PROCEEDINGS OF SCIENCE

Supporting Multiple Concurrent Usage Patterns on the HTCondor-based HTC Computing System at IHEP

Xiaowei JIANG,[a,b,*] **Jingyan SHI,**[a] **Ran DU,**[a] **Qingbao HU,**[a] **Gongxing SUN,**[a] **Libin XIA**[a,b] **and Xiaoyu LIU**[a,b]

[a] *Institute of High Energy Physics,*
19B Yuquan Road, Beijing, China
[b] *University of Chinese Academy of Sciences,*
19A Yuquan Road, Beijing, China

E-mail: jiangxw@ihep.ac.cn, shijy@ihep.ac.cn

IHEP is a multi-disciplinary research institution which is hosting or attending about 15 experiments in the field of high energy physics, including LHAASO, BES, JUNO, HEPS, DYW, ALI, GECAM, ATLAS, CMS, LHCb, etc[1]. Corresponding to these diverse experiments, also in the computing system, a range of usage scenarios has to be considered and the technologies employed should be suitable for the different requirements from these experiments and applications. The technological architectures and the corresponding scenarios are: 1. a local sharing pool for massive offline data processing; 2. a unified sharing pool with a special resource policy for WLCG (Worldwide LHC Computing Grid) and JUNO distributed computing grid; 3. a dHTC (distributed High Throughput Computing) pool for resources sharing between HTC (High Throughput Computing) and HPC (High Performance Computing) and between sites; 4. a near-real-time computing pool for astrophysics streaming data processing; 5. customized clusters for the cooperating institutions or universities. HTCondor[2] is a popular distributed batch system in HEP computing. This paper will discuss an HTC computing system supporting different usage scenarios, based on HTCondor at IHEP, including local batch computing, distributed computing and near-real-time computing. Currently, the whole computing system is managing about 50,000 CPU cores (including x86-arch CPU and arm-arch CPU) and some GPU cards.

International Symposium on Grids & Clouds 2022 (ISGC 2022)
21 - 25 March, 2022
*Online, Academia Sinica Computing Centre (ASGC), Taipei, Taiwan****

*Speaker

PoS(ISGC2022)010

1. Background

IHEP is a multi-disciplinary research institution which is hosting or attending about 15 experiments around high energy physics, including LHAASO, BES, JUNO, HEPS, DYW, ALI, GECAM, ATLAS, CMS, LHCb, etc. As a typical HEP research institutes, a computing platform is necessary for the data processing of these experiments. However, these experiments perform different computing tasks on IHEP's computing platform according to their requirements. As an example, the LHC (Large Hadron Collider) experiments and JUNO only schedule the small number of production jobs and analysis jobs to IHEP site without a strict time demand. But GECAM wants the jobs to be scheduled immediately and needs a near-real-time computing system. Therefore, a range of usage scenarios has to be considered and the technologies employed should be suitable for the different requirements from these experiments and applications.

2. Overview of Situations and Solutions

HTCondor is a popular distributed batch system in HEP computing. It's customarily regarded as a batch system focusing on high throughput computing. Originally, IHEP built a HTCondor cluster to replace PBS as the main computing system for HEP offline data processing[3]. With the increasing amount of computing tasks and the growing diversity of requirements from experiments, more solutions have been made. Given the available local expertise in using HTCondor, most of the computing solutions are designed and deployed based on HTCondor.

The technological architectures are designed and implemented for the corresponding scenarios. They are a local sharing pool for massive offline data processing, a unified sharing pool with a special resource policy for WLCG (Worldwide LHC Computing Grid) and JUNO distributed computing grid, a dHTC (distributed High Throughput Computing) pool for resources sharing between HTC (High Throughput Computing) and HPC (High Performance Computing) and between sites, a near-real-time computing pool for astrophysics streaming data processing and several customized clusters for the cooperating institutions or universities. Figure 1 shows the representation of the science use cases and the facilities, interlinked by the HTCondor system.

