DIRAC framework overview

Alessandro Paolini

Preamble

- WMS will not be ported on CentOS7 (UMD4)
- Currently the WMS support is in best effort way
- Sooner or later we should decommission it
- Evaluating alternatives:
  - DIRAC
  - Glidein-WMS
- DIRAC seems more integrated to the EGI infrastructure, so we are exploring this case

DIRAC overview

- Framework for submitting pilot jobs on computing resources
- Usually a central server for:
  - Holding the queue of pending payloads (jobs)
  - Matching the resources to payloads
  - Usage of the Pilot job paradigm
- Three components are necessary:
  - The Pilot Jobs
  - The TaskQueues
  - The TaskQueue Directors

Pilot jobs

- Empty resource reservation containers submitted to the available resources with the aim of executing the most appropriate pending payload in the central queue
- Perform basic sanity checks of the running environment before any binding with a given payload

Pilot jobs (II)

- *In situ* DIRAC installation (on WNs), with the most recent version of the configuration
  - Checking the working conditions: location of execution, available disk space, memory, cpu, Grid environment, running platform, …
- Execution of a DIRAC Job Agent:
  - Request the payload to the central DIRAC WMS and later executes it
  - Instantiates a Job Wrapper object for the payload execution
  - Instantiates a Watchdog process for monitoring the Job Wrapper (and payload) status, taking the proper actions in case of problems and reporting to the central WMS
- The Wrapper retrieves the input sandbox, checks the input data and software, executes the payload, report success or failure, and eventually uploads output sandbox and output data if required

TaskQueues

- A sorted groups of payloads waiting for execution with identical requirements to the possible Pilot Job requesting a match
- Among the requirements:
  - the user identity submitting the task
  - the amount of computing power requested
  - the computing platform required (if any)
  - the eligible or banned sites for the payload (if necessary)
- Priority assigned by the user at submission time for each task in the TaskQueue
- Overall group priority for each user group

TaskQueues (II)

- A new TaskQueue is created when a new payload that does not fit in any of the existing ones enters the system
- A TaskQueue is deleted when all pending tasks (payloads) have been removed by Pilot Jobs
- Priorities are consequently evaluated again
- Two steps matching resources-to-payload:
  - Found an appropriate TaskQueue
  - Select a pending payload and remove it from the TaskQueue

