Priority and Provisioning
Greg Thain
HTCondorWeek 2015

Overview

Important HTCondor architecture bits

Detour to items that doesn't fit elsewhere

Groups and why you should care

User priorities and preemption

Draining

Provisioning vs. Scheduling

- HTCondor separates the two
  - Very important
- Central Manager (negotiator) provisions
- Schedd schedules

Provisioning

- Negotiator selects a slot for a USER
- Based on USER attributes (and machine)
  - With some small thought to the users's job
- At slower frequency: the negotiation cycle
  - Don't obsess about negotiation cycle time

Scheduling

- Schedd takes that slot for the user
  - And runs one or more jobs on it
  - For how long?
    - `CLAIM_WORKLIFE = some_seconds`

Consequences

- May take much longer to start 1st job
- The central manager responsible for users
- The central manager responsible for groups
- Accounting happens in the CM


THE SCHEDD, DUDE
DOESN'T SCHEDULE!
memegenerator.net

Now, for the detour…

Schedd Policy: Job Priority

- Set in submit file with: **`JobPriority = 7`**
- … or dynamically with **`condor_prio`** cmd
- Integers, larger numbers are better priority
- Only impacts order between jobs for a single user on a single schedd
- A tool for users to sort their own jobs

Schedd Policy: Job Rank

- Set with
  - `RANK = Memory`
- In condor_submit file
- Not as powerful as you may think:
  - Negotiator gets first cut at sorting
  - Remember steady state condition

Concurrency Limits

- Useful for globally (pool-wide):
  - License limits,
  - NFS server overload prevention
  - Limiting database access
- Limits total number jobs across all schedds

Concurrency Limits (2)

- In central manager config
  - `MATLAB_LICENSE_LIMIT = 10`
- In submit file
  - `concurrency_limits = matlab`

Rest of this talk: Provisioning, not Scheduling

- schedd sends idle users to the negotiator
- Negotiator picks machines (idle or busy) to send to the schedd for those users
- How does it pick?

What's a user?

- Bob in schedd1 same as Bob in schedd2?
- If have same UID_DOMAIN, they are.
  - Default UID_DOMAIN is FULL_HOSTNAME
- Prevents cheating by adding schedds
- Map files can redefine the local user name

Or, a User could be a “group”

