WLCG Strategy towards HL-LHC

Executive Summary

The goal of this document is to set out the path towards computing for HL-LHC in 2026/7. Initial estimates of the data volumes and computing requirements show that this will be a major step up from the current needs, even those anticipated at the end of Run 3. 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. However, 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.

During 2017, the global HEP community has produced a white paper - the Community White Paper (CWP), under the aegis of the HEP Software Foundation (HSF). The CWP is 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, nor are they prioritised. That is the role of the present document - to prioritise a program of work from the WLCG point of view, with a focus on HL-LHC, building on all of the background work provided in the CWP, and the experience of the past.

At a high level there are a few areas that clearly must be addressed, that we believe will improve the performance and cost effectiveness of the WLCG and experiments:

- **Software:** With today's code the performance is often very far from what modern CPUs can deliver. This is due to a number of factors, ranging from the construction of the code, not being able to use vector or other hardware units, layout of data in memory, and end-end I/O performance. With some level of code re-engineering, it might be expected to gain a moderate factor (x2) in overall performance. This activity was the driver behind setting up the HSF, and remains one of the highest priority activities. It also requires the appropriate support and tools, for example to satisfy the need to fully automate the ability to often perform physics validation of software. This is essential as we must be adaptable to many hardware types and frequent changes and optimisations to make the best use of opportunities. It also 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**: there are clear needs in this area. For HL-LHC the level of pile-up anticipated 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, but new approaches could have larger benefits. This requires expert effort to achieve, but there is already a working group on reconstruction as a community effort. Another aspect is the full or partial use of fast Monte-Carlo in place of full Geant simulations. There is a huge potential saving. It may be realistic to propose that 50% of overall MC could be fast MC, which could provide close to a factor 2 improvement.
- **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, but also 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. Investigating 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. Investigating the opportunities with storage consolidation is a high priority. The idea of a “data-lake” where few large centres manage the long-term data, while needs for processing are managed through streaming, caching, and related tools, allows the cost of managing and operating large complex storage systems to be minimised. It also reduces complexity for the experiment. Importantly, such a structure 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. This is 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 before 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.

In the following we refer often to a "data-lake". This is really a label for a system that allows us to consolidate storage at various scales (national, regional, global). It is not just a mechanism for saving hardware and operational costs, but is also a real opportunity for building commonality, moving data management into the infrastructure layer, common optimisation of performance, common operations, and much better prospects for the long-term sustainability of the solutions. Of course, the promise of these potential opportunities must be demonstrated, and that is the goal of the program of work that we set out in the rest of this document.

There will be a WLCG Technical Design Report (TDR) for HL-LHC computing to be prepared in 2020. That point will be a major milestone in the work proposed here.

1. Introduction

The HL-LHC computing challenge is driven by the expected increase in event rate (between a factor 5 and 10 with respect to Run-2) for both data and Monte Carlo and the increase in event complexity, as we expect approximately 200 proton interactions in the same bunch crossing on average (compared with at most 60 during Run-2). Both ATLAS and CMS estimated the HL-LHC resource needs, projecting today's computing model with the parameters (event processing times, event sizes, running conditions) expected at HL-LHC. Both experiments concluded that in this naive scenario they will require approximately 20 times more resources with respect to today. The resource needs of Alice and LHCb have also been examined: they present a challenge already in preparation for Run-3 and they are being addressed in the respective computing TDRs. We do not foresee an increasing challenge for those experiments between Run-3 and Run-4.

The LHC funding agencies expressed very clearly that we should not expect the budget for LHC scientific computing to increase. Technology evolves and more processing power and storage can be acquired every year for the same cost, but not at a level of a factor 20 in the next 8 years. Appendix A in this document surveys the main market trends for hardware and conclude that: Moore's Law and Kryder's law have slowed down; The Device and Server markets show signs of saturation; Technology evolution is also slowing down due to the increasing complexity and large-scale investments needed; The current expectation for annual price-performance improvements over the next few years are 15% for CPU servers, 25% for disk storage and 20% for tape storage.

This average 20% gain in hardware capacity thanks to technology evolution will buy 4 times more hardware in 2026 for the same budget as that of today, which means we still have a gap of a factor 5 between what we need in 2026 and what flat a budget can provide. Such an estimate could be optimistic, since it does not take into account unforeseen or unaccounted increasing complexity or needs of the HL-LHC physics program and the derived consequences for computing. It does not consider the possible consequences of the increase in scale of the computing infrastructure and the inefficiencies arising from it. It also does not consider that market trends are very difficult to predict beyond a few years time and some of the expected gains might never materialize. Finally, some technologies on which

today we base our strategies for reducing cost, such as tape media, might not be so economical in the timescale of 10 years from now or even exist at all.

Conservatively, we need to set up a program to reduce the cost of LHC computing by almost one order of magnitude, while providing the adequate resources for the physics program. This reduction must be fulfilled in time for HL-LHC. The HEP Software Foundation recently produced a whitepaper[1] defining a roadmap for HEP software and computing in the 2020s ("Community White Paper - CWP), with the HL-LHC computing challenge as one of the main drivers. In this strategy document we present a WLCG specific prioritization of the R&D activities identified in the HSF community white paper. This prioritized list will serve as a baseline for delivering a WLCG TDR at the beginning of the next decade. We base our strategy on a few considerations about the cost and mission of the WLCG computing infrastructure. Data related aspects are the largest cost in WLCG today: in terms of hardware, storage (and particularly disk) is the highest cost in most countries (wrt CPUs); for the facilities, storage is the service that requires most effort in terms of operations. At the same time data is the highest value of the LHC scientific program and its curation is the highest WLCG priority: we need therefore to retain in-house ownership of our data, simplifying the infrastructure to reduce the cost and leveraging resources, such as the network, that are expected to increase at steeper rate than storage for the same cost. Compute capacity is by definition more volatile as it has the lifetime of one processing job (tens of hours at most). It can therefore be provided in several ways at different kind of facilities: from WLCG grid centers to High Performance Computers to academic and commercial clouds. The challenge for compute is to be able to flexibly provision resources on such different architectures, adapt the software and workflows and serve the data to the CPUs in an efficient manner.

There are several themes or areas where the computing costs can be addressed through R&D potentially leading to gains in overall cost, and optimising the physics output. These high-level areas consist of Computing Models, Experiment offline software, System Performance and Efficiency, Data and Processing infrastructures. The following sections will go through those areas and highlight the aspects that should be considered and prototyped. Sustainability, Data Preservation and Analysis Preservation must also be included in the long term planning.

