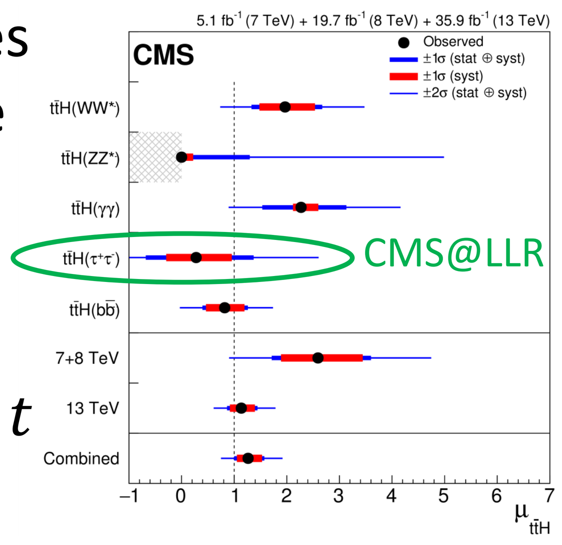
Deployment of a Matrix Element Method code for the $t\bar{t}H$ channel analysis on GPU's platform

G. Grasseau, F. Beaudette, A. Zabi, C. Martin Perez, A. Chiron, T. Strebler, G. Hautreux

Recent discovery of H boson in *ttH* channel (April 2018)

- In the Standard Model, the Higgs boson couples to fermions with a strength proportional to the fermion mass (Yukawa coupling).
- The decay to the $t\bar{t}$ final state is not kinematically possible.
- Probing the coupling of the Higgs boson to the t quark, the *heaviest* known fermion, is a high priority.



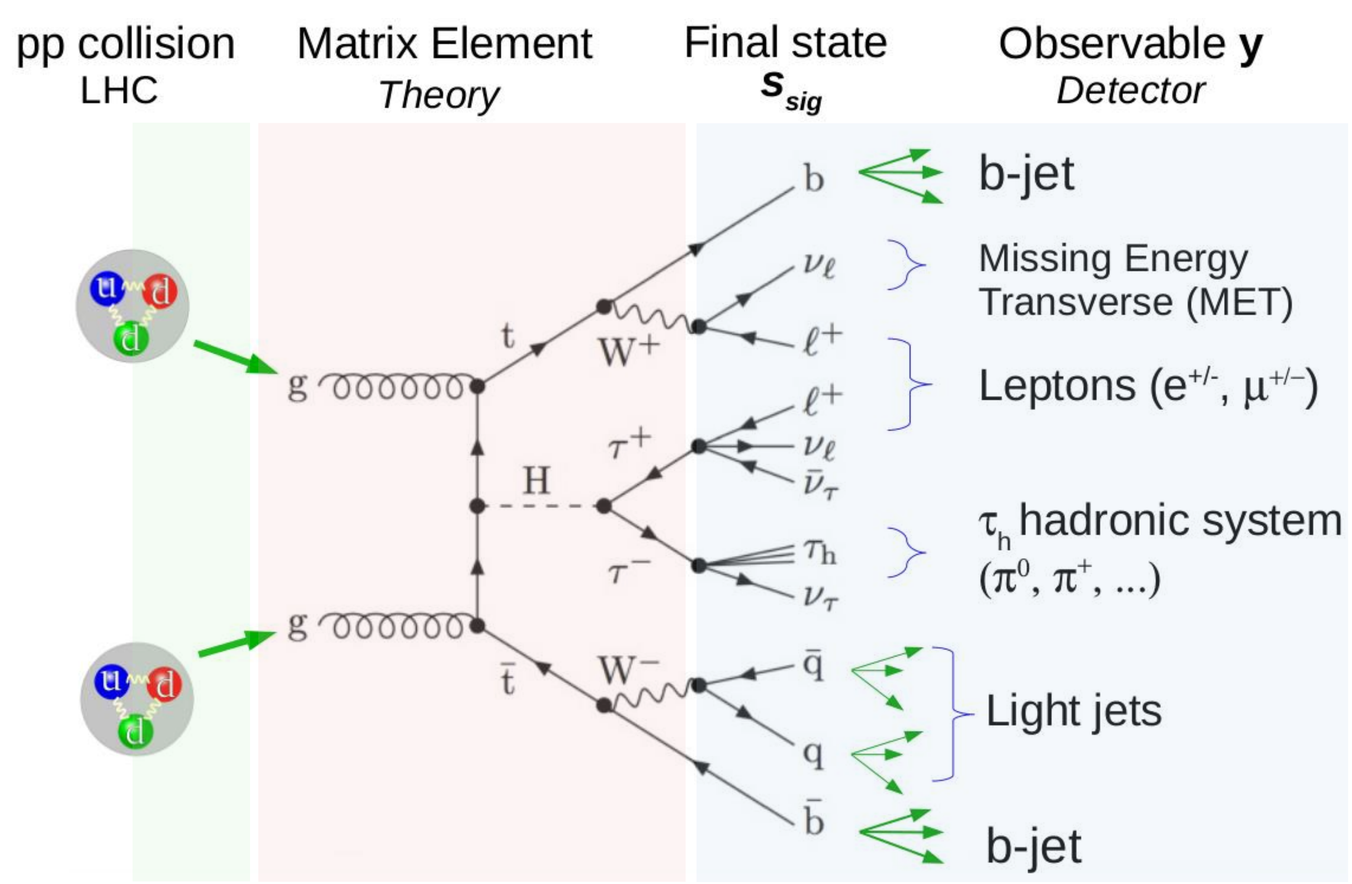
- The Higgs boson in association with $t\bar{t}$ final state can result from the fusion of a $t\bar{t}$ pair or through a radiation of t quark.
- First observation* of the simultaneous production of a **Higgs boson** with a $t\bar{t}$ pair ($t\bar{t}H$ channel) April 2018
- We (CMS@LLR) contributed to the $t\bar{t}H \rightarrow \tau\tau$ sub-channel

