Parallelization of Likelihoood function data analysis software based on RooFit package

Alfio Lazzaro, alfio.lazzaro@cern.ch, *CERN* Openlab, *Geneva*

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Jaipur (India), February 22 – 27, 2010

Introduction

- In data analysis, generally, all methods are based on optimization problems: find a *maximum* (for example in case of Statistical Significance Maximization or Maximum Likelihood) or a *minimum* (Expected Prediction Error) of a function
  - This is (in general) done by *numerical algorithms*
- The most largely used package in High Energy Physics for minimization is MINUIT
  - It uses the gradient of the function to find local minimum (MIGRAD), requiring
    - The calculation of the gradient of the function for each free parameter, naively $\left.\frac{\partial f}{\partial \hat{\theta}}\right|_{\hat{\theta}_0} \approx \frac{f(\hat{\theta}_0 + \hat{\mathrm{d}}) - f(\hat{\theta}_0 - \hat{\mathrm{d}})}{2\mathrm{d}}$ (2 function calls per each parameter)
    - The calculation of the covariance matrix of the free parameters (which means the second order derivatives)
  - The minimization is done in several steps moving in the Newton direction: *each step requires the calculation of the gradient*

Maximum Likelihood Fit

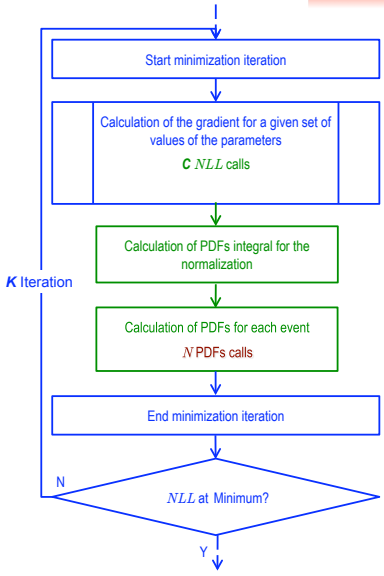
- In Maximum Likelihood fits we have to maximize the likelihood function, or minimize the Negative Log-Likelihood Function (NLL)

$$-\ln \mathcal{L} \equiv NLL = \ln\left(\sum_{j=1}^{s} n_j\right) - \sum_{i=1}^{N}\left(\ln \sum_{j=1}^{s} n_j \mathcal{P}_j^i\right)$$

j species (signals, backgrounds)
n_j number of events for specie j
$\mathcal{P}_j$ probability density functions (PDFs)
N number total of events to fit

  - The minimization is performed as function of free parameters: n_j number of events, parameters of $\mathcal{P}_j$
- The minimization requires the calculation of the NLL for each free parameter in each minimization step. Computational time depends on:
  1. the number **P** of free parameters and the complexity of the function
  2. the number N of events of the input sample
  3. Note, also, that $\mathcal{P}_j$ need to be normalized, i.e. calculation of PDFs integrals, which can be a slow procedure if we don't have analytical expressions
- The handling of NLL is performed by the RooFit package, inside the ROOT framework
- Complex fits of High Energy Physics measurements can *take hours to days*

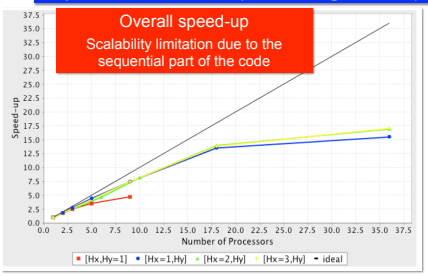
Parallelization

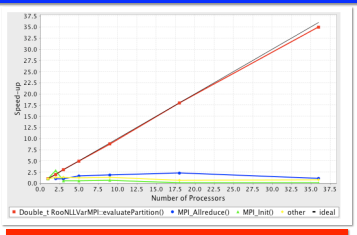


- Number of PDFs calculation calls during the minimization: **K** x **2P** x N
- Parallelization in two stages (reducing the number of communications between processors):
  - Gradient calculation (MINUIT): scattering of the **2P** NLL calls
  - NLL calculation (RooFit): scattering of the N PDFs calls
- Parallelization implemented using MPI, adopting a Cartesian topology (H_x x H_y processors)
  - H_x groups for gradient calculation
  - H_y groups for NLL calculation
- Total number of PDFs calls per processor: **K** x (**2P** / H_x) x (N / H_y)
- Code based on ROOT 5.26 and OpenMPI-1.3

Test @ INFN CNAF cluster, Bologna (Italy)

3 variables, 600K events, 23 free parameters
PDFs per each variable: 2 Gaussians for signal, parabola for background
Sequential execution time (Intel Xeon @ 2.66GHz): **~80 minutes**





`RooNLLVarMPI::evaluatePartition()` does the NLL calculation: excellent scalability

Conclusion

- Good scalability in case of large number of free parameters and large data samples: keeping low the number of communications, i.e. MPI overhead
- Main limitation due to the sequential part of the code for initialization and finalization of the fit. Working to improve these parts
- Under development: hybrid parallelization using MPI + OpenMP, parallelization of random events generation
- **Be ready for LHC intensive data analysis period!**