Report on Summer Student Program JINR-2015 (students.jinr.ru) summer program:

Slurm Simulator and IBM Watson Analytics for JINR Hybrilit Cluster Statistics Analysis

By **Olga S. Sedova** PhD student from Saint Petersburg State University (SPBU)

Under supervision of **Andrey V. Nechaevskiy** from the Laboratory of Information Technologies of Joint Institute for Nuclear Research (LIT JINR)

Dubna
2015

Table of Contents

About this document

This report intends to allow reproduction of work performed during the Summer Student Program 2015 at JINR, analysis and expansion of provided results.

Task

Inspect Slurm Simulator for potential use in Hybrilit cluster tasks load scheduling.

Setup

Given a Hybrilit cluster account, a VM for Slurm Simulator deployment, access to configured Slurm tasks log database. For cluster details please refer to official cluster site[1].

Slurm Simulator

Slurm Simulator was introduced in 2011 by Alejandro Lucero [1] as a tool for job trace execution using resource manager SLURM. By the use of Slurm Simulator it was shown that SLURM default settings are not always very effective for job scheduling and it is possible to change them for optimization of work with a particular cluster (see, for example recent articles [2, 3]). Therefore, it is of interest to run Slurm Simulator on Hybrilit cluster and find out whether SLURM manages Hybrilit resources optimally. And if not, what are possible strategies for SLURM optimization.

Getting ready with Slurm Simulator

We configured and installed Slurm Simulator following[2] manual. This included the following steps:

- SLURM compilation,
- MySQL setup,
- environment configuration,
- `users.sim` extraction on JINR cloud VM[3].

Moreover, Slurm Simulator required creation of links to the folder with the libraries:

```
ln -s /lib64/ /lib/x86_64-linux-gnu
```

and creation of two empty text files to be filled (mainly `sim_dir/sbin/rsv.trace` and `sim_dir/sbin/test.trace` files).

It should be noted that Slurm Simulator currently does not work with data on particular cluster specific job patterns.

[1] http://hybrilit.jinr.ru/
[2] https://github.com/SchedMD/slurm/blob/simulator/contribs/simulator/README
[3] http://cloud.jinr.ru

Slurm Simulator Session

We were running `slurmdbd -Dvv` daemon in UI mode, and simultaneously we started Slurm Simulator in separate window.

Slurm simulator was tested with 100, 1000, 100000, 1000000 and 3999999 tasks (fig. 1, 2). It was observed that even for a large number of tasks (3999999) SLURM needs no more than 0.7% of CPU resources for managing. So that SLURM (the default version of SLURM'15) was found to be effective for managing Hybrilit cluster.

