Computing at the HL-LHC

Predrag Buncic

on behalf of the

Trigger/DAQ/Offline/Computing

Preparatory Group

ALICE: Pierre Vande Vyvre, Thorsten Kollegger, Predrag Buncic; ATLAS: David Rousseau, Benedetto Gorini, Nikos Konstantinidis; CMS: Wesley Smith, Christoph Schwick, Ian Fisk, Peter Elmer ; LHCb: Renaud Legac, Niko Neufeld

Computing @ HL-HLC

- CPU needs (per event) will grow with track multiplicity (pileup)
- Storage needs are proportional to accumulated luminosity

Run1

Run2

Run3

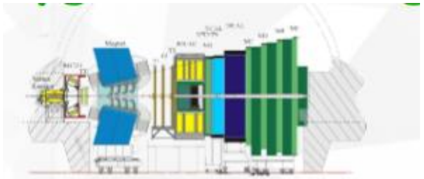
Run4

ALICE
+
LHCb

ATLAS
+
CMS

1. **Resource estimates**
2. **The probable evolution of the distributed computed technologies**

LHCb & ALICE @ Run 3



40 MHz

DAQ

40 MHz

LLT: μ,e/γ, hadrons

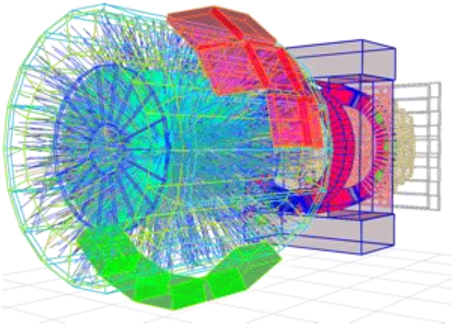
5-40 MHz

HLT: event recon.

20 kHz (0.1 MB/event)

Storage

2 GB/s



50 kHz

Reconstruction + Compression

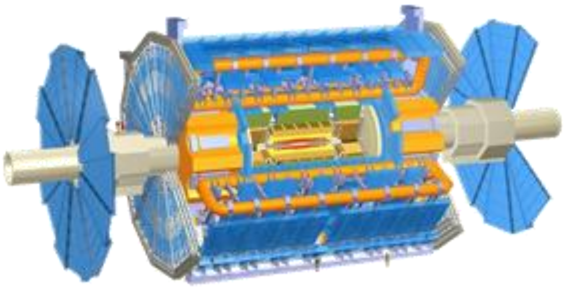
50 kHz (1.5 MB/event)

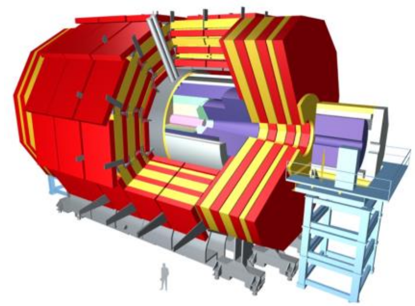
Storage

75 GB/s

← PEAK OUTPUT →

ATLAS & CMS @ Run 4

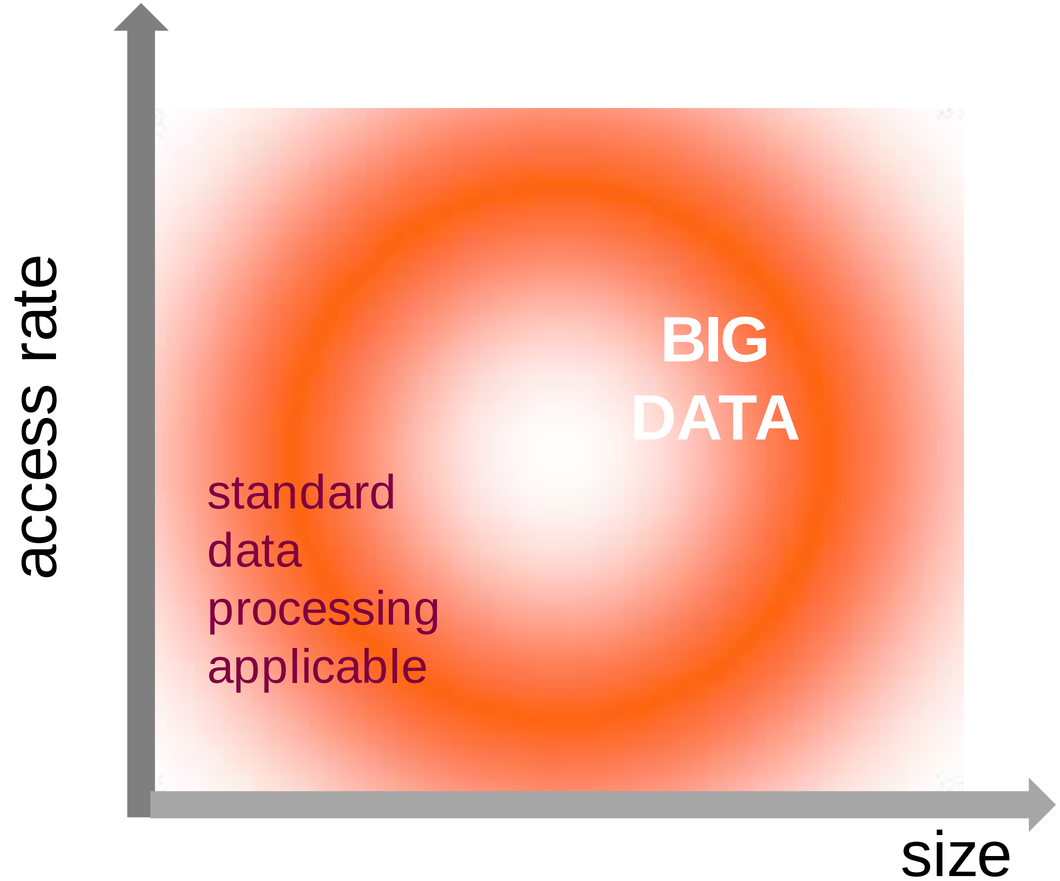




ATLAS	CMS
Level 1	Level 1
HLT	HLT
5-10 kHz (2MB/event)	10 kHz (4MB/event)
Storage	Storage
10-20 GB/s	40 GB/s

