AI Cluster Policy Description

TODO

1. There are multiple methods used to calculate priority reflected on the spreadsheet.
2. Double check the math for correctness.
3. Does the math reflect our intent? I think it does.
4. Multiple methods in calculating priority are reflected (blue to purple cells: `har_priority`, `normalized_priority`, `weighted_normalized_priority`
5. Choose one calculation to use. This decision doesn't prevent us from changing it later if we find another calculation works better.
6. If you have a suggestion: Please show us the work in a new column or by cloning the sheet.

Contribution Tracking and Priority Calculation Tool

AI Cluster Tracking and Priority calculation spreadsheet

AI Cluster committee members have access.

Sheet usage

- Red: Do not edit
- Green: user input (This will be Techstaff 95% of the time)
- \`groups\` sheet:
  - contributions get assigned a POSIX group. Group must have a primary contact, who then gets to set members for that group.
  - calculates contribution amount for use in \`contrib-priority\`.
  - tracks group name and primary owner
- \`log\` sheet
  - All contributions will get entered here.
  - Hardware contribution gets converted to USD by techstaff. A receipt of the purchase is good starting point.
- The group 'cs' is calculated on the spreadsheet but doesn't actually get any priority set. 'general' is for all CS users. It needs to be part of the total sum but is treated as a special case with 0 priority.
- \`contrib-priority\` calculation references contrib amounts calculated in \`groups\`.

Understanding Slurm Fairshare and Priority/Multifactor

Slurm comes with built in tools to calculation fair share priorities without anyone needing to do anything special. The cluster uses a fair share algorithm with multiple factors to adjust job priorities.

How Slurm calculates Job priority

Generally Slurm will use this formula to determine a jobs priority.

```
Job_priority =
    site_factor +
    (PriorityWeightAge) * (age_factor) +
    (PriorityWeightAssoc) * (assoc_factor) +
    (PriorityWeightFairshare) * (fair-share_factor) +
    (PriorityWeightJobSize) * (job_size_factor) +
    (PriorityWeightPartition) * (partition_factor) +
    (PriorityWeightQOS) * (QOS_factor) +
    SUM(TRES_weight_cpu * TRES_factor_cpu,
        TRES_weight_<type> * TRES_factor_<type>,
        ...)
    - nice_factor
```

The factors on the left that start with `PriorityWeight*` can and are set in the Slurm cluster configuration file (found on all nodes at /etc/slurm-llnl/slurm.conf). The factors on the right are calculated based previous job submissions for that particular user.

```
fe01:~$ cat /etc/slurm-llnl/slurm.conf |grep "^Priority"
PriorityType=priority/multifactor
PriorityDecayHalfLife=08:00:00
PriorityMaxAge=5-0
PriorityWeightFairshare=500000
PriorityWeightPartition=100000
PriorityWeightQOS=0
PriorityWeightJobSize=0
PriorityWeightAge=0
PriorityFavorSmall=YES
```

*Note that this example may not be up to date when you read this.

How we modify job priority to favor contributors

- We adjust those priorities with partitions for those who have donated either monetarily or with hardware. Hardware donations get converted a monetary value when logged on the spreadsheet.
- Every contribution gets assigned a POSIX group, hereon referred to as `$group`.

Here is a version of the partition configuration as it stands now (2021-02-10).

```
PartitionName=general Nodes=a[001-008]
PartitionName=cdac-own Nodes=a[005-008] AllowGroups=cdac Priority=100
PartitionName=cdac-contrib Nodes=a[001-008] AllowGroups=cdac Priority=5
```

Partition	Description	Priority
general	For all users	0
${group}-own	Machines $group has donated	100
${group}-contrib	A method to give slightly higher job priority to groups who have donated but do not own machines.	Variable based on spreadsheet calculation.

The key thing to notice before you continue reading is that nodes can be added to multiple partitions. `general` and `cdac-contrib` can submit to all nodes but with different priorities.

Calculating -contrib partition usage

We do the following calculation to determine the `*-contrib` partitions usage over the past 30 days in comparison to total cluster usage.

```
partition usage total time in seconds for 30 days
------------------------------------------------------ = percent used
all partition usage total time in seconds for 30 days
```

The percent will end up as an integer.

There is a python script that does this and sends techstaff a report. The repo is currently not available for everyone to see but I think that it should be eventually. In the mean time you can take a look at it on the front end nodes (/usr/local/slurm-tools/cluster_partition_usage.py).

You'll see on the spreadsheet we take subtract `percent used` from 100, ensure it's positive and call that "idleness".

Total amount of money contributed and "idleness" are the key factors in determining the priority of a groups 'contrib' partition.

This calculation will be run once a month and the relevant groups ${group}-contrib priority updated to reflect past months usage.

Note that the term "idleness" should not be taken literally. I don't know of way to actually calculate true idleness. I believe that the current calculation reflects the intended usage.

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~~
  - ~~userland tool to check quota~~
  - ~~home directory usage report~~
- monitoring
  - ~~basic node monitor~~
  - nfs or bandwidth monitoring
  - gpu
- sync script
  - ~~fix bug to ensure accounts and users are created~~