ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS045

HIGH-PERFORMANCE SCHEDULING OF MULTI-BEAM MULTI-BUNCH SIMULATIONS *

S. V. Furuseth†[1], X. Buffat, CERN, 1211 Geneva 23, Switzerland
[1]also at EPFL, 1015 Lausanne, Switzerland

Abstract

Coherent multi-bunch interactions through beam-beam forces or wakefields can cause severe impacts on the beams in circular colliders, if not well understood and countered. COMBI is a parallel multiparticle tracking code developed to study such interactions. Its implementation greatly limits its efficiency when considering realistic configurations featuring effects with different computational requirements, such as the multi-bunch interaction through wakefields, beam-beam interactions, transverse feedback and lattice nonlinearities. A new parallel scheduling method, pipelining the effects for each bunch, has greatly sped up the code. The new version of the code, COMBIp, is presented here.

INTRODUCTION

The dynamics of particle beams in circular colliders can be so involved that the most complex configurations must be evaluated by numerical simulations. Each beam consists of multiple bunches. The bunches are affected by both external electromagnetic fields produced by the machine, and interactions with each other. There are: (i) independent, intra-bunch effects such as the forces from the various magnets; (ii) intra-beam, inter-bunch interactions such as the kicks from electromagnetic wakefields [2]; (iii) inter-beam, inter-bunch interactions close to the collision points, called beam-beam interactions [3]. The modeling of the different effects will be referred to as calculations.

There exists a wide library of simulation codes developed specifically for circular particle colliders, exploiting the parallel infrastructure of modern computers in different ways. The parallelization in COMBI was designed to deal with multi-beam, multi-bunch simulations, and it can study both beam-beam interactions and wakefields [4, 5].

When simulating coherent multi-beam, multi-bunch effects, causality puts strong constraints on the order of the calculations. Based on the analysis of the challenges, a new parallel algorithm has been implemented in COMBI, named COMBIp. The performance of the new and old algorithms will be analyzed and compared.

CAUSALITY CAUSED CHALLENGES

In a circular collider, two beams move in opposite directions, as shown in Fig. 1. The locations must be traversed in order, meaning $19, 0, 1, \ldots$ for a bunch in beam 1 (B1), and $1, 0, 19, \ldots$ for a bunch in beam 2 (B2), to ensure causality. At each location there can be a calculation to be performed, or not, with various implications on the scheduling.

Independent, intra-bunch calculations require no communication between the bunches. These calculations have to be done in order for each bunch separately, to ensure causality. However, in a multi-bunch simulation, these calculations do not require any synchronization between the bunches.

Intra-beam, inter-bunch calculations, from now on referred to as intra-beam calculations, require communication between bunches in the same beam. Here we consider wakefields. In the ultra-relativistic limit, particles can only affect trailing particles. How the inter-bunch dependencies affect the order of these calculations is visualized in Fig. 2a. The wakefields produced by each bunch can be calculated independently and simultaneously for every bunch. Then the wakefields must be communicated to the trailing bunches, before the kicks from the wakefields can be calculated. Hence, a bunch cannot overtake another bunch beyond this calculation. Without inter-beam calculations, the individual bunches can easily be parallelized, by performing each calculation simultaneously for every bunch.

Inter-beam calculations are typically referred to as beam-beam interactions. In modern circular colliders, the two beams are kept separated except for close to the interaction points. How the inter-bunch dependencies affect the order of these calculations is visualized in Fig. 2b, assuming one long-range calculation on each side of a head-on calculation at location 0. Bunch n has to calculate its interaction with bunch $n-1$, then n, then $n+1$ of the other beam. As this is required for both beams, bunch $n-1$ has to finish its calculation with a bunch of the other beam, before bunch n can start its calculation with the same bunch. Without intra-beam calculations, or filling of every slot in the collider model, the bunches can still be parallelized efficiently by doing different calculations simultaneously.

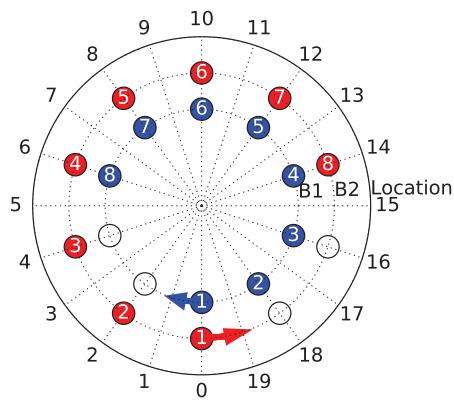
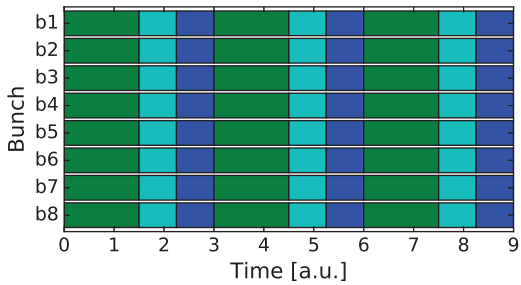


