CHEP 2013

20th **Computing in High Energy and Nuclear Physics**

every 18 months (America->Europe->Other)

14-18 October in Amsterdam by Nikhef

http://www.chep2013.org

bias towards Atlas & databases

Tracks

big overlap between sessions

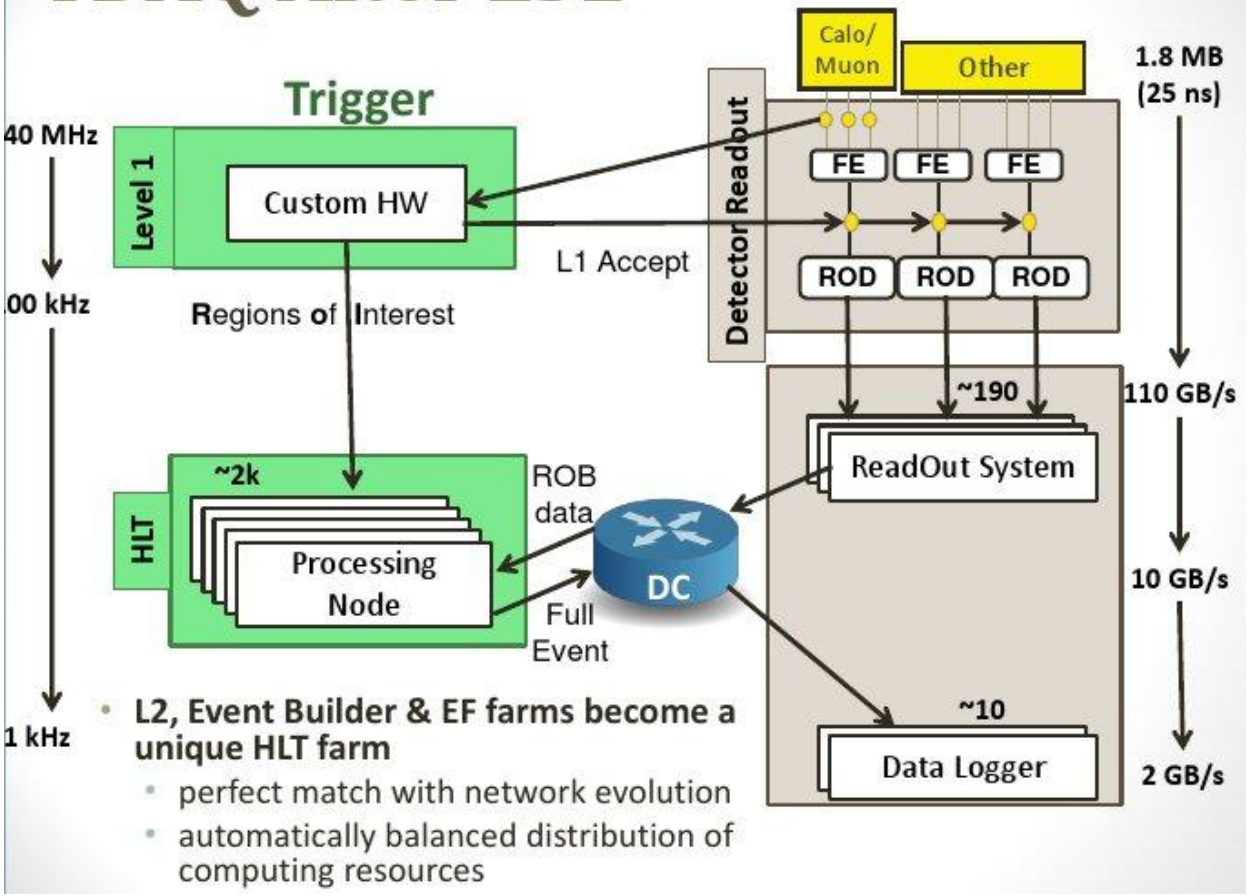
- data acquisition, trigger and controls
- event processing, simulation and analysis
- distributed processing and data handling
- data stores, data bases and storage systems
- software engineering, parallelism & multi-core programming
- facilities, production infrastructures, networking and collaborative tools

