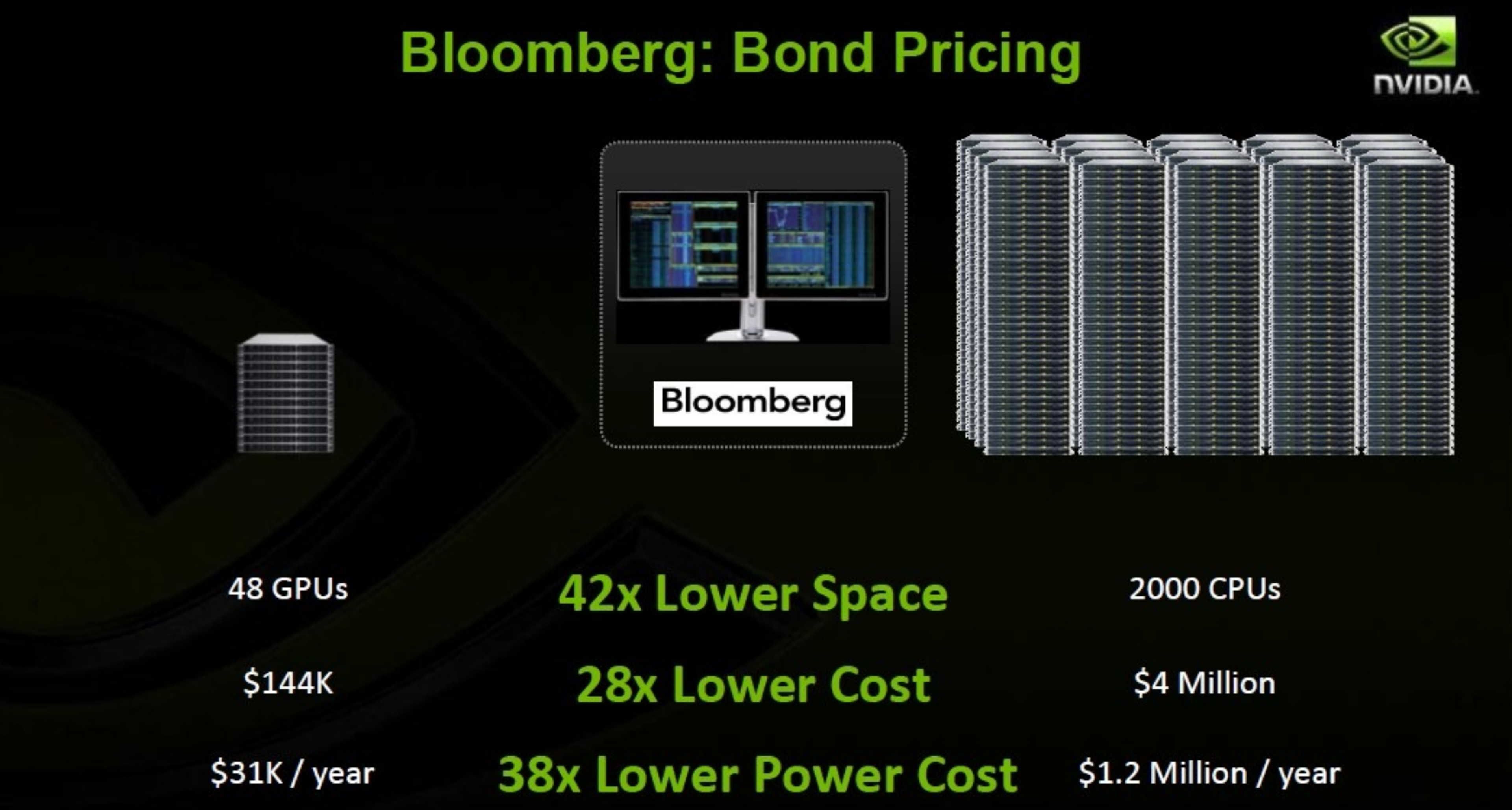
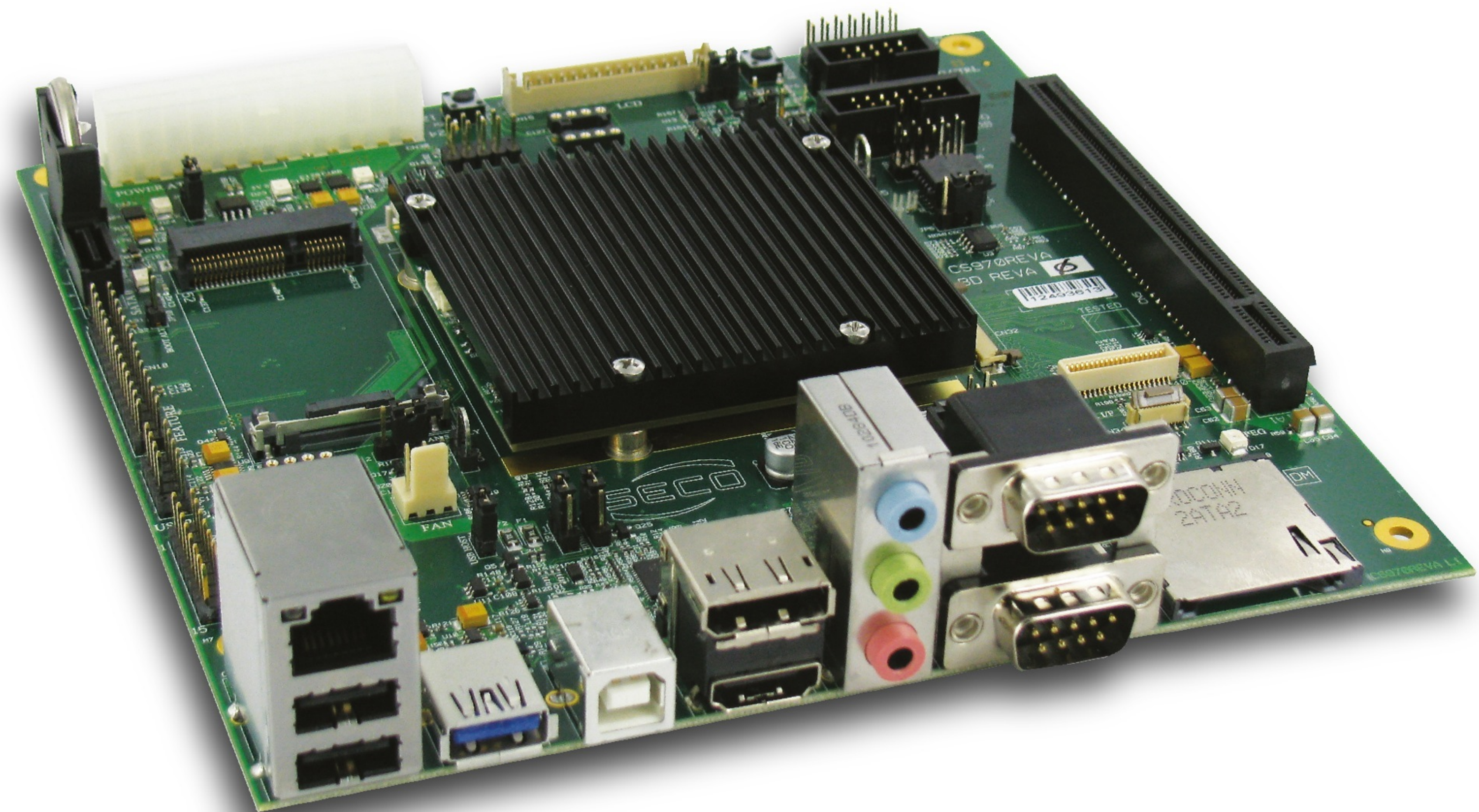
