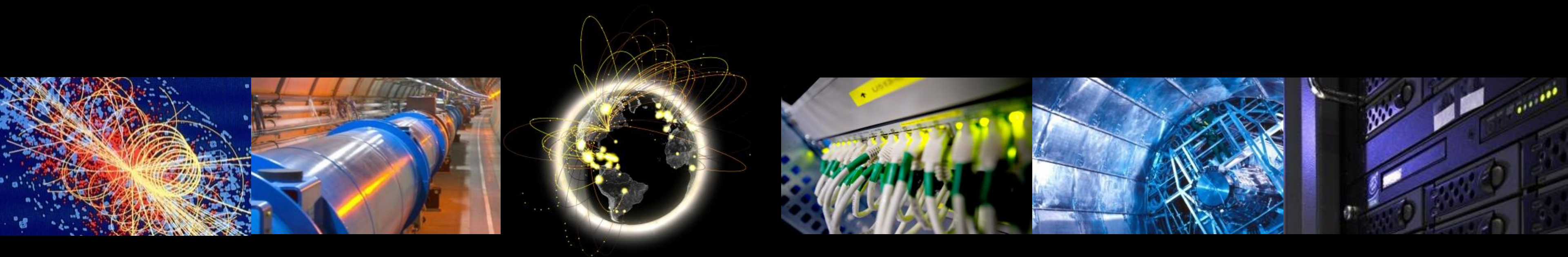
Cost and system performance modelling in WLCG and HSF: an update

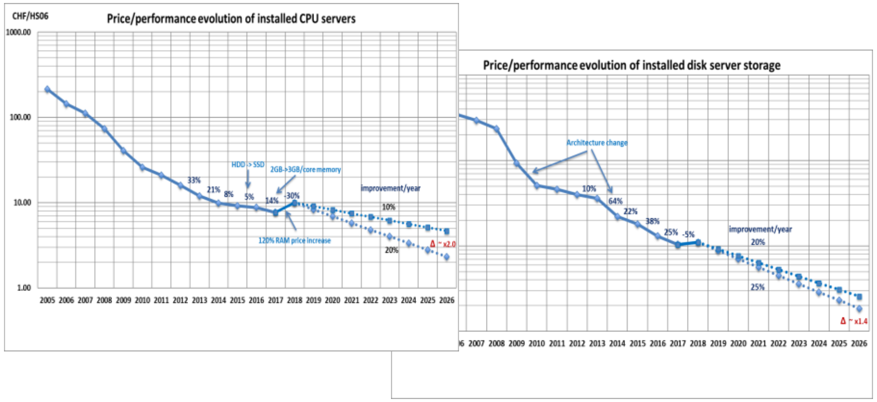
Andrea Sciabà
on behalf of the HSF/WLCG Systems performance and cost modelling WG

HEPiX Autumn/Fall 2018 Workshop
Barcelona, 8-12 October 2018

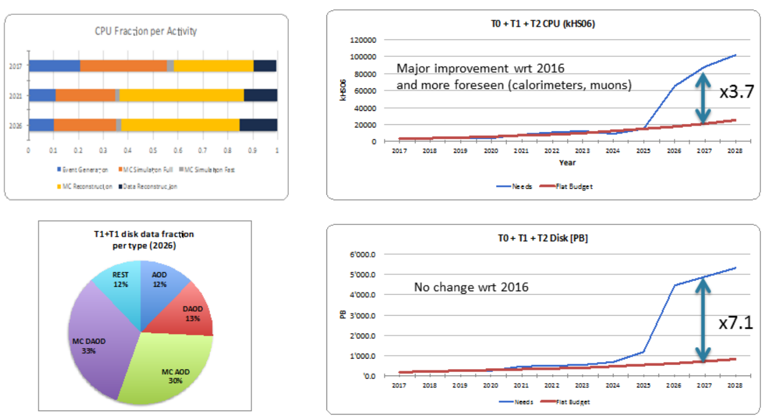


The High Luminosity challenge

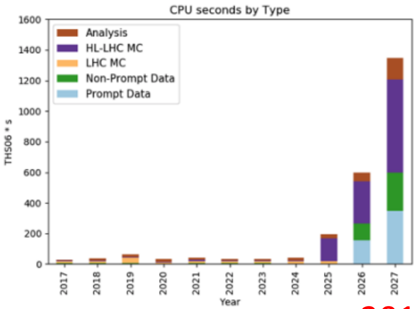
- There is still a significant gap between the estimations of needed and available resources
  - 10x increase in trigger rates, 5x increase in pile-up, NLO & NNLO, …
  - Price/performance advances slowing down, 15-20%/year at best
- CPU and disk short by a factor ≈ 5
  - Even if the gap is reducing! CMS disk estimates are 2x lower than one year ago
- Strong need to quantitatively understand our efficiency and how we can optimise performance

