Ian Bird

WLCG Workshop, Copenhagen

12th November 2013

HEP computing futures

Topics

- Summary of computing model update
- Longer term – HL-LHC
- What should HEP computing look like in 10 years
  - How should we address the problem?

Computing Model update

- Requested by LHCC
  - Initial draft delivered in September; final version due for LHCC meeting in December
- Goals:
  - Optimise use of resources
  - Reduce operational costs (effort at grid sites, ease of deployment and operation, support and maintenance of grid middleware)
- Evolution of computing models – significant improvements that have already been done; areas of work now and anticipated; including several common projects
- Evolution of grid model: use of new technologies
  - Cloud/virtualisation
  - Data federations, intelligent data placement/caching, data popularity service

Contents:

- Outline:
  - Experiment computing models
    - Ongoing changes wrt original model – plans for the future
    - Structured to allow comparison across the experiments
  - Technology review
    - What is likely for CPU, disk, tape, network technologies; expected cost evolutions
  - Resource requirements during Run 2
  - Software performance
    - General considerations and experiment-specific actions; in particular optimising experiment software – what has already been done, what is anticipated
  - Evolution of the grid and data management services
    - Aim to reduce costs of operation and support

Computing models

- Focus on use of resources/capabilities rather than "Tier roles"
  - Already happening: LHCb use of Tier 2s for analysis, CMS use for MC reconstruction; use of Tier 1s for prompt reconstruction, etc
  - Data access peer-peer: removal of hierarchical structure
- Data federations – based on xrootd – many commonalities
  - Optimizing data access from jobs: remote access, remote I/O
  - More intelligent data placement/caching; pre-placement vs dynamic caching
  - Data popularity services being introduced
- Reviews of (re-)processing passes; numbers of data replicas
- Use of HLT and opportunistic resources now important

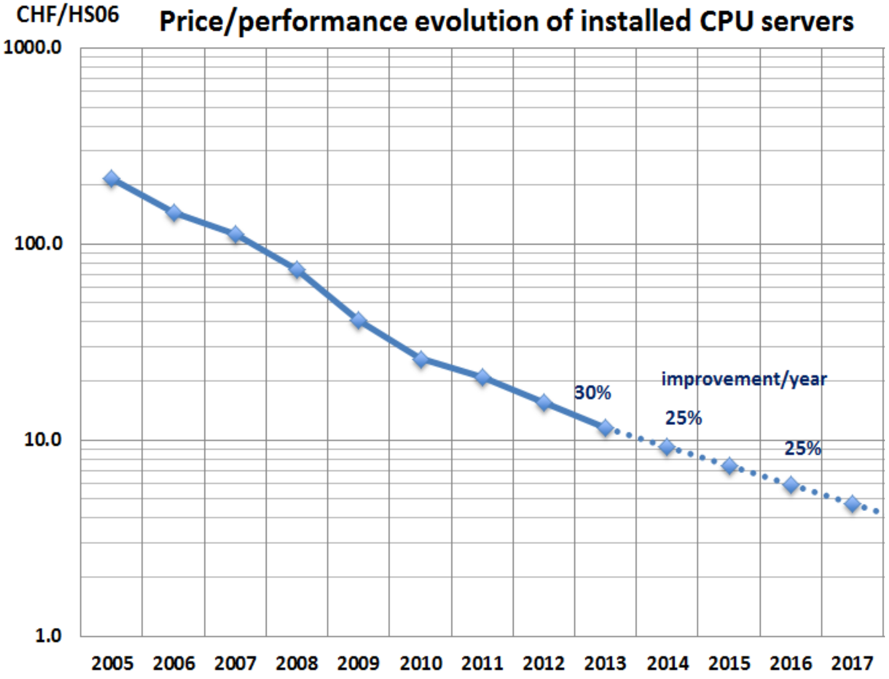
Software

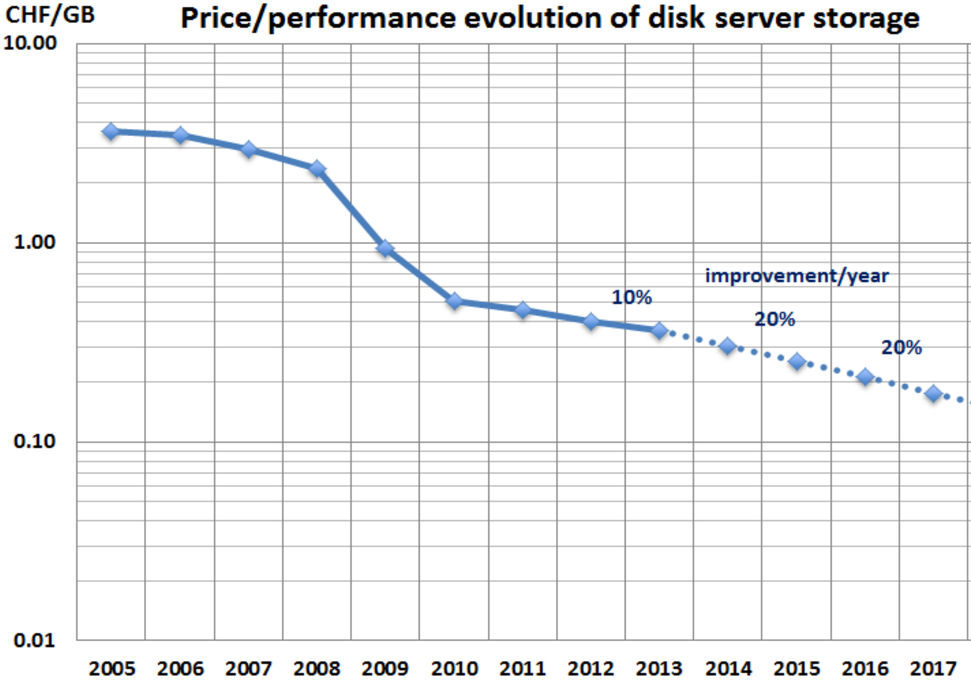
- Moore’s law only helps us if we can make use of the new multi-core CPUs with specialised accelerators etc. (Vectorisation, GPUs, …)
  - No longer benefit from simple increases in clock speed
- Ultimately this requires HEP software to be re-engineered to make use of parallelism at all levels
  - Vectors, instruction pipelining, instruction level pipelining, hardware threading, multi-core, multi-socket.
- Need to focus on commonalities:
  - GEANT, ROOT, build up common libraries
- This requires significant effort and investment in the HEP community
  - Concurrency forum already initiated
  - Ideas to strengthen this as a collaboration to provide roadmap and incorporate & credit additional effort

