dCache project status and update

Tigran Mkrtchyan[1,*], *Christopher* Green[2], *Dmitry* Litvintsev[2], *Lea* Morschel[1], *Marina* Sahakyan[1], and *Svenja* Meyer[1]

[1]Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany
[2]Fermilab, PO Box 500, Batavia IL 60510-5011, USA

Abstract. The dCache project delivers an open-source, massively scalable, distributed storage system deployed internationally to satisfy today's scientists' ever-demanding storage requirements. Its multifaceted approach supports different use cases with the same storage, from high throughput data ingest, data sharing over wide area networks, efficient access from HPC clusters, and long-term data persistence on tertiary storage. Even though dCache was initially developed for HEP experiments, today, it is used by various scientific communities, including astrophysics, biomed, and life science, each with their specific requirements. To match the needs of these new communities and keep up with the scaling demands of existing experiments, dCache is permanently evolving. With this contribution, we would like to highlight the recent developments in dCache regarding integration with CERN Tape Archive (CTA), advanced metadata handling, token-based authorization support, bulk API for QoS transitions, REST API to control interaction with the tape system, and future development directions.

1 Introduction

The dCache project started in 2000 as a collaboration between Deutsches Elektron-Synchrotron (DESY) and Fermi National Accelerator Laboratory. The mission back then was to develop a common storage software that combined commodity heterogeneous disk servers as a caching layer in front of tape storage. The dCache software has proved popular within WLCG. Various laboratories and universities have deployed dCache. Combined, they provide some 50% of the overall WLCG storage capacity outside of CERN[1]. Today, dCache has been used in production for over twenty years and is deployed throughout the world [1]. Increasingly, sites use dCache to support communities with needs beyond those of LHC experiments. For example, DESY facilities now serve photon sciences, biology, future accelerators, R&D, and more. From its earliest versions, dCache has addressed the challenges of new user communities and evolving workflows by developing and adopting innovative solutions to boost productivity [2]. Despite its maturity, dCache continues to adapt to changing technologies and user demands. Beyond distributed storage, it offers access and authentication protocols, supports third-party copy for inter-site data movement, and provides standard

[*] e-mail: tigran.mkrtchyan@desy.de
[1] This number is calculated based on the information provided by the WLCG monitoring system.

and HEP-specific protocols to meet scientific needs. dCache runs in heterogeneous environments, giving sites flexibility in hardware and OS choices, and its scalable architecture supports deployments ranging from a single node to hundreds, allowing seamless growth.

This paper will discuss the technical details of dCache recent developments.

2 Namespace scalability

The dCache architecture separates storage for data from storage for file metadata[3]. Its metadata service is implemented on top of a PostgreSQL database. Although the file metadata usually occupies only a tiny fraction of storage systems, for example, the metadata for the EuXFEL dCache instance is only $\approx$ 200GB for $\approx$ 140PB of stored experiment data, i.e., 0.000001%, the metadata operation latency plays a significant role in the overall distributed storage system performance. With a growing number of data-ingest and data-processing CPUs in modern scientific environments, the primary challenge for metadata servers arises from semantics required by POSIX specifications for *create*, *link*, *mkdir*, *rename* and *unlink* operations. The POSIX semantics[4] assume linearizable consistency, where filesystem updates become visible to all readers at the same point in time without synchronization operation[5]. However, not all workloads require strong POSIX semantics. Starting version 9.2, dCache provides a tunable consistency mechanism that improves the filesystem object creation rate by relaxing POSIX constraints and introducing eventual consistency for the parent directory attributes. Each metadata-aware component, such as an NFS server (NFS door) or a metadata server (PnfsManager), can be configured depending on the application's needs to provide different consistency guarantees. Three consistency levels are available: strong, weak, and soft. The consistency level can be selected based on the POSIX compliance required for a given use case. For example, the applications that only create data can use weak consistency, whereas others can fall back to soft or strong consistency. The test workloads have demonstrated an increase of up to x200 throughput of weak over strong consistency. Such a technique demonstrates that using commodity hardware and general-purpose databases can provide an HPC-scale distributed storage system comparable with metadata and I/O with high-end cluster filesystems.

