HPC Pledge-Equivalence

Problem

- HPC systems differ progressively from WLCG commodity based services
- Usability (or lack of)
  - Network access, OS, access to storage
  - Scheduling
  - Access, Authorisation and Accounting
  - Granularity of shares …..
- Architecture (diversity and focus)
  - CPUs (range of x86, power9, ARM..)
  - Accelerators
    - GPUs, TPUs, FPGA….
    - Represent majority of computational capability
  - Focus on high speed low latency internode networks

What is the value of a computer?

- Value is the **throughput** that can be achieved **for our use cases**
  - In the past HS06 mapped to this
  - HepScore will use a balanced mix of HEP workloads to ensure this
- HS06 and HepScore only make sense for an experiment when:
  - All relevant workflows can be run on the system
  - The threshold for usability is low enough that the resource can be integrated into the workflow/data management services (with reasonable effort)
- For many HPC systems this will not be the case for a long time
  - Standard HPC benchmarks don't work for us (LIN/LA-PACK FLOPS etc.)
- Throughput and usability have to be factorized
  - One is useless without the other
- We looked at the value only

Core concepts

- Pledged HPC resources can be allocated to tasks with a given granularity
  - Units of nodes or number of CPU cores and accelerator cores
- For each workload that can be run on the HPS the ***throughput*** is ***measured***
  - Filling the system in the way that maximises the throughput
- For each workload the t**hroughput on a conventional system** with known HS06/HepScore rating is ***measured***
- These measurements are used to calculate a ***HS06/HepScore Equivalence***

- More details can be found in the backup slides…

Consequences

- The part of the resource that can be used might represent only a fraction of the potential ability of the resource → poor economics
    - Underused accelerators, bad fit of resource needs and scheduling granularity etc.
    - This can be quantified with a metric for the **Realised Potential (RP)**
- RP can be measured by comparing the LINPACK R_{max} FLOPS of the pledged resource with the FLOPS that the workload consumes on the resource
    - This can be measured either on the system or derived from the equivalence HS06/HepScore
    - RP allows to compare the level at which an application can exploit the resource
- When experiments run several workflows on a resource, those with the highest HS06/HepScore Equivalence should be used with priority
- By providing queues for each workload the local accounting can be translated to the WLCG units

MB mandate for Cost Modeling and Benchmarking WG

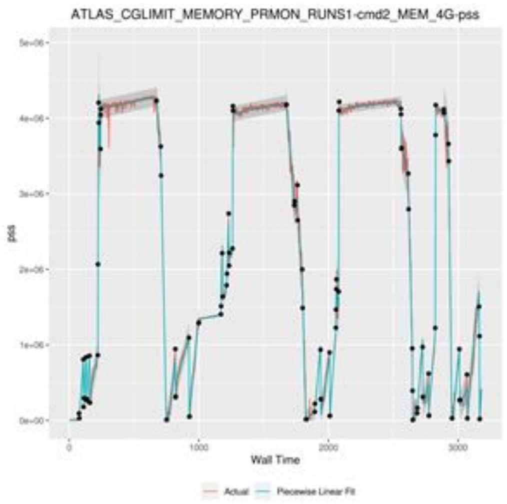
- Created a first draft document to collect ideas and derived from this a second, more concrete text
  - https://docs.google.com/document/d/1vxAwt8Eb3WkBwfVdfMAzGtV3gpnY-zrhOti3jtB260E/edit?usp=sharing
  - Main contributions by Andrea Sciabà and Domenico Giordano
- This has been circulated, covering experiments and site people
  - Many comments have been received
- A new version is currently being prepared
  - A short first part on the concept and principal approach
  - A longer annex describing the calculation of HS06/HepScore Equivalence in detail
    - Providing examples

