Qumulo Data Platform

Software Architecture Overview

February 7, 2020

Our Mission: Enable Creators and Builders

Qumulo’s mission is to help innovators unleash the power of their data wherever it resides. Innovators create new businesses, treatments, products, and art by transforming data into value. This transformation is built around a data lifecycle where collaborators capture, interact, transform, publish, then archive their data. Inside this lifecycle, “creators” (artists, researchers, etc.) use digital data to do their work. We see this transformation in action in several industries.

Life Sciences

Scientists explore data captured from sequencers to identify anomalies, then compute clusters transform the raw data into finished discoveries. Those discoveries are published to the research community, then the data is archived.

Media and Entertainment

Artists edit data captured from daily shoots to create initial scenes, then render pipelines transform that into finished film. Distributors publish that film to online outlets, and the finished content is archived.

Manufacturing & IoT

Logs and images are generated from sensors (with high volume, velocity, and variety) and analyzed in real time for component failures. Later, business analysts review the data to explore opportunities for process improvements, and by data scientists to build better machine learning models. Those models are published into the manufacturing line to improve production efficiency, and finished logs and images are archived.

Requirements of a data platform

In order to innovate, organizations depend on *unstructured data platforms*. These persistent data platforms power applications and micro services for builders and developers. They provide fast and reliable access to builders and creators, who transform data into outcomes and discovery. Innovators require that their unstructured data platforms:

Be cloud-ready
Platforms must be built for public, private, and hybrid cloud infrastructures, offering unstructured data across all environments. They should also seamlessly fit with the ecosystem of cloud services (e.g. AI/ML, IoT, or cloud object stores).

Scale
Platforms must be able to serve petabytes of data, billions of files, millions of operations, and thousands of users.

Work with standard tools
The most valuable creators in an innovative organization (artists, researchers, data scientists, and analysts) need to be able to use their tools without having to install custom drivers or make changes to their workflow.

Provide visibility and automation
Administrators must be able to create, manage, and tear down data services using RESTful APIs. They must be able to understand the performance and capacity utilization of their data services in real time in order to better diagnose issues and plan for the future.

Secure and enterprise-ready
Data is the lifeblood of innovative organizations and therefore must be protected using industry-standard identity and encryption tools. The data platform must satisfy company requirements for disaster recovery, backup, and user management.

The Challenges with Existing Solutions

Organizations that thrive on data to drive innovation are poorly served by available unstructured data platforms.

Open source and Windows-based file systems scale poorly, are difficult to automatically provision, and require substantial management.

Legacy file systems based on hardware appliances lack real-time visibility features, offer incomplete API surfaces, and can only support public cloud workloads via cloud-adjacent hardware-as-a-service offerings. Scale-up variants such as Netapp struggle to scale single namespaces beyond 100 TB.

Cloud object and file services support many cloud innovation workloads but do not enable creators to use their standard tools, mostly due to the lack of multi-protocol file support. Furthermore, they lack many of the security and enterprise features organizations require in order to move workloads to the public cloud.

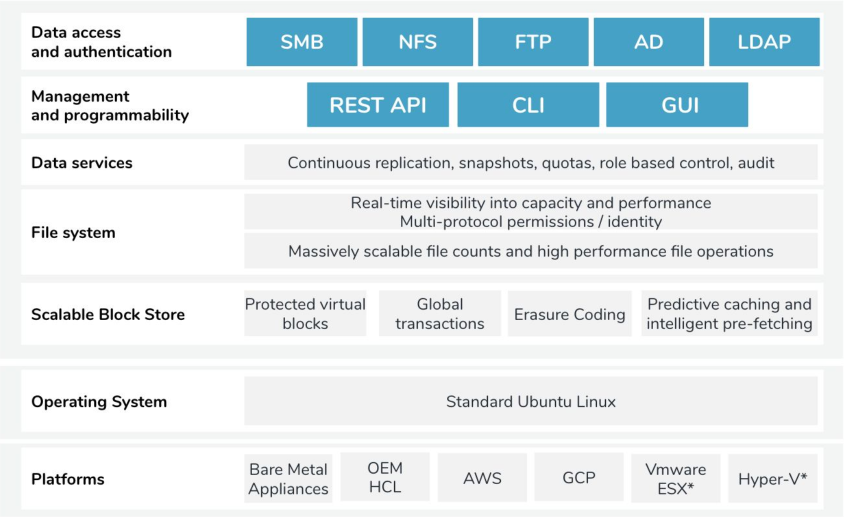
On-prem object stores offer low-cost data storage, but are fundamentally unfit for the interactive and transformative stages of the data lifecycle due to poor performance and lack of support for standard end-user tools.

Qumulo’s Software Architecture

Qumulo was founded to empower creators with an unstructured data platform for private and public clouds.

We package that platform into cloud-ready, scalable products which enable creators to use essential tools. We also provide robust APIs for management and real-time visibility to system usage, and meet the security and data protection requirements of Fortune 500 enterprises.

