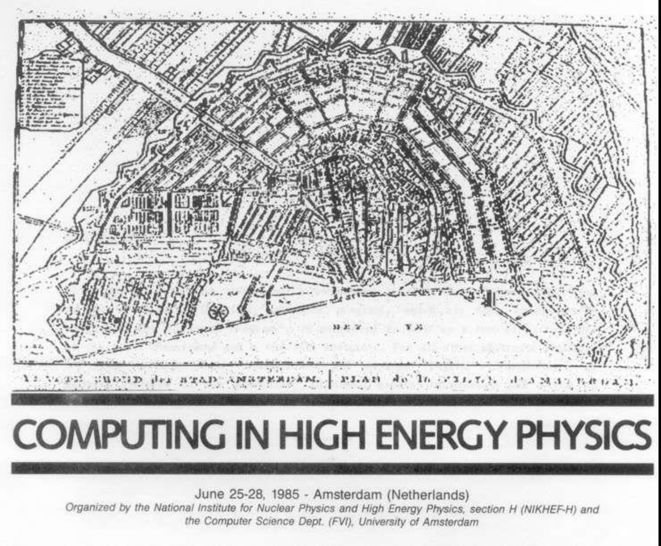
organized by NIKHEF in collaboration with partners

The 20th CHEP Conference

Amsterdam 2013

Amsterdam 1985

CHEP ´13
AMSTERDAM



Things have changed since 1985 …

… have completely gone away …

- “Portability Aspects of MODULA-2”
- “Using the 3081/E as a VAX Emulator”
- “A LAN with Real-Time Facilities *based on OSI Standards*”

… or have just changed a lot …

- “Satellite Communication”
- “LAN with an Experiment Command Interpreter and 2.5 MBaud Interfaces”

CHEP ´13 AMSTERDAM

http://www.chep2013.org/1985

… but not all that much!

- Multi-processor, multi-core & 'GPU'
  - "Loosely and Tightly Coupled Parallel Processors for High Energy Physics"
  - "Parallelism in Scientific Engineering Computation"
  - "Use of SIMD—SPMD Machines for Simulation in Particle Physics"
  - *Panel discussion*: "Vector and Parallel Processing in HEP"

CHEP '13 AMSTERDAM

- Large data volumes and transfers
  - "Data Storage - Where Do We Store Terabytes Of Data?"
  - "GIFT: An HEP Project For File Transfer"
- Resource sharing
  - ""Y": a Distributed Resource Sharing System in Nuclear Research Environment"

CHEP ´13 AMSTERDAM

http://www.chep2013.org/1985

A meta-summary

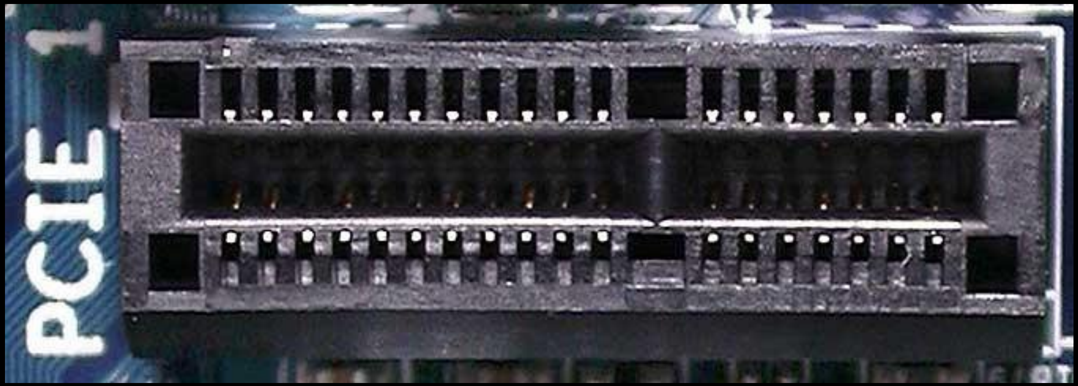
1. Data Acquisition, Trigger and Controls
2. Event Processing, ~~Simulation and Analysis~~
3. Distributed Processing and Data Handling
4. Data Stores and Storage Systems
5. Software Engineering
6. Facilities, Infrastructures, ~~Collaboration~~

- Parallelism & Multi-Core
- Data Preservation

Thanks to all speakers – you'll see mostly their slides!

with some blue-ish text which is typically mine …

CHEP´13 AMSTERDAM



DAQ, TRIGGER AND CONTROLS

Images from Wikimedia commons: Arnold Reinhold

“Trend to use more and more COTS equipment and all-software based solutions continues”

“DAQ systems outside HEP have been growing a lot: challenges comparable, similar ideas & synergy coming on”

General purpose DAQ tools

- First to appear in production for ‘lower volume’ experiments, but expanding to much more. E.g. for artDAQ:
  - DarkSide-50 at LNGS: 10 Mbyte event, ~ 80 Hz
  - LBNE (neutrino exp) FNAL -> Sanford: 4 GByte/s
- Happily borrowing good re-usable components from elsewhere
  - Run controls from IceCube
  - Configuration management from NOvA

Just the reco-algorithms are experiment-specific

Some Technical Details

Uses C++11 features

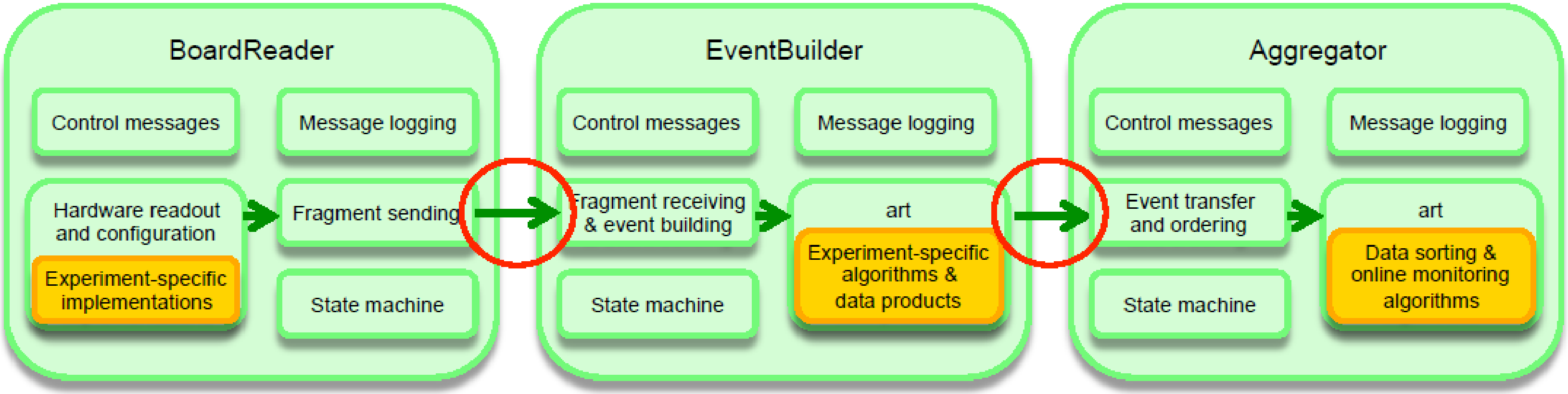
- e.g. move semantics to minimize data copies

MPI for data transfer

- Wrapper classes for sending and receiving MPI buffers

Process management

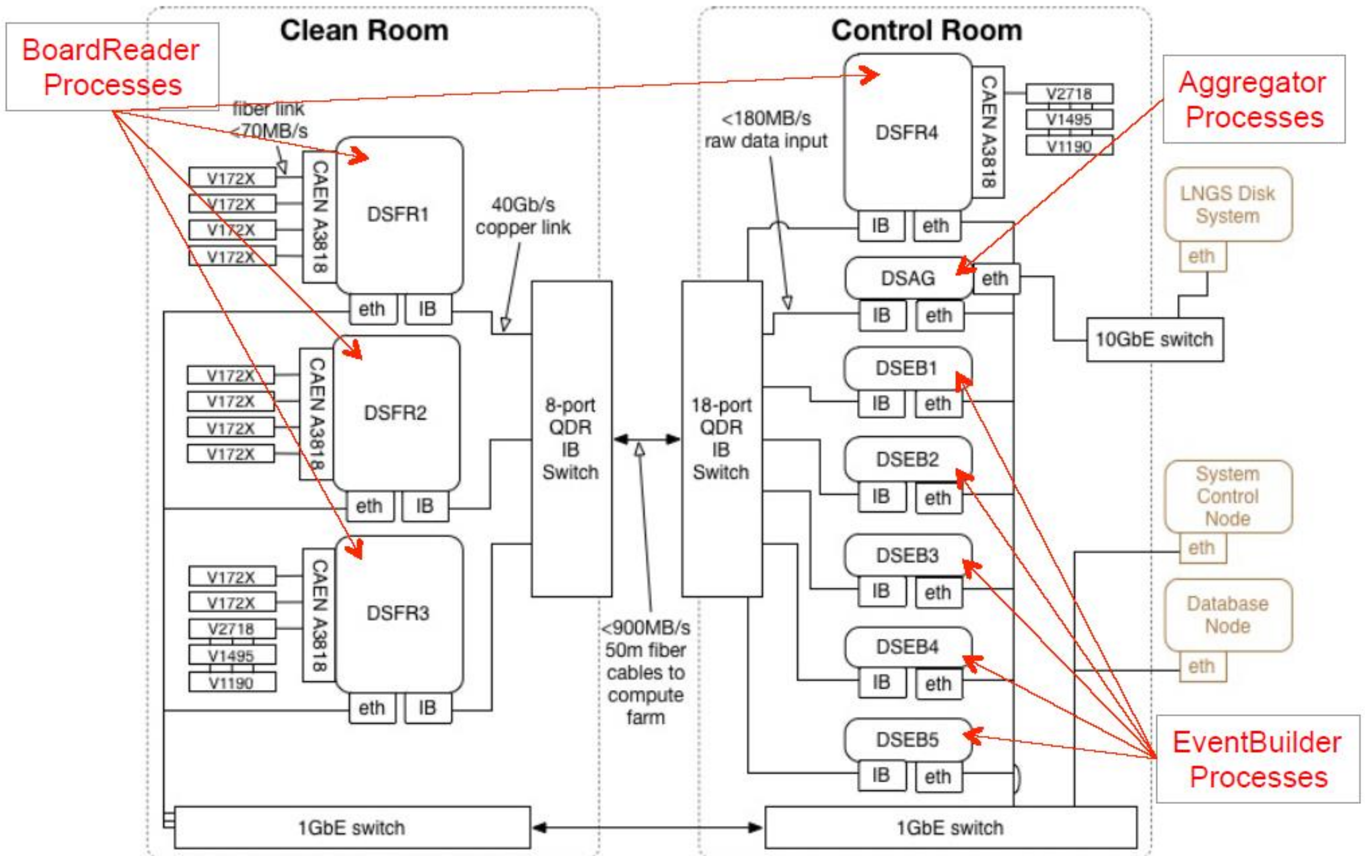
- Wrapper script around *mpirun* command



#466: Jim Kowalkowski et al. *The artdaq Data Acquisition Software Toolkit*

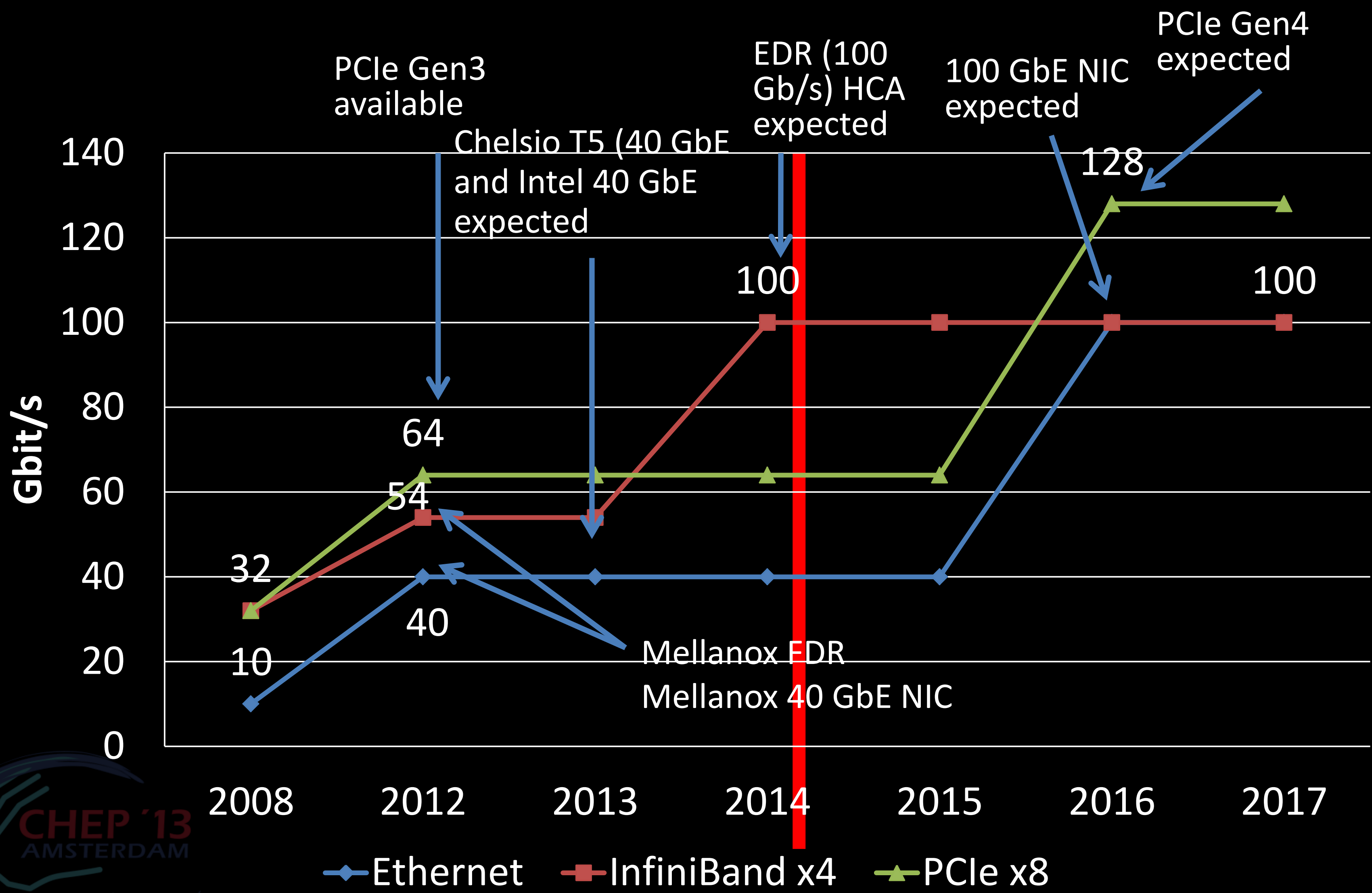
Re-use of commodity networking and hardware
Infiniband (40Gbps) and Ethernet (now: 10Gbps)

artdaq for DarkSide-50



#466: Jim Kowalkowski et al. *The artdaq Data Acquisition Software Toolkit*

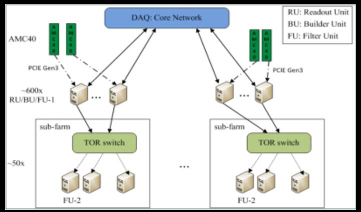
Evolution of host network cards



Graphic by Niko Neufeld (CERN) at Joint DAQ@LHC workshop, March 2013, https://indico.cern.ch/event/217480/session/1/contribution/40

Commodity components everywhere

Example: LHCb



- Emergence of 'low-cost' 32 **T**bps DAQ
  - Infiniband (40, 100Gbps) and Ethernet
  - FPGA receiver cards in standard server PCs
  - PCIe Gen3 fast (and simple!) enough for that
- Utilize the (expensive) network full-duplex & leverage available CPUs in read-out systems also for building & filtering

Example: Alice, LHCb

- Network speeds ~ 100Gbps in 2016? Needed since in 2018+ both Alice & LHCb want to go triggerless!

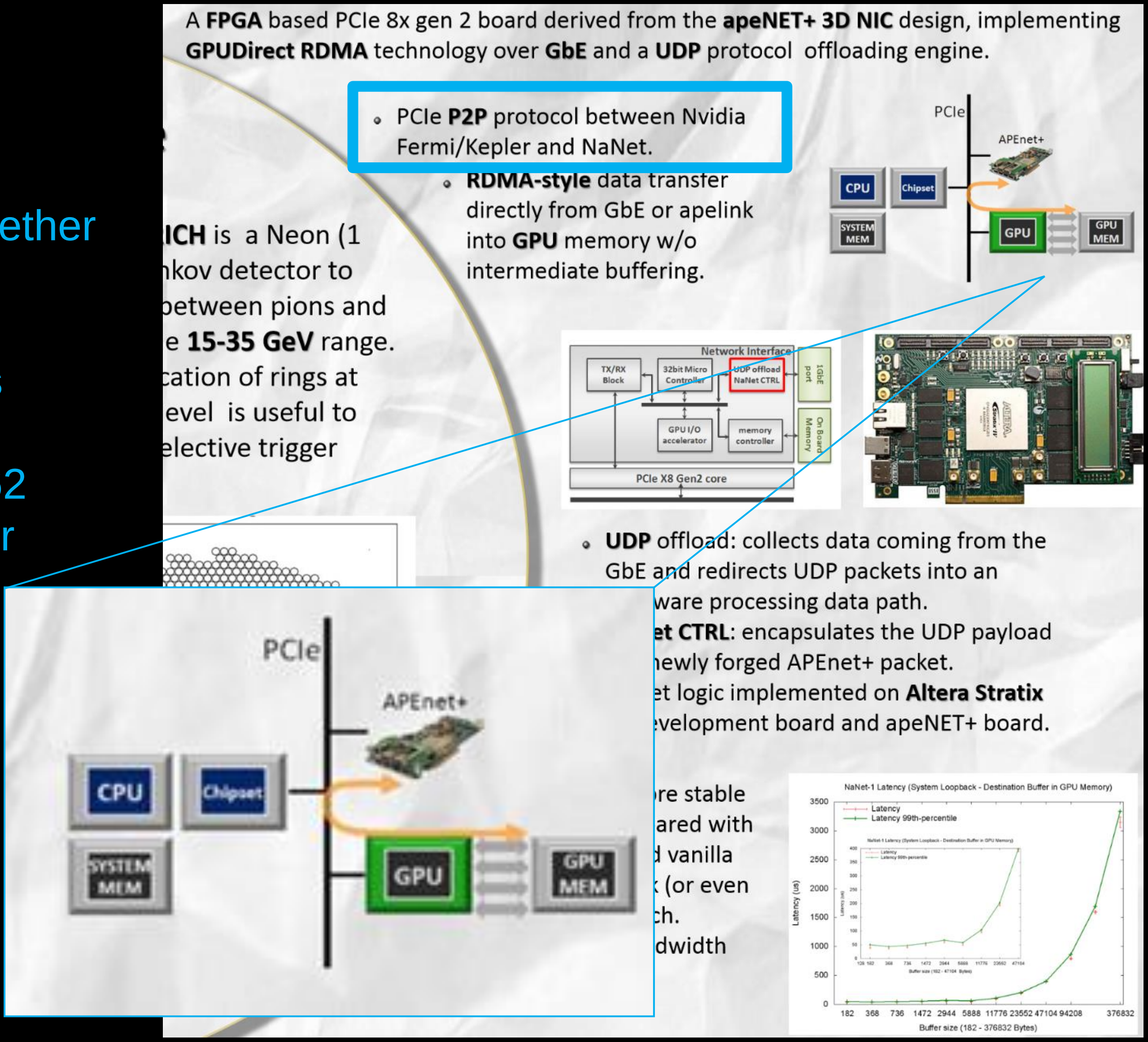
CHEP '13 AMSTERDAM

And the introduction of GPGPUs

NaNet

FPGA working together with (Nvidia) GPU in real-time over PCI2 Gen2 x8 bus

field-tested in NA62 for real-time trigger

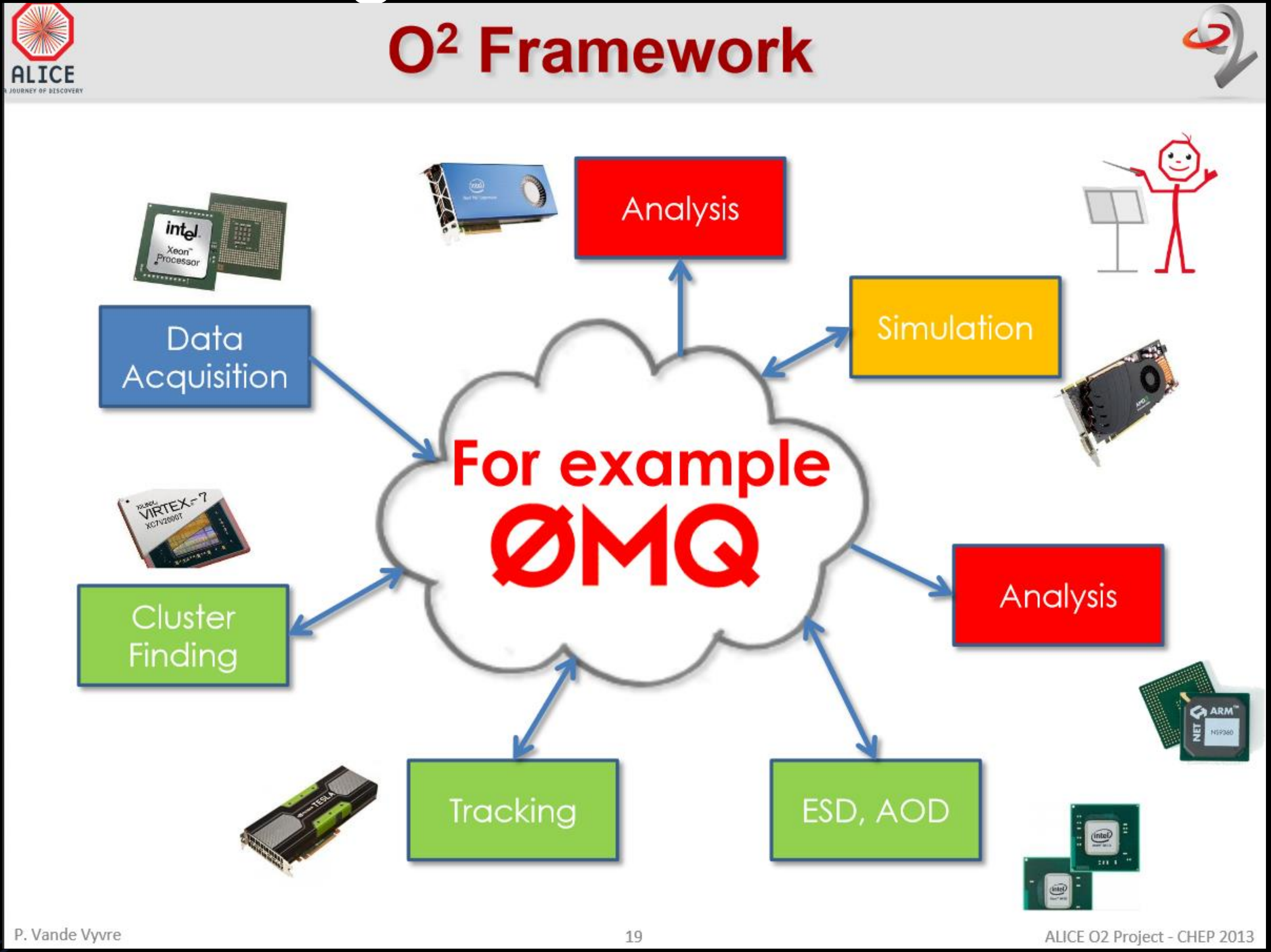


See also #297 H. Valerie *GPU Enhancement of the High Level Trigger to extend the Physics Reach at the LHC*

