Minutes of the ABP Computing Working Group meeting 1

15th September 2016

Participants: D. Amorim, H. Bartosik, E. Belli, X. Buffat, N.Biancacci, L. Carver, R. Costa, L. Deniau, A. Latina, K. Li, L. Mether, G. Rumolo, D. Schulte, D. Pellegrini, M. Serluca, P. Skowronski.

Arising matters

- Members of the ABP-CWG are reminded to update the wiki pages related to the computing tools that they are responsible for.
- The next meeting will be held on 15 September and will be focused on AFS and LSF phase-out, with the contribution of experts from the IT department.
- With this meeting we start a review activity of the beam physics tools that are developed, maintained or heavily used in the ABP group. In particular, Xavier will give a presentation covering the COMBI code.

The COMBI code

Xavier's slides are available here.

- COMBI stands for COherent Multibunch Beam-beam Interaction.
- The code simulates interactions of two counter rotating beams. Several "actions" can be included in the simulation: linear transport, head-on and long-range beam-beam interactions, noise sources, collimators, impedances, linear detuning, transverse feedback, synctrotron radiation.
- At CERN, and in particular in ABP, COMBI is used to simulate instability mechanisms involving colliding beams, to compute beam transverse functions that can be directly compared with machine observations, and to study emittance effects like the impact of noise with colliding beams.
- The head-on beam-beam interactions are traditionally simulated using and Hybrid Fast Multiple Method. However the present implementation is quite old and, in spite of being fast and robust for most of the cases, it is very difficult to maintain and would require serious refactoring to be parallelizable. Moreover it shows a saturation of the noise level that is not compatible with the present needs.
- Recently several alternatives to the HFMM were explored and the most attractive was found to be the implementation of a Fast Polar Poisson Solver. This proved to be faster and more precise than the HFMM. Moreover it is efficiently parallelized with OpenMP.
- Long-range beam-beam interactions are simulated using a soft-gaussian 4D model.
- Impedances are simulated following the approach implemented in HEADTAIL. The different bunches are sliced and kicks are applied to the particles using the charge and the position of the different slices and wake functions describing the impedances of in the accelerator.
- Synchrotron radiation can be simulated using the full implementation or a Gaussian noise model.
- The oldest routines are written in F77, the newer features are implemented in C++. The wrapping and the first level of parallelization based on MPI are written in C. A second level of parallelization is based on OpenMP.

- Using the second level or parallelization requires an MPI installation with the level of thread support MPI_THREAD_FUNNELED. The FPPS requires the FFTW3 library.
- The first layer of parallelization uses a master/slave paradigm. The master dispatches the actions to the slaves, each one associated to a different bunch. Some of the actions (e.g. beam-beam interactions) requires communication with a partner bunch.
- The performance scales as expected with respect to the number of bunches, nevertheless actions requiring heavy processing can act as bottlenecks.
- Unfortunately MPI idling processes are busy-waiting, this forbids a simple re-allocation of the resources of the waiting processes and therefore limits the performance of the second layer of parallelization. This can be overcome controlling the priorities at the OS level (implemented in COMBI) but this requires privileges that are usually not granted on a shared cluster.
- Another implementation of the second level of parallelization based on MPI and POSIX shared memory interface was also attempted but it was discontinued due to the low performance gain with respect to the complications in the implementation.
- Many studies are run on local machines (usually 4 to 12 cores). For most studies the code performs well, but requires proper parallel infrastructure, as well as manpower for setting up, testing and optimisation on each architecture.
- Preferred resources are: multiple nodes with fast connection, regular memory and possibly several cores per node.
- EPFL infrastructures (accessible on request for EPFL Ph D. students) offer such capabilities. Most results obtained with COMBI where based on their facilities but this are not generally available to other ABP members. The potential of the current CERN infrastructures needs to be investigated.
- Concerning the present status and prospects: the present implementation already allows fulfilling many of the needs. For accurate emittance studies and for the simulation of slow instabilities studies there is a need for more powerful computing resources (large number of CPUs per node, possibly large number of nodes for multibunch studies). The effort towards the implementation of a 6D simulation just restarted with the test and parallelization of the 6D soft-Gaussian solver.
- Apart from the FFTW3, needed to use the FPPS solver, no other external library is required.
- It is definitely possible to increase the modularity to interface the code with other tools (e.g. PyHEADTAIL). In this direction the FPPS solver was Python-wrapped and included in the PyPIC library.