Distributed computing

- Drivers:
  - Operational cost of grid sites
  - Ability to easily use opportunistic resources (commercial clouds, HPC, clusters, …) with ~zero configuration
  - Maintenance cost of grid middleware
- Simplifying grid middleware layer
  - Complexity has moved to the application layer where it better fits
    - Ubiquitous use of pilot jobs, etc.
  - Cloud technologies give a way to implement job submission and management
  - Run 2 will see a migration to more cloud-like model
  - Centralisation of key grid services – already happening
  - Leading to more lightweight and robust implementation of distributed computing

Technology outlook





- *Effective* yearly growth: CPU 20%, Disk 15%, Tape 15%
- Assumes:
  - 75% budget additional capacity, 25% replacement
  - Other factors: infrastructure, network & increasing power costs

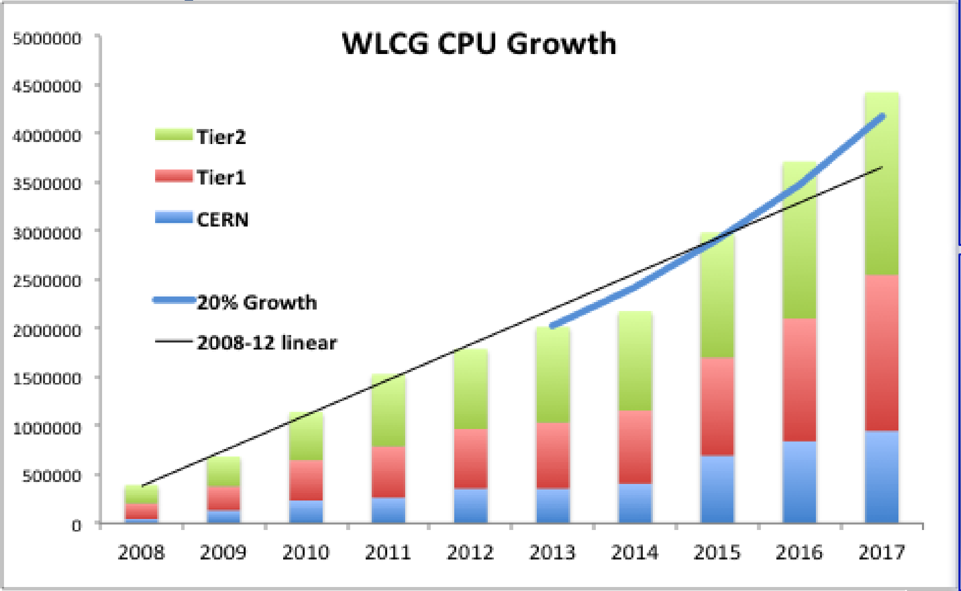
Evolution of requirements

Higher trigger (data) rates driven by physics needs

Based on understanding of likely LHC parameters;

Foreseen technology evolution (CPU, disk, tape)

Experiments work hard to fit within constant budget scenario

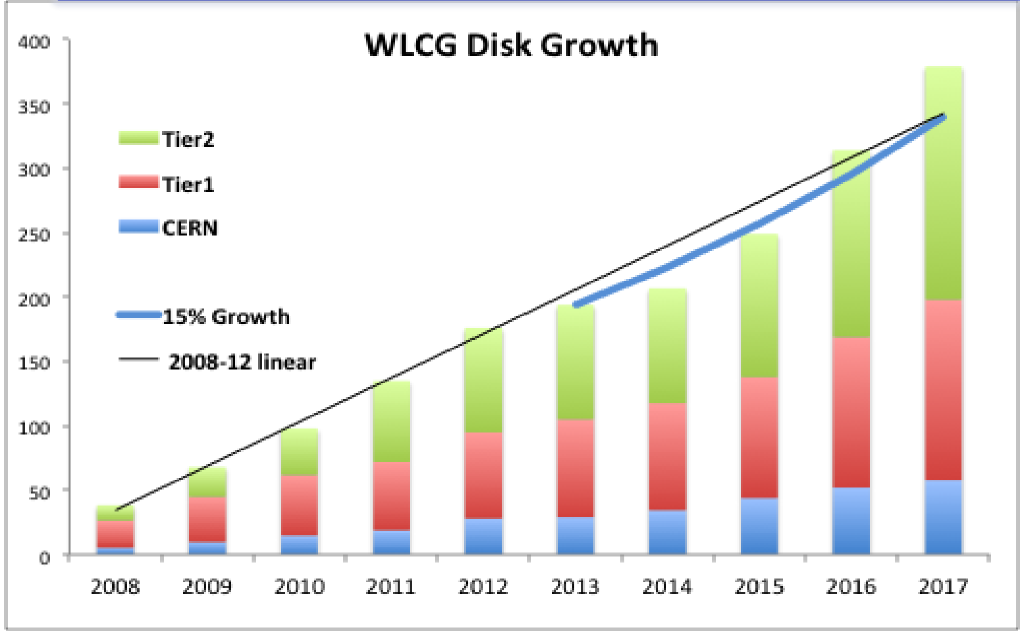


Estimated evolution of requirements 2015-2017 (**NB. Does not reflect outcome of current RSG scrutiny**)

