Volunteer computing and virtualization for high energy physics simulations

Predrag Buncic Stefan Roiser Holger Schulz
Ben Segal Peter Skands

2nd Workshop on adapting applications and computing services to multi-core and virtualization

22 June 2010, CERN

High Energy Physics (HEP) simulations

HEP simulations are performed using Monte Carlo (MC) generators

MC generators

- Working horses for e.g. background estimation, calibration ...
- Model physics of e.g. proton - proton collisions
- Rely on phenomenological assumptions $\rightarrow$ many different setups ("tunings")
- Don't always describe experimental data - comparisons needed
- Are time- and CPU-consuming (high statistics needed)

We are working on a web-ressource

- for comparison of different tunings and different generators
- for many important observables and measurements from Sp$\bar{p}$S to LHC era
- This requires a lot of CPU time

Virtual machine allows getting ressources through volunteer computing!

High Energy Physics (HEP) simulations

HEP simulations are performed using Monte Carlo (MC) generators

MC generators

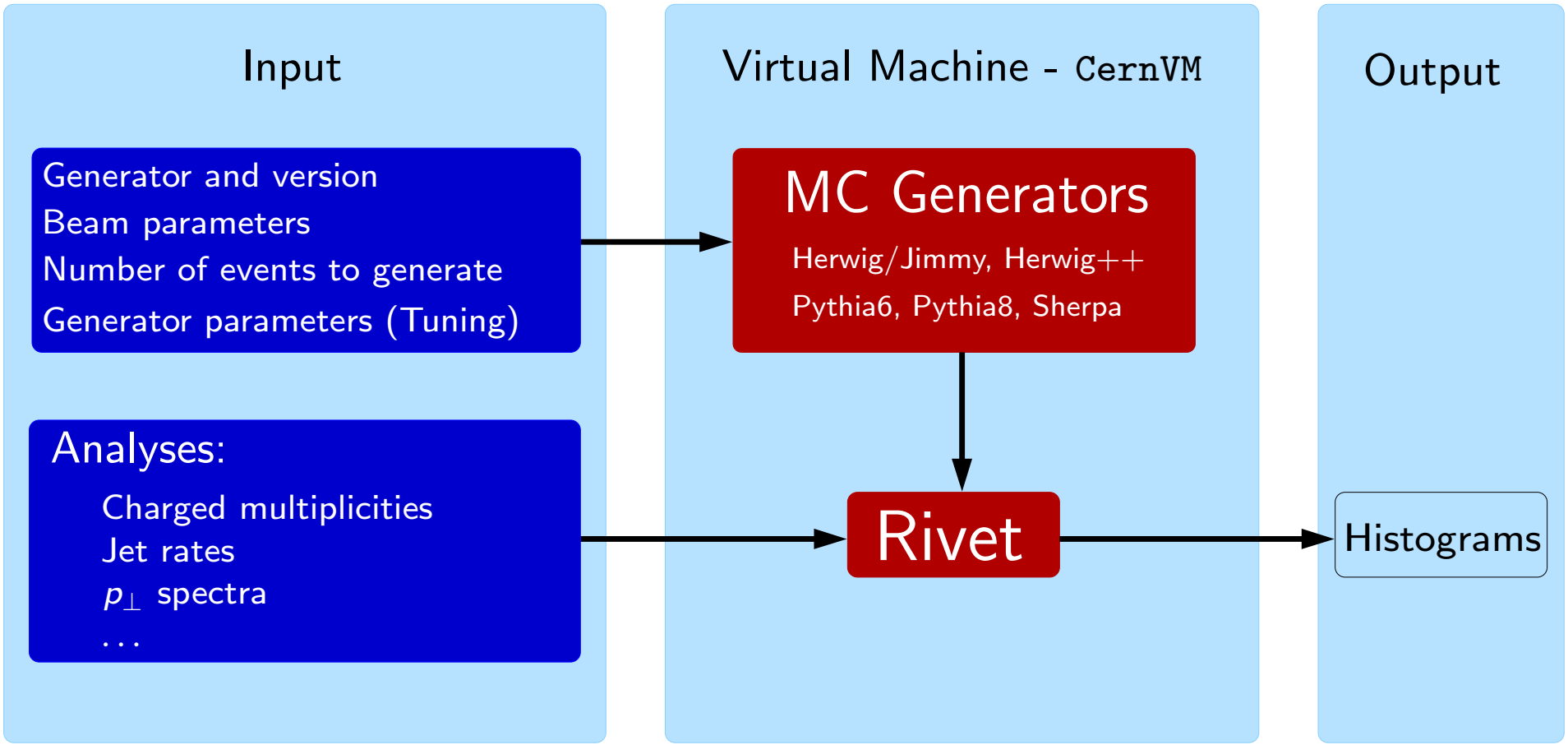
- Working horses for e.g. background estimation, calibration ...
- Model physics of e.g. proton - proton collisions
- Rely on phenomenological assumptions $\rightarrow$ many different setups ("tunings")
- Don't always describe experimental data - comparisons needed
- Are time- and CPU-consuming (high statistics needed)