#48: Roberto Ammendola *GPU for Real Time processing in HEP trigger systems*

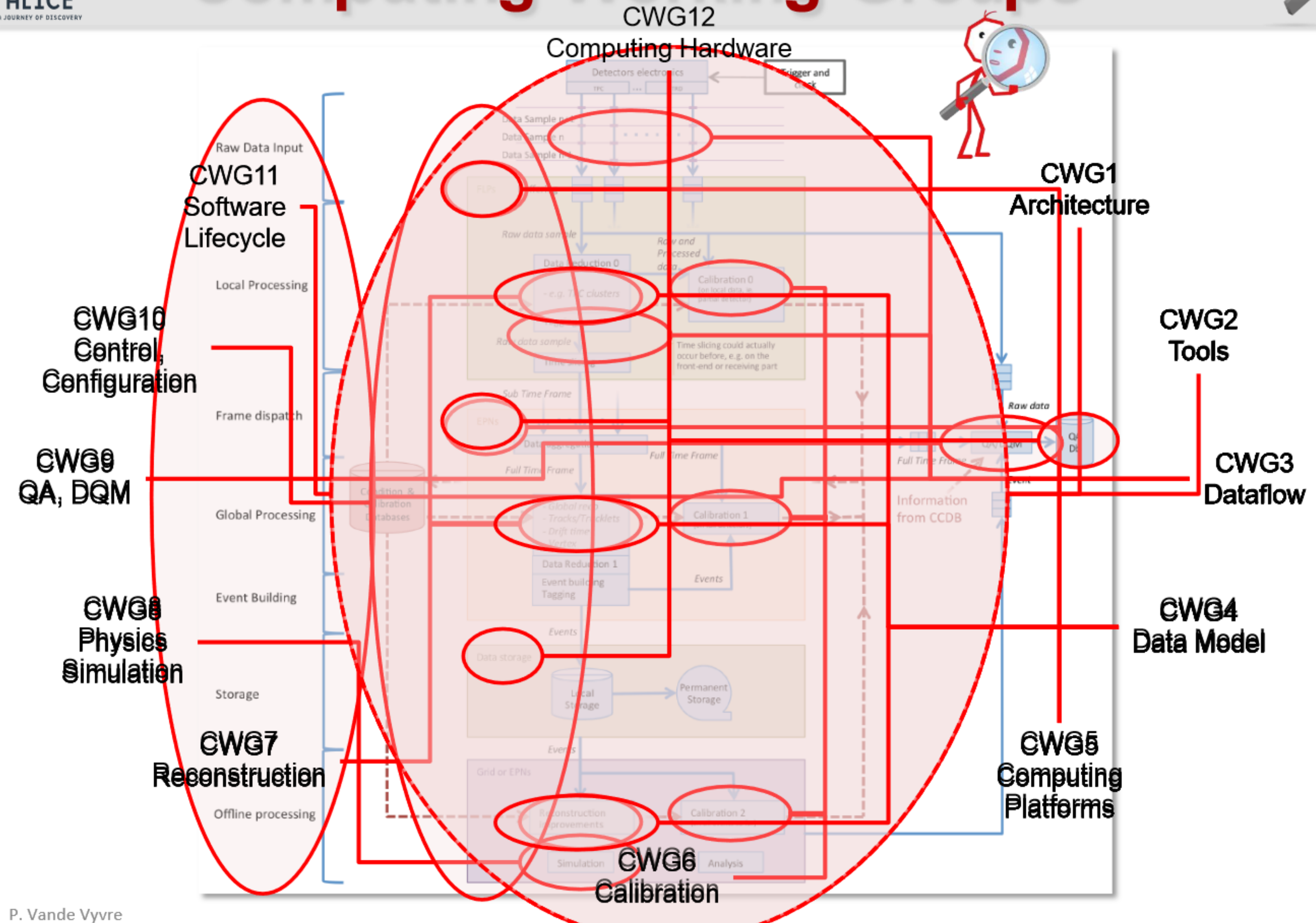
On-line moving off-line - or vice-versa?

Alice in 2018? A single framework
for DAQ, HLT and Off-line processing



#389: Pierre Vande Vrye *O2 a novel combined online and offline computing system for ALICE after 2018*

Computing Working Groups



Geant 4+

Concurrent event processing

~~Russian Roulette, MC and pre-mixing of min-bias hits~~

SIMULATION & EVENT PROCESSING

CHEP ´13
AMSTERDAM

Geant4 10+ exploiting multi-threading

Multi-threading

Porting applications …

- Few changes needed in user code:
  1. Change `main()` to use `G4MTRunManager` – **one line**
  2. Create Sensitive Detector & Field in a new method
  3. Adapt to **per-event RNG seeding** (potential change)
  4. Check User 'Action' classes (Step, Track, Event)
- Choice - handling Output: per thread or accumulate ?
  - Geant4 automatically performs reductions (accumulation) when using scorers or `G4Run` derived classes
- Testing
  - Check output of runs – MT vs 1-thread vs Sequential

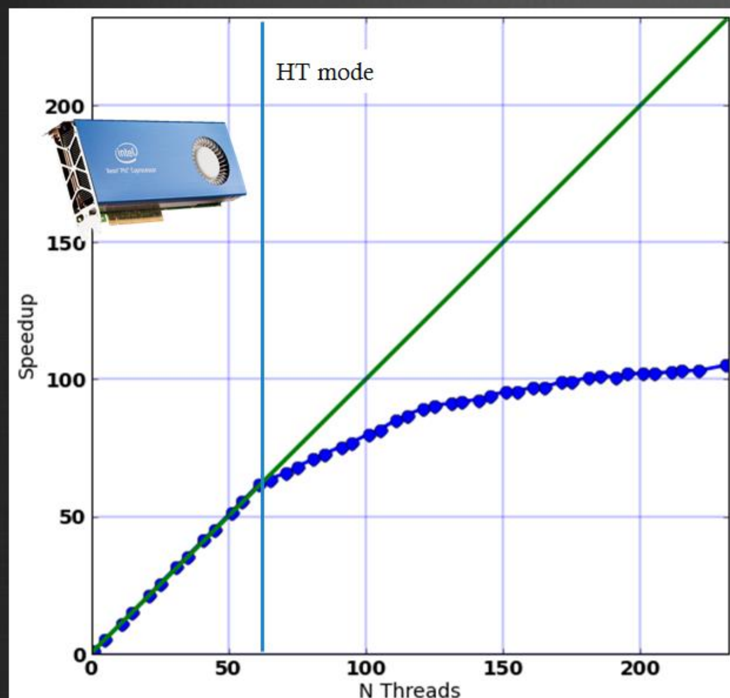
See: https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4MTForApplicationDevelopers

CHEP ´13 AMSTERDAM

Spending time parallelizing pays off!

Multi-threading

Performance – 2/4



- Intel® Xeon Phi™ coprocessor (MIC) (*)
  - 60 cores (4 HW threads each), 16Gb RAM
  - Excellent results: additional factor ~2 in events produced w.r.t. host only
  - Confirmed good scalability up to 240 threads
  - Full physics: 50 GeV pions with B-field on
  - Reduced use of memory
    - (see next slide)

(*) Analysis on full-CMS benchmark on latest September development release by A.Dotti, SLAC

Event processing concurrency

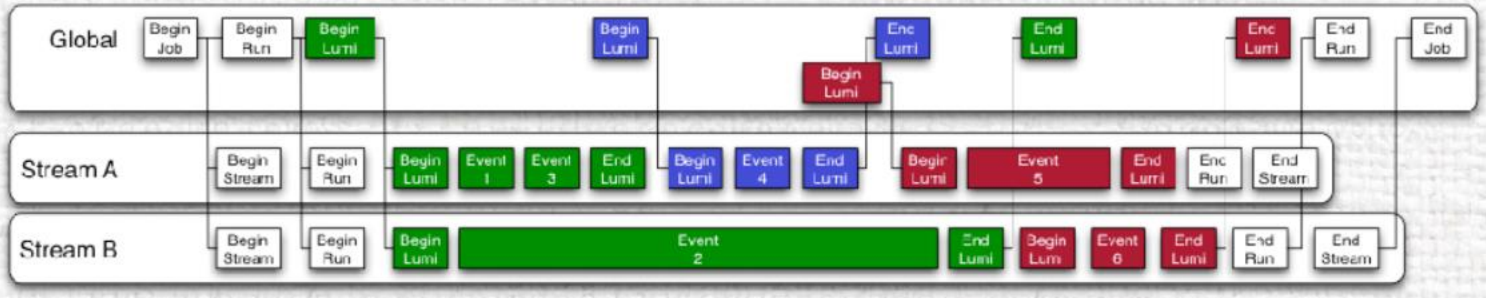
2012 prediction: declare complete success

2013 reality: many different approaches

- CMS: multiple events in parallel
- Gaudi: multiple algorithms in parallel
- FairRoot: multi-process with IPC

using Intel Threaded Building Blocks (TBB) &c

Concurrent Transitions



Global

Sees transitions on a 'global' s

see begin of Run and begin of

sees end of Run and end of Lu

Multiple transitions can be ru

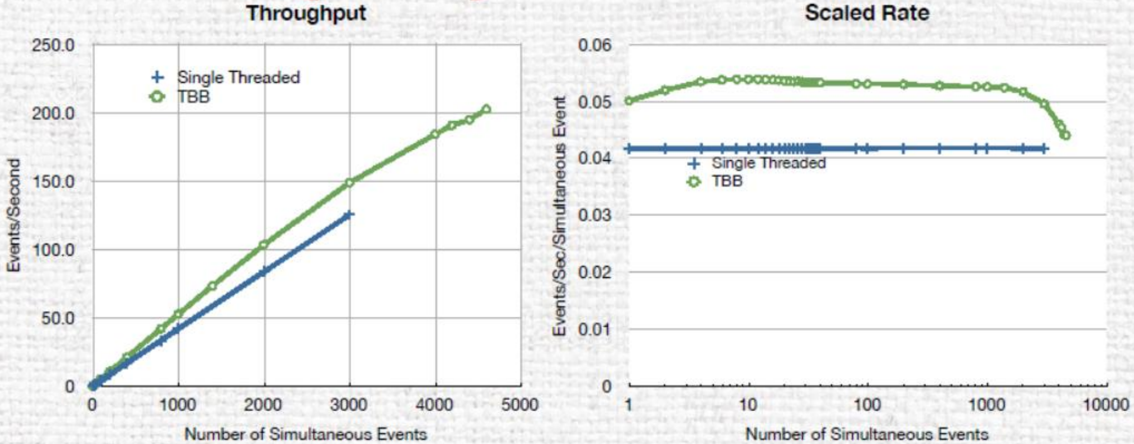
Events are not seen 'globally'

Stream

Processes transitions serially

begin run, begin lumi, events,

Scaling: Infinite Cores



32 core AMD Opteron Processor 6128 w/ 64GB RAM

All modules are calling usleep

TBB stops perfect scaling around 2000 simultaneous events (se)

Is using 1.3 threads/simultaneous event

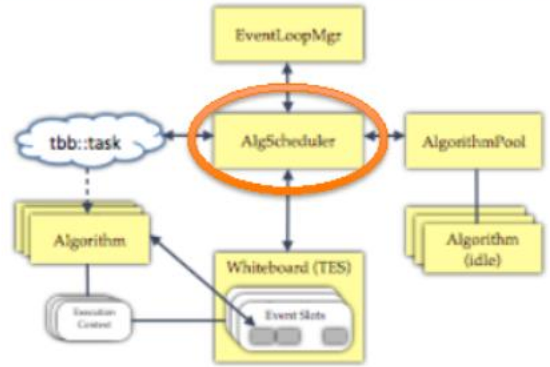
Single threaded framework hits memory limit at 3000 se

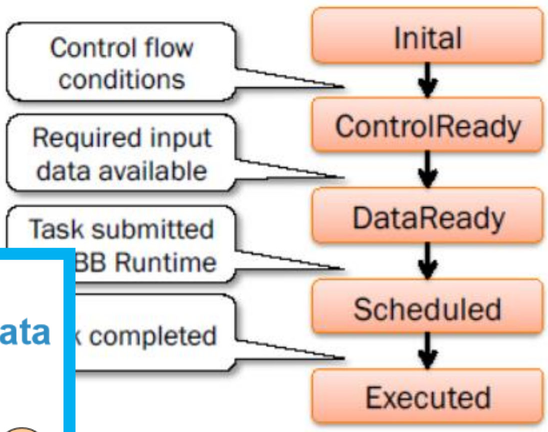
Gaudi: added task-level concurrency

The Forward Scheduler

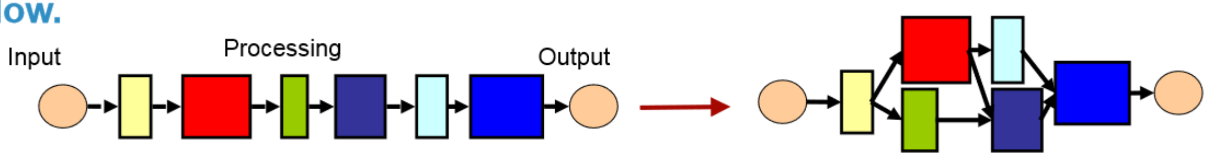
Keeps the state of each algorithm for each event

- Simple finite state machine
- Receive new events from loop manager
- Interrogate Whiteboard for new DataObjects
- Pull algorithms from AlgorithmPool if they are available
- Prepare a tbb::task for execution



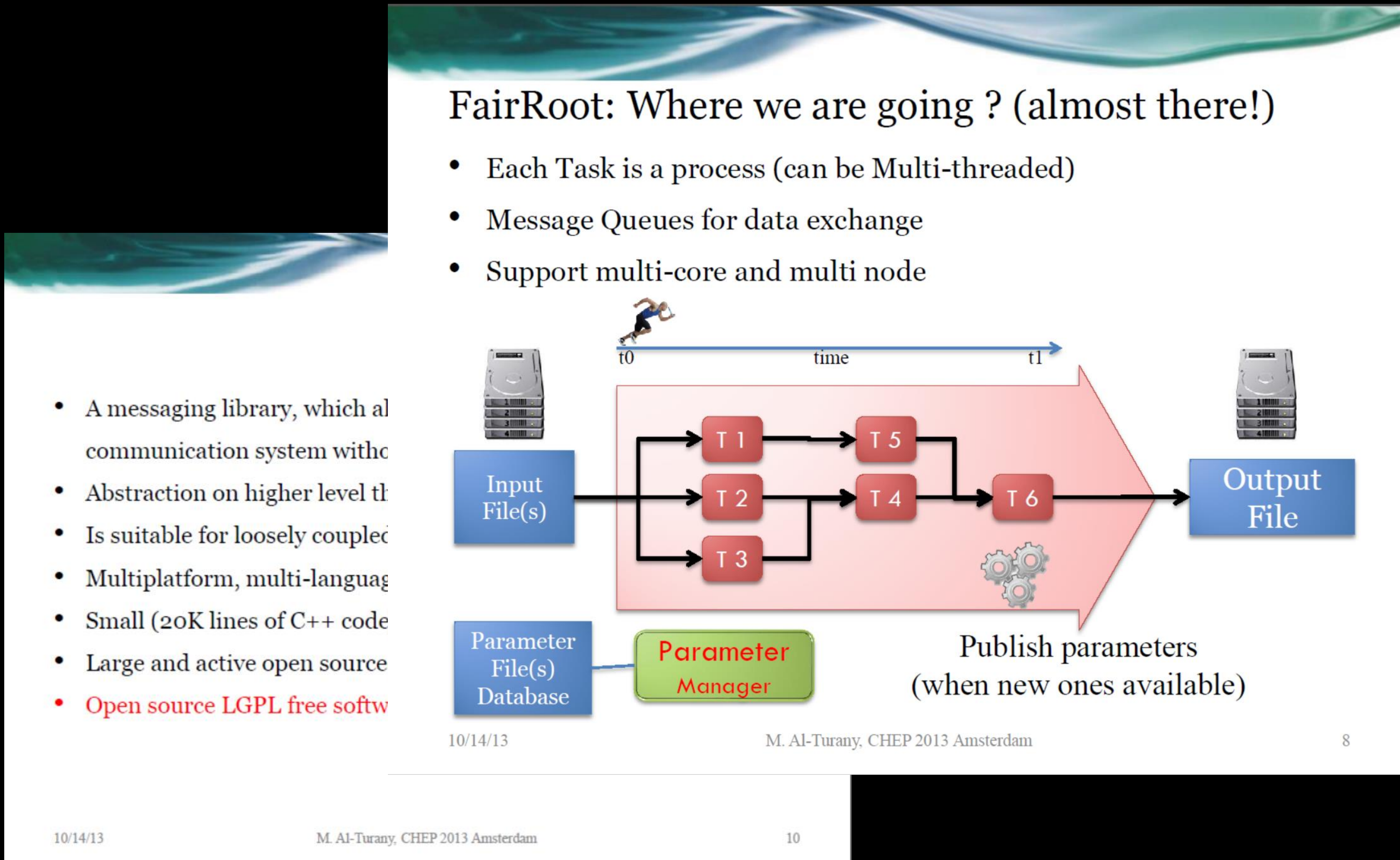


Resolve these dependencies automatically.
Run everything in parallel that isn't constrained by control flow or data flow.



#203 Benedikt Hegner: *Introducing Concurrency in the Gaudi Data Processing Framework*

Or a multi-process approach with IPC



#408 Mohammed Al-Turany *Extending the FairRoot framework to allow […] free streaming data*

Algorithms

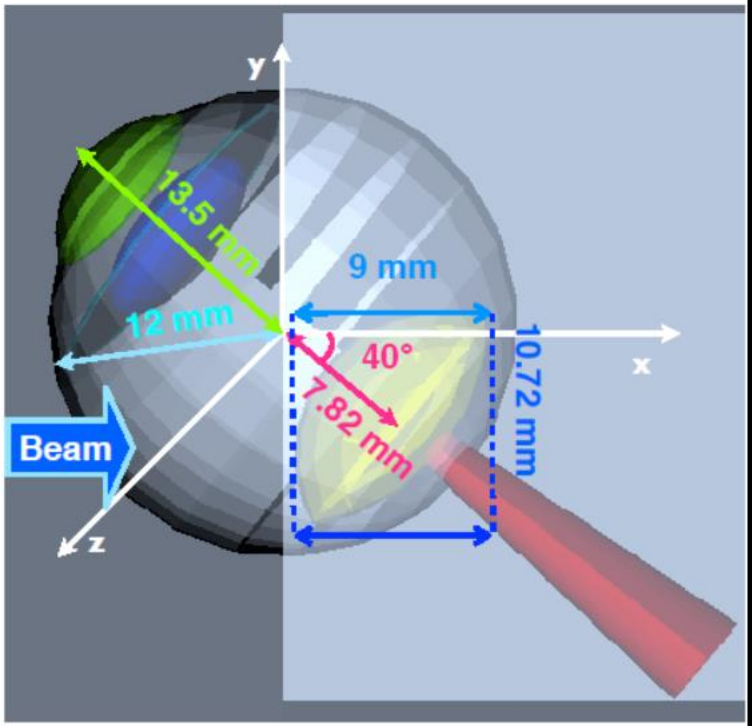
- Many talks from different collaborations
- Many algorithms are very specific designed for one experiment
  - CBM: Selected event reconstruction algorithms
  - Belle II: Track extrapolation using Geant4E
  - ....
- There are also developments which should be usable for a larger user community
  - CLAS: Bayesian Data Analysis in Baryon Spectroscopy
  - PANDA: Common Partial Wave Analysis Framework
  - ....
- How to find such developments which could be (re)used?
  - Database with information?
  - Web page?
- How can we come to a situation like with common frameworks?

For dealing with MC pile up and 'pre-mixing' of minbias MC, see hidden slides

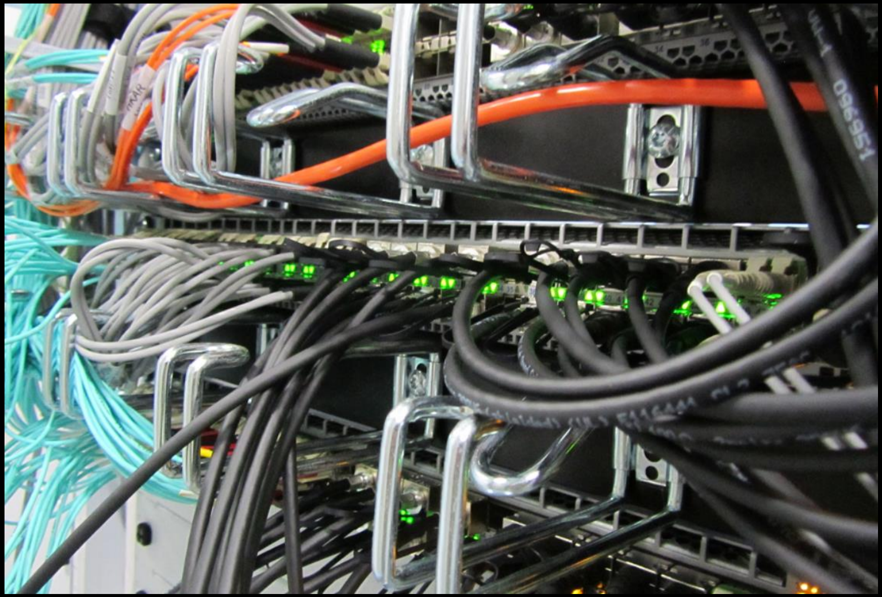
Simulation and MC outside HENP

The eye detector

- Eye anatomy deeply studied and a geometric schematization realized
- Accurate reproduction of all eye-components in the G4 simulation
- Dimensions parameterised as a function of the sclera radius
- Rotation possible to misalign tumour and sensitive sub-components



#166 Adele Rimoldi: *Geant4 studies of the CNAO facility system for hadron therapy treatment …*



Infrastructure, sites, and virtualisation

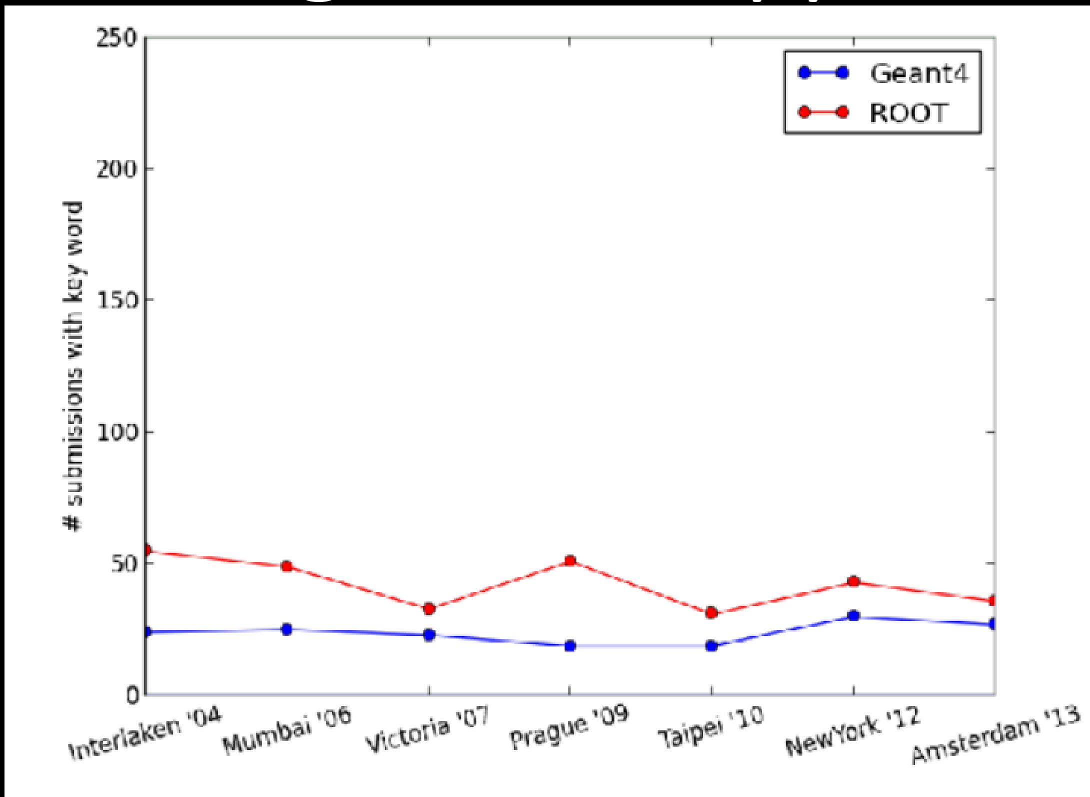
Experiment data models, data handling, and computing models

