PROCEEDINGS OF SCIENCE

Application of OMAT in HTCondor resource management

Qingbao Hu*

Institute of High Energy Physics,

Chinese Academy of Sciences, Beijing, China

E-mail: huqb@ihep.ac.cn

Wei Zheng

Institute of High Energy Physics,

Chinese Academy of Sciences, Beijing, China

E-mail: zhengw@ihep.ac.cn

Xiaowei Jiang

Institute of High Energy Physics,

Chinese Academy of Sciences, Beijing, China

E-mail: jiangxw@ihep.ac.cn

Jingyan Shi

Institute of High Energy Physics,

Chinese Academy of Sciences, Beijing, China

E-mail: shijy@ihep.ac.cn

PoS(ISGC2021)021

[*] Speaker

Abstract

Conventional computing resource management systems use a system model to describe resources and a scheduler to control their allocation, computing resources are divided into isolated parts to provide computing services for different experiments. To improve resource utilization and reduce the deployment complexity of differentiated operating environments, our computing resources are configured to support a running environment of all HTC (high throughput computing) experiment jobs. To prevent experiments with few computing resources from occupying a large amount of extra computing resources for a long time, we configured the running jobs' quota of each experiment to ensure the fairness. The conventional computing resource management systems does not adapt well to the ever-expanding resources scale and complex scheduling strategies.

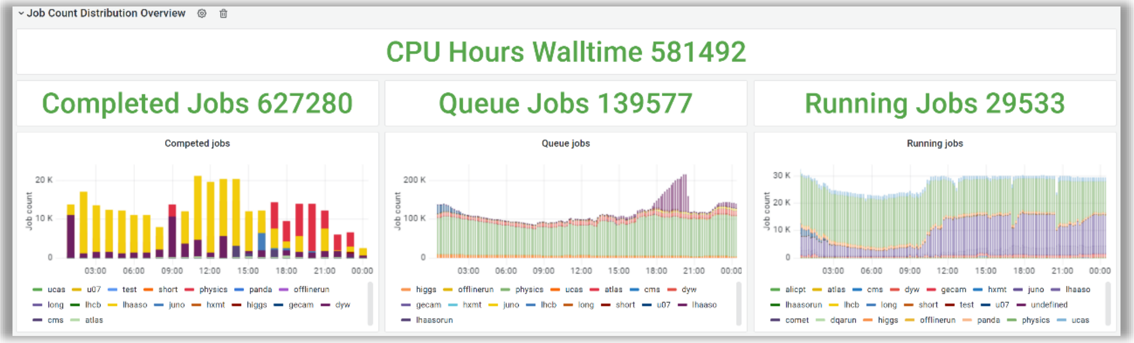
Faced with these problems, we developed and implemented a new framework based on device management database and Open Maintain Analysis Tools (OMAT)[1], a flexible and general approach to manage resources in a complex environment with that significantly reduces manual intervention. Novel aspects of the framework include a flexible configuration method for configuring the relationship between device, service, and experiment; alarm policies that quickly detect unallocated computing resources, and an operationally implementable way to quickly generate a scheduling policy and make it effective. This framework is robust, flexible, and scalable that can evolve with changes in resources and experiments.

The framework was designed to solve real problems encountered in the deployment of HTCondor, a high throughput computing scheduler system at IHEP of Chinese Academy of Sciences.

International Symposium on Grids & Clouds 2021, ISGC2021
22-26 March 2021
Academia Sinica, Taipei, Taiwan (online)

1. Introduction

At IHEP (Institute of High Energy physics), local computing clusters provide high throughput computing and high performance computing services. High throughput computing cluster consist of more than 23,000 cpu cores and providing computing resources for many of the experiments in which IHEP participates, including BESIII[2], DYW[3], CMS[4], LHAASO[5], JUNO[6] etc. In order to improve resource utilization, each work node support all job runtime environments of those experiments mentioned earlier, and configure the experiment quota at the scheduler to keep the fairness among all the jobs. Under this scheduling strategy of resource sharing[7], there are more than 100 million jobs are completed in the last year. As shown in figure 1, half a million jobs complete every day, more than 20,000 jobs are running at any given time, and tens of thousands of jobs submitted by users from various experiments are idle at any one time, competing for resources in the job queue.



CPU Hours Walltime 581492

Completed Jobs 627280

Queue Jobs 139577

Running Jobs 29533

Figure 1: HTC job status at IHEP in 24 hours *(the figures display the number and distribution of completed, queuing and running jobs group by experiment)*

As the batch system for managing compute-intensive jobs of IHEP, HTCondor[8] is used to manage the high throughput computing cluster. During HTC job distribution, each updated resource allocation policy takes effect quickly by reconfiguration of the "StartD" service on the corresponding node, so as to achieve the purpose of managing resources according to specific needs. Based on this method, this article combines service monitoring, asset management and data analysis technologies to design and implement a computing resource management system suitable for HTCondor, which is used to ensure the stable operation of operations and improve the success rate of operations.

This paper mainly studies how to collect and maintain the state of computing resources, how to adjust the resource's job receiving strategy, and how to quickly make the strategy effective.

2. Related Work

A principal consideration of resource management systems is the efficient assignment of resources to customers. The problem of making such efficient assignments is referred to as the resource allocation or scheduling problem, and it is commonly formulated in the context of a scheduling model that includes a system model, which is an abstraction of the underlying resources. The system model provides information to the allocator regarding the availability and properties of resources at any point in time. The allocator uses this information to allocate resources to tasks so as to optimize a stated performance metric.[9] Combined with the multi-application shared scheduling scenario in IHEP, how to allocate resources to customers effectively needs to be realized in two steps. The first step is to locate the computing resources that can be allocated to HTCondor. The second step is to filter out the computing resources that can be allocated to each type of application by combining the application characteristics of HTC and the attributes of computing resources. Then, generate the list of applications that can be supported at any point in time for each computing resource.

