Studies on Job Queue Health and Problem Recovery

Xiaowei Jiang[*]
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: jiangxw@ihep.ac.cn

Jiaheng Zou
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: zoujh@ihep.ac.cn

Jingyan Shi
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: shijy@ihep.ac.cn

Ran Du
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: duran@ihep.ac.cn

Qingbao Hu
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: huqb@ihep.ac.cn

Zhenyu Sun
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: sunzy@ihep.ac.cn

Hongnan Tan
Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
E-mail: Tanhn@ihep.ac.cn

Abstract. In a batch system, the job queue is in charge of a set of jobs. Job health is the most important issue concerned by users and administrators. The job state can be queuing, running, completed, error or held, etc, that can reflect the job health. Generally jobs can move from one state to another. However, if a job keeps in a state for too long time, there might be problems, such as worker node failure and network blocking. In a large-scale computing cluster, problems cannot be avoided. That means a number of jobs will be blocked in one state, and cannot be completed in an expected time. This will delay the progress of the computing task. For that situation, this paper studies on the unhealthy job state's reason, problem handling and job queue stability. We aim to improve the job health, and then we can improve job success rate and speed up users' task progress. Unhealthy reasons can be found from job attributes, queue information and logs, which can be analyzed in detail to acquire better solutions. Depending on who do the recovery, all the solutions are grouped into two categories. The first category is recovered by administrator. Most problems are automatically solved through integrating with the monitor system. When problem is solved, the corresponding job will be rescheduled in time, without involving users. The second category is automatically informing users to dispose unhealthy jobs by themselves. In accordance with the results of unhealthy analysis, the helpful suggestion might be recommended to users for quick recovery. Based on the foregoing methods, a job queue health system is designed and implemented at IHEP. We define a series of standards to pick out unhealthy jobs. Various factors relevant with unhealthy jobs are collected and analyzed in association. In case that unhealthy jobs could be recovered at admin side, automatic recovery functions are carried out to automatically recover the unhealthy jobs. In case that unhealthy jobs must be recovered at user side, alarms are sent to users via emails, WeChat, etc. The running status of job queue health system indicates that it's able to improve the job queue health in most situations.

International Symposium on Grids and Clouds (ISGC) 2018 in conjunction with Frontiers in Computational Drug Discovery
16-23 March 2018
Academia Sinica, Taipei, Taiwan

*Speaker.

1. Introduction

At IHEP(Institute of High Energy physics), high throughput computing cluster is managing about 11,000 cpu cores and providing computing resources for most experiments, including BESIII [1], DYW [2], CMS [3], LHAASO [4], JUNO [5], etc. After deploying one specific share policy [6], resource usage improves from around 50% to above 85%, nearly 100% at busy time. In 2017, and more than 50 million jobs are completed, average 150,000 jobs per day. So a large number of jobs are always concurrently stored in job queue. As shown in figure 1, the bars represent the count of idle jobs submitted by users from different experiments, that means large amount of jobs keep idle state at any time to compete for resources in job queue. Meanwhile, the resource usage is reaching up to nearly 90%, that means large amount of jobs keep running state in job queue. In some ways, large quantity of job means a variety of job failures.