We are working on a web-ressource

- for comparison of different tunings and different generators
- for many important observables and measurements from Sp$\bar{p}$S to LHC era
- This requires a lot of CPU time

Virtual machine allows getting ressources through volunteer computing!

A basic work cycle



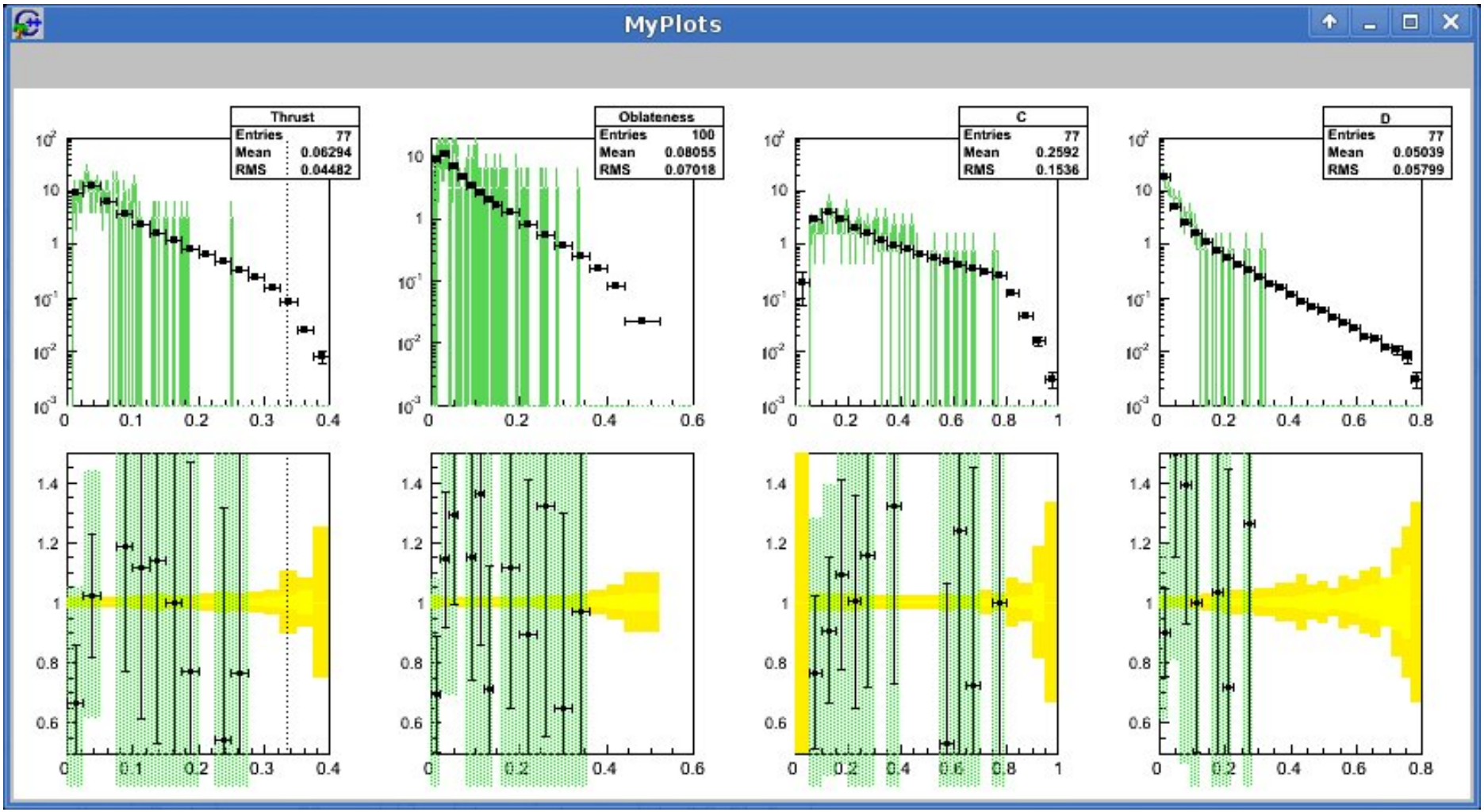
Requirements