At the IHEP computing cluster environment, some support systems are deployed to guarantee the stability of all computing services in the cluster, such as Cluster Information Management system, Cluster Monitoring system and Open Maintain Analysis Toolkits (OMAT). Cluster Information Management system records key information of computing cluster, and provides equipment management, experiment management and account management. The equipment management function is used to record the status of the equipment, such as in preparation, in repair, in use, scrapped; the purpose of equipment, such as storage server, cluster login node, HTC computing node, HPC computing node, GPU computing node, management server, etc; the experiment which purchase the equipment, such as BESIII, LHAASO etc. Based on this function, cluster administrator configures the appropriate software deployment template according to the purpose of equipment, and updates the equipment status to “in use” when the equipment’s software environment is ready. The experiment management function is used to record the experiment information supported by the cluster, such as which data access paths are required for the normal operation of the experimental job, the type of experimental job is HTC job or HPC job, etc. Cluster administrator update the system environment of all work nodes for the newly added experiment information so as to support the operation of all experiment jobs. The account management function is used to record the cluster account name, account membership experiment, account status and other information. Nagios[10], an open source software tool, has been deployed to implement the cluster service monitoring function. The cluster service monitoring system configures different detection service lists according to the properties of the devices. In order to ensure the normal operating environment of each HTC experiment job, some necessary service monitoring agents are configured for HTC computing nodes, such as whether the data storage path and software storage path of each experiment are accessible, whether the memory and local disk space of the node are exhausted, etc. Open Maintain Analysis Toolkits (OMAT), is an integrated framework based on a variety of open-source tools, which supports real-time data collection, fast correlation analysis of data from various data sources, has a variety of warning methods, and provides fast index, efficient query and rich visualization functions for monitoring data. Over the past few years, these support systems have kept HTC experiments running smoothly.

With the rapid growth of the scale of computing resources and the complex scheduling strategy of multiple experiments sharing resources, in order to achieve more efficient scheduling efficiency of HTC computing resources, a new generation of HTC computing resources management system, which based on Cluster Information Management system, Cluster Monitoring system and Open Maintain Analysis Toolkits, is designed and implemented.

3. The framework and detailed design

Figure 2 shows the framework of this HTC computing resource management system. It consists of the following principal components: a cluster information management module, computing resource management module, service monitoring module, data analysis module, policy update module and so on. The resource management system provides three interfaces for interactive operations for administrators and contains five main automated data analysis steps..

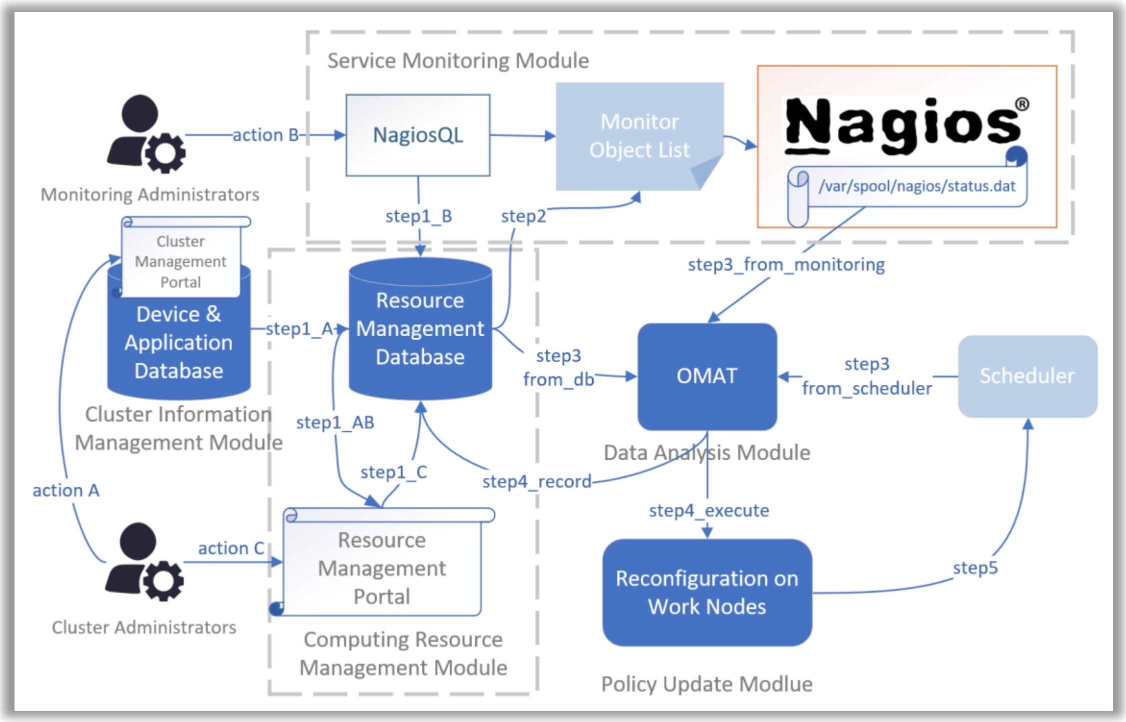

Service Monitoring Module
Monitoring Administrators
action B
NagiosQL
Monitor Object List
Nagios®
/var/spool/nagios/status.dat
step1_B
step2
step3_from_monitoring
Cluster Management Portal
Device & Application Database
step1_A
Resource Management Database
step3 from_db
OMAT
step3 from_scheduler
Scheduler
Cluster Information Management Module
step1_AB
Data Analysis Module
action A
step1_C
step4_record
step4_execute
action C
Resource Management Portal
Reconfiguration on Work Nodes
step5
Cluster Administrators
Computing Resource Management Module
Policy Update Modlue