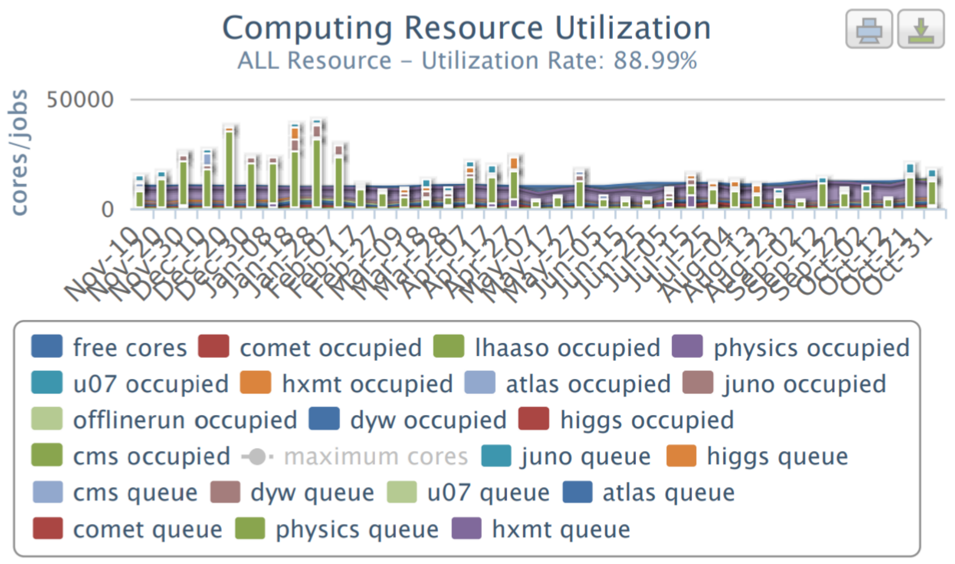


Figure 1: job and resource status at IHEP in recent year

As the local batch system of IHEP, HTCondor [7] is used to manage the high throughput computing cluster. Around HTCondor, several tools and softwares were developed for localized application, including central control system, unified front-end job toolkits, monitoring and accounting, as shown in figure 2. Therefore, the studies and exploits in this paper will be discussed based on HTCondor.

In HTCondor, job queue is maintained by SCHEDD. With a high performance, HTCondor's job queue can be effectively alive under pressure of hundred thousands of jobs. The number of HTCondor's job state is 7 (in the version of 8.6.12), "Idle", "Running", "Removed", etc. Integrated with the fine-grained conditions, more kinds of job states could be assumed. This paper studies on the job health, job queue stability and problem handling. We aim to guarantee job health, improve job success rate and speed up users' task progress.

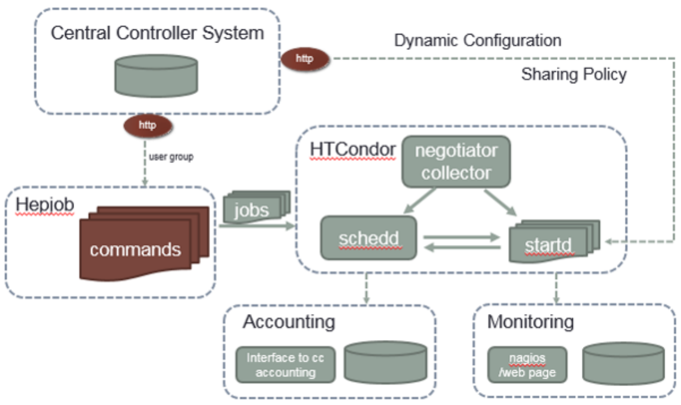


Figure 2: toolkits and softwares around HTCondor at IHEP

2. Unhealthy Job

Generally, the job state can be queuing, running, completed, error or held, etc. A job can smoothly change from one state to another. However, if the job keeps in one state for too long time, there might be problems, such as worker node failure or network blocking. In high throughput computing cluster, problems would not be avoided and a fair amount of jobs would be blocked in one state, this kind of job can not be completed in an expected time. We simply give a name "unhealthy job" to the jobs blocked in one state.

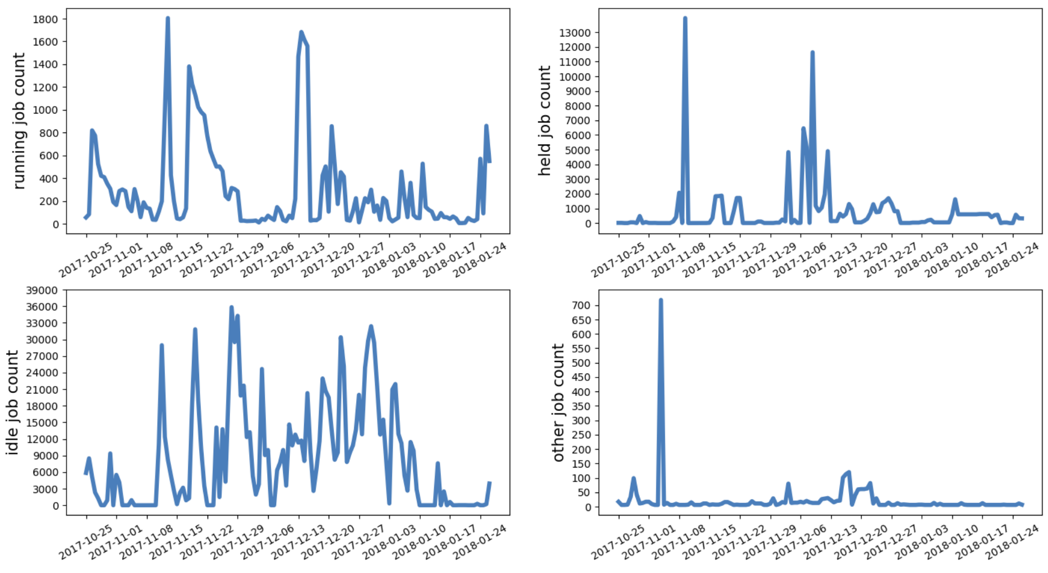


Figure 3: unhealthy job statistics during Oct. 2017 to Jan. 2018

The figure 3 shows unhealthy job statistics during Oct. 2017 to Jan. 2018 in HTC cluster of IHEP. The sub-figure *running* shows that the count of unhealthy running jobs, which are running over a limit time, peaks to 1,800; The sub-figure *held* shows that the count of unhealthy held jobs peaks to 13,000; The sub-figure *idle* shows that the count of unhealthy idle jobs, which are idle over a limit time, peaks to 35,000; The sub-figure *other* shows that the count of other unhealthy jobs, such as removed unsuccessfully, peaks to 700.

