L’evoluzione dei modelli di calcolo a LHC

Concezio Bozzi, INFN Ferrara

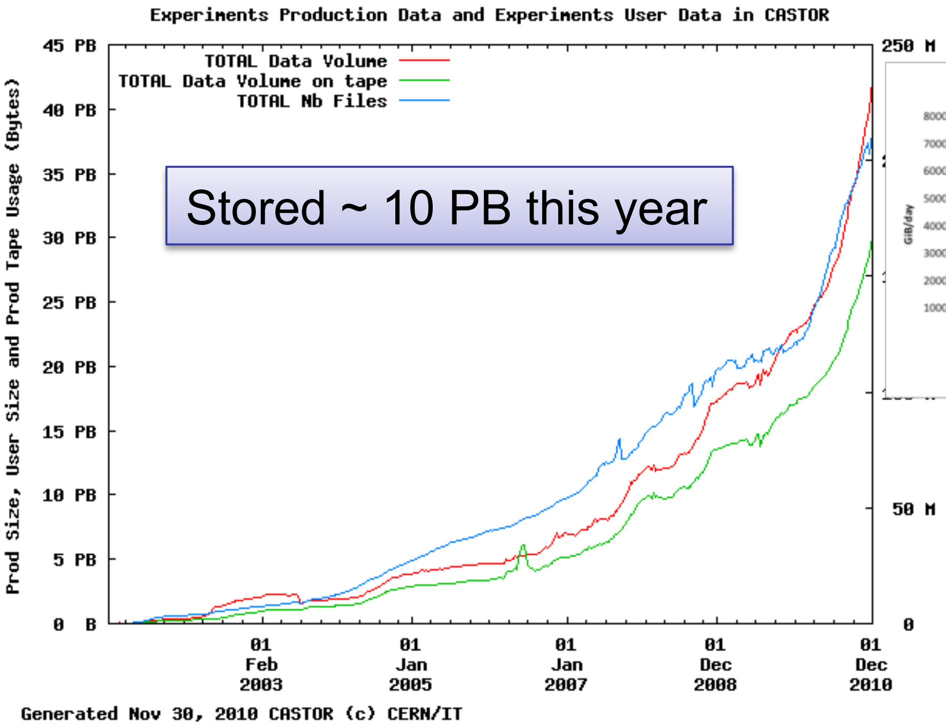
Workshop CCR su stato e prospettive del calcolo scientifico

Legnaro, 17 febbraio 2011

gratefully acknowledging

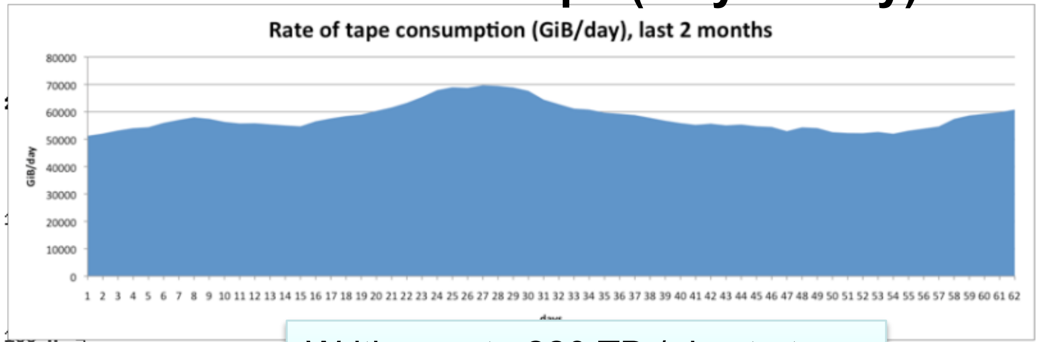
I. Bird, H. Newman, I. Fisk and R. Jones

Data did not fall on the floor

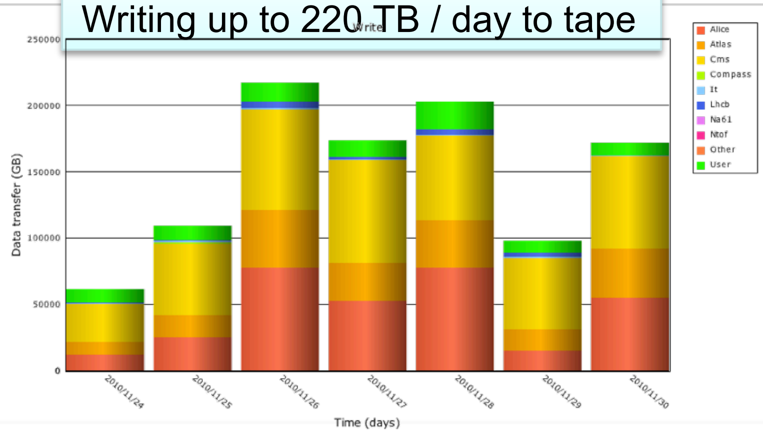


Stored ~ 10 PB this year

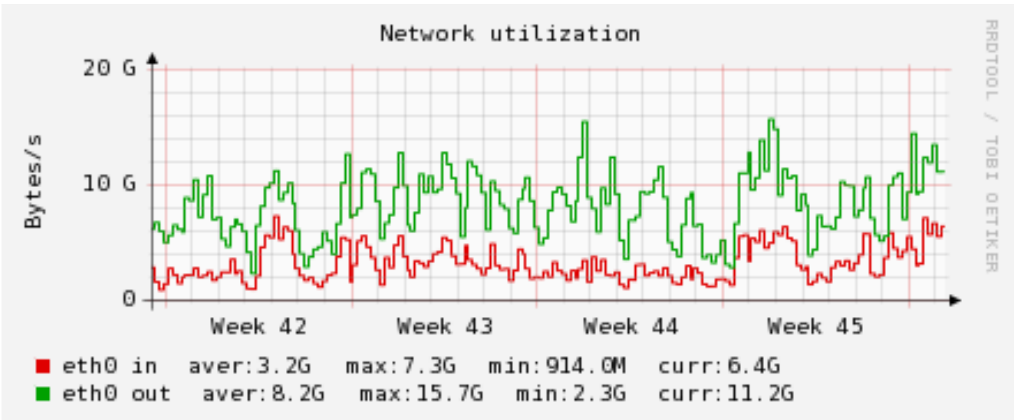
Data written to tape (Gbytes/day)



Writing up to 220 TB / day to tape



Disk Servers (Gbytes/s)

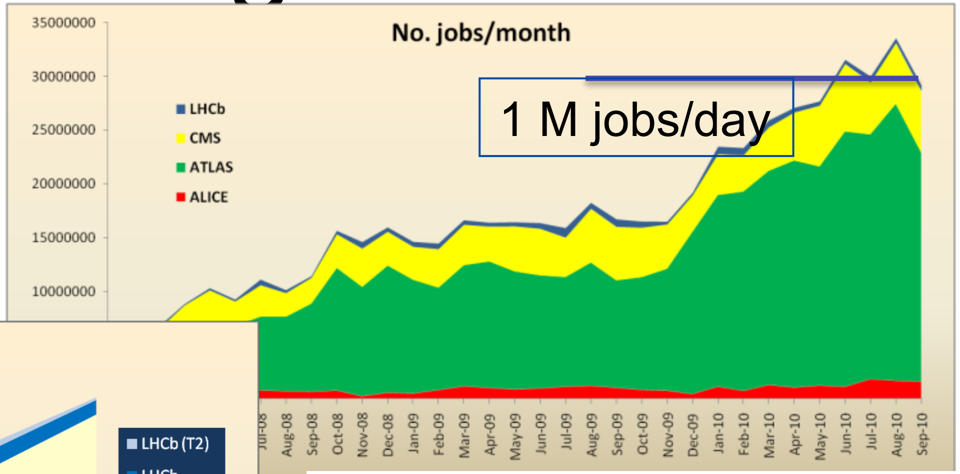


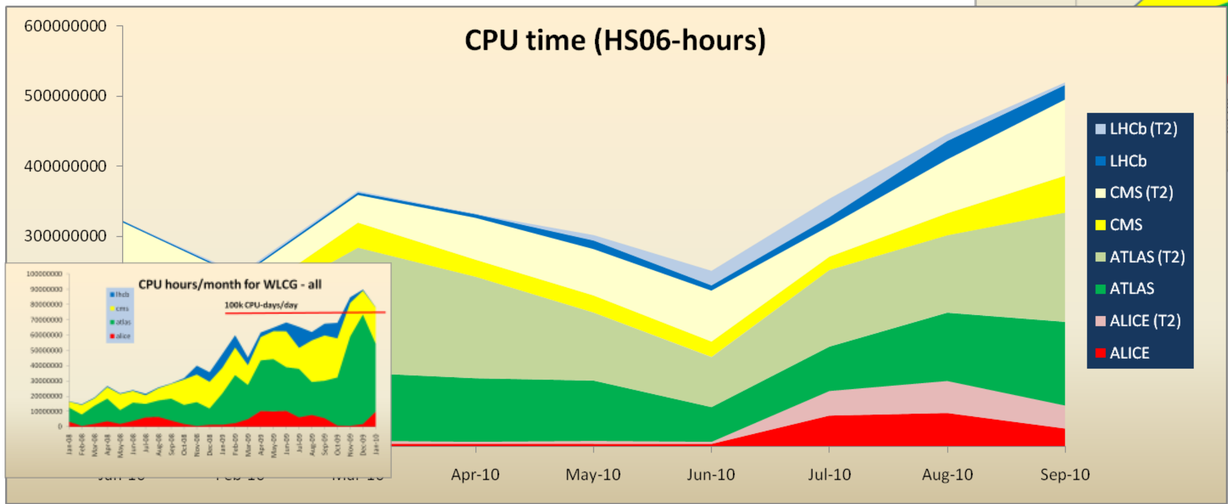
Tier 0 storage:

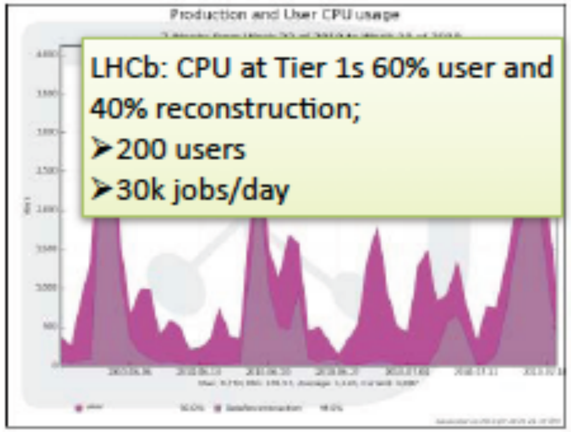
- Accepts data at average of 2.6 GB/s; peaks > 7 GB/s
- Serves data at average of 7 GB/s; peaks > 18 GB/s
- **CERN Tier 0 moves ~ 1 PB data per day**

WLCG Usage

- Use remains consistently high
  - 1 M jobs/day; >>100k CPU-days/day
  - Actually much more inside pilot jobs

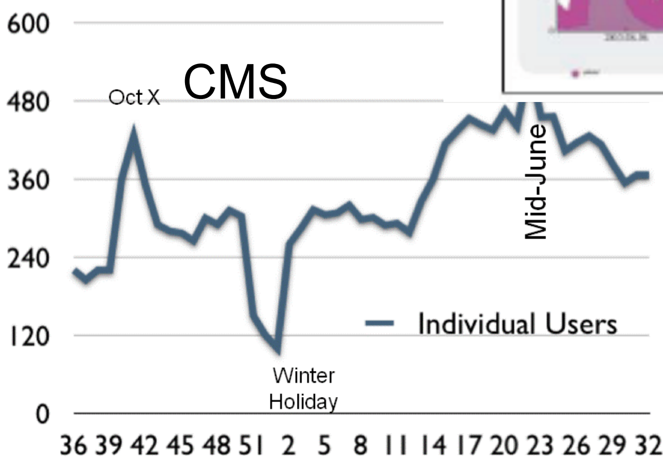






As well as LHC data, large simulation productions ongoing

ALICE: ~200 users, 5-10% of Grid resources

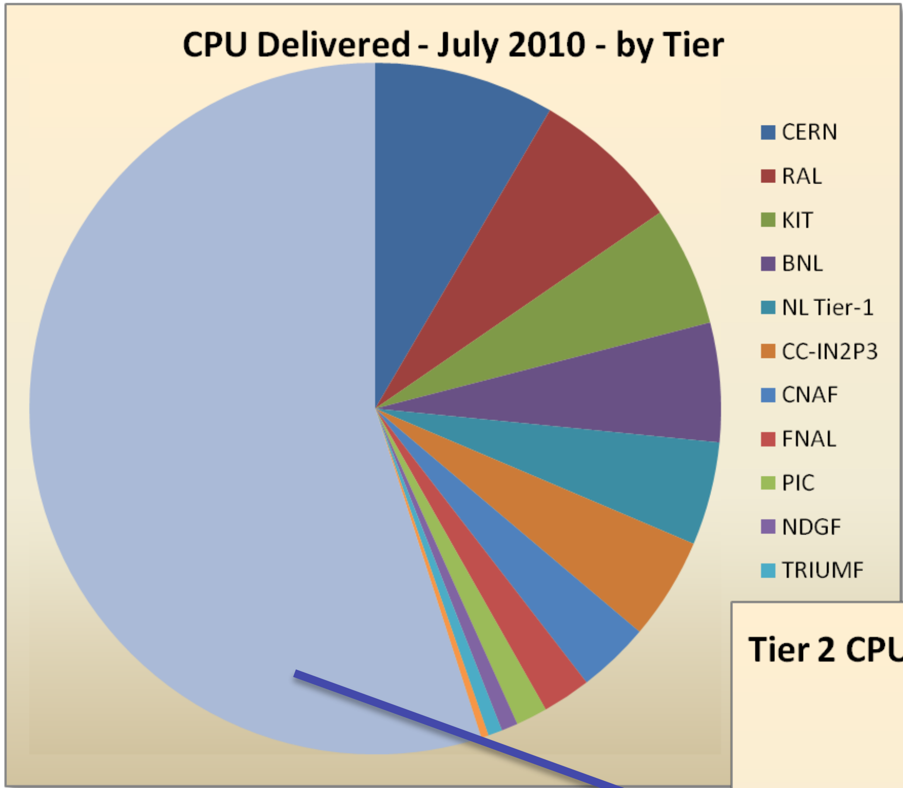


Large numbers of analysis users

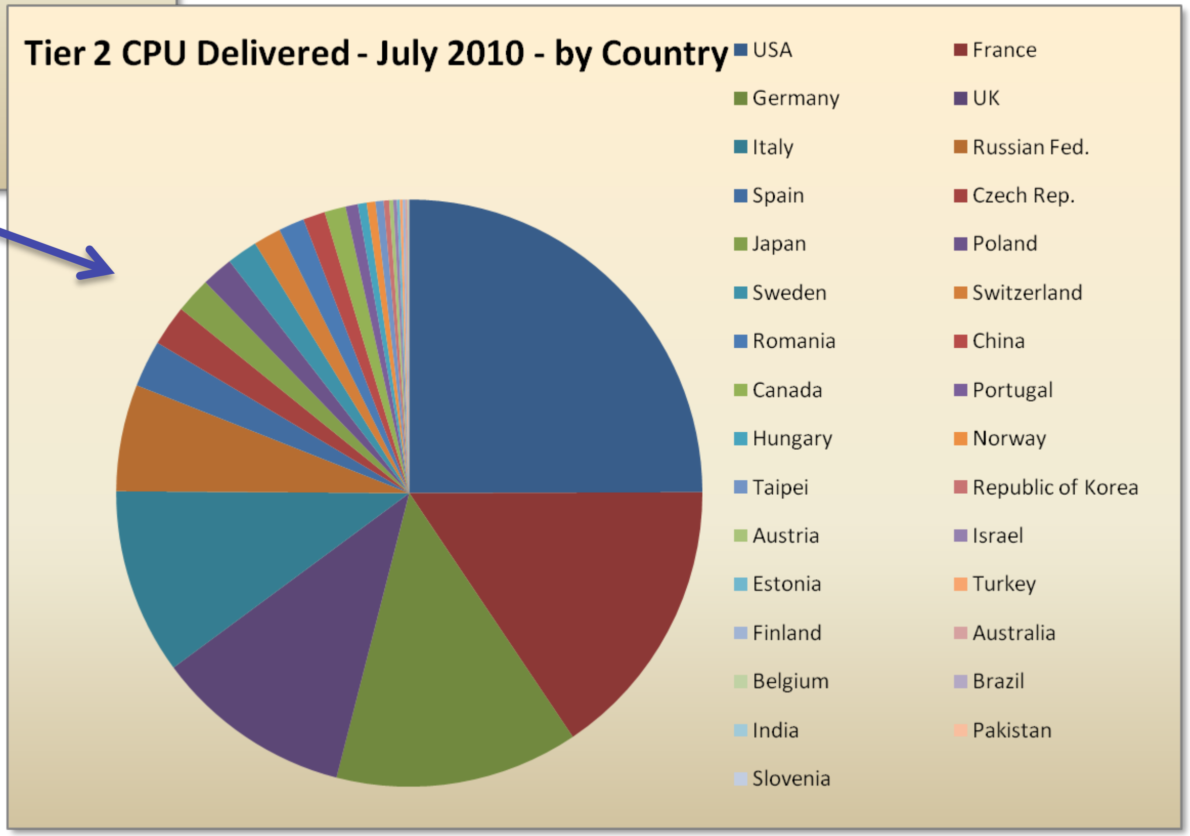
CMS ~800,
ATLAS ~1000,
LHCb/ALICE ~200

CPU – July

- Significant use of Tier 2s for analysis
  - frequently-expressed concern that too much analysis would be done at CERN is not reflected

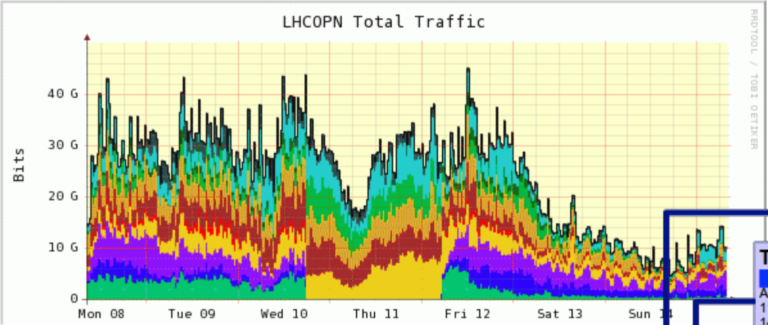


- Tier 0 capacity underused in general
  - But this is expected to change as luminosity increases



Data transfer

- Data transfer capability today able to manage much higher bandwidths than expected/feared/planned



