Accelerating online software development with CBM data challenges

J. de Cuveland[1]*, D. Hutter*[1]*, V. Friese*[2]*, S. Gorbunov*[2]*, P.-A. Loizeau*[2]*, A. Redelbach*[1]*, D. Smith*[2]*, F. Uhlig*[2]*, F. Weiglhofer*[1]*, and S. Zharko*[2]

[1]FIAS, Frankfurt, Germany; [2]GSI, Darmstadt, Germany

Development and testing of software designed particularly for CBM online computing have motivated three dedicated data challenges (DCs) in 2023. Real-time event selection in the context of data analysis of the mCBM full-system prototype demands highly efficient and robust software components.

Introduction

As discussed also in a previous report [1], efficient online data processing comes with special requirements for all data structures, interfaces, and algorithms. To this end, many efforts have been taken to separate the actual algorithms from framework layers and to focus developments on code relevant for reconstruction considering execution speed also from the beginning. In the context of online computing, aspects of integration of individual tasks are particularly relevant to ensure interoperability and reliable interfaces. Implementing regular data challenges every one to two months leads quite naturally to reasonable milestones and deadlines.

Executing the data chain using synthetic data also has the advantage that one can easily adopt the rate of incoming FLES data. Once the processing at a defined rate for the input is stable, an acceleration of incoming data can shed light on the scaling behaviour towards higher rates.

Since the CBM software cannot generate detector data formats from simulation, pre-recorded mCBM data of 2023 have been used. In order to emulate the actual online processing as closely as possible, we have chosen a setup based on mFLES nodes sending timeslice data to be processed by VIRGO nodes.

Setup

Four mFLES nodes have been used for replaying of archived timeslice (tsa) files, allowing a continuous playback at real time or also faster. In total, 80 tsa files were sent to eight Cbmreco binaries on each of the ten VIRGO nodes reserved.

These binaries are included in Apptainer containers based on Debian 12. Their execution was started by Slurm jobs on the nodes in the VIRGO cluster thus leading to online processing of 80 Cbmreco binaries von VIRGO nodes in parallel.

Most of the tests have utilized multi-core CPUs on VIRGO cluster nodes with AMD EPYC processor architecture. In this case also parallelisation using OpenMP has been used, typically setting the number of threads to twelve. For a shorter time also the execution of processing on three GPU VIRGO servers has been tested.

During a data challenge inspection of online results including input and output data rates is indispensable, similar to data taking during beamtimes. Therefore performance data are collected in an Influx database interfaced to Grafana displays. It should be emphasized that in the setup for DCs, monitoring and data quality assurance have been separated from the online algorithms. The algorithms provide monitoring data to higher-level instances, where their handling is managed, such as the InfluxDB as basis of the Grafana online display.

Archived files of two mCBM benchmark runs served as input source. The run number 2391 contains 960 tsa files and run 2488 includes 1123 tsa files, respectively. Data of corresponding timeslices have been merged and split again into 80 timeslice streams.

Integration of tasks and monitoring

The following basic tasks have been included in the reconstruction steps:

- Unpacker algorithms have been formulated to operate on one FLES microslice as the smallest independent data unit. The data challenges have included STS, TOF, MUCH, TRD, TRD2D, BMON and RICH unpackers. Several optimisations towards multi-threading on the task-level for unpackers have been implemented and are e.g. summarised for the STS in [2].

- Digis as output from unpacking can be used for digi-based triggering and subsequent event building. Identifying time-clusters in the time distribution of digis thus enables the definition of trigger-based events.

- Digis in the STS are basis of local STS reconstruction, consisting of finding STS clusters and hits. Developments for highly efficient parallel execution on CPUs and GPUs have been tested during the data challenges, see also [3].

- Similarly an improved version of local TOF reconstruction has been developed, c.f. [4], in 2023 and could be tested during the data challenges.

- Using hits from STS or TOF, also tracking has been included. Details of developments of tracking software for mCBM are summarised in [5].

