Introduction

In order to take full advantage of new computer architectures and to satisfy the requirement of maximizing the CPU usage with increasing amount of data to analysis, parallelization and vectorization have been introduced in the ROOT mathematical and statistical libraries, requiring minimal changes in user code.

Parallelization and Vectorization of ROOT Fitting classes

X. Valls, L. Moneta for the ROOT Team

As part of this effort, new generic classes supporting a task based parallelization mode have been defined in ROOT, which can be used for a wide range of computational tasks in the field of High Energy Physics. The support for different SIMD's libraries has also been included.

Tools for parallelism

Task level parallelism: TThreadExecutor

- Task-oriented, multithreaded MapReduce for ROOT
- Provides operations Map, Reduce, Foreach and even chunked mapping with partial reduction.
- Used in math fitting, TMVA(Boosted Decision Tree evaluation, Deep Neural Networks processing), Implicit multithreading operations in I/O (reading, deserialization and decompression of tree branches in parallel, parallel writing) and for parallel execution of functional chains in TDataFrame.

Instruction level parallelism: VecCore

- Provides efficient vectorization on all platforms by writing abstract, architecture-generic code that will map to each of its optional backends' concrete types, methods or instructions. Includes a scalar backend for the case when SIMD operations are not available.
- See the poster “Speeding up software with VecCore, a portable SIMD library” by Guilherme Amadio.

References

1. VecCore Library https://github.com/root-project/veccore
2. Vec : https://github.com/VcDevel/Vc
3. UME: SIMD https://github.com/edanor/umesimd
4. ROOT Data Analysis Framework https://root.cern