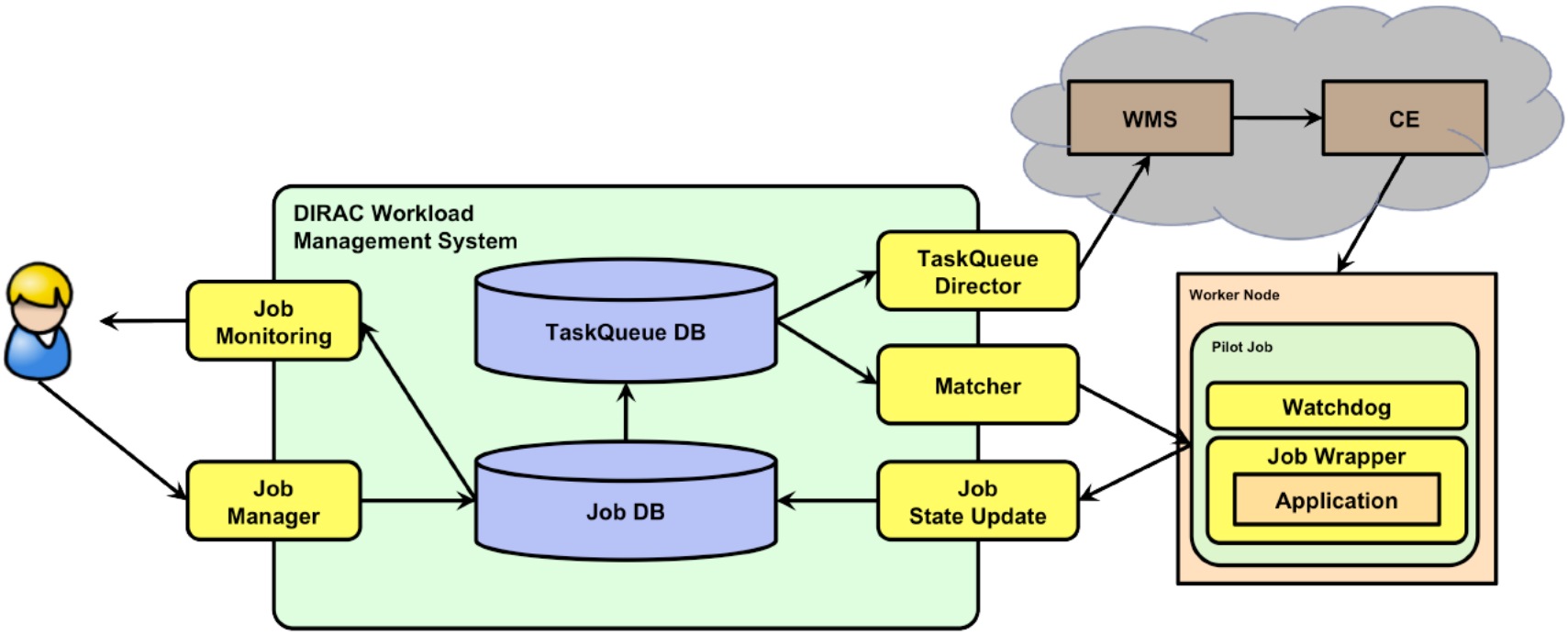
TaskQueue Directors

- Responsible for populating the computing resources with appropriate Pilots Jobs able to match the pending payloads
- work in cycles repeating the same logic on each iteration
  - (I) Query the central DIRAC WMS server to get a list of TaskQueues
  - (II) the Director attempts the submission of a configurable number of Pilot Jobs per iteration
    - Depending on both the priority of the TaskQueues and the total number of pending tasks on each of them
    - For each TaskQueue the availability of matching resources is checked by looking at the number of free slots on the queues
  - (III) It is evaluated the number of Pilot Jobs to be submitted for the TaskQueue and the submission is attempted

TaskQueue Directors (II)

- Each Director Agent is accompanied by a corresponding Pilot Status Agent
  - Monitors the submitted Pilots, flagging them in case of failures and declaring them stalled if they have not managed to execute after a certain amount of time
- Any Pilot Job receiving a payload for execution is immediately flagged in the central WMS server
  - it is essential preventing the system from stalling in case of bad resources, failing all Pilots received

DIRAC Workload Management System



Generic and Private Pilots

- Generic pilots: submitted with a generic identity (for instance a robot certificate)
  - Resource reservation made by (all the) VO (using the generic identity)
  - No information about the owner of the jobs on the chosen resources
- Private pilots: submitted with the user identity
  - Resource reservation made by user
    - Little decrease of performances
  - Information about the jobs owner is available on the chosen resources

Preferable usage and testing

- EGI would prefer using the private jobs for security and accounting reasons
  - With the opportunity in using the PUSP in future
- The idea is contacting some of the communities that are currently using the WMS for testing the DIRAC framework
- DIRAC developers suggested to contact VOs with higher number of jobs (pamela, pheno, vo.complex-systems.eu, glow, …) or alternatively with higher user of normalized CPU hours (virgo, theophys, compchem, auger, …)

Documentation

- The reference paper: https://indico.cern.ch/event/42258/contributions/1026691/attachments/890467/1253242/pilotFramework.pdf
- Pilots presentation: https://indico.cern.ch/event/161080/contributions/1413980/attachments/185973/261255/DIRAC-Pilots.pdf
- Dirac website: http://diracgrid.org/

Thank you for your attention.

Questions?

www.egi.eu