

Track 1 summary report

Presenter Name

The Track 1 coordinators

Luis Salinas, Axel Naumann and Niko Neufeld

The Track 1 advisors

David Britton, Jerome Lauret, Clara Gaspar

ACAT 2016

18-22 January 2016

UTFSM, Valparaíso (Chile)



Computing technology for Physics Research

Summary by Niko Neufeld and Graeme Stewart

Introduction


- Track 1 had a lot of **high quality contributions**
- They cover a very wide field of software and hardware techniques and technologies
- A selection will be presented and some synthesis attempted

“Not mentioned” does **not** mean “not interesting”



Statistics

- 39 abstracts submitted (1 moved to plenary)
- 20 talks accepted, 1 short-term cancellation
- 17 posters accepted out of which 15 were actually shown

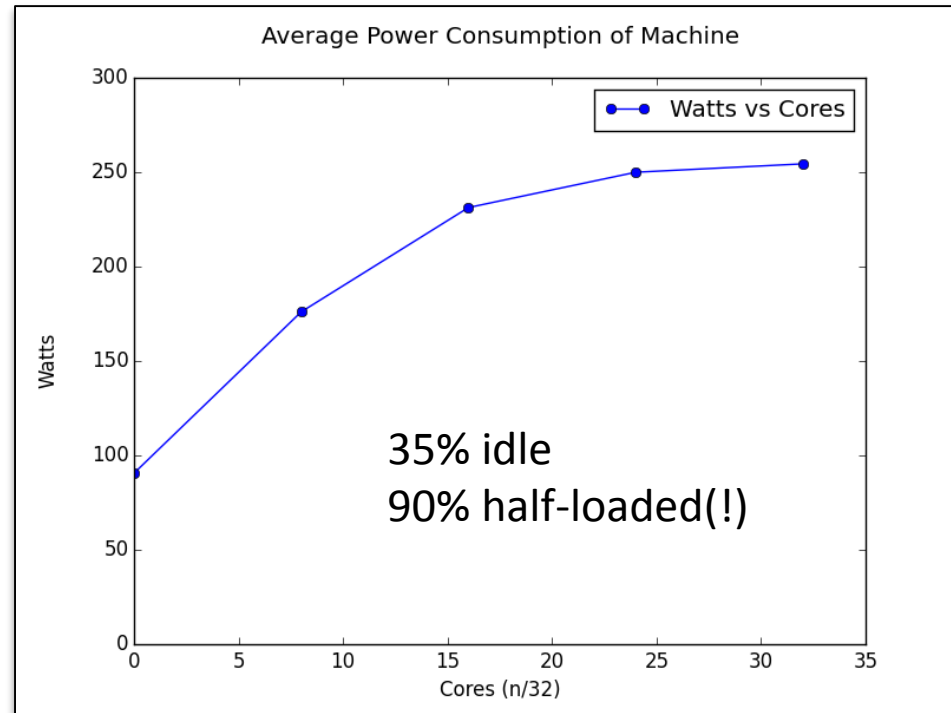


Fabrics, infrastructure, power, data-placement,
workload engines and all that...

Managing power

- Introduce hibernation policy → power savings
- Can be made more efficient using preemptible jobs
- Adapting to varying prices → cost saving
- How to properly integrate what the users *really* want?

	Machine Online	• Machine Offline
Jobs starting promptly	Hibernate if: <ul style="list-style-type: none"> • Machine has been idle for 5 minutes • At least 30 minutes since last wakeup • Machine COULD start jobs 	<ul style="list-style-type: none"> • Do Nothing
Jobs waiting in queue		<ul style="list-style-type: none"> • Wake machine if: • At least 30 minutes since last hibernation

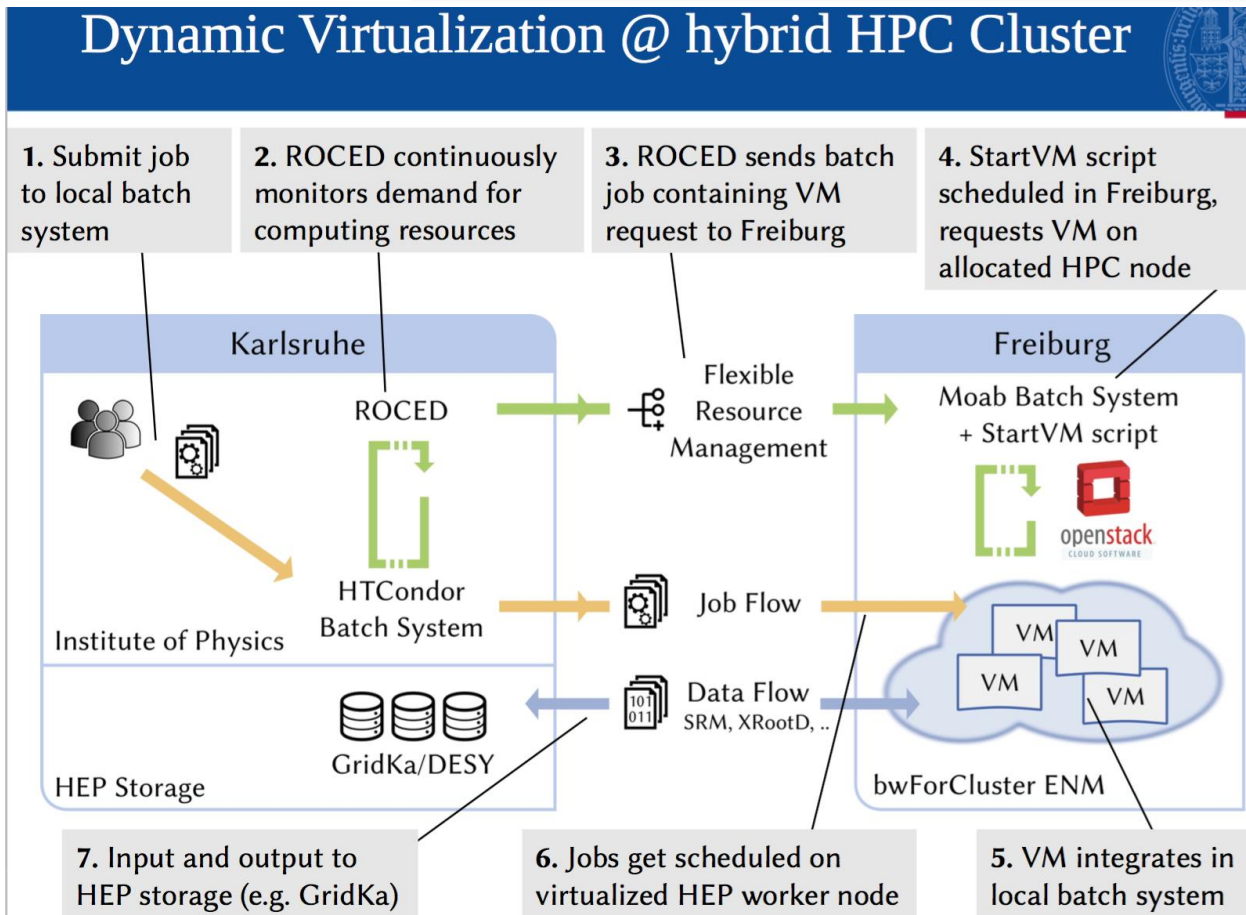


Greg Corbett “Reducing power consumption on demand”

Using HPC resources

Konrad Meier *“Dynamic provisioning of a HEP computing infrastructure on a shared hybrid HPC system”*

CMS “Elastic extension of a CMS Computing Centre resources on external Clouds”



Using HPC resources II

Markus Fasel "Using NERSC High-Performance Computing (HPC) systems for high-energy nuclear physics applications"

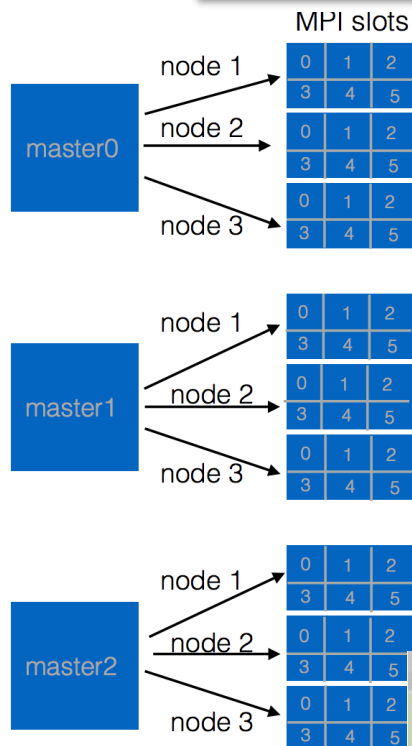
Tool which runs multiple serial jobs as a MPI job

- Submitter:
 - Splits a master into n sub jobs
- Worker (MPI):
 - Runs the subjobs (payload)
- Job description: config, json, xml

Key facts:

- PYTHON, mpi4py
- BSD-type license
- <https://bitbucket.org/berkeleylab/analisa>

Hiding complexity of resource management for the user



- HPCs not generally the best suited resources to our problems
- However they *do* exist and we *do* have access to them
- Running here can bring considerable resources
- Standardisation...?

