Extending the distributed computing infrastructure of the CMS experiment with HPC resources

J Adelman-McCarthy[1], T Boccali[2], R Caspart[3], A Delgado Peris[4], M Fischer[3], J Flix Molina[4,5], M Giffels[3], J M Hernández[4], D Hufnagel[1], E Kühn[3], T Madlener[6], A K Mohapatra[7], H Ozturk[8], A Pérez-Calero Yzquierdo[4,5], D Spiga[9], C Wissing[6] for the CMS Collaboration

[1]Fermi National Accelerator Laboratory, Batavia, IL, USA
[2]INFN Sezione di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy
[3]KIT, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
[4]CIEMAT, Av. Complutense 40, Madrid, Spain
[5]Port d'Informació Científica (PIC), Barcelona, Spain
[6]DESY, Notkestr. 85, 22603 Hamburg, Germany
[7]University of Wisconsin-Madison, Madison, WI, USA
[8]CERN, 1211 Geneva 23, Switzerland
[9]INFN Sezione di Perugia, Via A. Pascoli 23c, 06123 Perugia, Italy

E-mail: daniele.spiga@pg.infn.it christoph.wissing@desy.de

Abstract. Particle accelerators are an important tool to study the fundamental properties of elementary particles. Currently the highest energy accelerator is the LHC at CERN, in Geneva, Switzerland. Each of its four major detectors, such as the CMS detector, produces dozens of Petabytes of data per year to be analyzed by a large international collaboration. The processing is carried out on the Worldwide LHC Computing Grid, that spans over more than 170 compute centers around the world and is used by a number of particle physics experiments. Recently the LHC experiments were encouraged to make increasing use of HPC resources. While Grid resources are homogeneous with respect to the used Grid middleware, HPC installations can be very different in their setup. In order to integrate HPC resources into the highly automatized processing setups of the CMS experiment a number of challenges need to be addressed. For processing, access to primary data and metadata as well as access to the software is required. At Grid sites all this is achieved via a number of services that are provided by each center. However at HPC sites many of these capabilities cannot be easily provided and have to be enabled in the user space or enabled by other means. At HPC centers there are often restrictions regarding network access to remote services, which is again a severe limitation. The paper discusses a number of solutions and recent experiences by the CMS experiment to include HPC resources in processing campaigns.

1. Introduction

Currently the strongest particle accelerator is the Large Hadron Collider (LHC) at CERN, which collides protons at a center of mass energy of $\sqrt{s} = 13\,\text{TeV}$. One out of four experiments is the CMS detector that comprises almost 100 million readout channels. Even zero suppressed the data rate at 40 MHz would be several tens of Terabytes per second being unfeasible to store. Therefore the rate is reduced by several orders of magnitude by a trigger system that selects

events at a rate of roughly 1 kHz resulting in a manageable data rate in the order of 1 GB/s. To analyze the data roughly the same amount of events is generated by Monte Carlo (MC) simulations. Overall the CMS experiment creates dozens of Petabytes of data each year.

Traditionally the data is being processed using resources of the Worldwide LHC Computing Grid (WLCG) [1], that consists of more than 170 compute sites. The CMS experiment is supported at about 60 of them. Compute and storage resources are accessed via standardized components, Compute Elements (CEs) and Storage Elements (SEs), allowing the execution of workflows across many locations. Most of the CMS payloads are executed by a central production team following an agreed production plan from the entire collaboration.

2. Integration of HPC resources

In its present approach for distributed computing the CMS experiment is targeting a sizable number of sites with a spectrum of different applications and workflows. Since these sites are all very similar regarding their integration the required effort to operate such an infrastructure is still reasonable. In contrast, HPC applications are typically highly tuned for one specific machine in order to fully exploit that particular system. From a technical perspective HPC centers differ on a variety of specialized hardware setups and policy wise strict usage rules can apply mostly for security reasons. As a consequence, a variety of conditions can be encountered, when attempting to integrate these centers. Examples are less common operating systems, absence of local scratch disk space on compute nodes, low memory available per core, limited / absent outbound network connectivity, and unusual architectures. After several investments by the CMS collaboration over the past few years a number of HPC machines has been integrated and continuously been used in production mode throughout the years 2020 and 2021. Most of the machines have been able to execute all types of CMS workflows.

