a 10,000-node Grid Engine Cluster in Amazon EC2 , Scalable Logic

1.4 Document Amendment Procedure

This document is under the responsibility of CESGA. Amendments, comments and suggestions should be sent to Javier Cacheiro (jlopez [at] cesga.es).




1.5 Terminology

Table 2: Glossary

SGE	Sun Grid Engine → Oracle Grid Engine → Univa Grid Engine
SoG	Son of Grid Engine
OGS	Open Grid Scheduler
SLURM	Simple Linux Utility for Resource Management
TORQUE	Terascale Open-Source Resource and QUEue Manager
Maui	Maui Cluster Scheduler
Moab	Moab Cluster Suite including a non-open-source scheduler with many similarities with Maui

1.6 Conventions

This document uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

	This icon indicates tips that could be useful for the reader.
	This icon indicates important considerations that are easily missed.
	This icon indicates critical aspects that should not be overlooked.

2 Executive Summary

In this technical report we provide a comparison of various *free* batch systems (*aka* resource managers) available in order to make an informed decision of which one to choose to manage a computing cluster of medium size.

We mainly consider batch systems that offer *free* versions and distribute the source code. There are several batch systems that fulfil this requirement but according to our experience the main players in this field are: SGE, SLURM and TORQUE/Maui.

Sun Grid Engine (SGE) has been used at CESGA during the last 10 years, it offers quite a lot of features and it is very stable. Sun provided frequent updates that fixed existing bugs and added new functionalities over the years making the product very complete. The acquisition of Sun by Oracle has ended the Grid Engine open-source project but two alternative forks have appeared: **Open Grid Scheduler** (OGS) and **Son of Grid Engine** (SoG). Of the two the only one that can be really considered under active development is SoG with 4 new releases in 2013. The OGS variant has not received any update since May 8, 2012.

SLURM is an open-source batch system started at LLNL and that is now very popular in large supercomputing centres because it offers high scalability and pretty complete functionality by using additional SLURM plugins. SLURM was inspired by the closed source Quadrics RMS and its aim was precisely to satisfy the requirements of large supercomputing centers. SLURM uses a modular design with many optional plugins to add additional functionalities and it can be integrated with other schedulers like Maui.

TORQUE is a very popular batch system usually deployed in conjunction with the **Maui** scheduler. It is the most widely used batch system in worldwide grid infrastructures like EGI and it also very popular across small and medium clusters and it is the main batch system. TORQUE is a community effort based on PBS, including more than 1200 patches that solved several important issues in the original PBS.

All three solutions offer a complete set of functionalities that should cover the requirements of any small or medium site.

- **SLURM** would be the main choice for **very large sites** as it can be considered better suited for this sites due to its higher scalability. Actually SLURM is the main choice for Petaflop systems.
- **SGE** would be the main choice for **small and medium sites** that want an easy to deploy integrated solution with a complete management GUI.
- **TORQUE/Maui** would be the main choice for **grid sites** that will be integrated in large grid infrastructures like EGI.

A detailed comparison of the three batch systems is given in Chapter 8.

3 Structure of this document

The document is structured as follows:

In Section 4 we provide an introduction to resource management of supercomputers and the main free batch systems available.

In Section 5 we provide a description of batch systems based on SGE, including the open-source variants Son of Grid Engine (SoG) and Open Grid Scheduler (OGS).

In Section 6 we provide a description of SLURM as well as its functionalities and a list of supercomputing sites using it.

In Section 7 we provide a description of batch systems variants of PBS, mainly focusing in the TORQUE/Maui variant.

In Section 8 we compare the different batch systems both in terms of functionality, easiness of development and support.

Finally, in Section 9 we recapitulate the main conclusions of this technical report.

4 Introduction

In this technical report when we refer to a **batch system** we mean a software application for controlling unattended background program execution—commonly called jobs—in a given cluster of computers. A batch system is also referred as: *Resource Manager, Distributed Resource Manager (DRM), Distributed Resource Management System (DRMS), Workload Management System (WMS), or Job Scheduler.*

According to wikipedia:

"Some widely used cluster batch systems are [Moab](#), [Argent Job Scheduler®](#), [Oracle Grid Engine](#), [Portable Batch System](#), [LoadLeveler](#), [Condor](#), [OAR](#), [Simple Linux Utility for Resource Management](#) and IBM's [Platform LSF](#)."

