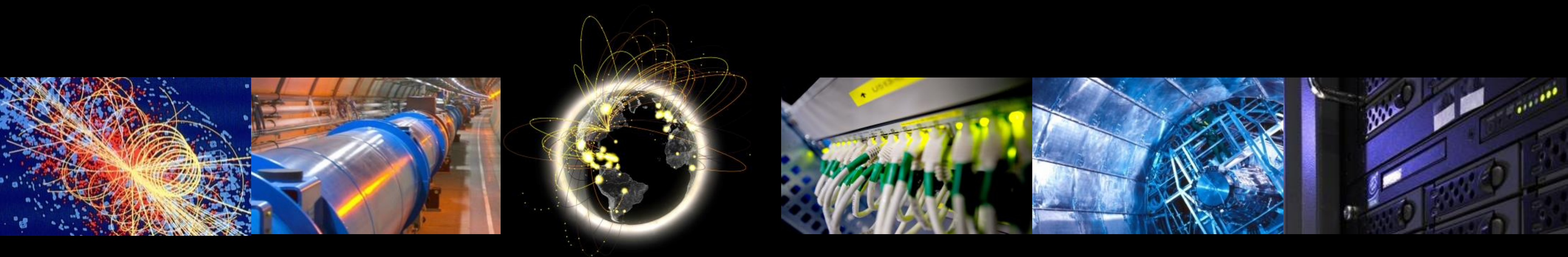
System Performance and Cost Modelling in LHC computing