data acquisition, trigger and controls

- off-the-shelf equipment
- all-software solutions
- synergy with out-of-HEP

not my domain

TDAQ After LS1

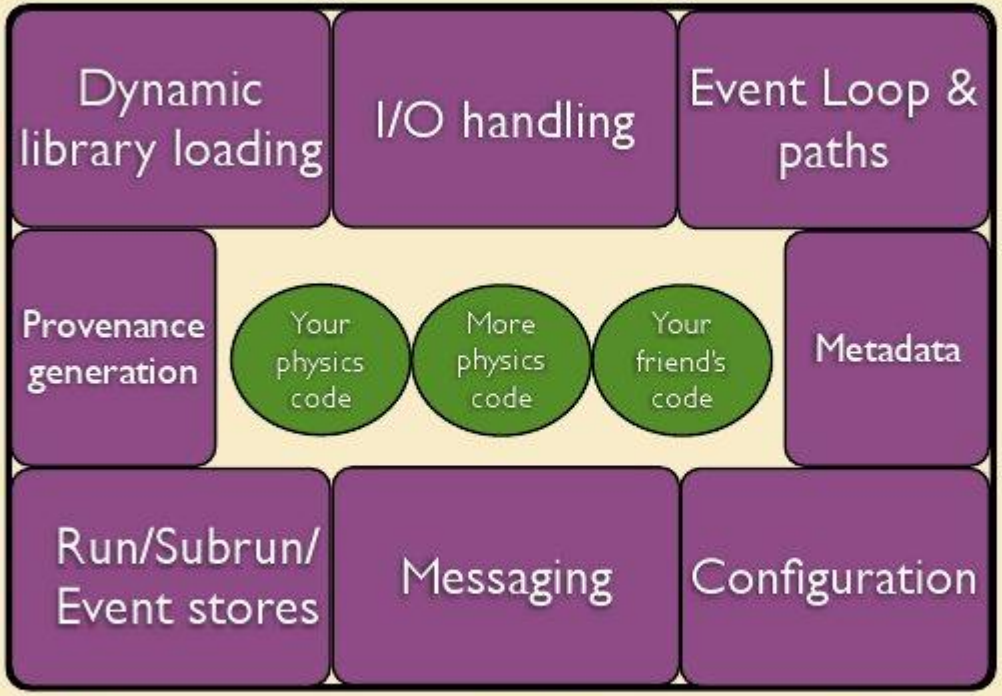


TriggerDAQ Example (Atlas)

event processing, simulation and analysis

- **Common frameworks**: Athena, Gaudi, Geant4, RooSomething,...
- **Concurency**: different approaches (Event-level, subevent-level, algorithm-level)
- **Simulation**: Pileup, Optimisation

What does a framework do?



Code you write

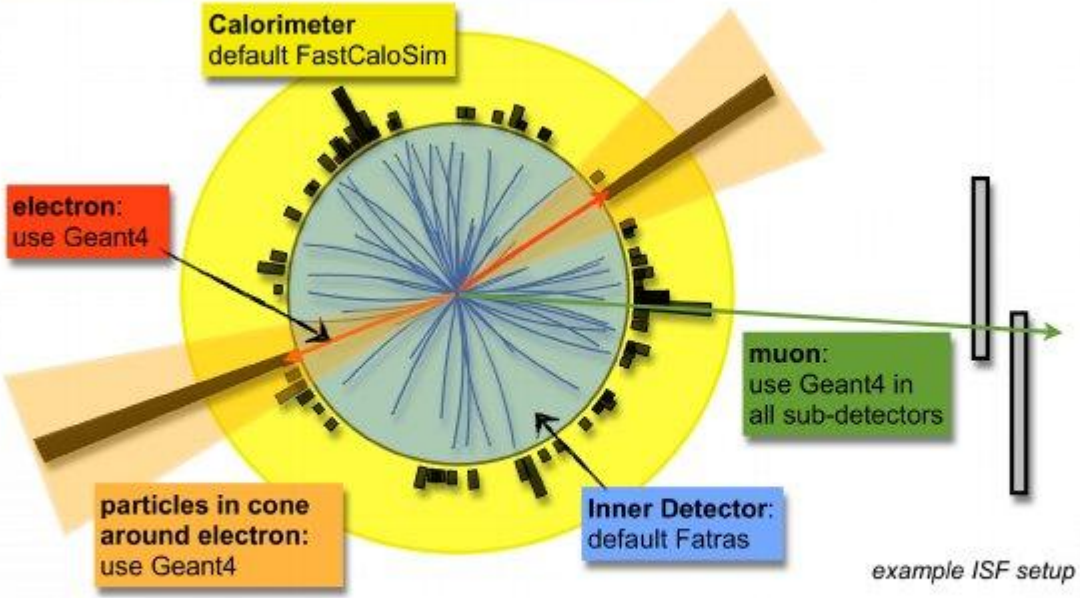
Code you use from the framework

Adam Lyon

what is Framework ?