2. Computing Models

The experiment data and computing models will be reviewed in the next few years in order to adapt to the running conditions of HL-LHC; several elements will impact the cost of computing:

2.1. The boundary between online and offline has been reviewed by Alice and LHCb for Run-3, while ATLAS and CMS will continue with the same model used in Run-2. It will be important to understand pros and cons of alternative models and what is the impact in terms of resources.

[1] Community Whitepaper: https://arxiv.org/abs/1712.06982

2.2. The HLT output rate will likely be one of the main drivers of the cost, as it will determine the amount of LHC data to process and store. Based on consideration of the expected HL-LHC instantaneous luminosity, values between 7.5 and 10 kHz have been considered in the past by ATLAS and CMS. Understanding the possibility to reduce the HLT output data rate without affecting the physics reach of the HL-LHC program should be one of the main goals in preparation for the TDR. For Run-2 an agreement was reached by ATLAS and CMS on what the expected HLT output rate should be (1 kHz) and it would be useful if the same consensus could be reached for HL-LHC.

2.3. Currently, the amount of resources needed in order to produce and store the required Monte Carlo samples represents between 50% and 75% of some experiment's computing budget. While this number is expected to decrease in HL-LHC because of the increased relative cost of reconstruction with respect to detector simulation, it will remain considerable and larger than 50%. Understanding the amount of Monte Carlo needed at HL-LHC will be critical to understanding the cost, and reducing those needs, again without affecting the physics reach of the experiments will be a key aspect.

2.4. Storage is the main driver of the WLCG hardware cost. The experiment Data Models are in continuous evolution and will likely be revisited in preparation for HL-LHC. Different experiments implemented different approaches in terms of data tiers: ATLAS produces O(100) DAOD (derived AOD) formats from AODs through centralised analysis trains, while CMS produces a single MiniAOD format and is looking into a further reduced data tier (and is experimenting switching to this NanoAOD). There is an opportunity to develop a model for HL-LHC which leverages the main advantages from the two approaches and reduces the storage needs, preserving or further improving the user experience. Several formats have been abandoned or descoped during Run-1 and Run-2, once the experiments gather a better understanding of the detector and develop refined tools and techniques particularly for calibration and alignment. It will be important to understand which formats will play a role at the start of HL-LHC and during its evolution. The kind of information stored in each format will also be reviewed, again trying to optimize cost and usability.

2.5. Related to data format, data compression, retention policies and access patterns will drive the inherent cost of data replication. The experiments will review data retention policies based on the experience of previous runs and the evolution of the data model. The cost of the HEP infrastructure can be reduced by exploiting adapted quality of services from different storage technologies and therefore a study of the role of tactical storage and of different archival technologies should be made. Those aspects will be elaborated more in Section 5 of this document. Data reproducibility also impacts the resource needs: the possibility to reproduce derived data in an automated manner would reduce the needs for storage for data retention. The possibility to rely on Virtual Data, i.e., storing the necessary information to produce the data rather than the data itself, should be studied and prototyped

as a challenging but effective possibility to economise resources through data reproducibility.

2.6. The HL-LHC data processing model will also need to be reconsidered. The experiments will understand in the coming years their expectations in terms of number of processing and reprocessing campaigns and the impact of the resources. It will be important to understand the time constraints of the different campaigns as well, to understand the resource usage profile and different options in terms of provisioning (dedicated hardware, elastic capacity, shared capacity). The analysis model will also play an important role and the experiments will understand how much of today's chaotic activity can be centralized and scheduled in planned and organised workflows.

3. Experiment Software

Evolving the software in a direction more tailored to the high particle density environment of HL-LHC will allow a resource reduction, particularly of CPU needs. The CWP identified several areas of R&D in the area of offline software where the community should invest effort to reduce the computing needs, while delivering the expected physics precision:

3.1. Particle Generators require an increasing level of precision, from Leading Order (LO) in LHC Run-1 to Next-to-Next-to-Leading Order (NNLO) in HL-LHC, which implies an increasing CPU time per event. On the positive side, the same generators are used by many experiments and the same underlying libraries are used by many generators, therefore collaborative work in this area would bring common benefits. Chapter 3.1 of the Community White Paper identifies R&D activities around two main themes:

3.1.1. Improving event filtering and reweighting, allowing de facto to generate less events for providing the same sample statistics.

3.1.2. Improving the parallelism and concurrency of the generator code, allowing the software to exploit at best modern hardware and facilities. In general, a strong collaboration between theorists and software experts is needed.

3.2. Detector Simulation and Digitization activities today are the main consumer of CPU in WLCG and will require considerable resources in HL-LHC. Geant4 is the common software package for detector simulation and improvements at this level will be a benefit for all experiments. Similarly to the case of particle generators, the Community White paper identified several areas of R&D (see Chapter 3.2):

3.2.1. Improving the physics description of the processes, adapting to new use cases in modern detectors.

3.2.2. Modernizing the software to be able to leverage vectorization and multithreading, so that it can more optimally exploit modern hardware architectures.

3.2.3. Evaluating different strategies at the level of signal vs background event mixing need to be prototyped and tested. Event pre-mixing and

fast simulation of background samples seem the appropriate starting point. The simulation software needs to be modularised to be able to complement full and fast simulation algorithms in the same event sample.

3.3. Reconstruction tasks are expected to adsorb an increasing fraction of computing resources by the time of HL-LHC. Some workflows, like tracking on silicon detectors and jet clustering on high granularity calorimeters, show a superlinear behavior in time per event at increasing pile-up, due to their combinatorial nature; this needs to be combined also with the increased channel count planned for the Phase II detectors. In general, it is expected that novel reconstruction techniques will be needed in order to fit into a reasonable HL-LHC computing budget. Lines of R&D include:

3.3.1. Using enhanced vectorization techniques, which allow for a larger transistor utilization in recent CPUs;

3.3.2. Using computing frameworks that are adapted to allow an easier utilization of many computing cores - ultimately this appears to be the only way to avoid unaffordable increases in the cost of memory;

3.3.3. The potential of using accelerators in order to help event reconstruction; this includes GPGPUs, FPGAs and possibly custom ASIC chips and implies the need for software able to efficiently run on heterogeneous architectures;

3.3.4. In general, we expect reconstruction code to become more complex, and more difficult to write; this can be partially mitigated by experiment frameworks and modern software technologies like OpenCL, or client-server architectures encapsulating heterogeneity.