The purpose of this paper is to provide an overview of the architecture of Qumulo’s software platform in order to illustrate how our product delivers the aforementioned benefits to innovators and creators. To illustrate the architectural differentiation and value of our system, we will explore the major layers of our software. At each layer we will describe the purpose of the layer, how it delivers elements of our value proposition, and the innovation driving that value.



Data access and authentication	SMB	NFS	FTP	AD	LDAP	
Management and programmability	REST API	CLI	GUI			
Data services	Continuous replication, snapshots, quotas, role based control, audit					
File system	Real-time visibility into capacity and performance Multi-protocol permissions / identity Massively scalable file counts and high performance file operations					
Scalable Block Store	Protected virtual blocks	Global transactions	Erasure Coding	Predictive caching and intelligent pre-fetching		
Operating System	Standard Ubuntu Linux					
Platforms	Bare Metal Appliances	OEM HCL	AWS	GCP	Vmware ESX*	Hyper-V*

Fundamentals of the Qumulo System

Before diving into the individual components of the system, there are several foundational assumptions that are important in understanding the Qumulo architecture:

1. **Qumulo provides a distributed system that presents a single namespace.** Qumulo's systems are shared-nothing clusters of independent nodes. Each node provides capacity and performance. Individual nodes stay in constant coordination with each other. Any client can connect to any node and read and write in the namespace.

2. **Qumulo's File System is optimized for scale.** We ensure all aspects of our product can comfortably support petabytes of data, billions of files, millions of operations, and thousands of users.

3. **Qumulo's File System is highly available and immediately consistent**. Qumulo's unstructured data platform is built to withstand component failures in the infrastructure while still providing reliable service to clients. We do this through the use of software abstraction, erasure coding, advanced networking technologies, and rigorous testing. When data is written to Qumulo's software, we do not acknowledge that write to the service, user, or compute node until we have stored that data in persistent storage. Thus any read will be of a coherent view of the data (as opposed to eventually consistent models).

4. **Qumulo delivers software built for the public, private, and hybrid cloud**. Qumulo's software makes few assumptions about the platform on which it runs. It abstracts the underlying physical or virtual hardware resources in order to take advantage of the best public and private cloud infrastructure. This enables us to leverage the rapid innovation in compute, networking, and storage technologies driven by the cloud providers and the ecosystem of component manufacturers.

5. **The Qumulo File System is API-first**. Every capability built by Qumulo first emerges as API endpoints. We then present a curated set of those endpoints in our command line interface (CLI) and our management UI (GUI). This includes system creation, data management, performance and capacity analytics, authentication, and data accessibility.

6. **Qumulo ships innovation rapidly and regularly**. We release new versions of our software every two weeks. This enables us to rapidly respond to customer feedback, drive constant improvement in our product, and insist on production-quality code from our teams.

7. **Qumulo's customer success team is highly responsive, connected, and agile.** Each file system from Qumulo has the ability to connect to remote monitoring via our Mission Qontrol cloud-based monitoring service. Our customer success team uses that data to help customers through incidents, provide insight into product usage, and to alert customers when their systems are experiencing component failures. This combination of intelligent support and rapid product innovation powers an industry-leading NPS score of 80+.

Data Access and Authentication

Purpose

Enable access to data using standard applications and operating systems while ensuring enterprise-grade security.

How it works

Our data access layer supports the three file access protocols most commonly adopted by creators (NFS, SMB, and FTP). These protocols exist as independent and scalable resources on each node of a Qumulo cluster. End users see a single namespace which can expand in capacity and performance. This namespace can be accessed seamlessly from any Windows, Mac, or Linux computing device and, therefore, any unstructured data application.

Our authentication layer supports the two industry standard identity services: Active Directory (AD) and Lightweight Directory Access Protocol (LDAP). Qumulo's data services integrate with these global identity systems as managed by customers, enabling access to be controlled across compute, end users, and data. Connecting the Qumulo File System to identity services requires simple set-up and works well with complex and distributed identity services configurations (a common challenge in enterprise private and public cloud environments).

Each data access protocol uses a common authentication layer to interact with the data stored in our file system. This enables users to move between applications, operating systems and environments, all while accessing the same data. As data moves through the data lifecycle (from capture through transformation and archive) this separation of layers offers critical flexibility and reduces the number of systems customers need to maintain.

Points of Innovation

Qumulo's data access protocols enable users to leverage any standard Windows, Mac, or Linux operating system and any standard application without making changes to their environment.

Qumulo supports both stateful (SMB) and stateless (NFS) data access protocols from the same scalable namespace.

Qumulo enables enterprise-grade identity management in a scalable system with high availability.

Management and Programmability

Purpose Enable application owners to build integrated solutions with the Qumulo platform, and enable administrators to automate and manage their data services.

How it works The management and programmability layer is made up of three capabilities; a REST API, a command line interface (CLI), and a management interface (GUI).

The REST API

The REST API is a superset of all capabilities in the Qumulo data platform. From the API, customers can:

- Create a namespace (in the cloud using a Terraform or Cloud Formation template)
- Configure all aspects of a system (from security such as identity services or management roles, to data management such as quotas, to data protection such as snapshot policies or data replication, to adding new capacity)

- Gather information about their system (including capacity utilization and performance hotspots)

- Access data (including read and write operations)

The API is “self-documenting,” making it easy for developers and administrators to explore each end point (and see example outputs). Qumulo maintains a collection of sample uses of our API on Github (https://qumulo.github.io/).

