Job submission requirements

Dennis Box, Joe Boyd, Rick Snider
on behalf of REX

Definitions and context
Basic objectives
Requirements

NuComp meeting
Fermilab
Oct 13, 2010

Definitions and resources

- Computing element (CE)
  - ◈ A grid headnode + the associated worker nodes serviced by that node
- “Local batch” (via General Physics Computing Facility)
  - ◈ Pool with priority access to sub-set of machines per experiment
  - ◈ To be used mainly for small-scale projects/testing
  - ◈ All experiment-specific resources available (home areas, data/sw disks)
  - ◈ Not the grid (?)
- “GP grid” (aka “local grid” or “Fermigrid”)
  - ◈ Various centrally maintained CEs
  - ◈ Most experiment-specific resources available (no home areas)

Definitions and resources

- Other resources on Fermigrid (a seriously overloaded term…)
  - On-site CEs purchased specifically for other experiments
  - Opportunistic or pre-arranged access
  - Reasonable connectivity to experiment disk, but no guarantee of mounts
- Grid at large (aka “OSG”)
  - CEs, storage accessible only via fully grid compliant mechanisms
    - Should generally assume high-latency, intermittent connectivity
  - At least two flavors:
    - Collaborating institutions with priority access rights
    - Non-collaborating sites with only opportunistic / pre-arranged access
  - Typically some work to integrate remote sites into an experiment

What does “job submission” include?

- “Job submission” cuts through several layers
  - Submission client
    - What end users see: includes the feature set and user interface
    - Independent of underlying batch system(s)
      - Can provide a uniform way to access all available resources
  - Job submission and management infrastructure
    - Talks to the submission client and the batch system
    - May include pieces that live on several machines + pieces submitted with the job
    - Exploits features specific to a particular batch system
  - Batch system configuration
    - Provides features to support management of resource utilization

  Note that the relevant “batch system” may not be the one in operation at a given site.

    - Can “overlay” one batch system on another

Some notes on the discussion that follows

- Cast in declarative terms, but really seeking input as much as trying to present a vision
- Attempted to abstract the requirements from any given batch system
    - ◈ Have adopted some of the language of Condor for conceptual purposes (although direct mapping of conceptual requirements onto actual features is obviously a good thing).

Read the requirements slides in the following way:

- A requirement (the big bullet items)
    - ◈ Discussion, considerations, issues, examples

      but not “requirements” in all the other bullets

Basic objectives

- For end-users
  - To provide access to distributed (grid) computing resources
    - “local” resources in this context = one instance from a set of grid resources
  - To simplify the task of utilizing these resources to solve complex or large-scale computing problems
- For experiment management
  - To allow experiments to manage utilization of the available resources to meet physics objectives
- For computing system operators
  - To provide mechanisms to manage utilization of the available resources in order to maximize computing throughput
  - To minimize the effort required to do so across multiple experiments

The underlying assumption: limited computing resources available

Job submission requirements

(in no particular order)

- Common submission client for all IF experiments
  - Isolates users from direct interaction with batch system
    - Allows uniform interface (although options may differ between experiments)
  - Provides primary mechanism for simplifying complex job submissions
    - Automatically generate submission configuration files for particular use cases
    - Ex: jobs that require pre-staging of data from tape before processing begins
    - Ex: jobs that require certain steps to complete before others can start
  - Allows instrumentation of job submission
    - Collect monitoring, debugging data beyond that of the underlying batch system
      - For example, “your job died because you used this switch incorrectly, try this instead”
    - Collect data for usage analysis
      - May require application-level information
- Common submission infrastructure for all IF experiments
  - Mostly same as above
  - Reduces required support load

Job submission requirements

- Provides support for steering of jobs to specific resources
  - GPCF, GP grid, Fermigrid, OSG
  - Specific CEs or sets of CEs when useful (eg, a particular OSG site)
    - Reasons could include testing or the location of a resource or dataset of interest
    - A critical feature during times of OS migration

Job submission requirements

- Supports the concept of “groups” for accounting and priority
  - Need to distinguish members of different experiments (ie, VO membership)
    - Provide priority access to particular machines
    - Provide certain number of slots with priority access on a given CE or set of CEs
    - Limit opportunistic users
  - Provide a structure for experiment to manage limited computing resources
    - Define special groups for various types of processing, for instance:
      - Service groups for centrally managed data production, MC production
      - High priority groups for rapid processing for certain jobs
      - Low priority groups for things that should only run if absolutely nothing else needs CPU
      - Etc.
    - Can set slot limits, steer to particular resources, etc, based upon group
    - Users select the submission group. Several default groups available to all users.
    - Experiment management can set high priority group membership

Job submission requirements

- Supports specification of resource requirements “external” to the job
    - Input and output data sources
    - Required access to experiment code base, etc.
    - User-imposed limit on number of simultaneously executing jobs
    - Approximate job execution time
        - Could allow limited number of short jobs to execute with higher priority than otherwise equal long jobs
        - Need at least a “test queue” for very small number of very short jobs

These specifications can be used for job steering, throttling, or other resource management algorithms

Job submission requirements

- Supports job ordering dependencies
  - For example, pre-staging data files from tape prior to executing the jobs that consume them
  - REX will implement and support experiment-defined workflows when possible and appropriate
- Supports logging of job submission information not available via batch system
  - Needed for operations, resource management, and planning
- Operational requirements
  - Not yet defined, but are considering how / if to define requirements for:
    - Deployment
      - Eg, shouldn't need stop everything just to update the code, or change configuration
    - Robustness

Job submission requirements

- Provides extensible and maintainable code base
    - Extensible in the sense that (these *are* requirements):
        - Experiment-specific customization do not require modification of core code
        - Submission configuration adaptable from the command line
            - Allows rapid adaption to changes in underlying batch system (usually out of our control)
- Returns error messages that users can understand, respond to
    - Easier said than done...
- Provides tools to assist with tarball creation
    - Will be a necessary part of working on the grid...
- Provides sensible defaults so that the most simple command is almost always the correct one to use
    - Most users, most of the time

When do I get all this?

- Short term goals
  - Provide basic functionality
    - The system evolving from *_jobsub does this
  - Agree on the requirements
- Intermediate goals (work in progress now)
  - Re-write and unification of 'jobsub' scripts (in beta now)
  - Extensibility provided via sub-classing for each experiments
  - Will provide easy transition for users
- Longer term
  - Infrastructure that ships monitoring suite with the user job
  - Everything else

Need to work out details of how to proceed when done with requirements