3.4. Fast simulation will play a critical role in reducing the cost of HL-LHC computing. Depending on the level of precision that can be achieved, a varying fraction of HL-LHC simulation could be done through fast simulation. Beside parameterized detector simulation, various solutions should be evaluated and prototyped at all levels of the simulation chain. Examples are parametrized digitization and truth seeded reconstruction.

3.5. Machine Learning techniques offer an alternative approach to many of the current methodologies, from data analysis to event selection to anomaly detection and data quality. In the next two years the potential impact of ML techniques in reducing computing cost should be estimated.

4. System Performance and Efficiency

4.1. Cost Model

The computing challenge in HL-LHC will consist in delivering the compute and storage capacity for affordable cost (the same cost as today, i.e., a flat budget). To meet this challenge we need to understand the relationship between the performance of the WLCG system, its cost and the implications in delivering the service to the experiments. We need therefore to build a cost model which we can use to understand the impact of different future strategies quantitatively . The model should take into account the cost of the hardware, infrastructure and operations and provide

a quantitative assessment for any proposed change in terms of computing model, workflow model, data placement, data access and data processing strategy, offline software evolution. The initial part of this work, which will be an iterative process, started with the Performance and Cost Model working group and consists in understanding which are the relevant metrics, identifying the important workflows and collecting tools to benchmark the system performance.

The Performance and Cost Model working group identified the following areas as central to achieving these goals:

4.1.1. Identification and description of the most computing and storage intensive workloads for each experiment, concentrating on the data transformations. This needs to be done within the scope of the current computing models and be expanded to the future models

4.1.2. Provide access to these workloads so that measurements can be performed independently

4.1.3. Provide a testbed, covering different architectures to assess the impact of workloads on different resources.

4.1.4. Define and identify or implement the means to measure a metrics that characterises the resource utilisation of our workloads. This is the most critical activity since the modeling will express all needs of an application based on these quantities.

4.1.5. Develop a cost evaluation process to map given resource needs to local costs and efforts

4.1.6. Develop a model that describes the time dependent resource utilisation, this might require at some stage discrete event based modeling

4.1.7. Build a common resource request calculation model where the specific aspects of different experiments become parameters.

4.1.8. Identify and promote performance analysis tools and develop a common vocabulary

All these activities are already active focussing at the current existing workloads.

4.2. Software Performance

Steady progress in offline software performance has been made over the years and common patterns concerning the internal structure of the code have emerged. While optimisation of the data layout in memory and code simplification has brought gains, in general no significant hotspots with easy improvements remain and resource usage is spread across hundreds to thousands of methods. Three main aspects of those common patterns need to be considered:

- A high level of abstraction coming from our use of C++ as descriptive language frequently obfuscates the nature of the calculations (an example is replacing loops with functions/methods) and the underlying data layout. Because of this, even compiler tools struggle optimizing the code in terms of

parallelism and vectorization, even if the mathematical concepts behind inherently lead to vector operations.

- Current object design and data structures, often directly determined by the EDM, can impact on the efficiency of data access and is can be unsuitable for concurrent processing. The allocation of collections of objects one at a time and discontinuously is slow for later iteration over the collection. This allocation pattern is not at all suitable for using accelerators, requiring slow and expensive rewriting of objects. It also increases the cost of object (de)serialisation and data that is needed in the same processing step is more likely stored non-contiguously, which complicates efficient I/O on the hardware level, creates overheads and makes efficient readahead over WANs difficult. There is broad agreement in the community that the fundamental reason for this pattern is the way object oriented programming has been adopted by HEP. In addition frequent allocation and deallocation of small pieces of memory for transient objects is observed. This has the effect of creating memory management overheads and scattering data throughout the memory, reducing the effective speed at which it can be accessed.
- The current code sometimes lacks a modular scheme enabling a switch between code paths during compilation, while retaining the core physics logic. This is mostly due to the way we retain legacy code, through inheritance and templates. As a consequence, it is difficult to leverage architecture specific features of hardware, with loss in performance as a result. Furthermore, when porting software to different architectures, we lack a lightweight and automated system to evaluate the impact of numerical differences and today the physics validation process of the results is tedious work requiring experts time.

The HEP community has identified several areas of potential improvement in software performance along those lines and work in the experiments started already. Revisiting the current Event Data Models with a focus on efficiency of storage, remote data access and processing is also critical for this activity and is elaborated more on the next section on I/O performance.

4.2.1. Define and promote C++ programming techniques suited for addressing performance in the different areas of developments. This requires a more formal approach by codifying styles and setting up the necessary educational activities to reach a sufficiently high percentage of the community. Training has to be offered at different levels, from the novice developer to the highly experienced experts. Several activities in this direction have started independently in different experiments and schools and these activities should be enlarged and coordinated. Chapter 4 of the Community White Paper offers more insight.

4.2.2. Invest in developing more automation for physics validation, evaluating numerical differences and possibly their impact on physics. The focus is in developing procedures to facilitate code refactoring, where differences in the physics distribution coming from different and

improvements algorithms are not expected. Differences arising from architectures and compile strategies should be handled automatically and only in rare cases the intervention of an expert should be needed. Without this ability deep changes to the current code base become practically impossible. As such the availability of automated physics validation is one of the fundamental enablers for progress.

4.2.3. Evolve the code in the direction of modularity, both functional and for different target architectures. This will help exploiting the capabilities of current and future hardware. The sequences of the code that are carrying out the computational work need to be made explicit and compact so that porting to different platforms can be achieved.

4.2.4. Eventually the community should pursue refactoring the code, focusing on the efficient use of memory and the capabilities of modern hardware as the foundation for further improvements. Refactoring this code using performance oriented programming styles and data constructs will not only help with improving the efficiency on general purpose architectures, but also will ease the exploitation of different computing architectures and increase the maintainability of the code. Refactoring the code should not be confused with redeveloping the code and in fact can follow an adiabatic approach, starting from the areas of larger gains.

4.3. I/O performance

Based on an analysis of the I/O performance and its interdependence with the EDM for the different data transformations, a development program to improve the I/O performance for those workloads that require significant data movements and access is needed. The consolidated (or federated) distributed storage approach (also referred here as “Data Lake”) is expected to lead to a system that relies on a smaller number of sites, connected by high speed networks. This will lead to a larger fraction of remote data access for which the effect of EDMs with a granularity different than the optimal size for storage and network I/O will be critical.

4.3.1. The performance difference between scheduled pre-staging and caching on demand has to be understood for the different workloads. Within the storage hierarchy the massively different latencies, ranging from microseconds to hours, will require different strategies for the transition and the aggregation of data into units suitable for the exploitation of the different systems used at the different tiers.

