EPJ Web of Conferences **214**, 03014 (2019) https://doi.org/10.1051/epjconf/201921403014
CHEP 2018

Research and Exploit of Resource Sharing Strategy at IHEP

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

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

Abstract. At IHEP (Institute of High Energy Physics, Chinese Academy of Sciences), computing resources are contributed by different experiments including BES, JUNO, DYW, HXMT, etc. The resources were divided into different partitions to satisfy the dedicated experiment data processing requirements. IHEP had a local Torqu&Maui cluster with 50 queues serving for above 10 experiments. The separated resource partitions leaded to imbalance resource load. In a typical situation, BES resource partition was quite busy without free slot but still with lots of jobs in idle, while JUNO resources are free and wasted seriously.

After moving resources from Torque&Maui to HTCondor in 2016, job scheduling efficiency has been improved a lot. In order to balance the imbalance resource load, we designed an efficient sharing strategy to improve the overall resource utilization. We created an unified pool shared by all experiments. For each experiment, resources are divided into two parts: dedicated resource and sharing resource. The slots in dedicated resource only run jobs from its own experiment, and the slots in sharing resource are shared by jobs from all experiments. Default ratio of dedicated resource to sharing resource is 1:4. To maximize the sharing effectiveness, the ratio is dynamically adjusted between 0:5 and 4:1 based on the number of jobs submitted by each experiment.

We have developed a central control system to decide how many resources can be allocated to each experiment group. This system is implemented at two sides: server side and client side. A management database is built at server side, which is storing resource, group and experiment information. Once the sharing ratio needs to be adjusted, resource group will be changed and updated into database. The resource group information is published to the server buffer in real-time. The client periodically pulls resource group information from server buffer via https protocol. And resource scheduling configuration at client side is changed based on the resource group information. With this method, share ratio can be modified and deployed dynamically.

Sharing strategy is implemented with HTCondor. ClassAd mechanism and accounting-group in HTCondor facilitate to utilize the sharing strategy at IHEP computing cluster. With the sharing strategy, resource usage has been improved dramatically.

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

1 Introduction

1.1 Background

At IHEP (Institute of High Energy Physics, Chinese Academy of Sciences), computing resources are mainly used for physics data processing, they are purchased and contributed by different experiments including BES[1], JUNO[2], DYW[3], LHAASO[4], etc. All the resources are grouped into several dedicated partitions, each partition is only dedicated to its contributor experiment or specific application without sharing. With Torque[5]&Maui[6], IHEP computing center had built a local cluster with 50 queues, served for above 10 experiments for over 10 years[7], but it encountered a bottleneck on scale which is unable to perform well under the high throughput situation.

1.2 Problems

The separated resource partitions leaded to imbalance resource load. As shown in Figure 1, BES resource partition is quite busy at some time points, meanwhile, there are still lots of jobs in BES queue; Oppositely, at the corresponding time, most of DYW resources are free without any job running. In a reversed case, DYW partition is busy but BES partition is free. Apparently, separately using resources will waste of numerous computing resources. Besides, the separated resource partition will not satisfy the needs in some specific situations.

