A Lightweight Submission Frontend Toolkit - HepJob

Xiaowei JIANG[1,*], *Ran* Du[1,**], *Jingyan* Shi[1,***], *Jiaheng* Zou[1,****], and *Qingbao* Hu[1,†]

[1]Institute of High Energy Physics, Chinese Academy of Sciences

Abstract. A typical HEP Computing Center normally runs at least one batch system. As an example, at IHEP (Institute of High Energy Physics, Chinese Academy of Sciences), we've used three batch systems: PBS, HTCondor and SLURM. After running PBS as a local batch system for 10 years, we replaced it by HTCondor (for HTC) and SLURM (for HPC). During that period, problems came up on both user and admin sides.

Introduction of the new batch systems implies necessity for users to acquire additional knowledge specific for every batch system, in particular, batch commands. In some cases, users have to use both HTCondor and SLURM in parallel. Furthermore, HTCondor and SLURM provide more functionality, which means more complicated usage mode, compared to the simple PBS commands. On admin side, HTCondor gives more freedom to users, which brings an additional challenge to site administrators. Site administrators have to find the solutions for many problems: preventing users from requesting the resources they are not allowed to use, checking if the required attributes are correct, deciding where requested resources are located (SLURM cluster, the cluster of the virtual machines, the remote sites, etc).

To meet the above requirements, HepJob was designed and developed. HepJob provides a set of simple user commands, for example: hep_sub, hep_q, hep_rm, etc. In the submission process, HepJob checks all the attributes and ensures all attributes are correct; assigns proper resources to users (the user and group info is obtained from the management database); routes jobs to the target site; performs other steps as required.

Users can start with HepJob very easily and administrators can take the necessary management actions in HepJob.

1 Introduction

1.1 Background

By the end of 2019, IHEP computing resources are serving more than 2800 users from above 13 experiments. Historically, in total three batch systems were provided to users, namely PBS[1], HTCondor[2] and SLURM[3]. In 2016, PBS(Torque&Maui) was replaced by HTCondor as the batch system of HTC Clusters. Five HTC clusters have about 18,000 CPU

*e-mail: jiangxw@ihep.ac.cn
**e-mail: duran@ihep.ac.cn
***e-mail: shijy@ihep.ac.cn
****e-mail: zoujh@ihep.ac.cn
†e-mail: huqb@ihep.ac.cn

cores in total and are located at different sites. Moreover, a new HPC cluster managed by SLURM is being expanded.

1.2 Motivation&Purpose

The issues came out when we replaced PBS by HTCondor and started to build SLURM Cluster.

Cluster interfaces have changed completely which implied necessity of learning of the new commands by users. Extended functionality of HTCondor and SLURM introduces additional complexity for users. For example, HTCondor job description file required for submission, represents a set of key-values pairs, which is quite different compared to PBS syntax.

The experiments have developed their own interfaces of connecting to computing clusters. Introducing of the new batch systems required changes in the experiments software. Moreover, many users developed customer scripts for PBS which have to be adapted for the new batch systems. Overall, this implied a lot of changes both for users and experiments.

Several experiments have on-site computing clusters to do preprocessing or basic analysis. For example, LHAASO[4] has a computing cluster in Daocheng City and a cluster in Chengdu City, DYW[5] has a computing cluster in Daya Bay. But our users usually submit jobs in Beijing City where an entry point for the computing resources is located, they have to explicitly specify the remote cluster where a job is submitted to. The separate clusters are not connected together.

A unified and easy-to-use submission frontend tool can address all those issues. The frontend tool provides a set of simple commands, which is quite easy to start. The interface to batch systems becomes transparent, users don't need to learn HTCondor and SLURM, they only interface the new frontend tool. For the separate or individual clusters, the frontend tool helps to select the site which the jobs are going to, based on experiment permission or job type. Additionally, the frontend tool can perform some checks before a job is submitted. This would be helpful to prevent jobs from being interrupted due to issues with file permissions of the job script or user/group permissions of a given resource. Therefore, we designed and developed a frontend tool named 'HepJob'. The details of HepJob will be described in the following sections.