3. Basic Workflow

For the purpose of repairing unhealthy jobs, several toolkits and softwares are developed to prevent, check, recover and alarm unhealthy jobs. The basic workflow of repairing unhealthy jobs can be summarized as follow:

1. **prevention**: with the unified front-end job toolkits, which are developed by ourselves and named with 'hepjob' [8], effective preventions can be taken.

2. **unhealthy job check**: as the characters of unhealthy jobs, unhealthy jobs can be checked out from job queue. Then, these jobs go through filtering, simplifying and classifying. After that, unhealthy jobs are classified in different classifications and appended with simplified tags, which can indicate the main reason of unhealth.

3. **problem handling**: in accordance with the different classifications and reason-tags of unhealthy jobs, the corresponding recovery operations will be done at worker node side, job queue side and user side.

4. **alarm**: The unhealthy information should be sent to users and administrators in time. The measures of alarm are various, a new popular measure (WeChat [9] public account) will be taken, which is freely provided by Tencent [10].

4. Unhealthy Job Prevention

In HTCondor, each job in job queue has a set of attributes under classad mechanism. These attributes are updated in real-time by SCHEDD. Some attributes keep useful information which can indicate the unhealthy job. For example, 'HoldReason' shows the reason of why job changes to held state. For another example, 'EnteredCurrentStatus' shows the time when job changes to current state, which is facilitated to evaluate whether a job is over a limit time. Meanwhile, there are other incorrect or imprecise attributes leading to failed match, that will make jobs blocked in a frozen idle state. If any prevention measure could be taken to check out and correct this kind of attributes, lots of unhealthy job might be avoid.

In our environment, prevention measures are implemented in hepjob—the unified front-end job toolkits. hep_sub, one of commands provided by hepjob, is used to submit jobs. In hep_sub, the pre-check functions help users to set the right match configuration. As figure 4 shows, pre-check is implemented in configuration setter.

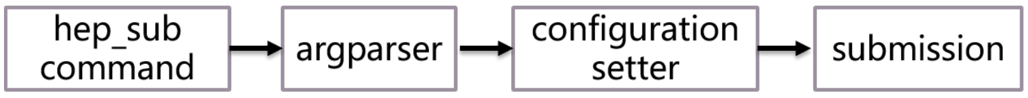


Figure 4: hep_sub process

As an example, accounting group [11] is the major attribute in job classad, because resources are divided into different partitions based on accounting groups, user only can request the resources in his/her groups. Once hep_sub is executed on the terminal, accounting group will be checked whether user group is equal to the requesting resource group and whether there are available resources in his/her group.

5. Unhealthy Job Check

Unhealthy job check process is completed in three steps, as shown in figure 5. **Extracting**: as the pre-defined conditions, which are defined based on the relevant attributes of unhealthy job, unhealthy jobs can be periodically checked out from job queue. **Filtering**: some unhealthy jobs still need to be filtered, because there are often special users requesting special requirements. Therefore, a filtered user list is set to ignored special users' unhealthy jobs. **Analyzing**: the rest of unhealthy jobs will be analyzed and the analysis results will be saved in a specific structure. The result structure is depending on the next handling process. In our case, the result structure is defined as a dictionary.

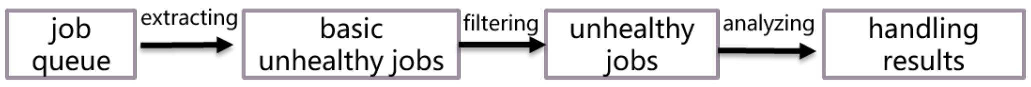


Figure 5: unhealthy job check process

6. Problem Handling

Once the unhealthy jobs are sent from the previous analysis, the handling process will start. As figure 6 shows, the handling process is completed in three branches based on different kinds of problems.