Figure 2: The framework of HTCondor resource management at IHEP

3.1 Cluster information management module

The Cluster Information Management Module is a part of Cluster Information Management system, provides information on cluster equipment types, current device status, and experiments supported by the cluster. When the software environment of the new device is deployed, or the hardware failure causes the equipment to be unable to provide service, the cluster administrator will change the equipment state information recorded by this module in time, the same as “action A”.

3.2 Computing resource management module

Computing resource management module, as the most core module in the framework, undertakes the functions of resource management, policy management and state management. It contains resource management database and resource management portal.

The resource management database is designed with several data tables to records the necessary information for resources sharing scheme, as show in figure 3.

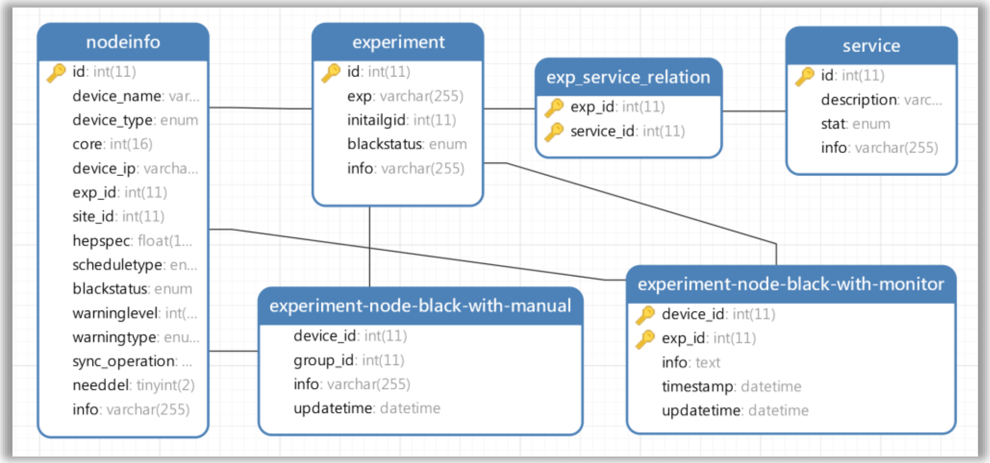


Figure 3: The main database table information of the resource management database

1. The table “nodeinfo” is used to maintain a list of current available resources to the HTC scheduler, by periodically synchronizing equipment, which type is HTC computing cluster node and status is “in use”, from the cluster information management module, the same as “step1_A”. The “sync_operation” field of this table is used to maintain the update status of the node's policy, that is, whether the latest resource sharing policy is successfully deployed on the node (this operation is performed in “step4_execute”). Under special circumstances, some nodes that already exist in the “nodeinfo” table will be temporarily removed from the computing resource pool by the cluster administrator through the resource management portal for debugging experimental jobs or other purposes. The “blackstatus” field of these nodes will be set to “black”, which means that these nodes are prohibited from providing computing services (this operation is performed in “action C”).

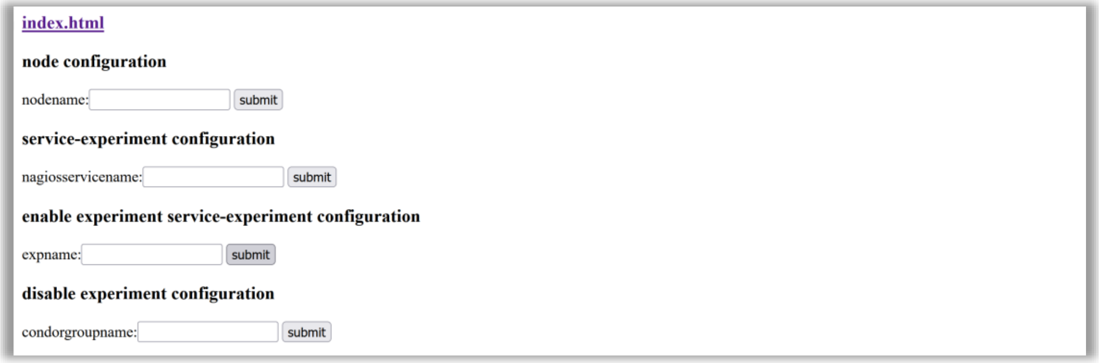
2. The table “experiment” is used to maintain a list of experiments that HTC can support, also by synchronizing experiment information from the cluster information management module, the same as “step1_A”. This table is also used “blackstatus” field to record the resource sharing policy supported by each experiment. This content of this field is white, which means that newly added computing resources support receiving the job of this experiment by default; content of this field is gray, which means that newly added computing resources do not support receiving the job of this experiment by default, and the administrator needs to configure the computing resources of this experiment separately; content of this field is black, which means that the entire computing resource is prohibited from receiving the job of this experiment, and will no longer provide computing services for the experiment.