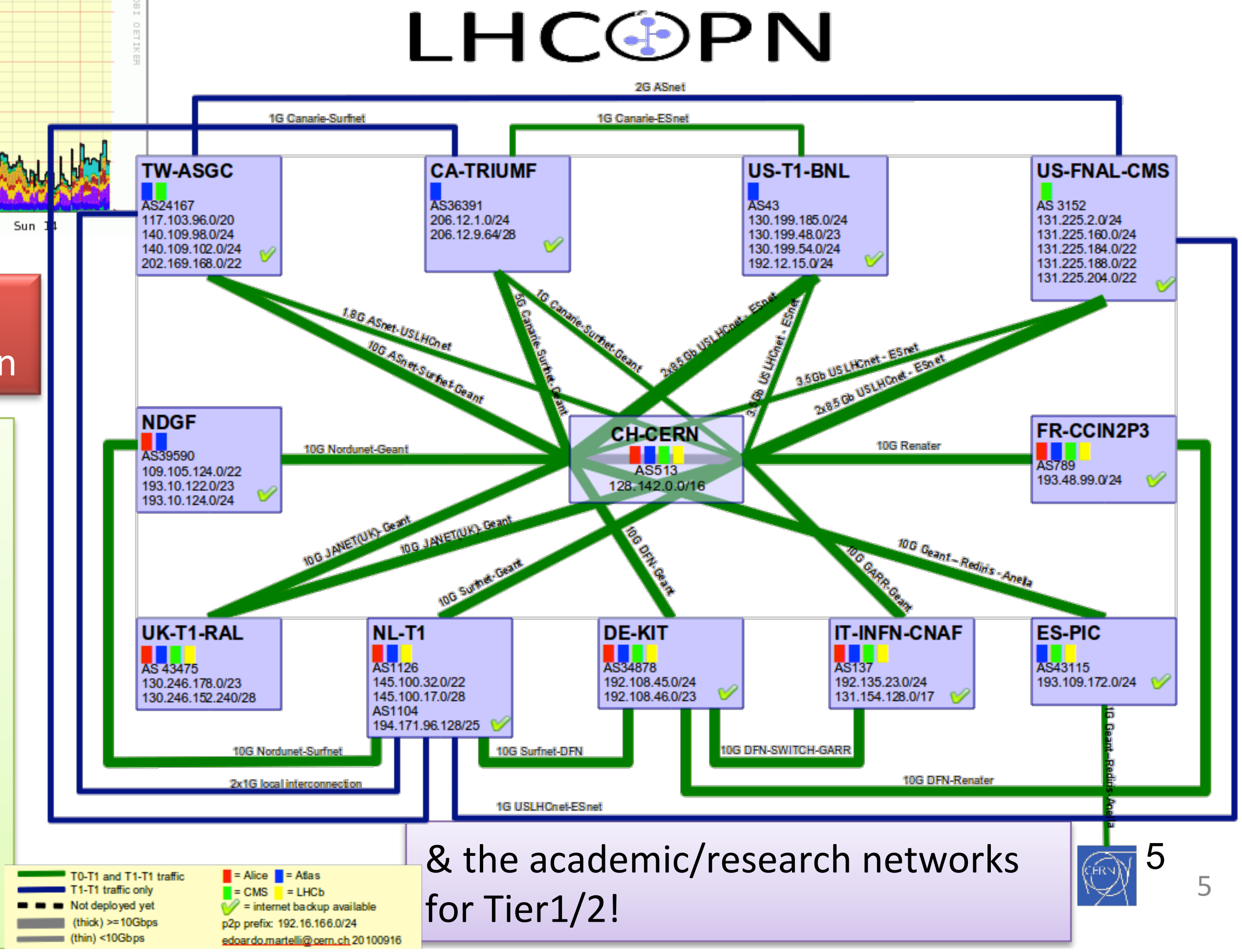
Fibre cut during STEP'09:
Redundancy meant no interruption

Data transfer:

•SW: gridftp, FTS (interacts with endpoints, recovery), experiment layer

•HW: light paths, routing, coupling to storage

•Operational: monitoring



& the academic/research networks for Tier1/2!

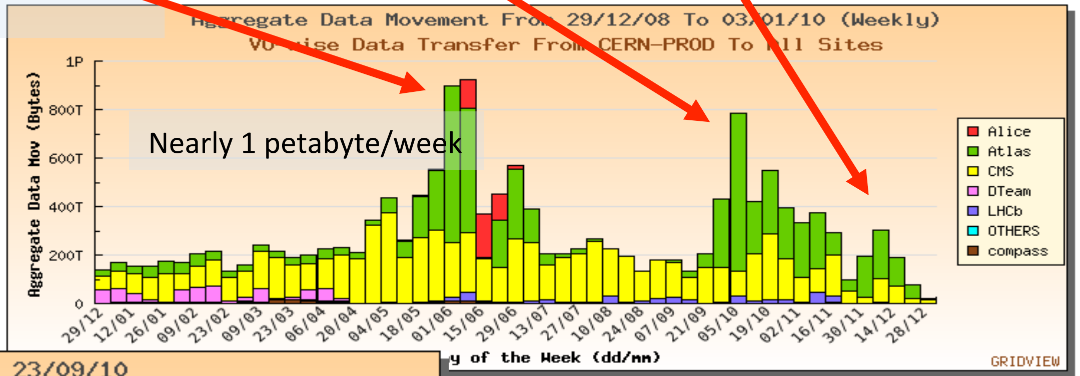
Data transfers

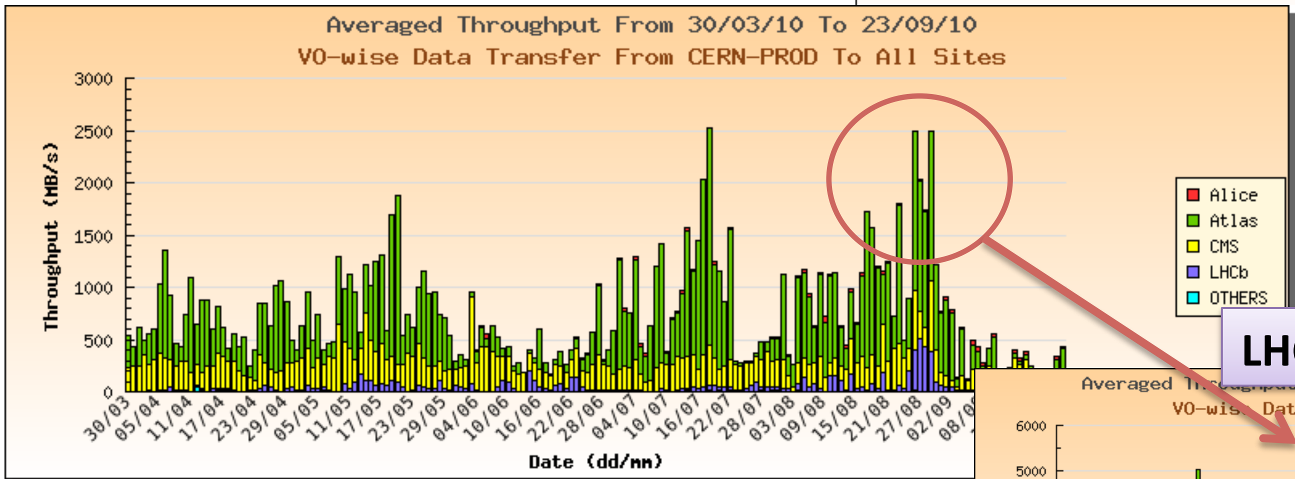
Final readiness test (STEP'09)