4.3.2. The optimal granularity for data management and data access have to be understood for all tiers in the hierarchy and workloads. This will lead to the definition of suitable aggregation layers that combines objects that are moved together between the storage tiers. The elementary objects for data access and data management needs to be defined.

4.3.3. Storage and access requirements have to be understood and for some activities, like event streaming and caching have to be decoupled. For example, compression to reduce storage cost would favour a specific data organization while optimizing caching and latency hiding may require a different one. To allow both representations to exists in parallel a transformation has to done. To decide where this is best done, at the storage service, caching layer or client layer, requires R&D work.

4.3.4. Over the years ROOT has been providing the shared solution when it comes to implementing the experiments data structures. When developing new EDMs and exploring new approaches for storage and access of data ROOT is the natural level on which commonalities are leveraged. For this it is essential that already prototype work is carried out within this context. Latency hiding, bandwidth leveling and shifting packaging/unpacking workloads between client and server are examples for areas where commonalities can be implemented via ROOT.

5. Data and Compute Infrastructures

The WLCG storage and computing resource capacity is expected to grow by roughly a factor 4 in the next 10 years, if we assume flat funding and extrapolate the current growth. Again extrapolating the current patterns, the LAN and WAN network connectivity for the largest centers (today's T1s and major T2s) is expected to grow by up to a factor 100 in the same time frame. The HL-LHC data and processing model therefore would leverage the network capacity and reduce the relative storage needs by:

- Consolidating storage resources in a smaller set of larger data centers, from O(100) we have today to O(10); one large data center could be geographically distributed in several physical locations connected by fast enough network.
- Leveraging processing (compute) resources at a much larger number of heterogeneous facilities, some of which might host the data;
- Enabling the capability to process data remotely and/or cache the data in volatile storage.

One needs to keep in mind that the foreseen increase in bandwidth will surely benefit bulk data transfers but not necessarily client access performance which today is already dominated by network latency. The imbalance in transfer efficiencies between two areas will continue to increase and the impact needs to be carefully studied and alleviated. The following R&D activities will allow in the next two years to evaluate and measure the expected gains in in the above scenario. The R&Ds are described in more details in Chapter 3 and Chapter 4 of the Community White Paper [CWP].

5.1. Storage Consolidation

The storage services are the most expensive and complicated to operate (compared for example to compute and network) at WLCG sites, according to a recent study[2]. Reducing the number of storage endpoints will therefore reduce cost and we expect a relatively large number of sites to run processing services only. Other centers will continue operating storage but, especially at national level, consolidate different endpoint into a single distributed instance, spanning different sites. We should try to understand which technologies will allow to do this efficiently, both in terms of performance and cost.

5.1.1. For a given existing WLCG storage technology, build a prototype with one entry point for the namespace and data pools spread across few data centers. Such data centers should be connected by at least a 10Gb/s network with a reasonable latency (to be determined). Apply a simple policy for data locality such as data is available in one of the geographical locations.

5.1.2. Test local and remote data access in the described setup for different experiment workflows. Files should be accessed through a protocol enabled in ROOT I/O, which might differ from the protocol used for internal storage management. The measurements will provide the reference values for more complicated setups and allow to validate the simplest scenario where sites will consolidate the storage into one distributed instance and run processing capacity

5.1.3. Implement a more sophisticated storage setup and data replication policy. For example, with 2 copies of the same data spread across multiple (say 3) locations. Perform the same measurements as in 5.1.2., relying on the storage technology to serve the “best” replica of the file to the CPU node. This setup will allow to validate a more realistic scenario where different sites will consolidate the storage and leverage the distributed nature of the instance to ensure data custodiality (multiple replicas of the same file) and facilitate data access from CPUs at the same site.

5.1.4. Perform the same studies as in 5.1.3. but accessing data from CPUs that are not located at any of the site offering storage. Start with a processing site still in the “proximity” of the storage, but then test the case where the processing site is far away (> 30 ms) from the storage.

5.1.5. Introduce a caching mechanism based on the technology investigated in the R&D 5.2 (Caching Technologies) where the remote processing site in 5.1.4. is enabled with a cache populating the data from the distributed storage. Consider different access patterns based on what the caching technology can support. Repeat also the measurements in 5.1.2 in presence of a cache. Assess for which workflows and in which scenarios caching data at a local storage is preferred with respect of directly reading data from remote storage.

[2] https://twiki.cern.ch/twiki/bin/view/LCG/WLCGSiteSurvey

5.2. Caching

5.2.1. Prototype a caching technology/strategy based on a ROOT-supported protocol, such as xrootd, HTTP, posix. File, event, object (sub-event column-wise) and block level caching solutions should all be investigated and prototyped. Define the behaviour of the cache in case of “miss” (file/event not found) such as:

5.2.1.1. Serve the file/event from the persistent storage to the client, while the cache is being populated

5.2.1.2. Populate the cache and serve the client from the cache (while the cache is being populated or after the cache has been populated)

5.2.1.3. Identify a model for cache management, including size and policies, to minimise trashing of the cache content.

5.2.2. Different content delivery methods should be looked at, namely a Content Delivery Network (CDN) and Named Data Networking (NDN) approach.

5.3. Storage, Data Access and Data Transfer Protocols

5.3.1. Ensure that the WLCG data management services will not require the existence of SRM in the future (with some caveats for tape access) and that functionalities today relying on SRM can be achieved with other means. The SRM interface served well as abstraction layer to access and manage the storage, but came with sometimes too large overheads impacting performance. In addition, requiring such interface to storage in the future would imply precluding the use of more modern, open source and widely adopted technologies (such as S3 storage). A study should include also access to tape which today relies entirely on SRM.

5.3.2. Investigate and test alternative protocols to gridFTP for data transfer. Using the protocol should not rely on the existence of SRM, both in terms of functionality and performance. Ideally, the same protocol could be used for both data transfer and data access (so being supported in ROOT I/O), reducing the number of protocols to be supported by the storage. The protocol should support 3rd party data transfer to enable scheduling through a service such as FTS. The protocol should be able to efficiently handle transfers of large files (20 GB range) as well as small files (1 MB range) over short (30ms) and long (200ms) distances. The possibility to consistently enable vector reads would allow to reduce the actual data traffic and should therefore be pursued. Lightweight authentication and support of session reuse is an important point to consider. The case of file sizes well below 1MB and above 10 GB should to be studied as well, at

least as proof of concept. The protocol should also allow checksum verification during transfer for common algorithms such as adler32.