3. The table “service” is used to maintain information about the monitoring services deployed by the Cluster Service Monitoring System, by synchronizing Nagios’s backend database, the same as “step1 B”. The status of newly introduced service is inactive by default, and will be change to activated status after the administrator associate it with some experiments.

4. The table “exp_service_relation” is used to record the mapping information between services and experiments, that is, which experiments are affected by each service exception on nodes. The “blackstatus” field of newly introduced experiment is “black” by default, and will be change to “gray” or “white” after the administrator associate it with some services (this operation is performed in “action C”).

5. The table “experiment-node-black-with-manual” is used to record the mapping relationship between nodes and experiments, that is, which experiment each node is assigned to support (this operation is performed in “action C”). Newly introduced nodes are allocated by default to those experiments that support full node sharing.

6. The table “experiment-node-black-with-monitor” is used to record the relationship between the node where the abnormal service occurs and the experiment which affected by the service (this operation is performed in “step3_record”).



index.html

node configuration

nodename:

service-experiment configuration

nagiosservicename:

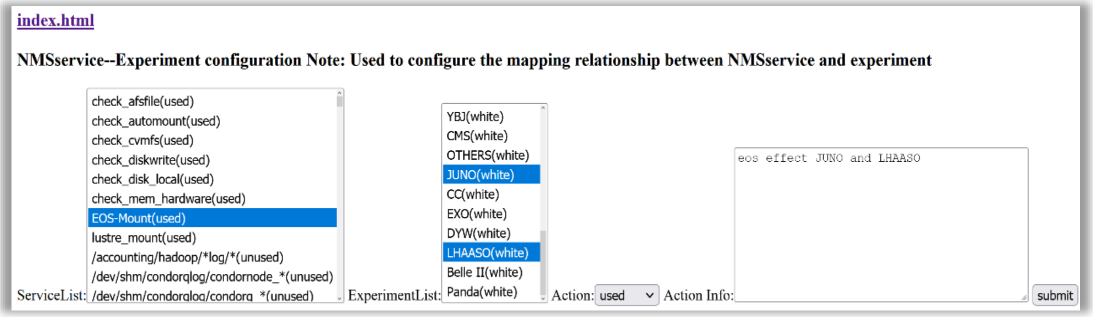
enable experiment service-experiment configuration

expname:

disable experiment configuration

condorgroupname:

Figure 4: The index webpage of resource management portal



index.html

NMSservice--Experiment configuration Note: Used to configure the mapping relationship between NMSservice and experiment

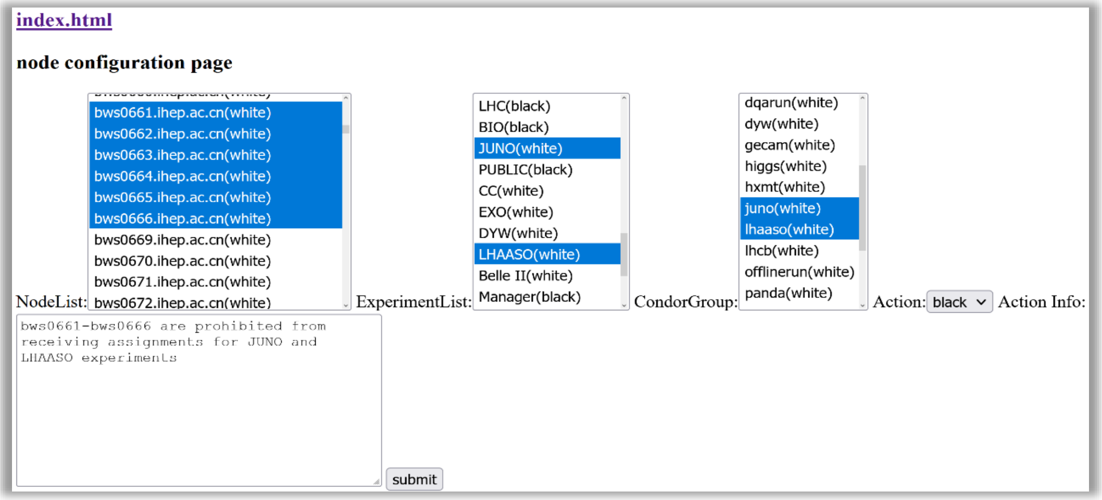
ServiceList:
check_afsfile(used)
check_automount(used)
check_cvmfs(used)
check_diskwrite(used)
check_disk_local(used)
check_mem_hardware(used)
EOS-Mount(used)
lustre_mount(used)
/accounting/hadoop/*log/*(unused)
/dev/shm/condorqlog/condornode_*(unused)
/dev/shm/condorqlog/condorq_*(unused)

ExperimentList:
YBJ(white)
CMS(white)
OTHERS(white)
JUNO(white)
CC(white)
EXO(white)
DYW(white)
LHAASO(white)
Belle II(white)
Panda(white)

Action: used

Action Info: eos effect JUNO and LHAASO

Figure 5: The service-experiment configuration webpage of resource management portal



index.html

node configuration page

NodeList:
bws0661.ihep.ac.cn(white)
bws0662.ihep.ac.cn(white)
bws0663.ihep.ac.cn(white)
bws0664.ihep.ac.cn(white)
bws0665.ihep.ac.cn(white)
bws0666.ihep.ac.cn(white)
bws0669.ihep.ac.cn(white)
bws0670.ihep.ac.cn(white)
bws0671.ihep.ac.cn(white)
bws0672.ihep.ac.cn(white)

ExperimentList:
LHC(black)
BIO(black)
JUNO(white)
PUBLIC(black)
CC(white)
EXO(white)
DYW(white)
LHAASO(white)
Belle II(white)
Manager(black)

