Paving the Way for HPC: An XRootD-Based Approach for Efficiency and Workflow Optimizations for HEP Jobs on HPC Centers

Robin Hofsaess[1], **Manuel Giffels**[1], **Artur Gottmann**[1], **Andreas Petzold**[1], **Günter Quast**[1], **Matthias Schnepf**[1], and **Achim Streit**[1]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany

E-mail: Robin.Hofsaess@cern.ch

Abstract.

Starting from 2025, for the fulfillment of the German WLCG compute obligations, a continuous transition from dedicated university Tier-2 centers to shares on national HPC resources is planned. This undertaking necessitates a thorough understanding of the requirements and challenges involved to ensure the efficient utilization of an HPC center as a full-fledged replacement. In this contribution, we will discuss the process of optimizing the utilization of HPC centers for HEP workflows including the essential analysis of possible limitations using the example of HoreKa, a local HPC cluster at the Karlsruhe Institute of Technology. Furthermore, we present a generalizable XRootD-based approach for data access bottleneck mitigation leading to a more efficient HEP job execution. First experiences with our deployed Proof of Concept at HoreKa are promising, demonstrating a performance on par with the German Tier-1 center, GridKa, for Monte Carlo production campaigns. Based on the results, we are convinced that the efficient utilization of HPC centers as replacements will be successful.

1 Introduction

The Worldwide LHC Computing Grid (WLCG) as the backbone of the High Energy Physics (HEP) community today still mainly consists of distributed, dedicated resources within its tiered structure, provided and maintained by the HEP community itself, e.g. at universities or research institutions. These rather homogeneous resources are tailored to the requirements of typical HEP workflows that for example often rely on a high throughput of data to process the sheer amount of data in the exabyte era of the LHC.

But even if the WLCG in its current form as world's biggest scientific computing Grid is a great success story, we are nevertheless observing the beginning of a transformation in the global computing infrastructure. Especially multipurpose High Performance Computing (HPC) centers started to play a more and more important role in recent years. While the majority of the available compute resources is still provided by the traditional grid sites, a non-negligible contribution[1] of compute power is already being added to the WLCG capabilities by HPC centers, often opportunistically integrated.

And this transformation is underlined by the future German HEP computing strategy[4]. It foresees a continuous transition away from dedicated university Tier-2 centers[2] to shares on national HPC centers

---

[1] At the time of writing (July 2024), the contribution amounts to more than 200 million core hours for CMS.

[2] Remark: The Tier-2 centers at research institutions, like e.g. DESY or GSI, are excluded from the transition.

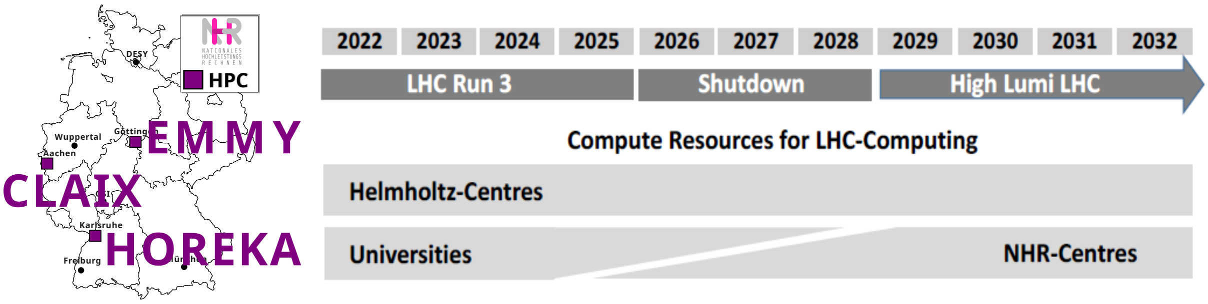

Figure 1: The future German HEP Computing strategy. Starting from 2025, the dedicated German university Tier-2 centers will continuously be replaced by shares on national HPC centers within the NHR Alliance: CLAIX (Aachen), EMMY (Göttingen), and HoreKa (Karlsruhe). *Graphic adapted*: [4]

within the NHR[3] Alliance (Nationales Hochleistungsrechnen, eng: National High Performance Computing). While the HPC centers will take over the contribution of compute power to the German WLCG pledges, the storage will be moved to the two Helmholtz Centers, the Karlsruhe Institute of Technology (KIT) and the Deutsches Elektronen-Synchrotron (DESY). The process will start next year and is due to be completed with the start of the HL-LHC, as visualized in fig. 1.