Data driven analysis

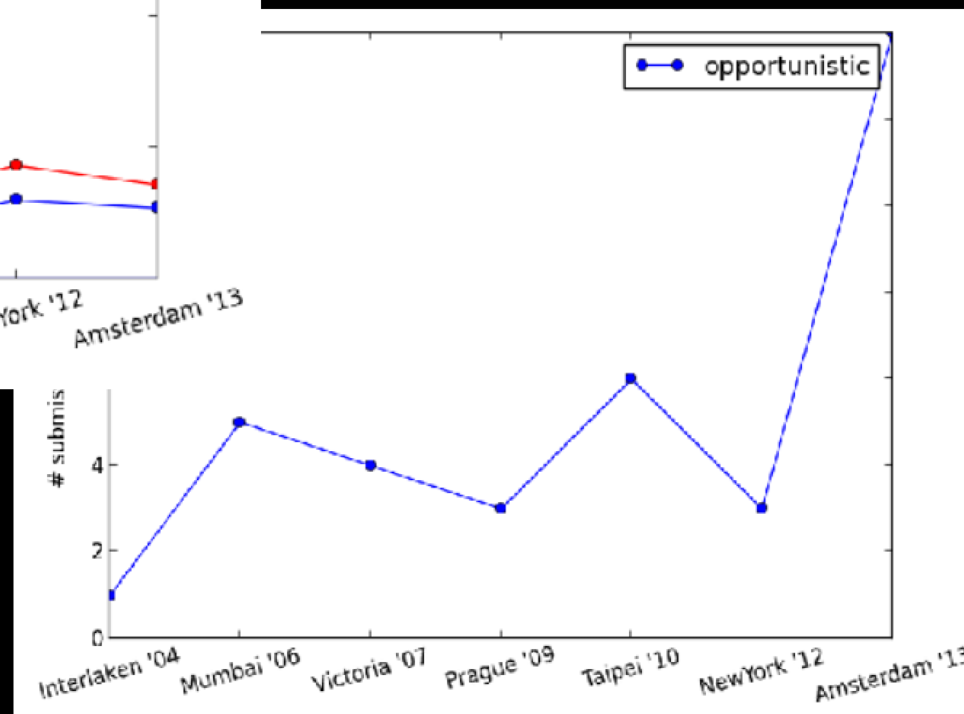
DISTRIBUTED PROCESSING AND DATA HANDLING

CHEP ´13 AMSTERDAM

Emergence of opportunistic computing



- Steady rise of virtualisation
- IaaS resources used via same overlay mechanisms used in grids

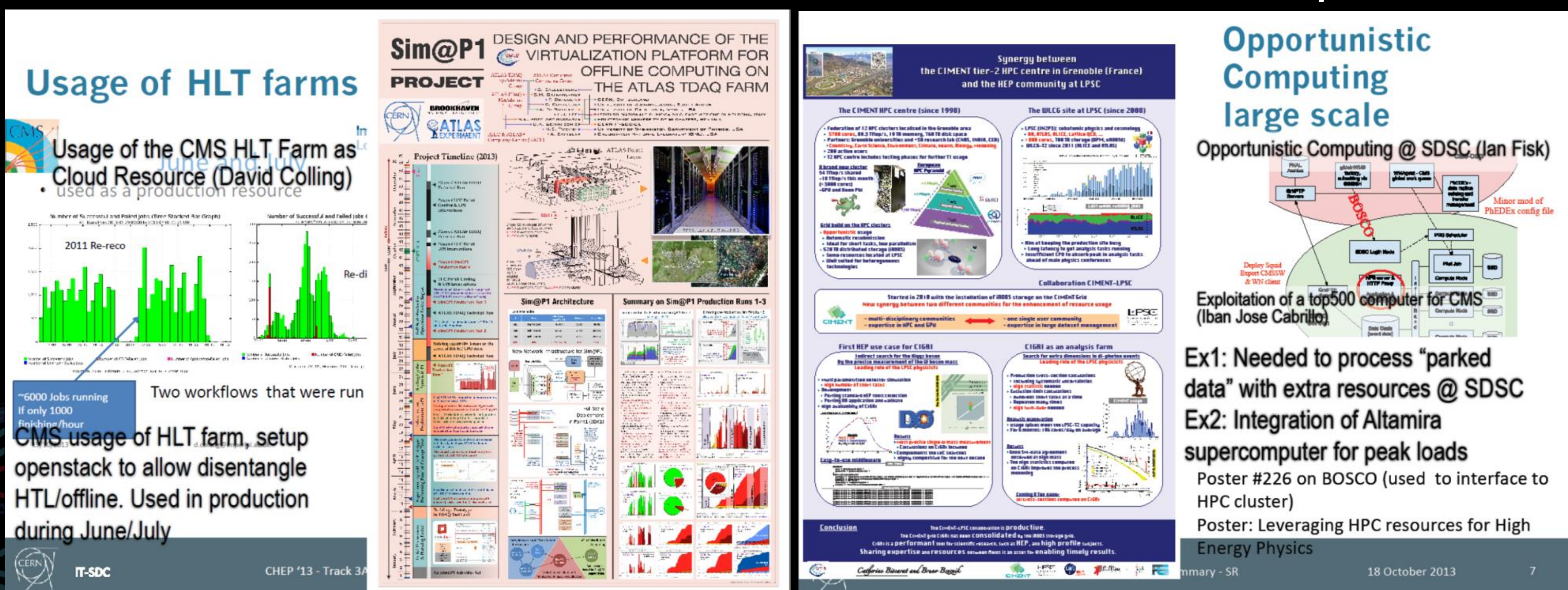


Number of abstract with given keyword vs. time

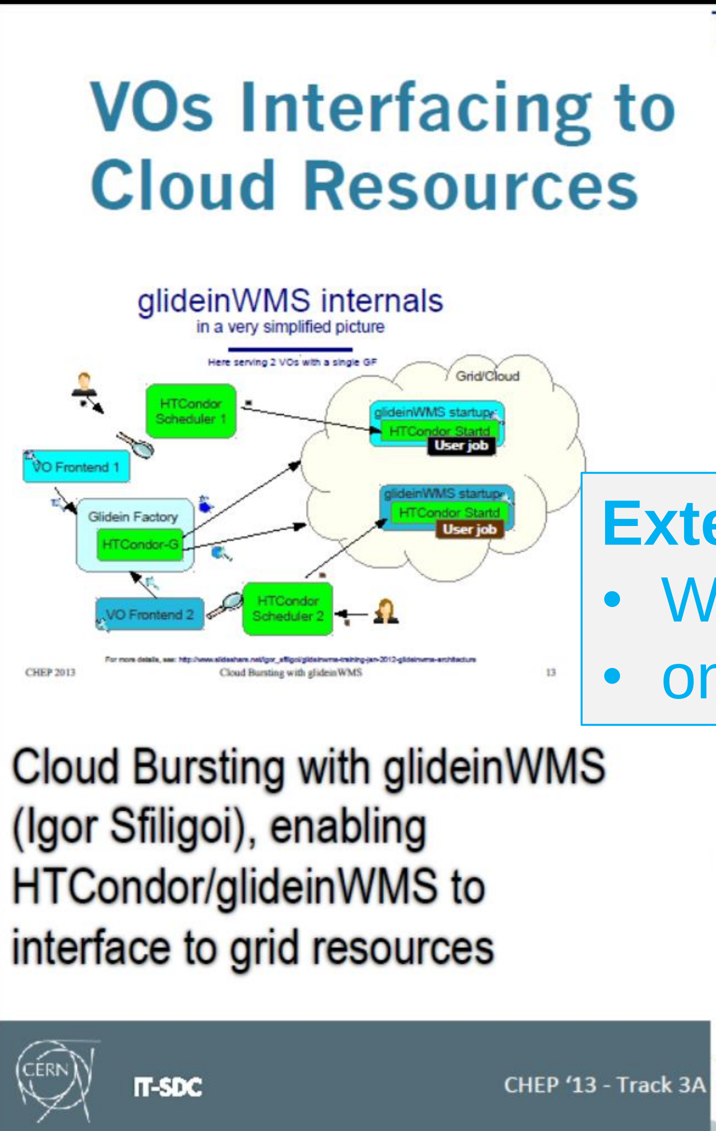
Stefan Roiser, Davide Salomoni: *Track 3A summary talk*

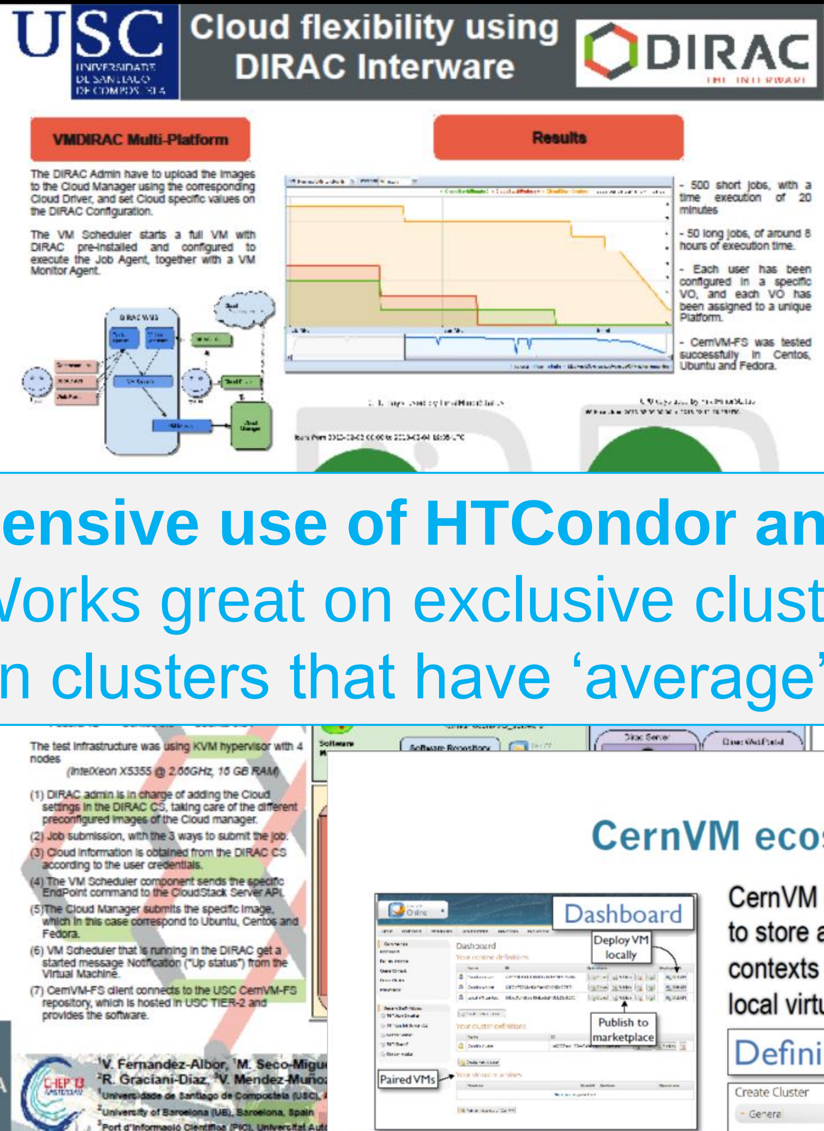
More ‘chaotic’ data organisation

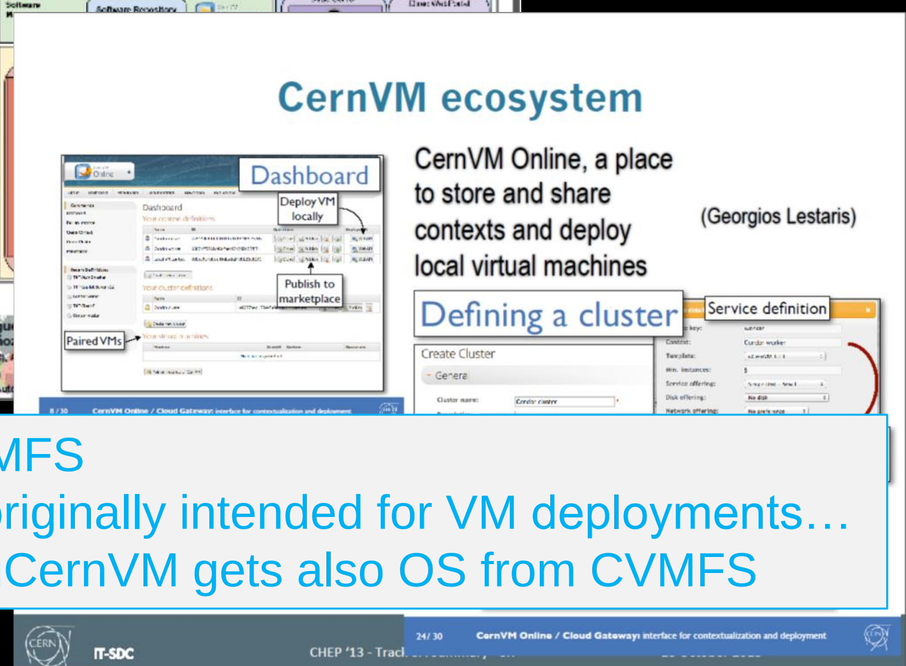
- Left-over CPU resources used from HPC centres, desktop grids, HLT farms, …
- Data pre-placement is gone
  - At least for the ‘small’ data sets of today



Stefan Roiser, Davide Salomoni: *Track 3A summary talk*







Extensive use of HTCondor and 'elastiq'

- Works great on exclusive clusters, and
- on clusters that have 'average' occupancy

CVMFS

- originally intended for VM deployments…
- uCernVM gets also OS from CVMFS

Stefan Roiser, Davide Salomoni:
Track 3A summary talk

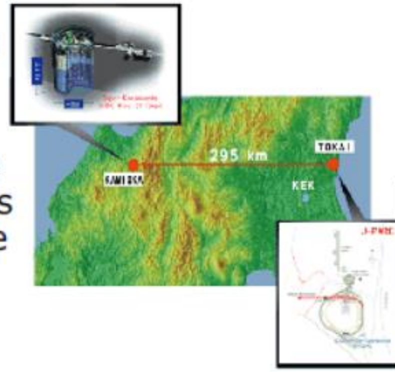
Adapting Experiment Frameworks

- DIRAC emerges as multi-user framework
  - CLIC/ILC, BelleII, BESIII, …
  - Actively integrate non-HEP use cases
- Atlas & CMS compute
  - Workload management stays separate
  - But job management maybe merged in PanDA
  - data model: file-level granularity (Atlas Ruccio)
  - Leverage new services like FTS2 and GFAL2
- But: outside the 'LHC bubble'
  iRODS and databases are the popular choice
  and SAM still lives!

KEK iRODS Data Management System

Data Management for T2K

- Tokai to Kamioka (T2K) Neutrino experimental group
- The experimental data is stored to KEK storage
- The group needed to provide an easy way to quickly access data collected to evaluate the quality of the data from outside of KEK
- iRODS provided the solution

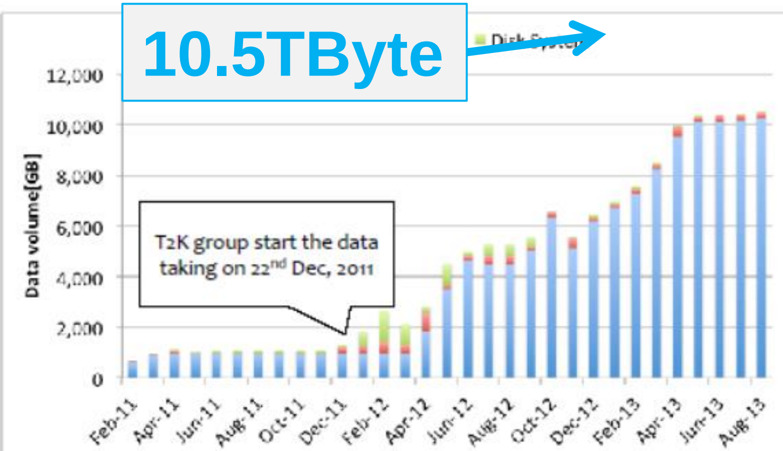


Also multi-PByte iRODS systems

Wataru Takase

IROD: Integrated Rule-Oriented Data System

Amount of data in KEK-T2K



10 13

Federation with QMUL

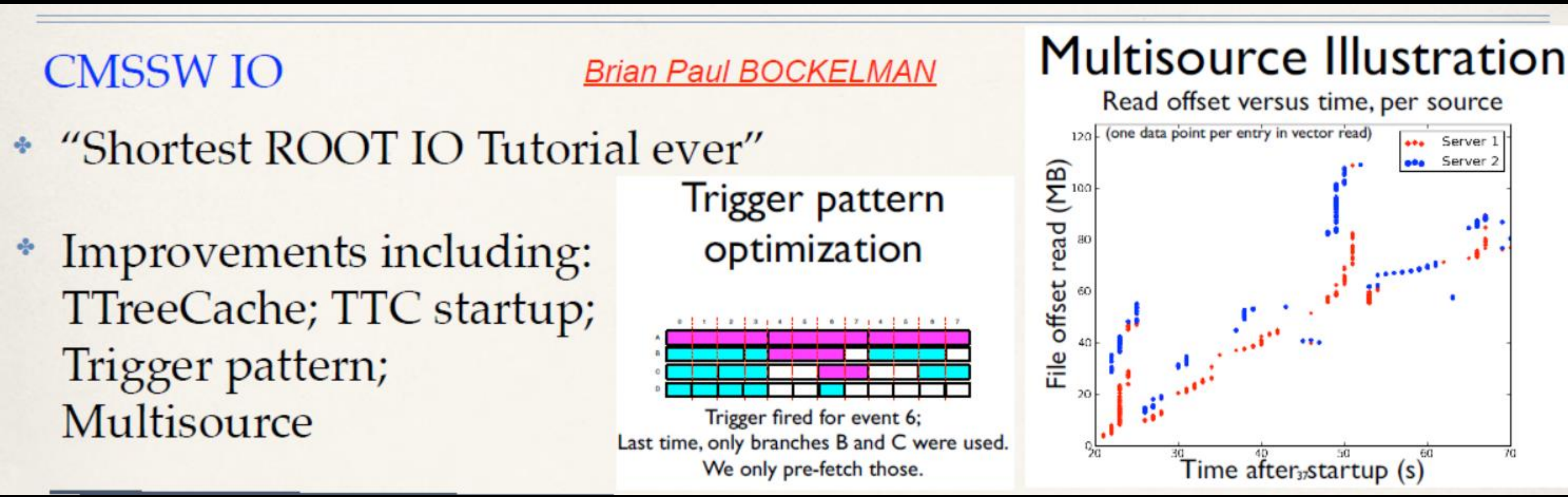
- Data replication among 2 sites
- Share each site data

9

iRODS also typical general-purpose solution for groups in OSG

Nurcan Ozturk:
Track 3B summary talk

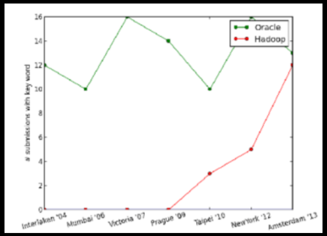
And outside the box …



- Long-latency (remote) file access ***can*** be efficient!
  - You need to tune RootIO to make it work well
  - Parallel transfers (multi-source) also helps
  - Read how to do it right in Brian's talk!

