Track 1: Computing Technology for Physics Research

Jérôme Lauret
David Britton
Axel Naumann

Topics and statistics

- Topics from Track 1 – very diverse and still very vibrant
  - Programming languages, software quality, IDE and UI
  - Distributed and parallel computing
  - New architectures, many-cores
  - Virtualization
  - Online Monitoring and Control, HLT
- Disclaimer
  - Tried to put a word for everyone … if only a word, don't over-read it
  - May have a few possibly outrageous statements …
  - Did not include a posters summary, but acknowledge the work there …
- Numbers & Statistics
  - 23 parallels, 11 posters, 5 plenaries, 1 panel
  - Acceptance as parallel: 68%
- Other statistics
  - By centers: 36% from CERN, 64% coming from other places …
  - Academia: 32% comes from universities, 68% from National Laboratories or facilities
- By experiments (rounded)
  - Other: 36%
  - ATLAS: 25%
  - ALICE: 11%
  - OpenLab: 7%
  - RHIC & DESY: 7% each
  - CMS & LHCb: 4% each
- In other words: 57% coming from non-LHC experiments; 64% non-CERN related is encouraging ; 32% from academia / university (is this low?) ; OpenLab appears as a clear entity

We can only learn and expand through diversity – outreach to larger community essential for ACAT

University contribution IS vital for a sane research – the core workforce often comes from there

Very good session, many thanks to the speakers and contributors for their work – special thanks to the public for making the panel lively …

Architecture, GPU, Multi-Core & Languages / compilers

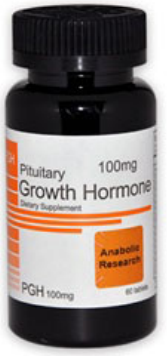
Where do we go from here? - The next phase of computing in HEP – Sverre Jarp (CERN / OpenLab)
Track finding using GPUs – Christian Schmitt (Meinz / ATLAS)
Challenges in using GPUs for the reconstruction of digital hologram images – Peter Hobson (Brunel)
Evaluation of likelihood functions on CPU and GPU devices – Snee Lindal Yngve (CERN / OpenLab)
Efficient Pseudo-Random Number Generation for Monte-Carlo Simulations Using Graphic Processors – Federico Carminati (CERN / ALICE)
Multicore in Production: Advantages and Limits of the Multi-process Approach - TSULAIA, Vakhtang (LBNL / ATLAS)
Can 'Go' address the multicore issues of today and the many-core problems of tomorrow? – Sebastien Binet (LAL / ATLAS)
Lessons from Static Analysis on HEP Software – Axel Naumann (CERN)

What did I learn and concluded?

- From Svere → setting the mood
  - CPU are as archaic as they used to be (same problems with latencies, locality, …)
  - Generally clock speed will not increase, other dimensions will (threads, cores, vectors, pipeline, superscalar and compute nodes and sockets) – and precision?
    - Hardware vectors keeps growing – community not using much of it
    - Assumed to be 10% at most of a machine capability
    - Many trying but beating one dimension only
  - Tera-Flops machine and Exa-scale coming and many architecture and hardware believed to become tomorrow commodities (at least some of it): Xeon, Atom, Tilera, … CPU, GPU, …

Not only a broad set of programing talents would be needed but frankly, **the plethora of architectures and devices tend to indicate a need for** not only agile software but **agile API (aAPI) and a new language or approach savior.**

In the tranches …



- Lots of (heroic) efforts to speed up framework – heard factors of x4, x10, x200
  - This shows **inconsistent comparative measurements and metrics across efforts/experiments**

- Work done & related
  - Christian Schmitt – CUDA use in ATLAS for seed finding, need to convert STL containers into array of C like structs + C-style arrays (perhaps OpenCL later)
  - Vakthang Tsulaia for ATLAS showed an Athena MP approach – event processed in separate sub-process (essentially fork()) , separate IO + merger – conclusion tend to go toward
    - Definitely saves lots of memory
    - "stop gap to save memory inflation and scaling as a function of CPU, but not as efficient … for now"
  - Yngve Sneen Lindal – OpenLab and exploring RooFit + OPenMP (SIMD) and Hybrid imps, OpenCL/OpenMP – both can coexist well wile CUDA may be more suitable for large memory programs
  - Federico Caminati - ALICE using CUDA and OpenCL for RAND – memory transfer in/out of GPU problem

  - Related - Sebastien Binet – 'can "go" save us' tend to conclude it is NOT yet ready
    - Addresses non-multi-core issues from yesterday and some multi-core issues of tomorrow – gcc-go coming
    - Binding to C++ is hardly possible (SWIG trick) no dynamic libraries and no dynamic library loading, no operator overloads, …
    - *My own spin*: we will learn on the way – it may evolve but it may not be "the " language

Do we have a clear path?