According to our experience, popular commercial solutions include IBM's Platform LSF, Univa Grid Engine, Adaptive Computing's Moab, and Altair's PBS Pro. Popular open-source solutions include SGE forks (SoG and OGS), SLURM, TORQUE/Maui, and Condor.

In this technical report we will mainly consider batch systems that offer free versions and distribute the source code, so we will focus in the second group. We will devote the next sections to analyse each of them separately in more detail.

In Section 5 we analyse SGE, including the open-source variants Son of Grid Engine (SoG) and Open Grid Scheduler (OGS). The main issue with SGE has been the discontinuation of their open-source version after the acquisition of Sun by Oracle. The development of new functionalities is now only possible through the commercial version by Univa and to a lesser extent the SoG version.

In Section 6 we analyse SLURM, this is a new batch system that was started at LLNL to satisfy their needs for a batch system to manage large supercomputers. Right now it is very popular and many users are migrating from other solutions like SGE to SLURM. Most of the petascale supercomputers use SLURM.

In Section 7 we provide a description of batch systems variants of PBS, mainly focusing in the TORQUE/Maui variant.

In Section 8 we compare the different batch systems both in terms of functionality, easiness of development and support.

5 SGE

5.1 Introduction

Sun Grid Engine (SGE) has been used at CESGA during the last 10 years, it offers quite a lot of features and it is very stable. Sun provided frequent updates that fixed existing bugs and added new functionalities over the years making the product very complete. The acquisition of Sun by Oracle changed the play-field, by first halting the free version of Grid Engine and later selling Grid Engine to Univa.

The first decision of Oracle was to stop the release of the courtesy free binaries and later also stopped releasing the source code under the SISSL license. Finally the current situation is that there are three independent versions:

- **Univa Grid Engine:** commercial version produced by the same team that previously produced SGE in Sun.
- **Open Grid Scheduler (OGS):** open-source project based on the latest free release by Sun (SGE6.2u5) maintained by Scalable Logic—the company previously hired by Sun to take care of the open-source version of SGE—and Gompute.
- **Son of Grid Engine (SoG):** open-source project based on Univa's free code.

Of the two open-source projects the only that can be considered under active development is SoG with 4 new releases in 2013. Mainly all the work under SoG should be attributed to just one person: Dave Love from the University of Liverpool.

The OGS variant has not received any update since May 8, 2012 when they released Grid Engine 2011.11p1 which represented an update of the first release of the project: 2011.11—a repackage of SGE6.2u5 with latest bug fixes.

On October 22, 2013 Univa announced it acquired the intellectual property and trademarks for the Grid Engine technology and that Univa will take over support.

The original Grid Engine open source project website closed in 2010 but versions of the technology are still available under its original **Sun Industry Standards Source License** (SISSL). Those projects have forked from the original project code and are known as Son of Grid Engine (SoG) and Open Grid Scheduler (OGS).

SGE is designed to foster maximum flexibility. Most behaviours in the software can be configured or overridden with administrator-defined scripts. The ability to configure the Oracle Grid Engine software to meet such a wide variety of needs has helped to make it one of the world's most widely deployed DRM systems.

SGE is written in C++ and, until SGE 6.2u5, the source code was released under the SISSL license.



SGE source code is difficult to read and debug. Even if Grid Engine was an open source project there is a lack of developers documentation making very difficult for external developers to contribute to the project.

Architecture

SGE consists of a central *qmaster* daemon running on a primary management node with the option to use several secondary management nodes with *shadowd* daemons that monitor the status of *qmaster*. Each compute node runs an *execd* daemon.