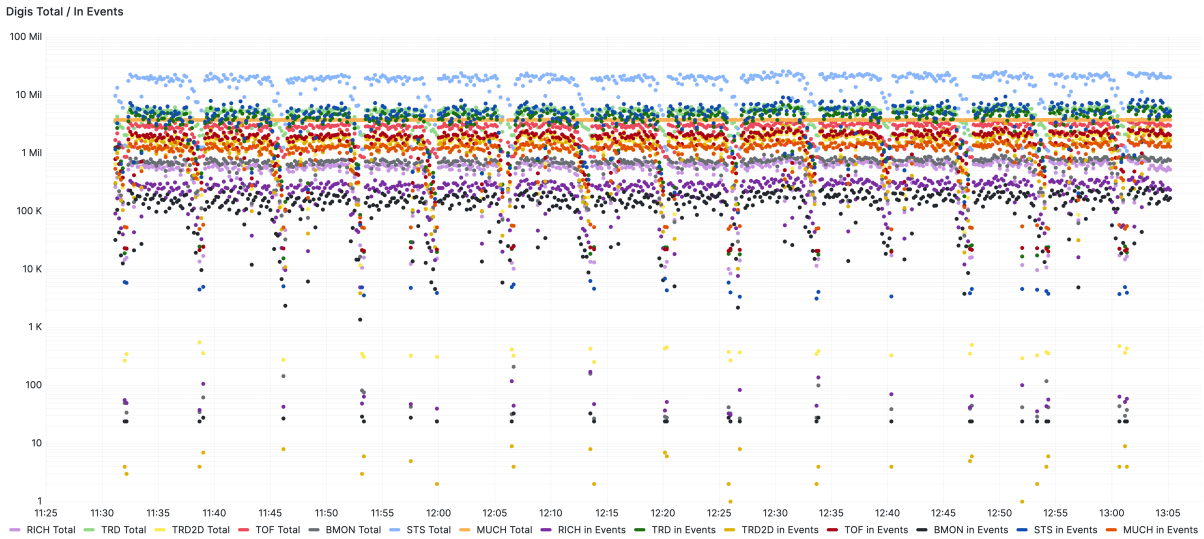

Figure 1: Numbers of digis per timeslice from different mCBM detector systems processed in DC 3. Also the corresponding numbers of digis after event selection are shown.

Results

Replaying timeslice data from mFLES nodes to processing at VIRGO worked smoothly. As an example, a total input rate of 8000 GB in 90 min has been measured during DC 2, corresponding approximately to 15 GB/s. During that time event building has been performed, typically at a rate between 1 GB/s and 2 GB/s. Variation of the input data rate allowed to increase and test incoming rates of up to a factor of eight as compared to the original speed.

Also the operation of 10 VIRGO nodes operating on timeslices in parallel has been realized. The prepared online processing binaries worked for hours demonstrating the stability of the systems. It should be underlined that this includes unpacking of timeslices containing STS, TOF, MUCH, TRD, TRD2D, BMON and RICH data as a basis for subsequent digi-based triggering plus event building. Fig. 1 gives an overview of numbers of digis from different mCBM detector systems processed in DC 3 displaying also the corresponding numbers of digis after event selection. It is interesting to note that the largest number of unpacked digis are from mSTS, whereas the fraction of mSTS digis actually selected is lower than e.g. for mTOF digis, also as a result of the definition of events.

Finally the software for mCBM online tracking using hits of TOF and STS has been integrated successfully. Also the online view of histograms for quality control of results from tracking has been achieved, forming the basis of future online selection of events containing characteristic tracks for physics analyses.

In order to store data in a resource-efficient way, outgoing data can be compressed. One option for file compression based on Zstandard has also been tested. For a period of two hours the processing has been done on three GPU servers using AMD Radeon Instinct MI100 at VIRGO. During that time STS unpacking plus subsequent HitFinding optimised particularly for parallel GPU processing have been performed successfully. The actual performances of individual tasks can be further investigated including comparison between runtimes on CPUs, GPUs and utilisation of VIRGO resources.

Outlook

The data challenge campaigns in 2023 have made software developments for CBM online computing more visible. At the same time also the principal readiness and robustness for data taking have been demonstrated. In order to enable fast validation of the reconstructed objects, an extension of online quality assurance histograms is foreseen. Also in the future regular data challenges are planned, focusing on integrations of new online software developments and workflows focusing more on event selection. As a complement of enabling software for physics analysis, also an efficient usage of computing resources at the VIRGO HPC cluster can be further investigated. Investigating performances based on runtimes for different numbers of threads will allow more efficient utilisation of CPU cores and at the same time show the potential for further optimisation of CBM online algorithms.

References

[1] J. de Cuveland and V. Friese, “Towards an online-capable software stack”, GSI Scientific Report 2021

[2] S. Heinemann, “CBM Note for STS unpacking”, to be published

[3] F. Weiglhofer, “Optimized STS hit finding using XPU”

[4] D. Smith, “Online-capable hitfinder for TOF”, this report

[5] S. Gorbunov, “Developments for tracking in mCBM”, this report