Our Monte Carlo application is ideal for volunteer computing

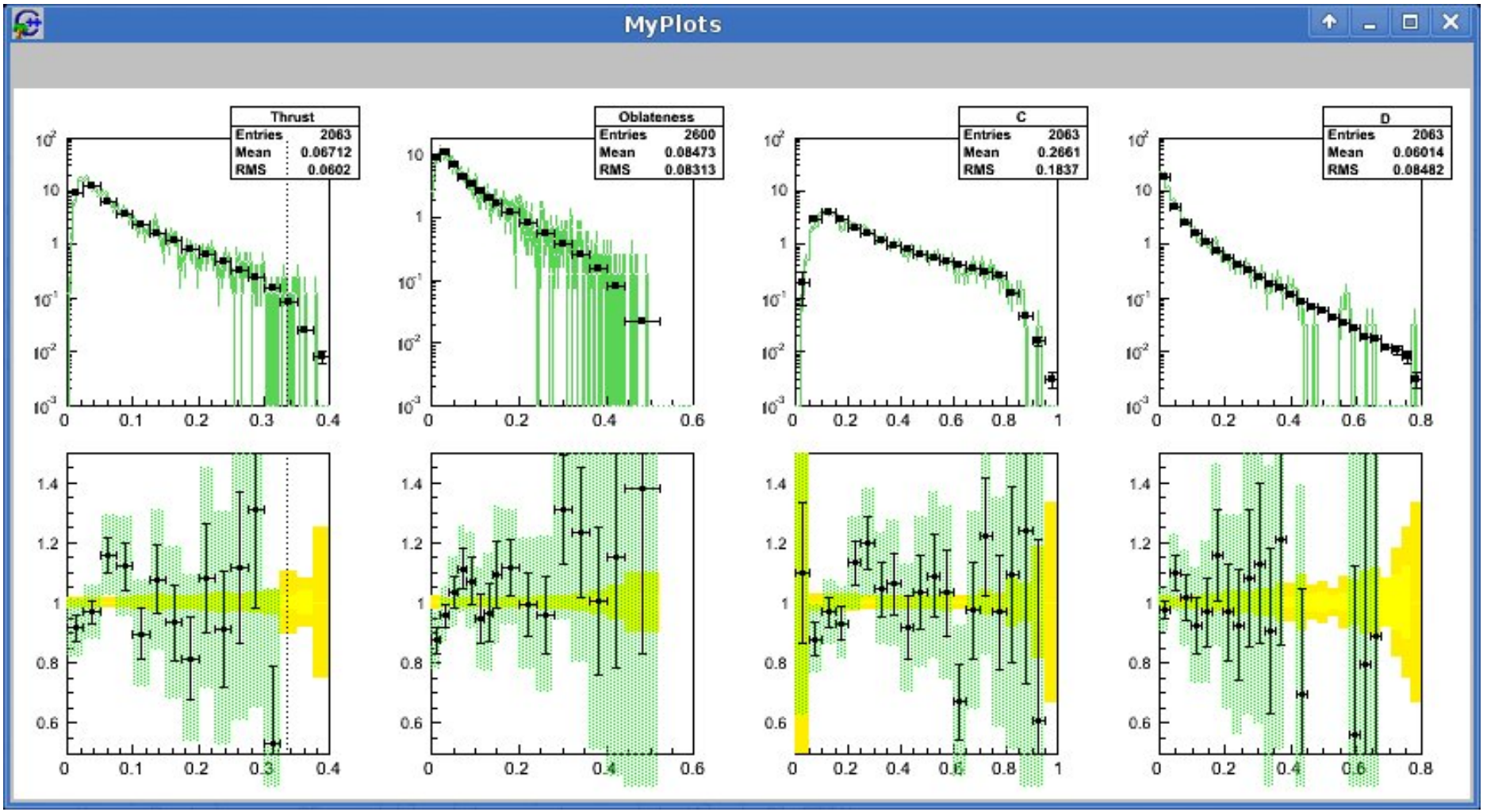
- Required input: small text files $\mathcal{O}(1\text{kB})$
- Size of output histograms: $\mathcal{O}(100\text{kB})$
- $\rightarrow$ network traffic small, CPU is $\sim$ all we need
- Neccessary software available in `CernVM` (`hepsoft`)

It doesn't have to be boring for the users...

