EFFICIENCY MEASUREMENT SYSTEM FOR THE COMPUTING CLUSTER AT IHEP

V. Ezhova [a], V. Kotliar [b]

Institute for High Energy Physics named by A.A. Logunov of National Research Centre "Kurchatov Institute", Nauki Square 1, Protvino, Moscow region, Russia, 142281

E-mail: [a] Victoria.Ezhova@ihep.ru , [b] Viktor.Kotliar@ihep.ru

Every day IHEP central computing cluster produce thousands of calculations related to research activities, both IHEP and GRID experiments. A lot of machine resources are expended on this work. So, we can estimate the size of the spent resources used for all types of tasks, make decisions for changing cluster configuration and to do the forecast for the work of the computer center in general. In this work you can see the calculations of the efficiency index and the graphical representation of work of a cluster on the basis of account information. It is one of the main tasks within work on creation of system of uniform monitoring of computer center of IHEP.

Keywords: accounting system, Torque pbs, Elastic Search, Kibana

1. Introduction

In this work you can see a graphical representation of the work of the cluster based on the account information from the torque pbs cluster for real-time analysis. The work of the main scripts and a method for calculating cluster performance based on pbs data will be present in details, as well as the results of testing of some memory allocation systems will be described.

We pursued the goal of improving the effectiveness of the cluster through the efficient allocation of resources, for example, by introducing additional libraries.

2. Review of the work performed and calculation of efficiency

To start the task on the cluster, you might use the Torque PBS job management system. This system contains information about the necessary resources (the number of cluster nodes, the required amount of RAM and the necessary execution time). After starting the task, all this information is placed in a log file and stored on the server [1].

The python script sorts the accounting file of torque pbs into a python dict and throws off this information in JSON format. It uses the alogger.utils - small python library to parse resource manager logs and json module which can generate JSON from python objects and lists [2]. As next step bash script connects to Elastic Search (ES) and it sends JSON data to ES every minute to display it in the form of the schedule from Kibana (Figure 1).

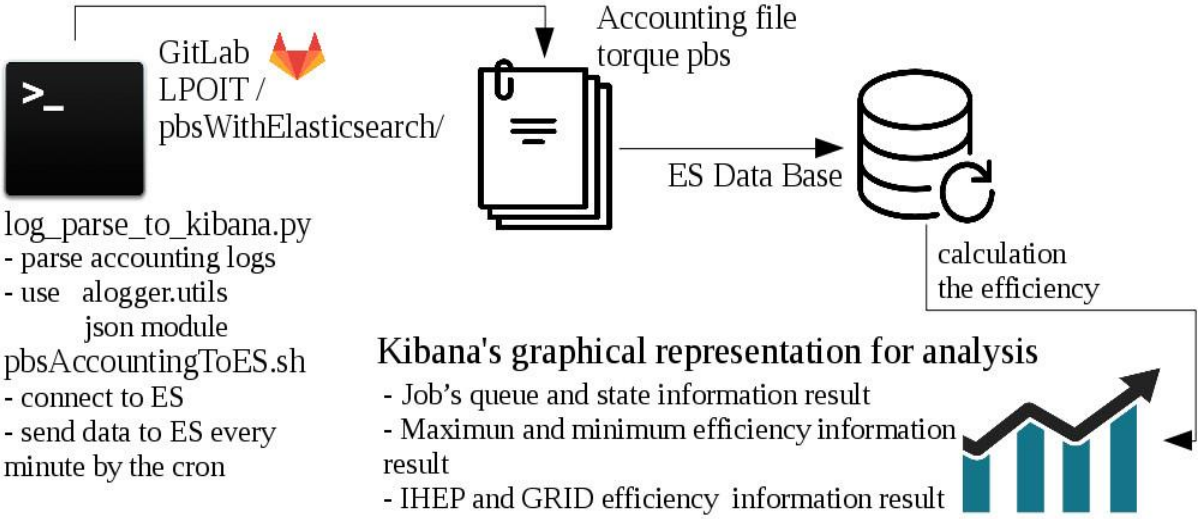


Figure 1. Overview of the efficiency measurement system

After the efficiency indicator is calculated it is used in visualizations. Here is the information for the analysis:

1) cput – maximum operating time of the processor or CPU time;
2) ncpus – how many cpus each allocated node must have;
3) ppn – how many processes to allocate per each node;
4) walltime – time of performance of a task in hours;
5) efficiency – result on the basis of the previous values (percentage format).