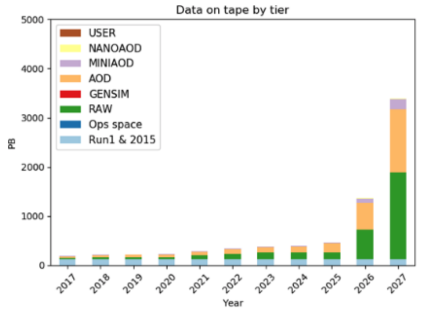


HL-LHC baseline resource needs (LHCC Sep. 2017)

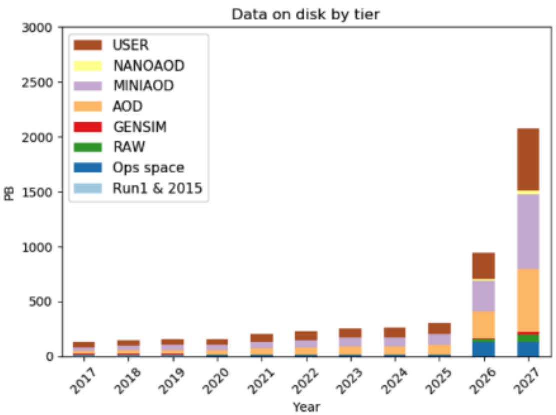


ATLAS EXPERIMENT Simone.Campana@cern.ch - LHCC Meeting 12/09/2017 13





2018 estimates



Source: T. Boccali

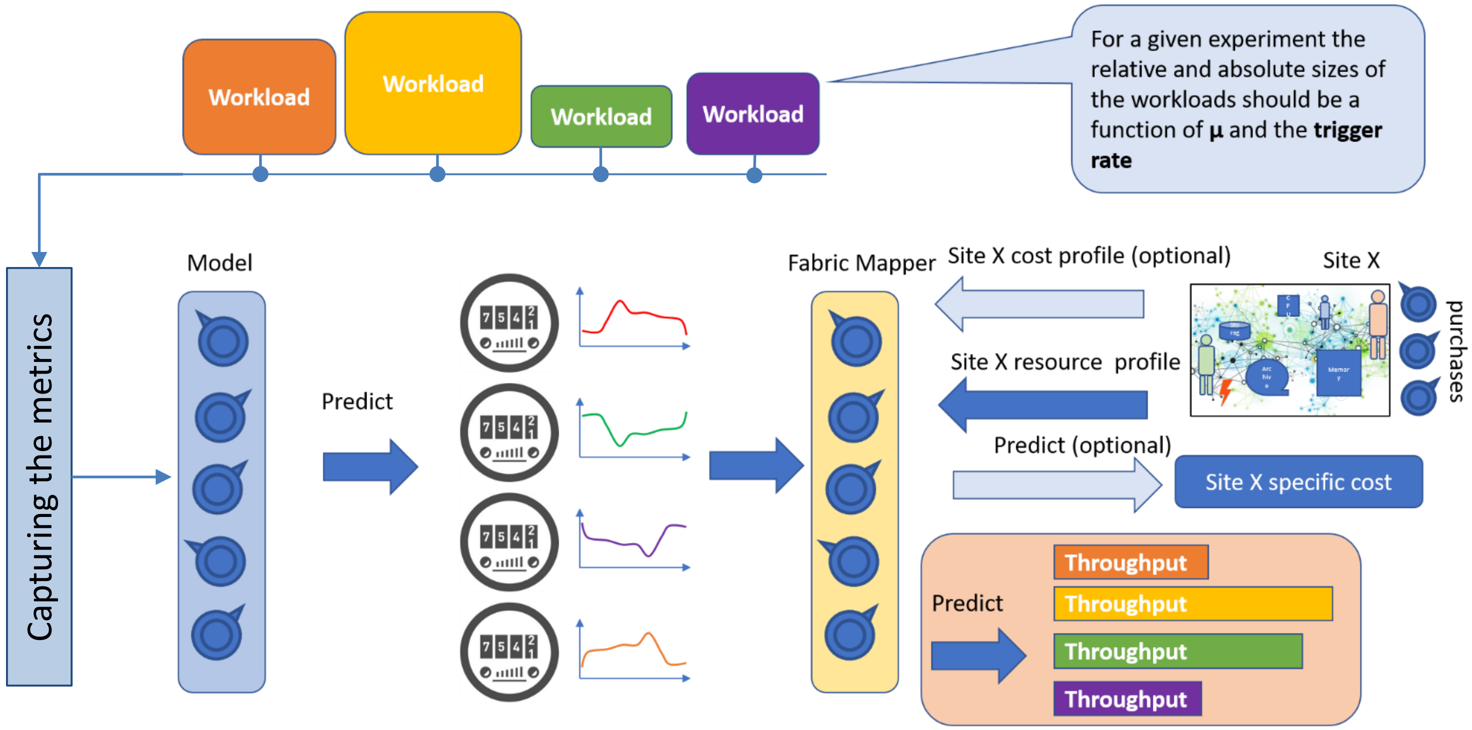
The Working Group

- 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
- Current areas of work and goals
  - Identify representative experiment workloads to run in a controlled environment
  - Define which metrics best characterise such workloads
  - Establish a common framework for estimating resource needs
  - Define a process to evaluate the cost of an infrastructure
  - Measure the impact of new storage configurations on applications and costs