Here it is to be mentioned that the utilization of HPC centers in Germany for HEP compute jobs is not entirely new. E.g. with HoreKa, a local HPC cluster at KIT, we already integrate a HPC resource opportunistically[4] into GridKa, the German Tier-1 center, since several years. At the moment, it is used for Monte Carlo (MC) production campaigns for CMS. The main difference to the current integration, however, lies in the future fulfillment of official pledges through the mandatory provisioning of HPC resources for the WLCG. On the one hand, this implies that the HPC centers need to be capable of fully replacing the former university Tier-2 centers in their obligations in terms of provisioning compute resources. And on the other hand, a reliable and efficient operation of such resources must be guaranteed, even if the HEP community is not in charge of the resources anymore as a provider, but becomes one user among many. A fluent and successful implementation of the future HEP computing strategy therefore requires new approaches and operational concepts.

The challenges associated with this new strategy are analyzed below and our approach for a more efficient and reliable integration of HPC centers is presented using the example of HoreKa.

2 The Utilization of HPC Centers for HEP Computing

In general, as it is already shown on multiple HPC clusters worldwide, it is definitely possible to utilize them for HEP workflows. However, common physics tasks are often entirely different to typical high-performance applications, which leads to challenges that mainly originate from the conceptual design of multipurpose HPC clusters. In comparison to the traditional Grid sites, they are rather heterogeneous among each other and have a different user focus, emphasizing performance and security rather than the transfer and processing of large amounts of (remote) data. With this, they usually show their advantages in strongly parallelized high-performance tasks, like e.g. climate model calculations, and are typically not inherently suited for trivial parallelizable event-based, data intensive workflows, like common physics data analyses or MC production campaigns.

From this discrepancy, different requirements and challenges arise for an efficient integration and operation of such compute resources compared to the traditional sites, which are examined in the following.

2.1 Requirements and Considerations

Before discussing efficiency improvements for HPC centers, we must first take a look at the integration itself. Due to the fact that the operational model entirely changes with a full transition to HPC, effectively resulting in a loss of direct control over the hardware, the (security) policies of the HPC centers must be in agreement with a minimum set of requirements to allow an integration and the execution of HEP jobs at all. As an example, a cluster that is not connected to the Internet may only be integrated with

---

[3] Further info: https://www.nhr-verein.de/en

[4] The integration is realized with the CoBalD/TARDIS stack, developed at KIT. For further info, see [5] and https://cobald-tardis.readthedocs.io/

considerable effort[5] or may even be impossible to integrate, as the access to remote data is fundamental. Therefore, we are able to formulate a minimum set of requirements[6] for HEP job execution:

- **Connectivity:** As mentioned in the example above, direct access[7] to the Internet is the first prerequisite. This is not only required for data access, but also for the entire Grid job ecosystem, like job slot provisioning and glide-in via pilot mechanism, monitoring and error reporting, efficiency measurements, accounting, et cetera, which is necessary for such an ”HPC site” to be a full-fledged replacement for traditional Grid computing sites.
- **Software:** To run HEP jobs, a defined and validated software environment is necessary. In a Grid environment, this is usually ensured by container solutions, like `singularity/apptainer`, and software provisioning via `CVMFS` [2], while for HPC, the software is often provided bare metal on request or as an automated service. The validation is therefore only ensured if the required (Grid) technologies are supported.
- **Permissions:** Typically, the security measures are very high at multi-user HPC centers to ensure that users cannot get in each other’s way. This, however, can interfere with the software requirements above, if for example `usernamespaces` are not available, hindering the usage of containers. Or restricted connectivity, as e.g. firewall policies play an important role concerning incoming and outgoing network traffic of all kinds. Furthermore, it is not uncommon that monitoring and debugging capabilities are restricted, which complicates reliable operation and may require compromises and workarounds/tinkering solutions.

All of the above points need to be considered before running HEP jobs at an HPC center becomes possible.