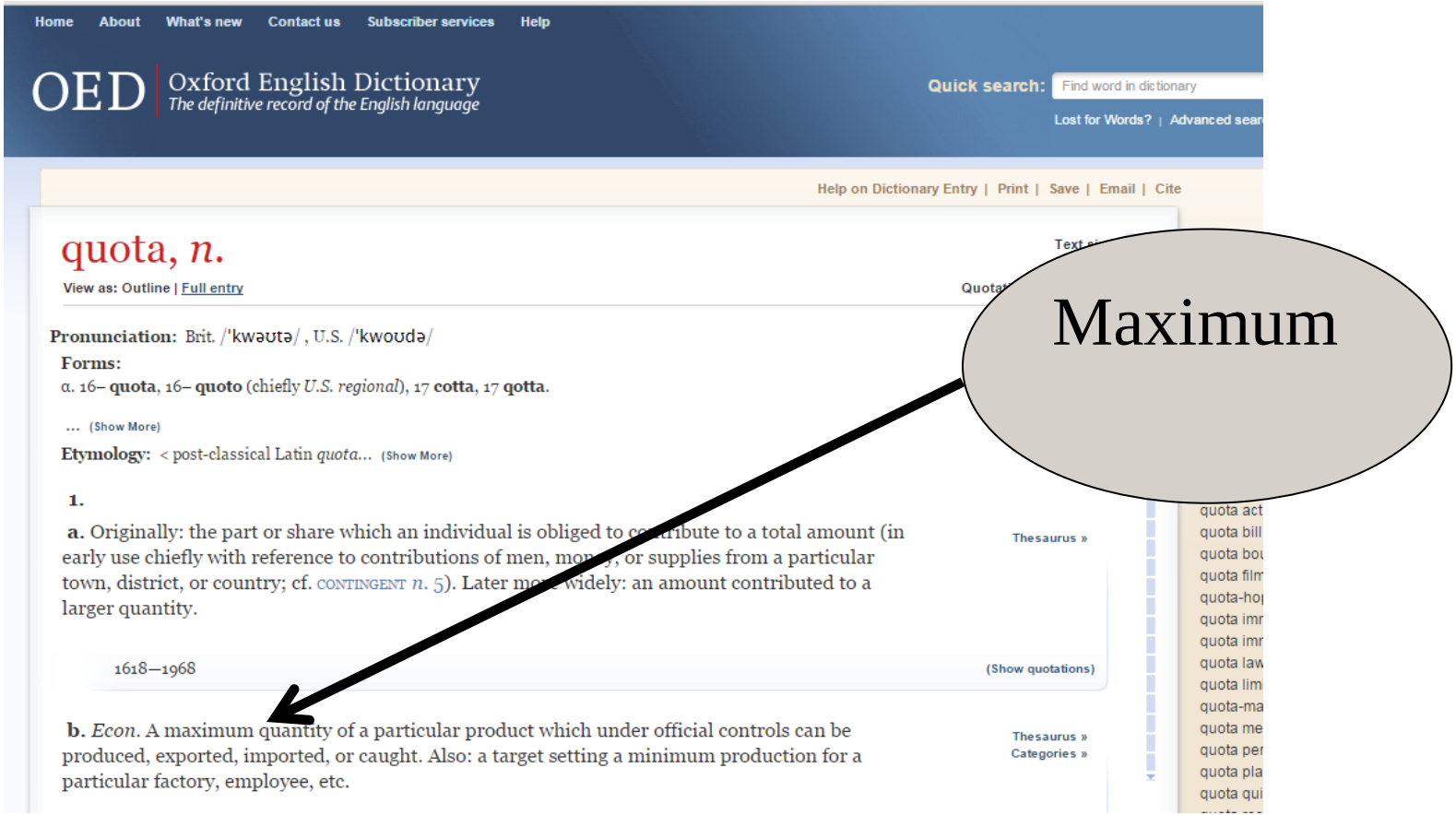
Accounting Groups (2 kinds)