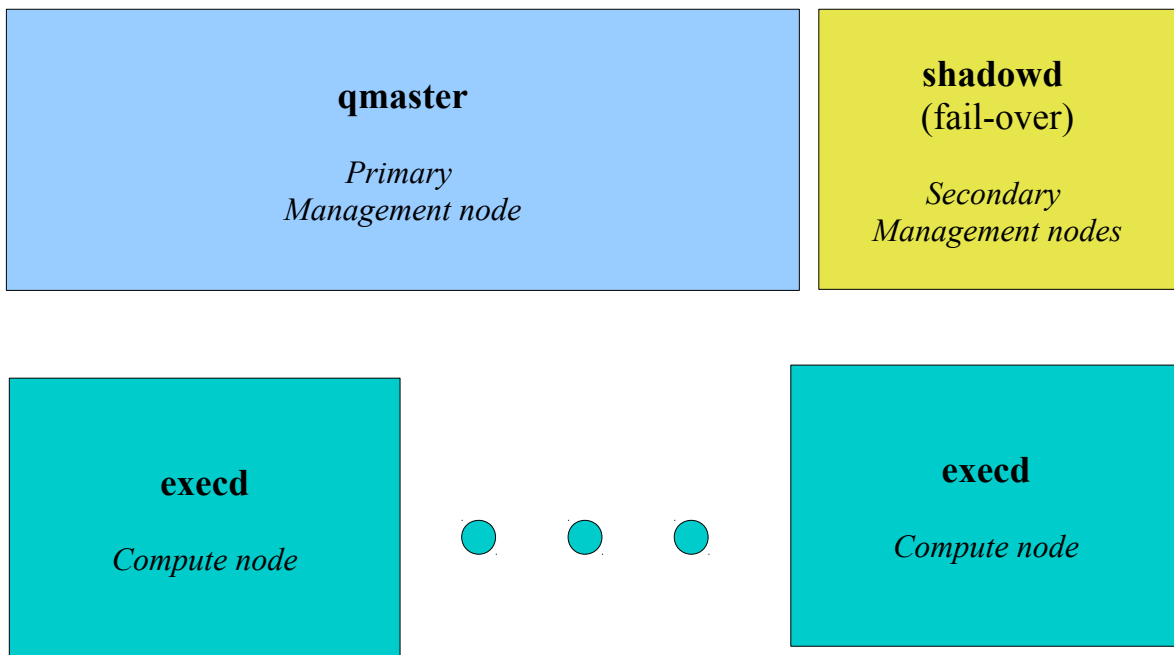


Figure 1: SGE Architecture

License

Open Grid Scheduler/Son of Grid Engine (both forks of the former FOSS Grid Engine project): Sun Industry Standards Source License (SISSL).



Even if OGS and SoG forks are source-code they use the SISSL license—now retired—that is more restrictive than the GNU GPL and Apache.

Support

Community support is provided through the users@gridengine.org mailing list.

Commercial support is provided by Univa.

5.2 Main Functionalities

These are the main functionalities included in SGE 6.2u5:

- Rule-based Resource Quota control
- GUI installer and SGE Inspect
- GUI management interface: qmon
- Topology-aware scheduling and thread binding
- Resource reservation (automatic)
- Advance reservation (manual)



Advanced reservations were first included in the latest release of SGE (SGE 6.2). They have not received additional development since their release and subsequent acquisition of Sun by Oracle. Unfortunately they have several bugs that have not been solved yet, like the fact that you can not actually use advanced reservations of more than 24 hours of duration.

This leads to several limitations for their use in production systems.

- **Qmaster daemon fault tolerance**
- Job fault tolerance
- Job Submission Verifier (job verification)
- Interactive jobs
- Enhanced remote execution (without using external rshd/rlogind/sshd processes)
- Hadoop integration
- Amazon EC2 integration
- Advanced scheduling algorithms: functional, policy-based, or priority.
- Job checkpointing support: BLCR
- Array jobs
- Array job interdependencies
- DRMAA support
- XML status reporting
- xml-qstat web interface



Apparently xml-qstat development was abandoned in 2011

- Parallel job support (MPI, OpenMP)
- Usage accounting
- Accounting and Reporting CONsole (ARCO): web interface
- Parallel make: distmake, dmake, and SGE's own qmake
- FLEXlm integration and multi-cluster software license management with *LicenseJuggler*

5.3 Usage across Supercomputing Sites

Texas Advanced Computing Center (TACC) used SGE to manage the workload on their Ranger system (**3,936 compute nodes**, 62,976 cores, 579 TFLOPS). This system was decommissioned on February 2013.

Currently, TACC uses SGE 6.2 for their Lonestar supercomputer (1,888 compute nodes, 302 TFLOPS) and SLURM 2.4 for their Stampede system (6,400 compute, 10 PFLOPS).



Even if Lonestar uses SGE, the largest TACC supercomputer Stampede (6,400 compute nodes, 10 PFLOPS) uses SLURM.