User-end problem must be handled at user-end, which are caused by user misuse and hard to be handled at admin-end, such as writing in a directory with no writable permission. But some efficient recommendations might be sent from administrators to users, helping them to recovery their jobs. Lasting problem and temporary problem should be verified in the monitoring system which is constructed based on nagios [12]. Without any interruption, the lasting problem can exist for a long time or ever, but lasting problem can be catched by monitoring system. Oppositely, the temporary problem can only exist for a short time or a flash time, and temporary problem can not be catched by monitoring system, but it does still have a bad influence on jobs. For lasting problem, central control system (developed by ourself) can solve them at worker node side, and then unhealthy jobs will be recovered and put back to job queue. For temporary problem, the corresponding unhealthy jobs are only reset to idle state and waiting for next matching, because temporary problem could vanish in a very flash time by itself. For double confirmation, alarm information still should be sent to users or administrators.

Due to the different categories of problems, recovery measures should be taken at three sides: worker node side, job queue side and user side.

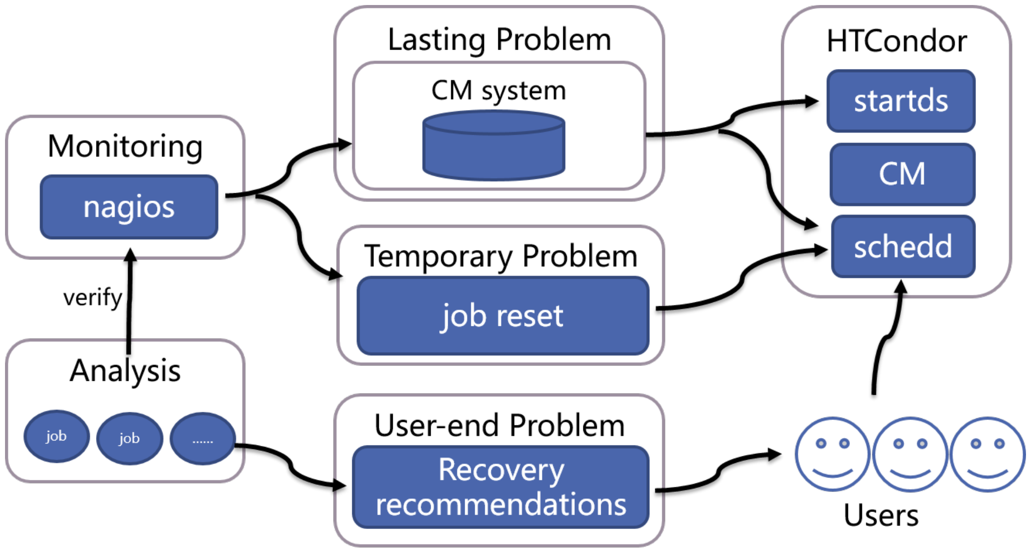
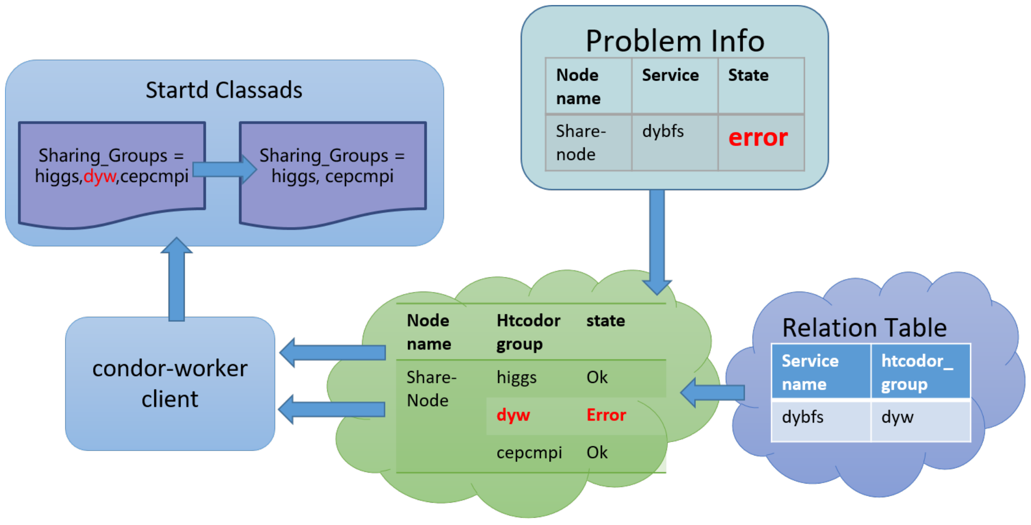


Figure 6: problem handling process

6.1 Problems recovery at worker node side

At worker node side, we developed a central control system(CC) to control the STARTD classad. Under the help of CC system, if a problem is happening on a problematic worker node and this problem only affects a specific group, this group will be forbidden to run its job on this worker node. The affected group would be removed from the startd attribute of the matching list. The process is shown as figure 7 . It's a simple but common problem handling process.