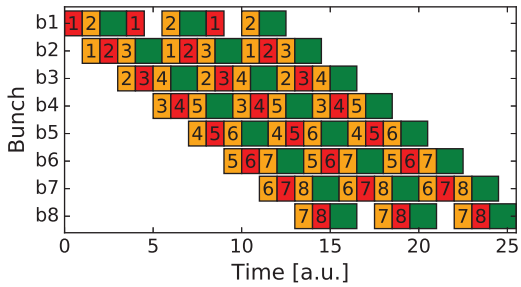
Figure 1: A circular collider model where two beams (B1 in blue, B2 in red) move in opposite directions. Both beams have 8 bunches in a row followed by two empty slots.

* Full study submitted to Computer Physics Communications [1].
† sondre.vik.furuseth@cern.ch

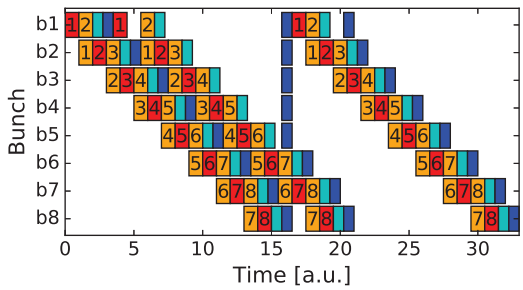
ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS045



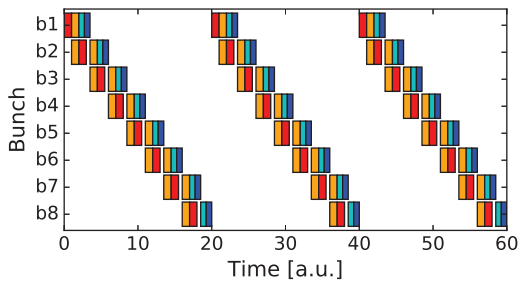
(a) Independent and intra-beam calculations.



(b) Independent and inter-beam calculations.



(c) Intra-beam and inter-beam calculations.



(d) Intra-beam and inter-beam calculations in COMBI.

Figure 2: Gantt charts [6] of the most efficient flow of calculations for 8 bunches (b1-b8) of one beam over three turns, while still ensuring causality in (a)-(c). The order of the calculations for each bunch is read left to right. A white gap signifies that the next calculation cannot yet be initiated, because it requires input from another bunch. Green is an independent calculation, Light blue is the calculation of that bunch's wakefields, Blue is the impact of other bunches' wakefields, Red is the head-on beam-beam calculation, and Orange is a long-range beam-beam calculation. The numbers on the beam-beam calculation blocks refer to which bunch of the other beam the calculation is with, assuming the head-on calculation is in location 0 in Fig. 1. The times of the different calculations are set artificially to be comparable. The equivalent flow of (c) in COMBI is shown in (d).

The bottleneck, preventing these parallel multi-beam, multi-bunch simulations from being efficient, arises when one includes both intra-beam and inter-beam calculations in the same simulation, as visualized in Fig. 2c. These calculations have different communication requirements. In result, there is a sizeable amount of white space corresponding to time when no calculation can be done for a given bunch. The main culprit in this model is that the kick from wakefields on bunch 1 (b1) in turn 2, cannot be calculated before bunch 8 has been there in turn 1 to produce its wakefield.

PARALLELIZATION

To combat the challenges laid out above, we have developed a new parallel algorithm. It has been implemented in COMBI, meaning that the physics is modeled exactly as before. COMBI is already implemented with a hybrid OpenMP-MPI master-worker parallelization, with one bunch per worker and one master overall [7, 8]. The bunches on each beam are fixed to a circular grid as in Fig. 1, and rotated synchronously to their next location in the collider model. In effect, all workers have to wait for the slowest calculation to finish, before the bunch grids are rotated to their next locations. The equivalent Gantt charts of Figs. 2a-c will include more white gaps between the calculations, as in Fig. 2d. The maximum times will increase from 9 in Fig. 2a to 40.5, from 25.5 in Fig. 2b to 60, and from 33 in Fig. 2c to 60 in Fig. 2d.

