PIC
port d'informació
científica

Systems Performance and Cost Modeling Working Group

Status and highlights of the working group

J. Flix (PIC/CIEMAT) , M. Schulz (CERN-IT), A. Sciabà (CERN-IT)

On behalf of the HSF-WLCG Working Group on Systems Performance and Cost Modeling

HEPiX Workshop - Amsterdam - 14th October 2019

WLCG
Worldwide LHC Computing Grid
&
HSF

The Working Group

Future WLCG resource requests are **far beyond** what technology improvements can bring in a flat-budget model scenario

Main motivation of this Working Group is to help WLCG to fit into the available resources for Run3 and Run4

- Develop a deep and better understanding of current workloads, resource utilisation, and site costs
- Explore the impact of future scenarios, estimating possible improvements in efficiency and potential savings in resources
- Develop tools and methods for the above

Systems Performance and Cost Modeling Working Group

The group was endorsed and created by Nov. 2017 [mandate]

- **Conveners:** J. Flix, M. Schulz, and A. Sciabà
- **Members:** ~35 active contributors → **wlcg-SystemPerformanceModeling@cern.ch**

Bi-weekly meetings:

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

Active participation from workload, workflow and framework developers, people who plan, engineer and operate IT systems and people who put all this in global WLCG

Contributors

Alaettin Serhan Mete, Alessandra Forti, Alessandro Di Girolamo, Andrea Sartirana, Andrea Sciaba, Andrea Valassi, Andrew Sansum, Andrey Kirianov, Antonio Pérez-Calero, Bernd Panzer-Steindel, Carlos Perez Dengra, Catherine Biscarat, Concezio Bozzi, Costin Grigoras, Daniele Bonacorsi, David Lange, David Smith, Davide Costanzo, Dirk Duellmann, Domenico Giordano, Duncan Rand, Eric Fede, Erik Mattias Wadenstein, Frank Wuerthwein, Gareth Roy, Graeme A Stewart, Helge Meinhard, Jan Iven, Johannes Elmsheuser, José Flix, Markus Schulz, Martin Gathhuber, Michel Jouvin, Michele Michelotto, Oxana Smirnova, Paul Millar, Raul Cardoso Lopes, Renaud Vernet, Servesh Muralidharan, Tommaso Boccali, Torre Wenaus, Xavier Espinal Curull, Xiaomei Zhang, Yves Kemp, Giulio Usai, Christopher Hollowell, Hadrien Grasland, Alastair Dewhurst, Shigeki Misawa, Martin Gasthuber

Credits go to them!

(current) Working areas

Current **areas of work and goals**

- Selected representative experiment workloads to run in a controlled environment
- Defining which metrics best characterise such workloads
- Establish a common framework for estimating resource needs
- Defining a process to evaluate the cost of an infrastructure with real costs from (big) computing sites in WLCG
- Measuring the impact of new storage configurations
- Identification of potential areas for savings

Workload characterisation for the LHC experiments

Identification of (today's) representative workloads

- **CPU intensive** **[e.g, MC Simulation]**
- **CPU+I/O** **[e.g, Reconstruction]**
- **I/O intensive** **[e.g, Merge, skimming)]**

Containerisation of workloads - run in a controlled environment

- Self-contained **docker images** [with the help of **HEPIX CPU Benchmarking WG**]

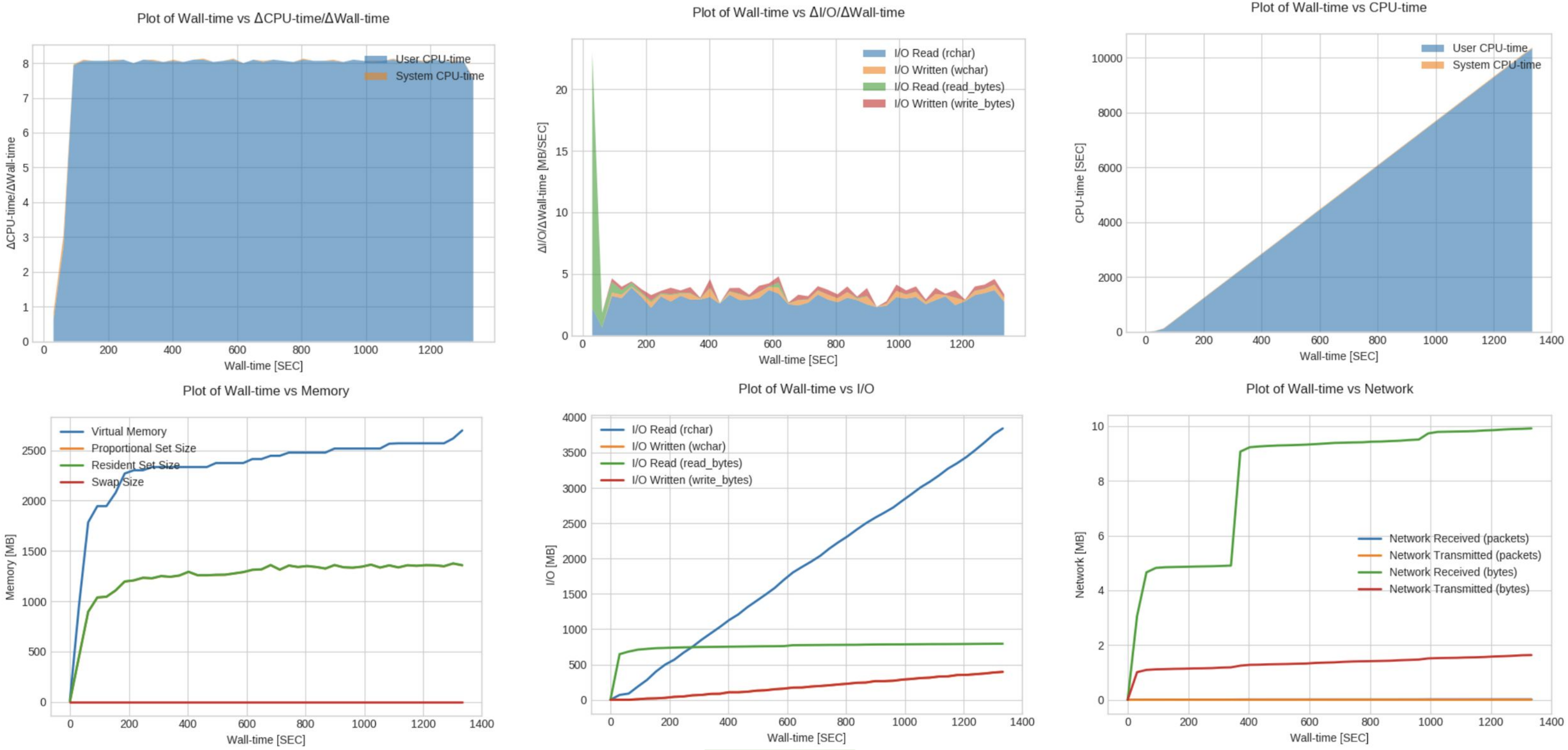
Performance metrics measurements (profiling)