CondorGroup:
dqarun(white)
dyw(white)
gecam(white)
higgs(white)
hxmt(white)
juno(white)
lhaaso(white)
lhcb(white)
offlinerun(white)
panda(white)

Action: black

Action Info:

bws0661-bws0666 are prohibited from receiving assignments for JUNO and LHAASO experiments

Figure 6: The nodes configuration webpage of resource management portal

The Resource Management Portal provides a configuration entry for cluster administrators, as shown in as show in figure 4. It presents the status of current experiments, services, and nodes to cluster administrators, the same as “step1_AB”. It is also used to query and configure the resource sharing relationships of each node and experiment that is shown in Figure 5, and the association relationships of experiments and services that is shown in Figure 6, these operations are performed in “action C”. Any configuration changes related to the node or service will affect the update status of the node’s policy, the same as “step1_C”.

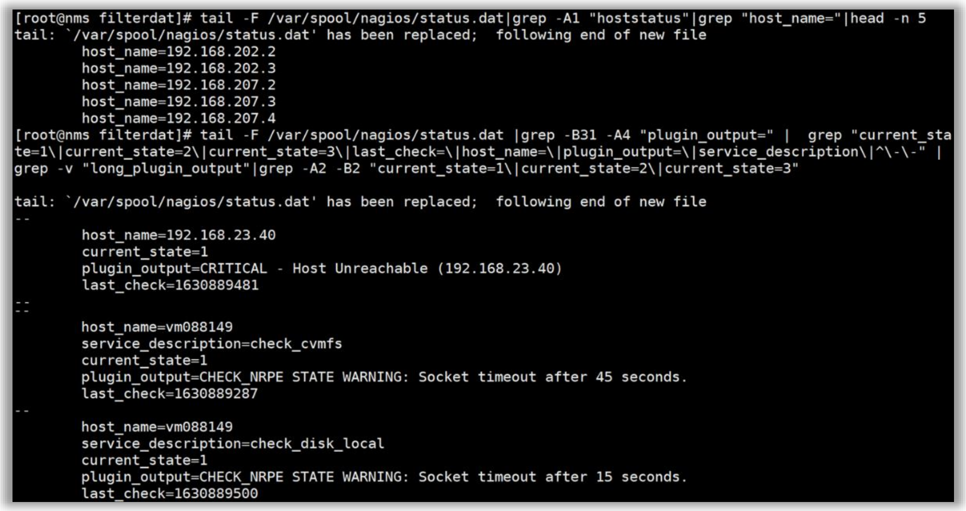
3.3 Service monitoring module

The Service Monitoring Module, based on Nagios monitoring tool, periodically retrieves node information from the “nodeinfo” table and updates the node list that needs to be included in the HTC cluster monitoring service template, the same as “step 2”. When the computing cluster environment changes, the monitoring administrator will adds new monitoring services according to the requirements of the operation environment of the experiment, like “action B”.

3.4 Data analysis module

The Data Analysis Module, as a typical stream processing and analysis task in the OMAT data analysis module, is used to associate the latest service monitoring data of the worker node with the configured worker node sharing strategy, and analyze the sharing strategy that the node actually needs to provide. The specific implementation steps include the following parts.

1. The service status monitoring function, which is responsible for real-time collection of the current monitored node list and abnormal service information of these nodes, as shown in Figure 7 . The output of these commands is pushed to the OMAT module via FileBeat[11], a popular log collection tool, the same as “step3_from_monitoring”.



```
[root@nms filterdat]# tail -F /var/spool/nagios/status.dat|grep -A1 "hoststatus"|grep "host_name="|head -n 5
tail: `/var/spool/nagios/status.dat' has been replaced;  following end of new file
        host_name=192.168.202.2
        host_name=192.168.202.3
        host_name=192.168.207.2
        host_name=192.168.207.3
        host_name=192.168.207.4
[root@nms filterdat]# tail -F /var/spool/nagios/status.dat |grep -B31 -A4 "plugin_output=" |  grep "current_sta
te=1\|current_state=2\|current_state=3\|last_check=\|host_name=\|plugin_output=\|service_description\|^\-\-" |
grep -v "long_plugin_output"|grep -A2 -B2 "current_state=1\|current_state=2\|current_state=3"

tail: `/var/spool/nagios/status.dat' has been replaced;  following end of new file
--
        host_name=192.168.23.40
        current_state=1
        plugin_output=CRITICAL - Host Unreachable (192.168.23.40)
        last_check=1630889481
--
--
        host_name=vm088149
        service_description=check_cvmfs
        current_state=1
        plugin_output=CHECK_NRPE STATE WARNING: Socket timeout after 45 seconds.
        last_check=1630889287
--
        host_name=vm088149
        service_description=check_disk_local
        current_state=1
        plugin_output=CHECK_NRPE STATE WARNING: Socket timeout after 15 seconds.
        last_check=1630889500
```

Figure7: Commands and output results for real-time collection of monitoring information

2. Locate unmonitored computing resources and triggers resource exception alerts, by comparing the set of nodes from “nodeinfo” table with the set of current monitored node from Nagios. Sets the state of these nodes to “require synchronization” and disables these node’s sharing state. This function is used to monitor and remedy the situation that the actual monitored object of the cluster monitoring module is inconsistent with the node to be monitored due to some unknown reasons.