Startd Classads
Sharing_Groups = higgs,dyw,cepcmpi
Sharing_Groups = higgs, cepcmpi
Problem Info
condor-worker client
Relation Table

Node name	Service	State
Share-node	dybfs	error

Node name	Htcodor group	state
Share-Node	higgs	Ok
	dyw	Error
	cepcmpi	Ok

Service name	htcodor_ group
dybfs	dyw

Figure 7: problem recovery at worker node

As the handling results, problem information will be extracted and formatted as three fields, node name, service and state. In CC system, there are a database storing the startd classad information. Two tables are related to group and health state, relation table and group status table.

According to relation table, the unhealthy group can be found and changed to error status. At worker node, a condor-worker client is listening for the group status table. Once group state is changed, the classad information would be pulled down and updated into STARTD classads. As figure 7 shows, dyw group is unhealthy on worker node and removed from the Sharing_Groups list.

6.2 Problem recovery at job queue side

After problem handling or reason analysis, some of unhealthy jobs can be recovered at job queue side.

1. Some of held jobs are directly released if they are impacted by temporary problems, and other held jobs are released after problems handling if they are impacted on by lasting problems. Certainly, some of held jobs only need to be handled at user-end.

2. Some of unhealthy idle jobs should be removed if they are blocked or over the limit time; others should be correct if they are configured with incorrect requirement conditions; the rest should be handled at user-end.

3. Some of unhealthy running jobs should be removed if they are over the pre-defined limit time; others should be recovered back to job queue and waiting next match, if they are impacted by problems on worker node; the rest should be handled at user-end.

4. Other unhealthy jobs, such as the jobs blocked in removed state, should be removed or recovered back to job queue.

At job queue side, via HTCondor's operations, such as condor_release, condor_rm, condor_qedit and so on, unhealthy jobs would be released, removed or corrected in time as the results of unhealthy job analysis and problem handling.

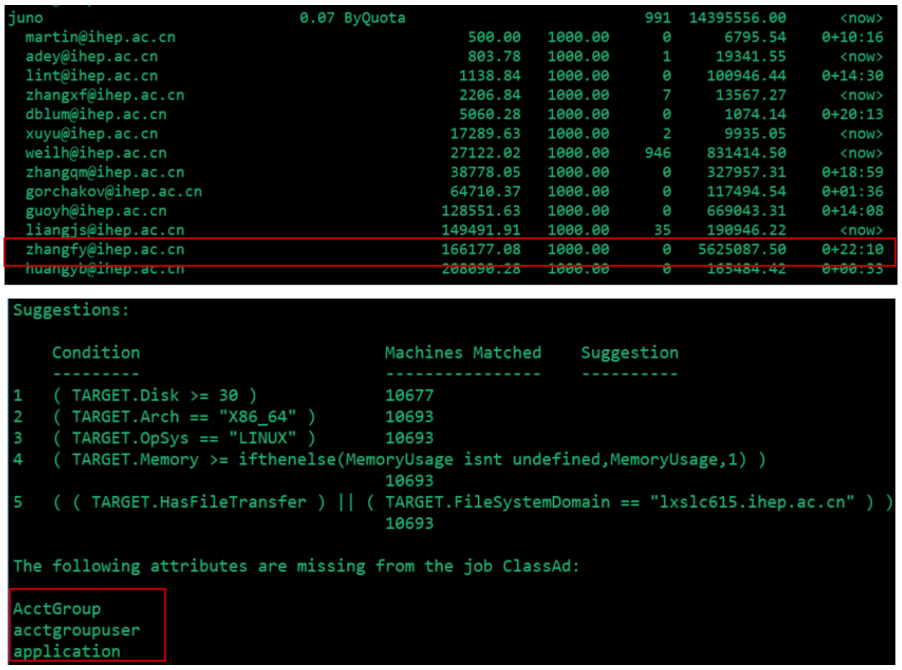
6.3 Problem recovery at user end

As the above mentioned, under the situations which unhealthy jobs can not be recovered by administrators, these jobs should be recovered or removed at user end. In our case, administrators only send alarms to users, once unhealthy jobs are checked out. However, some effective suggestions could be recommended to help them recover their unhealthy jobs in time.