- I would say – yes, as much as Mr Magoo
  - I am sure we will be saved at the last mnt by a suddenly appearing elevator (i.e. solutions) … But we are not there yet
- Some interesting points
  - Hybrid model? Peter Hobston from Brunel (Digital Holograms) – very nice analysis of the workflow (transfer in/out of memory, …) and statements / conclusions straight to the point
    - Many parts to change in a code to get it right – may not be feasible
    - Hybrid workflows: part on GPU part on Grid/Cloud?

Valuable investigative work but – Where is this going? How to make this all work together?

Grid/Cloud and multi-core – are we ready [doubt so]?

Lack of consistent approaches – three prone problem

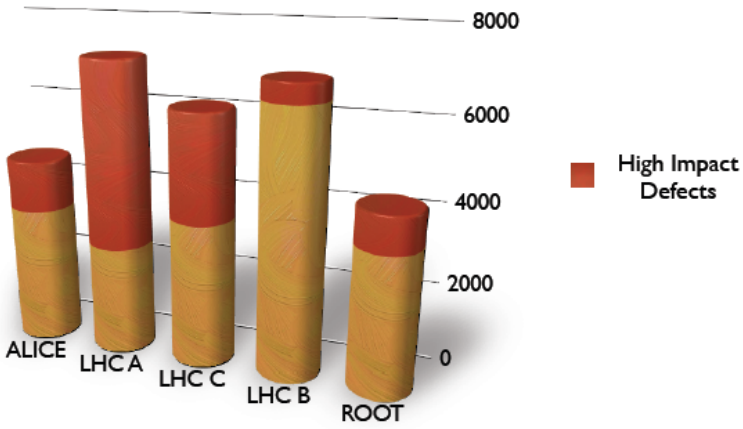
- **No common code & efforts for CPU / GPU tracking. GSI is trying this – Is such effort possible rather than having all track-based detectors going their own way?**
- **Within the same experiment: mix of CUDA, OpenCL, fork(), thread,**
- **%tage gain should be CAREFULLY stated (memory copy overhead, comparative to the whole workflow) or numbers are not meaningful – we need to define a standard matrix**

"Sequentiality" of a workflow (reconstruction) is killing everyone – do we have a wining strategy? Amdahl Law (often forgotten) is merciless in that regard.

Most approach need C-like structure (step back)
Has C++ killed our ability to exploit new architectures / hardware?

Static analysis of code

- Source code analysis in LHC
  - Detects coding issues, human errors, unchecked return values, …
  - Commercial: Coverity
  - Free: clang, checkcpp, cpplint
- Many problems found
  - Memory issues (pointer overflow, initialization, …)
  - Flow (logic flaw, conditional break, misspelled conditions, …)



C++ is too complex for us – coding is too difficult. Mitigation: testing, detection at commit, automated analysis. Is this a C++ theme?
Did we go in the wrong direction? Do we need a new language?

FS, data access, Data management & preservation

NFS 4.1 / pNFS, the final step – Patrick Fuhrmann (EMI/DESY)

Panel discussion – Jérôme Lauret (BNL), Dirk Düellmann (CERN IT), Patrick Fuermann (DESY), Jean-Yves Nief (CC-IN2P3), Samuel Skipsey (Gasgow U)

EOS disk storage at CERN – Andreas Peters (CERN / IT-DSS)

Advances in data management and Operations for ATLAS Data Management – Graeme Andrew Stewart (CERN / ATLAS)

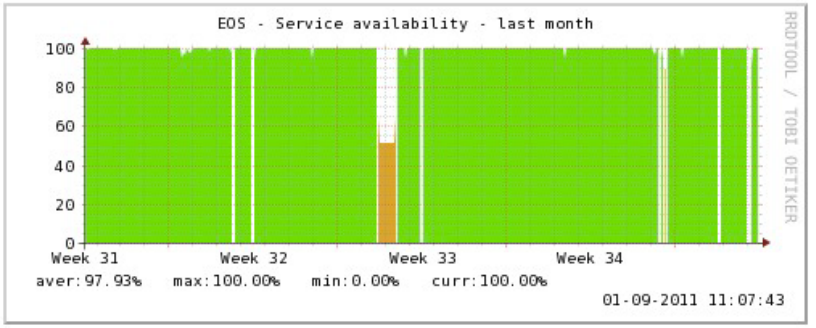
The LHCb DIRAC-based production and data management operations systems – Federico Stagni (CERN / LHCb)