5.3.3. Investigate different solution and optimisations at the level of the file access protocols and at the level of ROOT to reduce the impact of latency when reading remote data. Such solutions should include caching events or object in memory (TTreeCache) and asynchronous pre-fetching.

5.3.4. In general the constraints of access protocol and the constraints of storage implementation should be reconciled. The "object" size for data access needs (after TTreeCache) to have a minimum size with respect to bandwidth*latency to be efficient. The object size on the storage side and on the data management side impose quite some scalability challenge on each component implementation (e.g. namespace size). Still these "objects" are completely different things coupled by data access client (ROOT) and by data transfer (e.g. FTS). The balance between these connections is being looked at and should evolve to achieve better overall efficiency.

5.4. Data Lakes

Data lakes are an extension of storage consolidation, where geographically distributed storage centers, potentially deploying different storage technologies, are operated and accessed as a single logical entity.

As of today, experiments already rely on a certain level of non co-location between data and processing units, as they all implemented techniques and workflows for remote reading of data. Read-only storage federations such as AAA and FAX are a typical example.

Initially, storage centers in a lake should be connected through high bandwidth links (> 10Gb/s in 2017) and relatively low latency (< 100 ms). These parameters need to be evaluated in the R&D, however it may be that a single data lake would not span more than one continent.

5.4.1. Implement a relatively small distributed storage system, spanning more than 3 centers based on a technology evaluated in 5.1 (Storage Consolidation). The connectivity between centers should be adequate, as described above and the hardware performance and reliability as homogeneous as possible.

5.4.2. In this simple setup, study possible ways to implement different file replication and retention policies and understand how applying them to different kind of data would reduce the cost, preserving and possibly improving accessibility. Redundancy should be considered both at the level of hardware (RAID or erasure coding) and software (replication)

in a complementary manner, leveraging the geographically distributed nature of the system.

5.4.3. The security model (Authentication, Authorisation, Auditing) should also be studied at this early stage, exploring the possibility to treat the distributed storage as a unique administrative domain, for the benefit of efficiency. The implications of AAA should be evaluated against different criteria, such as efficiency, compliance with local policies, accessibility, information and data protection.

5.4.4. Prototype the possibility to attach an existing storage system to the lake as a “pool” and understand the implications for the above points. The existing storage system would be possibly based on a different technology and come with its internal data retention strategy. The existing storage system might come with its own namespace and internal structure. Start with a simpler existing storage, based on a mountable file system or object store. Storage solutions based on GPFS or CEPH are one example. Understanding the interplay between the lake storage and the underlying storages attached to it is a key aspect: ideally one would like to be able to manage data both with the native storage frontend and through the data lake gateway in an interchangeable manner.

5.4.5. Consider a setup where different nodes (or partitions of a node) consist of different hardware technologies such as very different IOPS specifications (HDD vs SSD) or latency (HDD vs TAPE). Prototype a solution where the lake behaves as hierarchical system and optimises the data organisation based on policy (first) and usage (after). Implement different QoS and retention policies based on the user requirements and usage patterns

5.4.6. Similar to the previous point, prototype the possibility to attach a volatile storage to the lake. The volatile storage should be used as tactical storage, hosting a redundant set of data, to optimise data access and the system should auto-recover in case the volatile storage disappears

5.4.7. Investigate the scenario where data in the lake needs to be processed at computational resources outside the lake, with a connectivity, say, O(10) lower with respect of what one can expect within the lake. In particular, understand the needs for a caching layer to be deployed at/near the computing sites or as a distributed content delivery system. Understand how this caching layer interplays with how the workload is scheduled on these resources.

5.5. Network

Networking will play a central role in HL-LHC as enabler for HEP computing. On the non technical aspects, WLCG should continue engaging with Funding Agencies and NRENs ensuring enough capacity is made available and the LHC traffic does not get segregated below a critical level. The Data Lake R&D should contribute defining what that critical level will be in the mid 2020s. The WLCG community should also understand the interplay between R&E networks and commercial networks in case of hybrid solutions where part of the resources are deployed in commercial cloud providers and some in research facilities. In particular, one needs to ensure that Acceptable Use Policies of R&Es and security considerations impact or limit the available network resources for HEP. On the technical side, several R&Ds should be launched to study how to better leverage the network resources in the data and processing infrastructures for HL-LHC

5.5.1. Network protocols at lower level than gridFTP should be studied and alternatives to TCP should be considered for the main use cases in WLCG with respect of data transfers and data access on the LAN and WAN. Consider both the scenario of dedicated network resources (such as the LHCOPN) or shared resources (like LHCONE)

5.5.2. Understand the potential benefits of programmable networks by enabling a prototype based on a SDN technology. Study how network scheduling could be integrated in the data lake software architecture and in the end-to-end transfers scheduled by the experiment's data management system.

5.5.3. Evolve the WLCG network monitoring infrastructure to collect more real time information concerning the healthiness and status of the network itself and expose them to the applications for adaptive network use.

5.5.4. Use of commercial cloud resources only accessible via the public Internet may require the use of overlay networks to securely attach such resources into the HEP computing system. The HEP community would greatly benefit if such connections could be done using standard protocols and mechanisms, independent from cloud and network providers. The HEP community should promote the development of a standard way to securely and efficiently connect remote cloud resources without compromise on performance and accessibility.

5.5.5. The possibility to leverage a caching system built into the network infrastructure and operated by the National Research Network providers should be evaluated. This is the model adopted by commercial Content Delivery Network providers and has obvious benefits in the way the traffic can be shaped and provisioned to the cache.

5.6. Processing Resources

Processing resources available to WLCG at the time of HL-LHC will be less co-located with the input data as we are used today. They will also be offered at a more heterogeneous set of facilities: as part of batch resources accessible through some sort of Grid middleware, as cloud resources, as allocations in large HPC facilities. R&D activities in this area should look into:

5.6.1. Explore scalable and uniform means of resource provisioning which incorporate dynamic heterogeneous resources. Ideally such provisioning layer could be commonly adopted by all experiments.

5.6.2. While data co-location with processing units will be extremely relaxed, an adequate data and cache aware brokering should be in place to favor co-location when convenient. Brokering workload and workflow management will probably remain an experiment specific aspect, however it should be investigated how commonality can be leveraged at this level as well, at least for the principles of data discovery and data awareness

5.7. Cloud Analysis Model

5.7.1. Prototype and evaluate a quasi interactive analysis facility that would offer a different model for physics analysis and would also be integrated into the data and workflow management of the experiments. Leveraging on the data lake experience, build a demonstrator where data is available in the data lake and can be accessed for quasi-interactive analysis through SWAN, scaling out the processing in a cloud or HPC backend.