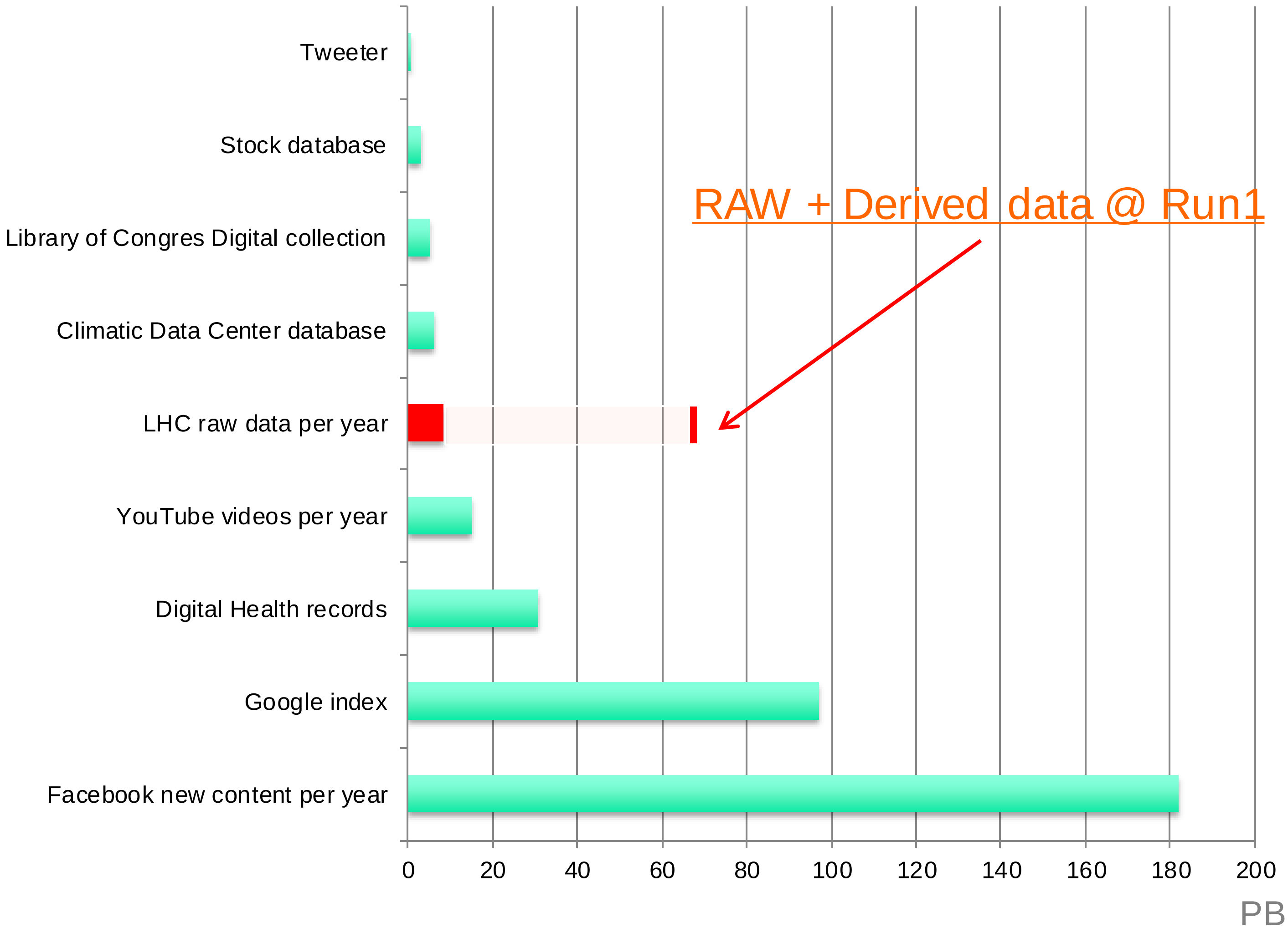
← PEAK OUTPUT →

Big Data

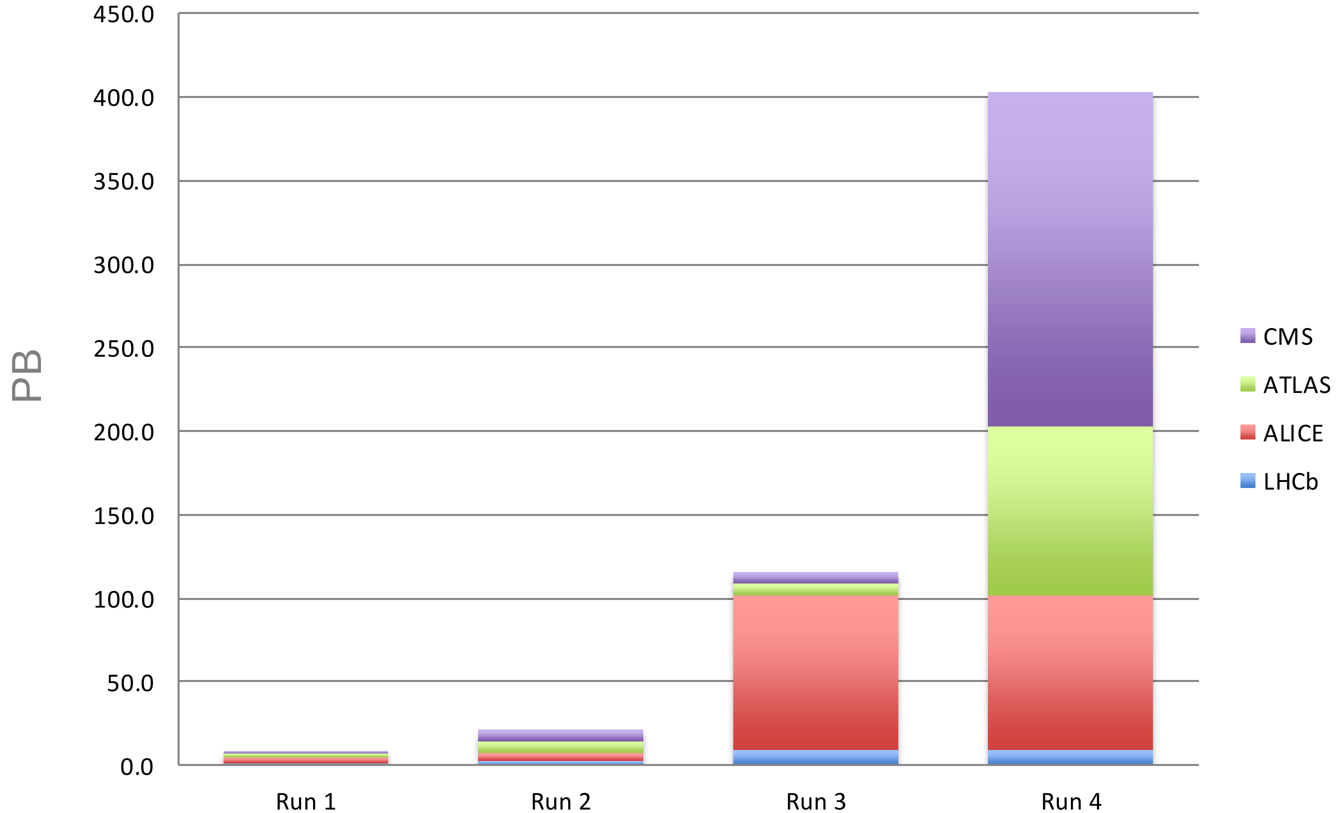


“Data that exceeds the boundaries and sizes of normal processing capabilities, forcing you to take a non-traditional approach”

How do we score today?