2008-2013: Actual deployed capacity

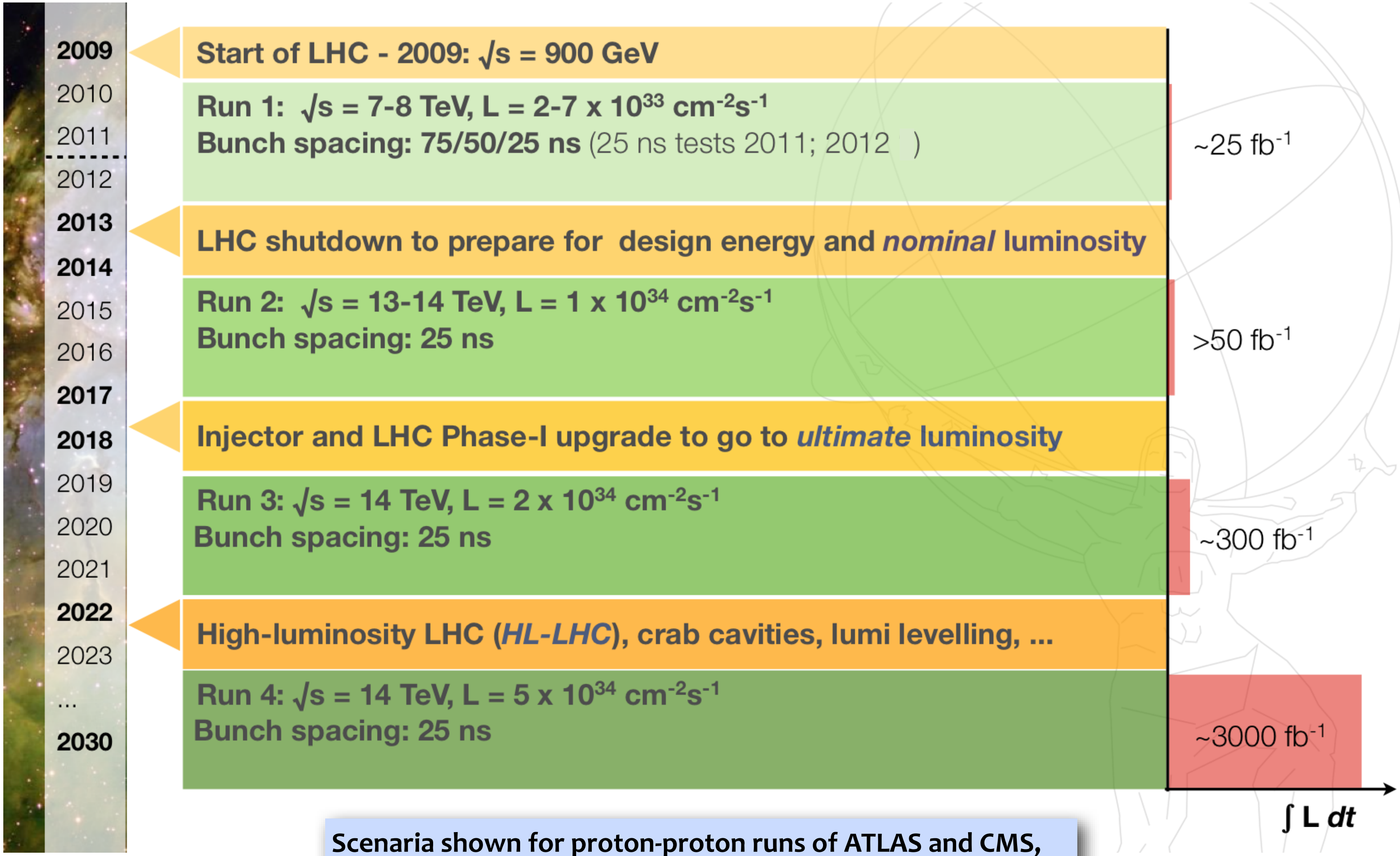
Line: extrapolation of 2008-2012 actual resources

Curves: expected potential growth of technology with a constant budget (see next)
 CPU: 20% yearly growth
 Disk: 15% yearly growth



Longer term?

A lot more to come …



2009 | Start of LHC - 2009: $\sqrt{s}$ = 900 GeV

2010, 2011, 2012 | Run 1: $\sqrt{s}$ = 7-8 TeV, L = 2-7 x 10^{33} $cm^{-2}s^{-1}$
Bunch spacing: 75/50/25 ns (25 ns tests 2011; 2012) | ~25 fb^{-1}

2013, 2014 | LHC shutdown to prepare for design energy and *nominal* luminosity

2015, 2016 | Run 2: $\sqrt{s}$ = 13-14 TeV, L = 1 x 10^{34} $cm^{-2}s^{-1}$
Bunch spacing: 25 ns | >50 fb^{-1}

2017, 2018 | Injector and LHC Phase-I upgrade to go to *ultimate* luminosity

2019, 2020, 2021 | Run 3: $\sqrt{s}$ = 14 TeV, L = 2 x 10^{34} $cm^{-2}s^{-1}$
Bunch spacing: 25 ns | ~300 fb^{-1}

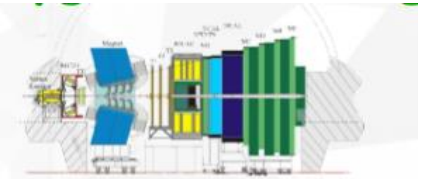
2022 | High-luminosity LHC (*HL-LHC*), crab cavities, lumi levelling, ...

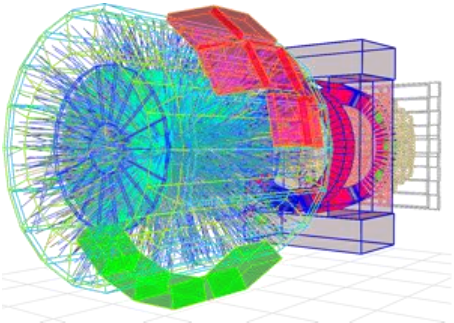
2023, …, 2030 | Run 4: $\sqrt{s}$ = 14 TeV, L = 5 x 10^{34} $cm^{-2}s^{-1}$
Bunch spacing: 25 ns | ~3000 fb^{-1}