- Set of parameters identified → tools assessments
- Two tools adopted/deployed: **PrMon** and **Trident**
- Testbeds available at CERN and PIC
- Application Behavioral Modeling under **Multidimensional Resource Restrictions**

Workload characterisation for the LHC experiments

[S. Mete talk]

CMS GEN-SIM

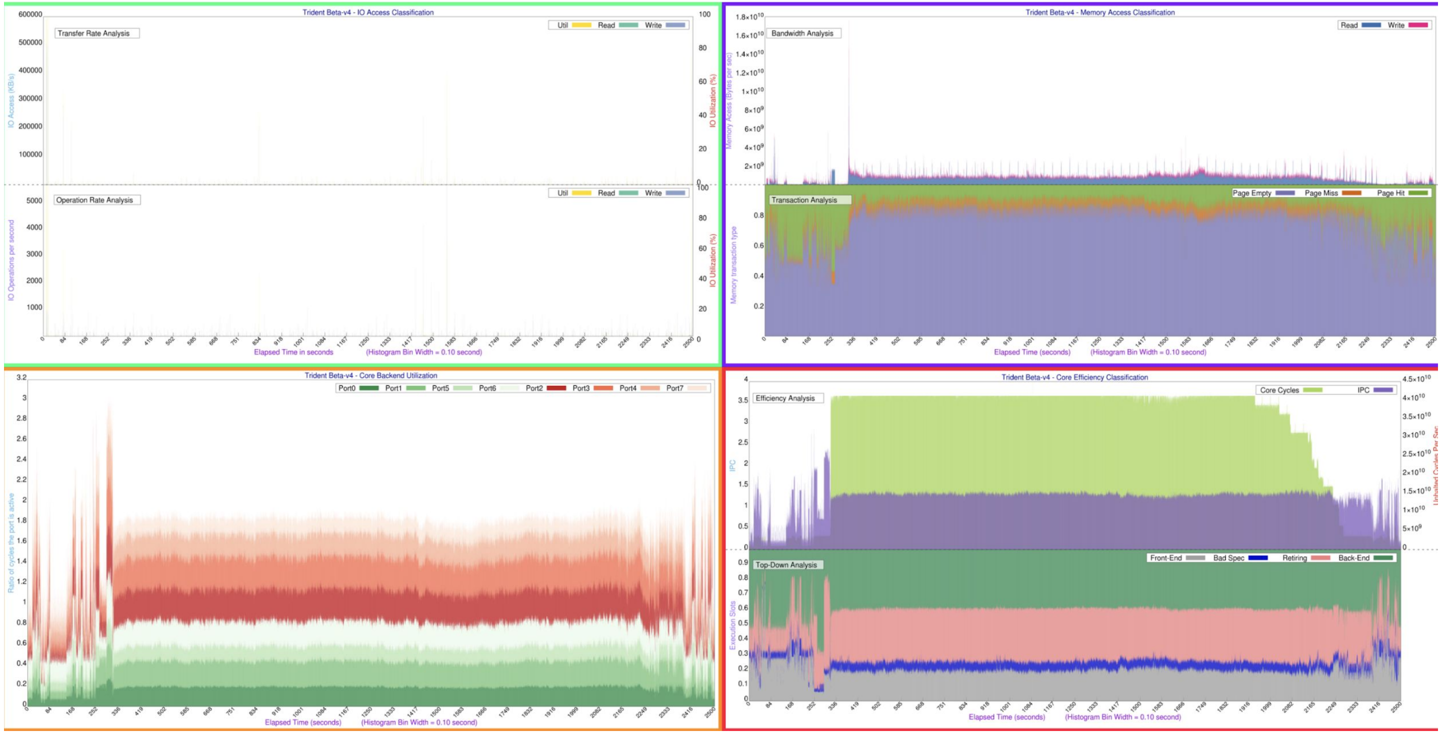


PrMon

Workload characterisation for the LHC experiments

[S. Mete talk]