- Several developments since the previous HEPiX workshop

WLCG Worldwide LHC Computing Grid

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

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

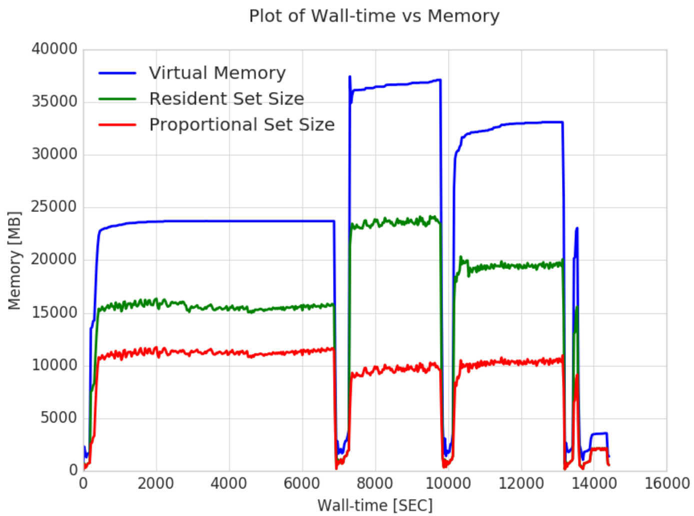
Memory

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

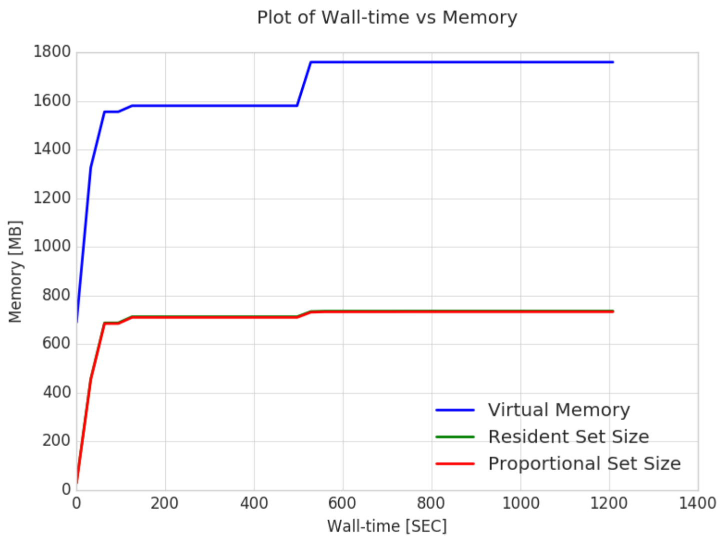
Metric	Type	Source	Scope	Command	Insight	Comments
Network usage	gauge	/proc/net/dev	global	Possible update to prmon	Aggregate Tx/Rx bytes to assess total network load	As /proc/net/dev is global some method of isolating a process is necessary to assess accurately (containers/namespaces?)
Network rates	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