On November 2012 Scalable Logic performed a scalability test of OGS launching a **10,000** node cluster on Amazon EC2 [R9].

6 SLURM

6.1 Introduction

SLURM is an open-source batch system designed to support very large clusters. The design is very modular with many optional plugins to extend its functionality.

SLURM was initially developed at Lawrence Livermore National Laboratory (LLNL) to manage their supercomputers. In 2010, the primary architects and developers of SLURM—Moe Jette and Danny Auble—founded SchedMD, which maintains the canonical source, provides development, level 3 commercial support and training services. Commercial support is also available from other vendors like Bull and Cray.

SLURM is the batch system used on many TOP500 supercomputers, including the 50% of the top 5 supercomputers of November 2013.

Architecture

SLURM consists of a central *slurmctld* daemon running on a primary management node with the option to use a secondary management node with a fail-over twin and a *slurmd* daemon running on each compute node. The *slurmd* daemons provide fault-tolerant hierarchical communications.

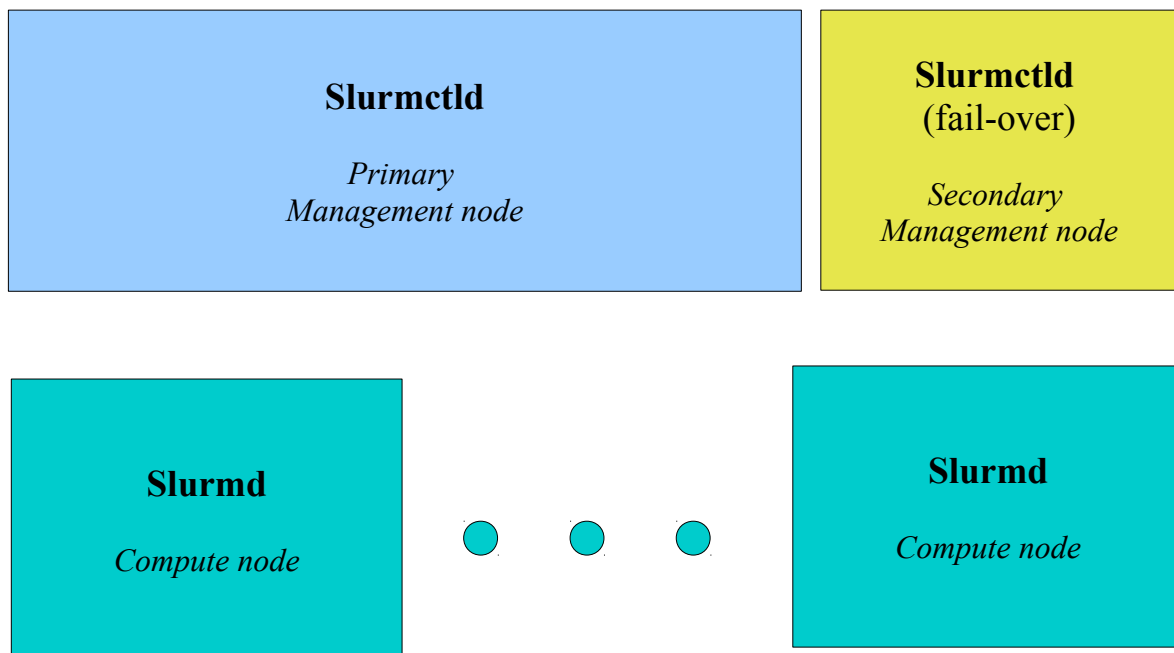


Figure 2: SLURM Architecture

License

SLURM is available under the GNU GPL V2.

Support

The support is provided through the following mailing list:

slurm-devel: slurm-dev@schedmd.com

<http://dir.gmane.org/gmane.comp.distributed.slurm.devel>

There are also several companies that provide commercial support:

- SchedMD: the core company behind Slurm
- Bull
- Cray
- Science + Computing

6.2 Main Functionalities

Scalability: It is designed to operate in a heterogeneous cluster with up to tens of millions of processors.



SLURM is used in production in supercomputers with as much as 98,000 nodes.

Performance: It can accept 1,000 job submissions per second and fully execute 500 simple jobs per second (depending upon hardware and system configuration).

Power Management: power usage consumed by a given job is recorded. Idle resources can be automatically powered down until needed.