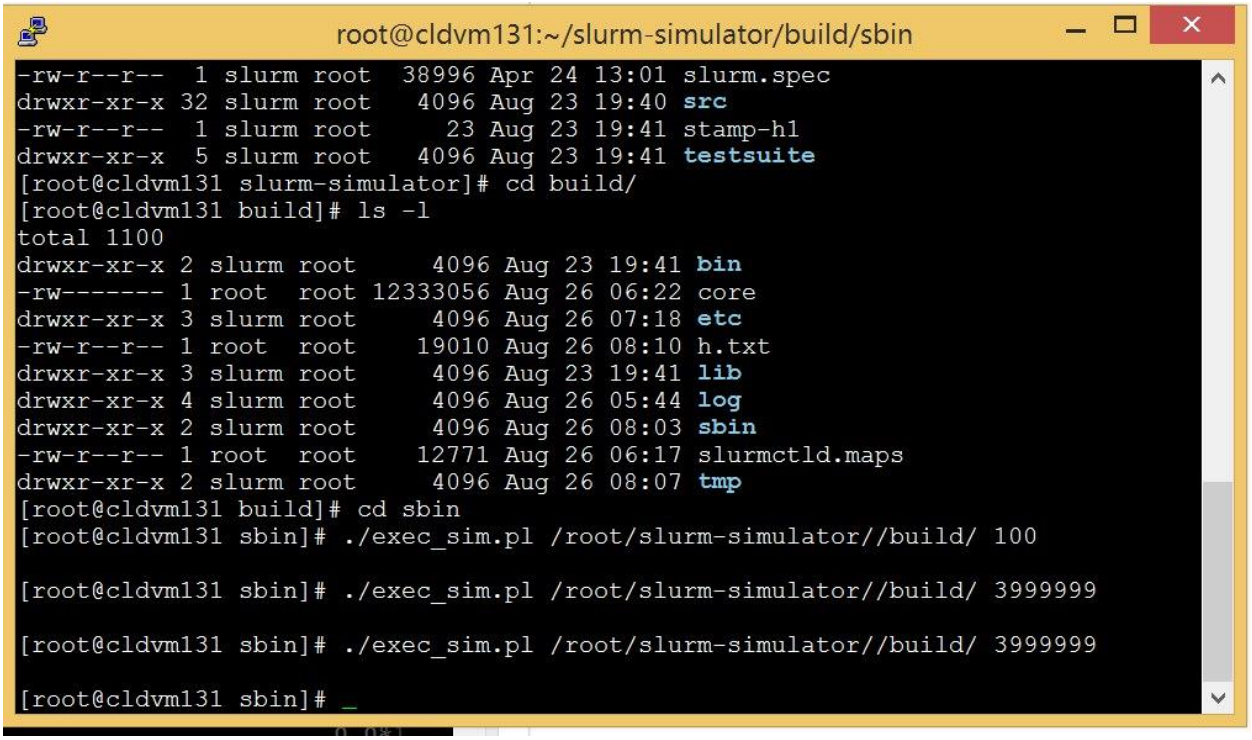

Fig. 1. An example of the Slurm Simulator test.

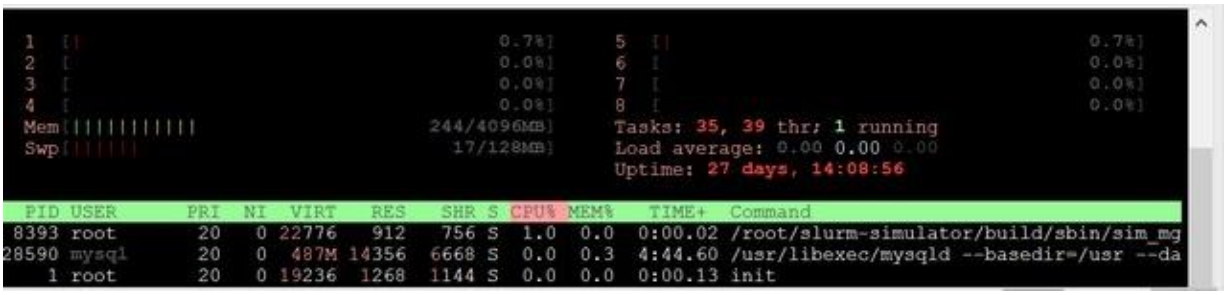

Fig. 2. Current cluster load during Slurm Simulator execution process.

Results of Slurm Simulator session

During the session, Slurm Simulator generates two main log files: `exec_sim.log` and `sim_mgr.log`.

The first of them contains information about simulation process (fig. 3).

```
Launching sim_mgr.../root/slurm-simulator//build//sbin/sim_mgr
Launching slurmctld.../root/slurm-simulator//build//sbin/exec_slurmctld.sh
Launching slurmd.../root/slurm-simulator//build//sbin/exec_slurmd.sh
0     0  9326     1  20   0   9200  1196 pipe w S+   pts/5      0:00 /bin/bash /root/slurm-
simulator//build//sbin/exec_slurmctld.sh /root/slurm-simulator//build/



Getting maps for process ID 9326
1     0  9327  9326  20   0   9200   636 wait   S+   pts/5      0:00 /bin/bash /root/slurm-
simulator//build//sbin/exec_slurmctld.sh /root/slurm-simulator//build/



Getting maps for process ID 9327
4     0  9328  9327  20   0 333492  3252 hrtime Sl+  pts/5      0:00 /root/slurm-
simulator//build//sbin/slurmctld -Dc 2



Getting maps for process ID 9328
Waiting...
Ok. We have 3999999 completed jobs
Killing simulation processes...
```

Fig. 3. Information about simulation process in `exec_sim.log` file.