3. Local HTC Cluster

In the field of high energy physics, many sites are using HTCondor to build the high throughput computing clusters. This is similarly the case at IHEP, where an HTCondor system has been deployed since 2016. With HTCondor, the IHEP computing center team has done work, including work on resource sharing policy[3], HepJob[4], OMAT[5], etc. During the latest two years, the scalability of the system and the implementation of HA (high availability) mechanisms are becoming the main work items. Figure 2 shows the basic structure of the local HTC cluster. To scale up the capacity of the job queue, a cluster of schedds (each schedd daemon maintains a job queue) is designed and deployed. According to the historic amount of jobs completed for each experiment, a job will be routed to a specific schedd associated with its experiment. The routing rule is defined in HepJob, which is a front-end job tool. The schedd cluster also has a role on providing high availability. If one of the schedds suffers a problem, the job will be easily mapped to another schedd with a simple

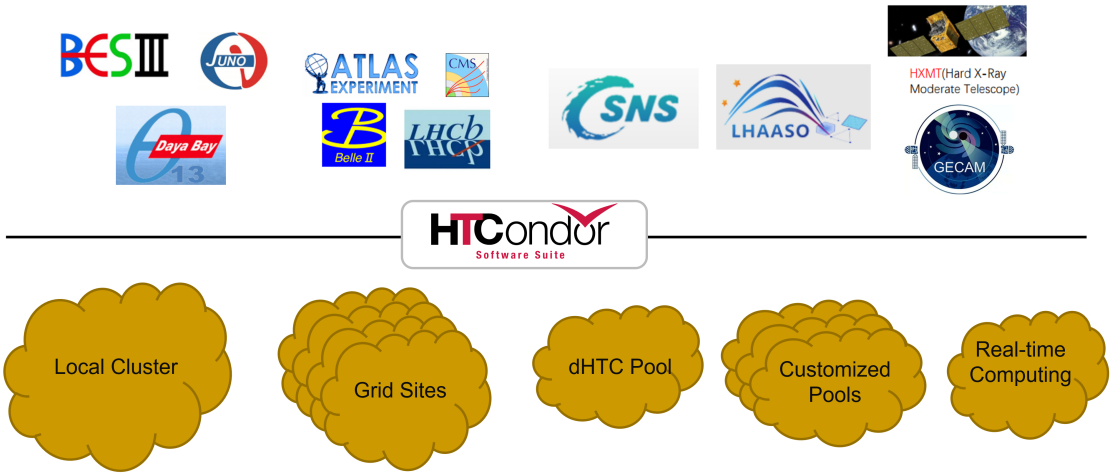

Figure 1: The Relationship between Science Use Cases and Facilities

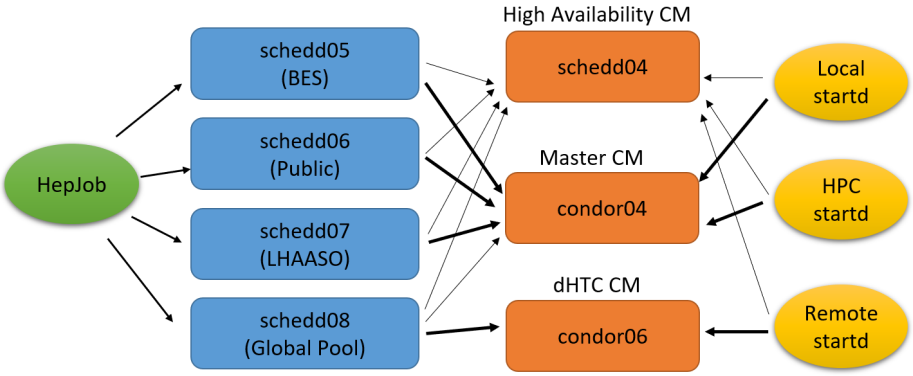

Figure 2: Structure of the Main Local Cluster at IHEP (*HepJob is a submission frontend tool; CM indicates the central manager of HTCondor; startd indicates the worker node of HTCondor*)

modification in HepJob. On the CM side, we also deployed a backup CM server to perform a HA role.

