Track 8

Performance increase and optimization exploiting hardware features

Niko Neufeld[1] Tommaso Boccali[2]
Amitoj Singh[3] Danilo Piparo[1]

1 Cern, 2 INFN Pisa, 3 FNAL

From 10.000 Km

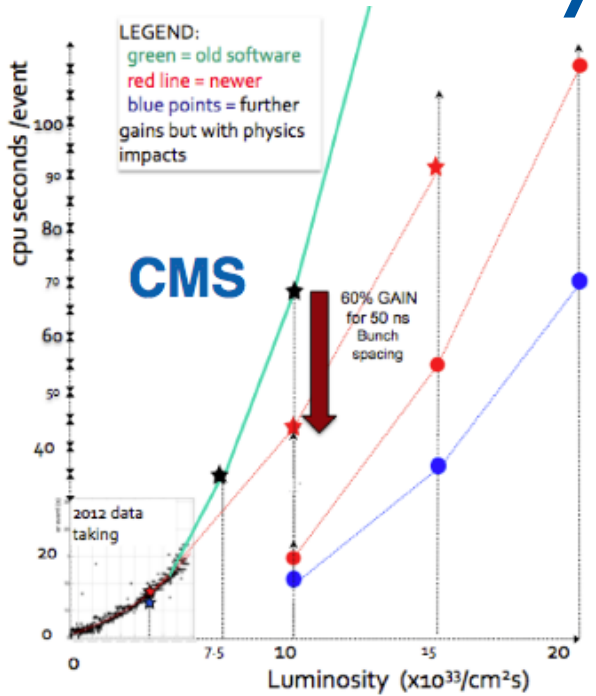
- 40 Excellent contributions
- A variety of topics covered
- Experiment independent techniques and innovations
  - Documented success stories from running and future experiments

Thanks to:

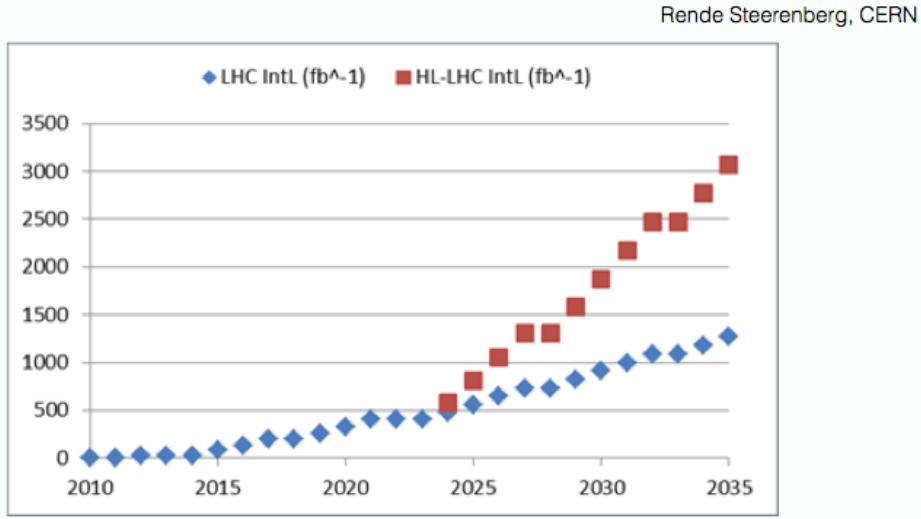
- The authors for making a success of the track!
  - Maybe one observation: late oral cancellation is somehow less than optimal
- The local organisers for the impeccable preparation and coordination!

Why This Track Is Important?

Diversity in Computing Technologies and Strategies for Dynamic Resource Allocation, O. Gutsche et al



Evolution of Computing and Software at LHC: from Run 2 to HL-LHC, G. Stewart



We cannot cope with such a complex environment hoping that our hw resources will increase accordingly

Evolution of existing software and computing models to keep pace with available and future HPC hardware is a must.

Disclaimer

Not mentioned != Not interesting

Themes

1) Strong Scaling

- FPGAs and Accelerators
- CPUs
- IO

2) Throughput and HPC

CHEP2015 Motto:
"Evolution of Software and Computing for Experiments"

Strong Scaling

FPGAs and Accelerators

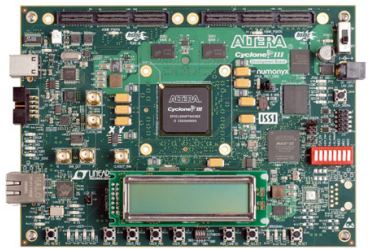
GPUs are now in production

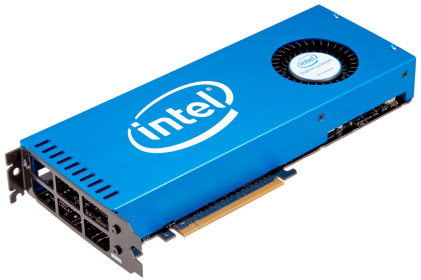
- Mainly online but also event generation, simulation and analysis