2 Overview of HepJob

HepJob is a unified entry point for computing clusters at IHEP. It's deployed as a client tool on the login nodes. From the implementation point of view, the tool represents rather a 'unified interface' than a server implementation. It reads the user/group info from a central database and translates HepJob commands into corresponding batch system interface. HepJob maintains all the information related to job, user and site in a file. To interface the clusters, users run HepJob commands. HepJob in its turn manages all communication with the batch systems as shown in Figure 1.

In general, as shown in Figure 2, HepJob command goes through four main steps: argument parser, job router, pre-checker and batch interface. Firstly, argument parser receives the complete command executed by user and would parse two types of arguments, simple arguments or a list of complex arguments from the template file (template function will be discussed in Section 3.1). Secondly, job router decides where a job is going, including batch system, site and pool. Thirdly, pre-checker checks the possible incorrect attributes which would often interrupt the process, including user info, group info, resource requirement and so on. Finally, if all steps described above go well, batch interface invokes the API of batch system to make the connection with the computing clusters.

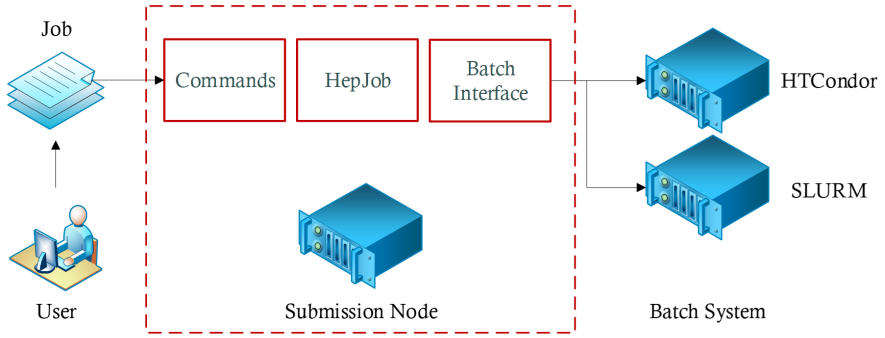

Figure 1: The Workflow of HepJob

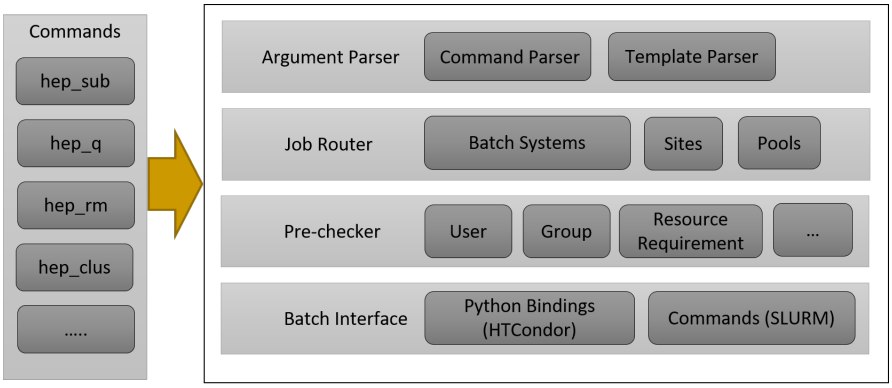

Figure 2: Structure and Functions

So far, HepJob is used to connect with two main batch systems, HTCondor and SLURM. More discussions related to HTCondor are expanded in Section 3, SLURM in Section 4.

3 Processing for HTCondor

The followings are three key ideas used in the process of connecting with HTCondor: template function, routing job and pre-checker.

