Use of the Information System in WLCG Experiments

Davide Salomoni, INFN

EGI Workshop “Towards an Integrated Information System”

December 1, 2011

Why I am here today

- In Summer 2011, the WLCG Management Board decided to set up some "Technical Evolution Groups" (TEGs).
  - Their mandate is "to document a strategy for the evolution of the technical implementation of the WLCG distributed computing infrastructure."
  - And "to provide a clear statement of needs for WLCG, which can also be used to provide input to any external middleware and infrastructure projects."
  - Deliverables are due in early 2012.
- One of these TEGs is the Workload Management TEG (WM TEG) – others TEGs deal with Data Management, Storage Management, Databases, Security and Operations.
  - Among its several tasks, the WM TEG covers the topic "Information Services".
  - Torre Wenaus (BNL, ATLAS) and myself are the two chairs of the WM TEG.
  - Besides reporting here on the current status of the Information System for WLCG experiments, we are interested in understanding how the IS is thought to evolve.

Input from the WLCG experiments

- Baseline for usage of the IS in WLCG comes from the document "WLCG Information System Use Cases"[1].
- Then from updates provided at the May 2011 Grid Deployment Board (GDB) meeting[2].
- Then from face-to-face meetings and discussions taking place as part of the WM TEG work[3].

- Foreword: all the 4 WLCG experiments (Alice, ATLAS, CMS, LHCb) have developed their own software framework, typically pilot-based.

1. https://twiki.cern.ch/twiki/pub/LCG/WLCGISArea/WLCG_IS_UseCases.pdf
2. https://indico.cern.ch/conferenceDisplay.py?confId=106644
3. Work in progress, wiki at https://twiki.cern.ch/twiki/bin/view/LCG/WorkloadManagementTechnicalEvolution

Alice

- Alice use the BDII to regulate the flow of jobs to a particular site through their pilot-based AliEn framework.
  - However, the list of CEs to be used is hard-coded.
- And to identify which CEs are in production mode (CE status).
  - I.e. only production CEs are used to submit jobs.
- The Alice VO-box (a system within a site boundary and responsible for job submissions to local CEs) checks the CE BDII to verify site occupancy.

ATLAS

- ATLAS maintain a cache of experiment-specific info for their software components (PanDA, dashboards, etc.)
  - This info is collected from the BDII, but also from other sources (GOCDB, the OSG Information Management System, other services).
  - It involves static and quasi-static information (like downtimes, queues being set offline, blacklisted sites).
  - BDII is used by PanDA (the ATLAS software framework) to discover and keep current the list of known endpoints/sites. The most dynamic part is SoftwareRunTimeEnvironment and the status of CEs.
  - BDII is periodically scanned and its info cached.
  - ATLAS maintain a site configuration db in Oracle – they may decide to add site attributes to this database when needed; for example, recently, fields for controlling many-core queues were defined.
  - A common source of problems is related to publication of disk space (it would be desirable to know how much is *in use* and how much is *available* – not necessarily coincident with how much is *installed*).
  - ATLAS relies on services such as FTS and LFC; these services query the BDII, which must therefore work reliably.

CMS

- The BDII information used by CMS is quasi-static.
- E.g. in CRAB (WMAgent) there are queries for CE status, SoftwareRunTimeEnvironment, CEUniqueId [static match for inclusion/ esclusion], OS version, but typically with a “trust but verify” model.
  - The list of sites is not automatically updated based on BDII info – they verify site functionality before enabling mass submissions.
  - Ditto for site attributes: they are not auto-updated (and, like ATLAS, they may define custom site attributes).
- The requirements are for relatively basic items, which should be easy on the sites to operate and not error-prone.
- CMS do not use dynamic info (slot utilization, system usage), which were found frequently unreliable / out-of-date. They suggest it is better to have a fast, simple, reliable, quasi-static IS. It is also questionable how much benefit would pilots gain from dynamic info.
  - E.g. they validate nodes directly before their “glideins” start on a given node. This avoids most “black-hole” problems.
- CMS also use services like FTS, which rely on the BDII.

LHCb

- DIRAC (the LHCb software framework) does not basically use the BDII.
  - DIRAC (like other frameworks) incorporates its own workload management system.
  - Endpoint info (e.g. list of CEs) is statically defined in the DIRAC Configuration Service.
- Like others, LHCb use services like FTS, relying on the BDII.

Quick comments

- WLCG experiments all have developed their own software frameworks and tend to use BDII for static/quasi-static information, and in general for limited purposes – the pattern is often “use for bootstrap, then refine with our own heuristics”.
- Quality control of the IS content needs to be automated. WLCG experiments have learnt by experience that no info is *at least* not worse than unreliable dynamic information.
- Pilot-based frameworks adopt late-binding of jobs to slots. How a more dynamic BDII would benefit pilot-job based frameworks is to be seen.
- Cached info is vital.
- Reliable storage information would be desirable, but generally it is currently not available.
- We need to understand how and if the current way of interacting with the IS may be possibly simplified or made more uniform across the experiments. This is part of the ongoing WM TEG work and a motivation for being here today.
- Other, future services related e.g. to the integration of Cloud resources might conceivably use the IS – however, it is still early days there. In the WM TEG we are certainly interested in following, for example, the work of the EGI fedcloud-tf on this.