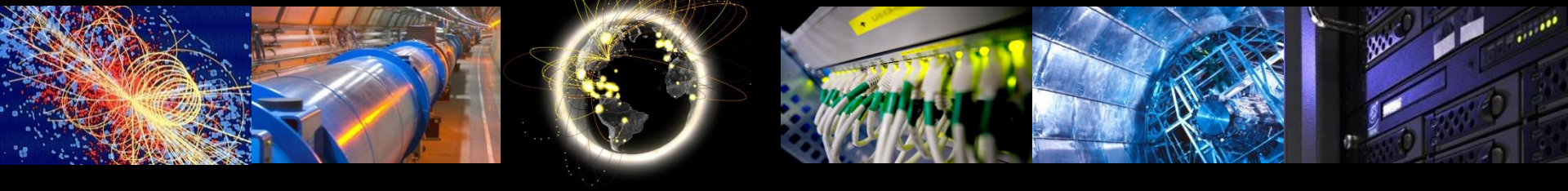
Cost and system performance modelling in WLCG and HSF: an overview

Andrea Sciabà
on behalf of the HSF/WLCG Systems performance and cost modelling WG

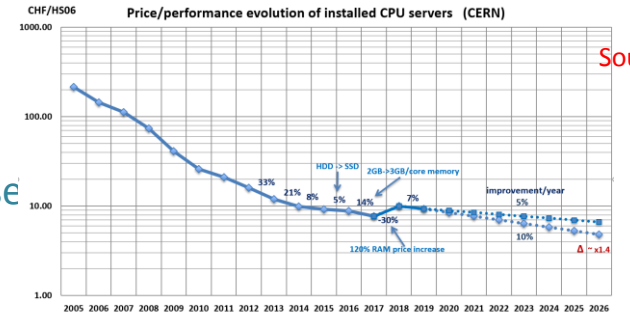
ATLAS Site Jamboree
CERN, 5-8 March 2019



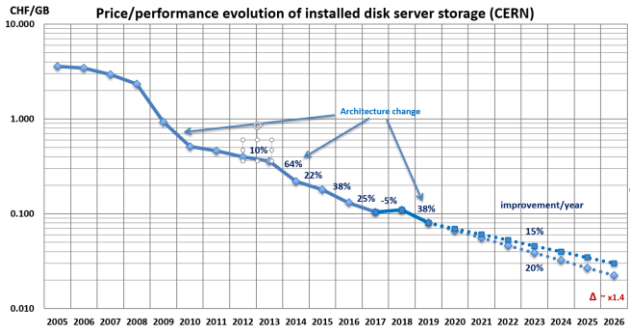
WLCG Worldwide LHC Computing Grid
HSF HEP Software Foundation

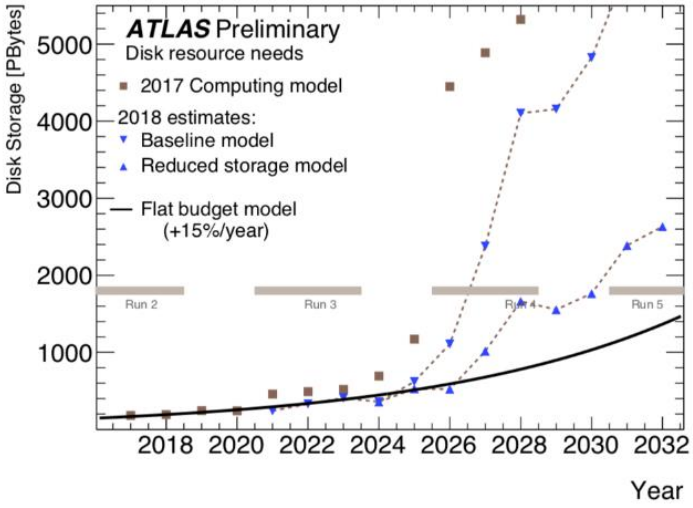
The High Luminosity challenge

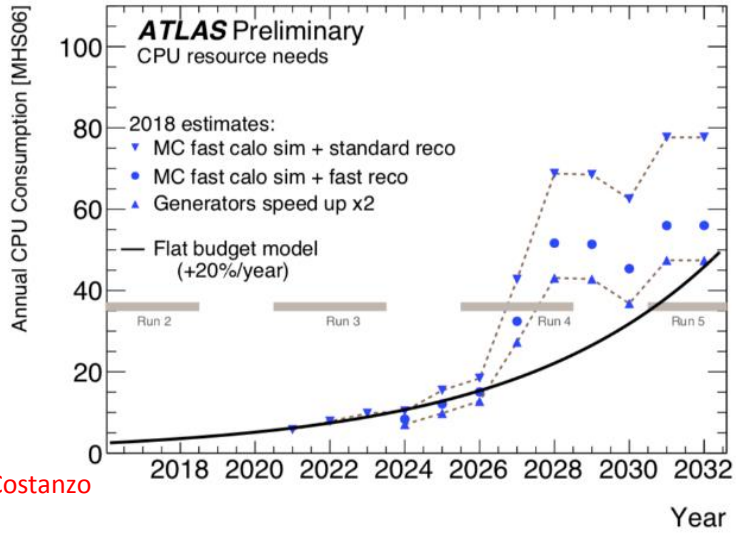
- There is still a significant gap between the estimations of needed and available resources
  - 10x increase in trigger rates, 5x increase in pile-up, NLO & NNLO, …
  - Price/performance advances slowing down, 10-15%/year at best
- CPU and disk short by a factor ≈ 2
  - Even if the gap is reducing!
- Strong need to quantitatively understand our efficiency and how we can optimise performance