Data: Outlook for HL-LHC



- Very rough estimate of a new RAW data per year of running using a simple extrapolation of current data volume scaled by the output rates.
  - To be added: derived data (ESD, AOD), simulation, user data…

Digital Universe Expansion

Data storage issues

- **Our data problems may still look small on the scale of storage needs of internet giants**
  - Business e-mail, video, music, smartphones, digital cameras generate more and more need for storage
- **The cost of storage will probably continue to go down but…**
  - Commodity high capacity disks may start to look more like tapes, optimized for multimedia storage, sequential access
  - Need to be combined with flash memory disks for fast random access
  - The residual cost of disk servers will remain
  - While we might be able to write all this data, how long it will take to **read** it back? Need for sophisticated **parallel** I/O and processing.

\+ **We have to store this amount of data every year and for many years to come (Long Term Data Preservation)**

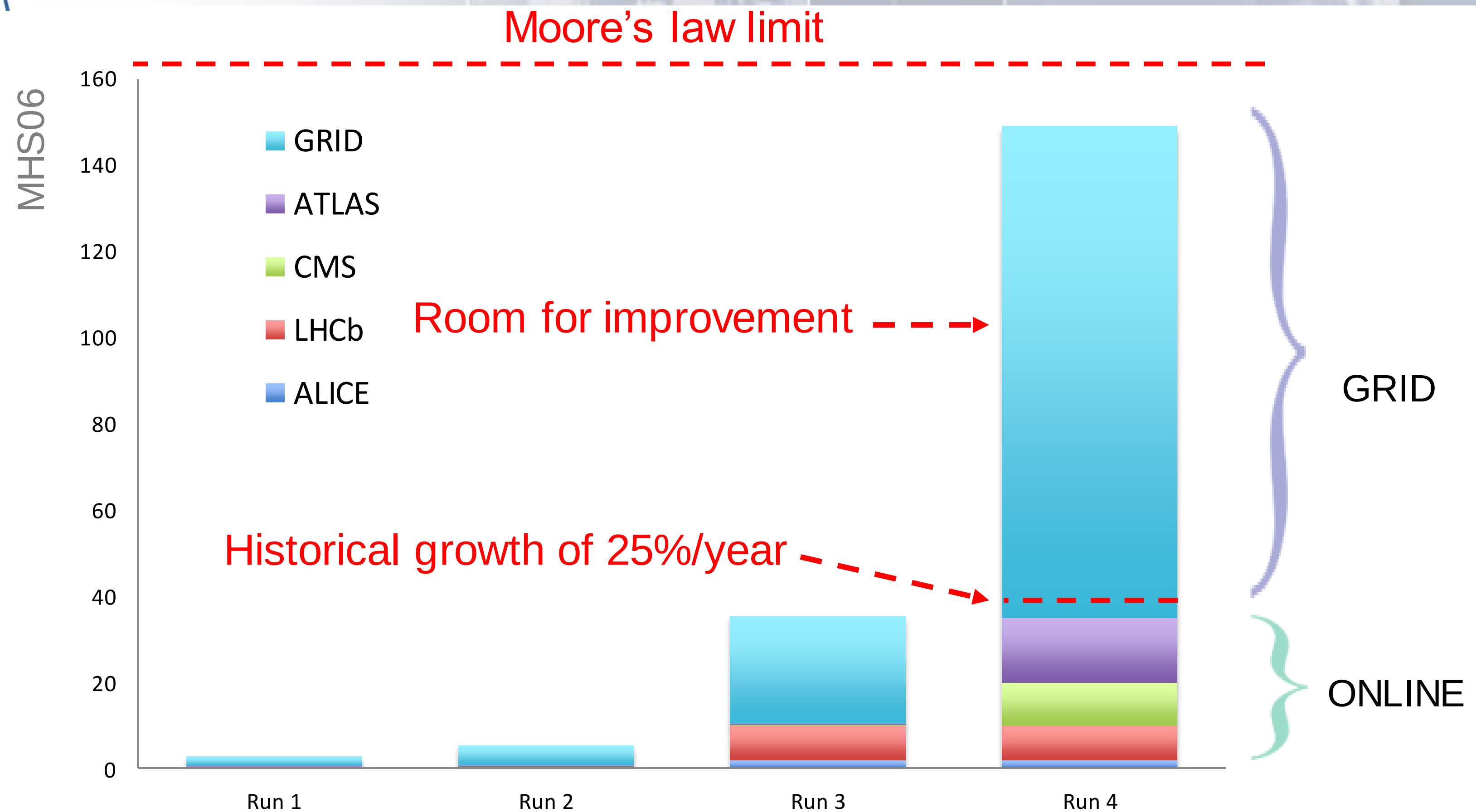
Exabyte Scale

- **We are heading towards Exabyte scale**

CPU: Online requirements

- **ALICE & LHCb @ Run 3,4**
  - HLT farms for online reconstruction/compression or event rejections with aim to reduce data volume to manageable levels
    - Event rate x 100, data volume x 10
  - ALICE estimate: 200k today's core equivalent, 2M HEP-SPEC-06
  - LHCb estimate: 880k today's core equivalent, 8M HEP-SPEC-06
  - Looking into alternative platforms to optimize performance and minimize cost
    - GPUs (1 GPU = 3 CPUs for ALICE online tracking use case)
    - ARM, Atom…
- **ATLAS & CMS @ Run 4**
  - **T**rigger rate x 10, CMS 4MB/event, ATLAS 2MB/event
  - CMS: Assuming linear scaling with pileup, a total factor of 50 increase (10M HEP-SPEC-06) of HLT power would be needed wrt. today's farm

CPU: Online + Offline



- Very rough estimate of new CPU requirements for online and offline processing per year of data taking using a simple extrapolation of current requirements scaled by the number of events.
- Little headroom left, we must work on improving the **performance**.

How to improve the performance?