Lively R&D ongoing: expand set of applications which use them

Exploitation of FPGAs definitively under study

- Trigger/online systems: perfect candidates



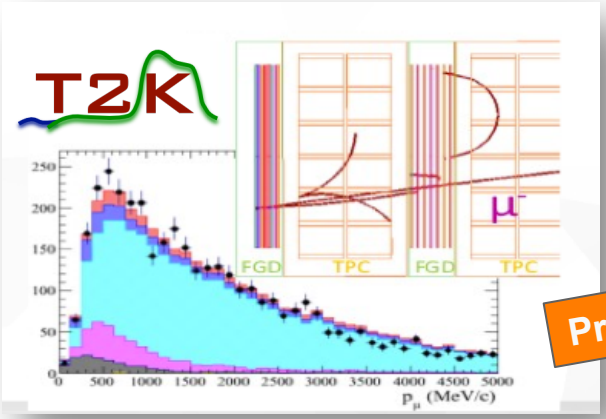


Data Analysis and Beyond

Event-by-Event approach (re-weight)

2800 CPU days -->in 140 GPU days

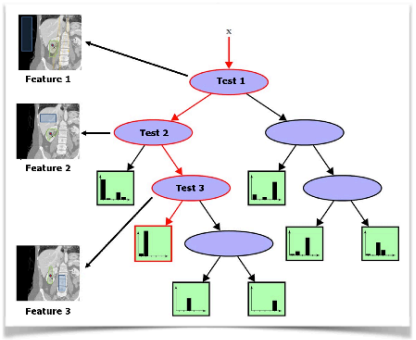
Still room for improvement: dp to sp!

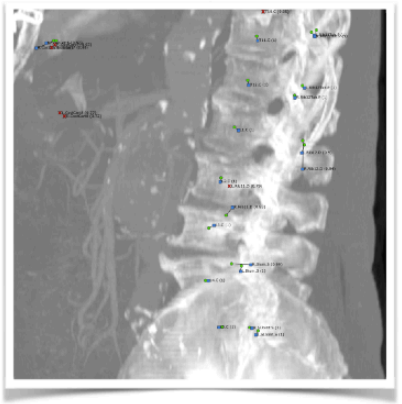


Production

GPU Accelerated Event-by-event Reweighting for a T2K Neutrino Oscillation Analysis, R. Calland

Medical Application!





Process computerised tomography data

threads	traversal time (s)	traversal speedup	execution time (s)	speedup
		80 trees		
1	1687.38	1	1687.56	1
8	213.24	7.9	213.31	7.9
16	134.76	12.5	143.92	11.7
GPU	7.96	212.0	12.18	138.6

R&D

Acceleration of ensemble machine learning methods using many-core devices, A. Washbrook et al.