Source: B. Panzer-Steindel







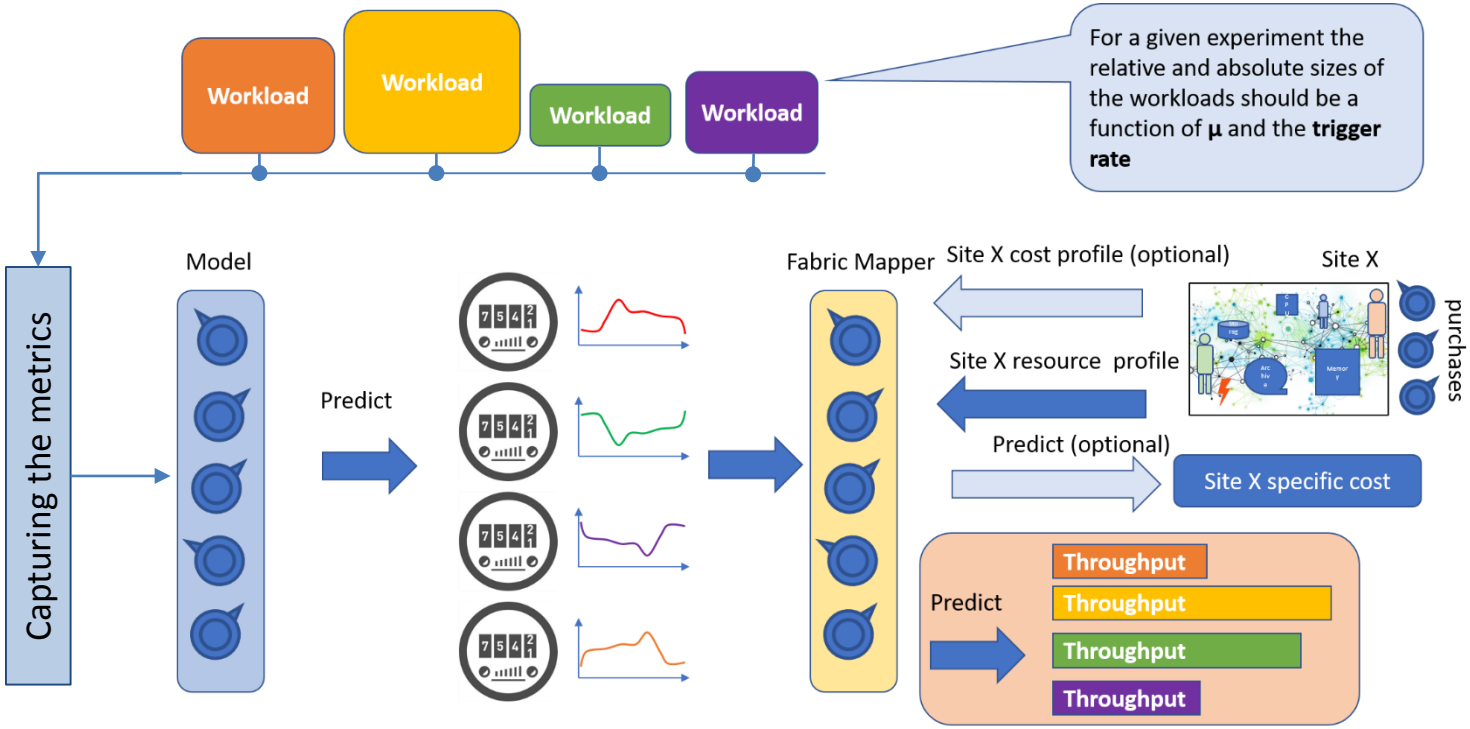
Source: D. Costanzo

The Working Group

- Main motivation is to help WLCG to fit into the available resources for Run3 and Run4
  - Develop a deep understanding of current workloads, resource utilisation, and site costs
  - Explore future scenarios, estimate possible improvements in efficiency
  - Develop tools and methods for the above
- Some of the current areas of work and goals
  - Identify representative experiment workloads
  - Define which metrics best characterise such workloads
  - Understand how to estimate resource needs
  - Define a process to evaluate the TCO of an infrastructure
  - Measure the impact of new storage configurations on applications and costs

Metrics and workload characterisation

- Identify the metrics that best describe a workload
  - To understand if the hardware is used efficiently → software experts
  - To quantify the resource utilisation on the node → site administrators
  - Record time series and extract summary numbers (averages, percentile values, etc.)



Metrics

- Started with a comprehensive list of basic metrics
- Try to have the smallest amount of parameters describing as completely as necessary the workloads
- Prmon (Github) is an HSF tool that collect most of these metrics
  - No overhead, reads from /proc/<pid>/smaps

Metric	Type	Source	Scope	Command	Insight	Comments
I/O rate	gauge	/proc/diskstats	global	iostat 1 1	Total IO operations ongoing, can calculate a %usage of theoretical maximum of spinning/ssd media	As /proc/diskstats is global some method of isolating a process is necessary to assess accurately (containers/namespaces?)
I/O bandwidth	gauge	/proc/<pid>/io	process	prmon	Total bytes read/written by a process, gives indication of rates and total usage	

I/O

Metric	Type	Source	Scope	Command	Insight	Comments
%usage	gauge	Tool internal	process	/bin/time <x> prmon	Gross measure of cpu utilisation, real/user/sys. Indicates potential overheads and multi-process scaling.	Use application metric of event loop time to change all of these per second metrics into per event (see below)
Thread #	gauge	/proc/<pid>/status	process	grep Threads	Gives a measure of how much of a running payload is parallel/serial.	Required for multi-threaded code
Process #	gauge	Process list	process	pstree -p <p> |wc	As above but for multi-process codebases.	Required for multi-process code

CPU

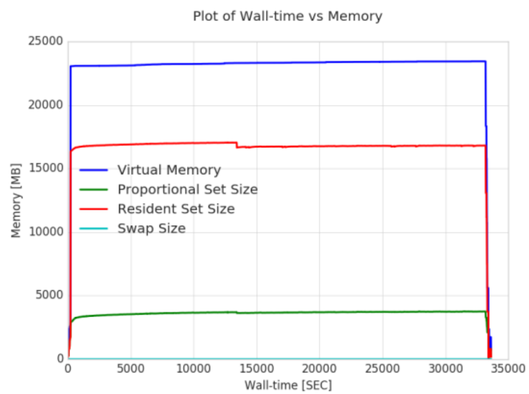
Metric	Type	Source	Scope	Command	Insight	Comments
Memory usage	gauge	/proc/<pid>/smaps /proc/<pid>/status	process	prmon	Allows understanding of how memory develops over time, can be used in conjunction with Process/Thread count to examine dependency.	*VMEM is application controlled, RSS is how much the kernel really maps, PSS accounts for shared pages better (important for parallel processing).*
Avg Mem	gauge	/proc/<pid>/smaps	process	prmon	Amount of memory that needs budgeted for the bulk of the runtime of the job payload.	*(see above)*
Max Mem	gauge	/proc/<pid>/smaps	process	prmon	Amount of memory that needs to be made available instantaneously - required for setting hard limits on a job payload to detect erroneous jobs.	(see above)