Cost and Performance Modeling Working Group

Status and Future

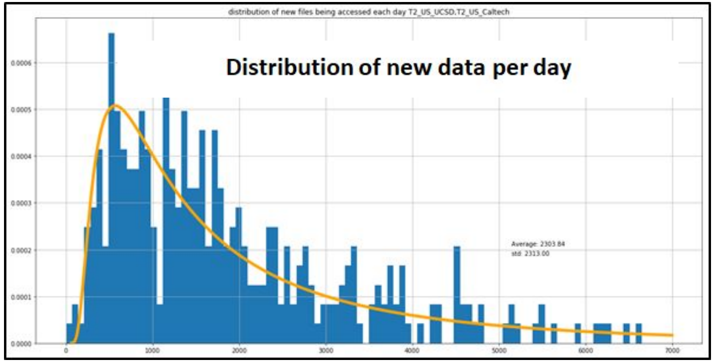
Recent progress (1/2)

- Resource needs estimation
  - All experiments have now a code-based machinery to extrapolate their resource needs up to Run4 scenarios (moving away from complex spreadsheets)
  - Input parameter set is almost the same
  - A common framework is still possible, but the functionality is already there
- Application performance studies
  - Revamped studies to measure effects on performance of restrictions in memory and network bandwidth and latency
  - Parameterization of PrMon time series (e.g. memory usage, I/O vs. time) using change point detection algorithms
  - TO DO: multidimensional parameterization when adding as additional parameter a constraint on memory, bandwidth, latency…
- Detailed profiling of CMS reference workloads using Trident
  - To find and understand bottlenecks and underutilization issues

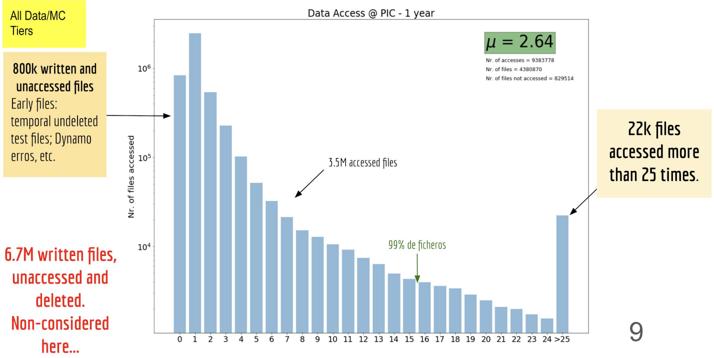


Recent progress (2/2)

- Effect of compiler options on performance
  - Looked at differences in performance in Geant4 using different gcc versions and dynamic vs. static compilation
  - Different detector geometries (LHCb, CMS, (ATLAS))
    - Caterina Marcon
- Cache simulation studies
  - Using ATLAS and CMS data access popularity records, study the effect of a site cache and optimize the trade-off between cache size and network traffic
  - Studying different cache management strategies
    - Purging least recently accessed files vs. estimating the “best” time to purge based on the full data access history of a file
  - Access patterns parameterized by simple model
  - Now migrating to the scope of DOMA Access
- Data access and popularity studies at PIC
  - Provides a more detailed picture than the general approach (including time of file creation and deletion) using the dCache billingDB
  - Will extend to CIEMAT and migrate data to CERN’s Analytix data analytics cluster



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



Outlook

- Many activities have now there center of gravity in other working groups
  - Reference workloads and their measurement and modeling → Accounting WG
  - Data access and storage cost optimisation → DOMA (access/QoS)
  - Detailed studies of workload behaviour on CPUs
    - Trident, code analysis → Various activities within experiments and HSF
  - Build and compiler studies (AutoFDO, static builds) → experiments and packages
- Others have become “standard” activities
  - Site cost calculations (two slightly different approaches to TCO)
  - T1 anonymised site cost assessment produced interesting spread in costs for storage
  - For Compute Resource Estimation ATLAS and CMS have now working Python scripts that are much more transparent than before
  - Pilot for storage and compute integration (BEER) (hyperconvergence) has been in production
- Focus on efficiency and performance is now common (evangelisation done)
  - Computing Schools, Working Groups within the experiments, HSF, GPU, ML…..
  - Generator workshops, several activities around simulation, faster reconstruction, compact analysis formats………

Why is this a problem and what can be done?

