PBS + Maui Scheduler

This web page serves the following purpose

- Survey, study and understand the documents about PBS + Maui scheduler.
- Carry out test drive to verify our understanding.
- Design schdeuling scenarios for different needs.
- Design rescuing standard operation procedure (SOP).

Contents

Survey

The first step for understanding Maui scheduler is to take a look at the project web site Maui Scheduler Home Page. We browse, survey and study the whole page and present the following summary of links for your reference. It may save your time while you are visiting this web page.

Document	What information provided?
Maui Scheduler Overview and Features	Present a general overview and features of Maui Scheduler. It's worth to spend a few mintues to read to get a bird's view of Maui Scheduler
SGI PBS power point introduction	Present an overview of Portable Batch System on SGI
OpenPBS Overview and Configuration	Contain detail explanation of concepts, architecture, and configurations of OpenPBS
Job Scheduling and Batch Systems (Linux Magazine Oct. 2002)	Give an introduction, installation, configuration, adiministrartion and running of OpenPBS on Linux.
Portable Batch System mini-HowTo	Give a step-by-step instructions of installation and configuration of PBS. Follow the steps, you will get a workable PBS environment
Portable Batch System	
OpenPBS Public Home	
Maui Scheduler for Linux Cluster	Present a concise overview of Maui system in PPT format
Job Scheduling with Maui (Linux Magazine Nov. 2002)	Give an introduction, installation, configuration, adiministrartion and running of Maui on Linux.
Maui Scheduler Open Cluster Software	1.FAQ(Why Java). 2.Maui Scheduler Molokini Edition (Java Distribution) 3.Give a detail description of the architecture of Maui in Java.
The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters	Present a very good introduction, architecture, mechanism and preformance of PBS+Maui system
Maui Scheduler Documentation	Includes installation, concepts and configuration
OSC Configuration and Tools for OpenPBS and Maui Scheduler	This page attempts to describe the configurations which OSC has made to OpenPBS and the Maui scheduler on the Center's Linux clusters, as well as some tools and scripts developed at OSC for making this combination easier to deal with. They are provided with the hope that they will be helpful to those at other sites who use these two software packages.
Maui Scheduler Administrator's Manual	Detail explanation of administartor's tasks with examples
Submitting Jobs to Maui Scheduler	Explain how to submit Jobs to Maui Scheduler by examples

CLIC Cluster Documentation	Contain very valuable information for PBS and Maui setup
The SDU Supercluster Horseshoe	A cluster example
SSS Resource Management and Accounting Downloads	Contain links to software and documentation download
Maui Scheduler Documentation	Contain links to Maui Adiministrator's guide and Users manual

Installation and Configuration

PBS

Installation

For front-end node or PBS server

1. Download and untar the source tarball.
2. configure
3. make
4. make install (/opt/pbs)

Once the code is built and installed on the front-end node, the pbs_mom and command components should be installed on the remaining cluster nodes.

Configuration

Runtime information is stored in the directory under $PBS_HOME, which is assumed to be /var/spool/pbs, the default location.

Before starting any daemons, a few configuration needed to be created or updated.

1. server_name: Each node needs to know what machine is running the server. This is converyed through the $PBS_HOME/server_name file. For Formosa, it should contain the line "srva0"
2. nodes: The pbs_server daemon must known which nodes are available for executing jobs. This information is kept in a file called $PBS_HOME/server_priv/nodes, and the file appears only on the node where the jobs server runs. The nodes should contain the lines

```
pca005 np=2
pca006 np=2
pca007 np=2
      .
      .
pca100 np=2
```

3. config: Each pbs_mom daemon needs some basic information to participate in the batch system. This configuration information is contained in $PBS_HOME/mom_priv/config on every node which pbs_mom daemon is running. The following lines should be in this file

```
$logevent       0x0ff
$clienthost     srva0
$restricted     *.cluster.nchc.org.tw
```

Scheduler Basics

Scheduling Iteration and Job Flow

Maui Scheduler Scheduling Flow Chart

Scheduling Iterations

- Update state information: Each iteration, the scheduler contacts the resource manager and requests up to date information on computer resources, workload, and policy configuration.
- Refresh reservations
- Schedule reserved jobs
- Schedule priority jobs
- Backfill jobs
- Update statistic
- Handle user requests: Any call requesting state information, configuration changes, or job or resource manipulation commands.
- Perform next scheduling cycle

Detailed Job Flow

- Determine basic job feasibility: The first step in scheduling is determining which jobs are feasible. This step eliminates jobs which have job holds in place, invalid job states (i.e. Completed, Not Queued, Deferred, etc), or unsatisfied preconditions.