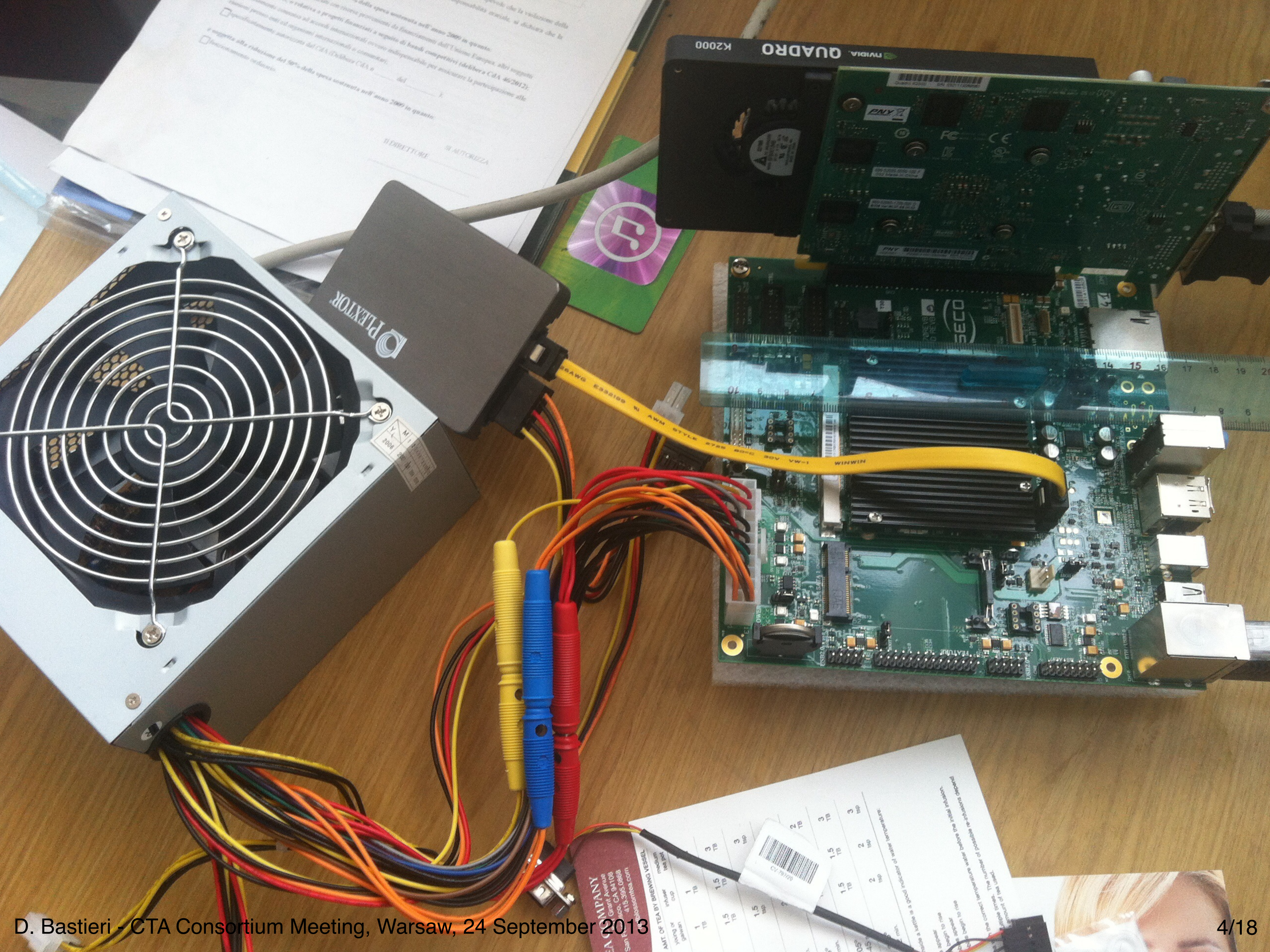
Highest GFLOPS/W, GFLOPS/$