#160 Brian Bockelman: *Optimizing High-Latency I/O in CMSSW*

Data access patterns

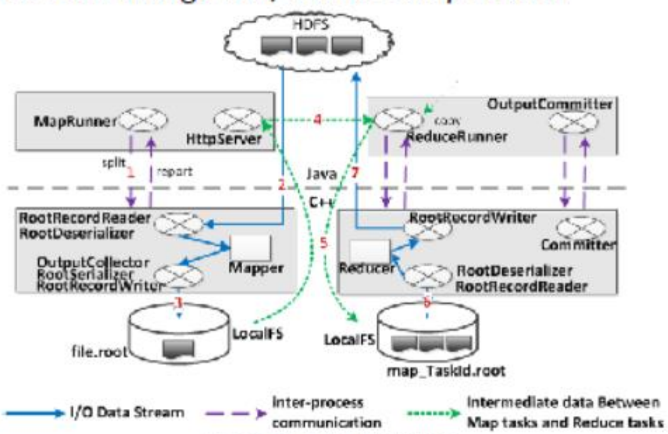


- Map-reduce ('hadoop') distributed data
  - Brings compute jobs to prelocated data on cluster
  - Distribute (multiple copies of) data across nodes
- Merger of Hadoop (Java) and C++ code
  - Demonstrated for Root with **no changes to Root** (but many to Hadoop …)
- 'NoSQL' databases
  - On the rise, but need to pick use carefully!
  - Useful in niches (e.g. monitoring-data collections)
  - Not better 'over-all' than conventional DBMS…

Hadoop

HEP MapReduce Procedure

- HEP MapReduce (different from Internet applications):
  - Java side is in charge of job splitting and scheduling
  - C++ side is in charge of I/O and computation



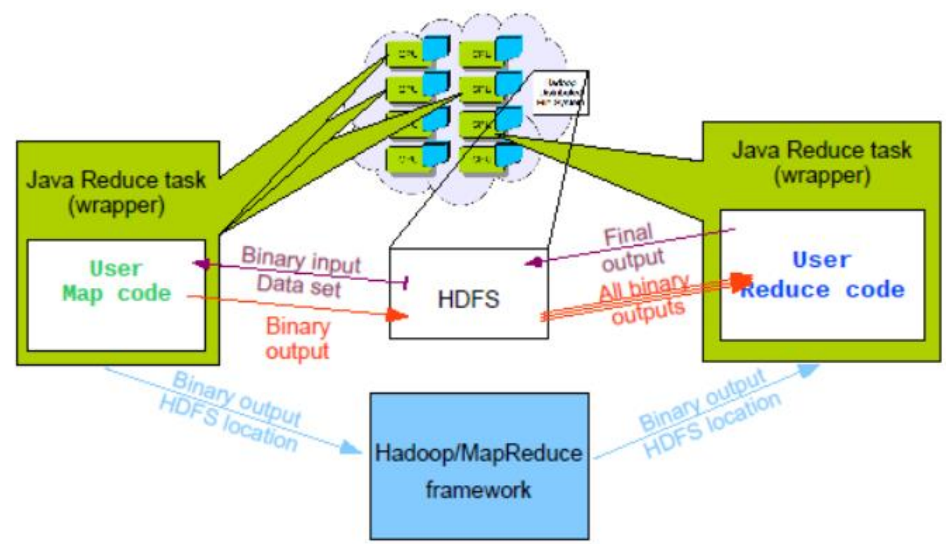
2013-10 INSTITUTE OF HIGH ENERGY PHYSICS

BESIII analysis on Hadoop (Sun Gongxing), wrote ROOT C++ classes to interface to Map/Reduce via libhdfs

ROOT on Hadoop (Stefano Russo), uses file == chunk on HDFS, wrappers around Map/Reduce -> no ROOT code changes

Under the hood..

```
# hadoop run RootOnHadoop "user Map code" "user Reduce code" "HDFS input dataset" "HDFS output location"
```

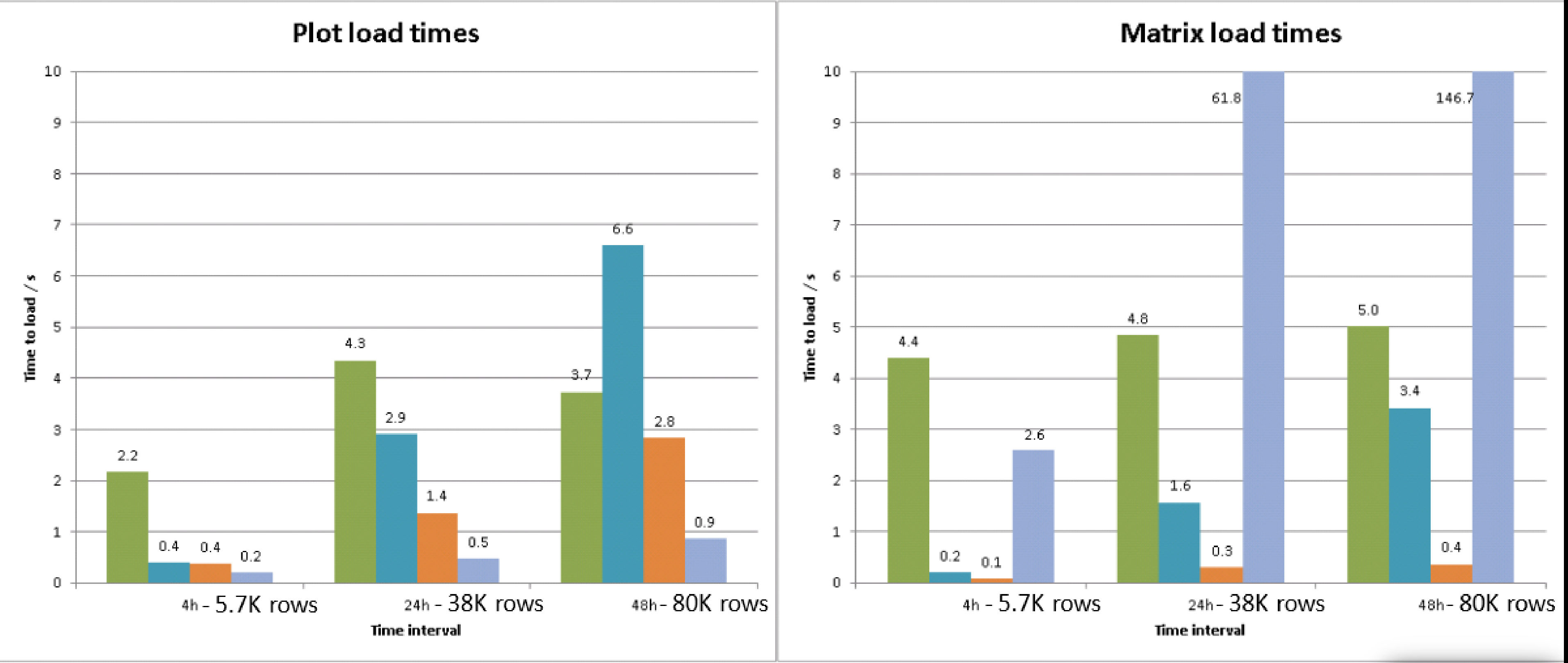


Stefan Roiser, Davide Salomoni:
Track 3A summary talk

OG: Oracle Grouping,
ENG: ElasticSearch NoGrouping,
EIG: ~IndexGrouping;
EQG: ~QueryGrouping

'NoSQL'

Data Out



- ENG is much faster than Oracle for small row counts but won't scale
- EIG is faster than Oracle in all cases but inflexible
- EQG is much faster for few distinct grouping values but won't scale

Computing model evolution

- All LHC expts revised the computing model
  - Atlas mimicking the CMS analysis trains
  - CMS takes an 'interesting' approach to multicore – essentially building single-node-multi-core miniclusters for their own
  - LHCb to leverage new DIRAC features, and gets rid of first-pass reco in 2015!
- More generic & pre-existing tools used
  - Mainly by non-LHC experiments
- Beyond HEP: data-driven analysis for SKA+

Belle II Computing Model

Distributed Computing System

- Based on existing, well-proven solutions plus extensions for Belle II
- DIRAC for job management
- AMGA for metadata
- CVMFS for software distribution
  (thanks to CERN and Steve Traylen for providing the Stratum-0 server, and to GridKa for the stratum-1 server)

DIRAC THE INTERWARE

ARDA METADATA CATALOGUE PROJECT

CernVM File system

root/svn/trunk/grid/BelleDIRAC

FrameworkSystem/	4326 (9 months ago) by myco: basic sites management service for BelleDIRAC
Web/	5519 (4 months ago) by hideki: remove unused AMGA API
WorkloadManageme...	6098 (2 months ago) by hideki: fix a bug
gbasf2/	6647 (2 weeks ago) by hideki: fix unnecessary AMGA initialization
README	4325 (9 months ago) by myco: init files for BelleDIRAC distribution
__init__.py	6348 (6 weeks ago) by hideki: release for 2nd MC campaign

Thomas Khur

Summary

Belle II

- Belle II will search for New Physics with O(50) times more data than current B factories
- Huge data volume is a challenge for the computing
  - Distributed computing system based on existing technologies and infrastructures
  - Workflow abstraction with projects and datasets
- First two MC production campaigns this year
  - ✔ Belle II distributed computing system works!
  - ✔ Bottlenecks and issues identified
  - Many thanks to technology and resource providers!
- Next steps:
  - MC campaign with more (cloud) sites
  - Further automatize and harden the system
  - Exercise user analysis on the grid

Nurcan Ozturk:
Track 3B summary talk

IceCube Computing Model

IceProd

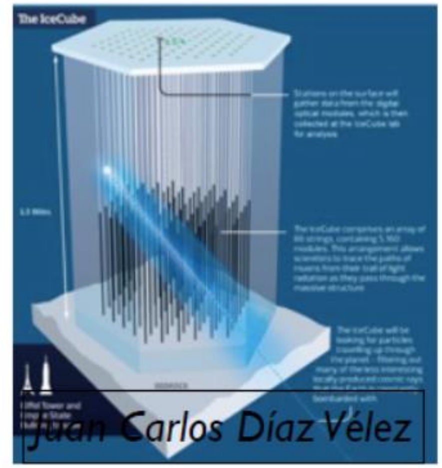
IceProd is a software package based on Python, XMLRPC and GridFTP. It is driven by a central database in order to coordinate, administer and drive production of simulations and processing of data.

It is not a replacement for batch queuing systems or grid middleware.

IceProd runs as a separate layer on top of other middleware and can take advantage of a variety of computing resources including grids and batch systems such as CREAM, Condor, NorduGrid, PBS and SGE

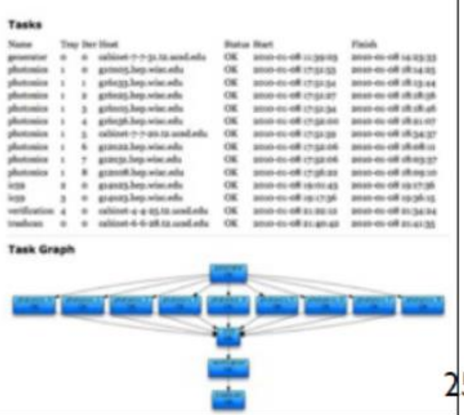
- IceProd
- Condor/gLite/Edg
- Globus/Grid Middleware
- Condor/PBS/SGE
- Operating System

Juan Carlos Díaz Vélez



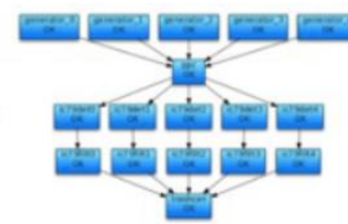
DAG (Directed Acyclical Graph) -based simulation

- Separate simulation segments into tasks
- Assign task to a node in DAG
- Different tasks can have specific hardware or software requirements



GPU-based Production

- DAGs allow for optimal usage of specialized computing resources such as GPUs.
- GPU nodes can be fully dedicated to GPU-tasks.
- Optimal DAG differ depending on the specific application.



Nurcan Ozturk:
Track 3B summary talk

Outside HEP: fully data-driven analysis

TarGet

Astro-WISE information system– fully datacentric

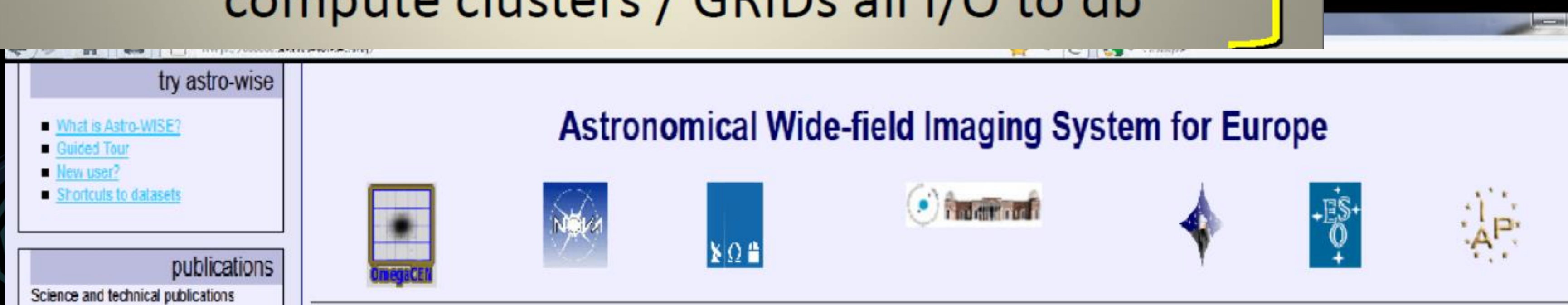
All data beyond pixel data is Metadata

Inherent data lineage and provenance

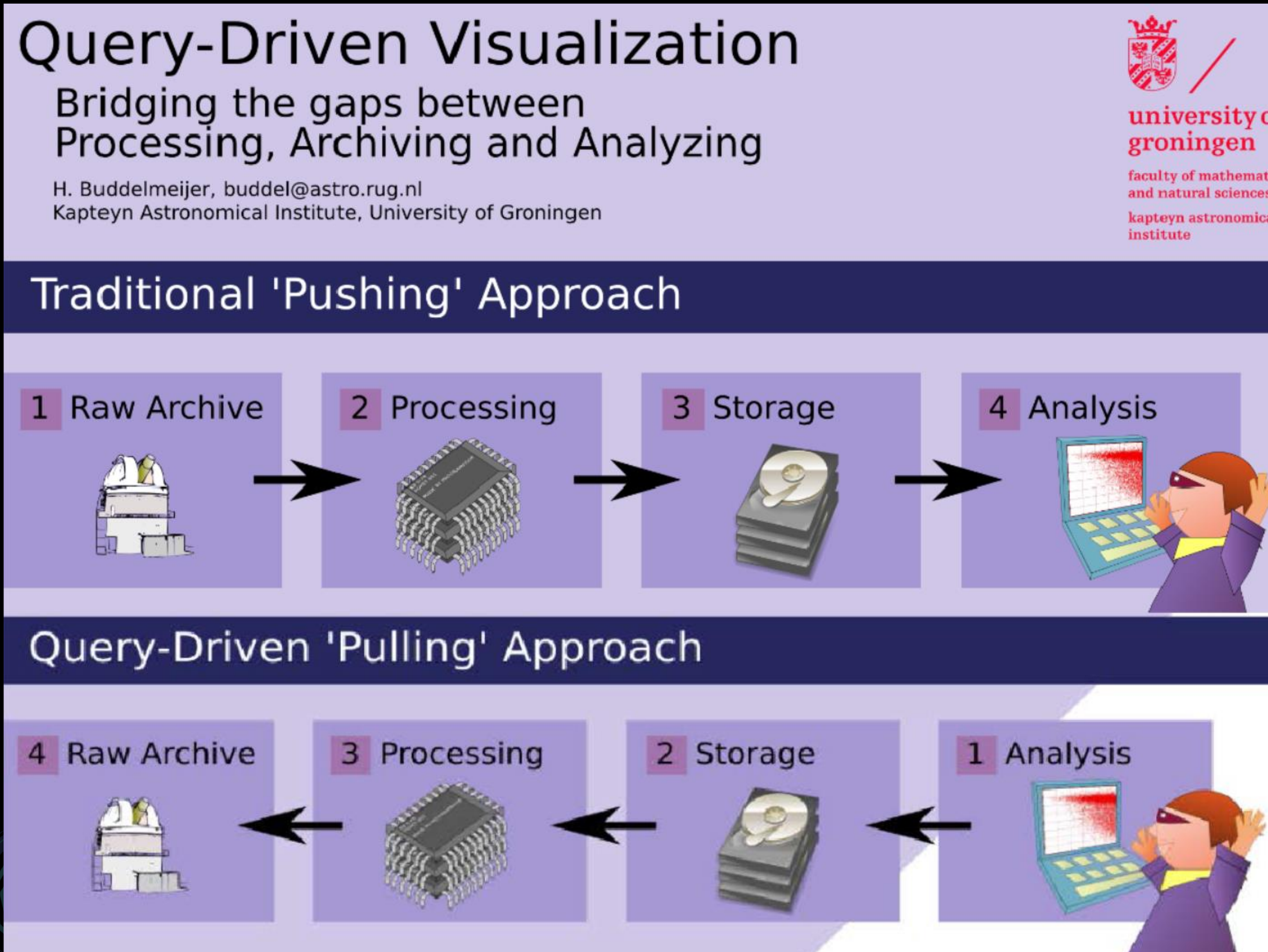
all pixel data <–>data servers

all Metadata <–> database

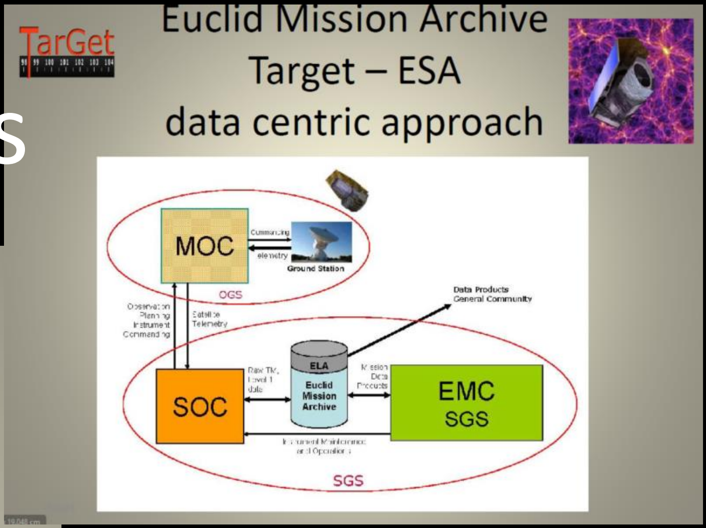
compute clusters / GRIDs all I/O to db

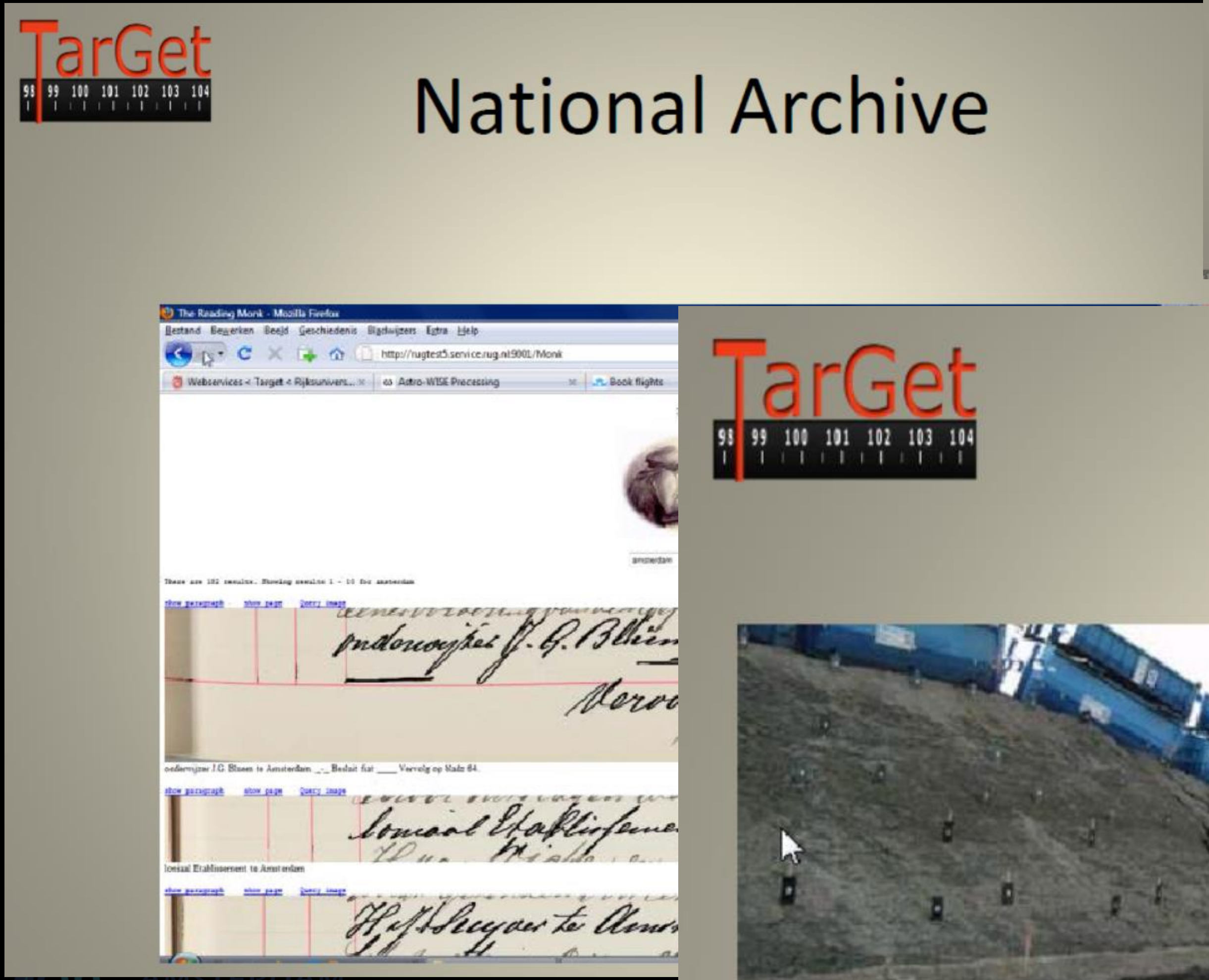


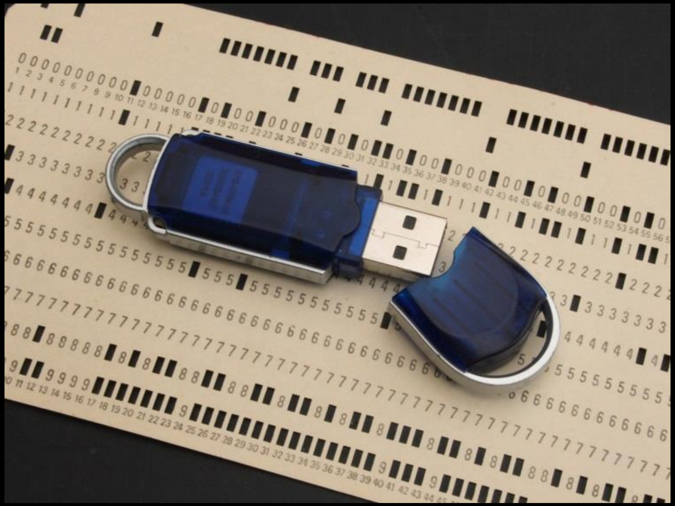
Edwin Valentijn - *Probing Big Data for Answers using Data about Data* – Wednesday plenary


Query-Driven Visualization
Bridging the gaps between
Processing, Archiving and Analyzing
H. Buddelmeijer, buddel@astro.rug.nl
Kapteyn Astronomical Institute, University of Groningen
university of
groningen
faculty of mathematic
and natural sciences
kapteyn astronomical
institute
Traditional 'Pushing' Approach
1 Raw Archive
2 Processing
3 Storage
4 Analysis
Query-Driven 'Pulling' Approach
4 Raw Archive
3 Processing
2 Storage
1 Analysis

Applicable to many domains







Storage large and small

File systems forever

The cheapest storage front-end

DATA STORES, DATA BASES AND STORAGE SYSTEMS

CHEP ´13
AMSTERDAM

Storage evolution: 2 extremes

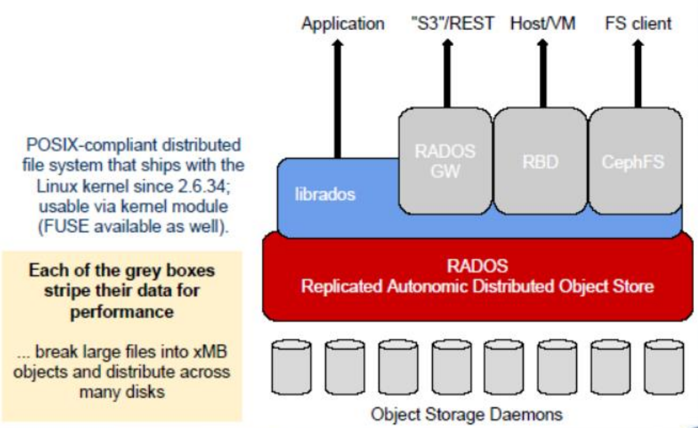
Local Note:
SURFsara and SURFnet soon to announce eduBox here based on ownCloud!

Ceph and owncloud

#68 Dan van der Ster: *Building an organic block storage service at CERN with Ceph*

Ceph's architecture CERN IT Departm

Daniel VAN DER STER



- Large scale test of Ceph at CERN - demonstrates usefulness at least as object store

We are attracting various use-cases

- OpenStack images and volumes
- RBD backends for other storage services (AFS/NFS/DPM)
- Object storage for novel applications: (tape buffer, Zenodo, OwnCloud)

We have very high hopes for Ceph at CERN!

- the design is *interesting*
- the performance so far is adequ
- operationally it is very attractive

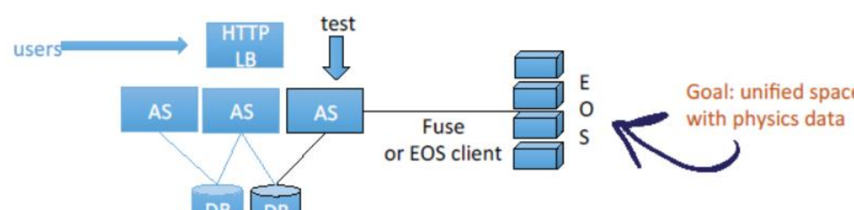
CephFS NOT ready for prime time yet

Everybody wants a filesystem: CephFS will be crucial

- CERN also testing owncloud for dropbox-like *cernbox* - beta service soon.