- Prioritize jobs: With a list of feasible jobs created, the next step involves determining the relative priority of all jobs within that list. A priority for each job is calculated based on job attributes such as job owner, job size, length of time the job has been queued, and so forth.
- Enforce configured throttling policies: Any configured throttling policies are then applied constraining how many jobs, nodes, processors, are allowed on a per credential basis. Jobs which violate these policies are not considered for scheduling.
- Determine Resource Availability: For each job, Maui attempts to locate the required computer resources needed by the job. In order for a match to be made, the node must possess all node attributes specified by the job and possess adequate available resources to meet the TasksPerNode job constraint (Default TaskPerNode is 1)
- Allocate resources to job
- Distribute jobs tasks across allocated resources
- Launch job

As indicated in the steps above, an important part of this scheduling process is the concept of a reservation. Internally, the Maui Scheduler maintains a table of these reservations which include information such as when the reservation starts, when it ends, and who owns it. Reserved nodes maintain pointers to the corresponding reservation, as do jobs that have had a reservation created for them.

The use of reservations allows a job to maintain a guaranteed start time. This enables other jobs to use these nodes as long as they do not delay this job's start time.

The backfill step is simply the process of determining the best combination of jobs which can run without delaying any of the reservations that have been created.

Maui Scheduling Scenario and Setting

Golden Rule: A job's priority is the weighted sum of its activated subcomponents. The fundamental formula is list below

```
(Component Weight) * (Subcomponent Weight) * (Priority Value)
```

Notice 1: By default, the value of all component and subcomponent weights is set to 1 and 0 respectively. The one exception is the **QUEUETIME** component weight which is set to 1. This results in a total job priority equal to the period of time the job has been queued, causing Maui to act as simple FIFO.

Notice 2: MAX_PRIO_VAL=1000000000 (one billion)

Scenario1:

Strictly First-In and First-Out with backfill algorithm

```
# ------------------------- maui.cfg -----------------------------
SERVERHOST              pca150
ADMIN1                  root
RMCFG[base]             TYPE=PBS

RMPOLLINTERVAL         00:00:30

DEFERTIME               00:00:10

SERVERPORT              42559
SERVERMODE              NORMAL

LOGFILE                 maui.log
LOGFILEMAXSIZE          10000
LOGLEVEL                3

# CRED Compoent

CREDWEIGHT              0

# Service (SERV) Component :

QUEUETIMEWEIGHT         60

# Backfill

BACKFILLPOLICY        FIRSTFIT
RESERVATIONPOLICY     CURRENTHIGHEST
```

Scenario2:

Prioritized userA has the privilege to have submitted job to be executed ahead of other user's submitted job in 1 hour.

```
# ------------------------- maui.cfg -----------------------------
SERVERHOST              pca150
```

```
ADMIN1                  root
RMCFG[base]             TYPE=PBS

RMPOLLINTERVAL         00:00:30

DEFERTIME               00:00:10

SERVERPORT              42559
SERVERMODE              NORMAL

LOGFILE                 maui.log
LOGFILEMAXSIZE          10000
LOGLEVEL                3

# Priority Weights

CREDWEIGHT              1

USERWEIGHT              1
USERCFG[userA]          PRIORITY=3600

# SERVWEIGHT is always 1 by system definition.
QUEUETIMEWEIGHT         60      # cause jobs to increase in priority by
                                # 60 points for every minute

# Backfill: http://supercluster.org/mauidocs/8.2backfill.html

BACKFILLPOLICY        FIRSTFIT
RESERVATIONPOLICY     CURRENTHIGHEST

# Node Allocation: http://supercluster.org/mauidocs/5.2nodeallocation.html

NODEALLOCATIONPOLICY  MINRESOURCE
```

Scenario 3

userA belongs to QOS(quality of service)=highpriv and userB belongs QOS=common. Under the definition of QOS=highpriv, userA can submit at most 20 jobs and use 100 cpus with priority=1000. Similarily, userB can submit at most 4 jobs and use 10 cpus with priority=100.

```
# -------------------------- maui.cfg -----------------------------
SERVERHOST              pca150
ADMIN1                  root
RMCFG[base]             TYPE=PBS

RMPOLLINTERVAL         00:00:30

DEFERTIME               00:00:10

SERVERPORT              42559
SERVERMODE              NORMAL

LOGFILE                 maui.log
LOGFILEMAXSIZE          10000
LOGLEVEL                3

# Priority Weights

CREDWEIGHT              1

USERWEIGHT              1
USERCFG[userA]          QDEF=highprio
USERCFG[userB]          QDEF=common

QOSWEIGHT               1
QOSCFG[common]          PRIORITY=10 MAXJOB=4 MAXPROC=10
QOSCFG[highprio]        PRIORITY=1000 MAXJOB=20 MAXPROC=100

# SERVWEIGHT is always 1 by system definition.
QUEUETIMEWEIGHT         60      # cause jobs to increase in priority by
                                # 60 points for every minute

# Backfill: http://supercluster.org/mauidocs/8.2backfill.html

BACKFILLPOLICY        FIRSTFIT
RESERVATIONPOLICY     CURRENTHIGHEST

# Node Allocation: http://supercluster.org/mauidocs/5.2nodeallocation.html

NODEALLOCATIONPOLICY  MINRESOURCE
```