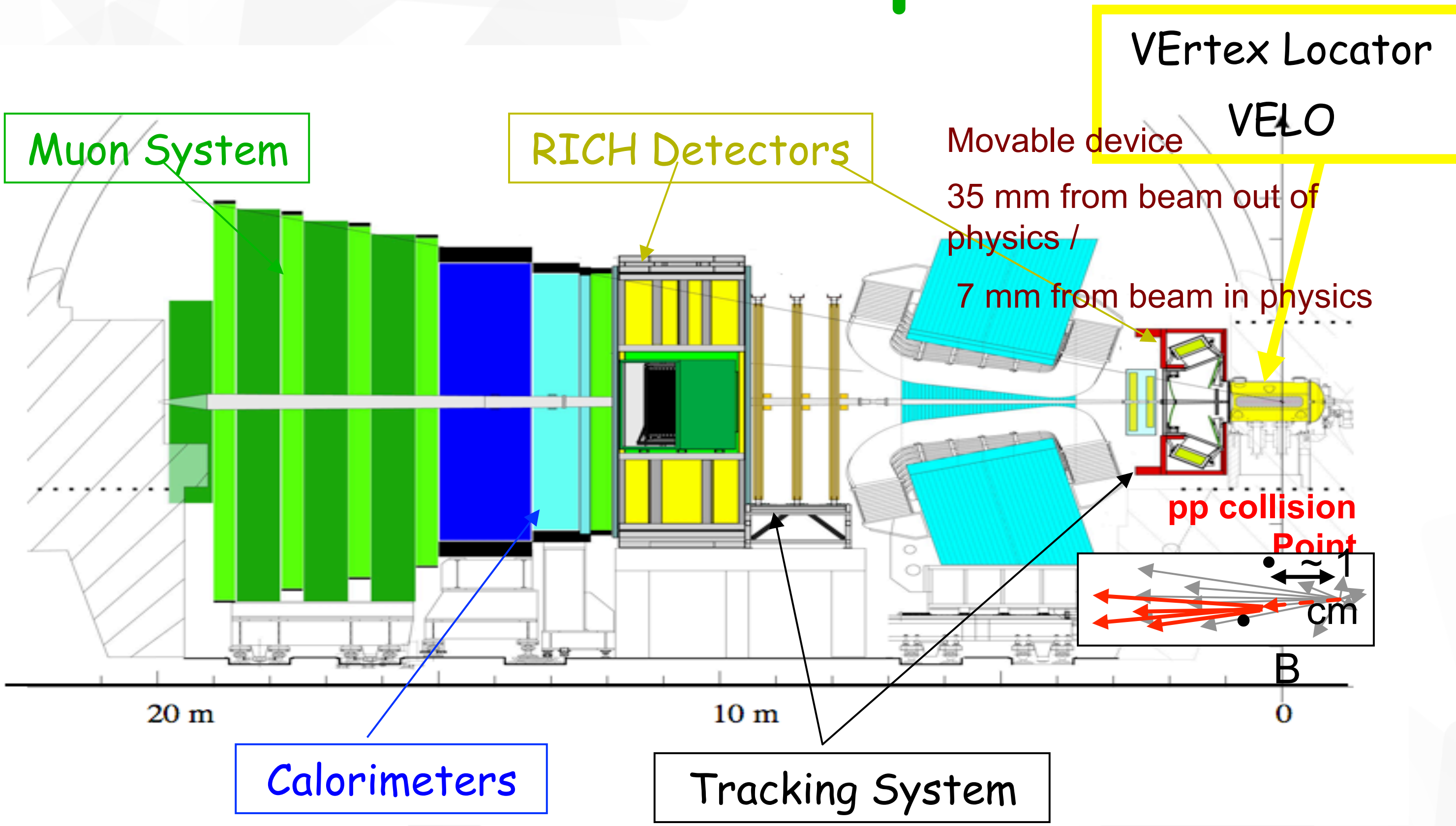
Kayla Devkit by SE.CO.




