Simulating high-energy heavy-ion collisions in the cloud

Gábor Bíró[1,2], Gergely Gábor Barnaföldi[2], Gábor Papp[1], Péter Lévai[2], Miklós Gyulassy[2,3,4,5], Xin-Nian Wang[4,5], Ben-Wei Zhang[5]

[1] Eötvös Loránd University, 1/A Pázmány Péter walkway, 1117 Budapest, Hungary
[2] Wigner Research Centre for Physics., 29-33 Konkoly-Thege Miklós street, 1121, Budapest, Hungary
[3] Lawrence Berkeley National Laboratory, Berkeley, California 94720 USA
[4] Columbia University, New York, NY 10027, USA
[5] Central China Normal University, 152 Luoyu Rd. Wuhan, P. R. China

biro.gabor@wigner.mta.hu; hijing@winger.mta.hu

Academia – Industry Matching Event

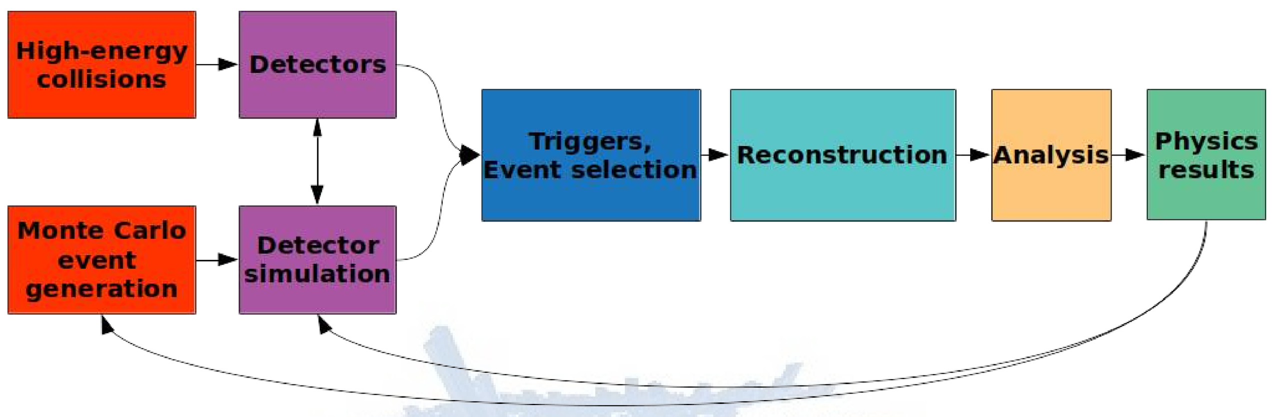
Artificial Intelligence, Machine Learning Workshop

