User View of Fermilab Batch Computing

Fermilab Batch Computing is implemented in several different workflows. Figures 1 and 2 diagram three methods for connecting experiment jobs with compute resources: CMS workflows, FIFE workflows, and direct submission. The simplest of these models is the direct submission wherein an experiment such as DES submits jobs to compute elements by issuing HTCondor commands directly. This method consists of user commands connecting directly with a single compute element and is therefore limited to that single cluster (in this case the Fermi GP Grid cluster). It requires that experimenters learn HTCondor commands and actively avoid potential pitfalls. HTCondor establishes the order in which jobs will start and matches job requirements (memory, CPUs, disk) to available computing resources. Within this implementation there is no ability to dynamically go beyond the confines of the dedicated compute cluster.

For almost all experiments with distributed-computing needs (CMS, NOvA, MicroBooNE, DUNE, etc.), additional layers are introduced between the experiment and the distributed resource pool. The purpose of the additional layers is to automate common tasks (data handling, resource provisioning and partitioning, authentication, etc.), allow service providers to negotiate with numerous compute elements behind the scenes using GlideinWMS, and sanitize the interaction with the HTCondor pool. The diagram shows three custom workflow management systems between the user and the HTCondor pool: FIFE, WMAgent, and CRAB.

WMAgent (production) and CRAB (end users) are utilized by CMS as a layer of abstraction that communicates with the HTCondor pool performing tasks such as authentication, data discovery, and automating the configuration files for resource provisioning with the HTCondor pool. At the next layer, GlideinWMS allows for the dynamic incorporation of additional compute resources (e.g. WLCG and OSG Sites, commercial clouds) and the configuration of those compute elements to accept jobs from the HTCondor pool.

The FIFE workflow is designed as a layer of abstraction similar to WMAgent and CRAB that also conglomerates the submissions from more than 20 VOs into a single service and workflow. FIFE jobsub performs automatic authentication, file transfers, configuration of resource provisioning requests, and integration of data handling interfaces (SAM) and tools (ifdh). The FIFE workflow also leverages GlideinWMS to dynamically provision a diverse set of distributed compute resources to the HTCondor pool ~~from traditional grid sites, HPC clusters along with resources from paid clouds such as AWS and community clouds. The FIFE Workflow requires greater flexibility in order to adapt to some of the specific requirements of each VO.~~

The Fermilab HEPCloud Facility extends the concept of delegated resource provisioning by granting the site additional capabilities to acquire resources on behalf of experiments. The design calls for the establishment of resource provisioning policies and a decision engine that will implement these policies. The Fermilab HEPCloud Facility will evaluate supply, demand,

capability, and cost of resources to determine which resources will be allocated for an experiment's computing needs.