3. Create a temporary node sharing policy, by analysis of the latest abnormal service information of nodes and the mapping between services and experiments from table

“exp_service_relation”, the same as “step3_from_db”. The abnormal service name of the node, the latest check time of the exception status, the specific affected experiment and other information are updated to “experiment-node-black-with-monitor” table. If the abnormal service information of the node has been recorded in this table, directly update the detection time field of the abnormal service, otherwise, insert the new abnormal service information of the node and set the content of “nodeinfo” table’s “sync_operation” field to “unsync”, the same as “step4 record”.

4. The temporary node sharing policy is maintained based on the following two rules. Instantly change the experiment sharing policy for nodes affected by the abnormal service, to prevent abnormal job running environments from persistently causing job errors. After the abnormal service of the node has been repaired for a period of time, the node will recover the sharing policy for the experiment which is impacted by the service. This recovery strategy is used to reduce the impact of monitoring service jitter on job scheduling. After any abnormal service related to worker nodes is detected, it is recorded in the “experiment-node-black-with-monitor” table forthwith. Abnormal service information that has not been updated for a long time means that the service has been repaired and the data will be deleted from “experiment-node-black-with-monitor” table.

5. Get a list of nodes that need to be updated by querying the “sync_operation” field of “nodeinfo” table. Compare the sharing policy for nodes assigned by cluster administrators with the temporary sharing policy effected by the result of these nodes’ monitoring service, create the sharing policy of nodes. Then get the node sharing state from “blackstatus” field of “nodeinfo” table and the experiment sharing state from “blackstatus” field of “experiment” table, the sharing policy is adjusted again. In order to improve the update efficiency, these policy files are concurrently actively pushed and deployed on nodes using the “check_nrpe” command. At last, modify the node’s “sync_operation” field from “unsync” to “sync” based on the execution result, the same as “step4_excute”.

6. The worker node heartbeat monitoring function, which is responsible for collection the set of nodes communicating with the HTC scheduler, by command “condor_status -master -pool”, the same as “step3_from_schedluer”.

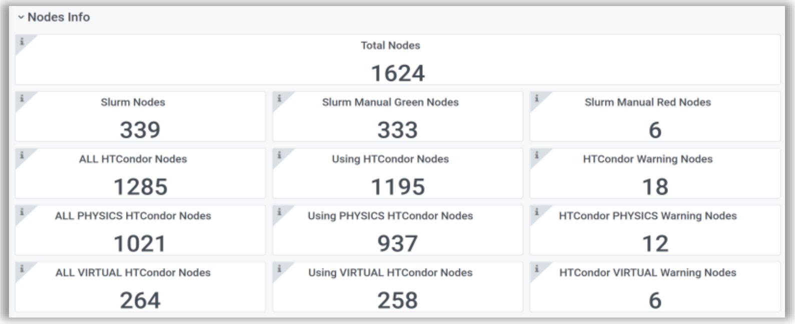
7. Detect the missing computing resources and trigger resource exception alerts, by comparing the set of sharing nodes from “nodeinfo” table with the set of current active nodes with the scheduler.

3.5 Policy update module

The policy update module, which is deployed on every HTC worker node, listen for policy information from the data analysis module using the NRPE service, receive and analyze the new sharing policy to create scheduler configuration and reload “StartD” service, the same as “step5”.

4. Information visualization webpage of HTC resource management system

The HTC resource management system has been deployed and running at IHEP computing center for more than two years, and the computing resources are allocated to different experiment . At the same time, the resource scheduling management information visualization webpage is designed in order to show the computing cluster management state to cluster administrator. It contains these following panels.



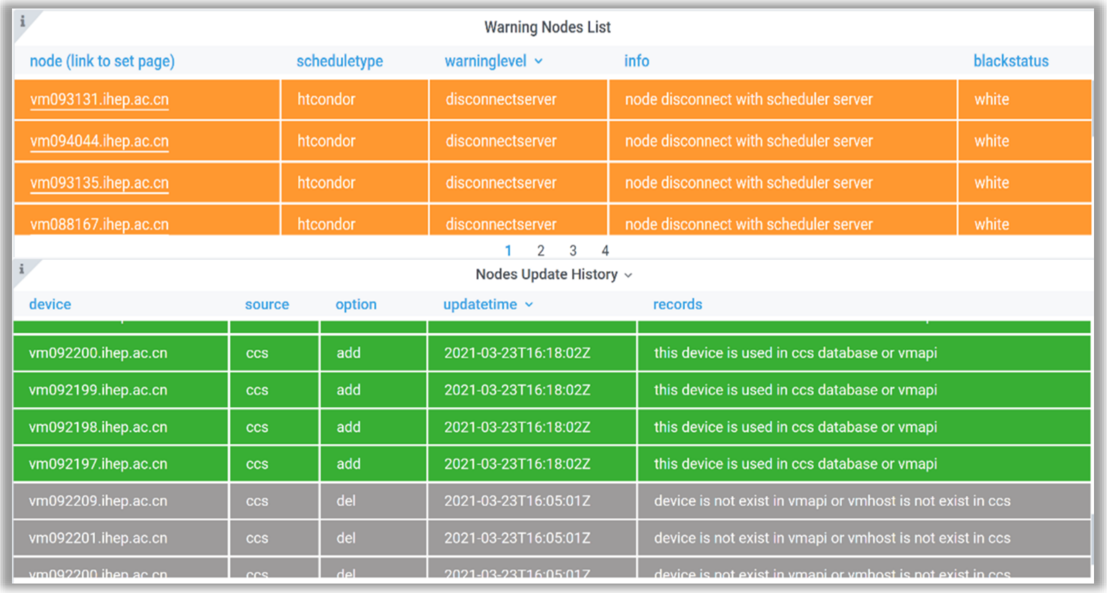
Nodes Info

Total Nodes
1624

Slurm Nodes 339	Slurm Manual Green Nodes 333	Slurm Manual Red Nodes 6
ALL HTCondor Nodes 1285	Using HTCondor Nodes 1195	HTCondor Warning Nodes 18
ALL PHYSICS HTCondor Nodes 1021	Using PHYSICS HTCondor Nodes 937	HTCondor PHYSICS Warning Nodes 12
ALL VIRTUAL HTCondor Nodes 264	Using VIRTUAL HTCondor Nodes 258	HTCondor VIRTUAL Warning Nodes 6