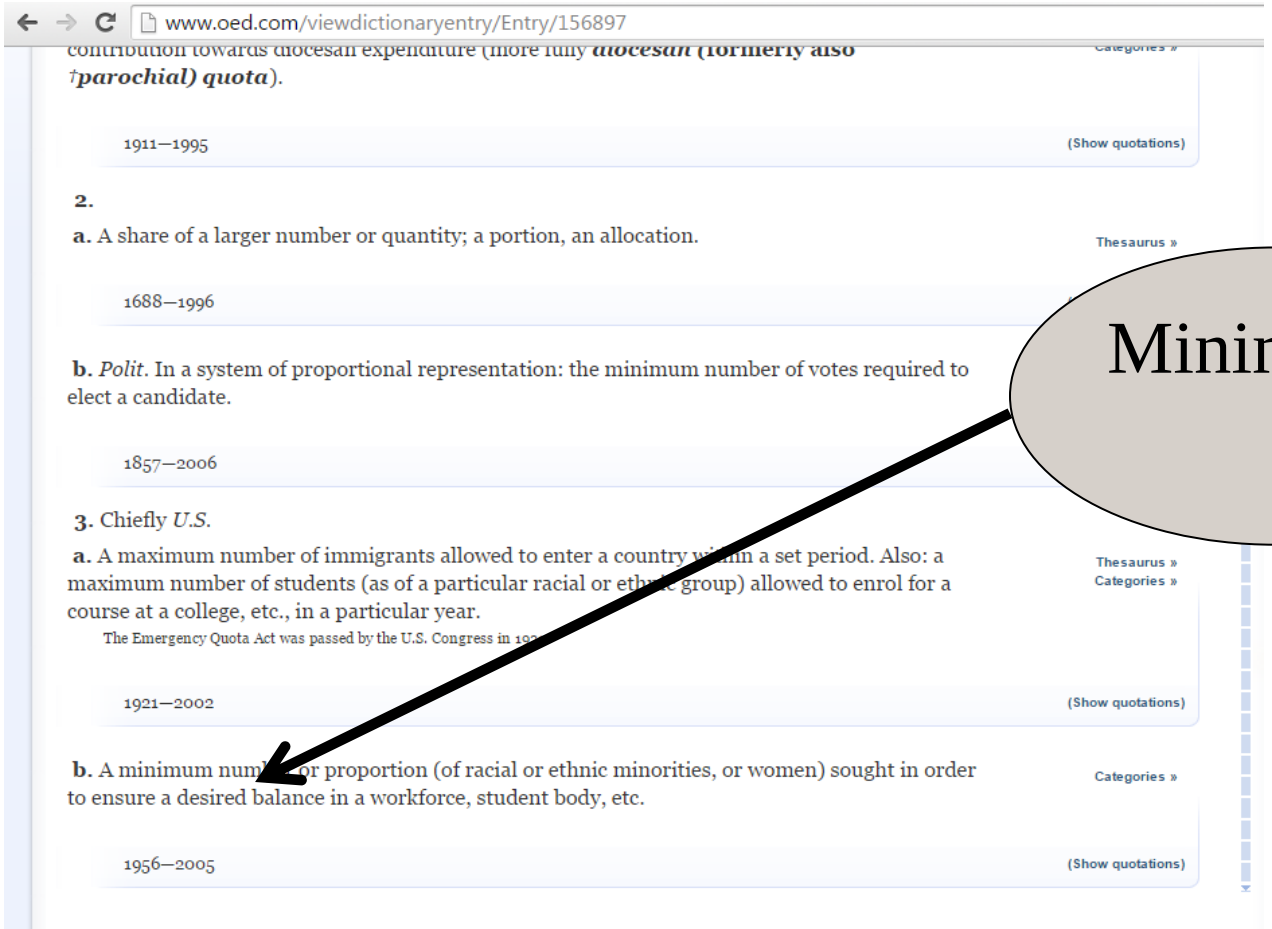
- Manage priorities across groups of users
  Can guarantee maximum numbers of computers for groups (quotas)
- Supports hierarchies
- Anyone can join any group

Accounting Groups as Alias

- In submit file
  - `Accounting_Group = group1`
- Treats all users as the same for priority
- Accounting groups not pre-defined
- No verification – condor trusts the job
- condor_userprio replaces user with group

Accounting Groups w/ Quota aka: “Hierarchical Group Quota”


Home
About
What's new
Contact us
Subscriber services
Help
OED
Oxford English Dictionary
The definitive record of the English language
Quick search:
Find word in dictionary
Lost for Words?
Help on Dictionary Entry | Print | Save | Email | Cite
quota, n.
View as: Outline | Full entry
Pronunciation: Brit. /ˈkwəʊtə/ , U.S. /ˈkwoʊdə/
Forms:
α. 16– quota, 16– quoto (chiefly U.S. regional), 17 cotta, 17 qotta.
... (Show More)
Etymology: < post-classical Latin quota... (Show More)
1.
a. Originally: the part or share which an individual is obliged to contribute to a total amount (in early use chiefly with reference to contributions of men, money, or supplies from a particular town, district, or country; cf. CONTINGENT n. 5). Later more widely: an amount contributed to a larger quantity.
Thesaurus »
1618—1968
(Show quotations)
b. Econ. A maximum quantity of a particular product which under official controls can be produced, exported, imported, or caught. Also: a target setting a minimum production for a particular factory, employee, etc.
Thesaurus »
Categories »
Maximum