- Cross reporting of progress isn't efficient and can lead to confusion
  - We recently looked for a presentation by a student and had to look in three WG agendas
    - Each one an excellent fit
- Keeping track of all activities is taking a lot of effort
- Relocate and rescope:
  - Move activities to the dedicated WGs with experiment and site participation
    - Workload, storage and access to DOMA and Benchmarking
    - Profiling to benchmarking or HSF
  - Identify topics that are unique to the Cost Performance Modeling WG
    - Site cost calculation with multiple sites
    - Confidential statistical analysis of cost differences
    - Refinements of the Resource Estimates
    - ....
  - Run topical micro workshops on unique topics with experiments and sites
    - Every 3 months could be linked to GDB, HSF-WLCG Workshops

Backup Slides

Why isn't it trivial to use HPC systems (efficiently)

- The summary of the Cross-Experiment HPC workshop gives a good overview of the software and operational challenges
- WLCG site services and experiment workloads have evolved together for almost two decades with many explicit and implicit agreements
    - Agreed authorisation and authentication system
    - Agreed hardware and software environment
        - Including memory and scratch space
    - Agreed set of edge services
        - Including systems like CVMFs with significant state
    - Agreed access to external networks from user level applications
    - Agreed access and behaviour of local storage
    - Agreed approach of providing resources
        - Resources are pledged for extended periods of time (at least several months)
    - Relatively fine grained contributions by each site (~20% max)

Why isn't it trivial to use HPC systems (efficiently)

- HPC systems evolved independently from WLCG targeting their user communities with their set of workloads and requirements
  - Variety of access restrictions
    - WAN-network, users …
  - Often large pledges for short times
    - O(100k) cores for weeks to months
  - Systems optimised for large parallel workloads
  - Huge variety of technology
    - CPUs, Accelerators, OS, shared file systems with different focus
    - Storage systems with different focus
  - Significant investment in interconnects for parallel jobs
  - Often the majority of computational capability comes from accelerators
  - Expecting and supporting the porting and tuning of workloads
      - Many codes are considered "classic" -- fluid dynamics, linear algebra…..

Why bother?

- Funding agencies will keep funding larger HPC systems
  - Strategic reasons
  - Like to see them fully utilised
- Next generation of HPCs will provide more computational power than all WLCG sites combined
- The technology of HPC nodes and server nodes converge
  - GPUs and other accelerators are the most likely path to cost effective computing in the future → porting to GPUs already started in all experiments (online and offline)
  - Data layout in memory needed for GPUs will help with CPU performance too
- The increasing use of machine learning in HEP
  - TPUs for inference, GPUs for training

Why bother?

- Our resource gap for HL-LHC

Those in dire need must be content with what they get

How to value HPC pledges?

