Development and implementation of first level event selection process on heterogeneous systems for high energy heavy ion collision experiments

V. Singhal[1],* V. Friese[2], and S. Chattopadhyay[1]
[1]*HBNI, VECC, 1/AF Bidhan Nagar, Kolkata-700 064, India and*
[2]*GSI Darmstadt, D-64291 Darmstadt, Germany*

Introduction

The computing demands for modern experiments in high-energy particle and nuclear physics are becoming increasingly challenging. This holds in particular for experiments relying heavily on real-time data processing. An example is the Compressed Baryonic Matter experiment (CBM) at the future Facility for Antiproton and Ion Research (FAIR), Darmstadt, Germany. In High Energy collisions, depending on the colliding ions, many particles are generated with extremely low production cross section, one such is J/ψ. J/ψ decays promptly into the dimuon channel. The multiplicity of charmonium is extremely small, therefore, to get reasonable statistics, CBM experiment is intended to run with unprecedented $\sim 10^7$ collisions per second. A Muon Chamber (MuCh) system will be employed for the detection of dimuon pairs originating from the nucleus-nucleus collisions. The MuCh system is being developed at VECC and it consists of alternating layers of segmented absorbers and detector stations. Each station consists of three detector layers. Due to high interaction rates, CBM will employ a free-streaming data acquisition with self-triggered readout electronics, without any hardware trigger. Efforts are concentrated towards the simulation of such raw data stream in which each individual detector channel message termed as *"digi"* will contain a global time stamp to generate free-flow data scenario. As collision rate is extremely high and J/ψ production is very low at FAIR energies (E_L = 10-40 AGeV), a real time selection process has been demonstrated in the titled thesis and provides a) an event selection *"trigger algorithm"* which decides the specific event contains J/ψ or not, b) to cope with 10MHz rate, use of heterogeneous computing has been investigated and c) time based software stack has been developed to simulate trigger-less real time collision and generated realistic free-running data stream.

First-level event selection process

The signature of a, $J/\psi \rightarrow \mu^+\mu^-$, candidate event is the simultaneous registration of two particles in the trigger station consisting of three detector layers provides three position measurements for a muon, which can be extrapolated backward to the target. On ba-

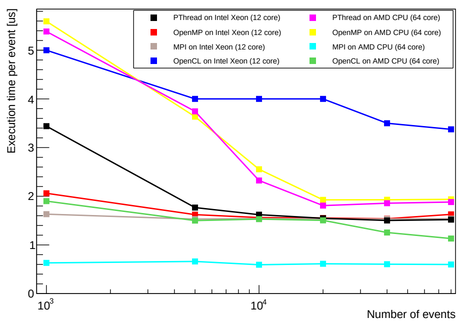

FIG. 1: Execution time per event as a function of the number of events processed at a time for the implementations with pthread, OpenMP, MPI and OpenCL on the Intel and AMD CPUs. The number of threads equals that of the available physical cores (12 for Intel, 64 for AMD).

*Electronic address: vikas@vecc.gov.in

sis of this signature, an **Brute Force event selection** algorithm has been developed. As compute complexity of algorithm is O(n^3), another algorithm, named **Selective**, is also developed. The complexity of selective algorithm is O(n^2) [1].

Heterogeneous computing

Modern computers come with a variety of concepts for concurrent data processing on many-core architectures, examples coprocessors, GPU, APU etc. The architecture of such systems is based on the SIMT (Single Instruction Multiple Thread) technology and come with a plenitude of processing units, allowing to run many threads in parallel. However, in order to make use of their computing potential, adequate programming paradigms are needed. To achieve the fast execution of event selection process, heterogeneous computing based on many/multi core machine has been investigated in detail [2]. The pure and heterogeneous parallel programming paradigms have been used for implementation & optimization of event selection algorithms. Detailed comparisons have been performed among pthread, OpenMP, MPI and OpenCL for two different types of hardware architecture Intel and AMD. The execution time for different configurations are shown in figure 1.

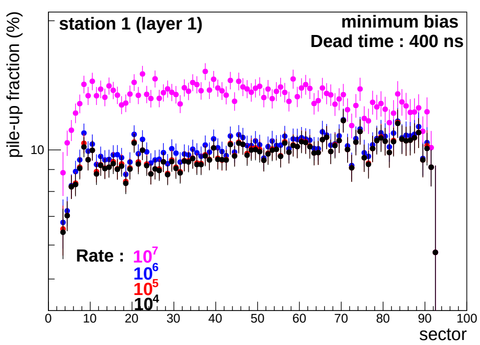

FIG. 2: Pile-up fraction in each sector of the first layer of the first MuCh station for a dead time of 400 ns and for different interaction rates.

Time Based Signal Generation

The data acquisition systems of most experiments in high-energy particle or nuclear physics are based on hardware triggers which defines an "event". The software framework used for simulation and analysis of such experiments are thus designed on an event-by-event scheme, where each event is treated as a separate and independent entity. But this triggered readout scheme is not feasible at CBM which intends to inspect up to 10^7 nuclear collisions per second, each producing several hundreds of particles to be registered in the detectors. These conditions will lead to use a trigger-less, free running data acquisition, where self-triggered front-end electronics register signals above a predefined threshold as caused by particles traversing the respective detectors and autonomously push the data forward. Such a system is not limited by latency, i.e. the time needed to generate a hardware trigger, but by data throughput bandwidth. It results in a continuous data stream in contrast to a series of events defined by the hardware trigger in conventional readout schemes. The association of raw data to physical events is based on precise time stamps.

We have investigated, the software to analyse this data stream - both in real-time and offline in detail and the time based simulation of the muon detector system MuCh incorporating response of detectors, front-end electronics and electronics noise has been developed and performance of the same has been studied. Consequence of a self triggered free running data stream is inter-event pile-up due to deadtime of the underlying readout chip which is STS-MuCh-Xyter. Figure 2 shows the pile-up fraction differentially for each sector in the first layer of the first MuCh station for different interaction rates, varying from 10^4/s to 10^7/s [3].

References

[1] DOI: 10.1109/INDICON.2015.7443569.
[2] V. Singhal et al 2020 CPC 253 107190.
[3] V. Singhal et al 2021 JINST 16 P08043.