Figure 8: The distribution of HTC computing resources

Currently, it manages more than 1200 HTC computing devices, including physical computing resources and virtual computing resources, as shown in figure 8.



Warning Nodes List

node (link to set page)	scheduletype	warninglevel	info	blackstatus
vm093131.ihep.ac.cn	htcondor	disconnectserver	node disconnect with scheduler server	white
vm094044.ihep.ac.cn	htcondor	disconnectserver	node disconnect with scheduler server	white
vm093135.ihep.ac.cn	htcondor	disconnectserver	node disconnect with scheduler server	white
vm088167.ihep.ac.cn	htcondor	disconnectserver	node disconnect with scheduler server	white

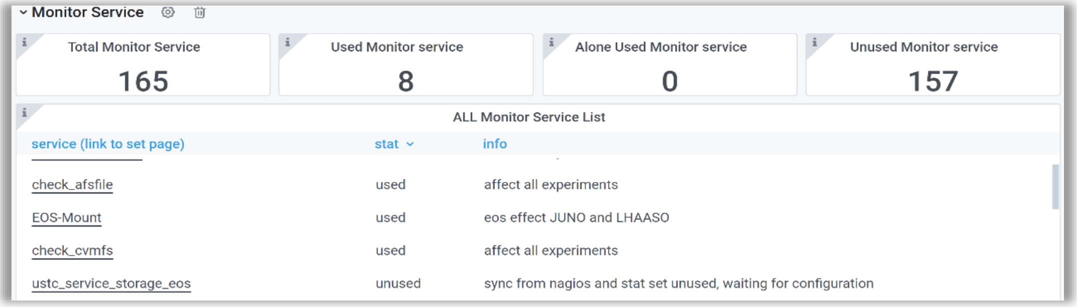
1 2 3 4

Nodes Update History

device	source	option	updatetime	records
vm092200.ihep.ac.cn	ccs	add	2021-03-23T16:18:02Z	this device is used in ccs database or vmapi
vm092199.ihep.ac.cn	ccs	add	2021-03-23T16:18:02Z	this device is used in ccs database or vmapi
vm092198.ihep.ac.cn	ccs	add	2021-03-23T16:18:02Z	this device is used in ccs database or vmapi
vm092197.ihep.ac.cn	ccs	add	2021-03-23T16:18:02Z	this device is used in ccs database or vmapi
vm092209.ihep.ac.cn	ccs	del	2021-03-23T16:05:01Z	device is not exist in vmapi or vmhost is not exist in ccs
vm092201.ihep.ac.cn	ccs	del	2021-03-23T16:05:01Z	device is not exist in vmapi or vmhost is not exist in ccs
vm092200.ihep.ac.cn	ccs	del	2021-03-23T16:05:01Z	device is not exist in vmapi or vmhost is not exist in ccs

Figure 9: The nodes' information of HTCondor resource management

Figure 9 shows the nodes that are currently disconnected from the scheduler and the history log of the synchronization node from the cluster information management module.



Monitor Service

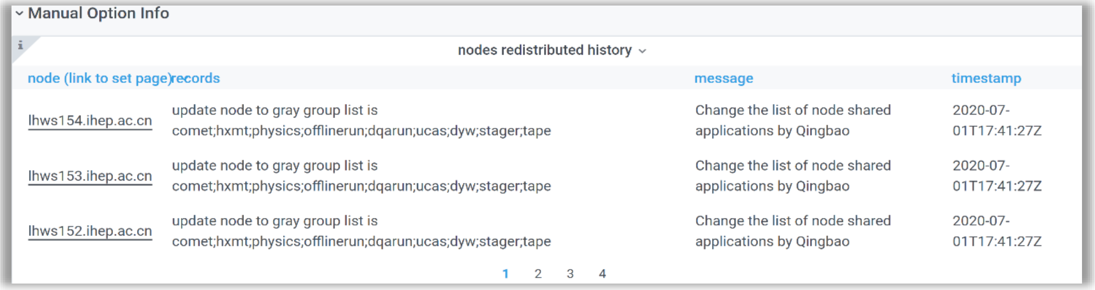
Total Monitor Service	Used Monitor service	Alone Used Monitor service	Unused Monitor service
165	8	0	157

ALL Monitor Service List

service (link to set page)	stat	info
check_afsfile	used	affect all experiments
EOS-Mount	used	eos effect JUNO and LHAASO
check_cvmfs	used	affect all experiments
ustc_service_storage_eos	unused	sync from nagios and stat set unused, waiting for configuration

Figure 10: The services' information of HTCondor resource management

Figure 10 shows a list of monitoring services that currently cover computing resources and which experiment are affected by them.