2.1. HEPCloud

HPC resources located in the US are integrated into the CMS systems via HEPCloud [2]. At the moment the existing HEPCloud system provides access to various HPC and commercial cloud resources, but in a targeted way. That means jobs have to be explicitly targeted for the desired resource, without HEPCloud itself making any decision to route the jobs. Work on the Decision Engine continues to make the job routing more intelligent and automatic. During 2020 and early 2021 HEPCloud provisioned resources from a few different HPC centers, namely: the Cori cluster at NERSC (National Energy Research Scientific Computing Center); the Bridges and Bridges-2 clusters at PSC (Pittsburgh Supercomputing Center); the Stampede2 and Frontera clusters at TACC (Texas Advanced Computing Center); the Comet and Expanse clusters at SDSC (San Diego Supercomputer Center). NERSC is an HPC user facility operated by Lawrence Berkeley National Laboratory for the United States Department of Energy (DOE) Office of Science. The other HPC, except Frontera, are funded through the United States National Science Foundation (NSF) XSEDE (eXtreme Science and Engineering Discovery Environment) project [3]. Frontera is also funded by NSF, but not part of XSEDE. These HPC resources represent a mix of various Intel and AMD CPUs.

The technical integration of HPC resources into the HEPCloud and CMS workflow management systems can be summarized as *make it look like a Grid site*. There are two major components to this approach. First is the connection of the HPC batch system to the CMS pilot submission system, usually accomplished via remote submissions through the HPC login nodes. Second is the configuration of the HPC runtime environment through the use of containers (they all support `Singularity`, except NERSC, where `shifter` is used). In addition, CVMFS [4] is available at all these HPC sites, either provided directly by the site or by mounting it in user space with the `cvmfsexec` package [5]. All these HPC sites provide outbound internet connectivity from the worker nodes as well, which makes integration a lot easier. Job input is

read from remote via the AAA federation [6] from other CMS sites, and any job output is staged out directly to FNAL storage.

2.2. CINECA

CINECA [7], a PRACE [8] Tier-0 facility in Bologna (Italy) currently hosts a system, Marconi [9], which was ranked no. 21 in the top500.org Nov 2019 list[10]. The Italian LHC community received an allocation on the Marconi A2 partition, which deploys 3600 nodes equipped with one Xeon Phi 7250 (KNL) each. Since the standard Marconi A2 nodes are not immediately usable by CMS workflows, CMS and CINECA agreed on a minimal set of changes:

- CVMFS was installed on the systems and Squids were deployed by CINECA on edge nodes;
- external outbound networking was partially opened, with routing active to CERN and CNAF IP ranges; input files from other sources were made available via an Xrootd proxy cache at CNAF prepared and managed in synergy with XDC [11] and ESCAPE [12];
- the `Singularity` virtualization tool was audited by CINECA;
- an HTCondor CE was allowed on a CINECA edge node, with access to the external IP ranges as above, and the ability to submit to the internal SLURM batch system;
- in order not to overload CINECA's GPN, a 40 Gbit/s connection was established with CNAF via an Infinera DCI router, on a private dark fiber which was already in place.

Following the CNAF Tier-1 strategy [13] the CINECA nodes were configured as an elastic extension of CNAF Tier-1 [14], and were receiving all the jobs targeted for the standard site. On top of that a key aspect of the integration is the payload selection strategy. The Marconi resources were enabled to specify additional requests with respect to CNAF nodes in order both to select most suitable workflow (low memory, low IO, resize-able jobs), and to veto specific payloads like end user analyses.

By the end of 2020 CMS had used 26.7 million CPU hours at CINECA Marconi A2. The workflow success rates and CPU efficiency was close to similar workflows running at standard WLCG sites. INFN will exploit stable allocations on the newer Marconi M100 [15] cluster (IBM Power9 architecture with NVIDIA Volta GPUs). Upon successful physics validation it could be the first non-x86 system to reach production level.

2.3. Managing temporarily available resources using `COBalD/TARDIS`

Managing temporarily available resources not dedicated to the WLCG requires operational effectiveness, dynamic integration as well as transparent and unified access to such resources. This is realized by using HTCondor as an overlay batch system (OBS) which abstracts resource management and hides the short-term availability of individual resources by presenting a single pool of all resources. The well-established technology of Grid CEs is used to present a single point of entry for the experiments. Together, this minimizes the efforts for integrating those resources into the existing computing infrastructure of the experiments [16].