PrMon monitoring plots: examples

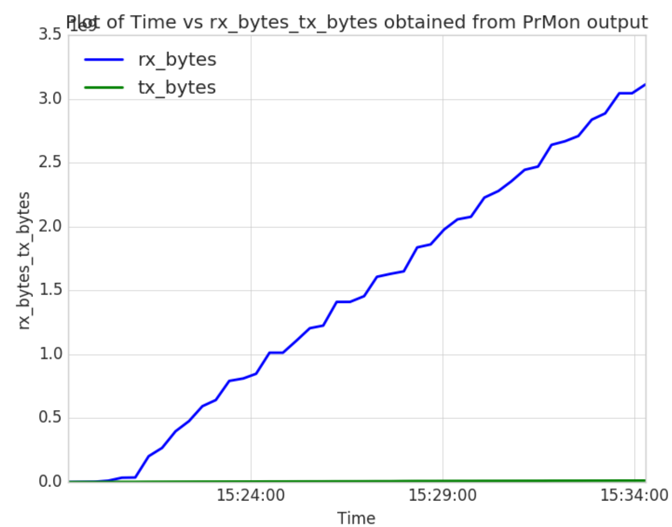


ATLAS Digi Reco - memory

CMS DIGI - IO



ALICE sim+reco - Memory

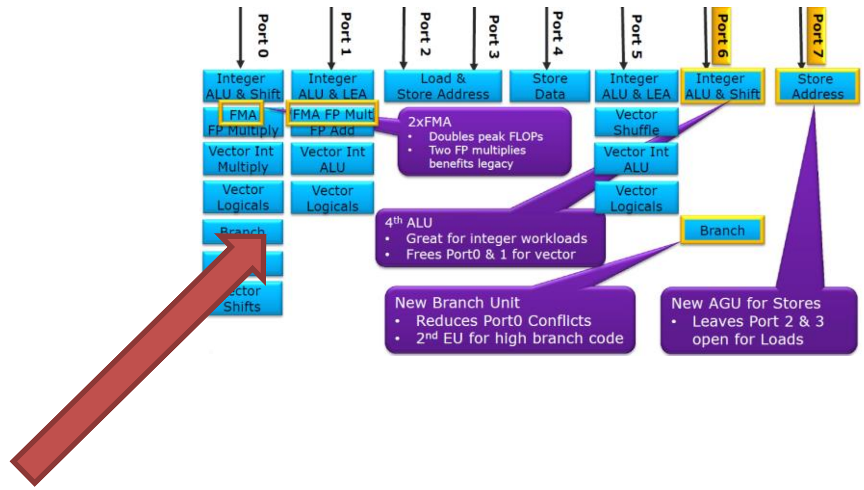


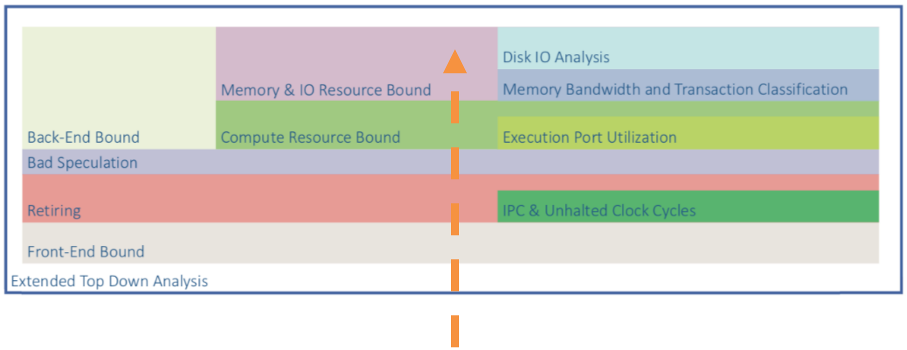
CMS DIGI - Network

WLCG
Worldwide LHC Computing Grid

Measuring performance with Trident

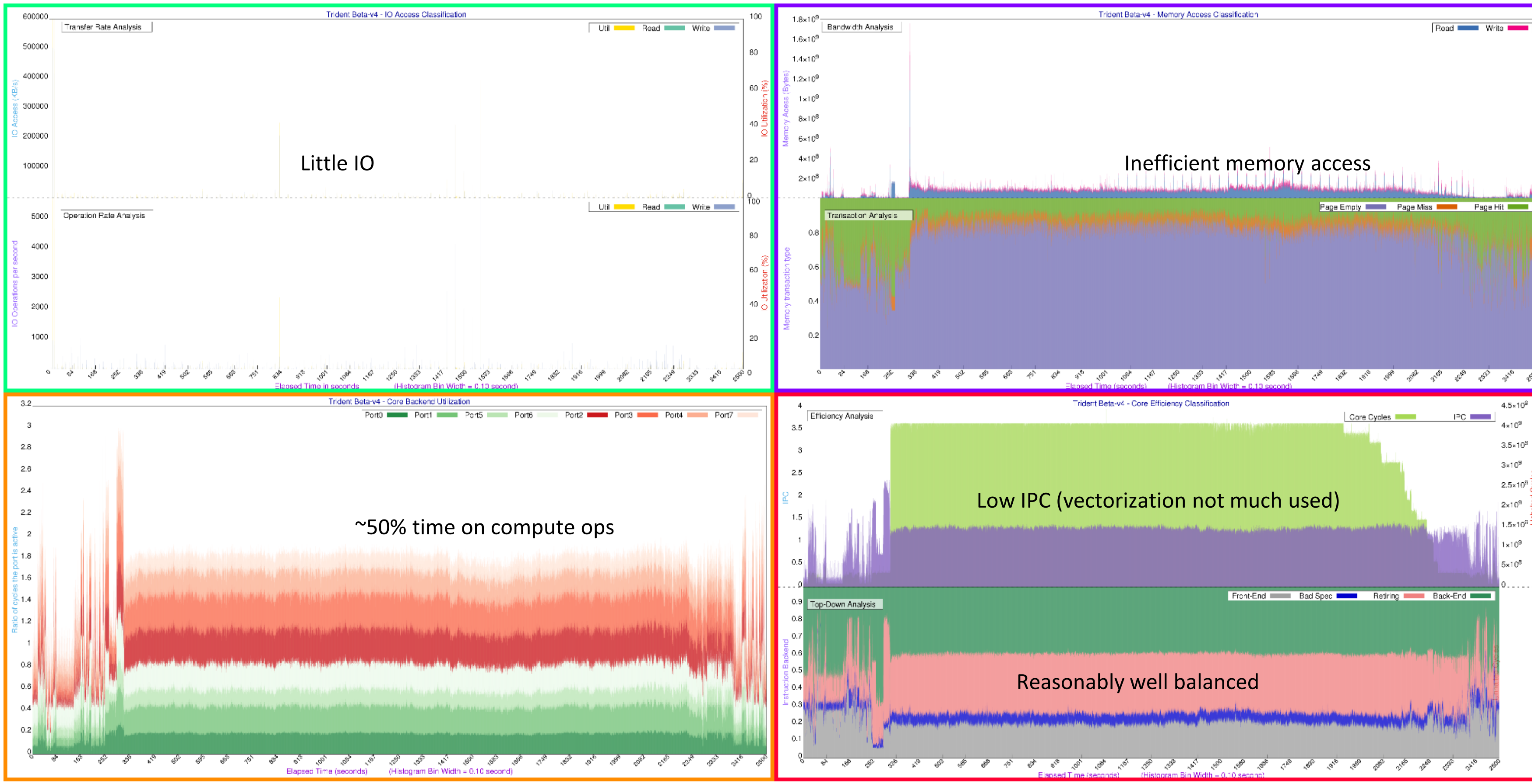
- Measures CPU, IO and memory utilisation based on hardware counters, memory and IO information
- Several metrics calculated
  - CPU: IPC, total cycles, top-down analysis (time spent on front-end, back-end, retiring, bad speculation)
  - Core backend utilization: compute (ports 0,1,5,6) vs memory (ports 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 (HS06, SpecCPU2017?)
- 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