See also posters "Integration Of PanDA Workload Management System With Supercomputers for ATLAS and Data Intensive Science"
"Scaling up ATLAS Event Service to production levels on opportunistic computing platforms"

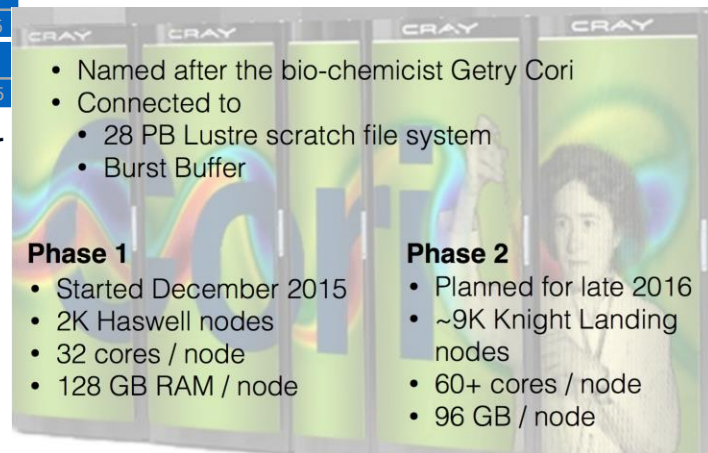
- Named after the bio-chemist Getty Cori
- Connected to
 - 28 PB Lustre scratch file system
 - Burst Buffer

Phase 1

- Started December 2015
- 2K Haswell nodes
- 32 cores / node
- 128 GB RAM / node

Phase 2

- Planned for late 2016
- ~9K Knight Landing nodes
- 60+ cores / node
- 96 GB / node



- Containment of job resources using cgroups
- Careful monitoring of many jobs
- **Have observed minimal < 2% run-time impact by limiting resident memory to 75% of requested** (needs to be validated for different workloads)

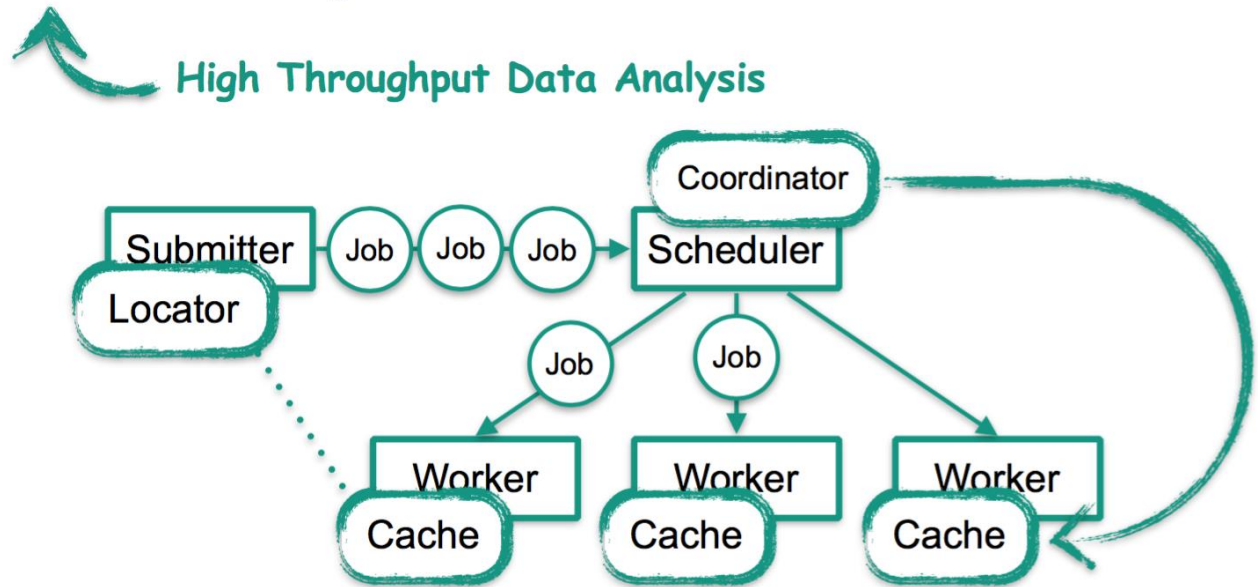
	Job Discription	CPU Req.	MEM Req.	Mem Needed?	Saving
1	ATLAS Multicore Simulation	8	16	10-12	25% - 38%
2	ATLAS Multicore Reconstruction	8	16	16	0%
3	ATLAS Single-core Simulation	1	3	1.8	40%
4	ATLAS Single-core Analysis	1	4	1.8	55%
5	ATLAS Sequential Single-Core Analysis	1	4	1.8	55%
6	ILC VO Jobs	1	2	1.2	40%
7	PhenoGrid VO Jobs	1	2	1.2	40%
8	GridPP VO Jobs	1	2	1.2	40%

- On this basis, we reduced memory allocation of all single core jobs to **1.8GB** for the last 6 months:
 - Running was stable.
 - **We achieved 10% higher CPU usage.**

For detecting problems see also: Eileen Kuehn, Manuel Giffels "A scalable architecture for online anomaly detection of WLCG batch jobs"