QUADRO
K2000
PLEXTOR

GPU @LHCb

LHCb Experiment



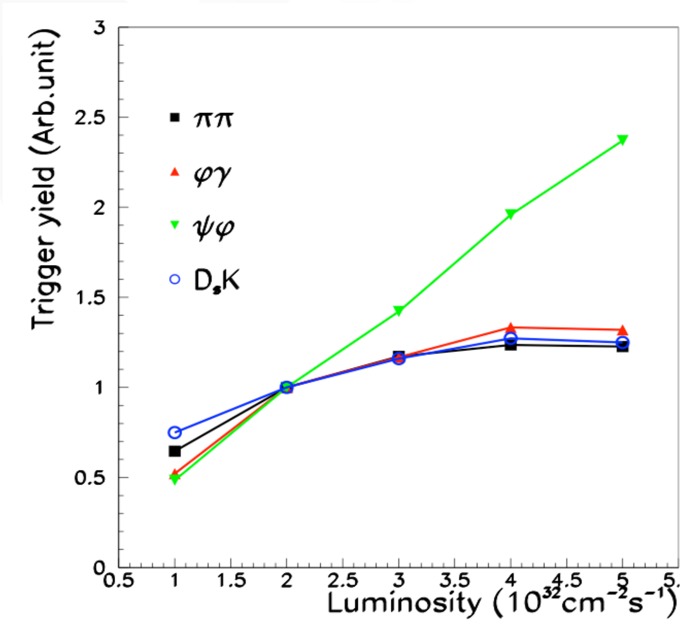
- One of the four LHC experiments
- Designed to study heavy flavor properties

LHC and LHCb

LHC accelerator steadily increasing:

- energy (TeV) 3.5 → 4 → 6.5 → …
- peak luminosity ($cm^{-2}s^{-1}$) 8×10^{33} → 2×10^{34} → …

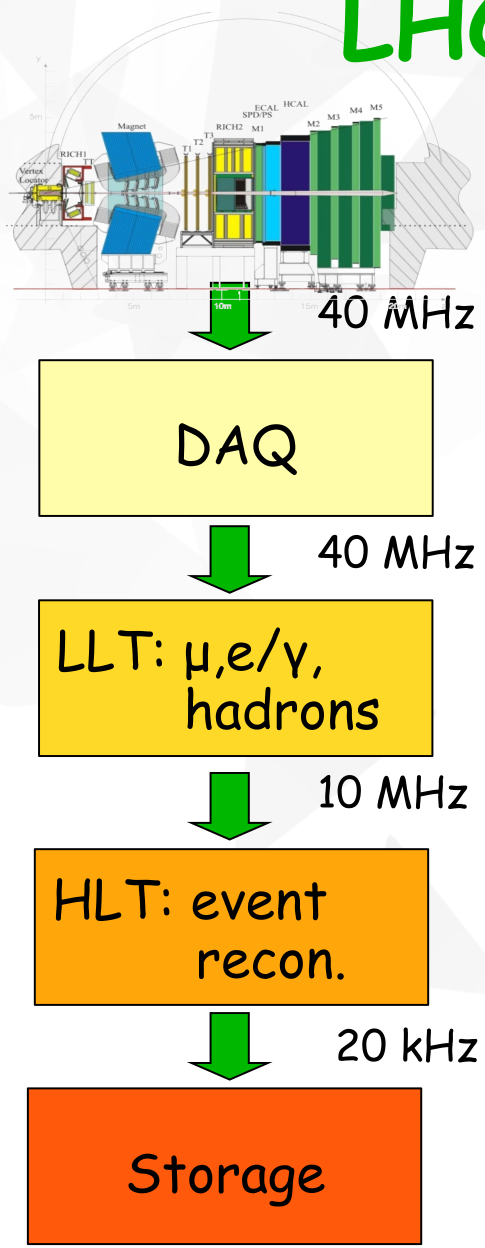
LHCb is currently limited by trigger and Data Acquisition



No gain, but muon based trigger. momentum cut raised to cope with readout.

Upgrade to obtain a physics gain beyond " $1/\sqrt{N}$ "

LHCb Trigger Upgrade



DAQ

- Major upgrade, read out each sub-detector at 40 MHz

Trigger

- Low Level Trigger (LLT)
  - reduce the rate to manageable level according to size of CPU farm (if needed)
- High Level Trigger (HLT)
  - full event reconstruction
  - offline quality level track reconstruction
  - CPU farm for the moment, but proposed to use accelerator, under study GPU

HLT Tracking on GPU

Activities :

1. Porting of the current VELO tracking code to GPU. The goal is to have a tracking code running by the end of 2014 to take data in parassitic mode in 2015 to demonstrate it works.
   To achieve that we have to :
   - Remove the Gaudi dependence to submit kernel to GPU ✔
   - Port the original code in CUDA (re-write) ✔
   - Code optimization
   - Compare performances : track efficiency, purity, fake, times, …

2. Use other algorithms still to reconstruct current VELO tracks.
   Currently using the Hough transform.
   Re-do exactly the previous study.