3 Analysis Facility support and POSIX access

One issue when dealing with large volumes of data is how to provide efficient data access. Many standard protocols lack the features to benefit from distributed data, which is common in multi-petabyte storage systems. Version 4.0 and earlier of the NFS protocol are examples of such behavior: all data access goes through a single node, limiting overall performance. However, with the introduction of the pNFS extension in NFS v4.1[6], the NFS protocol has become a practical way of accessing large-scale storage[7]. By separating metadata and the data access paths, clients can talk directly to the data servers. This allows distributed storage systems to grow in throughput and capacity by increasing the number of data servers. To the best of our knowledge, NFSv4.1/pNFS is the only open standard to access distributed storage. Figure 1 demonstrates pNFS-enabled distributed server architecture.

For this reason, dCache supports NFS, with an emphasis on pNFS[8]. This allows the storage system to be mounted using standard clients (such as the Linux kernel) without needing any driver or application changes.

The analysis framework ROOT[9] developed and maintained by CERN supports various network protocols with the ability to add more. This allowed using ROOT with dCache via HEP proprietary protocol, like DCAP or XRootD. With a growing demand for non-HEP

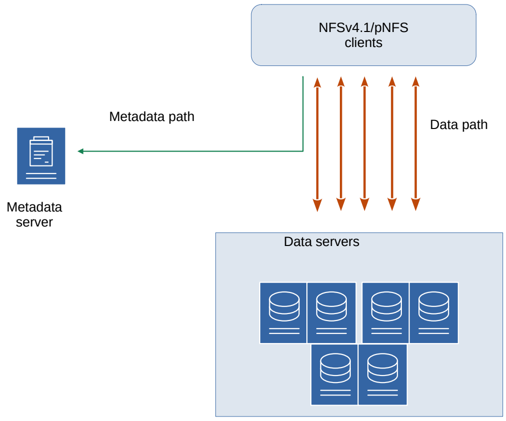


Figure 1: NFSv4.1/pNFS distributed architecture. The storage bandwidth grows with the number of data servers.

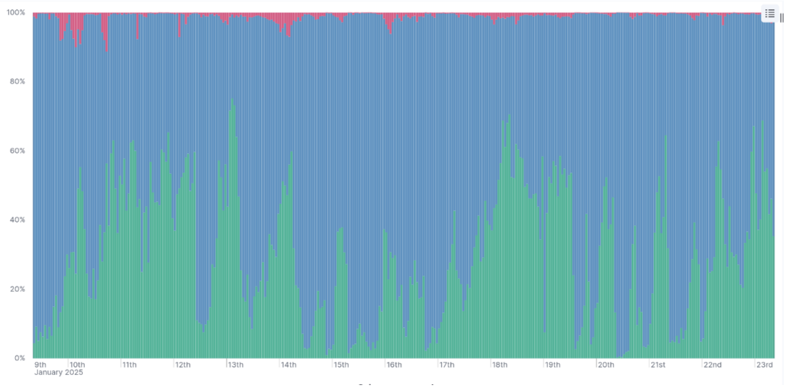


Figure 2: Local access protocol distribution percentage.

software, like Jupyter Notebooks and Apache Spark, POSIX-like data access is expected. In addition, with the availability of opportunistic HPC resources to HEP experiments, the provided POSIX interface must not require any special software installed on the worker nodes. The NFS-mounted dCache allows such communities to use standard Linux machines to access data stored in dCache without adapting their analysis software, which is not always possible. Moreover, with recent developments in the dCache inter-component communication protocol[10], the overhead of the internal communication is reduced, which improves the efficiency of HPC jobs, where file metadata access is as essential as the data access itself.

Figure 2 demonstrates HTCondor data access protocol distribution at the National Analysis Facility (NAF) at DESY. Most NAF local accesses are dominated by NFSV4.1/pNFS, shown in blue, while grid jobs often utilize xroot protocol[11], shown in green. The rise of xroot-based accesses over weekends demonstrates the interactive nature of NAF, where grid jobs use idle nodes as an opportunistic resource[12].

Another case of non-ROOT-based analysis is using commercial applications, such as MATLAB, under Microsoft Windows OS. To support MS Windows users to store and access data in dCache, the SMB protocol has been added. Rather than implementing the SMB protocol support directly in dCache, a protocol translation server runs the open source SAMBA[13] software while providing access to dCache storage via NFS.

