Integration of the Barcelona Supercomputing Center for CMS computing: towards large scale production

C. Acosta-Silva[1,2,*], *A.* Delgado Peris[3], *J.* Flix Molina[2,3,**], *J.M.* Hernández[3], *A.* Pérez-Calero Yzquierdo[2,3], *E.* Pineda Sánchez[4], and *I.* Villalonga Domínguez[4], on behalf of the CMS Collaboration.

[1]IFAE, The Barcelona Institute of Science and Technology, 08193 Bellaterra (Barcelona), Spain
[2]PIC, 08193 Bellaterra (Barcelona), Spain
[3]CIEMAT, Scientific Computing Unit, 28040 Madrid, Spain
[4]Autonomous University of Barcelona, 08193 Bellaterra (Barcelona), Spain

Abstract. The CMS experiment is working to integrate an increasing number of High Performance Computing (HPC) resources into its distributed computing infrastructure. The case of the Barcelona Supercomputing Center (BSC) is particularly challenging as severe network restrictions prevent the use of CMS standard computing solutions. The CIEMAT CMS group has performed significant work in order to overcome these constraints and make BSC resources available to CMS. The developments include adapting the workload management tools, replicating the CMS software repository to BSC storage, providing an alternative access to detector conditions data, and setting up a service to transfer produced output data to a nearby storage facility. In this work, we discuss the current status of this integration activity and present recent developments, such as a front-end service to improve slot usage efficiency and an enhanced transfer service that supports the staging of input data for workflows at BSC. Moreover, significant efforts have been devoted to improving the scalability of the deployed solution, automating its operation, and simplifying the matchmaking of CMS workflows that are suitable for execution at BSC.

1 Integration of HPC resources into CMS computing

The Compact Muon Solenoid (CMS) [1] experiment is actively pursuing an expanded utilization of High Performance Computing (HPC) resources to meet its escalating computing demands and gain access to cutting-edge computing technologies prevalent at HPC sites. The current landscape of international scientific projects witnesses a remarkable growth in HPC center funding, thereby prompting funding agencies to encourage their LHC national communities to leverage these resources to fulfill, at least in part, their computational needs. As CMS prepares for the future, particularly the anticipated surge in computing requirements during the mid to long term (Run 3 and High-Luminosity LHC), the relevance of adopting HPC becomes even more pronounced [2].

[*] The authors acknowledge support by Spanish funding agency AEI, grants PID2019-110942RB-C21 and PID2020-113807RA-I00, and PIC, the BSC and the Spanish Supercomputing Network (RES) for the resources.

[**] Corresponding author: `jflix@pic.es`

The seamless integration of HPC sites into the future High-Luminosity LHC strategy is considered a crucial component of the Worldwide LHC Computing Grid [3] (WLCG) vision. While CMS is dedicated to optimizing the utilization of non-Grid opportunistic CPUs, including those offered by HPC facilities, it currently faces challenges in transitioning from traditional computing resources available at WLCG sites to HPC resources.

1.1 Network requirements for HPC resource exploitation by CMS

The CMS processing applications need to access external services at run time, such as the conditions Database (CondDB) via the Frontier system [4], CMS software via CernVM-FS [5] (CVMFS), or input samples using Xrootd [6]. Additionally, CMS employs a late-binding resource allocation model, in which *pilot* jobs are submitted to compute nodes, and only at runtime do they retrieve real tasks to compute (*payload* jobs). For this, they require connectivity to a central scheduling queue at CERN.

In certain cases HPC centers do not allow outgoing connectivity from the compute nodes, effectively limiting the potential resources that CMS can integrate using standard mechanisms. Multiple R&D efforts have been performed in CMS in order to overcome these limitations [7–9]. Some of these rely on the existence of edge services or special nodes at the HPC center that can bridge network connectivity to the outer world. In this paper, we have addressed one of the most restrictive HPC scenarios encountered by CMS, where not only compute nodes are not exposed to WAN, but also no privileged nodes are available. We present the solutions that have been developed to enable its exploitation by CMS, going from testbeds [10] to a full-production at scale service.

2 The Barcelona Supercomputing Center

The Barcelona Supercomputing Center (BSC) [11] is the largest HPC center in Spain, and it will soon host one of the most powerful HPC facilities in the EU, expected to reach pre-exascale capabilities in the near future [12]. The biggest general-purpose cluster at BSC is MareNostrum (currently in its fourth phase, MN4), which comprises 3,456 compute nodes, each with two Intel Xeon Platinum chips fitted with 24 CPU cores, amounting to a total of 165,888 cores and a total memory of 390 TB. The MN4 peak power is 11.15 Petaflops.