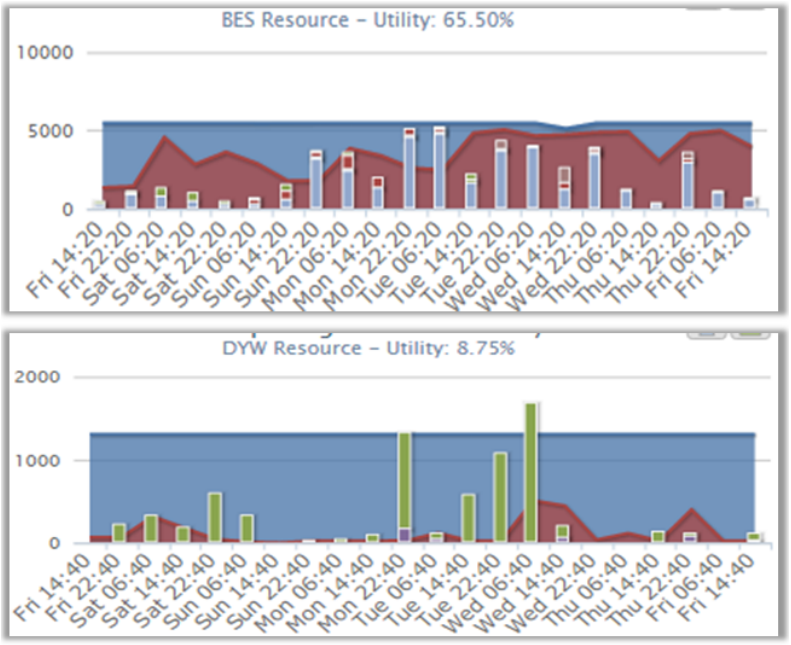

Figure 1: utilization status of resource partition for BES and DYW experiments (*bars represent the number of queuing jobs, red area represents the number of resources being used, blue area represents the number of free resources*)

For instance, in case an experiment meets an urgent task for processing a large scale of data, but the number of resources they need is much more than they have, it will delay the progress. In this case, if more resources from other experiments could be shared to this experiment, which would speed up the urgent task.

2 Basic Method

After moving resources from Torque&Maui to HTCondor[8] in 2016, job scheduling efficiency and resource usage have been improved dramatically. However, resource usage can not be improved again when it reached around 80%, that is limited by the separated resource partition. In order to break resource isolation, an efficient sharing strategy was presented to improve the overall resource usage. The strategy is implemented with two core components: Sharing Policy and Central Controller. **Sharing Policy** dynamically defines the sharing quota for each experiment group. **Central Controller** manages the sharing information which is published to worker nodes automatically.

3 Sharing Policy

In the sharing policy, all resources are collected into a unified resource pool which is shared by all experiment groups. Resources of each experiment are divided into two parts: dedicated resource and sharing resource. The slots in dedicated resource only run jobs from its own experiment group, and the slots in sharing resource are shared by jobs from all experiment groups. N_{all} (number of total resources) and Ng_i (number of resources for group i) are constrained by the conditions below:

$$N_{all} = \sum_{i=group_0}^{group_n} Ng_i \ , (i = physics, juno, dyw, hxmt, ...) \tag{1}$$

$$Ng = Ng_{dedicated} + Ng_{sharing} \tag{2}$$

$Rate_{sharing}$ is defined to evaluate the number of sharing resources for each group(each group has its own sharing rate), So $Ng_{sharing}$(number of sharing resources) and $Ng_{dedicated}$(number of dedicated resources) are evaluated based on the simple expressions below.

$$Ng_{sharing} = Rate_{sharing} * Ng \tag{3}$$

$$Ng_{dedicated} = (1 - Rate_{sharing}) * Ng \tag{4}$$

Default sharing rate is 0.2. To maximize sharing effectiveness, the ratio is dynamically adjusted between 0 and 1 based on the number of jobs submitted by each experiment group. Figure 2 is showing an instance about sharing and dedicated resources, an experiment owns its dedicated resources, shares part of its resources to other experiments and benefits extra resources from other experiments.

4 Central Controller

Central controller system was developed to allocate resources for each experiment group, the structure is shown in figure 3. Central controller system is implemented at two sides: server and client. A management database is built at server side, which is storing resource, group and experiment information. Once the sharing ratio needs to be adjusted, resource group information in database will be updated. The resource group information is published to the server buffer in real time.

