WLCG Strategy
Overview of Strategy document

Ian Bird
GDB
CERN; 11th April 2018

Background

- The goal of this strategy is to set out the path towards computing for HL-LHC in 2026/7.
- There is a strong desire to maximise the physics possibilities with HL-LHC, while at the same time maintaining a realistic and affordable budget envelope.
- The past 15 years of WLCG operation, from initial prototyping through to the significant requirements of Run 2, show that the community is very capable of building an adaptable and performant service, building on and integrating national and international structures.
  - The WLCG and its stakeholders have continually delivered to the needs of the LHC during that time, such that computing has not been a limiting factor.
- In the HL-LHC era that could be very different unless there are some significant changes that will help to moderate computing and storage needs, while maintaining physics goals.
- The aim of this document is to point out where we see the main opportunities for improvement and the work that will be necessary to achieve them.

Strategy

- In 2017, the global HEP community has produced the Community White Paper (CWP), under the aegis of the HEP Software Foundation (HSF).
  - A ground-up gathering of input from the HEP community on opportunities for improving computing models, computing and storage infrastructures, software, and technologies.
  - It covers the entire spectrum of activities that are part of HEP computing.
  - While not specific to LHC, the WLCG gave a charge to the CWP activity to address the needs for HL-LHC along the lines noted above.
  - The CWP is a compendium of ideas that can help to address the concerns for HL-LHC, but by construction the directions set out are not all mutually consistent, not are they prioritised.

- Strategy document – prioritise a program of work from the WLCG point of view:
  - A focus on HL-LHC, building on all of the background work provided in the CWP, and the experience of the past.

Strategy – outline

Themes

1. Software performance
2. Algorithmic improvements/changes
   - E.g. reco, fast MC, event generators
3. Reducing data volumes
4. Managing operations costs
5. Optimizing hardware costs

How do we convince FA's that we are in control of costs, while maximizing physics output?

1. Introduction
2. Computing Models
3. Experiment Software
4. System Performance & Efficiency
   - Cost Model
   - Software performance
   - I/O performance
5. Data & Compute Infrastructure
   - Storage consolidation
   - Caching
   - Storage, access, transfer protocols
   - Data Lakes
   - Network
   - Processing resources
   - Cloud analysis
6. Sustainability
   - Common solutions
   - Security infrastructure
7. Workplan
8. Appendix: technology evolution
9. Appendix: likely benefits

The HL-LHC challenge and Cost Model

- ATLAS and CMS will need x20 more resources at HL-LHC with respect of today
- Flat budget and +20%/year from technology evolution fills part of this gap but there is still a factor x5. Storage looks like the main challenge to address
- Market surveys (in appendix to the document) indicate this 20% might be optimistic, it varies with time and strongly depends on market & economy rather than technology
- Started a cost model providing a quantitative assessment of the prototyped solutions and evolution in terms of computing model, software, infrastructure

Computing Models

- Understand the HL-LHC running conditions and the input parameters arising from them: trigger rates, # Monte Carlo, seconds of data taking, ..

- Pursue aggressively the reduction of data formats
  - Compression
  - Tiering (AOD->MiniAOD->NanoAOD) or slimming the derived formats (e.g. DAODs from trains)

- Rely on less expensive media (e.g. TAPE) for a larger set of formats (LHC data is generally “cold”).
  - Implies evolution of the facilities and the workflow and data management systems

- Review the centralized processing and the analysis models. Shift more workload in the direction of organized production.

Software

- With today's code the performance is often very far from what modern CPUs can deliver.
- Due to a number of factors:
  - construction of the code,
  - not being able to use vector or other hardware units,
  - layout of data in memory,
  - end-end I/O performance
  - Etc.
- With some level of code re-engineering, it might be expected to gain a moderate factor (x2) in overall performance.
- This type of activity was the driver behind setting up the HSF, and remains one of the highest priority activities.

