ICTEA

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Montecarlo simulation of high energy physics using hardware accelerators

Carlos Vico Villalba, Pelayo Leguina López, Santiago Folgueras Gómez

CMS
Compact Muon Solenoid

EUROPEAN UNION
European Regional Development Fund

PCTI ASTURIAS

Grant PID2020-113341RB-I00 funded by:

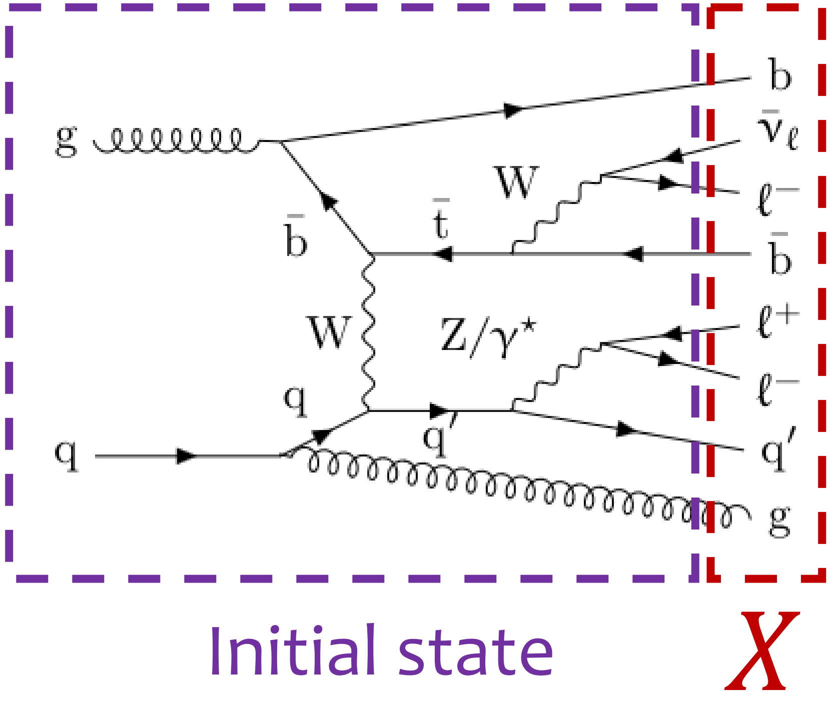
GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

AGENCIA ESTATAL DE INVESTIGACIÓN

GOBIERNO DEL PRINCIPADO DE ASTURIAS

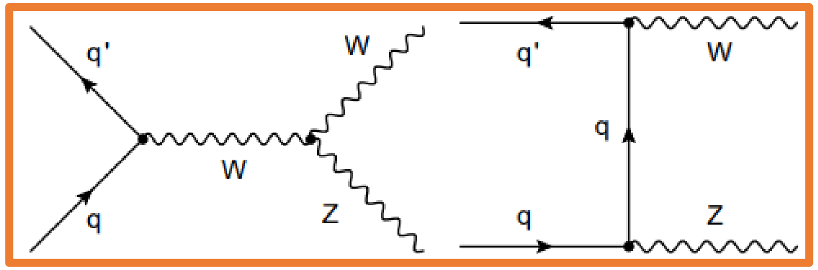
Why do we use MC simulation

- Already covered by other colleagues, just focusing on the premises here.
- **In High Energy Physics (HEP)** we want to access the structure of the proton at ~TeV scales.
  - In order to do so: we built the LHC $(pp \rightarrow X)$
  - Moreover, we built large detectors such as CMS to detect X.
  - From X (Final state) we infer the physics of the proton (Initial state).
- **Problem: in data, where's the X coming from?**
  - Same final states can come from multiple initial states.
- **Solution: use Montecarlo (MC) simulation**
  - Generate hundreds of millions of final states from a given initial state (signal).
  - Do this for all the initial states that can lead to the same final state (background).
  - In a sufficiently large dataset, data should behave as what the standard model (SM) predicts as signal + background.

