Slurm account synchronization with UNIX groups and users

Dr. Ole Holm Nielsen
Department of Physics
Technical University of Denmark
Ole.H.Nielsen@fysik.dtu.dk
Motivation

- Keeping Slurm accounts consistent with a site's user database is a well-known challenge, and there are very few publicly available tools for this purpose.

- We propose to use the already existing UNIX passwd and group information to define a mapping onto the Slurm account tree hierarchy.
Related work


- Define an account hierarchy in YAML files in *Git*.
- Map LDAP groups to Slurm *accounts*.
- Requires an LDAP/AD infrastructure (large organizations!).
- Open Source code: https://c4science.ch/source/slurm-accounts/
Keeping Slurm accounts consistent with a site's user database

- **On the one hand:** Users must be created in the Linux/UNIX system `passwd` database with a primary UNIX group `GID`:
  - username:password:UID:GID:GECOS:directory:shell

- **On the other hand:** Slurm employs “users” and “accounts” in the SlurmDB to define user access. Management with `sacctmgr`.

- Typically a Slurm `account` hierarchy may be defined as:
  - root->organization->department->group->user
    - SlurmDB basic entity is an Association=(User,Cluster,Partition,Account)

- The **Slurm username** must be == Linux username (as defined by an UID).

- A **Slurm account** is a “bank account” which may be used to aggregate users.
Operational challenges

- When system `passwd` and `group` databases change, how do we synchronize this onto the Slurm accounts?
- Can a user be a member of multiple Slurm accounts *(one-to-many membership)*?
- How do we administer Slurm accounts with respect to Slurm user limits, fairshare, and other factors?
- Are there any Open Source tools for such administration? *(Example: The EPFL project).*
This work: A simple Slurm account strategy

- Assume that the `passwd` and `group` system databases contain the authoritative user and `group` information for both Slurm as well as the system!
- Create a mapping of system UNIX groups onto Slurm accounts.
- Assign Slurm account names equal to the UNIX group names.
- User’s Primary UNIX group becomes his Slurm Default account.
- This is a natural “KISS” strategy because users and groups would likely be created on the system in the same way that they would be created in Slurm.
Our Slurm account solution

- Project on GitHub: https://github.com/OleHolmNielsen/Slurm_tools/tree/master/slurmaccounts

- Additional design choices:
  - If the user is a member of any secondary UNIX groups, the user is also added to these groups’ corresponding Slurm account.
    - This is a one-to-many mapping of users onto accounts!
  - Not supported: Setups where a single UNIX group is mapped onto multiple Slurm accounts.
  - Not supported: Per-user UNIX groups (where the primary groupname = username).
Setup example

- User `aaa` has the **primary** UNIX *group* `groupaaa`.
- A similarly named Slurm *account* “`groupaaa`” has been created.
- User `aaa` is a member of secondary UNIX *groups* `group1` and `group2`.
- Slurm *accounts* “`group1`” and “`group2`” have also been created.
Slurm account hierarchy config file

• How do we define the account hierarchy in a simple plain-text file?
• Inspired by the Slurm topology.conf file, define the /etc/slurm/accounts.conf file (5 fields separated by colons):

  account_name:parent_account:fairshare_value:Description_of_account[:group1[,group2]...]

• The optional field 5 is a comma-separated list of UNIX groups which are aliased to (mapped onto) the Slurm account_name.

• Example:

  dtu::20:DTU departments
  fysik:dtu:parent:DTU Physics:physics
  camd:fysik:parent:CAMD section:camdfaculty,camdstudent

• Remember: Slurm accounts are named after UNIX groups!
Ignoring some UNIX groups

It is possible to add also a “fake” account_name=NOACCOUNT where the UNIX groups listed in field 5 will be ignored from further processing, for example:

NOACCOUNT:::We ignore these groups: group3, group4
Importing existing Slurm accounts

• Isn’t it a hassle to import all of your existing Slurm account hierarchy into an accounts.conf file?

• No worry: The simple script slurmaccounts2conf parses your Slurm account tree and outputs an initial accounts.conf file.