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

CHEP 2018
Sofia, 9-13 July 2018



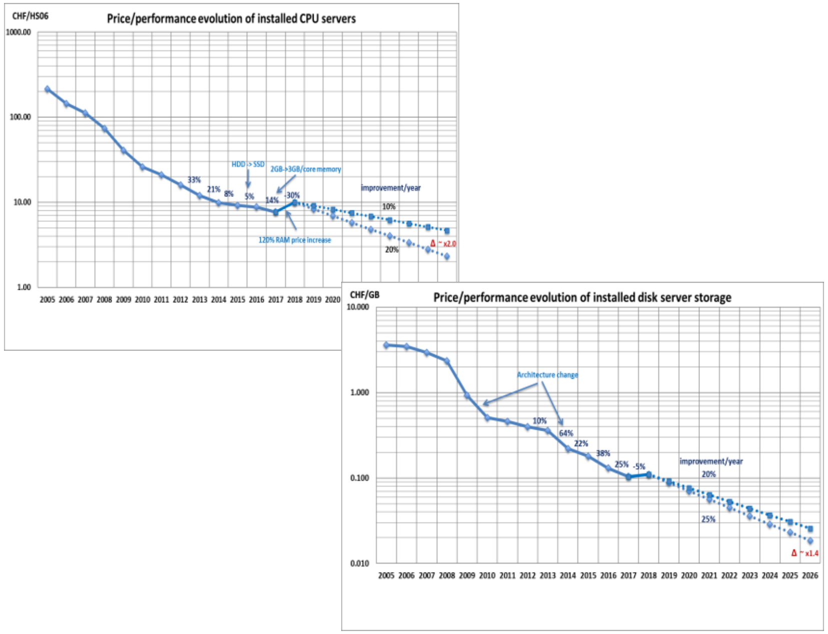
Worldwide LHC Computing Grid

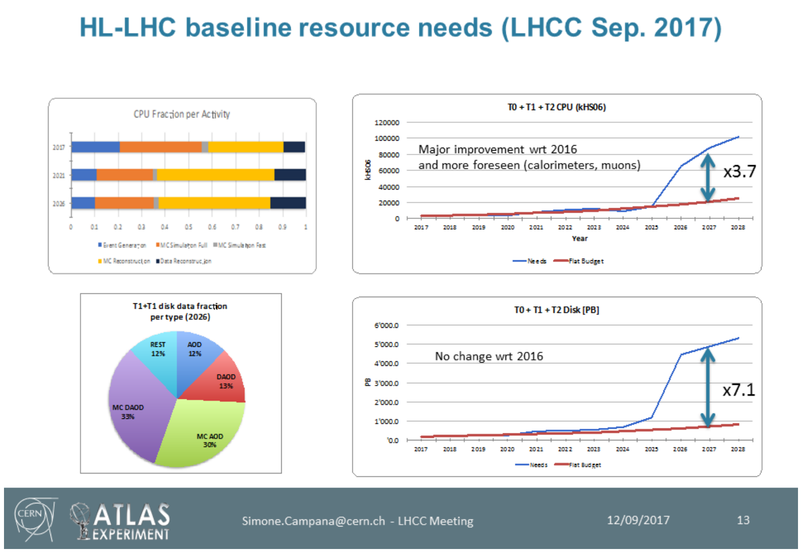
HEP Software Foundation

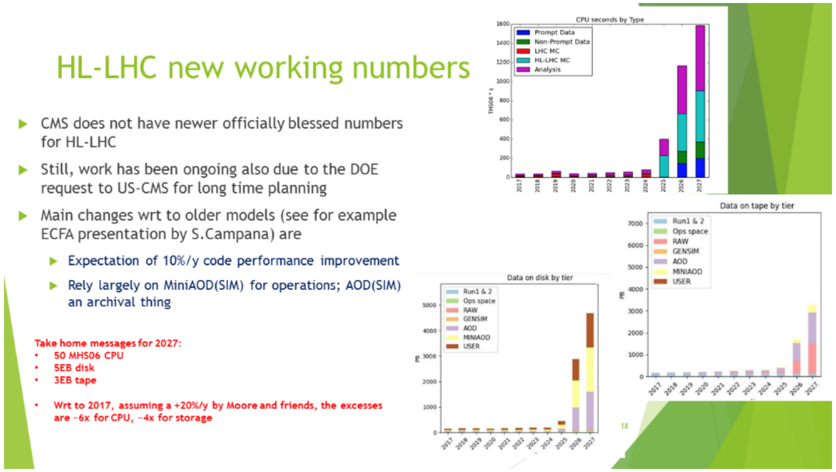
CHEP 2018
Sofia, Bulgaria

The High Luminosity challenge

- Despite ongoing efforts, we do not know yet how we will manage to process HL data with the expected levels of funding
- 10x increase in trigger rates, NLO & NNLO, 5x increase in pile-up
  – The latter involves >> linear growth in reconstruction time
- Price/performance advances slowing down
  – 20% yearly gains are very difficult
- CPU and disk short by a factor ≈ 5
  – Assuming no “revolutionary”changes
- Strong need to quantitatively understand our efficiency and how we can optimise performance







CMS

WLCG Worldwide LHC Computing Grid

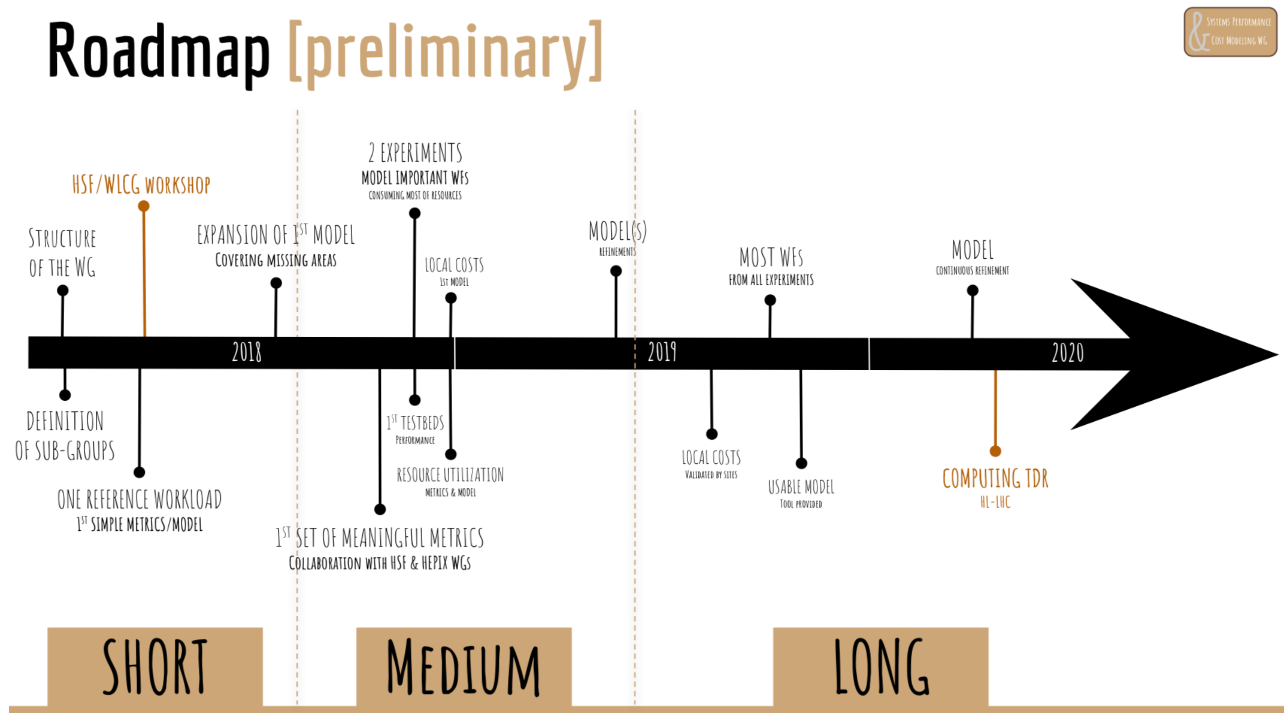
The Working Group

- WLCG and HSF joined forces to study how we can achieve a more cost-effective computing on the Run3/4 timescale
  - Start by developing a deep understanding of current workloads, resource utilization, and their impact on site costs
  - Proceed to explore future scenarios, estimate possible improvements in efficiency (in software, infrastructure and computing systems)
  - Develop tools and methods to do the above, that can be used in the community
  - At the same time, establish a "culture of performance"
  - Site cost cannot be compared but locally optimised
  - Active participation by experiments, sites and IT experts
    - Conveners: J Flix, M Schulz, A Sciabà
    - About 35 active members → wlcg-SystemsPerformanceModeling@cern.ch
    - Links with HEPiX benchmarking working group
    - Web site: https://twiki.cern.ch/twiki/bin/view/LCG/WLCGSystemsPerformanceModeling
    - Meetings: https://indico.cern.ch/category/9733/