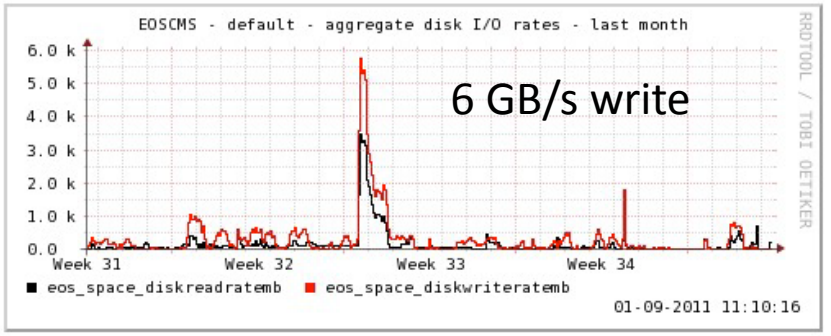
One click dataset transfer: toward efficient coupling of distributed storage resources and CPUs – Michal Zerola (NPIS-ASCR / STAR)

A Validation System for Data Preservation in HEP – Yves Kemp (DESY)

EOS disk storage at CERN

- A disk pool project – provides a layer between CASTOR and Clients for disk-only activities – resource become CASTOR+EOS
  - High availability, high performance
- POSIX access, Auth / ACLs, dynamic pool size, checksum (2 Mio checked in 80s)
- Usage: ATLAS (3.8 PBytes / 1.55 PBytes / 450 users) and CMS (2.3 PBytes / 800 users)
- Stability: 98%
- Move from rapid development (past 15 month) to reliable production mode

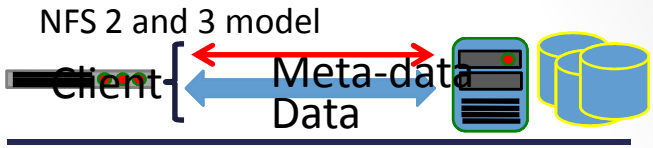


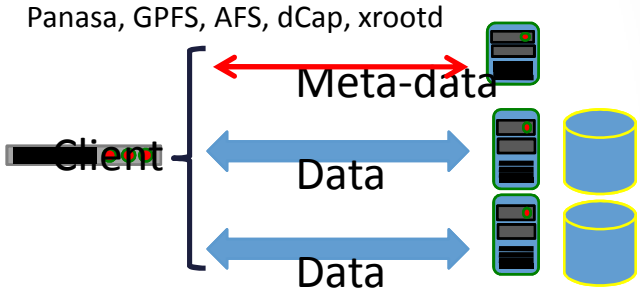


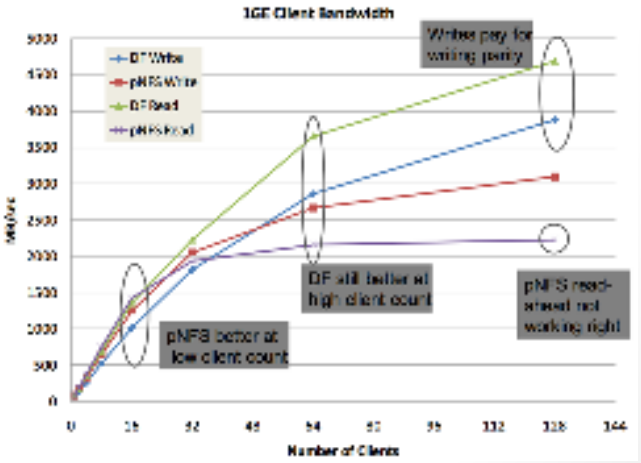
Past solutions do not seem to perform under demand and load – new scalable and distributed approach are being investigated …