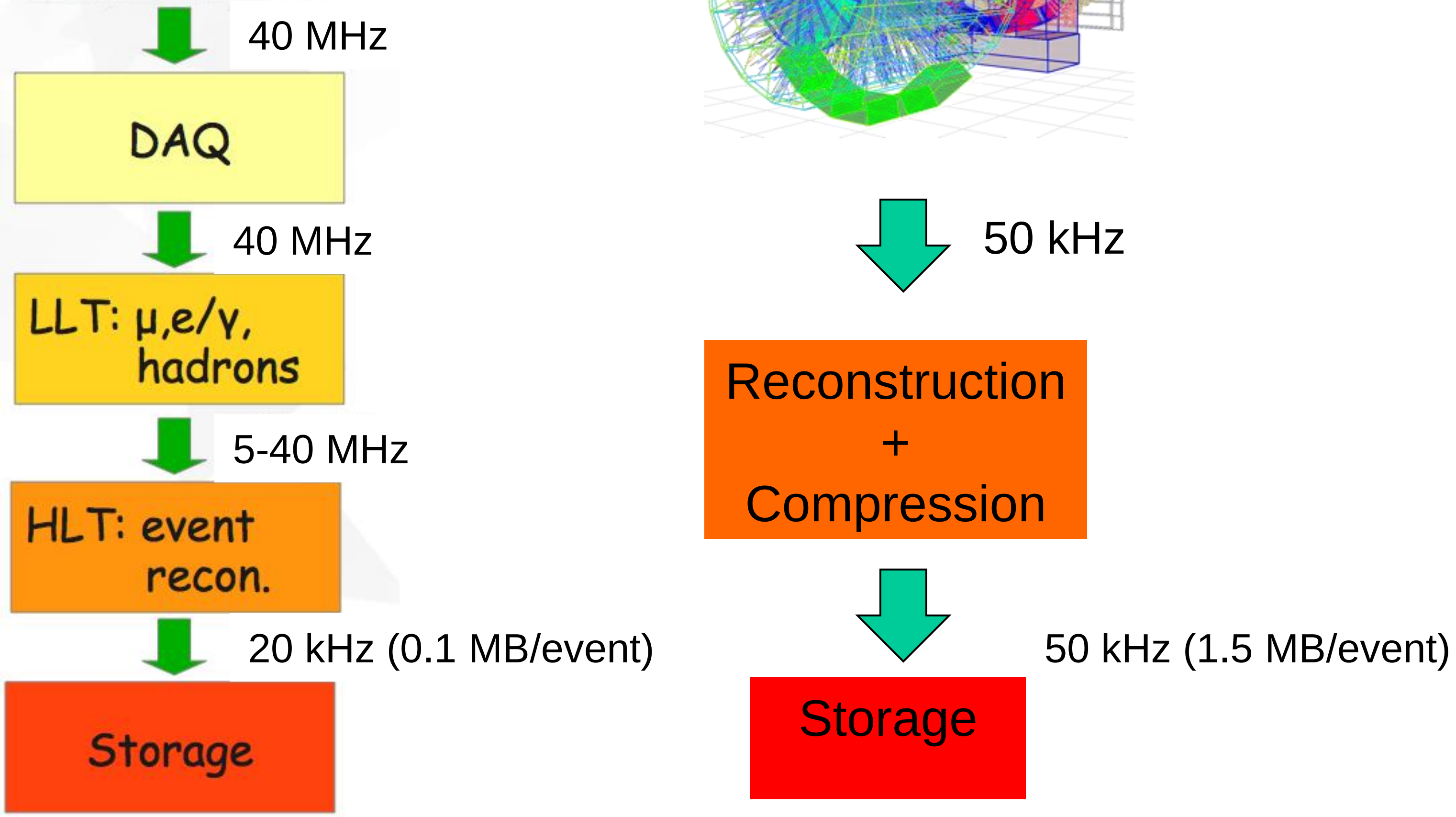
$\int L\, dt$

Scenaria shown for proton-proton runs of ATLAS and CMS, LHCb and Alice follow different strategies.

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

50 kHz

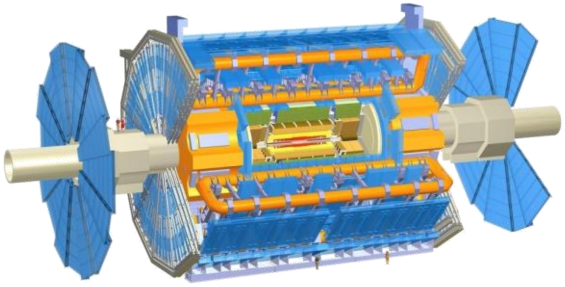
Reconstruction + Compression

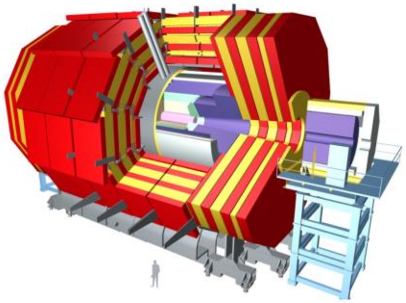
50 kHz (1.5 MB/event)

