# Extension of PBS Monitoring Tool

**Project Documentation**

**Project of Software Validation**

**Upgraded Web based application for PBS-Cluster**

**By**

**Shruti Mahesh Kulkarni-00369013**

**Advisor: Prof.Dr. Mr.Martin Thost**

# Table of Content

# Introduction to High Performance Computing 3

# About the Project 5

# Objective of the Project 5

# Technology Used 6

# User PBS Environment-PuTTy 6

# PBS Shell Commands 7

# PBS Resources 7

# Torque 7

# Architecture 8

# System Software 11

# Programming Environments and Tools 12

# 6.1. Programming Specification 12

# 6.2. Test Specification 12

# UML Diagrams 13

# Web based Application 18

# Test case 1: Sorting the Status 20

# Test case 2: qdel 22

# Test case 3: qrls 25

# 8.4 Test case 4: reloading the page after each 5 or 10 seconds

# 8.5 Test Case 5: Licence Monitoring 27

# Shell in PHP 28

# Software Validation 30

# 10.1 Exception Handling 30

# 10.2 Java Script Validation 31

# Migration of Web Application 32

# Shh Passwordless 34

# Bibliography 35

# 14. The Annex 36

# Introduction to High Performance Computing

Most recent developments, such as high performance and parallel computingofferings are having rapidly growing requirements for homogeneous and heterogeneous cluster solution approaches. High Performance Computing adds power to computation by providing a cluster solution for industrial applications using simulations, modeling, analysis in industries such as pharmaceuticals, CAD/CAM, ecommerce, life sciences, meteorology, aerospace, oil and gas exploration.

Small and medium sized company businesses have a necessity for coordination between their data hence they need to be connected together with a centralized cluster. This centralization cluster brings all together by virtually or physically close such that it is easy to manage, monitor and account the users. It helps the users to perform calculations, simulations, etc.

Basically, the high performance computer comes into picture and play a curial role by connecting the workstations with a network and organizing it centrally with a web based software application. The cluster provides a solution to mass production processing networks by the means of parallelization of web based applications present on the server. Batch systems used to distribute the application uniformly over the resources of the cluster.

The application gives the possibility of uniformly and providing availability to execute the user process threads. The batch system abstracts the technical details like CPU, memory used, Disk I/O information on each node and also the information regarding the nodes which are free for the users to perform their calculations on the web based application.

# About the Project

To handle the management and monitoring of the computational workload on clusters, the Portable Batch System (PBS) plays a vital role as a resource management system. To this management system the users submit jobs where they are queued up until the system is ready to run them. PBS selects which job to run and decides when and where to run the job in order to balance competing user needs and to maximize efficient use of the cluster resources.

A control script file which includes the script to execute the operation is submitted to the PBS server by a PBS command to run on the HPC cluster. The control script is a shell script containing the set of commands which the user wants to run on cluster computer nodes. The system that allows usage of PBS actions like job controlling (qsub, qdel, qrls) and job monitoring (qstat, qselect) on web based monitoring platform so that it is user-friendly by automating via PHP.

The submission of user process is nothing but the user jobs to the batch system. Here the PBS system acts batch system and the web based application is the PHP web based application. The PBS system processes the user process by standard specified commands which are known as the PBS commands or also are called as standard TORQUE commands. The user calculations are written in a program and the TORQUE commands are used at the shell command line that runs the user program.

For an instance, we can say that the user *qs*creates some calculation, so now the user has developed a logic in the form of program. Assuming it as user’s job and well now the job needs to submit it on the server to get output of the user calculations. With the help of TORQUE commands *qsub* the user submits the job.

*qsub job\_script*

# Objective of the Project

Currently the user’s use **PuTTY** as a platform to execute their commands. PuTTY is a free SSH, Telnet and Rlogin client for Windows system. PuTTY is command line interface to users to use TORQUE commands via SSH, Telnet and Rlogin network protocols. These protocols login from one multi user system to another system by the means of network or over the network.

# Technology Used

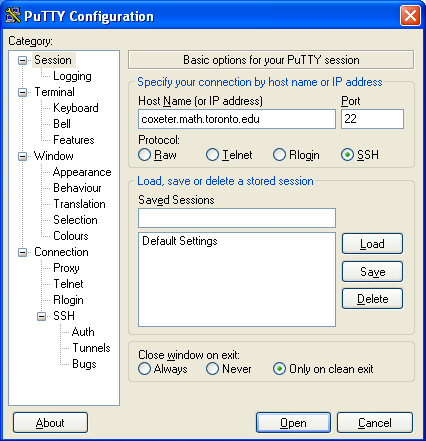
# 4.1. User PBS Environment-PuTTy

PuTTY supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as 3DES, Arcfour, Blowfish, DES, and Public-key authentication.

It also can emulate control sequences from x term, VT102 or ECMA-48 terminal emulation, and allows local, remote, or dynamic port forwarding with SSH.The network communication layer supports IPv6, and the SSH protocol supports the zlib@openssh.com delayed compression scheme.

It can also be used with local serial port connections.PuTTY comes bundled with command-line SCP and SFTP clients, called "pscp" and "psftp" respectively[[1]](#footnote-2).

To achieve a connection to server, the user connects to server via remote login and submits their calculations to specific the server as a pbs job. Main task to automate the process and provide it from frontend by which the motive to avoid PuTTY can be achieved.



# PBS Shell Commands

When you submit and monitor jobs on compute clusters, you are using TORQUE - a version of PBS. A popular open-source resource manager, TORQUE is used at thousands of research sites globally. With the commands available through TORQUE you can allocate resources, schedule and manage execution, monitor and view the status of your jobs. Some of TORQUE's commands are used at the shell command line, others are embedded in the shell script that runs your program.

# PBS Resources

In the batch script, you can specify the resources needed by your job. This does not mean that you can just specify the maximum amount of resources possible at all times. PBS tries to optimize the utilization of resources and to balance the job load. If current resources does not meet your requirement, PBS will let your job wait in the queue and pick a job in the queue, whose job requirement can be satisfied by the current resources, to run.

# Torque

Frequently Used Shell Commands

qsub submit a pbs job qsub [script] $ qsub job.pbs

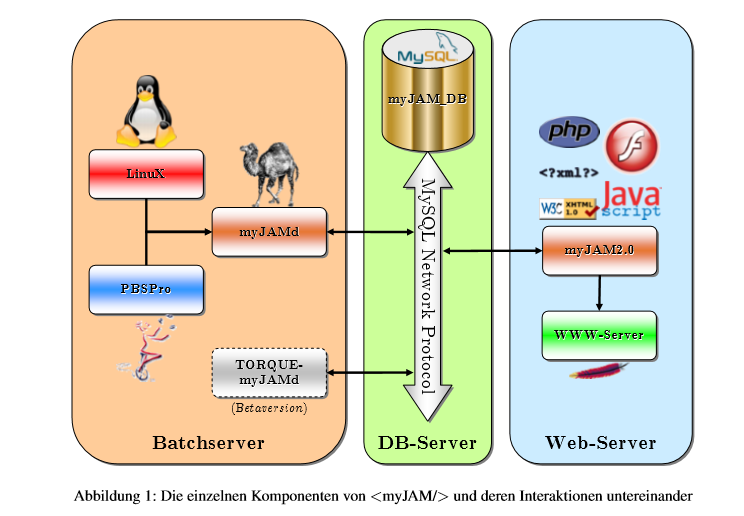
qstat show status of pbs batch jobs qstat [job\_id] $ qstat 44

qdel delete pbs batch job qdel [job\_id] $ qdel 44

qhold hold pbs batch jobs qhold [job\_id] $ qhold 44

qrls release hold on pbs batch jobs qrls [job\_id] $ qrls 44[[2]](#footnote-3)

# Architecture

[[3]](#footnote-4)

The tool <myJAM /> ("Job Accounting and Monitoring ", www.myjam.uni-duesseldorf.de) already supports the batch base operation of the central heterogeneous Linux computing cluster at the HHU Linux with a connection to the batch system PBS Pro and to the medium as a platform-form over cross open source software are provided. <myJAM /> consists of three main components:

<myJAM> Daemon: A \* nix daemon that runs on each server node PBS.It collects the information over the current and pending jobs from the batch systemor the operating system and stores it in the database.

Database: A MySQL database in which all information stored over projects, users, jobs and applications.

