10.3. Hierarchical Fair Share (HFS)

Hierarchical Fair Share (HFS) is a feature that allows user quotas and priorities to be managed within an administrator-defined hierarchy. MRG Grid provides the ability to specify HFS hierarchies of any depth and breadth.

Setting up hierarchical fair share

1. The hierarchical groups are specified in the global configuration file, using the configuration variable **GROUP_NAMES** (see also example 10.6). Groups are delimited from subgroups (or "children") by a period (**.**) character, in the same way that groups are delimited from users. If two groups are defined as ***group_physics*** and ***group_chemistry***, the subgroups are defined as ***group_physics.lab1***, ***group_physics.lab2***, ***group_chemistry.lab1***, and ***group_chemistry.lab2***. Users submit jobs to these subgroups by adding a plus (**+**) character and the name of the accounting group to the job submit description file. For example, if user ***mcurie*** wants to submit a job to ***group_physics.lab1***, they would add ***+Accounting_Group = "group_physics.lab1.mcurie"*** to their job submit description file:

   ```
   executable = /bin/sleep
   arguments = 600
   universe = vanilla
   should_transfer_files = YES
   when_to_transfer_output = ON_EXIT
   +AccountingGroup = "group_physics.lab1.mcurie"
   queue 100
   ```

   Each group must also have a quota declaration. Quota declarations can be either *dynamic* or *static*. If a group has no quota declaration, its quota will be assigned to zero with a warning in the ***Negotiator*** log. In most cases using dynamic quotas is considered preferable to static quotas due to the dynamic nature of grid computing, where the number of available slots can change over time.

   A static quota is expressed as a single integer, representing a specific slot count. This quota will remain the same regardless of changes in the number of available slots. A static quota is assigned using a configuration variable of the form **GROUP_QUOTA_*groupname***. For example, a static quota for ***group_physics*** is assigned using the configuration variable **GROUP_QUOTA_DYNAMIC_group_physics**. If static quotas combine to an amount greater than the quota of the parent group, the static quotas will be normalized to the parent quota (accompanied by a warning message in the ***Negotiator*** log). For example, if the quota for ***group_physics*** is 15, and its children ***group_physics.lab1*** and ***group_physics.lab2*** have static quotas of 10 and 20, then these child quotas will be normalized to 5 and 10 respectively.

   A dynamic quota is specified as a fractional value between ***0.0*** and ***1.0***, which represents a fraction of the quota allocated to the parent group. Dynamic quotas are assigned in proportion to the number of slots currently available. A dynamic quota is assigned using a configuration variable of the form **GROUP_QUOTA_DYNAMIC_*groupname***. For example, if the total number of slots is 20, ***group_physics*** has a dynamic quota of 0.5, ***group_physics.lab1*** has a dynamic quota of 0.2 and ***group_physics.lab2*** has a dynamic quota of 0.8, then ***group_physics.lab1*** will be assigned a slot quota of (20)(0.5)(0.2) = 2, and ***group_physics.lab2*** will be assigned (20)(0.5)(0.8) = 8. When dynamic quotas sum to greater than 1, they are normalized so that they sum to 1, with a warning message in the ***Negotiator*** log.

   When the total quota of the subgroups within a parent group is less than the quota for the parent group, the remainder is assigned to the parent and is available for users to submit jobs against in the parent accounting group. If the total quota of the subgroups in a parent group equal the total quota for the parent group, the parent group will have zero quota; any jobs submitted to the parent will run only if its children

have unused quota ("surplus"). For example, if the total number of slots is 20, ***group_physics*** has a dynamic quota of 0.5, ***group_physics.lab1*** has a dynamic quota of 0.2 and ***group_physics.lab2*** has a dynamic quota of 0.3, then ***group_physics.lab1*** will be assigned a slot quota of (20)(0.5)(0.2) = 2, and ***group_physics.lab2*** will be assigned (20)(0.5)(0.3) = 3 and ***group_physics*** itself will be assignedthe remainder (20)(0.5)-(2+3) = 5.