Use data-popularity for caching

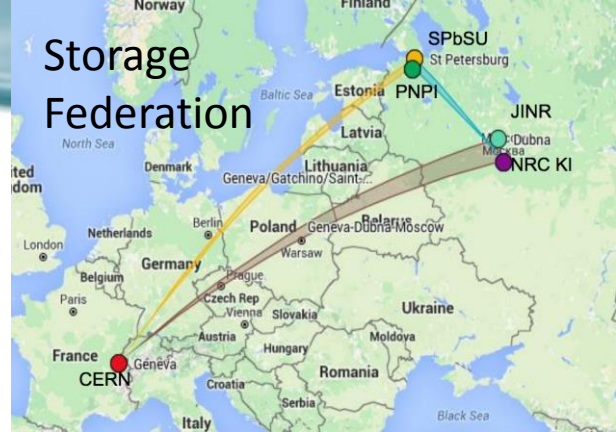
HTDA Batch System Extension



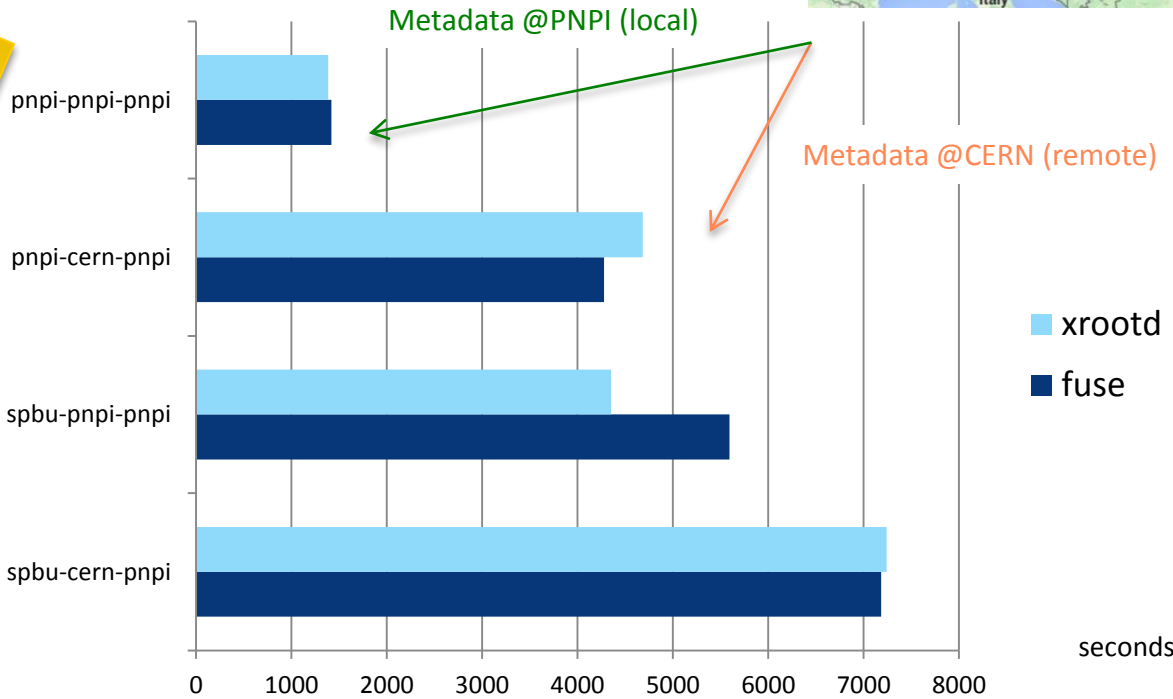
Max Fischer, Manuel Giffels "Data Locality via Coordinated Caching for Distributed Processing"

- Caches maintain data copies on worker nodes
- Locator provides locality information for jobs
- Coordinator schedules files for caching on nodes

Locality of meta-data / EOS federation



Alexei Klimemtov "Data Locality via Coordinated Caching for Distributed Processing"



- **Legend**
 - PNPI-PNPI-PNPI : client@PNPI, data@PNPI, MGM@PNPI
 - PNPI-CERN-PNPI: client@PNPI, data@PNPI, MGM@CERN
 - SPBU-PNPI-PNPI: client@SPBU, data@PNPI, MGM@PNPI
 - SPBU-CERN-PNPI : client@SPBU, data@PNPI, MGM@CERN

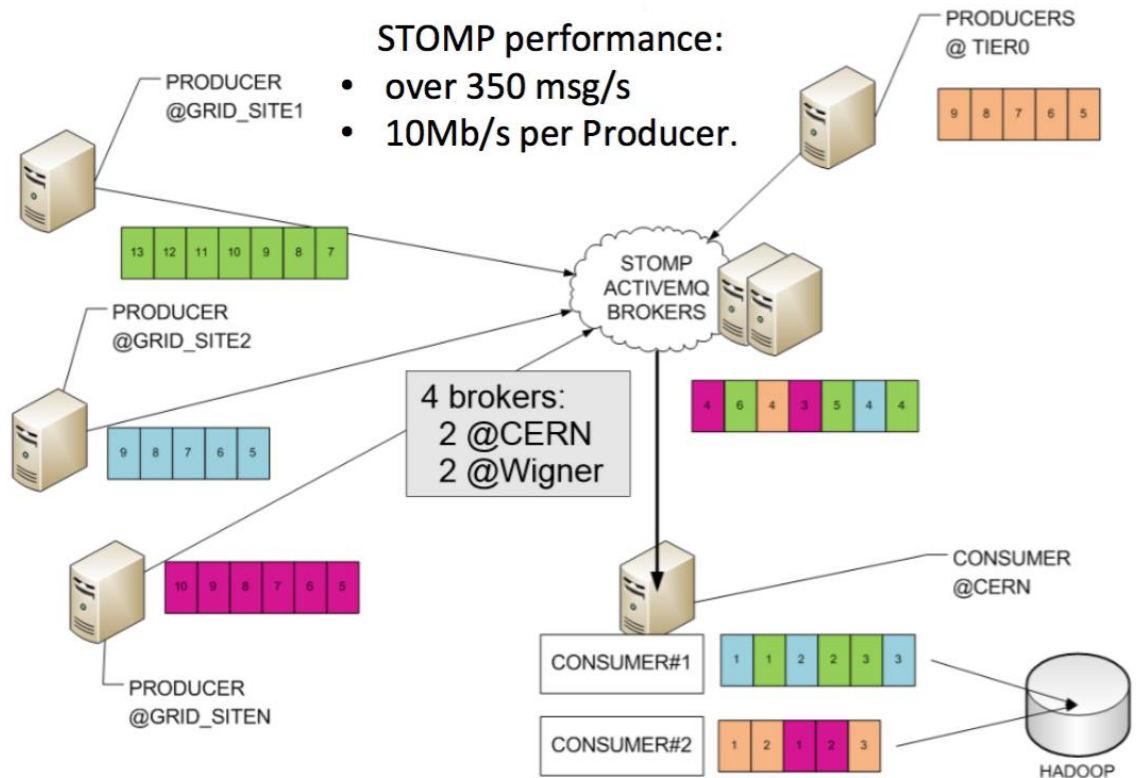
Fedor Prokoshin "The ATLAS EventIndex:
data flow and inclusion of other meta-data"

EventIndex uses 350 B/
event

Allows event-picking and
counting and duplicate
detection:

Uses ActiveMQ and central
Hadoop infrastcture at
CERN

*Objective: Find the
data I want!*



Measured performance for sending real events:

- **200K event/s**
- **60Mb/s**

(1Broker, 6 Producers, 4 Consumers, 50K events/job)

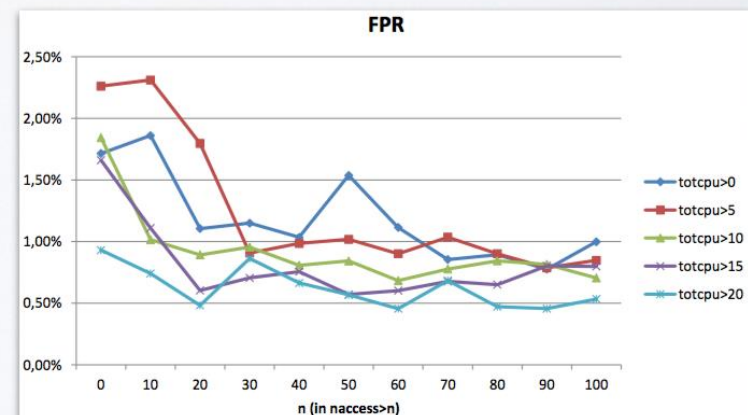
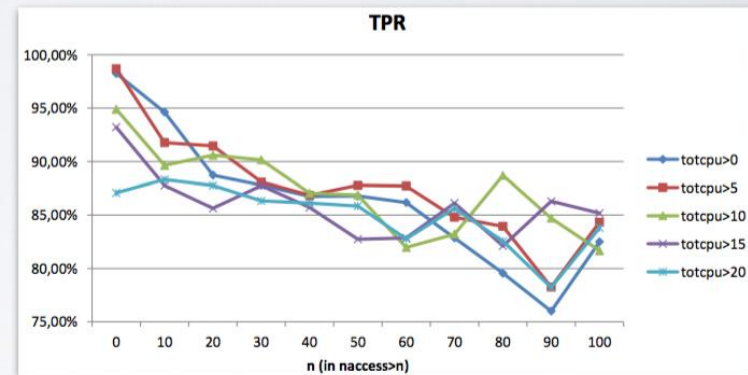
“Will these data be popular?”