These data have been taken from the log pbs files [3]. The last indicator was calculated on the basis of all previous (Figure 2).

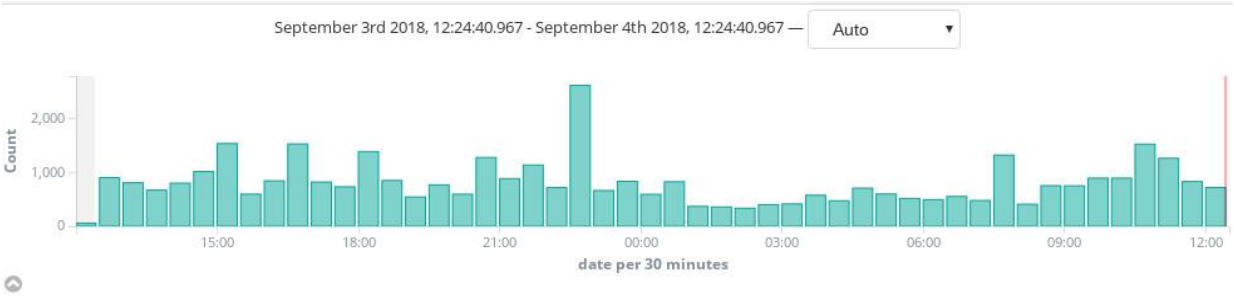
Script

```
doc['cput'].empty ? 0 : doc['ppn'].value ? ((doc['cput'].value/doc['walltime'].value)/doc['ppn'].value)
: doc['ncpus'].value ? ((doc['cput'].value/doc['walltime'].value)/doc['ncpus'].value)
: (doc['cput'].value/doc['walltime'].value)
```

Figure 2. Calculation of a new efficiency indicator in Kibana

Cput is the main reference point. If there is no such index it is set to default zero value. In order to calculate the efficiency index, cpu time is converted to the number type. The indicators ncpus and ppn were also taken into account and they are mutually exclusive. CPU time is divided into one of these values, if it presents. As a result, the calculated value is divided by the walltime of a task given by type number.

Kibana tools allow you to display this data in the table (Figure 3).



Time	cput	walltime	ppn	efficiency
September 4th 2018, 12:23:43.000	3,840	3,858	-	99.533%

Figure 3. Job's data information results

3. Kibana's graphical representation

Display of the data loaded into ES — it is work for Kibana. Kibana is an open source (Apache Licensed), browser based analytics and search dashboard for Elasticsearch.

Several graphs were constructed reflecting the growth or decrease in the efficiency of resources use by a group of tasks based on efficiency indicator. The first one shows a vertical bar graph that reflects the number of tasks in the execution queue depending on the type (Figure 4). You can see that more tasks are related to the ihep-short queue. These tasks typically perform fast processing of small amounts of data and are characterized by a large number of input /output operations.