The Command Line Interface (CLI)

The Qumulo CLI offers most (but not all) of the API and is focused on system administrators. The CLI offers a scriptable interaction method for working with a Qumulo system. The CLI offers roughly 200 unique commands (as of Qumulo Core Version 3.0.3). A full list of commands can be found in our Knowledge Base (www.care.qumulo.com).

The Management Interface (GUI)

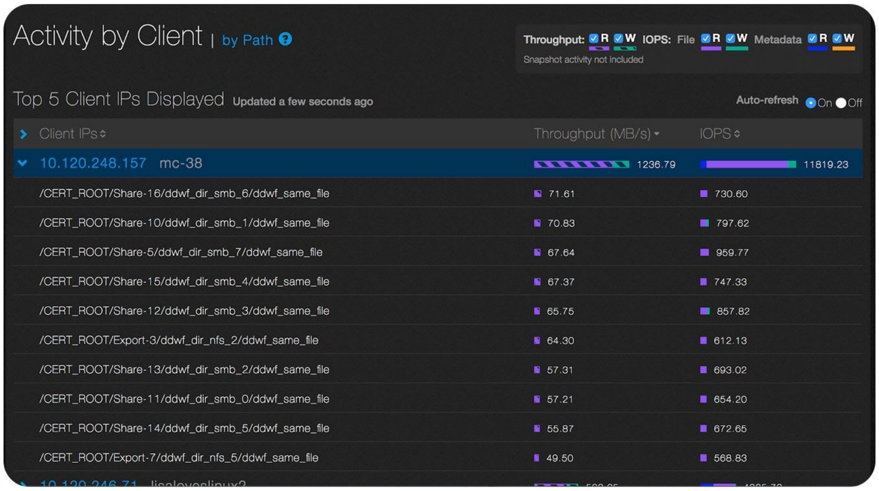
The Qumulo GUI offers a user-focused way to interact with a Qumulo system. The GUI is a web-based interface, served from the system, with no separate VM or service needed. The GUI is organized around six top-level navigation sections: Dashboard, Analytics, Sharing, Cluster, API & Tools, and Support.

Dashboard

This is the “home page” of the Qumulo UI. It offers a series of easily digestible insights into the activity, growth, performance and health of a Qumulo system. The dashboard visualizes available and used capacity (including data, metadata, and snapshots), capacity growth and file count growth, aggregate performance of the system, and balance of workloads.

Analytics

This section offers system administrators a granular and real-time view of their systems not available in other data platforms. These analytics offer visibility into the performance utilization of each workload (either by client or by data path) and into the capacity growth of their system, including a view of which portions of the file system are growing or shrinking by time period, enabling customers to figure out which workloads are consuming their capacity.



Activity by Client | by Path

Throughput: ☑ R ☑ W IOPS: File ☑ R ☑ W Metadata ☑ R ☑ W
Snapshot activity not included

Top 5 Client IPs Displayed Updated a few seconds ago

Auto-refresh ● On ○ Off

Client IPs	Throughput (MB/s)	IOPS
10.120.248.157 mc-38	1236.79	11819.23
/CERT_ROOT/Share-16/ddwf_dir_smb_6/ddwf_same_file	71.61	730.60
/CERT_ROOT/Share-10/ddwf_dir_smb_1/ddwf_same_file	70.83	797.62
/CERT_ROOT/Share-5/ddwf_dir_smb_7/ddwf_same_file	67.64	959.77
/CERT_ROOT/Share-15/ddwf_dir_smb_4/ddwf_same_file	67.37	747.33
/CERT_ROOT/Share-12/ddwf_dir_smb_3/ddwf_same_file	65.75	857.82
/CERT_ROOT/Export-3/ddwf_dir_nfs_2/ddwf_same_file	64.30	612.13
/CERT_ROOT/Share-13/ddwf_dir_smb_2/ddwf_same_file	57.31	693.02
/CERT_ROOT/Share-11/ddwf_dir_smb_0/ddwf_same_file	57.21	654.20
/CERT_ROOT/Share-14/ddwf_dir_smb_5/ddwf_same_file	55.87	672.65
/CERT_ROOT/Export-7/ddwf_dir_nfs_5/ddwf_same_file	49.50	568.83

Sharing

This section enables administrators to make data accessible to creators by creating SMB shares and NFS exports. It also enables administrators to manage capacity usage through quotas. Finally the sharing section is where administrators go to manage the cluster's connection to AD and LDAP identity services.

