Distributed Data Management and Distributed File Systems

Maria Girone
CERN, CH-1211, Geneva 23, Switzerland

Abstract. The LHC program has been successful in part due to the globally distributed computing resources used for collecting, serving, processing, and analyzing the large LHC datasets. The introduction of distributed computing early in the LHC program spawned the development of new technologies and techniques to synchronize information and data between physically separated computing centers. Two of the most challenges services are the distributed file systems and the distributed data management systems. In this paper I will discuss how we have evolved from local site services to more globally independent services in the areas of distributed file systems and data management and how these capabilities may continue to evolve into the future. I will address the design choices, the motivations, and the future evolution of the computing systems used for High Energy Physics.

1. Introduction

For several decades the computing resources for high energy physics experiments were localized in one center, which was generally close to the detector, and contained all the services that supported the computing infrastructure. There was no concept of a distributed file system because activities were focused on the central facility. Data management meant ensuring that files migrated to and from tape. In 2000 the MONARC model [1] introduced a hierarchical computing environment and the computing capabilities at the LHC were proposed to be provided by hundreds of computing centers on 5 continents. The Worldwide LHC Computing Grid (WLCG) was one of the first successful computational grids. The motivation for this came from the desire to pay for computing in the local countries, to better capitalize on local infrastructure, and to take advantage of local expertise. With this focus on distribution, services needed to be replicated and synchronized. Two of the most critical services for creating a functional grid were the distributed file systems and the distributed data management systems. These services facilitate all of the higher level functionality: enabling consistent and reproducible processing results across globally separated facilities and moving the petabytes of input data and resulting derived data. A consistent processing environment provided by a distributed file system allows experiments to reproduce results by ensuring there are no differences in the environment and therefore no differences in the results derived at independent sites. A global data management service ensures samples are archived, replicated, and served to the community, which allows members to work effectively from any center.

The services for environment and data management described above are examples of how making dedicated and independent global services can improve flexibility and open resources as will be discussed later. Looking forward it is interesting to consider how further development of
services for advanced data management and large scale processing could achieve a more flexible and dynamic computing system, if we allow the services to scale and evolve independently.

2. Services

At the time of MONARC the sites were defined by a set of independent local site services that were essentially copies of the original functions at the single central facility. The services primarily interacted with each other and there was a defined set of protocols for external communication: the site processed data read by the local site storage; the site database served local applications; and the site software environment served jobs running locally. At the boundary the site had a computing element to send in jobs, a storage element to send in files, and an information system interface to describe the site. Sites on the same tier were similar in terms of capability. In the original design the sites were required to maintain full independence because the network was perceived as insufficiently provisioned and unreliable. The bulk of this paper will motivate the evolution from the original model of a variety of sites that were largely independent relying on local services, to a system where many of the services have grown to be globally distributed, and how that evolution continues. The network is one of the most reliable services and as the capacity has grown it has helped fuel this evolution.

2.1. The Software Environment

The software environment is one of the first examples of a globalized service. At the start of LHC Run 1, the software environment was deployed in a variety of ways: OpenAFS was used, though even at the time it was difficult to manage across wide area distribution; BitTorrent was used successfully by ALICE, but at some facilities there were issues about security and opening ports; NFS was used locally on sites; grid jobs were sometimes deployed to come in, access the processing resources, and then populate of the NFS system; and site admins installed software locally. Many of these solutions were seen as non-scalable, operationally intensive and/or sources of high latency. The LHC experiments sought a better and more consistent solution.

The CERN Virtual Machine File System (CVMFS) [4] was proposed during Run 1. CVMFS is a service that was adopted by the grid, but did not originate as a grid project. CVMFS was designed to enable Virtual Machines to have environments without big machine images, but the design choices made it directly applicable also to a grid environment. Large-scale software distribution is an ideal application for a caching system like CVMFS because between two software releases in high energy physics only about 10% of the files change, which allows caching the other 90% from previous releases. Operations like stat and ls are localized to the client and improve the scalability. In 2012, the WLCG Operations and Tools Technical Evolution Group made a recommendation [7] that all experiments should use CVMFS for their software deployment. The effort was expected to be moderate because it involved service deployments at all sites and small changes to the experiments processing frameworks, but the impact was predicted to be significant.

CVMFS was an enabling technology. The experiments could expect to find the same software environment everywhere within hours and with much less effort. CVMFS is populated in one central location and the contents are propagated to all sites automatically within hours. CVMFS is also completely independent from the other grid services. Since it was designed not to run initially on the grid, it could be equally well deployed on local sites with no grid interfaces, desktops, virtual machines and it made a consistent software environment for users no matter where they were.