The Transparent Adaptive Resource Dynamic Integration System (`TARDIS`) [17] has been developed at the Karlsruhe Institute of Technology (KIT) to enable dynamic resource provisioning, integration into the OBS and life cycle management. `TARDIS` provides a variety of interfaces for common batch systems as well as common cloud APIs [16]. `TARDIS` builds on `COBalD` [18], a lightweight framework to balance opportunistic resources. In contrast to similar resource managers available, it does not rely on predictions for the current workloads in distributed computing infrastructures. Instead `COBalD` uses a feedback control loop to estimate required resources based on observed usage [19].

KIT operates such an OBS and entry points.Via this OBS, CMS jobs are run on resources of the KIT HPC cluster (ForHLR II) [20], (HoreKa) [21] and KIT university cluster (TOPAS), for a total of around 2.1 million core hours in 2020 and over 15 million core hours in 2021.

All integrated resources provide x86 compatible CPUs, allow outgoing network connections and support `Singularity`. This allows to provide the common WLCG environment using `Singularity` containers. The required Grid and experiment software is made available using CVMFS, either mounted on the nodes themselves or provisioned in user space using `cvmfsexec`.

Recently, a similar setup was deployed at the RWTH Aachen University to integrate resources of the CLAIX HPC cluster [22] into the co-located RWTH Aachen WLCG centre, temporarily doubling its available compute resources and providing around 1.2 million core hours used by the CMS experiment in December 2020 and around 2.3 million core hours in 2021.

3. Production experiences

HPC sites might require special treatment operationally, due to their differences from WLCG sites. The workflow assignment procedure determines the selection of sites that are going to execute the payloads. For HPC sites the procedure needs special treatment.

In the cases of CINECA and `COBalD/TARDIS` there is no need for special actions from the operational perspective, because the HPC resources appear as CPU extensions of existing sites and the workflow selection is managed at site level by cherry-picking suited payloads. In order to cope with the slow KNL core performance CINECA selects at site level workflows with expected execution times that meet the batch queue limit.

For the assignment of workflows to HEPCloud the CMS computing operations team experimented with several methods. In a first approach long pending jobs at other sites got re-routed to HPCs, but this often let to insufficient job pressure for HPCs and also to jobs that were not well suited for an HPC site. The second approach was to manually assign complete workflows to HPC sites. While this led to good utilization of HPC resources, it causes significant operational overheads. The third approach was an automation of the second: Workflows, which do not need intermediate data on the local storage, are assigned to all available CMS sites including the HEPCloud ones in an automated manner. HEPCloud sites start to compete for the workflows together with WLCG sites and get their share over the HTCondor batch system. This works best so far to utilize the HPC resources behind HEPCloud, preventing manual work and avoiding data access failures.

Overall HPC allocations have provided a significant contribution of CPU hours to CMS. Table 1 summarizes the provided wall clock times for the years 2020 and 2021. The amount of wall clock hours more than doubled form 2020 to 2021. In the recent months HPCs have contribute between 5% to 10% to the overall CPU capacity of the CMS experiment.

Table 1. Consumption of CPU wallclock (WC) time, measured in k hours, of HPCs systems exploited by CMS in 2020 and 2021.

HPC	Country	Integration method	WC k hours 2020	WC k hours 2021
SDSC	US	HEPCloud	6269	21480
TACC	US	HEPCloud	11730	51656
PSC	US	HEPCloud	12198	20583
NERSC	US	HEPCloud	33014	76334
ForHLRII	DE	`COBalD/TARDIS`	1233	1311
HoreKa	DE	`COBalD/TARDIS`	—	13266
CLAIX	DE	`COBalD/TARDIS`	1274	2347
CINECA	IT	Site Extension	26721	2634
BSC	ES	HTCondor SplitStarter	—	12373

4. Developments on access to network-segregated nodes

One of the main hurdles when trying to incorporate HPC systems in the CMS distributed computing resides in the need to access external services at run time, such as access to condition data via Squids. On top of that, the late binding pilot mechanism requires the compute node to contact central HTCondor services at CERN or FNAL. Multiple R&D efforts are ongoing in CMS in order to mitigate these limitations.