www.oed.com/viewdictionaryentry/Entry/156897
†parochial) quota).
1911–1995
(Show quotations)
2.
a. A share of a larger number or quantity; a portion, an allocation.
Thesaurus »
1688–1996
b. Polit. In a system of proportional representation: the minimum number of votes required to elect a candidate.
1857–2006
3. Chiefly U.S.
a. A maximum number of immigrants allowed to enter a country within a set period. Also: a maximum number of students (as of a particular racial or ethnic group) allowed to enrol for a course at a college, etc., in a particular year.
Thesaurus »
Categories »
1921–2002
(Show quotations)
b. A minimum number or proportion (of racial or ethnic minorities, or women) sought in order to ensure a desired balance in a workforce, student body, etc.
Categories »
1956–2005
(Show quotations)
Minimum

HGQ: Strict quotas

- "a" limited to 10
- "b" to 20,
- Even if idle machines
- What is the unit?
  - Slot weight.
- With fair share of users within group

- Must be predefined in cm's config file:

```
GROUP_NAMES = a, b, c
GROUP_QUOTA_a = 10
GROUP_QUOTA_b = 20
```

- And in submit file:

```
Accounting_Group = a
Accounting_User = gthain
```

Group_accept_surplus

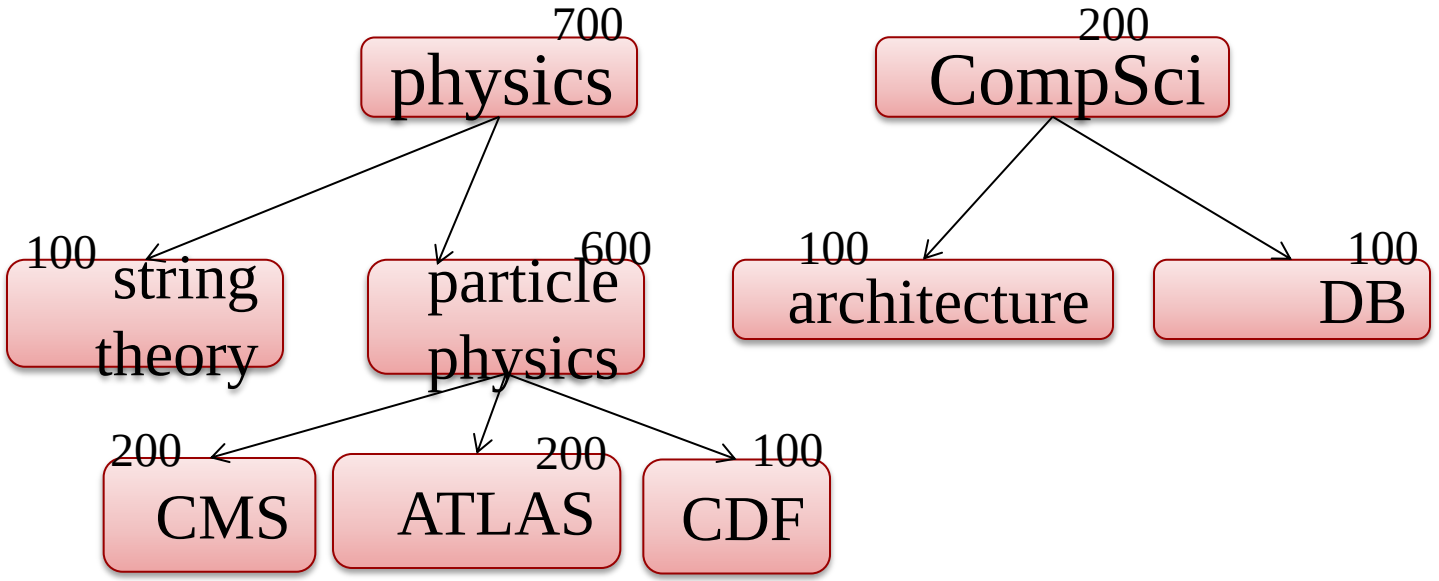
- Group_accept_surplus = true
- Group_accept_surplus_a = true
- This is what creates hierarchy
  - But only for quotas