Introduction

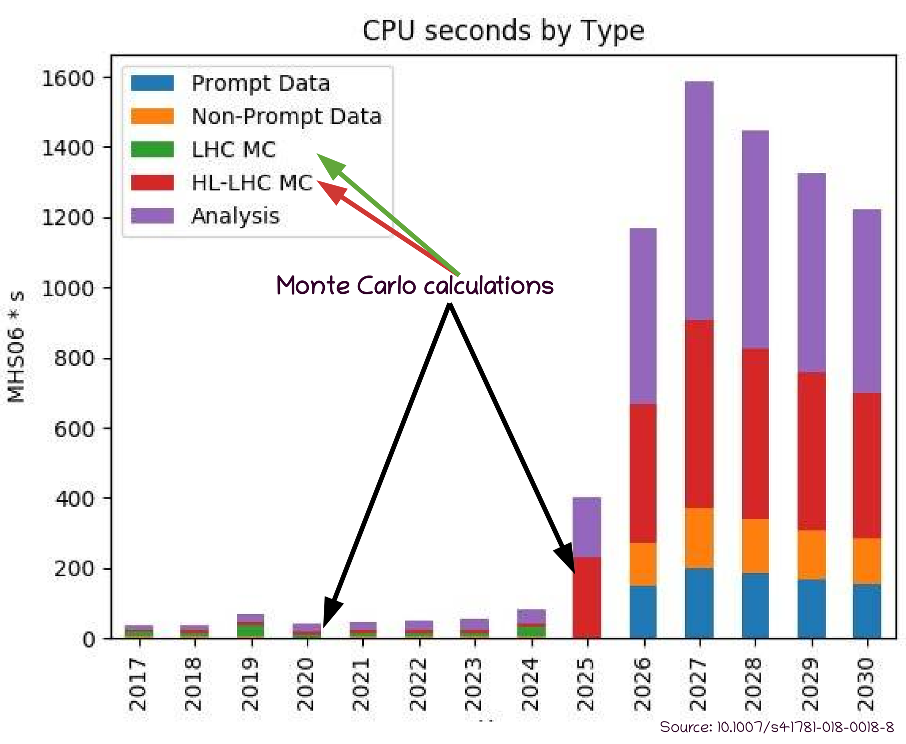
At the world largest particle accelerators such as the Large Hadron Collider at CERN or the Relativistic Heavy Ion Collider at BNL hundreds of thousands of interesting interactions may occur in every second. A special subset of these events are the high-energy heavy-ion collisions, aiming to investigate the birth of the Universe itself. These experimental measurements are always accompanied by numerical calculations, such as Monte Carlo event generators. However, these calculations are computationally very intensive: even with a state-of-the-art desktop machine many CPU hours (days, weeks sometimes) is needed to simulate only a few seconds of experimental data. Additionally, with the future improvements of the LHC it will be an even bigger challenge to catch up computationally.

Computing in HEP

In High Energy Physics the investigated phenomena are performed in a statistical basis. The numerical calculations should follow this behaviour. As a consequence, each simulated collision event should repeated many times.



The difficulty is in the difference between the rate of the real and simulated events: for the simulation of a central lead-lead collision the necessary computational time for a single CPU core is several order of magnitude larger, than in reality. Additionally, with the forthcoming upgrade of the LHC and the detectors this difference will even increase.