3.1 Templates

A job would have a long list of attributes, like memory size, number of cpu cores, number of nodes, partition, site, etc. Normally, individual jobs belong to the experiment tasks consisting of multiple jobs with specific requirements. These tasks can run for a long period of time. In this case, a pre-defined argument template would be a possible solution to simplify the submission process. In HepJob, the template is saved as a JSON file containing all necessary attributes. Adding a template argument in the submission command will trigger the template parser. HepJob will merge the arguments parsed from the template file and from user commands. The process is shown in Figure 3.

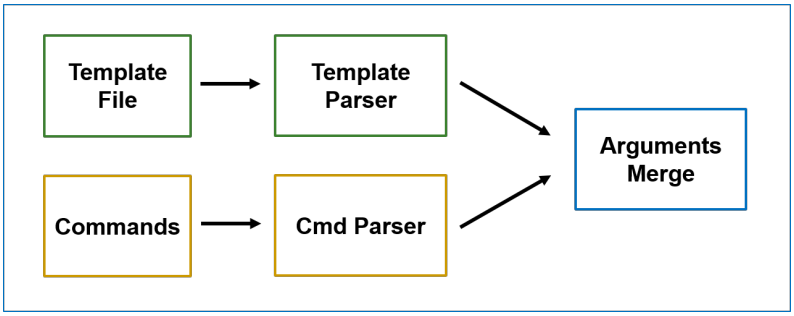

Figure 3: Processing with the Template Function

As an example, in our local cluster, ATLAS users have always requested to use multi-core and big memory resource which is not a common requirement. For this use case corresponding resources have been setup. The 'atlasbm' template used to process ATLAS users jobs is demonstrated in Figure 4. Job attributes and resource configuration can be handled by site administrators using template without any change required on the user side.

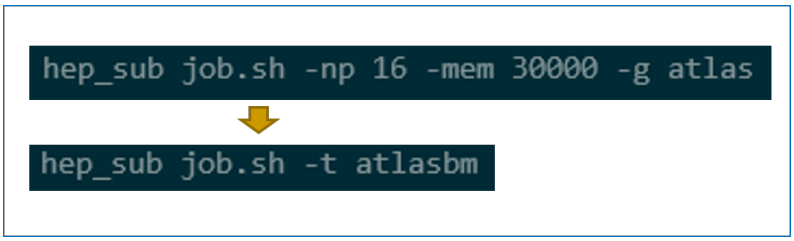

Figure 4: Template Usage Example

3.2 Routing Job

As we discussed in Section 1.1, our resources are distributed across various pools, sites and batch systems. In order to access all those resources, user submission instructions would become too complicated. In reality, many of our users have to face this issue. To address it, all entry points data for job processing are brought together in HepJob and recorded in the routing table in the JSON format. A job looks for its destination based on the information in this routing table.

Figure 5 shows the basic idea of a routing job:
- choose a batch system
- define a target site
- define a proper pool

3.3 Pre-check

Many problems can cause job failure, for example, bugs in the user code, wrong file permissions, incorrect environments and so on. If there is an issue with the job itself, job submitter would not know about the problem for a long while, unless the job starts running and exits