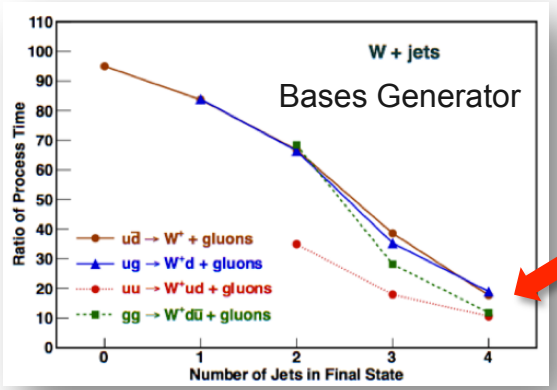
Event Generation and Simulation

Generate MC events on GPU

Port existing generator **Production**

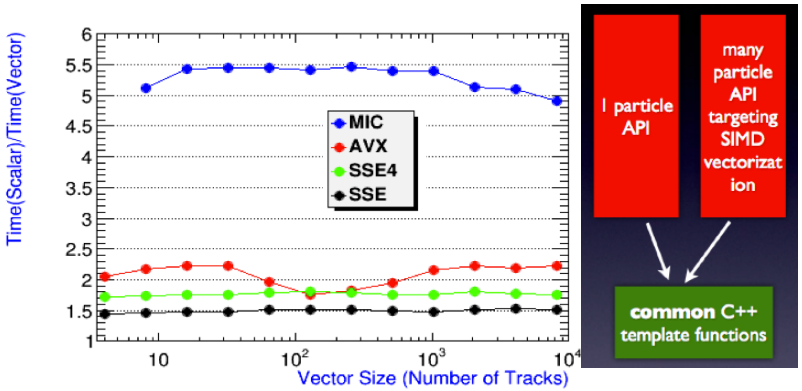
Possible development: dp to sp

Core i7 2.7 GHz
GPU: C2075



Porting / Rewriting

10/20 X



New algorithms: no branching **R&D**

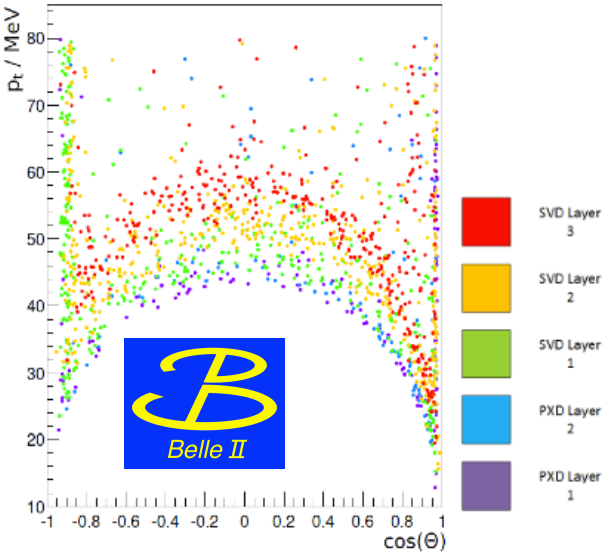
New data structures: vectorization

MIC: Extreme architecture!

Fast event generation on graphics processing unit (GPU) and its integration into the MadGraph system, J. Kanzaki

Detector Simulation on Modern Coprocessors, P. Canal et al.

Trigger and Online – FPGA



Online Analysis of Hits in the BelleII Pixel detector for Separation of Slow Pions from Background, S. Baer

Discard πs with pt<60 MeV

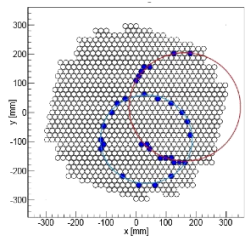
Tracking impossible: not enough hits!

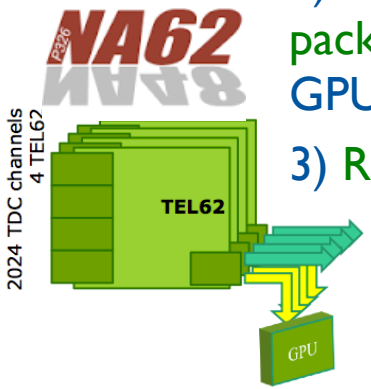
NN approach in a Virtex6 FPGA

Closed-source (Neurobayes)

Perfomance sufficient for deployment!

Ready





Goal: online Cherenkov reco

1) Merge readout streams with custom card

2) 9 to16 bits decompression of packets on FPGA, direct pipe to GPU (no host)

3) Ring reco on GPU

R&D

A multi-port 10GbE PCIe NIC featuring UDP offload and GPUDirect capabilities, A. Biagioni et al.

FPGA: powerful solution, low level programming.

Change model to OpenCL?

Evaluation of 'OpenCL for FPGA' for Data Acquisition and Acceleration in High Energy Physics applications, S. Sridharan

R&D

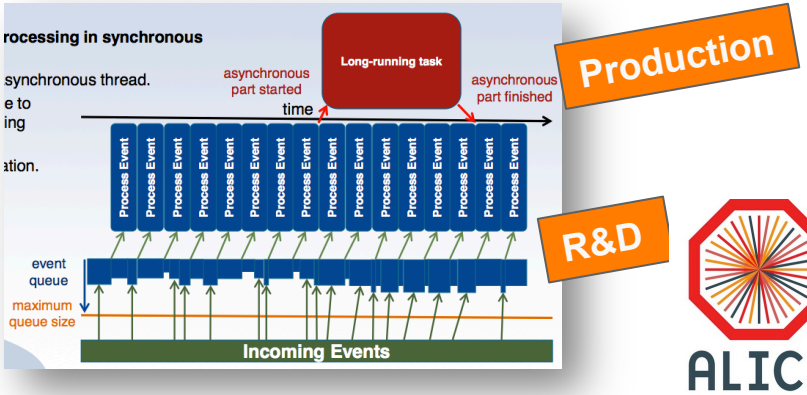
Trigger and Online – GPUs

TPC needs calib. @ online

1)Fast tracking + 2)Async processing (spawn thread)

GPU based tracking also used in Run1

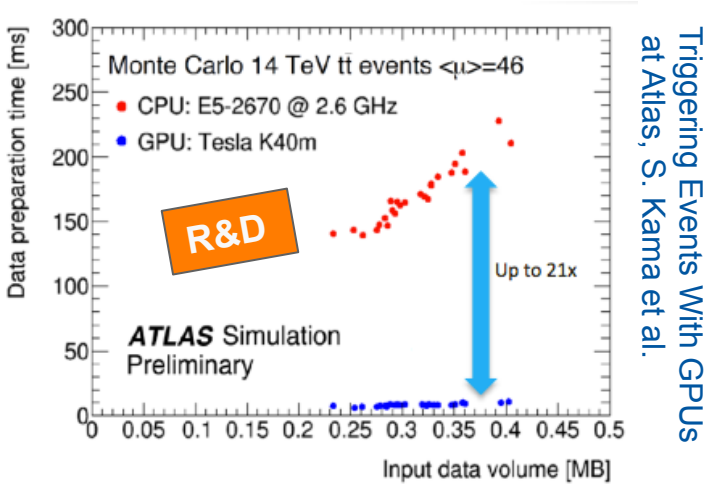
CPU, Cuda and OpenCL versions



Production

R&D

ALICE



Triggering Events With GPUs at Atlas, S. Kama et al.