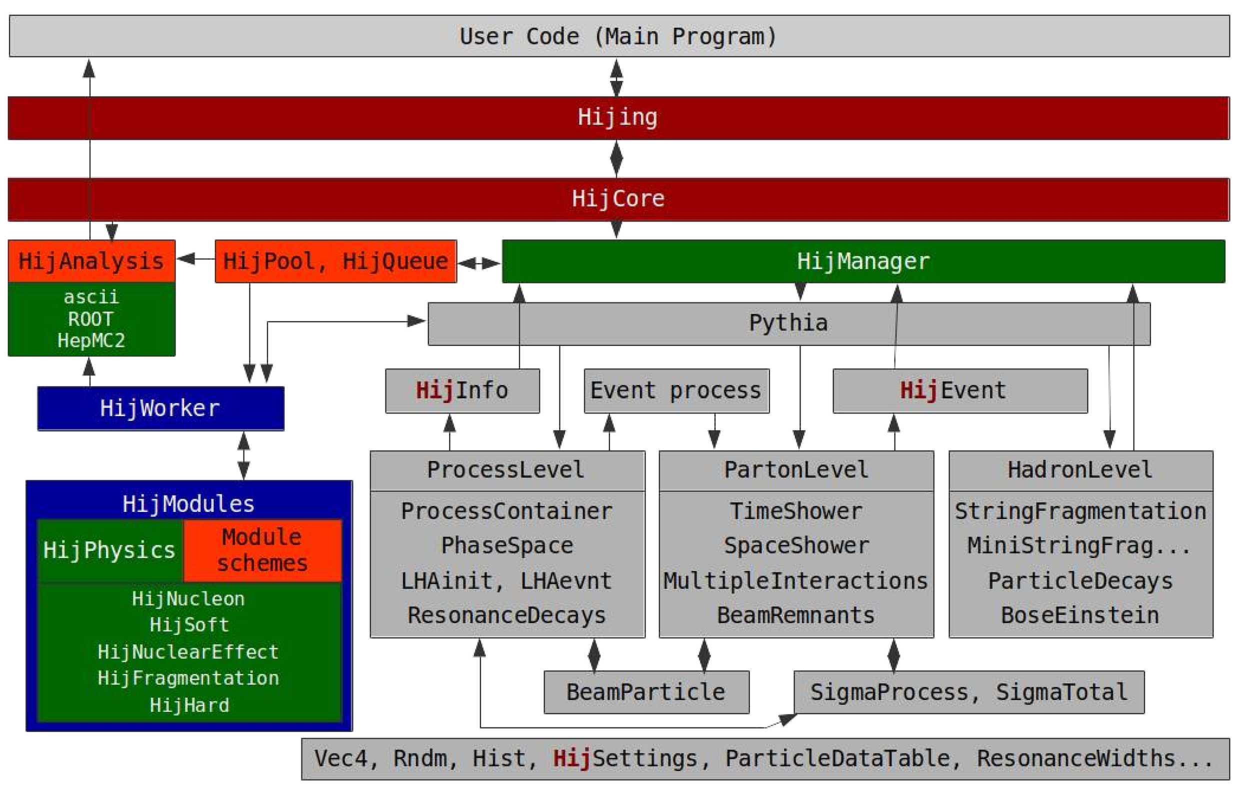


Although the hardware components become more and more sophisticated each year, today the performance gain in the calculations is not straightforward. In order to reduce the computational time and the additional costs, state-of-the-art HPC solutions are needed.

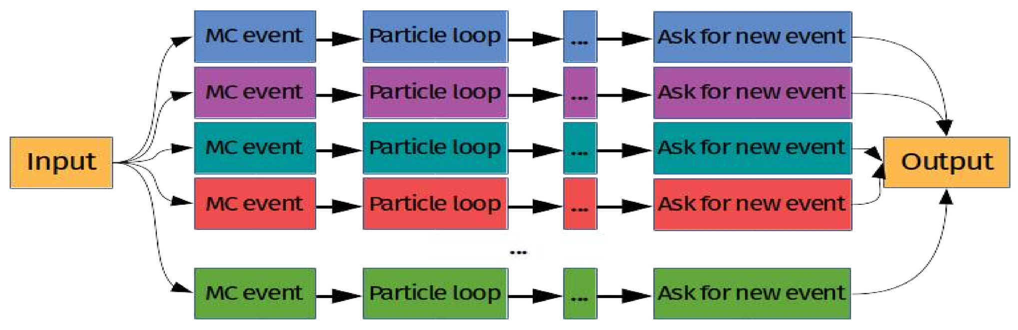
HIJING++

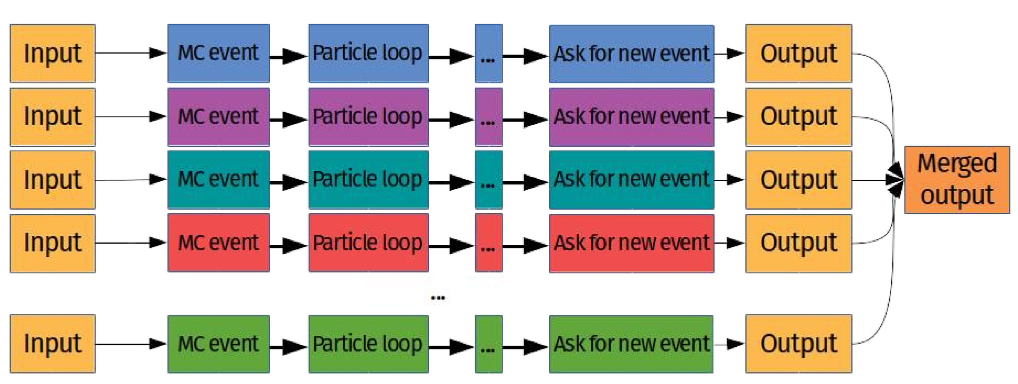
The **HIJING++** framework is the next generation of high-energy heavy-ion Monte Carlo event generators. Equipped with the latest theoretical models, it is designed to perform precise calculations in a flexible, fast, CPU parallel way.

With a modular structure, using modern C++ features, it is aimed to serve the high-luminosity era of HEP, with the capability to extend, improve and adapt.



