DATACORE | Swarm

Object Storage Performance Benchmark

Utilizing the Science and Technology Facilities Council's Super Data Cluster Environment

This whitepaper presents the results from benchmarking of DataCore Swarm object storage on a multi-Terabit converged Ethernet Software-Defined Storage Super Data Cluster deployed by the UK Science and Technology Facilities Council's (STFC) Scientific Computing Department (SCD) for the JASMIN project.

Many datasets, both model and observational data, are too big to be easily shipped around; JASMIN enables scientists to bring their processing to the data. It offers flexible data access for scientists to collaborate in selfmanaging group workspaces, enabling models and algorithms to be evaluated alongside curated archive data and data to be shared and evaluated before being deposited in the permanent archive.

JASMIN is funded by the Natural Environment Research Council (NERC), the UK Space Agency (UKSA), educational institutions and private industry to provide the UK and European climate and earth-system science communities with an efficient data analysis environment. It is managed jointly by the STFC Scientific Computing Department and the Centre for Environmental Data Analysis (CEDA), which is part of STFC Rutherford Appleton Laboratory (RAL) Space. The JASMIN infrastructure is located at RAL in Oxfordshire, UK.

EXECUTIVE SUMMARY

In technical terms, JASMIN is half supercomputer and half data center, and as such, provides a globally unique computational environment. JASMIN blends Petabytes of storage, thousands of cloud virtual machines, batch computing and WAN data transfer. The infrastructure provides compute and storage linked together by a highbandwidth network in a unique topology with significant compute connected with much greater bandwidth to disk than is typical of a normal data center. It has a supercomputer's network and storage, but without quite as much compute.

The first phase of JASMIN was funded in 2011 and deployed in 2012. Since then, hundreds of Petabytes of data have been accessed and analyzed by thousands of researchers. Due to increasing capacity needs, a growing researcher base and a shift in access from traditional file protocols to RESTful interfaces, the infrastructure managers at RAL started looking for methods to streamline infrastructure management, tenant management, access controls, and private and public file sharing. Their research led them to object storage. First and foremost, the object storage solution selected needed to achieve read performance characteristics similar to existing parallel file systems.

The following tables show the performance requirements defined by STFC and the Swarm results. Testing was conducted using 2 Gigabyte files with sequential reads and erasure-coded data via S3 and NFS.

S3 AGGREGATE THROUGHPUT

	MINIMUM PERFORMANCE REQUIREMENTS	SWARM RESULTS	% ABOVE MINIMUM REQUIREMENT
S3 Read	21.5 GB/s	35.0 GB/s	63%
S3 Write	6.5 GB/s	12.5 GB/s	92%

NFS SINGLE INSTANCE THROUGHPUT

	MINIMUM PERFORMANCE REQUIREMENTS	SWARM RESULTS	% ABOVE MINIMUM REQUIREMENT
NFS Read	150 MB/s	349 MB/s	132%
NFS Write	110 MB/s	392 MB/s	256%

Based on these benchmark results along with other functional tests, Swarm was selected as an object storage solution for JASMIN. The performance figures achieved are the result of Swarm's underlying parallel architecture and read results are similar to parallel benchmark tests run on a similar sized parallel file system in an equivalent environment. The following paper describes in detail the infrastructure architecture, testing methodologies and results achieved.

ENVIRONMENT

NETWORK

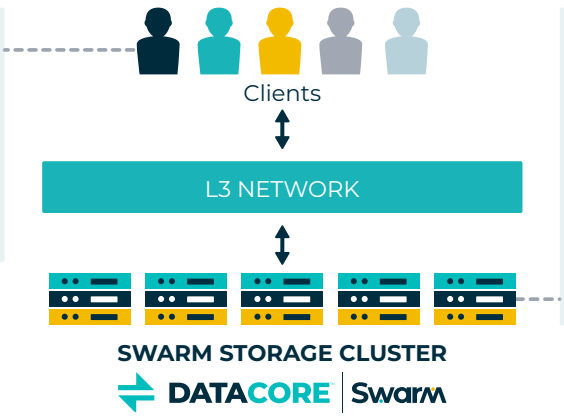
STFC employs an High-Performance Computing (HPC) "leaf/spine" routed CLOS network with 100Gb spine switches and 100Gb leaf or top of rack (TOR) switches. Every TOR is connected to every spine switch and there is equal uplink/downlink bandwidth on every TOR switch. This design delivers a super low-latency, non-blocking network where there are only 3 switch hops of <100nS between any network endpoint—orders of magnitude lower latency than an ordinary network.

