Batch Submission System
For Intensity Frontier Experiments

Dennis Box, Stephan Lammel

2012-Jun-28
v0.9

In particle physics most data processing, selection and large scale analyses are done on Grid facilities. For convenience users have wrapped the more complex Grid commands in scripts to simplify the submission and status checking of jobs. The sophistication of such wrapping varies among experiments, wrappers are very custom and re-use within the Intensity Frontier community has been by cloning the scripts. This has led to a large collection of rather similar utilities distributed across experiments and user areas that need to be updated in case of Grid middleware changes and enhancements.

Enabling new Grid features and implementing a common data handling approach for current and next generation neutrino experiments have highlighted the problem. The computing Sector of Fermilab is responding to this with a web-based monitoring and a common job submission system inside the FIFE (For Intensity Frontier Experiments) project.

Introduction

Job submission to the Open Science Grid (OSG), as overseen by the Fermilab Running Experiments (REX) department, is accomplished through suites of software used to steer work flows and data throught distributed performing desired scientific analyses. The software used for these purposes have been developed by many organizations, both internal and external to Fermilab.

Each experiment has chosen from among the tools that existed at the time to pick the ones best suited their needs. These were crafted into their own unique submission systems with their own advantages and idiosyncrasies.

This proliferation of submission systems presents a challenge for a manager or operator trying to understand the overall state and behavior of the system, or a user or technician diagnosing bottlenecks or failures. A scientist transferring to a new experiment is expected to learn the new analysis framework, but usually has to master new submission and monitoring system as well.

Experiments have invested years of effort in their infrastructure, analysis framework and submission interfaces. They work. As experiments wind down, there are dwindling resources for changing what is seen as successful and useful interfaces. At the same time, the infrastructure the experiments run on will continue to evolve, as hardware goes out of warranty and is replaced, vendors discontinue support or go out of business, and security patches change expected behavior. This plan addresses the problem of managing a diverse user community using different interfaces as the underlying infrastructure the interfaces use changes over time.

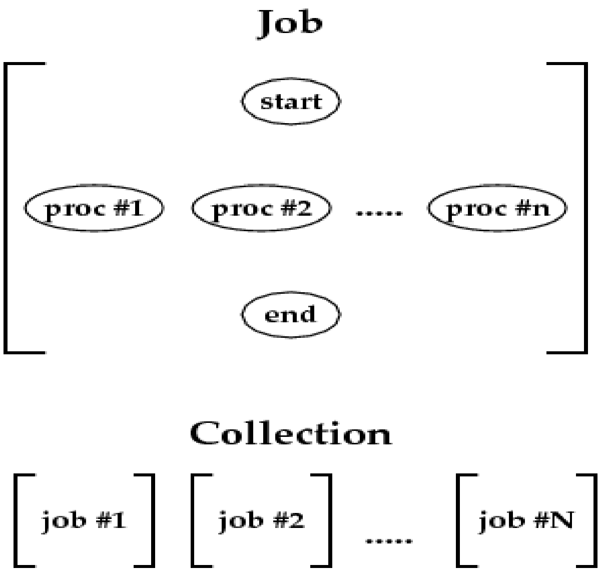
I. Overview of the Batch Submission Systems

The batch submission system will consist of two parts: a client part, independent of OSG, LCG or gLite tools that will be packaged for easy installation at on-site and off-site servers, desktop or user laptops and a server part that will reside on a FIFE gateway machine at Fermilab.

The batch submission system will connect with the user on one side and Grid middleware on the other side during job submission and during job execution with Grid middleware and user script/executable. The batch system will also provide simple binding to the REX data handling system SAM (to start up and shut down file delivery for jobs processing a SAM dataset).

The batch submission system will handle transfer of job script/executable and configuration files from the submitter host to the job-executing worker node. It will handle transfer of log files and tar up plus transfer of left-over files on the worker node. The batch submission system will also provide users with the option of a notification when the job finished.

The batch submission system will support the most common job structures of particle physics, that is provide repeated, potentially parallel execution of the same script/executable. Each such instance is called a processing section. The instance is flagged to the script/executable via environmental variable. Especially for data handling jobs a one-time starter section proceeding any processing section (and one trailer section executing after all processing sections complete) is desirable. A combination of zero or one start and end section and one or more processing section is called a job. To repeat the same job, for instance to run the same analysis/selection job over multiple datasets, the batch submission system will support job collections, i.e. multiple execution of the same job (each with zero or one start, one or more processing and zero or one end sections).



The batch submission system will assign unique identifiers to each job and collection and provide tools to track progress based on the identifier. (Start and end sections can be NOPs.) Users can specify options for the job sections during submission. The batch submission system will pass them to the script/executable of the start, process and end sections.

