dCache

A flexible, free petascale storage system

Jürgen Starek
On behalf of the dCache development team
2019

Driven by the extreme demands on storage systems that exist within the high energy physics domain, dCache evolved as one of the leading storage systems for scientific data. It has been used to store many hundreds of Petabytes of physics data across the globe and proved its reliability over almost two decades of operation. While providing many advanced features for scientific data management and being able to deliver outstanding performance, dCache is still flexible enough to be configured as a general-purpose cloud storage system and to be run on cost-effective off-the-shelf hardware.

dCache.org

Overview

dCache is a storage system for extremely large amounts of data. It was originally developed in the context of scientific data management and has been shown to scale far into the Petabyte range — for example, the data catalogue of the LOFAR experiment, widely recognized as one of the largest astronomical data collections, is stored on a (rapidly growing) 40 PB dCache instance, and there are various petascale dCache instances used in the particle physics community as well.

dCache unifies the storage space from many, potentially quite heterogeneous, storage servers into a single virtual file system. It takes care of data hot spots and failing hardware and can ensure that at least a minimum number of copies of each dataset reside within the system for resilience or performance optimization. Furthermore, dCache supports a diverse set of access protocols and authentication and authorization methods.

dCache is Free Software, available under a permissive AGPL license without licensing costs, and has been under active development for almost two decades. Its developers have close ties to the research community and understand the needs of scientific data management firsthand.

Technical Aspects of the System

Conceptually, dCache is a distributed Enterprise Java application targeting Linux platforms. It uses a microservice-style architecture that allows flexible, scalable distribution of system components across various physical machines.
For small-scale deployments or evaluation purposes, a Java Runtime Environment (JRE) and a database like PostgreSQL are the only system requirements.
For production setups, some additional software components will be beneficial. Providing an external Apache Zookeeper cluster will increase throughput and resilience of dCache. A load balancing HTTP proxy server can help distribute large numbers of requests. Finally, a powerful monitoring system that can process classical log files as well as Kafka and / or SSE events should be used to provide operational insights.

Virtual File System Tree

dCache strictly separates filenames from the actual physical location of the files it stores. This has the advantage that a namespace entry that looks to users like a single file can actually have several physical instances on multiple storage media or in several data centres around the world. Having one filename point to various physical locations can help increase resilience against failures and increase performance by routing many parallel requests for the same file to different physical hard disks.

Access Protocol Support

Users can access the data stored in a dCache instance through a variety of protocols. dCache's *doors* provide a pluggable mechanism to add new protocols to an existing instance, so that a single dCache instance can look like an FTP, WebDAV and xrootd server hosting the same data at the same time.

POSIX-style data access is available via dCache's NFS door. The NFS 4.1 server implementation in dCache is one of the most modern NFS implementations on the market, enabling scalable deployments that automatically distribute traffic across as many server endpoints as provided by the administrator.

Supported access protocols

General purpose	Scientific use cases
FTP	GridFTP
WebDAV	xrootd
NFS	DCAP
	SRM

Authentication and Authorization

Administrators can mix various sources of identity information and authorization (AuthN) methods in one dCache instance. dCache's AuthN system is called *gPlazma*, and it is flexible enough to support use cases ranging from a single user (e.g. a functional account used for a data catalogue's web application) to thousands of users from widely dispersed institutes with potentially non-overlapping trust structures.

Supported authN methods

General purpose	Scientific use cases
Username / Password	X.509
LDAP	SciToken
OpenID Connect	
Kerberos	
Macaroons	

Operations on the file system can be controlled using the standard Unix permission model. For more complex needs, dCache additionally supports NFS 4-style Access Control Lists (ACLs). Through ACLs, administrators can specify more complex permission sets thatn would be possible through Unix permissions and create very detailed permission models both for existing files and, through inheritance, for files yet to be written.

High Availability

dCache services support replication within an instance. A central Apache Zookeeper cluster keeps track of the services' state and allows continued operation in case of individual services failing. This architecture also allows server-by-server, "rolling" updates of a system without any downtimes.

Administrative Interfaces and Scriptability

In order to facilitate administering large clusters, dCache has sophisticated interfaces for scripting and interactive control.

- *Administrative actions* on dCache services are generally performed through an SSH-based interactive shell. Additionally, a RESTful API, well-suited for use in scripts, offers a way to query and change system state.
- *File lifecycle information* is available through a Kafka message bus interface as well as a Server-Sent Events (SSE) interface offering inotify semantics.
- Even scripts that can only access an NFS mounted filespace can *query and change some file statuses* by using a special syntax for filenames that represent commands, e.g. `cat “.(get)(file.dat)(checksums)”`.

Data Management and Lifecycle Features

Structure of the Virtual Storage Space

dCache does not simply consider the storage on its servers a „flat set of hard drives“. Instead, it offers several logical structures to help administrators to subdivide the unified storage space and to manage data flows within it:

- *Pools* form the basic logical unit for storage within dCache. A pool is comparable with a disk partition in that several of them can be used to subdivide a server's RAID, but pools are purely logical constructs that dCache creates on the file system level. They are generally several dozens of Gigabytes in size.
- *Pool Groups* aggregate pools into logical groups for management purposes.
- *Links* can route traffic to pools, according to filter criteria for protocols, netmasks, pool preferences etc.

Support for Tertiary Storage Systems

dCache data pools can be configured to move data from disk to one or more attached tertiary storage systems, e.g. tape libraries. Generally, files that are available on tape are removed from disk as soon as disk space runs low on a 'least recently used' basis.