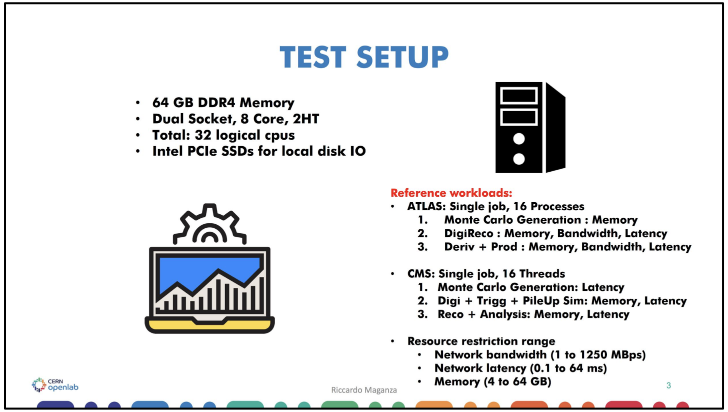
ATLAS – MC Simulation



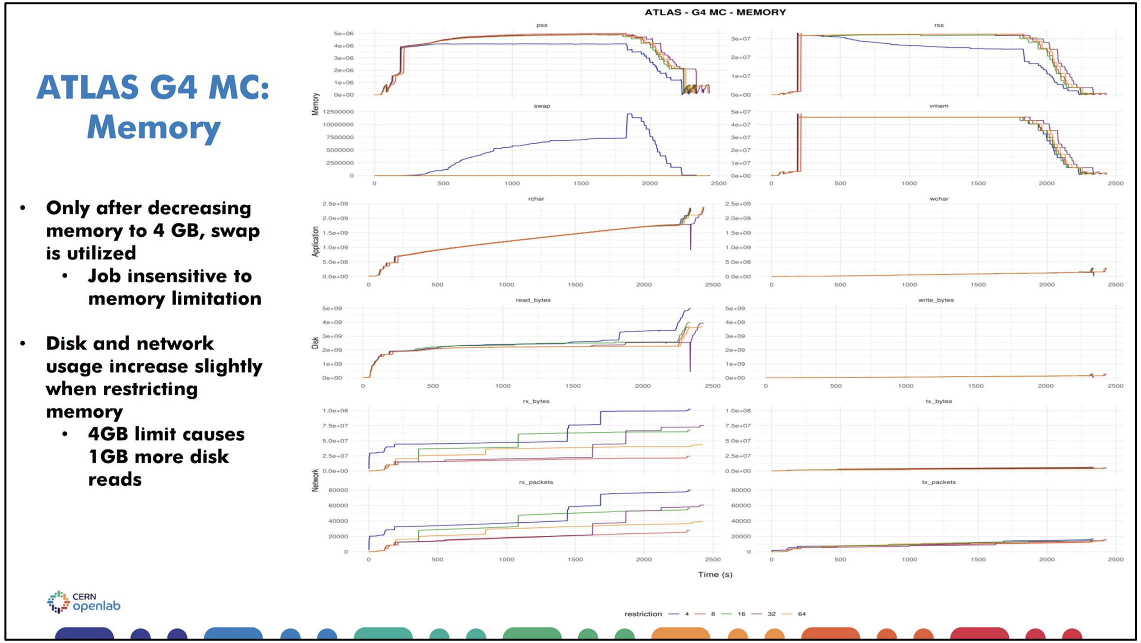
Trident

Workload characterisation for the LHC experiments

[R. Marganza talk]

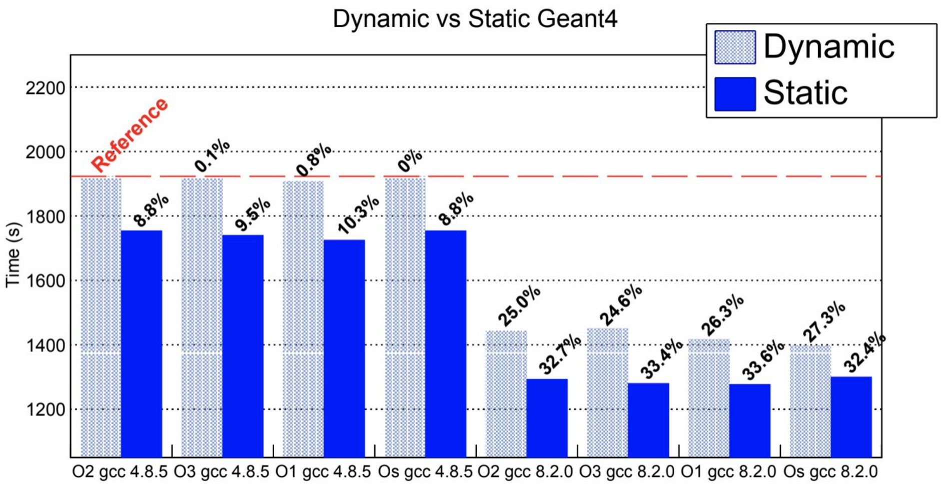


Resource Restrictions on ATLAS/CMS applications



Different compilers on G4 sim execution time

[C. Marcon talk]



Running standalone **Geant4 simulations**, using either dynamic (used by LHC experiments) or static libraries, disentangling them from other libraries that may affect performance

Two versions of the **GCC** compiler, namely **4.8.5** and **8.2.0**, have been used for these studies, and four GCC optimization levels (Os, O1, O2 and O3)

