

Parallelization of TMVA High Level Algorithms in ROOT



Omar Zapata

CMS

CERN 2017



Outline



ROOT

Data Analysis Framework



- TMVA
 - Motivation
 - Current status
 - New architecture
 - Parallelization
- Future outlook
- Conclusions



Motivation

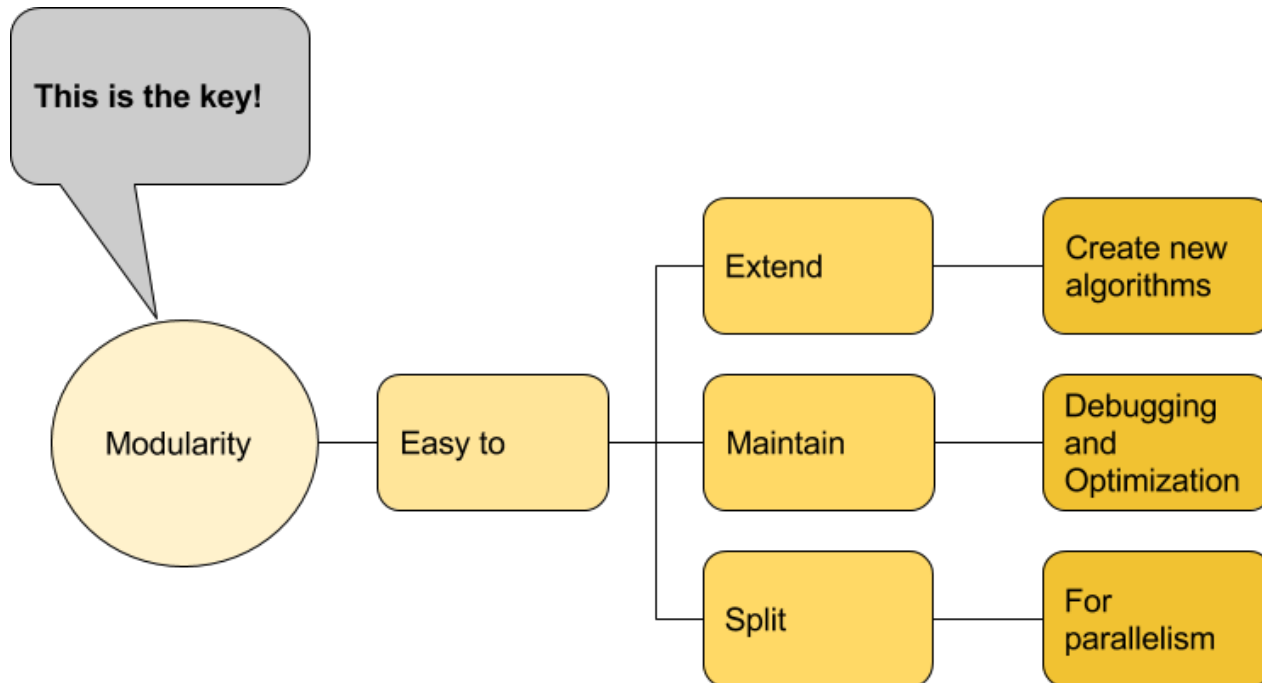
Some new developments in parallel computing in TMVA is growing without a global structure or design with different paradigms that can produce an unmaintainable software condensing all in some few classes with a lot of dependencies.

The idea with this proposal is to implement a clear architecture for multiple parallel paradigms in TMVA that lets extend the different algorithms with the different paradigms.