WLCG Worldwide LHC Computing Grid

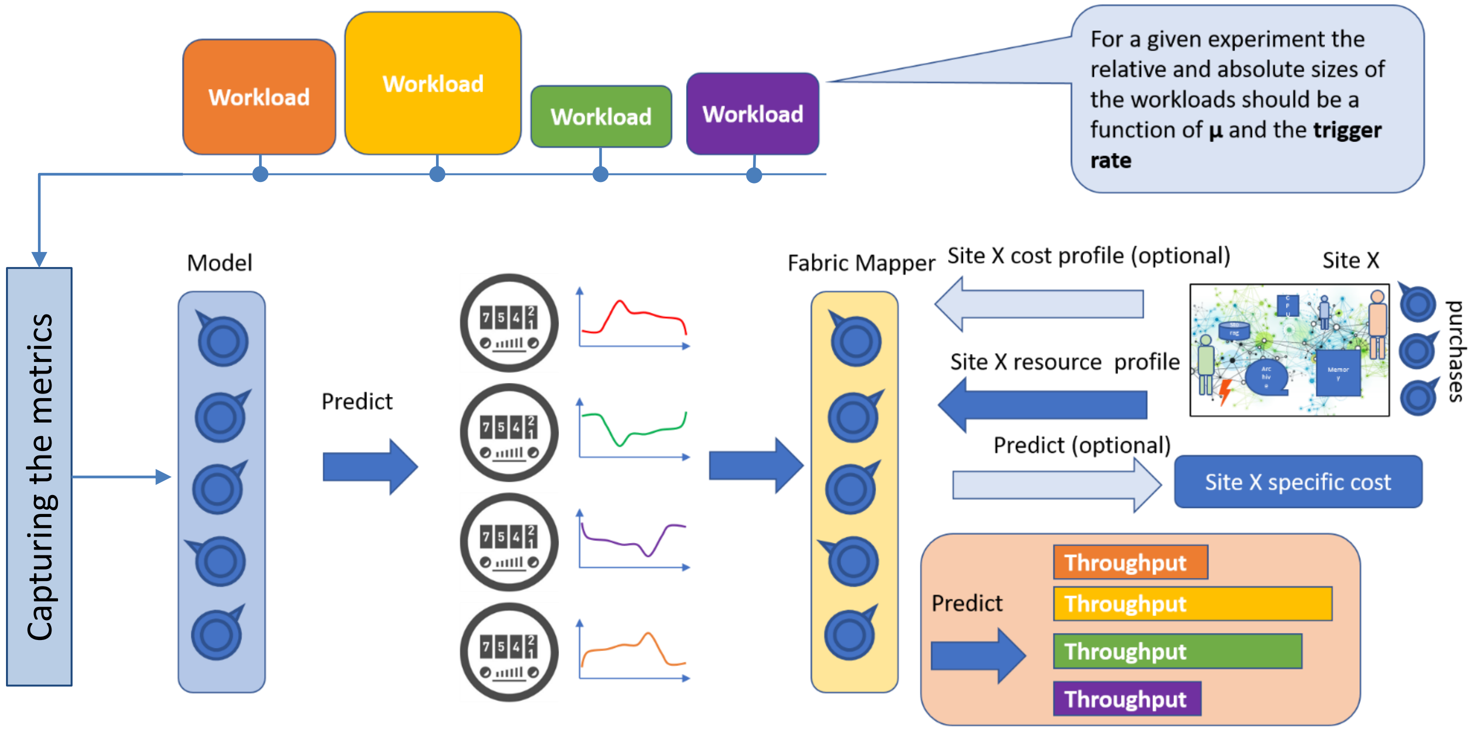
Areas of work

- Several goals have been identified for the short, medium and long term and some are well under way or even completed
  - Identify representative experiment workloads that can be run in a controlled environment and package them for easy distribution
  - Define which metrics best characterise such workloads
  - Set up a distributed testbed to run tests
  - Establish a common framework for estimating resource needs
  - Define a process to evaluate the cost of an infrastructure as a function of the experiment requirements



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, 95th percentile values, etc.)



Current metrics

- Started with an already comprehensive list of basic metrics
- Will expand / contract as needed – work in progress
- The goal is to have the smallest amount of parameters that describes as completely as necessary the workloads

I/O

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	

CPU

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

Memory

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)