SWARM OBJECT STORAGE ARCHITECTURE

At its core, Swarm is built around a "pure" object storage architecture that is simple, symmetrical and does not rely on traditional storage technologies such as caching servers, file systems, RAID or databases. Instead, data is written raw to disk together with its metadata, meaning objects are "self-describing." Identifying attributes such as the unique ID, object name and location on disk are published from the metadata into a patented and dynamic shared index in memory that handles lookups. This design is quite "flat," infinitely scalable and very low latency as there are zero IOPS (input/output operations per second) to first byte. It also eliminates the need for an external metadata database both for storing metadata and as a mechanism for lookups. Automatic load balancing of requests using a patented algorithm allows for all nodes and all drives to be addressed in parallel, removing the need for external load balancers and front side caching—both of which can present significant performance challenges in an HPC environment (where total aggregate throughput is the goal rather than single-stream performance). The simple, flat, self-balancing and self-optimizing Swarm architecture can handle thousands of concurrent client requests and deliver the full throughput value of all its drives in parallel. This allows ramp-up tests to be run to the point where the full aggregate throughput potential of the cluster is being delivered, as the results in this paper show.

All clients with at least the following specification:

128GB RAM
16x cores
RHEL 7.x
2x 25GB NIC ports



12x Supermicro Swarm Storage chassis, each with:

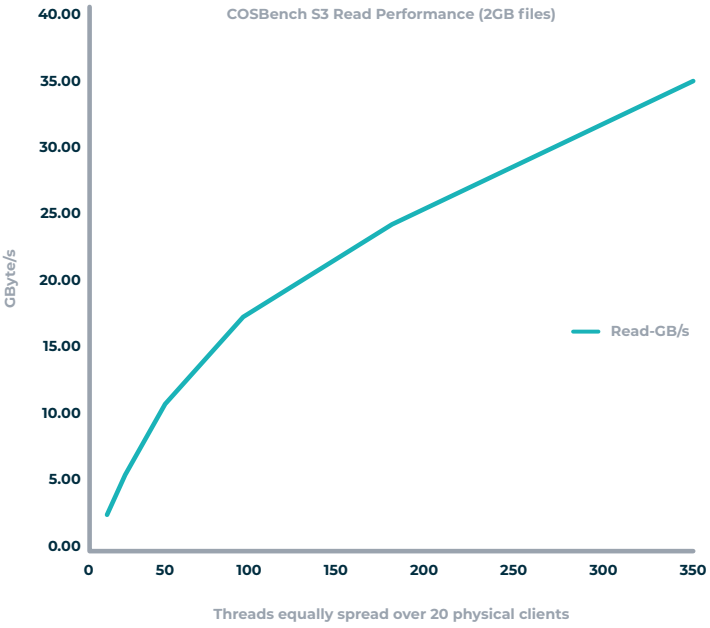
45x 12TB drives
2x 25GB NIC ports
256GB RAM, 12GB RAM per process
24x cores, 20 processes, ~2disks per core
5:2 erasure coding

DATASET & TESTING METHODOLOGY

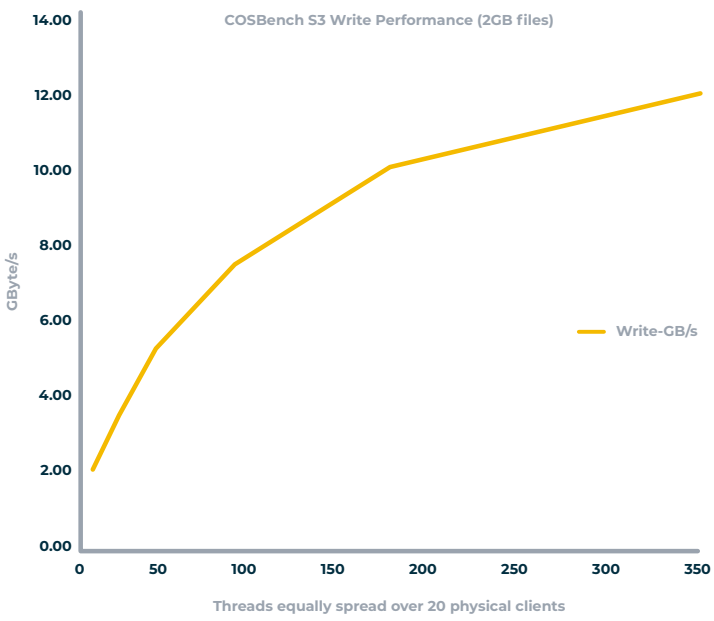
For S3 testing, COSBench was used to run ramp-up tests leveraging up to 20 physical client machines to measure the throughput potential of the entire Swarm cluster. Sequential tests were run using 2 Gigabyte erasure-coded files.

For NFS testing, a DD test was used in conjunction with custom scripting to run ramp-up tests leveraging up to 8 physical client machines to measure the throughput potential of a single SwarmFS interface. Sequential tests were run using a mixture of 1 and 10 Gigabyte erasure-coded files.