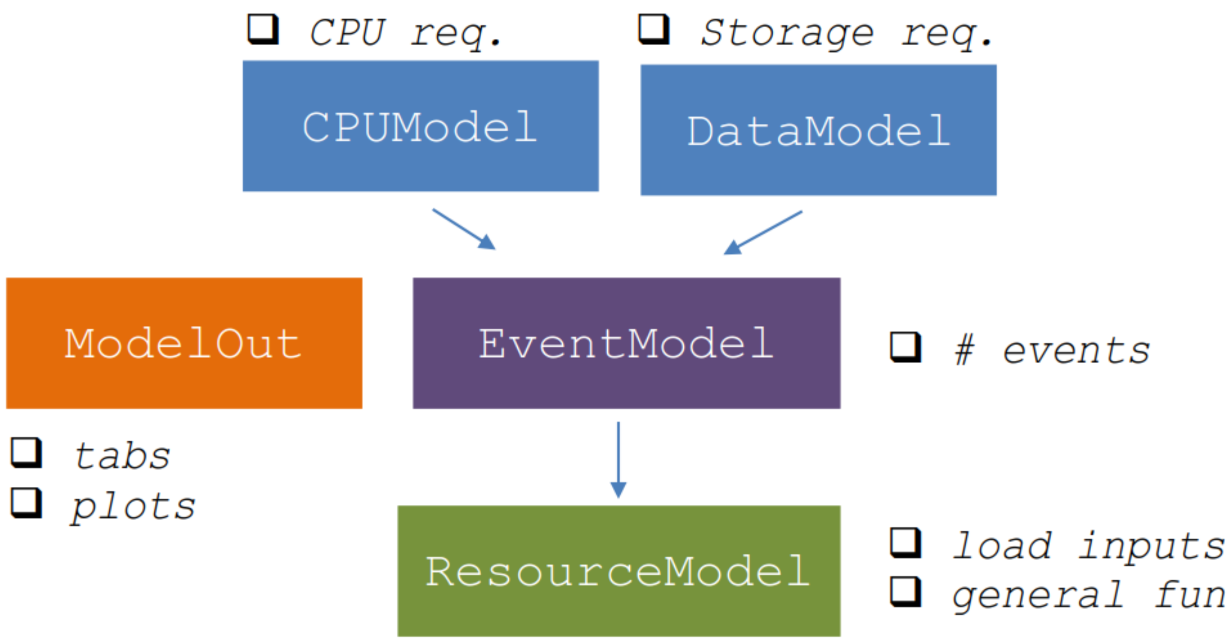
Resource estimation (1/2)

- 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, generic and customisable
  - Using as an input the characteristics of the workflows
  - Reproduce with reasonable accuracy 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
    - Being evaluated by ATLAS and LHCb

WLCG Worldwide LHC Computing Grid

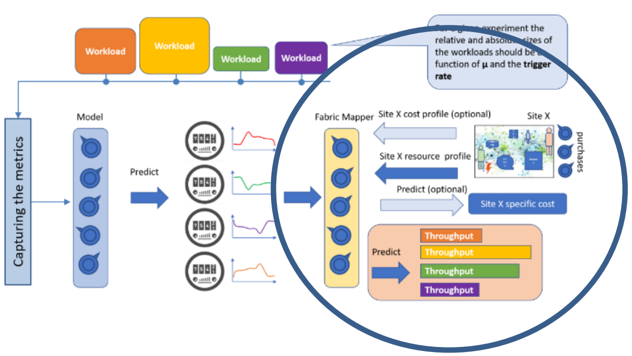
Resource estimation (2/2)