Network

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

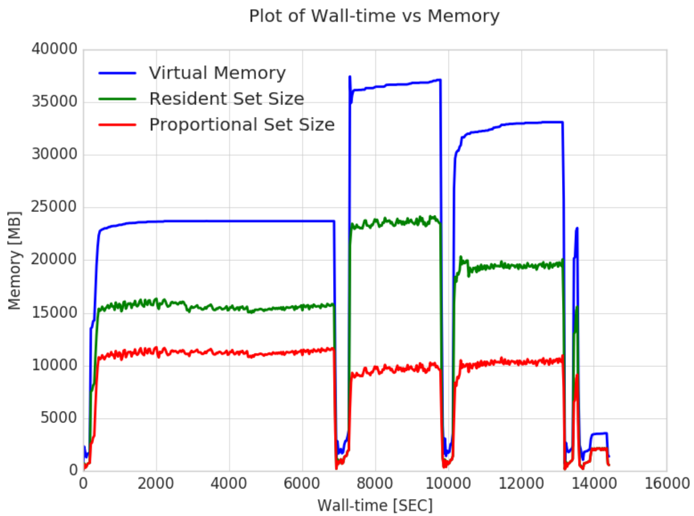
WLCG Worldwide LHC Computing Grid

Metrics measurement

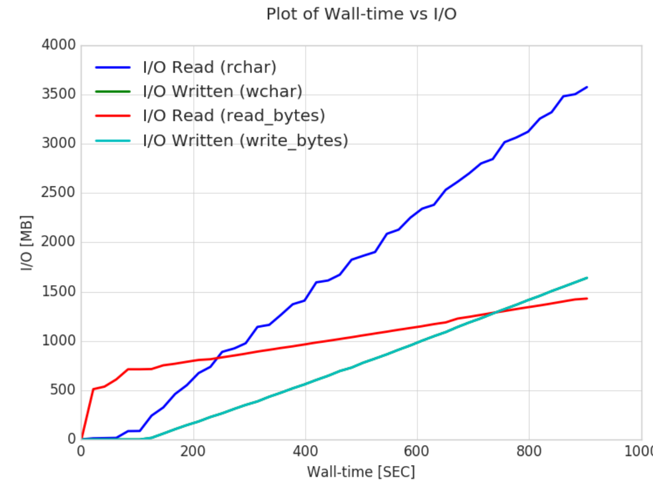
- PrMon is a tool to monitor resource usage of a process tree
  - Derived from the ATLAS MemoryMonitor
  - https://github.com/HSF/prmon
  - It includes most of the previously listed metrics (from /proc)
    - VMEM, RSS, PSS
    - rchar/wchar (bytes read/written by the process) , read_bytes/write_bytes (bytes read/written from/to the storage layer)
    - User time, system time, wallclock time
    - rx_bytes, tx_bytes, rx_packets, tx_packets
  - Actively worked on
- Trident
  - Measures CPU, IO and memory utilisation based on hardware counters
  - Very detailed, almost no overhead
  - See Servesh' and David's poster "Trident: A three pronged approach to analysing node utilisation" (link)
- Collection of reference workloads from the LHC experiments
  - Event generation, Geant4 simulation, digitisation, reconstruction, derivation steps
  - Local file access or remote access via xrootd
- Making power and complex tools accessible for users and site managers on all levels