HLT Tracking on GPU

3. Implement an algorith to reconstruct tracks in the upgraded VELO detector (pixel).
   Identify the more suited algorithm and study the performances.

In collaboration with other group in LHCb perform the full track identification by combining VELO + Forward tracks.

Future possible extension:

- add to tracks PID information at trigger in order to be able to distinguish pion from kaon already at trigger level.

Low-Power Data Reduction and Analysis

Denis Bastieri
CUDA Research Center
Univ. of Padova

CTA Consortium Meeting
Warsaw, 24 September 2013

Data Crunching: recipe from OAR

1) Evaluate pedestal offsets from 2k random events
2) Real data input (2GB = 50 s on MAGIC II @200Hz)
3) Pedestal subtraction
4) signal integration via *sliding window* (short[] -> int)
5) ADC counts (int) -> (× calibration) -> phe (float)
6) phe sorting/clustering/cleaning
7) evaluation of first 10 momenta
8) data output

L.A. Antonelli, S. Buson, D. Gasparrini, S. Lombardi, F. Lucarelli & G. Pivato

Input/Output

Data stored in a SSD, connected via SATA

Tests from July 9th to Sept. 10th ≈ 60 days

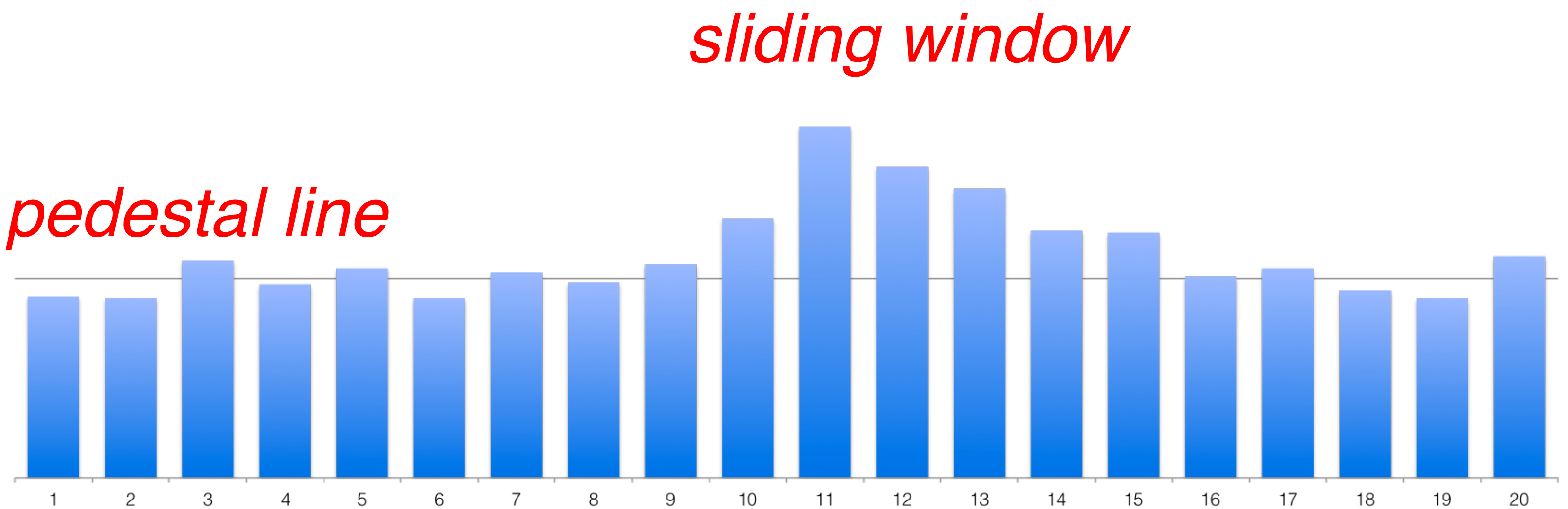
for a 2GB chunk	*Input*	*Output*	*Input &Output*
<time> in s	***15 s***	***15 s***	***35÷45 s***

Sustained data transfer rate: 1 Gb/s

pedestal/calib

1) Evaluate pedestal offsets from 2k random events => no impact on overall timing of the data crunching.

3) Pedestal subtraction
4) signal integration via *sliding window* (short[] -> int)
5) ADC counts (int) -> (× calibration) -> phe (float)



ped/cal timing

4) Integrate over the *sliding window* (short[] → (int) → float)

3+5) exploit fma.s $0 = $0×$1+$2

Virtually no difference in timing between pedestal subtraction and pedestal subtraction + conversion [cts → phe]

Typically 25-30 s.