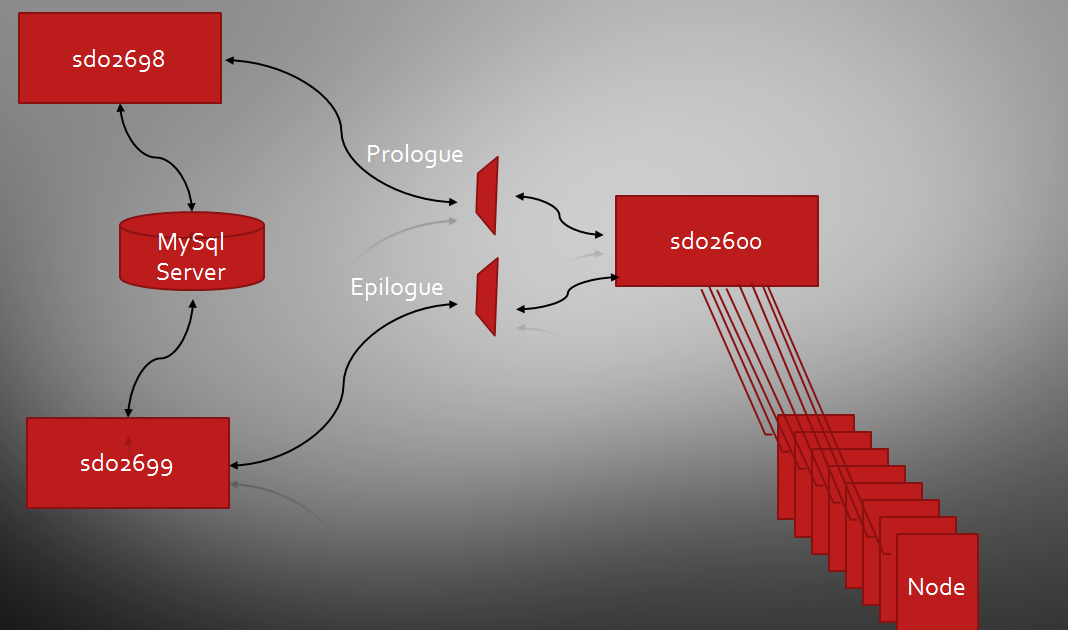
Web-Frontend: A highly interactive web application, with the information collected in the database according to different criteria can be analyzed and visualized. Also users and projects can manage become.

Each of these three components runs independently of theanderen. each other.Die Kommunikation findet ausschließlichuber das standardisierte Communication is exclusively via the standardized ” " MySQL NetNetwork Protocol“ statt, weshalb jede Komponente auch auf einem eigenen Server laufenwork Protocol " instead, which is why every component running on a separate serverkann. can. Grundsatzlich konnen mehrere <myJAM/>-Daemons auf verschiedenen Batch-SeBasically can more <myJAM /> - daemons on different batch server vern laufen, um mehrere Batchsystemeuberwachen zu konnen.provides protection to batch systems over guards.Durch die Modularitat und die wohldefinierte Kommunikation ist es relativ einfach, neue

By modularity and well-defined communication, it's relatively easy to add new Daemons fur weitere Batchsysteme zu entwickeln.daemons to develop for another batch systems. Die Datenbank und das Web-Frontend The database and the web frontend havebedurfen keiner Anpassung. no adjustment. Aus dem gleichen Grund konnten auch einfach neue Front- For the same reason could also simply new frontends entwickelt werden, z. B. ein Kommandozeilen-Interface.ends are designed for instance, a command line interface.

Zusatzlich kommen noch Prolog- und Epilog-Skripte zum Einsatz. In addition, the prologue and epilogue scripts are used. These scriptsvon den meisten Batchsystemen unmittelbar vor (Prolog) und nach (Epilog) dem Start by most batch systems immediately before (prologue) and after (epilogue) the start a job executed. <myJAM/> verwendet diesen Mechanismus, um job- <myJAM /> uses this mechanism to jobbezogene Daten (z. B. Knotennummer, Zeitstempel) in der Datenbank zu speichern. to store related data (eg., node number, time stamp) in the database.

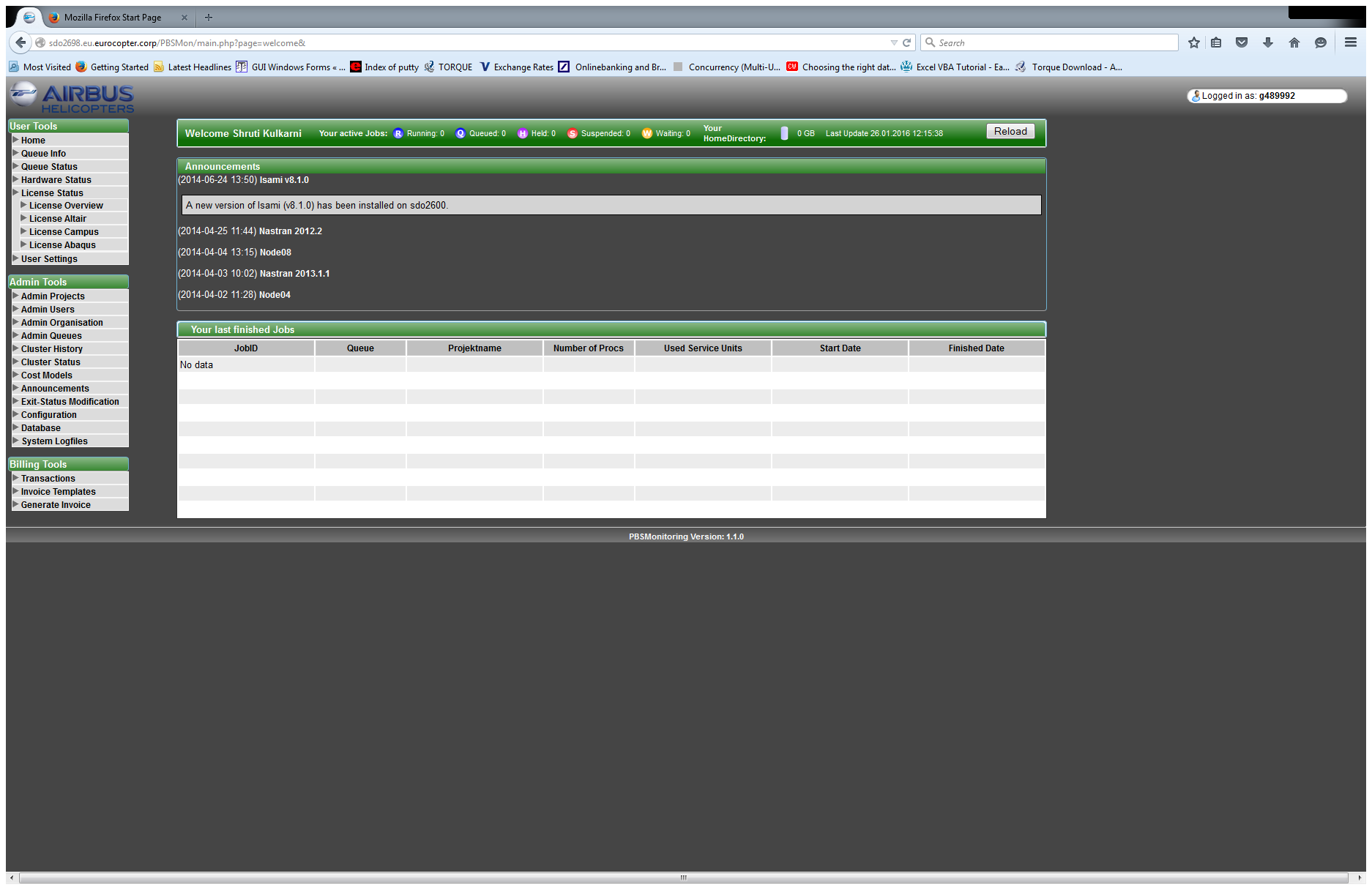
**Extension of PBS Monitoring architectural point:**

****

The <myJAM /> - Web front end is a highly interactive web application that for User und Administratoren gleichermaßen das zentrale Interface fur das Monitoring von Jobs, and administrators alike, the central interface for monitoring by Jobs, Warteschlangen oder des ganzen Clusters, fur die Verwaltung von Usern und Projekten, so- Queue or the entire cluster, for the management of users and projects, as well how to visualize current and historical analyzes and statistics. Das Frontend The frontendnutzt neben W3C-konformem XHTML [Pem02] auch objektorientiertes PHP5 [PHP], Ja- utilizes both W3C compliant XHTML also object oriented PHP5, JavaScript / AJAX und OpenFlashChart [Gla].vaScript / AJAX and Open Flash Chart.