WLCG Worldwide LHC Computing Grid

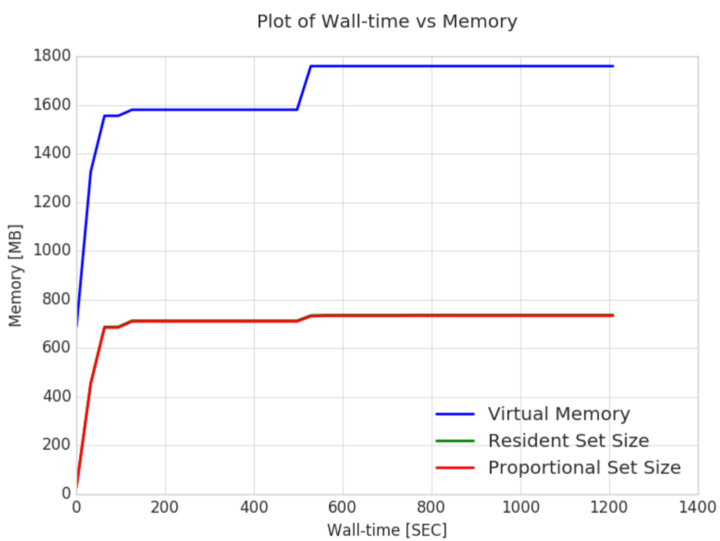
PrMon monitoring plots: examples



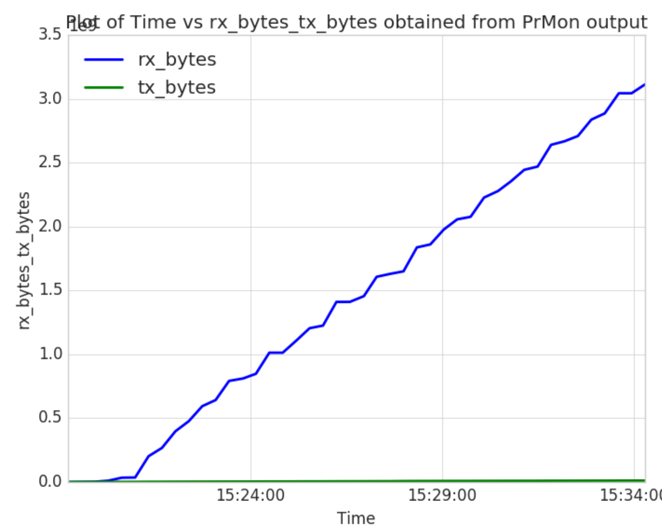
ATLAS Digi Reco - memory



CMS DIGI - IO



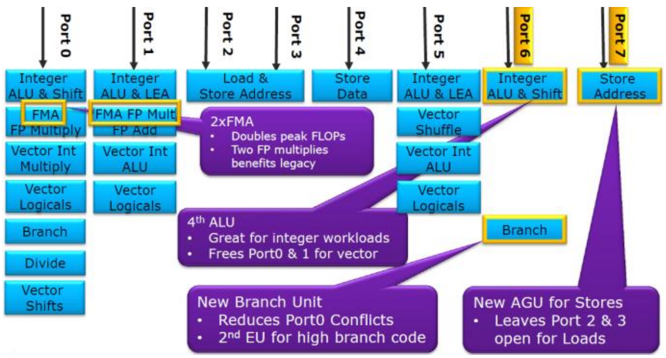
ALICE sim+reco - Memory

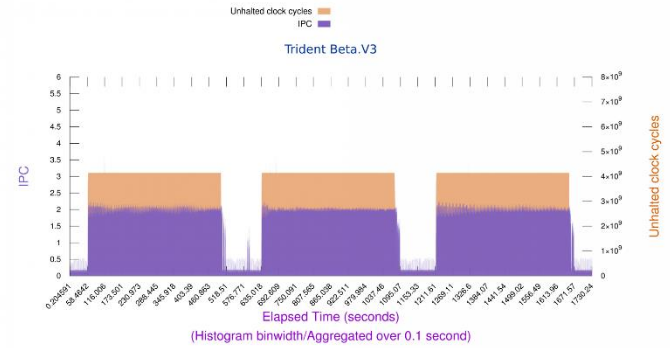


CMS DIGI - Network

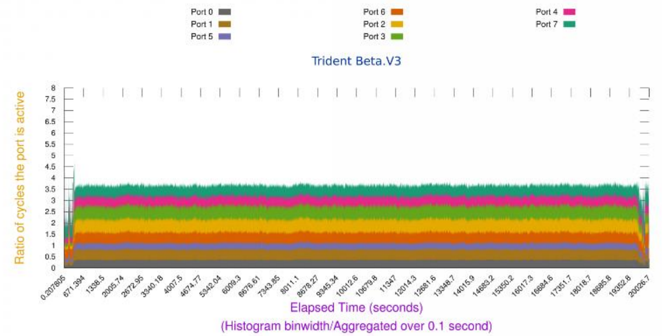
Measuring performance with Trident

- Several metrics calculated
  - CPU: IPC, top-down analysis (time spent on front-end/back-end, retiring/bad speculation), execution unit port utilisation
  - Memory: bandwith usage, transaction classification (page-hit, page-empty, page-miss)
- Can be used to see how workloads differ (or resemble) each other and the benchmarks we use (HS06, SpecCPU2017?)
- CPU counters are a powerful (but complex) tool and Trident makes them accessible





IPC for HS06 namd



ATLAS G4 CPU port utilisation

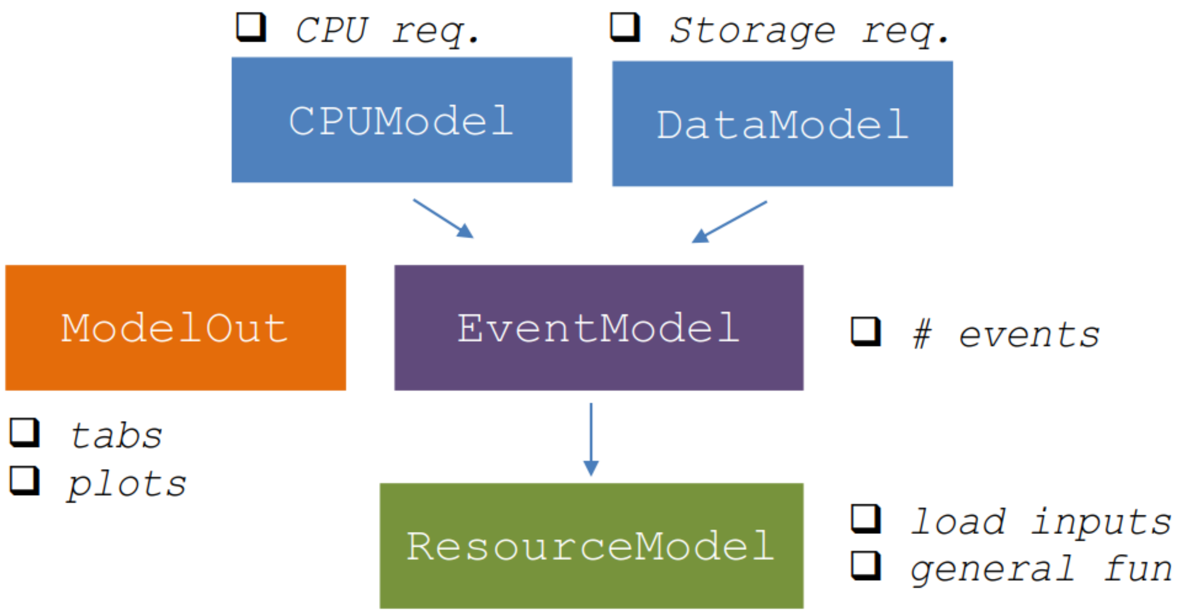
Resource estimation (1/2)

- The goal is to define a common framework for modelling the computing requirements of the LHC experiments
  - Models as collection of parameters and standard calculations, to be as generic and customisable as possible
  - Takes as one of its inputs the characteristics of the workflows
  - Reproduce with reasonable accuracy (but not supersede!) the official estimates from the experiments
  - Allow to play with different scenarios to explore potential gains
- 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
    - Agreed as a common basis for future development

