

# SLURM Database Use Accounting and Limits



SLURM Users Group Meeting  
October 2012

Danny Auble  
da@schedmd.com

SchedMD LLC

SchedMD LLC  
<http://www.schedmd.com>

# Outline



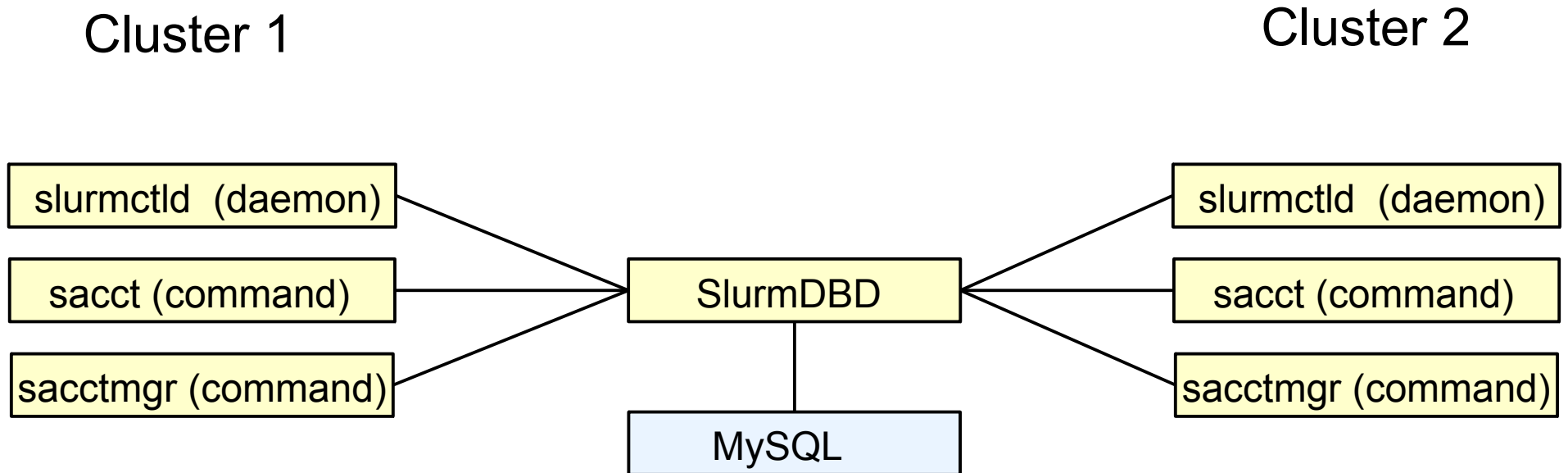
- System architecture for database use
- Definitions
- Accounting commands
- Resource limits
- Accounting configuration

# Architecture



- It is recommended to maintain one database containing the information about all computers and users at a site
  - One database per cluster is possible, but increases the maintenance effort and eliminates the multi-cluster option.
- MySQL is the only fully supported option
- Data maintained by user name
  - A uniform mapping of user name to ID across all computers is strongly recommended