- CMS:
  - Run multiple events in parallel, within one event run multiple modules in parallel, and within one module run multiple tasks in parallel
  - Use Intel Threaded Building Blocks (TBB) for all the parallelization
- ATLAS:
  - Use scheduler to start task when input data is ready
  - New scheme is implemented using TBB
- FairRoot
  - Use Multi-Process instead of Multi-Threading
  - Communication and synchronization through message (data) exchange

Current simulation performances



example ISF setup

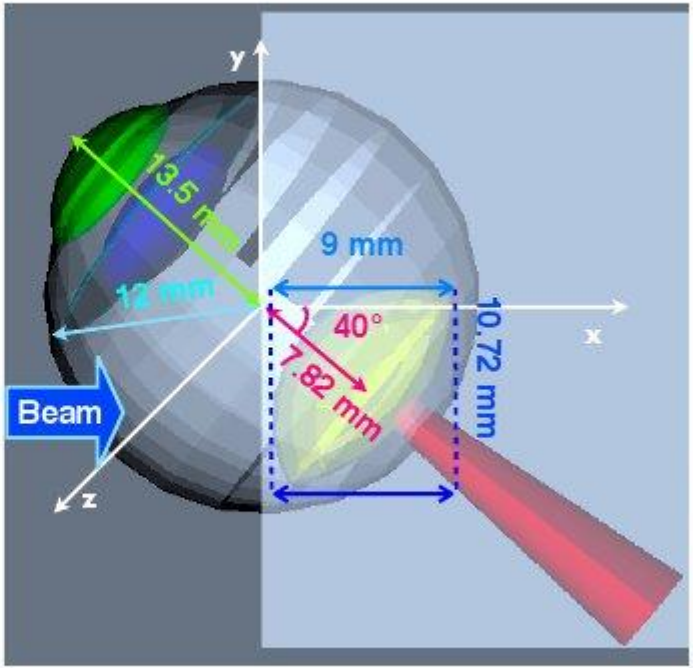
- ***Idea***: use different simulation techniques for the same event, depending on region or particle type
- Main feature: **flexibility** with respect to particles => simulator assignment
- Designed to be compatible with multithreading and multiprocessing

Chiara Debenedetti

flexibility in simulation

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



interesting application of Geant4

distributed processing and data handling

- busiest track
- opportunistic computing
- virtualisation, grids, clouds: CernVM
- **NoSQL** (mostly Hadoop - > 10 talks)
- **Common Frameworks**: Dirac, PanDA
- Data management, Federated resources, MultiCore, Distribution

data stores, data bases and storage systems

- IO, storage & data management: optimisation
- **databases**: Oracle -> NoSQL
- **metadata**: growing interest, databases or Root-files
- data preservation (long term)

Metadata services: ATLAS

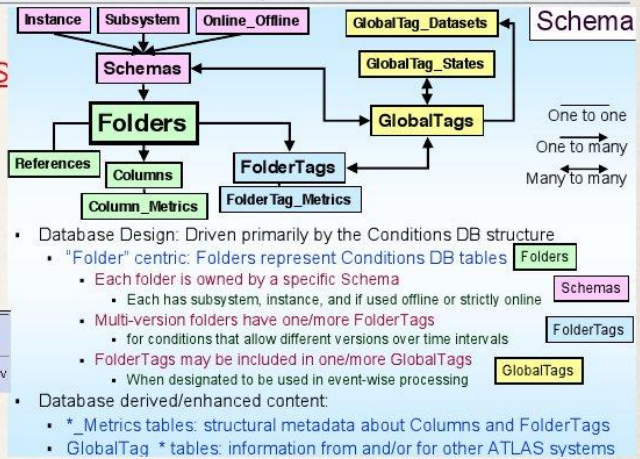
ATLAS Conditions metadata

Elizabeth GALLAS

- Structure on-top of LCG cond. DB
- Enhanced functionality and global view interfaces