New architecture for TMVA

- Why?
 - Need to:
 - Parallelize the algorithms
 - Create more algorithms
 - Modularize the Factory





Parallel architecture for TMVA

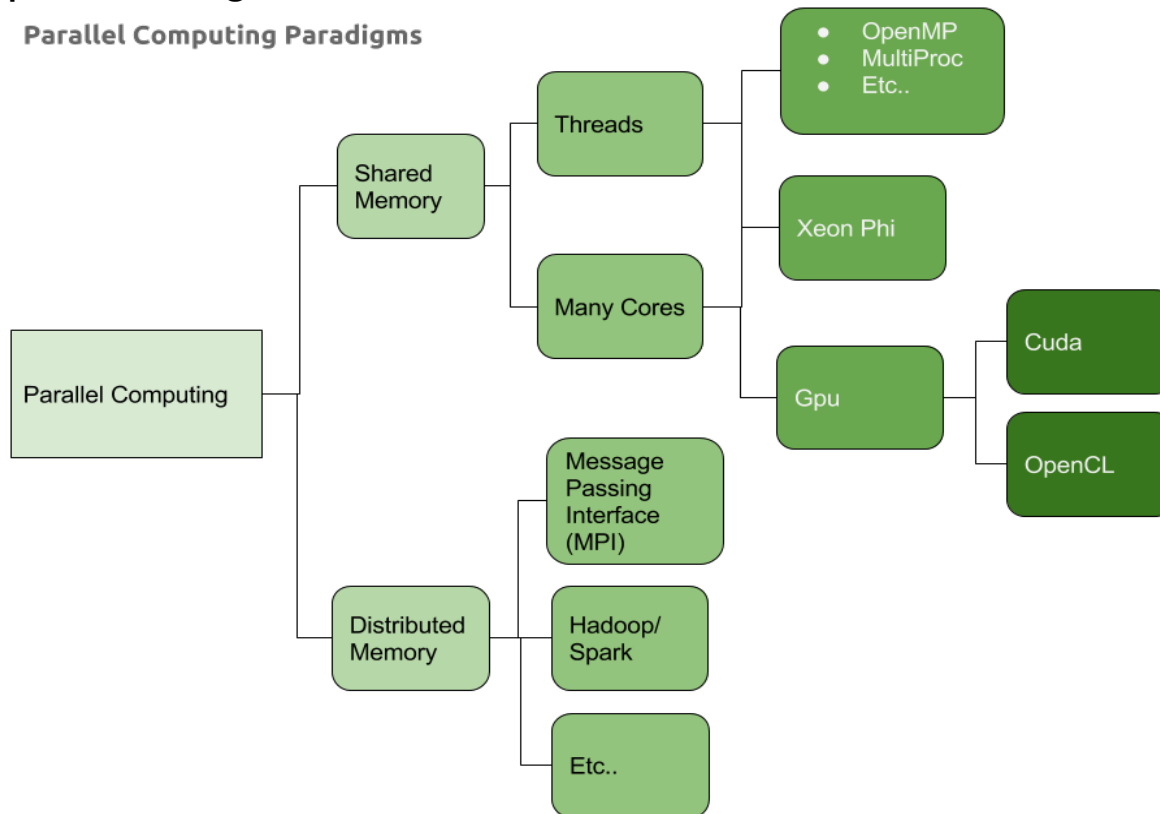
- Why parallel computing

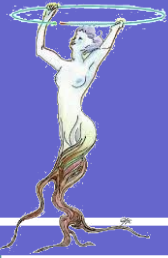
- Huge amount of data
- Expensive algorithms