SLURM provides support for energy-aware scheduling

Fault Tolerant: Includes the option to run a secondary *slurmctl* daemon.

Flexibility: *Plugins* exist to support various interconnects, authentication mechanisms, schedulers, etc.

Resizable Jobs: Jobs can grow and shrink on demand. Job submissions can specify size and time limit ranges.

Status of individual job tasks: It provides the status running jobs up to the level of individual tasks helping to identify load imbalances and other anomalies.

Integration into the EGI grid platform: SLURM integration into UMD—the middleware used in EGI—is being tested and it is almost ready for production under EMI-3.

6.3 Usage across Supercomputing Sites

SLURM is very popular between large supercomputing centers. On the November 2013 Top500 list, five of the top ten systems use SLURM, including the number one system: Tianhe-2.

Tianhe-2: designed by [The National University of Defense Technology \(NUDT\)](#) in China has **16,000** nodes, each with two Intel Xeon IvyBridge processors and three Xeon Phi processors (33.86 Petaflops).

Sequoia: an [IBM](#) BlueGene/Q system at [Lawrence Livermore National Laboratory](#) with **98,304** compute nodes (17.17 Petaflops).

Piz Daint: a [Cray](#) XC30 system at the [Swiss National Supercomputing Centre](#) with 28 racks and **5,272** hybrid compute nodes (6.27 Petaflops).

Stampede: a Dell cluster at the [Texas Advanced Computing Center/University of Texas](#) with 6,400 hybrid compute nodes (5.17 Petaflops).

TGCC Curie: a Bull cluster at CEA with 5,416 compute nodes (2 PetaFlops).

Several large **grid sites** from EGI are **planning to move from TORQUE/Maui to SLURM** now that SLURM is being integrated in the middleware—from EMI-3.

7 TORQUE/Maui

7.1 Introduction

TORQUE is a community effort based on the original OpenPBS project including more than 1,200 patches that have greatly improved many areas of the original project, improving scalability, enabling fault tolerance, and adding additional features with extensions contributed by many HPC organizations across the world.

TORQUE is the most widely used batch system in worldwide grid infrastructures like EGI and it also very popular across small and medium clusters and it is the main batch system.



TORQUE is probably the most widely used batch system nowadays. Most sites of the EGI infrastructure use TORQUE/Maui to manage their clusters.

TORQUE provides a very basic scheduler, for this reason it is usually integrated with the open-source Maui scheduler or the proprietary Moab Workload Manager from Adaptive Computing.

Maui Cluster Scheduler is currently maintained and supported by Adaptive Computing, although most new development has come to a standstill. A next-generation non-open-source scheduler is part of the Moab Cluster Suite and borrows many of the same concepts found in Maui. Maui's developers state that the licence satisfies some definitions of open-source software and that it is not available for commercial usage.



Adaptive Computing is no longer enhancing the open source Maui Cluster Scheduler product. For this reason, Maui development has almost stopped and the focus of Adaptive Computing is now their proprietary Moab Cluster Suite.

Architecture

TORQUE/Maui consists of a central *pbs_server* daemon running on a primary management node in conjunction with a separated daemon for the maui scheduler. Each compute node runs a *pbs_mom* daemon.