END OF THE TIME BUDGET!

clustering on K2000

Transfer CPU <-> GPU dominates: $\mathcal{O}(30s)$

30s-budget can accommodate also for ped/cal

1) pxl sorting

2) set hi threshold

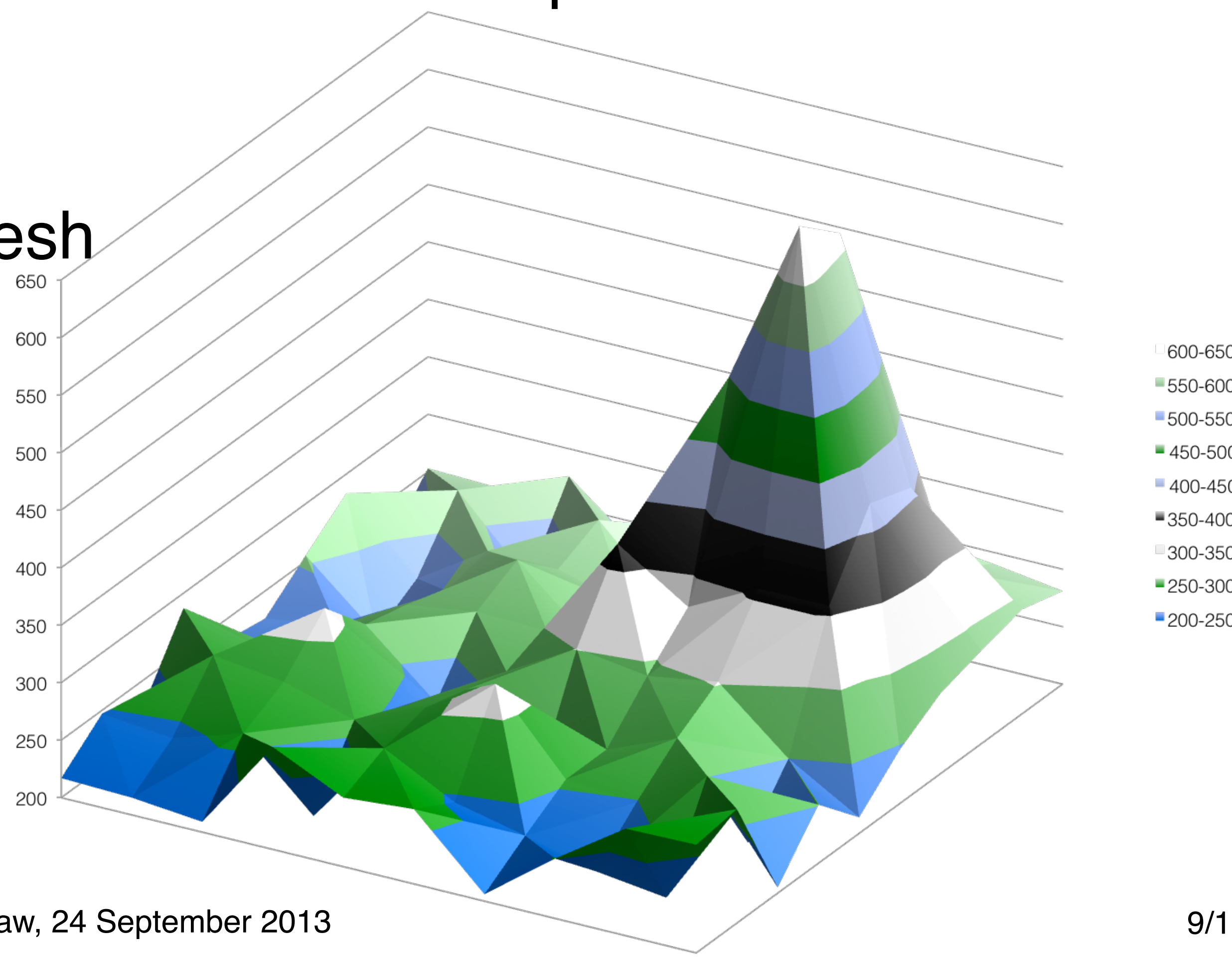
3) check NN > lo/thresh

4) else at zero

5) evaluate first
   10 momenta

12V rail ≤ 1.6A

=> $P \leq 20W$



Data Crunching: summary

1) Sustainable I/O rate of 1 Gb/s. Beware of conflict!
2) Output is much smaller than input: Forget it!
3) @2GB/50s just enough time for ped/cal proc. by ARM, presumably at 5W
4) Cleaning+10 momenta should run on GPU (20W)
5) If ext. GPU (+20W), much computer power unused
6) ARM (Tegra3) should run at 5W, meas. undergoing

Università degli Studi di Padova
Dipartimento di Fisica e Astronomia "Galileo Galilei"
Laurea Magistrale in Fisica

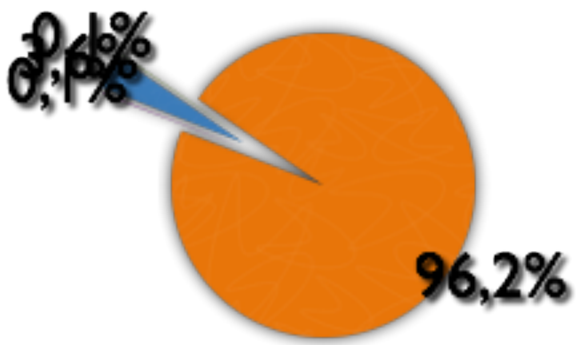
The imprint of new physics in AGN cutoffs

author: Andrea Pigato

supervisors: dr. Denis Bastieri
dr. Sara Buson

Computational Cost

- Selection (~5 minutes)
- Livetime cube (~2,5 hours)
- Exposure map (~3 minutes)
- Fitting & SED (~70 hours ext.)