Global_Tag_Name	Lock Stat	Description	Create Date	Folder Tag Count	AMI Dataset Count	Processing Date Range	AMI Project (s)
COMCOND-BLKPA-006-11	1	Based on COMCOND-BLKPA-006-10, new BULK tag with mu=0 conditions for run 200805 TMikh:CondionsTagsComCondBlkpa006111	2013-May-22 14:14	266	3352	2013-May-27: 2013-Sep-16	data12_8TeV

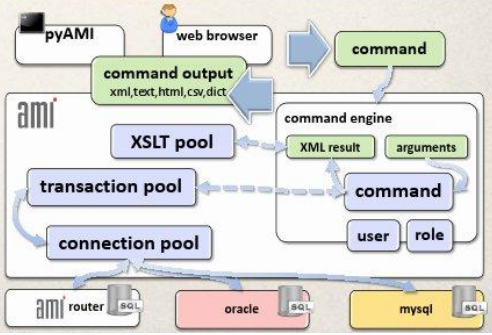
- Also significant cleanup in LS1



ATLAS Metadata Interface (AMI)

Jerome FULACHIER

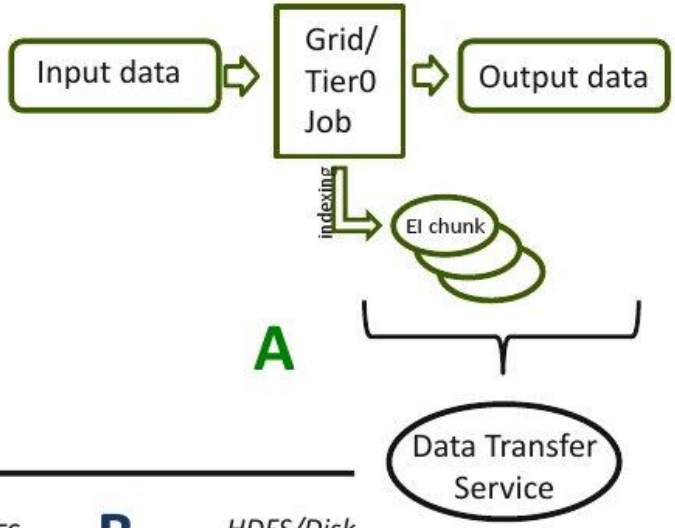
- 10 year architecture history covering:
  - User/role; Commands; query language; connection/transactions; production infrastructure; development methods ++