In HTCondor, with its open features, the reason of unhealthy jobs can be easily found out and make solutions for users. As the top sub-figure in figure 8 shows, users' next match time can be predicted as the real priority and PRIORITY_HALFLIFE if their jobs are idle for a long time; As the bottom sub-figure shows, the incorrect attributes can be extracted, analyzed and formatted as a recovery recommendation for users. The recommendation functions should be implemented in the future work.

7. Alarm

Alarm is an important part in the unhealthy recovery process. Whatever problems are handled automatically or manually, alarm messages should be sent to administrators. With the alarm messages, administrator could take effective measures to manually recover unhealthy jobs in time or verify whether the automatical problem handling does work. Meanwhile, alarm messages should be sent to users, reminding them to recover unhealthy jobs in time or informing them that unhealthy



```
juno                              0.07 ByQuota                                 991  14395556.00      <now>
  martin@ihep.ac.cn                                        500.00   1000.00      0      6795.54    0+10:16
  adey@ihep.ac.cn                                          803.78   1000.00      1     19341.55      <now>
  lint@ihep.ac.cn                                         1138.84   1000.00      0    100946.44    0+14:30
  zhangxf@ihep.ac.cn                                      2206.84   1000.00      7     13567.27      <now>
  dblum@ihep.ac.cn                                        5060.28   1000.00      0      1074.14    0+20:13
  xuyu@ihep.ac.cn                                        17289.63   1000.00      2      9935.05      <now>
  weilh@ihep.ac.cn                                       27122.02   1000.00    946    831414.50      <now>
  zhangqm@ihep.ac.cn                                     38778.05   1000.00      0    327957.31    0+18:59
  gorchakov@ihep.ac.cn                                   64710.37   1000.00      0    117494.54    0+01:36
  guoyh@ihep.ac.cn                                      128551.63   1000.00      0    669043.31    0+14:08
  liangjs@ihep.ac.cn                                    149491.91   1000.00     35    190946.22      <now>
  zhangfy@ihep.ac.cn                                    166177.08   1000.00      0   5625087.50    0+22:10
  huangyb@ihep.ac.cn                                    208090.28   1000.00      0    165484.42    0+00:33
```

```
Suggestions:

    Condition                            Machines Matched    Suggestion
    ---------                            ----------------    ----------
1   ( TARGET.Disk >= 30 )                10677
2   ( TARGET.Arch == "X86_64" )          10693
3   ( TARGET.OpSys == "LINUX" )          10693
4   ( TARGET.Memory >= ifthenelse(MemoryUsage isnt undefined,MemoryUsage,1) )
                                         10693
5   ( ( TARGET.HasFileTransfer ) || ( TARGET.FileSystemDomain == "lxslc615.ihep.ac.cn" ) )
                                         10693

The following attributes are missing from the job ClassAd:

AcctGroup
acctgroupuser
application
```

Figure 8: case of problem recovery at user end

jobs have been recovered automatically. In our case, two types of alarm mode are applied, WeChat and Mail.

1.WeChat: it's a multi-purpose messaging and social media application developed by Tencent. WeChat provides a public account for companies or communities freely. The public account manages users tree levels, including user, group and application. We can define system administrators in different groups or applications. As a alarm mode, its advantage is quick and real-time, like the mobile message.

Figure 9 shows the basic structure of WeChat alarm. Once alarm message should be sent, WeChat interface will organize the content with the pre-defined templates. WeChat server will transmit alarm massages to administrators.

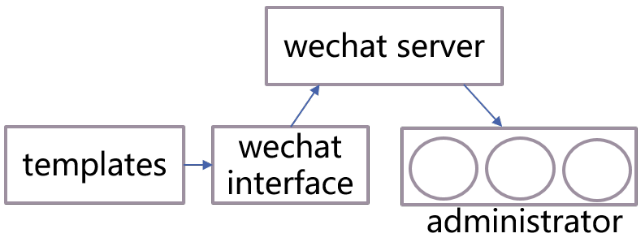


Figure 9: WeChat alarm process

2.Mail: it's commonly used in alarm system. In our case, mail mode is similar with WeChat mode. Different part is that mail mode has no user management, so that we build a user-info database to manage user information. When alarm mail should be sent, mail interface will organize the content with the pre-defined templates and get user email address from user-info database. Then, mail server will transmit the mail to the corresponding user. The basic process is shown in figure 10.