Client-Server Approach for offloading

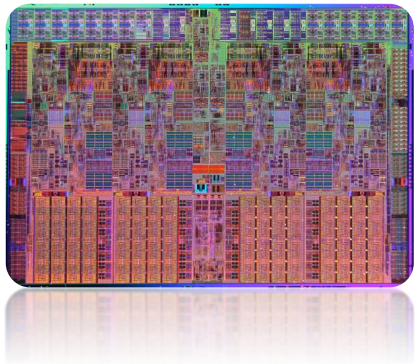
Promising demonstrator

ATLAS EXPERIMENT

Fast TPC online tracking on GPUs and asynchronous data-processing in the ALICE HLT to enable online calibration, D. Rohr

CPUs

- Vectorisation widely adopted in production
- Memory access optimisation is more critical than ever due to NUMA
- Active research in the area of power consumption and efficiency
- Care needed to identify the « right » set of benchmarks



Vectorisation

Vectorisation in some of the LHCb code

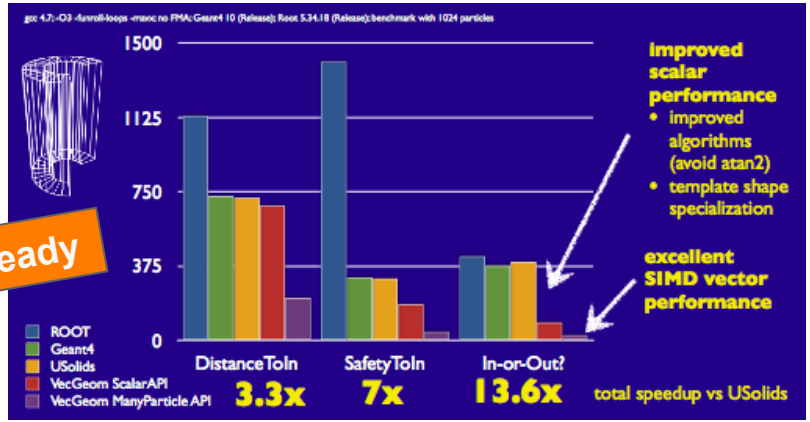
No demonstrator - production setup!

Rethink software was a must: « parallel by design »

library	PrPixel addHits		RICH EigenGeom	
	X5650 @ 2.67GHz	E5-2630 v3 @ 2.40GHz	X5650 @ 2.67GHz	E5-2630 v3 @ 2.40GHz
sequential	1.00x	1.00x	1.00x	1.00x
intrinsics	1.94x	2.57x		
gcc intrinsics	1.45x	2.09x		
vc	1.66x	2.26x	1.35x	1.49x
Vectorclass	1.60x	2.23x	1.35x	1.50x

Ready

SIMD studies in the LHCb reconstruction software, D. Campora et al.



A new generic C++ geometry library for detector-particle simulation, S.Wenzel et al.

Geometry library

Redesign of algorithms (e.g. get rid of math functions when possible)

Multiparticle Vector interface

Compatible with production uSolids!

The Role of Memory

Performance benchmark of LHCb code on state of the art x86 architectures, R. Schwemmer et al.

