Journal of Physics: Conference Series **513** (2014) 062050 doi:10.1088/1742-6596/513/6/062050

Challenging data and workload management in CMS Computing with network-aware systems

Bonacorsi D[1] **and Wildish T**[2]

[1] University of Bologna and INFN, Italy
[15] Princeton University, Princeton, NJ, USA

E-mail: awildish@princeton.edu

Abstract. After a successful first run at the LHC, and during the Long Shutdown (LS1) of the accelerator, the workload and data management sectors of the CMS Computing Model are entering into an operational review phase in order to concretely assess area of possible improvements and paths to exploit new promising technology trends. In particular, since the preparation activities for the LHC start, the Networks have constantly been of paramount importance for the execution of CMS workflows, exceeding the original expectations - as from the MONARC model - in terms of performance, stability and reliability. The low-latency transfers of PetaBytes of CMS data among dozens of WLCG Tiers worldwide using the PhEDEx dataset replication system is an example of the importance of reliable Networks. Another example is the exploitation of WAN data access over data federations in CMS. A new emerging area of work is the exploitation of Intelligent Network Services, including also bandwidth on demand concepts. In this paper, we will review the work done in CMS on this, and the next steps.

1. Introduction

The CMS computing model was designed almost 10 years ago [1]. It describes a hierarchical organisation of computing sites and the high-level dataflows between them. It defined the way we would analyse data, following the paradigm of *move the jobs to the data*. At that time we thought that the network would be one of our weakest services, that links would be slow and error-prone, that transferring data with high throughput and efficiency would he hard.

Today, links of 10 Gbps are common, which is considerably more than anticipated at the time of the computing model, and the network is far more robust than had been expected. We have evolved from the strictly hierarchical model of the past to an almost fully-connected mesh, and are able to transfer data effeciently between most pairs of sites. Data-transfers among the ensemble of CMS sites routinely amount to over 1 PB of data per week, yet we are far from saturating the bandwidth on most of our links.

Nonetheless, we could do better. Accurate monitoring of the network at the fabric level can allow us to predict when bulk transfers will complete with greater accuracy, or allow us to choose optimal sources for a particular destination. Knowing when a set of files will arrive means we can co-schedule the analysis or processing jobs to arrive at the computing element just as the data arrives at the storage element, minimising latencies in the entire system.

Eventually, network control by reserving virtual circuits with dedicated bandwidth will prove even more powerful. Reserving high-bandwidth channels for specific dataflows will reduce transfer latencies by large factors, which opens the possibilities even further.


Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

In this paper we review the past and current state of network usage in CMS. We consider how network-aware systems can be integrated into our computing model, and examine some of the ways in which we can exploit them. We look at work that is being done now, and also consider what could be achieved in the longer term.

2. The past and present of network use in CMS

2.1. The past

The original computing model had the following characteristics:

- the Tier-0 is connected only to a hub of 7 Tier-1 sites. Data-flow from the Tier-0 is outbound, nothing gets imported from outside. N.B. At the time of writing there are actually 8 Tier-1 sites, as JINR prepares to take over from ASCG.
- The Tier-1 sites are all interconnected, each to every other. Data flows between them regularly, as they exchange data on behalf of their client Tier-2 sites.
- Each Tier-2 site is connected to one and only one Tier-1 site. Data is pulled from the Tier-1 for analysis, and Monte Carlo data produced at the Tier-2 is uploaded to the Tier-1 for custody or further distribution. Any data that the Tier-2 needs which is not available at its host Tier-1 is first retrieved from its source by the Tier-1 site, then sent to the Tier-2.
- In principle, we allowed the idea of interconnections between Tier-2 sites, but we expected this would be a small component of the topology, hard to commission and maintain.

It was thought that this model would simplify operations by limiting the number of network links to around 100, since maintaining these links in an operational state was assumed to be hard. Debugging problems spanning many network-domains was assumed to be something to avoid, and a hierarchy allowed us to maintain mostly in-domain links, with only major sites linking across domains. This topology is shown in figure 1.