G4 simulation with a CMS-based geometry (from A. Dotti **[*]**) has been used as a **Reference**

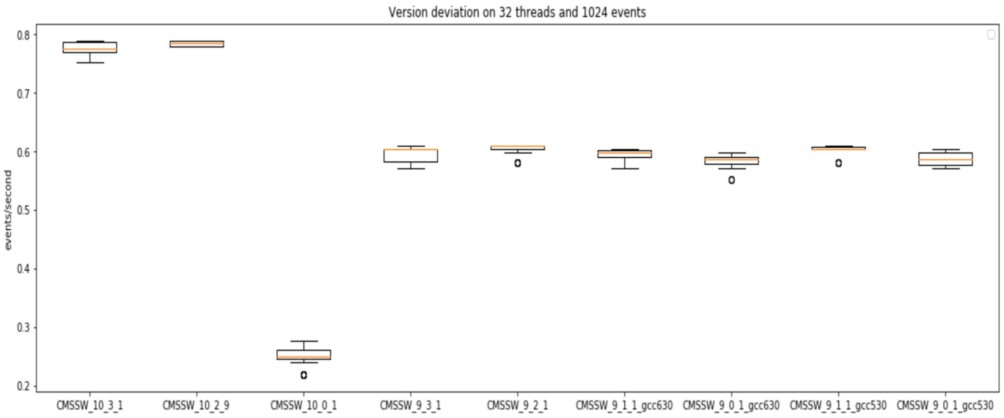
The static approach, for both the GCC versions considered, generally reduces the execution time by more than 10%

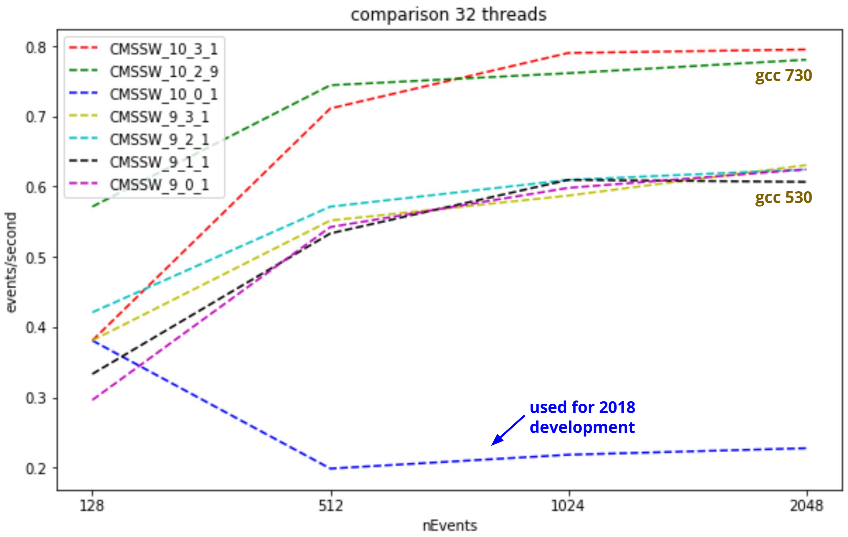
Switching from GCC 4.8.5 to GCC 8.2.0 results in an average of ***30% improvement*** in the execution time, for both build approaches

[*] **https://gitlab.cern.ch/adotti/Geant4HepExpMTBenchmark**

CMS GEN-SIM performance vs CMSSW versions

[A. Alonso talk]





Since GEN-SIM does not vary much from different CMSSW versions, we infer newer versions of gcc improves its performance

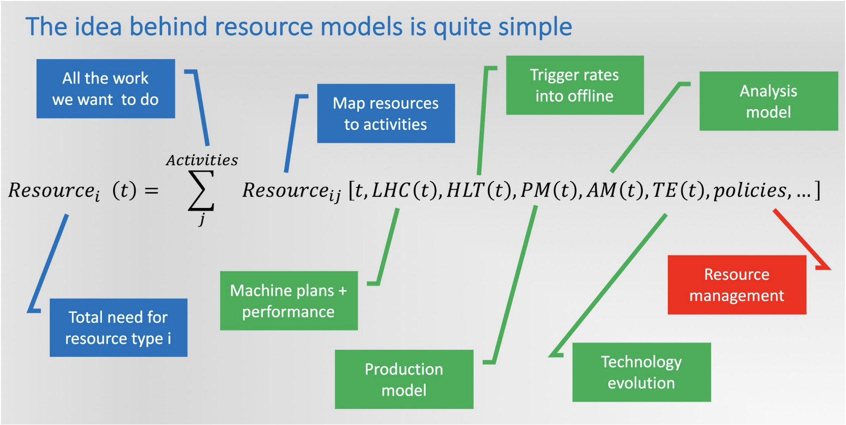
This ***25% increased performance*** (in events/second) are similar to the studies made by C. Marcon