GROUP_AUTOREGROUP

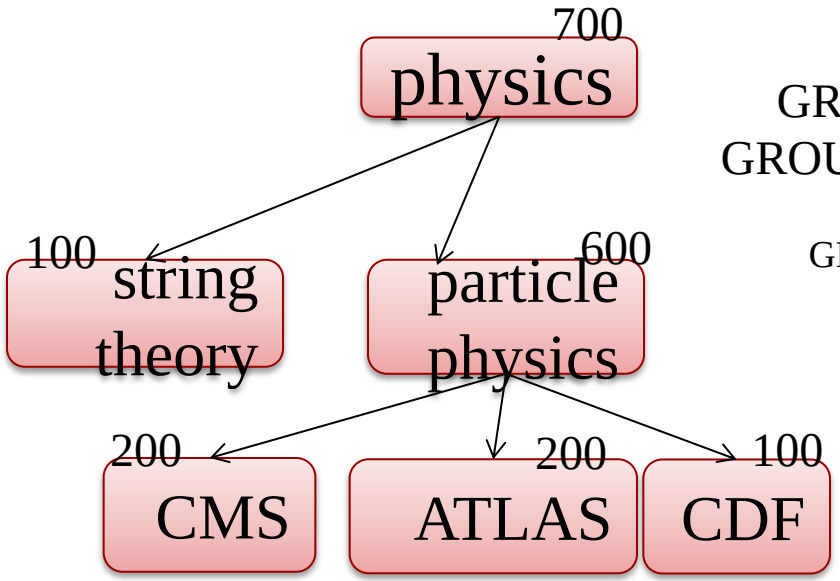
- Allows groups to go over quota if idle machines

- “Last chance” wild-west round, with every submitter for themselves.

Hierarchical Group Quotas



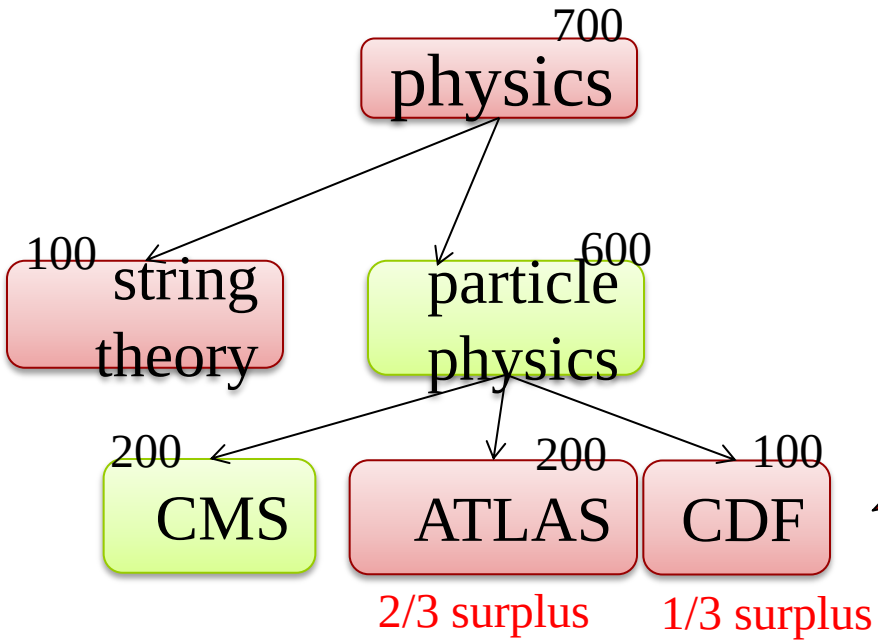
Hierarchical Group Quotas



GROUP_QUOTA_physics = 700
GROUP_QUOTA_physics.string_theory = 100
GROUP_QUOTA_physics.particle_physics = 600
GROUP_QUOTA_physics.particle_physics.CMS = 200
GROUP_QUOTA_physics.particle_physics.ATLAS = 200
GROUP_QUOTA_physics.particle_physics.CDF = 100

group.sub-group.sub-sub-group…

Hierarchical Group Quotas



Groups configured to accept surplus will share it in proportion to their quota.