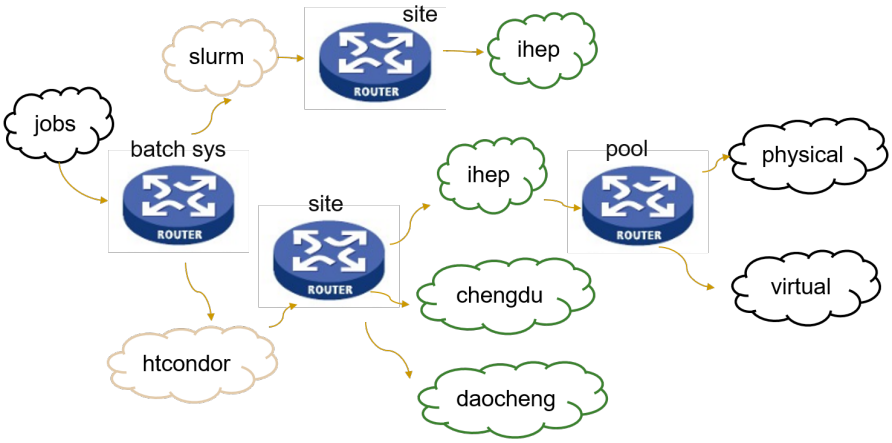

Figure 5: Routing Job

with an error. Another problem can be caused by the wrong requirements. In this case, resource matchmaking can not be performed and the job will stay in the queue for a long time. In both cases too much time is wasted by the user. This is a common issue on the user side. Simple analysis and corresponding checks would allow to avoid some of such problems. We keep collecting problematic cases which can be avoided and checked by HepJob before a job is submitted. When a potential problem is found, user gets a warning. Figure 6 shows the metrics which are checked in Hepjob until now. Besides checks of submission command, the pre-check is also needed to be done in other commands, like querying job, removing job, editing job and so on.

- Requirements Check:
  - accounting_group
  - max_memory
  - max_cpu
  - operating system
  - singularity
  - job universe
  - site
  - …
- Problem Check:
  - output/error file existence
  - job script file existence
  - job script executable permission
  - submission permission
  - job amount limit
  - incorrect environments
  - …

Figure 6: Pre-check Metrics

4 Plugins for SLURM

In contrast with HTCondor, SLURM job submission arguments are defined in the head of job script. HepJob handles only part of HPC submission arguments. It is implemented via HepJob SLURM plugin. Further work on SLURM compatibility will follow the growth of the

HPC cluster. Currently for SLURM, HepJob does not apply template function and complete function of routing jobs. However, an efficient pre-check function has been implemented. HepJob submission to SLURM with a corresponding workflow is demonstrated in Figure 7.

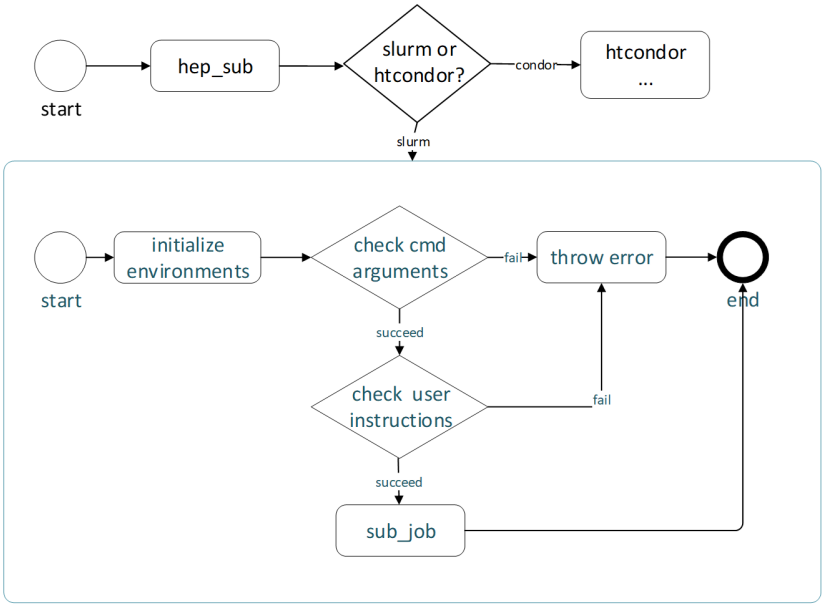

Figure 7: The Submission Workflow to SLURM

HepJob submission command transfers the arguments to SLURM plugin. To ensure the necessary arguments are set correctly, the plugin checks the existence and value type of the received command arguments. SLURM obtains job requirements from the command arguments and the header of the job script. 'User instructions' check consists of verifying whether users do not request illegal resources because of incorrectly defined group or partitions.

5 Performance Tests and Current Status

A simple test to compare time required for submission of 100 jobs to Condor with and without HepJob has been performed in our production environment. Results are shown in Figure 8. In average, condor_submit takes around 40ms for submitting a job from our submission node, while hep_sub takes around 100ms. 100ms for each submission is acceptable performance in the real production environment.