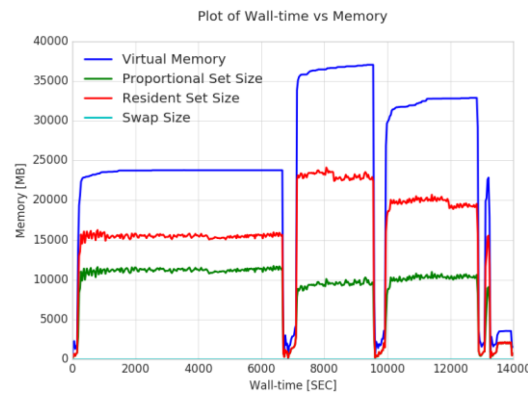
Memory

Metric	Type	Source	Scope	Command	Insight	Comments
Network usage	gauge	/proc/net/dev	global	Possible update to prmon	Aggregate Tx/Rx bytes to assess total network load	As /proc/net/dev is global some method of isolating a process is necessary to assess accurately (containers/namespaces?)
Network rates	gauge	Socket statistics	process	ss -ip	Per process rates, can be used to assess /cvmfs usage.	More work needed to understand if the numbers provided are useful

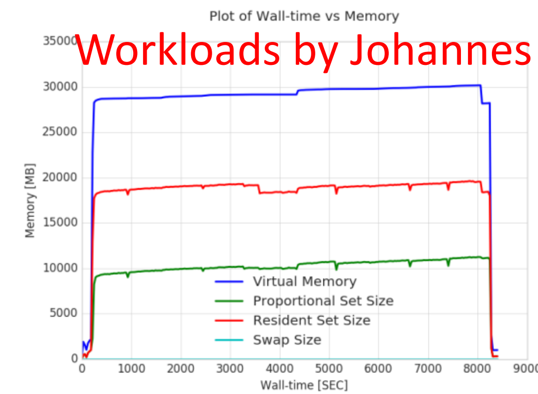
Network

PrMon monitoring plots: ATLAS examples

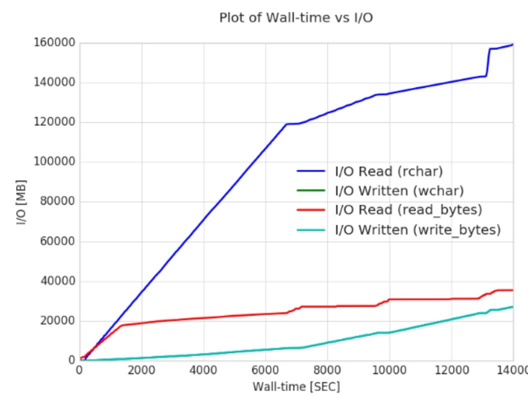




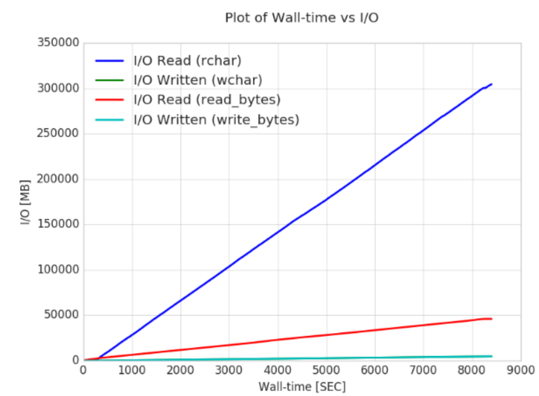
Workloads by Johannes



Simulation



Digi-Reco



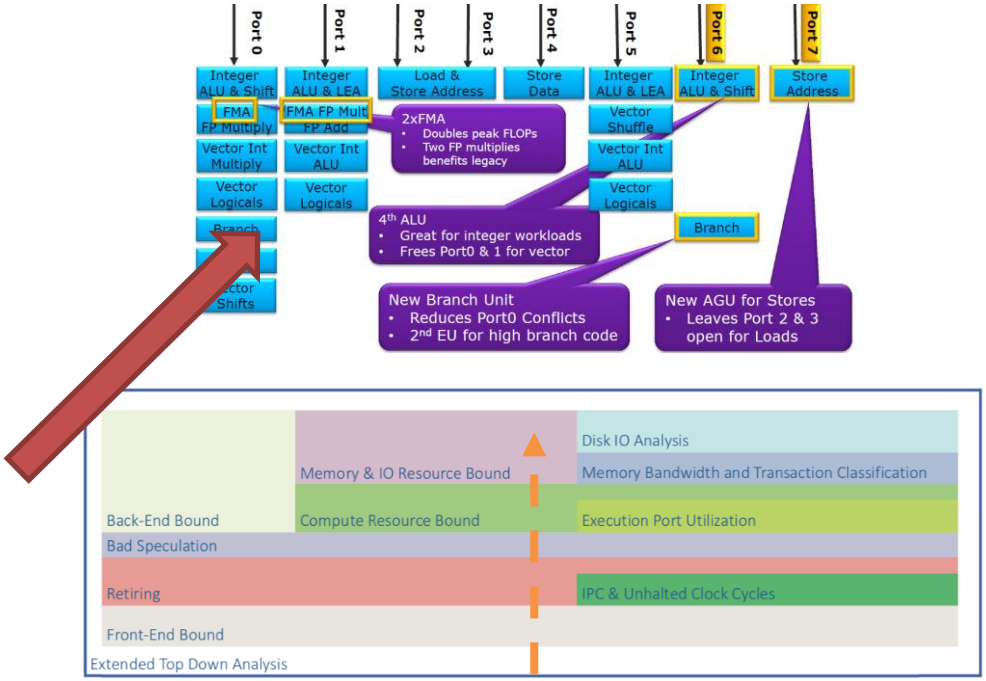
Derivation

Type	Events	Processes	Walltime (h)	CPU eff (%)	Mem/proc (GB)	Read rate/proc (MB/s)	Write rate/proc (MB/s)
Sim ttbar G4	1000	8	9.3	100	0.44	0.015	0.009
Digireco	2000	8	3.9	87	1.12	0.32	0.24
Deriv	95741	8	2.3	98	1.20	0.68	0.07