- Intended final configuration

Jakub MOSCICKI



Wahid Bhimji: *Track 4 summary talk (and upstream contributors)*

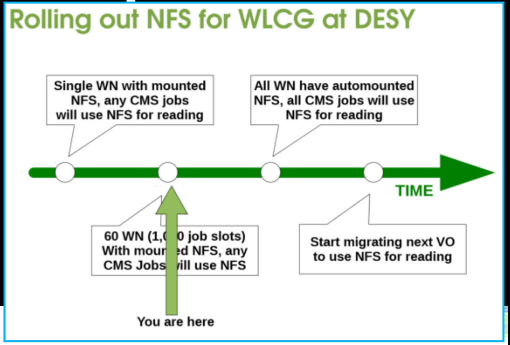
Distributed file and object stores do work – and NFSv4.1 may help as FS

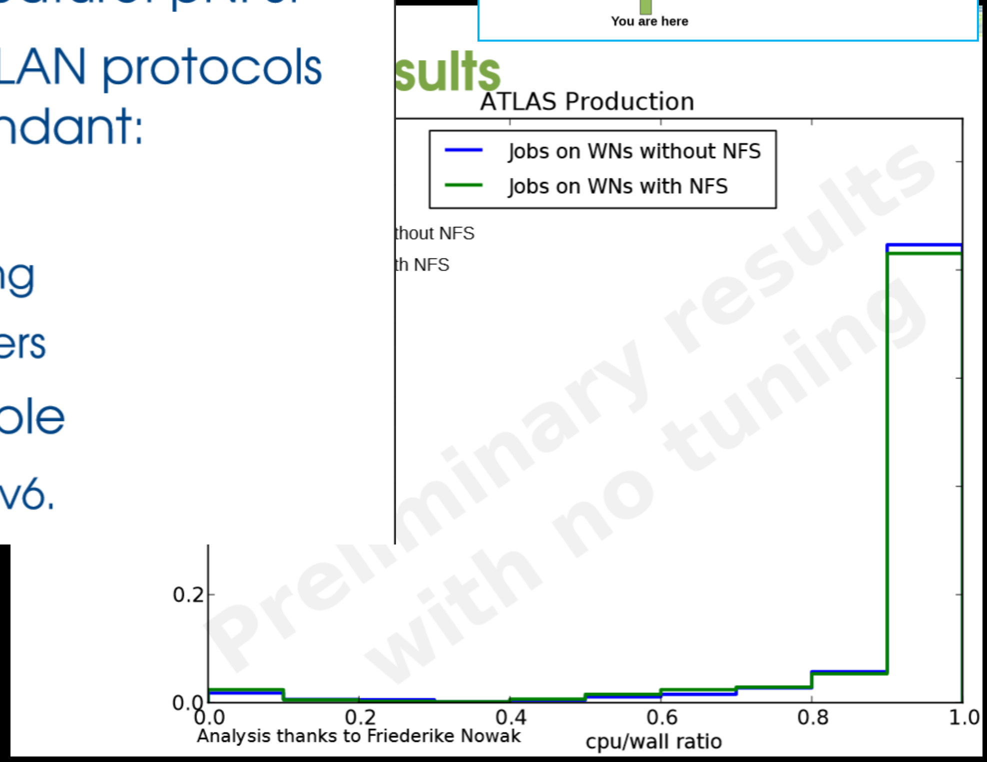
dCache.org

One slide summary: NFS and pNFS

- NFS v4.1 introduced an optional feature: pNFS.
- pNFS means that HEP-proprietary LAN protocols (dcap, xrootd, rfio) become redundant:
  - Don't have to maintain a client
  - Build-in support for client-side caching
  - Lots of exciting innovations from others
- For WLCG, only just become feasible

  requires WNs running Scientific Linux v6.

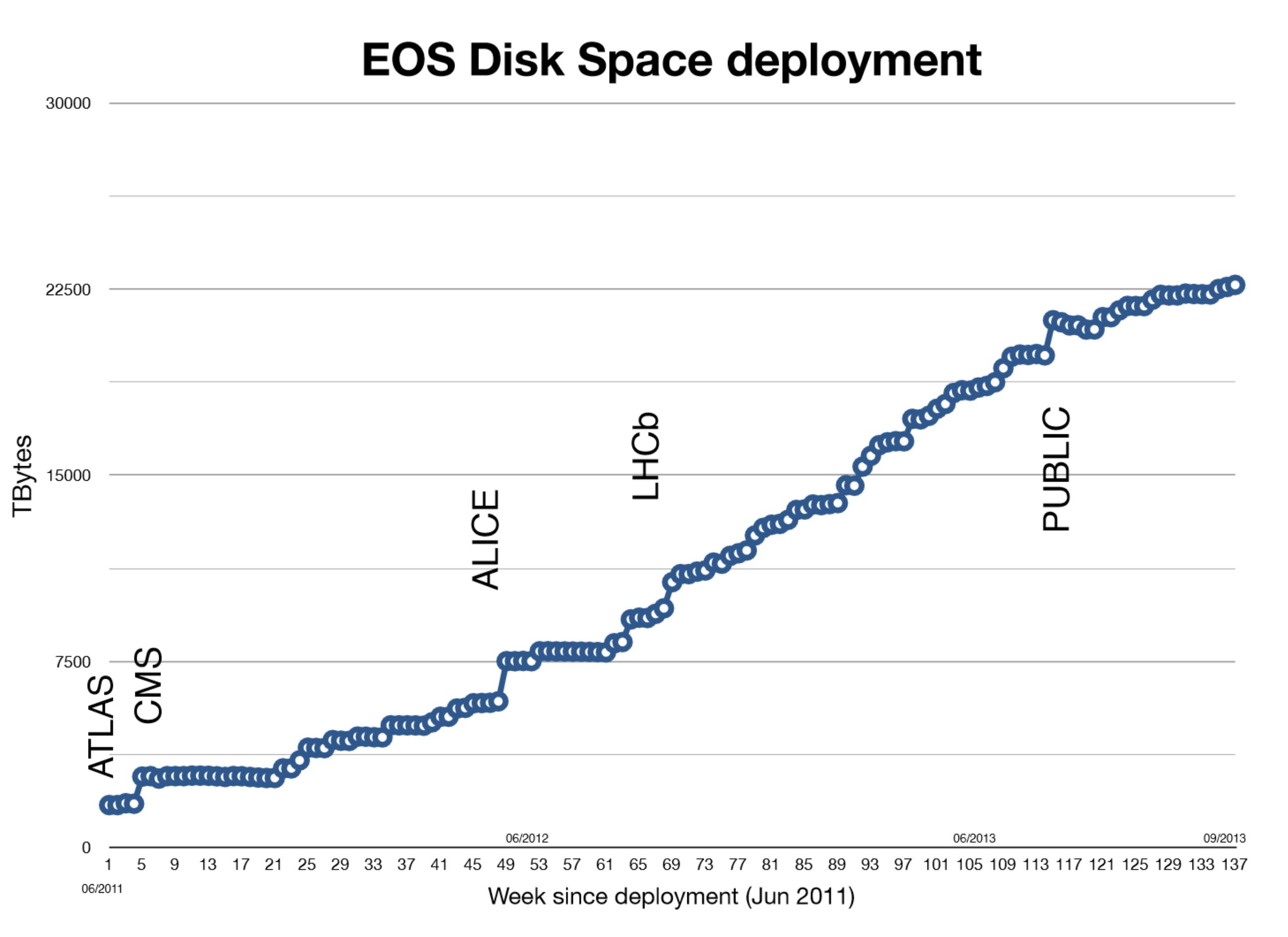




CHEP '13 AMSTERDAM

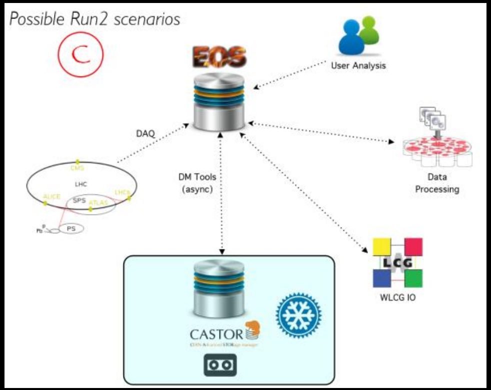
#376 Paul Millar: *dCache: Big Data storage for HEP communities and beyond*

Storing LHC data on EOS or CASTOR



For LHC Run2 several scenarios open:

- DAQ to CASTOR, then copy to EOS
- DAQ to both EOS and CASTOR
- DAQ to EOS, use CASTOR as dark storage



Note HEP-specific protocols continue concurrent with industry standadization and new products like pNFS

Tape is not dead: still fastest per-device throughput (240MB/s) and cheap

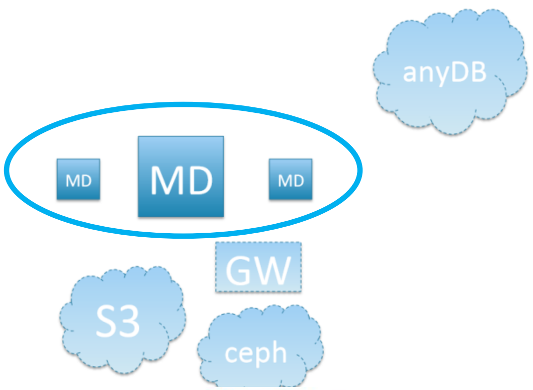
#83 Xavier Espinal: *Disk storage at CERN: handling LHC data and beyond*

Cheap meta-data management for DPM (and a scaling test …)

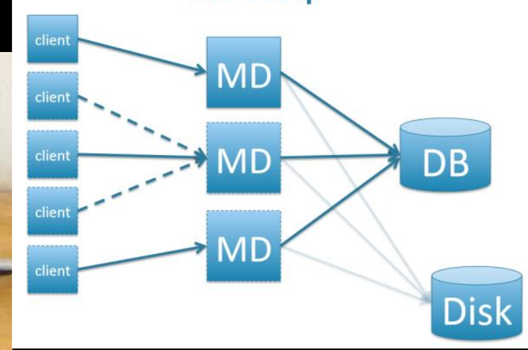
Performance per Euro

	Raspi	VM	EC2 Medium
Requests/sec	**25.8**	**394.18**	**162.27**
Buying Price	25	**428**[2]	-
Power Cost over 3 years[1,2]	4.89 Euros	41.94	-
Cost over 3 years[1]	29.89 Euros (+100)	469.94 (+100)	**1297.2**
500 Request/sec over 3 years	750	940	3891

So we focused on Metadata



The Setup



external equipment is much more expensive!

[1] Buying Price + Power Costs or EC2 Pricing (reserved instance)
[2] 12VMs running on DELL PowerEdge M620 24 core 16GB RAM, price from website

Virtualization

Software Defined Networking

Integrating CPU, memory, and communications

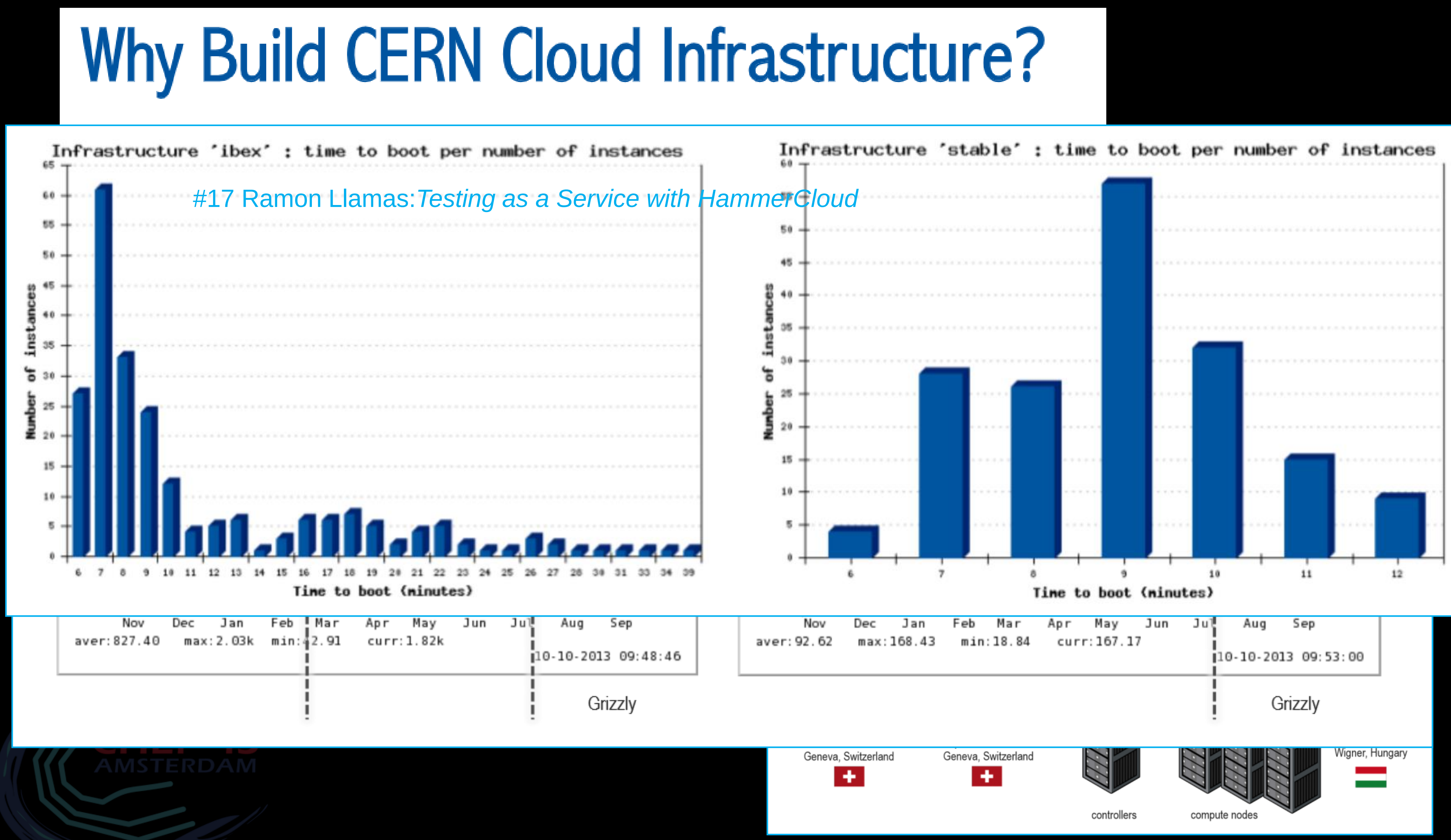
INFRASTRUCTURE AND NETWORKS

Virtualization and ‘clouds’

- On-demand deployment of resources using virtualisation clearly the way to go
  - Still hear the ‘cloud’ buzz word, but what we see is ‘agile infrastructure’
  - CERN pushing ahead due to dual-site setup *lots of talks related to Wigner commissioning*

CHEP ´13 AMSTERDAM

CERN Cloud production service Grizzly



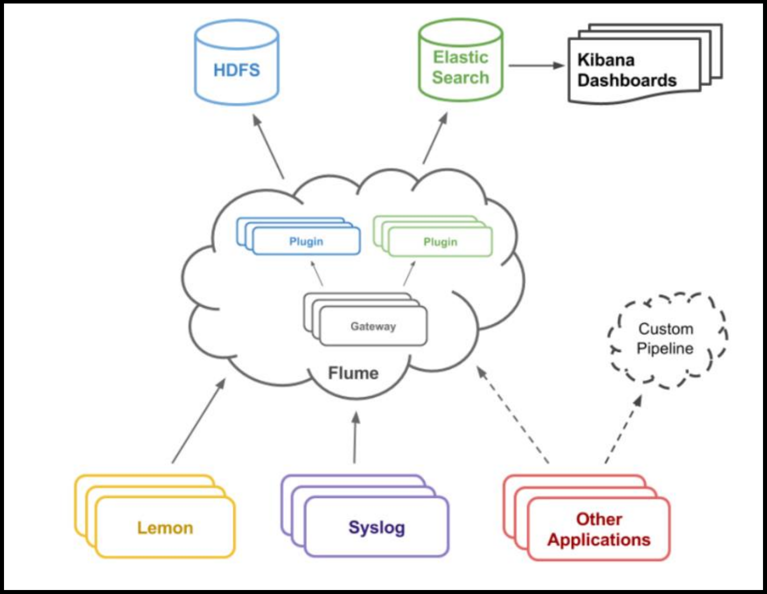
#217 Belmiro Moreira: *Production Large Scale Cloud Infrastructure Experiences at CERN*

Problems shift, but do not go away …

- Adding virtualisation gives you more potential failure points to monitor …
- And new services even more …

Motivation

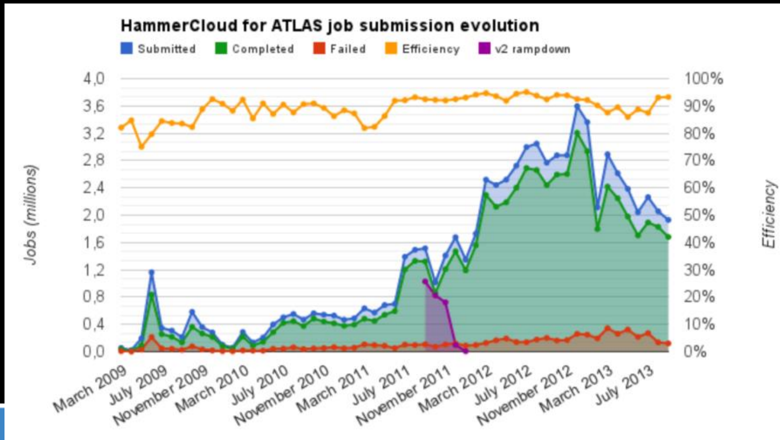
- Several independent monitoring activities in CERN IT
- Combination of data from different groups necessary
- Understanding performance became more important
- Move to a virtualised dynamic infrastructure

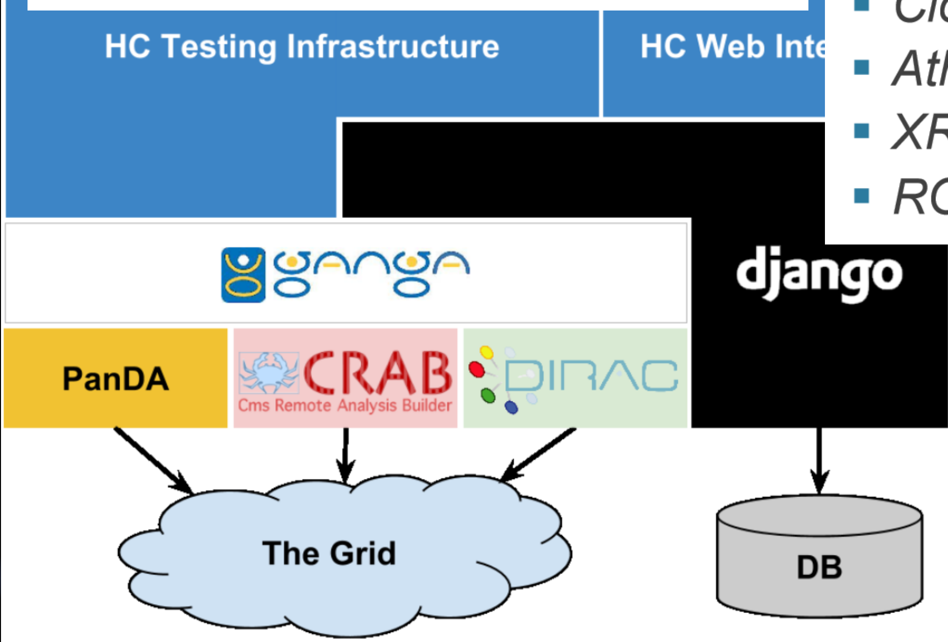


So just use Hadoop and ElasticSearch just to mine the monitoring data!

#206 Pedro Andrade: *Agile Infrastructure Monitoring*

From monitoring to testing
HammerCloud as a multi-purpose tool





New use cases

- Stress testing of sites
- Functional testing of sites
- AFT/PFT testing suite

} **12,000 test/year**

- *Benchmarking testing* ***NEW!***
- *Cloud resource validation* ***NEW!***
- *Athena nightly build system* ***NEW!***
- *XRootD federation (FAX)* ***NEW!***
- *ROOT I/O and WAN tests* ***NEW!***

- Need to cope with increasing demand,
- Requested by users and tools on demand
- *Elastic* testing infrastructure

Moved to Grizzly OpenStack cloud

#17 Ramon Llamas: *Testing as a Service with HammerCloud*

Commercial clouds: they *work* fine …

(but:
at a cost)

Early Success in January 2012

100 nodes, 200 CPUs used at commercial provider for G4 production tasks

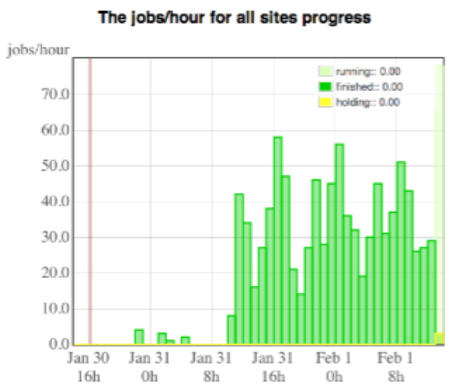
Panda production jobs at HELIX

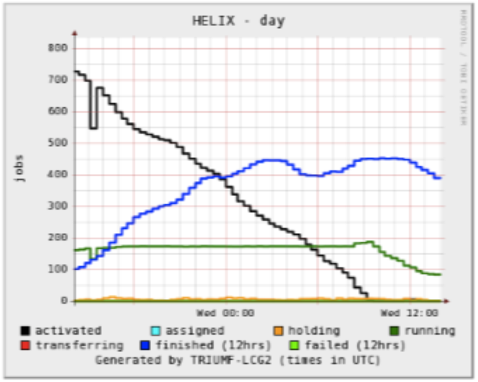
Selection parameters:
type:production **hours:**48

Click for help

Summary of production **jobs for the last** 2 **days, jobset**

1010 jobs. Click job number to see details.
States: running:63 holding:5 finished:942
Users (1): borut.kersevan@ijs.si:1010
Releases (1): Atlas-16.6.7:1010
Processing types (1): simul:1010
Job types (1): managed:1010
Task ID (2): 693280:#1000 694592:#10
Transformations (1): AtlasG4_trf.py:1010
Working groups (1): physics:1010
Creation Hosts (2): voatlas110.cern.ch:651 voatlas111.cern.ch:359
Sites (1): CERN.HELIX:1010
Regions (1): CERN:1010
Clouds (1): CERN:1010





…ture

We have built a cloud execution framework using Condor to schedule the job slots, CVMFS as the machine images, EOS for data storage, and Ganglia for monitoring:

Condor: orchestrates the jobs; a condor_schedd boots with the cloud VM and calls home to a condor_master at CERN to get its workload.
CVMFS: holds all the software and configuration.
EOS: input and output of physics data. Allows remote access from cloud sites across the WAN.
Ganglia: basic monitoring of machines.