In close collaboration with Linux kernel developers, dCache is an early adopter and demonstrator of the evolution of the NFS protocol and gives Early Access to new developments, like extended file attributes[14] over NFS or NFS-over-TLS[15].

4 Integration with CERN Tape Archive

Even though large hard disk-based storage systems cost, space, and volume-effective today, magnetic tapes are still the best (and cheapest) option for long-term data archival, especially for so-called cold data, the data that is rarely accessed.

dCache has a flexible tape interface that allows for connectivity with any tape system. There are two ways that a file can be migrated to tape. Either dCache calls a tape system-specific copy command or through interaction via an in-dCache tape system-specific driver called a nearline storage provider. The latter has been shown (by NDGF, TRIUMF, and KIT Tier-1s) to provide better resource utilization and efficiency[16].

The CERN Tape Archive (CTA)[17] is an open-source storage system developed by the CERN IT storage group to replace the legacy CASTOR system used to manage experiment data on tape. Its architecture is designed to meet the requirements of LHC Run 3 as well as HL-LHC, thus matching the most data-intensive scientific workloads. In addition, the CTA project is actively pursuing the system flexibility to allow wider adoption by other sites, ease of contributions from other developers, and elimination of CERN-specific dependencies in the provided binary packages[18].

Out of the box, CTA comes with a frontend that communicates with EOS, the disk system deployed at CERN. However, as CTA's queuing system is not EOS aware, other frontend implementations are possible. For seamless integration of CTA, the dCache developers at DESY implemented a CTA-specific nearline storage provider called dcache-cta[19, 20], and a corresponding CTA frontend component. The communication between dCache and the new frontend is based on Google's gRPC library and not limited to dCache, therefore, can be used by other disk systems used by CTA. The dCache-CTA integration is demonstrated in Figure 3.

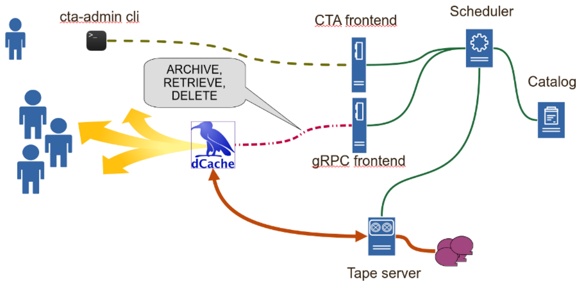


Figure 3: dCache integration with CTA.

Since the Summer of 2024, dCache with CTA has been the only system used by DESY to store experimental data, demonstrating transfer rates that are close to maximum throughput allowed by tape drive specification.

5 Data Labeling

Traditional file systems organize data in a hierarchy of directories[21]. The directory structure is built as a hierarchy of containers that groups files and directories by a logical unit, e.g., experiment run number, beam time, data type, etc. The data processing application can then be pointed to such a directory to process all data in that directory. However, sometimes only a subset of data files is required based on one specific attribute. Recent changes in dCache have introduced dynamic grouping of files into *virtual directories* based on a user-provided grouping key - a file label. These labels can be added, removed, and queried via the dCache REST API. A single file can have multiple labels and thus exist in multiple virtual directories simultaneously. To ensure seamless integration with existing workflows, the virtual directories are exposed to the user as regular read-only directories and are hence accessible via all supported protocols, such as NFS and WebDav. Future developments aim to integrate end-user-provided metadata extraction and automate data labeling.

6 Quality of Storage Service

Modern experiments like those at European XFEL[2] define a strict data policy plan that ensures that the scientific data are Findable, Accessible, Interoperable, and Reusable (FAIR)[22]. An example of such a data management plan is demonstrated in Figure 4[3].