Preparation for LHC startup

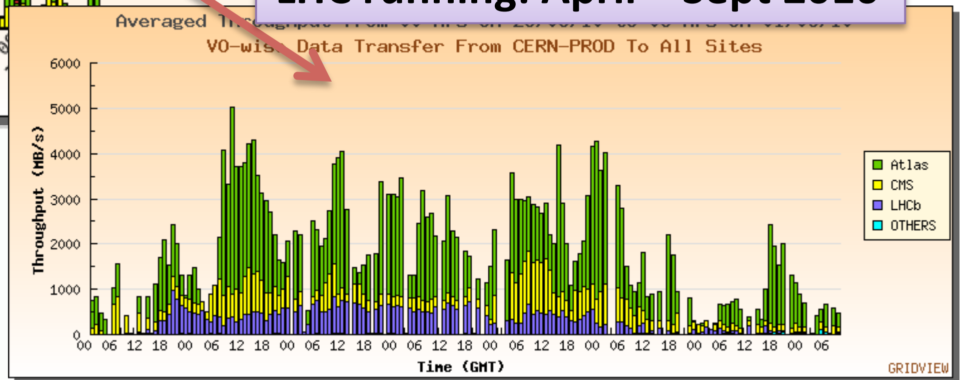
LHC physics data

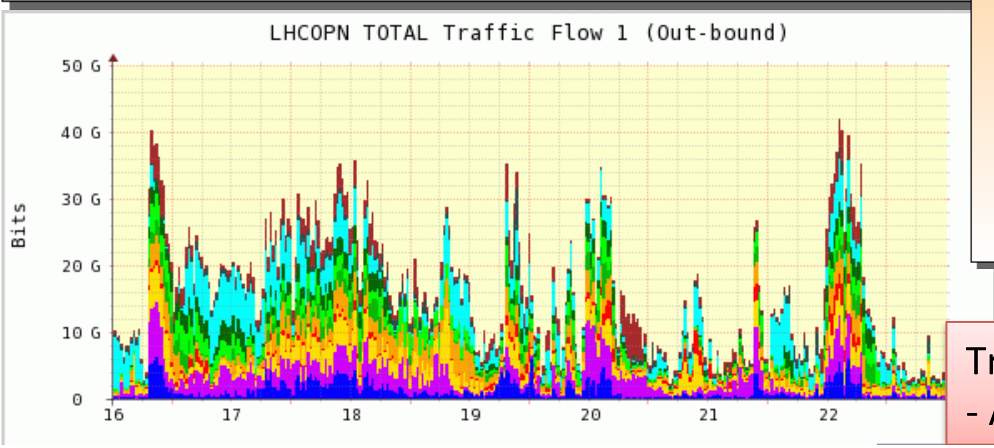
2009: STEP09 + preparation for data





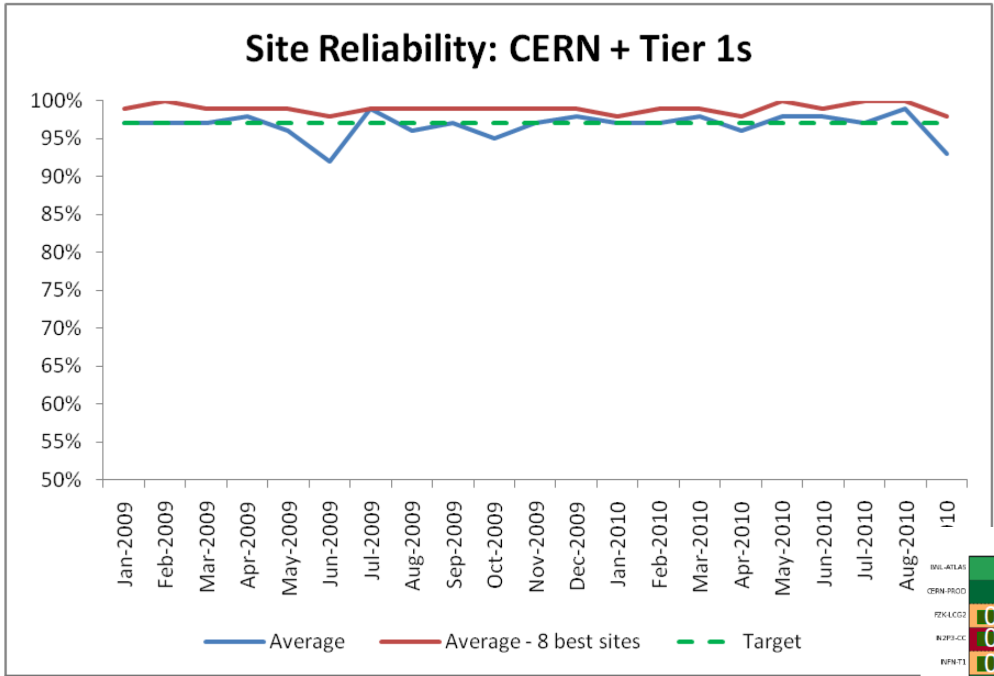
LHC running: April – Sept 2010

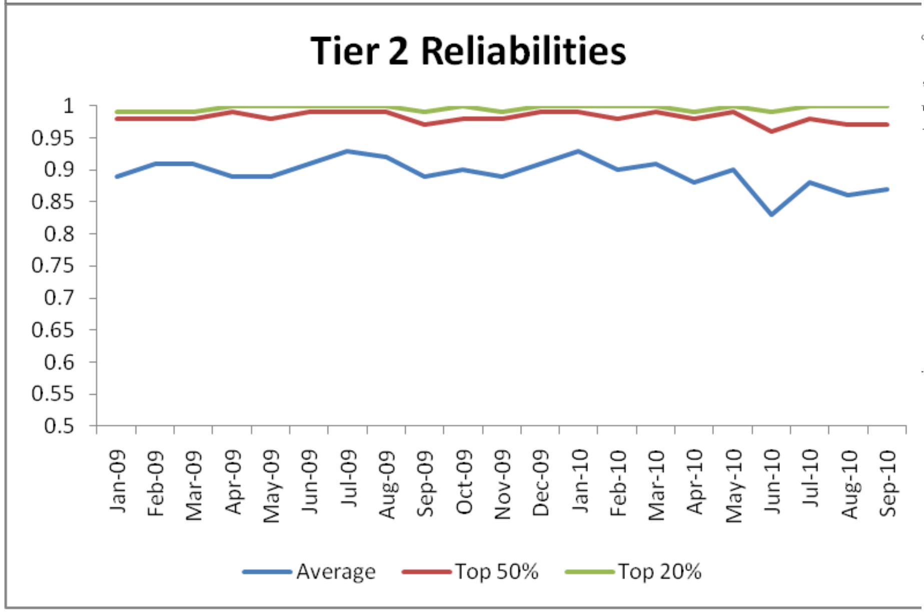




Traffic on OPN up to 70 Gb/s!
- ATLAS early reprocessing campaigns

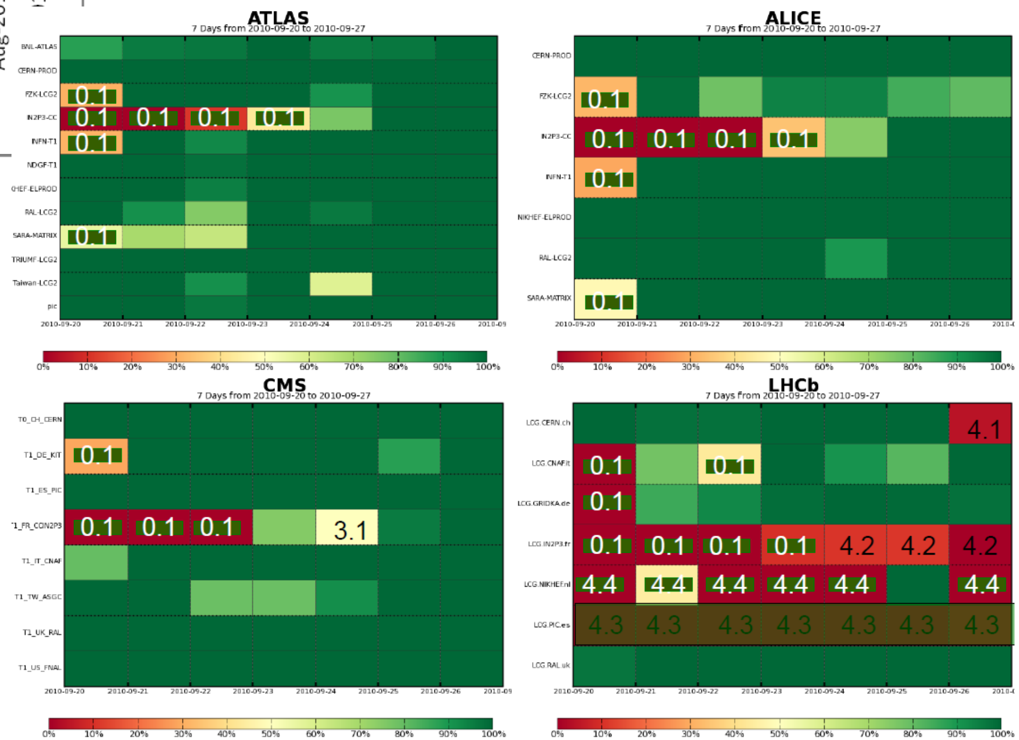
Reliabilities





Experiment-measured site availabilities:

Includes down times during security patching;
At times ~50% of resources were unavailable.



From testing to data:

Independent Experiment Data Challenges

2004

e.g. DC04 (ALICE, CMS, LHCb)/DC2 (ATLAS) in 2004 saw first full chain of computing models on grids

Service Challenges proposed in 2004
To demonstrate service aspects:
- -Data transfers for weeks on end
- -Data management
- -Scaling of job workloads
- -Security incidents ("fire drills")
- -Interoperability
- -Support processes

SC1 Basic transfer rates

2005

SC2 Basic transfer rates

SC3 Sustained rates, data management, service reliability

2006

SC4 Nominal LHC rates, disk→ tape tests, all Tier 1s, some Tier 2s

2007

- Focus on real and continuous production use of the service over several years (simulations since 2003, cosmic ray data, etc.)
- Data and Service challenges to exercise all aspects of the service – not just for data transfers, but workloads, support structures etc.

2008

CCRC'08 Readiness challenge, all experiments, ~full computing models

2009

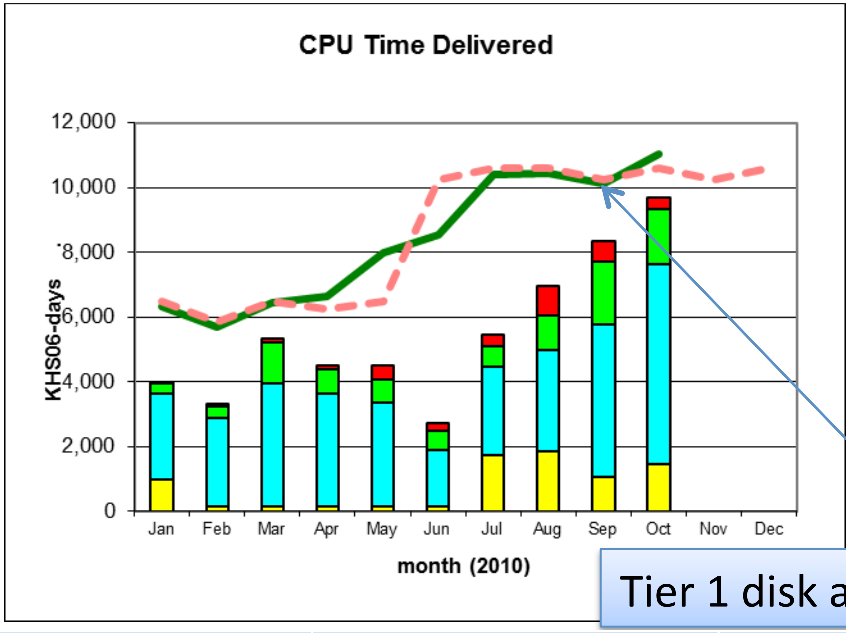
STEP'09 Scale challenge, all experiments, full computing models, tape recall + analysis