NFS 4.1 / pNFS, the final step

- Transitioning from single data server to distributed data
  - Community diverge to provide missing scalability feature (Industry: Panasas, BlueArc, Lustre, GPFS / HENP: RFIO, dCap, Xrootd, …) – pNFS / NFS v4.1
  - May help re-converge
- pNFS is an extension of NFS protocol, first version supports multiple data servers.
  - pNFS is known to dCache
  - Industry moving there – Panasas DirecFlow seem better than pNFS but "standard" may wins
- Regular mount point, POSIX & ACLs
  - Client caching done by clients / vendors
- Availability uncertain –
  - Industry: whoever comes first gets "graded" ☺ - perhaps 2012
  - Server: dCache, DPM, StoRM
  - Client in
- Other server implementation and usage in the community need to be seen …



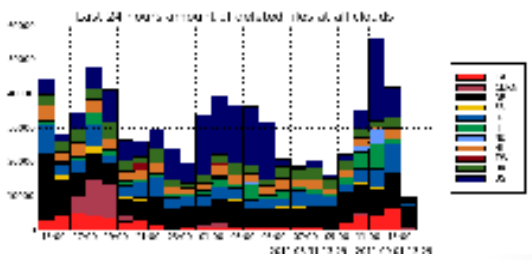




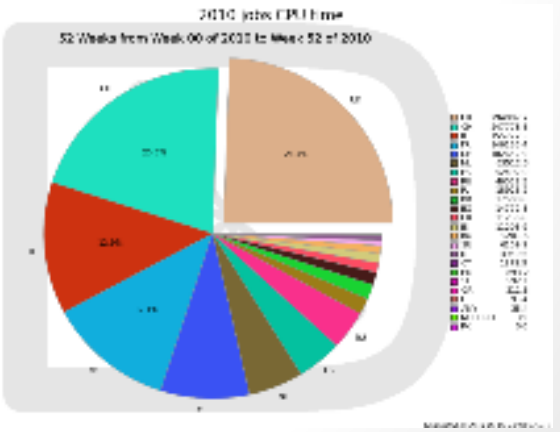
Distributed data coming from pNFS – industry behind
Path seem encouraging & coming pretty soon

DM – consolidation & status

- Graeme Andrew Stewart provide an overview of the ATLAS DM advances
  - A working scalable service able to cope with current load
  - Central Catalog: 20 M read/day, 1 M writes/day
  - ActiveMQ for Oracle writes to provide enhanced scalability for tracer
  - Optimization and system tuning – "consolidation"
  - New/enhanced features: deletion (with care of overlaps with multiple datasets – Perf 1.5-2.0M files/day), consistency (replacement and mark suspicious), checksum
  - DM will soon be under a re-design / re-thinking – part of evolution to face new realities and demands

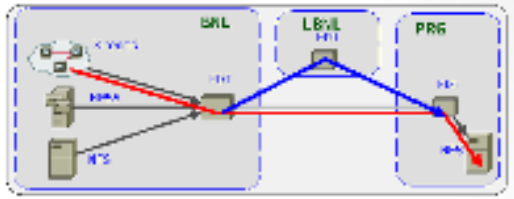


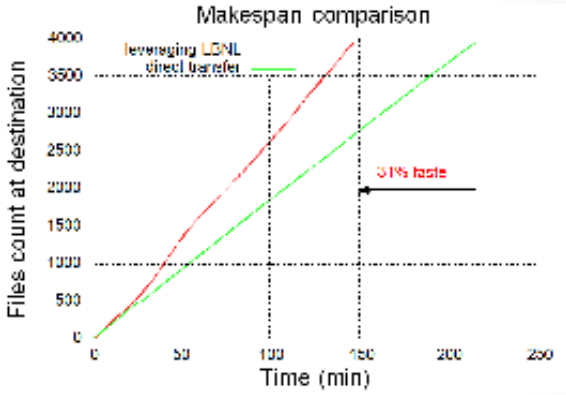
- Federico Stagni presented a review of the LHCb DM system
  - DIRAC, a community grid system with security: SSL/X509
  - JDL based for workload management with support for templates, job transformations, … data handling within (move with jobs)
  - GUI interfaces for monitoring – large UK component for non-data, predominantly CERN for real-data (30%, next is 15-16% UK and FR)
  - Very efficient with < 5% cycles wasted
  - Seem scalable, ready and working well for production



DM – little R&D

- Michal Zerola (well, me really) presented for STAR the development of a data placement planer for a NP-hard problem
  - One click data transfer – specify what, where to transfer and click
  - Make use of advanced Computer Science techniques: CP and MIP for the solver
  - Showed to provide shorter transfer time over P2P and multi-network path transfer faster by ~ 30% comparing to one site to one site, system is adaptive to failures (storage service, network downtime) and load balance
  - Next phase: coupling storage & CPU





- Yves Kemp presented a validation system for data preservation in HEP
  - Gave the impression of a dynamic (re-compile-code framework) for a long term preservation problem
  - Made peace afterward ☺ - the components in place are there to allow re-install of software in the EXACT same old OS but with evolving VM formats … in other words: no brainer → Code preservation – isolation in VM with Cloud like idea - "*Don't give me a better result, just give me the same result*"
  - Software, Libs, OS layered "in" + test suite – test software but also services it needs to access (do they still work?)