Related study done in LHCb
Nikita Kazeev “LHCb data processing optimization using Event Index”

POPULARITY METRICS

Valentin Kuznetsov , Daniele Bonacorsi
“predicting CMS dataset popularity”

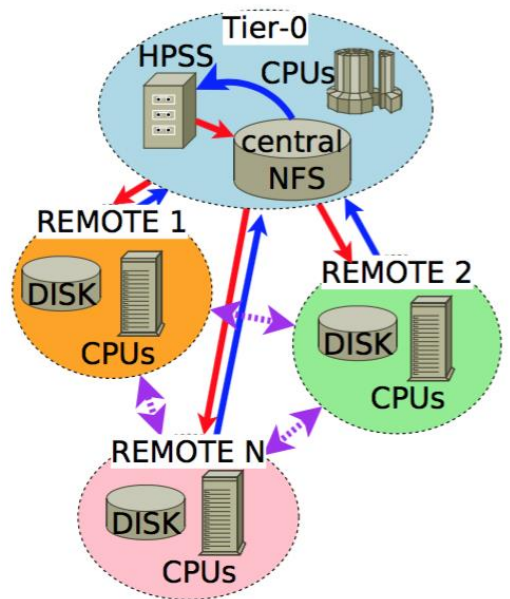
- Define popular dataset as those which passed a given cut, perhaps even combined cut
- A good choice:
 - # accesses > 10 & tot CPU hrs > 10
- Train model (see next) and look at FP yield
- Study cut effect on yield of FP vs data tier



Dzmitry Makatun "Multi-resource planning: Simulations and study of a new scheduling approach for distributed data production in High Energy and Nuclear Physics"

In a world with thin pipes, varying CPU and disk resources and network links:

How do we best schedule processing?



— Input data flow
 — Output data flow
 Links between remote sites

Forwarding allows using full topology of scientific internet

All you need is a (better) plan

Plan execution

Handler

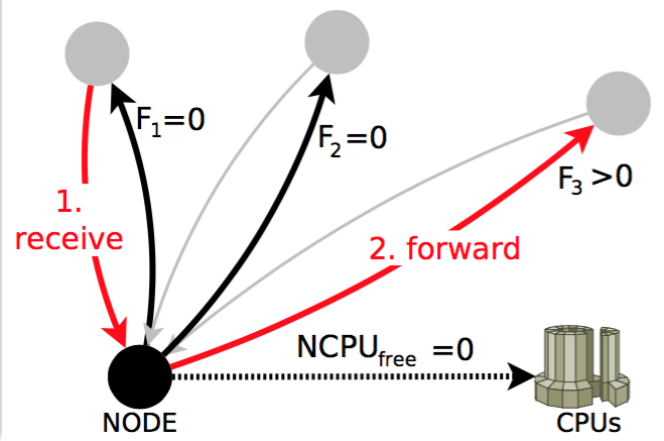
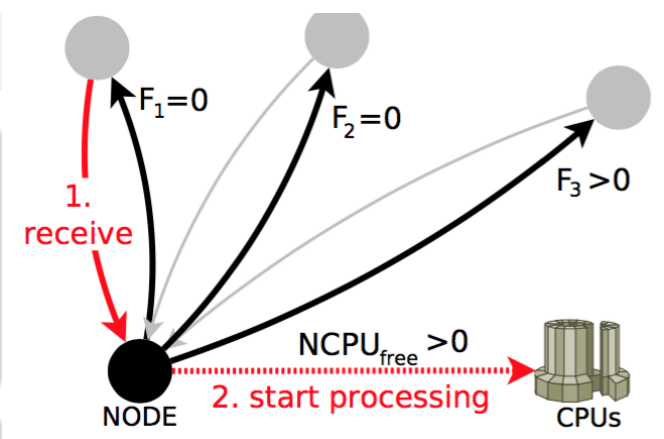
- Service runs at each site
- Sends status data
- Executes the plan

When a new file arrives:

Process the file (if $NCPU_{free} > 0$)

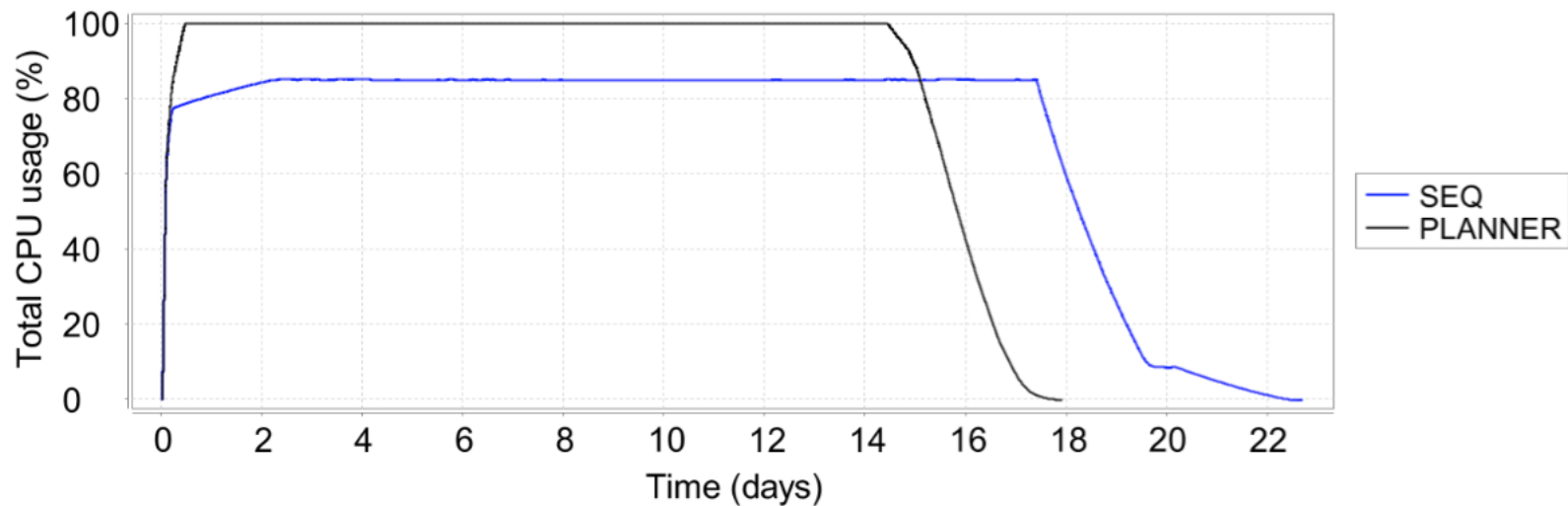
OR Forward it over one of the links (if $\exists F_l > 0$), decrease F_l

OR Keep the file until a new plan or a free CPU appears



Large scale simulation of planner

11 sites
50 kCPUs
500 k jobs



Much higher CPU utilization
Adaptive!

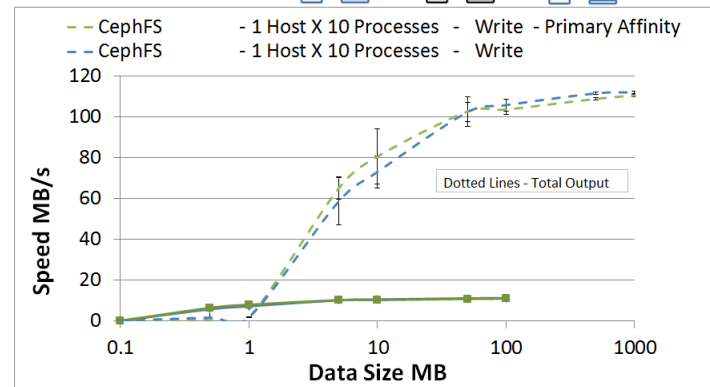
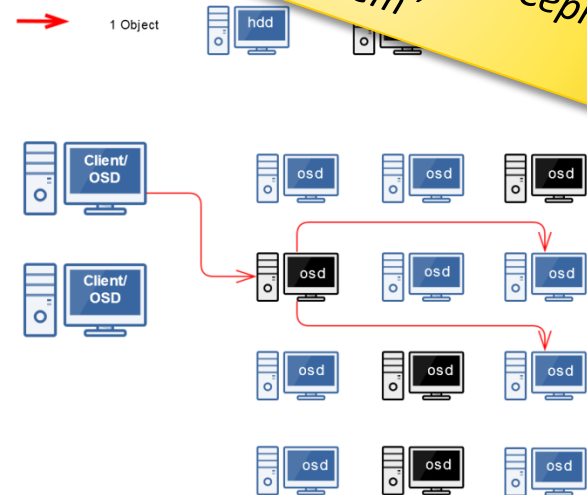
Leveraging the resources in the Online clusters


- Move processing toward online domain requires storage (Online QA, real-time calibration, ...)
- Users can store and recover data to a local storage while running jobs – “a” global space helps share results across processing nodes
- Distributed file system & aggregators exists (Xrootd, Object storage, ...) – users most familiar with POSIX FS.