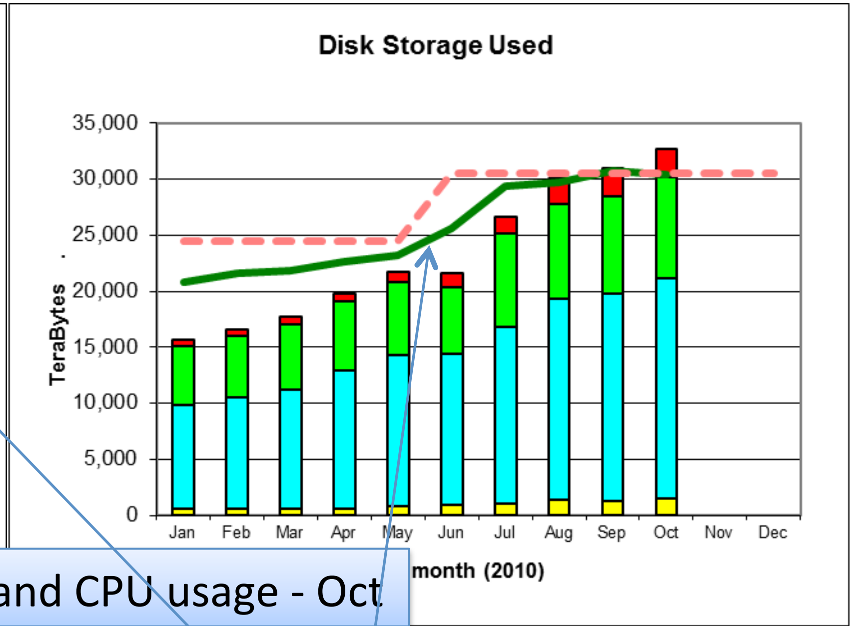
2010

8

Resource usage

- Now Tier 1s and Tier 2s start to be fully occupied; as planned with reprocessing, analysis, and simulation loads



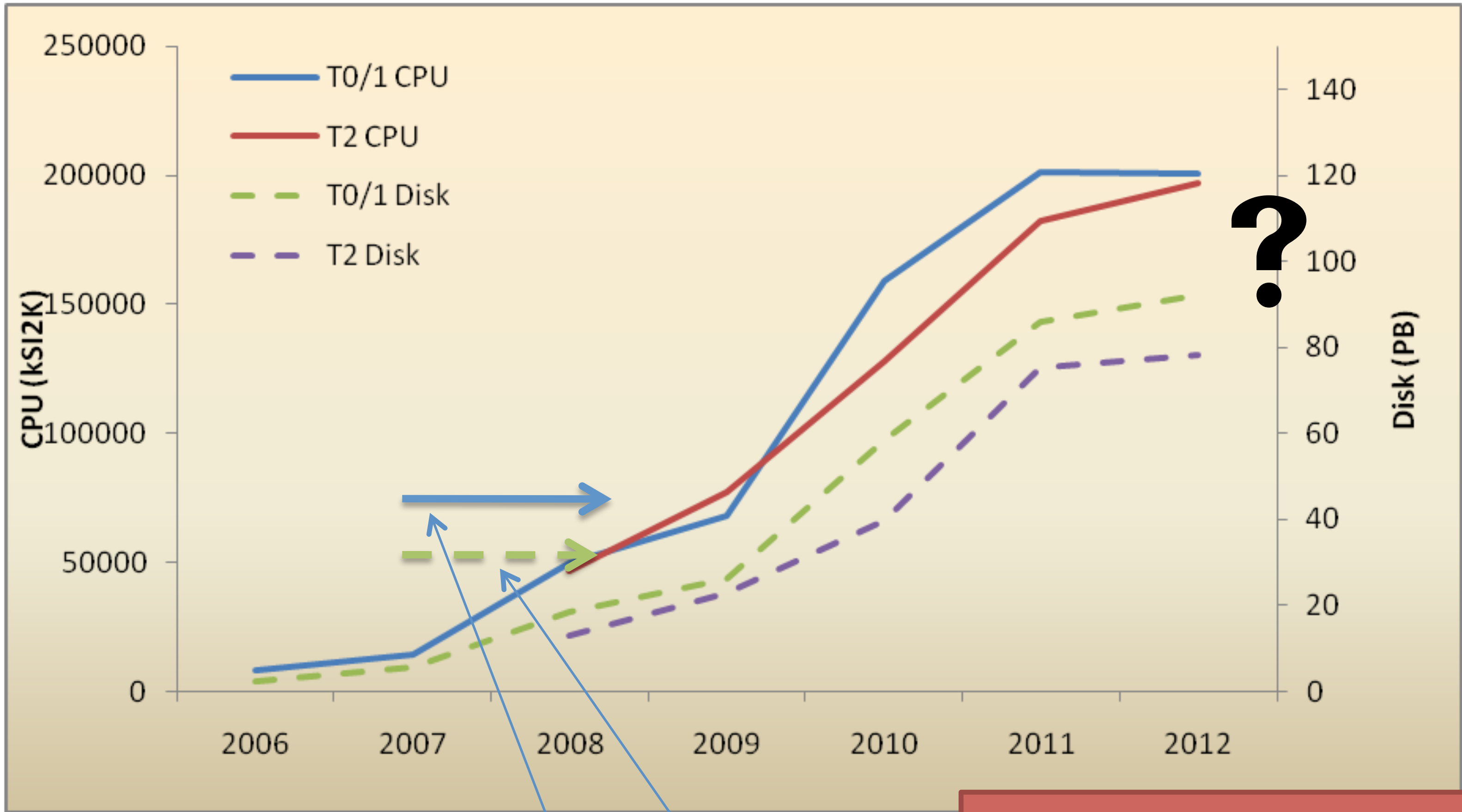


Tier 1 disk and CPU usage - Oct

Tier 1 use - Oct	CPU use/pledge	Disk use/pledge
ALICE	1.04	0.25
ATLAS	0.94	0.89
CMS	0.54	0.74
LHCb	0.27	0.79
Overall	**0.78**	**0.75**

NB: Assumed effic factors
0.85 for CPU
0.70 for disk

Resource Evolution (no run in 2012)



Expected needs in 2011 & 2012

Need foreseen @ TDR for T0+1 CPU and Disk for 1st nominal year

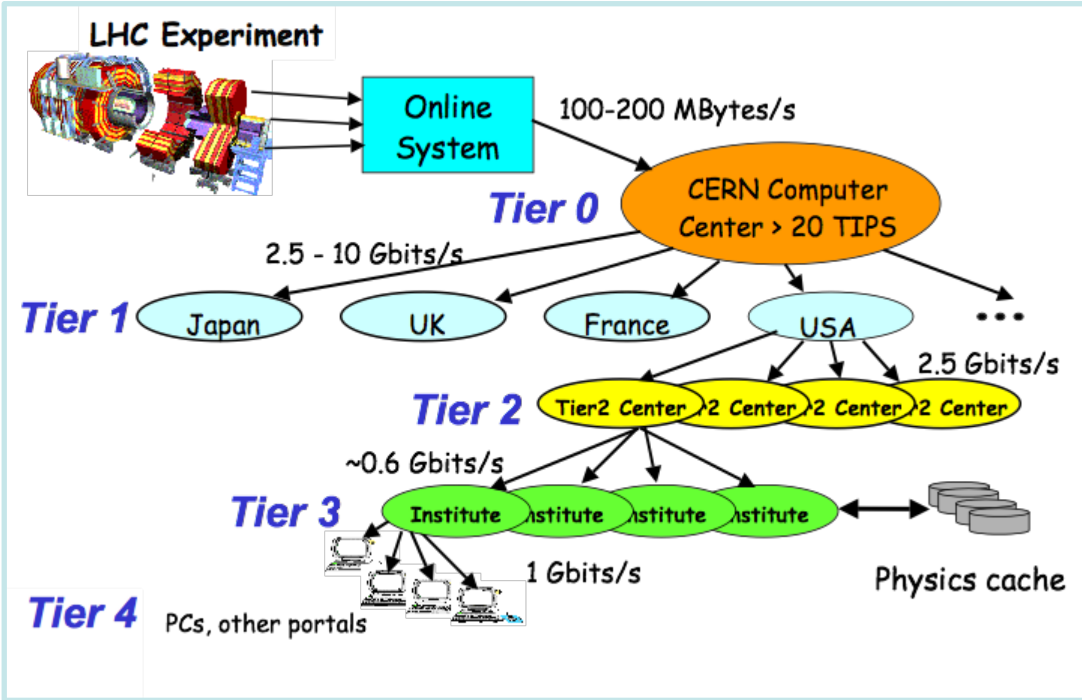
NB. In 2005 only 10% of 2008 requirement was available. The ramp-up has been enormous!

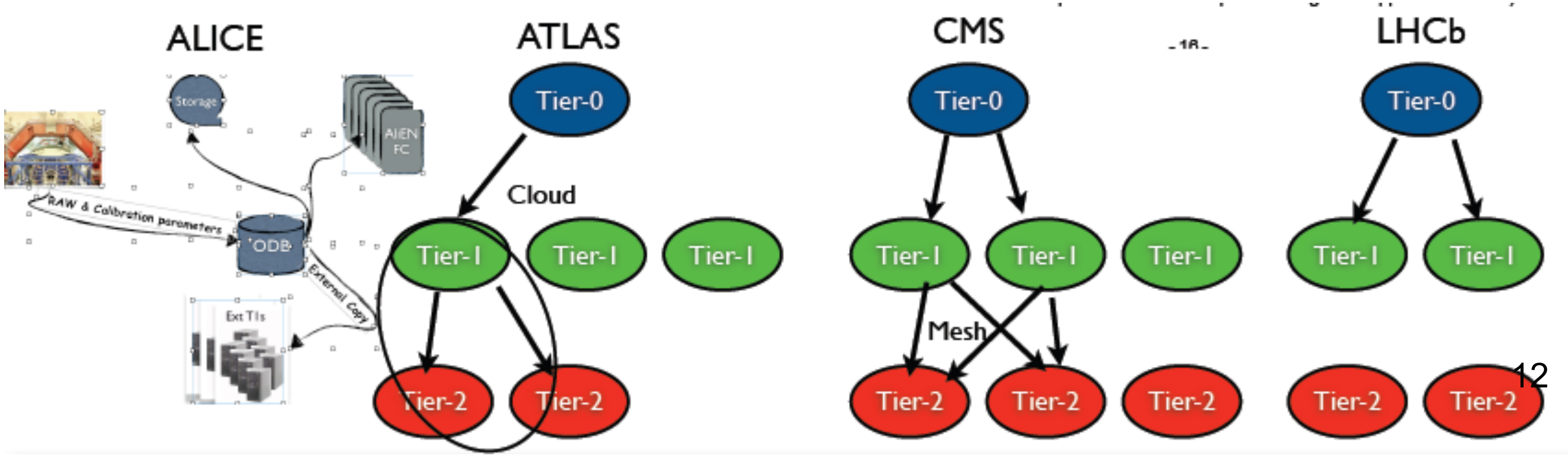
Elements of a computing model

- Basic parameters
  - How many events, how many event types
  - Event size, event types
  - Processing times
- Data distribution
  - Filtering, skimming, slimming
  - How many copies in Tier1/Tier2 ensembles
- Data processing
  - “Scheduled” activities: how many processes in a year? How long is a reprocessing cycle? How many versions on disk?
  - “Chaotic” activities: how many analysis groups/users? How frequently do they access data? How much time for a full pass?