In 2020, a collaboration agreement was signed between BSC and the Spanish LHC community, making LHC computing a strategic project, with access to a stable share of BSC CPU. Up to 7% of the MN4 slots, with additional opportunistic use, are dedicated to LHC activities (in particular, for ATLAS, CMS and LHCb). The aim is to provide a significant fraction of the CPU budget pledged by the Spanish LHC community, enough to fully cover the Spanish share of the LHC experiments data simulation needs. This corresponds approximately to 50% of the Spanish CPU pledge to WLCG, expected to consolidate by 2024.

The MN4 compute nodes have no external Internet connectivity, making their use by CMS a difficult task. A shared file disk storage system (GPFS [13]) is mounted on the compute nodes and also on the user login machines, and can be made available to the outside via sshfs. However, although the MN4 login machines (the only entry points) allow incoming connection through ssh, they provide no outgoing network connectivity. In addition, only short-duration processes can be executed on them. Therefore, no edge or proxy services can be run on those login nodes. Additionally, the BSC does not provide dedicated access points to internally allocated disk storage areas that can be directly integrated in the data management infrastructure used by CMS for its distributed storage. All these constraints make the whole scenario a real challenge for its exploitation by CMS.

3 Integration of BSC resources for CMS: adopted solutions

Given the demanding scenario regarding the utilization of BSC resources by CMS, significant integration efforts have been required to ensure that BSC can effectively run CMS jobs. A collaboration was formalized between the HTCondor development team and CIEMAT members at the Spanish WLCG Tier-1 center at Port d'Informació Científica (PIC, in Barcelona) to address the communication channel between HTCondor processes required in the CMS late binding model. The use of the shared filesystem as a communication path for HTCondor daemons was adopted, replacing the standard network communication between them. The proposed *split-starter* model [14], illustrated by Figure 1, was implemented to solve the BSC case and provide a general solution for the case of network-restricted resources.

This model requires a bridge node, installed at PIC, running HTCondor startd daemons. This bridge node mounts a pre-determined area of the GPFS shared filesystem at BSC via sshfs. Additionally, it also submits jobs to the BSC Slurm workload manager [15], via the BSC login node, in order to acquire resources at BSC. Slurm jobs act as pilot jobs, instantiating HTCondor starter processes on the BSC compute nodes. These BSC starter processes employ a rendezvous area in GPFS to connect back to the respective mirror starter processes running at PIC's bridge. Once this happens, startd processes at PIC's bridge, managing the resources allocated at BSC compute nodes, can join the CMS Global Pool [16] of resources. As the CERN-PIC connection is established, job wrapper scripts and input sandboxes are passed as tar files from the bridge to the BSC, via sshfs and GPFS file copy. Likewise, execution status files and produced results are copied back from BSC to PIC. From a functional perspective, the resources allocated at BSC can be regarded as standard compute nodes at PIC. In order to enable their late-binding matchmaking to the central task scheduling processes at CERN (schedds), the BSC slots are tagged as an extension of PIC, which facilitates their monitoring, commissioning, and accounting as part of the PIC contribution to CMS.