We have parallel HEP data processing

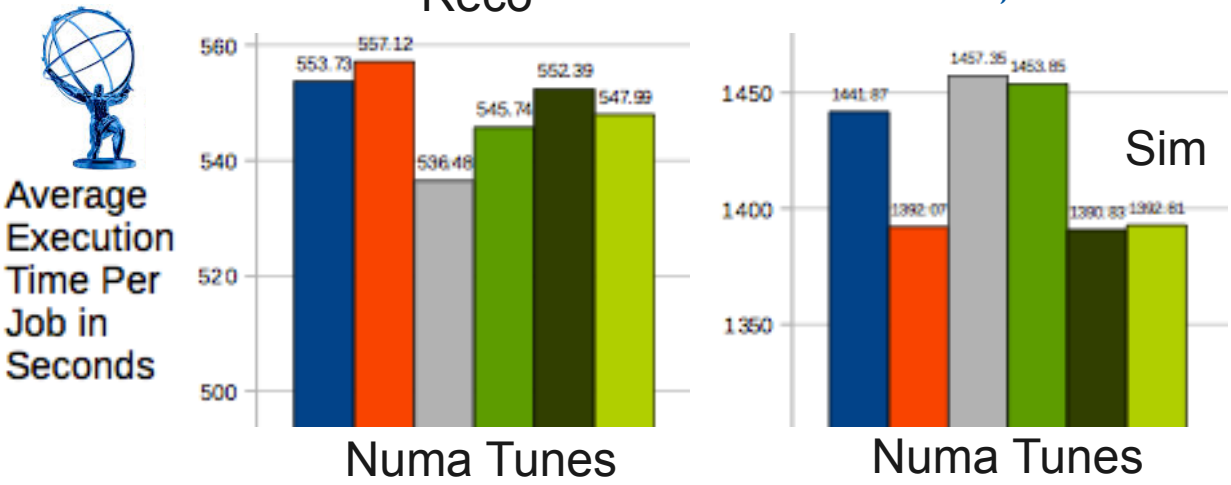
→ Study Numa effects

Here: no MT but MP

Spawn children on same socket: no uncore events → +14% decision rate

CPU	Decisions/s No NUMA	Decisions/s NUMA	NUMA Gain
Intel X5650 (8 cores)	599.6	648.8	1.08
Opteron X272	632.35	682	1.08
E_2630 v3 (8 cores)	865	986	1.14
E_2650 v3 (10 cores)	1129	1210	1.07

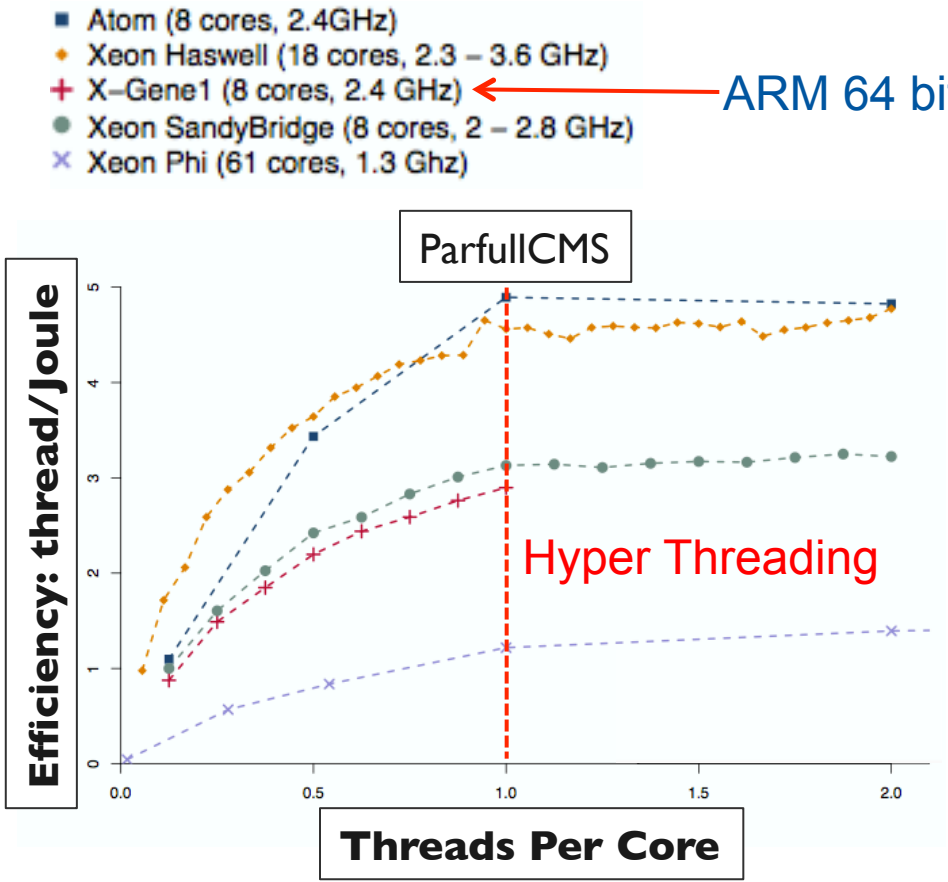
Standalone, real LHCb HLT application available on DVD!



The same tune has different effects on different workflows!

The Effect of NUMA Tunings on CPU Performance, C. Hollowell et al.