Experiment models have evolved

- Models all ~based on the MONARC tiered model of 10 years ago
- Several significant variations, however

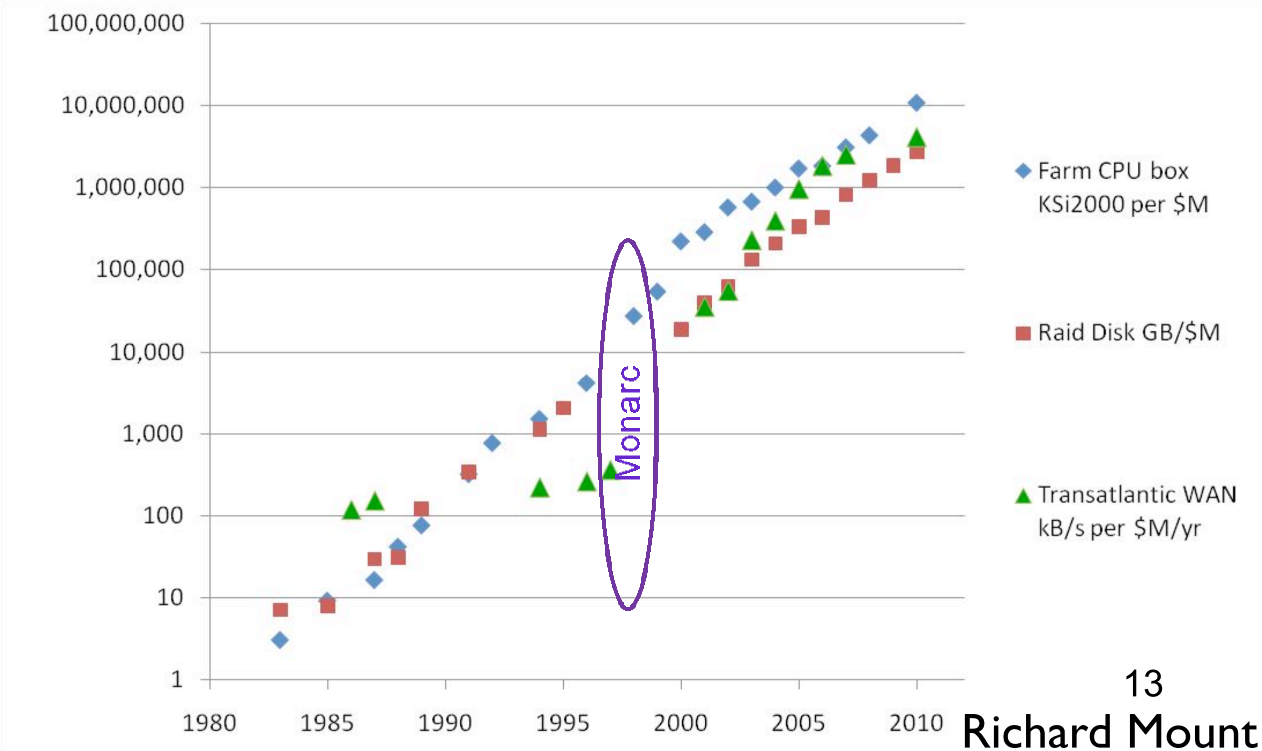




The Monarc rationale

- The MONARC computing model of 2000 relied heavily on data placement
- Jobs were sent to datasets already resident on sites
- Multiple copies of the data would be hosted on the distributed infrastructure
- General concern that the network would be insufficient or unreliable

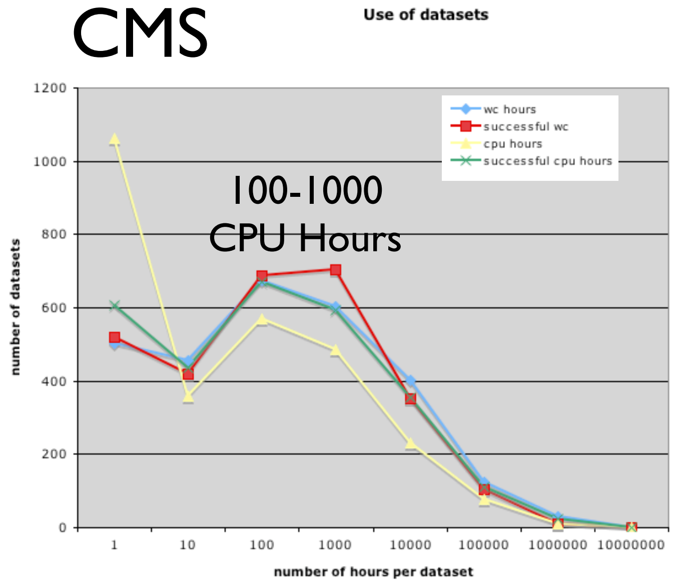
- As we have just seen, this is no longer the case nowadays
- Look at ways to make more efficient use of the resources

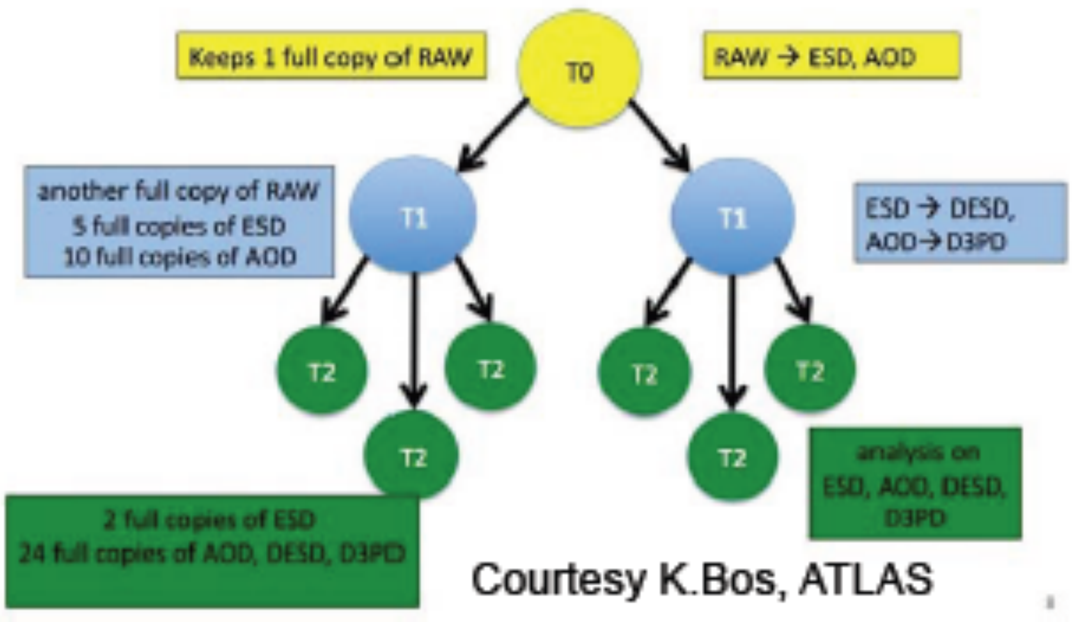


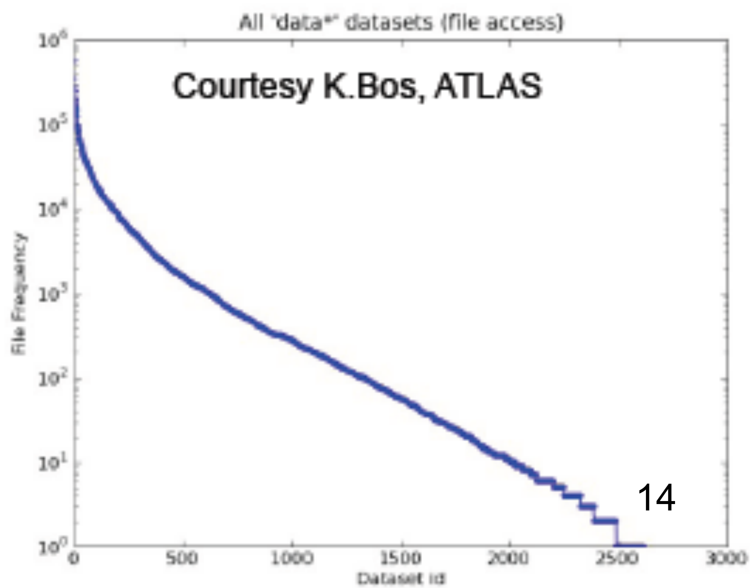
Richard Mount

Data placement and usage today

- Small subset of data distributed is actually used
- Don't know a priori which dataset will be popular
  - CMS has 8 orders magnitude in access between most and least popular
- Data is only popular for a short time (~2 weeks)
- Data duplication increases disk usage
  - ATLAS: per 1 PB raw data, creates 7 PB derived data