A corollary of this is that for Tier-2 sites to exchange data, the data would have to be sent via one or two Tier-1 sites. The overhead was thought to be worthwhile, given the expected difficulty of transferring directly between Tier-2 sites; Tier-1 sites would have higher link-speeds to the Tier-2s, and greater network expertise available to resolve any problems.

It was further assumed, at the time of the Computing Model, that data-taking would start with relatively low network bandwidths between the Tier-0 and Tier-1 sites, and with lower speeds from the Tier-1 to Tier-2 sites. We expected that bandwidth would be a scarce resource, that our data-transfer needs would saturate the available bandwidth.

This influenced the design of the data-transfer management system, PhEDEx [2]. When a failure occurs transferring a particular file, PhEDEx backs off from aggressively retrying the transfer, and only retries after a reasonable interval. If there is a second failure, it backs off further still. The assumption is that if something is wrong it will eventually be fixed, meanwhile there are plenty of other files that could be successfully transferred, which would make better use of the bandwidth than persistently trying and failing on a single file. Consequently, PhEDEx cannot offer any guarantees of delivery within deadlines, only that it will persist until transfers eventually succeed.

Note that some of the low-level transfer tools PhEDEx uses, such as FTS, are implementing some of the same retry and backoff features we already employ in PhEDEx. This will not make those features obsolete in PhEDEx, as we do not wish to couple ourselves to specific transfer tools. The interplay between low-level and high-level error-handling mechanisms will need some consideration, but as is usual with networked applications, error-handling will still be needed at all levels.

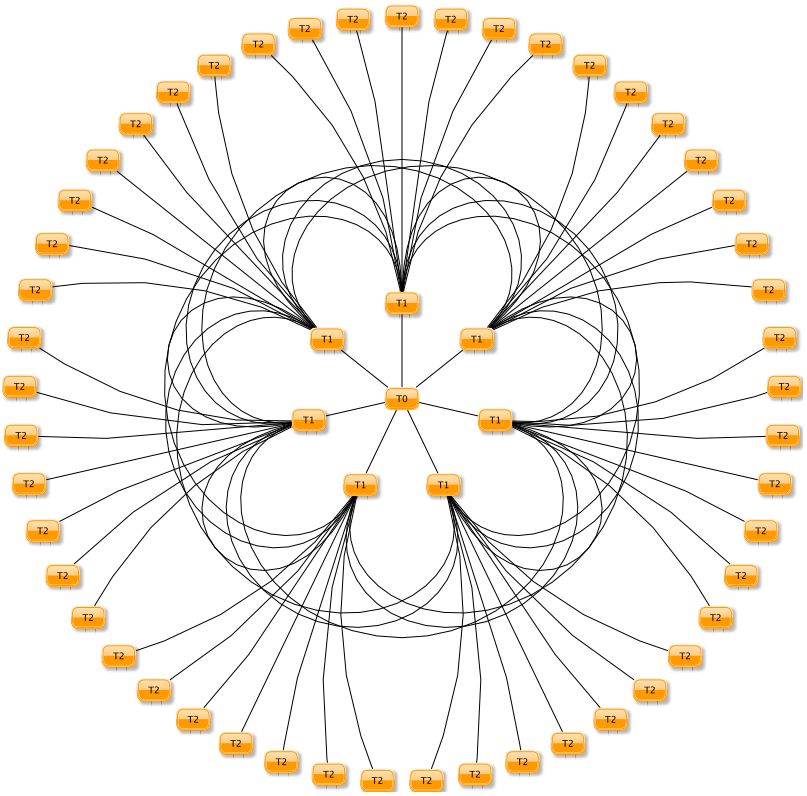

Figure 1. The original network topology envisioned by CMS. A Tier-0 is connected to a hub of Tier-1 sites, which are also all interconnected. Each Tier-1 serves several Tier-2 sites. Each Tier-2 site connects only to its parent Tier-1, not to any other site. The total number of network links is of the order of 100, depending on the exact numbers of Tier-1 and Tier-2 sites

2.2. The present

In practise, the LHC experiments have found the network to be highly reliable. Sustaining transfers between Tier-2 sites has proven to be very effective, offloading a lot of work from the Tier-1 sites and reducing the time taken to complete the transfers. The resulting network topology is almost a complete mesh, as shown in figure 2.