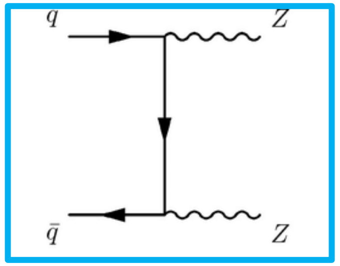


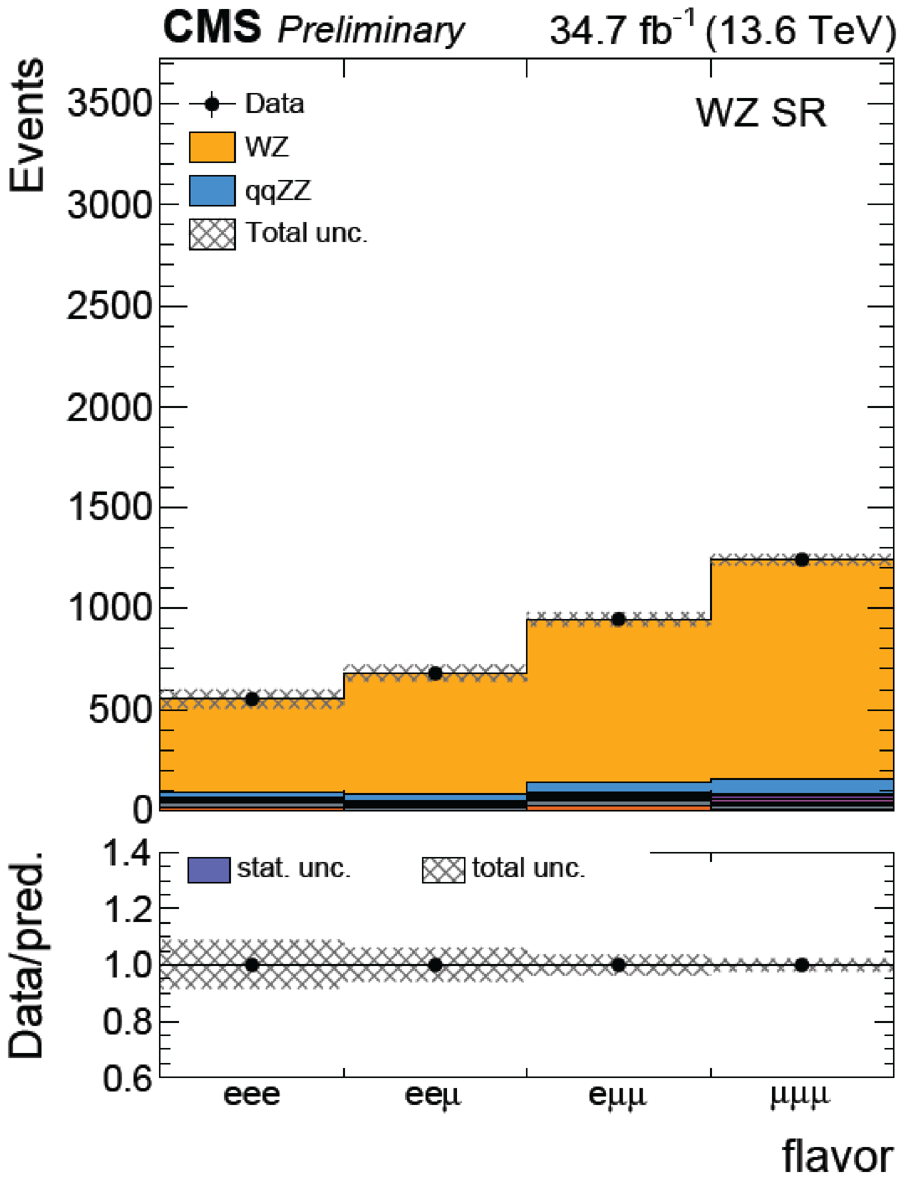
- From Miguel's talk: WZ analysis.