- **Clock frequency**
- **Vectors**
- **Instruction Pipelining**
- **Instruction Level Parallelism (ILP)**
- **Hardware threading**
- **Multi-core**
- **Multi-socket**
- **Multi-node**

Very little gain to be expected and no action to be taken

Potential gain in throughput and in time-to-finish

Gain in memory footprint and time-to-finish but not in throughput

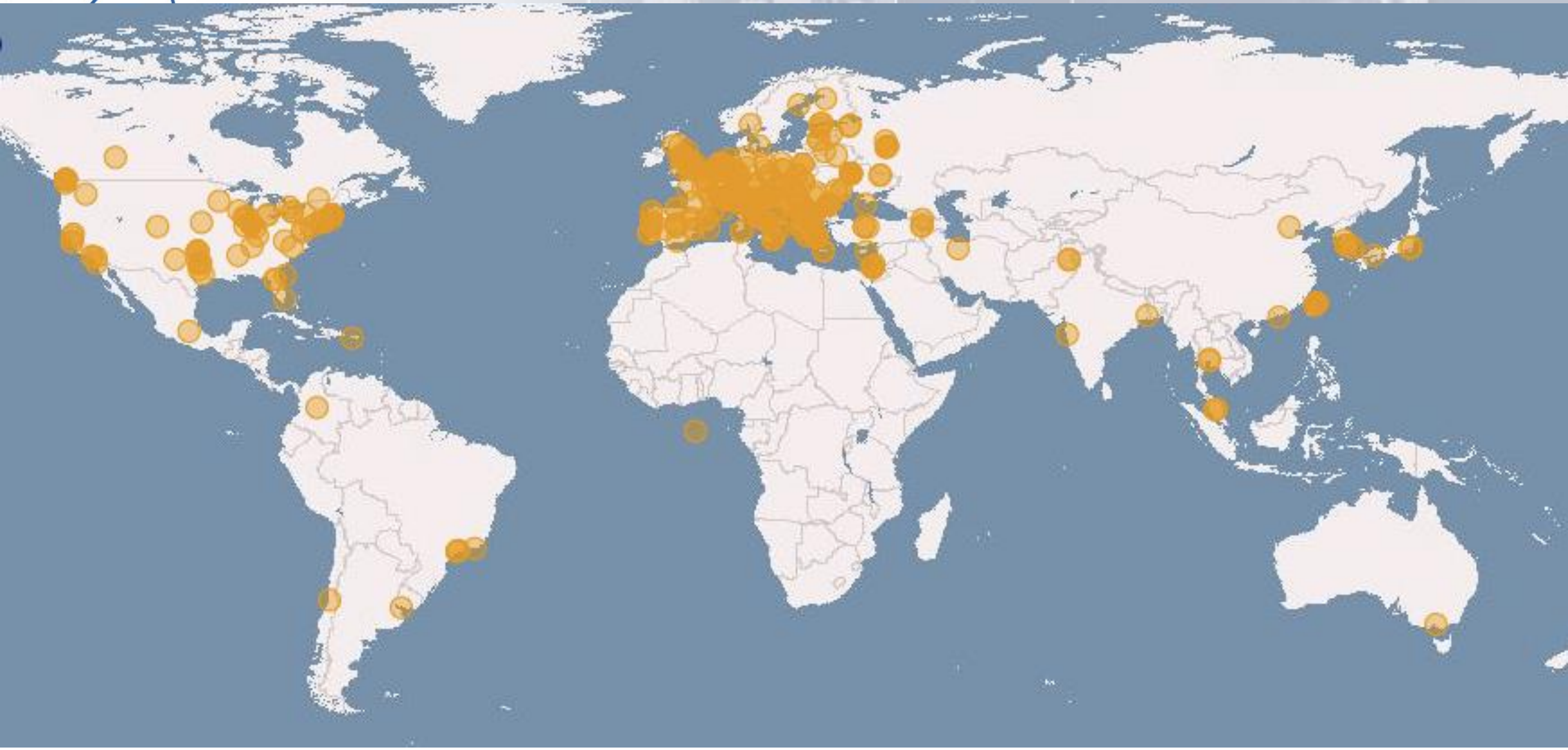
NEXT TALK

Running independent jobs per core (as we do now) is optimal solution for High Throughput Computing applications

THIS TALK

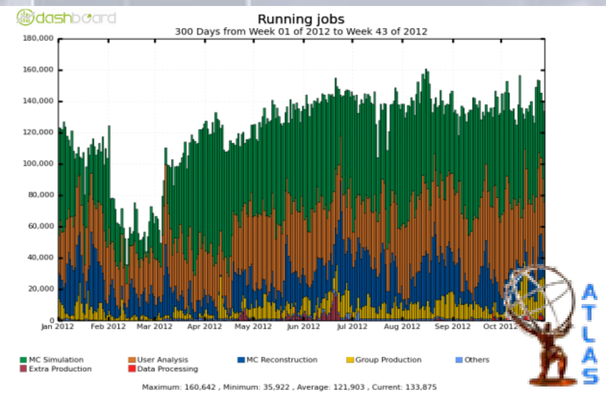
Improving the algorithms is the only way to reclaim factors in performance!

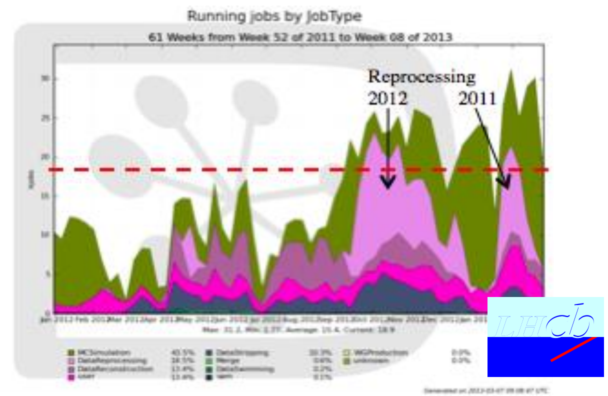
WLCG Collaboration



- Distributed infrastructure of 150 computing centers in 40 countries
- 300+ k CPU cores (~ 2M HEP-SPEC-06)
- The biggest site with ~50k CPU cores, 12 T2 with 2-30k CPU cores
- Distributed data, services and operation infrastructure