- Which paradigm

- Many

Parallel Computing Paradigms



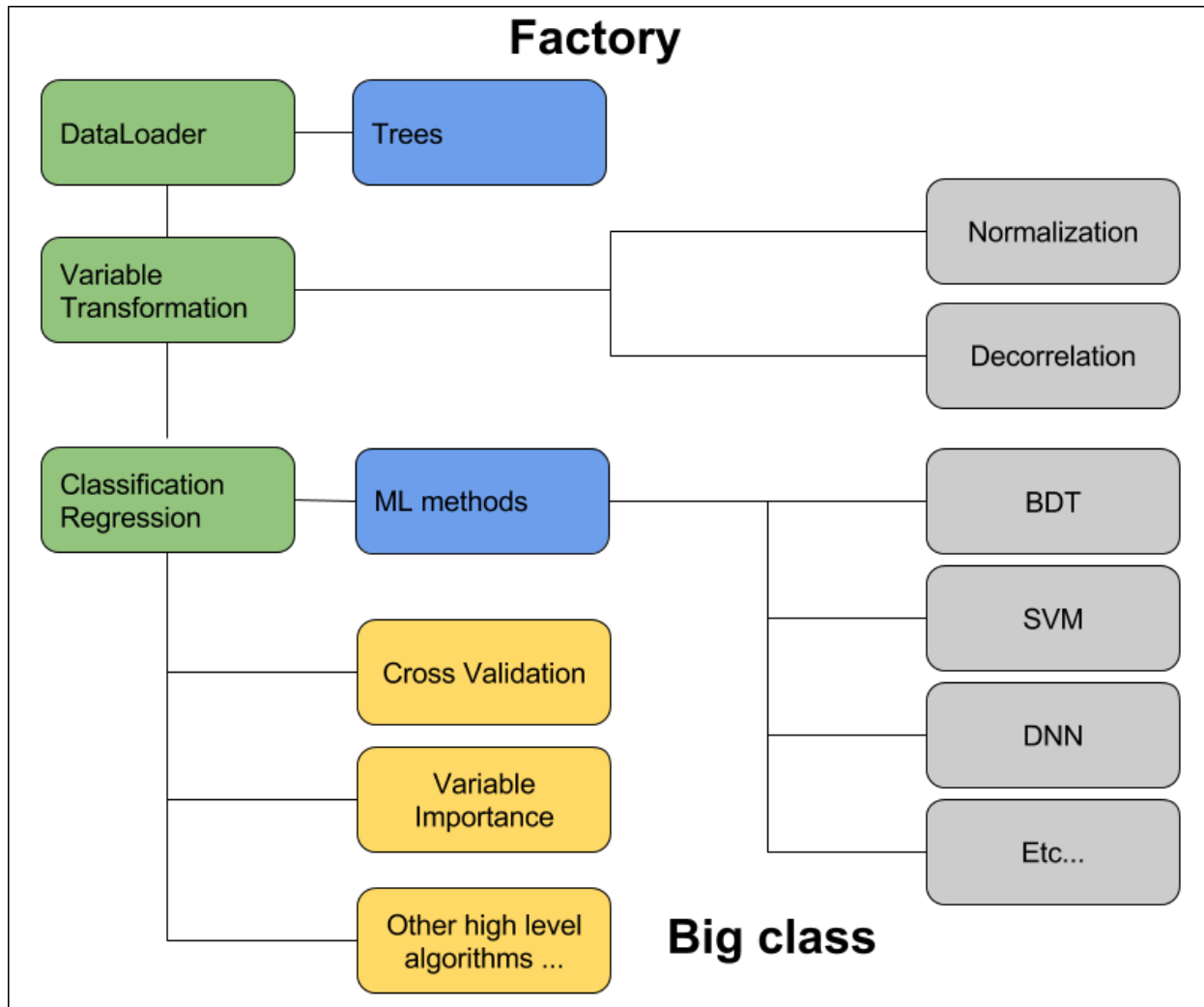


Algorithms

- We divide the machine learning algorithms in two parts
 - The low level algorithms are the methods to Train/Tests a model like (Boosted Decision Trees, Multilayer Perceptron, Support Vector Machine etc..)
 - High level algorithms are algorithms to do exploratory analysis or validation, like cross validation, variable importance, hyper parameter optimization etc..
- This proposal describe the architecture for high level algorithms.