2.2 Identification of Challenges and Limitations

The next step for an efficient utilization of HPC centers is to identify and understand challenges, bottlenecks, and limitations to find feasible solutions that guarantee a reliable and efficient operation. Since every HPC center is different, an in-depth analysis of the individual conditions is required to track down the causes of possible inefficiencies that may differ.

Already from the general requirements and considerations above, several challenges of varying severity can be derived. E.g., the provisioning of software via `CVMFS` is crucial, but it is not mandatory to be available on bare metal, since there are solutions available working in userspace, like `cvmfsexec`. Rather critical are disabled `usernamespaces`, as it prevents the usage of nested containers, making a utilization of an HPC center very complicated to impossible. Such challenges, however, can be resolved in most cases by agreements with the HPC sites, as they are a matter of policies. More complicated are infrastructure limitations, like slow WAN connections, which may lead to higher failure rates and a worse CPU efficiency of HEP jobs running at such resources. They are not as easy to resolve and require advanced solutions, since we do not have influence on the infrastructure anymore.

If now a problem is observed with jobs running at an HPC site, these considerations from the theoretical point of view are important to narrow down the possible causes. Practically, a good starting point is obviously monitoring data that may give a hint on the inefficiencies. But, since the monitoring capabilities are often limited at HPC centers, finding the actual reason for an inefficient job execution can already be challenging. For example, if a low CPU efficiency of a HEP job is observed, the first guess would be that the bandwidth could be limiting the job. But how to proof that it is really the WAN connection limiting? Maybe, it is sufficient, since HEP jobs are often streaming the necessary data in smaller packages, and the inefficiencies may emerge from something else? Furthermore, it has to be exclude that inefficiencies do not originate from inefficient workflows or bad scheduling and pilot efficiencies, which is only possible with additional monitoring data from other sites running the same

---

[5] If at least some connectivity nodes exist, proxy solutions may work or other creative approaches like at Barcelona Supercomputing center: [1]

[6] Due to the limited scope of this proceeding, other considerations like internal networks or storage space are not discussed here, but are also important points to be mentioned.

[7] In principle, also an ”offline” processing could be considered and may be possible, but this in turn would require, e.g. for a Monte Carlo simulation workflow, massive local storage capabilities in the order of PBs to make the required files locally available – which are then typically only partially read, introducing a potential huge loss in efficiency.

campaigns. And is it really the only limitation, or do we may have several issues coinciding, making it even more complicated to actually pin down the reasons for inefficiencies.

This exemplary discussion is intended to give an impression on the possible difficulties in optimizing HPC integration. Due to the site restrictions and limited monitoring capabilities, it can be difficult to even pin down the reasons behind inefficiencies. And since the efficient integration is mandatory, also in terms of sustainability, the importance of fully understanding the underlying constraints and challenges as part of the optimization is not to be underestimated.

To conclude, several questions need to be answered before thinking of possible steps to enhance the integration in terms of reliability and efficiency.

2.3 Analyzing Inefficiencies at HoreKa

For HoreKa, as a real-world example, we started analyzing the setup for identifying potential bottlenecks as we observed a lower CPU efficiency and an increased failure rate, especially for data intensive workflows, in a direct comparison to GridKa. The general configuration of HoreKa is depicted and described in fig. 2.

From the theoretical perspective, the external connectivity of the worker nodes with 1G is clearly the main bottleneck when analyzing the overall setup. But practically, this is not trivial to proof for such an HPC center, as described above, since the monitoring capabilities are restricted and do not allow in-depth I/O monitoring per job or node for tracing back the problem to a possible saturation of the external link. In addition, HoreKa as an opportunistic resource within the WLCG is already running CMS production jobs, which means that no direct comparison on job level is possible, since running the same job two times on different centers, as necessary for a real ”benchmarking scenario”, is not intended and would require a lot of development work on collaboration level. As a consequence, it is only possible to compare overall tendencies in terms of e.g. CPU efficiency or failure rate per campaign.

When comparing the distributions of the CPU efficiency and the I/O wait times based on available CMS monitoring data for jobs of campaigns simultaneously running at HoreKa and GridKa, a clear shift to lower efficiencies for jobs running at HoreKa is recognizable, especially if they are data-intensive. Ultimately, we still have no real proof for the above assumption, but the clear indications have prompted us to use this as the starting point for optimizations of the HPC integration. Additionally, our investigation revealed non-ideal configurations that were mainly contributing to the increased failure rate. While those issues were rather easy to resolve leading to a significant reduction of job failures, the improvement of the CPU efficiency is way more complicated and will be addressed in the next section.