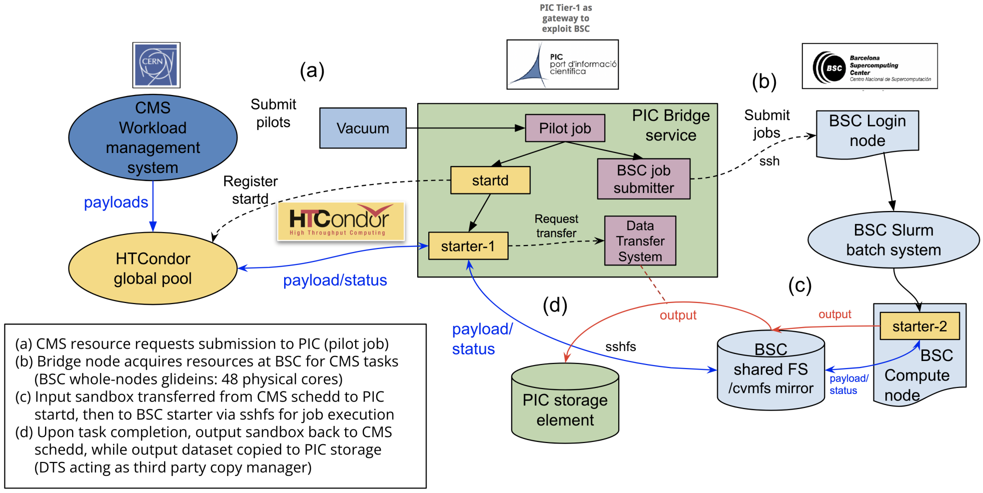
PIC Tier-1 as
gateway to
exploit BSC
PIC
port d'informació
científica
Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación
(a)
(b)
(c)
(d)
CMS
Workload
management
system
Submit
pilots
Vacuum
PIC Bridge
service
Pilot job
startd
BSC job
submitter
Submit
jobs
ssh
BSC Login
node
Register
startd
payloads
HTCondor
High Throughput Computing
starter-1
Request
transfer
Data
Transfer
System
BSC Slurm
batch system
HTCondor
global pool
payload/status
payload/
status
output
sshfs
output
starter-2
BSC
shared FS
/cvmfs mirror
payload/
status
BSC
Compute
node
PIC storage
element
(a) CMS resource requests submission to PIC (pilot job)
(b) Bridge node acquires resources at BSC for CMS tasks
(BSC whole-nodes glideins: 48 physical cores)
(c) Input sandbox transferred from CMS schedd to PIC
startd, then to BSC starter via sshfs for job execution
(d) Upon task completion, output sandbox back to CMS
schedd, while output dataset copied to PIC storage
(DTS acting as third party copy manager)

Figure 1. HTCondor split-starter model applied to the BSC case, as well as other services to get a production-ready environment, described in the main text.

The CMS processing software (CMSSW) is transferred to sites via CVMFS, through a system of tiered HTTP caches, built upon the standard web caching tool Squid. Since CVMFS is not available at BSC, we used the cvmfs_preload client to initially replicate the whole CMS CVMFS repository at the BSC storage (12.6 TB, 183M files), a process that took around two weeks to complete. Since then, periodic updates are run at PIC directly into an sshfs mount of BSC shared filesystem. This process is much faster (it completes in a few tens of minutes). CMS jobs run at BSC in custom singularity [17] container images with a preferred OS system for CMS execution tasks, and a pointer to the area in which all the CMSSW releases are locally available.

The lack of outbound network connectivity from BSC compute nodes prevents CMS jobs from reading conditions data at run time from a Frontier cache located at PIC. One attempted solution was to read conditions data from pre-placed static SQLite files, which required developments in CMSSW. This solution was very difficult to operate, since it required identifying suitable workflows, using appropriate CMSSW versions and matching available conditions data files, and entailed the manual non-trivial production of SQLite files.

A more sophisticated solution was finally implemented to enable the connection of the BSC compute nodes to the Frontier cache service at PIC. It involves splicing two SSH tunnels through BSC's login node, a reverse one from the bridge node, and a forward one from the compute node, as sketched in Figure 2. Two tunnels are created for each Slurm job, one of them being set as a backup proxy for the Frontier client. The CMS pilot checks that the tunnels are working properly before starting each new payload, but, if the tunnel process is interrupted for any reason, it is re-established automatically. This solution relieves CMS from producing ad-hoc SQLite conditions files, makes it easier to find suitable workflows for BSC, and provides a lighter and more standard way to run workflows at BSC. Since processes on login nodes are allowed to consume only up to 5 minutes of CPU time, the established ssh tunnels cannot be used to transfer massive amounts of data (e.g. input or output data files).

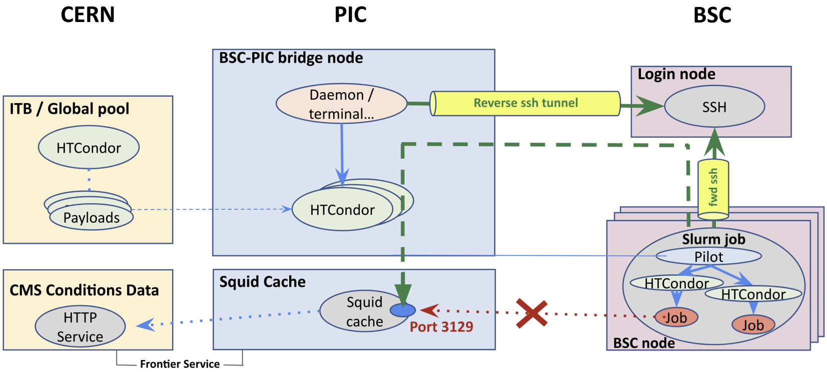

Figure 2. Reverse/forward ssh tunnels to connect BSC compute nodes to the PIC Frontier cache service.