Signal (WZ)



Background



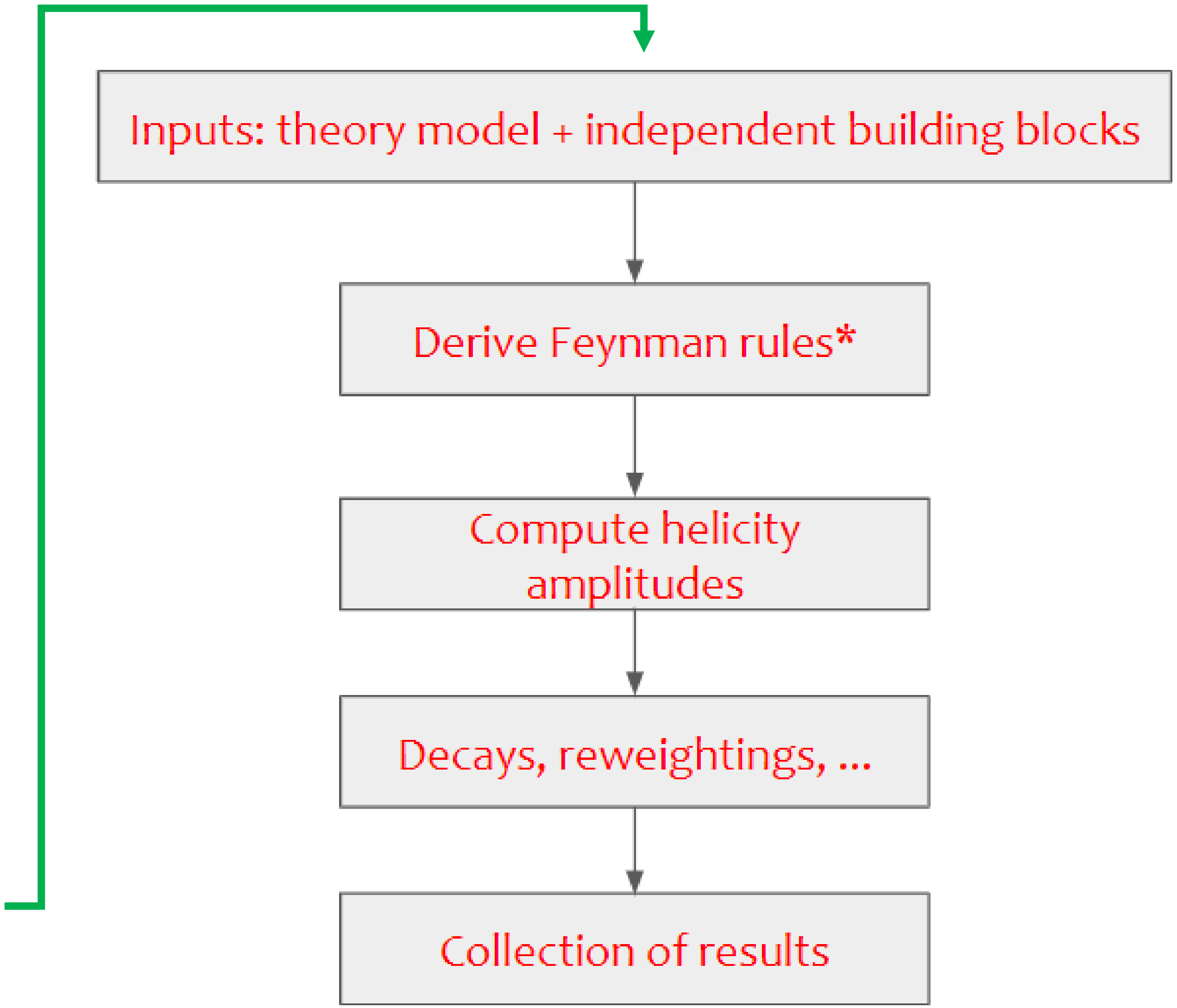


- In data, we don't measure **directly** W and Z bosons → we measure their decay products.
  - So we measure e.g. $\ell^{\pm}\nu\ \ell^{+}\ell^{-}$
  - But those 3ℓ and ν may come from a WZ initial state or a qqZZ final state…
- By comparing WZ MC + ZZ MC to data, we learn most likely the 3ℓ come from a WZ initial state (with our selections).
- But there's a component of ZZ in our data which is not entirely negligible.

Take-home message: we need MC simulation in HEP.

What are MC generators

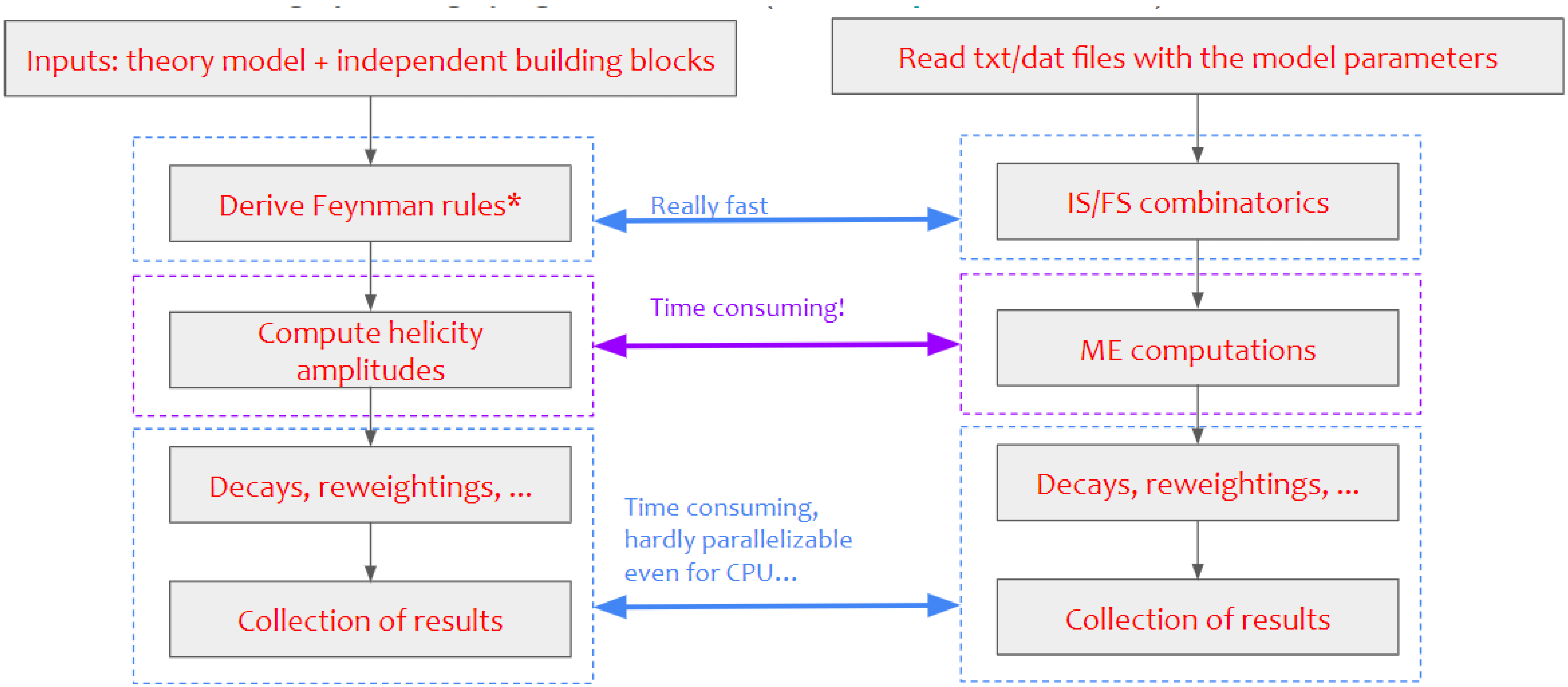
- Starting with the real contents of the talk…
- How do we **generate** MC? What are **MC generators**?
  - Essentially, software packages that compute matrix element calculations.
- There are several on the market. The most common ones:
  - Pythia8: up to leading order (LO) in QCD.
  - Herwig7: up to LO in QCD.
  - Madgraph5_aMC@nlo: up to next to leading (NLO) order in QCD
  - Sherpa: up NLO in QCD.
  - Powheg: up to NLO in QCD.
- Typically MC generator (e.g. madgraph) software start from very basic inputs from the user, and write (typically) fortran code to compute the matrix elements, taking into account all the theory ingredients one needs to do so.