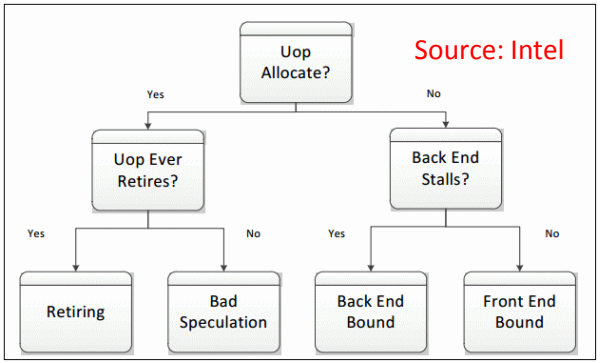
WLCG Worldwide LHC Computing Grid

Measuring performance with Trident

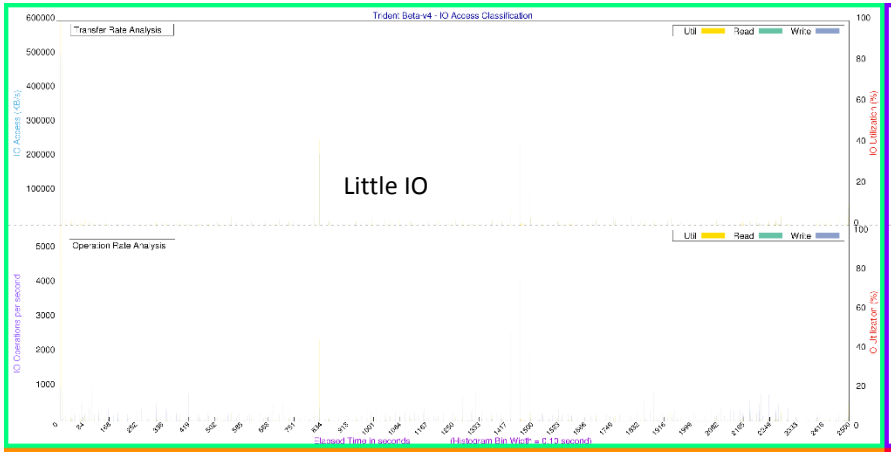
- Measures CPU, IO and memory utilisation based on hardware counters, memory and IO information
- Several metrics calculated
  - CPU: IPC, total cycles, top-down analysis (front-end bound, back-end bound, retiring, bad speculation)
  - Core backend utilization: compute (ports 0,1,5,6) vs memory (ports 2,3,4,7)
  - Memory: bandwidth usage, transaction classification (page-hit, page-empty, page-miss)
- Can be used to see how workloads differ (or resemble) the benchmarks we use (e.g. HS06)
- CPU counters are a powerful (but complex) tool and Trident makes them accessible

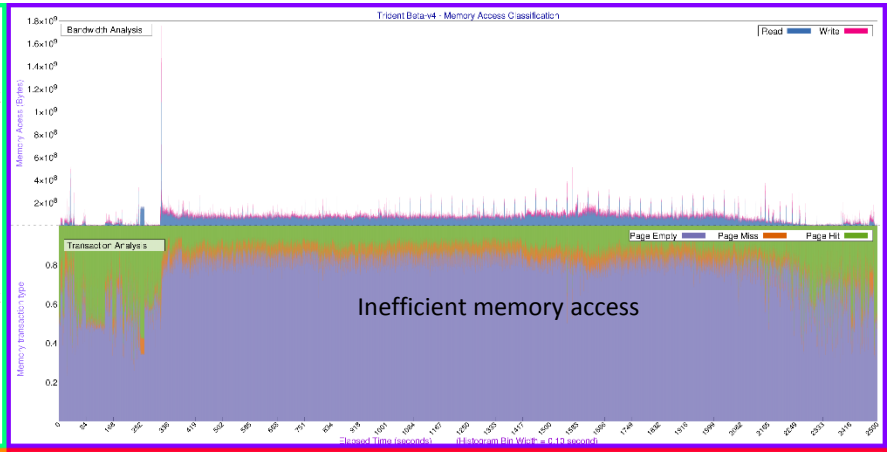


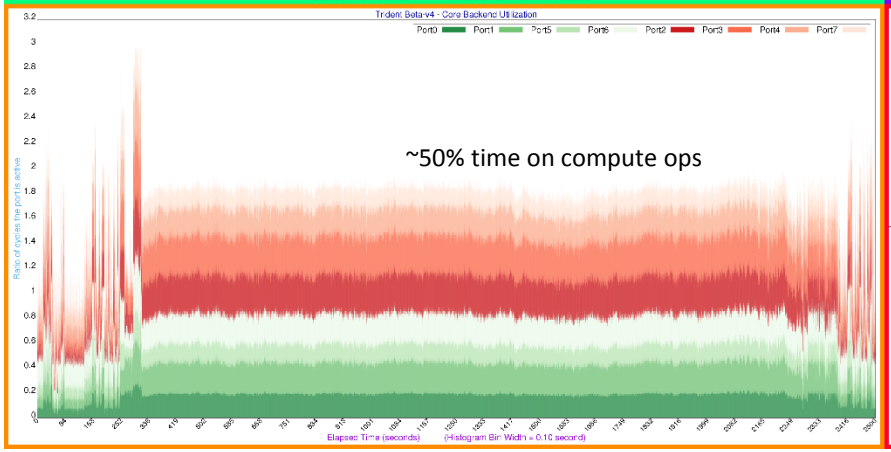
Full exploration of CPU utilisation

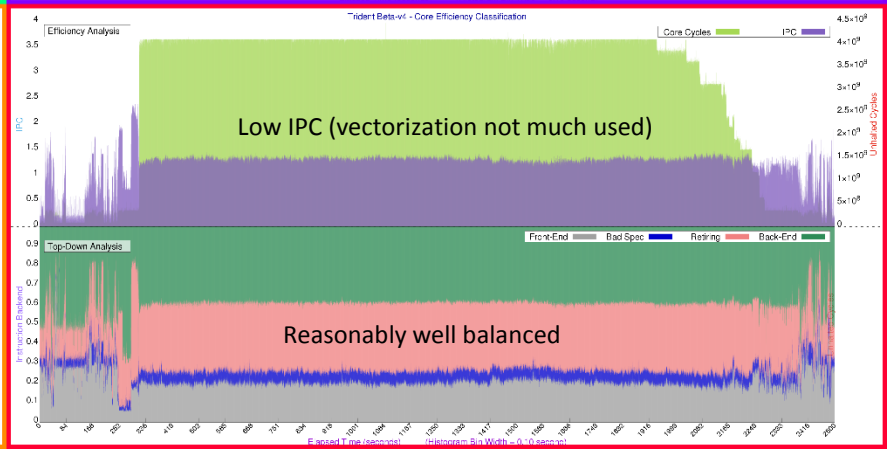


Trident plots: ATLAS Geant4









More on top-down analysis [here]