Cluster

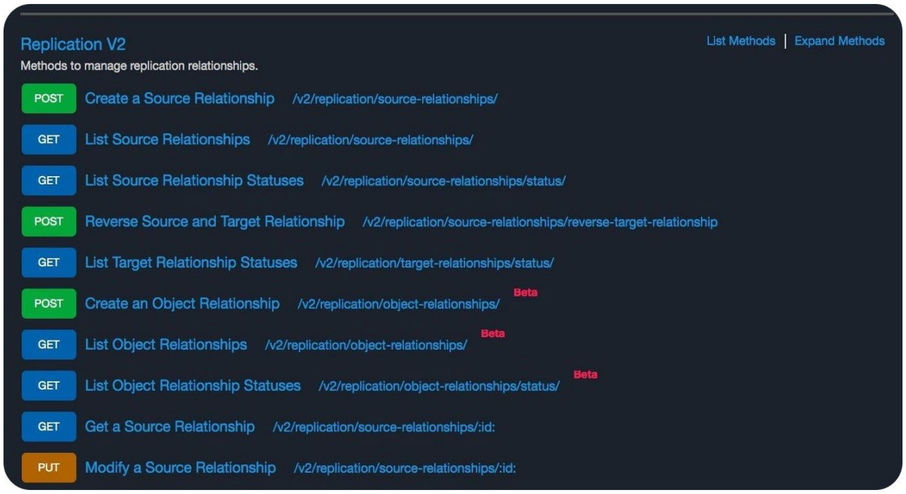
This section enables administrators to configure snapshot policies, continuous replication policies, networking, and time and date services. It allows them to see the nodes of the system, and to add new nodes.

API & Tools

This section enables customers to download the Qumulo command line tool and explore our self-documenting API. It includes the ability to "try" each endpoint and see sample JSON outputs.

Support

This section enables customers to perform software upgrades, connect to remote monitoring and authorize Qumulo's customer success team to connect remotely to a Qumulo system.



Replication V2

List Methods | Expand Methods

Methods to manage replication relationships.

POST	Create a Source Relationship	/v2/replication/source-relationships/
GET	List Source Relationships	/v2/replication/source-relationships/
GET	List Source Relationship Statuses	/v2/replication/source-relationships/status/
POST	Reverse Source and Target Relationship	/v2/replication/source-relationships/reverse-target-relationship
GET	List Target Relationship Statuses	/v2/replication/target-relationships/status/
POST	Create an Object Relationship	/v2/replication/object-relationships/ Beta
GET	List Object Relationships	/v2/replication/object-relationships/ Beta
GET	List Object Relationship Statuses	/v2/replication/object-relationships/status/ Beta
GET	Get a Source Relationship	/v2/replication/source-relationships/:id:
PUT	Modify a Source Relationship	/v2/replication/source-relationships/:id:

Points of Innovation

The Qumulo API offers a complete superset of all capabilities in the Qumulo system, enabling customers to develop against Qumulo's software and manage their Qumulo system entirely through modern infrastructure management tools.

The Qumulo UI offers simple (and understandable) tools to manage Qumulo systems, which reduce IT expenditure in terms of both cost and time.

The dashboard and analytics UI provide real-time, actionable insights into any Qumulo system. Users can understand how well their Qumulo platform is serving creators and offers insights into the workloads of those creators.

Data Services

Purpose

Protect, secure, and manage data in the Qumulo platform using the enterprise-grade tools that CIOs and CSOs expect from data platforms.

How it works

The Data Services layer is made up of five capabilities - snapshots, replication, quotas, audit, and role-based access control (RBAC).

Snapshots

Data stored in a Qumulo system can be viewed both in its current form and in previous versions via snapshots. These snapshots use a unique write-out-of-place methodology that only consumes space when changes occur. This makes Qumulo's snapshots both efficient and performant. Snapshots are controlled by a snapshot policy that articulates the portion of the namespace to be protected, the frequency of the snapshots, and how long the snapshots are kept.

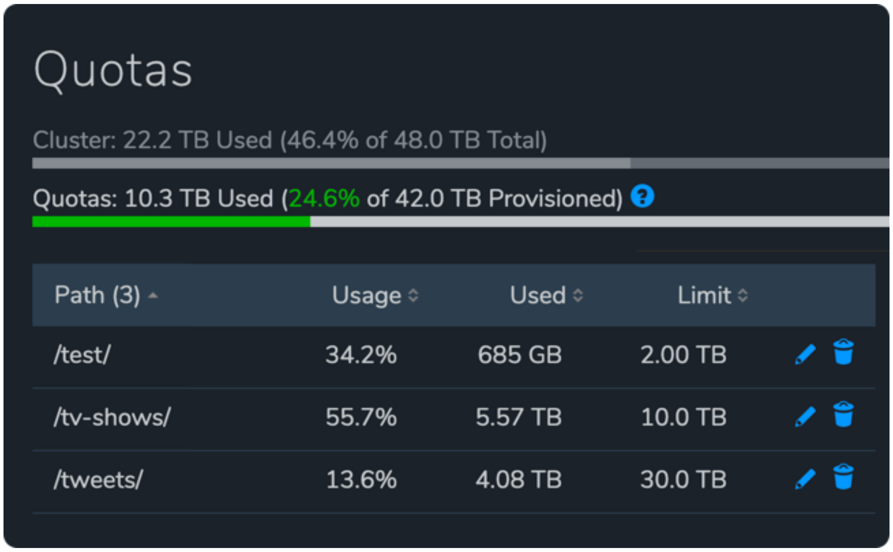
Snapshots policies can be linked with replication policies. This enables snapshots to be replicated to a second Qumulo system and enables frequent snapshots to be kept on one Qumulo system and less frequent snapshots on another (a common enterprise data loss and ransomware protection strategy). Snapshots can be restored by administrators via the UI/CLI/API, or by end users via client OS tools (e.g. "Previous Versions" in Windows). The limit on the total number of snapshots in a Qumulo system is measured in the tens of thousands, higher than most other systems.