Scenario 4:

A more general scenario. For detail explanation, please see the comments below.

```
# -------------------------- maui.cfg -----------------------------
SERVERHOST              pca150
ADMIN1                  root
RMCFG[base]             TYPE=PBS

RMPOLLINTERVAL         00:00:30

DEFERTIME               00:00:10

SERVERPORT              42559
SERVERMODE              NORMAL

LOGFILE                 maui.log
LOGFILEMAXSIZE          10000
LOGLEVEL                3

# Priority Weights
# Credential (CRED) Component : The priority calculation for the credential
#       component is:
#
#       Priority += CREDWEIGHT  * (
#               USERWEIGHT      * J->U->Priority +
#               GROUPWEIGHT     * J->G->Priority +
#               ACCOUNTWEIGHT   * J->A->Priority +
#               QOSWEIGHT       * J->Q->Priority +
#               CLASSWEIGHT     * J->C->Priority )

CREDWEIGHT              1

USERWEIGHT              1
USERCFG[DEFAULT]        PRIORITY=50
USERCFG[userA]          PRIORITY=100
USERCFG[userB]          PRIORITY=200

GROUPWEIGHT             1
GROUPCFG[nchc]          PRIORITY=100 QDEF=highprio
GROUPCFG[chem]          PRIORITY=10 QDEF=common

QOSWEIGHT               1
QOSCFG[common]          PRIORITY=33 MAXJOB=4 MAXPROC=10
QOSCFG[highprio]        PRIORITY=1000 MAXJOB=20

CLASSWEIGHT             1
# up to 5 jobs submitted to the class "serial" will be allowed to execute
# simultaneously and will be assigned the QOS highprio by default.
CLASSCFG[serial]        MAXJOB=5 QDEF=highprio
CLASSCFG[n1]            QDEF=highprio

# Service (SERV) Component :

QUEUETIMEWEIGHT         60

# Backfill: http://supercluster.org/mauidocs/8.2backfill.html

BACKFILLPOLICY        FIRSTFIT
RESERVATIONPOLICY     CURRENTHIGHEST

# Node Allocation: http://supercluster.org/mauidocs/5.2nodeallocation.html

NODEALLOCATIONPOLICY  MINRESOURCE
```

Reservation Setting

Every reservation consists of 3 parts, a set of resource, a timeframe, and an access control list.

Case 1:

Reserve pca010, pca011, pca012, pca013 to user mike from 18:00:00 to 24:00:00 on September 2.

In this case:

- Resource: pca010, pca011, pca012, pca013
- Access Control List (ACL): mike
- Timeframe: 18:00:00_09/02 - 24:00:00_09/02

Command:

```
# setres -s 18:00:00_09/02 -e 24:00:00_09/02 -u mike 'pca01[0-3]'
```

Return: reservation 'mike.1' on 4 nodes

Case 2:

Mike can use the following command to use his reserved nodes:

```
qsub -W x="FLAGS:ADVRES:mike.1 job.sh
```

Rescuing Standard Operation Procedure (SOP)

If some abnormal behavior of PBS+Maui is encountered, the first step is to check the required daemons and the configuration files.

Prerequisite Checking List

1. Checking whether `pbs_server` and `maui` daemons are alive on `srva0` by issuing the following commands

   ```
   srva0:~# ps -e | grep pbs_server
   srva0:~# ps -e | grep maui
   ```

   If one of or both daemons is not alive, restart it by issuing the following commands

   ```
   srva0:~# /etc/rc2.d/S98pbs_server start
   srva0:~# /etc/rc2.d/S99maui start
   ```

2. Checking the configuration files of PBS+Maui on `srva0`
   - Checking `/var/spool/pbs/server_name`, the typical content should be

     ```
     srva0
     ```

   - Checking `/var/spool/pbs/server_priv/nodes`, the typical content should be

     ```
     pca005 np=2
     pca006 np=2
     pca007 np=2
     ...
     pca150 np=2
     ```