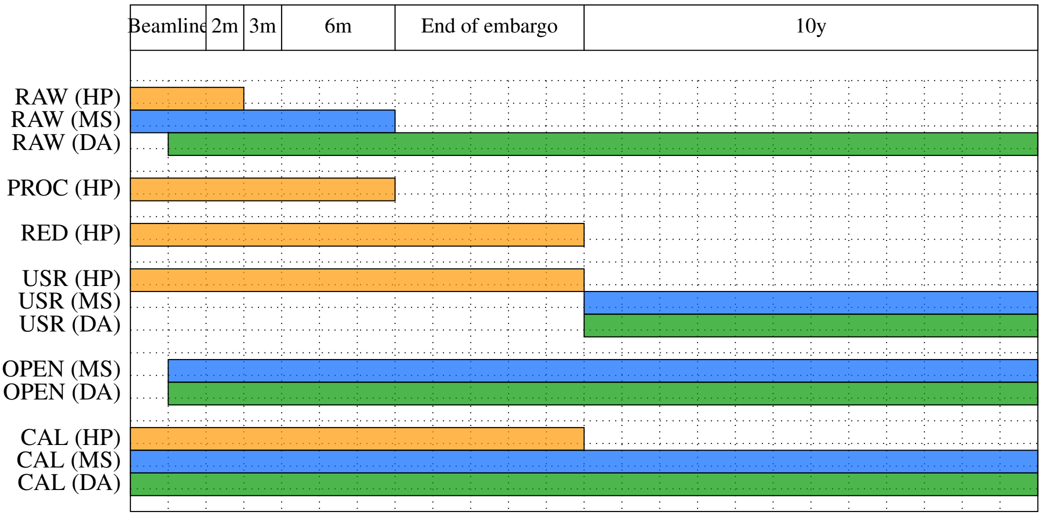


Figure 4: Retention periods of scientific data across storage classes and storage systems at European XFEL. QoS classes: HP - High Performance Storage; MS - Large-volume Mass Storage; DA - Deep Archive (tape). Data types: RAW - experiment raw data; PROC - processed data; USR - user analysis data; OPEN - open data; CAL - calibration data.

To match with defined policy and automate data movement within the system, dCache has *Quality of Service* module that allows defining multiple data locations and transitions between them. An example of such a policy definition is demonstrated in Listing 1. The

[2] https://xfel.eu
[3] Source: https://www.xfel.eu/users/policies/index_eng.html

policy specifies *xfel-raw-policy* that it is required to stay six months on disk with a tape copy, and then only tape copies should be maintained.

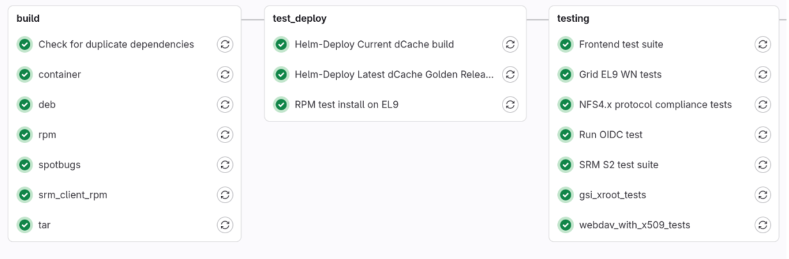
Listing 1: Example of XFEL Raw data QoS lifecycle.

```
"name": "xfel-raw-policy",
  "states": [
    {
      "duration": "P6M",
      "media": 1x DISK, 1x HSM
    },
    {
      "media": 2x HSM
    }
  ]
```

7 Development process and quality assurance

From the outset, the dCache has been a distributed software project with developers at DESY, Fermilab, and, later, from NeIC[4]. Being a critical part of site infrastructure, the dCache project has stringent requirements on software quality and build reproducibility. From early on, the dCache project has adopted semantic versioning[5] and time-based release policy[23] as a strategy to make software packages available to the sites. With the growing number of software branches to support and the increasing complexity of testing, the manual build and test steps were replaced with continuous integration (CI) systems that trigger automatic build and test procedures on code changes.

The pipeline stages are logically divided into three major groups with the following responsibilities:



build
- Check for duplicate dependencies
- container
- deb
- rpm
- spotbugs
- srm_client_rpm
- tar

test_deploy
- Helm-Deploy Current dCache build
- Helm-Deploy Latest dCache Golden Relea...
- RPM test install on EL9

testing
- Frontend test suite
- Grid EL9 WN tests
- NFS4.x protocol compliance tests
- Run OIDC test
- SRM S2 test suite
- gsi_xroot_tests
- webdav_with_x509_tests

Figure 5: A reduced dCache CI pipeline in GitLab. The auxiliary stages, like Kubernetes namespace creation or log collection, are not shown.