The local HTC cluster can meet the current requirements form HEP experiments at IHEP. Figure 3 shows the status of completed jobs showing an average of 524k jobs completing per day and Figure 4 shows that a resource utilization above 90% can be reached over a one-month average, based on data form 2022.

4. Computing Entry and Batch System for Grid Sites

Before 2020, CreamCE[6] and Torque[7] were used as the entry point for computing on the batch system at the IHEP-Beijing site. With the phasing out of CreamCE, CreamCE was replaced by HTCondorCE and PBS was replaced by HTCondor in the middle of 2020. Currently, the IHEP-Beijing grid site is serving for ATLAS, CMS, LHCb, BelleII, JUNO, CEPC and the connection to ILC is being made[8]. The number of CPU cores for each user community is shown in Figure 5.

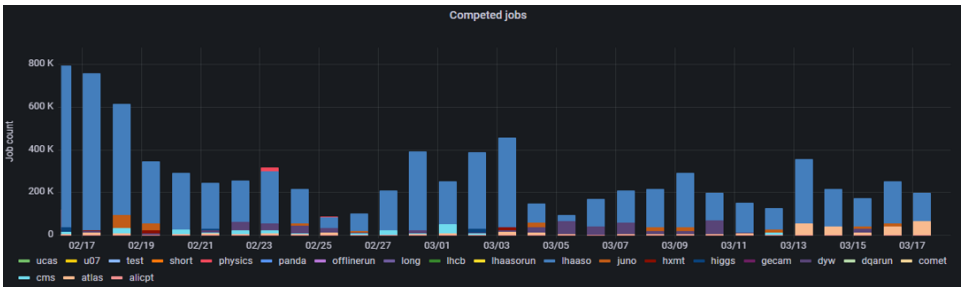

Figure 3: Completed Jobs Per Day in IHEP HTC Cluster

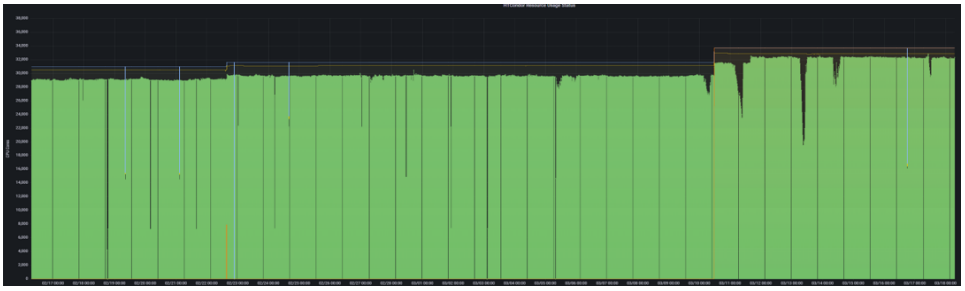

Figure 4: Resource Utilization of IHEP HTC Cluster

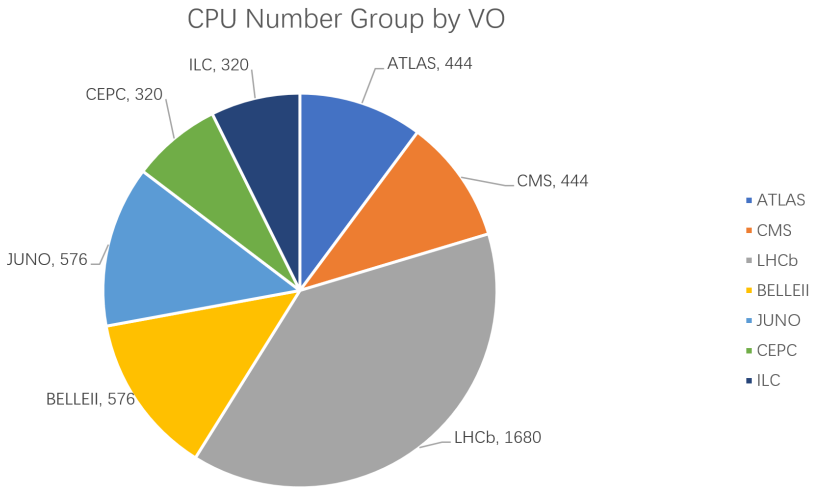

Figure 5: Number of CPU Cores Assigned per Community at IHEP-Beijing Site

According to the scale of the whole pool, a single CE (Computing Element) and batch system should be enough to serve all these communities. The main issue is to guarantee that the jobs for a specific user community should be directed to and executed on those nodes have been pre-assigned for use by that specific community. By using the Job Router and Accounting Group add-on to HTCondor, the job scheduling issue can be solved. The Job Router is a component add-on to the HTCondor Schedd server to transform jobs from one type into another according to a configurable policy[9], and the Accounting Group is a function to decide the priority of a group and fair share between groups[10]. The simplified architecture of HTCondorCE and HTCondor is shown in

Figure 6.

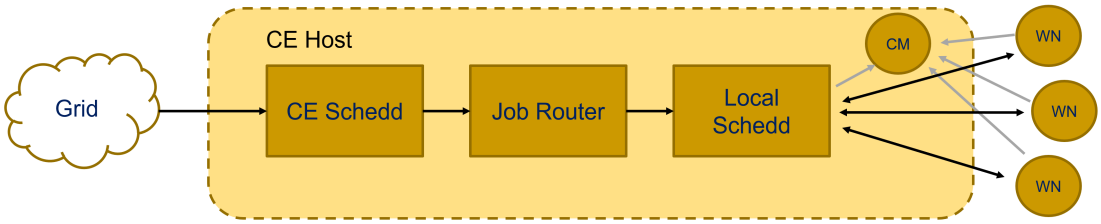

Figure 6: Computing Element for Grid Sites at IHEP

A grid job is submitted to the CE host at a site. Firstly, the job goes into CE schedd and several attributes related to user community are added to the job classads. Secondly, based on the job router configurations that indicate the mapping relationship between the community-related classad and accounting group, the job will be sent to the local schedd and the job classad attributes related to accounting group will be added. Thirdly, the job will be scheduled to the worker node where the constraint of the corresponding accounting group is set. By means of this workflow, the jobs submitted by different communities can find the designated resources from the sharing pool.