The batch submission system will allow users to select the batch farm the job/collection is executed on, specify resource requirements like memory, OS, scratch space, etc., chose between batch queues and make the job/collection known to the FIFE monitoring system.

The batch submission system will provide accounting based on experiment (and group). Experiment (and group) membership may be restricted and require authorization. Tools for managing membership will enable delegation to the experiments.

II. Current Job Submission

Intensity Frontier Job Submission

Intensity Frontier experiments modeled their submission systems on the architecture pioneered by the MINOS experiment. Batch computing resources both on local clusters and on on-site Grid systems are used. A condor control file is generated by a script that accepts input and output directives as arguments. As new IF experiments began using the Grid, they copied MINOS's scripts and modified them to their own needs. This resulted in a plethora of scripts, minos_jobsub, nova_jobsub, minerva_jobsub etc. OSG Grid proxies are generated and refreshed by cron scripts that are the responsibility of each individual users to set up.
An effort is underway to unify these scripts into a single 'jobsub' script (1), re-written in an object oriented manner using plugins to manage experiment-specific needs.

Chaining of jobs is handled through a separate application, a DAG generator where users chain together jobsub commands in an xml style control file which generates condor DAG control files. The system is in a prototype stage. It was designed to handle arbitrarily complex workflows. (2)

IF job submission is currently tightly coupled to the central network attached storage provided by a Bluearc server, which means that Grid sites outside Fermilab cannot currently be used for IF jobs.

CDF Job Submission

The CDF experiment has been running for over 20 years and its submission interfaces are consequently well developed. All batch computing resources are Grid systems, on-site and off-site, using OSG and LCG middleware. A client server model is used, where users submit jobs through the interface on the client, which then contacts a server which constructs condor control files which are in turn submitted to condor. Monitoring and job control (kill, hold, resume) commands also have client interfaces which talk to the server which in turn talks to the batch system. OSG Grid proxies are the responsibility of the server in this model, a valid Kerberos ticket is required from the client before the server will honor a request, and once honored, the Grid proxy is generated and then kept alive for the lifetime of any user jobs the server controls.

Job dependency chaining at CDF is handled by the client submission interface and is restricted to a simple use case: starting a SAM file delivery project, running N jobs to consume the files, and a post-execution process to end the SAM file delivery.

One added complexity to the CDF system is that the server must be able to obtain Kerberos principals for all users in order to generate Grid proxies for them. This requires some security measures not seen on many batch systems:

- A security exemption allowing the server machine to have special Kerberos principals that provide user identification but are owned by the admin account (cdfcaf).
- Infrastructure for generating and maintaining Grid proxies of users
- Restricted access to the server machine. Users can only interact with server through the client interface.

CDF job submission is less tightly coupled to condor than IF submission. Condor is not the batch system that was originally used. As a consequence, all direct interfaces to condor are only from the server, changing to a different batch system would not require changing client code or interfaces. As all of CDF uses the same interface to submit to batch systems, and the underlying submit infrastructure has evolved over the last 15 years (a geologic age in computing), the CDF interface is already evolvable and adaptable.

D0 Job Submission

The D0 submission system evolved, as in the case of CDF, from large multi-processor UNIX machines. Batch computing resources on both local clusters and on on-site and off-site Grid systems are used. The local clusters run Portable Batch System (PBS) software with small enhancements, for instance, to provide Kerberos ticket handling. For job submission users use a wrapper script that deals with parameter setting, transfer of configuration input files, etc. For job control and monitoring PBS commands are used directly. On-site Grid resources are used for data processing and on- and off-site Grid resources for Monte Carlo generation/simulation. Grid submission is used by only very few users with custom scripts. For Grid jobs a glide-in workflow management system (glide-in WMS), provides forwarding of jobs to both OSG and LCG Grid sites.

III. Submit Command Options

The job submission system provides middleware for convenient submission of typical HEP jobs to the Grid. The FIFE submit command has three types of options: 1) options to specify job/collection configuration input files, to control log and output tar files and notification; 2) options to describe the job or collection work flow structure/parameters and 3) options to list extra resources and specify experiment/group and queuing parameters.