# Architecture



# SlurmDBD

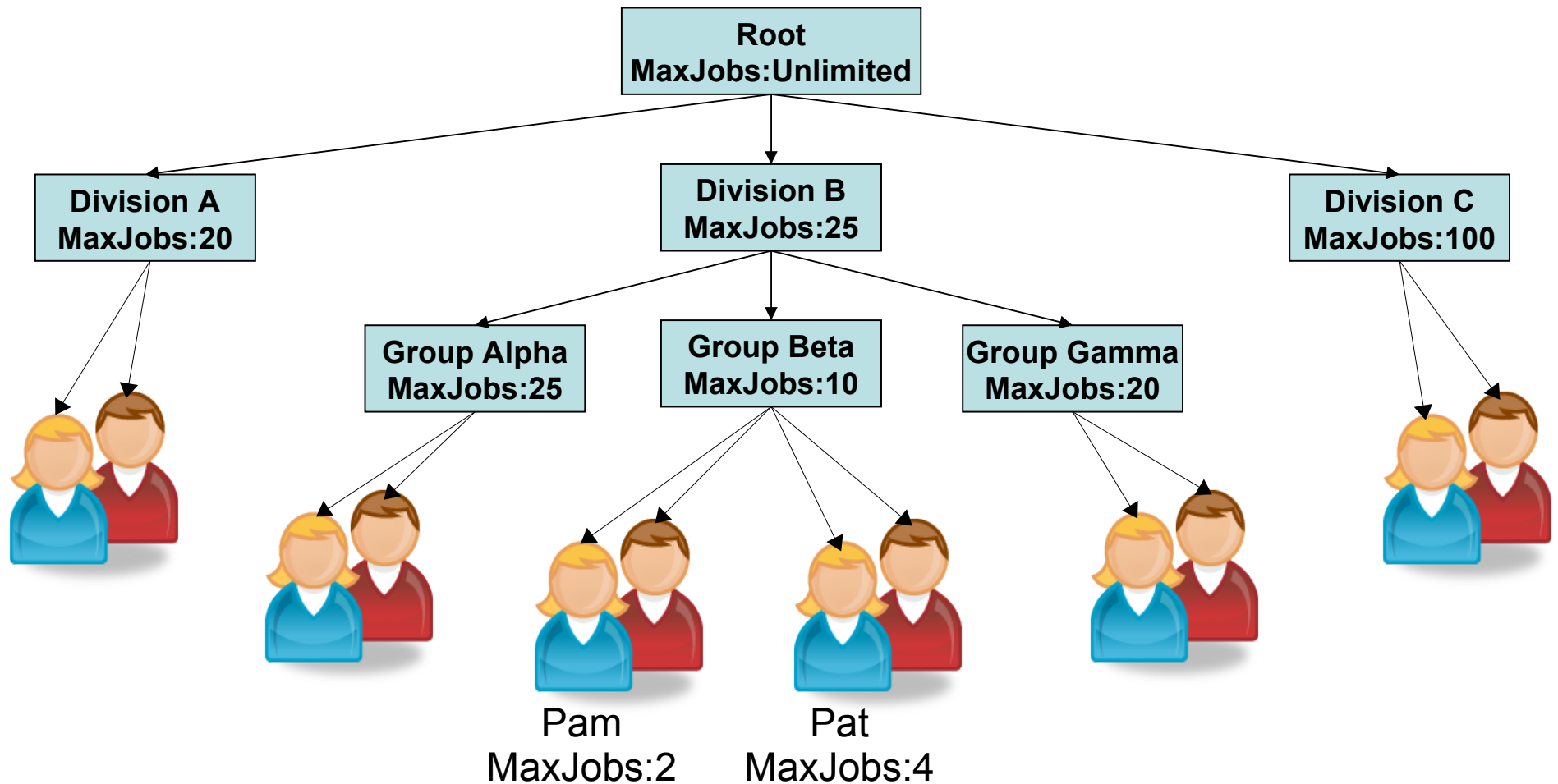
- SlurmDBD == SLURM DataBase Daemon
- An intermediary between user and the database
  - Avoid granting users direct database access
  - Authenticate communications between user and slurmdbd (using Munge)
  - Only slurmdbd needs permission to read/write the database
- Pushes update information out to slurmctld on the clusters
- slurmctld daemon will cache data if slurmdbd not responding

# Association

- Association is a combination of cluster, account, user name and (optional) partition name
- Each association can have a fair-share allocation of resources and a multitude of association specific and group (association + children) limits
- NOTE: Each account name must be unique. The name can not be repeated at different points in the hierarchy

User=Pam Account=Beta FairShare=20% MaxTime=2hours MaxJobs=2 etc.

# Hierarchical Account Example



# sacctmgr Command

## Part 1



- sacctmgr used by any user to view and by privileged users to modify the SLURM database
  - Manage clusters
  - Manage accounts
  - Manage users
  - Manage QOS
  - Manage association/QOS limits
  - Manage fair-share



# sacctmgr Command

## Part 2

- sacctmgr user AdminLevel definitions
  - None – regular user, no special privileges
  - Operator - can add, modify, and remove any database object (user, account, etc), and add other operators.  
On a SlurmDBD served slurmctld these users can
    - View information that is blocked to regular users by a PrivateData flag
    - Create/Alter/Delete Reservations
  - Admin – Same privileges as operator for the database. Same privileges as SlurmUser or root for the slurmctld.