Figure 1 outlines the architecture of CVMFS, which relies on a stratum zero service which is read-write and all subsequent tiers are

![Figure 1. Architectural Diagram of CVMFS](image)
read-only. Immutable data in the distributed tiers dramatically increases the scalability of the whole system and allows caching. There are infrequent atomic updates to the central location and then read access to all of the clients. All the traffic through the system is handled by HTTP and relies on the scalable technology of squid caches. There are minimal protocol requirements and there is an aggressive hierarchical caching. The local client projects a mounted file system, which is also read-only and POSIX compliant. By introducing more strata and more caches the system can scale to be arbitrarily large.

CVMFS content has grown linearly over the last 5 years and reached 12 TB of space stored in CVMFS in 2012. In addition, the number of files has similarly grown and is now at 200M. CVMFS has spread to all inhabited continents. There are at least 64k nodes in use at 160 sites and CVFMFS is a WLCG critical service.

2.2. Data Management

Both software distribution and data distribution are challenges that move files around, but they differ in fundamental ways. The most obvious difference is that software environment must manage of the order of 10TB of information and data management must handle of the order of 100 PB. The level of replication for the software environment is equal to the number of sites, while in data management the average sample replication is a factor 2 to 3. The latency requirements indicate that full synchronization is expected within a few hours of the environment, while replicating a large sample of data is permitted to take days. The attributes of software release and data management are shown in Table 1. Both these problems have required years of development effort to achieve the functionality that LHC currently has deployed, but the solutions are not interchangeable due to fundamentally different scale and latency requirements.

| Software Distribution vs. Data Management |   |
|------------------------------------------|---|
| Size of samples                          | O(10TB) | O(100PB) |
| Level of Replication                     | All sites | Average sample replication factor 2-3 |
| Latency                                  | hours | days |
| Update rate                              | Packages are updated frequently (incl. nightly builds) | New datasets are created less frequently |

Table 1. Attributes of Software Distribution vs. Data Management

In order to move the petabytes of data between sites of the LHC computing system each experiment has developed a data management system. There is considerable commonality in the underlying support services, and each has developed similar basic services for bookkeeping and metadata, file catalogs, replica catalogs, accounting, requests and subscriptions, and file transfer. The LHCB solution for data management was embedded in the larger computing infrastructure Dirac [3]. ATLAS redesigned their data management system for Run2 and the new system is called Rucio [11]. The ALICE system for data management is embedded into the larger ALIEN framework [2]. In CMS the data management system, PhEDEx, is one of the longest continuously developed services in the experiment [12]. The underlying services used to support activities are not identical between the four experiments, but there is substantial overlap. Three of the four experiments use the WLCG File Transfer Service (FTS) to move files. Three of four experiments rely on Oracle for the underlying database. All four experiments need to open files both locally and through the grid interface from the same mix of site storage elements.

Currently, there are close to 200 sites in WLCG with 246PB of disk and 267PB of tape. There is 140PB of unique LHC data and 280PB under management, spread over 1B files. The
breakdown of data by experiment is shown in Figure 2. Even during the LHC Long Shutdown (LS1) period the experiments moved on average 10GB/s with the bulk of the data coming from the 2 multi-purpose experiments. More than half a petabyte was moved per day and the total volume during LS1 was nearly an exabyte. The average transfer throughput during 2014 is shown in Figure 3 from the WLCG transfer dashboard [6].

![Figure 2. LHC Unique Data and Replication by Experiment](image)

2.2.1. Dynamic Data Placement  The primary method for data management has been to subscribe samples to local sites. The processing and storage are coupled and only data available locally is visible. Flat static subscriptions are based on the assumption that samples have similar amounts of access. Studies in the early days of Run 1 by the experiments showed this assumption to be false and that the amount of accesses varies by orders of magnitude between datasets. Instead of trying to pattern and predict what samples would be most heavily accessed, ATLAS pursued a dynamic data placement solution based on the concept of replicating the data when large accesses were measured [10]. Processing jobs were submitted to a Tier-1 location with most of the samples available. If an increased waiting time was observed, a second copy of the data could be subscribed to a Tier-2 and the jobs re-brokered to run on the new site. Additional samples are removed when the amount of access wanes.

2.2.2. Data Federation  More responsive and dynamic than automated data placement is the technique of data federation. Data Federation seeks to integrate data management and data access, resulting in a global data management system that can handle replication and transfer of data to storage or to running applications. ALICE has used data federation tools for their data management from the beginning. The other experiments have been following and deploying this in preparation for Run 2. All experiments currently rely on Xrootd [5] for the wide area delivery of the data files, though several use the local experiment catalog for discovery. Several collaborations are investigating alternative web service based delivery systems, but they are
only in the prototype phase. The basic functional diagram of Xrootd is shown in Figure 4. A user from an application asks the xrootd redirector to provide a file; the file is discovered and delivered from a particular server and sent directly to the application.