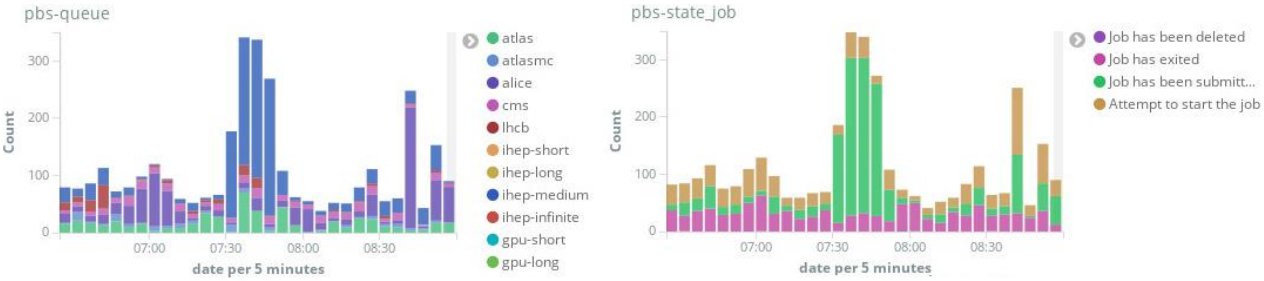


Figure 4. Job's queue and Job's state information results

Second shows the histogram of the statuses of tasks, the number of cancelled, interrupted, completed tasks (Figure 4). You can see that the overwhelming number of tasks are completed tasks, those that are currently executing and those that are in the execution queue.

Another two graphs show the effectiveness of the accomplishing tasks of IHEP users and grid tasks (Figure 5).

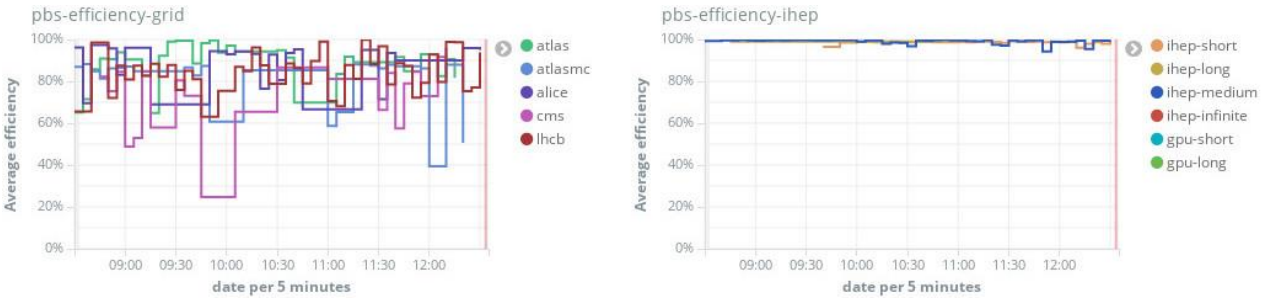


Figure 5. IHEP and GRID efficiency's information results

The last linear graph reflects several critical indicators at once: average efficiency level, maximum and minimum efficiency, Upper and Lower Standard Deviation of efficiency (Figure 6).

Figure 6. Maximum and minimum efficiency information results

4. Memory allocation systems

However, the memory usage of each process is bounded with Linux resources limitation. The memory management subsystem is one of the most important parts of the operating system. It was supposed to try different memory allocation systems on the computing nodes of the cluster and to check their effect on the efficiency and memory usage.

For an experiment the tcmalloc package was configured on the cluster working nodes for one week. The tc in tcmalloc stands for thread cache — the mechanism through which this particular allocator is able to satisfy certain (often most) allocations lucklessly [4].

During tcmalloc usage it is seen on the graph that a sharp decline of cpu usage occurred in compare with previous gradual usage. With regard to the efficiency of using the cluster, the most noticeable decline is visible for Atlas LHC experiment (Figure 7).

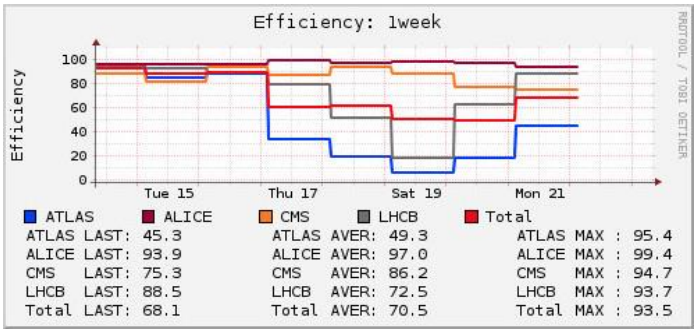


Figure 7. Tcmalloc - recession of efficiency

The tcmalloc allocator does not return memory to the operating system. As more threads are used, the advantage of using lock-free algorithms shows that tcmalloc drops behind. As a conclusion - tcmalloc is expedient to use for a certain type of tasks, scripts and for IHEP cluster it is unsuitable.