CEPHFs offers Posix FS w/o single point of failure

In practice:

- Very stable performance survived even DNS breakdown
- Study judicious use of SSD → the use-case needs to be demanding enough to see advantage
- CEPH has no single point of failure, but CEPHFS seems to be only as fast as the weakest performing component!

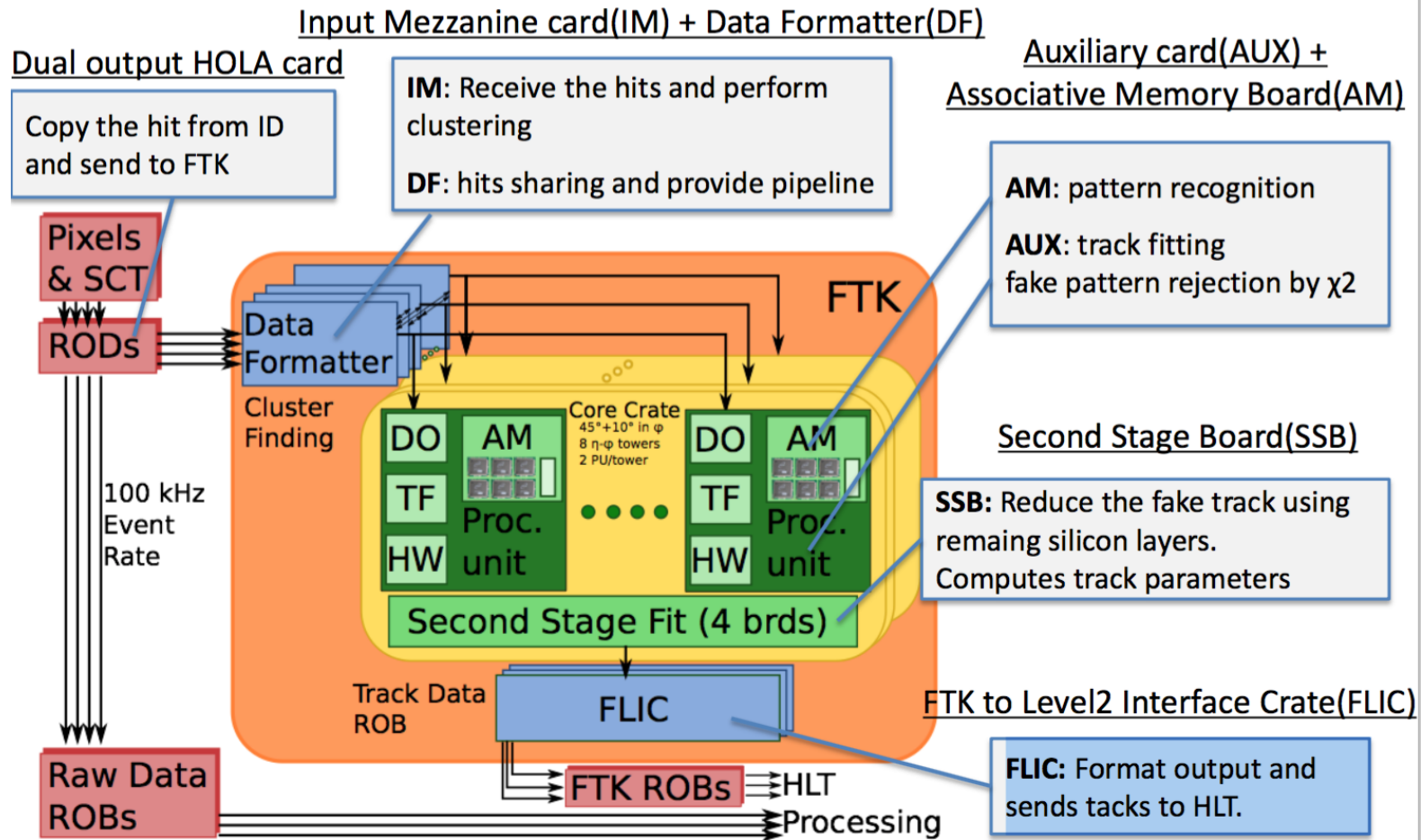




Triggers soft vs hard – the debate continues

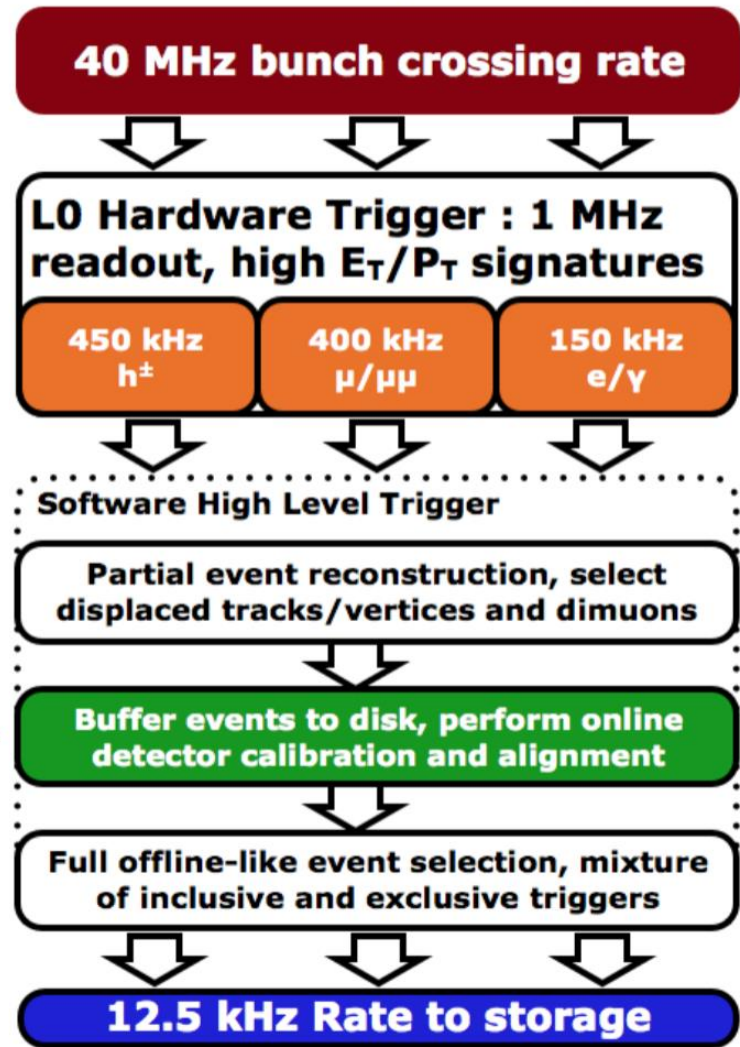
New and better triggers

Naoki Kimura "ATLAS FTK a - very complex - custom parallel supercomputer"



Towards 99.99 offline quality in LHCb triggers

Run 2 (2015)



Johannes Albrecht "The LHCb High Level trigger in Run 2"

For CMS future trigger see poster by D. Cieri "Hardware Demonstrator of a L1 Track Finding Algorithm with FPGAs for the Phase II CMS Experiment"



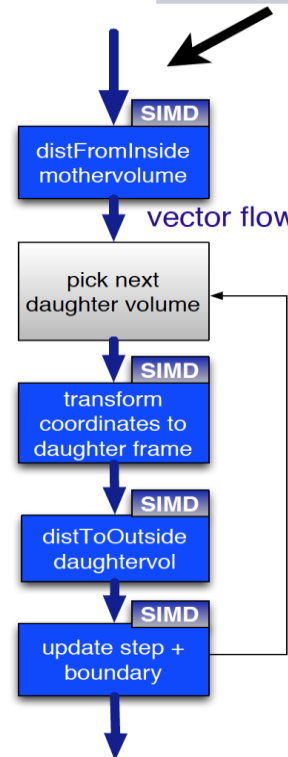
Accelerate me! Make me better! Make me parallel!