  (It just calls “sacctmgr show accounts” and prints the hierarchy)
Maintaining Slurm *accounts* with *accounts.conf*

- The script `slurmaccounts` reads `accounts.conf` and prints out `sacctmgr` commands which may be executed in order to update the Slurm database:

  ```
  sacctmgr add account …
  sacctmgr modify account …
  sacctmgr delete account …
  ```

- **Try it out:**
  No dangers are involved because `slurmaccounts` does not modify the Slurm database, but only prints Slurm commands which you should review before actually executing them.

  You can run the scripts as any unprivileged user!
Slurm user administration

Slurm users’ fairshare, QOS and limits (and defaults) are managed by the 
/etc/slurm/user_settings.conf file:

[DEFAULT|UNIX-group|username]:[Type]:value

Type examples:

- fairshare
- GrpTRES
- GrpTRESRunMins
- QOS
- DefaultQOS
- MaxJobs
- MaxSubmitJobs
- MaxJobsAccrue
- GrpJobsAccrue

Examples:

- DEFAULT:QOS: normal
- DEFAULT:DefaultQOS: normal
- DEFAULT:GrpTRES: cpu=1200
- DEFAULT:GrpTRESRunMins: cpu=3000000
- DEFAULT:MaxJobs: 500
- DEFAULT:MaxSubmitJobs: 5000
- DEFAULT:MaxJobsAccrue: 50
- DEFAULT:fairshare: 2
- user01:GrpTRES: cpu=2500
- user01:GrpTRESRunMins: cpu=4500000
- user02:QOS: normal, high
- camdfac:fairshare: 5
- camdvip:fairshare: 3
- cmdstud:fairshare: 2
Importing existing Slurm users

Done only initially:

- The `slurmusersettings2conf` script will capture the existing **Slurm user settings** and print them in the format of the `user_settings.conf` file.

- **DEFAULT** settings are determined by the highest frequency of values.
How does it work in practice?

• Initial setup is done with the `slurmaccounts2conf` and `slurmusersettings2conf` scripts.

• Slurm accounts are updated (infrequently) in `/etc/slurm/accounts.conf`  
  Executing `slurmaccounts` prints the required `sacctmgr` account commands.

• **UNIX users** are maintained in the system `passwd` and `group` databases.  
  Executing `slurmusersettings` will print the required  
  “`sacctmgr create/delete/modify user`“ commands.

• Every time you have a new user, or a user is modified or deleted,  
  just run `slurmusersettings`!

• User & account limits, fairshare etc. are maintained in `/etc/slurm/user_settings.conf`.  
  Executing `slurmusersettings` prints the required `sacctmgr modify user` commands.
### NOTICE: User sajal has NO DEFAULT ACCOUNT. Assume that this is a new Slurm user to be created
### Password entry: sajal:x:246025:1250:Sajal:/home/niflheim/sajal:/bin/bash
### NOTICE: User sajal has default account=, add to new default account=camdvip (primary UNIX group)
/usr/bin/sacctmgr -i create user name=sajal defaultaccount=camdvip MaxJobsAccrue=30 MaxSubmitJobs=5000
fairshare=3 DefaultQOS=normal MaxJobs=500 GrpTRES=cpu=1500 QOS=normal GrpTRESRunMins=cpu=4000000

### Slurm account sajal error: No password entry
/usr/bin/sacctmgr -i delete user sajal

### User mab with primary UNIX group ntchfac and account ntchfac is a secondary member of the UNIX group ntchvip
/usr/bin/sacctmgr -i add user mab account=ntchvip

# User ohn currently has MaxSubmitJobs=200 but configuration is MaxSubmitJobs=50
# User ohn currently has MaxJobs=200 but configuration is MaxJobs=50
/usr/bin/sacctmgr -i modify user where name=ohn set MaxSubmitJobs=50 MaxJobs=50

### The UNIX groups cephuser,modules,slurm are aliased to the Slurm account: NOACCOUNT
Potential improvements

Slurm user settings in `/etc/slurm/user_settings.conf` file are currently:

```
[DEFAULT|UNIX-group|username]:[Type]:value
```

but could be generalized so that the `DEFAULT|UNIX-group|username` field would be replaced by a complete `Slurm Association`:

```
username:cluster:partition:account:[Type]:value
```

If there is a need, I could look into this.
Please send feedback to Ole.H.Nielsen@fysik.dtu.dk.
Advertisement: Ole’s Slurm tools

- I have made my Slurm tools available on GitHub: https://github.com/OleHolmNielsen/Slurm_tools/

- The most useful/popular tools are:
  - `pestat` Print Slurm nodes status with 1 line per node including job info.
  - `showuserjobs` Print the current node status and batch jobs status broken down into userids.
  - `showuserlimits` Print Slurm resource user limits and usage.
  - ...