Here, unused particle physics surplus is shared by ATLAS and CDF.

GROUP_ACCEPT_SURPLUS_physics.particle_physics.ATLAS = true
GROUP_ACCEPT_SURPLUS_physics.particle_physics.CDF = true

Gotchas with quotas

- Quotas don’t know about matching
- Assuming everything matches everything
- Surprises with partitionable slots
- Managing groups not easy

- May want to think about draining instead.

Enough about Groups

- Remember: group quota comes first!
- Groups only way to limit total running jobs per user/group

- Haven’t gotten to matchmaking yet

Negotiation Cycle

- Gets all the slot ads from collector
- Based on new user prio, computes submitter limit for each user
- Foreach user, finds the schedd, gets a job
  - Finds all matching machines for job
  - Sorts the machines
  - Gives the job the best machine (may preempt)

Negotiator metric: User Priority

- Negotiator computes, stores the user prio
- View with **`condor_userprio` tool**
- Inversely related to machines allocated (lower number is better priority)
  - A user with priority of 10 will be able to claim twice as many machines as a user with priority 20

User Priority

- (Effective) User Priority is determined by multiplying two components

- Real Priority * Priority Factor

Real Priority

- Based on actual usage, starts at .5
- Approaches actual number of machines used over time
  - Configuration setting **`PRIORITY_HALFLIFE`**
  - If PRIORITY_HALFLIFE = +Inf, no history
  - Default one day (in seconds)
- Asymptotically grows/shrinks to current usage

Priority Factor

- Assigned by administrator
  - Set/viewed with **`condor_userprio`**
  - Persistently stored in CM
- Defaults to 1000 (**`DEFAULT_PRIO_FACTOR`**)
- Allows admins to give prio to sets of users, while still having fair share within a group
- “Nice user”s have Prio Factors of 1,000,000

condor_userprio

- Command usage:

```
condor_userprio -most
```

```
                                     Effective   Priority
User Name                            Priority     Factor   In Use (wghted-hrs) Last Usage
------------------------------------ --------- ---------- ------ ------------ ----------
lmichael@submit-3.chtc.wisc.edu           5.00      10.00      0        16.37    0+23:46
blin@osghost.chtc.wisc.edu                7.71      10.00      0      5412.38    0+01:05
osgtest@osghost.chtc.wisc.edu            90.57      10.00     47     45505.99      <now>
cxiong36@submit-3.chtc.wisc.edu         500.00    1000.00      0         0.29    0+00:09
ojalvo@hep.wisc.edu                     500.00    1000.00      0    398148.56    0+05:37
wjiang4@submit-3.chtc.wisc.edu          500.00    1000.00      0         0.22    0+21:25
cxiong36@submit.chtc.wisc.edu           500.00    1000.00      0        63.38    0+21:42
```

Prio factors with groups

```
condor_userprio -setfactor 10 group1.wisc.edu
condor_userprio -setfactor 20 group2.wisc.edu
```

Note that you must get UID_DOMAIN correct

Gives group1 members 2x resources as group2

Sorting slots: sort levels

```
NEGOTIATOR_PRE_JOB_RANK =
    RemoteOwner =?= UNDEFINED
JOB_RANK = mips
NEGOTIATOR_POST_JOB_RANK =
    (RemoteOwner =?= UNDEFINED) *
(KFlops)
```