Geometry performance (Phi vs Xeon) in GeantV

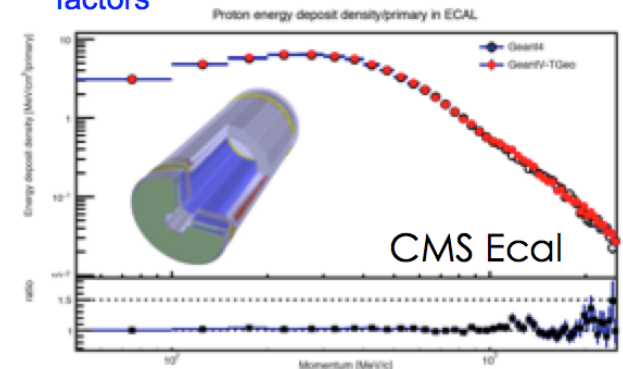
Andrei Gheata "From CPU to accelerators"

- **Geometry is 30-40% of the total CPU time in Geant4**
- A library of vectorized geometry algorithms to take maximum advantage of SIMD architectures
- **Substantial performance gains also in scalar mode**
- Testing the same on on GPU
- Moving from prototype to demonstrator

	16 particles	1024 particles	SIMD max
Intel Ivy-Bridge (AVX)	~2.8x	~4x	4x
Intel Haswell (AVX2)	~3x	~5x	4x
Intel Xeon Phi (AVX-512)	~4.1	~4.8	8x



- Current prototype able to run an exercise at the scale of an LHC experiment (CMS)
 - Simplified (tabulated) physics but full geometry, RK propagator in field
 - **Very preliminary results needing validation, but hinting to performance improvements of factors**

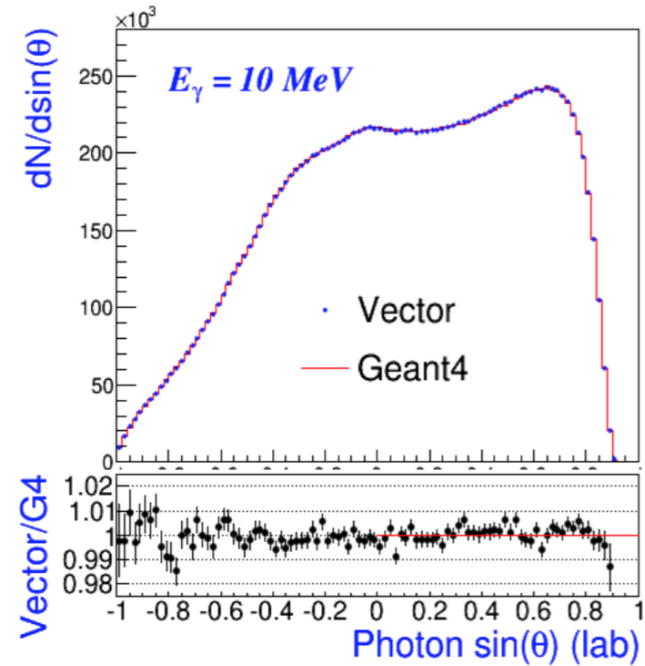
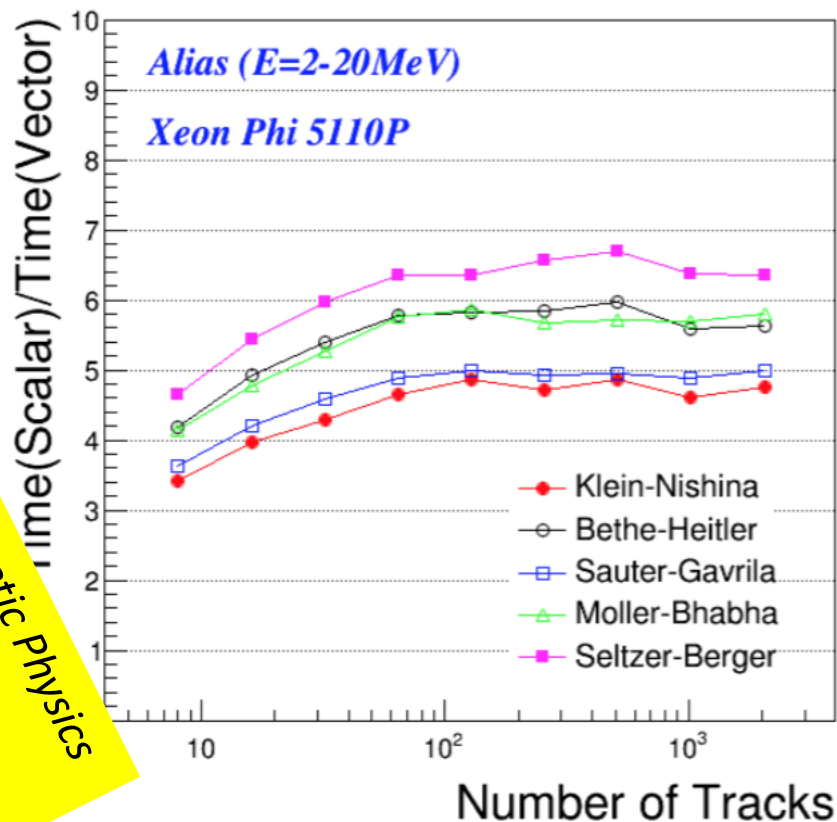


Going parallel

Goal: write EM models, to deal with multiple tracks ,to be accurate, fast and "portable"

Algorithms need to be redone – use alias sampling to vectorize pdfs for multiple tracks

Soon Yung Jun "Electromagnetic Physics Models for Parallel Computing Architectures"

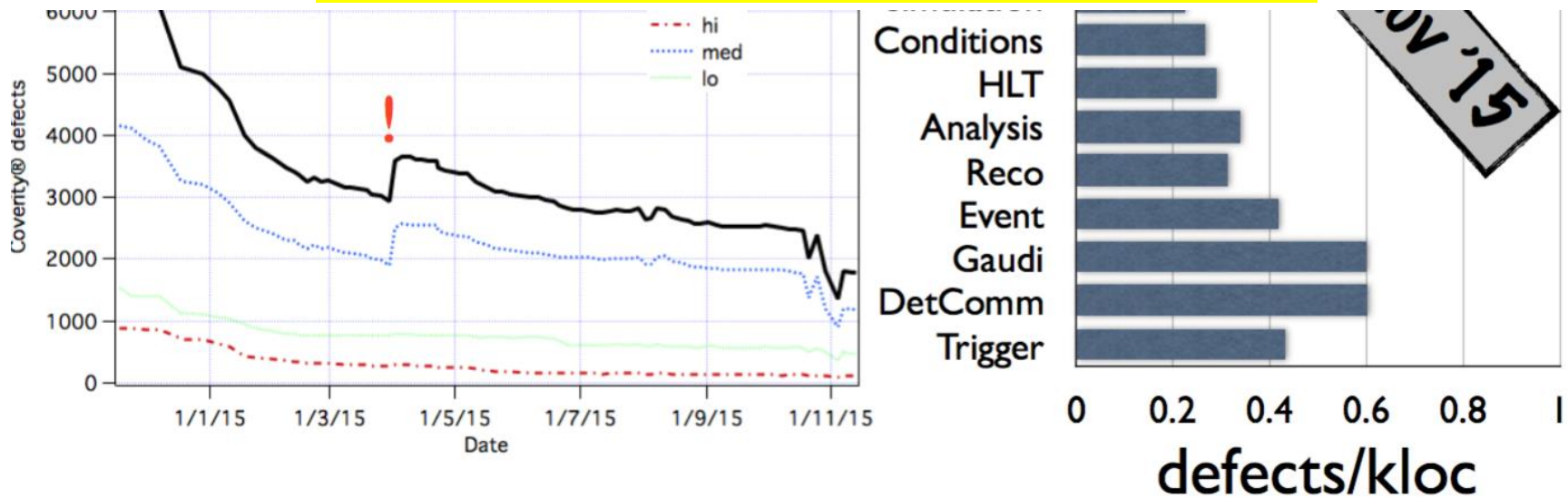


1% agreement with G4

Speedup on XeonPhi >6x

We are better than some of us sometimes think!