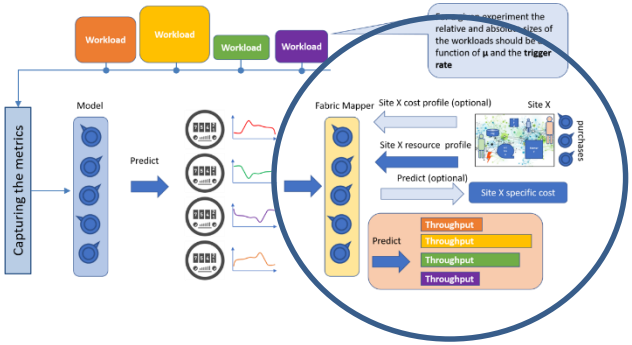
Source: Servesh Muralidharan

Resource estimation

- The initial goal was to define a common framework for modelling the computing requirements of the LHC experiments
  - Models as collection of parameters and generic and customisable calculations
  - Using as input the characteristics of the workflows
  - Reproduce with reasonable accuracy the official estimates
  - Allow to play with different scenarios
- Current status
  - A first iteration of the framework was obtained by refactoring and generalising (to a certain extent) a framework used by CMS
    - https://github.com/sartiran/resource-modeling
  - Elicited strong interest from other LHC experiments
- However, not yet clear if there is still interest for a common development
    - All experiments have refreshed their estimates using their own updated frameworks, e.g. CMS has a sophisticated Python framework and ATLAS a semi-pythonised tool (data from spreadsheet read as pandas DF)

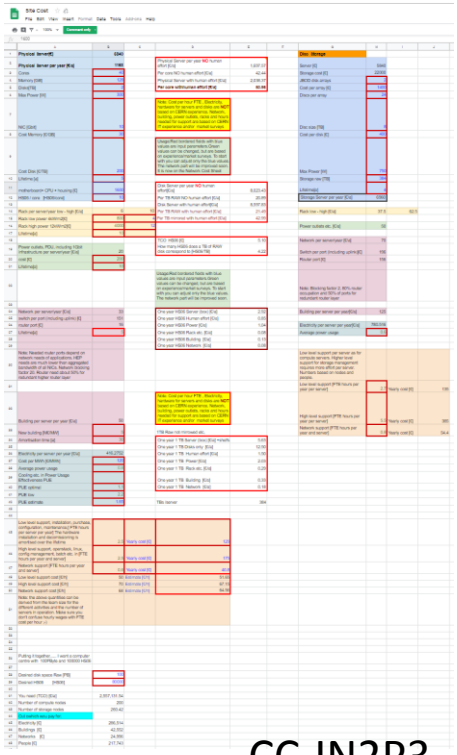
Site cost estimation models

- Develop a method to assess how well an infrastructure is matched to the needs of the experiment workloads
  - Fabric should be tuned to maximise the capacity over cost
  - Several site people in the WG went through a cost estimation exercise starting from an “example” workload
  - Actual model developed in IN2P3 and successfully applied to T1 to model yearly investment per sector
    - https://indico.cern.ch/event/304944/contributions/1672219/ (CHEP 2015)
- A model should include
  - Hardware: servers, racks, switches
  - Electricity: to run the hardware, cooling
  - Infrastructure: rooms, routers
  - Manpower

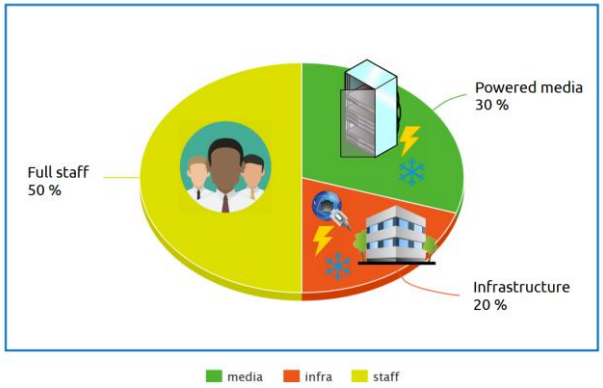


Total cost of ownership

- Two approaches being considered
- Bottom-up approach (Markus)
  - Based on Bernd Panzer's internal memos
  - Cost of hardware, power, network, building, electricity, FTEs included
  - Calculates yearly cost of 1 HS06 and of 1 TB of raw disk
  - Implemented as a self-documented spreadsheet that anybody can clone and use for his/her own site
- "Holistic" approach (Renaud)
  - Take the complete budget of the data centre
  - Remove tertiary expenses
  - Categorise per sector
  - Deduce TCO per unit of resource capacity
- The goal is to come up with a unified approach usable by all WLCG sites