5. Distributed HTC Pool

With the development of high energy physics, the scale of related to IT infrastructure is also increasing. In the case of IHEP, the current number of CPUs has increased 4 times compared to seven years ago and several new sites or data centers have been built by IHEP, including the Daocheng site and the Dongguan data center. In the meantime, several domestic computing sites have been built by other cooperating institutes, such as the University of Lanzhou, the University of Shandong and other institutes. The incorporation of multiple sites or data centers brings a new requirement to integrate all the sites. Figure 7 shows the corresponding logical structure of the computing job distribution among the participating sites. The resources are located in the Beijing data center, the Dongguan data center and several edge sites. Some commercial off-the-shelf cloud resources were also tested. On the user side, the HepJob tool is still used to interact with the job queue, so that the IHEP users don't need to change the way of interacting with the jobs.

Following the development of distributed high throughput computing, the HTCondor Glidein[9] was decided on as the fundamental technology to build the dHTC pool. Figure 8 shows the basic design of the dHTC pool at IHEP. The Glidein Factory is a factory to produce and submit glidein jobs. The Sentry keeps watching on the user job queue and decide if a specific number of glidein jobs needs to be produced with the pre-defined policy. If a glidein job is decided to produce, with the site information in SI Repo (site information repository), several attributes releated to site and resource requirement will be added on to the glidein job and the glidein job will be submitted to the Factory Scheduler, which maintains a glidein job queue and a job router tool. The Factory Scheduler re-directs the glidein jobs to the targeted data centers or sites based on the added-on attributes. When a glidein job starts running on a remote work node, the glidein tool will be lauched