The pipeline relies on Linux containers for dCache test deployment and the required infrastructure for integration and user tests. The Kubernetes service provided by the DESY-IT System group is utilized for container orchestration. All container-based applications and services are deployed using Helm charts. The dCache-specific Helm charts[24] mimic a multi-node setup. Multiple versions of dCache components can be deployed to test backward compatibility and interoperability. The complete build, test, and publish pipeline comprises

[4] Nordic e-Infrastructure Collaboration
[5] https://semver.org/spec/v2.0.0.html

over thirty jobs covering the entire development life-cycle. A reduced version of the pipeline is shown in Figure 5[6]. The pipelines triggered by the release process additionally publish signed packages in the *dcache.org* download area. For each pipeline, a new Kubernetes namespace is created, which allows more optimal resource isolation, such as avoidance of running memory-intensive dCache processes with build jobs from other pipelines. Before the namespace is destroyed, the logs from all containers are collected for debugging purposes.

Although dCache developers use the GitLab service at DESY for their CI, the main code repository remains in GitHub. This means that all code changes are committed and pushed to GitHub. A dedicated GitHub action[25] is used to synchronize code changes with GitLab at DESY and trigger the CI pipeline.

8 Summary and Future Work

The dCache project continues evolving as a robust and scalable open-source storage solution, meeting scientific communities' diverse and growing demands. This paper highlights recent developments in namespace scalability, enabling high-performance metadata operations while maintaining flexibility in consistency models. The improved support for POSIX access via NFSv4.1/pNFS enhances interactive data access, such as at the National Analysis Facility at DESY. The seamless integration with the CERN Tape Archive (CTA) establishes dCache as a viable long-term storage solution for experimental data, ensuring high-throughput archival capabilities. The Quality of Storage Service (QoS) service enables automated and policy-driven data lifecycle management, helping sites to comply with FAIR data principles. Finally, the adoption of modern DevOps practices ensures high-quality software releases. Future work will focus on further scalability and enhanced support for HPC-like workloads.

References

[1] P. Fuhrmann, V.Gülzow, dCache, storage system for the future, pp. 1106–1113, (2006). 10.1007/11823285

[2] A. Millar, T. Baranova, G. Behrmann, C. Bernardt, P. Fuhrmann, D. Litvintsev, T. Mkrtchyan, A. Petersen, A. Rossi, K. Schwank, dCache, agile adoption of storage technology, Journal of Physics: Conference Series **396**, 32077–087, (2012). 10.1088/1742-6596/396/3/032077

[3] T. Mkrtchyan, K. Chitrapu, D. Litvintsev, S. Meyer, A.P. Millar, L. Morschel, A. Rossi, M. Sahakyan, DB Back-ended Filesystem for Science, in *Proceedings of the 35th GI-Workshop Grundlagen von Datenbanken, Herdecke, Germany, May 22-24, 2024*, edited by U. Störl (CEUR-WS.org, 2024), Vol. 3710 of *CEUR Workshop Proceedings*, pp. 58–63, https://ceur-ws.org/Vol-3710/paper9.pdf

[4] IEEE Standard for Information Technology–Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7, IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008) pp. 1–3951 (2018). 10.1109/IEEESTD.2018.8277153

[5] G. Ntzik, P. da Rocha Pinto, J. Sutherland, P. Gardner, A Concurrent Specification of POSIX File Systems, in *32nd European Conference on Object-Oriented Programming (ECOOP 2018)*, edited by T. Millstein (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2018), Vol. 109 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 4:1–4:28, ISBN 978-3-95977-079-8, ISSN 1868-8969, https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ECOOP.2018.4

[6] The up-to-date pipeline version is available at https://gitlab.desy.de/dcache/dcache/-/pipelines/latest

[6] S. Shepler, M. Eisler, D.B. Noveck, Network file system (NFS) version 4 minor version 1 protocol, RFC **5661**, 1 (2010). 10.17487/RFC5661