In order for Xrootd to provide data from remote independent data servers it has to maintain a distributed file system, so that sources of data can be found in a dynamically changing environment when servers come and go and replicas are made and deleted. All of the servers have the same directory structure, but they do not have to have the same contents. When the redirector asks the servers if they have a particular file, only a check in one directory is needed. This simple query is fast and allows a high transaction rate between the redirector and the servers because no costly searches are required. CMS demonstrated 200Hz of file opens from a single Tier-2 to remote Xrootd servers. This is more than a factor of 10 higher than is predicted to be required for average stable running.

Most of the data, even in the existence of data federation, is served from local storage to local processors. The most aggressive target for wide area access to data is from the CMS experiment, and even it is only 20% of the total access. CMS has measured the connectivity of clients and servers in the worldwide data federation. The bulk of the connections are between Europe and North America, but connections between Asia and South America are used. The total aggregate transfer rate exceeds 2GB/s, which results in the largest sites serving petabytes of data over the course of a year.

One negative aspect of data federation is that it adds complexity. In addition to the local site with local storage and local processing, there are remote sites that have to perform correctly and successfully in order that processing is able to succeed. CMS has noticed in their deployment that not all the sites are equal. CMS did not choose to exclude sites, simply because they were not as reliable as others, but did not choose to ignore the differences either. The solution proposed was to have a production federation, carrying the bulk of the traffic, and a commissioning federation. Only if a file was unavailable from the production federation would the less reliable sites be used through a commissioning redirector.

3. Looking Forward

Adopting independent global solutions for software distribution made new resources available and simplified both support and operation. The data management solution has evolved from static subscriptions of data to dynamic data placement based on load and more recently all the experiments have investigated data federation solutions. Data federations are the first step towards a central data management service, however most of the access is still processing accessing storage at the local site. Even with introduction of basic data federation our two most expensive services, processing and storage, are tightly coupled.

Table 2 outlines the differences between processing services and storage services. The time scales for use and ability to dynamically switch between users are fundamentally different between processing and storage. It is interesting to consider what potential benefits there could be in introducing an advanced data management service. A true global service for data distribution would be able to serve data to multiple types of processing resources independently of location.

In this model of treating data management as a service, one can imagine evolving to a state
where the HEP specific computing systems and services would be dedicated to data management and data delivery, while the processing would be generic and logically separated. The processing could be dynamic and might be provided by traditional sources or by a larger pool of providers, a cloud resource, or a supercomputer allocation.

In order to have a data management service delivering to processing resources that are more heterogeneous, high energy physics computing will need to make the next generation of data delivery systems, which will rely more on moving data. This introduces an even tighter coupling with the network itself. The first step is to have a better understanding of the performance and attributes of the network available to the workflow and data handling systems. ALICE has tightly integrated the MonALISA monitoring framework [8] into their distributed computing environment and can measure real-time values of the performance of network connections. This allows the determination the location of the closest alternative copy of a data file dynamically.

Looking forward there are a variety of next generation network projects that are proposed in the other LHC experiments. Some simply provide networking information to the data management system to make more intelligent choices. ATLAS is providing PerfSonar data to the workflow management system to improve the choices made for locating replicas and executing jobs [9]. Network information can be particularly helpful in an environment with data federation. The processing no longer needs to be coupled to local storage in the presence of data federation, but wide area access to data files couples the processing to the network. CMS is looking to influence the structure of the network by making virtual circuits when needed, to allow reservations and more predictable traffic and higher performance. By using a network reservation system, CMS could get much more predictable bandwidth than using the same capacity network in the presence of contention.

The most ambitious new network projects are based on the concept that the network routing should contain information about the content and not simply the packets. The new systems are referred to as Named Data Networks, NDN, and they will work to deliver particular data samples to a client. Data is stored on the network and might be served from multiple locations. A number of the concepts discussed earlier like dynamic data placement and data federation are proposed to be handed in the network layer in Named Data Networks. NDN systems remain in the exploratory and development phases and represent the tightest coupling between the networking infrastructure and the data management system.

Wide area access to data through the federation is flexible, but for repeated access to samples it is not as efficient as local access. The introduction of local caching helps maintain high efficiency while maintaining the flexibility of data federation. Caching can be thought of as opportunistic storage: replicating storage that it repeatedly accessed and removing copies that are no longer needed. In a dynamic and complex system the caching needs to be smart enough to know the origin and access patterns, such that it does not cache files that are hosted close by and does not cache information from the files that are never accessed.