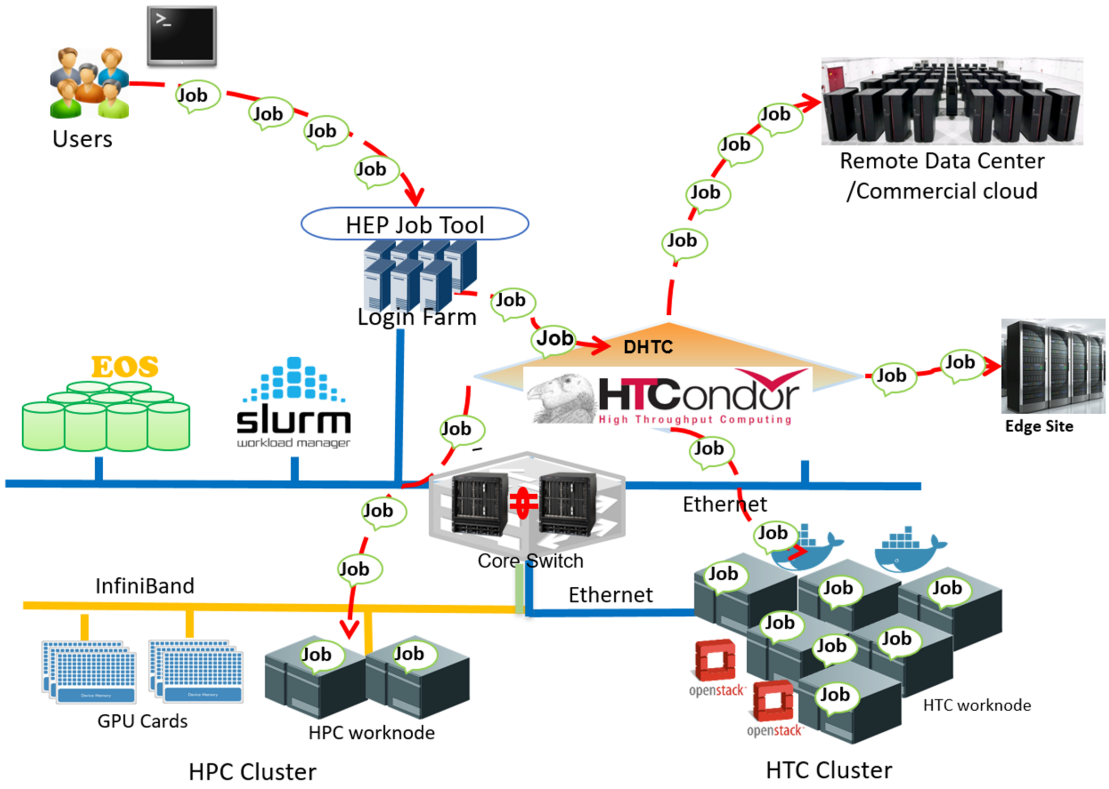

Figure 7: The Basic Situation of dHTC at IHEP

and the work node will be reported back to the global pool. Till now, we are still using a simple policy where more incoming user jobs trigger more glidein jobs to be submitted.

Currently, some specific jobs from the LHAASO experiment have been running in the dHTC pool and BES simulation jobs are under testing and migration.

6. Customized Clusters in Edge Sites

As mentioned in the previous section, several domestic sites were built with the cooperation of IHEP CC team. For the computing service at these sites, HTCondor was selected as the batch system and some different policies were made for each site. On the resource side, the static and dynamic slots are set and CPU and GPU are both added in, which all need to set different configurations or install extra tools. On job side, both single-core job as well as parallel job are supported, but not the multi-node job. Most of the clusters are typical HTCondor cluster and therefore do not warrant further discussion here. The main issue is to centrally manage the multiple sites in parallel. To achieve this, we have selected a solution is based on puppet[12].