HELIX NEBULA THESCIENCECLOUD

CHEP´13 AMSTERDAM

#84 Ramon Llamas:*Helix Nebula and CERN: A Symbiotic Approach to Exploiting Commercial Clouds*

Own data centre still very cost-effective – factor 3 compared to Amazon

You pay for the Amazon profit
and for the low occupancy ('elasticity')

Wong, BNL

	USATLAS	RHIC
Server	$228/yr	$277/yr
Network	$28/yr	$26/yr
Software	$3/yr	$3/yr
Staff	$34/yr	$34/yr
Electrical	$12/yr	$16/yr
Space	$27/yr	$13/yr
Total	$332/yr ($0.038/hr)	$369/yr ($0.042/hr)

Includes 2009-2013 data

BNL-imposed overhead included

Amortize server and network over 4 or 6 (USATLAS/RHIC) years and use only physical cores

RACF Compute Cluster staffed by 4 FTE ($200k/FTE)

About 25-31% contribution from other-than-server

- Cost of computing/core at dedicated data centers compare favorably with cloud costs
  - $0.04/hr (RACF) vs. $0.12/hr (EC2)
  - Near-term trends
    - Hardware ↘
    - Infrastructure ↗
    - Staff ⇨
    - Data duplication ↗
- Data duplication requirements will raise costs and complexity – not a free ride

#14 Tony Wong: *Operating dedicated data centers - Is it cost-effective?*

The network is not 'just there'

'Back to the future'

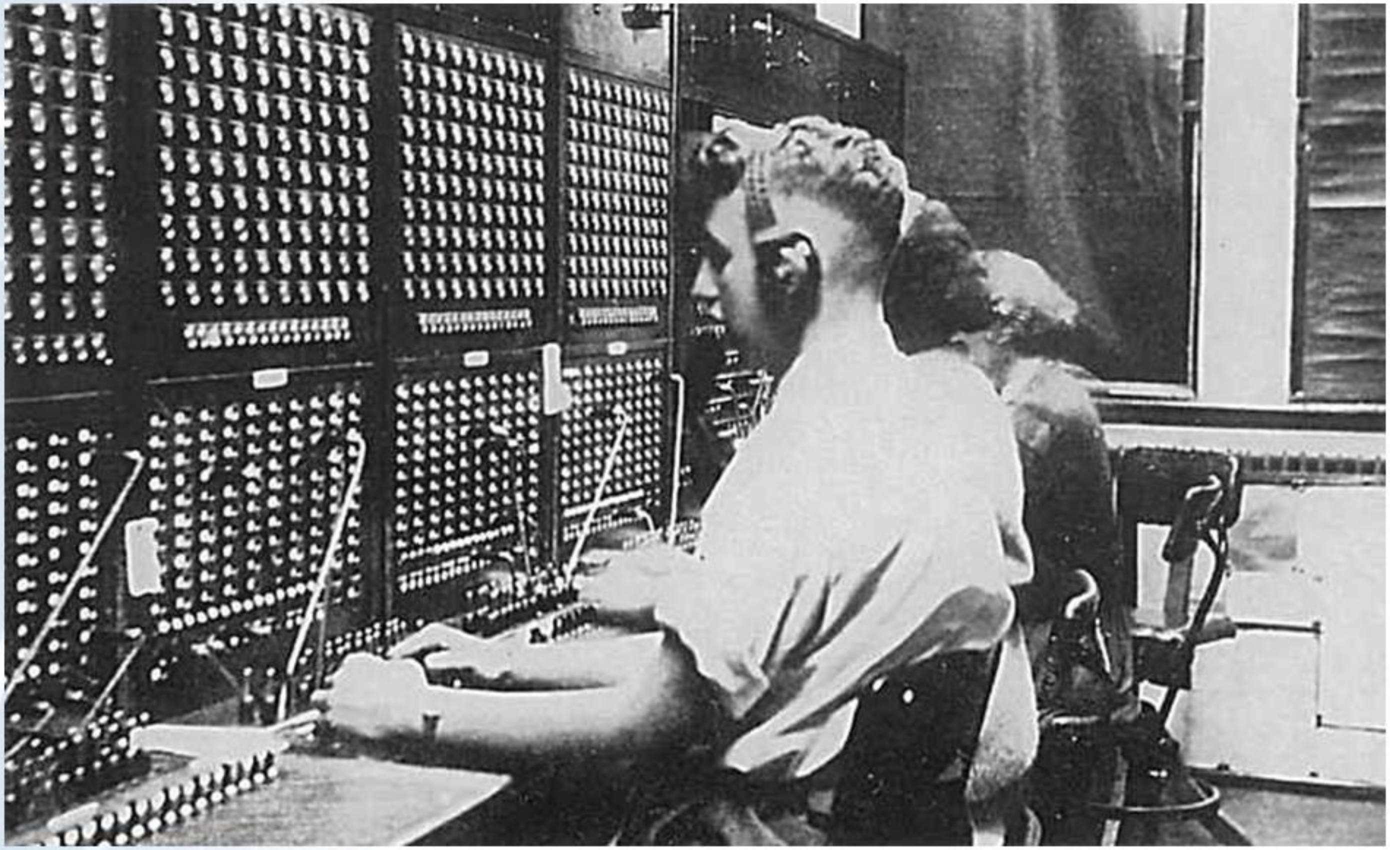
ESnet

- o "What we can do on LANs today is indicative of what we wish to be able to do on wide area networks."
- o "Just as we expect a computer to perform as if we are the only user, we expect the network to give that same appearance."

First workshop report for ESnet on intersite networking, 1986

Network community is still struggling to meet application requirements captured in 1986!

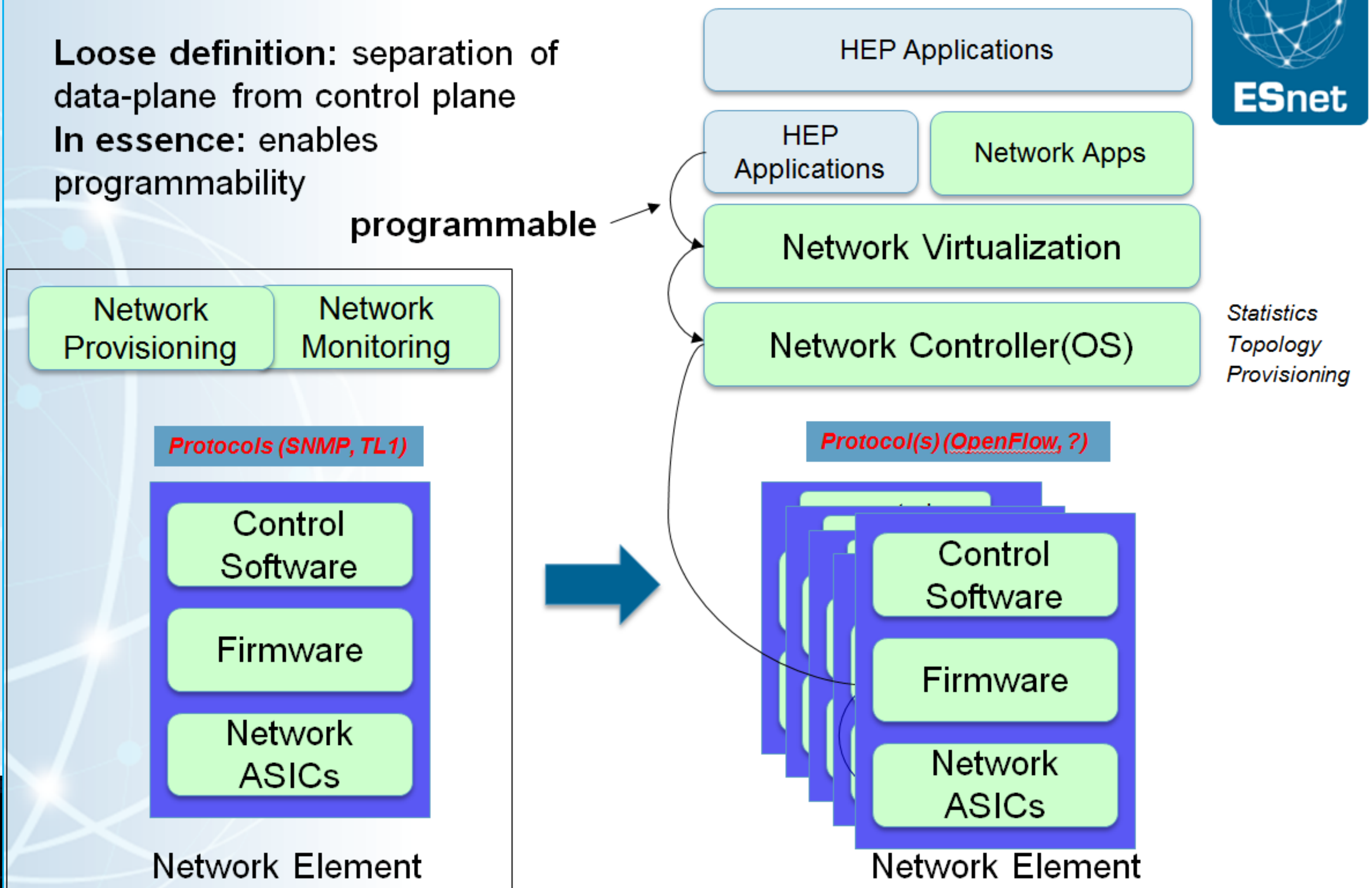
What is common between modern networks and analog phone switches?



What is SDN?

Loose definition: separation of data-plane from control plane

In essence: enables programmability



Inder Monga – *Software Defined Networking* – Thursday plenary

SDN is about network abstraction

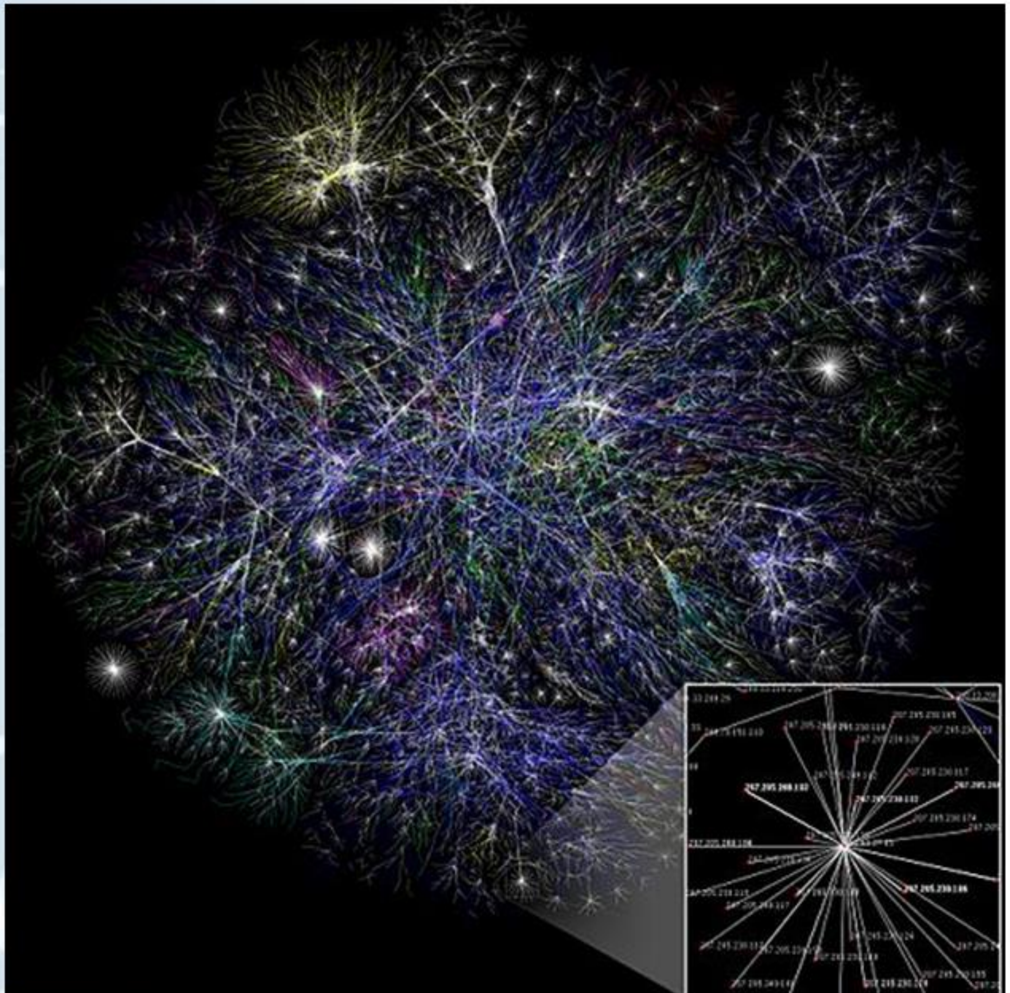
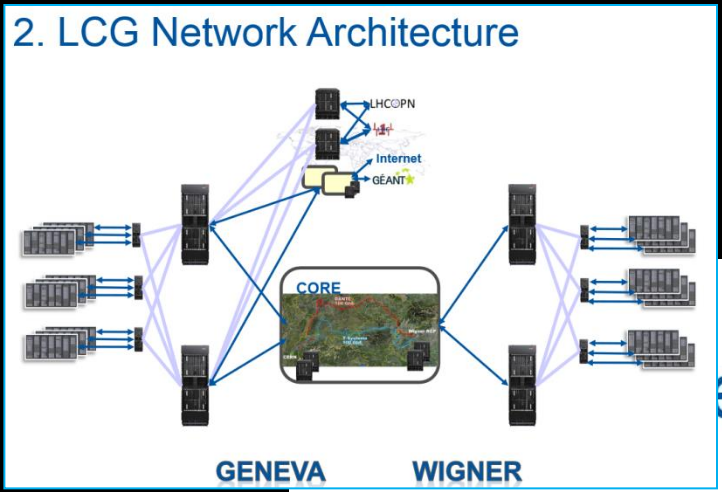


Image by Matt Britt (used with permission under Creative Commons Attribution 2.5)

Complexity of the Internet infrastructure ‘black box’:

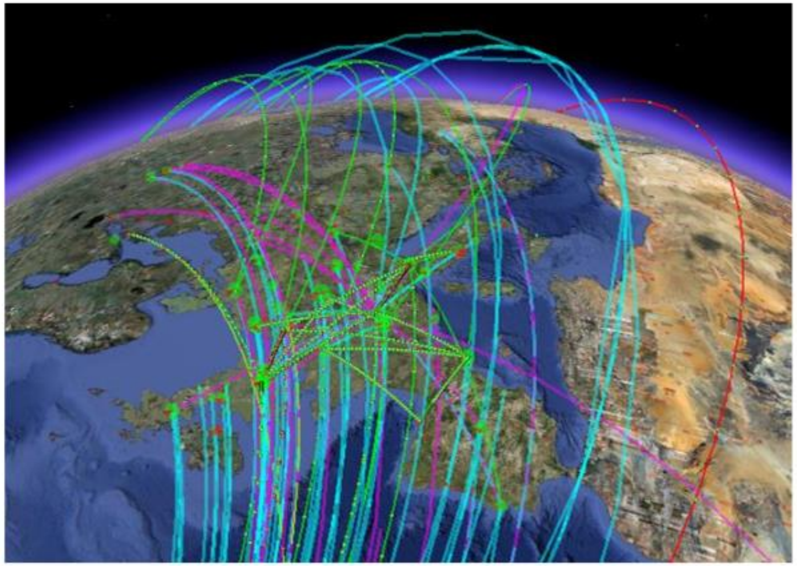
It’s ***strength*** and ***weakness***

CERN internet peers



etwork

- Public general purpose connections
  - Full BGP Internet routing table
  - Geant, CIXP, ISPs
- Private WLCG
  - LHCOPN
    - 70Gbps peaks to T1
  - LHCONE



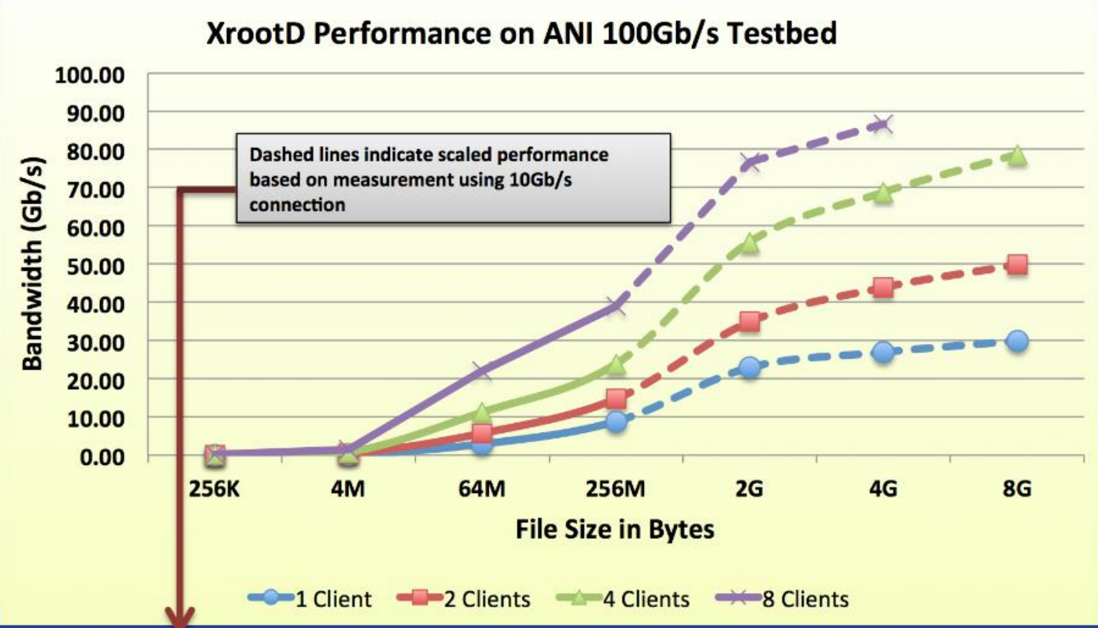
Brocade Routers	8
BGP Peerings	86
Aggregated BW	232 Gbps
IPv4/IPv6 Dual Stack	YES

#30 David Rueda: *Network architecture and IPv6 deployment at CERN*

100Gbps is coming fast – literally ☺ Xrootd cannot keep up and needs **4GByte+ files** and >8 parallel clients

XRootD Tests

- Data Movement over XRootD, testing LHC experiment (CMS / Atlas) analysis use cases.
  - Clients at NERSC / Servers at ANL
  - Using RAMDisk as storage area on the server side
- Challenges
  - Tests limited by the size of RAMDisk
  - Little control over xrootd client / server tuning parameters



Dataset (GB)	1 NIC measurements (Gb/s)	Aggregate Measurements (12 NIC) (Gb/s)	Scale Factor per NIC	Aggregate estimate (12 NIC) (Gb/s)
0.512	4.5	46.9	0.87	–
1	6.2	62.4	0.83	–
4	8.7 (8 clients)	–	0.83	86.7
8	7.9 (4 clients)	–	0.83	78.7

Calculation of the scaling factor between 1 NIC and an aggregated 12 NIC for datasets too large to fit on the RAM disk

U.S. DEPARTMENT OF ENERGY

And 400GbE is also here already

ICFA SCIC

400G Production-Ready Waves Demonstrated
400GE Link in Production (RENATER)

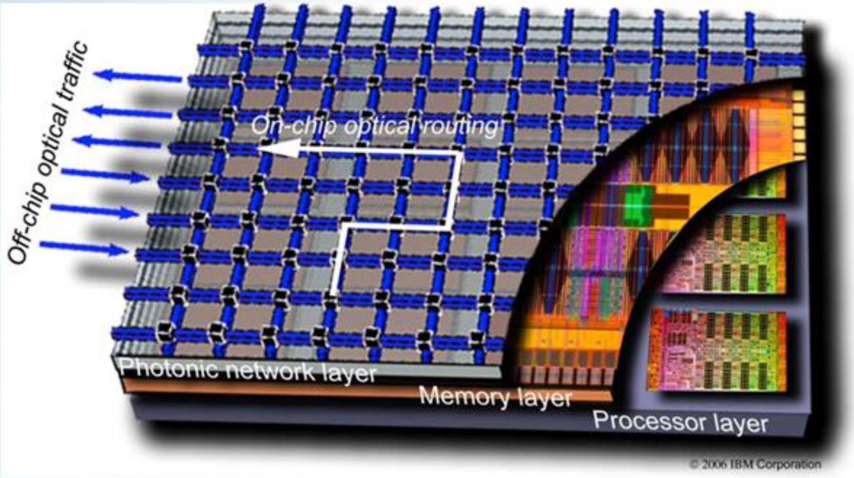
Chinese telecoms equipment vendor Huawei successfully completed a field trial using new optical fiber transmission technologies on Vodafone's live network, reaching 2 Terabit/s transmission over 3,325 km, or 2066 miles. This capacity is ~20 times higher than current commercially deployed 100G systems.
http://www.huawei.com/en/about-huawei/newsroom/press-release/hw-202114-vodafone.htm

February 6: Orange, Alcatel-Lucent provide a live 400G link to RENATER (Paris – Lyon)

France Telecom-Orange and Alcatel-Lucent have deployed the world's first optical link with a capacity of 400 Gbps per wavelength in a live network. Following a successful field trial, the 400-Gbps-per-wavelength fiber-optic link is now operational between Paris and Lyon (289 miles).
[System capacity: 17.6 Tbps on 44 400G waves.]

http://www.lightwaveonline.com/articles/2013/02/orange--alcatel-lucent-provide-live-400g-link-to-renater.html

A fun peek into the future…just imagine



With silicon photonics integration, each chip will have a network interface

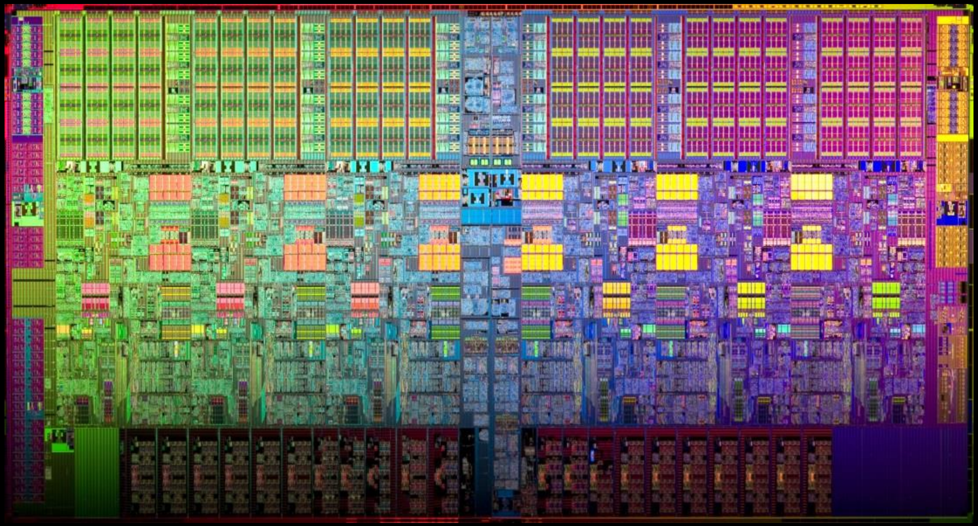
That implies each chip could be network addressable

If so, we could design servers without needing NIC cards – no difference between communication within the motherboard or outside.

With HEP applications like FAX, file systems or memory can be mounted remotely to my chip while 'streaming data for analysis.'

With SDN, can effectively route IP and non-IP protocols (like ROCE)

SDN could revolutionize how computing is done, are we ready for that?