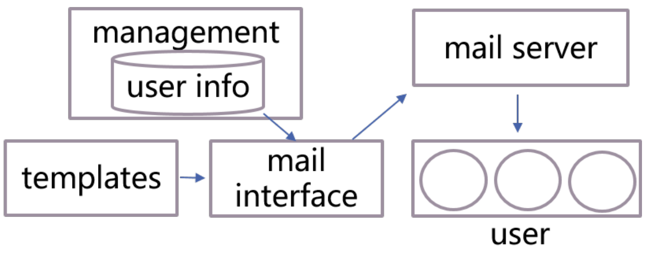


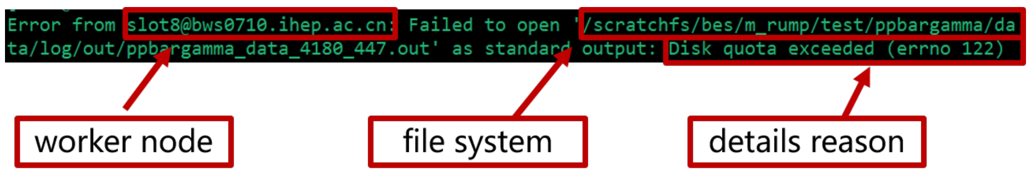
Figure 10: mail alarm process

8. Application Case

This paper shows a basic process of unhealthy job recovery. And we developed a set of toolkits to complete prevention, check, problem handling and alarm. As the typical unhealthy job, held job will be a representative application case to show the basic handling process.

1.**prevention**: job script file must be existing, writable and executable, or it will cause held job. In hepjob, the job script file will be pre-checked. If the file cannot pass the pre-check, the alarms and suggestions will be printed to help user fix it.

2.**check**: all held jobs are regarded as unhealthy jobs. So all of them will be checked out from job queue by the 'Held' job state. In HTCondor, the attribute 'HoldReason' saves a formatting held reason, so that the detailed reason can be extracted from this attribute. As a case of held job shown in figure 11, the problem worker node, the error file system and the detailed reason can be extracted from the attribute 'HoldReason'.



```
Error from slot8@bws0710.ihep.ac.cn: Failed to open '/scratchfs/bes/m_rump/test/ppbargamma/da
ta/log/out/ppbargamma_data_4180_447.out' as standard output: Disk quota exceeded (errno 122)
```

Figure 11: 'HoldReason' in held job classad

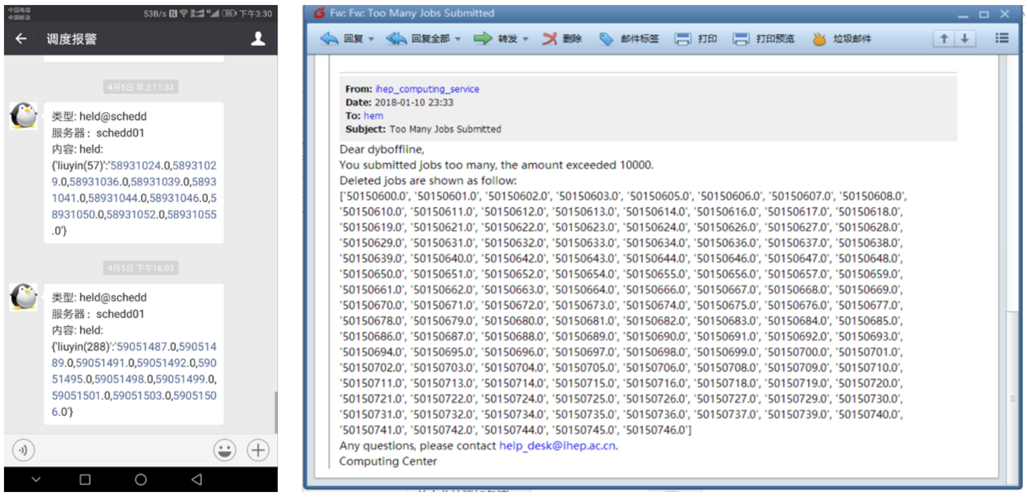
Table 1 shows a small part of held reasons collected as defined format.

3.**handling**: one specific held job recovery has been described and shown in figure 7. When group "dyw" is removed from worker node, the held jobs impacted by this node will be changed to idle and wait for next match.

4.**alarm**: once held job is checked or recovered, alarm messages will be sent to administrators. Some alarm messages via WeChat are shown in figure 12. Meanwhile, an alarm message via email is also shown in, but it's not for held jobs.