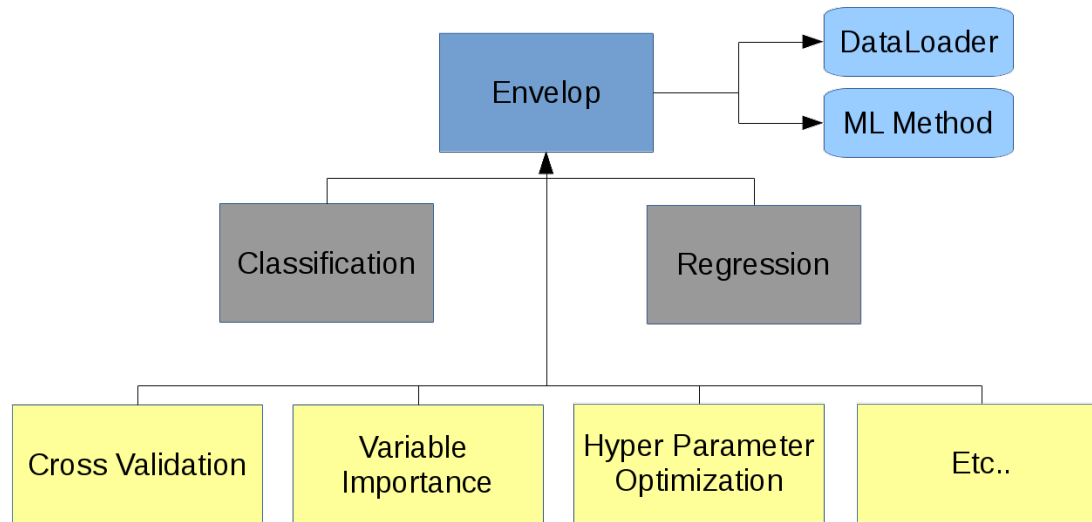


Current status





Current status



<https://github.com/root-project/root/blob/master/tmva/tmva/inc/TMVA/Envelope.h>

<https://github.com/root-project/root/blob/master/tmva/tmva/inc/TMVA/CrossValidation.h>

<https://github.com/root-project/root/blob/master/tmva/tmva/inc/TMVA/VariableImportance.h>

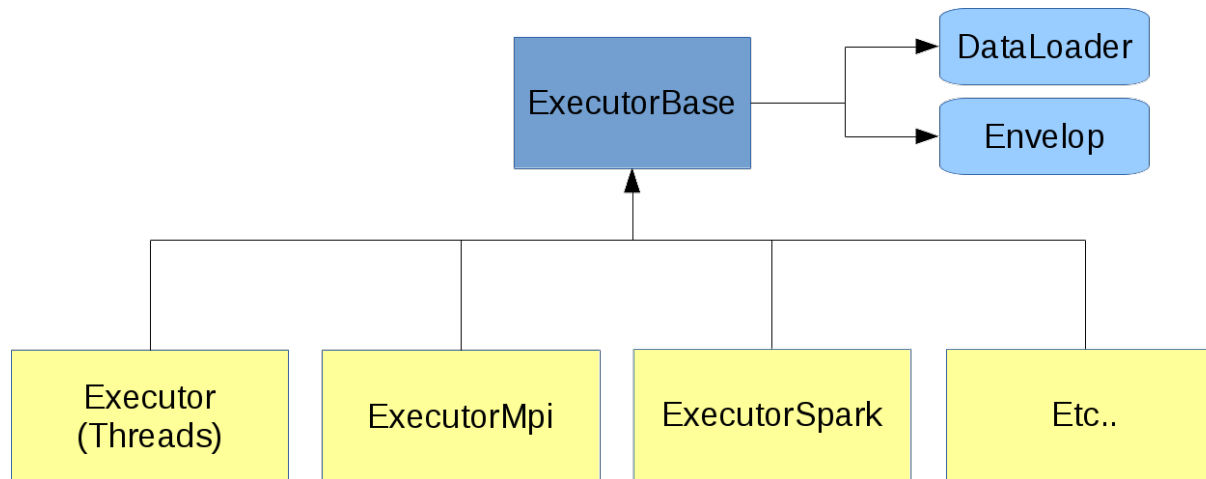
<https://github.com/root-project/root/blob/master/tmva/tmva/inc/TMVA/HyperParameterOptimisation.h>

TODO:

- * Create a class Classification and Regression that inherit from Envelop
- * Deprecate Factory to move to the new architecture or rewrite factory that in the new architecture don't require the huge amount of code that is using.