1. Config, Log and Tar File Options

-i job/collection configuration input file(s)

-t job/collection config input tar file (tar must contain start/proc/end section scripts/executables)

-o job/collection output tar file URI (default FIFE_DEPOT/<job/collection id>/<section id>.tar)

-l job/collection log file URI (default FIFE_DEPOT/<job/collection id>/<section id>.log)

-m email address for job complete notification

2. Job/Collection Work Flow Options

-s script/executable of start section

-x script/executable of process section

-e script/executable of end section

-n number of processing sections in a job

-N number of jobs in the collection (default is -N 1, i.e. submit a job not a collection)

-X arguments to be passed to start, processing and end section scripts/executables

-S dataset name in case of SAM data handling (defaults start and end sections to SAM)

3. Resource and Queuing Options

-G experiment/group specification

-Q queue name

-R list of extra resources required

-T time limit (default based on queue; triplet to allow different start/proc/end section)

-F farm selection

-H hold the job/collection

-A execute the job/collection only after another job/collection finished

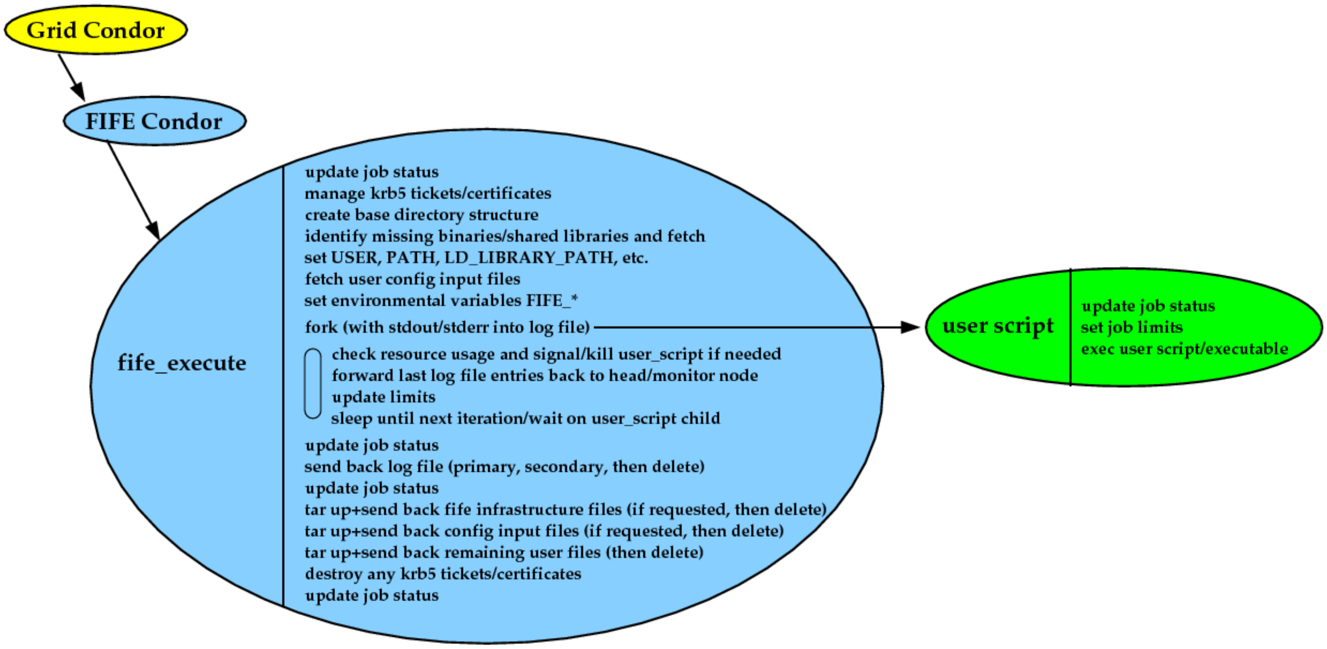
-v verbose

-V version

VI. Section Execution Wrapper

The batch submission system provides middleware for convenient execution of typical HEP jobs on the Grid. The batch submission system sits thus between OSG/LCG Grid worker node starter process and the user script/executable being run. The queuing system of the FIFE batch submission system is separate from the OSG/LCG Grid queuing/batch system. (Both are currently based on the Condor High Throughput Computing package.) The OSG/LCG Grid batch system will schedule the worker-node-startup process of the FIFE batch submission system. This process will then advertise its resources and receive a section to run.

An execution wrapper handles configuration input file fetching, manages authentication tickets and certificates, provides job status updates to the FIFE monitoring system, watches and updates job resource usage and handles transfer or log and any left-over files after the job finished.