Table 1: part of formatting held reasons

Status	Type	Type content	Reason
held	iwd	Cannot access initial working directory	No such file or directory
held	startd	Error from %: failed to open % as standard output	Input/output error (errno 5)
held	startd	Error from %: failed to open % as standard output	Permission denied (errno 13)
held	startd	Error from %: failed to open % as standard output	Transport endpoint is not connected (errno 107)



类型: held@schedd
服务器： schedd01
内容: held:
{'liuyin(57)':'58931024.0,58931029.0,58931036.0,58931039.0,58931041.0,58931044.0,58931046.0,58931050.0,58931052.0,58931055.0'}

类型: held@schedd
服务器： schedd01
内容: held:
{'liuyin(288)':'59051487.0,59051489.0,59051491.0,59051492.0,59051495.0,59051498.0,59051499.0,59051501.0,59051503.0,59051506.0'}

From: ihep_computing_service
Date: 2018-01-10 23:33
To: hem
Subject: Too Many Jobs Submitted

```
Dear dyboffline,
You submitted jobs too many, the amount exceeded 10000.
Deleted jobs are shown as follow:
['50150600.0', '50150601.0', '50150602.0', '50150603.0', '50150605.0', '50150606.0', '50150607.0', '50150608.0',
'50150610.0', '50150611.0', '50150612.0', '50150613.0', '50150614.0', '50150616.0', '50150617.0', '50150618.0',
'50150619.0', '50150621.0', '50150622.0', '50150623.0', '50150624.0', '50150626.0', '50150627.0', '50150628.0',
'50150629.0', '50150631.0', '50150632.0', '50150633.0', '50150634.0', '50150636.0', '50150637.0', '50150638.0',
'50150639.0', '50150640.0', '50150642.0', '50150643.0', '50150644.0', '50150646.0', '50150647.0', '50150648.0',
'50150650.0', '50150651.0', '50150652.0', '50150654.0', '50150655.0', '50150656.0', '50150657.0', '50150659.0',
'50150661.0', '50150662.0', '50150663.0', '50150664.0', '50150666.0', '50150667.0', '50150668.0', '50150669.0',
'50150670.0', '50150671.0', '50150672.0', '50150673.0', '50150674.0', '50150675.0', '50150676.0', '50150677.0',
'50150678.0', '50150679.0', '50150680.0', '50150681.0', '50150682.0', '50150683.0', '50150684.0', '50150685.0',
'50150686.0', '50150687.0', '50150688.0', '50150689.0', '50150690.0', '50150691.0', '50150692.0', '50150693.0',
'50150694.0', '50150695.0', '50150696.0', '50150697.0', '50150698.0', '50150699.0', '50150700.0', '50150701.0',
'50150702.0', '50150703.0', '50150704.0', '50150705.0', '50150706.0', '50150708.0', '50150709.0', '50150710.0',
'50150711.0', '50150713.0', '50150714.0', '50150715.0', '50150716.0', '50150718.0', '50150719.0', '50150720.0',
'50150721.0', '50150722.0', '50150724.0', '50150725.0', '50150726.0', '50150727.0', '50150729.0', '50150730.0',
'50150731.0', '50150732.0', '50150734.0', '50150735.0', '50150736.0', '50150737.0', '50150739.0', '50150740.0',
'50150741.0', '50150742.0', '50150744.0', '50150745.0', '50150746.0']
Any questions, please contact help_desk@ihep.ac.cn.
Computing Center
```

Figure 12: alarm case

9. Summary

Unhealthy jobs always sit in job queue, especially in HTC cluster, which lead to delaying experiment data processing. Quick and effective unhealthy job recovery can help users to save the time spending on job maintenance and promote the progress of experiment data processing.

This paper shares a basic handling process of unhealthy jobs, including prevention, check, problem handling and alarm. And a set of toolkits are developed to implement this process. However, there are lots functions which will be optimized and implemented in the future work.

1. The automatical handling process should be optimized. Now, only held jobs could be recovered automatically, other unhealthy jobs are still recovered manually after alarming. But the implement process is similar with held job recovery, it can be completed soon.

2. More effective recommendations should be generated and provided to users. Lots of useful recommendation information can be found out from HTCondor's components. These information would make more contributions.

3. More logs should be recorded. Logs do not only stored the useful information for troubleshooting but also give some results to prove the efficiency.

Acknowledgements

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, Appi, Japan, February 16-20, 2004,* pages 86-90, 2004.

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

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

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

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

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

[7] 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.

[8] Xiaowei Jiang, *a front scheduling tool for high energy physics, SCE2017, Weihai, China, July 3-7, 2017.*

[9] WIKIPEDIA, *https://en.wikipedia.org/wiki/WeChat*

[10] WIKIPEDIA, *https://en.wikipedia.org/wiki/Tencent*

[11] Xiaowei Jiang, Jiaheng Zou, Yaodong Cheng, Jingyan Shi, *A multi-group and preemptable scheduling of cloud resource based on HTCondor,Journal of Physics(2017): Conference Series,* volume 898, number 9, pages 092051

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