Similarly, the central interface is used for monitoring by jobs, qWarteschlangen oder des ganzen Clusters, fur die Verwaltung von Usern und Projekten, sueues or the entire cluster, for the management of users and projects, as well how to visualize current and historical analyzes and statistics. The user submits a job on sdo2600 and visualized on sdo2699 or sdo2698 where the PBS Monitoring Tool is deployed with a connection to mySQL database server in common. The sdo2698 is specifically used for development server and is a cloned server of sdo2699. The project main focus is on the new development on the development server and automation of user operations for production server.

# System Software



# 6. Programming Environments and Tools

# 6.1. Programming Specification

1. The PBS tool allows the user to monitor the status of the running jobs on sdo2600 PBS Cluster.
2. The user can check the Queues info and Status, the node where the job is running, the requested memory …
3. The user is also able to check the Hardware status and available licenses.
4. The user can sort the jobs as per state i.e if the user wishes only to view the jobs in running state.
5. The user can delete user’s own job.
6. In Hardware Overview page, the option for refreshing the page each 5 seconds or 10 seconds.

# 6.2. Test Specification

1. Test Case 1: Sorting the status
2. Test Case 2: qdel
3. Test Case 3: qrls
4. Test Case 4: reloading the page each 5 or 10 seconds
5. Test Case 5: Licence Monitoring

# UML Diagrams

# Usecase diagram

# 

# Class diagram

Class Diagram.png

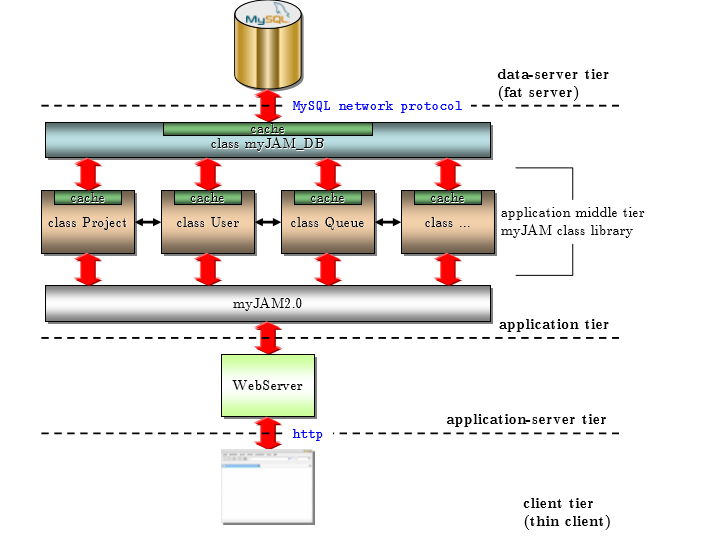
# Sequence diagram-Test Case 2-qdel

qdel_ac.png

# Sequence Diagram-TestCase 3-qrls

qrls.png

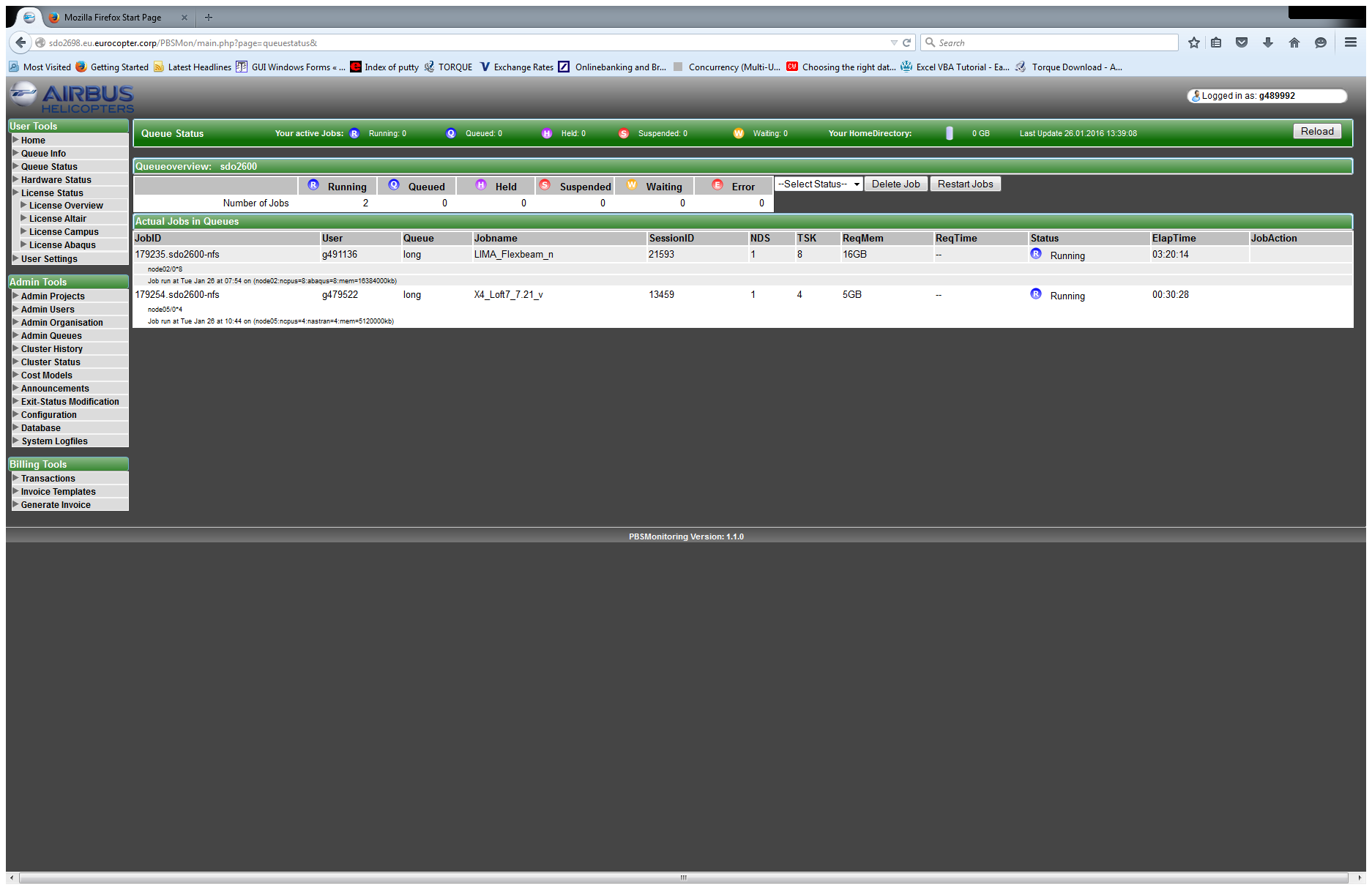
# Standard Deployment diagram

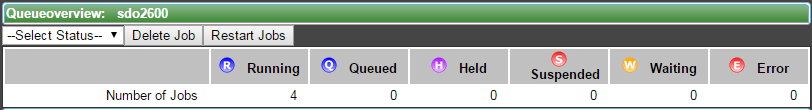


# Web Based Application

# 8.1 Test case 1: Sorting the Status

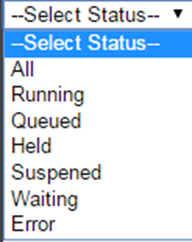
The Queue Status page is divided in 2 parts: The Queue Overview and Actual Jobs in Queues