Power Efficiency



The race is heating up: Intel is not a spectator

Arm64 bits: not a « panacea »

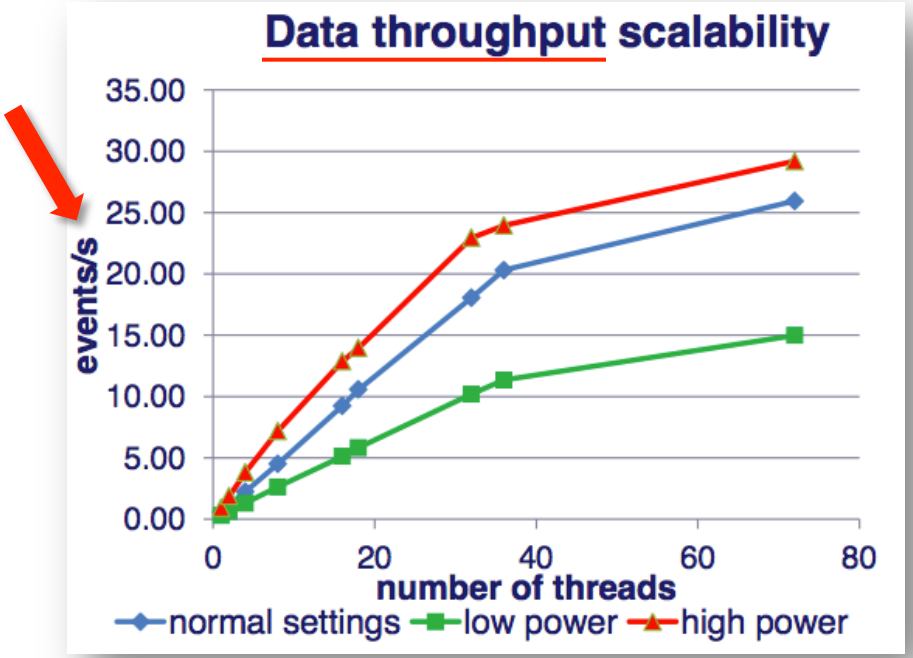
- Wait for next generations: stay tuned!

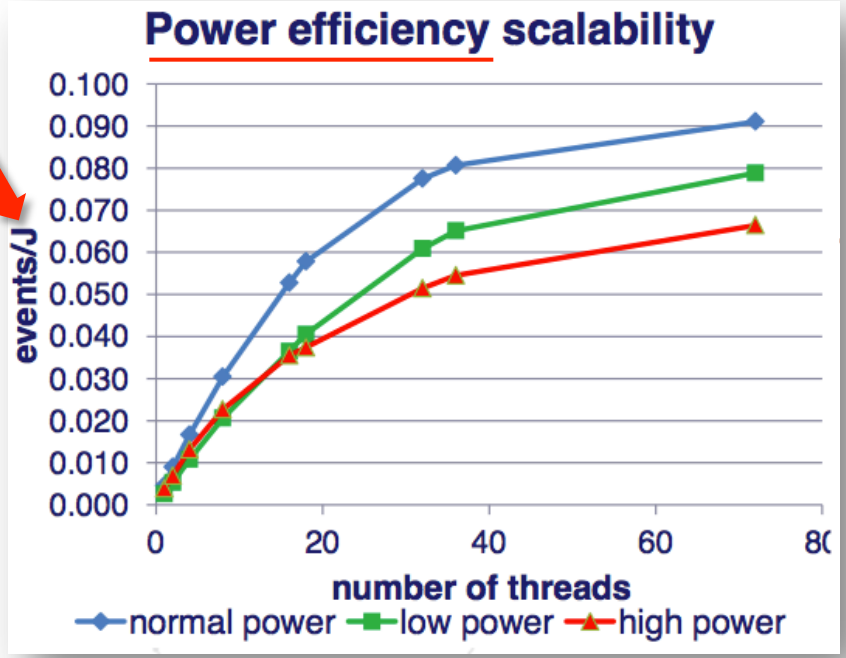
Future Computing Platforms for Science in a Power Constrained Era, G. Eulisse et al.

High Degree of Complexity

Haswell-EP E5 2699v3 18 cores/socket

ParFullCMS Benchmark (G4MT)





Evaluating the power efficiency and performance of multi core platforms using HEP workloads, P. Szostek et al.

Partial Bottomline

Measuring performance became more and more challenging - complex hardware, heterogeneous platforms, many benchmarks

Taking advantage of cutting edge hardware features is possible but it requires an evolution of our software:

- E.g. Redesign and re-implementation of algorithms and data structures: such an effort needs resources
- Understanding of the physics involved and strong technical skills must be in our community (study, train, collaborate and learn)

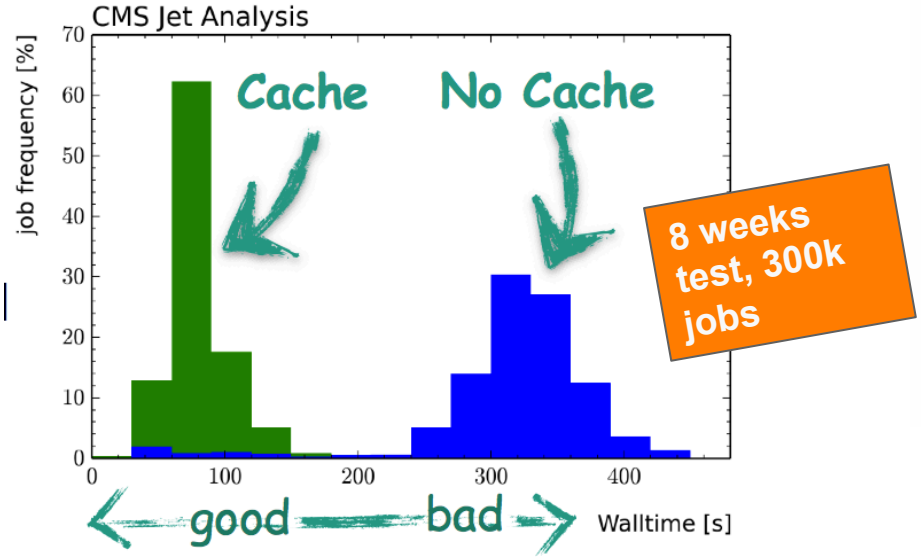
These activities are really taking momentum!