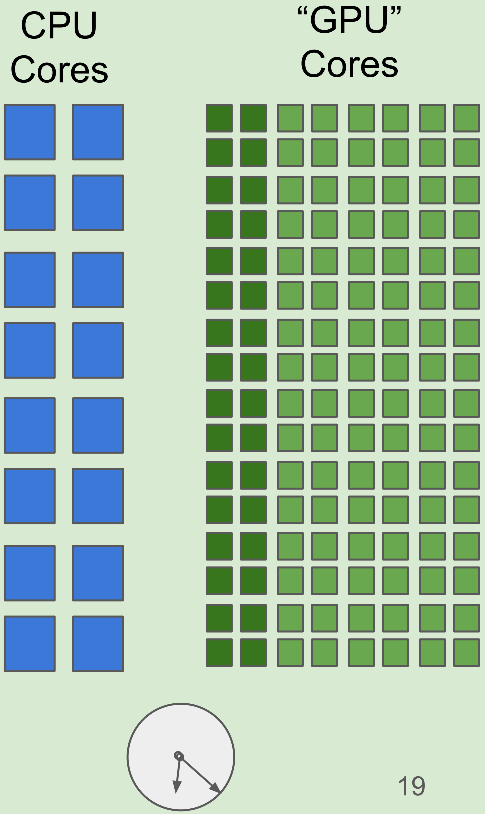
- Current WLCG pledges are in units of HS06
  - One set of abstract benchmarks combined to a number
- This will soon be replaced by HepScore
  - A mix of containerised experiment workloads mixed by weight factors based on usage
    - Massively oversimplified by me
  - This works well when all workloads can be run on the target
- Currently only selected workloads from the experiments run on most HPC systems (see Alexei's table of ATLAS' use of HPCs)
  - Porting, constraints, ……
- For several/most workloads only limited support for GPUs exists
  - This will change, but given the diversity of technologies this will remain a challenge for many years
- → Using standard HPC benchmarks won't work for us
  - See also Shigeki Misawa's note: GPU-Benchmarking.pdf ,

Pledge valued by *usable* Throughput

- A pledged resource that can't be used has no value for an experiment
- Process to determine value of the resource has to be based on usage of the resource
  - Access and compatibility problems have to be addressed to a certain degree
- Compare *achievable* throughput with throughput on conventional resource
- Not all workloads will exploit all architectures
  - Increasing variety of architectures and environments
- If not all workloads of an experiment can be run determine value for each workload that can be run
- Next slides describe the approach in some detail

Mapping the usable capacity to the WLCG currency

- Start with a simplified model of the HPC pledge:
  - The pledge is based on what a job can request:
  - N_C[hours] of M_C [cores] plus N_A[hours] of M_A[accelerator-cores]
  - Accelerator cores (CUDA cores) are the minimum unit that can be scheduled
- An experiment measures the throughput of workload A_i
  - Z_i events in x_i [core-seconds] and y_i[accelerator-core-seconds]
  - In case the accelerators aren't used the y_i is 0
  - From this the average core and accelerator seconds needed to process one event is calculated
- For all workloads that an experiment can/will run on the pledged HPC system this measurement is repeated:
  - A_1, A_2....



Mapping usable capacity to the WLCG currency II

- Subsequently these workloads (not necessarily the identical code) are run on a traditional system with known “HS06” rating
  - The average “HS06”-seconds required to produce one event of workload A_i allow to convert the the HPC pledge back to known units
- There is no reason to assume that a workload will use cores and accelerator cores in the same ratio as they are pledged (especially when full nodes are pledged)
  - The “HS06”-equivalence is determined based on the resource that is saturated first

Example (very artificial)

- On a HPC system with 32 cores and 512 GPU cores per node an experiment gets 100 nodes for 200 days
- The experiment's reconstruction code can make use of the GPUs and CPUs
- For 10000 events on 8 cores and 100 GPU cores the code needs 12500sec
    - 1 event requires 10 core seconds and 125 GPU-core seconds

- Reconstruction run on 8 "traditional" Cores @ 10"HS06" takes 20000 sec
- 1 event requires 160 "HS06"-seconds := 10 core and 125 GPU-core seconds
- The pledge is: 55.3 10^9 core sec + 885 10^9 GPU-core sec
- This pledge can produce: 5.53 10^9 events, limited by cores →
- The value of the pledge corresponds for this workload to: 885 10^9 "HS06"sec
    - The 194 10^9 unused GPU-core seconds are not contributing to the equivalence value because they don't contribute to the throughput

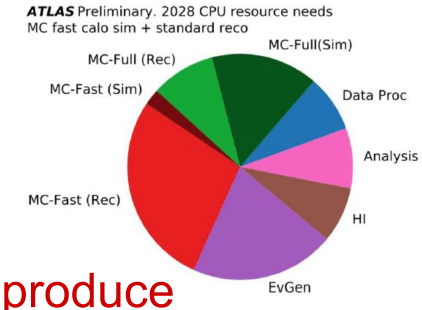
There is more than one workload for an experiment!