# Account Coordinator



- Users given permission to add users or sub-accounts, modify fair-share and limits to the accounts and users they are coordinator over
  - Limits of their association can not be increased, but child associations (users and sub-accounts) of that association can increased up to that limit

# sacctmgr Examples of Use

```
sacctmgr add cluster tux
```

```
sacctmgr add account science Description="science" Organization=science
```

```
sacctmgr add account chemistry,physics parent=science \  
Description="physical sciences" Organization=science
```

```
sacctmgr add user adam DefaultAccount=chemistry
```

```
sacctmgr show associations
```

```
sacctmgr modify user adam account=chemistry set MaxJobs=2
```

# Other Commands Accessing Database

- `sacct` – Generates detailed accounting information about individual jobs or job steps
  - Filtering options by user, computer, partition, time, etc.
- `sreport` – Generates aggregated accounting reports
  - Reports resource usage by Cluster, Job, Reservation, or User
  - Data is based on conglomerate data not individual jobs or steps

# Resource Limits



- All “Max” limits given to a parent association will be inherited by it's children where the limit hasn't been set
  - An admin can set a child higher than it's parent
- Association-level (Group) limits: Applies to an association and all children (e.g. account Beta)
  - GrpCPUMins, GrpCPUs, GrpJobs, GrpMemory, GrpNodes, GrpSubmitJobs, GrpWall, etc
- Can be used to set limits on a finer-grained basis than SLURM partition limits

# Configuration – slurm.conf

## Part 1



- ***ClusterName=...***
  - Name of cluster for accounting purposes
- ***JobAcctGatherType=jobacct\_gather/linux***
  - Define how to gather job accounting information. Only required if step metrics (cpu, mem, etc) is desired.
- ***JobCompType=jobcomp/none***
  - Define where to record job completion data
  - Redundant if job accounting enabled
- ***TrackWckey=no***
  - A wckey is an orthogonal way to do accounting against maybe a group of unrelated accounts.

# Configuration – slurm.conf

## Part 2

- ***AccountingStorageType=accounting\_storage/slurmdbd***
  - Define where to record job accounting data
- ***AccountingStorageHost=...***
  - *The name or address of the host where SlurmDBD executes*
  - Defaults to localhost
- ***AccountingStoragePort=...***
  - *The network port that SlurmDBD accepts communication on.*
  - Defaults to 6819
- ***AccountingStoragePass=...***
  - If using SlurmDBD with a second MUNGE daemon, store the pathname of the named socket used by MUNGE to provide enterprise-wide authentication (i.e. /var/run/munge/moab.socket.2). Otherwise the default MUNGE daemon will be used.

# Configuration – slurm.conf

## Part 3

- **AccountingStorageEnforce=...**
  - **Associations** – prevent the running of job unless user and account defined in the database
  - **Limits** – prevent user from exceeding user or account limits. Automatically sets associations to be enforced
  - **Wckey** – This will prevent users from running jobs under a wckey that they don't have access to. By using this option, the 'associations' option is automatically set. The 'TrackWCKey' option is also set to true.
  - **QOS** – Require all jobs to use valid QOS (Quality Of Service). Jobs must specify QOS or use their default. Limits must be set to enforce qos limits
  - **All** – all of the above



# Configuration – slurmdbd.conf

## Part 1

- ***AuthType=auth/munge***
  - Define how authenticate communications
- ***StorageType=accounting\_storage/mysql***
  - Define where to record job accounting data
- ***StoragePort=...***
  - Defaults to the mysql default (3306)
- ***StorageUser=...***
  - User name used to access the database
- ***StoragePass=...***
  - Password used to access the database (can be blank)
- ***PrivateData=...***
  - Limits access to accounting information to who can query what from the database.
  - Current limits: account, job, reservation, usage, user ('all' for all)

# Configuration – slurmdbd.conf

## Part 2

- **Options to Purge elements.** All take format of #time, i.e. 2months. Default is all elements are preserved indefinitely.
  - *PurgeJobAfter*
  - *PurgeStepAfter*
  - *PurgeEventAfter*
  - *PurgeSuspendAfter*
- **Options to Archive elements.** All are booleans – default = no
  - *ArchiveJobs*
  - *ArchiveSteps*
  - *ArchiveEvents*
  - *ArchiveSuspend*
- **ArchiveScript**
  - This script can be executed every time a roll-up happens (every hour, day and month), depending on the Purge\*After options.
- **ArchiveDir**
  - If ArchiveScript isn't set this is the directory where all archive files will be placed after a purge event occurs

# Upgrading



- slurmdbd can communicate with SLURM commands and daemons at the same or recent earlier versions (slurmdbd v2.5 can communicate with version 2.4 or 2.3, slurmdbd v2.4 will not recognize v2.5 RPCs)
- ALWAYS UPGRADE SLURMDBD FIRST

# More Information



- SLURM documentation on line at:

<http://www.schedmd.com/slurmdocs/accounting.html>

[http://www.schedmd.com/slurmdocs/resource\\_limits.html](http://www.schedmd.com/slurmdocs/resource_limits.html)

<http://www.schedmd.com/slurmdocs/qos.html>