* At NLO it also computes at least UV counterterms, but I personally don't think we should think at NLO for the moment.

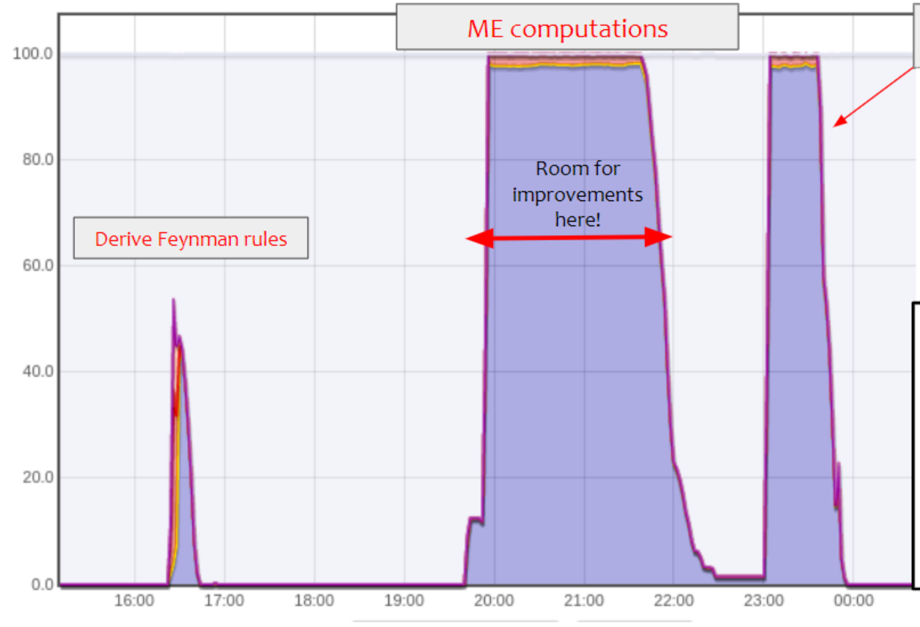
MC generators from a computational point of view

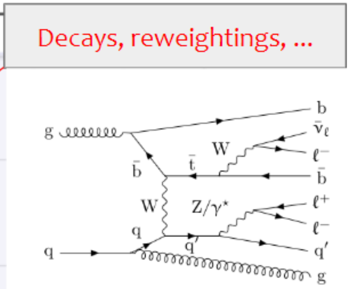
- From a computational point of view, each step of the generation has a "computational interpretation".



How do MC generators run

- Up until very recently, all MC generators were run on CPU.
  - Not the best solution for parallelized computations.
  - Running times not good.