The basic structure is shown in Figure 9. Puppet has a central server to manage and distribute the configurations. With a gitlab[11] repository, the configuration of a site is stored in a puppet module. If a new update is submitted, the puppet server will send the updated configuration to the targeted site and make the configuration effect. Similarly, for the HTCondor administrative configuration, any change will be committed to the puppet repository centrally and then the change

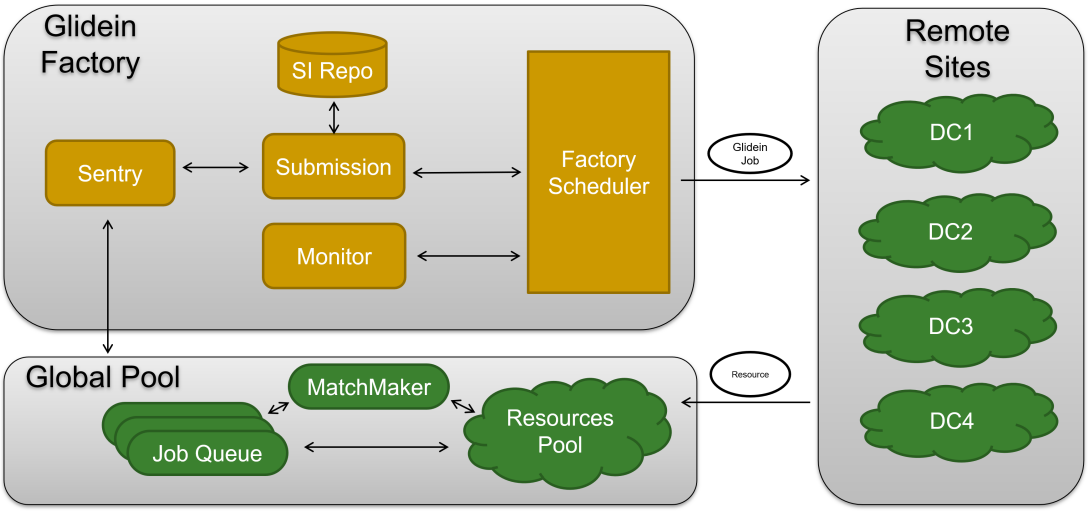

Figure 8: Structure of dHTC Pool

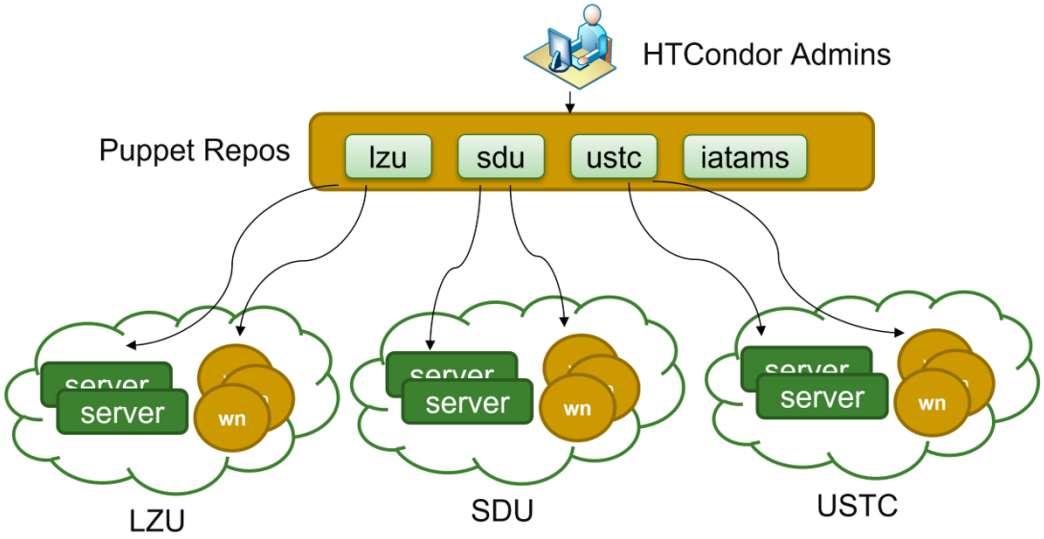

Figure 9: The Basic Workflow of the Puppet Solution (*central management of the edge sites and their worker nodes at three representative sites, using a common Puppet Repository managed by the central HTCondor Administrators*)

will be published to the schedd servers, central manager servers, or worker nodes on site. In this way, the HTCondor system administrators can manage several sites in parallel.