Thermal death (beyond classic x86 cores)

Vectorization: how we learn anew what we though we lost

The OO curse, or the 'how-to-waste-CPU-cycles' how-to guide

C++11: a language to make concurrency understandable

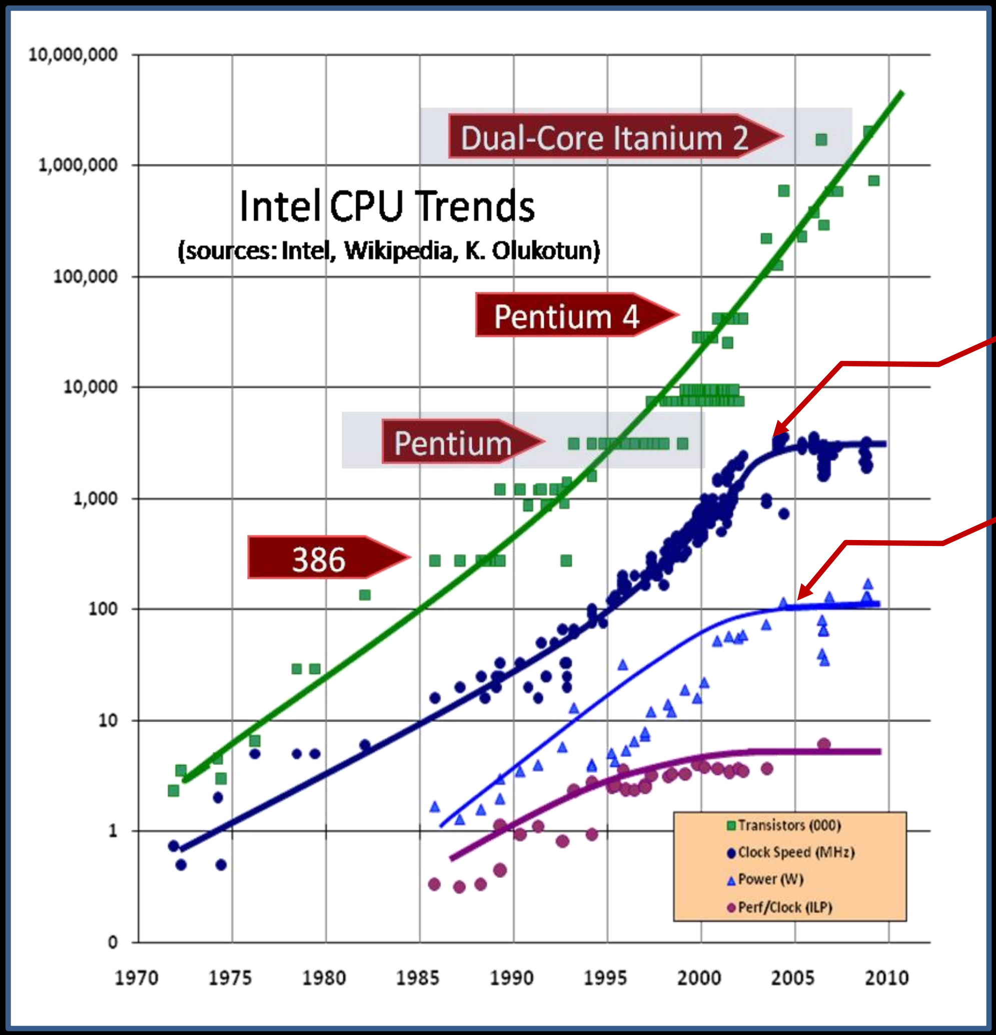
Prepare for the future!

PARALLELISM
MULTI-CORE AND VECTORISATION

CHEP ´13 AMSTERDAM

Image: die of the Intel XEON 5680 CPU

Lulled to sleep?



2004: Jayhawk cancelled by Intel

2005: end of Dennard scaling

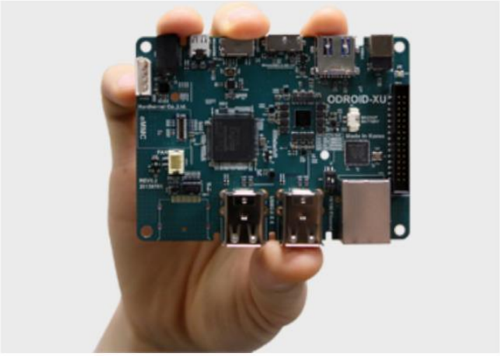
Figure source: <http://www.gotw.ca/publications/concurrency-ddj.htm>,
data from Intel, public sources and Wikipedia, and Kunle Olukotun (Stanford University)

WLCG as Distributed Supercomputer - Power

- Not only would the the WLCG be one of the top supercomputers in terms of performance if it were considered as such, but it also shares another characteristic which is less obvious.

- Using the mix of hardware available at FNAL (and known power use), we estimate the aggregate power cost to be of order 10MW

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National University of Defense Technology China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3120000	33862.7	54902.4	17808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560640	17590.0	27112.5	8209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1572864	17173.2	20132.7	7890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705024	10510.0	11280.4	12660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786432	8586.6	10066.3	3945
6	Texas Advanced Computing Center/Univ. of Texas United States	**Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462462	5168.1	8520.1	4510
7	Forschungszentrum Juelich (FZJ) Germany	**JUQUEEN** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458752	5008.9	5872.0	2301
8	DOE/NNSA/LLNL United States	**Vulcan** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	393216	4293.3	5033.2	1972
9	Leibniz Rechenzentrum Germany	**SuperMUC** - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR IBM	147456	2897.0	3185.1	3423
10	National Supercomputing Center in Tianjin China	**Tianhe-1A** - NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050 NUDT	186368	2566.0	4701.0	4040



- Fedora 19 ARMv7-A, hard floats, gcc 4.8, ODROID kernel

	Cores	TDP	Gen-Sim Evt/min/core	Gen-Sim Evt/min/W	G4MT Evt/min (threads)	G4MT Evt/min/W
	4	4W	1.14	1.14	34.2 (4)	8.6
ODROID XU+E	4/4	5.5W?			45 (4) (est.)	8.2
dual Xeon L5520	2x4	120W	3.50	0.23	307.2 (16)	2.6
dual Xeon E5-2630L	2x6	190W	3.33	0.21		

516000 events/kWh

156000 events/kWh

#303 Peter Elmer: *Explorations of the viability of ARM and Intel Xeon Phi for Physics Processing*

Reminder

The changes are fundamental

- The many-core computing change
  - Large core counts is primary focus
  - Slower cores with reduced memories, reduced operations, and much greater vector processing.
  - The larger leadership class machines have been headed in this direction, but also include multiple high speed interconnects.
- Even the simpler commodity-computing environment is changing.
  - ARM based server ar
  - PDAs replacing lapto
  - Integrated CPU-GPU

- Other key aspects
  - NVIDIA K20: unusual synchronized thread operation
  - Xeon Phi: Big 512 bit VPU registers

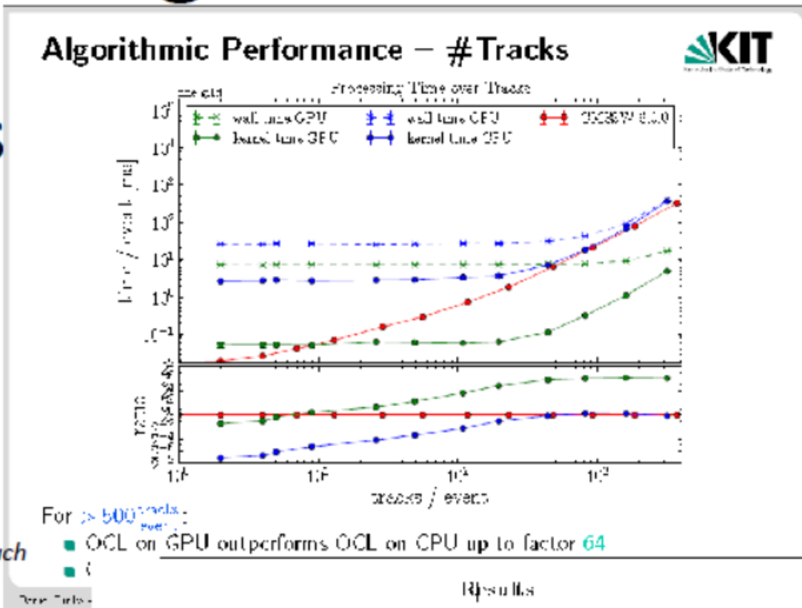
	Phi	K20
Power	~300W	~235W
Cores	60	~2500
Threads	60*4=240	2048*13=26624
Memory / thread	8GB/240=33MB	32 regs + 24KB shared

Most interesting is how many-core affects event processing applications

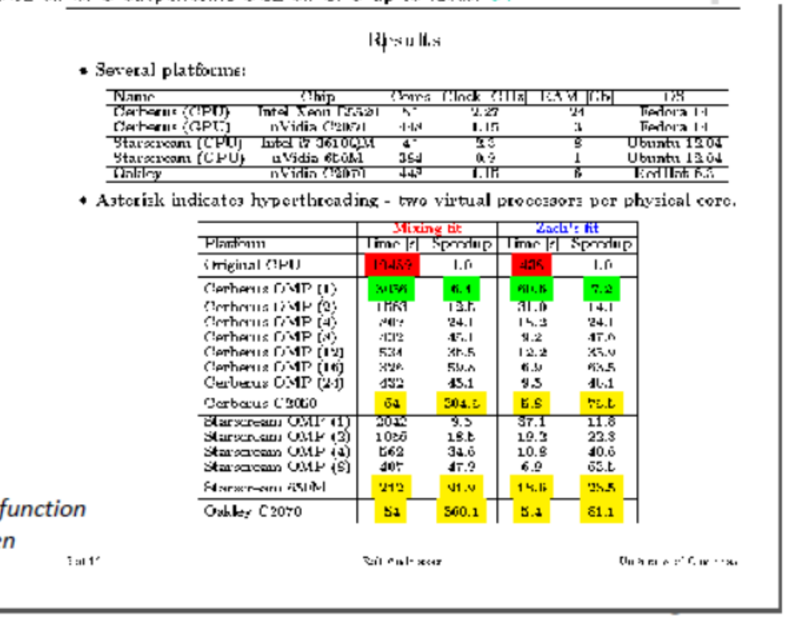
re-implement algorithms?

- Many examples for GPUs from every processing step!
  - Track Fitting
  - GooFit
  - GEANT
  - …

Parallel Track Reconstruction in CMS Using the Cellular Automaton Approach
Daniel Funke et al.



This approach gets better speed-up but is much more work.
(must revalidate physics
 performance)



Results

- Several platforms:

Name	Chip	Cores	Clock (GHz)	RAM (Gb)	OS
Cerberus (CPU)	Intel Xeon E5520	8*	2.27	24	Fedora 14
Cerberus (GPU)	nVidia C2050	448	1.15	3	Fedora 14
Starscream (CPU)	Intel i7 3610QM	4*	2.3	8	Ubuntu 12.04
Starscream (GPU)	nVidia 650M	384	0.9	1	Ubuntu 12.04
Oakley	nVidia C2070	448	1.15	6	Red Hat 6.3

- Asterisk indicates hyperthreading - two virtual processors per physical core.

	Mixing fit		Zach's fit	
Platform	Time [s]	Speedup	Time [s]	Speedup
Original CPU	19489	1.0	436	1.0
Cerberus OMP (1)	3058	6.4	60.6	7.2
Cerberus OMP (2)	1561	13.5	31.0	14.1
Cerberus OMP (4)	807	24.1	15.3	24.1
Cerberus OMP (8)	432	45.1	9.2	47.0
Cerberus OMP (12)	534	36.5	12.2	35.0
Cerberus OMP (16)	328	59.6	6.9	63.5
Cerberus OMP (24)	432	45.1	9.5	46.1
Cerberus C2050	64	304.5	5.8	75.5
Starscream OMP (1)	2042	9.5	37.1	11.8
Starscream OMP (2)	1056	18.5	19.3	23.3
Starscream OMP (4)	562	34.6	10.8	40.6
Starscream OMP (8)	407	47.9	6.9	63.5
Starscream 650M	212	91.9	15.6	28.5
Oakley C2070	54	360.1	5.4	81.1

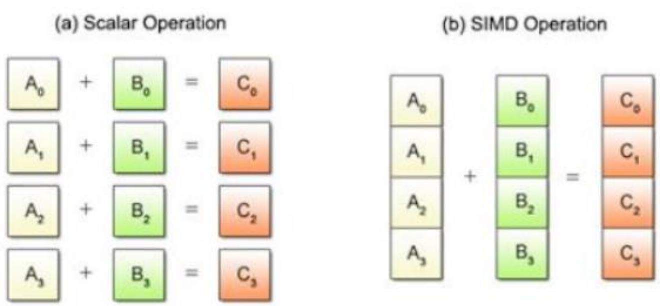
GooFit: Massively parallel function evaluation, Rolf Andreassen

Vectorization – making your code fast today and tomorrow

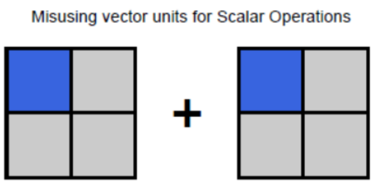
Vectorization

- Traditional a **+** b operation: combine input a with input b; yields one output
  Single Instruction, single data

- Current CPUs run **+** operation
  memory) at the same cost, e.g
  Single Instruction, Multiple Da

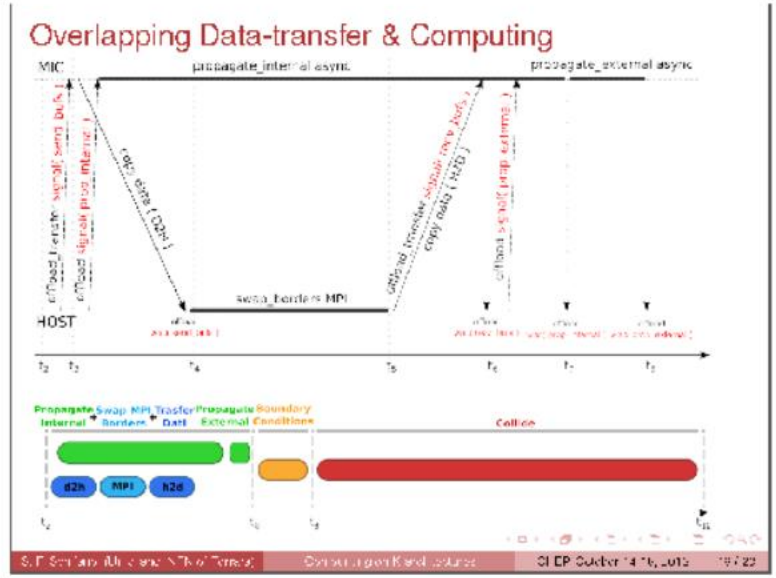


- We often use only one "slot" out of four
  - likely even more dramatic in the future
  - already more dramatic for GPUs
- Vectorized code commonly sees throughput increase by factor 2



Think about…

- Communication overhead with accelerators
  - Overlap it with computation
  - Batch more tasks
- Precision
- Memory Layout



Forward Scheduler 1/2

Pros:

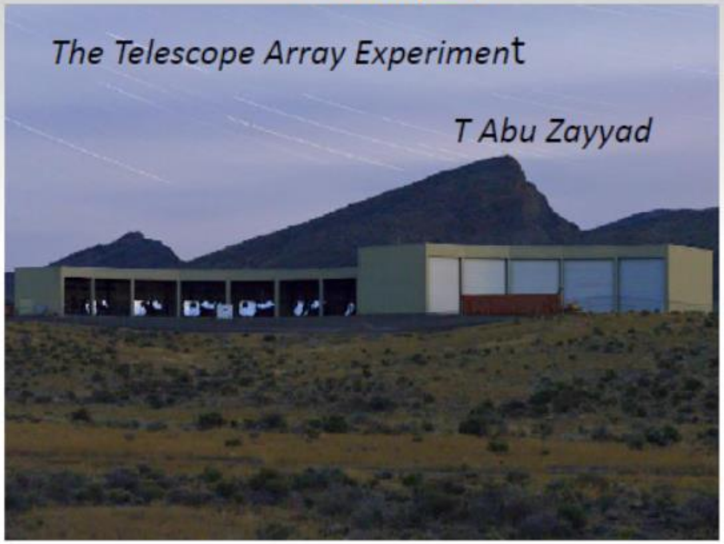
+ Scheduling intrinsically immune to deadlocks
+ Possible to lump data from different events for computations on accelerators
+ Possibility to run same algorithm repeatedly on the same core: cache friendly

Vectorization in Practice

- Vectorized code must
  - not use virtual functions
  - align data as vectors
- Very intrusive
  - changes interfaces
  - changes data formats
- But it gives you a very noticeable performance boost already today

Precision

Middle Drum TA/TALE FD Observatory Site (14 + 10 Telescopes)



Excellent speed-up on GPU

Tested single & double precision

"If single is OK – then don't use double"

Performance comparison – running time

- Desktop CPU vs. Desktop GPU

E (eV)	100 evt (cpu) t in sec.	100 evt (gpu) t in sec.	1000 evt (cpu) t in sec.	1000 evt (gpu) t in sec.
10[illegible]	13.91	0.759 (18x)	140.97	2.860 (49x)
10[illegible]	18.76	0.875 (21x)	164.59	3.348 (49x)
10[illegible]	24.64	1.006 (24x)	196.99	4.230 (47x)
10[illegible]	40.72	1.297 (31x)	364.44	6.834 (53x)

'if results are very sensitive to double precision (and surely if you see the difference between AMD and Intel CPUs!), you likely need to revise your implementation!'

From the past - ignored for a long time

Writing of pipelineable and vectorisable code used to be part of a standard physics bachelor curriculum! But it dropped out and efficiency suffered …

- **ILP (Instruction Level Parallelism)**
  - Our LHC programs typically issue (on average)
    - only 1 instruction per cycle
  - This is very low!
    - Core 2 architecture can handle 4 instructions
    - Each SSE instruction can operate on 128 bits (2 doubles)
  - We are typically only extracting 1/8 of maximum

We are not getting out of first gear!!

5 September 2007 | 13 | jbe

Sverre Jarp, *How good is the match between LHC software and current/future processors?* , **CHEP2007**
https://indico.cern.ch/event/3580/contribution/490

How to waste your CPU? We did it!

1. Unpredictable conditional jumps inside tight loops are Evil™
2. Memory layout:

C++ Memory Layout, Or The Curse Of Object Oriented Data

```
class XYZ {
  double x;
  double y;
  double z;
};
```

- Assume algorithm that calculates

```
for (int i = 0; i < fManyXYZ.size(); ++i) {
  sum += fManyXYZ[i].x * fManyXYZ[i].x
       + fManyXYZ[i].z * fManyXYZ[i].z;
}
```

Axel Naumann: *C++ Evolves!* – Monday plenary

Array of Pointers to Structs

```
class XYZ {
  double x;
  double y;
  double z;
};
```

- “TObjArray<XYZ>”, vector<XYZ*>

xyz xyz xyz xyz

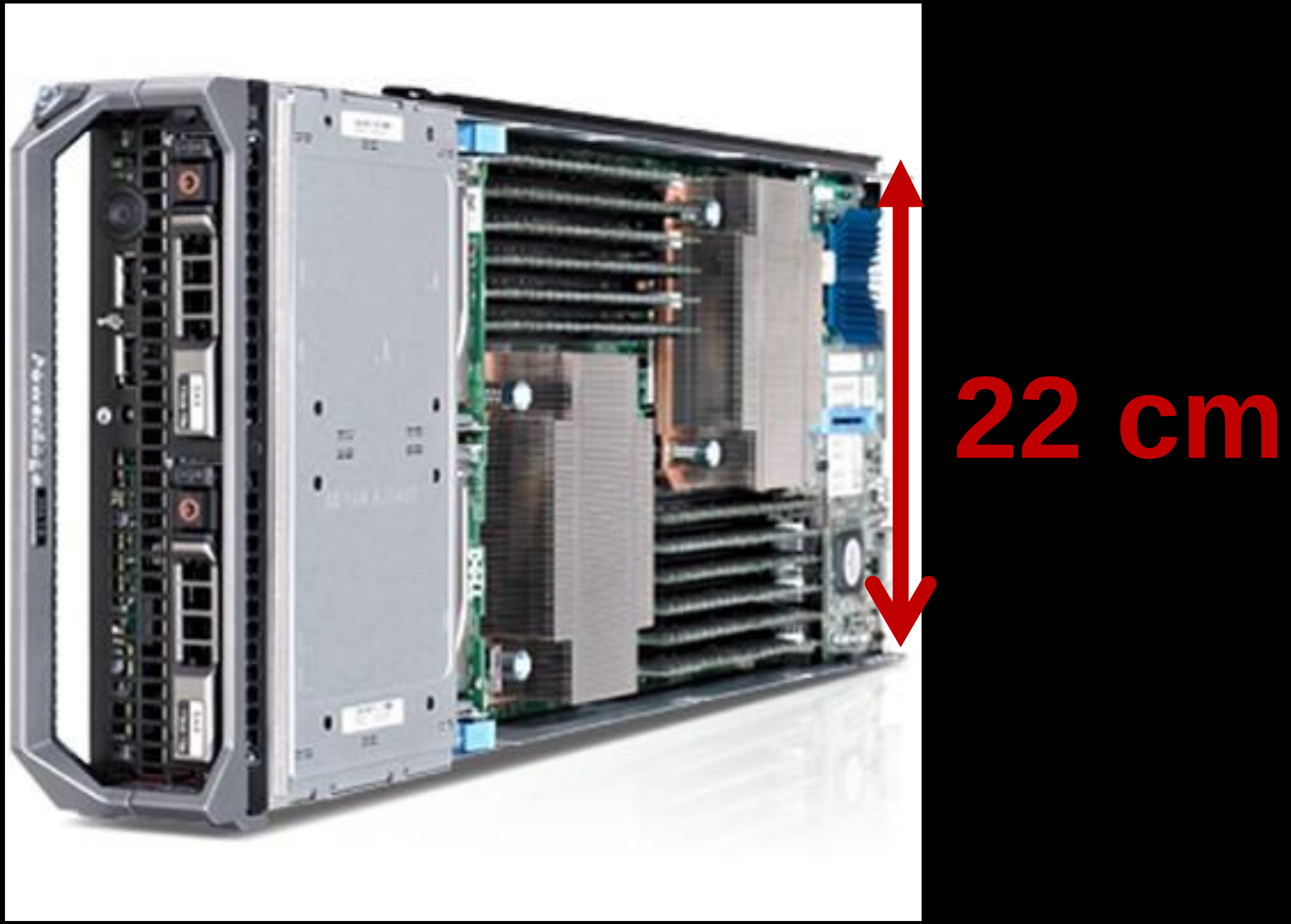
Using ‘new’ to instantiate each XYZ causes this mess!

xyz xyz xyz xyz xyz xyz xyz xyz

```
for (int i = 0; i < fManyXYZ.size(); ++i) {
  sum += fManyXYZ[i].x * fManyXYZ[i].x
       + fManyXYZ[i].z * fManyXYZ[i].z;
}
```


Axel Naumann: *C++ Evolves!* – Monday plenary

Quite a long way away …



Array of Structs

```
class XYZ {
  double x;
  double y;
  double z;
};
```

- vector<XYZ>, XYZ[N]

```
xyz xyz xyz xyz xyz xyz xyz xyz xyz yxz
```

+2

Slightly better …

```
for (int i = 0; i < fManyXYZ.size(); ++i) {
  sum += fManyXYZ[i].x * fManyXYZ[i].x
       + fManyXYZ[i].z * fManyXYZ[i].z;
}
```

Structs of Arrays (SOA)

```
class XYZ {
  double x;
  double y;
  double z;
```

• structOfArray<XYZ>

SOA Element Access

```
NaiveXYZSOA {
  vector<double> x;
  vector<double> y;
  vector<double> z;
} fManyXYZ;
```

- Accessing element of object number i

  what used to be `fManyXYZ[i].x`

  now becomes `fManyXYZ.x[i]`

- Workaround, wonderful R&D tool: Intel's Arrow Street [Costanza, Mon 17:25, Effectenbeurszaal]

- Proper solution to convert vector<Jets> into struct of arrays:

```
for (int i = 0; i < fManyXYZ.size(); ++i) {
  sum += fManyXYZ[i].x * fManyXYZ[i].x
       + fManyXYZ[i].z * fManyXYZ[i].z;
}
```


Axel Naumann: C++ *Evolves!* – Monday plenary

C++11 compilers ***can*** generate good code (given the right constructions)
… just like what you would have done anyway (don’t you?)

C++ ++

- Finally we got C++11
- Easier to use
- Can we trust the compiler to generate optimal code?