- 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



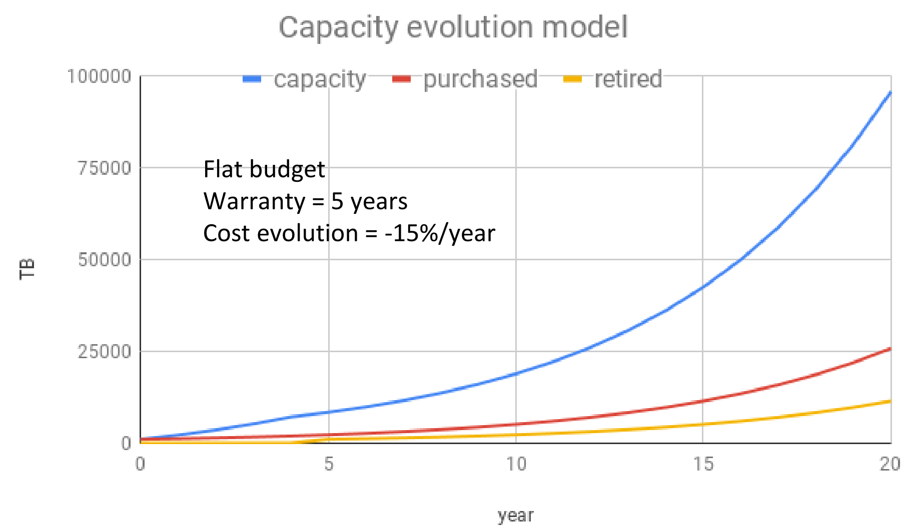
Site cost estimation models

- 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



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



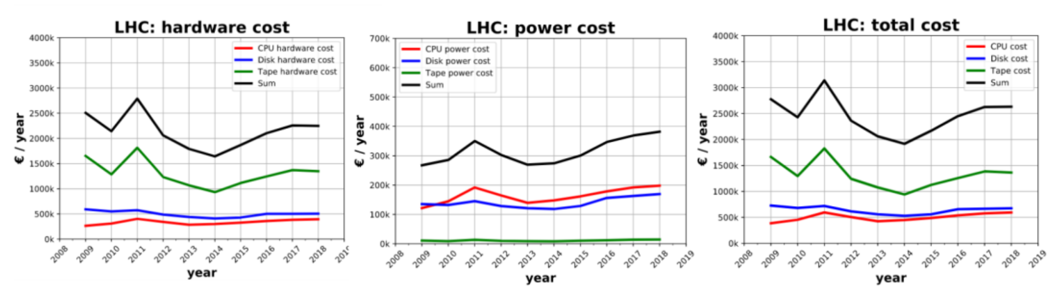
Investment (t) = Capacity (t) * Modeled Cost (t)

€/year — site (flat) budget; HS06, TB — site capacity; € / HS06, TB / year — 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

hardware cost + power cost = total cost



Source: R. Vernet

Storage Impact: preliminary estimates

- Concentrate persistent storage at a small number of large sites ("data lake") and use caches at T2's?
  - Manpower for storage (from the 2015 WLCG survey): ~2.5 FTE at T1's, ~0.5 FTE at T2's
    - Increases very slowly with size
    - Assumed much lower (~0.1 FTE) for cache-only sites
    - 13 T1 × 2.5 FTE + 157 T2 × 0.5 FTE ≈ 110 FTE → 15 "T1" × 3 FTE + 155 "T2" × 0.1 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 1 TB of MC, or 1 TB of disk for another copy?
  - HDD failure rates in EOS ≈ 1%/year → ≈ 1 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 ≈ 1000 HS06 · s and is ~400 kB
    - Regenerating the 1 PB of AOD lost → $2.5 \cdot 10^9$ events → $2.5 \cdot 10^{12}$ HS06 · s = 80 kHS06 · y
    - CPU needed costs the same as 20 PB of raw disk (~20% of the cost of full data redundancy)
  - Huge advantage of 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

WLCG Worldwide LHC Computing Grid

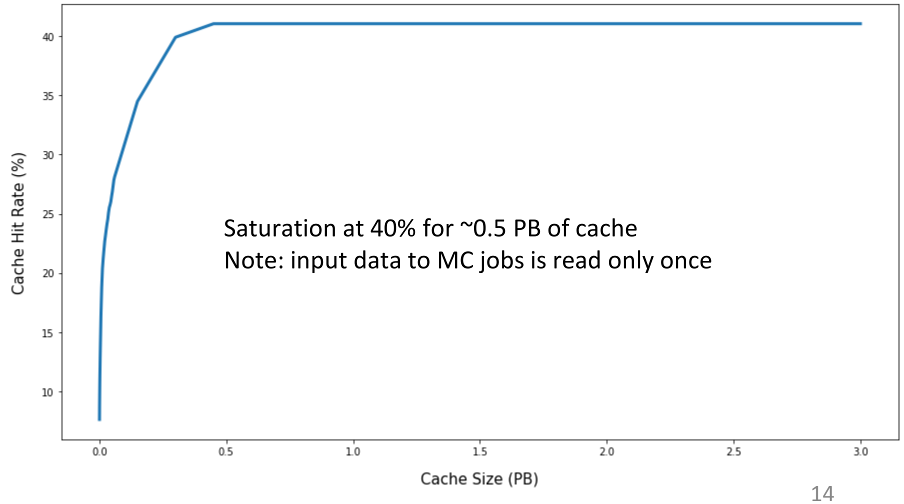
Preliminary studies on caches at T2’s

- 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
- Cache simulation at Prague T2
  - Site has 6 PB of disk
  - Used one month of real data access history
  - Assume the 2nd time a file is read, is read from a cache
  - Need to extend to more sites and more experiments
  - Cache size much less than current disk