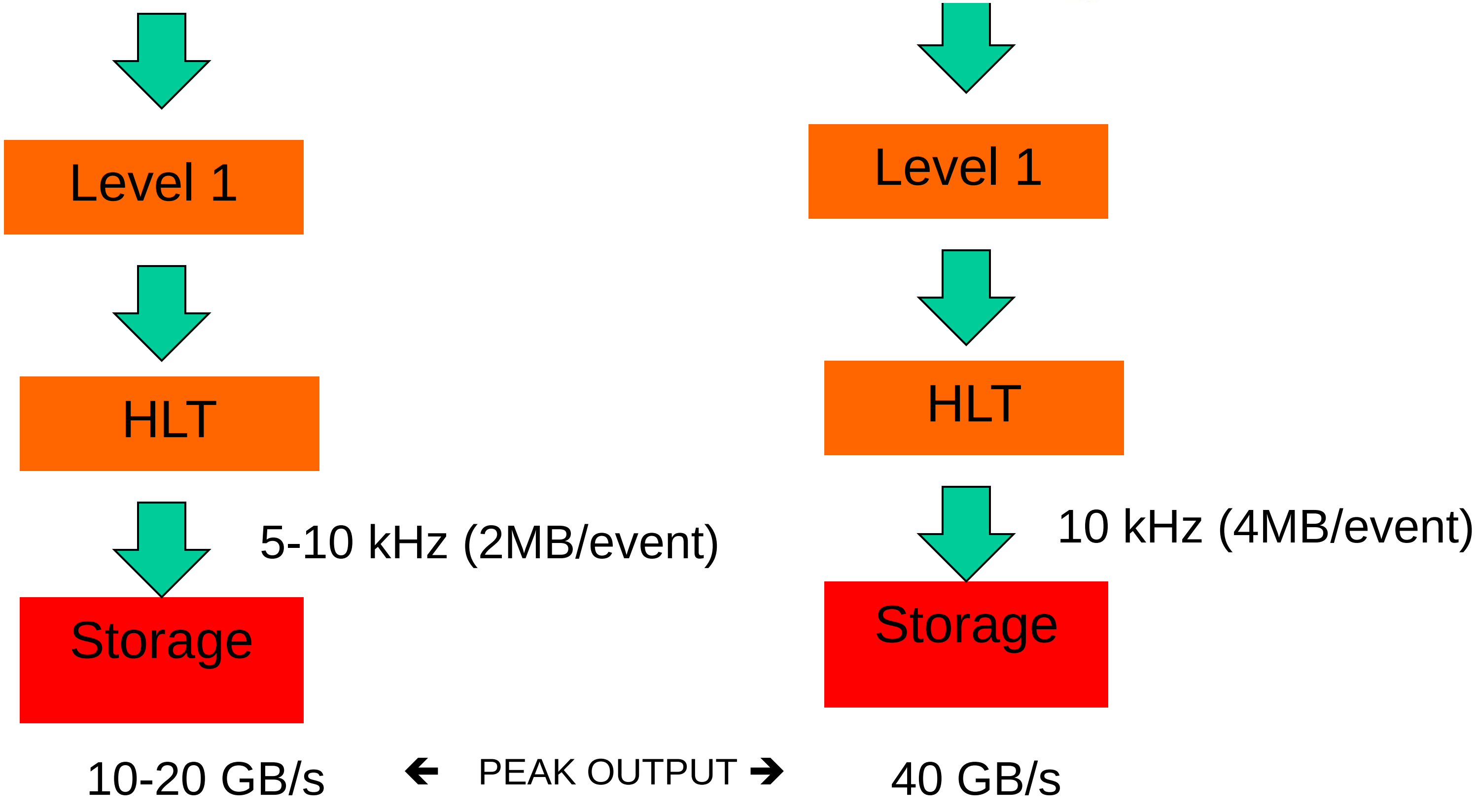
Storage

2 GB/s ← PEAK OUTPUT → 75 GB/s

ATLAS & CMS @ Run 4







Level 1

HLT

5-10 kHz (2MB/event)

Storage

10-20 GB/s

Level 1

HLT

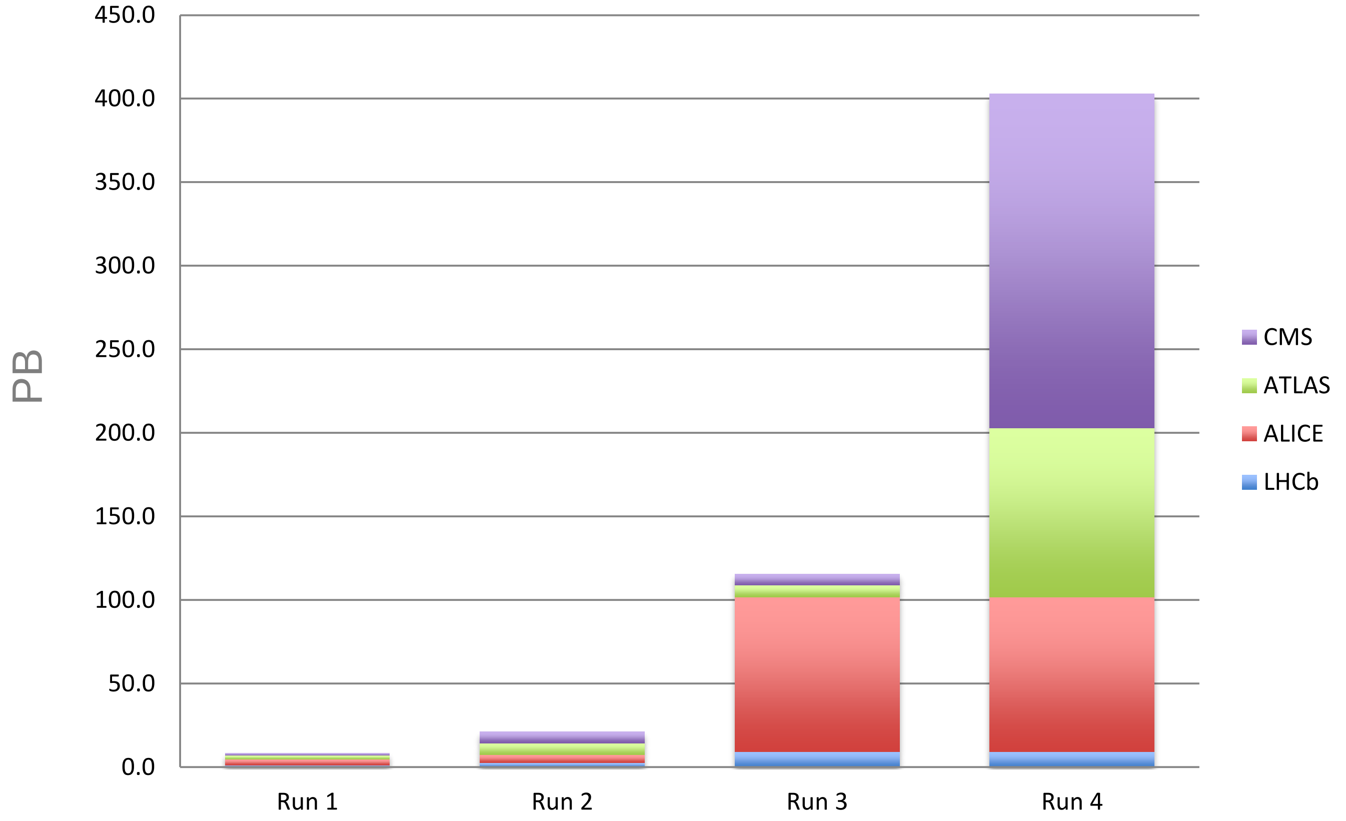
10 kHz (4MB/event)