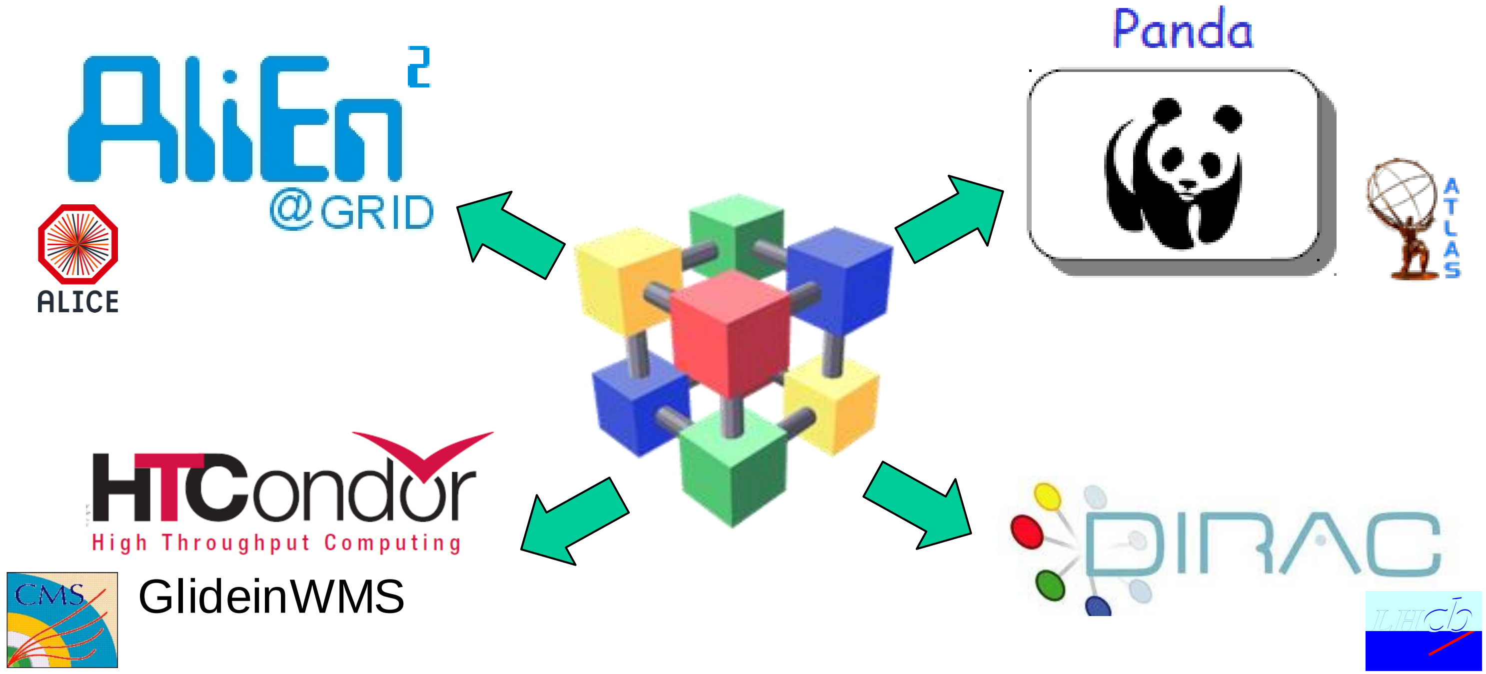
Distributed computing today





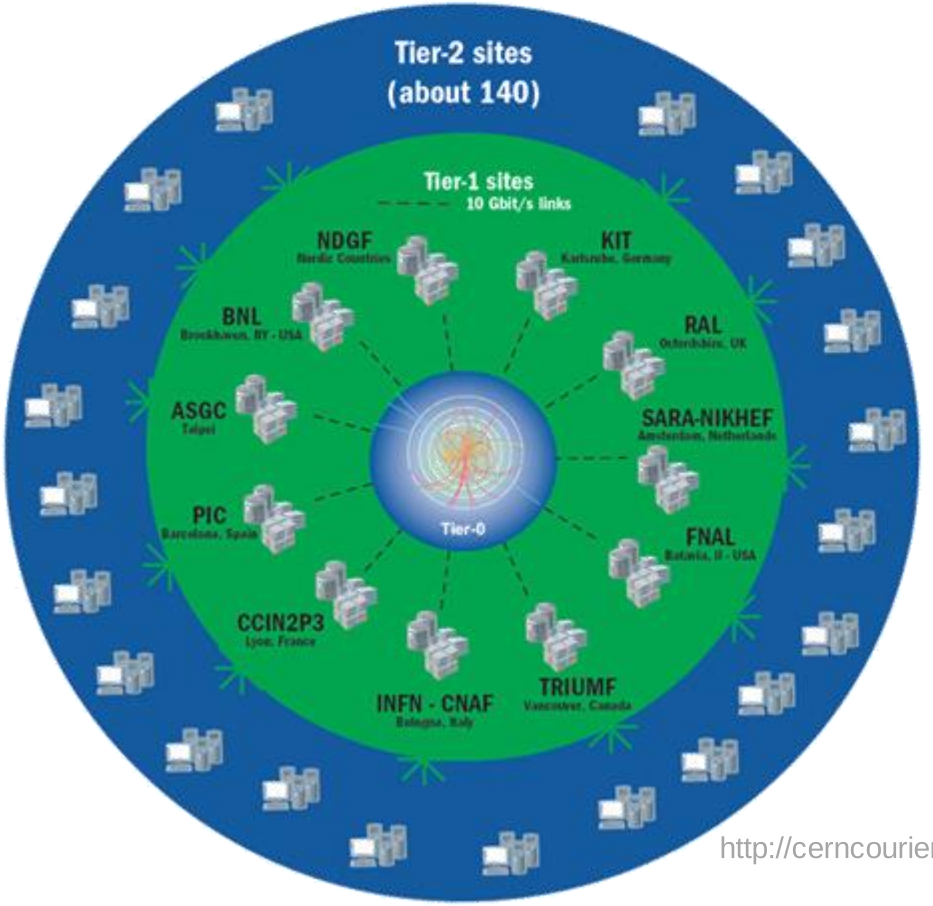
- Running millions of jobs and processing hundreds of PB of data
- Efficiency (CPU/Wall time) from 70% (analysis) to 85% (organized activities, reconstruction, simulation)
- 60-70% of all CPU time on the grid is spent in running simulation

Can it scale?



- Each experiment today runs their own Grid overlay on top of shared WLCG infrastructure
- In all cases the underlying distributed computing architecture is similar based on “pilot” job model
  - Can we scale these systems expected levels?
  - Lots of commonality and potential for consolidation