The second file `sim_mgr.log` contains information about how many threads have been created for the allocation of tasks and which ones were the most loaded (fig. 4).

Current Hybrilit load analytics using IBM Watson Analytics

About the product

IBM Watson Analytics is a product that allows making data preparation, refinement, management, analysis and visualization easily. Options are automated and available from the cloud.

Working with Hybrilit statistics in IBM Watson Analytics

We have taken statistics on cluster usage as CSV-file using database

```
MySQL -uUserName --password=PassWord -h drift.jinr.ru -B -e "use slurm jobs; SELECT uid,
user name, jobid, name,starttime, endtime, partition, FROM UNIXTIME(starttime) AS os start time,
FROM UNIXTIME(endtime) AS os end time FROM jobcomp table LIMIT 10000000;"| sed
"s/'/\'/;s/\t/\",\"/g;s/^/\"/;s/$/\"/;s/\n//g" > slurm_stats.csv
```

And copied it into a local machine

```
scp osedova@hydra.jinr.ru:/nfs/hybrilit.jinr.ru/user/o/osedova/slurm_stats.csv ~/Documents/slurm_stats.csv
```

Using Watson we analyzed the statistics of running tasks on the Hybrilit cluster only for the first eight months of 2015 year.

By asking Watson a few questions on current cluster state, we created some visualizations.

The breakdown of the number of different jobs by cluster partition is shown in fig. 5. We can see from the figure, that the most loaded partitions of Hybrilit cluster are CPU and GPU.

```
Found sbatch program at /root/slurm-simulator//build//bin/sim_sbatch
Found scontrol program at /root/slurm-simulator//build//bin/scontrol
Can not open rpc_threads.info file
Initializing semaphores...
Trace initializarion done. Total trace records: 1
Inserting new reservation trace record for time 0
Trace initializarion done. Total trace records for reservations: 1
INFO: Creating _time_mgr thread
Leaving some time for slurm threads to be ready ...
Sync: waking up thread 0
Sync: waking up thread 32
SIM_MGR[4294966696][1440583660][408523]: Checking for 2 threads [0000000100000001], last_cycle
(created,exited): 0,0
Let's dump the shared memory contents
SIM_MGR[4294966696][1440583668][714459]: created,exited: 12,3
sleep_map_array:       ff00000007
thread_exit_array:               0
Total fast threads: 1
Total thread create counter: 12
Total thread exit counter: 3
Dumping thread data ...

Thread         pid            f address              sl/new/noend/join     creation
       last_sleep     last_wakeup    wait_mean
======         ===            =========              =================     ==========
       ===========    ========       ========
0              9332           0000000000000000              -1/0/0/0
       00000000       00000000       00000000       0
1              9332           0000000000426fcc              1/1/0/0               4294966696
       4294966696     00000000       0
2              9332           000000000042be00              1/1/0/0               4294966696
       4294966696     00000000       0
32             9328           0000000000000000              -1/0/0/0
       00000000       00000000       00000000       0
33             9328           00007f05b7cd40b3              5/1/0/0               4294966696
       4294966696     00000000       0
34             9328           00007f05b7cd4663              -1/1/0/1
       4294966696     00000000       00000000       0
35             9328           000000000057802d              -1/1/0/0
       4294966696     00000000       00000000       0
36             9328           00007f05b72c5b24              -1/1/0/0
       4294966696     00000000       00000000       0
37             9328           00000000004b199e              -1/1/0/0
       4294966696     00000000       00000000       0
38             9328           0000000000437ffa              -1/1/0/0
       4294966696     00000000       00000000       0
39             9328           0000000000437dac              -1/1/0/0
       4294966696     00000000       00000000       0
40             9328           0000000000438764              -1/1/0/0
       4294966696     00000000       00000000       0
41             9328           0000000000438764              -1/1/0/0
       4294966696     00000000       00000000       0
Killing slurmctld and slurmd with SIGSEGV
```

Fig. 4. File `sim_mgr.log` Shows how many threads have been created for the allocation of tasks and which ones were the most loaded.