   - Checking `/var/spool/pbs/server_priv/acl_svr/acl_hosts`, the typical content should be

     ```
     *
     ```

   - Checking `/var/spool/maui/maui.cfg`, please see Maui Scheduling Scenario and Setting for correct scheduler setting.
3. Checking whether `pbs_mom` is alive on each computation nodes, for example, checking `pca005`

   ```
   pca005:~# ps -e | grep pbs_mom
   ```

   If the daemon is not alive, then restart it by issuing the following command

   ```
   pca005:~# /etc/rc2.d/S99pbs_mom start
   ```

   Note That

   For convenience, you may use your favor cluster tool (C3) or your own cluster tool to scan the whole cluster to find and fix problems effectively and efficiently.

4. Checking the configuration files of PBS on each computational node
   - Checking `/var/spool/pbs/server_name`, the typical content should be

     ```
     srva0
     ```

   - Checking `/var/spool/pbs/mom_priv/config`, the typical content should be

     ```
     $clienthost srva0
     $logevent 0x63
     $restricted *.cluster.nchc.org.tw
     ```

Diagnosing and Solving Problem

If abnormal behavior still exist after go through the Prerequisite Checking List. Then proceed to cross-examine the following files to find the possible causes.

- Server side: /var/spool/pbs/server_logs, /var/spool/maui/log/maui.log
- Client side: /var/spool/pbs/mom_logs

The following is some possible causes and how to solve them.

Job Cannot be Executed

Scenario:

Job is queued but not running though there is still enough free resource for satisfying the request.

Diagnosis and Solution:

- Obtain the list of nodes that have been allocated to the job by issuing the following command

  ```
  /opt/maui/bin/checkjob $JOB_PID
  ```

- Examine the log files or error message on the node list.
- Check the availability of each node in the allocated node list,
  - If the node is down, restart the pbs_mom daemon.
  - If the loading is too high, kill illegal processes.
  - If disk is full, delete unnecessary files.
  - If sshd is dead, restart it

Inconsistance of Job's Status

Scenario:

From Maui `showq` command, a job is running but in the other hand, from PBS `qstat`, it is still queued.

Diagnosis and Solution:

- Find out which node block the Job, by using `checkjob` to find the leader of the job.
- Look at leader's /var/spool/pbs/mom_log/`date` to check the block node's hostname
- Login the node and check the content of the "config" file, located at /var/spool/pbs/mom_priv
- The file looks something like the follows:

  ```
          $logevent 0x1ff
          $clienthost master
          $clienthost slave1
          $clienthost slave2
          $clienthost slave3
          $clienthost slave4
          $max_load  2.0
  ```

- Add one line to the "config" file if it does not exist

  ```
          $clienthost $leader_hostname
  ```

- Restart pbs_mom daemon

Non Delivery of Output

Scenario:

If the output of a job is not delivered to the user, it is typical saved in the directory PBS_HOME/undelivered and the job PID still exist and the status is marked as 'E'.

Diagnosis and Solution:

- If the specified destination is not writable, change the file mode to writable.
- The PBS spool directory on the execution host does not have the correct permission mode 1777 (drwxrwxrwx).
- The output file is too big to transfer to the destination directory.

Unable to Remove Zombie Jobs from PBS Queue

Scenario:

Cannot remove a zombie job from pbs queue by issuing the termination commands such as

```
/opt/maui/bin/canceljob $PID
```

or

```
/opt/pbs/bin/qdel $PID
```

Diagnosis and Solution:

Diagnosis

The pbs_mom was supposed to run a new job but never got the command to actually carry it out. On the other hand pbs_server thought that pbs_mom has the job and already carried it out. This inconsistance causes the job's status shown running but actually didn't get running on the client nodes. If such a situation occurs, the zombie job cannot be removed by the ordinary terminated commands.

Solution

- Login PBS server
- Stop pbs_server daemon by issuing the following command

  ```
  /etc/rc2.d/S98pbs_server stop
  ```

- Delete the jobs file in /var/spool/pbs/server_priv/jobs

  ```
  cd /var/spool/pbs/server_priv/jobs
  rm -rf $PID.JB
  rm -rf $PID.SC
  ```

- Start pbs_server daemon by issuing the following command

  ```
  /etc/rc2.d/S98pbs_server start
  ```

- Delete the jobs files in /var/spool/pbs/mom_priv/jobs on the client nodes.

  ```
  rsh $ClientNode
  cd /var/spool/pbs/mom_priv/jobs
  rm -rf $PID.JB
  rm -rf $PID.SC
  ```