4.1. HTCondor split-starter model

The Barcelona Supercomputing Center (BSC) is the main HPC center in Spain, hosting a general-purpose cluster (MareNostrum) with a peak power of 11.15 Petaflops, on 3,456 nodes with 165,888 cores and 390 TB of memory. The BSC execute nodes have no external connectivity, and consequently, substantial integration work has been required in order to be used by CMS. The only available communication channel with the outside world are incoming SSH connections into the BSC user login machines. A shared file system (GPFS), mounted on the execute nodes and on the login machines, is accessible from the outside via sshfs. A collaboration was formalized between the HTCondor team and CIEMAT members at the Spanish WLCG Tier-1 at PIC to use GPFS as communication path for HTCondor [24]. The objective of such work is not only to help solving the BSC case, but to provide a general solution for similar scenarios elsewhere.

This implementation was first tested to run realistic CMS jobs in September 2020. A test workflow was defined to execute simulation jobs that read pre-placed input data. All necessary application software was included in a custom `Singularity` image used for the jobs, and a database file with the experimental conditions for the simulation was copied to BSC storage. This prototype underwent scalability exercises, where adequate performance from all components was observed. A single submitter node (bridge) at PIC was able to fill up 10,000 CPU cores at BSC. This bridge service can be horizontally scaled up, so higher scales can be easily achieved.

Additional functionalities have been added to the initial prototype towards its production rollout. The whole CMS CVMFS software repository was replicated to BSC storage (37M files, 13 TB), so that custom singularity images are no longer necessary. This repository is kept up to date using the cvmsfs_preload client tool. The transfer of output files produced by BSC jobs is now handled by a custom-made data transfer service (DTS). File transfers between BSC and PIC storage are implemented synchronously with the jobs through transfer requests to the DTS at the post-job stage. The whole system has been successfully exercised executing realistic simulation workflows orchestrated by the WMAgent-based CMS production management system.

4.2. R&D on tsocks

`tsocks` is a networking tool designed to trap system calls and allow rerouting TCP+UDP connections via multiple means. CMS developed a solution using tsocks in combination with a SOCKS5 tunnel to overcome networking restrictions at HPC sites[25]. While generic outgoing networking from compute nodes can be filtered, the same nodes are usually able to reach hosts, which are instead able to route externally via some gateway node. In the simplest implementation, the user running a payload on a compute node does not need to get admin level access to these "edge" nodes, but can setup a tunnel simply using an `ssh` connection. Once established, preloading the `libtsocks.so` library allows for a full redirection of traffic via the SOCKS5 tunnel.

4.3. R&D on HTTP CONNECT TCP tunnel

Another approach under testing relies on Squid proxies to provide TCP tunneling capabilities, since CMS usually needs to have them available locally at all resources to access conditions data. A client can connect to them via a HTTP CONNECT call and instruct them to open a TCP connection to route traffic to an arbitrary other host and port. By default this is only

enabled for port 443, but can be enabled for other ports with a simple configuration change. Very similar to the previously mentioned tsocks approach, a preloader is needed to use existing TCP clients [26]. Already mentioned bandwidth limitations make it hard to see this being used for all possible traffic by CMS jobs at scale (in addition the HTTP CONNECT tunnel is only for TCP connections, not UDP), but it could still be useful for tunneling specific traffic.

Along these lines further investigations based on Virtual Private Network (VPN) are being carried out. When the user namespace feature is enabled on the compute node and on the edge node running the VPN server, a tunneling setup could be achieved, that runs entirely in user space without the need for root privileges. If namespaces are not enabled, then the strategy falls back to using a SOCKS server instead of the virtual network interface.

5. Conclusions and future steps

The WLCG infrastructure is generally considered as a success and it has allowed to carry out the necessary tasks for the scientific program of the experiments. However it is mainly used by particle physics research. There is a recent trend towards research infrastructures shared by many scientific disciplines. The evaluation of HPC resources for particle physics is a direct consequence. HPC sites that allow external network connections have been used successfully for a large fraction of CMS workflows. The integration effort as well as the additional operations load was found to be reasonable. HPC resources without external network connectivity pose quite a challenge regarding their integration into the existing system. Nevertheless there are promising R&D efforts to at least overcome these limitations partly.

For the near future a goal of CMS is to increase the capability to integrate HPC resources, to exploit the huge amount of GPUs as well as CPU architectures other than x86. Both will be beneficial to validate the physics performance of the CMS software and its readiness for other platforms.

Acknowledgements

The authors and the CMS collaboration as a whole would like to thank the cited HPC centers for their collaboration, support, and interest in the computing activities of CMS.

