Managing large scale computing for MPD experiment. [*]

Krzysztof Dygnarowicz[a†], Jakub Zieliński[b], Łukasz Sawicki[b]

[a]The Faculty of Electronics and Information Technology Warsaw University of Technology,
[b]Faculty of Physics Warsaw University of Technology

As the part of the TeFeNICA we received tasks associated with physical analysis, large scale computing and data management. The main goal of our project was to create scripts in bash or programs in C/C++ and generate events which could be used to analyse. Next step was launching our work on other clusters. Article shortly describes what tools can be used to achieve this targets.

PACS numbers: 07.05.Kf

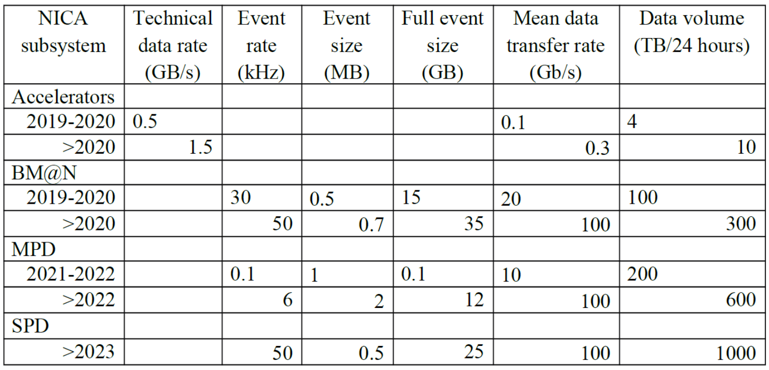
1. Introduction

One of the challenges we must face during NICA project is amount of data we want to collect, process and storage. Data should be accessible to all members of collaboration and allow to make physical analyses. With many clusters available, we have lots of different Linux distributions (CentOS, Ubuntu), queue system (SLURM, Sun Grid Engine) or same software but in different versions which we use during computing. At some point we must unify it and gave a tool to run tasks independently from platform.

2. Big Data problem

During creating, technical design of NICA network and computing infrastructure at JINR , there had been estimated volume of data which will be created during work of complex.

[*] Presented at the IV NICA Days 2019 Conference associated with the IV Slow Control Warsaw 2019, Warsaw, Poland, October 21-25 2019



NICA subsystem	Technical data rate (GB/s)	Event rate (kHz)	Event size (MB)	Full event size (GB)	Mean data transfer rate (Gb/s)	Data volume (TB/24 hours)
Accelerators						
2019-2020	0.5				0.1	4
>2020	1.5				0.3	10
BM@N						
2019-2020		30	0.5	15	20	100
>2020		50	0.7	35	100	300
MPD						
2021-2022		0.1	1	0.1	10	200
>2022		6	2	12	100	600
SPD						
>2023		50	0.5	25	100	1000

Fig. 1. Table showing plans in amount of data produced by NICA [1]

Assumed 1PB of data in 2023 can be only achieved with usage of proper tools which should be tested first 1. In this moment we have software version of MPD detector, called MPDRoot, and can simulate impacts and reconstruction of events of colliding gold ions. This software can be used now to generate large amount of data.
We are doing this now because we want:

- To find MPDRoot errors

  During physical analyse of this data, model can be easily verified and show errors.

- To make reconstruction programs

  When MPD will start to work we should have some basic reconstruction programs.

- To know what to implement in FPGA

  Some of the analyse, could show us which data is most important and where we should improve frequency or accuracy of measurements.

3. Producing data

For this task our group created script to automate job with following outline:

1. Get information of current JOBID in queue system, use it to name folder
2. Link generator of event (particles collision) and MPDRoot
3. Generate data (runqmd.bash for URQMD generator)

4. Run them by detector (runMC.c)
5. Reconstruct event from detector data (reco.c)
6. Copy data to safe storage
7. Erase unnecessary data

Script firstly was running on NICA cluster [2] which at the beginning had limitations of 200 jobs per user. Next step is to perform script on other clusters like HybryLIT. For this it had been modified to use with SLURM job system. The basic commands are described in instruction [4]. Further improvements can be achieved with grid computing.

Supercomputers can be connected in grid where they share their resources. Example of this system is RDIG (Russian Data Intensive Grid)[3] which to manage jobs using DIRAC. "DIRAC is software framework for distributed computing providing a complete solution to one (or more) user community requiring access to distributed resources." [5] It allows to integrate different clusters, manage hardware and run tasks on them. Tests show that our first approach to problem will not work in this case.

4. Conclusion

During work we proposed system which uses JOBID to identify output of jobs. After this we decide that name of the output of the file should give full information of used generator, energy, centrality, number of events etc.

This pattern is currently used on HybryLIT cluster by Andrey Moshkin:

- **vHLLE-UrQMD_AuAu_11.5GeV-06.6-10.4fm_200ev_1.root**

During generation on HybryLIT and NICA cluster it has been estimated that one job should work for 200 events to optimize usage of CPU time and RAM storage. Moreover clusters have limitation, for example one job could not take more than one day or access to some storage spaces can be done only indirectly through the working script. DIRAC at this point did not had standardized generators (URQMD,vHLLE) and user need to install them on his own.

The instructions we prepared should be filled with basic workflow pattern and information of available modules. Next step will be receiving feedback from people, who will use our instructions.

Links to programs and instructions:

https://github.com/jzielins97/NICA
https://github.com/Lukaszz99/work
https://gitlab.com/kdygnaro/tefenica_mpd_lit

REFERENCES

[1] http://mpd.jinr.ru/wp-content/uploads/2018/06/NICA_computing_TDR_1.03.pdfAccess 15.03.2020

[2] http://mpd.jinr.ru/howto-work-with-nica-cluster/Access 15.03.2020

[3] http://ca.grid.kiae.ru/RDIG/Access 15.03.2020

[4] http://hlit.jinr.ru/en/user_guide_eng/#_3enAccess 15.03.2020

[5] http://diracgrid.org/Access 15.03.2020