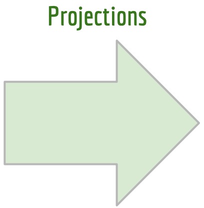
Resource estimation

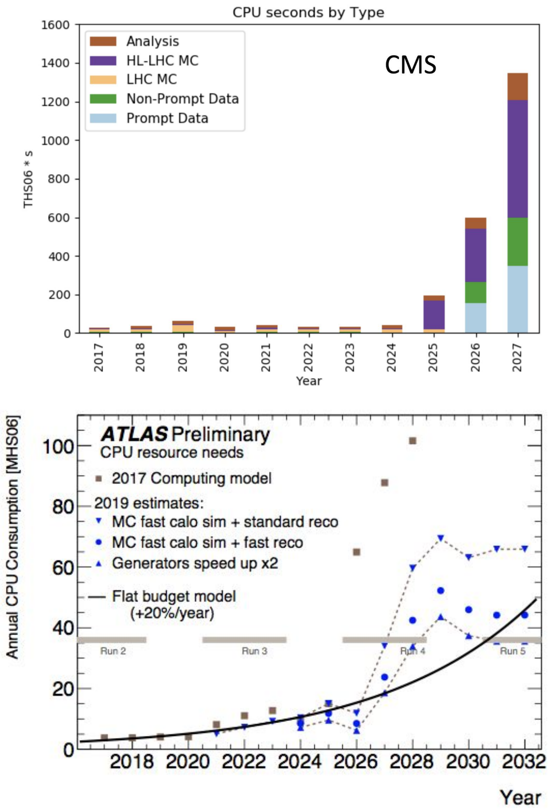
- Taking inspiration from the **complex spreadsheets** that experiments used to calculate their resource needs
- Create a **simplistic model** that can be progressively refined. Agreed that it is much better that this is provided by means of a code, rather than continuing with the spreadsheets
- Experiments did efforts in this direction **[python codes]**
- Discussions of the state of the art of the resource estimation up to HL-LHC for each experiment and explore commonalities

Resource estimation

[D. Lange talk]
[G. Usai talk]







Models based on **current practices** → Difficult to anticipate impact of R&D or behavioral changes before they happen
Model includes data management policies, LHC Schedule, Campaigns...
Potential **new metrics** to be included: Caches, Tape I/O, Network, Accelerators, ...
Flexibility to add new paradigms (opportunistic resources, application performance, LHC commissioning periods...)

Site costs and estimations

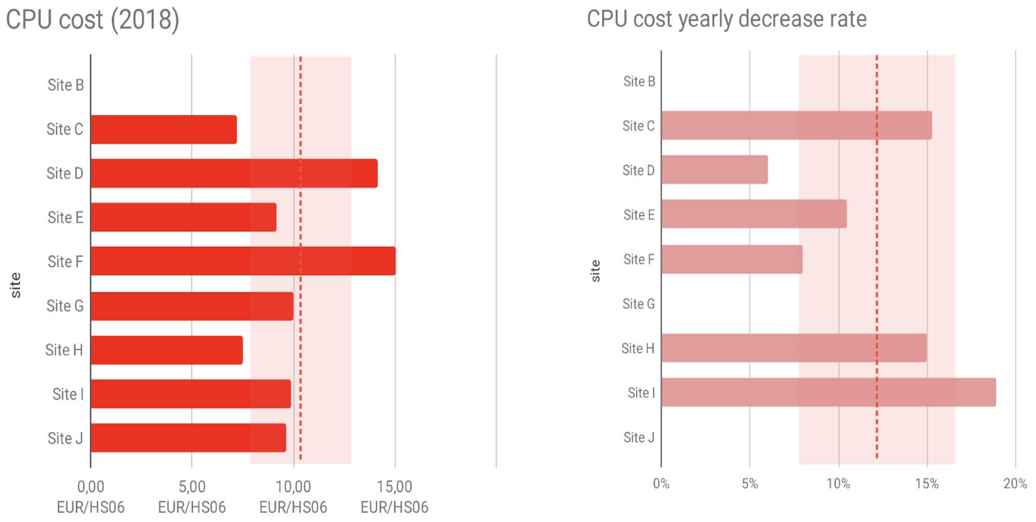
- Outlining a **method** to calculate the cost of hardware to expand the site capacity to hold the most important workloads, where the cost is split into several components
- **Survey** sent to the big sites (Tier-1s) to understand major costs
- Data collected (**confidential**) analyzed
  - Current costs on CPU, Disk, Tape, power (manpower not considered yet)
  - Evolution over last years
  - Variation across sites
- **TCO study is difficult**, however this information is helpful to understand improvements impact into infrastructure costs

Site costs and estimations

[R. Vernet talk]