Network speeds are much higher than originally anticipated. The Tier-0 and Tier-1s are connected by the LHC Optical Private Network (LHCOPN [3]) with a 120 Gbps backbone. Many of the Tier-2s have 1-10 Gbps links to many other CMS sites. This is more than enough to satisfy our basic needs.

Throughout the first run of LHC in 2010, CMS has transferred 1 PB of data per week among its consitituent sites (figure 3). In 2007, a dedicated effort was made by the *Debugging Data Transfers* task force to commission the full-mesh transfer topology.

Since then, all transfers are classified as 'Production' (real or Monte Carlo data) or 'Debug' (for comissioning links or maintaining a heartbeat to monitor link status). Debug traffic accounts for one third of the total activity, since even a small transfer rate on so many links soon adds up to a lot of traffic. Comissioning new links is still an intensive process, but now that we have the expertise and monitoring tools we need, maintaining links is a routine operations task.

Figure 4 shows the Production traffic during the last year of Run 1, from May 2012 to May 2013. Traffic from the Tier-0 to the Tier-1s rises and falls in unison, as expected. Traffic from the Tier-2s to the Tier-1s is more erratic, with some Tier-1s receiving data almost all the time while others receive data only sporadically. The traffic to the Tier-2s shows even less structure;

some Tier-2s are active for much of the year but many receive data in short bursts at irregular intervals. Clearly we do not have smooth traffic flows, but instead have bursts on links which then remain idle for some considerable time. Average numbers do not describe our dataflow well.

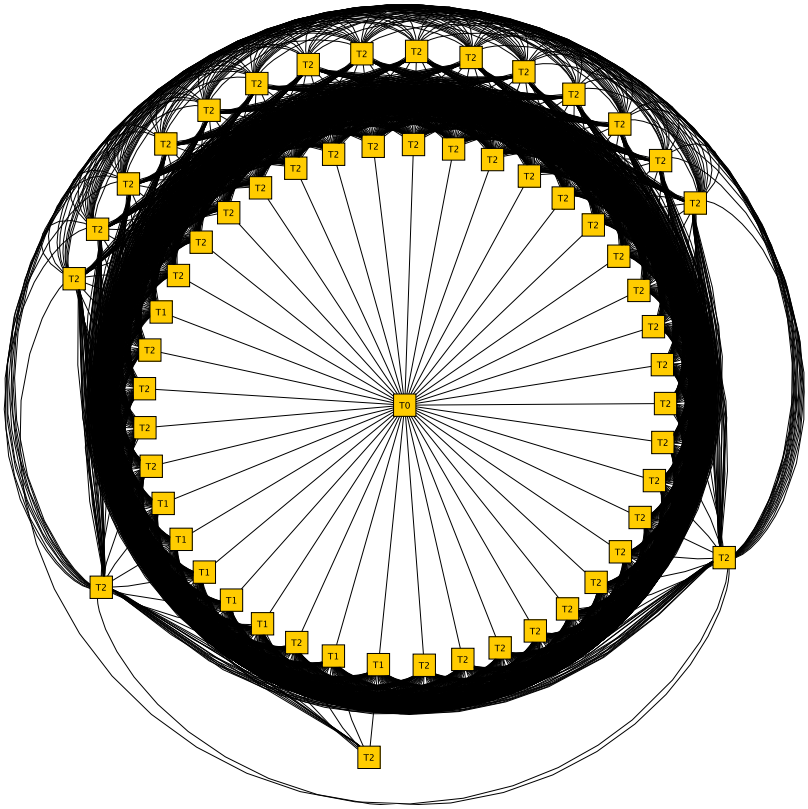

Figure 2. The actual network topology in use by CMS today. The Tier-0 is connected to all the Tier-1 sites and most of the Tier-2s, with about 60 outbound links in total. The Tier-1 and Tier-2 sites are almost all connected, bi-directionally, for about 100-120 links per site. Only a handful of sites are poorly interconnected with the rest of CMS, and work is underway to improve those sites' connectivity. There are about 2000 active links in total.