3 Data Access Bottleneck Mitigation with XRootD

Resolving the issue of a too slow connection is complicated as we cannot just increase the available external bandwidth of the worker nodes. Luckily, in case of HoreKa, the login nodes provide a faster connectivity with 50G each, as indicated on the left in fig. 2. From this, we started developing an XRootD-based [3] approach for mitigating the data access bottleneck by leveraging the faster bandwidth of the login nodes. XRootD is a Grid software framework and transfer protocol which is one of the default Grid tools for data provisioning and transfers within the WLCG. Fortunately, it provides additional server-sided functionalities that allow proxying and caching for data transfers. Our concept for a more efficient data access combines these two features. By deploying a fully containerized XRootD caching proxy on a faster externally connected login node, we are able to reroute the incoming data traffic of the jobs running on the worker nodes, de facto increasing the available bandwidth significantly. Additionally, transferred data can be cached on the RDMA-enabled parallel filesystem. This adds two more benefits to the setup. First, a prefetching mechanism is available that allows to fetch remote data blocks in advance. They are stored in the cache and can be accessed by the jobs via the internal connection, essentially transforming the XRootD caching proxy into a fast buffer for the requested data, resulting in optimized data transfers and therefore an overall increased CPU efficiency. And second, cache hits increase the transfer speed extremely, since the transfers are realized over the 200G internal InfiniBand, or even directly via RDMA, if a file is cached entirely, so-called *direct cache access*.

The full concept is exemplary shown in fig. 2 and reads like this: A HEP job starts running on a worker node and requires remote data. But instead of pulling the data directly over the 1G bandwidth from remote **(1)**, it redirects the request for data over the internal InfiniBand network to the XRootD caching proxy on the login node **(2)**. The proxy then takes over the remote data transfer by utilizing the 50G external bandwidth **(3)**. The transferred data is cached on the fly on the parallel filesystem **(4)** and the job receives the data utilizing the 200G InfiniBand or via direct cache access (RDMA), if the file is fully cached **(5)**, resulting in a greatly improved data access. With prefetching enabled, the setup

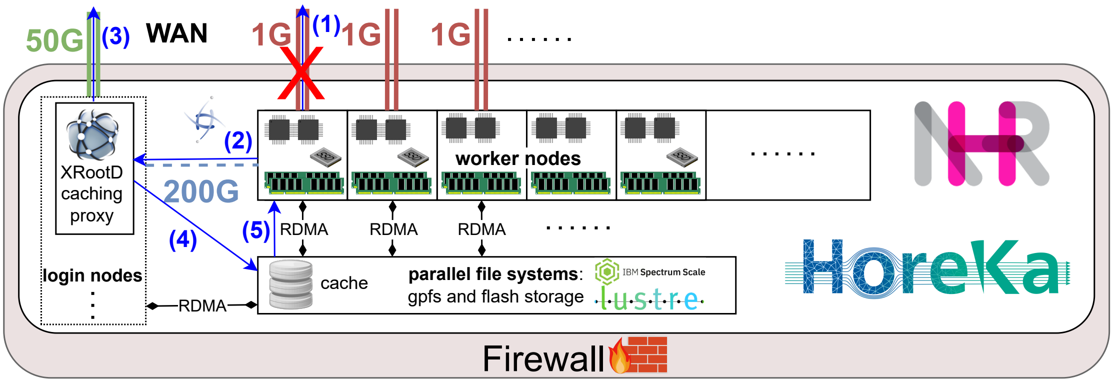

Figure 2: The current setup of the HoreKa HPC cluster. Each HoreKa CPU worker node (WN) provides twelve 8-core slots. Internet access is available with 1G for each WN, while the login nodes have a WAN connection with 50G each. All external traffic is routed through a firewall. Internally, each worker is interconnected with the login nodes via fast InfiniBand (200G). In addition to on-node flash storage, multiple parallel filesystems (gpfs, lustre) are available and accessible from the worker and login nodes via RDMA.
The approach for a mitigation of the data access bottleneck is indicated by the blue arrows. By rerouting the data over a faster externally connected login node and utilization of the parallel filesystem as a cache, we are able to bypass the limitations and optimize the data access for HEP jobs significantly.

therefore turns into a kind of ”caching buffer” for the HEP jobs. When the job has finished, the stage-out of the results then currently happens over the 1G connection of each node for technical reasons[8].