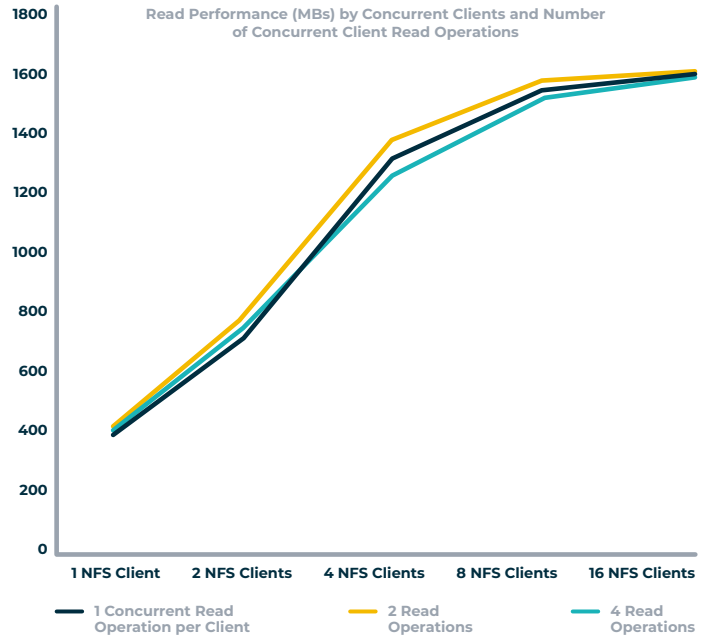
S3 READ RESULTS

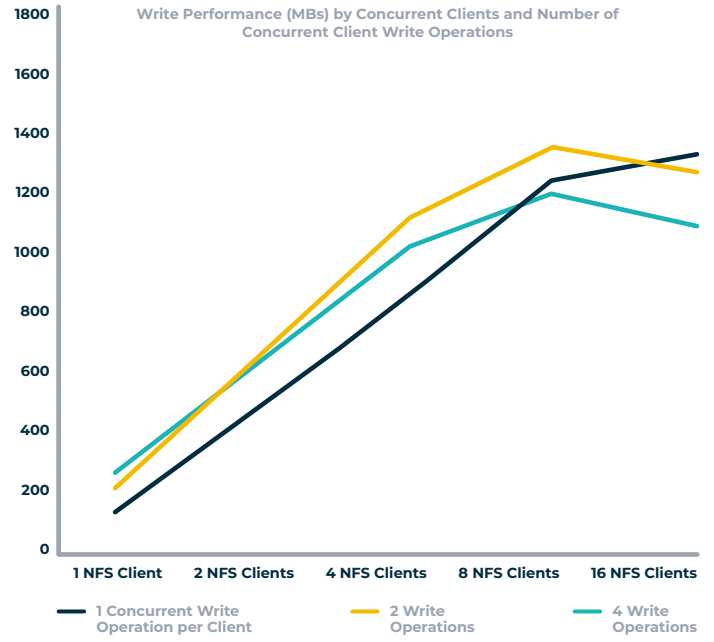


S3 WRITE RESULTS



SINGLE SwarmFS INSTANCE PERFORMACE TESTING





SINGLE SwarmFS INSTANCE READ RESULTS

Best single-read operation result: single physical NFS client using single DD read operation against a single SwarmFS instance:

READ: 349 MB/s

Best multi-operation result: 8 physical NFS clients using one dd read operation per client (8 concurrent reads) against a single SwarmFS instance:

READ: 1,562 MB/s (195MB/s average per read operation)

SINGLE SwarmFS INSTANCE WRITE RESULTS

Best single-write operation result: single physical NFS client using single copy write operation against a single SwarmFS instance:

WRITE: 392 MB/s

Best multi-operation result: 8 physical NFS clients using one copy write operation per client (8 concurrent writes) against a single SwarmFS instance:

WRITE: 1,280 MB/s (160MB/s average per write operation)

CONCLUSION

The performance numbers presented in this paper are the result of actual benchmark tests executed on production infrastructure at scale. The high levels of performance achieved resulted from the elimination of common networking and storage bottlenecks that are not present in the STFC leaf-spine CLOS network and the Swarm Object Storage platform.

Swarm does not require the use of any front side-caching mechanism or load balancers. Swarm's simple, flat, architecture makes it low latency, self-balancing and highly symmetrical. This enables Swarm to handle many concurrent requests in parallel yielding the full throughput potential of all the drives in the system. As detailed, these benchmarking results exceeded performance minimum requirements by 60% or more in all cases.

In an HPC environment where high aggregate throughput as well as durability and accessibility of data over a common protocol (such as S3 wrapped in a multi-tenancy framework) are required, Swarm represents a new storage paradigm. To meet these requirements, Swarm delivers:

- Read performance equivalent to parallel file systems for S3 throughput
- NFS and S3 access to the same objects
- Easy management of thousands of tenants and billions of objects
- Simple internal and external file access and file sharing

To learn more, visit STFC, JASMIN, and DataCore Swarm.

0621