Minutes of the ABP Computing Working Group meeting

20th July 2017

Participants: D. Amorim, V. Baggiolini, H. Bartosik, E. Belli, X. Buffat, R. De Maria, P. Dijkstal, A. Falabella, M. Hostettler, G. Iadarola, N. Karastathis, A. Lasheen, K. Li, A. Mereghetti, M. Migliorati, A. Oeftiger, Y. Papaphilippou, D. Pellegrini, A.C. Pesah, A. Romano, G. Rumolo, R. Scrivens, G. Sterbini, M. Titze,C. A. Valerio Lizarraga

G. Rumolo asked for feedback on the HTCondor usage.

1. M. Titze is experiencing issues when running MAD-X job. Based on the log file it seems that several attempts to connect to a machine are performed. Eventually the job start properly, but it seems that the multiple connection attempts take longer the that the running of the job. Also, some jobs sporadically fail due to lack of credential. H. Bartosik said that he didn't experience such issue, but he is using 8 core whereas M. Titze is using 4.
2. C. A. Valerio Lizarraga said that it would be convenient to be able to run jobs from EOS.
3. R. De Maria reported that the scheduler was not available at time, which was usually due to a pending upgarde of the system. Also, some job sporadically disappeared from the queues. This effect disappeared after an upgrade, but it was not explained. Jobs with significant I/O to eos are often unstable, this occurs in few cases when interacting with AFS. He avoided the issue by avoiding direct I/O outside of the compute node and rather use HTCondor features to copy the results to another support once the job is finished. K. Li mentioned that, when using such features, the monitoring of the running jobs'evolution becomes impossbile. H. Bartosik mentioned that one can use ssh to connect to the compute node on which the job is running to monitor it. K. Li added that the data of a crashed job is lost in such a configuration.
4. X. Buffat reported that the submission of jobs within a job is no longer possible with HTCondor. The use of DAG is however working well.
5. R. De Maria suggested to gather these observations and solutions on the ABP-CWG wiki page.
6. After the meeting, A. Oeftiger added that a small fraction of his jobs on a sufficiently long queue where killed at Wall time.

A. Falabella presented the HPC cluster at CNAF. The cluster is composed of 384 CPUs at 2.1GHz with hyperthreading enabled. The scheduler is based on LSF. gcc, icc, ifort, python 2.7, OpenMPI and mvapich2 are installed. Other software can be installed upon request. He gave instructions on how to apply for an account on the cluster. The users from ABP are asked to send the request to the working group administrators (abp-cwg-admin@cern.ch), who will forward it to the administrator of the cluster.
The access is done through a front end, that shouldn't be used to perform computationally intensive tasks. The jobs should be submitted to the hpc_acc queue, they will be scheduled according to the fairshare policy. The maximum duration of a job is currently 21 days, but can be reviewed according to the needs of ABP. The default queue is acc_short and is limited to 6h.

The list of nodes needs to be explicitly given in the submission of the jobs to the scheduler. If the user does not have a preference on which node to use, the full list has to be specified.
A. Falabella asked whether a processor reservation policy with backfilling would be useful. This policy prevents single (few) core jobs to fill the available resources delaying multicore jobs singificantly. Nevertheless, the idling cores are used to run jobs of sufficiently short duration. This feature can be efficient if the users specify the length of the jobs. G. Iadarolla mentioned that they already encountered an issue of this type and the solution proposed would solve it. A. Falabella proposed to implement a separate queue, in which processor reservation would be enabled.
Further request can be addressed to the support.
The cluster will be extended with few node equipped with general purpose GPUs by the end of the year.

R. De Maria asked whether it would be possible to have access to a computer to perform the data processing, avoiding large data transfer to CERN. A. Oeftiger said that this is already possible by requesting an interactive session on the compute nodes.

G. Iadarola and H. Bartosik thanked A. Falabella for the support in the early phase of the machine operation. H. Bartosik mentioned that he did not need to use the machine file and the list of node. A. Falabella said that it is probably a feature of OpenMPI, whereas H. Bartosik used mvapich2.

R. De Maria presented PyTimber and PyLSA. Currently the hardware of the accelerator complex is mainly controlled through FESA classes (C++). They are interfaced with Java for control / operation (LSA, CALS, JAPC). The tools presented here are Python interfaces to the control system meant to facilitate the data analysis.
PyTimber is used to access accelerator data logged in the CALS (CERN Accelerator Logging Service) database. This software is used widely within ABP and in other groups (BI,PH), representing around 100 users. Examples of high level classes used for specific analysis on different equipments are available, allowing for fast learning and ease of use.
The development of PyTimber is shared between ABP and BI. CO is rather focused on the development of NXCals, a new technology that will become available in production by LS2. After which it is possible that PyTimber becomes a layer on top of NXCals with the same API, or it could be discontinued if the new technologies provide the necessary features. V. Baggiolini mentioned that NXCals is based on a different paradigm, where the data is not loaded locally, but rather the algorithm is send to be executed on computers where the data is stored. Consequently, it might not be straight forward to create an API working as PyTimber.
PyLSA is a Python interface to LSA (LHC Software Architecture). The API is not stable yet, nevertheless it is already possible to retrieve accelerator settings (read-only). V. Baggiolini mentioned that CO would like to keep control on the locations where the Java code are running, in order to avoid issues during upgrades. The development of PyLSA is shared between ABP, CO and OP.

G. Iadarola asked about an efficient way to handle the different data types available from the CALS database (e.g. vectornumeric with varying size). R. De Maria developed a database (pagestore) which allows to handle this type of data in a reasonably efficient way. Most standard ways have drawbacks in terms of efficiency or rigidity.

R. Scrivens suggested to extend these tools to the injectors. R. De Maria said that any help is most welcome to try an implement the necessary adjustments.