# Minutes of the ABP Computing Working Group meeting

# 3 May 2018

**Participants:** L. Barraud, H. Bartosik, X.Buffat, L. Deniau, S. Furuseth, P. Heckmann, N. Hoymir, G. Iadarola, K. Iliakis, J. Komppula, K. Li, C. Lindqvist, P. Llopis, J. Molson, N. Mounet, A. Petrenko, T. Polzin, H. Rafique, J. Repond, T. Richard, O.Rios, G. Rumolo, L. Sabato, H. Saberi, A. Sublet.

This is an extended meeting to discuss experience with the HPC batch cluster operated by IT using the SLURM system. Users of the cluster from other departments and groups also joined the meeting.

## HPC clusters at CERN

C. Lindqvist, P. Llopis, and N. Hoimyr presented the status of the CERN HPC service. Slides can be found here.

- The HPC cluster is conceived for applications and use cases that do not fit the standard batch High Throughput Computing (HTC) model (batch system operated with HTCondor). These are typically parallel MPI applications.
- Computations that do not have these requirements should run on the standard batch service. The user community covers different departments (BE, TH, HSE, TE, EN).
- The HPC service is presently made of:
  - **The HPC Batch Cluster** made of 78 nodes (16 core / 128Gb) with low latency 10Gb ethernet.
  - Two **HPC Infiniband clusters** each made of 72 nodes with 20 cores (40 with HT) equipped with 10Gb Ethernet and Infiniband "Fat tree".
- Data on the usage of the cluster in 2018 is presented. The main observations are:
  - There are about 20 active users since cluster launched last year;
  - The activity of BE/ABP often within a single host (16 cores); these kind of jobs should run on HTCondor, for this purpose it would be nice to have supported MPI installations on the standard batch system.
  - Long running multi-host jobs are mainly launched by BE/RF (Gdifdl), TE/VCS (Plasma studies) and HSE/SEE (Fire simulations)
  - The shorter queue (batch-short) is at times under-used;
  - Mvapich2 seems the most stable MPI implementation;
  - Overall we have a stable running cluster.
- A survey among the users was conducted in 2017. The main outcomes were:
  - Beam simulation software used by BE is typically installed and setup within the team;
  - A large fraction of the software is open source, except commercial engineering applications;
  - Engineering software (CFD, Field Calculations etc) distributed by IT or supplier;
  - Only part of the user community are familiar with batch systems and MPI software setup, and with detailed computing requirements.

- Access to the cluster is granted via e-groups. Users who have been granted access can login to the head node via ssh. They can compile code on the head node using MPI if needed and submit jobs using SLURM (using srun, sbatch or salloc). One of the available MPI installations can be selected via "module load".
- The clusters are equipped with fast local storage and have access to AFS and EOS.
- The IT HPC team is presently working on:
  - Job priorities (fairshare and accounting groups);
  - Backfill of idle resources with regular batch work;
  - Tuning of applications when sources available;
  - Improvements based on user feedback.
- The IT team is very interested in having feedback from the user community (e.g. which MPI flavours and versions are used, are applications more CPU, Network, I/O bound). They are interested in providing support to improve the application efficiency and for this purpose they are interested in having benchmark cases and templates from the users in order to be able to perform tests in realistic conditions.

## HPC cluster usage and perspectives

Different users presented their experience and future needs (slides can be found here):

- **COMBI simulations** for the study of transverse coherent effects in the presence of beam-beam interactions:
  - One process is allocated for each simulated bunch. Each MPI process with a heavy CPU load can spawn threads (OpenMP);
  - Studies run so far considered only one bunch per beam and could be run on a single node. Soon the studies will be extended to multi-bunch effects and this will require using multiple nodes;
  - The users acknowledged the fast reaction in supporting installation and debugging;
  - Output data transfer from hpcscratch often becomes a bottleneck. It was suggested to use eoscp instead of cp.
- **PyHEADTAIL simulations** for the study of transverse coupled-bunch effects in the presence of impedances and transverse feedback:
  - In general the HPC cluster works well for these applications;
  - Issue in the MPI h5py limits the maximum number of processor to one node;
  - Support would be appreciated in tuning the simulation setup.
- **PyORBIT simulations** to study space-charge effects:
  - The software has been installed very recently on the cluster;
  - The users would like to have a local project space to share data and software instead of using AFS/EOS;
  - At the moment scaling with number of CPUs looks quite poor, this is being investigated;
  - The expected load is of about 25 jobs/week each using about 40 cores. After an exploration period of 1-2 months this could increase.

- **Electron Cloud simulations**. For these applications parallel computing is used for e-cloud instability simulations. The tools are developed in-house (PyECLOUD, PyHEADTAIL, PyPARIS):
  - Presently these studies are running in a different facility (collaboration CERN/INFN-CNAF).
  - Typical load: 500 CPU-cores, running most of the time, typical job occupancy: 4-16 CPU-cores, RAM: 1 GB/core, typical simulation duration: about 2 weeks, chopped down in jobs of about 4h (chained jobs, useful when cluster reliability is not great);
  - Multibunch e-cloud instability simulations will be coming soon. The code is under development, first test runs should be coming in summer, production only later. For these applications the single simulation will be much heavier: 100-1000 CPU cores, for 1-2 weeks;
  - The simulations are split in smaller runs for checkpointing purposes.
- **BLonD simulations** to study longitudinal beam dynamics:
  - BLonD is a python code with C++ extensions (based on ctypes);
  - mpi4py to co-ordinate Python processes, calling C++ libraries, parallelized with OpenMP;
  - These features are at an early stage of the development. In the immediate future resources will be required only for development, testing and optimization. For these purposes jobs will be launched requiring up to 40 CPU cores.