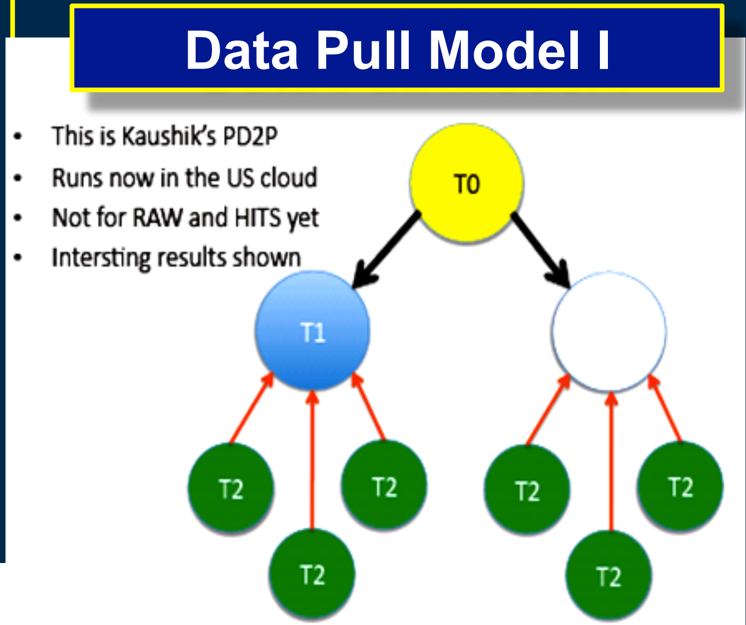


Evolution of data placement

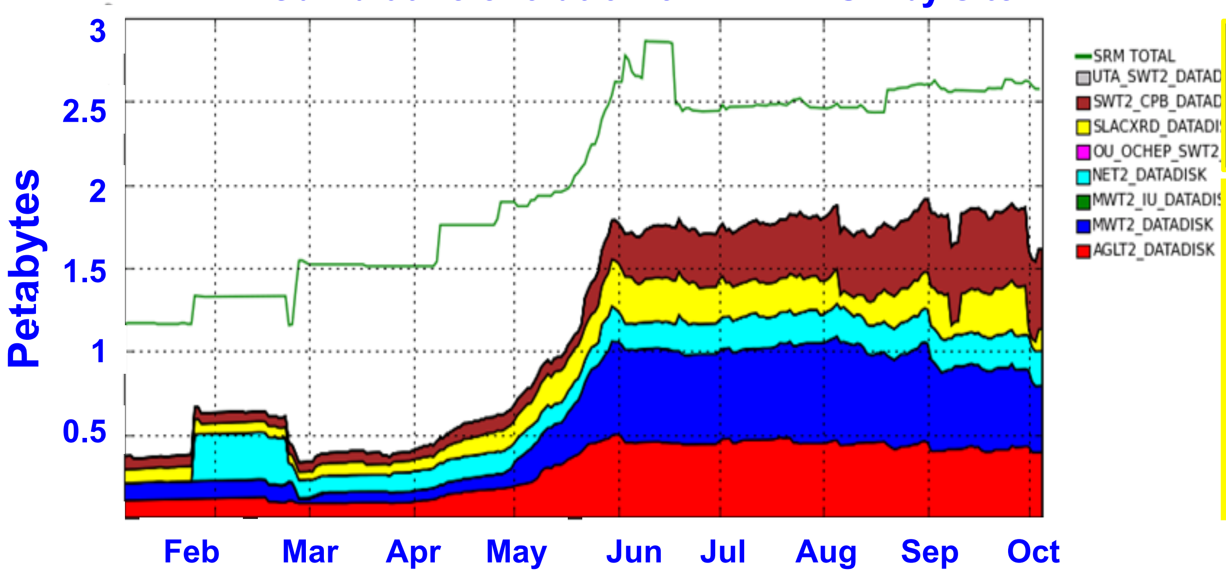
- Move towards caching of data rather than strict planned placement
- Download the data when required
  - Selects popular datasets automatically
  - When datasets no longer used will be replaced in the caches
- Data sources can be any (Tier 0, 1, 2)
- Can still do some level of intelligent pre-placement
- Understanding a distributed system built on unreliable and asynchronous components means
  - Accepting that catalogues may be not fully updated
  - Data may not be where you thought it was
  - Thus must allow remote access to data (either by caching on demand and/or by remote file access)

Pull Model in Atlas BNL Cloud

- **PD2P: Atlas implementation of the pull model**
  - Tier1 used as repository (Tier0-Tier1: Push)
  - Dynamic data placement at Tier2s
  - Dataset is subscribed to a Tier 2 if no other copies are available (except at a Tier 1), as soon as any user needs it
- **Deployed in the US (BNL) cloud in June**



Cumulative evolution of DATADISK by site



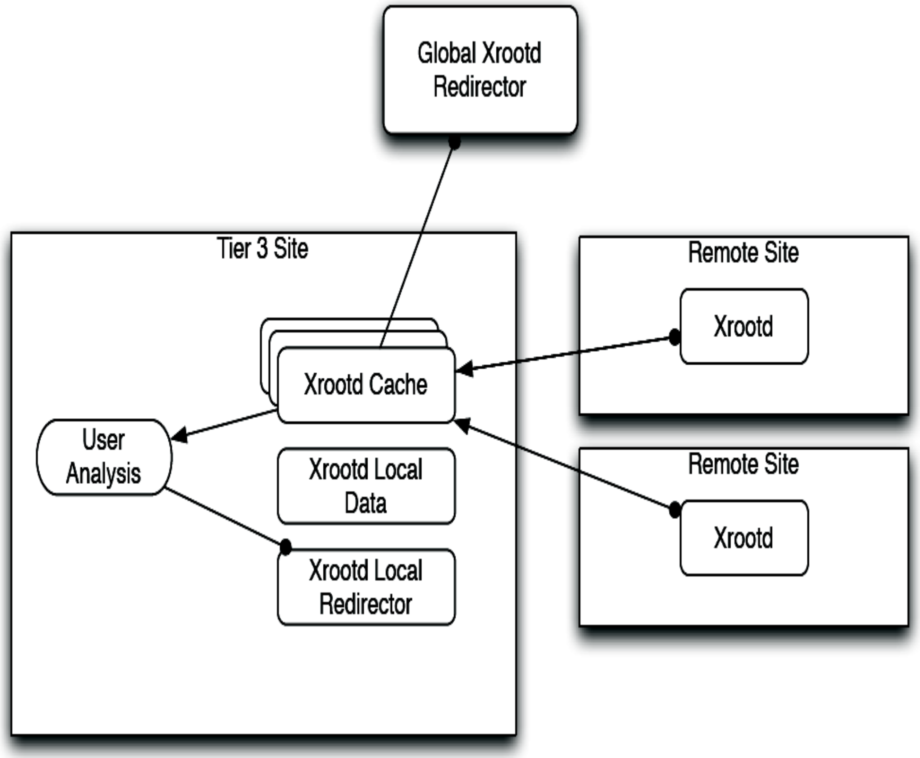
Kaushik De, Atlas Week Oct 2010

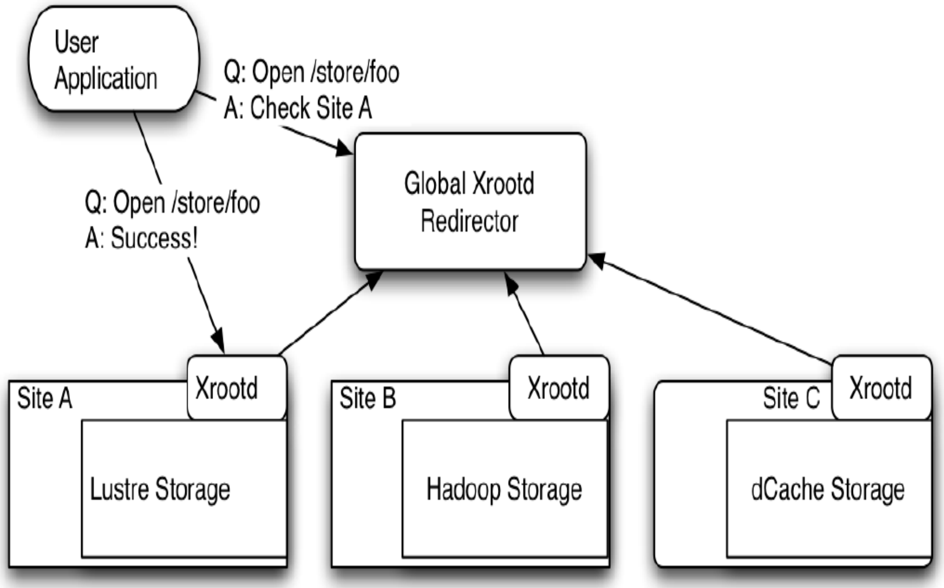
Before: Exponential rise from right after LHC start

Much slower rise in disk utilization since July

Remote Data Access and Local Processing with Xrootd (CMS)

- Useful for smaller sites with less (or even no) data storage
- Only selected objects are read (with object read-ahead).
  No transfer of entire data sets
- CMS demonstrator: Omaha diskless Tier3, served data from Caltech and Nebraska (Xrootd)