- It requires the appropriate support and tools,
  - E.g. the need to fully automate the ability to often perform physics validation of software. Essential to be adaptable to many hardware types and frequent changes and optimisations

- It requires that the community develops a level of understanding of how to best write code for performance, again a function of the HSF

Algorithmic improvements

- For HL-LHC anticipated pile-up means that current reconstruction algorithms must be improved significantly to avoid exponential computing time increases.
  - It is estimated that a considerable improvement could be obtained with some tuning of current algorithms,
  - New approaches could have larger benefits.
  - This requires expert effort to achieve, there is already a working group on reconstruction as a community effort.
- Full or partial use of fast Monte-Carlo in place of full Geant simulations.
  - There is a huge potential saving. May suggest that 50% of overall MC could be fast MC, which could provide close to a factor 2 improvement.

Software Performance

- Review the data layout (EDM) as one of the main bottlenecks in dealing with I/O
- Define and promote programming styles suitable for different areas of development
- Invest in developing a more automated framework for physics validation evaluating numerical differences and the impact of physics
- Evolve the code in the direction of modularity, to allow exploring capabilities of future hardware. Make the computational code sequences more explicit and compact
- Engage in ad adiabatic code refactoring, focused on efficient use of memory and generally on performance oriented programming

Event generators

- As the precision of the experiments increases the generators need to simulate higher-order effects, and the related computing time is now becoming significant, and is expected to grow towards HL-LHC.
  - The generators need to gain very large factors of improvement to prevent this from becoming a problem.
- There are 2 aspects,
  - the optimisability of the code itself,
  - the capability of weighting effectively rather than generating huge numbers of filtered events.
- The generator community must take this in hand.

Reducing data volumes

- A key cost today is the amount of storage required.
- Mechanisms for reducing that volume will have a direct effect on cost
  - removing or reducing the need for intermediate data products that must be stored,
  - managing the sizes of derived data formats, for example with “nanoAOD”-style even for some fraction of the analyses will have an important effect.
- There is a big potential here, but needs work from the experiments.

Managing operations costs

- Here there are a number of strategies.
- Storage consolidation is a high priority.
  - The idea of a “data-lake” where few large centres manage the long-term bulk data,
  - Access to processing managed through streaming, caching, and related tools,
  - Can help minimise the cost of managing and operating large complex storage systems
- Reduces complexity for the experiment.
  - Gives the opportunity to move common data management tools out of the experiments and into a common layer.

- This allows better optimisation of performance and data volumes, easier operations, and common solutions.
  - It also makes it easier to introduce common workflow solutions.
  - Storage consolidation can save cost on expensive managed storage, but requires that we are able to hide the latency via streaming and caching solutions.
  - Feasible as many of our workloads are not I/O bound, and data can be streamed to a remote processor effectively with the right tools.

Optimising hardware costs

- There is an opportunity to reduce storage cost also by more actively using tape (or cold storage).
  - With a highly organised access to tape it could replace the need to keep a lot of data that is today kept on disk.
- The judicious use of virtual data (re-create samples rather than store) is another opportunity.
  - This could save significant cost, but requires the experiment workflows to be highly organised and planned.
- Moving away as far as possible from random access to data
  - except for the final highly refined analysis formats
- Other considerations include
  - the optimisation of the amount of storage vs compute, and
  - optimising the granularity of data that is moved - between dataset level and event level.