CPU

- Current cost and yearly decrease

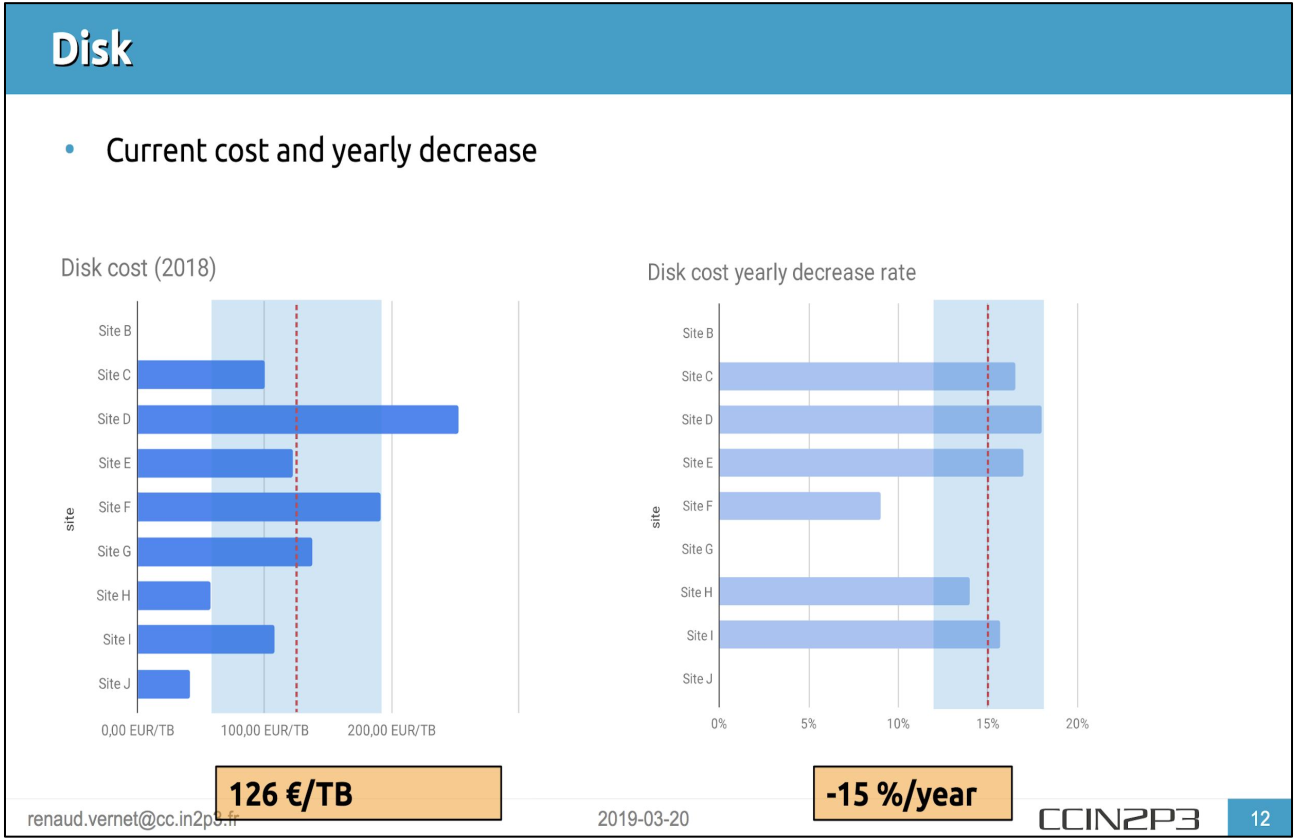


10.3 €/HS06

-12 %/year

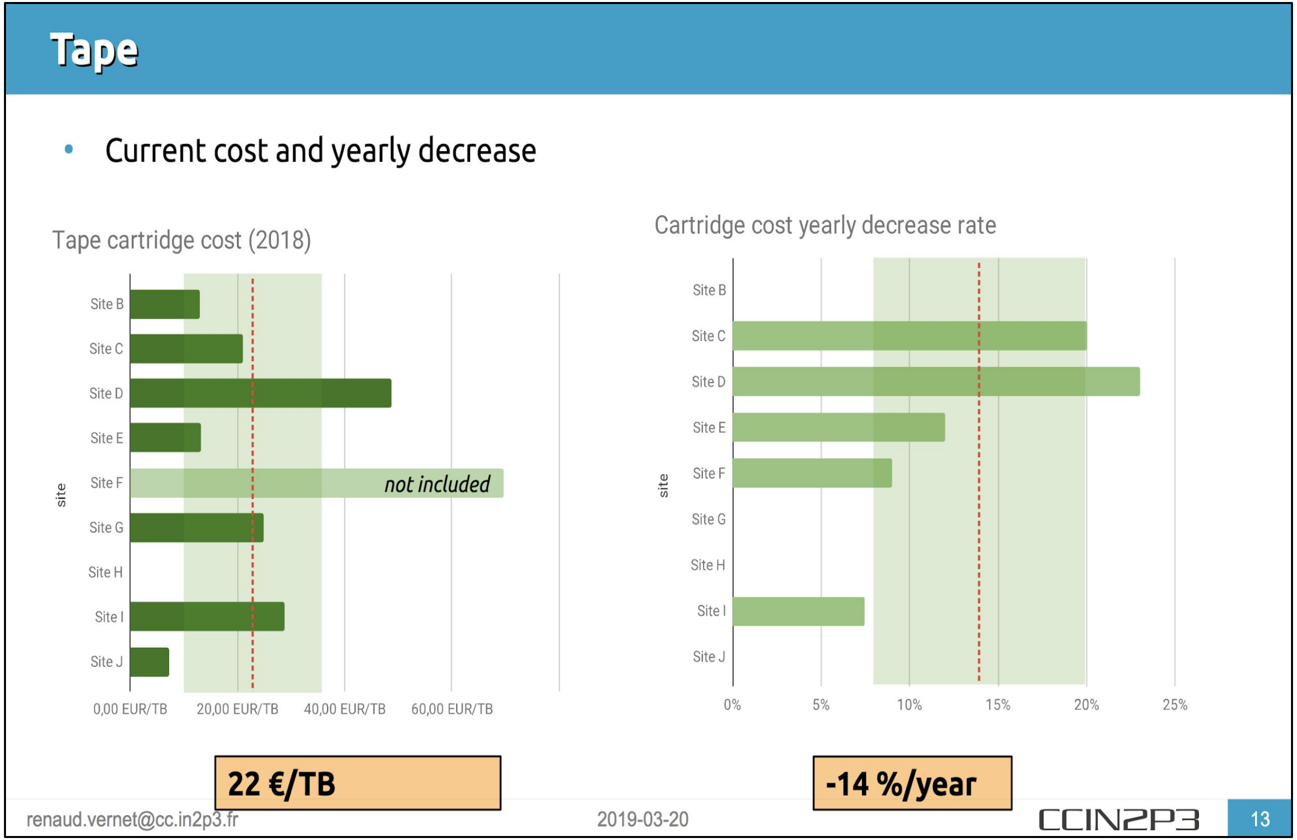
renaud.vernet@cc.in2p3.fr 2019-03-20 CCIN2P3 10