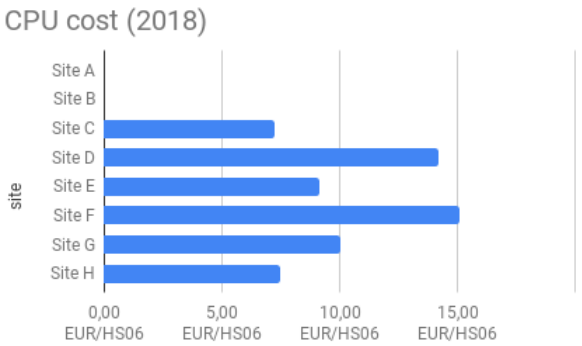
CC-IN2P3

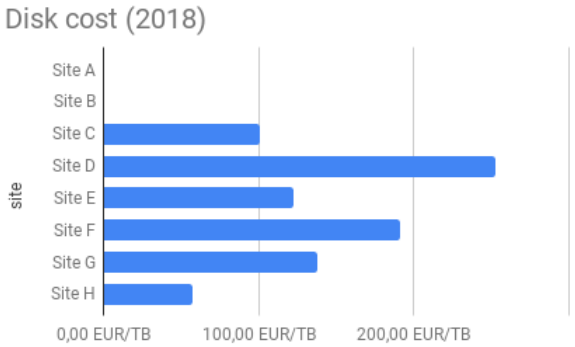


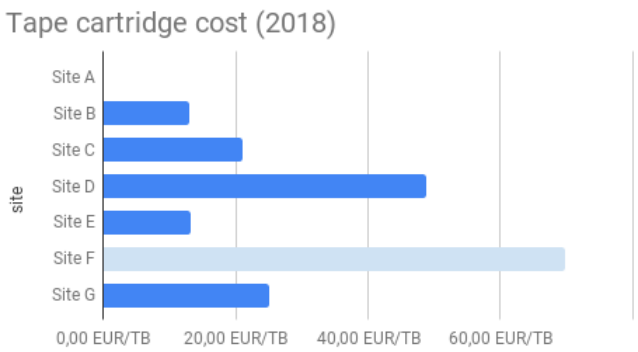
Source : R. Vernet

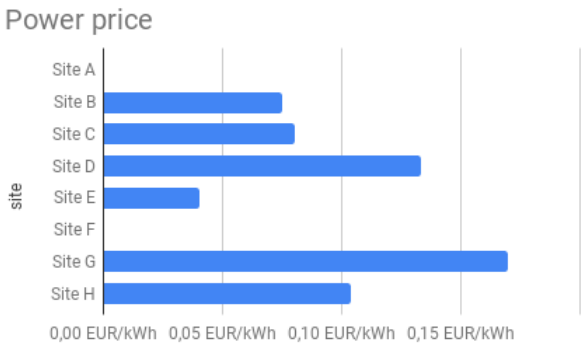
Site cost Tier-1 survey

- Renaud launched a survey among Tier-1 sites to understand their costs for CPU, disk and tape (and see the variation from site to site)
  - Questionnaire available here
  - Six Tier-1 answered (5 are ATLAS), 4 promised to answer, 1 maybe will, 3 didn't reply
  - Answers are confidential
- No firm conclusions possible with so little statistics, but already some interesting information
  - Considerable differences among sites!
- All Tier-1's should answer, but also Tier-2 sites are very welcome to participate!
  - The goal here is to have an accurate view of the spread of resource costs and their time evolution









Preliminary studies on caches at T2's

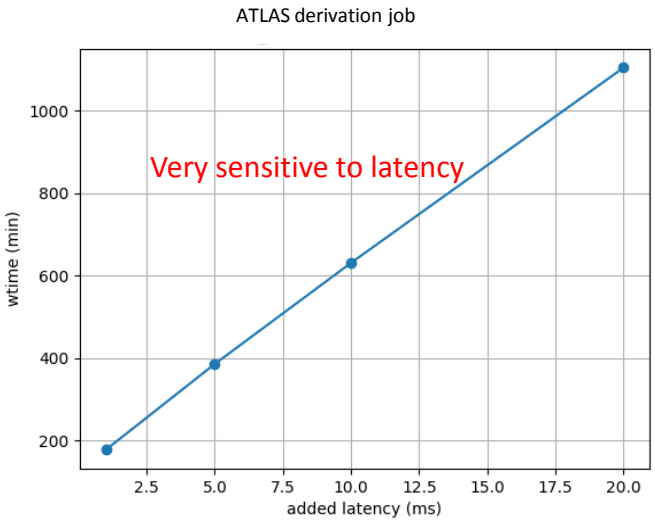
- Three-fold advantage: reduce latency at application level, reduce data transfers and reduce disk
  - But cache must scale with number of clients
- Tested throughput of ATLAS jobs with an Xcache instance at Meyrin
  - Data on WN (local), vs. remotely read from Meyrin, vs. remotely read from Wigner (with or without Xcache)
  - Latency hiding very successful

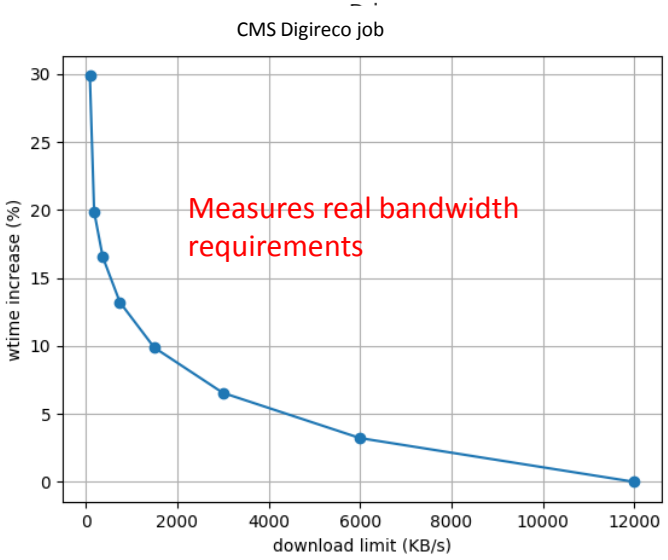
Credits: Lucrece Laura Akira

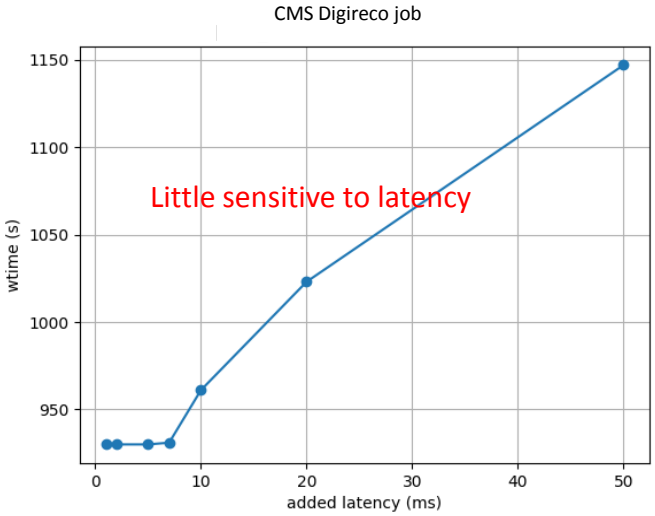
Job type	Run conditions	Run time (min)	Relative run time
DIGI-RECO	local	240	1.0
	Remote far, no cache	480	2.0
	Remote far, empty cache	262	1.09
	Remote far, pop'd cache	250	1.04
Derivation	Local	147	1.00
	Remote close	151	1.03
	Remote far, no cache	1217	8.28
	Remote far, empty cache	155	1.05
	Remote far, pop'd cache	153	1.04

Throughput vs latency: preliminary studies

- Added artificial latency and bandwidth limitations to network and studied the effect on application throughput
  - Using cgroups and iptables
  - Compared resilience to latency and bandwidth of different applications







Other areas of potential savings

- Many "small" improvements can stack to provide significant gains
  - Numbers below are based on exploratory work and are not to be taken literally – the goal is to stimulate more accurate estimates
    - Some savings could be reduced by "side effects", e.g. storage consolidation could cause loss of resources for some funding schemes