Since the collision events are independent from each other, different schemes of parallelism are possible. The difficulty is with the merging of results.





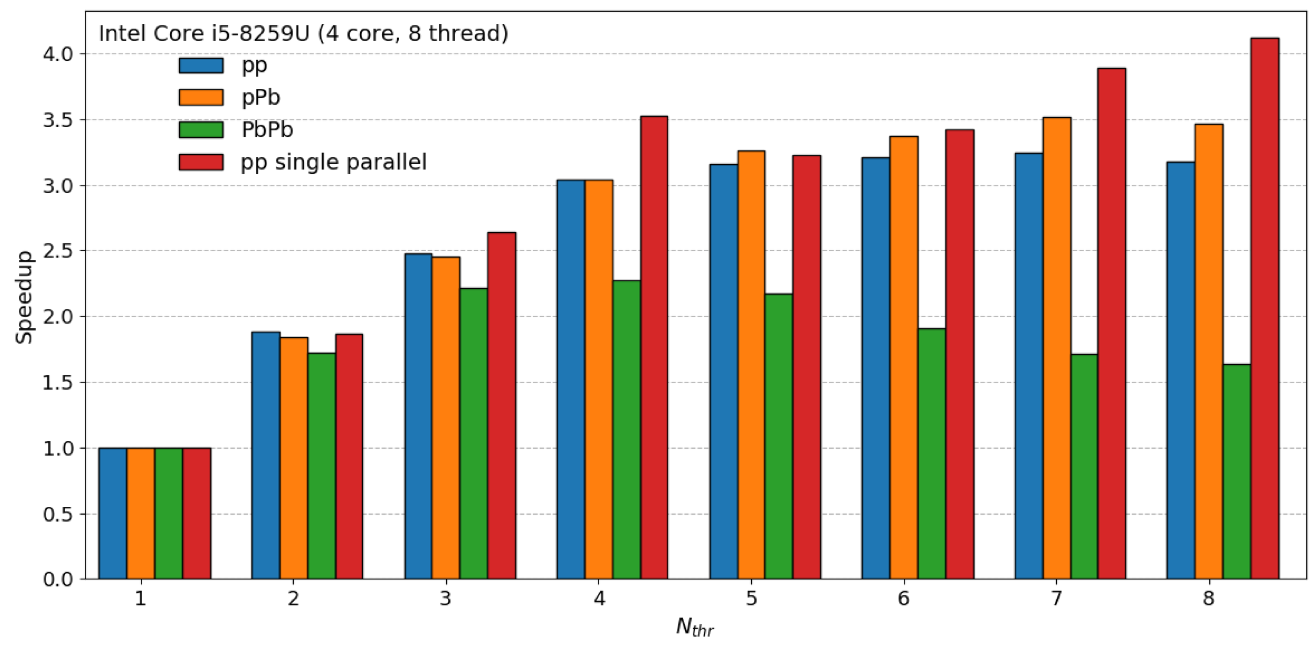
HIJING++ is able to utilize the multiple CPU cores of the machine and takes care of the event merging and the optional post-processing. The challenge is to optimize the job distribution in a variable environment in order to avoid bottlenecks.

Using multicore architectures in distributed cloud infrastructures, even higher speedup can be achieved.