Slew of topics generates from there …

Panel discussion on DM

- Data preservation
  - Preservation of data is one thing, preservation on how to do this is another. Knowledge preservation is not a trivial issue
    - Experiments should start (perhaps even before they get data).
    - Some tips: validation of analysis, preserve notes on how to reproduce, documentation, codes, data, … and verify ASAP!
  - Preservation of Meta-Data
  - But also – storage capacity increase / yesterday's data at the mercy of a one tape loss (T10Kc / 5 TB today, 20 TB cartridge on roadmap). May want to checksum at tape repack
  - Some issues of perceptions and expectations on "preservation"
    - Data loss is UNAVOIDABLE. 100% accurate / up-to-date catalogs are like unicorn – sounds good in stories but not real!
    - Use mitigation such as mark files un-available (replicas may still be fine) when node/disks are down.
- Model for DD
  - "On-demand" versus pre-placed
  - CPU oriented jobs, should be fine – IO oriented jobs, beware of network bandwidth requirements (LAN or WAN) – 1,000 jobs access to remote "far-away" data is expensive
  - Cache size need to be adequate and tailored to problem – this needs to be measured (cache hit) otherwise, not wise to schedule jobs without data plan . Data can be asked to be placed PIOR to submission.
  - Sequential IO: read-ahead works super! But random event fetching? Unlikely …
  - Overall, smooth transition from one model or another or combined models are bound to occur
    - "outsourcing" the data especially as networks become more beefy)
    - Sites like "some" datasets – no doubt that pre-placement will remain
  - Should RMS know about cache states? YES

Panel – other items

- Multi-core & Network and DM issues?
  - Multi-core
    - Svere –Need to get back to efficiency by locality (and avoid scattered objects)
    - But Mega-Multi-core machine may generate lots of data "in-situ" – could be difficult to move it (TeraFlop machine then ExaScale) push in that direction by the 'powers up there'
      - Start to see network Mega-Highways between a few sites?
      - More processing in the same workflow (don't move data and do analysis along data reconstruction)?
      - Reduce data size further?
      - All of the above?
    - New communities? LSST @ 150 PBytes over 10 years / other similar generating TB per day per experiment x dozens?
    - Solution may come from private sector (video industry, …)
  - IO problems with multi-core?
    - In some cases, model of data on local disk is broken
    - SSD expensive, cannot see it as long-term solution (perhaps for a while)
    - Balance IO/CPU is shifting toward IO – problems are coming and we will see

As the we ramp-up in the multi-core era, will we see more distributed IO in the coming years via dedicated IO facilities or services? IO harvesting? Hadoop like services?

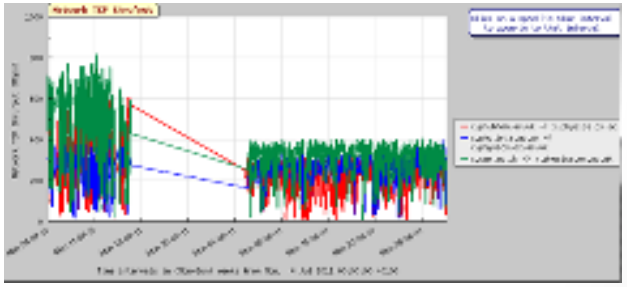
Would combined workflows (production and analysis) help/needed to reduced outputs?