HepJob has been deployed in production since 2016 when its first version was released. Figure 9 shows the job statistics by experiment group from January to October in 2019, at IHEP. Currently, almost all the HTC jobs and part of HPC jobs are being operated via HepJob. HepJob popularity is explained by the fact that HepJob shields users from the job submission complexity. As demonstrated by the latest statistics, users prefer submission via HepJob though it is slower compared to direct one.

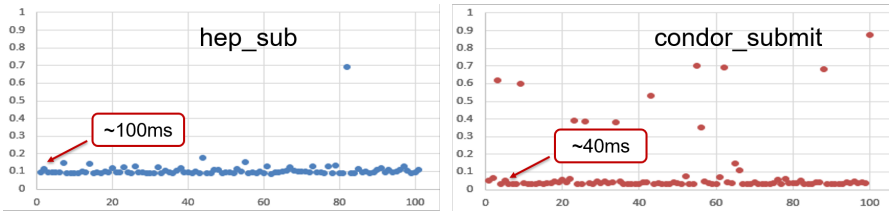

Figure 8: The Results of Time Consume Test

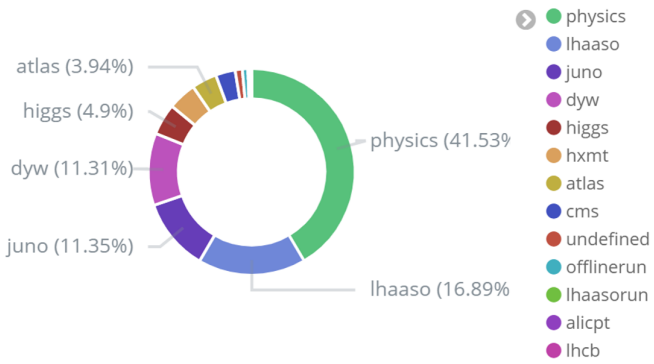

Figure 9: Job Statistics by Group

6 Conclusion

As mentioned in the previous section, HepJob has been successfully used in IHEP for 4 years and allows to effectively operate distributed heterogeneous resources. And the users benefit a lot from HepJob.

In future, we plan to add more helpful functions for users and collect more pre-check metrics. On administrator side, a better and easy-to-deploy solution will be worked out in order to make HepJob maintenance and management easier. Furthermore, we're considering to do some prediction before submitting job in HepJob, which would help the job to find the best resource. Besides, the compatibility with SLURM will be improved.

Authors would like to thank all the colleagues who contributed ideas or developments to this work. Besides, this work was supported by several projects of the National Natural Science Foundation of China (No.11775250, No.11805225, No.11805226, No.11875283 and No.11805223) and the Youth Innovation Promotion Association of Chinese Academy of Sciences.

References

[1] OpenPBS Introductions, *https://http://www.openpbs.org/*, online, accessed 16-Jun-2020

[2] HTCondor Description, *https://research.cs.wisc.edu/htcondor/description.html*, online, accessed 14-Mar-2020

[3] SLURM Overview, *https://slurm.schedmd.com/overview.html*, online, accessed 14-Mar-2020

[4] Richard A Erickson, Michael N Fienen, S Grace Mccalla, Emily L Weiser, Melvin L Bower, Jonathan M Knudson, Greg Thain, *Wrangling distributed computing for high-throughput environmental science: An introduction to HTCondor*. PLoS Computational Biology, 01 October 2018, Vol.14(10), p.e1006468

[5] Cao,Zhen For Lhaaso Collaboration (Corporate Author), *Status of LHAASO updates from ARGO-YBJ*, Nuclear Inst. and Methods in Physics Research, A, 01 April 2014, Vol.742, pp.95-98

[6] Jun Cao, *Daya Bay neutrino experiment*, Proceeding of NuFACT05.Nucl.Phys.Proc. Suppl.155, 229-230, 2006