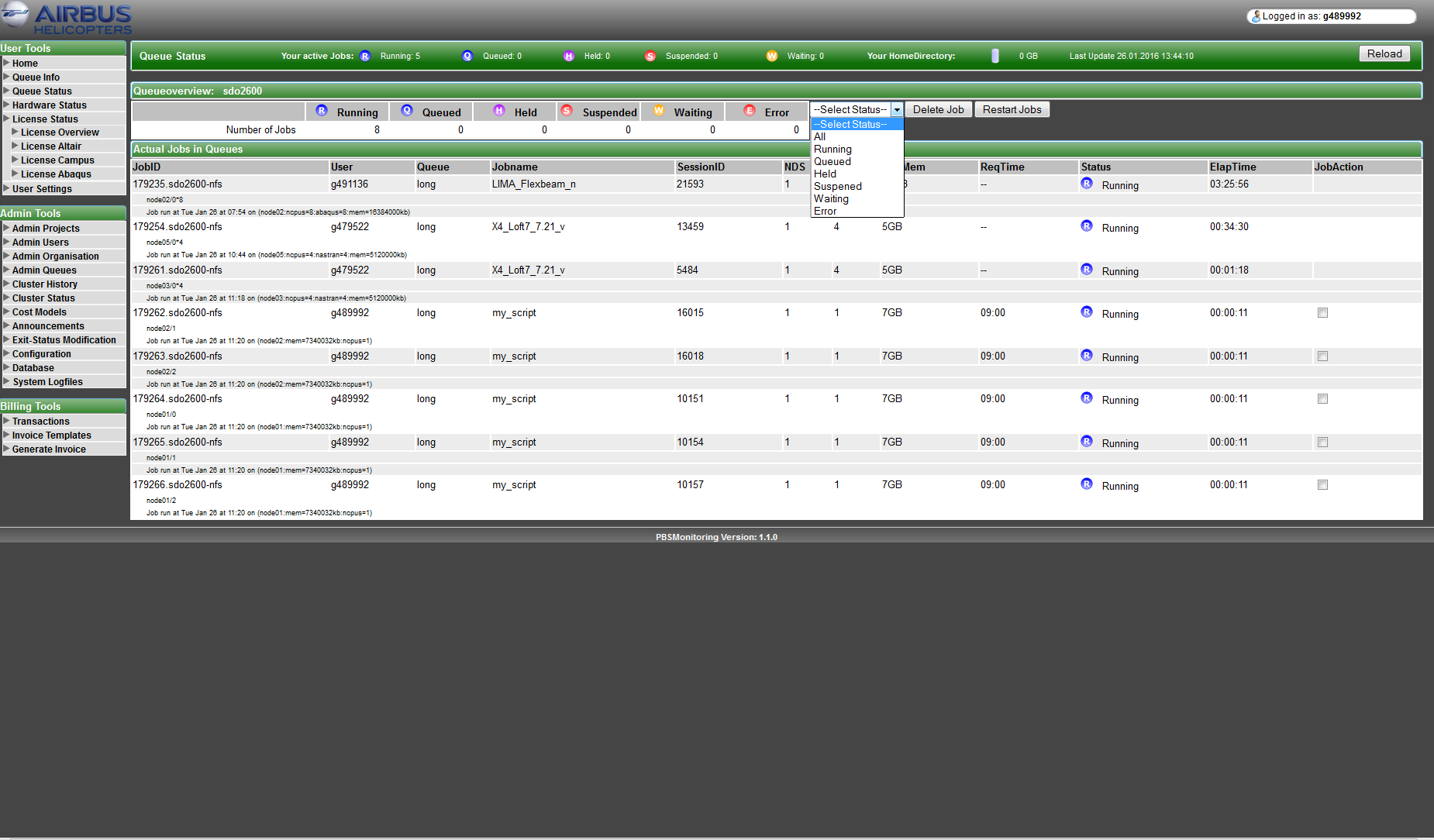
Queue Overview: Summarized the status of all the jobs submit through the Queuing System (PBS-Submit –c...)

Status Dropdown Box:

Sort the Actual jobs by it’s status

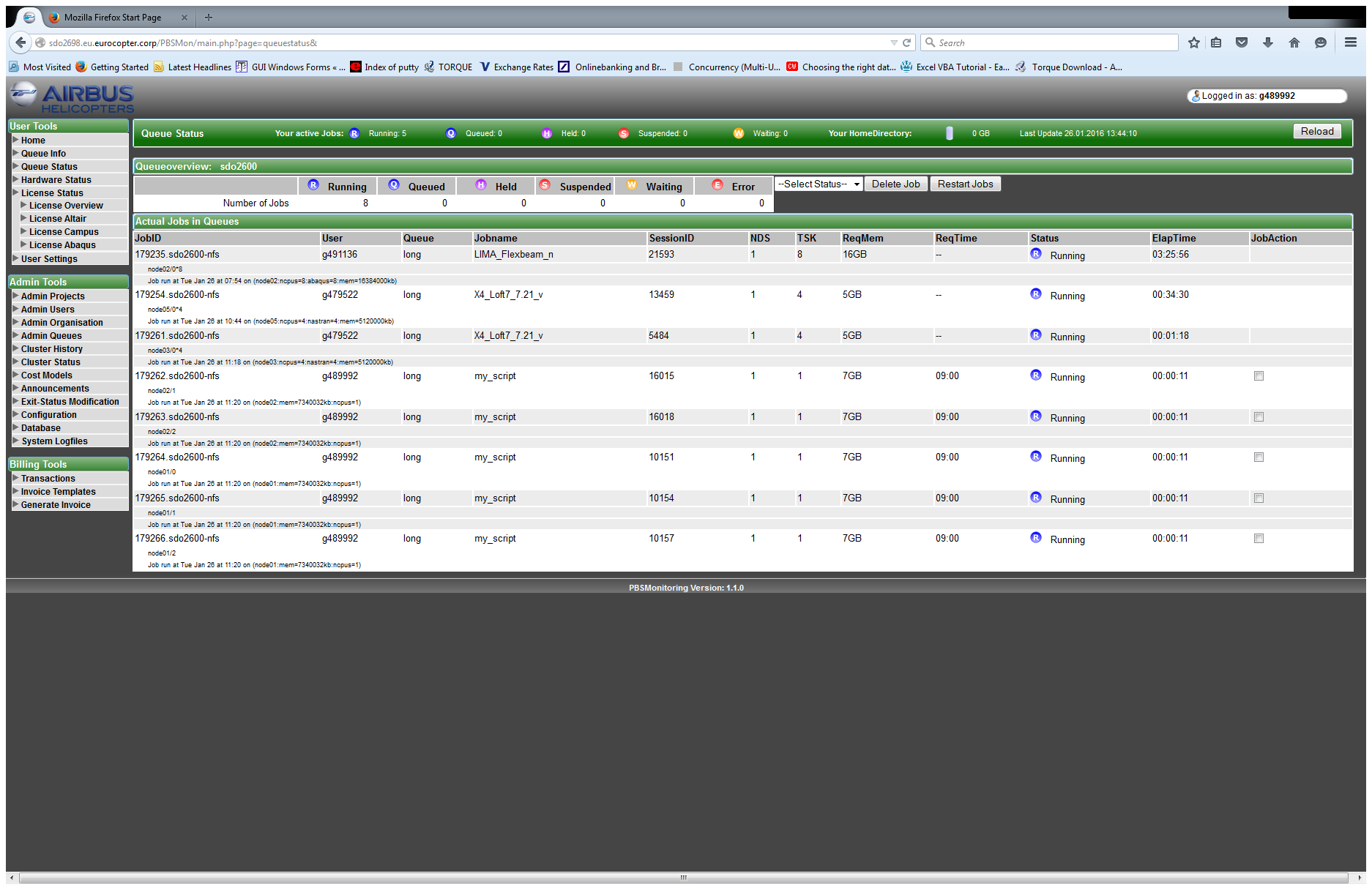
Options:-

1. All
2. Running
3. Queued
4. Held
5. Suspended
6. Waiting
7. Error

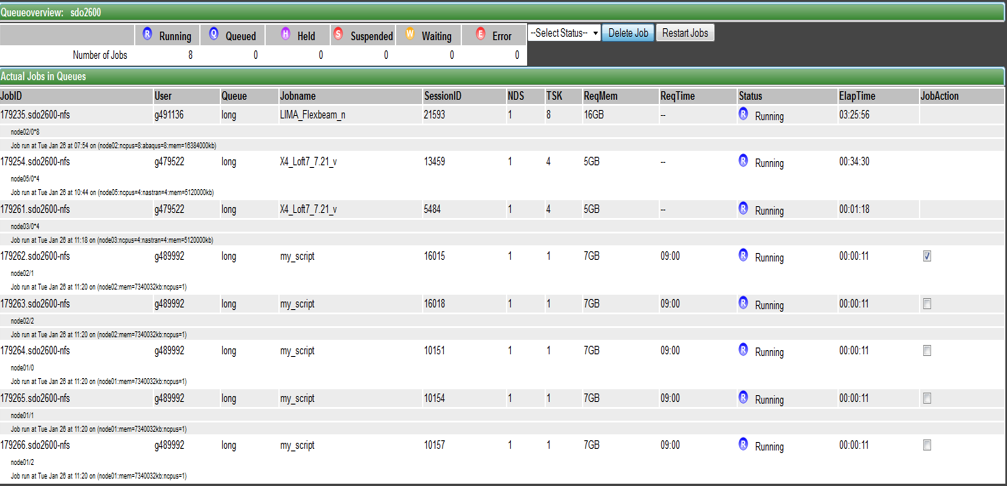


# 8.2 Test case 2: qdel

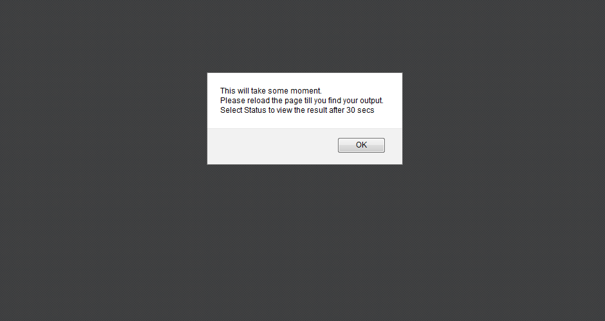
Below here an example, the user submitted 5 jobs using *qsub* my\_script.The jobs created by user will have checkbox and can perform operations only user owned jobs.

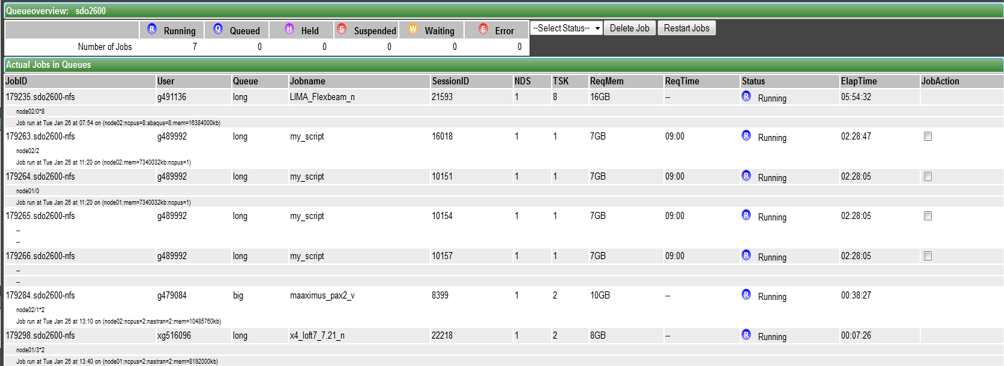


The user checks the specific job to delete and presses the delete button by which the job gets deleted with a notification of assigning 30 seconds of time to execute the deletion process.



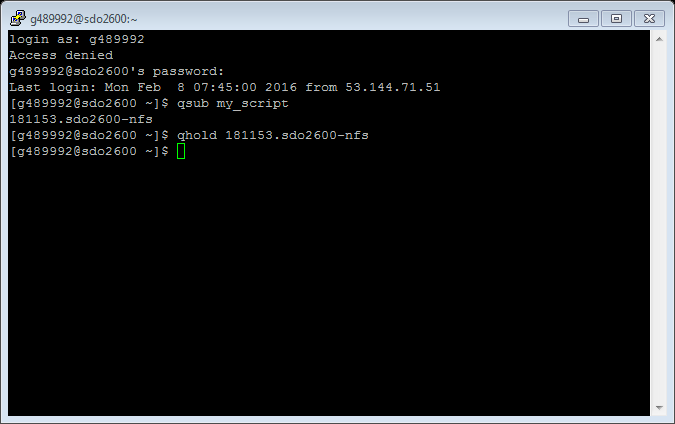
Test case 2 : qdel validation:

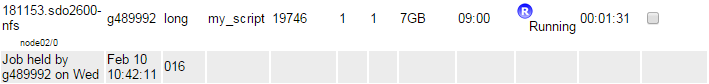


Therefore, the job is deleted and there are 4 jobs running instead of 5.

# 8.3 Test case 3: qrls

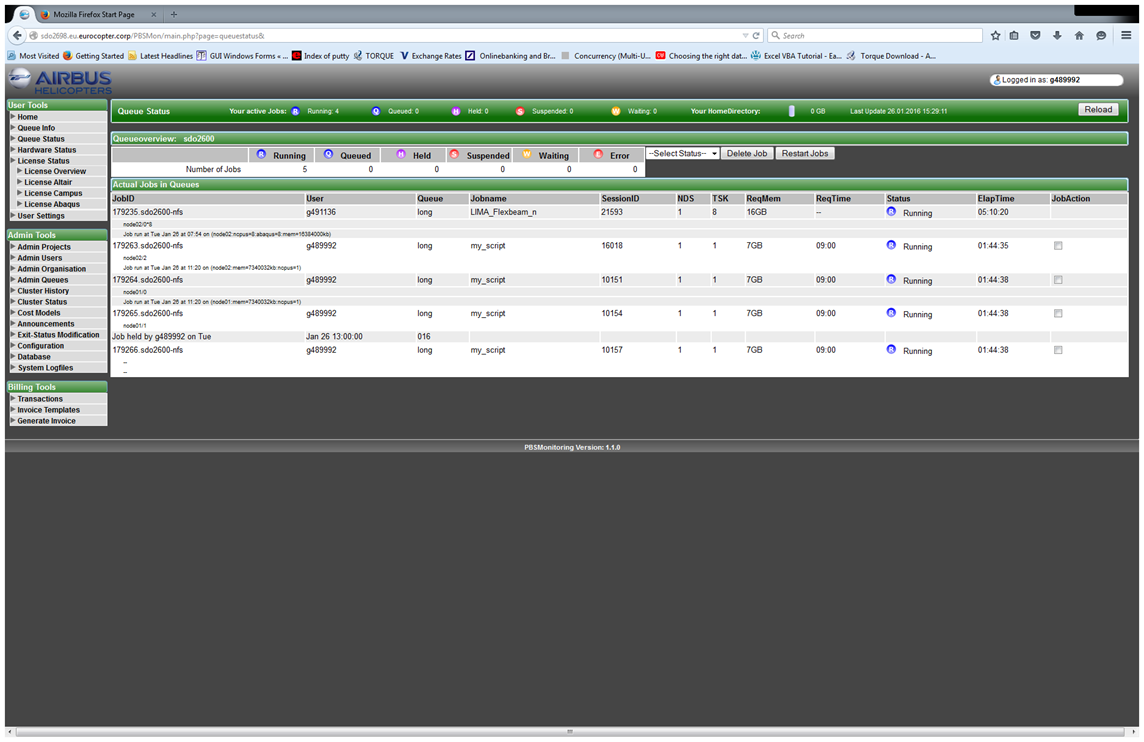
If the user executes the *qhold* command on a job then the job goes into the held state but on the PBS Monitoring ; the job status is still in running .To restart the job use the restart button which provides the*qrls* functionality.





After executing Restart button , the outcome shows two hypen (-).It is an indicator that the job is restarting.