Shaun Roe, Graeme Stewart *“C++ Software Quality in the ATLAS Experiment”*



Industry report 2012

“Coverity’s analysis found an average defect density of .69 for open source software projects that leverage the Coverity Scan service, and an average defect density of .68 for proprietary code developed by Coverity enterprise customers. Both have better quality as compared to the **accepted industry**

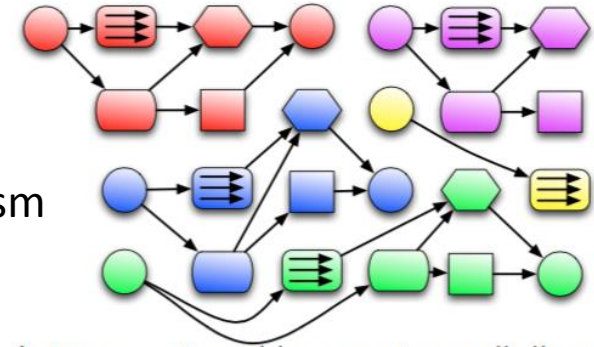
Going parallel

Memory / core goes down
 NUMA become more complex
 Memory bandwidth will be key to
 Use CPUs

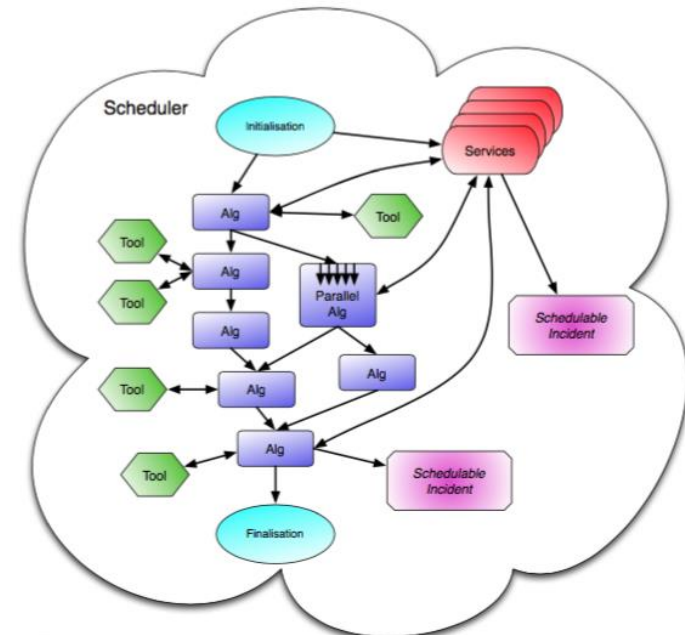
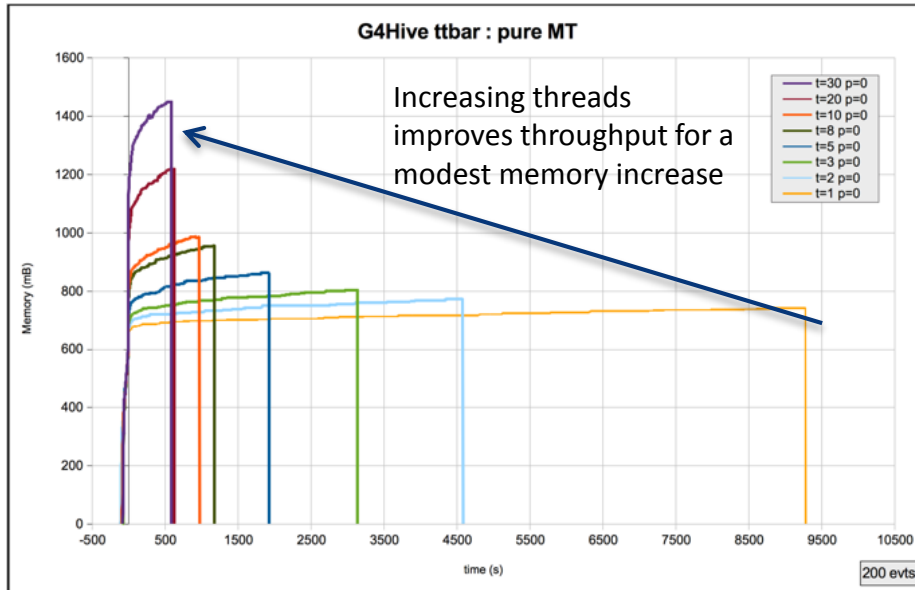
Re-training the development community is not easy!

AthenaMT exploits

- Multiple event parallelism
- Inter-event parallelism
- In-algorithm parallelism
- Uses Intel TBB under the hood



Inter-event and in-event parallelism (colours are events, shapes algorithms)

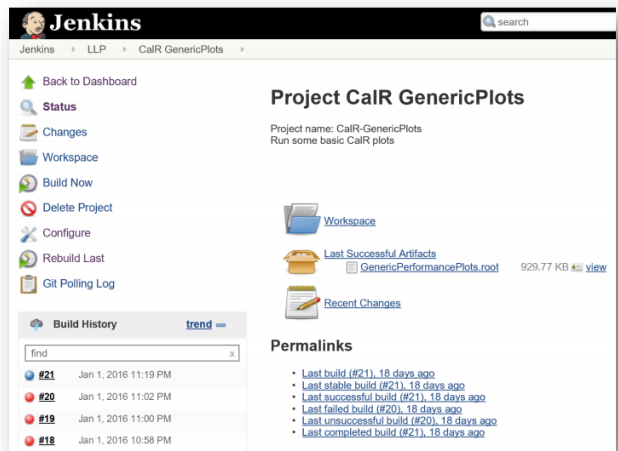


Framework element interaction within a single event

Can physics users go easier and faster...?

Use a declarative syntax to make it clearer what you are doing
More powerful when combining multiple plots

Re-run your physics analysis each time you have a change



Automatic regression tests with Jenkins

Declarative: Events

```
.SelectMany(jets)
.Where(|jet.eta| < 2.0)
.Do(hist.Fill(jet.pT));
```

Environment

Gordon Watts "Experiments Toward a Modern Analysis Environment"



Not our standard environment, but ideas attractive (c.f. ROOT plenary talk)





Beyond LHC and ACAT16

LHC is not the only big fish in the pond...