I/O

- Fundamental aspect for « voluminous data » applications
- Typical bottleneck for analysis workflows.
- Two possible solutions explored:
  - Hierachical coherent caching
  - In memory computing
- Getting Smarter!

I/O

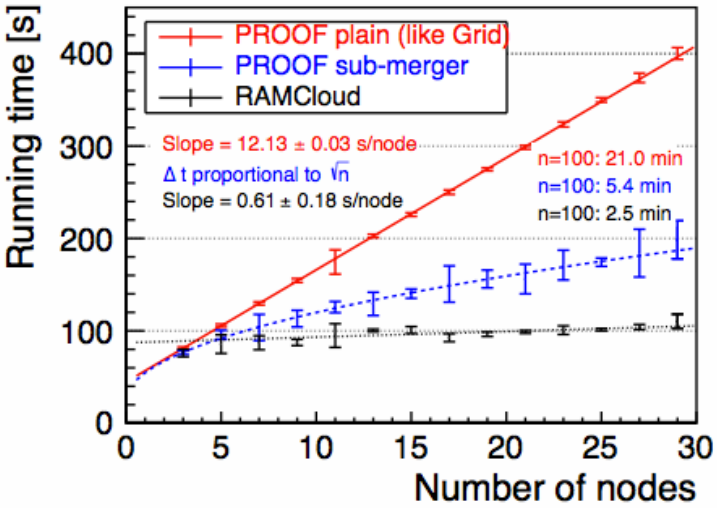
High Performance Data Analysis via Coordinated Caches, M. Fisher



University Cluster: I/O intensive HEP data analysis

Provide an SSD cache to nodes via UFS

No ad-hoc protocol, just declare consumed datasets in jdl



Histogram merging: typical serial problem

12k histos, 19M bins

RAMCloud: BigData technology

General purpose distributed storage solution competitive with highly tuned HEP tool

Large-Scale Merging of Histograms using Distributed In-Memory Computing, J. Blomer

Throughput and HPC

HPC Farms

- Huge machines: 10% of Titan → 300M cpu h unused per year
- We are guests there: opportunistic usage
- adopt policies made by others…

… But it's in their interest to fill up the machines

- HW resources not (yet) on our bill, free help from HPC experts

- Generation and simulation: perfect candidates for such systems
  - Little IO, a lot of CPU
- Event level parallelism again leveraged!
  - Run during machine draining or among big fishes

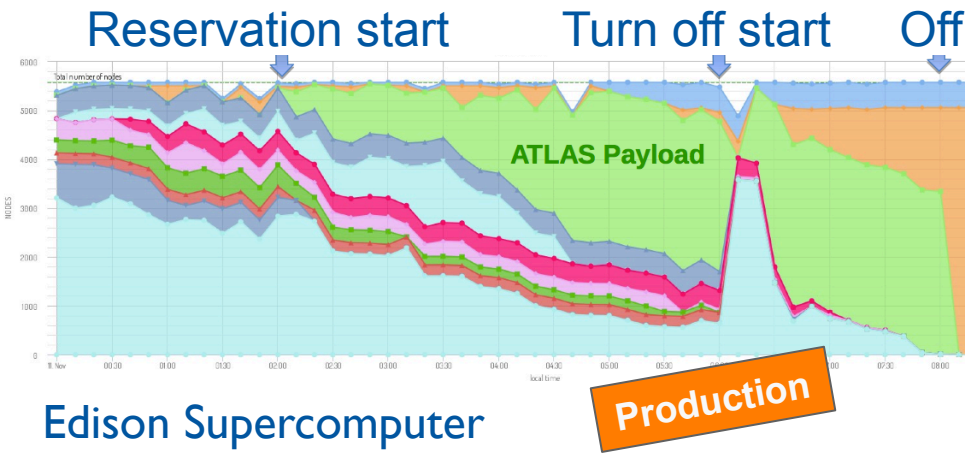
HPC Farms

HEP Gen-Sim jobs would be the « sand » filling the gaps

Picture taken from: Fine grained event processing on HPCs with the ATLAS Yoda system, V. Tsulaia et al.