Replication

Replication enables users to copy, move, and synchronize data across multiple Qumulo systems. Our replication technology offers two core capabilities: efficient data movement and granular identification of changed data. Qumulo's replication is continuous, meaning that any new changes to a replicated directory will be identified and moved, asynchronous, and uni-directional. Our replication technology leverages snapshots to create a list of changed file regions in a given time period, which are then moved to a second Qumulo system over an encrypted data transfer protocol. The list of changed files is available as its own API endpoint, which third party ISVs use to integrate Qumulo into data backup systems. Qumulo's replication works across any two Qumulo systems including on-prem to cloud, cloud-to-cloud, and across cloud regions. Replication is used to enable petabyte-scale backup, especially when coupled with snapshot replication disaster recovery including failover and failback. It also enables hybrid cloud and cloud bursting, multicloud and multi-region infrastructure, and remote collaboration scenarios.

Quotas

Quotas enable users to control the growth of any subset of a Qumulo namespace. Quotas act as independent limits on the size of any directory, preventing data growth when the capacity limit is reached. Unlike other products, Qumulo quotas take effect instantaneously, which enables administrators to identify rogue workloads using our real-time capacity analytics and instantly stop rampant capacity usage. Quotas even follow the portion of the namespace they cover when directories are moved or renamed.



Quotas

Cluster: 22.2 TB Used (46.4% of 48.0 TB Total)

Quotas: 10.3 TB Used (24.6% of 42.0 TB Provisioned)

Path (3)	Usage	Used	Limit
/test/	34.2%	685 GB	2.00 TB
/tv-shows/	55.7%	5.57 TB	10.0 TB
/tweets/	13.6%	4.08 TB	30.0 TB

Audit

Audit enables security administrators to track all actions taken in a Qumulo namespace and configuration changes to the system. Audit captures all data access and modification, attempts to access a Qumulo system, sharing of data through new shares or exports, and changes to system configuration or data protection schemes. Audit sends those activity logs to any standard remote syslog server.

Role-Based Access Control

Role-Based Access Control (RBAC) enables security administrators to use their identity services to control which users or groups have rights to make changes to a Qumulo system or view the UI. RBAC offers several pre-configured "roles" that have rights to take action in or on the system (admin, observer, data admin). Administrators add users and groups from AD or LDAP to those roles. Administrators can also create custom roles to match unique security regimes in their organization (e.g., a "backup administrator" role).

Points of Innovation

Snapshots in Qumulo's system are efficient, performant, and scalable (to 40k or more).

Snapshot replication enables built-in, petabyte-scale backup for any Qumulo system, no matter where it's located.

Replication enables any Qumulo system to copy, move, or sync data to any other Qumulo system (from on-prem to cloud, cloud region to cloud region, or across clouds).

Quotas in Qumulo are real-time and do not require lengthy file system enumerations (aka "tree walks") to take effect.

Audit integrates simply with modern infrastructure management tools such as Splunk.

Object store replication enables creators to capture data in a Qumulo namespace, copy it to S3 to be transformed by a cloud service (e.g. ML or image recognition) and then be returned to the Qumulo system.

The Qumulo File System

Purpose

Organize data in understandable structures, enable workloads with massive file counts, empower creators to collaborate on data sets as they move through the data lifecycle, and provide real-time insight into performance and capacity utilization, even when systems scale to petabytes and billions of files.

How it works

The Qumulo File System organizes all data stored in a Qumulo system into a namespace. This namespace is POSIX-compliant and maintains the permissions and identity information that support the full semantics available over the NFS or SMB protocols. Like all file systems, the Qumulo File System organizes data into directories, and presents data to SMB and NFS clients. However, the Qumulo File System has several unique properties: the use of B-trees, a real-time analytics engine, and cross protocol permissions (XPP).

B-Trees in the File System

The Qumulo File System can scale to billions of files without experiencing the problems common in other systems like running out of inodes slowdowns, inefficiency and long component failure recovery. We accomplish this by using a collection of technologies, one of which is the B-tree. B-trees are particularly well-suited for systems that read and write large numbers of data blocks because they are “shallow” data structures that minimize the amount of I/O required for each operation as the amount of data increases. With B-trees as a foundation, the computational cost of reading or inserting data blocks grows very slowly as the amount of data increases.