3. Making better use of the network

So the question we ask ourselves is: *how can we make better use of the network to improve analysis throughput?* Our current performance metrics are based on quantities like the time to transfer a dataset, the failure rate on a transfer link, or for batch jobs, the time to complete a given set of jobs and the CPU efficiency. These metrics are all related to the performance of the underlying machinery, not of the analysis itself. For example, we have no metrics that translate to the performance of an analysis that needs to examine multiple datasets.

3.1. The short term

That aside, there are things we can do to improve the situation already.

In the near-term future we have two projects that are approaching maturity. One is the *Any data, Anytime, Anywhere* (AAA [4]) project that aims to improve CPU efficiency by addressing a common problem, that of batch jobs failing because an input file cannot be read. By falling

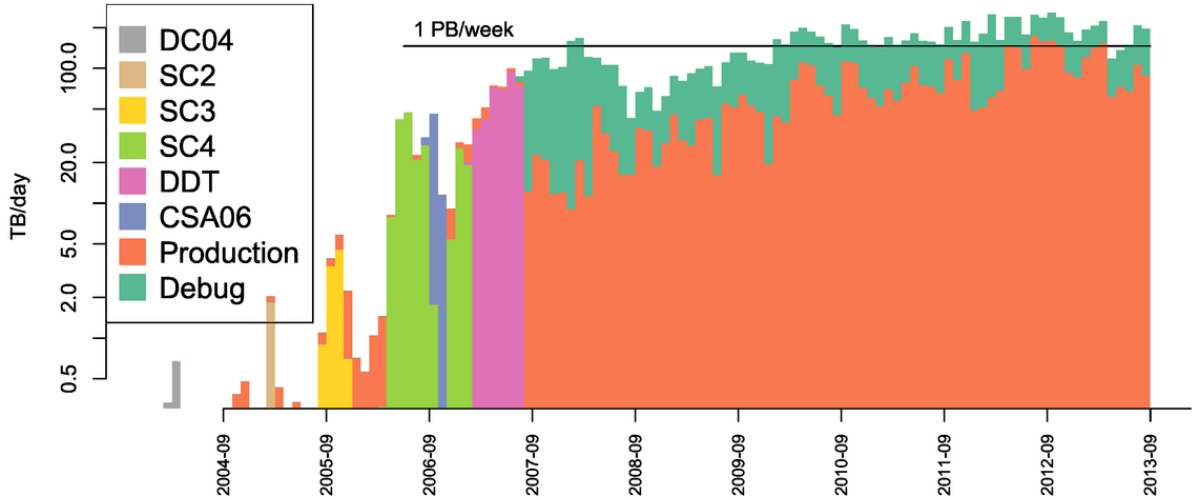

Figure 3. Total data transferred by PhEDEx per week since its inception in 2004. The rate has been above 1 PB per week throughout the first LHC run (from early 2010 to mid 2013). The DDT transfers (Debugging Data Transfers task force) marks the transition from MONARC topology to a fully integrated transfer mesh. Since then, only Production (real and Monte Carlo data) and Debug (link-comissioning and debugging transfers) have been needed. (N.B. Note the log scale)