Change	Effort Sites	Effort Users	Gain
Moving cold data to tape only	Some large sites	Frameworks some	15% disk costs
Scheduling and site inefficiencies	Some	Some	10-20% gain CPU
Reduced job failure rates	Little	Some-Considerable	5-10% CPU
Compiler and build improvements	None	Little	15-20% CPU
Improved memory access/management	None	Considerable	10-15% CPU
Exploiting modern CPU architectures (e.g. vectorisation)	None	Considerable	100% CPU
Paradigm shift algorithms (ALICE HLT)	Some	Massive	Factor 2-100 CPU (GPU)
Paradigm shift online/offline data (LHCb and ALICE)	Little	Massive	2-10 CPU 10-20 Storage

Notes
- ALICE HLT: new tracking based on cellular automata on vector processors, reported 10x better on CPUs (more on GPUs)
- ALICE/LHCb online/offline: raw data not kept, immediately reconstructed on HLT, no re-reconstruction

Source: M. Schulz

Next steps

- After 1.5 years of activity it's time to re-examine the roadmap and the goals
- Some preliminary ideas
  - Archive the results of all current and future studies
  - Harmonise studies performed on data from different experiments
    - Typically popularity data
  - Make sure that such studies are well coordinated with DOMA
  - Make WLCG sites more aware of the tools being developed for calculating costs
- To be further discussed at the HOW workshop

Conclusions

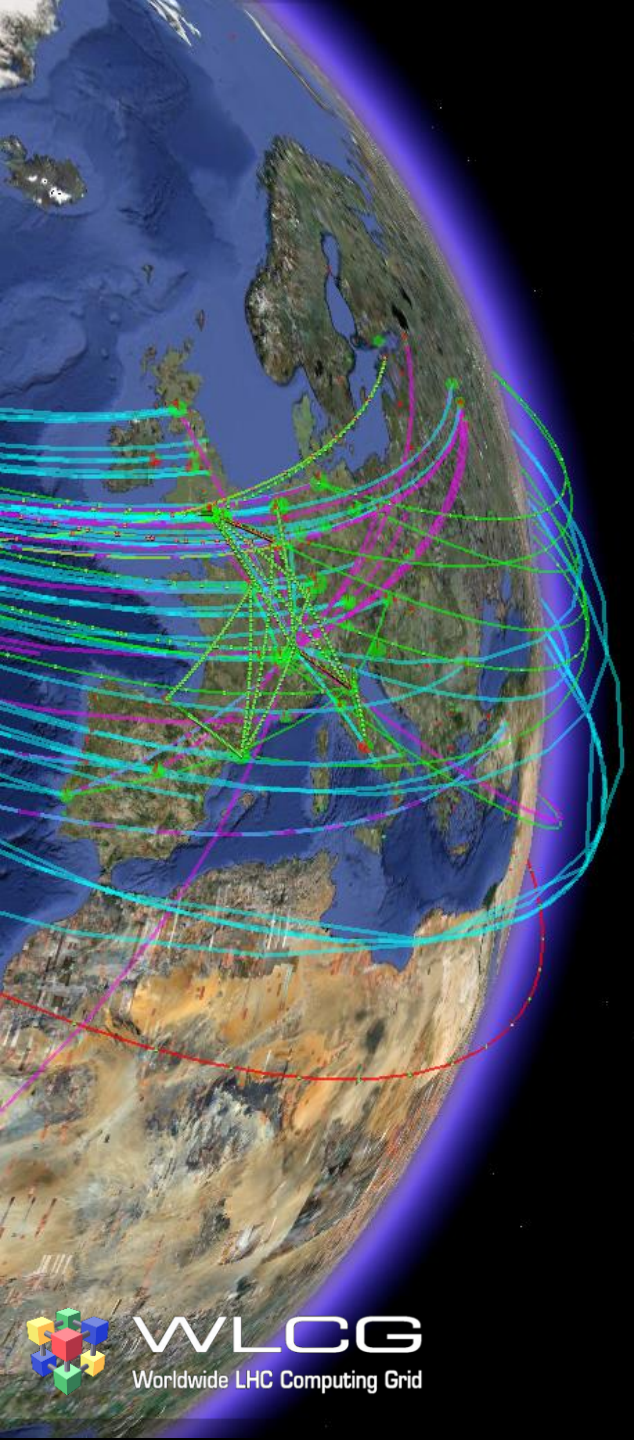
- This working group was established to improve our understanding of the performance and the cost of computing for LHC (and HEP) and its evolution
  - HL-LHC requires us to squeeze all the performance we can get at all levels
- The WG is active on many fronts and is already achieving important results
  - Reference workloads and performance analysis tools
  - Model for site cost estimation
  - Resource need estimation
  - Effect of storage caches
  - Effect of network bottlenecks
  - …
- Work is still in progress but the time scale is long…

Membership

- C Biscarat, T Boccali, D Bonacorsi, C Bozzi, R Cardoso Lopes, D Costanzo, D Duellmann, J Elmsheuser, E Fede, J Flix Molina, D Giordano, C Grigoras, J Iven, M Jouvin, Y Kemp, D Lange, H Meinhard, M Michelotto, G D Roy, A Sansum, A Sartirana, M Schulz, A Sciabà, O Smirnova, G Stewart, A Valassi, R Vernet, T Wenaus, F Wuerthwein