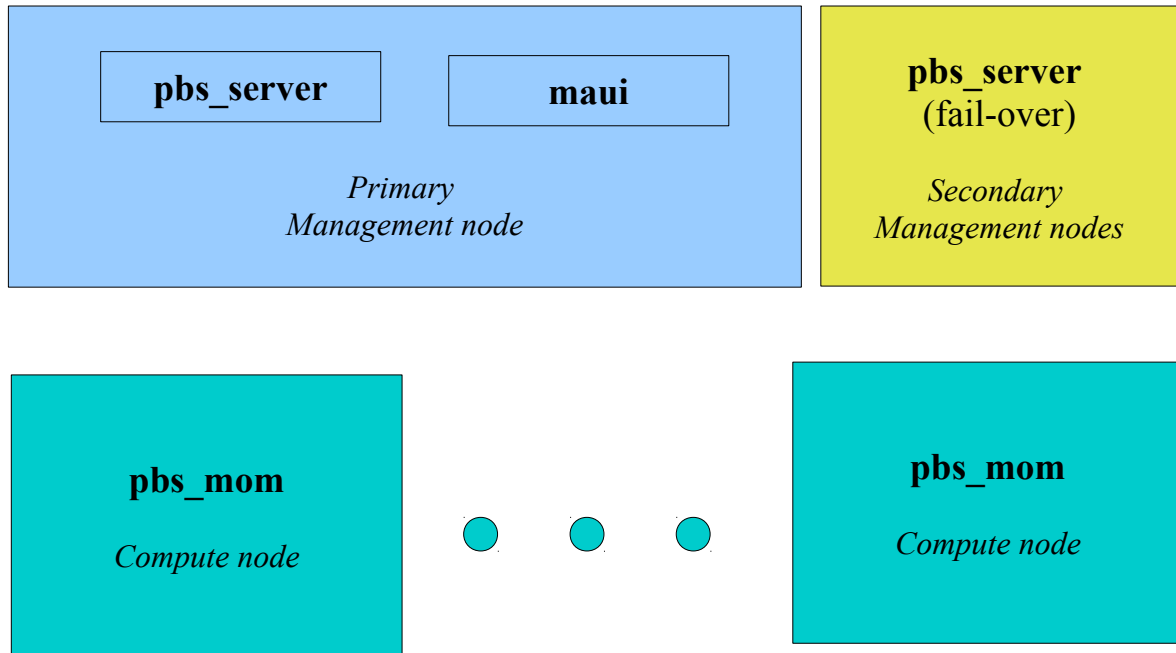



Figure 3: TORQUE/MMaui Architecture



TORQUE/Maui has the disadvantage of using two independent implementations for the master and scheduler. This usually leads to duplications in certain functionalities, the requirement to maintain two independent configurations (one for torque and one for maui), and the difficulty to propagate changes from one to the other.

License

TORQUE and Maui source-code is available but they use a non-common licenses.

TORQUE: It uses the OpenPBS license. Cluster Resources, Inc. describes it as open-source and Debian classifies it as non-free owing to issues with the license.

Maui: It uses the [Moab Scheduling System - End User Open Source License](#).



Even if TORQUE and Maui are free and distribute the source-code the licensing model is not clear and it is not clear that TORQUE and Maui are free for commercial use.

Support

The support is provided through the following mailing lists:

- torqueusers: torqueusers@supercluster.org
<http://www.supercluster.org/pipermail/torqueusers/>
- mauiusers: mauiusers@supercluster.org
<http://www.clusterresources.com/pipermail/mauiusers/>

There are also several companies that provide commercial support:

- Adaptive Computing (Moab)
- Altair (PBS Works)

7.2 Main Functionalities

TORQUE provides several enhancements over standard OpenPBS and it is actively developed by Adaptive Computing adding new functionalities with each new release (about 10 releases per year).



Maui is a very complete scheduler offering a lot of advanced scheduling functionalities including fair-share, backfilling and reservations.



There are two independent interfaces to manage TORQUE/Maui.

- TORQUE: *qmgr*, *pbsnodes*, *qstat*, *qsub*
- Maui: *showq*, *diagnose*, *setspri*

Sometimes this leads to errors and, in general, it makes more difficult to diagnose problems.

- Fault Tolerance: possibility to use a HA *pbs_server* configuration
- QOS support including service targets
- Dynamic priorities
- Fair-share scheduling
- Deadline based scheduling
- Reservation of resources
- Node health checks
- Allows the collection of statistics for completed jobs
- Ability to handle larger clusters (over 15 TF/2,500 processors)

- Ability to handle larger jobs (over 2000 processors)
- Support for cpusets
- Support for array jobs
- GPU support
- Very good support in gLite and other grid middlewares

7.3 Usage across Supercomputing Sites

Most EGI grid sites use TORQUE/Maui, including: Cyfronet, DESY, PIC, Hellasgrid, INTFN, etc.

San Diego Supercomputer Center (SDSC) uses TORQUE with Catalina scheduler in all its supercomputers including the largest one: Gordon (**1,024** compute nodes + 64 I/O nodes).

