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

SLUG 2018 presentation *Keeping Accounts Consistent Across Clusters Using LDAP and YAML* (https://slurm.schedmd.com/publications.html)
by Christian Clémençon, Ewan Roche, Ricardo Silva (EPFL, Switzerland):

- 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

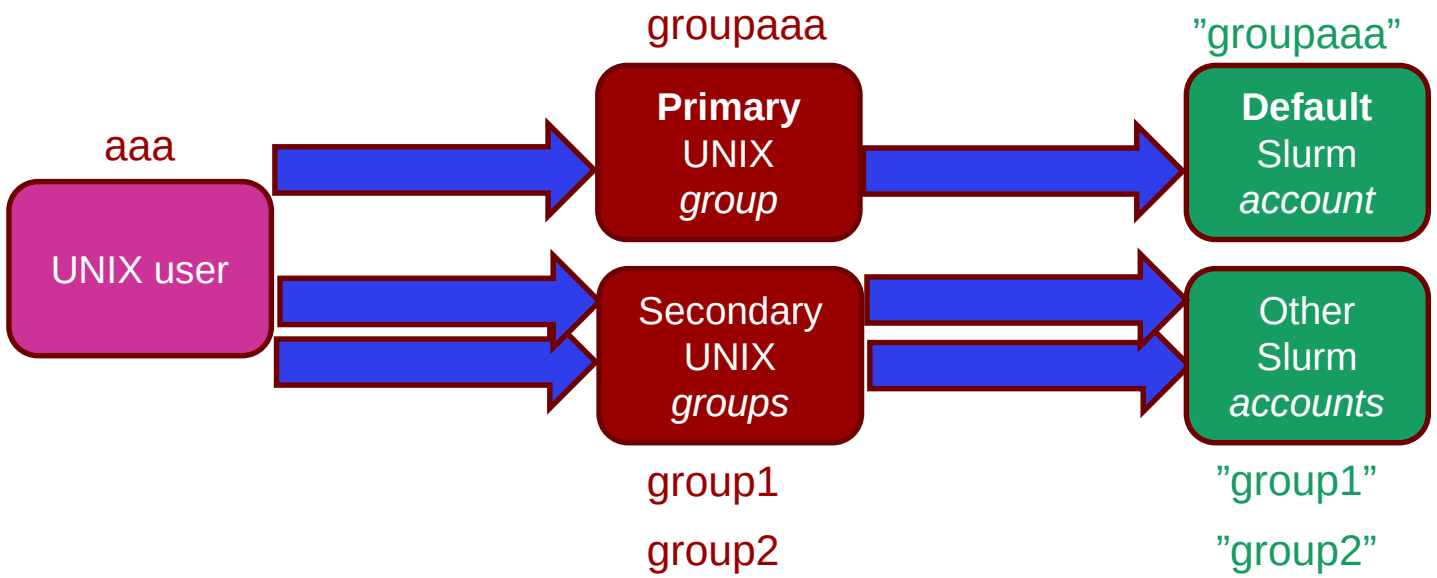
- Assume that the *passwd* and *group* system databases contain **the authoritative** user and *group* information for both Slurm as well as the system!
- Create a mapping of system UNIX *groups* onto Slurm *accounts*.
- Assign Slurm *account* names equal to the UNIX *group* names.
- User's Primary UNIX *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
camdstud: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.

slurmusersettings output examples

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 drng maint
Hostname       Partition     Node Num_CPU  CPUload  Memsize  Freemem  Joblist
                           State Use/Tot              (MB)     (MB)  JobId User ...
    c123          xeon40     idle   0  40   20.87*   384000   381731
    d055          xeon8*    alloc   8   8    8.01    23500      397* 1387753 mzhang
    g011          xeon16    alloc  16  16    6.83*    64000    48769  1404463 arnem
    g019          xeon16    alloc  16  16    6.62*    64000    59038  1404463 arnem
    g026          xeon16    alloc  16  16    6.79*    64000    58986  1404463 arnem
    g027          xeon16    alloc  16  16    6.65*    64000    58944  1404463 arnem
    g028          xeon16    alloc  16  16    6.79*    64000    58977  1404463 arnem
    g033          xeon16    alloc  16  16   16.01     64000      746* 1373441 mzhang
    g041          xeon16    alloc  16  16   16.01     64000     1637* 1368172 achrii
    g043          xeon16    alloc  16  16   16.01     64000     1893* 1368172 achrii
    g044          xeon16    alloc  16  16   16.01     64000     2013* 1368172 achrii
    g045          xeon16    alloc  16  16   16.01     64000     2013* 1368172 achrii
    g047          xeon16    alloc  16  16   16.01     64000     2338* 1368172 achrii
    g048          xeon16    alloc  16  16   16.01     64000     1718* 1368172 achrii
    g050          xeon16    alloc  16  16   16.01     64000     2217* 1368172 achrii
    g053          xeon16    alloc  16  16   16.01     64000     1868* 1368172 achrii
    x002          xeon24    alloc  24  24   16.46*   256000   247114  1404380 geokast
    x025          xeon24    alloc  24  24   24.08    256000    14682* 1387618 mabr
    x038          xeon24    alloc  24  24   14.79*   256000   245387  1404381 geokast
    x092          xeon24    alloc  24  24   12.73*   256000   107615  1403724 aegm
    x132          xeon24    alloc  24  24    3.95*   256000   207626  1391964 askov
    x150          xeon24    alloc  24  24   21.04*   256000   232080  1403728 askov
```