Storage

40 GB/s

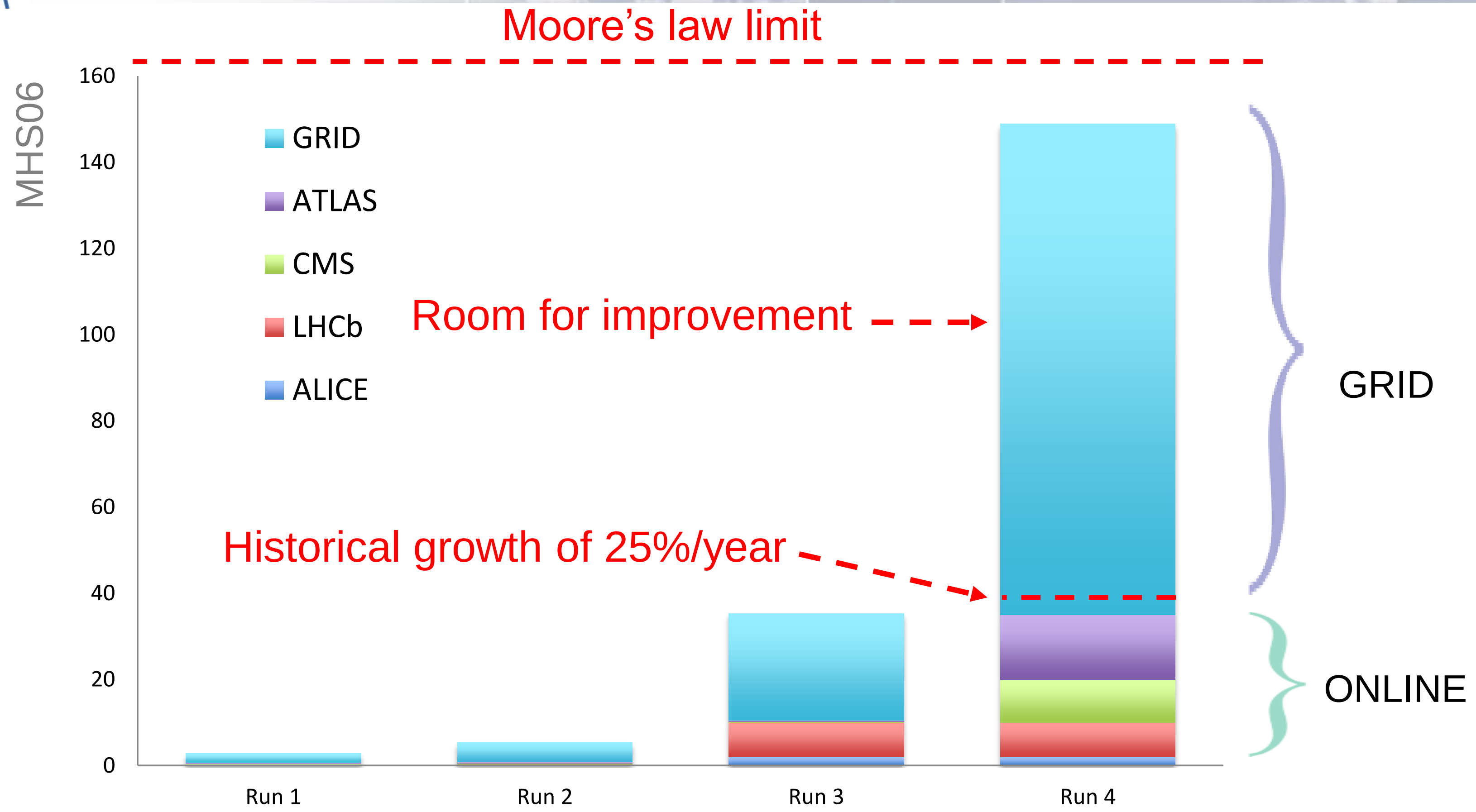
← PEAK OUTPUT →

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…

CPU: Online + Offline



Moore’s law limit

Room for improvement

Historical growth of 25%/year

GRID

ONLINE

- 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**.

Summary

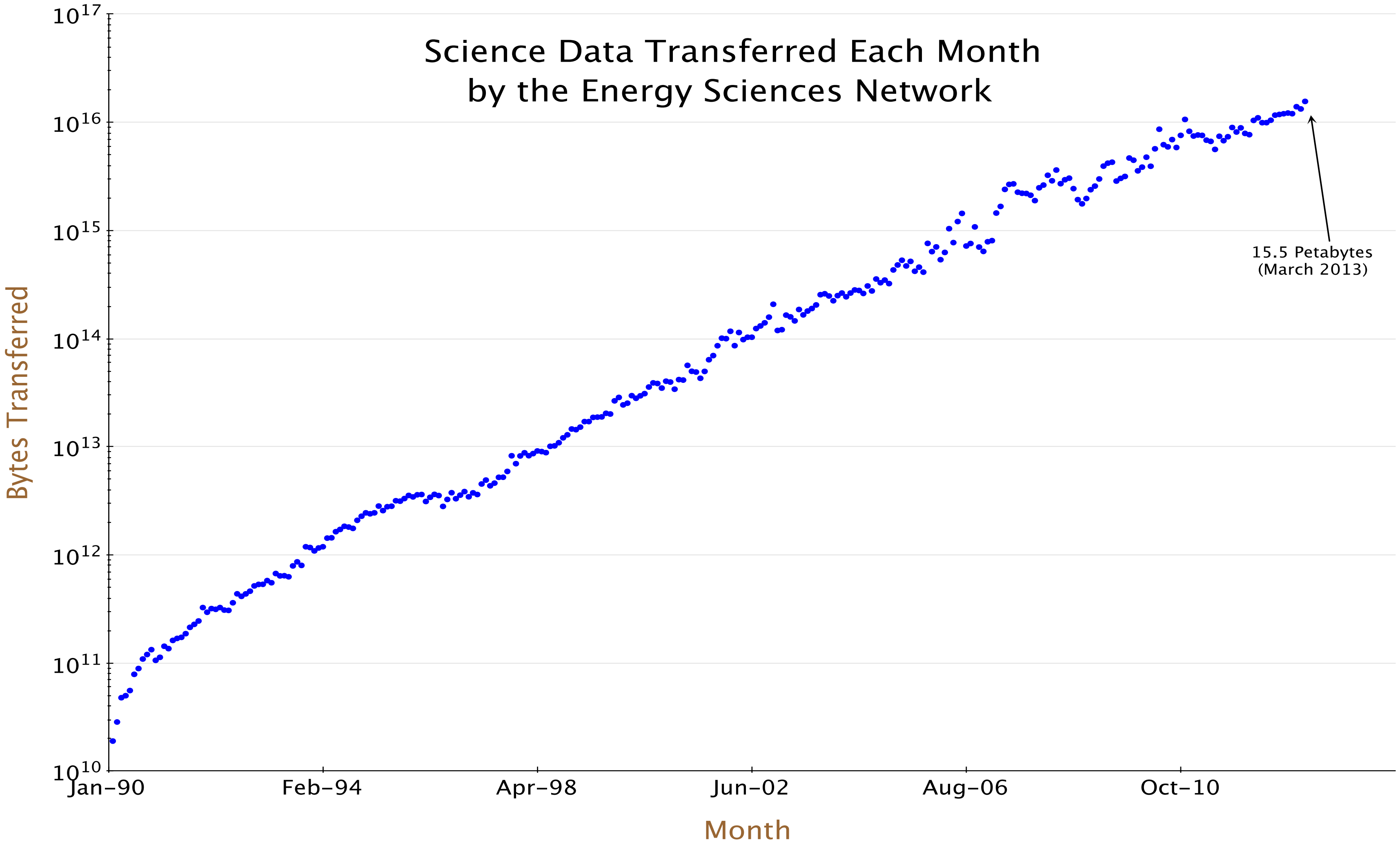
- ❑ HL-LHC : high pile-up and high read-out rate