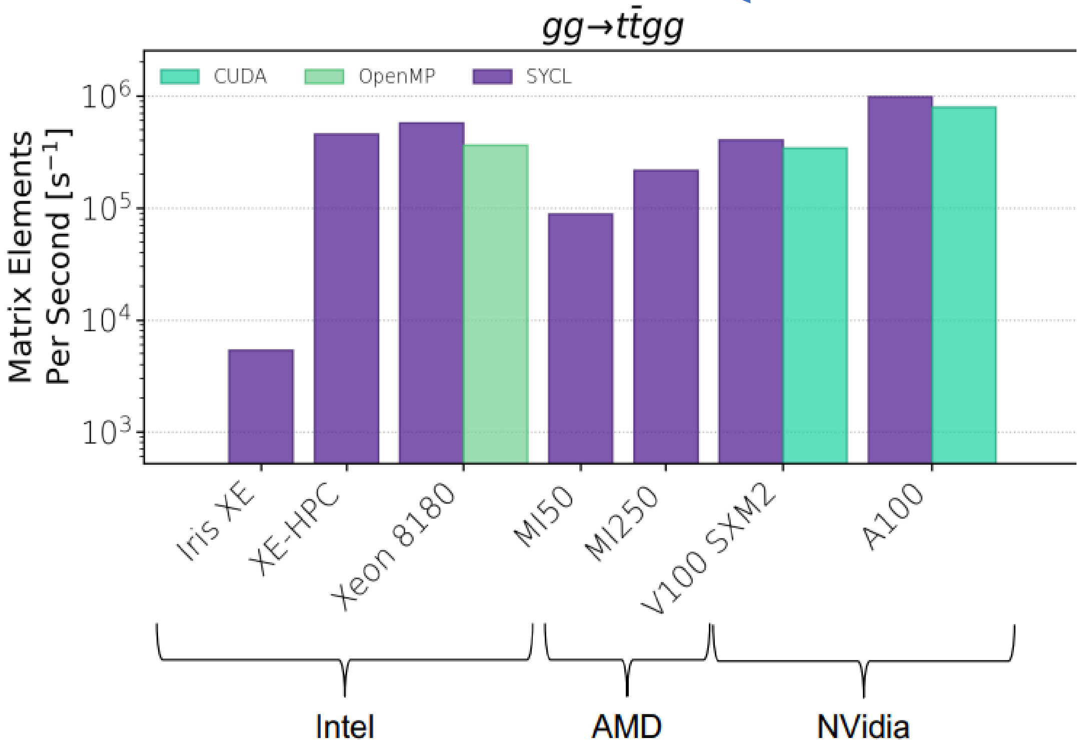
- "Simple" process
- ~2hrs in 64 cores (normally we don't use such big machines for processes like this one)
- Integrating an accessible corner of phase space (mll > 30 GeV)
- Below 30 GeV, process is undoable because computations do not converge quickly and gets kicked out due to walltime errors.

- Biggest problem is not one week to compute all matrix elements.
- Biggest problem is that many times the job gets killed because it takes too much time to run.
- This can significantly impact when the MC simulations are available for experimentalist to use on their analyses…
- CMS produces millions of millions of events **per year** in order to do analysis.
  - So any delay of 1 week, can result on a month-scale delay (or even more).

So this is a real problem!

Our project

- Very recently, Madgraph5_aMC@NLO authors deployed a version of their code that can run on GPUs.
- This version significantly improves computation times (see this talk).



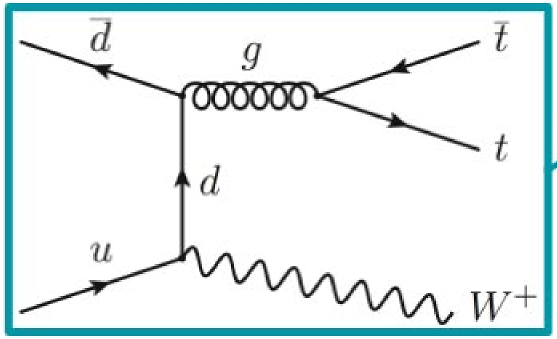
So our idea is: can we do this on hardware based accelerators?