Results The first results of our Proof of Concept deployed at HoreKa are promising and indeed confirmed our assumptions of the external bandwidth as the cause for the inefficiencies, described in the section before. And the identification of configuration problems due to the in-depth analysis of HoreKa in preparation for our PoC lead to an overall more stable and efficient operation. Already with a rather default setup of the *XRootD buffer* without in-depth optimizations on XRootD level we were able to achieve a performance of CMS MC production jobs comparable to GridKa in terms of CPU efficiency and job failure rates. For the full HEP job mix required for a total replacement of a former Tier-2 center, however, further validation studies are necessary that will follow in the upcoming months.

4 Conclusion and Outlook

The transition from traditional Grid computing resources to more versatile HPC centers marks a significant change in the HEP computing environment. The integration challenges we discussed, particularly using the example of HoreKa, illustrate the complexity of optimization approaches as a basis for a more efficient, reliable and sustainable use of HPC centers for the HEP community. Limitations and challenges identified within this contribution underline that the efficient integration can be complicated. But with our PoC that tends to be as general as possible, we have shown that an increase in efficiency can be achieved leading to a comparable performance of our HPC resources to that of traditional Grid sites. Specifically for MC production jobs running at HoreKa, we were able to reach an overall similar CPU efficiency with our combined approach of prefetching (*buffer*) and caching of data over a faster connected login node of the HPC center. On top of that, the PoC provides additional monitoring capabilities that are very valuable for an efficient operation as a Grid site.

In addition, our PoC still offers opportunities for further improvement. While the rather default XRootD configuration currently deployed already works well, with fine-tuning of the XRootD caching proxy even better results can be expected in the future. Horizontal scaling as an extension of the setup is also conceivable. Utilizing several login nodes as a cluster of buffers connected via a local XRootD

---

[8] XRootD does currently not include a mechanism for credential forwarding with X509 proxies. This may change in the future with tokens.

redirector for load balancing may even make more bandwidth per job available. And ultimately, in case of HoreKa, moving the PoC to dedicated transfer and caching nodes located at the close-by Tier-1 Grid center, GridKa, with a direct connection to the HoreKa worker nodes via fast InfiniBand eventually allowing the jobs to access data via LHCOne over GridKa would be a great option for the future and could work as a prototype for other centers.

In conclusion, based on our results, we are convinced that the envisaged transition to HPC resources will be successful.

References

[1] C Acosta-Silva, A Delgado Peris, J Flix Molina, JM Hernández, A Pérez-Calero Yzquierdo, E Pineda Sánchez, and I Villalonga Domínguez. Integration of the Barcelona Supercomputing Center for CMS computing: Towards large scale production. In *EPJ Web of Conferences*, volume 295, page 07027. EDP Sciences, 2024.

[2] Predrag Buncic, C Aguado Sanchez, Jakob Blomer, Leandro Franco, Artem Harutyunian, Pere Mato, and Yushu Yao. Cernvm–a virtual software appliance for lhc applications. In *Journal of Physics: Conference Series*, volume 219, page 042003. IOP Publishing, 2010.

[3] Alvise Dorigo, Peter Elmer, Fabrizio Furano, and Andrew Hanushevsky. XROOTD-A Highly scalable architecture for data access. *WSEAS Transactions on Computers*, 1(4.3):348–353, 2005.

[4] Komitee für Elementarteilchenphysik. Perspektivpapier der Teilchenphysiker:innen in Deutschland zur Bereitstellung der Computing-Ressourcen in Deutschland für Speicherung und Auswertung der Daten des Large Hadron Colliders. Paper published on the website of the project management organization, 2022. `https://www.ketweb.de/sites/site_ketweb/content/e199639/e312771/KET-Computing-Strategie-HL-LHC-final.pdf`. Accessed: 2024-07-04.

[5] RF von Cube, M Fischer, M Giffels, A Jung, T Kress, A Nowack, G Quast, A Schmidt, and M Schnepf. Transparent expansion of a WLCG compute site using HPC resources. 2022. ACAT Proceeding, Submitted.