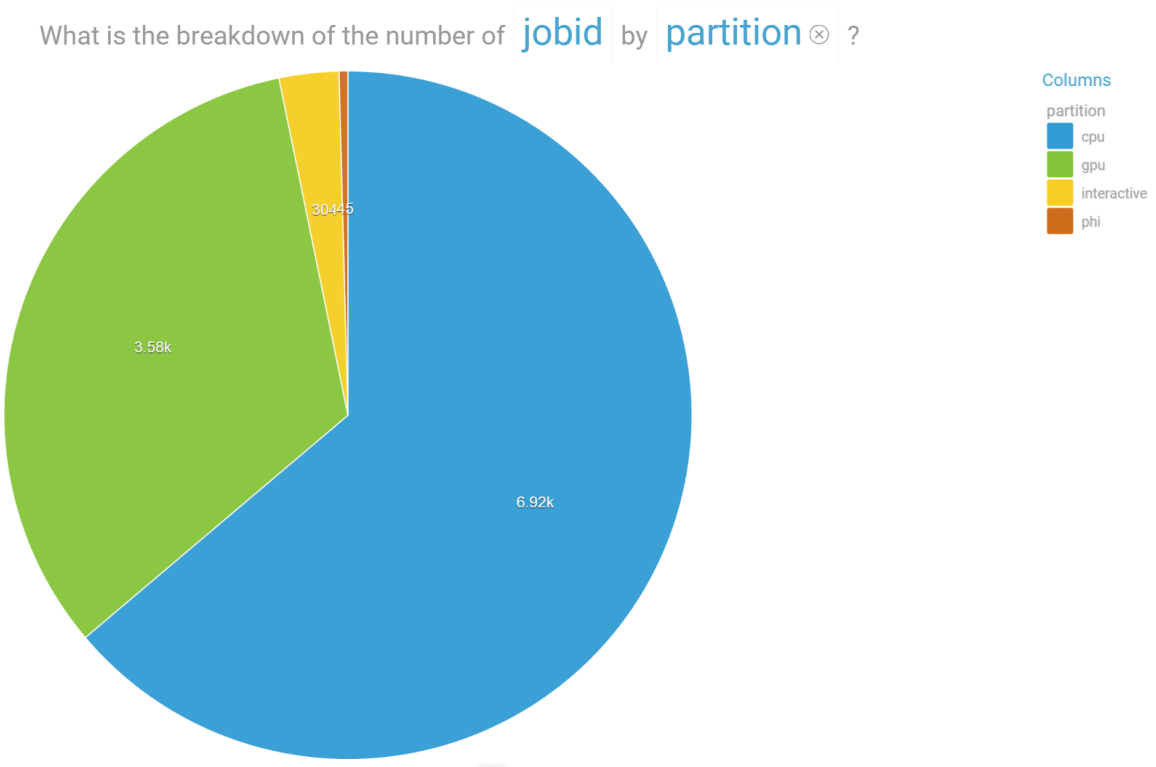

Fig. 5. Breakdown of the number of different jobs by partition

The number of different jobs that have been performed on each of CPU, GPU, Interactive and Phi partition over month is illustrated in fig. 6.