****

# 8.4 Test case 4: reloading the page after each 5 or 10 seconds

The Hardware overview contains technical information about all available nodes. The displayed values are updated every minute. The Hardware overview is divided into the following groups:

– General

– CPU and Core

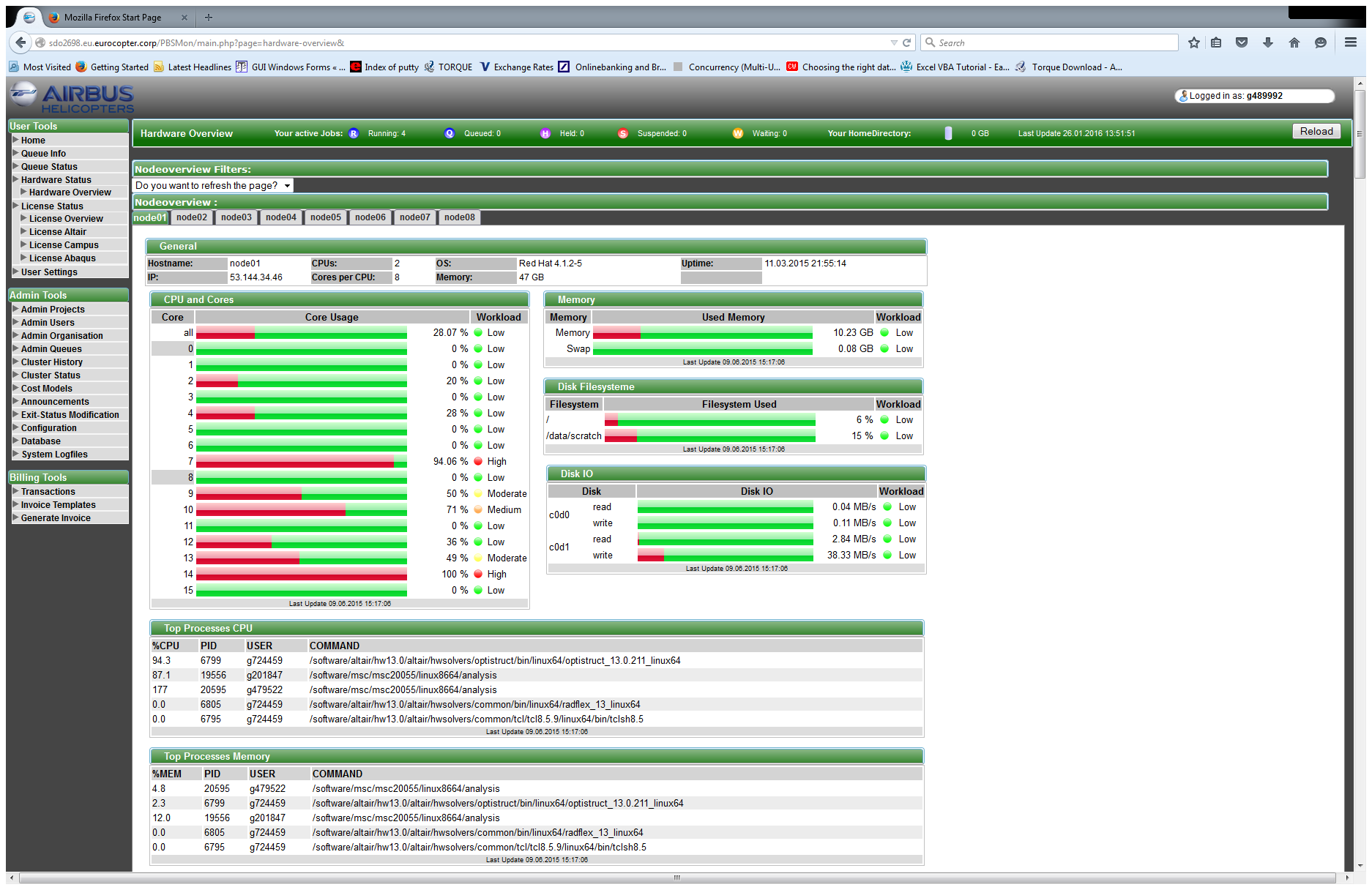
– Memory

– Disk Filesysteme

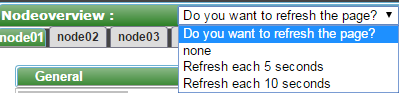
– Disk IO

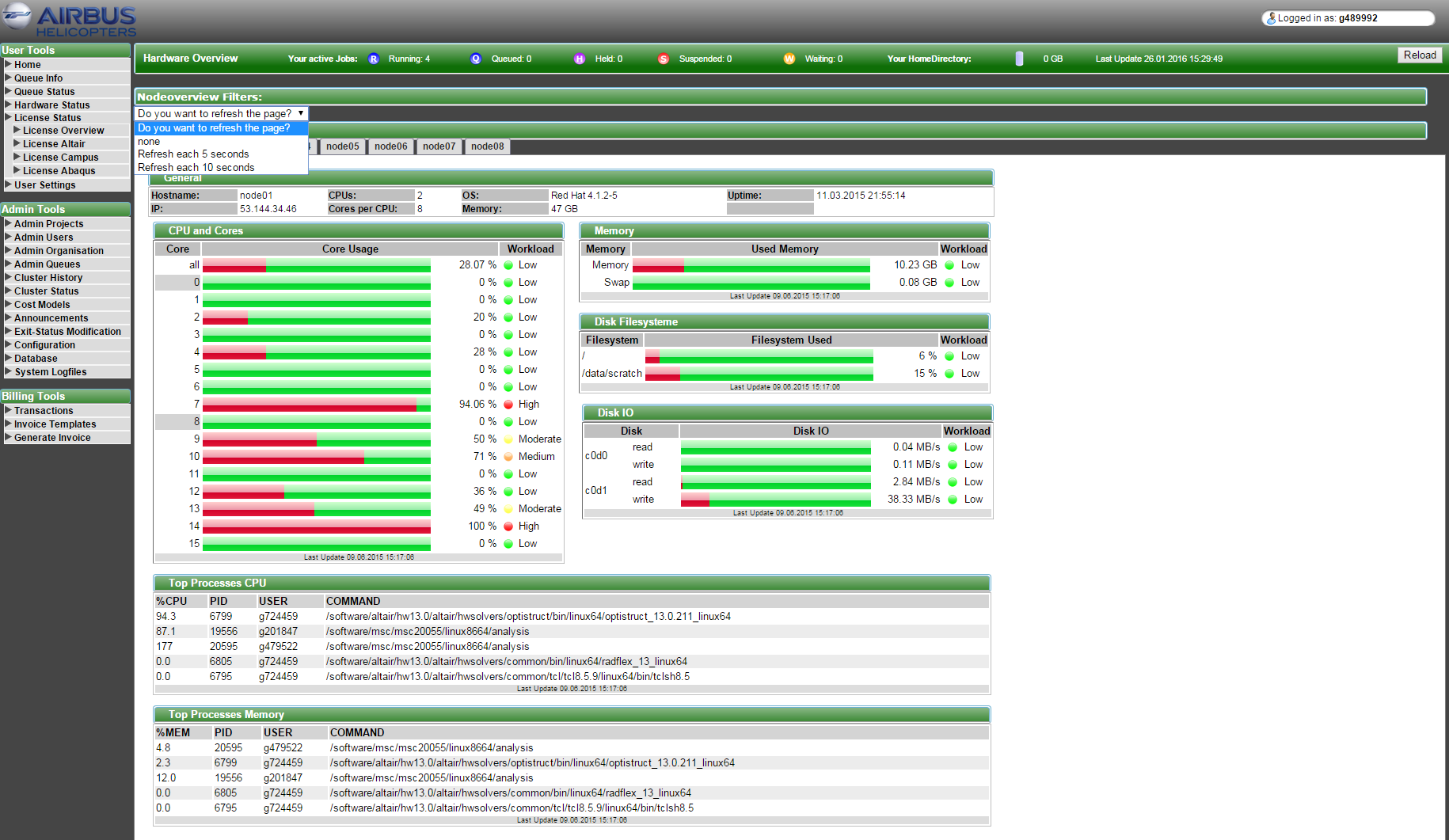
– TOP Processes CPU and TOP Processes Memory

Select the appropriate node panel (node01, node02, node03, node05, node06, node07) to have an overview of its performance.



The dropdown box for refreshing page is provided for each 5 seconds or 10 seconds. The option none refreshes after each 30 seconds. Google Chrome is recommended to be used because Mozilla Firefox and IE browsers generate in built pop’s within them.