Oak Ridge uses PBS and Moab in their Titan Cray XK7 system (**18,688** compute nodes, 27 PFlops)

8 Comparison

Functionality	SGE	SLURM	TORQUE /Maui
MPI support	X	X	X
Interactive jobs	X	X	X
Advance reservation	X	X	X
Checkpointing	X	X	
Resource Quota	X		X
DRMAA	X	X	X
Usage accounting	X	X	X
License management: Flexlm integration	X	Planned for next release (14.03)	
CPU Core Binding	X	X	
LDAP integration	X	X	X
Fault-tolerant master	X	X	X
Integration with Hadoop	X	X	
Support for array jobs	X	X	X
Accounting of job's power consumption		X	
Fair-share scheduling	X	X	X
Energy-aware scheduling		X	
Support for interactive configuration changes (without restarting the daemon)	X		
Resizable jobs		X	

Table 1: Batch system features comparison.

	SGE	SLURM	TORQUE/Maui
Source code	C++	C	C
Active development	Low	Yes	Torque: Yes Maui: No
Active community	Medium January: 147 messages	Medium January: 156 messages	Medium January: 127 messages
Quality of the documentation for new developers	Bad	Good	Bad
Time to solve new bugs	High <i>Open bugs: 1133</i> <i>Oldest bug: July 2001</i>	Low <i>Open bugs: 36</i> <i>Oldest bug: Feb 2013</i>	Unknown <i>No public bug tracking system available</i>
Releases in 2013	SoG: 4 OGS: 0	12	TORQUE: 10 Maui: 0
Easiness to read and follow the source code	Difficult	Easy	Difficult
Scalability	Medium (10,000 nodes)	High (100,000 nodes)	Medium (20,000 nodes)

Table 2: Batch system comparison (non-functional)

9 Conclusions

In this technical report we provided a comparison of the three most widely used *free* batch systems: SGE (SoG and OGS), SLURM, and TORQUE/Maui; with the aim of helping site administrators to make an informed decision of which one to choose to manage their computing clusters.

Sun Grid Engine (SGE) has been used at CESGA during the last 10 years, it offers quite a lot of features and it is very stable. Sun provided frequent updates that fixed existing bugs and added new functionalities over the years making the product very complete. The acquisition of Sun by Oracle changed the play-field, by first halting the free version of Grid Engine and later selling Grid Engine to Univa.

The first decision of Oracle was to stop the release of the courtesy free binaries and later also stopped releasing the source code under the SSISSL license. Finally the current situation is that there are three independent versions:

- **Univa Grid Engine:** commercial version produced by the same team that previously produced SGE in Sun.
- **Open Grid Scheduler (OGS):** open-source project based on the latest free release by Sun (SGE6.2u5) maintained by Scalable Logic—the company previously hired by Sun to take care of the open-source version of SGE—and Gompute.
- **Son of Grid Engine (SoG):** open-source project based on Univa's free code.

Of the two open-source projects the only that can be considered under active development is SoG with 4 new releases in 2013. Mainly all the work under SoG should be attributed to just one person: Dave Love from the University of Liverpool.

The OGS variant has not received any update since May 8, 2012 when they released Grid Engine 2011.11p1 which represented an update of the first release of the project: 2011.11—a repackage of SGE6.2u5 with latest bug fixes.

SLURM is an open-source batch system started at LLNL and that is now very popular in large supercomputing centers because it offers high scalability and pretty complete functionality by using additional SLURM plugins. SLURM was inspired by the closed source Quadrics RMS and its aim was precisely to satisfy the requirements of large supercomputing centers. SLURM uses a modular design with many optional plugins to add additional functionalities and it can be integrated with other schedulers like Maui.

TORQUE is a very popular batch system usually deployed in conjunction with the **Maui** scheduler. It is the most widely used batch system in worldwide grid infrastructures like EGI and it also very popular across small and medium clusters and it is the main batch system. TORQUE is a community effort based on PBS, including more than 1200 patches that solved several important issues in the original PBS.

All three solutions offer a complete set of functionalities that should cover the requirements of any small or medium site.

- **SLURM** would be the main choice for **very large sites** as it can be considered better suited for this sites due to its higher scalability. Actually SLURM is the main choice for Petaflop systems.
- **SGE** would be the main choice for **small and medium sites** that want an easy to deploy integrated solution with a complete management GUI.
- **TORQUE/Maui** would be the first choice for **grid sites** that will be integrated in large grid infrastructures like EGI.



In terms of future development SLURM seems to be the most promising of all the batch systems evaluated including a strong development community.