AI Cluster Admin

TODO

Since I'm still working on it, I don't guarantee any uptime yet. Mainly I need to make sure TRES tracking is working like we want. This will involve restarting slurmd and slurmctld which will kill running jobs.

- ~~generate report of storage usage~~
- ~~groups (Slurm 'Accounts') created for PI's.~~
    - ~~e.g. ericj_group: ericj, user1, user1, etc~~
- ~~grab QOS data from somewhere (gsheet or some kind of DB)~~
- ~~Properly deploy sync script~~
    - ~~Systemd unit~~
    - ~~main loop~~
- ~~research on slurm plugin to force GRES selection on job submit. Might be able to use:~~
    - ~~SallocDefaultCommand~~
    - ~~Otherwise look for 'AccountingStorageTRES' and 'JobSubmitPlugins' and /etc/slurm-llnl/job_submit.lua <- used to force user to specify '-gres'.~~
    - ~~jobs that do not specify a specific gpu type (e.g. gpu:rtx8000 or gpu:rtx2080ti) could be counted against either one but not specifically the you actually used.~~
    - ~~From 'AccountingStorageTRES' in slurm.conf: "Given a configuration of "AccountingStorageTRES=gres/gpu:tesla,gres/gpu:volta" Then "gres/gpu:tesla" and "gres/gpu:volta" will track jobs that explicitly request those GPU types. If a job requests GPUs, but does not explicitly specify the GPU type, then its resource allocation will be accounted for as either "gres/gpu:tesla" or "gres/gpu:volta", although the accounting may not match the actual GPU type allocated to the job and the GPUs allocated to the job could be heterogeneous. In an environment containing various GPU types, use of a job_submit plugin may be desired in order to force jobs to explicitly specify some GPU type."~~
- ~~ganglia for Slurm:~~ http://ai-mgmt2.ai.cs.uchicago.edu
    - figure why summary view is no longer a thing.
- ~~update 'coolgpus'. Lose VTs when this is running.~~
    - ~~coolgpus: sets fan speeds of all gpus in system.~~
    - ~~Goal is to statically set fan speeds to 80%. The only way to do this is with fake Xservers... but that means you lose all the VTs. Is this a compromise I'm willing to make?~~ It is.
- home directory
    - setup backups for home dirs
    - default quota
    - home directory usage report
- monitoring
    - basic node monitor
    - nfs or bandwidth monitoring

Fairshare

Check out the fairshare values

```
kauffman3@fe01:~$ sshare --long --accounts=kauffman3,kauffman4 --
users=kauffman3,kauffman4
             Account       User  RawShares  NormShares    RawUsage NormUsage
EffectvUsage  FairShare    LevelFS GrpTRESMins     TRESRunMins
-------------------- ---------- ---------- ----------- ----------- ---------
-- ------------- ---------- ---------- ------------------------------ ------
------------------------
kauffman3                                 1    0.000094         428 1.000000
1.000000             0.000094 cpu=475,mem=2807810,energy=0,+
 kauffman3           kauffman3          1    1.000000         428 1.000000
1.000000   0.000094   1.000000 cpu=475,mem=2807810,energy=0,+
kauffman4                                 1    0.000094           0 0.000000
0.000000                  inf cpu=0,mem=0,energy=0,node=0,b+
 kauffman4           kauffman4          1    1.000000           0 0.000000
0.000000   1.000000        inf cpu=0,mem=0,energy=0,node=0,b+
```

We are using the FairTree (fairshare algorithm). This is the default in Slurm these days and from what I can tell probably better suits our needs. It is no big deal to change to classic fairshare.

As the system exists now. One Account per User.

```
Account: kauffman
  Member: kauffman
User: kauffman
```

We will probably assign fairshare points to accounts, not users.

QOS

When submitting jobs users will have to include '–account=<groupnmame>' to get the priority levels associated with that account.

Priority levels: normal: [default] value=0 low: value=100 medium: value=500 high: value=1000

groupname is a Slurm 'account', with users of the cluster added.

As an admin the following would be created:

create group and set allowed QOS levels. Multiple levels can be specified. # Meaning you can set 'low,medium,high' with a default QOS of low # sacctmgr create account jonaslab # sacctmgr -i modify account jonaslab set qos=low # sacctmgr -i modify account jonaslab set defaultqos=low

Now add 'kauffman3' to the group # sacctmgr create user kauffman3 account=jonaslab

These values get used in the multifactor calculation to set the total priority on any given job.

The math/algorithm is available on SchedMDs site if anyone wants to come up with something

optimal. I've guessed at values that seem reasonable and should do what we want. https://slurm.schedmd.com/priority_multifactor.html#general The values on the left side of the + signs are values we can set.

It will be up to us to know when to remove any groups access to higher priorities. I imagine some sort of boolean in a spreadsheet or database.

If you do not use the '–account=<groupname>' switch it will use the users default account which has the default priority (normal) set.

Anyways... a more readable version of the policy would be helpful for me to try to match what we think we want to what we can do.