For example, B-trees are ideal for file systems and very large database indexes. In Qumulo’s File System, B-trees are block-based. Each block is 4096 bytes, and each 4K block may have pointers to other 4K blocks. The Qumulo File System uses B-trees for many different purposes. There is an inode B-tree, which acts as an index of all the files. The inode list is a standard file system implementation technique that makes checking the consistency of the file system independent of the directory hierarchy. Inodes also help to make update operations, such as directory moves, efficient. Files and directories are represented as B-trees with their own key/value pairs, such as the file name, its size and its access control list (ACL) or POSIX permissions. Configuration data is also a B-tree and contains information such as the IP address of the cluster.

Real-Time Analytics Engine

Qumulo offers insight into the capacity and performance utilization of data in a Qumulo File System. This enables customers to see, within a 15-second time period, which portions of the file system have grown (or shrunk), which applications are consuming performance resources, and which parts of the file system are most active. This enables customers to troubleshoot applications, manage capacity consumption, and plan using real data. These insights are powered by two technologies: capacity metadata aggregation and file system sampling.

Capacity metadata aggregation

In the Qumulo File System, metadata such as bytes used and file counts are aggregated as files, and directories are created or modified. This means that the information is available for timely processing without expensive file system tree walks. The real-time analytics engine maintains up-to-date metadata summaries across the file system namespace. It uses the file system's B-trees to collect information about the file system as changes occur. Various metadata fields are summarized inside the file system to create a virtual index. As changes occur, new aggregated metadata is gathered and changes are propagated up from the individual files to the root of the file system.

As each file (or directory) is updated with new aggregated metadata, its parent directory is marked as out-of-date and another update event is queued for the parent directory. In this way, file system information is gathered and aggregated while being passed up the tree. The metadata propagates up from the individual node, at the lowest level, to the root of the file system as data is accessed. Each file and directory operation is accounted for.

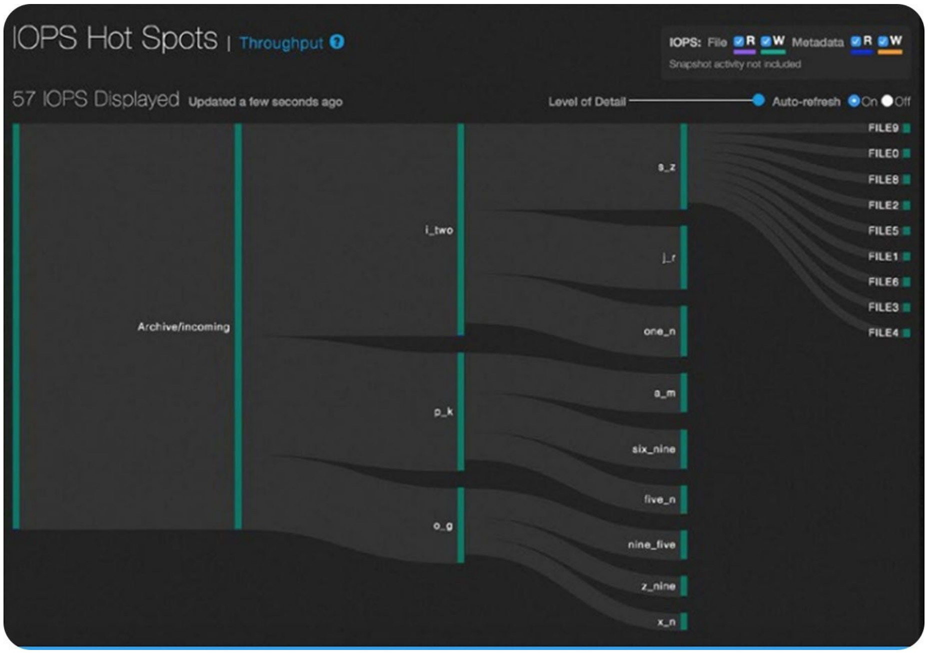
In parallel to the bottom-up propagation of metadata events, a periodic traversal starts at the top of the file system and reads the aggregate information present in the metadata. When the traversal finds recently updated aggregate information, it prunes its search and moves on to the next branch. It assumes that aggregated information is up-to-date in the file system tree from this point down towards the leaves, including all contained files and directories, and does not have to go any deeper for additional analytics. Most of the metadata summary has already been calculated, and, ideally, the traversal only needs to summarize a small subset of the metadata for the entire file system. In effect, the two parts of the aggregation process meet in the middle with neither having to explore the complete file system tree from top to bottom.

File system sampling

One example of Qumulo's real-time analytics is its performance hot spots reports. Representing every throughput operation and IOPS within the GUI would be infeasible in large file systems. Instead, Qumulo's real-time analytics engine uses probabilistic sampling to provide a statistically valid approximation of this information. Totals for IOPS read-and-write operations, as well as I/O throughput read-and-write operations, are generated from samples gathered from an in-memory buffer of tens of thousands or more entries that is updated every few seconds.

The report shown displays the operations that are having the largest impact on the cluster. These are represented as hotspots in the GUI.

Qumulo's ability to use statistically valid probabilistic sampling is only possible because of the summarized metadata for each directory (bytes used, file counts) that is continually kept up-to-date by the real-time analytics engine.