[7] D. Hildebrand, P. Honeyman, Exporting Storage Systems in a Scalable Manner with pNFS, in *22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 2005), Information Retrieval from Very Large Storage Systems, CD-ROM, 11-14 April 2005, Monterey, CA, USA* (IEEE Computer Society, 2005), pp. 18–27, https://doi.org/10.1109/MSST.2005.14

[8] J. Elmsheuser, P. Fuhrmann, Y. Kemp, T. Mkrtchyan, D. Ozerov, H. Stadie, LHC data analysis using NFSv4.1 (pNFS): A detailed evaluation, Journal of Physics: Conference Series **331**, 052010, (2011). 10.1088/1742-6596/331/5/052010

[9] R. Brun, F. Rademakers, P. Canal, A. Naumann, O. Couet, L. Moneta, V. Vassilev, S. Linev, D. Piparo, G. GANIS et al., root-project/root: v6.18/02 (2020), https://doi.org/10.5281/zenodo.3895860

[10] L. Morschel, O. Adeyemi, V. Garonne, D. Litvintsev, P. Millar, T. Mkrtchyan, A. Rossi, M. Sahakyan, J. Starek, S. Yasar, dcache – efficient message encoding for inter-service communication in dcache: Evaluation of existing serialization protocols as a replacement for java object serialization, EPJ Web Conf. **245**, 05017 (2020). 10.1051/epjconf/202024505017

[11] XrootD, The XRootD software framework, https://xrootd.org/ (2024)

[12] B. Christoph, T. Finnern, M. Flemming, A. Gellrich, T. Hartmann, Y. Kemp, B. Lewendel, J. Reppin, K. Sever, S. Sternberger et al., Consolidating the interactive analysis and grid infrastructure at desy, EPJ Web Conf. **245**, 07003 (2020). 10.1051/epjconf/202024507003

[13] Samba Contributors, samba, online; last access on 07.01.2025, https://samba.org

[14] M. Naik, M. Eshel, File system extended attributes in nfsv4, RFC **8276**, 1 (2017). 10.17487/RFC8276

[15] T. Myklebust, C. Lever, Towards remote procedure call encryption by default, RFC **9289**, 1 (2022). 10.17487/RFC9289

[16] Musheghyan, Haykuhi, Petzold, Andreas, Heiss, Andreas, Ressmann, Doris, Beitzinger, Martin, The gridka tape storage: various performance test results and current improvements, EPJ Web Conf. **245**, 04026 (2020). 10.1051/epjconf/202024504026

[17] M. Davis, V. Bahýl, G. Cancio, E. Cano, J. Leduc, S. Murray, Cern tape archive — from development to production deployment, EPJ Web of Conferences **214**, 04015 (2019). 10.1051/epjconf/201921404015

[18] M. Davis, J. Afonso, R. Bachmann, V. Bahýl, J. Vera, J. Leduc, P. Cortés, F. Rademakers, L. Wardenær, V. Yurchenko, The cern tape archive beyond cern an open source data archival system for hep, EPJ Web of Conferences **295** (2024). 10.1051/epjconf/202429501048

[19] T. Mkrtchyan, J. Chodak, M. Karimi, R. Lueken, S. Meyer, P. Suchowski, C. Voss, dcache integration with cern tape archive, EPJ Web of Conferences **295** (2024). 10.1051/epjconf/202429501016

[20] The dCache Collaboration, dcache/dcache-cta: v0.14.0 (2024), https://doi.org/10.5281/zenodo.14421668

[21] Tanenbaum, Andrew S., Modern operating systems, 3. ed. edn. (Pearson Prentice Hall, Upper Saddle River, NJ, 2008), ISBN 0136006639; 9780136006633

[22] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.W. Boiten, L.B. da Silva Santos, P.E. Bourne et al., The fair guiding principles for scientific data management and stewardship, Scientific Data **3**, 160018

(2016). 10.1038/sdata.2016.18

[23] E.S. Raymond, The cathedral and the bazaar, First Monday **3** (1998). 10.5210/FM.V3I2.578

[24] dCache developers, dCache Helm cart (2025), `https://github.com/dCache/dcache-helm`

[25] GitHub, Inc., GitHub Actions documentation (2025), `https://docs.github.com/en/actions`