Grid Condor
FIFE Condor
fife_execute
update job status
manage krb5 tickets/certificates
create base directory structure
identify missing binaries/shared libraries and fetch
set USER, PATH, LD_LIBRARY_PATH, etc.
fetch user config input files
set environmental variables FIFE_*
fork (with stdout/stderr into log file)
check resource usage and signal/kill user_script if needed
forward last log file entries back to head/monitor node
update limits
sleep until next iteration/wait on user_script child
update job status
send back log file (primary, secondary, then delete)
update job status
tar up+send back fife infrastructure files (if requested, then delete)
tar up+send back config input files (if requested, then delete)
tar up+send back remaining user files (then delete)
destroy any krb5 tickets/certificates
update job status
user script
update job status
set job limits
exec user script/executable

V. Implementation Plan

In the above sections the current and envisioned, future job submission systems of the Run II, neutrino and new IF experiments have been described. Immediate action is required for some of the current job submission systems to counteract the proliferation of cloned jobsub scripts and to add requested new features. The new FIFE job submission system will be developed in stages and current job submission systems migrated into it/current jobsub scripts connected to it.

1. The first stage is to version, package, test and release a FIFE job submit command and set of experiment wrapper script. Included in this package will be the existing DAG generator scripts. The package will be made available through ups/upd and experiments encouraged in installing/using it.

2. Data handling example jobs based on the IFDH package of the REX/DH group will be provided together with default start/end sections for file delivery based on SAM datasets. Experiments will be coached on how to best use the data handling tools, advantages and disadvantages of file list processing versus SAM dataset, etc. This stage will effectively ‘gridify’ data access allowing data analysis from worker nodes without BlueArc mounts.

3. Direct usage of condor commands will be monitored and analyzed and additional status checking, job management or monitoring commands developed as needed. They will become part of the FIFE job submission package.

4. To improve job status, monitoring and error handling the section execution wrapper will be fully developed. This will include fetching section scripts/executables and configuration input files.

5. An output tar file server will handle temporary storage for job output, unload “scratch” activity

from the BlueArc and equalize on-site and off-site execution further. This output server will provide disk space to spool batch left-over and log file output. Users will have a quota with large overdraw option, a brief, ~72 hour, grace period after which automatic cleaning will keep the area from filling up.

6. In the final stage, job submission will be switched to a client—server system. Condor software installation will be removed from all the central interactive machines. Submission systems will be consolidated. The goal is to have one submission system for all remaining Run II and current/future IF experiments.
7. A post stage of the job submission system development is working with each experiment, decoupling job executables/scripts from BlueArc filesystems, reviewing data access and data flow of their various jobs and thus fully 'gridify' the jobs.

Detailed Plan Task List

NB only covers steps 1 and 2 above. Steps 3-7 are under review for management approval.

1. KITS product will be created, tentatively named jobsub_tools v0_1

1.1. will contain jobsub, setup_condor.sh, dagNabbit.py condor_wrappers [Dennis -nearly done]

1.2. documentation updated on wiki [John,Dennis]

MILESTONE jobsub_tools v0_1: a productized jobsub and DAG generator useable by experiments
Duration: 1 week. Completed.

1.3. Refactor jobsub into core, pre, and post sections

a) refactor current jobsub to use standard library python modules [Dennis]

b) refactor current setup_condor.sh to use pre and post.sh (needed?) [Dennis]

MILESTONE jobsub_tools v0_2: a productized jobsub and DAG generator useable by experiments. Test suite verifies that tested functionality is correct.
Status: under way, completion target early July 2012.

2. IDFH incorporation into jobsub and DAG generator [Mike Joe Dennis Marc]

2.1. cpn and srmcp replaced with idfh cp

a) idfh SAM interface used

b) idfh monitoring/error logging interface used.

c) SAM getnextfile loop implemented

- handles errors gracefully
- talks to monitoring/logging about its status

2.2. Dag generator uses idfh to complete requirements listed in reference (2) specifically:

- station startup given dataset name
- station cleanup

MILESTONE jobsub_tools v0_4: jobsub and DAG generator undestand SAM
Duration: 1 Month . Started.

2.3. Experiment switchover.

a) 'secret handshake' implemented in jobsub, not turned on at condor/glideinwms level

b) translation scripts – minerva_jobsub to jobsub awaiting on approval of steps 3-7

Risks

This project depends on and is coupled to the ifdh product. If this product is delayed then step 2. in the detailed task list will be delayed as well.

Users may reject the idea of being forced to use released versioned software and reverse engineer the secret handshake, then continue using custom scripts with condor_submit embedded in them.

Conclusions

When this project is complete we will have a single submission system that is maintainable, and more capable of using grid sites and features than what we have now.

References

(1) Requirements for the Grid job submission system for the Intensity Frontier experiments
Dennis Box , Joe Boyd , Rick Snider V2, 1-Nov-2010
https://cd-docdb.fnal.gov:440/cgi-bin/RetrieveFile?docid=4082;filename=if-job-sub-requirements-v1.pdf;version=10

(2) Requirements for DAG submission system for Intensity Frontier experiments
Dennis Box, Joe Boyd, Rick Snider, April 2012 (DRAFT)