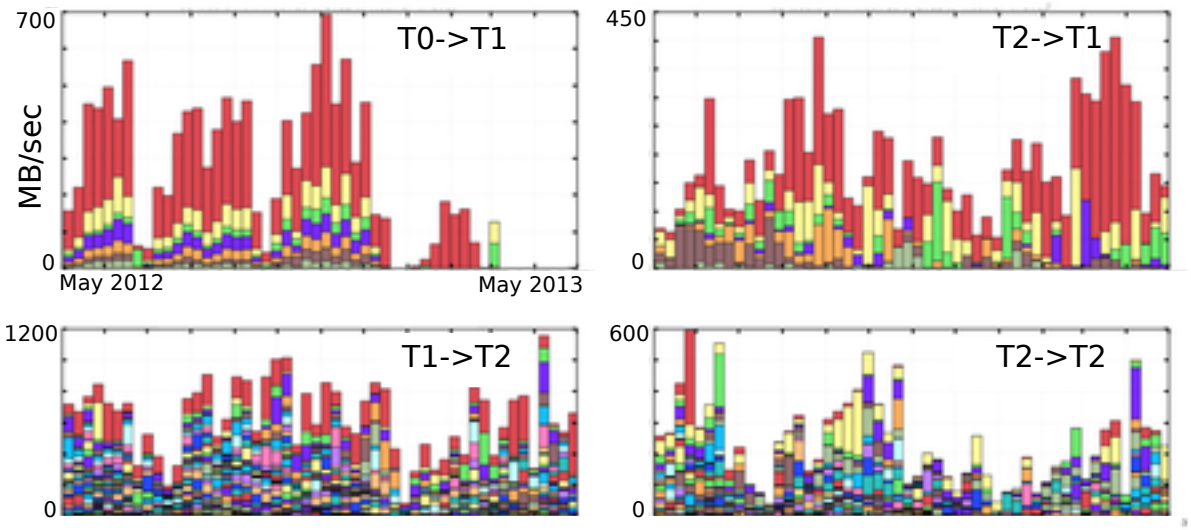

Figure 4. Transfers by destination during the last year of run 1, from May 2012 to May 2013. Note the variation in vertical scales; the Tier-0 to Tier-1 traffic and Tier-2 to Tier-1 traffic is small compared to the traffic to the Tier-2s. The Tier-0 to Tier-1 plot shows export of raw and reconstructed data. The Tier-2 to Tier-1 plot shows upload of Monte Carlo data produced at the Tier-2s. The Tier-1/2 to Tier-2 plots shows data exported for analysis. The volume of traffic between Tier-2s was not anticipated in the original MONARC model.

back to an xrootd [5] data federation, the batch job can be redirected to read another copy of the file across the WAN. The overall application performance will be lower than reading directly from the LAN, but still higher (on average) than having the job fail and be resubmitted. This is especially true if the file-read failure is persistent, which means that resubmitting the batch job will only result in another failure.

The other project is based on the CMS *Data Popularity Service* [6]. By measuring the access patterns of CMS batch jobs we can determine which data remain idle on disk for long periods of time, and use that knowledge to remove redundant copies, thereby freeing up disk space for more valuable data. Conversely, data that is accessed heavily can be replicated to other sites, to balance the load.

3.2. The longer term

Longer term, it is clear that having more responsive, dynamic data-placement will be important. We aim to reduce latencies inherent in the PhEDEx workflow from tens of minutes to a minute or less. Instead of backing off immediately on failed transfers, PhEDEx can retry from another replica, even if it is nominally from a less favourable source location.

These changes will reduce the latency for completing blocks of data on disk, so that analysis activities can start sooner. Beyond this, making data-transfers more aware of the state of the network, and giving PhEDEx the ability to actually control the network, will give significant improvements.

We can obtain network state information through monitoring tools such as PerfSONAR [7] or MonaLISA [8]. PhEDEx today maintains internal statistics on network performance, based on the rate at which data is successfully transferred between sites, but it knows nothing of the underlying network fabric or its condition. Such information can allow PhEDEx to make more intelligent decisions when choosing where to fetch data from, and to respond quickly to changes in circumstances, without waiting for a shift in the observed behaviour of transfers.

Explicit control of the network, by reserving virtual circuits with bandwidth guarantees, opens up many more possibilities:

- Improving the knowledge of the delivery time for bulk data. Having a more deterministic estimate of the time to complete transfers makes it feasible to co-schedule batch jobs with the data
- Just-in-time replication of datasets. Data can be replicated rapidly at short notice, in response to, for example, batch jobs being sent to idle resources that do not have their data present
- Better use of opportunistic resources. We currently use opportunistic resources mostly for simulation, because we have no easy way of getting data in or out in a timely manner. Virtual circuits could change that, so data can be pumped in, processed, and pumped out again. That allows for use of resources that are available periodically, such as clusters that are idle overnight.

One key to making this happen will be refactoring the two main roles of PhEDEx. PhEDEx currently serves as a data-store and a data-movement tool. It maintains knowledge of what data exists at sites, so that batch jobs can be directed there or the data can be copied from there to somewhere else. It also manages the transfer of that data, deciding which sources to use and which files to send at a given time to achieve the overall movement of data.