Performance tests

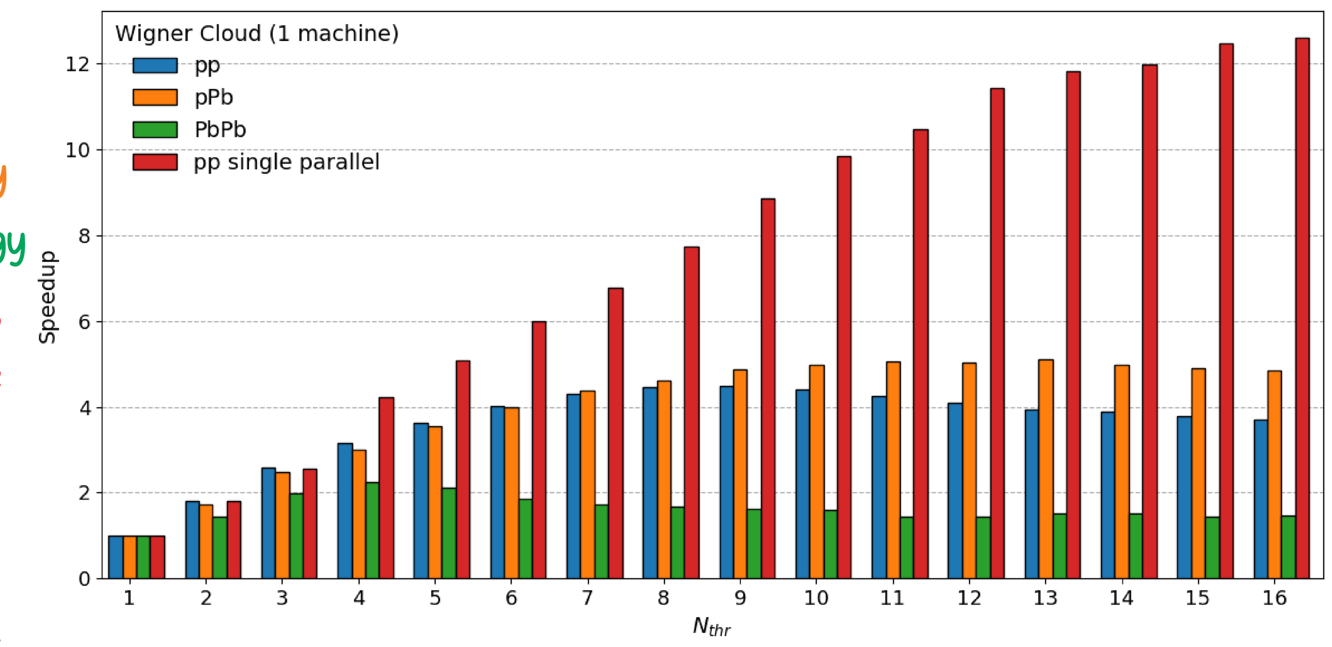
In this current study we investigated the performance of HIJING++ with 3 different setups that are common in scientific work:

- Intel Core i5-8259U (top figure)
- Intel Xeon based machine at Wigner Data Center (middle figure)
- 6 machine cluster at Wigner Data Center, launched several instances simultaneously in 3 thread mode (bottom figure)



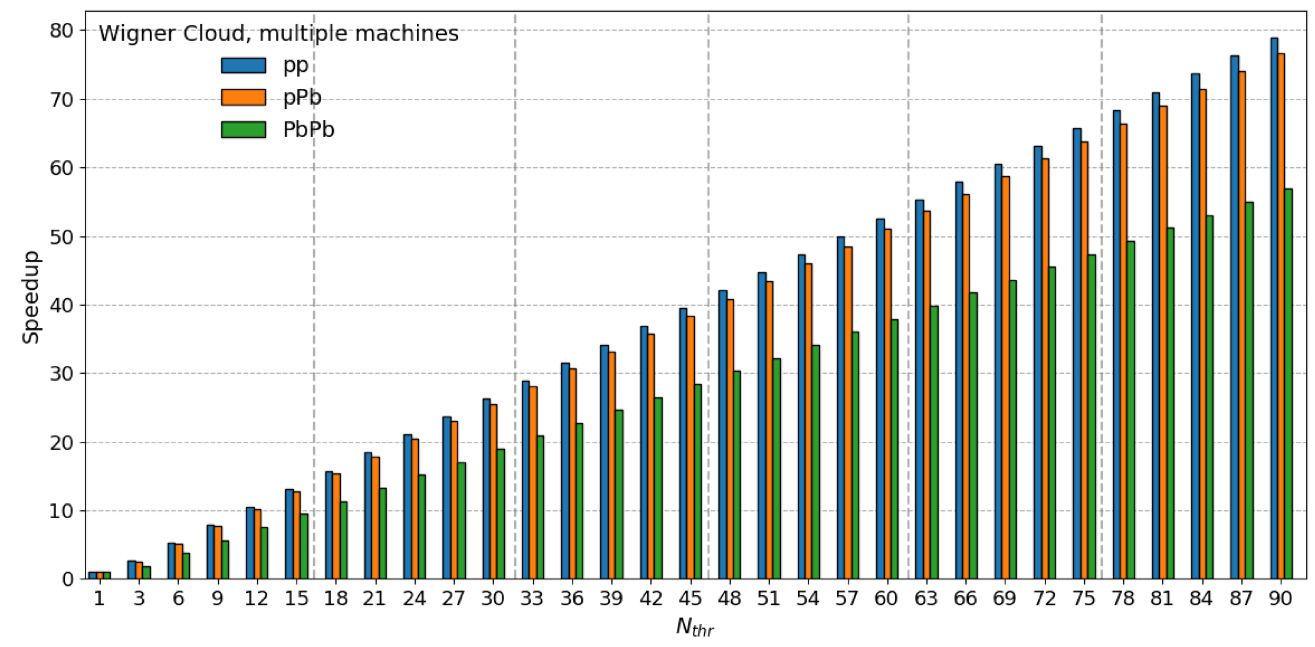
The investigated physical systems with 7 different analysis:

- Proton-proton collisions at 5.02 TeV CM energy
- Min. bias proton-lead collisions at 5.02 TeV CM energy
- Most central lead-lead collisions at 5.02 TeV CM energy
- Proton-proton collisions at 5.02 TeV CM energy, launched several instances simultaneously in single thread mode



The main observations are:

- On the Core i5-8259U machine the scaling is not trivial due to the sophisticated power management of the CPU
- The scaling in the largest lead-lead system breaks down quickly due to possible bottlenecks
- The scaling with the **HIJING++** thread management breaks down around N_{thr}~3-4
- Using multiple 3 thread instances, the speedup on the Wigner Cloud Machines scales linearly



There are several considerations to make:

- For collecting data, the **HIJING++** thread management creates a separate histogram for each thread and merges them with given frequency, which is a possible source for a bottleneck in the analysis interface, especially at large thread number
- The *"run several instance simultaneously and merge the results afterwards"* type of parallelization gives room for normalization errors
- With the most complex, most central lead-lead collisions the achievable maximum speedup seems to be lower - possible reasons:
  - Race conditions at the random number generation and/or histogram filling occur
  - Cache faults (the typical size of such an event is ~100-1000 times larger than in proton-lead collisions)
  - Further optimization of the scheduler is necessary

Summary

- HIJING++ is going to be the next generation of high-energy heavy-ion Monte Carlo event generators designed to be maintainable and extensible in a long term
- The modular structure provides room for improvement
- The built-in scheduler and analysis interface provides an easy usage on multicore machines
- The single-machine scalability needs improvement, while on cloud architectures a huge speedup can be achieved

Acknowledgement

This work was supported by the Hungarian-Chinese cooperation grant No. MOST 2014DFG02050, Hungarian National Research Fund (OTKA) grant K120660 and K123815. We acknowledge the support of the Wigner Data Center and Wigner GPU Laboratory.

References

[1] X.N. Wang, M. Gyulassy, Phys. Rev. D44, 3501 (1991), W.T. Deng, X.N. Wang, R. Xu, Phys. Rev. C83, 014915 (2011).
[2] T. Sjöstrand, Comput. Phys. Commun. 191, 159 (2015)
[3] Albrecht, J., Alves, A.A. et al. Comput Softw Big Sci 3: 7 (2019)
[4] G. Bíró, G.G. Barnaföldi, G. Papp, T.S. Biró, Universe, 5, 134 (2019)
[5] G. Bíró, G.G. Barnaföldi, G. Papp, M. Gyulassy, P. Lévai, X.N. Wang, B.W. Zhang, PoS HardProbes 045 (2019)
[6] G. Bíró, G. Papp, G.G. Barnaföldi, D. Nagy, M. Gyulassy, P. Lévai, X.N. Wang, B.W. Zhang,, MDPI Proc 10 1, 4 (2019)