WLCG Worldwide LHC Computing Grid

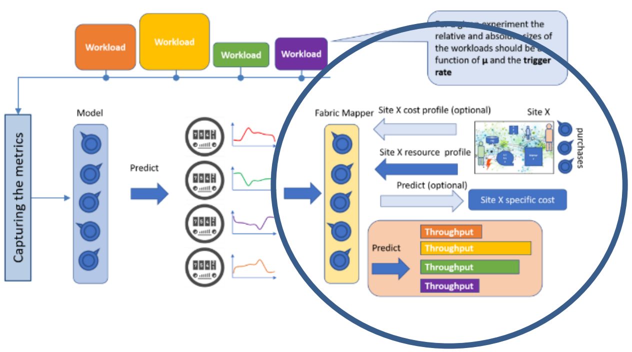
Resource estimation (2/2)

- LHC parameters (trigger rates, live fractions, shutdown years, …)
- Computing model (event sizes and processing times, improvement factors, …)
- Storage model (numbers of versions, replicas, …)
- Infrastructure (capacity model, T1 disk and tape, …)
- Time granularity is yearly
  - While resource needs vary over the year
- No network estimates (for now)
- Extrapolation to HL-LHC relies on very uncertain estimates – the workloads don’t exist yet



Site cost estimation models

- Develop a method to assess how well an infrastructure is matched to the needs of the experiment workloads
  - Capacity can be matched to local cost
  - Fabric can 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
    - The goal is not to compare sites, but to provide tools to optimise expenditure
  - 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)
- Main sectors
  - Hardware: servers, racks, switches
  - Electricity: to run the hardware, cooling
  - Infrastructure: rooms, routers
  - Manpower



Infrastructure costs at CCIN2P3: hardware

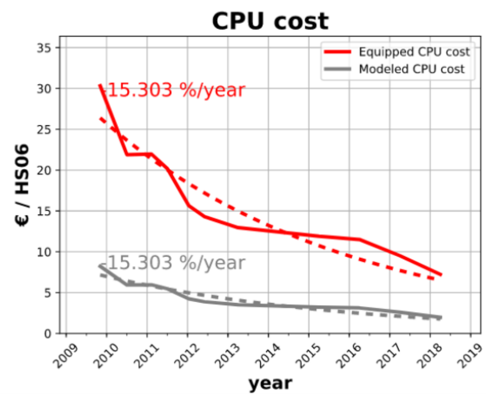
- Main conditions
  - Exponential decrease of costs
  - Flat budget
    - Used for capacity replacement + capacity increase
  - Replace hardware when warranty expires

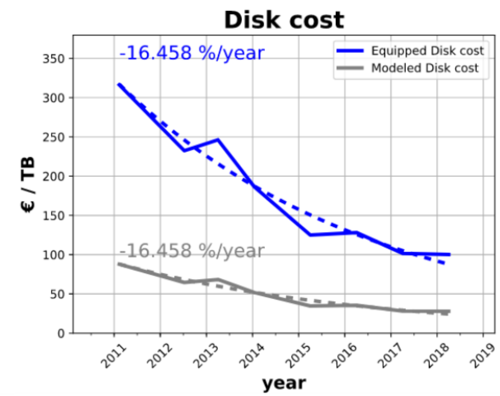
Investment (t) = Capacity (t) * Modeled Cost (t)

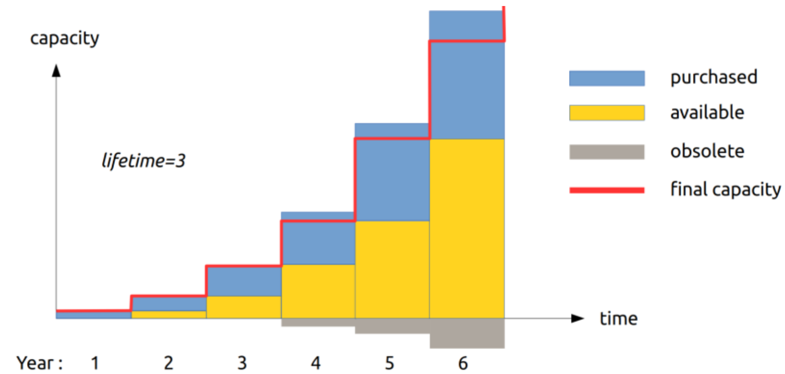
Investment (t)	Capacity (t)	Modeled Cost (t)
€/year	HS06, TB	€ / HS06, TB / year
site (flat) budget	site capacity	related to hardware cost evolution yearly cost