# 8.5 Test Case 5: Licence Monitoring

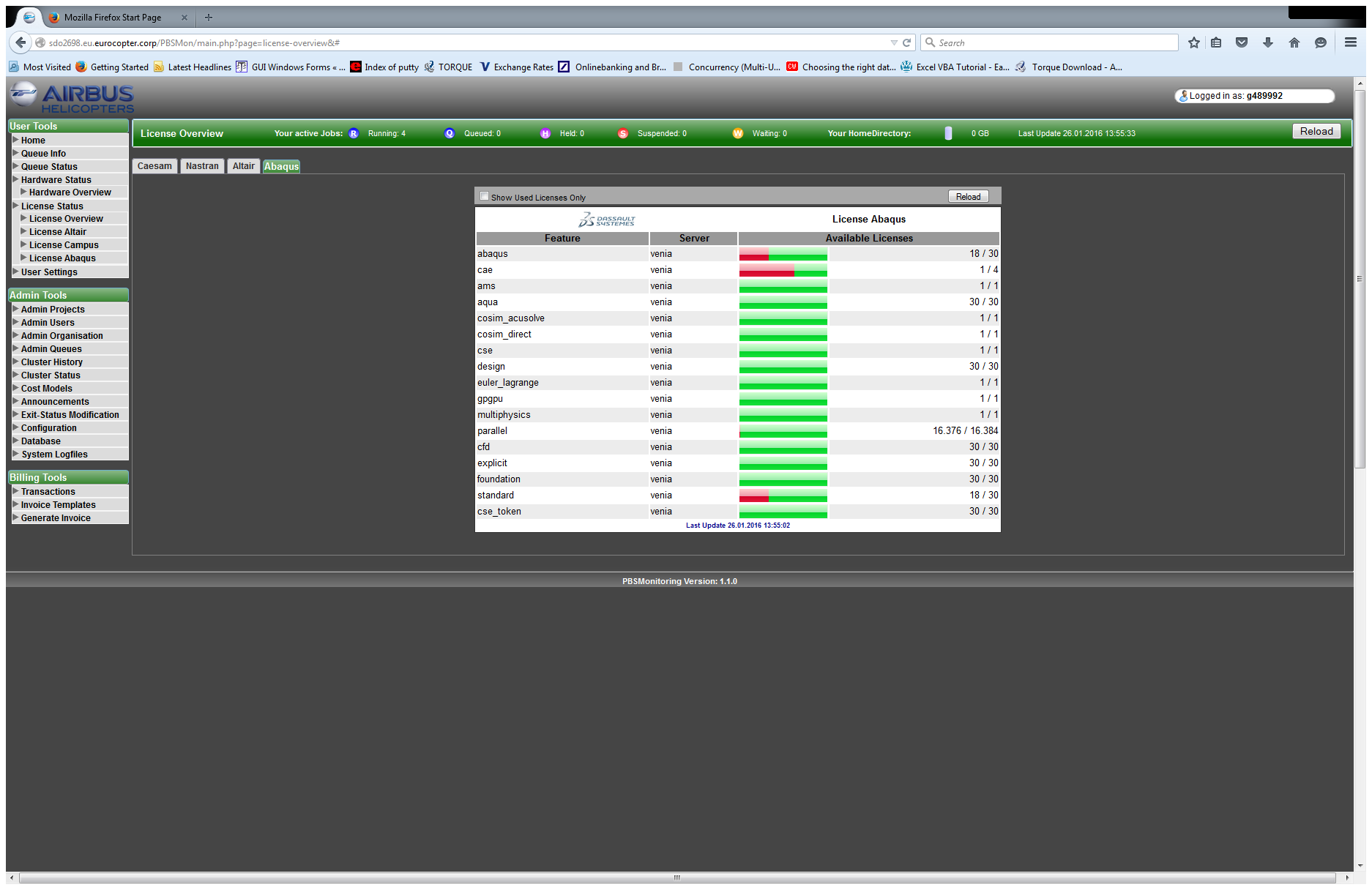
It is divided in 4 sub-sections: License Overview, License Altair, License Campus, License Abaqus. The license overview page provide all information about the most important license servers.

This information is updated every minute.By selecting the corresponding panel (Caesam, Nastran, Altair,Abaqus), you will see the available licenses on the server.You also have the possibility to check only the current used licenses by selecting the option Show used Licenses only.

License Altair: Provide a graphical representation of the used altair Licenses in a chosen period of time.

License Campus: Provide a graphical representation of the used MSC Licenses in a chosen period of time.

License Abaqus: Provide a graphical representation of the used Dassault Systems Licenses in a chosen period of time.



# Shell in PHP

**exec()**

**Syntax:**

string exec ( string $command [, array &$output [, int &$return\_var ]] )

exec() executes the given command.[[4]](#footnote-5)

1. Test Case 2:qdel

**PHP exec:**

foreach($value as $checkbox\_id)

{

echo $checkbox\_id;

$logged\_user=$\_SERVER['PHP\_AUTH\_USER'];

logged\_pwd=$\_SERVER['PHP\_AUTH\_PW'];

exec("/usr/bin/qdel $logged\_user $checkbox\_id");

}

**Shell Script qdel**

#!/bin/bash

#set -x

echo "command started with \"$\*\"" >/tmp/qdel.log

USER\_NAME=$1

JOBID=$2

if [ ! "$USER\_NAME" ] || [ ! "$JOBID" ]; then

echo "$0 <UserID> <JOBID>"

exit 10

fi

echo "Executing \"ssh $USER\_NAME@sdo2600 \"qdel $JOBID\"\"" >>/tmp/qdel.log

ssh 1>&2 2>>/tmp/qdel.log -vvv $USER\_NAME@sdo2600 "qdel $JOBID"

1. Test Case 3: qrls

**PHP exec:**

foreach($value as $checkbox\_id)

{

echo $checkbox\_id;

$logged\_user=$\_SERVER['PHP\_AUTH\_USER'];

logged\_pwd=$\_SERVER['PHP\_AUTH\_PW'];

exec("/usr/bin/qdel $logged\_user $checkbox\_id");

}

**Shell Script qdel**

#!/bin/bash

#set -x

echo "command started with \"$\*\"" >/tmp/qdel.log

USER\_NAME=$1

JOBID=$2

if [ ! "$USER\_NAME" ] || [ ! "$JOBID" ]; then

echo "$0 <UserID> <JOBID>"

exit 10

fi

echo "Executing \"ssh $USER\_NAME@sdo2600 \"qdel $JOBID\"\"" >>/tmp/qdel.log

ssh 1>&2 2>>/tmp/qdel.log -vvv $USER\_NAME@sdo2600 "qdel $JOBID"

# Software Validation

# 10.1 Exception Handling

1. Test Case 2: qdel

Exception Handling:

* Permission denied
* Httpd connect se linux -> on

[root@sdo2699 bin]# cat /tmp/qdel.log

command started with "g489992 180305.sdo2600-nfs"

Executing "ssh g489992@sdo2600 "qdel 180305.sdo2600-nfs""

OpenSSH\_5.3p1, OpenSSL 1.0.1e-fips 11 Feb 2013

debug1: Reading configuration data /etc/ssh/ssh\_config

debug1: Applying options for \*

debug2: ssh\_connect: needpriv 0

debug1: Connecting to sdo2600 [53.144.34.45] port 22.

debug1: connect to address 53.144.34.45 port 22: Permission denied

ssh: connect to host sdo2600 port 22: Permission denied

* Test successful

1. Test Case 3: qrls

Exception Handling:

* Permission denied
* Httpd connect se linux -> on

[root@sdo2699 bin]# cat /tmp/qdel.log

command started with "g489992 180305.sdo2600-nfs"

Executing "ssh g489992@sdo2600 "qdel 180305.sdo2600-nfs""

OpenSSH\_5.3p1, OpenSSL 1.0.1e-fips 11 Feb 2013

debug1: Reading configuration data /etc/ssh/ssh\_config

debug1: Applying options for \*

debug2: ssh\_connect: needpriv 0

debug1: Connecting to sdo2600 [53.144.34.45] port 22.