VM, Clouds, Grids, service provisioning

Building an Outsourcing Ecosystem for Science – Kate Keahey (ANL / MCS)
Computing on Demand (PoD) – Anar Manavof (GSI)
Monitoring the Grid at local, national and global levels – Peter Gronbech (Gridpp)
Dynamic deployment of a PROOF-based analysis facility for the ALICE experiment over virtual machines using PoD and OpenNebula – Dario Berzano (INFN / ALICE)
Integrating Amazon EC2 with CMS production framework – Andrew Malone, (Vanderbilt / CMS)
Offloading peak processing to Virtual Farm by STAR experiment at RHIC – Jan Balewski (MIT / STAR)

GridPP monitoring

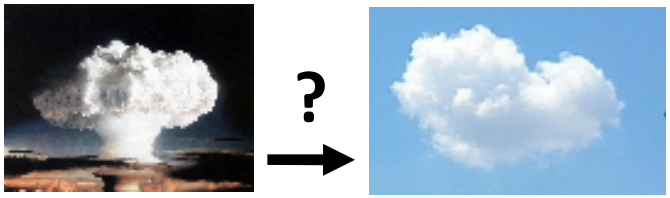
- Pete Gronbech overview of monitoring in the UK
- Multiple level of monitoring: local, national, global
  - Ganglia, PBDSWebMon, Cacti
  - GRIDMON
  - Pakiti, sys logger, nagios
  - Very nice talk – left a taste of "a zoo of monitoring tools" …
- Conclusions
  - Probably too much information to ever fit on one dashboard
  - Systems Administrators will continue to need multiple screens to keep track of many web pages
  - They will have to try to consolidate these with customized dashboards, Or perhaps ...



Immediate thoughts

- **Several tools developed in the UK – how is this re-integrated into the WLCG project (if at all)? – in-house knowledge and efforts see seldom survival rate over larger projects tools – Collecting the lesson learn essential.**
- **We know how to monitor but (far) too many dashboards & not enough alarms and auto-detection (must be a focus onward)**

Clouds



- Clouds: A form of distributed computing
  - Coming from industry, with virtualization at its heart
  - Innovation – Lose (and diverse) steering with many low hanging fruits
    - from VM to VC ⇔ from disconnected resources to Cloud
    - Sustainable isolation / virtualization
    - Low barrier / easy of use (Grid tend to be complex / may can start a Cloud within a hour)
  - Infrastructure clouds are increasingly getting adopted in scientific community
    - Genome science , dark matter studies, Sky computing – FutureGrid and Grid 5k, Babar (VM for data preservation / Cloud scheduler), STAR (burst of resources), Ocean Observatory … observable science
- Usage and patterns
  - On-demand processing / high elasticity: observatories, experiments, conference deadlines, …
  - Multiple providers: risk-mitigations, …
  - Seamless integration of heterogeneous resources
- We can't have unlimited cycles – but how they are provisioned should be driven by need rather than technology limitations. Infrastructure clouds provide such mechanism – it is now a question of how to use it best
  - Simplified infrastructure building
  - Support all aspects of complex deployment: a Grid in your pocket
  - Provide a "power adapter": automatically regulate how much or how little of this power you will use
  - New concepts – like initd, cloudinit.d light configuration for portability and repeatability

amazon web services · NIMBUS · openstack · Eucalyptus

What would you do with dynamic resources?

"You've got to be thinking anyway, you might as well think big"