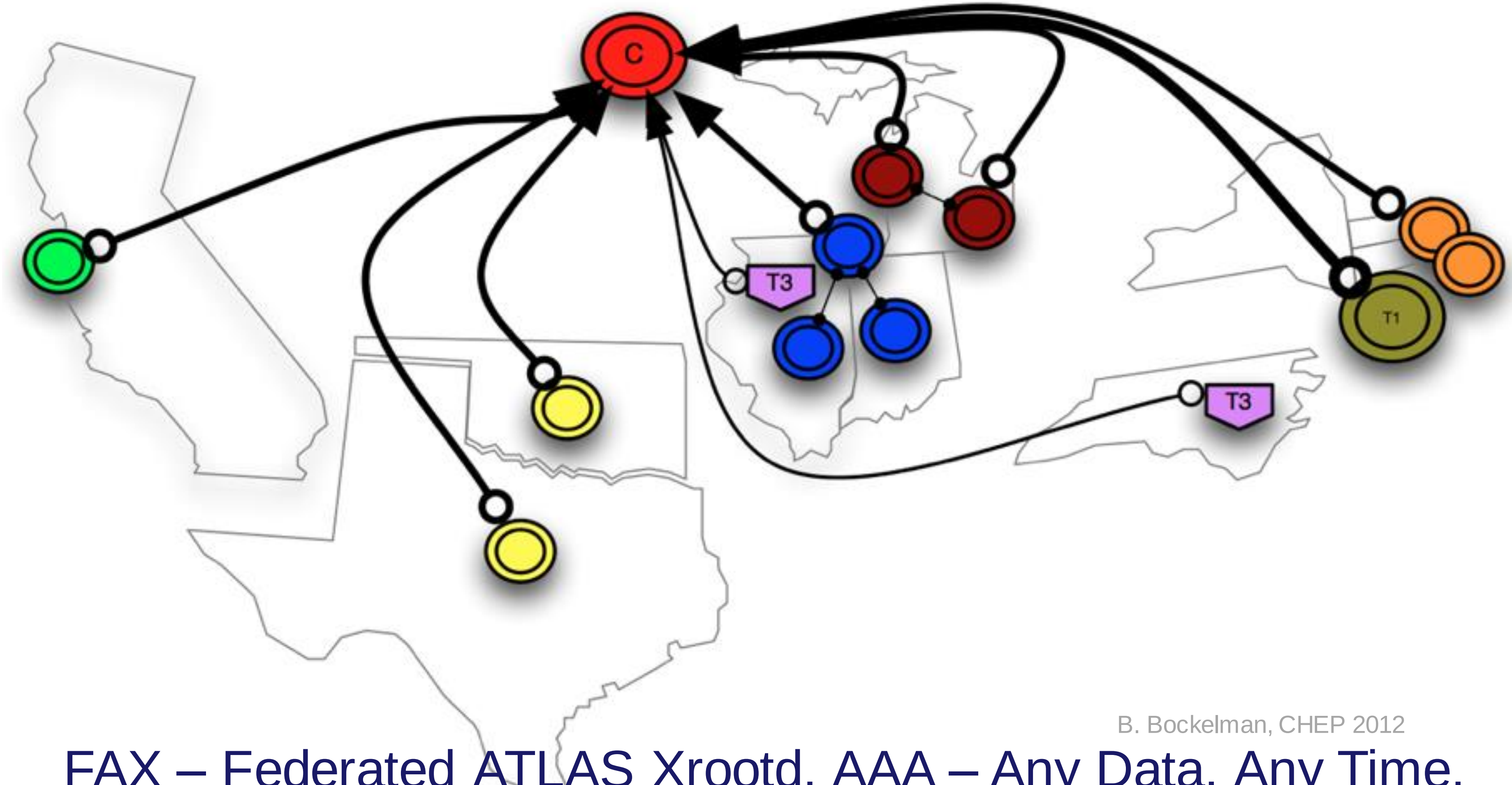
Grid Tier model



http://cerncourier.com/cws/article/cern/52744

- Network capabilities and data access technologies significantly improved our ability to use resources independent of location

