



Contribution ID: 124

Type: Sectional reports

## Analysis of the effectiveness of various methods for parallelizing data processing implemented in the ROOT package.

Thursday 8 July 2021 14:45 (15 minutes)

The ROOT software package has a central role in high energy analytics and is being upgraded in several ways to improve processing performance. In this paper, we will consider several tools implemented in this framework for calculations on modern heterogeneous computing architectures.

PROOF (Parallel ROOT Facility – an extension of ROOT system) uses the natural parallelism of data structures located in files of a special format, providing direct access to any particular value. PROOF [1,2] divides common work into small fragments – packets. We have investigated how the processing speed depends on the minimum or maximum packet size in seconds or events, the size of the first packet used for calibration. Our calculations also showed that when processing data using PROOF, it is desirable to use the highest possible structuring of the primary data.

Using the heterogeneity of modern computing architectures, ROOT can be used in common with OpenCL technology. Unlike CUDA, the use of which is limited only by graphics processing units, OpenCL technology is perfectly adapted to various families of microprocessors, so a program developed for one type of computing architecture can be easily transferred to another. The expediency of performing calculations on a GPU is considered, depending on the type of data processing algorithms.

Implicit multithreading [3], implemented in ROOT since version 6.06, is based on one of the key innovations of the framework – the columnar data format. Data components (variables, structures, or objects) are converted into independent buffers, which are periodically compressed and written to memory. Implicit multithreading parallelizes loops over buffers during transformation and compression stages.

When processing large amounts of data, read and write speed can be critical. A new function for asynchronous file merging, implemented in the TBufferMerger class [4], allows to write data in parallel from multiple streams to a single output file. Our calculations show good scalability of macro execution time on the number of processor cores used.

### References

1. Brun R. et al. Parallel interactive data analysis with PROOF //Nuclear Instruments and Methods in Physics Research. A559, pp 13-16, 2006.
2. Solovjeva T.M., Soloviev A.G. Comparative study of the effectiveness of PROOF with other parallelization methods implemented in the ROOT software package //Computer Physics Communications,v.233, p 41-43, 2018.
3. Piparo D. et al. Expressing Parallelism with ROOT //Journal of Physics: Conf. Series 898 (2017) 072022.
4. Amadio G., Canal F., Guiraud E. and Piparo D. Writing ROOT Data in Parallel with TBufferMerger // EPJ Web of Conferences 214, 05037 (2019).

### Summary

**Author:** SOLOVJEVA, Tatyana (Jinr)

**Presenter:** SOLOVJEVA, Tatyana (Jinr)

**Session Classification:** HPC

**Track Classification:** 5. High Performance Computing