The activities at CINECA are funded via a PRACE Project Access Grant (# 2018194658), and were partially supported via the EU projects ESCAPE (Grant agreement 824064) and XDC (Grant agreement 777367). The CINECA integration was made possible also thanks to the support by INFN-CNAF, and in particular by S. Dal Pra, S. Zani and L. Morganti.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

This research is part of the Frontera computing project at the Texas Advanced Computing Center. Frontera is made possible by National Science Foundation award OAC-1818253.

References

[1] I. Bird et al., "Architecture and prototype of a WLCG data lake for HL-LHC", *EPJ Web Conf.* **214** (2019) 04024. https://doi.org/10.1051/epjconf/201921404024

[2] P. Mhashilkar et al., "HEPCloud, an Elastic Hybrid HEP Facility using an Intelligent Decision Support System", *EPJ Web of Conf.* **214** (2019) 03060 https://doi.org/10.1051/epjconf/201921403060

[3] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, N. Wilkins-Diehr, "XSEDE: Accelerating Scientific Discovery", *Computing in Science & Engineering* **160**, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80

[4] J. Blomer et al, "CernVM-FS: delivering scientific software to globally distributed computing resources", Proceedings of the first international workshop on Network-aware data management, November 2011, pp 49–56, https://doi.org/10.1145/2110217.2110225
[5] J. Blomer, D. Dykstra, G. Ganis, S- Mosciatti, J. Priessnitz, "A fully unprivileged CernVM-FS", *EPJ Web of Conf.* **245** (2020) 07012, https://doi.org/10.1051/epjconf/202024507012
[6] K. Bloom *et al.*, "Any Data, Any Time, Anywhere: Global Data Access for Science," arXiv:1508.01443
[7] CINECA homepage, https://www.cineca.it/en
[8] PRACE homepage, https://prace-ri.eu/
[9] Marconi overview webpage, https://www.hpc.cineca.it/hardware/marconi
[10] Top500 list Nov 2019, https://top500.org/lists/top500/list/2019/11/
[11] XDC homepage, http://www.extreme-datacloud.eu
[12] ESCAPE homepage, https://projectescape.eu
[13] L. Dell'Agnello et al., "INFN Tier–1: a distributed site", The European *Physical Journal Conferences* **214** (2019) 08002, https://doi.org/10.1371/journal.pone.0177459
[14] T. Boccali, S. Dal Pra, D. Spiga et al., *EPJ Web Conf.* **245** (2020) 09009 DOI: 10.1051/epjconf/202024509009
[15] Marconi100 overview webpage, https://www.hpc.cineca.it/hardware/marconi100
[16] M. Fischer, M. Giffels, A. Heiss, E. Kuehn, M. Schnepf, F. von Cube, A. Petzold, G. Quast, "Effective Dynamic Integration and Utilization of Heterogenous Compute Resources", *EPJ Web of Conf.* **245** (2020) 07038, https://doi.org/10.1051%2Fepjconf%2F202024507038
[17] M. Giffels, M. Schnepf, et al., "TARDIS – Transparent Adaptive Resource Dynamic Integration System", https://doi.org/10.5281/zenodo.2240605
[18] Max Fischer, et al., "COBalD – The Opportunistic Balancing Daemon", http://doi.org/10.5281/zenodo.1887872
[19] M. Fischer, E. Kuehn, M. Giffels, M.J. Schnepf, A. Petzold, A. Heiss, "Lightweight dynamic integration of opportunistic resources", *EPJ Web of Conf.* **245** (2020) 07040, https://doi.org/10.1051%2Fepjconf%2F202024507040
[20] ForHLR II homepage, https://www.scc.kit.edu/en/services/10398.php
[21] HoreKa homepage, https://www.scc.kit.edu/en/services/horeka.php
[22] CLAIX homepage, https://www.itc.rwth-aachen.de/cms/IT-Center/Forschung-Projekte/High-Performance-Computing/~eucm/Infrastruktur/?lidx=1
[23] BSC homepage, https://www.bsc.es/
[24] C. Acosta-Silva, A. Delgado Peris et al., "Exploiting network restricted compute re-sources with HTCondor: a CMS experiment experience", *EPJ Web of Conf.* **245** (2020) 09007
[25] M. Mariotti, D. Spiga, T. Boccali, "A possible solution for HEP processing on network secluded computing nodes", *Proceedings of Science* **378** 002
[26] https://github.com/rofl0r/proxychains-ng