- Slurm deployment HOWTO guide: https://wiki.fysik.dtu.dk/niflheim/SLURM
### Example of `pestat` output

```
[root@que ~]# pestat -Fd
Print only nodes that are flagged by * (RED nodes)
Omit nodes with states: down drain drain maint

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Partition</th>
<th>State</th>
<th>Num_CPU</th>
<th>CPUload</th>
<th>Memsize (MB)</th>
<th>Freemem (MB)</th>
<th>Joblist</th>
<th>JobID</th>
<th>User ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>c123</td>
<td>xeon40</td>
<td>idle</td>
<td>0 40</td>
<td>20.87*</td>
<td>384000</td>
<td>381731</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d055</td>
<td>xeon88'</td>
<td>alloc</td>
<td>8 8</td>
<td>8.01</td>
<td>23500</td>
<td>397*</td>
<td>1387753</td>
<td>mzhao</td>
<td></td>
</tr>
<tr>
<td>g011</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>6.93*</td>
<td>64000</td>
<td>49769</td>
<td>1404463</td>
<td>arnem</td>
<td></td>
</tr>
<tr>
<td>g019</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>6.62*</td>
<td>64000</td>
<td>59038</td>
<td>1104463</td>
<td>arnem</td>
<td></td>
</tr>
<tr>
<td>g026</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>6.79*</td>
<td>64000</td>
<td>58996</td>
<td>1404463</td>
<td>arnem</td>
<td></td>
</tr>
<tr>
<td>g027</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>6.85*</td>
<td>64000</td>
<td>55944</td>
<td>1404463</td>
<td>arnem</td>
<td></td>
</tr>
<tr>
<td>g028</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>6.79*</td>
<td>64000</td>
<td>55977</td>
<td>1404463</td>
<td>arnem</td>
<td></td>
</tr>
<tr>
<td>g033</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>706*</td>
<td>1973661</td>
<td>mzhao</td>
<td></td>
</tr>
<tr>
<td>g041</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>1637*</td>
<td>1386172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g043</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>1853*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g044</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>2013*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g045</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>2338*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g047</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>1718*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g048</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>3217*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g050</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>1860*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>g053</td>
<td>xeon16</td>
<td>alloc</td>
<td>16 16</td>
<td>16.01</td>
<td>64000</td>
<td>1860*</td>
<td>1368172</td>
<td>schnir</td>
<td></td>
</tr>
<tr>
<td>x002</td>
<td>xeon24</td>
<td>alloc</td>
<td>24 24</td>
<td>16.46*</td>
<td>256000</td>
<td>247114</td>
<td>1404300</td>
<td>geokast</td>
<td></td>
</tr>
<tr>
<td>x025</td>
<td>xeon24</td>
<td>alloc</td>
<td>24 24</td>
<td>24.08</td>
<td>256000</td>
<td>146502*</td>
<td>1367618</td>
<td>mabr</td>
<td></td>
</tr>
<tr>
<td>x036</td>
<td>xeon24</td>
<td>alloc</td>
<td>24 24</td>
<td>14.79*</td>
<td>256000</td>
<td>245387</td>
<td>1404381</td>
<td>geokast</td>
<td></td>
</tr>
<tr>
<td>x092</td>
<td>xeon24</td>
<td>alloc</td>
<td>24 24</td>
<td>12.73*</td>
<td>256000</td>
<td>107615</td>
<td>1403724</td>
<td>aegm</td>
<td></td>
</tr>
<tr>
<td>x102</td>
<td>xeon24</td>
<td>alloc</td>
<td>24 24</td>
<td>3.95*</td>
<td>256000</td>
<td>207626</td>
<td>1391964</td>
<td>askov</td>
<td></td>
</tr>
<tr>
<td>x150</td>
<td>xeon24</td>
<td>alloc</td>
<td>24 24</td>
<td>21.04*</td>
<td>256000</td>
<td>232080</td>
<td>1403728</td>
<td>askov</td>
<td></td>
</tr>
</tbody>
</table>
```