Thoughts about HL-LHC Computing Models

Maria Girone (CERN), Ian Fisk (Simons Foundation), Oliver Gutsche (Fermilab)

For all of its success the computing at the LHC has relied almost exclusively on dedicated resources purchased for the LHC program. This has allowed the experiments to have sufficient computing to make groundbreaking discoveries, but fails to exploit other resource opportunities and prevents dynamic expansion. It is important to be able to leverage as many types of resources as possible to ensure the most cost effective operations. It is important to be able to expand dynamically because it allows activity to be scheduled for peak and not average. Up to now, since most computing resources were dedicated, all activity needed to be scheduled with a flat profile because unused resource were wasted. Workflows can take months to complete. Being able to effectively burst to factors more than the average resource level would fundamentally change how people work and schedule, much less time would be spent waiting for organized processing to complete.

In this white paper we focus on the changes needed in resource definition, workflow management, and maintenance and operations to allow greater use of dynamically provisioned resources.

On HL-LHC time scales, compute resources will be available in various forms and architectures scattered around the globe and exhibiting a large variety of availability and reliability characteristics. The challenge will be to feed data to processes that run on these resources.

In the context of this paper, the basic parts of a HL-LHC computing model are:

1. Compute
2. Storage
3. Network
4. Analysis

We distinguish central processing workflows which are organized by a few entities from general analysis which is performed by many individual researchers or groups of researchers.

In finding solutions, we propose to follow the following guidelines:

1. Solutions need to serve multiple communities/experiments/collaborations also outside the field of HEP to avoid developing/maintaining multiple systems with similar or same functionality

2. Solutions should have a large cross sections with industry solutions to lower the maintenance cost for science and facilitate migration of experts from industry to science and vice versa.

The guiding principle is to move functionality of higher level systems into lower levels of the infrastructure stack.

Compute for central processing workflows is assumed to be mostly provided by volatile resources that can scale elastically with demand. Multiple architectures with various degrees of core counts and special hardware like GPUs will be available and workflows need to be able to use them. A common workflow management layer should combine all lessons learned from the current LHC systems (DIRAC, WMAgent, Panda, …) and other systems, including all tracking of completeness of work, automatic failure recovery and monitoring. We claim that the problem of defining work and workflows is sufficiently common that merging this functionality makes sense. A common resource provisioning layer would be responsible for provisioning resources on the various forms of available compute resources. A very tight coupling between workflow management and resource provisioning will be needed to be able to dynamically partition work depending on the provisioned resources, as well as provision resources depending on the available work. We don't see this existing in open source or community based projects. We propose to integrate this functionality into existing batch systems like HTCondor or create a new stakeholder-driven community project.

Storage will be provided in two forms. On the one hand we have *managed storage* that we know from traditional distributed computing models. It is provided to VOs and/or experiments by sites and funding agencies and is managed, in a manual or automatic way, by policies set by the VO or experiment. The other form describes new forms of *caching storage*, which can be used opportunistically and dynamically, but are not managed by policies but rather by access patterns. The majority of managed storage will be provided at larger facilities. Organized data movement handles the distribution of data between these facilities. Workflows running on any compute resources stream the data from managed storage facilities through data federations dynamically using caching storage instances on lower levels of the federation. Output needs to be handled by the federations as well and archived back at the managed storage facilities.

Network will not only be the vehicle for orchestrated data movement between managed storage facilities and for streaming data to compute through the federation dynamically using the available caching storage instances. The network will play a much more active role in providing access to data, managing the caching storage instances and providing metadata tracking and orchestrated transfer functionalities. It will marry the current transfer system functionality, data federation functionality and data book keeping catalogs with the underlying network fabric and provide a global optimization of data flows.

Analysis of data has two components: analysis-specific processing and data selection for interactive analysis. The first component transforms data using code from the central software frameworks of the experiments. Compared to central workflows, analysis workflows need to be able to execute new algorithms or different incarnations of central algorithms written by individual researchers or small groups of researchers. Analysis workflows are predominantly executed on dedicated compute resources (compared to the general volatile compute resources used for central workflows) with direct access to managed storage facilities to benefit from higher support levels and local optimizations. The same compute resources are also used for the second component of analysis taking care of selecting subsets of the data and calculating new properties from the data (skimming & slimming) with very high turn around. Output is downloaded for interactive analysis by the researcher or groups of researchers.

R&D proof-of-concept ideas: We would like to propose the following R&D proof of concepts:

1. Computing: a more common and consistent integration of dynamically provisioned workflow management techniques into the HTCondor workflow system. A focus on flexibility in resource implementation and scalability is needed.
2. Storage: improvements in data caching are needed in opportunistic storage, including access pattern-driven dynamic caching in federations.
3. Network: investigate to move the functionality of transfer systems, data federations and metadata catalogs into the network stack and provide optimized interfaces for the common resource provisioning and workflow management layer to optimize data placement and streaming data access from applications.