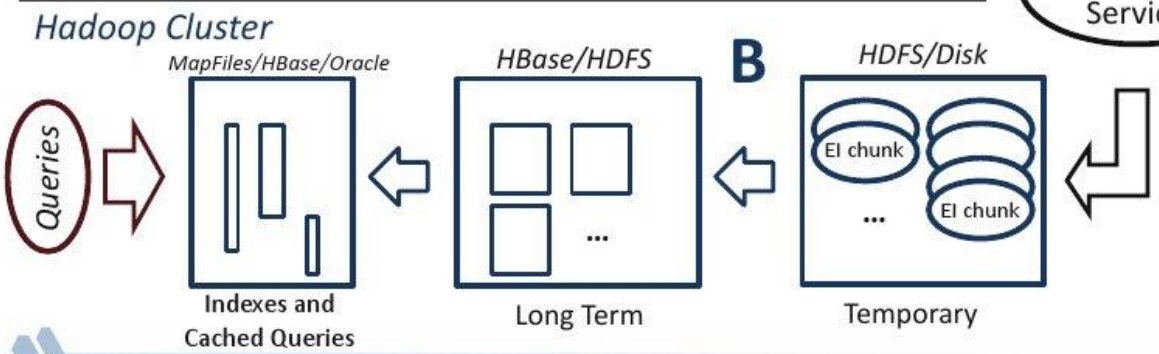


Atlas was a metadata pioneer

The general idea

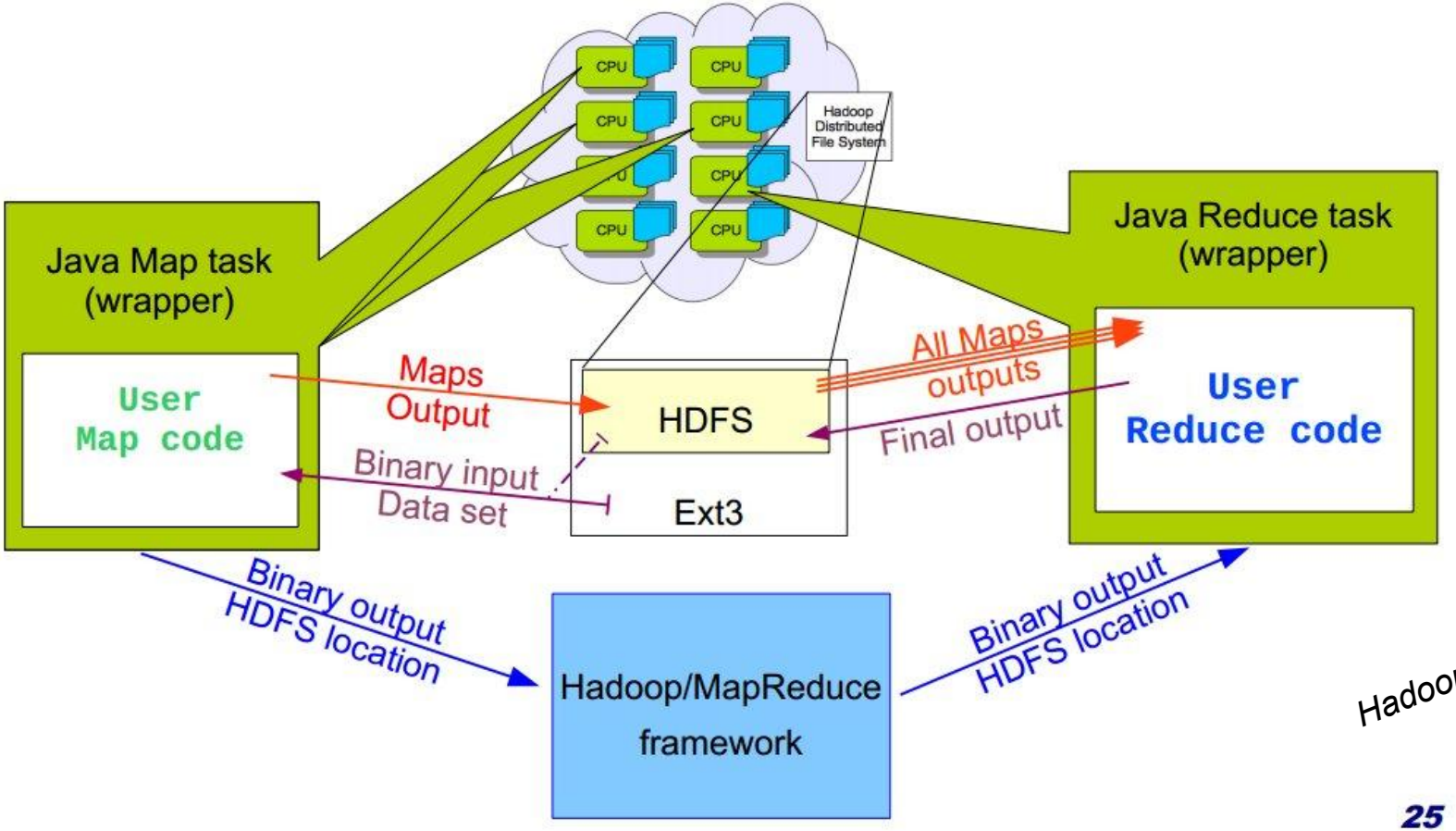
- Individual jobs will still produce the index data as EI chunks.
- Rather than being put into long term storage with the other output data, the EI chunks will be collected by a transfer service integrated with the Grid job management system, possibly using a messaging protocol, and put into a temporary storage where they can be checked and processed.
- These chunks are then transformed into long term data structures suitable for MapReduce processing.
- Finally, use case specific indexing and query caching can be added to improve performance.