New parallel architecture for TMVA



Prototype

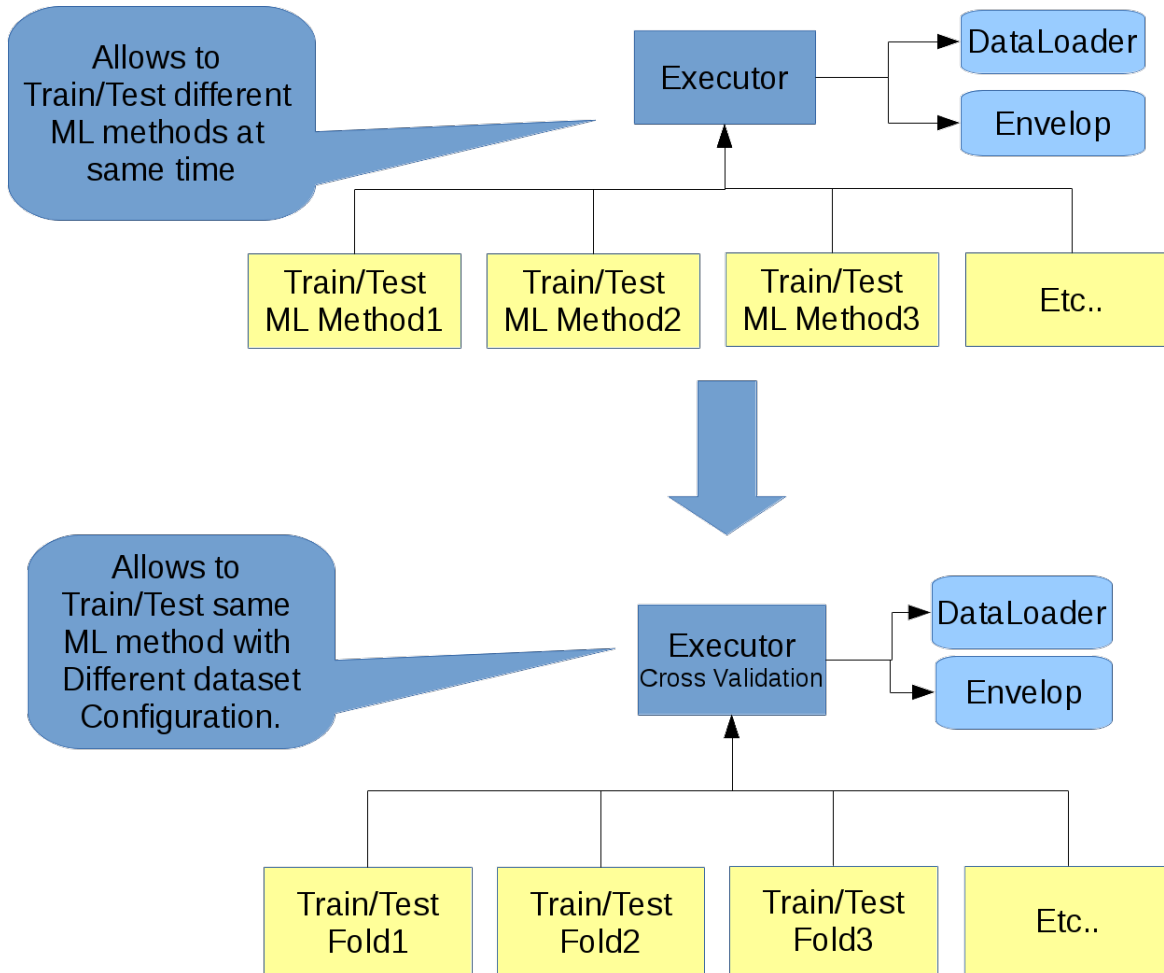
<https://github.com/oprojects/root/blob/master-tmva-pexecutor/tmva/ptmva/inc/TMVA/ParallelExecutorBase.h>