Site costs and estimations
[R. Vernet talk]



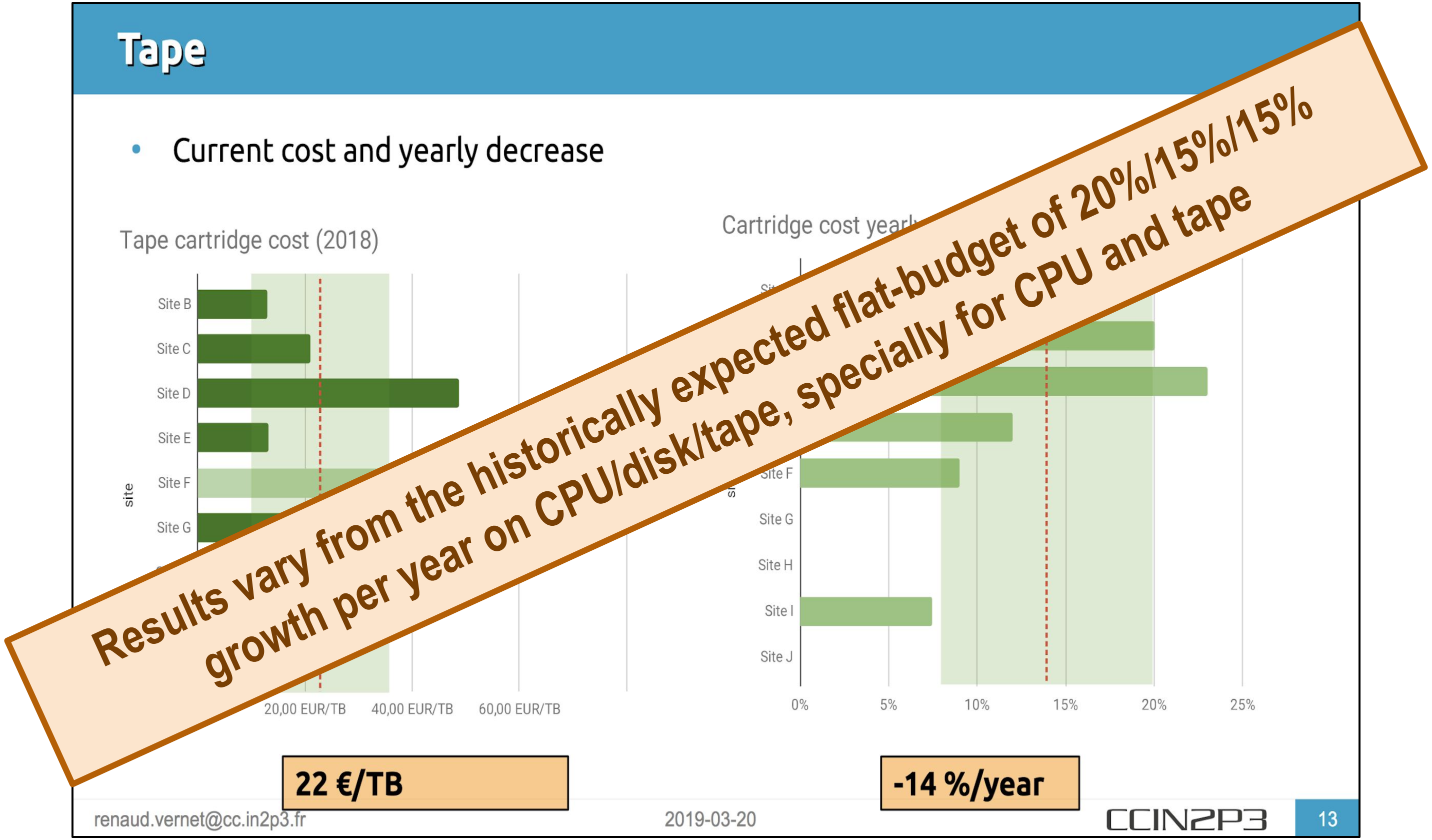
Site costs and estimations

[R. Vernet talk]



Site costs and estimations

[R. Vernet talk]



Measuring impact of new storage configs

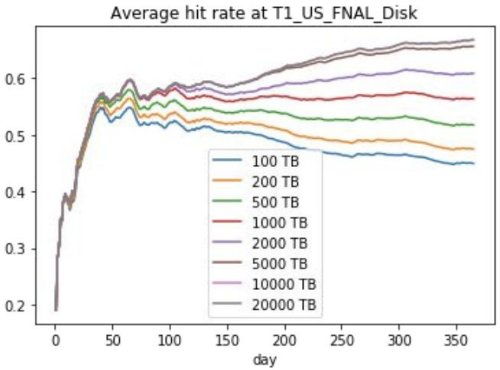
Trying to understand different approaches on **data access** and **popularity** from the experiments, and how to combine results and incorporate it into the model, in close collaboration with DOMA group

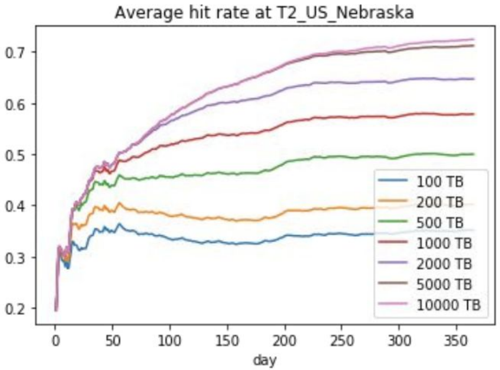
- How data **caches** deployments might impact on storage resources deployed?
- Alternative storage models (**Data Lake Strawman**)
- **How applications behave** reading/populating caches?
- Which are the implications in the **network**?
- Which is the **cost impact** of all this together?