➔ large increase of processing needs

- ❑ With flat resource (in euros), and even with Moore's law holding true (likely, provided we maintain/improve efficient use of processors), this is not enough (by ½ to one order of magnitude)

➔large software improvement needed

- ❑ Future evolution of processors: many cores with less memory per core, more sophisticated processors instructions (micro-parallelism), possibility of specialised cores➔
  - o Optimisation of software to use high level processors instructions, especially in identified hot spots (expert task)
  - o Parallel framework to distribute algorithms to cores, in a semi-transparent way to regular physicist software developer
- ❑ LHC experiments code base more than 15 millions of line of code, written by more than 3000 people➔a whole community to engage, starting essentially now, new blood to inject
- ❑ We are sharing already effort and software. We can do much more: concurrency forum http://concurrency.web.cern.ch



2023: expect 10 Tb/s networks

Network access to facilities and data will be cheap
Moving data around is expensive (needs disk!)

Problems

- No economies of scale (ops costs);
  - 10 large centres much better than 150 smaller
- Too distributed – too much disk cache needed
- Current inability to effectively use CPU
  - Evolution of commodity or HPC architectures, Break down of Moore’s law (physics)

Opportunities

- Fantastic networking – as much as you want
- No reason at all to have data locally to physicist
- Much more “offline” goes “inline” – don’t store everything
  - HLT farms will significantly increase in size – why not carry this further?
- Change in funding models needed

Long term ?

- Current models do not simply scale – need to re-think
- What is the most cost-effective way to deploy computing?
- Proposing to hold series of workshops to brainstorm radical computing model changes for the 10-year timescale.
  - How can we benefit from economies of scale?
  - How does HEP collaborate with other sciences (big-data, e-infrastructures, etc)

HEP computing in 10 years?

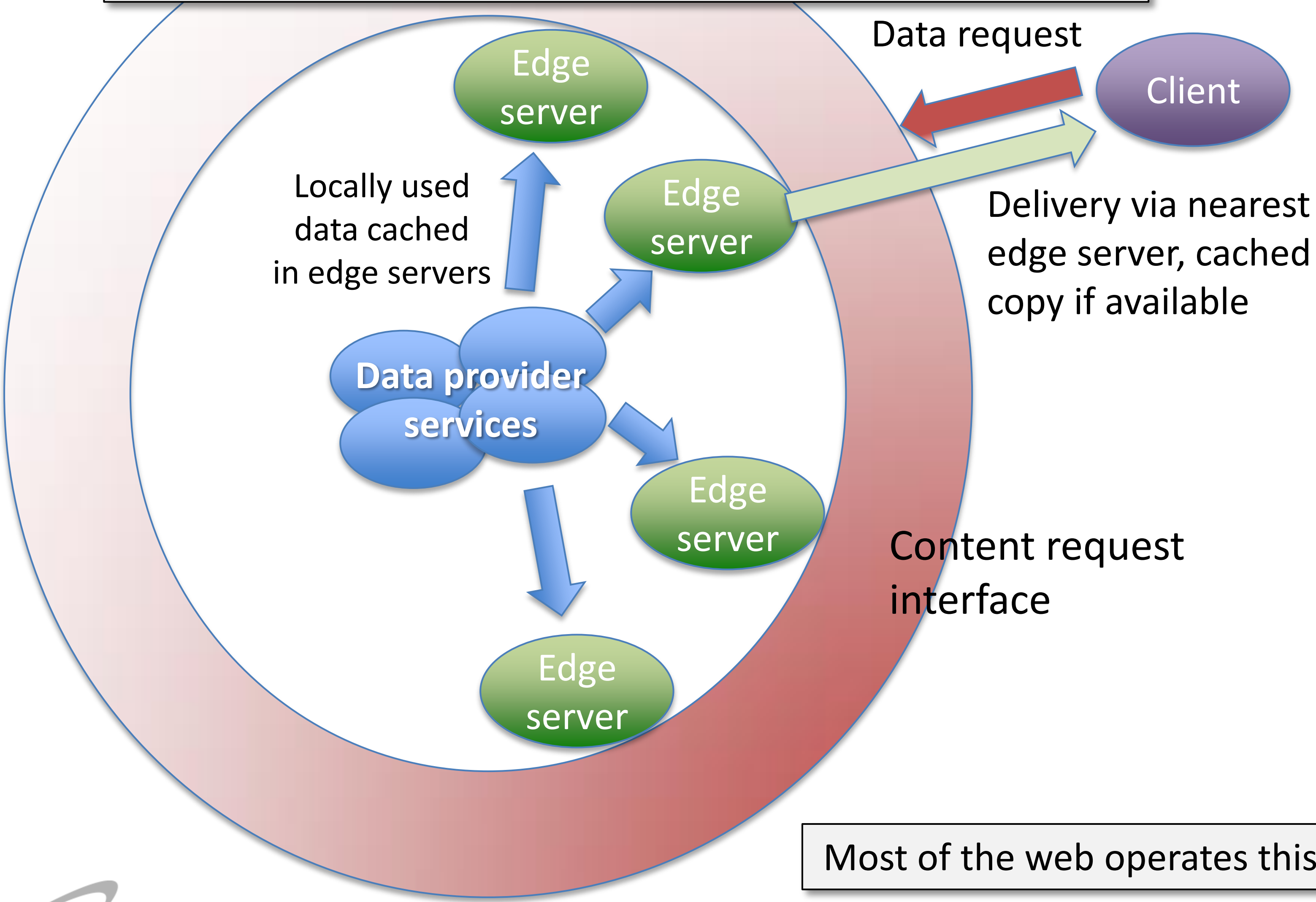
- We still use the computing model of 1970’s
- Opportunity to really re-think how we produce “science data”
  - And how physicists can use or query it
- Opportunity to (re-)build true commonalities
  - Within HEP, and with other science, and other big-data communities