*A. M. Sirunyan et al. (CMS Collaboration), "Observation of tt̄ H Production", Phys. Rev. Lett. 120, 231801 (2018)

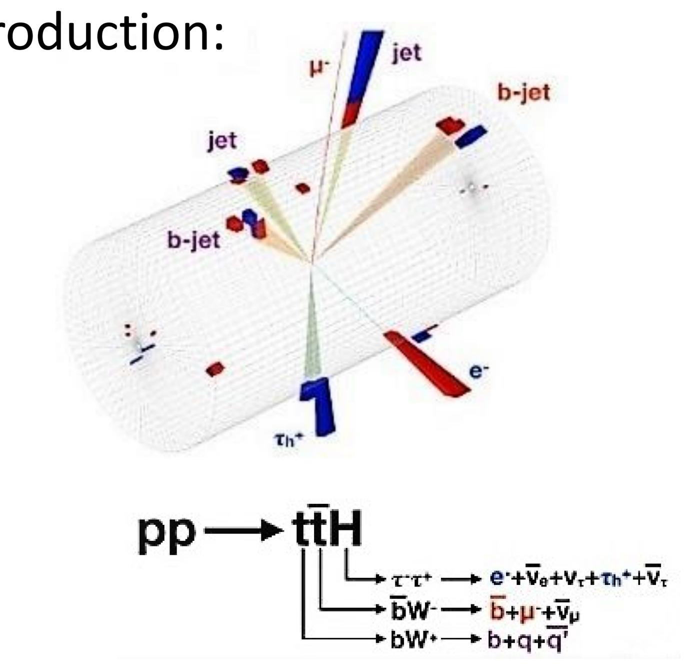
Matrix Element Method (MEM)

Unlike to supervised methods (neural networks, decision trees, …), the Matrix Element Method (MEM) allows classifying events thanks to the physics process. This sophisticated method is however very CPU time consuming and requires huge powerful computing platform to perform the CMS analyses carried out at our laboratory in a reasonable time.

Possibly a lot of integrals like below are computed for one event with different dimension values (from 3 to 7)



An event candidate for $t\bar{t}H \rightarrow \tau\tau$ production:



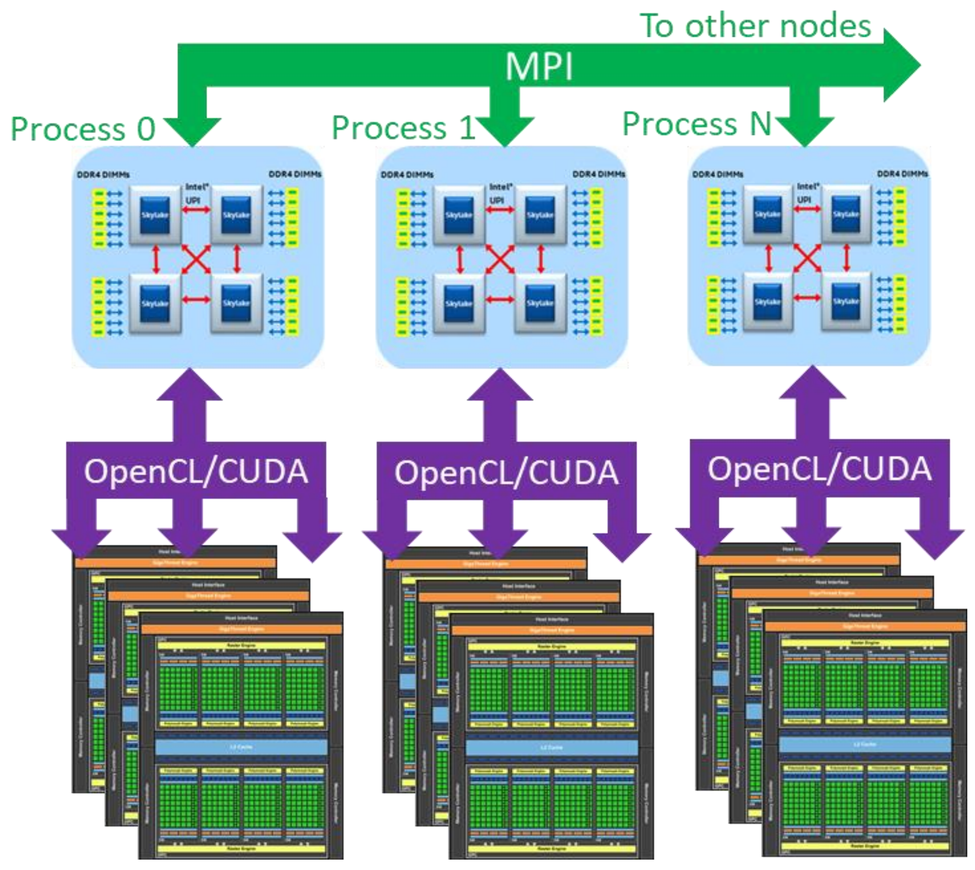
Weight of an event **y** | *p* processes | Parton Density Function (PDF) | Kinematics constrains | Matrix Element | Transfer Function

$$w_i(\mathbf{y}) = \frac{1}{\sigma_i}\sum_p \int d\mathbf{x}dx_a dx_b \frac{f(x_a,Q)f(x_b,Q)}{x_a x_b s}\delta^2(x_a P_a + x_b P_b - \sum p_k)|\mathcal{M}_i(\mathbf{x})|^2 W(\mathbf{y}||\mathbf{x})$$

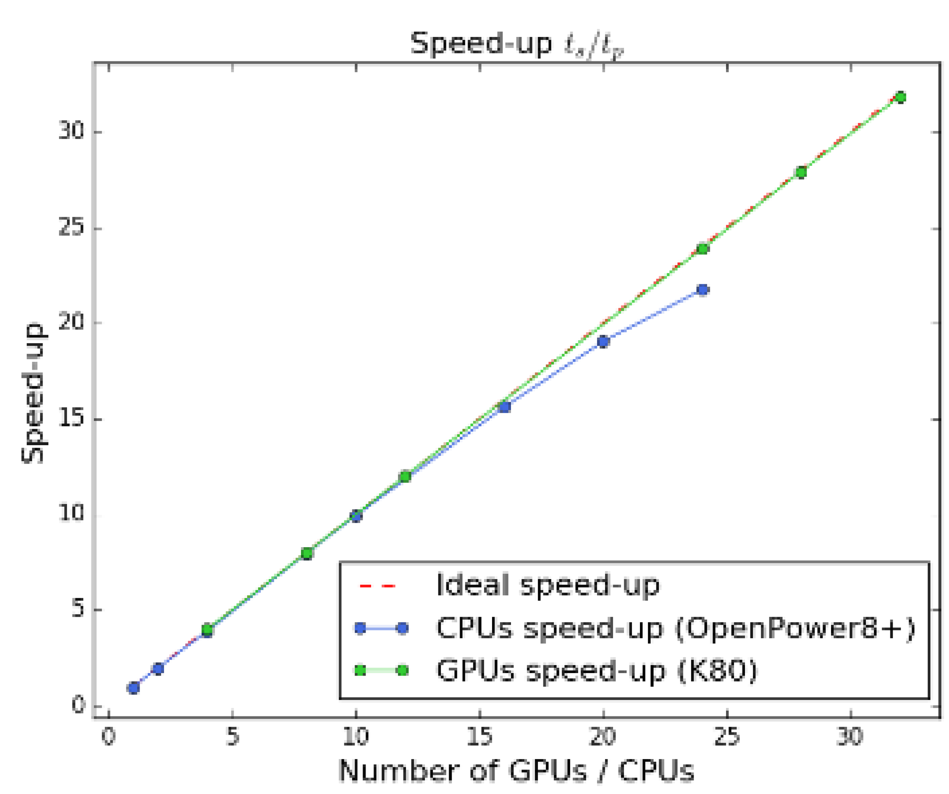
MEM code deployment and performance

The MEM code has been deployed on GPUs platform at CC-IN2P3 Computer Center. Two versions have been compared: MPI C++ and MPI/OpenCL/CUDA (compilation `gcc -O3` and `nvcc`). Up to 8 nodes have used for this benchmark. Each node contains:

- Intel Xeon 2 x E5-2640, 2 x 8 cores@2.6 GHz
- 2 NVidia K80 cards -> 4 Kepler GPUs per node



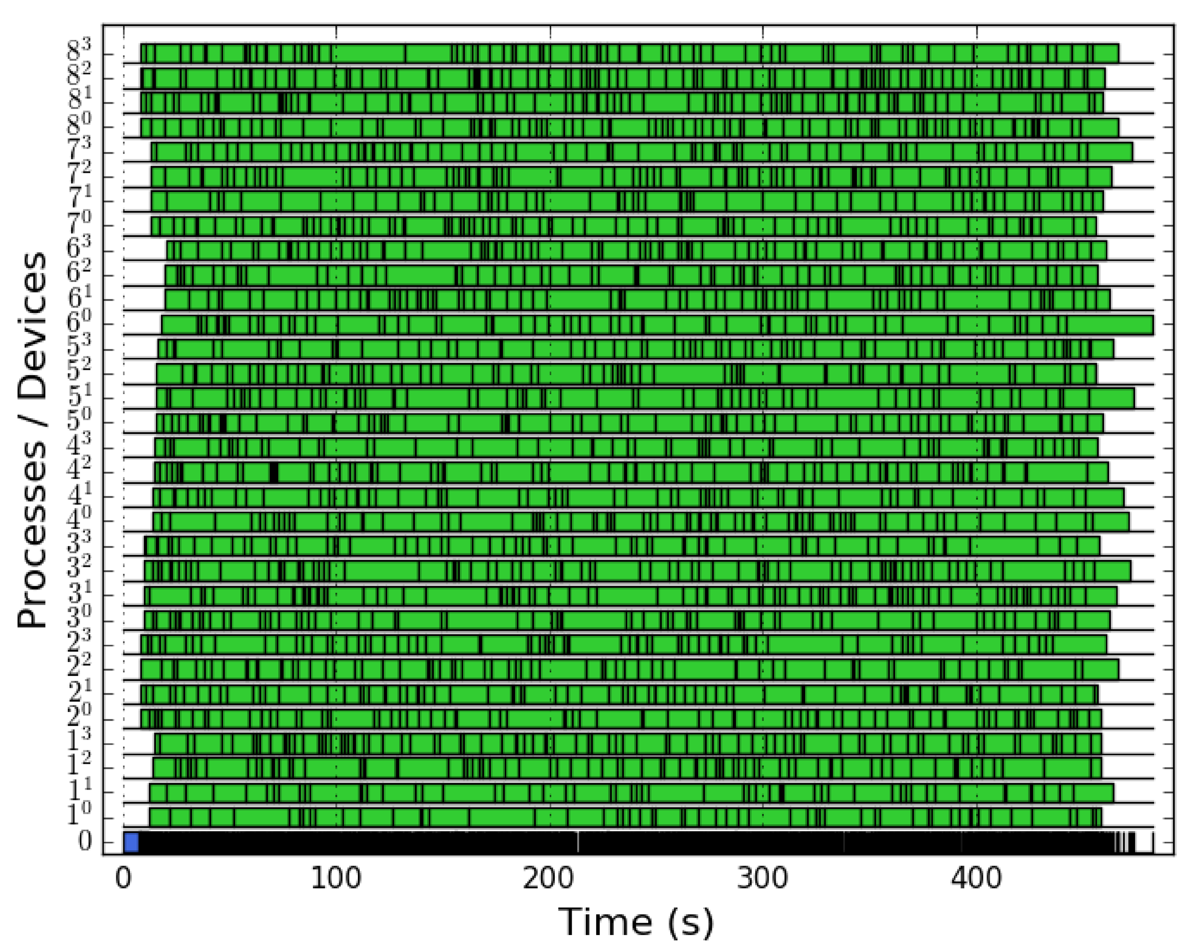
Combination of MPI / OpenCL / Cuda to obtain an important computing power



Study of the GPUs scalability with the MEM code

Given a data set containing 2395 events:

- 55 days of compute time is required on 1 core (or 3. 5 days on a 2 x 8 cores node)
- 450 sec. are sufficient with 32 GPUs (8 nodes)



The great speed-up obtained come from the GPU computing power and from the recoding the C++ to C flat kernels.