6. Sustainability

The infrastructure and software stack needs to remain operative, efficient and competitive over the project lifetime, which spans decades. The data and the capability to process the data needs to be retained for longer that the project lifetime.

6.1. Common Solutions for infrastructure and software stack

Experiment specific solutions both in the areas of software and computing systems and services will likely be hard to support and maintain in the medium and long term. Many funding agencies express clearly the view that only common software and services development should be funded even in the short term. In the area of offline software, the usage of common tools and libraries should be favored. Geant and ROOT are good examples of generally adopted packages in HEP and should therefore serve as base for commonality. In terms of computing frameworks, implementing an increasing number of functionalities closer to the infrastructure level and in general through common middleware services will will favour their adoption by multiple experiments. The File Transfer Service and the Condor scheduler are good

examples in the data and workload management areas of middleware solutions today widely adopted and that could serve the basis of a data and compute provisioning layer. In addition, experiments should seek commonalities whenever possible also at the level of high level services such as data management, workflow management, information systems and monitoring and analytics.

6.2. Security Infrastructure

The shift towards federated identities and the adoption of new authorization standards by the industry is a strong signal for WLCG to adapt its authorization infrastructure. It is necessary to continue to connect with users globally as well as peer organisation, infrastructures and cloud services. The current needs of the WLCG sites and experiments are being established, and a review of the main available authorization architectures is being conducted, in order to prepare a transition of WLCG towards a sustainable and highly interoperable authorization infrastructure. Chapter 3.13 of the Community White Paper reviews the current infrastructure, identifies future challenges, and defines an R&D roadmap. Although it is clear that WLCG has to evolve away from X.509 at least for end users, there has been no community wide strategy. Several independent efforts to provide an authorization infrastructure supporting federated identity and authorization without certificates have been started and it is essential that a common vision be agreed upon. Different solutions are being implemented in the Research & Education sector and a number of translation services will be required to allow interoperable services. The plan for WLCG can be summarized in the following actions:

6.2.1. Collect and agree on a well-defined set of requirements from LHC experiments and WLCG sites regarding VO Membership Management and WLCG Service Authorization. These requirements must support and be consistent with with existing security policies, operational security requirements, IGTF Levels of Assurance and the EU General Data Protection Regulation.

6.2.2. Review the current AAI (Authentication and Authorisation Infrastructure) and the tools being considered for the future by WLCG partners. Evaluate existing or proposed AAI in the HEP community (e.g, in EGI, INDIGO-Datacloud, OSG) for their suitability for WLCG. Review VO management tools (group management) and evaluate how the current VO registration and user management workflow can be expanded to accommodate federated identities. Analyse the aspects related to user authentication, service authentication and authorization, membership management tools token translation services and suspensions mechanisms.

6.2.3. Propose a design for WLCG that ensures both a suitable production service and maximum interoperability in the long term. The costs of preparing and maintaining the authorization infrastructure, the security model, the compliance with existing data protection and conformance to the WLCG requirements defined previously are all key aspects. The

scope of the proposal should include the additional services required, such as token translation services or blocking services.

6.2.4. Contribute to the definition a JSON Web Token schema, the building block for token-based authorization solutions such as OpenID Connect and OAuth2, for a common or compatible authorization token profile to be used by collaborating infrastructures.

6.2.5. Produce a proof-of-concept (or multiple) to demonstrate a “certificate free” workflow of a VO user, including VO registration and job submission.

6.2.6. Prepare a pilot service demonstrating interoperability and scalability of a future production service. The pilot service will include both the user registration/management workflow in the VO and the full authorization chain required to access Web/non-Web WLCG services using federated identities.

This work will be conducted under the WLCG Authorization Working Group and in collaboration with the AARC EU project.

7. Workplan

In Fig.1 and Fig.2 we draft a tentative time schedule for the tasks which we expect will have a higher impact in addressing the HL-LHC computing challenge. We identified four main tasks and several subtasks, with several dependencies among them. We also set yearly milestones with deliverables, while we expect to monitor progress more regularly than that. We foresee major progress in the various R&D activities in the next two years and therefore year 2020 should represent the right time for a major checkpoint in preparation for a HL-LHC TDR.