metadata in Hadoop in Atlas

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



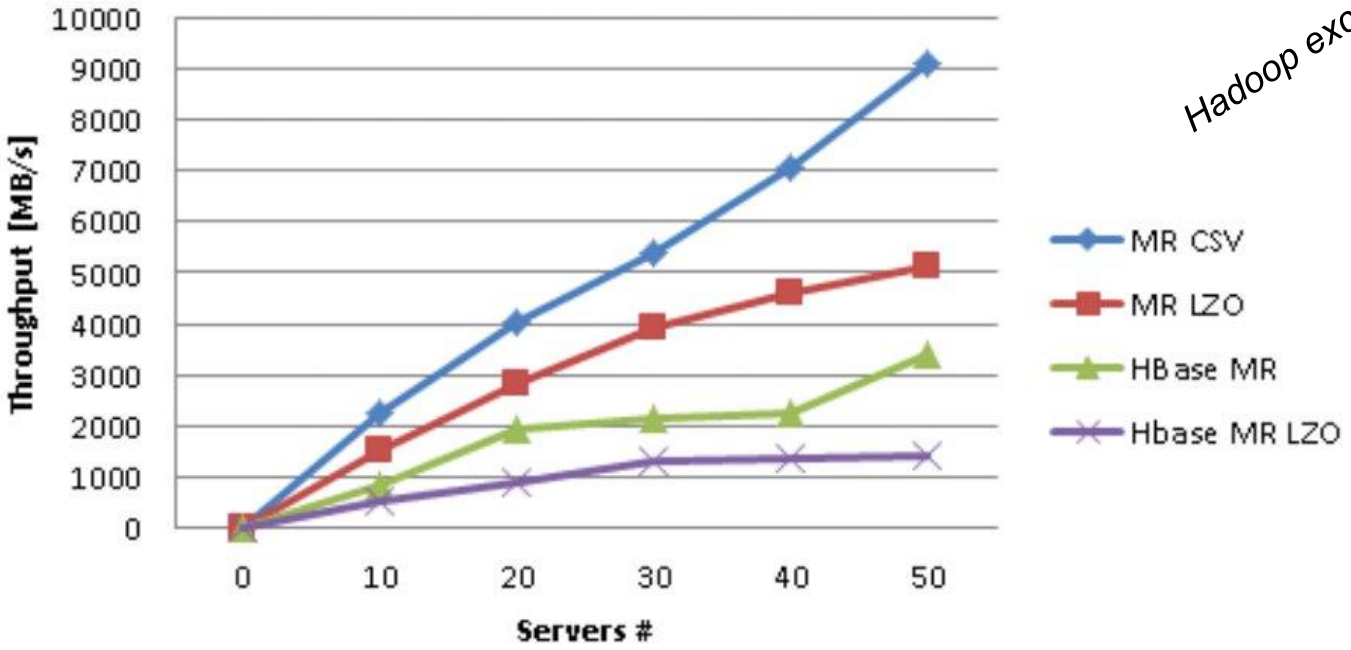
Hadoop with Root

Going beyond 5 node cluster...

Hadware used:

CPU: 2 x 8 x 2 x 2.00GHz **RAM: 128GB**

Storage: 3 SATA disks 7200rpm (~120MB/s)



Hadoop excellent scaling

software engineering, parallelism & multi-core programming

- vectorisation
- **concurency**: even-level, inside event, per algorithm,...
- suffering from language with almost no support for concurency (C++) => many home-grown solutions
- **beyond x86**: Arm, Xeon Phi, GPU,...

The computer-farm of the future

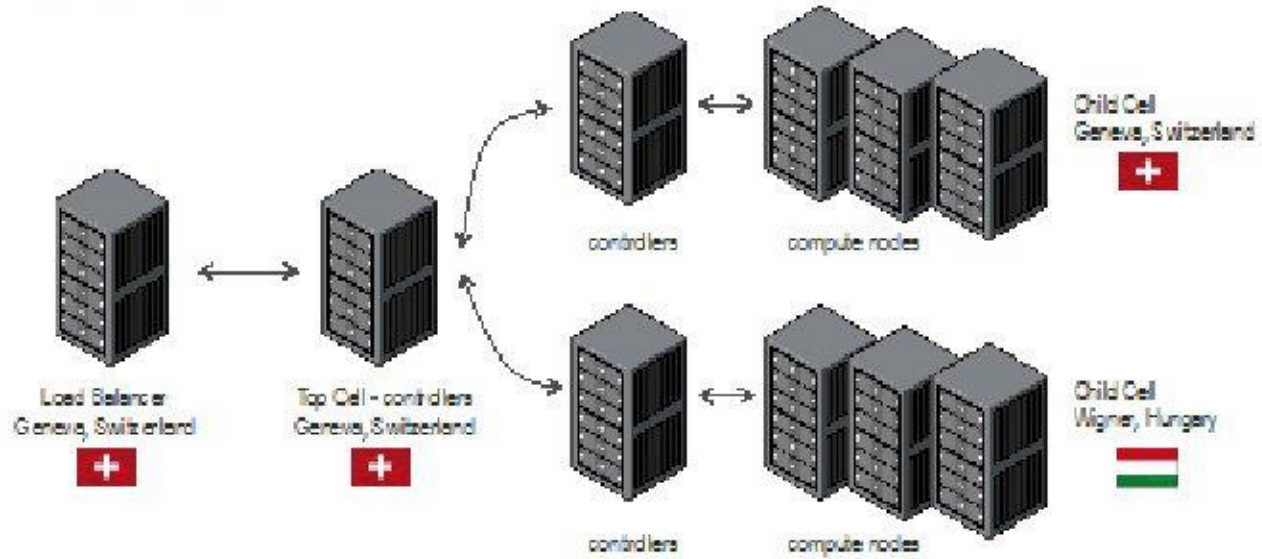
Not really ...

