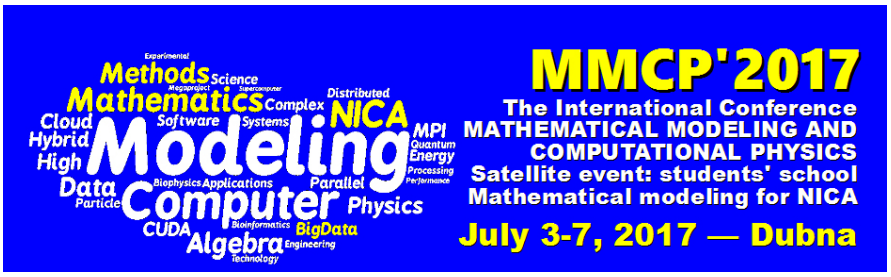

Contribution ID: **113** Type: **not specified**

Comparing the effectiveness of PROOF with others methods of parallelizm for the experimental data processing.

Friday 7 July 2017 16:00 (15 minutes)

The modern scientific research requires careful modeling of experiments, as well as the fast and qualitative processing of a large amount of data. Under that the comparison of the used models and the optimization of the program code are performed. Optimization implyes both algorithmic code improvement and the usage of high-performance and parallel technologies.
In high-energy physics, the standard of data analysis and visualization has become ROOT - the object-oriented framework been developed at CERN. ROOT based software is used in most of modern experiments. ROOT provides several options for parallelism. First, it contains PROOF[1] - Parallel ROOT Facility - ROOT extension, which performs interactive analysis on large sets of ROOT files in parallel on multiprocessor machines. Secondly, it allows the use of technologies OpenMP and MPI.
In current paper we compare the effectiveness of all ROOT methods for parallelizing computations, depending on the type of experimental data analysis, on their volume and on the computing platform.
The PROOF cluster is built according to the standard master-workers scheme. Due to the multi-level architecture, which allows creating a hierarchy of master and submaster nodes, this approach can be easily adapted to a wide range of virtual clusters. The user, working in the ROOT session, can start the processes communicating with the PROOF-cluster and submit requests for job processing. Upon receiving request for job processing, a special ROOT application - proofserv - starts on the wizard and on the working nodes for each user session. The process executed on the master coordinates the work between the work nodes and combines the results. At work nodes, the proofserv process does performed the computational work itself, processing individual jobs.
Parallel calculations can also be implemented with the TThread class. A single process can have multiple threads. The actual work is done in the class TThreadImp (or TPosixThread or Twin32Thread).

Reference

1. Brun, R. et al. Parallel interactive data analysis with PROOF // Nuclear Instruments and Methods in Physics Research. –2006. –A559. –P. 13–16.

Author: Dr SOLOVJEVA, Tatiana (JINR)

Co-author: Dr SOLOVIEV, Alexei (JINR)

Presenter: Dr SOLOVJEVA, Tatiana (JINR)

Session Classification: Mathematical methods and software for experimental data processing