- Yes, and for each this approach has to be repeated
- Therefore the equivalence-value differs by workload on the same system and mechanisms have to be used to attract those workloads with the highest equivalence rating to a given resource
  - Until the need for this workload is satisfied
  - This will complicate scheduling on the experiments site…
- Theoretical this becomes a very complex matrix that has to be measured regularly
  - Number of workloads X number of accelerator types X number of WLCG experiment → O(1000)
- In practice each experiment will use 0(10) specific systems with few different architectures
  - Only some of their workloads will make use of the offered accelerators

Why isn’t this extra work?

- When an experiment moves a workload to a given HPC system
    - The flow of jobs to the system has to be tested
    - The code that has been modified to use GPUs has to be verified on this system
    - During these steps the necessary measurements will be taken anyway (to understand the system and ported code)

What does it mean for experiments



- A pledge is only valued at the rate at which the system can produce throughput for the specific workload of a specific experiment
  - A wet neuro HPC with no external network will be rated at 0 “HS06”-equivalent sec for all workloads
- Potential problems:
  - Massive pledges for a specific short time window on resources that can run a narrow subset of workloads
    - We can’t produce a year’s worth of MC-simulation in the first week of March……
  - Large fraction of the resources are too specific to produce all types of workloads needed…
  - A lot of effort is needed to adapt to these systems:
    - New bridge/edge services, new packaging of software, scaling of workflow management systems, …..
    - Code changes …
    - WHO PAYS FOR THIS?????
  - ……..

And the resource owner/funding agency?

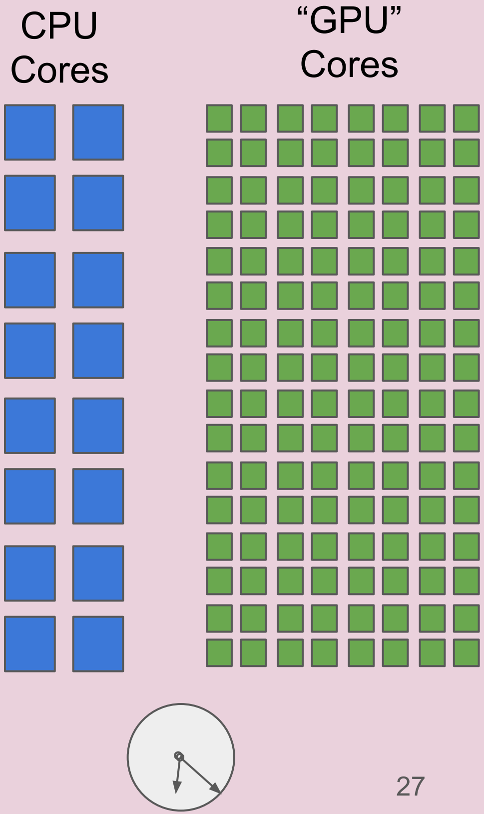
- Can compare the value for the users with traditional resources → costs
- From an Economist's point of view the experiments should not invest in making efficient use of the accelerators!
  - They will get the same "HS06" equivalence with and without investment in optimisation
  - Unless some metrics can be found that can be used as an incentive
  - Unused accelerator time doesn't affect the value of the pledge
  - Porting to an accelerator costs effort → best option is not to port!!!!
- We should add a metric for the utilisation of the potential performance of the resource to indicate how far we are away from making best use of the pledged resource

Metric for ***R**ealised* ***P**otential*

- Modern HPCs are can provide computing power at the exascale
- More and more this depends on accelerators (GPUs etc.)
- Most scientific applications can only exploit a fraction of the potential
  - Often below 20%
- A metric to quantify this is can guide decisions
  - Effort for code adaptation
    - The funding by users **and** providers
  - Tracking of progress
  - Prioritisation of workloads