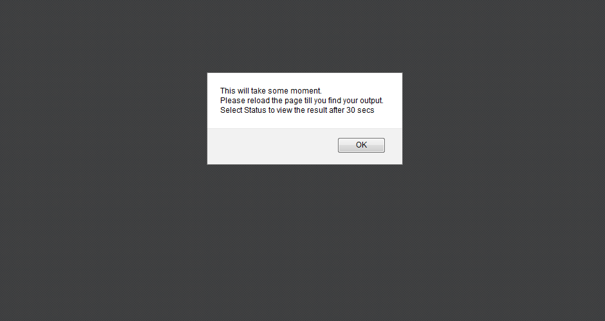
debug1: connect to address 53.144.34.45 port 22: Permission denied

ssh: connect to host sdo2600 port 22: Permission denied

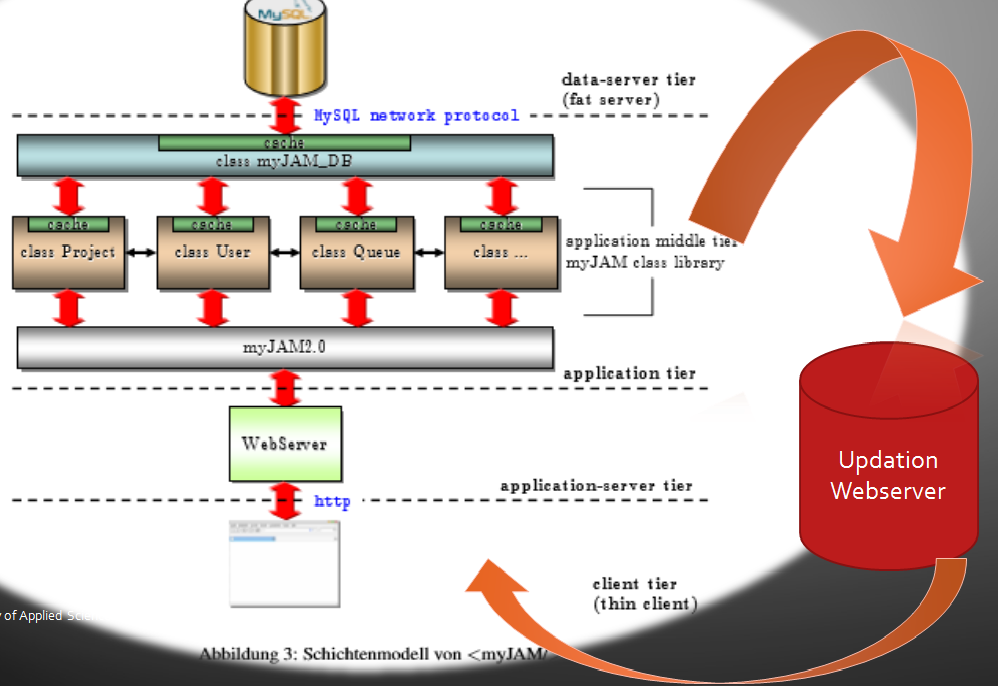
* Test successful

# 10.2 Java Script Validation

The entire project has only one validation which is provided only for TORQUE commands because the commands needs time to execute operation from user specific domain. The command is called then it execute from frontend PHP to shell script and then to server. As the third interface is created, it consumes 30 second time.



# Migration of Web based Application



The updation server in figure above is a cloned web based server with the web application deployed on it. The basic idea of cloning is to migrated the web application from development server to the production server. The updation server and the main server share the same database server. After the server development is done, it is migrated but later after the migration process is done there are various bugs immerged. The bugs are mentioned below

**Error on migration on to sdo2699 from sdo2698**

* **debug1: Connection established.**
* **debug1: identity file /var/www/.ssh/identity type -1**
* **debug1: identity file /var/www/.ssh/identity-cert type-1**
* **debug1: identity file /var/www/.ssh/id\_rsa type -1**
* **debug1: identity file /var/www/.ssh/id\_rsa-cert type -1**
* **debug1: identity file /var/www/.ssh/id\_dsa type -1**
* **debug1: identity file /var/www/.ssh/id\_dsa-cert type -1**
* **debug1: identity file /var/www/.ssh/id\_ecdsa type -1**
* **debug1: identity file /var/www/.ssh/id\_ecdsa-cert type-1**
* **debug1: Remote protocol version 2.0, remote software version OpenSSH\_4.3**

To overcome this exception the identity ,rsa ,dsa files are given access via ssh keygen. How to do it is explained in section 12.

# Shh Passwordless

To use Linux and OpenSSH to automate your tasks. Therefore you need an automatic login from host A / user a to Host B / user b. For passwordless , because to call ssh from a within a shell script. Log in on A as user a and generate a pair of authentication keys. Do not enter a passphrase:

a@A:~> ssh-keygen -t rsa

Generating public/private rsa key pair.

Enter file in which to save the key (/home/a/.ssh/id\_rsa):

Created directory '/home/a/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

The identification has been saved in /home/a/.ssh/id\_rsa.

The public key has been saved in /home/a/.ssh/id\_rsa.pub.

The key fingerprint is:

3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 a@A

Now use ssh to create a directory ~/.ssh as user b on B. (The directory may already exist):

a@A:~> ssh b@B mkdir -p .ssh

b@B's password:

Finally append a's new public key to b@B:.ssh/authorized\_keys and enter b's password one last time:

a@A:~> cat .ssh/id\_rsa.pub | ssh b@B 'cat >> .ssh/authorized\_keys'

b@B's password:

From now can log into B as b from A as a without password:

a@A:~> ssh b@B

Depending on the version of SSH to do the following changes:

Put the public key in .ssh/authorized\_keys2

Change the permissions of .ssh to 700

Change the permissions of .ssh/authorized\_keys2 to 640

# Bibliography

1. <https://www.dfn.de/fileadmin/3Beratung/DFN-Forum2/115.pdf>
2. PBS User Guide-http://www.bc.edu/offices/researchservices/cluster/torqueug.html
3. PBS Administration Guide-http://www.pbsworks.com/documentation/support/PBSProAdminGuide11.2.pdf
4. Linux Cheat sheets
5. php.net/manual/en/function.exec.php
6. <https://wiki.archlinux.org/index.php/TORQUE>
7. <http://php.net/manual/en/function.exec.php>
8. <http://www.tecmint.com/ssh-passwordless-login-using-ssh-keygen-in-5-easy-steps/>

# 14. The Annex

# The Annex-Introduction to Company

This chapter presents the organization andthe procedures AIRBUS HELICOPTERSDeutschland Airplane Door Systems has established to achieve Agency–approval as Engineeringorganization. The engineering responsibility in the frame ofthe Airbus Enginerring organization of theA380 pax doors and cargo doors wastransferred in 2002 to AIRBUSHELICOPTERS DEUTSCHLAND Donauwörth[[5]](#footnote-6).

In 1979 it was started with the engineering ofthe first AIRBUS A310 assemblies (shells, door frame shellsand floorframes) in Donauwörth.Since 1983, starting with the transfer ofengineering activities to Donauwörth, paxdoors, cargo doors and bulk cargo doorsfor Airbus aircrafts are supported and further developed, in the frame of the AirbusEnginerring Organization, in Airplane DoorSystems Engineering. In this time thefirst type of the Single Aisle family wasborn with the A320. Beside the emergency exit doors, the bulk cargo doors andthe cargo doors, for this A/C we have designed parts of the belly fairing. Based onthe doors and belly fairings for the A320development, we enhanced these doorsand fairings during the following years forthe A321, A319 and A318.In September 2009 the contract to develop and produce Passenger, Service andBaggage Doors for the Mitsubishi Regional Jet (MRJ) with Mitsubishi AircraftCorporation was signed.AIRBUS HELICOPTERSDEUTSCHLAND is responsible for theentire development and production whilecertification will be arranged byMitsubishi.In January 2014 the company was renamed to AIRBUS HELICOPTERS DEUTSCHLAND GmbH[[6]](#footnote-7).

The technologies used in the Enginerring Organisation include metal as well as nonmetal structures, e.g. Doors, Fairings,Crew-Rest-Containers and Stairs of the various Airplane models.The EngineeringOrganization (ETVE) prepares and **technically approves** the engineering data and compliance demonstration documents for

* Engineering Changes on products, parts and appliances,
* The engineering data for continued airworthiness of the product in effect of the engineering change and
* the engineering data for repair engineering of the product

that needs airworthiness approval.

Staffs of the following organizational units take part in the AHD EngineeringOrganization:

ETVEA Architecture & Integration  
ETVED Design   
ETVEI Tools & Support   
ETVEM Airworthiness, Safety & Reliability and Configuration Management  
ETVEP Engineering Project Management  
ETVER Research & Technology   
ETVES Stress, Fatigue & Testing   
EXDC Customer Service

1. https://en.wikipedia.org/wiki/PuTTY [↑](#footnote-ref-2)
2. http://www.rcc.uh.edu/hpc-docs/49-using-torque-to-submit-and-monitor-jobs.html [↑](#footnote-ref-3)
3. https://www.dfn.de/fileadmin/3Beratung/DFN-Forum2/115.pdf [↑](#footnote-ref-4)
4. http://php.net/manual/en/function.exec.php [↑](#footnote-ref-5)
5. ENTWICKLUNGSBETRIEBS-HANDBUCH [↑](#footnote-ref-6)
6. ENTWICKLUNGSBETRIEBS-HANDBUCH [↑](#footnote-ref-7)