Jemalloc was tested for the same period (Figure 8). Jemalloc is a general purpose malloc implementation that emphasizes fragmentation avoidance and scalable concurrency support. Unlike

tcmalloc focused on separate tasks, this library is intended for use on the cluster for the whole stream of tasks [5].

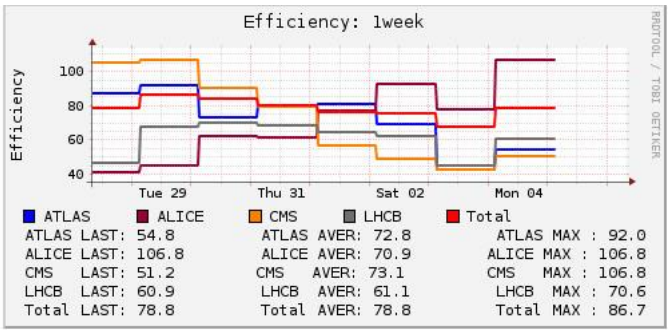


Figure 8. Jemalloc - recession of efficiency

After inclusion of libjemalloc.so.1 library there was a growth of overall performance of the cluster (in particular CPU utilization). But jobs processing has failed - more than a half of jobs and about 27% of multicore jobs were interrupted (Figure 9).

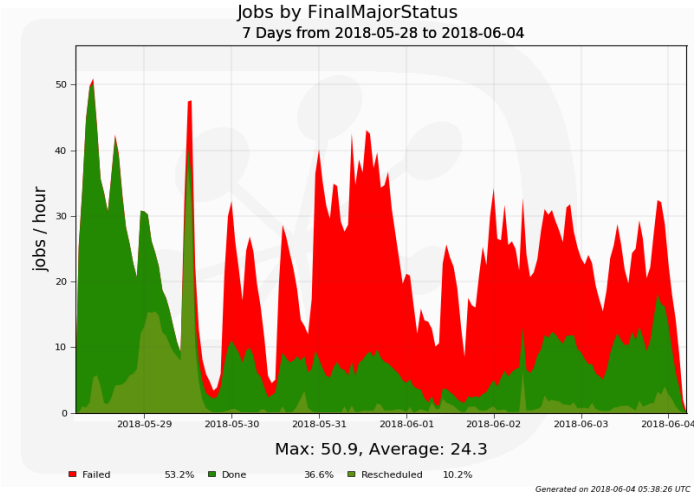


Figure 9. Jemalloc forced jobs to fail

That means tcmalloc and jemalloc are not useful for IHEP cluster.

5. Cluster Management System

In the future there is the plan to create additional control component for IHEP Cluster Management System (CMS) with objectives to analyse efficiency indicators of overall cluster performance and to manage the cluster in a way of improving resource usage efficiency. At this stage CMS consists of event-driven management system, configuration management system, monitoring and accounting system and a ChatOps technology which is used for the administration tasks.

6. Conclusions

In this article it was calculated the efficiency indicator, which based on the accounting file of torque pbs. Also the examples of efficiency's graphs were given. It will help to analyse efficiency of working cluster in the future.

For an experiment the memory allocators were tested. In our case the result of tests was not useful for IHEP cluster. We will continue to improve the efficiency of the computing cluster. As future works it is planned to use Cluster Management System (CMS) with objectives to analyse efficiency indicators.

References

[1] Ezhova V., Kotliar V. Accounting system for the computing cluster at IHEP // CEUR Workshop Proceedings. February 2017: Vol. 1787.- pp. 518-524
[2] Python library to parse resource manager logs [python aloger]. Available at: https://pypi.org/project/python-alogger/ (accessed 24.10.2018)

[3] TORQUE Resource Manager [Torque home page]. Available at: http://docs.adaptivecomputing.com/torque/3-0-5/4.1queueconfig.php (accessed 24.10.2018)

[4] TCMalloc : Thread-Caching Malloc [TCMalloc overview]. Available at: https://gperftools.github.io/gperftools/tcmalloc.html (accessed 24.10.2018)

[5] Jemalloc Documentation [jemalloc home page]. Available at: http://jemalloc.net/ (accessed 24.10.2018)