The management of input and output data files, consumed and produced by CMS jobs at BSC, and exposed to CMS, is also a critical task that has been addressed. Input files need to be made accessible to jobs running at BSC and output files have to be transferred to a CMS storage element and registered in the CMS data book-keeping and location services. Since BSC jobs cannot directly access any external CMS storage, due to the described network restrictions, a custom Data Transfers Service (DTS) has been designed and implemented in order to manage input and output data transfers between BSC and PIC storage. The DTS runs at PIC, copying files from BSC to PIC data storage systems (GPFS and dCache [18] respectively) and vice versa. Regarding job output data, after a payload job is completed, a job wrapper script inspects the files that were produced and injects transfer requests for them into the DTS queue. Once these transfers have been completed (it typically takes a few seconds), the wrapper sets the correct exit code for the job. This ensures that the files are properly registered at PIC storage element. To support transparent transfer of job input data files into BSC, payload jobs are instructed to consume input data from a local XRootD server, which is shipped with the singularity container where the jobs are run. This server is able to execute a configurable command for each received file request (*open* operation). This command creates a DTS transfer request for the desired file, and keep the client on hold until the file has been transferred to GPFS and is available for direct read. In both modes, DTS executes parallel scp transfers (with configurable concurrency) between BSC and PIC storage and implements sanity checks, such as size or checksum validations.

3.1 Integration of BSC resources for CMS: large scale production

Integration tests started by September 2020, and the results were reported in previous proceedings [10]. Having performed the initial functionality validation of the deployed prototype, including its integration with CMS systems, various scale tests were also performed before considering it a production-like system.

Currently, automated operations rely on the so-called vacuum operation model [19], where startd daemons are launched autonomously by the resource center (PIC in this case), rather than submitted by the experiment (CMS). Technically, a batch of scouting glideins is periodically launched, acquiring CPU from BSC, and launching a HTCondor startd process that connects to the CMS HTCondor Global Pool. If no compatible tasks are present in the CMS workload schedulers (based on a configurable HTCondor matchmaking expression), glideins expire and CPUs are returned to BSC scheduler. When suitable workload is matched, filling up the slots, the bridge node sends further resource requests to BSC. Each Slurm job is paired with a single HTCondor startd daemon at PIC, which in turn is configured to advertise the allocated resources (i.e. a full BSC compute node) as a dynamically partitionable slot. When payload jobs are matched to this resource, the partitionable slot is fragmented into multiple starter processes, each one executing a single payload job and managing the fraction of resources such job requires.

The management of the utilization of the BSC nodes has been optimized in order to minimize the amount of time the node is acquired but not executing CMS payloads. CMS pilots, with a lifetime of 48 hours, have an internal draining mechanism, that stops acquiring new payloads when the end of the pilot's lifetime approaches (currently set to 4 hours). Additionally, a mechanism to ramp-up resources has been implemented to reach the given maximum load we can use at BSC. This maximizes the use and utilization efficiency of the resources (as displayed in Figure 3). Moreover, to compensate for I/O CPU inefficiencies of the CMS payloads, the number of declared logical cores within a pilot, typically 56, is larger than the number of available cores in the node (48). This maximizes the CPU usage efficiency while keeping memory utilization on the compute nodes under control.

Given the network restrictions at BSC, the initial focus has been set on executing simulation workflows with no input data (GEN-SIM). The rest of the chain (DIGI-RECO, I/O intensive, and requiring reading input files) is then executed at PIC (or other sites remotely reading those inputs from PIC). The suitability of CMS workloads to be executed at BSC is determined according to a set of criteria, such as workflows not requiring any input data, tasks belonging to a certain campaign or study, or even containing only simulation jobs (i.e. not involving for example experimental data reprocessing tasks). Technically, such restrictions are enforced by using a similar approach to that of CMS site-customizable pilots [20].

Production CMS simulation workloads (GEN-SIM, no input data required) were run at scale and the infrastructure and services deployed were proven capable to sustain a scale of

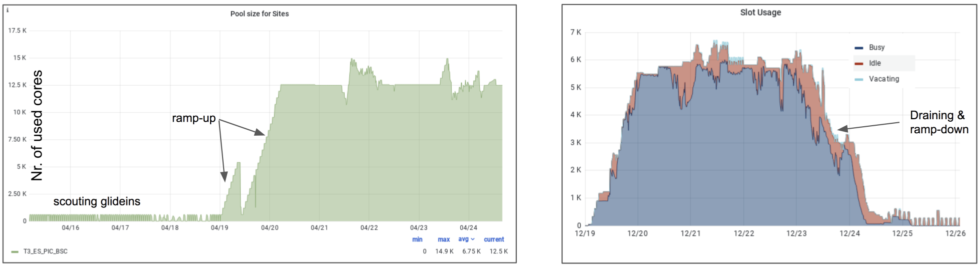

Figure 3. Ramping-up and ramp-down implementations for BSC resource utilization.