The edge sites are also integrated in the central monitoring system. Figure 10 shows a snapshot from the monitoring system showing the current status of edge sites.

7. The 'Quick' Computing Cluster

Satellite-based experiments have a requirement for real-time computing. With a typical HTC cluster, it's hard to handle a fully real-time computing situation. So we called this kind of HTC clus-

Monitoring of One Platform for Multiple Data Centers
last 1 week

Sites	CPU Resources (CPU Cores)	CPU Resource Utilization	Disk Storage Capacity	Data Storage	Completed Jobs HTC&HPC	Job Run Time (CPU Hour)
IHEPCC	39,996	88.97%	72.12 PB	43.22 PB	2,752,083	5,953,860
DongGuan	31,120	39.51%	6.38 PB	2.30 PB	493	1,399,793
DaoCheng	3,616	47.93%	4.27 PB	3.54 PB	2,036	344,790
CSNS	5,852	31.24%	802.7 TB	375.5 TB	469	290,286
SDU	1,180	8.863%	352.9 TB	238.8 TB	716	16,538
USTC	3,018	26.97%	1.17 PB	767.2 TB	3,539	145,605
LZU	1,768	2.091%	341.8 TB	217.3 TB	585	6,215

Figure 10: the Current Status of Edge Sites

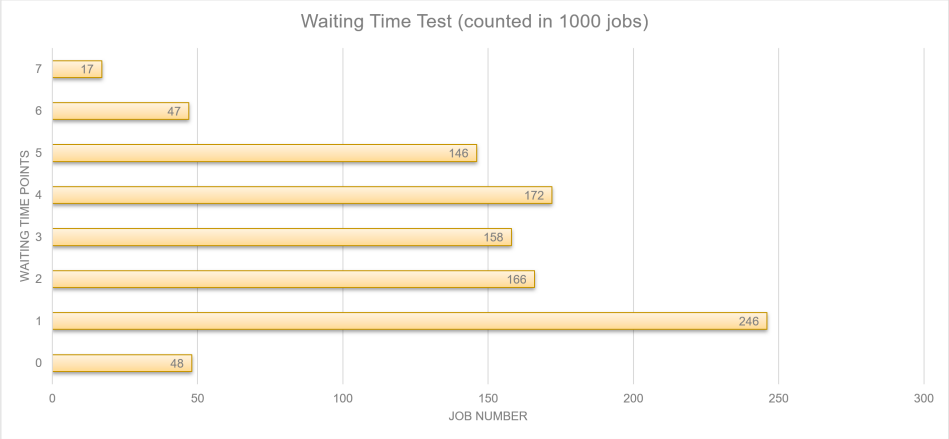

Figure 11: Waiting Time Test in Quick Computing Cluster

ter "quick computing cluster". Two concurrent approaches have been selected for perform real-time computation. One is keeping a cluster dedicated for satellite experiments without any disturbance. The other one is optimizing the configurations of the HTCondor negotiator to speed up the negotiation. The configurations of NEGOTIATOR_INTERVAL and NEGOTIATOR_CYCLE_DELAY are set to a lower value than the default one. With these approaches, most job can be quickly scheduled in short time. A simple test is done and the result is shown in Figure 11. In this test, 1000 jobs are submitted to an empty job queue, and then count the waiting time of these jobs to estimate the negotiation speed. The result indicates the average waiting time of all the jobs are about 2.9 seconds which is an acceptable data for a near-real-time computing task.