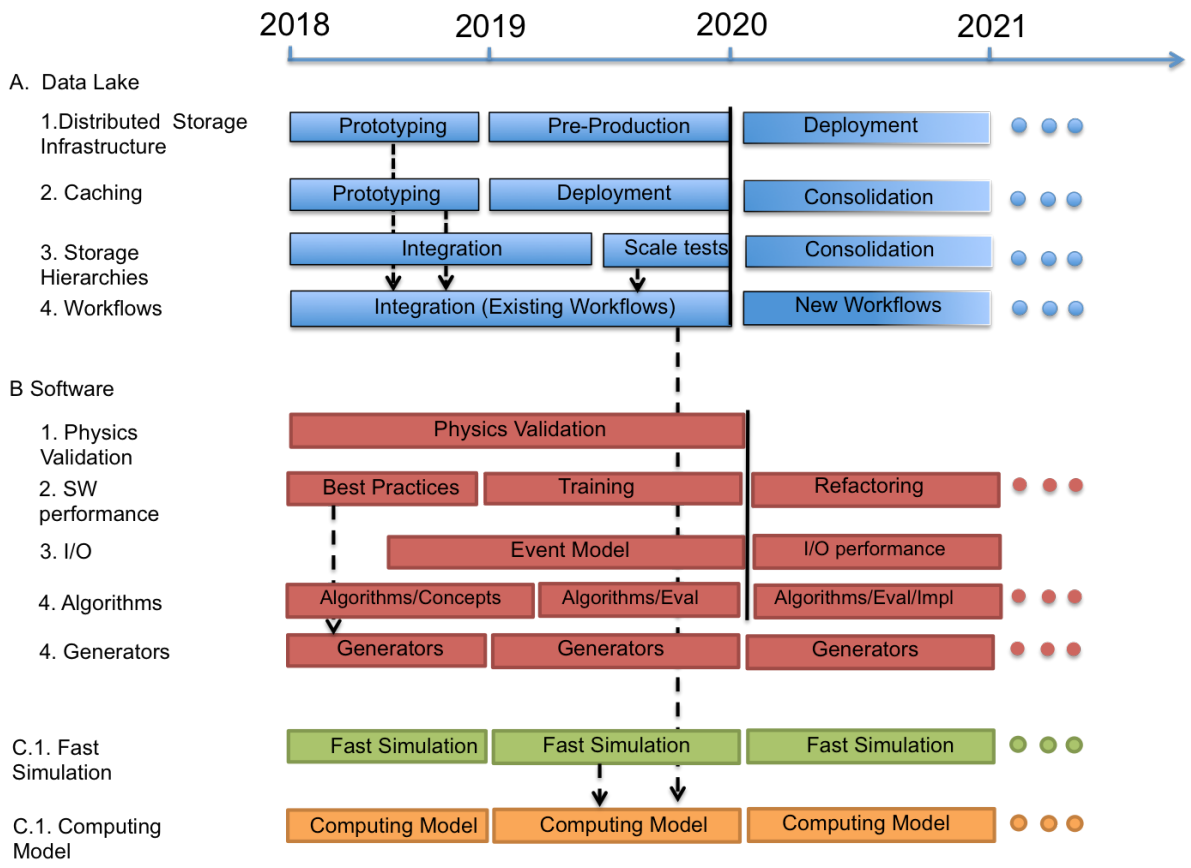


Fig. 1: the expected time schedule for the tasks with higher impact in addressing the HI-LHC computing challenge

Item	M1 (2019)	M2 (2020)	Long Term
A.1. Distributed Storage Infrastructure	Deploy and operate a prototype across 5 T1s	Offer the prototype as pre-production service to the experiments for beta testing	Upgrade the prototype to a full scale production service
A.2. Caching	Prototype different caching solutions with different protocols	Deploy the valuable solutions to complement the distributed storage pre-production service	Deploy and operate a fully scaled content delivery network
A.3. Storage Hierarchies	Evolve the computing systems of the experiments to fully leverage a high latency and low cost multi-tiered storage architecture	Stress tests the experiment services and the facilities to measure the effectiveness of the multi-tiered model	Run an increasing number of workflows accessing data from high latency storage
A.4. Workflows	Define the relevant metrics based on the cost model. Setup a system allowing to measure the impact of different technologies/decisions on workflows	Evolve the existing workflows to leverage the benefits introduced by the new technologies introduced in M1 and M2	Introduce new workflows, tailored to the new infrastructure and services introduced in M1, M2 and beyond
B.1. Physics Validation	Setup a system automating the Physics Validation process. The validation should require close to no human effort at least when refactoring the code (n algorithmic change)		
B.2. Software Performance	Document coding best practices according to the criteria in Section 4	Organize training based on those best practices and start applying them to the newly written code	Refactor the existing code, based on the best practices.
B.3. I/O	Review the Event Data Model to benefit of new technologies and adapt to new data access pattern (latency hiding). Define the optimal granularity for data management and data access.		Evolve the I/O layer based on the criteria in 4.3
B.4. Algorithms	Identify potential algorithms and compile for the different tasks their characteristics	Evaluate the impact of the most promising algorithms with realistic tests use cases	Implement new algorithms(focus on reconstruction) for HL-LHC leveraging B.2.
B.5 Generators	Regular improvements along the lines of Section 3.1 with yearly checkpoints		
C.1. Fast Simulation	Regular Improvements in Fast Simulation with yearly checkpoints. By 2020, it should be clear what will be the impact of Fast Simulation in managing the HL-LHC cost and the implications to the computing models.		
D.1. Computing Model	Regular Evolution of the computing models, incorporating the findings from all the work plan and setting directions accordingly. Yearly checkpoints		

Fig. 2: for each task and subtask in Fig. 1, a brief description of the deliverable and milestones.

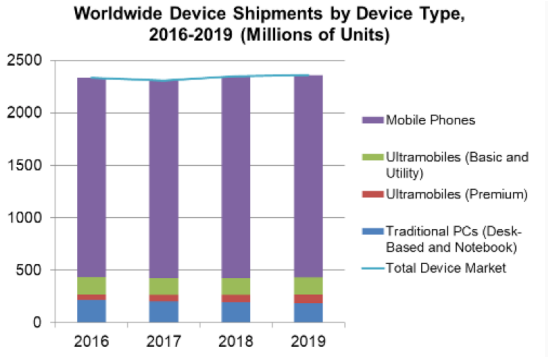
8. Appendix A: Technology and Market Trend

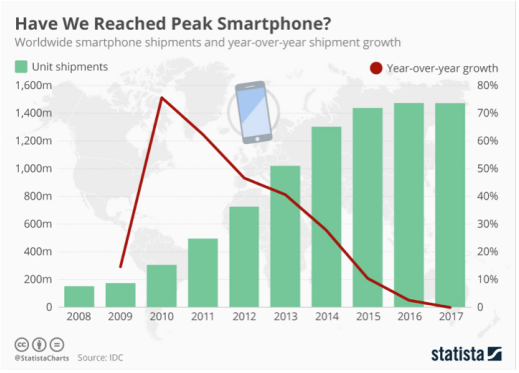
8.1. Industry developments

The current state of the computing industry is characterized by aggregation and consolidation of companies in the various sectors. Only very few companies are dominating the markets for all aspects of computing equipment.

Server CPU:	Intel (99%)
PC CPU:	Intel (79%), AMD (21%)
Device CPU:	ARM (100%)
FPGA:	Xilinx (49%), Intel (38%)
GPU:	Intel (68%), Nvidia (18%), AMD (14%)
Hard Disks:	Western Digital (41%), Seagate (37%), Toshiba (22%)
DRAM:	Samsung (50%), Hynix (25%), Micron (19%)
NAND:	Samsung (35%), Toshiba((20%), Western Digital (17%), Micron (10%)
Solid State Disks:	Samsung (35%), Western Digital (19%), Intel (9%), Kingston (7%)
Tape Drives:	IBM (100%)
Tape Media:	Fujifilm, Sony

All device markets show very low increases or negative growth rates with clear signs of saturation.





http://www.digitaltvnews.net/?p=30009
https://www.statista.com/chart/12798/global-smartphone-shipments/

8.2. Processors

The processor market is dominated by Intel in the PC and server area and ARM in the general device market (phones, tablets). So far, all attempts to break the Intel monopoly in the server market with PowerPC, ARM or AMD have failed. More competition was expected towards the end of 2017 with the introduction of new models from AMD (EPYC) and the 3rd generation of specific ARM processors (Qualcomm, Cavium, etc.). Both new architecture show promising values of price/performance and power/performance, but are still not competitive in the medium and high end server area.