<https://github.com/oprojects/root/blob/master-tmva-pexecutor/tmva/ptmva/inc/TMVA/ParallelExecutor.h>

<https://github.com/oprojects/root/blob/master-tmva-pexecutor/tmva/ptmva/inc/TMVA/ParallelExecutorMpi.h>

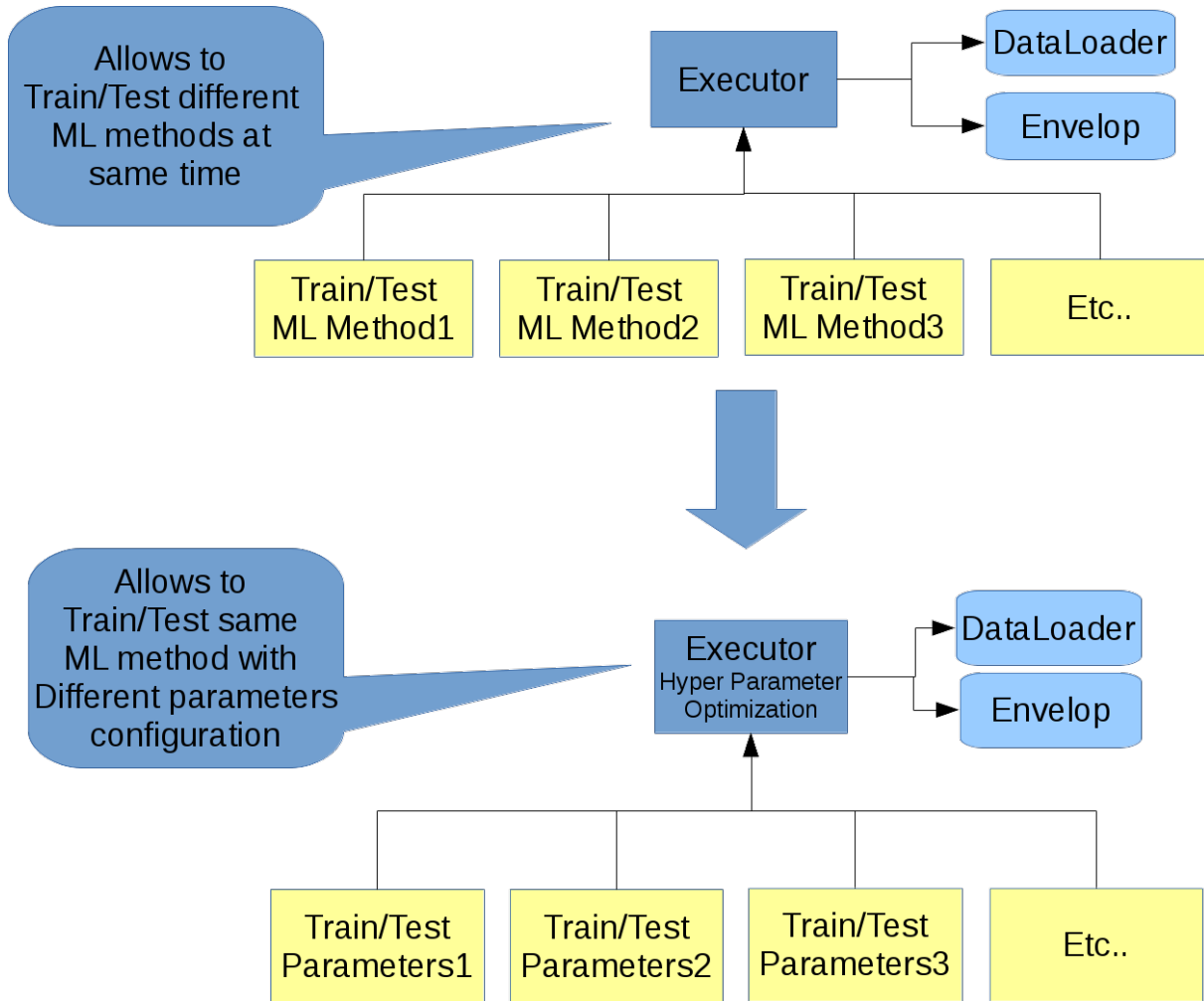


New parallel architecture for TMVA



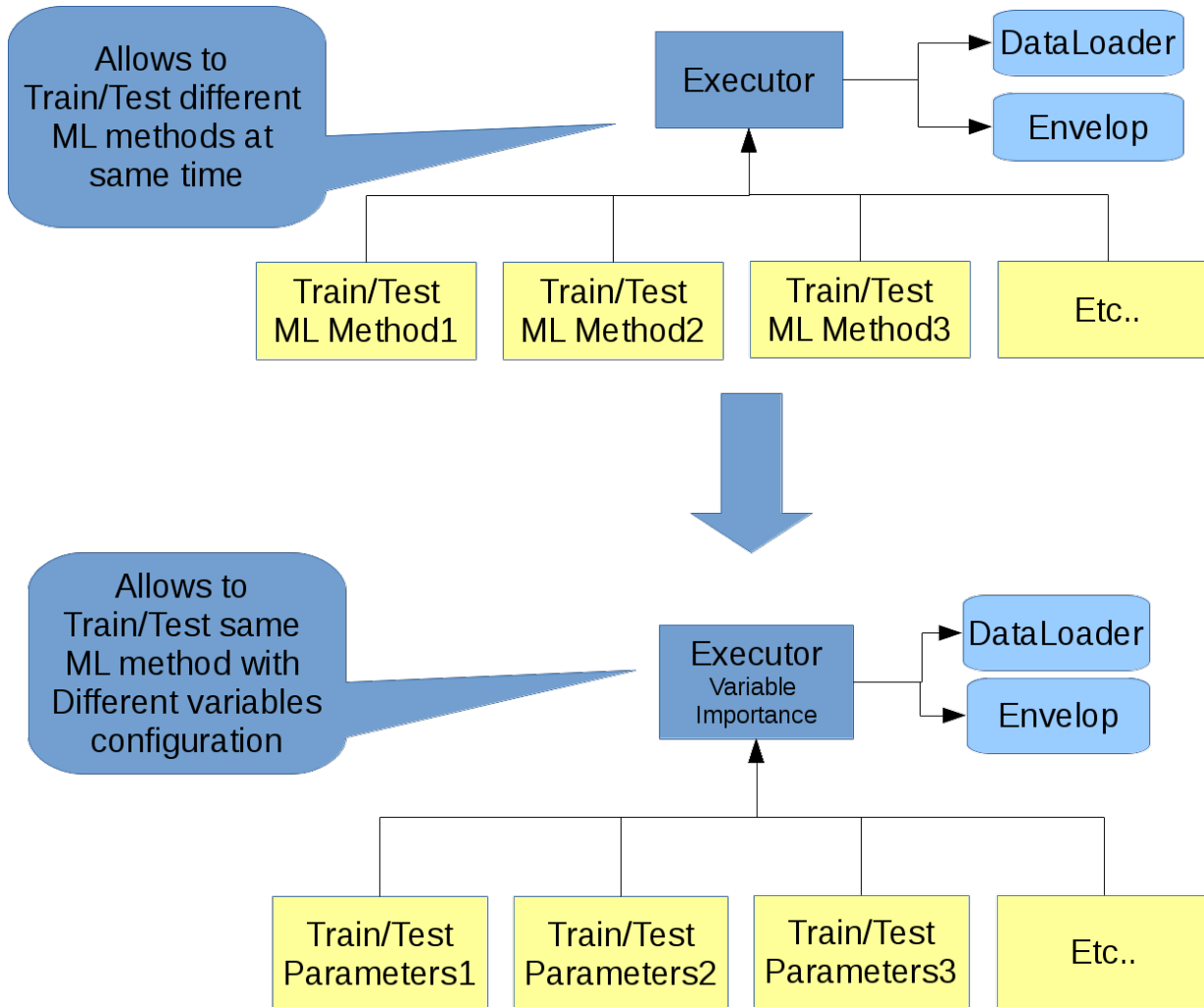


New parallel architecture for TMVA





New parallel architecture for TMVA





Future outlook

- In general TMVA needs
 - Finish the new architecture
 - create the classes Classification and Regression
 - Integrate TMVA to ROOT plugins system
 - Doxygen documentation
 - More tests for ctest