As the example of GECAM experiment[12], several tasks working on Storm Event Searching should be started in real time when the job landing in the quick computing cluster. These tasks include single-core and multi-core jobs. Currently, the quick computing cluster is performing as intended for GECAM and Figure 12 shows the statistics of completed job in a month of 2022.

Figure 12: Job Statistics in Quick Computing Cluster (*job statistics from the GECAM experiment, as run on the Quick Computing Cluster. The period shown on the x-axis is from February 21st till March 21st 2022. The vertical axis shows the completed job count, in bins of one hour.*)

8. Summary

This paper introduces five computing solutions based on HTCondor corresponding to five scenarios at IHEP. For each scenario, we present or introduce a job distribution and scheduling mechanism, and then develop tools or systems to implement or deploy the method in production. These solutions are the local HTC cluster, the Grid CE, the distributed HTC pool, customized edge sites and the near-real-time quick computing cluster. The total number of CPU cores contained within these five solutions is above 50,000.

For the future work, the integration of these pools or sites will be the focus. Till now, we have managed to use HTCondor Glidein to integrate several sites and the HPC Pool. So the solution based on the HTCondor Glidein can be a potential way to integrating the clusters or sites mentioned in this paper.

Acknowledgements

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.12175255, No.11805226, No.12105303, No.11905239) and the Youth Innovation Promotion Association of Chinese Academy of Sciences (No.2020017).

References

[1] IHEP Introduction, *http://english.ihep.cas.cn/se/fs/*, online, accessed 25-Apr-2022.

[2] E M Fajardo, J M Dost1, B Holzman, T Tannenbaum, J Letts, A Tiradani, B Bockelman, J Frey and D Mason, *How much higher can HTCondor fly?*, J. Phys.: Conf. Ser. 664 062014.

[3] Xiaowei JIANG, Jingyan Shi, Jiaheng Zou, Qingbao Hu, Ran Du, and Zhenyu Sun, *Research and Exploit of Resource Sharing Strategy at IHEP*, EPJ Web of Conferences 214, 03014 (2019)

[4] Xiaowei JIANG, Ran Du, Jingyan Shi, Jiaheng Zou and Qingbao Hu, *A Lightweight Submission Frontend Toolkit HepJob*, EPJ Web of Conferences 245, 03026 (2020)

[5] Hu,Qingbao, Zheng,Wei, Jiang,Xiaowei and Shi,Jingyan, *Application of OMAT in HTCondor resource management*, Proceedings of Science, v 378, October 22, 2021

[6] C. Aiftimiei et al, *Design and Implementation of the gLite CREAM Job Management Service*, Future Generation Computer Systems, Volume 26, Issue 4, April 2010, pp. 654-667 Massimo Sgaravatto, Sergio Traldi, Luigi Zangrando

[7] Torque Documentations, *https://support.adaptivecomputing.com/wp-content/uploads/2021/02/torque/torque.htm#topics/torque/0-intro/torquewelcome.htm*, online, accessed 29-August-2022

[8] Yaosong Cheng, *IHEP Site Report*, HEPIX SPRING 2022 online workshop

[9] Introduction of the job router, *https://htcondor.readthedocs.io/en/latest/grid-computing/job-router.html?highlight=job%20router#the-htcondor-job-router*, online, accessed 25-Apr-2022

[10] Introduction of accounting group, *https://htcondor.readthedocs.io/en/latest/admin-manual/user-priorities-negotiation.html?highlight=accounting%20group#group-accounting*, online, accessed 25-Apr-2022

[11] D Schultz, B Riedel and G Merino, *Pyglidein – A Simple HTCondor Glidein Service*, 2017 J. Phys.: Conf. Ser. 898 092018

[12] Puppet Introduction, *https://forge.puppet.com/*, online, accessed 25-Apr-2022

[13] Gitlab Introduction, *https://about.gitlab.com/*, online, accessed 25-Apr-2022

[14] Xu, Y.B., Li, X.Q., Sun, X.L. et al, *The design and performance of charged particle detector onboard the GECAM mission*, Radiat Detect Technol Methods 6, 53–62 (2022)