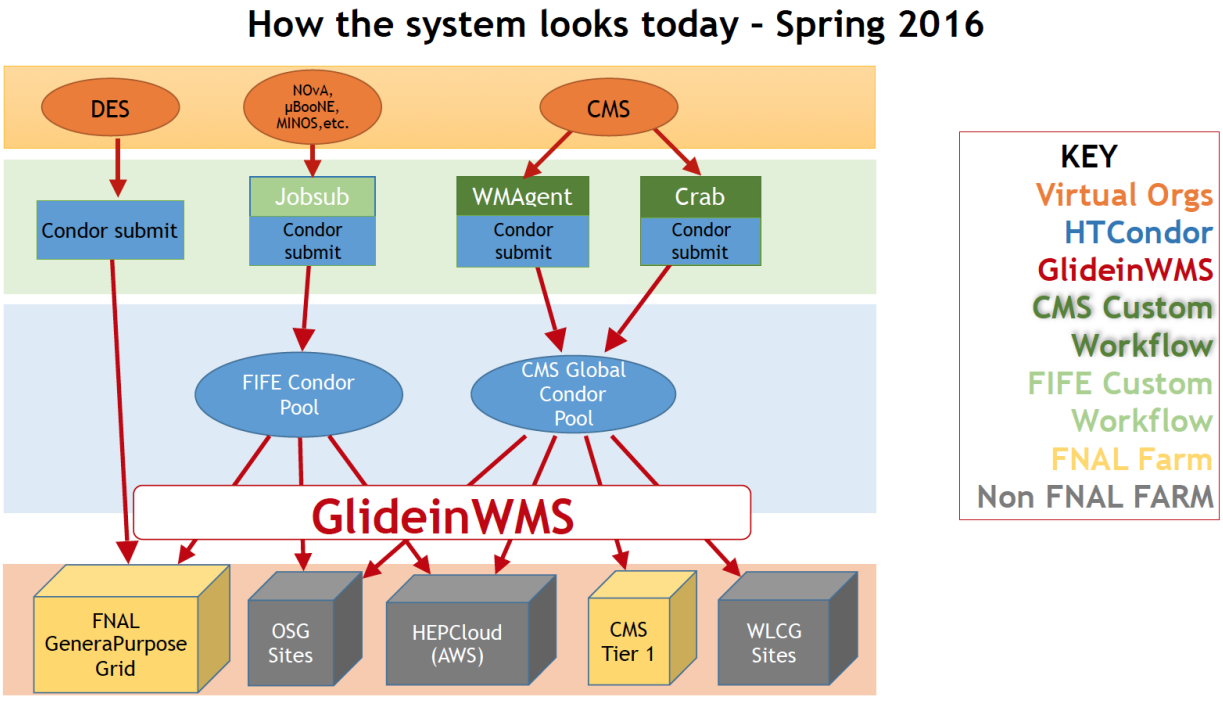

Figure 1. The diagram shows three of the batch computing workflows currently used at Fermilab.

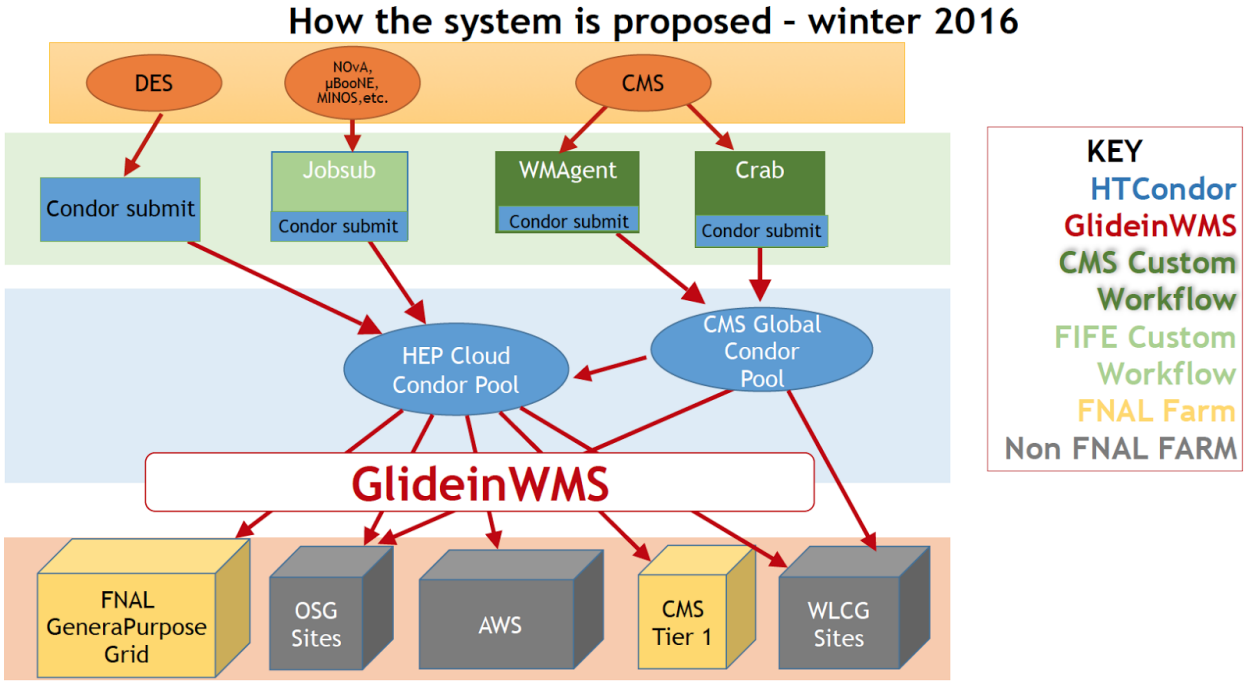

Figure 2. The diagram shows the proposed reconfiguration of batch computing to take advantage of the HEPCloud soon to be implemented. HEPCloud will include a resource provisioning policy and decision engine to incorporate additional resources based on need.