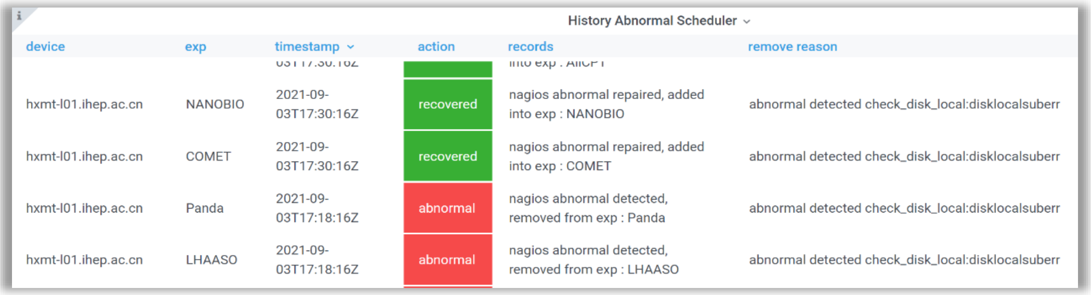


nodes redistributed history

node (link to set page)	records	message	timestamp
lhws154.ihep.ac.cn	update node to gray group list is comet;hxmt;physics;offlinerun;dqarun;ucas;dyw;stager;tape	Change the list of node shared applications by Qingbao	2020-07-01T17:41:27Z
lhws153.ihep.ac.cn	update node to gray group list is comet;hxmt;physics;offlinerun;dqarun;ucas;dyw;stager;tape	Change the list of node shared applications by Qingbao	2020-07-01T17:41:27Z
lhws152.ihep.ac.cn	update node to gray group list is comet;hxmt;physics;offlinerun;dqarun;ucas;dyw;stager;tape	Change the list of node shared applications by Qingbao	2020-07-01T17:41:27Z

Figure 11: The nodes' sharing strategy change log

The administrator can modify the allocation policy of the node through resource management portal, and the computing resources that are not covered by the monitoring are forbidden to be provided to the experiment service. The nodes' sharing strategy change log are shown as figure 11.



History Abnormal Scheduler

device	exp	timestamp	action	records	remove reason
		03T17:30:16Z		into exp : AliCPT	
hxmt-l01.ihep.ac.cn	NANOBIO	2021-09-03T17:30:16Z	recovered	nagios abnormal repaired, added into exp : NANOBIO	abnormal detected check_disk_local:disklocalsuberr
hxmt-l01.ihep.ac.cn	COMET	2021-09-03T17:30:16Z	recovered	nagios abnormal repaired, added into exp : COMET	abnormal detected check_disk_local:disklocalsuberr
hxmt-l01.ihep.ac.cn	Panda	2021-09-03T17:18:16Z	abnormal	nagios abnormal detected, removed from exp : Panda	abnormal detected check_disk_local:disklocalsuberr
hxmt-l01.ihep.ac.cn	LHAASO	2021-09-03T17:18:16Z	abnormal	nagios abnormal detected, removed from exp : LHAASO	abnormal detected check_disk_local:disklocalsuberr

Figure 12: The nodes' sharing policy based on monitoring service

Detect abnormal service information from Nagios in real time, analyze policies based on features, and disable or restore resource-sharing services between nodes and experiments, as shown in Figure 12.

5. Conclusions and Future Research

HTCondor computing resource management system based on the data acquisition and data stream processing functions of the OMAT module has been applied in the IHEP computing cluster. Based on data stream processing technology, the service monitoring information, service and experiment mapping relationship, node and experiment mapping relationship are combined to realize the dynamic adjustment of node and experiment sharing policy. The system avoids the situation that abnormal service nodes cause abnormal jobs, and ensures the stability of computing services. Separately provide resource management, experiment management, and service management configuration webpages to provide convenience for cluster administrators to manage resources. The implementation of idle resource warning strategy further guarantees the full utilization of resources. The resource management information visualization webpage presents detailed computing resource information to help administrators understand the cluster resource usage.

At IHEP, Slurm scheduling system is used to manage High Performance Computing (HPC) experiments. The current resource management system will be further improved to support efficient management of both HTC and HPC computing resources.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.11805226, No.11775250).

References

[1] Qingabo Hu, *Application of real-time stream processing technology in cluster monitoring of high energy physics computing*, SCE2019, Zunyi, China, July 15-19, 2019

[2] Zhen-An, *Status of BEPCII/BESIII project*, In Accelerator and particle physics, Proceedings, 9th Winter Institute, APPI 2004, Appi, Japan, February 16-20, 2004, pages 86-90, 2004.

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

[4] Univ Pisa Scuola Normale Super Pisa Pisa Tenchini, R, *The CMS experiment at the CERN LHC*, JOURNAL OF INSTRUMENTATION,3, 2008.

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

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

[7] Jingyan Shi, *The Dynamic Scheduling Strategy of HTCondor at IHEP*, HTCondor Week 2017.

[8] Douglas Thain, Todd Tannenbaum, and Miron Livny, *Distributed computing in practice: the Condor experience*, Concurrency and Computation: Practice and Experience, February-April, 2005, Vol. 17, No. 2-4, pages 323-356.

[9] R. Raman, M. Livny and M. Solomon, *Matchmaking: distributed resource management for high throughput computing*, Proceedings. The Seventh International Symposium on High Performance Distributed Computing (Cat. No.98TB100244), 1998, pp. 140-146

[10] Guthrie, M., *Instant Nagios starter an easy guide to getting a Nagios server up and running for monitoring, altering, and reporting*. Birmingham: Packt Pub.

[11] FileBeat Introduction, https://www.elastic.co/beats/filebeat, online, accessed 06-Sep-2021