Long Term strategy

- HEP computing needs a forum where these strategic issues can be coordinated since they impact the entire community:
  - Build on leadership in large scale data management & distributed computing – make our experience relevant to other sciences – generate long term collaborations and retain expertise
  - Scope and implementation of long term e-infrastructures for HEP – relationship with other sciences and funding agencies
  - Data preservation & reuse, open and public access to HEP data
  - Significant investment in software to address rapidly evolving computer architectures is necessary
  - HEP must carefully choose where to invest our (small) development effort – high added value in-house components, while making use of open source or commercial components where possible
  - HEP collaboration on these and other key topics with other sciences and industry

The Content Delivery Network Model

Content delivery network: deliver data quickly and efficiently by placing data of interest close to its clients



Data request

Client

Edge server

Locally used data cached in edge servers

Edge server

Delivery via nearest edge server, cached copy if available

Data provider services

Edge server

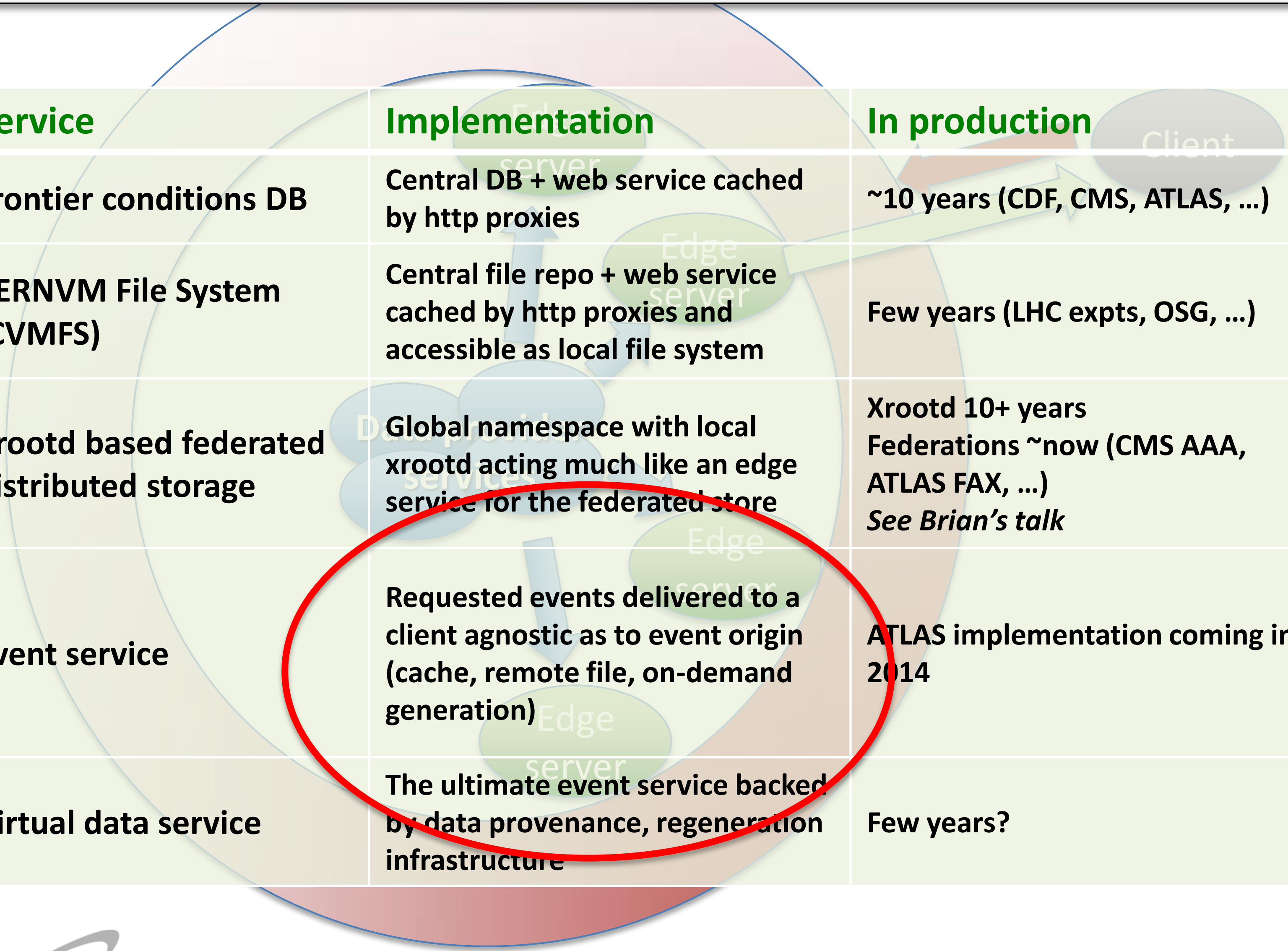
Content request interface

Edge server

Most of the web operates this way

The Content Delivery Network Model

A growing number of HEP services are designed to operate broadly on the CDN model



Service	Implementation	In production
Frontier conditions DB	Central DB + web service cached by http proxies	~10 years (CDF, CMS, ATLAS, …)
CERNVM File System (CVMFS)	Central file repo + web service cached by http proxies and accessible as local file system	Few years (LHC expts, OSG, …)
Xrootd based federated distributed storage	Global namespace with local xrootd acting much like an edge service for the federated store	Xrootd 10+ years Federations ~now (CMS AAA, ATLAS FAX, …) *See Brian's talk*
Event service	Requested events delivered to a client agnostic as to event origin (cache, remote file, on-demand generation)	ATLAS implementation coming in 2014
Virtual data service	The ultimate event service backed by data provenance, regeneration infrastructure	Few years?

What might this look like?

- Inside the CDN “torus”
  - Large scale data factories – consolidation of Tier 1s and large Tier 2s;
  - Function to deliver the datasets requested
  - No need to be transferring data around – essentially scale the storage to the CPU capacity
    - Connected by v. high speed networks
- Distinction between “online” and “offline” could move to this boundary at the client interface

- At this point can think about new models of analysing data
  - Query data set rather than event-loop style?