3 days per source for ~5 years worth of data

ScienceTools-09-31-00 · 2 × Xeon E5620 · 24 GB RAM

Maximum Likelihood Approach

- Parameters estimation through maximization
- Hypotheses testing through Wilks' s theorem

Null hypothesis max likelihood, *h* parameters

$$\mathrm{TS} = -2\ \log \frac{\mathscr{L}_0}{\mathscr{L}} \underset{N\to\infty}{\to} \chi^2_{m-h}$$

non fixed parameters

Alternative hypothesis max likelihood, *m* parameters

Unbinned Likelihood

Poisson statistics: $p(n,\lambda) = \dfrac{\lambda^n e^{-\lambda}}{n!}$

Total number of predicted photons

$$\log \mathscr{L}(\{\alpha_k\}) = \sum_{i\in P} \log J(E,\vec{p};\{\alpha_k\}) - \Lambda_{tot}(\{\alpha_k\})$$

Set of bins with an observed photon

Algorithm details

For each source: assign a spectrum
(functional shape + free/fixed parameters)

use MINUIT to minimize *-log(Likelihood) (*~10^2 iterations)

→ For each observed event ($10^4 \div 10^6$)

→ For each source of the sky model (~10^2)

→ Compute the probability that the event originated from the source given the source spectrum

→ Sum all *log*(probabilities) to obtain the new *Likelihood*

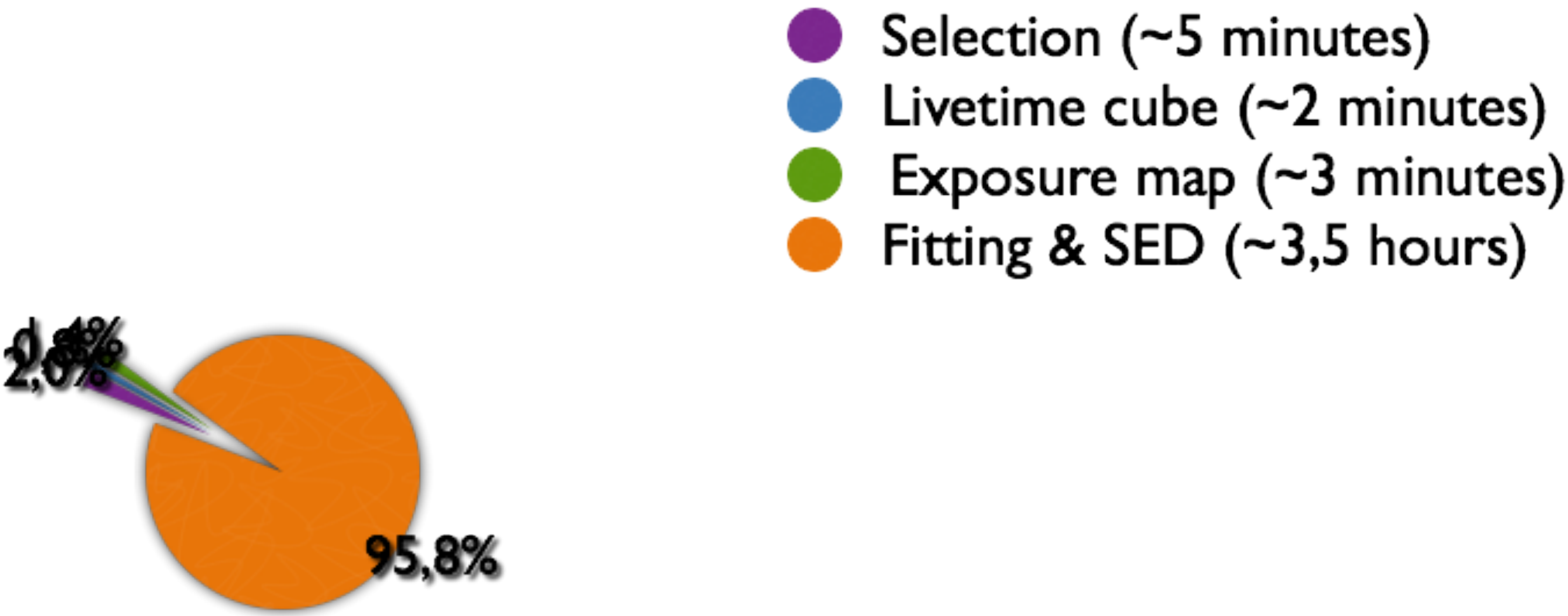
Once *-log(Likelihood)* has been minimized