First naive approach

- In HPC performance is often measured in LINPACK* R_{max} FLOPS (see Top500)
- These numbers are known not only for complete systems, but also for CPUs/cores
  - For accelerators, especially GPUs, benchmarks exists that also produce FLOPS ratings
- These numbers are lower than the theoretical maximum of FLOPS a system can produce, but higher than the number of floating point operations most user workloads can utilise. (well known…)
- The pledged resource can be expressed in LINPACK Floating Point Operations



* soon to be replaced by LAPACK

First naive approach II

- To get an indicator for the level at which the resource is utilised the efficiency of the usage by a specific workload is needed
- For the CPU cores WallTime/CPUTime can be used as CPU efficiency
- For Accelerators similar FLOP utilisation rates can be found
  - Nvidia provides several tools for this, so does Intel (nvidia-smi, intel_gpu_top ….)
- To condense these numbers they have to be converted into the same units
  - Linpack FLOP is a candidate
- Then the efficiency of a workload that uses CPU and Accelerator cores can be estimated by comparing the overall LINPACK-FLOP with the LINPACK-FLOP used (Realised Potential)

Example (very artificial based on first example)

- On a HPC system with 32 cores and 512 GPU cores per node and experiment gets 100 nodes for 200 days
    - Each CPU core is rated at 7 GigaFlops, each GPU core at 3 GigaFlops
    - The total pledge corresponds to: $3\ 10^{21}$ FLOP
- 1 event requires 10 core seconds and 125 GPU-core seconds
    - Running on 8 CPU-cores and 100 GPU cores → 78% of GPU cores can be used
- The efficiency of the CPU cores was 82%, the efficiency of the GPU cores 60% with an effective efficiency of 47%
    - The GPU-cores have to be adjusted for the 78% of cores that can be used
- Therefore during the pledge the workload uses: $1.5\ 10^{21}$ FLOP
- The ratio of the theoretical capability and the used one is: 0.5
- This can be called Realised Potential (RP)
    - This has to be adjusted for the availability/reliability of the resource and the scheduling inefficiencies of the workload management system

What now?

- This ratio can be compared by the resource provider with other workloads
  - From HEP and other communities
- If the ratio is very low the resource provider can either:
  - Prioritise workloads with better efficiency scores
  - Fund effort to improve the score
    - Since the RP score is related to the fraction of the resource utilised it is directly related to the investment and operational cost of the machine
    - funding agencies can make informed decisions on how much effort should be funded to improve the score
    - The score can be tracked to verify that the funded effort has been used effectively
      - This is a bit of an oversimplification….

Why isn’t this straight forward

- HPCs often run a mix of workloads that are complementary in their use of resources
    - Connectivity limited workloads share nodes which are computationally bound etc.
    - Backfilling should be taken into account
- Optimising the RP score can be done without improving the throughput!!!!
    - We assume good will …..
- The cost for accessing the HPC resource is still not taken into account
- We only look at single node workloads
    - Correct for most HEP workloads
    - Event generators (Sherpa) and some analysis code can make use of multiple nodes
    - Will become important in the future
    - Same throughput based approach can be maintained

Summary

- By measuring for each HPC system and each workload an individual "HS06"-equivalent value pledges can be based on the throughput they provide for the experiments
    - At the cost of some complexity
- A second metric is desirable to assess the fraction of the potential of the pledged resource that is exploited (Realised Potential)
- Many costs (for the experiments) are not taken into account!!!!
    - bridge/edge services
    - changes to the workflow and data management systems
    - development/build/packaging
    - Scheduling changes …
    - Maybe a scaling factor can be justified:
        - WLCG sites: 1   HPC sites: 0.x -1 based on complexity of usage