Conclusion

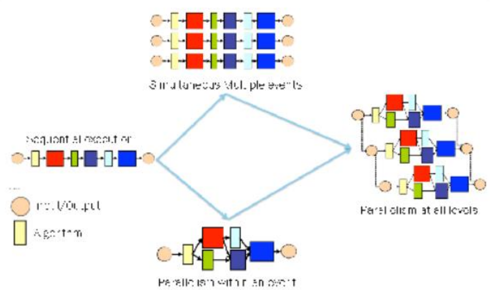
- One goal of the design of C was to provide a language with "no room below it", that is, to leave no reason to use a lower level language instead.
- This goal influenced the design of many of the "higher-level" features of the language, some of which we addressed in this talk
- Modern C++ compilers are sufficiently advanced to realize this goal in many cases.
- Modern C++ has many features to allow more concise and expressive code that is easier to maintain

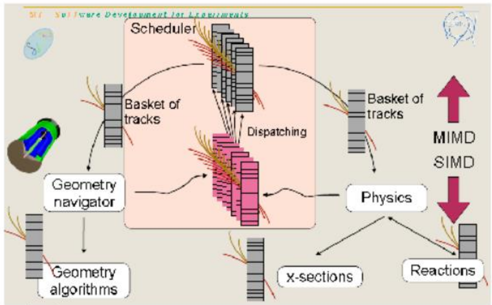
So: Write code for clarity and maintainability, using "high-level" features as they are intended, without worry about runtime efficiency.

Improving robustness and computational efficiency using modern C++, Marc Paterno et al.

Concurrency

- Adapt applications and application frameworks to many-core systems to exploit different sources of parallelism.
  - Multiple events
  - Within an event
  - Within an algorithm





A well-separated pairs decomposition algorithm for kd-trees implemented on multi-core architectures

Raul H. C. Lopes
Ivan D. Reid
Peter R. Hobson

But: compilers ***cannot*** fix inefficient memory layout!
So no arithmetic on vectors of objects please, but on objects with vectors in them!

And dont ‘new’ stuff inside a loop … heap scatter is Bad™

Garbage collection in C++11

in case you cant' remember where you left your data …

C++11: The Dark Side?

- Also caters coding wizards
  - variadic templates; lambdas; const_expr; user defined literals;...
- Complex code remains an option in C++11
- Still, C++11 dramatically improves even novices' code

std::shared_ptr is reference counted ("garbage collector")

- Can prevent hours of debugging memory errors!

Performance and optimizable code

- Hashed containers (finally!): std::unordered_map / std::unordered_set to be used instead of e.g. std::map<std::string,...>

- Container initialization

```
std::vector<int> v{12,42,17,12,9};
```

```
std::vector<int> v;
v.push_back(12);
v.push_back(42);
v.push_back(17);
v.push_back(12);
v.push_back(9);
```

- Move semantics

```
for (std::map<std::string, std::vector<MyClass> >::const_iterator
   i = m.begin(), e = m.end(); i != e; ++i) {
```

- auto

Simplified code

```
for (auto i = begin(m), e = end(m); i != e; ++i) {
```

- Bjarne Stroustrup: “C++11 feels like a new language”

- Old C++ code usually compiles in C++11 “mode”, ROOT had about 8 changes on 3 million lines of code:
  - token#pasting CPP macros
  - x={...} initializers
- Object file compiled with C++11 should not be linked against old C++: all C++11 or none

CHEP ´13 AMSTERDAM

Where do you find *good* examples?

- ROOT, Geant, frameworks should demonstrate the advantage of simple code, clear ownership, improved standard library

```
TH1::AddFunction(std::unique_ptr<TF1>)
```

- C++11 and after brings us closer to the ultimate goal:
  - Write correct code and analyses easily!
  - From data taking to physics result quickly!

CHEP ´13 AMSTERDAM

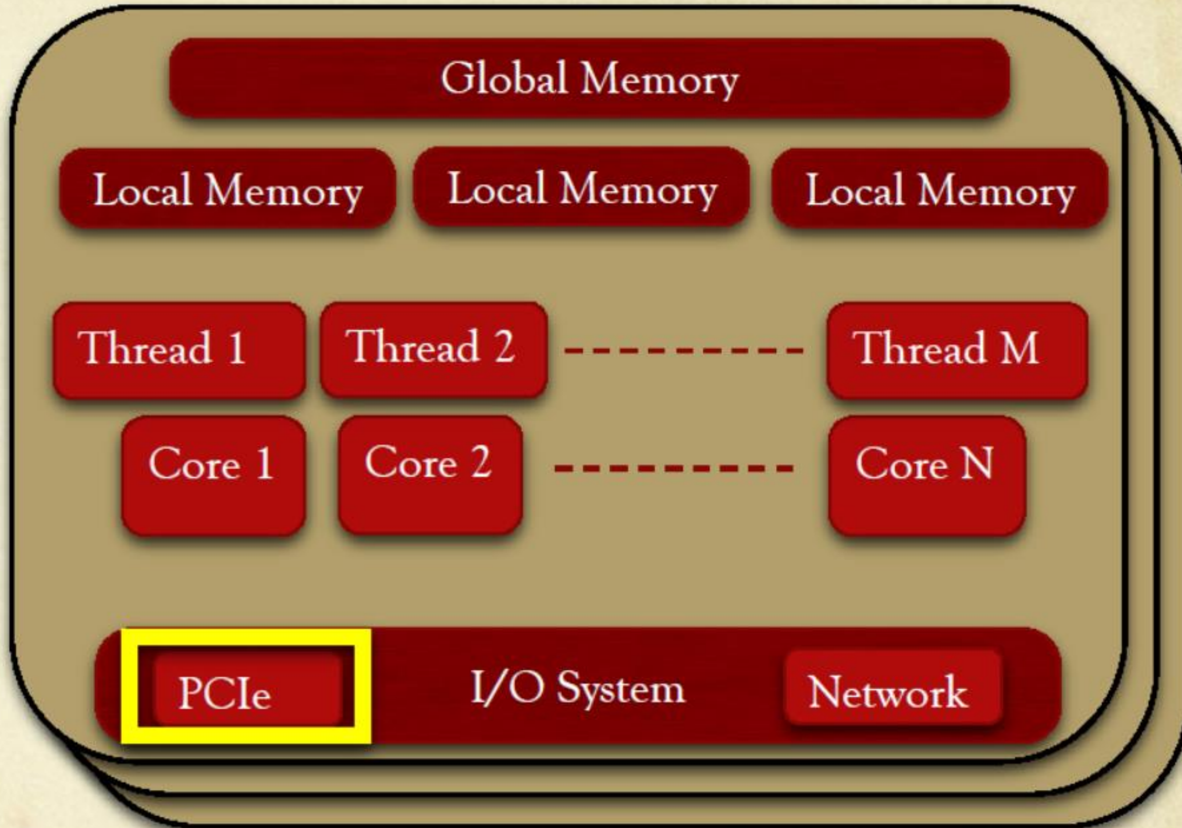
CPU information changes rapidly

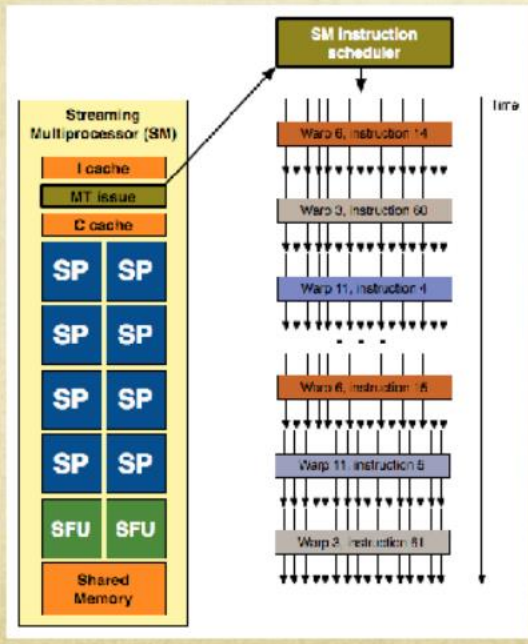
- Haswell servers (E5v3) readily available
- Good projections exist for Broadwell ('small' step) and Skylake (new microarch.)
  which are newer than CHEP2013 data

- Common mantra: **you need concurrency** (thread or data) to profit!
  - **Be friendly for SIMD** vectorization
  - Expect *much* more cores, some of which may be 'lighter' than others (2016+)

The many-core coprocessor

- Small memories
- Simple cores
- Data transfer costs
- Lock step operations
- Specialized languages



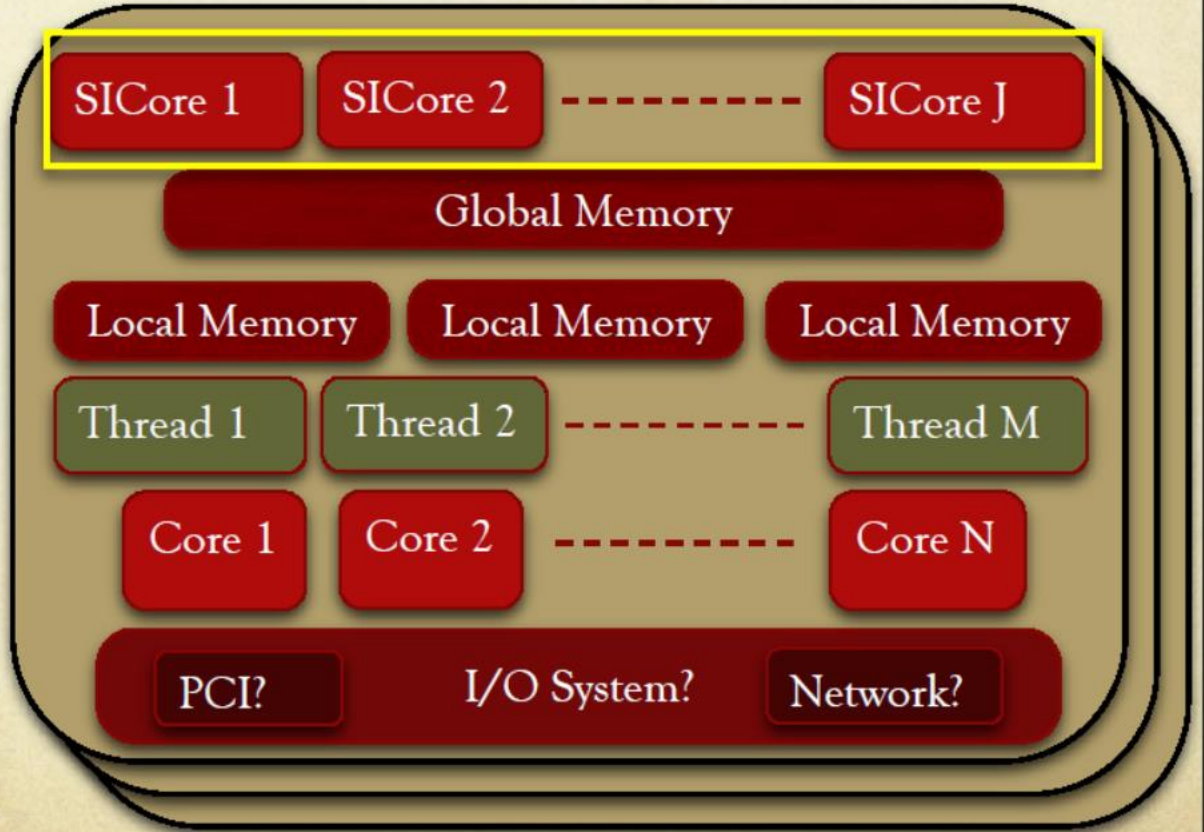


	Phi	K20
Cores	60	~2500
Threads	240	~26624
Memory / thread	~33MB	32 regs+24KB shared

Is this still a coprocessor?

- SICore = serial, integrated core
  - ARM?
  - PowerPC?
- Shared memory
  - Low overhead communications
  - Single address space possible
  - Usual coordinated access needed
- A Hybrid Processor

Not too far away ;-)



Issues that are already visible

- Mixed execution locations needed
  - For algorithm phases
  - For modules
  - **Heterogeneity is back** and will be a big issue
  - Unclear how much visibility and control the framework will have here
- Unclear how to keep a balanced load: many run-time parameters
- New methods of collecting summary data and diagnostic information from algorithm phases will likely be needed
  - Conditional logic troubles
  - Deviations in work load problems
  - Temporary storage of extra data issues
  - Making it widely available concerns
- What is good for the lighter weight processors has also been good for the heavier weight processors

```
tevere:occc-badexample:1311$ cat occc-dm.c
int i;main(){for(;i["]<i;++i){--i;}"];read('-'-'-',i+++"hell\
o, world!\n",'/'/'/'));}read(j,i,p){write(j/p+p,i---j,i/i);}
tevere:occc-badexample:1312$ 
```

SOFTWARE ENGINEERING

Treat your code as a scientific product!

Conclusion

26

- Rucio is the new data management system of ATLAS

> Maybe the real question is, ``What should we expect from our university faculty?''
> Is it of intrinsically higher worth to be able do QCD calculations in your head and invent new massless fields than to love CRTP and see your way to code a subtle new statistical algorithm? And which is a more valuable skill to teach our students?
>
> *FYI: The 'curiously recurring template pattern' (CRTP) is a C++ idiom in which a class X derives from a class template instantiation using X itself as template argument*
>
> **#493 Robert Lupton (Princeton, LSST):Writing Stellar Software – Monday Plenary**

  - Conduct enforced where possible by software
- Resulted in
  - High throughput of essentially error free code
  - Easy injection of new engineers into the team

#242 Mario Lassnig: *The ATLAS Data Management Software Engineering Process*

Documentation?

- **No** documentation talks this CHEP!

C. L. Dodgson (The Hunting of the Snark. 1876)

"The method employed I would clearly explain,
While I have it so clear in my head,
If I had but the time and you had but the brain ---
But much yet remains to be said."

The Butcher

My current theory is that we should give up on scientists writing introductory and how-to documents and instead employ professionals working in close collaboration with the development team.
One thing we're just starting to play with is a `stackoverflow` clone to provide the help desk and simultaneously the top-level documentation that we need.

Where will your data be 30 years from now?

DATA PRESERVATION

CHEP ´13
AMSTERDAM

Digital Curation

- Curation : The activity of, managing and promoting the use of data from its point of creation, to ensure it is fit for contemporary purpose, and available for discovery and re-use. For dynamic datasets this may mean continuous enrichment or updating to keep it fit for purpose.
- Archiving : A curation activity which ensures that data is properly selected, stored, can be accessed and that its logical and physical integrity is maintained over time, including security and authenticity.
- Preservation : An activity within archiving in which specific items of data are maintained over time so that they can still be accessed and understood through changes in technology.
- Digital curation : looking after and somehow "adding value" to digital data, ensuring its current and future usefulness. This probably implies creating some new data from the existing, in order to make the latter more useful and "fit for purpose".

Preservation methods

- **Preserving the original look-and-feel**
  - Emulation
    - Development of emulators to new platforms etc.
    - Active testing and technology watch
- **Preserving the content**
  - Migration
    - Format development watch (format libraries)
    - Development of transformation processes, testing, implementation, monitoring
    - Preparation for recoveries
- **Preserving the bits**
  - Integrity
    - File validation and monitoring
    - Management of copies
    - Both objects and metadata

DPHEP

2020 Vision for LT DP in HEP

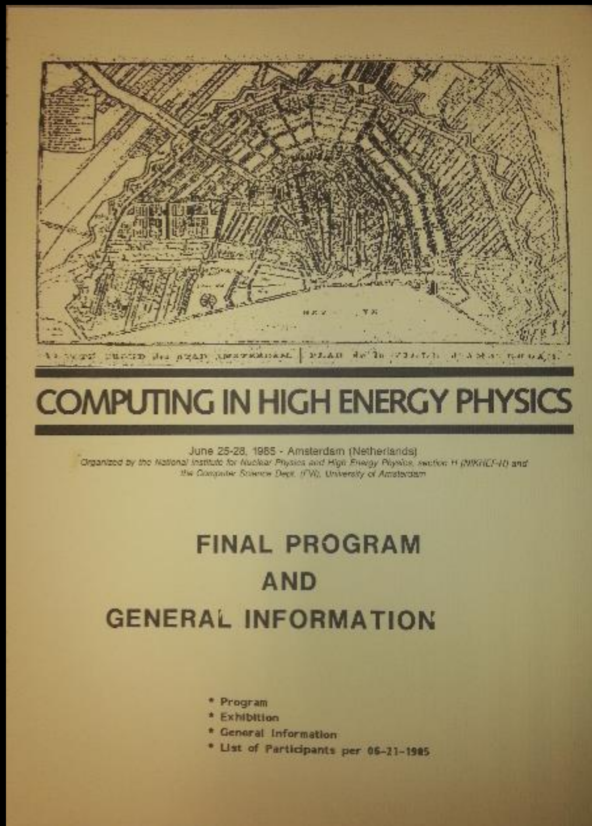
- ***Long-term – e.g. LC timescales*: *disruptive change***
  - By 2020, all archived data – e.g. that described in Blueprint, including LHC data – easily findable, fully usable by designated communities with clear (Open) access policies and possibilities to annotate further
  - Best practices, tools and services well run-in, fully documented and sustainable; built in common with other disciplines, based on standards
  - **DPHEP portal**, through which data / tools accessed

➢ **Vision achievable, but we are far from this today**

So what about *your* data?

- Keeping the bits is 'easy'
  - But deciding *which* bits to keep is something *you* should do, since only managed bits will stay!
  - I still have 50+ Exabyte 8mm tapes (and a tape drive on my shelf!), but realistically will never read them back … but the physics ntuples and PAW macro are on disk … although my logbook is not
- But what about
  - Meta-data: what do the bits mean
  - Processes: how convert bits to physics

  … when you are long since gone?
- **Do others understand your logbook??**


COMPUTING IN HIGH ENERGY PHYSICS

June 25-28, 1985 - Amsterdam (Netherlands)
Organized by the National Institute for Nuclear Physics and High Energy Physics, section H (NIKHEF-H) and the Computer Science Dept. (FVI), University of Amsterdam

FINAL PROGRAM
AND
GENERAL INFORMATION

* Program
* Exhibition
* General Information
* List of Participants per 06-21-1985

SOME FINAL WORDS

CHEP ´13 AMSTERDAM

Thanks!

- To all track conveners for their summaries
  Niko Neufeld, Tassos Belias, Andrew Norman, Vivian O'Dell, Rolf Seuster, Florian Uhlig, Lorenzo Moneta, Pete Elmer, Nurcan Ozturk, Stefan Roiser, Robert Illingworth, Davide Salomoni, Jeff Templon, David Lange, Wahid Bhimji, Dario Barberis, Patrick Fuhrmann, Igor Mandrichenko, Mark van de Sanden, Solveig Albrand, Francesco Giacomini, Liz Sexton, Benedikt Hegner, Simon Patton, Jim Kowalkowski, Maria Girone, Ian Collier, Burt Holzman, Brian Bockelman, Alessandro de Salvo, Helge Meinhard, Ray Pasetes, Steven Goldfarb
- The plenary speakers
- And all participants for the talks and posters!

CHEP ´13
AMSTERDAM

More?

http://chep2013.org/indico

http://chep2013.org/boa

http://chep2013.org/contrib/<*n*>

Soon:

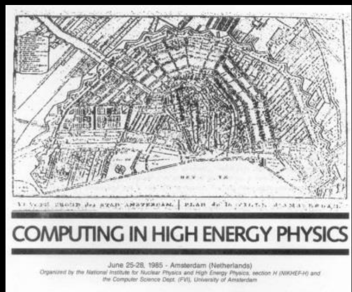
http://chep2013.org/journal

CHEP2015

April 13 – 17 2015

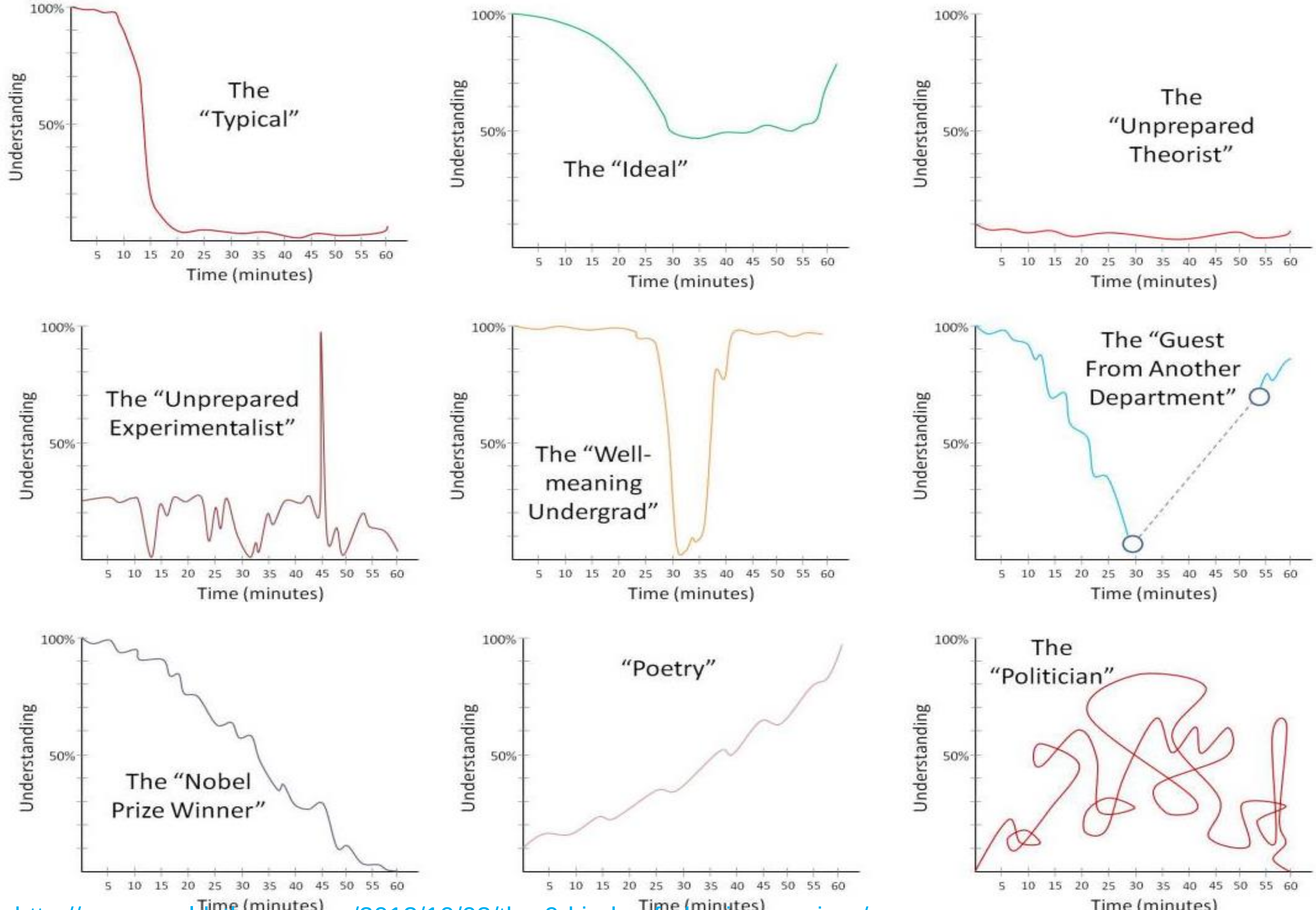
UTokyo and KEK, with RIKEN, OIST, and UOsaka

Amsterdam 1985



Amsterdam 2013

Okinawa 2015!



http://manyworldstheory.com/2013/10/03/the-9-kinds-of-physics-seminar/

From the plenaray talk of Stefano Spataro on *Computing for future experiments*