facilities, production infrastructure, networking and collaborative tools

- remote hosting, clouds
- collaborative Tools: Indico, Vidyo, **social networks**

not my domain

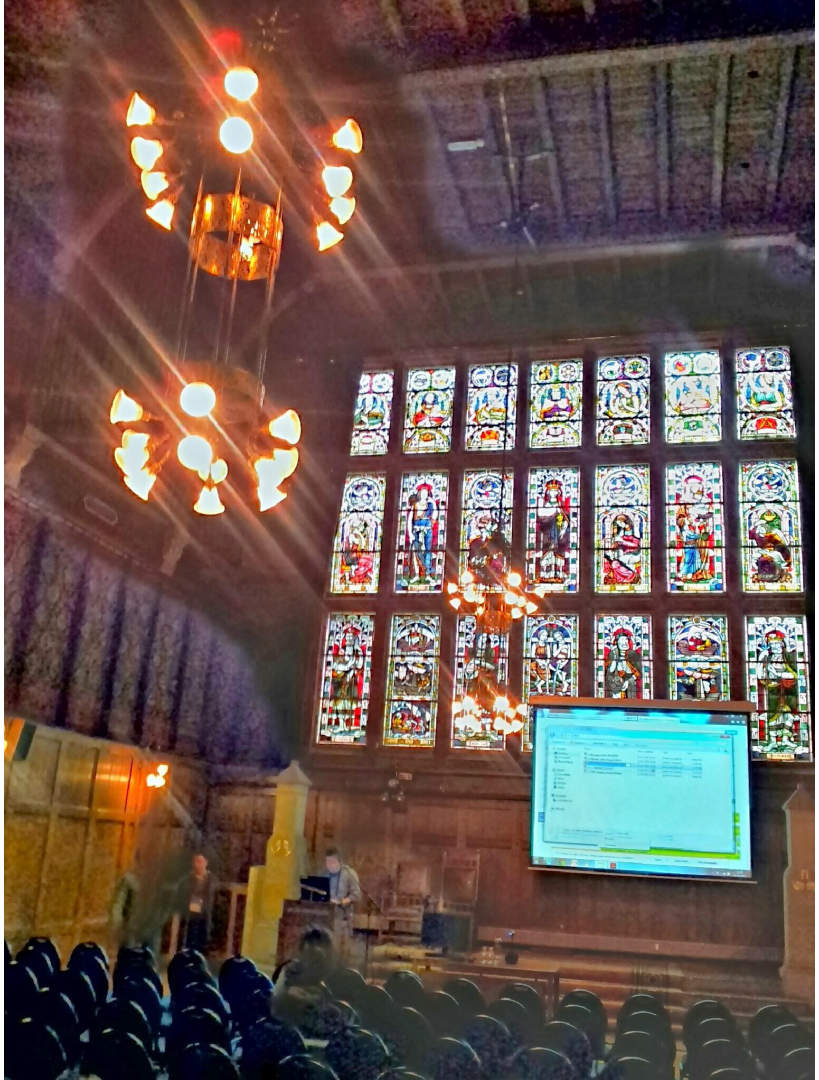
Architecture Overview



Summary

- Big Data (NoSQL, mostly Hadoop)
- Concurency, Parallelism, Vectorisation
- Virtualisation
- Commodity solutions (sw/hw)
- Personal observations:
  - Still fighting with C++ (trying to use it for tasks, for which it has very limited support)
  - Root problems (to interoperate with others) is often interpreted as its force - live discussion during Panel-Session
  - Very little about graphics

poster session
&
cafe



parallel session

conference dinner