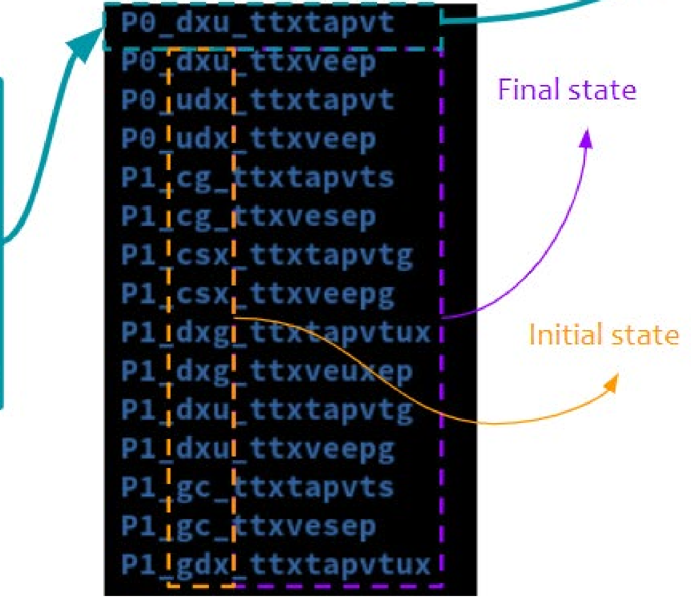
- **FPGAs** are:
  - **Highly** parallelizable
  - In some cases not as fast as GPU.
  - But less power consuming.
  - Hardware based! really versatile.

See talk by P. Leguina.

Our Project: current status

- The project is still at early stages.
- The first plan is to take an easy process to do (even analytically) with Madgraph5_aMC@NLO and see if we are able to implement it on a FPGA.
- The parallelization comes in the matrix element calculations.





Inside this there's just code that computes the ME for this particular **SubProcess**

The idea is to fit each of this into a FPGA that computes all matrix elements simultaneously

Our Project: where do we plan to do it

PRODUCT BRIEF

AMD XILINX

Versal AI Core Series VCK190 Evaluation Kit

OVERVIEW

Versal® AI Core series VCK190 evaluation kit, equipped with the best AI performance-in-portfolio VC1902 device, is built for designs requiring high throughput AI inference and signal processing compute performance. Delivering 100X greater compute power than today's server-class CPUs, with various connectivity options and standardized development flows, the VCK190 is an ideal evaluation and prototyping platform for cloud, network, and edge applications.

HIGHLIGHTS

Evaluate Versal AI Core Series Capabilities

> Equipped with Versal ACAP VC1902 production silicon

> AI and DSP Engines providing 100X greater compute over today's server-class CPUs

> Pre-built partner reference designs for rapid prototyping

Breadth of Connectivity Options for Rapid Development

> PCIe® Gen4 interface for high compute performance markets

> HDMI for video processing applications

> SFP28/QSFP28/RJ-45 for networking applications

> DDR4 and LPDDR4 memory interfaces

> FMC expansion connectors supporting a variety of optional plug-in cards

Co-Optimized Tools and Debug Method

> Fully supported by Vivado® Design Suite, Vitis™ unified software platform, and Vitis AI