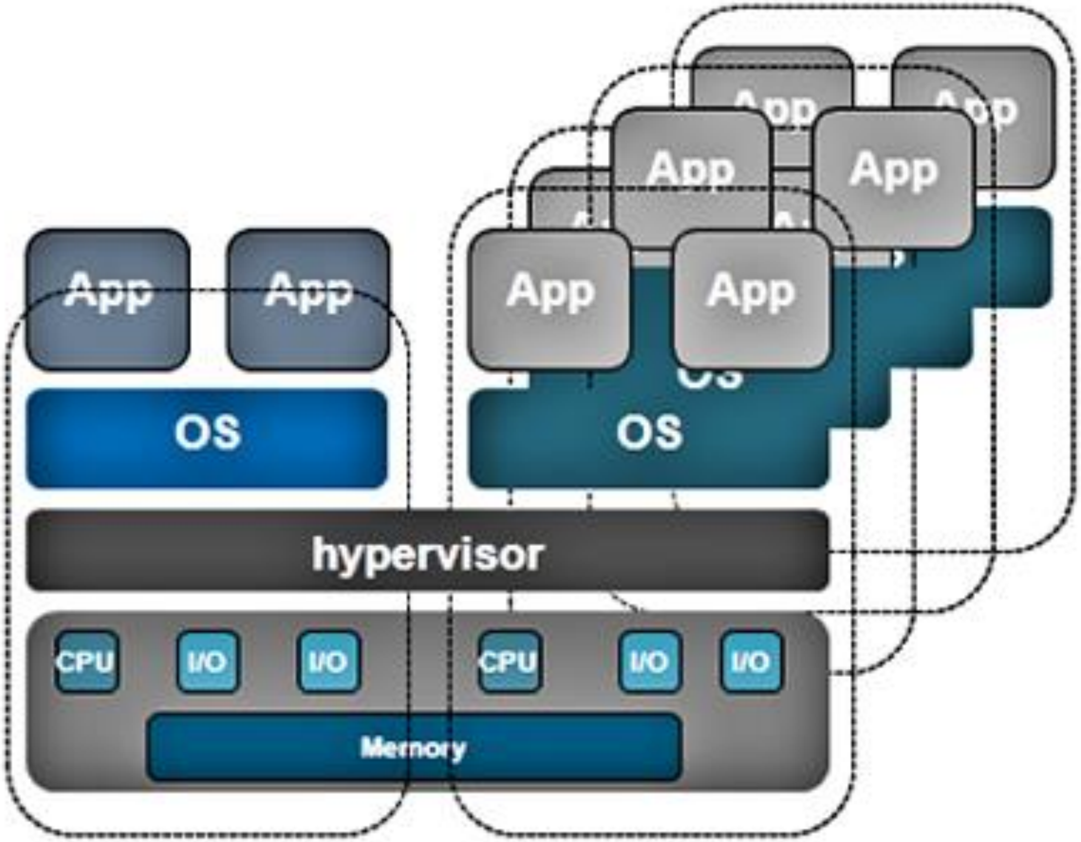
Global Data Federation



B. Bockelman, CHEP 2012

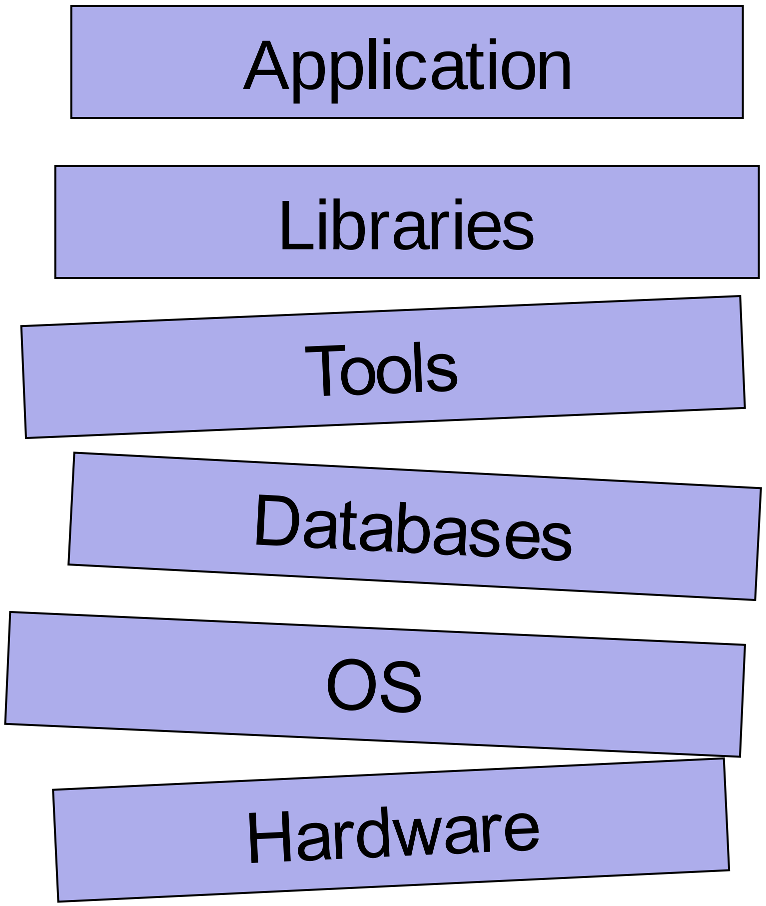
FAX – Federated ATLAS Xrootd, AAA – Any Data, Any Time, Anywhere (CMS), AliEn (ALICE)

Virtualization



- Virtualization provides better system utilization by putting all those many cores to work, resulting in power & cost savings.

Software integration problem



Traditional model

- Horizontal layers
- Independently developed
- Maintained by the different groups
- Different lifecycle

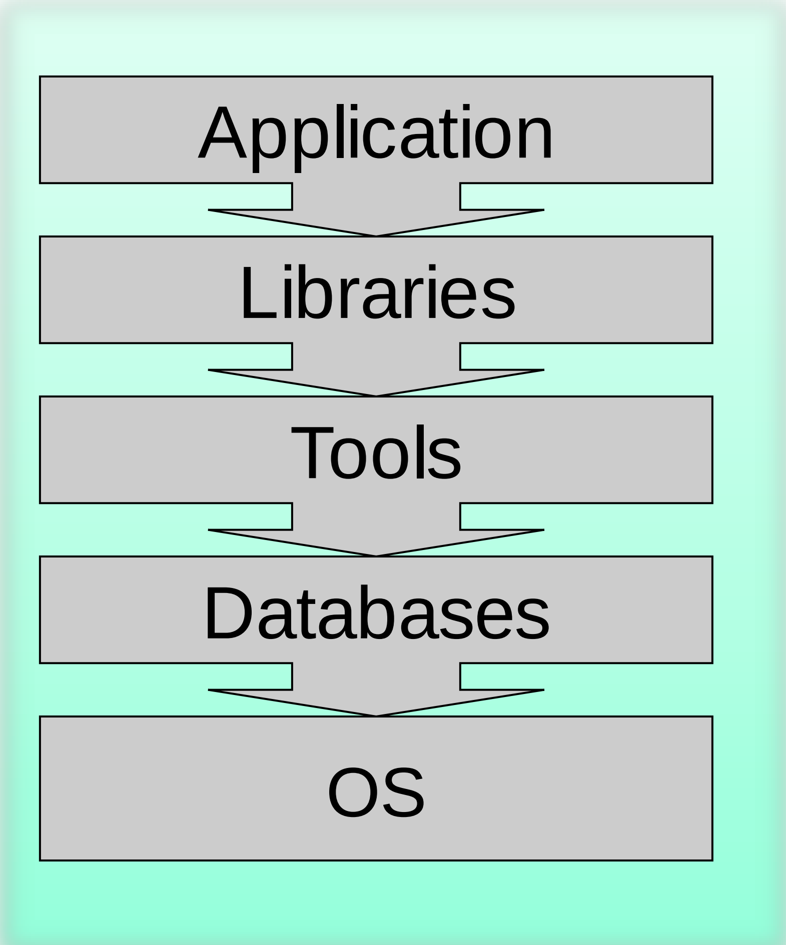
Application is deployed on top of the stack

- Breaks if any layer changes
- Needs to be certified every time when something changes
- Results in deployment and support nightmare