~15k CPU cores in BSC's MN4 (with 500 MB/s aggregate sustained output rate from BSC to PIC, with peaks even saturating the available 10 Gbps connectivity between PIC and BSC). The average CPU efficiency of these production tasks was higher than 90%. A single bridge node at PIC[1] is capable of handling all this load and the data transfers from BSC to PIC.

Operations have progressed successfully, making fruitful use of the resources that BSC has been providing CMS via the PIC bridge node. Running at scale, we proved the ability to consume ~5M hours/month, and we have been able to fully employ the successive CPU allocations granted to our team so far, since 2020. Automated accounting algorithms have been set to properly report the BSC usage by CMS through APEL accounting to the EGI Accounting Portal [21].

The BSC management council has granted 47.25 million CPU-core hours for CMS since 2020. Figure 4 shows the granted allocation cycles and resource usages. The periods in 2020 corresponded to the initial commissioning phase, and in 2021 the services were configured and tested at scale. During 2022 the resources have been exploited as a production service running at scale, as stated in this report. The average level of utilization by CMS of the allocated hours is very good (101%). Considering the total amount of resources that BSC has allocated to the LHC experiments participated by Spain, namely ATLAS, CMS and LHCb, CMS has consumed around 33%.

3.2 Integration of BSC resources for CMS: next steps

Along with improving monitoring, alarms and automated recovery for the services, further refinements can be applied to our setup to improve automated operations, quality and performance of the service.

An alternative mode of operation, and our ultimate goal, is to achieve a tighter integration with standard CMS operations, in which pilots are launched and managed by the GlideinWMS system, instead of running in a vacuum-like model. This will require that GlideinWMS pilot jobs, submitted centrally to claim BSC resources via PIC, are configured appropriately, so that startds employ the split-starter mode of operation, customized for the BSC case.

Additionally, the singularity container used by payloads has an xrootd-server that intercepts any xrootd data requests at runtime. This opens the door to serve input data to jobs at BSC using the DTS. This functionality has been tested, but it is not yet used in production.

It is also worth mentioning that we will soon be enabling and testing all these functionalities in the upcoming MN5 facility, a machine ~17 times bigger than MN4 (~200 petaflops), in which we also expect an enhanced network connectivity between BSC and PIC (PIC is

[1] AMD EPYC 7452 server, with 64 cores (hyper-threading enabled), 128 GB RAM, and 25 Gbps network.

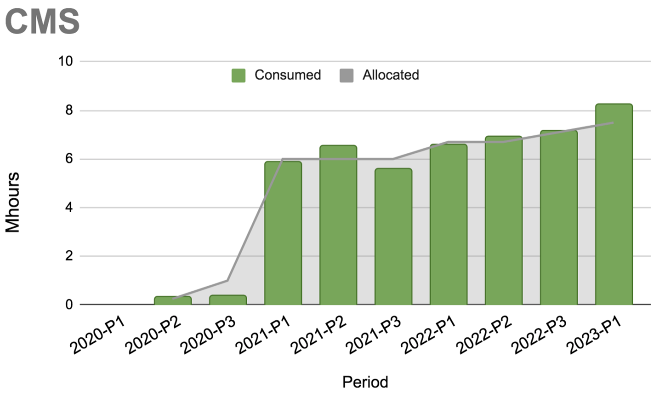

Figure 4. Allocated time every four months and the final allocation resource usage of BSC CPU resources by CMS through PIC bridge node, since 2020.

already at 200 Gbps; BSC expects 100 Gbps for MN5). This will boost the resources we can exploit and improve the performance of the jobs that are executed at BSC.

For the next years, annual increases on BSC utilization of around $15-20\%$ are expected, according to the resource needs of the experiments and the committed Spanish contribution share. Suggested improvements are being discussed with BSC management, in particular to improve the inbound/outbound connectivity of compute nodes, installation of edge services, or even deploy a close storage element, or even a cache system, in order to be aligned to other European HPC centres that are exploited for LHC computing and offer a more friendly environment.