The advances of the new parallelization algorithm, COMBIp, are detailed in [1]. They can be summarized in 4 key points: (i) the bunches are autonomous; (ii) all calculations for each bunch are put in separate pipelines; (iii) the communication between the bunches is asynchronous; (iv) there can be multiple bunches per process. Due to the first 3 advances, the Gantt charts for the processes are the most efficient ones shown in Figs. 2a-c. Since there also can be multiple bunches per process, a process does not have to stall when a bunch have to wait. By pairing b1 and b5, b2 and b6, etc. in Fig. 2c on 4 processes, the white pauses of one bunch are filled with the calculations of the other bunch, improving the efficiency of the simulation.

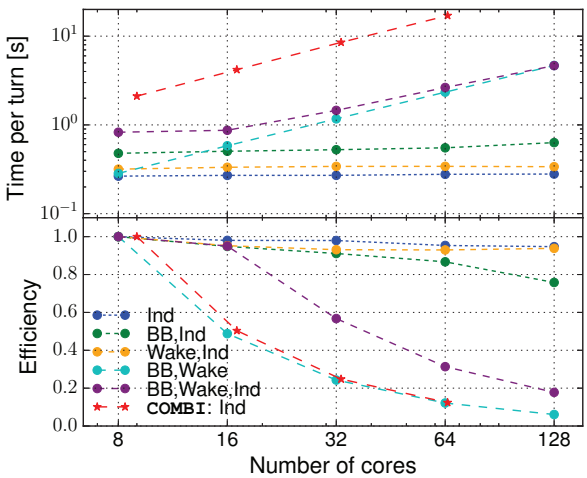
TIMING RESULTS

The parallel algorithms have been tested in detail for their performance in relevant configurations. The timings have been done on the Deneb cluster at EPFL, with nodes containing 2 Ivy Bridge processors running at 2.6 GHz, with 8 cores each [9]. All simulations have been run with 10^6 particles per bunch, over 100 turns. They were run 4 times, whereupon the average wall time was calculated. The weak scaling is presented here, the strong scaling is detailed in [1]. The weak scaling of an algorithm is how the wall time varies with the number of cores, N_{cores}, for a fixed problem size per core [10]. For weak scaling, the goal is to prevent the wall time from increasing. The efficiency is calculated as $t_{\text{ref}}/t_{\text{par}}$, where t_{ref} is the time of a small reference simulation with 1 or a few cores and t_{par} is the time of the parallel simulation for which we want to know the efficiency.

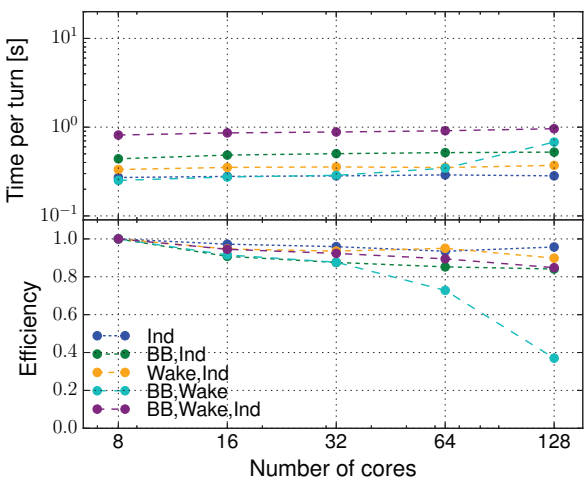
ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS045

To test the weak scaling, the bunch sizes were kept constant, while the total number of bunches were equal to the number of cores, subtracted 1 for the master process in COMBI. The weak scaling of COMBI and COMBIp is presented in Fig. 3. Five different collider models have been tested with the new algorithm. The models consisted of various combinations of the following: (i) Linear phase advance including chromaticity, independent for each bunch ("Ind"); (ii) A section with one head-on calculation and one long-range calculation on each side ("BB"); (iii) A wakefield calculation ("Wake"). The models with beam-beam calculations had $N_{\text{cores}}/2$ bunches per beam, while the others had N_{cores} bunches in the first beam only. As the beam-beam calculations do not make sense for one bunch, the scalings start at a reference simulation with 8 bunches.

The weak scaling with 1 bunch per 1-core-process is presented in Fig. 3a. The efficiency of the COMBI algorithm falls quickly as the number of bunches increases, due to the synchronization after each calculation, exemplified in Fig. 2d. Note that the COMBI simulations are run with only the independent calculation, and should be compared to the blue curve labeled "Ind", which is simulated with COMBIp. The bottleneck for collider models with both intra-beam and inter-beam calculations is clearly visible on the curves labeled "BB,Wake" and "BB,Wake,Ind". The efficiency of the model corresponding to the curve labeled "BB,Ind" falls off starting from 64 cores. This can partly be explained by the white triangles at the beginning and end of Fig. 2b, because only 100 turns are simulated. The work required by the wakefield calculations scales with the number of bunches. This scaling is negligible up to 128 bunches.



