Table of Contents

DIRAC Task Queue Matching

This page describes how the DIRAC handles matching using Task Queues.

Scenario

One of the most CPU consuming tasks any workload management system has to handle is matching. Every time a resource is available, the workload management system has to decide which job will use it. To do so, DIRAC WMS has to select a job that matches the resource attributes but also one that follows the priority rules. And it has to be able to select one job from up to a million waiting jobs.

Matching phases

Before the matching takes place, DIRAC tries to group together jobs to ease matching. All jobs with identical requirements are grouped in one *TaskQueue* before they can be matched. By doing this the number of possible choices is narrowed down.

Task Queue creation

Each job will be placed into a *_TaskQueue* that has the same requirements as the job.

Task Queue Requirements

Job requirements used to group into *TaskQueues* are:

- *Owner DN* : DN of the user that submitted the job
- *Owner Group* : Group used to submit the job by the user
- *Setup* : Setup to which this job was submitted
- *CPU Time* : Required CPU time for the job to run
- *Submit Pools* : Pools to be used to submit pilots for this job
- *Pilot Types* : Pilot types that can run this job (possible choices are *private* or nothing right now)
- *Sites* : Sites that can run this job
- *GridCEs* : Grid CEs that can run this job
- *GridMiddlewares* : Grid middleware required to run this job
- *BannedSites* : Sites that can **not** run this job
- *LHCbPlatforms* : Operating system and architecture required

Task Queue Implementation

TaskQueues are implemented as a DB schema. There is a main table for *TaskQueues* containing:

- *TaskQueue* id
- Owner DN
- Owner Group
- Setup
- CPU time
- Priority

This requirements can only have one value, so they are stored in a single row per *TaskQueue*. CPU time has an infinite amount of possible values so it's restricted to 4 different values (500, 5000, 50000, 300000). The Job CPU time is moved to the next value. For instance if job requires 10 CPU time, the *TaskQueue* will have 500. If the job CPU time exceeds 300000 the *TaskQueue* will have 300000 at maximum.

The rest of the requirements can have multiple values. For each field there is a table having *TaskQueue* id and one value per row. There can me more than one row with the same *TaskQueue* id.

Jobs are stored in a table containing:

- *TaskQueue* id
- Job Id
- User job priority

Resource matching

When the *JobAgent* requests a match, it sends the *Matcher* a dictionary with the resource description. The matching algorithm will select *TaskQueues* that have requirements that match those resource attributes.

- Owner Group must match (if private pilot)
- Owner DN must match if the Owner Group does not have *JOB_SHARING* property (if private pilot)
- Setup must match
- CPU time must be lower or equal than the one presented by the resource
- Pilot type must be accepted by the job
- Site must be accepted by the job
- Site cannot be in the Banned sites list of the job
- If Job requires a Grid CE, it must match
- LHCbPlatform must match if required by the job

Submit pool and Grid middleware are only used by the pilot submission mechanism. To select a *TaskQueue* the Matcher follows the next steps:

1. Selects all the possible *TaskQueues* that match the resource attributes
2. Orders them by CPU time in descending order and selects the highest CPU time
3. Of the remaining sorts them by RAND() divided by *TaskQueue* priority in ascending order
4. Get the first one

Once the Matcher has selected a *TaskQueue*, it

1. Orders the jobs in the *TaskQueue* by RAND() divided by "UserPriority in ascending order, and gets the 'UserPriority' of the fist job.
2. Selects all the jobs in the *TaskQueue* that have the previously selected 'UserPriority' and orders them by Job Id in ascending order
3. Get the first ten and order them in random order
4. Get the first one

This topic: LHCb > DIRACWMSTaskQueueMatching
Topic revision: r4 - 2009-02-17 - AdriaCasajus