If we successfully separate the data management system and the processing systems, then

| Time of Use | Processing | Storage |
|-------------|------------|---------|
|             | Minutes, hours, days | Days, weeks, years |
| Sharable Resource | Opportunistic use through prioritized access | No concept of storage priority and freeing space is normally time consuming |
| Loss from Preemption | Lose the CPU time spent | Deleting files can lose all the work needed to create or collect them |

Table 2. Attributes of Processing compared to Storage
we can consider using these services on different time scales. Currently in resource planning it is important to show that all the computing resources will be used continuously and over long periods of time. A number of activities like data processing, (re)processing and simulation come as requests for months of work. If it were possible to burst to a much higher number of resources, these predictable activities could be completed in shorter times and potentially using less operational effort on average. A diagram of peak scheduling is shown in Figure 5. This might be fundamentally changing the efficiencies of the experiment collaborations and the way people work, separating the processing and storage services, and basically allowing them to grow and scale independently, and allowing provisioning for peak.

![Figure 5. Diagram of Average and Peak Scheduling](image)

Large scale data serving opens large scale computing resources for peak scheduling. Data management as a service could deliver to commercial providers, super computers, or multi-VO infrastructures like the grid projects of big individual sites. By having the capacity to deliver data over the wide area in an efficient way, or to local temporary caches, there is the ability to open these types of computing resources.

4. Outlook
Computing for the LHC is evolving to define sets of global services that cross site boundaries. Each of the individual changes along this evolution has shown benefits; benefits for software distribution, with reduced operations and new resource opportunities, expanding in the number of resources we can run on, and changing people expectations about the level of services they could expect. The same is true for some of the complicated services of data management. The evolution of data management services have the potential to improve efficiency of how we use storage, open new resources for our use (supercomputers, cloud providers, and non-traditional sites), and change how resources are scheduled and tasks are completed.

5. References
[1] Michael Aderholz et al. Models of Networked Analysis at Regional Centres for LHC Experiments (MONARC), Phase 2 Report, 24th March 2000. Technical Report CERN-LCB-2000-001. KEK-2000-8, CERN, Geneva, Apr 2000.
[2] S Bagnasco, L Betev, P Buncic, F Carminati, C Cirstoiu, C Grigoras, A Hayrapetyan, A Harutyunyan, Peters A J, and P Saiz. AliEn: ALICE environment on the GRID. J. Phys.: Conf. Ser., 119:062012, 2008.
[3] JP Baur, Ph Charpentier, K Ciba, R Graciani, E Lanciotti, Z Mathe, D Remenska, and R Santana. The LHCb Data Management System. J. Phys.: Conf. Ser., 396:032023, 10 p, 2012.
[4] P Buncic, C Aguado Sanchez, J Blomer, L Franco, A Harutyunian, P Mato, and Y Yao. CernVM - a virtual software appliance for LHC applications. J. Phys.: Conf. Ser., 219:042003, 2010.
[5] Furano F Dorigo A, Elmer P and Hanushevsky A. Xrootd - a highly scalable architecture for data access. WSEAS Transactions on Computers, 2005.

[6] Maria Varvara Georgiou. Real-time statistic analytics for the WLCG Transfers Dashboard with Espe. Technical Report CERN-STUDENTS-Note-2014-180, CERN, Geneva, Sep 2014.

[7] Maria Girone and Jeff Templon. Final Report of the Operations & Tools Technical Evolution Group. Technical report, CERN, Geneva, Feb 2012.

[8] C Grigoras, R Voicu, N Tapus, I Legrand, F Carminati, and L Betev. MonALISA-based Grid monitoring and control. Eur. Phys. J. Plus, 126(1):9, 2011.

[9] Alexei Klimentov, Kaushik De, Artem Petrosyan, Jorge Horacio Batista, and Shawn Patrick Mc Kee. Integrating Network Awareness in ATLAS Distributed Computing Using the ANSE Project. Mar 2015.

[10] T Maeno, K De, and S Panitkin. PD2P : PanDA Dynamic Data Placement for ATLAS. May 2012.

[11] C Serfon, M Barisits, T Beermann, V Garonne, L Goossens, M Lassnig, A Nairz, and R Vigne. Rucio, the next-generation Data Management system in ATLAS. Technical Report ATL-SOFT-PROC-2014-009, CERN, Geneva, Oct 2014.

[12] Anthony Wildish. The CMS Data Management System. Technical Report CMS-CR-2013-388, CERN, Geneva, Oct 2013.