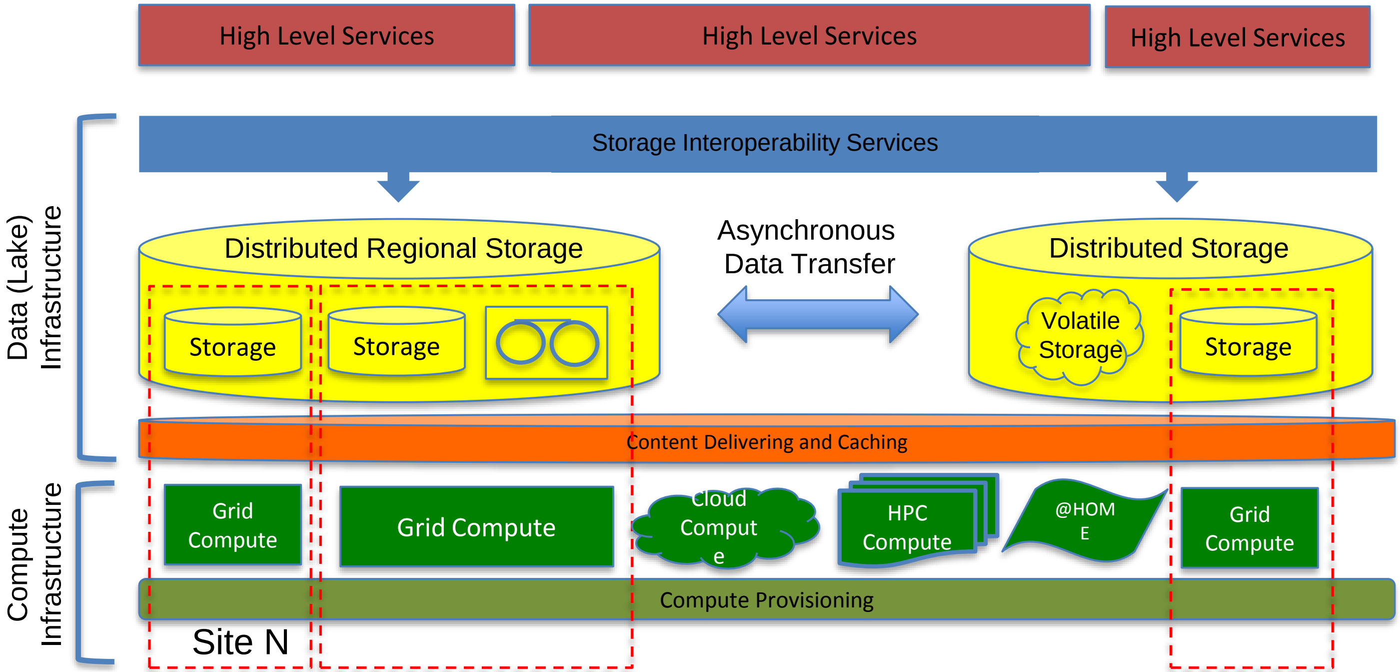
Interoperability and Data Preservation

- Review the security model and evolve it toward federated identities. Move away from X509, prototype a token-based solution ensuring interoperability and sustainability

- Favor common solutions across the stack (from high level services to infrastructure).
  - A very strong message in this direction from all funding agencies: little or no support in the future for experiment specific solutions.

- This is the basis for a data and analysis preservation strategy.

Data and Compute Infrastructures



Strategy – outline

<u>Themes</u>

1. Software performance
2. Algorithmic improvements/changes
   - E.g. reco, fast MC, event generators
3. Reducing data volumes
4. Managing operations costs
5. Optimizing hardware costs

How do we convince FA's that we are in control of costs, while maximizing physics output?

1. Introduction
2. Computing Models
3. Experiment Software
4. System Performance & Efficiency
   - Cost Model
   - Software performance
   - I/O performance
5. Data & Compute Infrastructure
   - Storage consolidation
   - Caching
   - Storage, access, transfer protocols
   - Data Lakes
   - Network
   - Processing resources
   - Cloud analysis
6. Sustainability
   - Common solutions
   - Security infrastructure
7. Workplan
8. Appendix: technology evolution
9. Appendix: likely benefits

R&D projects

- Specific R&D projects are being proposed
  - Should have explicit timelines, goals, metrics, etc.
  - See later for first proposals
- Integrate with existing working groups where practical
- Use GDB slots to engage community and show progress/manage direction