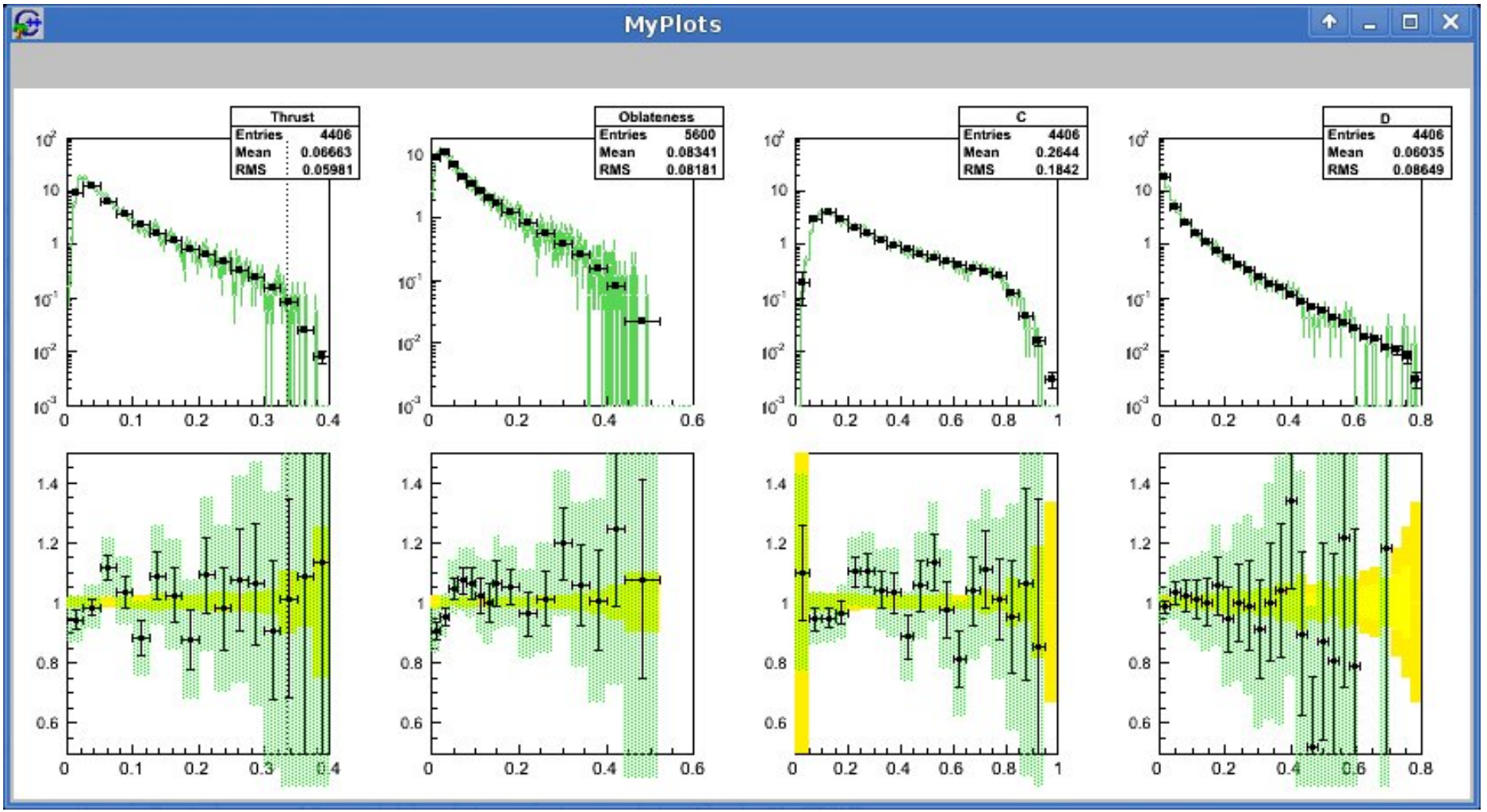
We have a prototype of a 'screensaver' that updates the histograms e.g. every 500 events:



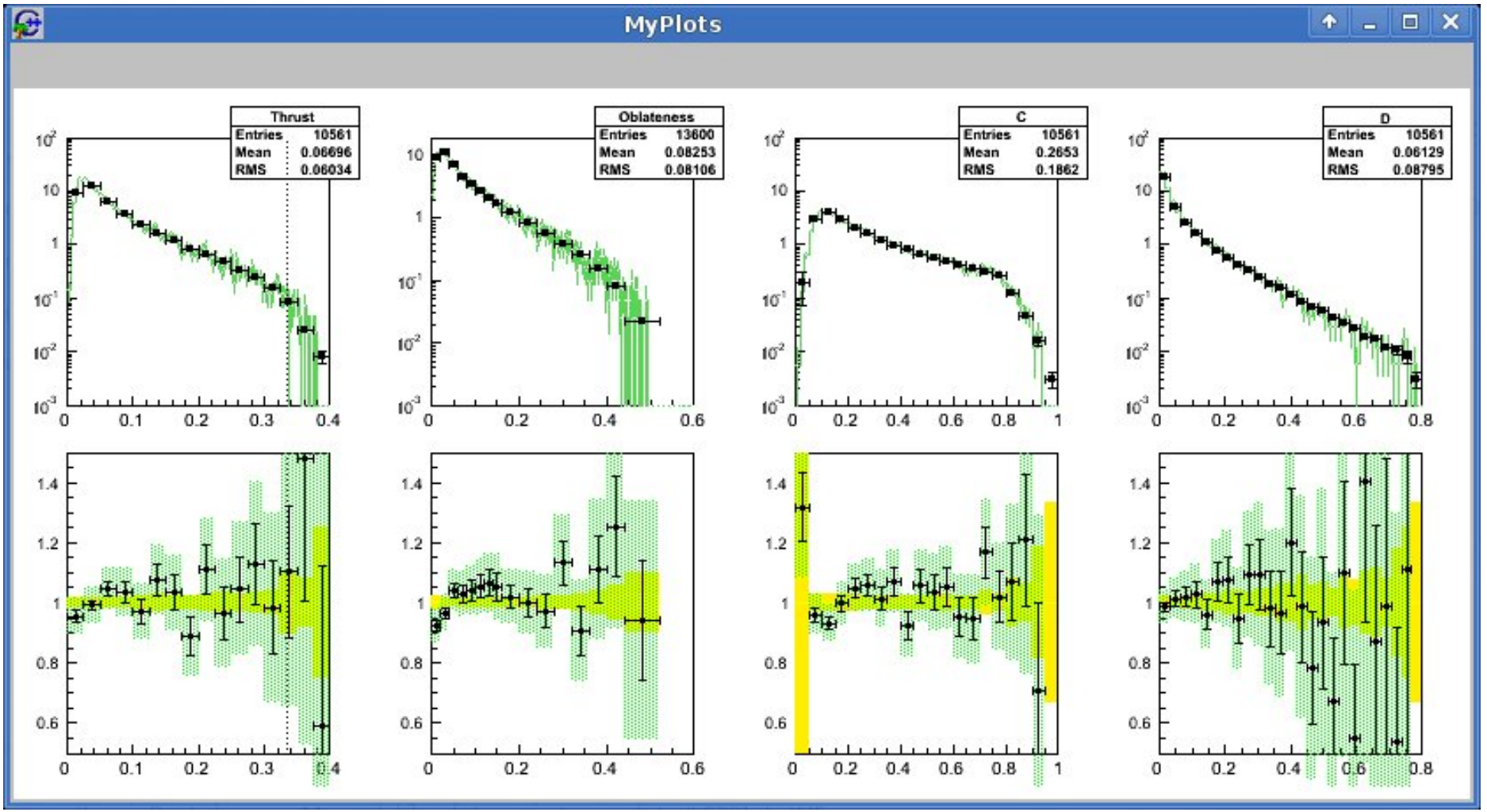
We have a prototype of a 'screensaver' that updates the histograms e.g. every 500 events:



We have a prototype of a 'screensaver' that updates the histograms e.g. every 500 events:



We have a prototype of a 'screensaver' that updates the histograms e.g. every 500 events:



Summary and Outlook

- We have a virtual machine with all the software needed to run Monte Carlo generators (`CernVM`, `hepsoft`)
- This is an ideal application for volunteer computing (low data transfer, small software dependencies)
- The missing bit is the dispatcher
- We have a prototype for a screen-scaver that might encourage volunteers
- One could as well send the virtual machine to a commercial cloud

Thank you!