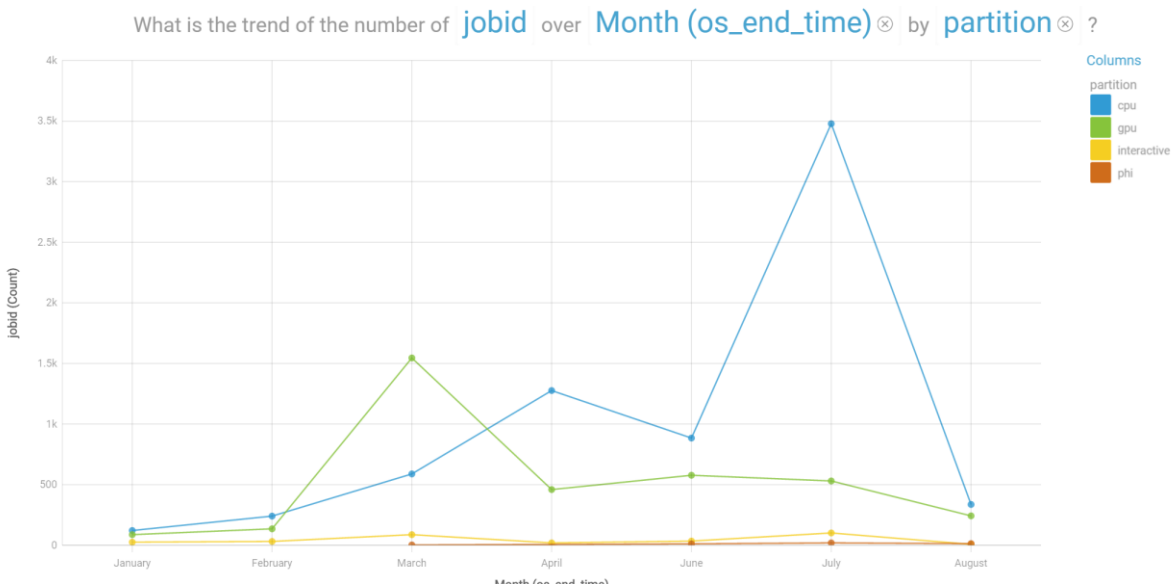

Fig. 6. Number of different jobs over month by partition

The duration of different jobs that have been performed on each of CPU, GPU, Interactive and Phi partition over month is shown in fig. 7.

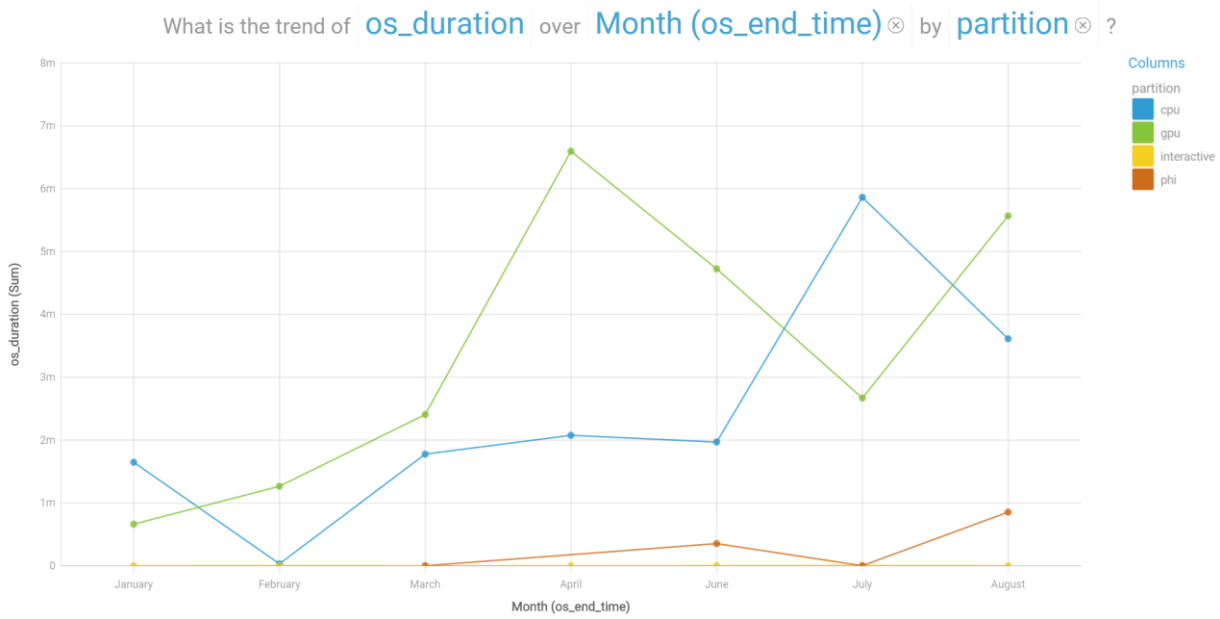

Fig. 7. Duration of jobs over month by partition

Note that there currently are 4 GPU nodes, one Mix node and one Phi node. A 100% load for one node per 1 month would take ~2678400 seconds, and 86400 seconds per day.