Credits: Lucrece Laura Akira

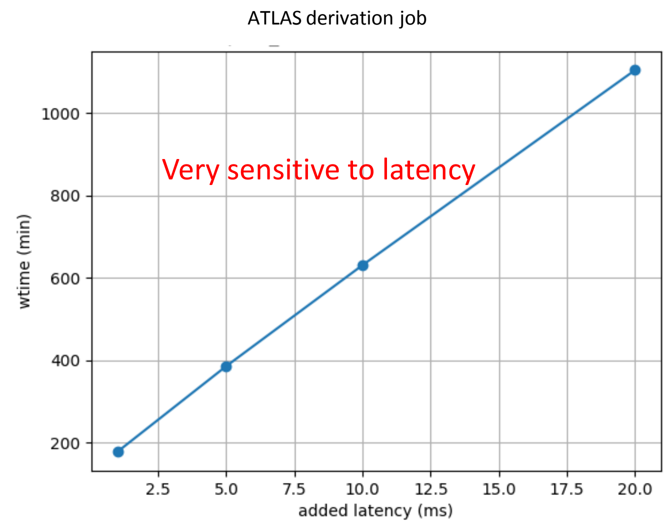
Job type	Run conditions	Run time (min)	Relative run time
DIGI-RECO	local	240	1.0
	Remote far, no cache	480	2.0
	Remote far, empty cache	262	1.09
	Remote far, pop’d cache	250	1.04
Derivation	Local	147	1.00
	Remote close	151	1.03
	Remote far, no cache	1217	8.28
	Remote far, empty cache	155	1.05
	Remote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

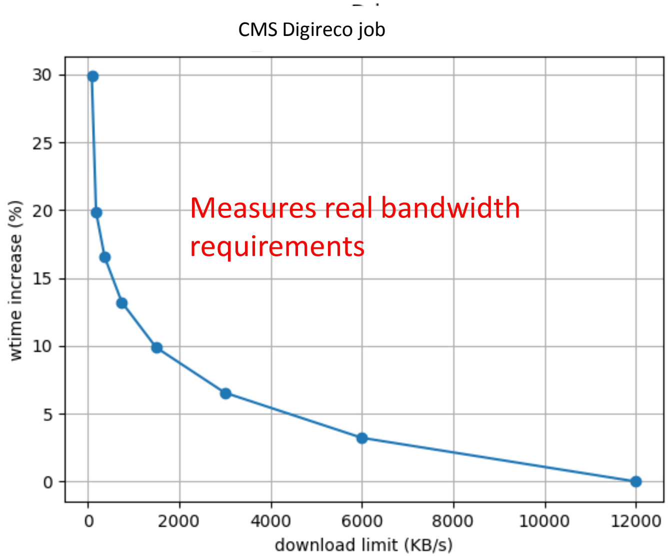
Credits: Irvin Umana Chacón

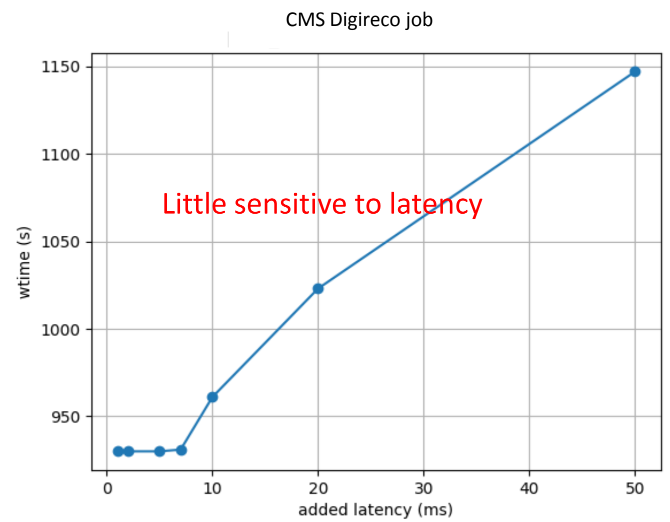


Throughput vs latency: preliminary studies

- 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
  - 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", 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

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
  - Framework on resource need estimation
  - Effect of storage caches
  - Effect of network bottlenecks
  - …
- Working closely on some topics to other bodies (e.g. the DOMA working groups)
- Work is still in progress but the time scale is long…
  - Active participation from more people is always very welcome!

WLCG Worldwide LHC Computing Grid

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

WLCG Worldwide LHC Computing Grid