(a) 1 bunch per 1-core-process.



(b) 8 bunches per 8-core-process.

Figure 3: Weak scaling for different collider models, with independent (Ind), beam-beam (BB) and wakefield (Wake) calculations. The weak scaling was calculated for COMBI in the simplest configuration with 1 bunch per worker process.

The weak scaling with 8 bunches sharing each 8-core-process is presented in Fig. 3b. There is a clear improvement by having multiple bunches on the same process in COMBIp. The root limitation, previously discussed qualitatively and shown here quantitatively, is how many bunches causality allows to be calculated in parallel. The solution is therefore to distribute more bunches on each process, such that each calculation takes a shorter wall time, instead of trying to calculate every bunch in parallel. By achieving better load balancing in this manner, the bottleneck is shifted from limiting the number of bunches to limiting the number of processes. Thus, the achievable speedup is higher in simulations of the LHC with up to 2808 bunches per beam [11].

CONCLUSION

The constraints due to causality in multi-beam, multi-bunch simulations have been discussed in this paper. Simulations with either intra-beam or inter-beam calculations can be performed efficiently. In simulations including both of them, causality leads to a bottleneck of how many bunches can be calculated in parallel.

A new parallel algorithm has been implemented in COMBI, named COMBIp, to improve the efficiency. The key points of the new algorithm are that each bunch is autonomous, its calculations are ordered in a pipeline, the required communication between the bunches is performed asynchronously, and there can be multiple bunches per process. The new algorithm has achieved a speedup of up to the number of bunches per beam, compared to the previous algorithm implemented in the code. The performance is close to independent of causality constraints when simulating collider models with either intra-beam calculations or inter-beam calculations in the ultra-relativistic limit. The predicted bottleneck for collider models with both is now a limit on the number of compute nodes that can be used efficiently, instead of a limit on the number of bunches that can be simulated efficiently. The new algorithm is designed to efficiently simulate realistic models of the LHC.

ACKNOWLEDGEMENTS

The authors would like to thank K. Li and M. Moan for fruitful discussions. The calculations have been performed using the facilities of the Scientific IT and Application Support Center of EPFL, Switzerland.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-WEPTS045

REFERENCES

[1] S. V. Furuseth and X. Buffat, "Parallel High-Performance Multi-beam Multi-bunch Simulations", submitted for publication.

[2] M. Dohlus and R. Wanzenberg, "An introduction to wake fields and impedances", *CERN Yellow Reports: School Proceedings*, vol. 3, 2017, doi:10.23730/CYRSP-2017-003.15

[3] W. Herr and T. Pieloni, "Beam-beam effects", in *Proc. CERN Accelerator School*, Trondheim, Norway, Aug. 2013, pp. 431–459, doi:10.5170/CERN-2014-009.431

[4] COMBI, https://twiki.cern.ch/twiki/bin/view/ABPComputing/COMBI

[5] F. W. Jones, W. Herr, and T. Pieloni, "Parallel beam-beam simulation incorporating multiple bunches and multiple interaction regions", in *Particle Accelerator Conference 2007*, Geneva, Switzerland, Jul. 2007, doi:10.1109/PAC.2007.4440383

[6] W. Clark, W. N. Polakov, and F. W. Trabold, *The Gantt chart, a working tool of management*. New York, NY, USA: The Ronald press company, 1922.

[7] X. Buffat, "Transverse beams stability studies at the Large Hadron Collider", Ph.D. thesis, Inst. of Physics, École polytechnique fédérale de Lausanne, Lausanne, Switzerland, 2015, CERN-THESIS-2014-246.

[8] T. Pieloni, "A study of beam-beam effects in hadron colliders with a large number of bunches", Ph.D. thesis, Inst. of Physics, École polytechnique fédérale de Lausanne, Lausanne, Switzerland, 2008, CERN-THESIS-2010-056.

[9] Deneb, https://scitas.epfl.ch/hardware/deneb/

[10] P. S. Pacheco, *An Introduction to Parallel Programming*. Amsterdam, Netherlands: Elsevier Inc., 2011.

[11] O. S. Brüning *et al.* (eds.), "LHC Design Report", CERN, Geneva, Switzerland, Rep. CERN-2004-003-V-1, 2004.