Cross-Protocol Permissions (XPP)

In order to empower creators to share data across the innovation lifecycle, Qumulo must enable the same data to be accessed by a variety of operating systems (e.g. Windows, Mac, Linux). However, those systems rely on NFS and SMB which have very different languages for expressing identity. As an example, a lab could create data using a genomic sequencer, which runs Windows, and therefore express identity using the rich language of ACE's and ACL's. Then a researcher might analyze that data using a Windows client, which also uses SMB and therefore understands ACLs. At some point the researcher will want to coordinate a parallel compute operation using an HPC cluster running Linux, which expects POSIX permissions (e.g. mode bits). The research organization would be faced with the choice to either simplify all permissions to the least restrictive set, thereby allowing the required access, or to move data into an entirely separate namespace, which would break the collaboration flow and increase IT costs.

Qumulo solved this problem by creating a file system that can translate and rationalize multiple permission languages, such that any client will see the permissions they expect without sacrificing protocol expressivity. We call this technology XPP. Cross-Protocol Permissions (XPP) enables mixed SMB and NFS protocol workflows by preserving SMB ACLs, maintaining permissions inheritance, and reducing application incompatibility related to permissions settings.

Cross-Protocol Permissions is designed to operate in the following ways:

- When there is no cross-protocol interaction, Qumulo operates precisely to protocol specifications.
- When conflicts between protocols arise, cross-protocol permissions minimize the likelihood of application incompatibilities.
- Enabling cross-protocol permissions won't change rights on existing files in a filesystem. Changes may only happen if files are modified while the mode is enabled.

Points of Innovation

The Qumulo File System can scale to billions of files while preserving high efficiency, resiliency to component failure and high performance.

Customers can diagnose workflows, identify misbehaving applications, manage capacity consumption and plan for the future using real time data on performance and capacity utilization, even in billion file and petabyte-scale file systems.

Creators and builders can collaborate across the data lifecycle, using standard tools for end-user clients and HPC compute, from the same Qumulo namespace by virtue of cross-protocol permissions.

The Scalable Block Store

Purpose

Bring the Qumulo data platform to any private and public cloud environment, enable massive scale, guarantee consistency across a system, protect against component failure and power high performance and interactive workloads.

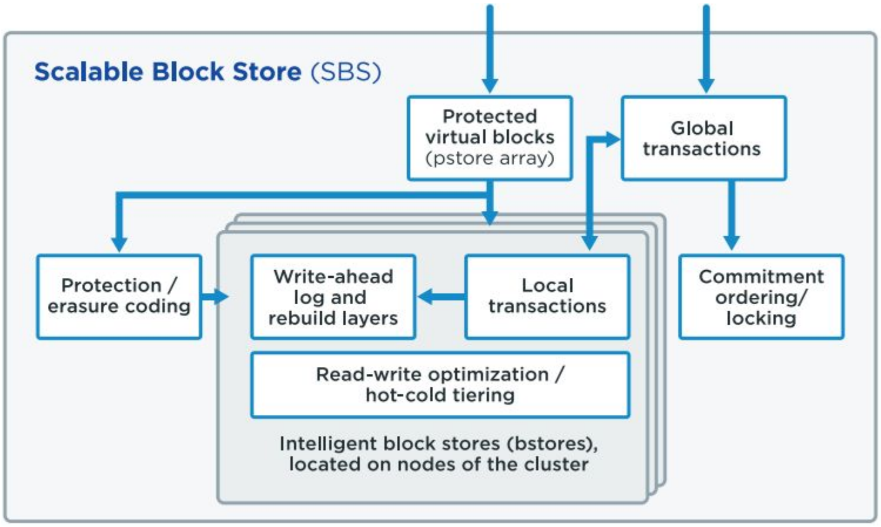
How it works

The foundation of the Qumulo platform is the Scalable Block Store (SBS). The SBS leverages several core technologies to enable scale, portability, protection and performance: a virtualized block system, erasure coding, a global transaction system and an intelligent cache.

The Virtual Block System

The storage capacity of a Qumulo system is conceptually organized into a single, protected virtual address space. Each protected address within that space stores a 4K block of bytes. Each of those "blocks" is protected using an erasure coding scheme to ensure redundancy in the face of storage device failure. The entire file system is stored within the protected virtual address space provided by SBS, including the directory structure, user data, file metadata, analytics and configuration information.

The protected store acts as an interface between the file system and block-based data recorded on the attached block devices. These devices might be flash devices or hard drives, either in a dedicated server in the private cloud or a virtual server in the public cloud. By using 4K blocks, the virtual block system enables highly efficient storage of all file sizes (large to small).


Scalable Block Store (SBS)
Protected virtual blocks (pstore array)
Global transactions
Protection / erasure coding
Write-ahead log and rebuild layers
Local transactions
Commitment ordering/ locking
Read-write optimization / hot-cold tiering
Intelligent block stores (bstores), located on nodes of the cluster

Erasure coding

Each virtual block is part of a larger protection group called a Protected Store (or pstore), which leverages erasure coding to distribute and protect data across a distributed Qumulo system. That pstore is the organizing container for data protection. Within and across pstores, data blocks leverage Reed-Solomon algorithms to create "parity" copies of data blocks which are used to rebuild blocks that are damaged by a component failure. The number of parity blocks determines the redundancy of the cluster with larger clusters requiring more redundancy than smaller ones as they contain more components that could fail. These pstores are then distributed across a Qumulo system to control for component failure in a .