Strategic Decisions:
Remote Access
Vs
Data Transfers

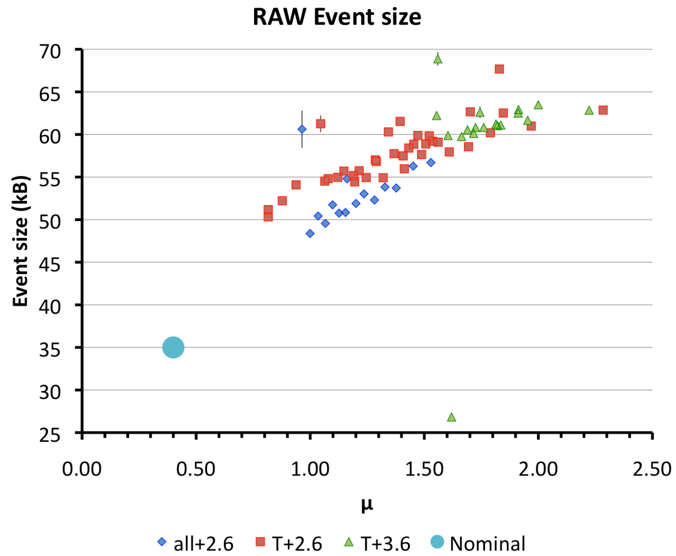
Brian Bockelman, September 2010

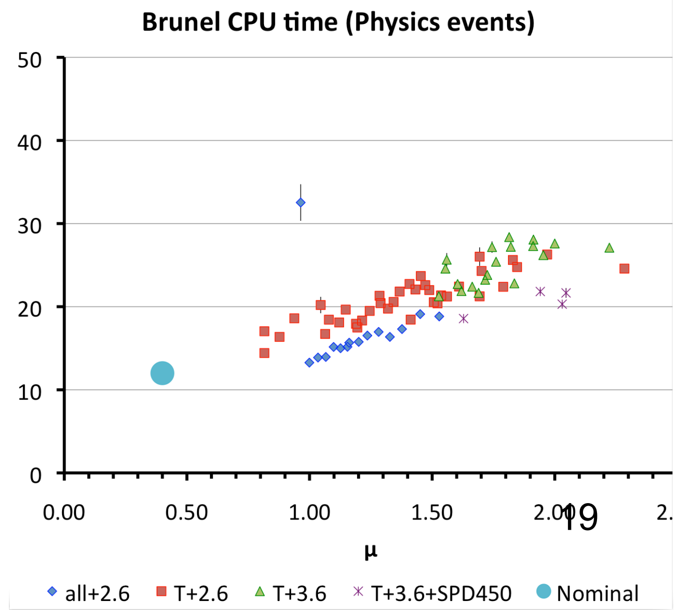
Implications for networks

- Hierarchy of Tier 0, 1, 2 no longer so important
- Tier 1 and Tier 2 may become more equivalent for the network
- Traffic could flow more between countries as well as within (already the case for CMS)
- Network bandwidth (rather than disk) will need to scale more with users and data volumes
- Data placement will be driven by demand for analysis and not pre-placement

Processing challenges

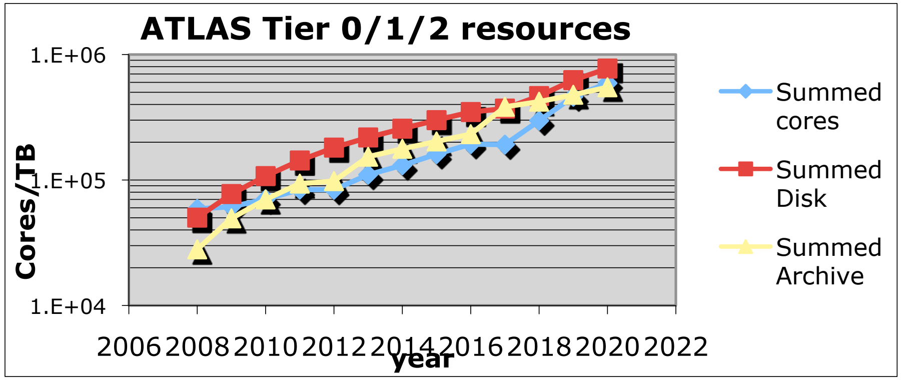
- Event sizes: a concern for most experiments
  - Processing times increase with collisions per bunch due to pile-up
- LHCb processing time quadratic with event size
  - Full luminosity, events twice design size, increased memory, 4x design processing time
    - File sizes being reduced, 2x speed up of reconstruction, x10 for stripping of events
    - Possible due to model flexibility
- ATLAS: despite big improvements, CPU time for MC generation still an issue
- Not all bad: CMS processing times and event sizes smaller than planned
  - Low-luminosity effect, plus speed-up of code





Future Challenges

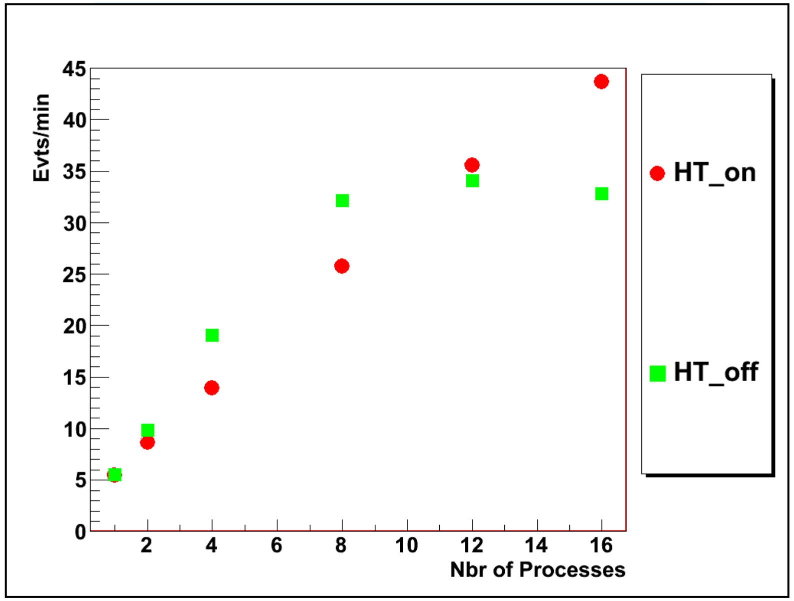
- We assume we can use growth in CPU
  - But this implies changing architectures
  - And handle the data throughput



- Experiments already working to deal with multi cores
  - Many cores and GPGPUs are down the line
- We need to use them or be very clear why we cannot

Parallelism

- Generally work smarter!
  - E.g. AthenaMP Event level parallelism
    - Share common memory between parent and daughter processes to allow many on a single node
    - Some speed-up using event loop parallelism
  - Also share common pages between processes with KSM
    - Real gains in memory use, but some slow-down
  - Cache as much as you can (e.g. pile-up events)
  - Also Non-Uniform Memory Access, simultaneous multi-threading
- Issues: hard to monitor performance in parallel jobs



- Other approaches
  - Job level parallelism (e.g parallel Gaudi) & hyperthreading
  - CMS working on this sort of ‘workflow’ parallelism
  - Pinning of processes to cores or hyperthreads with Affinty

Virtualisation and "clouds"

- .... Another hype / marketing / diversion ???

- Yes, but
  - Virtualisation is already helping in several areas
    - Breaking the dependency nightmare
    - Improving system management, provision of services on demand
    - Potential to help use resources more effectively and efficiently (many of us have power/cooling limitations)
    - Use of remote computer centres
  - Cloud technology
    - Let's not forget why we have and need a "grid"; much of this cannot be provided by today's "cloud" offerings
      - Collaboration (VO's), worldwide AAI and trust, dispersed resources (hw and people),
    - Although we should be able to make use of commercial clouds transparently

What about Grid middleware?

The *Basic* Baseline Services – from the TDR (2005)

- Storage Element
  - Castor, dCa[…]
  - Storm added in 2007
  - SRM 2.2 – deployed in production – Dec 2007
- Basic transfer t[…]
- File Tran[…]
- LCG File […]
- LCG data mgt tools - lcg-utils
- "Posix" I/O – ✓
  - Grid File Access Library (GFAL)
- Synchronised databases T0←→T1s
  - 3D project

SRM is too complex

OK, but why not HTTP?

OK for some use cases

OK, but must sync with storage
No need for distributed catalogue

Frontier/Squid for many use cases

- Informa[…]
  - BDII
- Compute El[…]
  - Globus/[…]
  - web ser[…]
  - Support for multi-user pilot jobs
- […]
  - WMS, LB
- VO Management System (VOMS), MyProxy ✓
- VO Boxes
- Application […]
- Job Monitoring Tools
- APEL etc. ✓

LDAP → messaging?
Static vs dynamic info

Still have LCG-CE, not yet replaced; MUPJs!

Actual LHC use cases much simpler
Pilot frameworks may supercede it

→ Virtual machine

→ CVMFS or Squid

→ MSG, Nagios, etc

What about grid middleware?

- Clearly a thinner layer today than originally imagined
  - And the actual usage is far simpler
- Experiment layer is deeper ... And different from one to the other
- Experiments had to work hard to (mostly) hide the grid details from users
- Pilot jobs are (almost) ubiquitous in all experiments
- Simplification of some services is possible and helps long term maintenance and support
- The current grid infrastructure can sit transparently over virtualised (cloud) services
  - And provide a potential path for evolutionary change

Automation, monitoring and testing

- Operations are still too effort-intensive
  - increase automation
- Monitoring is essential to keep system going and understand its usage patterns
  - More to be done for storage systems
  - Tendency to have too much!
  - Keep distinct views for experiments, sites, and managers
- Lots of testing results in outstanding availability and reliability
  - Revealed many configuration problems (e.g. ATLAS Hammercloud)

Conclusioni

- Il sistema di calcolo distribuito degli esperimenti a LHC ha funzionato molto bene in questo primo periodo di presa dati
- Le risorse a disposizione degli esperimenti erano "comode"
  - Che succederà quando LHC arriverà a regime?
- I modelli di calcolo si stanno evolvendo allo scopo di ottimizzare l'utilizzo delle risorse sfruttando gli "asset" consolidati
  - Bisogna capire bene le implicazioni sulla rete
- Occorre rimanere al passo con le tecnologie di punta…
  - Cambiamenti di architettura per many-core? GPU?
  - Virtualizzazione?
  - Cloud computing?
- …continuando a garantire il buon funzionamento di quanto è stato fatto finora
  - Automatizzare, testare, monitorare