At client side, two ClassAd attributes (IHEP_SHARED_GROUPS and IHEP_OWNING_GROUPS) are defined in HTCondor's startd configuration. A resource group would be added in or deleted from IHEP_SHARED_GROUPS if a worker

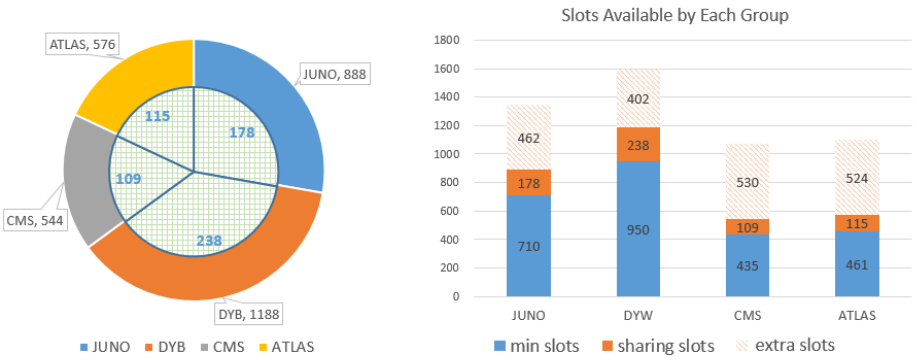

Figure 2: a case of sharing and dedicated resources (the value unit is cpu core)

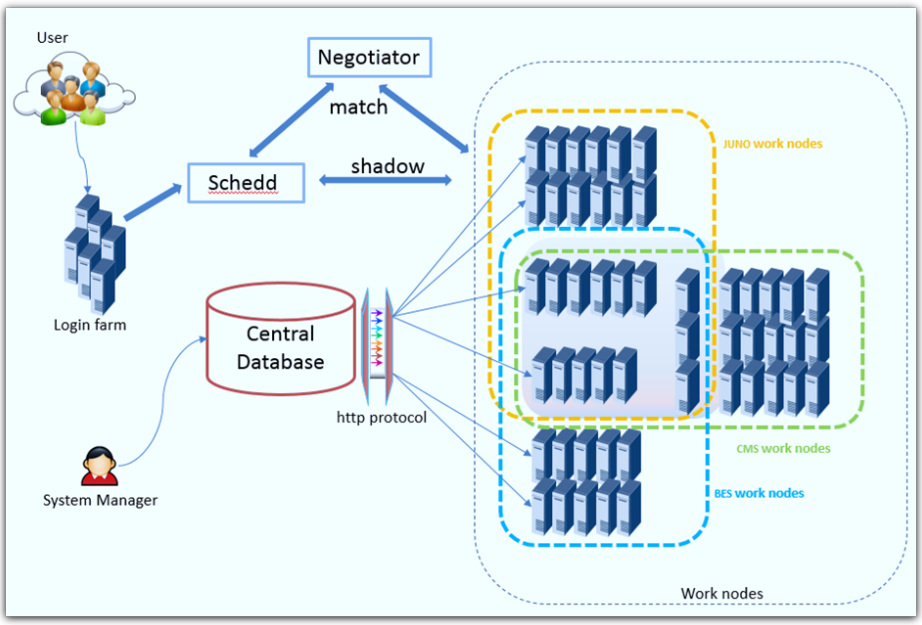

Figure 3: central controller structure

node needed to be shared or unshared, IHEP_OWNING_GROUPS is initially assigned with the contributor group which is used for priority in sharing policy. The Client periodically pulls resource group information from server buffer via https protocol and updates IHEP_SHARED_GROUPS attribute. In this process, share ratio can be regulated and deployed in computing cluster dynamically.

5 Conclusion

With the sharing strategy in this paper, the overall resource utilization of IHEP computing cluster has dramatically increased from around 50% to around 90%, as shown in figure 4. Just as the comparison shown in figure 5, the overall resource utilization without sharing policy during 2015 Oct to 2016 Oct is much lower. Besides, the total wall-time without sharing strategy in 2016 is 40,645,124 CPU hours, while with sharing strategy, the number in 2017 is 73,341,585 CPU hours, increasing by 80.44%. These results indicate sharing strategy is efficient. And the more CPU hours means that more tasks of data processing obtain their resources, which would promote the progress of experiment.