Further reading

- Indico category
  - https://indico.cern.ch/category/9733/

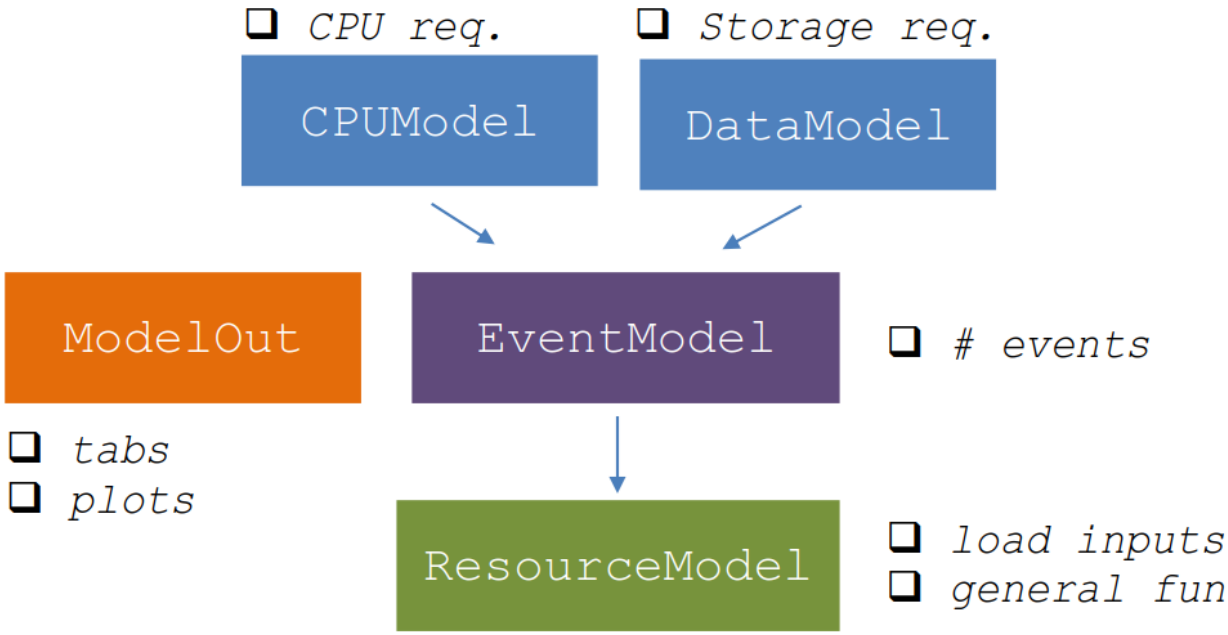


Backup slides

WLCG
Worldwide LHC Computing Grid

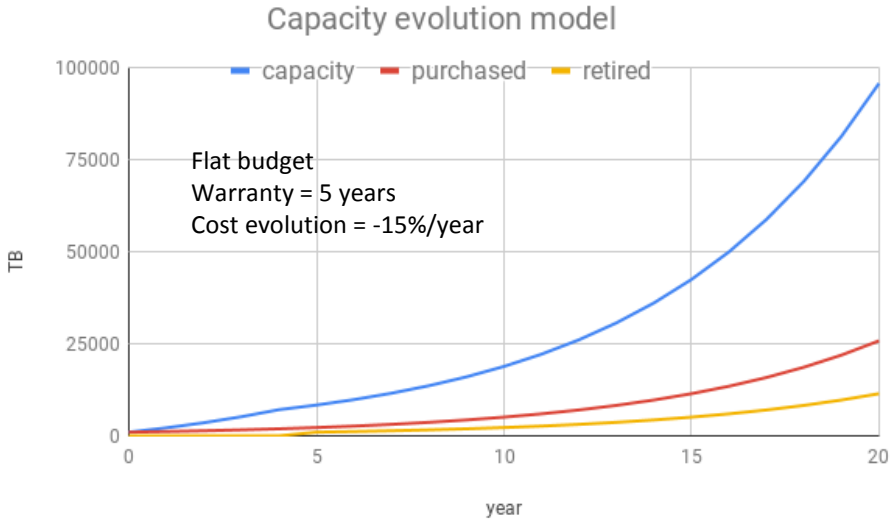
Resource estimation model

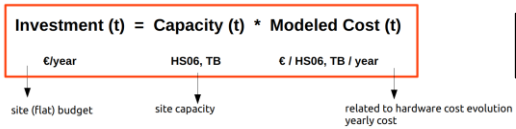
- LHC parameters (trigger rates, live fractions, shutdown years, …)
- Computing model (event sizes and processing times, …)
- Storage model (numbers of versions, replicas, …)
- Infrastructure (capacity evolution model, T1 disk and tape, …)
- No network estimates (for now)
- Extrapolation to HL-LHC relies on very uncertain estimates – the workloads don't exist yet



Example: Infrastructure costs at CCIN2P3

- Main conditions
  - Exponential decrease of costs
  - Flat budget
    - Used for capacity replacement + capacity increase
  - Replace hardware when warranty expires
- Verification of the model
  - Compare modeled budget with reality
    - Excellent match for CPU and disk
    - Less precise for tape
- Power consumption
  - Assume exponential decrease of unitary power consumption and exponential variation of power prices
- The model can be applied to any existing WLCG site
  - Work ongoing to confidentially collect relevant data from Tier-1 sites

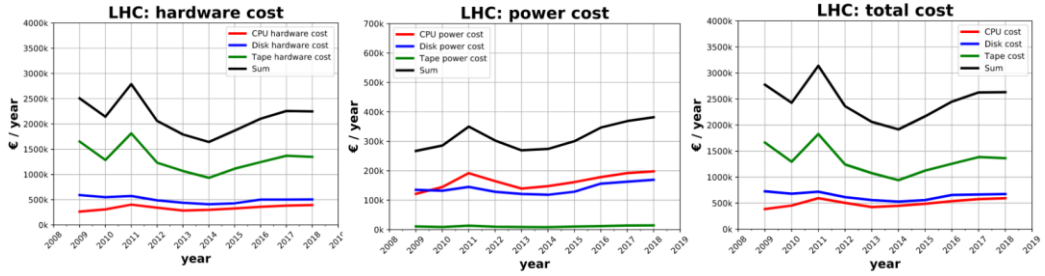




$$c^*(t) = c(t)\frac{r}{1-(1-r)^{\tau}}$$

c^* = modeled cost
c = real cost
τ = warranty time
r = cost decrease rate

hardware cost + power cost = total cost



Source: R. Vernet

Storage Impact: examples

- Another example of what we do: estimating alternative scenarios. Examples:
- Effect of concentrating persistent storage at a small number of large sites ("data lake") and use caches at T2's?
  - Manpower for managed storage: ~2.5 FTE at T1's, ~0.5 FTE at T2's, weakly dependent on size, for cache storage ~0.1 FTE
  - 110 FTE → 60 FTE (-45%)
- Replace redundant storage with non-redundant disk everywhere?
  - Assuming that lost data can be re-generated, what is cheaper – the CPU to regenerate the lost data, or buying enough disk for another copy?
  - HDD failure rates in EOS ≈ 1%/year → ≈ 2 PB lost/year for a major experiment
  - Yearly, 4 HS06 cost about the same as 1 TB (at a major site)
  - 1 MC AOD event costs ≈ 4000 HS06 · s and is ~400 kB
    - Regenerating the 2 PB of AOD lost → $5 \cdot 10^9$ events → $20 \cdot 10^{12}$ HS06 · s = 630 kHS06 · y
    - CPU needed costs the same as 160 PB of raw disk (~80% of the cost of full data redundancy)
  - Buying CPU instead of disk?
    - "pessimistic" estimate, as lost MC might be on tape or replicated at other sites
    - May even decide to regenerate only when data is required again