Pre-GDB on CE Extensions
Review of the Proposed Extensions

Davide Salomoni, INFN
July 9, 2012

The WM TEG Final Report

- https://espace.cern.ch/WLCG-document-repository/Technical_Documents/
- For what regards CE Extensions:
  - Chapter 4, Commonalities in Pilot Frameworks
    - In particular, the part on “streamed submission”
  - Chapter 7, I/O-bound and CPU-bound Tagging
  - Chapter 5, Support for Jobs Requiring Whole Nodes or Multiple Cores

Streamed Submission

- Extend the CE interface to support a "streamed submission service"
  - Constantly keep N jobs queued at a given queue until a given condition is satisfied (e.g., no more work to be done, or until a certain time)
  - With a JDL expression specifying requirements shared across the "streamed" jobs.
  - This includes associated operations to list, kill and update the submission frequency.

I/O-bound and CPU-bound tagging

- Goal: give sites more options on how/where to schedule resources
  - I/O throughput on storage systems has certain (site-dependent) optimum and maximum levels
    - Exceeding these levels can bring to inefficiencies (for one or more VO's) or even to storage instabilities
- If jobs are tagged (by VO's) as being "more CPU" or "more I/O" bound, and if this information is propagated to sites, then sites may be able to apply decisions on how to best handle them
  - Rather than relying on simple (or even not simple) heuristics
  - E.g. spread jobs declared to be I/O-intensive on multiple nodes

Whole-nodes / Multiple cores (WN/MC)

- A possible, existing solution is to use dedicated queues; this has been discussed in the report.
- Rather than relying on this "static" solution, the WM TEG believed that a better solution is to let VO's specify – and CE's handle – explicit requirements on what type of resources are needed.
  - This is typically a site requirement, especially coming from sites supporting more than one VO (and perhaps non-WLCG VO's as well)

Considerations/Requirements

- The solution to the WN/MC problem should not unnecessarily complicate a site configuration.
  - E.g. through deployment of solutions that are “unique” to WLCG
- As many LRMS’ as possible should be supported
  - However, the detailed configuration of LRMS’ should be left at site level. E.g., advance reservation, if needed, should not be provided by CE’s.
- The solution should be applicable to both early-binding and late-binding jobs or workloads.
- Details are to be expressed in the JDL associated to the job.
- Last but not least, the solution must be implementable and deployable in finite time, a.k.a. ASAP.

Use cases

- "I want a total of X GB of RAM for my job" – applicable to all the following cases.
- "I want a whole node"
  - This may be a real (physical) node or a virtual machine – the point is that with "whole-node" requirements all the hardware (real or virtual) visible to the job is made available to the job itself and only to that job.
- "I want my job to have N cores" (N being a fixed number.)
- "I want my job to have a mimimum of N and a maximum of M cores" (i.e. requirement of a variable number of cores.)
  - Typically, "give me all the cores you can give me, and I'll use them"
  - How about memory requirements in this case? The job could specify the minimum amount of memory *per-core*. The total memory would then be this amount times the # of allocated cores.
    - How does a job know how many cores has it been allocated in a shared environment? This is the role of the *enviroment variables* discussion.
- Other use cases / requirements might be useful, e.g., "I want a minimum amount of local disk space", but are probably best left for a later stage.

What now?

- Need to define:
  - How can this be supported by the IS
  - How can this be supported by CE technologies
  - What (if any) implications for other subsystems are there (e.g., accounting)
  - A timeline for implementation and testing.