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

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





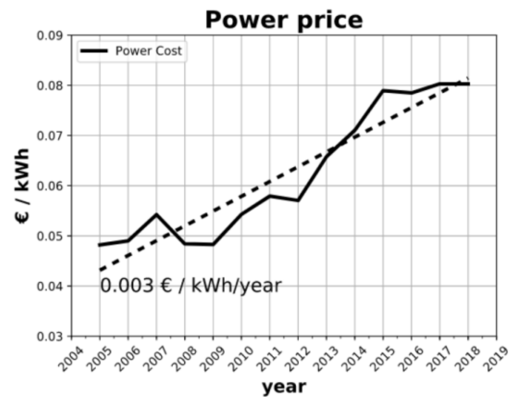


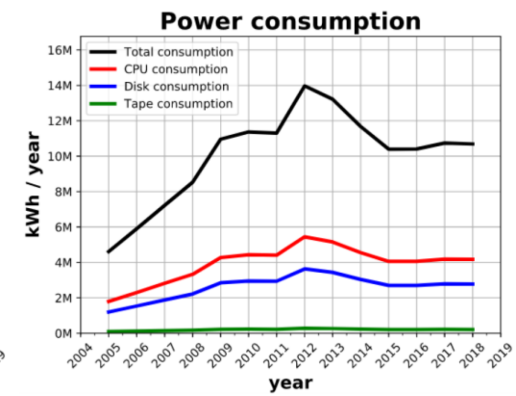
Source: R. Vernet

- Model predictions checked within 20% of reality
  - Most of the uncertainty comes from tape

Infrastructure costs at CCIN2P3: power and total

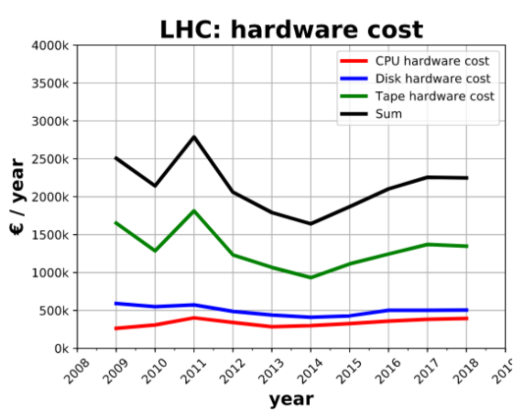
- Power consumption cost changes more difficult to predict
- Predicting future costs is possible
- Other sites are invited to use the same principles

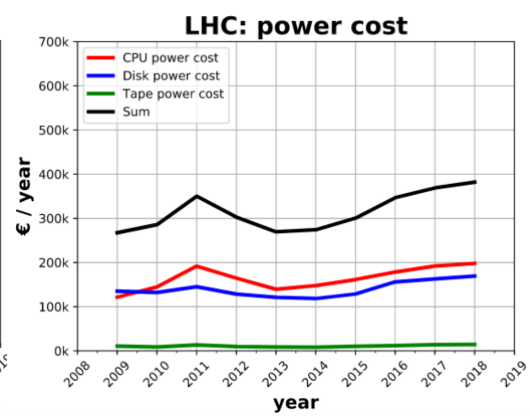


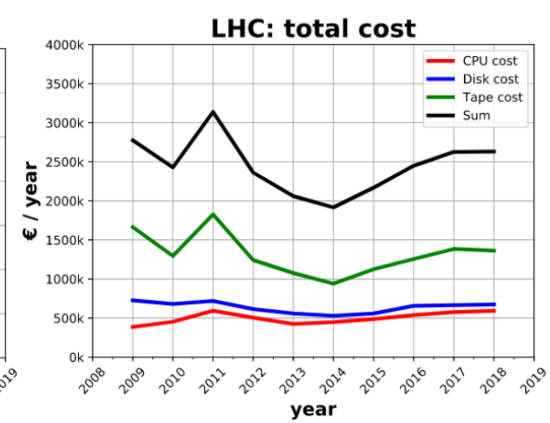


- CPU: 39%
- Disk: 26%
- Tape: 2%
- Rest: 33%

hardware cost **+** **power cost** **=** **total cost**







Areas of potential savings

- Many “small” improvements can stack to provide significant gains
- A quantitative estimation is highly desirable
  - OK to quantify not very realistic scenarios as it still provides a measure of the “gap”
  - 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”. Eg.: storage consolidation could cause loss of resources for some funding schemes → another argument for advocating a careful evaluation
    - https://indico.cern.ch/event/704519/

Change	Effort Sites	Effort Users	Gain
managed storage on 15 sites + caches	Some on large sites/gain on small sites	little	40% decrease in operations effort for storage
Data redundancy by tape backup	Some large sites	Frameworks some	30% disk costs
Reduced data replication and cold data	little	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% CPU
Exploiting modern CPU architectures	None	Considerable	100% CPU
Paradigm shift algorithms (ALICE HLT)	Some	Massive	Factor 2-100 CPU
Paradigm shift online/offline data (LHCb and ALICE)	Little	Massive	2-10 CPU 10-20 Storage

- Cumulative evolutionary changes
  - Storage costs: -45% less cost
  - Site operations for storage: -40%
  - CPU: +200% throughput

Source: M. Schulz

WLCG Worldwide LHC Computing Grid

HL-LHC predictions

- What will change?
  - Running conditions (luminosity, pile-up, trigger rate)
  - Event generation (LO + NLO + NNLO)
  - Detector simulation (full + fast simulation)
  - Detectors (some completely new, with much more fine-grained information)
  - Reconstruction (new algorithms, momentum cuts)
  - Analysis (new data formats)
  - Software (new algorithms, machine learning, vectorization)
  - Fabric (many-core CPUs, GPUs, accelerators)
- Need to develop sensible models for future workloads
- Initially, lots of unknowns, huge uncertainties
- Create “fake” workloads?