4 Conclusions

A viable operational model for the transparent integration of the available BSC resources into the CMS workflow management system has been designed, implemented and tested, and put into production. It minimizes central operational efforts for CMS, while maximizing CPU resources utilization. This model overcomes the severe network restrictions imposed by BSC by providing an alternative way to access external services.

Big efforts have been invested in the integration of BSC CPU resources for CMS use, which involved various teams. The HTCondor team developed the split-starter mode. The CIEMAT/PIC team interfaced with CMS and BSC, deployed the PIC bridge service, enabled access to PIC Frontier service, solved the replication of CMSSW releases to BSC, developed the DTS to handle output files, and put everything in place for operations. The CMS central team developed the handling of conditions data via files in CMSSW, although this solution was not finally adopted. The CMS central team injects suitable workflows into the Global Pool so they can run at BSC, given the deployed setup.

A fully working system has been tested at large scale and it is now regularly running CMS simulation production workloads at scale. This mode of operation is transparent for workflows that separate the GEN-SIM part (executed at the BSC) and the DIGI-RECO part (executed at PIC). We can run at saturation, given the BSC maximum number of allowed Slurm jobs we can run in parallel as a user. We are not limited by our setup at the moment. These developments have placed us at the top of user's ranking at BSC and we do expect to expand to other CMS workflows in the near future.

References

[1] *Cms experiment*, https://home.cern/science/experiments/cms (2023), accessed: 2023-07-31

[2] J. Albrecht, A.A. Alves, G. Amadio, G. Andronico, N. Anh-Ky, L. Aphecetche, J. Apostolakis, M. Asai, L. Atzori, M. Babik et al., Computing and software for big science **3**, 1 (2019)

[3] *Worldwide LHC Computing Grid*, https://wlcg-public.web.cern.ch/ (2023), accessed: 2023-07-31

[4] B. Blumenfeld, et al, *CMS conditions data access using FroNTier*, in *Journal of Physics: Conference Series* (IOP Publishing, 2008), Vol. 119, p. 072007

[5] C. Aguado Sanchez, et al, *CVMFS-a file system for the CernVM virtual appliance*, in *XII Advanced Computing and Analysis Techniques in Physics Research* (2008), p. 52

[6] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, WSEAS Transactions on Computers **1**, 348 (2005)

[7] A. Pérez-Calero Yzquierdo, et al, *CMS strategy for HPC resource exploitation*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 09012

[8] D. Hufnagel, et al, *HPC resource integration into CMS Computing via HEPCloud*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 03031

[9] T. Boccali, et al, *Extension of the INFN Tier-1 on a HPC system*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 09009

[10] Acosta-Silva, et al, *Exploitation of network-segregated CPU resources in CMS*, in *EPJ Web of Conferences* (EDP Sciences, 2021), Vol. 251, p. 02020

[11] *Barcelona supercomputing center*, https://www.bsc.es/ (2023), accessed: 2023-07-31

[12] *Digital single market: Europe announces eight sites to host world-class supercomputers*, https://ec.europa.eu/commission/presscorner/detail/en/IP_19_2868 (2023), accessed: 2023-07-31

[13] F.B. Schmuck, et al, *GPFS: A Shared-Disk File System for Large Computing Clusters.*, in *FAST* (2002), Vol. 2

[14] C. Acosta-Silva, et al, *Exploiting network restricted compute resources with HTCondor: a CMS experiment experience*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 09007

[15] A.B. Yoo, et al, *Slurm: Simple linux utility for resource management*, in *Workshop on job scheduling strategies for parallel processing* (Springer, 2003), pp. 44–60

[16] J. Balcas, et al, *Using the glideinWMS system as a common resource provisioning layer in CMS*, in *Journal of Physics: Conference Series* (IOP Publishing, 2015), Vol. 664, p. 062031

[17] G.M. Kurtzer, V. Sochat, M.W. Bauer, PloS one **12**, e0177459 (2017)

[18] P. Fuhrmann, et al, *dCache, storage system for the future*, in *European Conference on Parallel Processing* (Springer, 2006), pp. 1106–1113

[19] A. McNab, F. Stagni, M. Ubeda Garcia, Journal of Physics: Conference Series **513**, 032065 (2014)

[20] A. Pérez-Calero Yzquierdo, et al, *Evolution of the CMS Global Submission Infrastructure for the HL-LHC Era*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 03016

[21] *Egi accounting portal*, https://accounting.egi.eu/ (2023), accessed: 2023-07-31