The current fabrication of microprocessors is using the so called “14nm node” which corresponds to structure sizes of 50-70nm on the chips. The next generation of 10 nm nodes is already tested and will go into full production this year. IBM has shown prototypes of 5nm node manufacturing. The speed of the manufacturing improvements has slowed down considerably during the last years. The market leader Intel was using his Tick-Tock model in a 2-year interval, one year for a decrease of the structure sizes on the chip and one year for improvement of the

microprocessor architecture. This has moved to one-year structure size improvement followed by three years architecture changes.

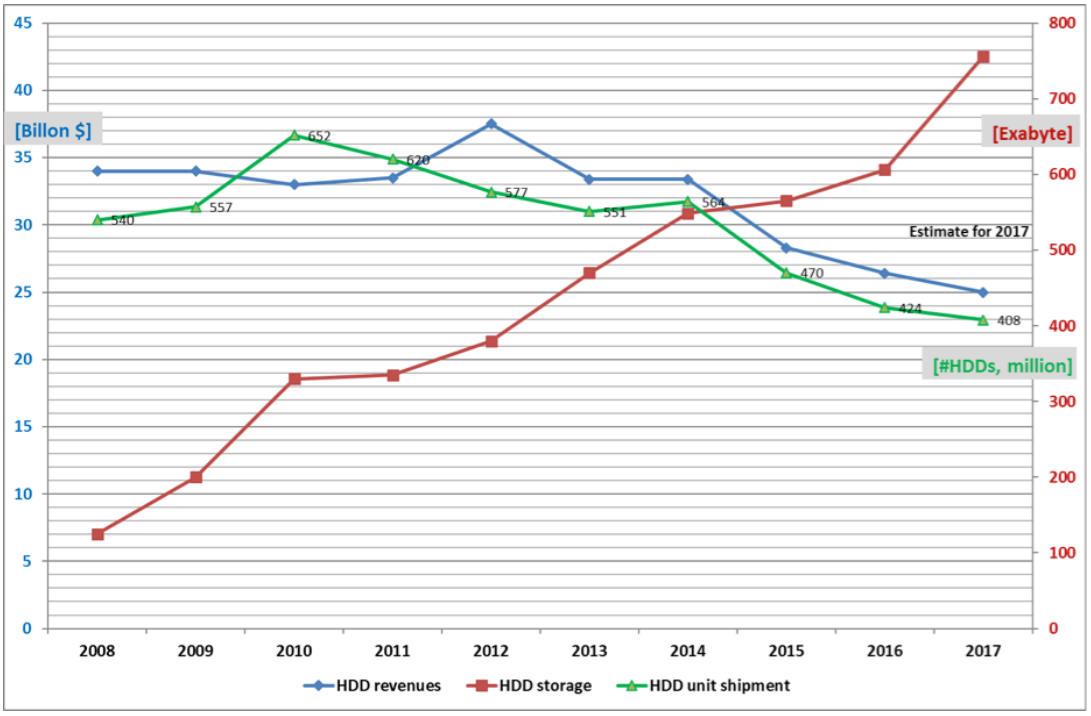
There are currently only four companies (Intel, Samsung, Globalfoundries, TSMC) at this level of structure sizes (<=10nm) which can afford the very large (~10B$) capital investments into new fabrication facilities.

The server market (10 million units and 55 B$ revenue per year) shows clear sign of saturation with very small or no growth rates. The anticipated price-performance improvements for CPU servers in 2017 was about 15%, which was at CERN confirmed with purchases in Q1 2017. The next generation of processors (Skylake) should have improved the situation further at the end of 2017. But the large increase in DRAM prices (and SSD prices) actually led to an increase of CPU server price/performance by more than 30% in Q4 2017. The memory price situation will improve only towards the end of 2018, but the DRAM industry is also facing an upcoming technology problem: end of structure size scaling below 10nm.

The design of new processor architectures is currently focused on neural networks, graphics and audio processing. The goal is to bring more data processing capabilities into smartphones and IoT, especially increasing the Machine Learning functions. This will possibly reduce the need for networking and cloud storage and processing (keyword: Locality). The assumption of a future price vs performance improvement of 15% per year might be optimistic.

8.3. Disk Storage

Falling revenues and unit sales currently characterize the traditional HDD storage market due to the Desktop PC market declining now since 5 years, increasing usage of SSDs in Notebooks and a strong competition of SSDs in the high-end enterprise disk area. The following plot is based on numbers from TRENDFOCUS (http://www.trendfocus.com/) reports.



The technology change from PMR (Perpendicular Magnetic Recording) to HAMR (Heat Assisted Magnetic Recording) or MAMR (Microwave Assisted Magnetic Recording) with much higher bit densities has been delayed and now first HDDs are only anticipated for the end of 2018. In the meantime, two intermediate technologies are used to increase bit densities at the 20% level (SMR and TMR). For the next years, one can still expect a cost improvement of about 25% for Hard Disk storage.

The cost evolution of SSDs is also improving rapidly. However, for capacity disks the standard HDD will still have a cost advantage of a factor 10 for at least the next 5 years. The underlying NAND technology moved from 2D-structures to 3D two years ago. The decreasing structure sizes were causing unsurmountable data reliability problems (wear-level). The key characteristics of the new 3D structures are the following: 2D structure size increase from ~10nm to ~90 nm (improved reliability) and move to linear density improvement (the 3D layer improvement moves in steps of 16 per generation) instead of exponential (factor 2). Thus the price/density improvements will slow down.

8.4. Tape Storage

In 2017 the cost for LTO tape media has reached 0.01 Euro/GB and shows an improvement rate of about 20% per year. The latest report from the LTO consortium shows a continuously decreasing amount of tape media sold while the amount of cumulative space is still increasing (10 EB per quarter compared to 150 EB for HDD). After Oracle stopped the development of Enterprise Tapes in the beginning of 2017, there is only IBM left manufacturing tape drive heads for LTO and IBM Enterprise tape drives. The yearly production of heads is currently about 230000 compared to 450000 in 2014. The only two companies (Sony and Fujifilm) manufacturing LTO

tape media are entangled in a patent ‘war’ since 2016. The future development in the tape area needs to be watched closely. A move from tape to hard disk storage would increase the costs by at least a factor 3.

A more detailed report about the technology and market evolution of storage can be found here: https://twiki.cern.ch/twiki/bin/view/Main/TechMarketDocuments