Static and dynamic quotas may be mixed. When a parent group's children have both static and dynamic quotas, the children with static quotas are assigned first. Any remaining quota is shared among children with dynamic quotas. For example, if ***group_physics*** has a slot quota of 10, and ***group_physics.lab1*** has a static quota of 2, the ***group_physics.lab2*** has a dynamic quota of 0.5, then ***group_physics.lab1*** will be assigned its static quota of 2 and ***group_physics.lab2*** will be assigned (10-2)(0.5) = 4. Subgroups may be assigned static or dynamic quotas regardless of which kind of quota was declared for the parent.

```
GROUP_NAMES = group_physics, group_chemistry, group_physics.lab1,
group_physics.lab2, group_physics.lab3, group_physics.lab3.team1,
group_physics.lab3.team2, group_physics.lab3.team3,
group_chemistry.lab1, group_chemistry.lab2

GROUP_QUOTA_DYNAMIC_group_physics = .4
GROUP_QUOTA_DYNAMIC_group_chemistry = .4
GROUP_QUOTA_DYNAMIC_group_chemistry.lab1 = .4
GROUP_QUOTA_DYNAMIC_group_chemistry.lab2 = .6
GROUP_QUOTA_DYNAMIC_group_physics.lab1= .2
GROUP_QUOTA_DYNAMIC_group_physics.lab2= .2
GROUP_QUOTA_DYNAMIC_group_physics.lab3= .6
GROUP_QUOTA_DYNAMIC_group_physics.lab3.team1 = .2
GROUP_QUOTA_DYNAMIC_group_physics.lab3.team2 = .2
GROUP_QUOTA_DYNAMIC_group_physics.lab3.team3 = .4
```

2. A group may have more slots assigned than it is currently using. In this case the group is said to have "surplus quota." Surplus quota is passed up to parents for sharing. For example, a parent group with zero quota can use any surplus quota from its children to run jobs submitted against it.

   By default, child subgroups will not use surplus quota from their siblings or parent. However, this behavior may be overridden with the **autoregroup** feature which allows groups to use quota that is unused by other groups. To enable **autoregroup** for a group, set the **GROUP_AUTOREGROUP_*groupname*** configuration variable to ***TRUE***. In some configurations, it may be desirable to define the default surplus sharing behaviour of all groups to be ***TRUE***. This can be accomplished by setting the configuration variable **GROUP_AUTOREGROUP = TRUE**. Individual groups can have **autoregroup** disabled by setting **GROUP_AUTOREGROUP_*groupname* = FALSE**.

```
GROUP_AUTOREGROUP_group_physics = TRUE
GROUP_AUTOREGROUP_group_physics.lab3 = TRUE
GROUP_AUTOREGROUP_group_physics.lab3.team1 = TRUE
GROUP_AUTOREGROUP_group_chemistry = TRUE
GROUP_AUTOREGROUP_group_chemistry.lab1 = TRUE
GROUP_AUTOREGROUP_group_chemistry.lab2 = TRUE
```

   Surplus quota is shared in the following way: for each group, unused quota is collected from itself and its children. The ***surplus quota*** then is shared proportionately by the group and its children. Children with **autoregroup** set to ***FALSE*** do not share quota. The parent will share this quota regardless of its **autoregroup** setting. A group can always share surplus from below it in the hierarchy, but will only share surplus from above if **autoregroup** is enabled. ***Surplus quota*** is first shared to groups with nonzero quota, in proportion to slot quota. For example, starting with 20 slots, if ***group_physics*** has dynamic quota 0.5, ***group_physics.lab1*** has static quota 2 and ***group_physics.lab2*** has dynamic quota 0.5, then ***surplus quota*** is shared between ***group_physics***, ***group_physics.lab1*** and ***group_physics.lab2*** using the ratios (4/10), (2/10) and (4/10) respectively (i.e. 0.4, 0.2 and 0.4).

   If a group requires less than its share of the surplus, the above algorithm iterates and remaining quota is

re-shared among remaining candidates, until no further groups require any, or all ***surplus quota*** is assigned. Any remaining ***surplus quota*** is then shared evenly between groups with zero quota (typically, only a parent group will have zero quota). Again, if groups require less than their share, this iteration occurs. Any remaining ***surplus quota*** is then passed up the hierarchy to be shared at the next higher level.

All top-level accounting groups are implicitly children of a predefined ***root*** group, which appears with the name "<none>" in the ***Accountant*** and in ***Negotiator*** log output. The ***root*** group behaves similarly to any other defined group: it will be assigned any quota not assigned to its children, and it will share surplus on equal footing with its children. Any job submitted with no accounting group (or with an undefined accounting group) will be assigned to the ***root*** group.

Accounting group configurations involving dynamic quotas may result in groups with fractional slot quotas. For example, if ***group_physics*** and ***group_chemistry*** each have a dynamic quota of 0.5, and the number of available slots is 9, then their slot quotas will each be assigned 4.5. These fractional "remainders" cannot be used as-is, since jobs utilize whole slots.

After all slot quotas have been assigned, and the surplus has been shared, the accounting group hierarchy is traversed to recover any fractional slots. At each group, recovered fractional remainder is collected, and any whole slots are distributed among the group and its children, using a round robin algorithm. Groups with **autoregroup** disabled do not share recovered remainders. As with surplus, the parent group can share its subgroup remainders regardless of its **autoregroup** setting. After distribution, any unused remainder is passed up the hierarchy to be shared at the next higher level.

Prev Up Home Next

10.2. Job Priorities Chapter 11. The Virtual Machine Universe