As a data-store, PhEDEx will become more dynamic. Not just the static map of CMS-managed sites, but opportunistic resources, from single machines to large clusters, must be able to be registered and un-registered on demand. This extends the scope of managed resources considerably.

As a data-movement tool, PhEDEx will become more like a content delivery network. It will be able to rapidly data between any sites or machines that it knows about, from the scale of a few tens of GB to the limits of the machine storage. Where appropriate, use of virtual circuits will guarantee performance.

3.3. Integrating virtual circuits into PhEDEx

There are several places in the PhEDEx software stack where control of virtual circuits can be integrated.

- In each transfer job. This already exists, using FDT [8], but is far from optimal. There is no policy to guide when a circuit should be reserved and when it is not worth doing, and no central optimisation to make sure that circuits can be fully utilised once they are booked.
- In each *FileDownload* agent [2], i.e. per destination. This is more interesting, especially if PhEDEx optimises the queue of each agent to make maximal use of the circuits.
- CMS-wide, where circuits are booked and managed by the central management agents. This has the potential to yield the best results, since the global flow of data can be controlled to make maximal use of each link.

The ANSE [9] project has begun work to integrate network circuit control into PhEDEx, following the hierarchy of options listed above. It manages the creation of virtual circuits using tools such as DYNES [10], so bridges the gap between the network fabric and the experiment's application layer.

It will be necessary for CMS to develop policies for managing virtual circuits, to decide when it is worth creating one and when it is not. For example, low priority traffic may not require a circuit, while high priority traffic will.

Reliable virtual circuits will be essential to this work, of course. There is ongoing work to extend the existing virtual circuit implementations to be compatible cross-domain, so that circuits spanning the breadth of the CMS organisation can be created and operated reliably.

4. Conclusion

CMS has been very successful so far with its data transfer management. PhEDEx is a mature tool, and our current data transfer needs are far from saturating the network bandwidth currently availble to CMS. Nonetheless, in order to continue to satisfy the needs of the experiment into the future, and to increase the flexibility and scope of our overall workflow and dataflow management tools, we need to make better use of the network. No longer treating it as a black box, we need to both respond to deeper knowledge of the network state, and to be able to control its behaviour wherever possible. This will affect our data transfer management at all levels, both operationally and in our software. Core components of PhEDEx will need to be adapted to exploit these network-aware systems, and the experiment will need to develop policies and procedures to use these new capabilities to the full. This work has already begun with the collaboration between CMS and the ANSE project.

References

[1] Grandi C, Stickland D and Taylor L 2005 The CMS Computing Model *CERN-LHCC-2004-35/G-083, CMS note 2004-031*
[2] Egeland R, Wildish T and Metson S 2008 Data transfer infrastructure for CMS data taking, *XII Advanced Computing and Analysis Techniques in Physics Research (Erice, Italy: Proceedings of Science)*
[3] Demar P and Bradley S 2013 WAN Data Movement Architectures at US-LHC Tier-1s *submitted to CHEP 2013*
[4] Bloom K 2013 CMS Use of a Data Federation *submitted to CHEP 2013*
[5] XROOTD, http://xrootd.slac.stanford.edu/

[6] Barreiro Megino F H, Cinquilli M, Giordano D, Karavakis E, Girone M, Magini N, Mancinelli V and Spiga D 2012 Implementing data placement strategies for the CMS experiment based on a popularity model *J. Phys.: Conf. Ser. 396 032047*
[7] Campana S et.al. 2013 Deployment of a WLCG network monitoring infrastructure based on the perfSONAR-PS technology *submitted to CHEP 2013*
[8] Legrand I, Newman H, Voicu R, Cirstoiu C, Grigoras C, Dobre C, Muraru A, Costan A, Dediu M and Stratan C 2009 MonALISA: An agent based, dynamic service system to monitor, control and optimize distributed systems *Computer Physics Communications, Volume 180, Issue 12, December 2009, Pages 24722498*
[9] Melo A et.al 2013 Integrating the Network into LHC Experiments: Update on the ANSE (Advanced Network Services for Experiments) Project *submitted to CHEP 2013*
[10] McKee S et.al 2013 Application Performance Evaluation and Recommendations for the DYNES *submitted to CHEP 2013*