Studies from experiments and sites data popularity/access, simulation of caches, data losses, latency effects to applications,... **very active area!**

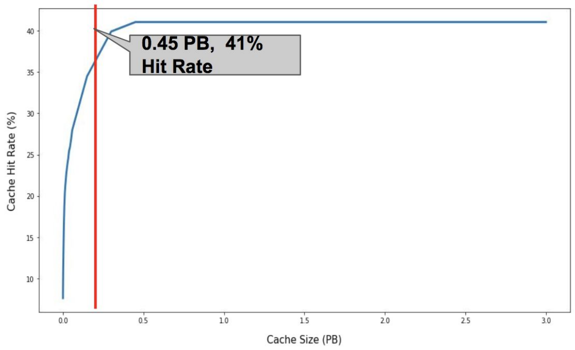
Measuring impact of new storage configs

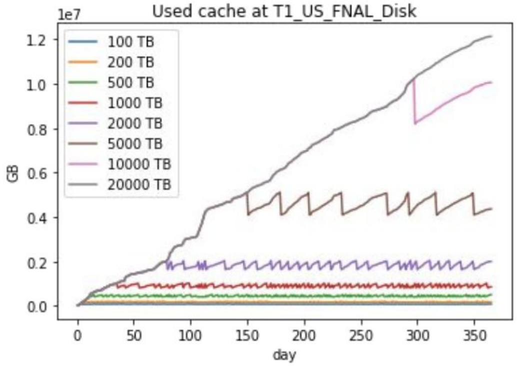
Simulating Data Caches [CMS]
[talk]



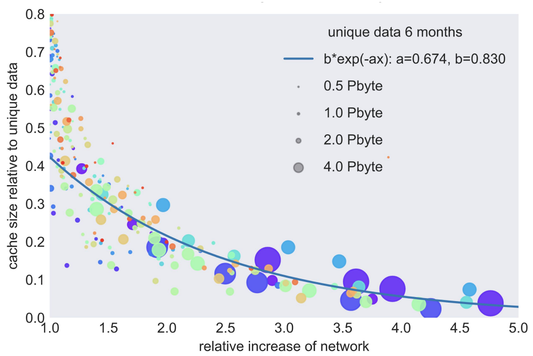


Prague Tier-2 cache simulation [ATLAS]
[talk]

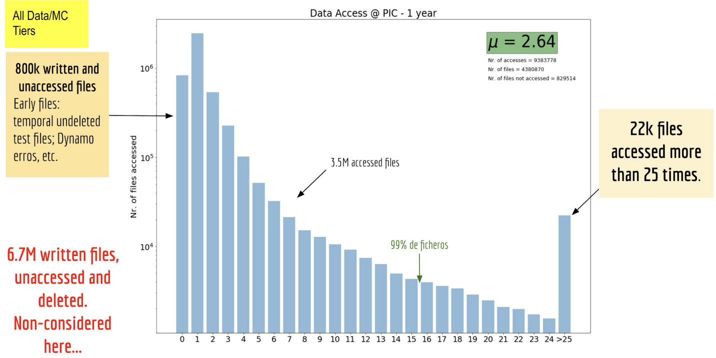




Cache effects on the network [ATLAS]
[talk]



PIC Data Access and Popularity studies [CMS]
[talk]



Gluing all together: potential savings

[M. Schulz talk]

Change	Effort Sites	Effort Users	Gain
managed storage on 15 sites + caches elsewhere	Some on large sites/gain on small sites	little	40% decrease in operations effort for storage
Caches at most sites (dataLake strawman)	Some everywhere	Frameworks some	15% of storage
Reduced data redundancy	Some large sites	Frameworks some	30-50% disk costs
Reduced data replication and cold data	little	Frameworks some	30% disk costs
Compact data formats for analysis	none	Some	>15% disk costs
Scheduling and site inefficiencies	Some	Some	10-20% gain CPU
Reduced job failure rates	Little	Some-Massive	5-10% CPU
Compiler and build improvements	None	Little-Some	15-20% CPU
Improved memory access/management	None	Realistic	10%-15% CPU
Exploiting modern CPU architectures	None	Massive	100% CPU
Paradigm shift algorithms	Some	Massive-Infinite	Factor 2-100 CPU
Paradigm shift online/offline data	Little	Massive-Infinite	2-10 CPU 10-20 Storage

8

Conclusions

Resources in WLCG will be more and more **constrained** in the next years

- Attention is focusing on increasing efficiency and cost-effective scenarios

Performance studies to characterize workflows being done

A common model to accurately estimate **resource needs** and best allocate spending is being developed

Efforts to understand **site costs** has progressed well (for big sites)

Active progress in understanding the effect of new **storage configurations**

Collaboration with other Working Groups (DOMA, Technology watch, ...)

Systems Performance and Cost Model WG as a **long term activity**

*** Want to volunteer? contact the authors of this talk!**

HEPiX

Thanks!
Questions?