Power of NEGOTIATOR_PRE_JOB_RANK

- Very powerful
- Used to pack machines
  - `NEGOTIATOR_PRE_JOB_RANK = isUndefined(RemoteOwner) * (- SlotId)`
- Sort multicore vs. serial jobs

More Power of NEGOTIATOR_PRE_JOB_RANK

Best fit of multicore jobs:

```
NEGOTIATOR_PRE_JOB_RANK =
(1000000 * (RemoteOwner =?= UNDEFINED))
    - (100000 * Cpus) - Memory
```

If Matched machine claimed, extra checks required

- **`PREEMPTION_REQUIREMENTS`** and **`PREEMPTION_RANK`**
- Evaluated when **`condor_negotiator`** considers replacing a lower priority job with a higher priority job
- Completely unrelated to the **`PREEMPT`** expression (which should be called evict)

A note about Preemption

- Fundamental tension between
  - Throughput vs. Fairness
- Preemption is required to have fairness
- Need to think hard about runtimes, fairness and preemption
- Negotiator implementation preemption
- (Workers implement eviction: different)

PREEMPTION_REQUIREMENTS

- MY = busy machine // TARGET = job
- If false will not preempt machine
  - Typically used to avoid pool thrashing
  - Typically use:
    - **`RemoteUserPrio`** – Priority of user of currently running job (higher is worse)
    - **`SubmittorPrio`** – Priority of user of higher priority idle job (higher is worse)

PREEMPTION_REQUIREMENTS

- Replace jobs running > 1 hour and 20% lower priority

```
StateTimer = \
  (CurrentTime - EnteredCurrentState)
HOUR = (60*60)
PREEMPTION_REQUIREMENTS = \
  $(StateTimer) > (1 * $(HOUR)) \
  && RemoteUserPrio > SubmittorPrio * 1.2
```

Preemption with HQG

By default, won't preempt to make quota

But, "there's a knob for that"

```
PREEMPTION_REQUIREMENTS =
(SubmitterGroupResourcesInUse <
SubmitterGroupQuota) &&
(RemoteGroupResourcesInUse >
RemoteGroupQuota) && ( RemoteGroup =!=
SubmitterGroup
```

PREEMPTION_REQUIREMENTS is an expression

- ```
  ( MY.TotalJobRunTime >
    ifThenElse((isUndefined(MAX_PREEMPT) ||
    (MAX_PREEMPT =?= 0)),(72*(60 *
  60)),MAX_PREEMPT) )
  ```

- ```
  && RemoteUserPrio > SubmittorPrio * 1.2
  ```

PREEMPTION_RANK

- Of all claimed machines where PREEMPTION_REQUIREMENTS is true, picks which one machine to reclaim
- Strongly prefer preempting jobs with a large (bad) priority and a small image size

```
PREEMPTION_RANK = \
  (RemoteUserPrio * 1000000)- ImageSize
```

Better PREEMPTION_RANK

Based on …

- Runtime?
- Cpus?
- SlotWeight?

MaxJobRetirementTime

- Can be used to guarantee minimum time
- E.g. if claimed, give an hour runtime, no matter what:

- MaxJobRetirementTime = 3600
- Can also be an expression

Partitionable slots

- What is the “cost” of a match?
  - SLOT_WEIGHT (cpus)
- What is the cost of an unclaimed pslot?
  - The whole rest of the machine
  - Leads to quantization problems
- By default, schedd splits slots
- “Consumption Policies”: some rough edges

Draining and defrag

- Instead of preemping, we can drain
- Condor_drain command initiates draining

- Defrag daemon periodically calls drain

Defrag knobs

```
DEFRAG_MAX_WHOLE_MACHINES = 12
DEFRAG_DRAINING_MACHINES_PER_HOUR = 1
DEFRAG_REQUIREMENTS = PartitionableSlot &&
        TotalCpus > 4 && …
DEFRAG_WHOLE_MACHINE_EXPR=
        PartitionableSlot && cpus > 4
```

Summary

- Many ways to schedule
- Knobs: We got ‘em!