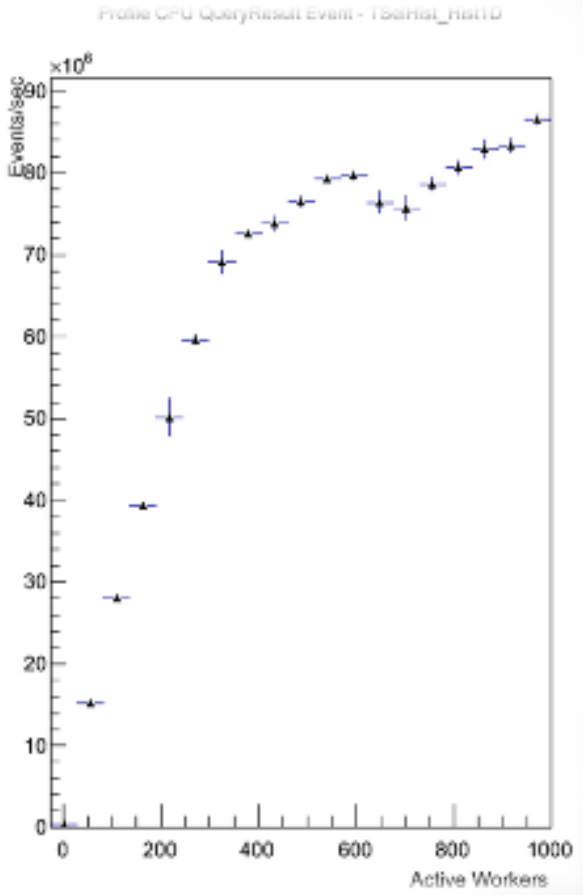
Examples

- Dario Berzano fro ALICE showed an example of "elastic" software provisioning (PROOF)
  - Dedicated cluster – efficient use of cycles
  - Overload → raise more PROOF nodes, read data over Xroot
  - Transparent deployment of virtual PROOF nodes
  - Interesting (and self-consistent with CERTs) to propagate SSH keys inside the VMs for remote connection
- Andrew Melo in CMS uses Cloud / EC2 for expanding resource pools
  - Starts a VC looking like yet another OSG resource/site to CMS
  - Condor starts VMs
  - Use regular approach to job submission on grid afterward (CE service)
  - Will be Looking to use EC2 / SE as well in future – IO in/out of EC2 unclear
- Jan Balewski in STAR showed a use of large amount of Cloud resources on Magellan for real-time real-data processing
  - Physics oriented exercise – Speed up science deliverables – real-time production
  - Data streamed from online to the Cloud – no batch system but a provider/consumer model (very much like a "pull" model)
  - The whole software is shipped out in a VM (2 GB image) with embedded database "snapshot" service
  - Software provisioning via VM ensures result reproducibility and QA

From service provisioning to absorb demand & load to on demand resource provisioning to *scaling up* for speeding science, Clouds seem beyond proof of principles but real usage / exploitations are happening NOW

Computing on demand

- Anar Manafov expand the conceptual ideal of Computing on demand with PoD
  - Hardware on demand – with or without RMS
  - Software on demand (tools or analysis software) …
- Tried other avenues to create a cluster of scalable yet elastic PROOF WN – did not succeed
- PoD fills the gap
  - Start server
  - Submit
  - Enjoy
  - Can start 100 of nodes in a few seconds
  - Pod-remote from a UI = a laptop
- Benchmark
  - scales well to 350 VMs / master (efficiency border)
  - Issues with packettizer under work at 400 VMs+
- Believes that
  - PoD is a scalable and a perfect tool for PROOF on the Cloud – simple solution
  - Working on 900+ servers per master
  - Addresses “on-demand” interactive analysis



Note however that those methods change the balance between batch / other work. Data remains at the same site –Xroot access

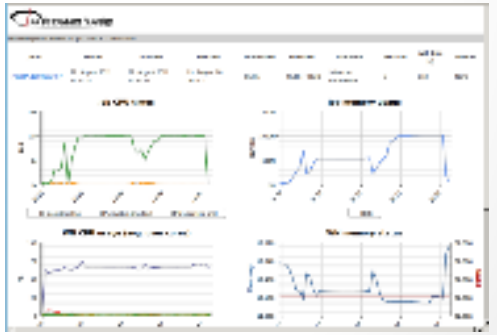
Other topics

The SALAMI Project – David De Roure (Oxford)
The AAL project : Automated Monitoring and Intelligent Analysis / ATLAS – Luca Magnoni (CERN / ATLAS)
Application of Remote Debugging Techniques in User-Centric Job Monitoring – Tim Dos Santos (Wuppertal / ATLAS)
Moving ROOT Forward – Fons Rademakers (CERN)
Online Measurement of LHC Beam Parameters with the ATLAS High Level Trigger.– Emanuel Strauss (SLAC / ATLAS)
An Exploration of SciDB in the Context of Emerging Technologies for Data Stores in Particle Physics and Cosmology - David Malon (ANL)
The PROOF Benchmark Suite Measuring PROOF performance – Sangsu Ryu (KISTI, Korea)
Do regions of ALICE matter? (Social relationships and data exchanges in the Grid) – Federico Carminati (CERN / ALICE)

Monitoring and Debugging

- Luca Magnoni - Automated Monitoring and Intelligent Analysis / ATLAS
  - A supporting work for a scalable TDAQ, information collection and analysis
  - Collection leveraging ActiveMQ
  - Real-time event processing using CEP (Complex Event Processing)
    - ESPER - Powerful Event Processing Language (EPL)
    - SQL-like
  - Detection "knows" about common problems
    - Generating alert, notification, statistics as soon as incoming events meet the constraints of the rule
  - System working since June

- Tim Dos Santos presented work which reminded me "something similar" done by Tim Muenchen (ACAT08)
  - Status of the work – tried in the context of PanDA / ATLAS
  - Remote debugging on the Grid – user space tools gathers and monitor data from apps
  - User-centric information: Memory usage, issues (locks & timeouts), …
  - Web UI to present the results
  - Development includes MQ messaging to control the job (encrypted msg)



Work dynamic continues on many front – Inventive work, new approaches and ideas with a net trend: all kind of MQ flavors often used today for information broadcasting / collection.