 - New features from new c++ standards
 - To update de users guide
 - Deprecate the old website <http://tmva.sourceforge.net/> and may to create a new one with updated documentation.
- Parallelization
 - Implement the design of the architecture for high level algorithms
 - Create a design to parallelize low level algorithms in a standart way.
 - Serialize more needed classes.
- Integrate new TMVA features.

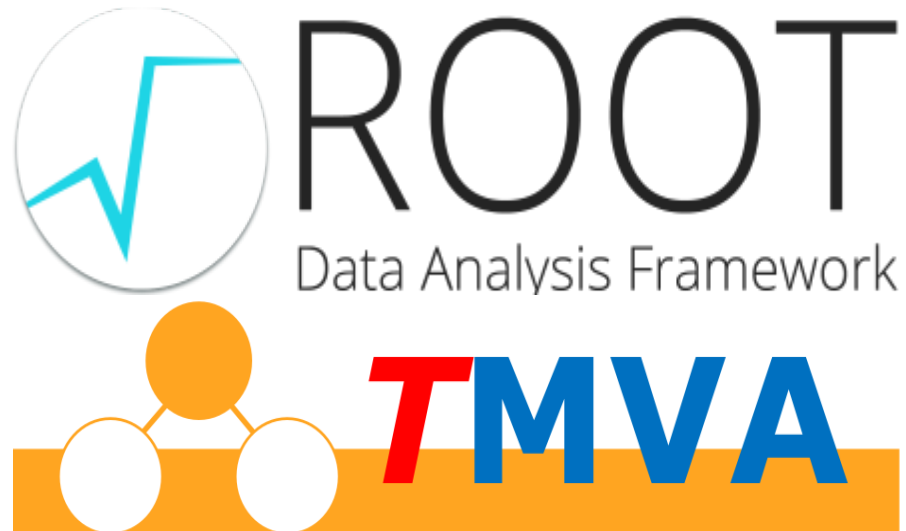


Conclusions

- Modularity in TMVA
- Parallel architectures
 - Theads
 - MultiProc
 - Spark
 - Mpi
 - Gpu
- We need restructure TMVA accoding the new C++ standarts and we need to think in the future of TMVA with the new architecture for ROOT 7 to rewrite it using modern architectures.



More Information



Website

<http://oproject.org>



To finish

Thanks !

BackUp

- A possible design for low level algorithms can be to specify in the booking options with parallel architecture do you want to use, in a similar way that the current DNN method.

```
factory->BookMethod(dl, TMVA::Types::kDNN,"DNN","Architecture=GPU:..." );  
factory->BookMethod(dl, TMVA::Types::kBDT,"BDTG","Architecture=CPU:..." );  
factory->BookMethod(dl, TMVA::Types::kBDT,"SVM","Architecture=Spark:..." );  
....
```