TARGET APPLICATIONS

> Data Center Compute

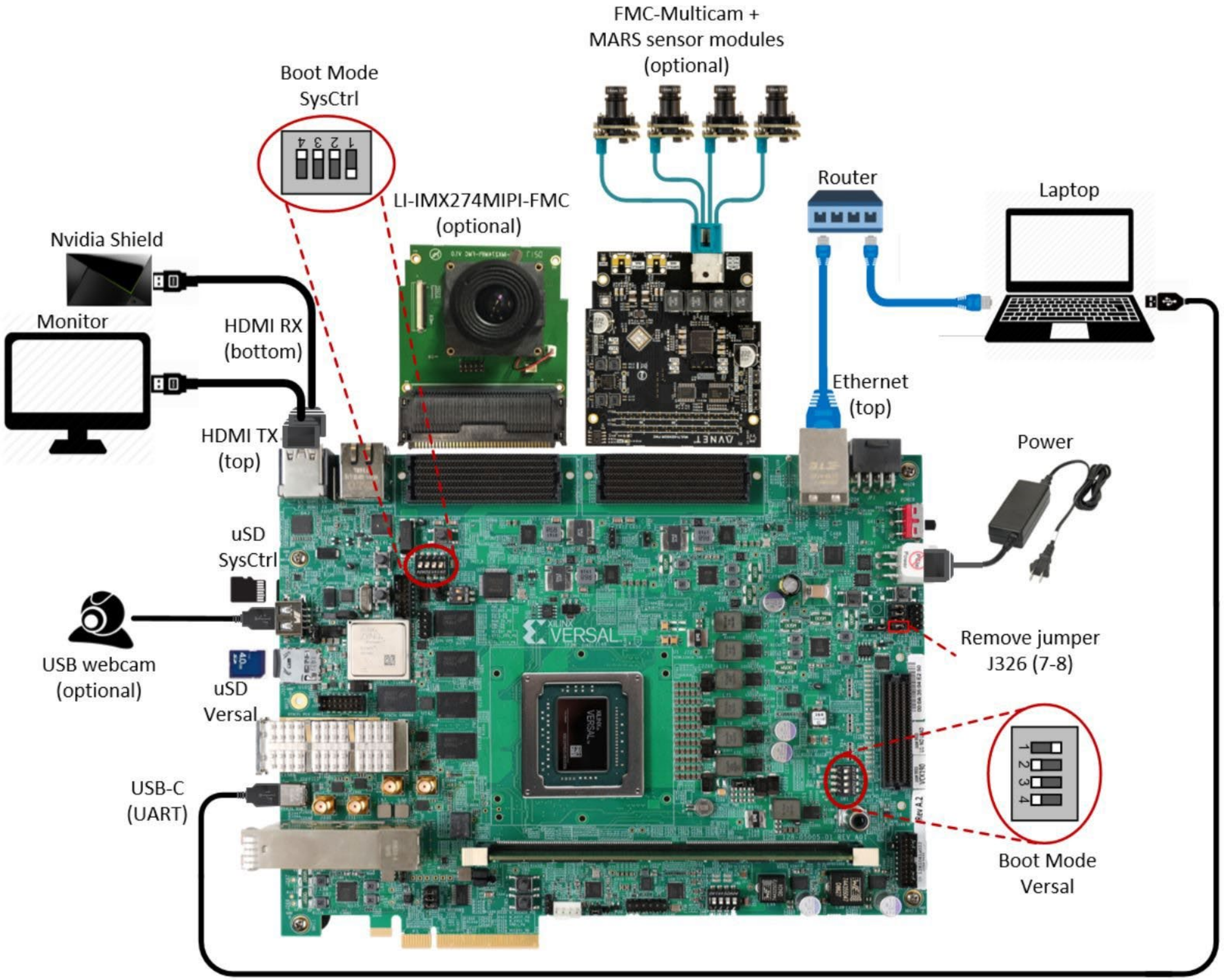
> 5G Radio & Beamforming (DFE)

> Cable-Access (Head-End)

> Wireless Test Equipment

> Automotive/ADAS prototyping

> A&D Radar, Early Warning



Conclusions

- MC simulations are key for the proper development of experimental high energy physics.
  - Sometimes not even theoretically limited…
  - But computationally limited!
  - Good thing here: going beyond in computing is (sometimes) easier than improving on matrix element integration.
- There's already an approach, taken by the authors of madgraph, that is based on GPUs.
- Can we take our expertise on FPGA and see how well it behaves for this task?
  - We will see ;).

Thank you for your attention!