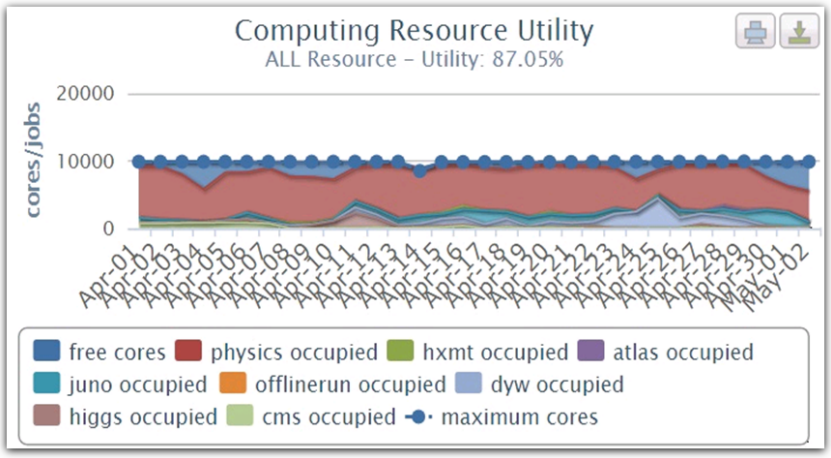

Figure 4: overall resource utilization at IHEP cluster during a month of 2018

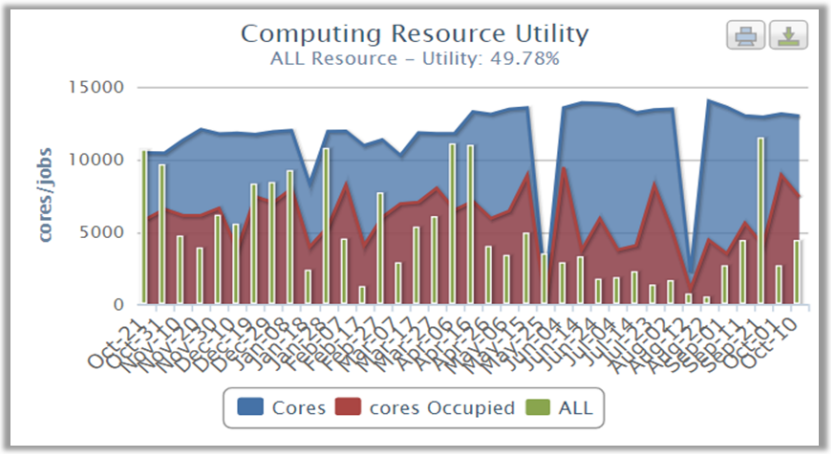

Figure 5: overall resource utilization resource at IHEP cluster during 2015 Oct to 2016 Oct

This work was supported by the National Natural Science Foundation of China (No.11775250, No.11475210, No.11605221, No.11805225, No.11805226, No.11805223, No.11875283 and No.11605223).

References

[1] Zhen-An, *Status of BEPCII/BESIII project*, In Accelerator and particle physics, Proceedings, 9th Winter Institute, APPI 2004, Japan, February 16-20 (2004)

[2] C. Jollet, *The JUNO experiment*, Nuovo Cim, C39(4):318 (2017)

[3] F. P. An et al, *The muon system of the Daya Bay Reactor antineutrino experiment*, Nucl.Instrum.Meth, A773:8-20 (2015)

[4] G. Di Sciascio, *The LHAASO experiment: from Gamma-Ray Astronomy to Cosmic Rays*, Nucl.Part.Phys.Proc, 279-281:166-173 (2016)

[5] http://docs.adaptivecomputing.com/torque/6-0-0/help.htm

[6] http://docs.adaptivecomputing.com/maui/index.php

[7] Bowen Kan and Jingyan Shi and Xiaofeng Lei, *A new Self-Adaptive disPatching System for local clusters*, Journal of Physics: Conference Series, 664 (2015)

[8] https://research.cs.wisc.edu/htcondor/description.html