ROOT

- Fons Rademakers presented ROOT advancements
  - Balancing act between development and stable production mode tool
  - Cling
  - Improvement in IO - // Tree merge, support for Google storage, …
  - New RooStats features
  - PROOF improvements – packetizer
  - Multi-master improvements
  - iOS based devices support (tablets and smart phone) except graphics modulo: no X11, no user shared libs, touch GUI (not the normal one) for control, fingers instead of mouse / no-click, …
  - Graphics improvements – native display on OS-X
  - New TeX engine
  - Infrastructure / gmake, cmake / Drupal based Web site / documentation goes in DocBook / new continuous build system & coverity
  - …

Changes to be phased in incoming releases

Consolidation of IO and languages (LLVM/Clang), "hot" topics today

Focus on new device / portable devices (iPad, SmartPhone) takes care of new world reality [the only project integrating development for new devices]

Others …

- SALAMI – interesting work and other community with large datasets and different problems
- Work in ATLAS to understand how to feedback information to HLT for vertex reconstruction
  - Considering beam position & Vertex resolution, pileups.
  - Thought to be a thorough work – could we learn from/in past experiments?
- Very interesting presentation from ALICE – do region matters
  - Grid is a grid of people – social network analysis of connections
  - Not sure how the data would be used but may be used to help enhancing communication in some region (with some social context & specificities to consider)
- Exploration of SciDB by David Malon
  - Emerging DB technology
  - Probably not a solution for HENP (Objectivity “fun” anyone?)
  - Array Query Language – AQL (looks like SQL)
    - Possible import of ROOT files into SciDB (python script)
  - But may be usable for HENP but other communities
    - LSST may leverage and we may learn something in the process
    - EPICS effort from BNL
- PROOF benchmark
  - To understand and improve RPOOF – multiple dimension studied (N clients, N cores)
  - TProofBench in 5.29 and beyond

Some conclusions

- Very active track with lots of outstanding work (research and a well as consolidation) to understand
  - How things work and how they can work better
  - How we can scale higher
  - How can we integrate new technologies
- Technology emergence – exploitation
  - Clouds & computing on demand
  - Here to stay in my view – virtualization is too practical hence too tempting + interfaces getting better and better
- Data management – many questions still and landscape not clear with incoming new technology – should keep an eye in the coming years
- **Little on new algorithm this year**
  - Refocus on ACAT goals (Advanced Techniques) needed?
  - However – Rise of MQ based technology & systems (was barely visible @ CHEP in Taiwan)
- **We should pay attention to incoming new devices – they are part of our every day world**
  - iPad / SmartPhone …
  - Or even the fact that **new screens & displays are tactile (finger rather than mouse**)
  - … and so may be Windows 8 and new OS directions

Some conclusions – Multi-core & GPUs

- Frameworks for RHIC, the LHC and older experiments are already written and in production mode – changes not always easy
- ~~Perhaps~~ **newer experiment** (FAIR? ILC? EIC? Further out?) could address (**have a mission to address**) **multi-core problems** at the start [some are]

- Clear problems /challenges with new architectures – many tries / no clear path
  - **No re-usable algorithms – we must focus on a "core" strategy (multi-core ready libraries? Math, track algorithmic, …)**
  - **No "standard" approach or global strategy** – all is tried (fine); integration remains unclear (OpenMP, CUDA, OpenCL, fork, …) . They do not always mix up well!!
  - **Where are our software Architects?**

- Improvements over standard approach not always clear – **need standard matrix and consistent measurements to have meaningful comparisons**
  - Please present and state: relative %tage gain, absolute %tage gain
  - ALWAYS consider the time to copy data in/out of memory

- Other: Give up on full //ization? Workflow separation? Hybrid approach? Analysis and production merged into one workflow? IO harvesting and collector facility?

- Provocative but … **Did we take the wrong turn with C++?**
  - Good mileage in software development in one hand BUT systematically appears and *presented as a show stopper to exploiting new architecture*
  - **Feel is that we need for "a" new language?**

Cross experiments (or even within) workshops on this topic alone


Thank
You!