Qumulo's implementation of erasure coding, coupled with our granular virtual address space, enables Qumulo systems to rapidly and predictably rebuild data from failed components. Additionally, this allows the systems to leverage the densest flash- and disk-based media available in the public and private cloud, and to operate large-scale systems with reliable performance and strong data protection guarantees. Finally, this protection system enables customers to confidently use 100% of the available protected space in a Qumulo system, in contrast to other systems which perform poorly or unpredictably beyond ~80% utilization.

Global Transaction System

Because Qumulo is a distributed shared-nothing system that makes immediate consistency guarantees, we require a mechanism to ensure that every node in the system has a consistent view of all data. We accomplish this by ensuring that all reads and writes to the virtual address space are transactional.

This means that when a file system operation requires a write operation that involves more than one block, the operation will either write all the relevant blocks, or none of them. Atomic read and write operations are essential for data consistency and the correct implementation of file protocols such as SMB and NFS. For optimum performance, SBS uses techniques that maximize parallelism and distributed computing while also maintaining transactional consistency of I/O operations. For example, SBS is designed to avoid serial bottlenecks, where operations would proceed in a sequence rather than in parallel. SBS's transaction system uses principles from the ARIES algorithm commonly used in databases for non-blocking transactions, including write-ahead logging, repeating history during "undo" actions, and logging "undo" actions.

However, SBS's implementation of transactions has several important differences from ARIES. SBS takes advantage of the fact that transactions initiated by the Qumulo File System are predictably short, in contrast to general purpose databases where transactions may be long-lived. A usage pattern with short-lived transactions allows SBS to frequently trim the transaction log for efficiency. Short-lived transactions allow faster commitment ordering.

Additionally, SBS's transactions are highly distributed, and do not require globally defined, total ordering of ARIES-style sequence numbers for each transaction log entry. Instead, transaction logs are locally sequential in each of the virtual blocks and coordinated at the global level, using a partial ordering scheme that takes commitment ordering constraints into account. The advantage of SBS's approach is that the absolute minimum amount of locking is used for transactional I/O operations, and this allows Qumulo clusters to scale to many hundreds of nodes.

Intelligent Caching and Prefetching

Qumulo systems store billions of files and petabytes of capacity. However, at any given moment only a small portion of that data is in the active working set of a creator or innovator. In order to guarantee that those creators have the fastest possible performance, and therefore to prevent customers from buying disparate systems for each stage of the data lifecycle, Qumulo makes several performance guarantees in our product:

1. All metadata,which is the most often read in any data set, lives on the fastest durable media in the system (i.e. flash).

2. Virtual blocks which are read frequently (as measured by a proprietary “heat index”) are stored on flash, virtual blocks which are read infrequently are moved to colder media if available.

3. As data is read, the system observes client behavior and intelligently prefetches new data into memory on the node closest to the client in order to speed up access times. We do this through the intelligent application of a series of predictive models which observe data naming schemes, data birthing order and read patterns within large files. The system intelligently leverages the most effective model for any workload, and turns off models that are wasteful.

Points of Innovation

The Qumulo SBS abstracts underlying hardware components, enabling the Qumulo data platform to run in public and private cloud environments.

Qumulo's SBS offers uniquely efficient storage across all file sizes.

The combination of a virtualized block system and erasure coding enables customers to use all of their available space and to leverage the densest storage devices offered in the private and public cloud.

Qumulo SBS enables customers to comfortably build very large systems. Our limit as of March 2020 is 100 nodes and 36PB in a namespace, though increasing that limit is a function of testing, not architecture.

Qumulo’s global transaction system enables massively scalable performance with highly efficient distributed locking to guarantee immediate consistency.

Qumulo’s predictive caching and prefetching enable high performance on the most active data while also enabling systems to scale large enough to serve the entire innovation data lifecycle.

Conclusion

Qumulo built a data platform that can serve the entire data lifecycle from capture, through transformation, to archive, in the private and public cloud. To accomplish this, Qumulo provides a system that is cloud-ready, scales, enables creators to use standard tools, provides automation capabilities and visibility and is secure and enterprise ready.

Cloud-ready

The Qumulo data platform is available in both the public, private, and hybrid cloud.

Scale

Qumulo systems confidently scale to billions of files and petabytes of data. Qumulo systems also scale in performance to meet the demands of the most challenging workloads.

Standard tools

The Qumulo data platform supports Windows, Mac, and Linux clients.

Automation and visibility

Qumulo systems provide a robust API for programmability and automation and real-time insight into the capacity and performance utilization of the system.

Secure and enterprise-ready

Qumulo systems offer the identity, control, management and encryption tools which enterprises require from their infrastructure.