Finally, the duration of jobs over month for whole Hybrilit cluster is shown in fig. 8. It is seen from the figure, that there is a trend that would lead to entire cluster load quite soon.

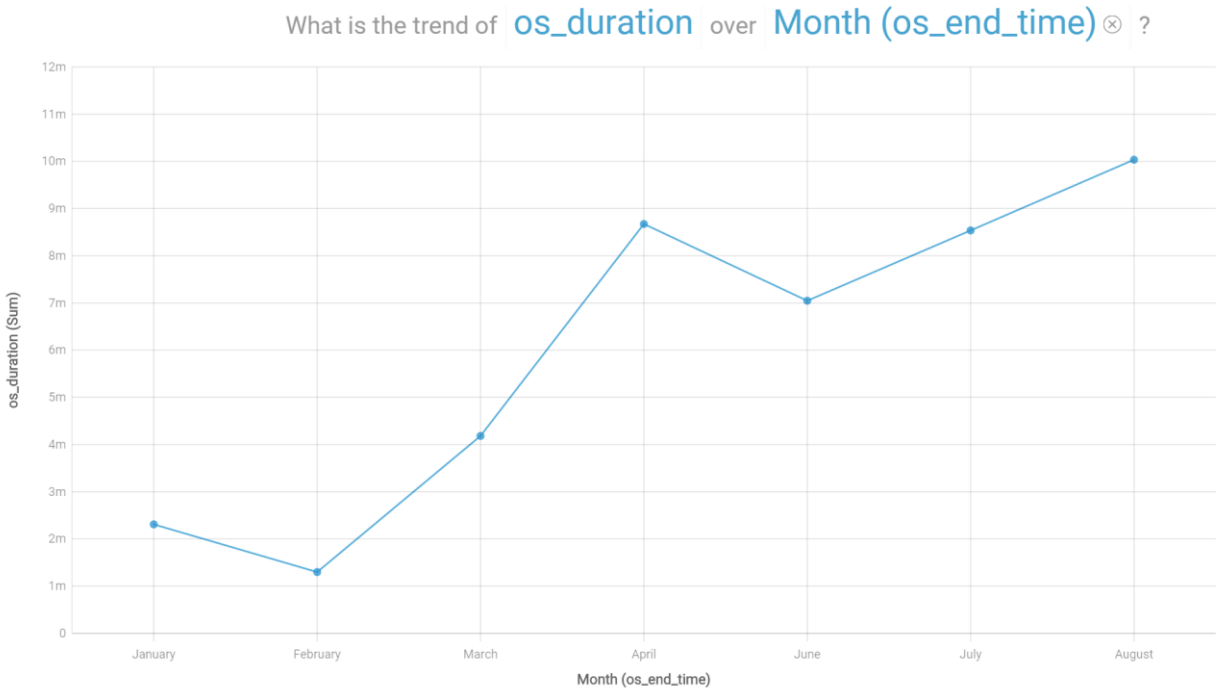

Fig. 8. Duration of jobs over month for whole Hybrilit cluster

Conclusion

During the Summer Student Program 2015 at JINR, a VM was configured to run Slurm Simulator. It was found out that main Slurm Simulator output is essential for Slurm development and extension but is not very helpful for our particular case. We look forward to further work with Slurm community. Statistics of running tasks on the Hybrilit cluster was analyzed using IBM Watson Analytics. Some visualizations were created. Experience gained with IBM Watson Analytics can be useful in further understanding of Hybrilit data.

References

1. Lucero, Alejandro. Simulation of batch scheduling using real production-ready software tools. Proceedings of the 5th IBERGRID, 2011.
2. S. N. Leonenkov. Expanding the functionality of SLURM resource manager. XVI International Supercomputer Conference «Scientific service in the Internet: A variety of supercomputing worlds», 9/26/2014.
3. Wang, Ke, Xiaobing Zhou, Hao Chen, Michael Lang, and Ioan Raicu. "Next generation job management systems for extreme-scale ensemble computing." In Proceedings of the 23rd international symposium on High-performance parallel and distributed computing, pp. 111-114. ACM, 2014.