If a file that is not available from disk is requested, the data is fetched from the tertiary storage system to a dCache disk, from where it is delivered to the requesting client. Both operations, the migration to and from the tertiary storage system, are completely invisible to dCache end users.

The tape system connection scripts are easily customizeable for various systems, and example scripts exist for connecting dCache with cloud storage services like Amazon S3 as overflow backends.

File Replication for increased Resiliency

dCache's resilience service can ensure that, at any given time, a specified number of physical copies of a file are present on the system to provide tolerance against hardware failures. The granularity of file distribution can be freely configured, enabling on-a-different-machine, in-a-different-rack, on-a-different-netsegment or even in-a-different-country setups.

Dynamic File Replication for Load Balancing and enhanced Throughput

Whenever a file is requested that has more than a single copy available, dCache selects a pool with a comparatively low load to serve the incoming request, thus providing load balancing.

dCache can, additionally, be configured to automatically create more copies of a file in high-load situations, in order to spread the load across the system even better. These extra copies can later be removed in a fully automated, transparent fashion.

Rebalancing Data Distribution and Server Lifecycle

Larger instances will see new machines being brought into operation and old machines being retired frequently. dCache's *migration* and *rebalancing* functionality supports all stages of a machine's lifecycle: For new, empty storage servers, migrating data helps getting them to an equal fill ratio with the rest of an instance's machines, thus evening out the spread of I/O operations across all the available hardware.

During the operation of a dCache instance, distribution of files across the available machines is a self-balancing process. However, in exceptional cases like the deletion of very large amounts of data, a rebalancing operation can again help out in getting usage of machines evened out across the instance.

Additionally, the migration module can be used to clear data off of a machine at the end of its service life.

Adapted to different Use Cases

Grid Architectures

Historically, dCache has been developed within the context of the Worldwide LHC Computing Grid WLCG. This has led to the system being optimized for the typical grid computing workloads.

All relevant grid-specific data transfer protocols are supported, and dCache is well-suited for integration with transfer services and the upcoming federated storage architectures in the WLCG environment.

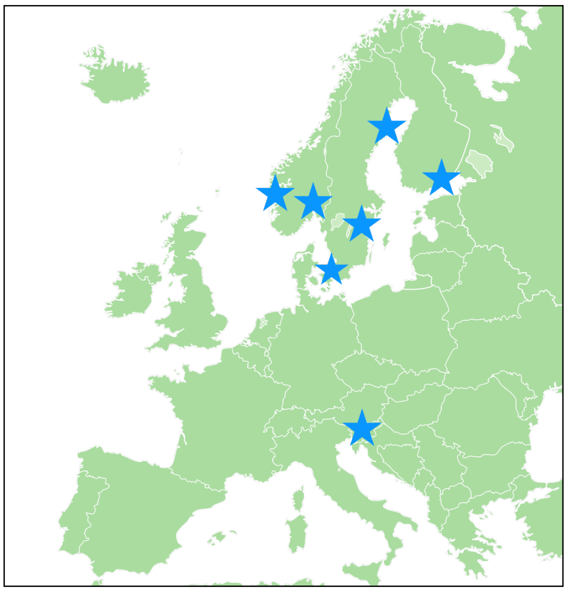
X.509 certificates are widely used throughout the Grid world to authenticate users and to give jobs a „proxy certificate“ that carries authN information across grid sites. dCache fully supports X.509 along with the other authN schemes.

Since the early 2000s, dCache has consistently played a major role in the WLCG data management system, currently hosting many dozens of Petabytes at around 20 large institutes around the world.

Federated Storage

Federating storage resources that are distributed across multiple data centres is becoming more and more important, especially in the context of scientific computing. dCache can both be deployed as a federated dCache-only instance and as part of large storage federations that consist of a mix of storage systems.

A dCache instance can internally consist of geographically and organizationally distributed machines. This enables cross-datacenter replication for resiliency purposes as well as for making data available in various LANs without the need for high-latency WAN transfers.



The NDGF dCache instance spans datacentres across Scandinavia and Slovenia, but is administered and used as a single instance.

Data Management for Scientific Experiments

But also outside of classical Grid environments, dCache's features make it ideally suitable for the reliable, long-term storage and management of scientific data. Being able to transparently use tape libraries as cost-effective nearline storage is a great benefit for all experiments with a need to archive rarely-accessed data for a long time.

One speciality of dCache is that it is conceptually a write-once file system, that means a file is immutable once it is written. This is a deliberate design decision within the system, intended to ensure that raw data files will always be unperturbed and matching the typical scientific workflow of taking data and incrementally adding artifacts from processing steps.

General Purpose Cloud Storage

With its flexible door mechanism enabling different access protocols, dCache is well-suited for integration with industry-standard cloud systems. It has been shown to perform well as a backend for enterprise-scale sync and share services.

Resources and Documentation

The dCache project's web page is at https://dcache.org, and the source code is available from the project's GitHub page at https://github.com/dCache.
The dCache project offers several documents detailing various aspects of the system:

dCache documentation

Purpose	Document
Installation	The dCache Book, Chapter 2: "Installing dCache" https://www.dcache.org/manuals/Book-5.2/install.shtml
Administration and Operation	The dCache Book https://www.dcache.org/manuals/Book-5.2/
End-user guide	The dCache Users' Guide https://www.dcache.org/manuals/UserGuide-5.2/
Source code and building	https://github.com/dCache/dcache/blob/master/BUILDING.md
Developers' guide	https://github.com/paulmillar/dCache-developers-guide