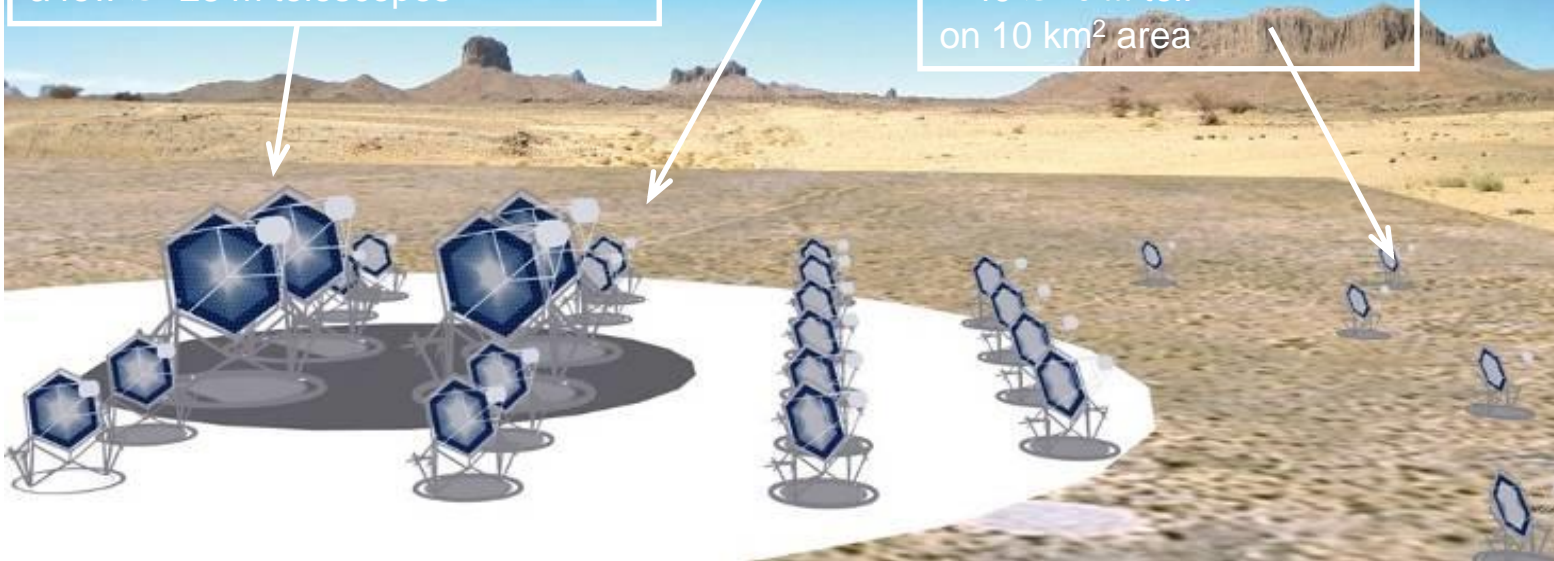
2 Arrays: North+South → All-Sky Coverage

More than 30 countries and ~300 M€ of budget.
More than 1000 members in the consortium.
Including the vast majority of the experts from existing experiments.

Medium Size Telescope - MST
core array
100 GeV-10 TeV
~ 40 $\varnothing=12$ m telescopes

Large Size Telescope - LST
low energy section
 $E_{\text{thresh}} \sim 10$ GeV
a few $\varnothing=23$ m telescopes

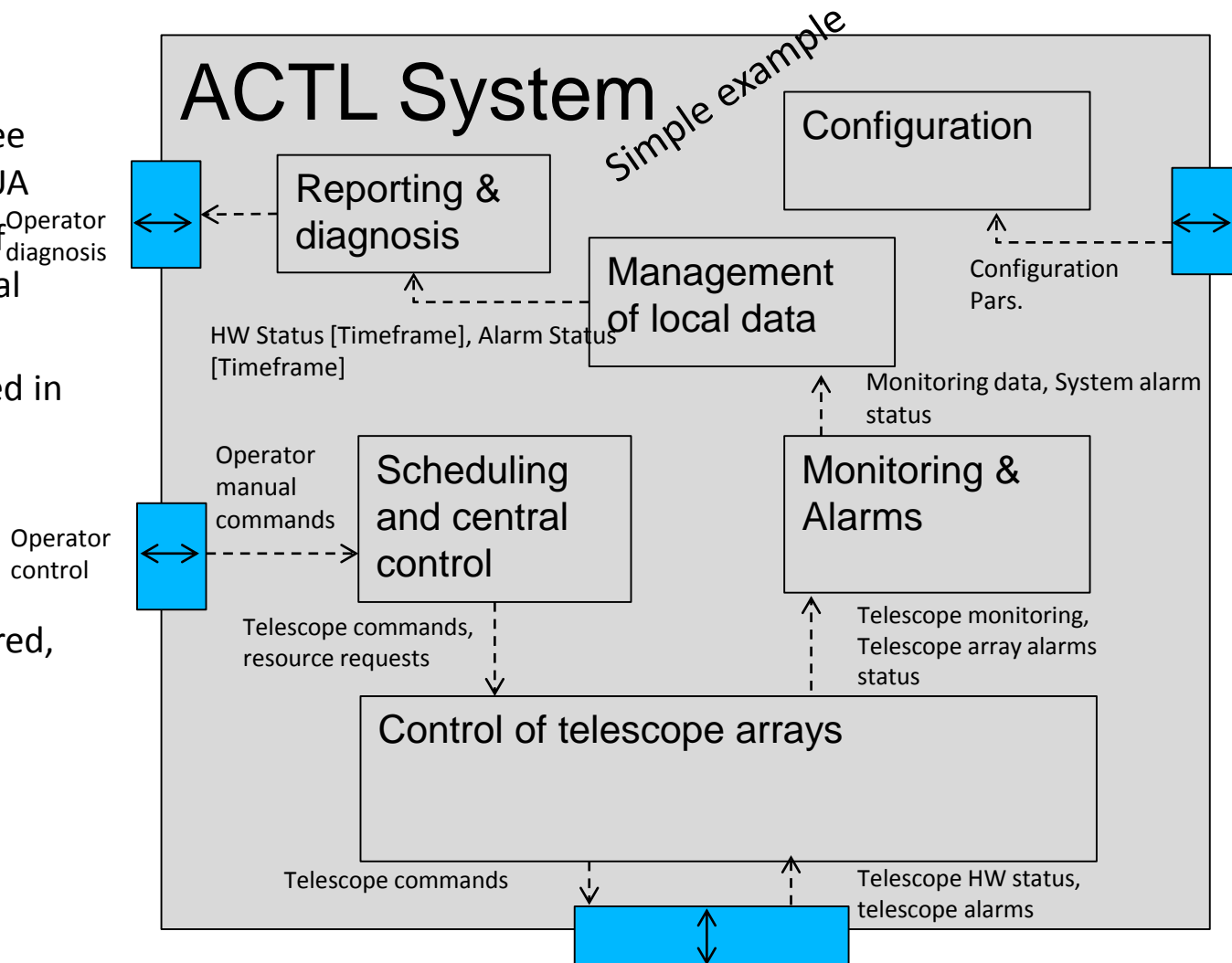
Small Size Telescope - SST
high energy section
~ 40 $\varnothing=6$ m tel.
on 10 km² area



P. Wegner "The software system for the Control and Data Acquisition for the Cherenkov Telescope Array"

CTA Array Control and Data Acquisition - ACTL

- Software based on Alma's software (see plenary) and OPC UA
- Information flow of 'abstract', functional data elements
- System decomposed in functionalities
- Functions have structure and granularity
- Architecture centered, using only COTS hardware
- 30+ year life-time



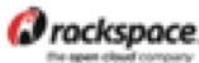


Many more projects with



ORACLE

SIEMENS



BROCADE



Yandex



F. Rademakers "New Technologies for HEP - The CERN openlab"

KINETIC

Open Storage Project

- Put/Get/Delete/... with a few extra's
- Checksum: can be verified by the drive
 - No need to read data for scrubbing
- Version: test-and-set functionality
 - Drive-side concurrency resolution

Test if we can improve EOS performance

Conclusions & synthesis

- Track 1 had a lot of high-quality contributions
- A lot of hard work went into improving and exploiting things we know:
 - Better use of computing fabrics, batch-facilities, less power, more CPU for the same money, better user-experience (run-time), exploitation of up-to-now unused resources (HPC, commercial clouds)
 - Accelerators are moving from the hype-peak to the productivity plateau
→ even if we don't use a specific accelerator technology, the efforts which went into making them usable for us will make our code better and – in most cases – faster

Conclusions and synthesis 2)

- We are not alone
 - Probably not we (human beings) in this universe ☺ but more importantly this conference showed (again) that there are other very large, long-lived scientific instruments out there (ALMA, CTA) → we can learn from each other
 - As well as our own ‘new kids on the block’ (Belle II)
- In online computing the ever higher data-rates are met with more and more sophisticated hardware (ATLAS, CMS track-triggers) and/or with offline-quality near-online data-reduction in software (LHCb) → the “debate” between these two approaches will continue



Conclusions and outlook 3)

- We didn't see a game-changer this time round
- But for sure they are out there, waiting to be discovered
- Looking forward to the next ACAT to see which of them will come forth