Difficult to do upgrades

- Even worse to switch to new OS versions

Decoupling Apps and Ops



Application driven approach

1. Start by analysing the application requirements and dependencies
2. Add required tools and libraries

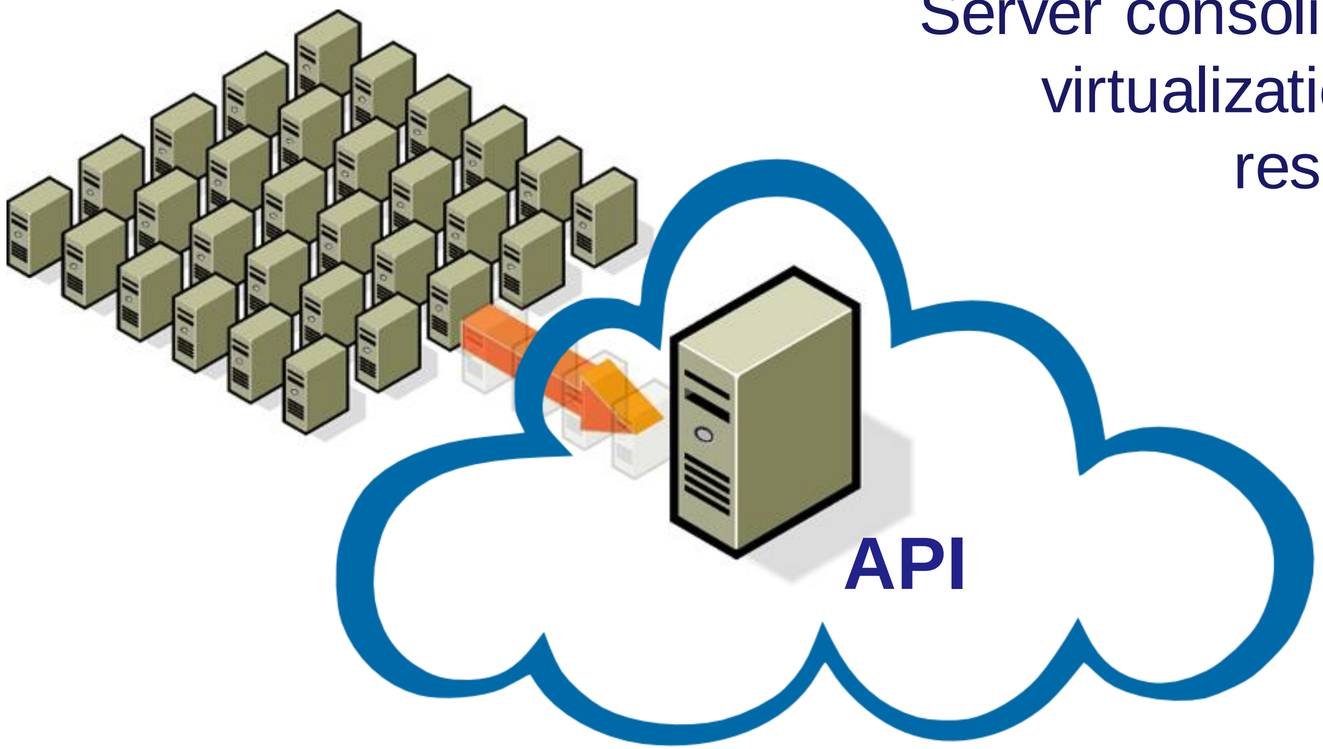
Use virtualization to

1. Build minimal OS
2. Bundle all this into Virtual Machine image

Separates lifecycles of the application and underlying computing infrastructure

Cloud: Putting it all together

Server consolidation using virtualization improves resource usage



IaaS = Infrastructure as a Service

- Public, on demand, pay as you go infrastructure with goals and capabilities similar to those of academic grids

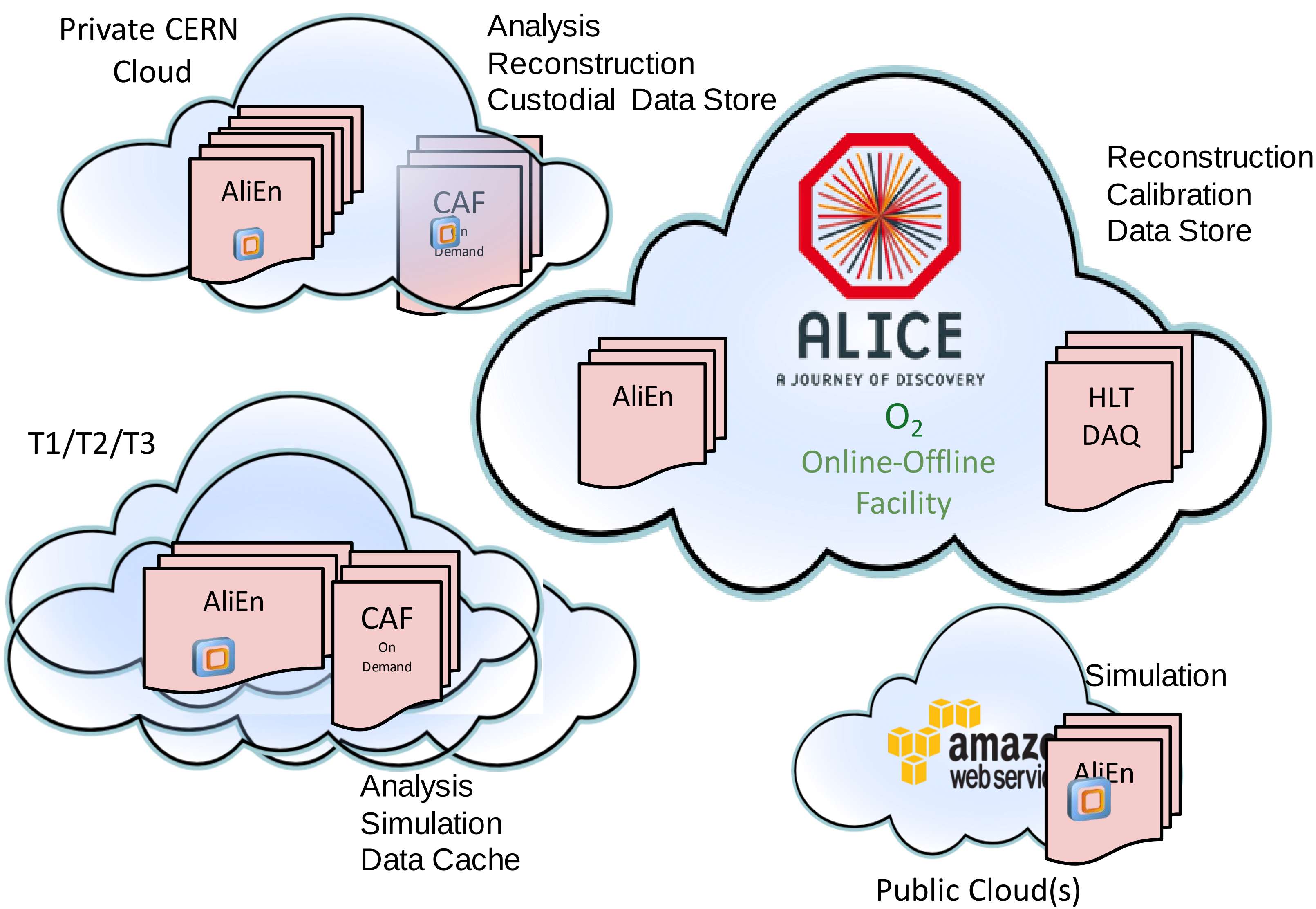
Public Cloud

- At present 9 large sites/zones
  - up to ~2M CPU cores/site, ~4M total
  - 10 x more cores on 1/10 of the sites compared to our Grid
  - 100 x more users (500,000)

Could we make our own Cloud?

- ✓ **Open Source Cloud components**
  - Open Source Cloud middleware
  - VM building tools and infrastructure
    - CernVM+CernVM/FS, boxgrinder..
  - Common API
- ✓ **From the Grid heritage…**
  - Common Authentication/Authorization
  - High performance global data federation/storage
  - Scheduled file transfer
  - Experiment distributed computing frameworks already adapted to Cloud
- **To do**
  - Unified access and cross Cloud scheduling
  - Centralized key services at a few large centers
  - Reduce complexity in terms of number of sites

Grid on top of Clouds



And if this is not enough…

- 18,688 nodes with total of 299,008 processor cores and one GPU per node. And, there are spare CPU cycles available…
- Ideal for event generators/simulation type workloads (60-70% of all CPU cycles used by the experiments).
- Simulation frameworks must be adapted to efficiently use such resources where available

Conclusions

- **Resources needed for Computing at HL-LHC are going to be large but not unprecedented**
  - Data volume is going to grow dramatically
  - Projected CPU needs are reaching the Moore's law ceiling
  - Grid growth of 25% per year is by far not sufficient
  - Speeding up the simulation is very important
  - Parallelization at all levels is needed for improving performance (application, I/O)
  - Requires reengineering of experiment software
- **New technologies such as clouds and virtualization may help to reduce the complexity and help to fully utilize available resources**