→ Remove one source at a time from the model

→ Refit

→ Use Wilks' theorem to assess the *TS* of the source

Computational Cost



from 3 days to **4 hours** per source for ~5 years worth of data

New Pipeline · NVIDIA S2050 · 3 GB RAM

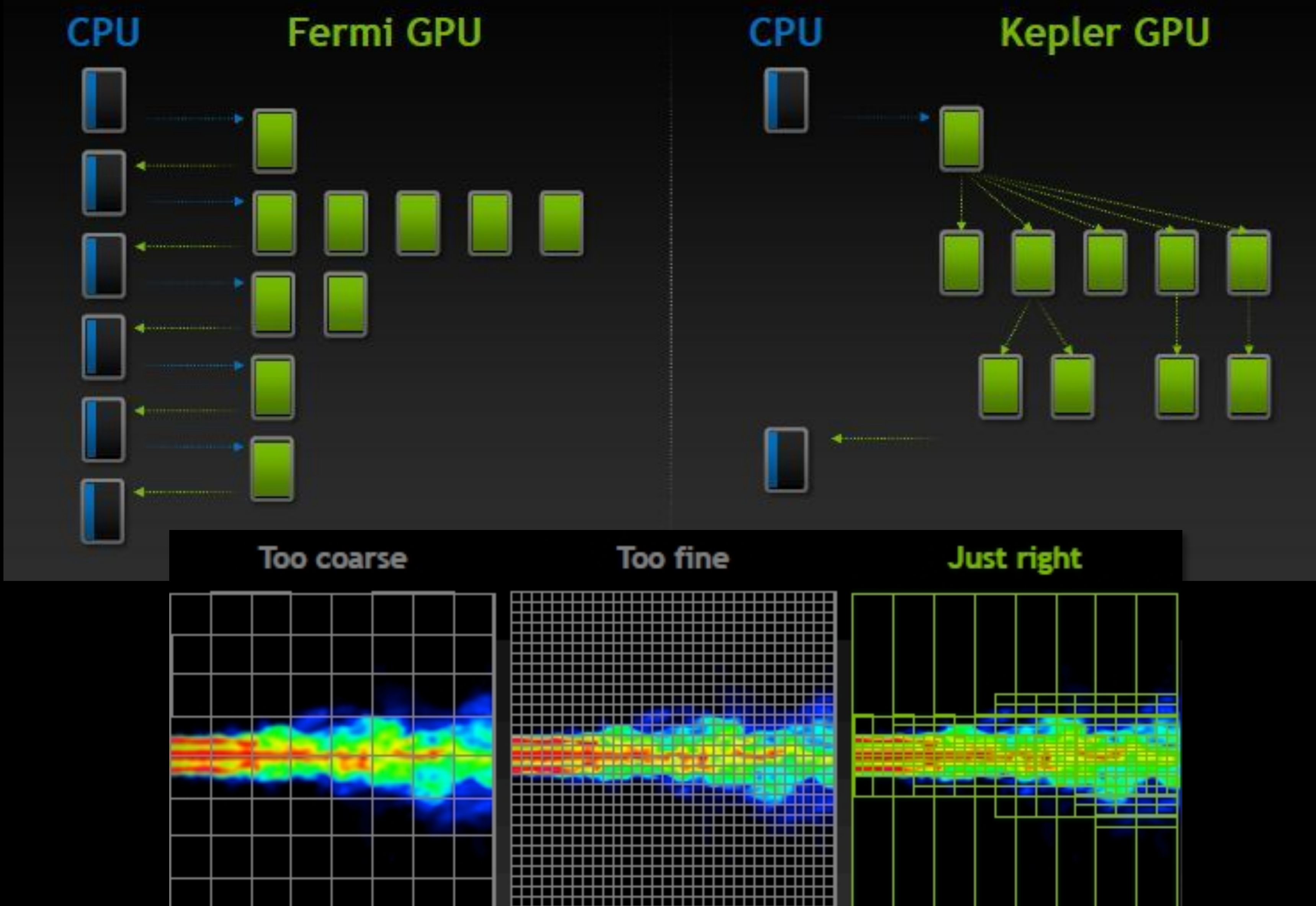
Future developments

- *What about higher level data analysis?*
- de Naurois & Rolland arXiv:0907.2610
- Template library
- Maximize the likelihood, given the data
- *How to reduce CPU<->GPU data transfer?*
- Levenberg-Marquardt *vs.* MINUIT
- Minimizer resident in GPU memory

Giovanni Lamanna

Dynamic Parallelism

GPU Adapts to Data, Dynamically Launches New Threads



Conclusions

1) Low-power data reduction implemented @5+20W
2) data flow @1Gb/s, data processing @~2GB/min
3) Pedestal/calibration feasible on ARM @5W.
4) Additional analysis:
   either reduce processing bandwidth < 2GB/min
   or spawn it to GPU's cores (add 20W)
5) Watt meas. needs some time (and a datalogger)
6) CTA timeframe suits well with *Logan/Parker* (SoC, 10×, FinFET)