GenSim

Fine grained event processing on HPCs with the ATLAS Yoda system, V. Tsulaia et al.



Edison Supercomputer

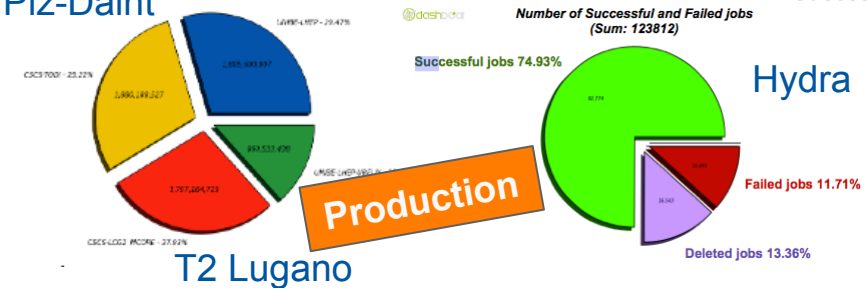
Yoda system: MPI based, HPC friendly event dispatching system

Inject work while machine starts to turn off or while reserved for big jobs

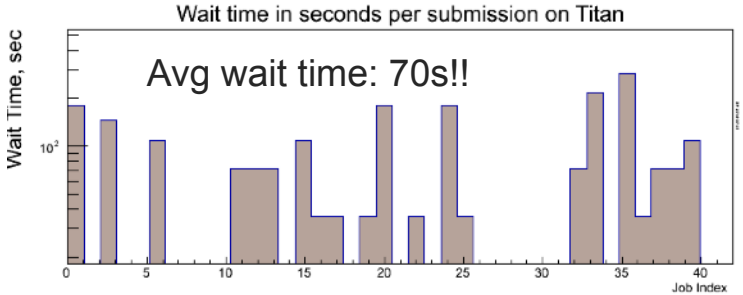
Simulation with Geant4

Piz-Daint

T2 Bern



T2 Lugano



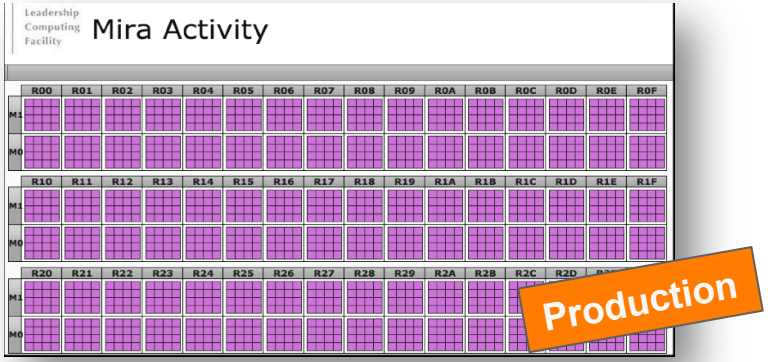
ATLAS computing on the HPC Piz Daint machine, Hostettler M. et al

Bringing ATLAS production to HPC resources - A use case with the Hydra supercomputer of the Max Planck Society, Mazzaferro L. et al

Integration of PanDA workload management system with Titan supercomputer at OLCF, S. Panitkin et al.

Generation and Analysis

Simulation of LHC events on a million threads, T. Childers et al.



Run heavy event generation: W+6 jets

Too expensive: no grid. Enable science with HPC!

Many runs: one filled Mira entirely for 20 minutes - ~4M gen evts

Possibility to share these events? Public, theorerical community, experiments

Analysis: machine learning technique

Process full CMS Run1:

4100h sequentially on CPU → 6 k20 GPUs: 10 days

OpenCL and Cuda approach: Mic, CPU tested as well!

Asset also for gaming cards: ~300 Eur/Chf/$

Ready

$$P(\mathbf{x}|\Omega) = \frac{1}{\sigma_\Omega} \underbrace{\iiint dx_1 dx_2 d\mathbf{y}}_{\text{ROOT GSL VEGAS}} \underbrace{\mathcal{P}_s(x_1, x_2)}_{\text{LHAPDF}} \underbrace{|\mathcal{M}_\Omega(x_1, x_2, \mathbf{y})|^2}_{\text{MADGRAPH generated code}} \underbrace{W(\mathbf{x}, \mathbf{y})}_{\text{« hand-made »}}$$

Matrix Element Method for High Performance Computing platforms, D. Chamont et al.

Partial Bottomline

- HPC: precious resources and strategies for HEP
  - Already in production for some workflows, e.g. Generation and Simulation
- More science: sophisticated generators to access corner of phasespace (e.g. Vector Boson + jets)
- Some supercomputers: PowerPC architecture - Big Endian (Intel LE)
  - Need to review our software to run there: generators, G4 already made it
  - New Zoo of architectures? Mainly two: Power8(9) and AArch64
  - Opportunity to improve the codebases