HL-LHC computational complexity

- Event size
  - Linear in μ, apart from the most compact analysis formats
- Reconstruction time
  - Dependency with μ is linear for
    - Calibration
    - Pattern recognition for low μ
    - Linking of tracks and calorimetric objects for low μ
  - It will be exponential for high μ for
    - Pattern recognition
  - Overall, it can be modelled as $t(\mu)=a\mu+be^{\mu-\mu crit}$
- Simulation time
  - Event generation and simulation independent from μ
  - Digitisation linear in μ
- Analysis time
  - Independent from μ

WLCG Worldwide LHC Computing Grid

Collaborations

- The cost model WG is by construction tightly connected with other groups and communities
- HEPIX
  - Mainly on benchmarking and fabric technology evolution
- WLCG DOMA (Data Organisation Management and Access)
  - Aims at greatly reduce the cost of storage by consolidation, caching, rationalization of protocols and services
- WLCG Archival Storage Working Group
  - Improve understanding of the cost of tape archives
- CERN EP
  - R&D on software to meet the challenges of Run3 and HL-LHC
- HSF
  - Collaborating on software optimization and tools

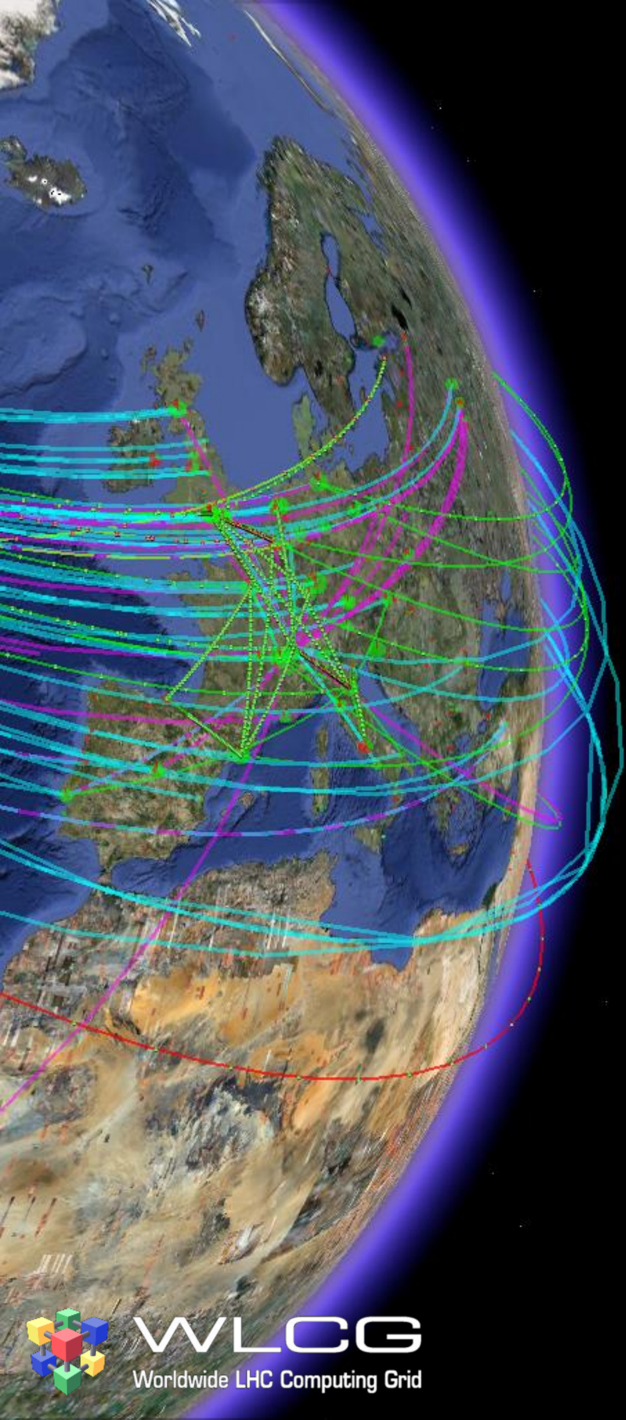
Conclusions

- The WLCG/HSF systems performance working group was established to improve our understanding of the evolution of the cost of computing for LHC (and HEP)
  - 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
  - Framework on resource need estimation
- Work is still in progress but the time scale is long
  - One of the biggest challenges is to produce reliable estimates for HL-LHC
- Several interactions with many other activities and bodies in the community
  - Active participation from more people is always welcome and encouraged!

WLCG Worldwide LHC Computing Grid

Author list

- C Biscarat, T Boccali, D Bonacorsi, C Bozzi, R Cardoso Lopes, D Costanzo, D Duellmann, J Elmsheuser, E Fede, J Flix Molina, A Forti, M Gasthuber, 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



BACKUP SLIDES

Worldwide LHC Computing Grid

Workload metric summary

Type	Events	Duration (hours)	CPU efficiency (%)	PSS/process (MB)	Disk read rate (kB/s)	Disk write rate (kB/s)	Network traffic (kB/s)
ATLAS sim	1000	9.4	98	500	140	70	negligible
ATLAS digi reco	2000	4.0	84	1500	2600	1900	negligible
ATLAS derivation	?	2.3	96	1400	5600	580	negligible
CMS GENSIM	500	0.5	97	200	600	240	negligible
CMS DIGI premix	500	0.25	58	400	1600	1900	3300
ALICE pp	1	0.3	100	700	600	60	negligible

WLCG Worldwide LHC Computing Grid