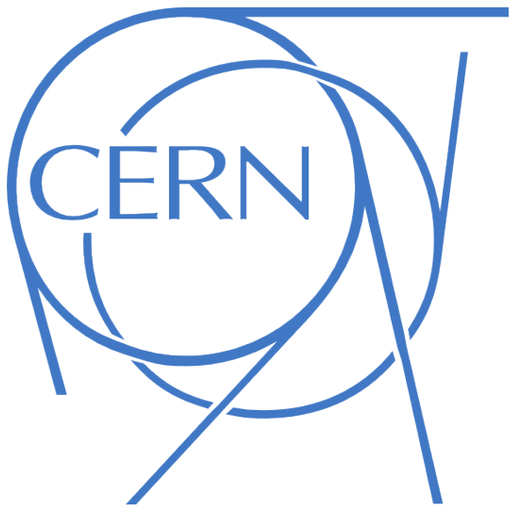
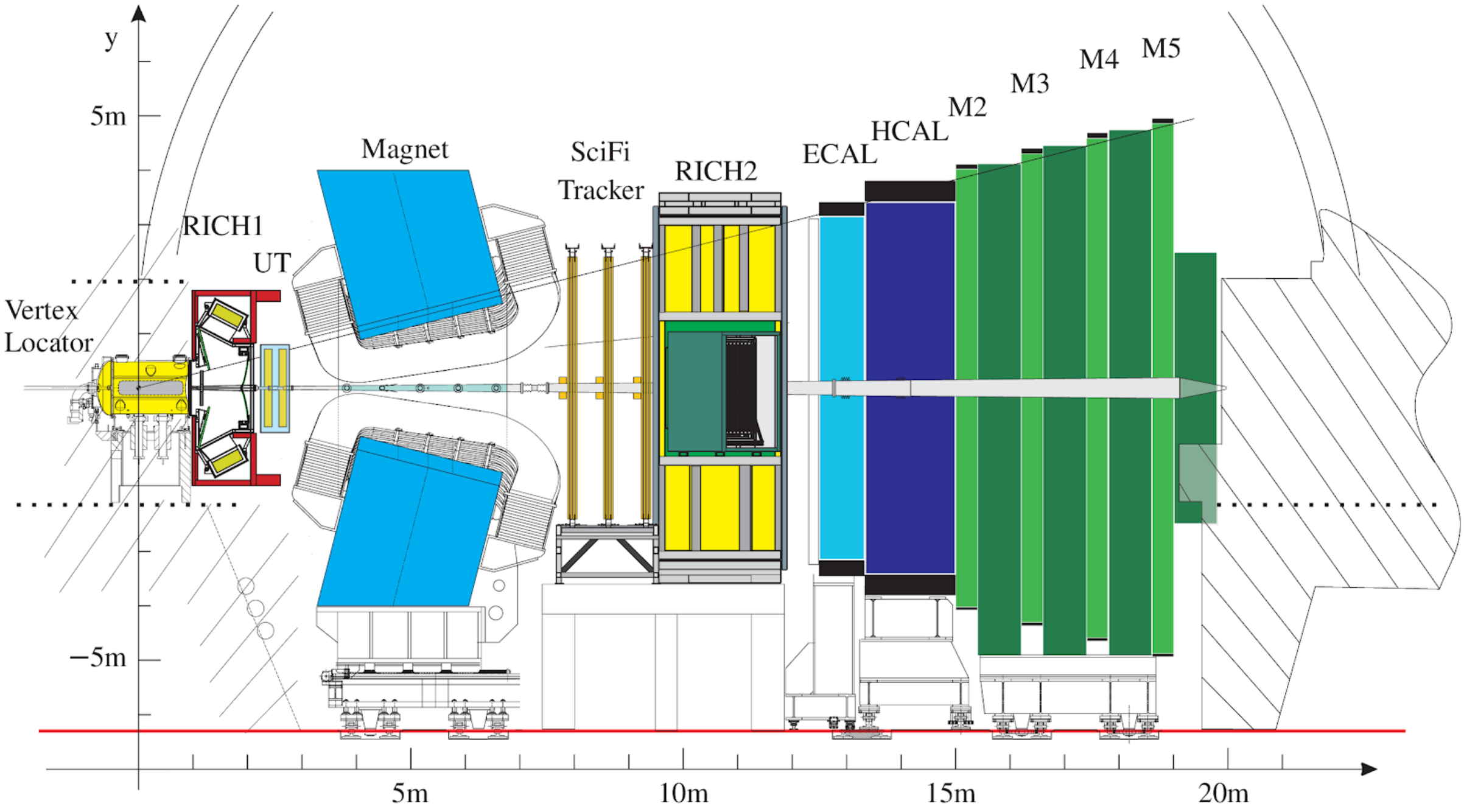
The LHCb HLT2 storage system: a 40 GB/s system made from commercial off-the-shelf components and open-source software



Pierfrancesco Cifra for the LHCb Online team

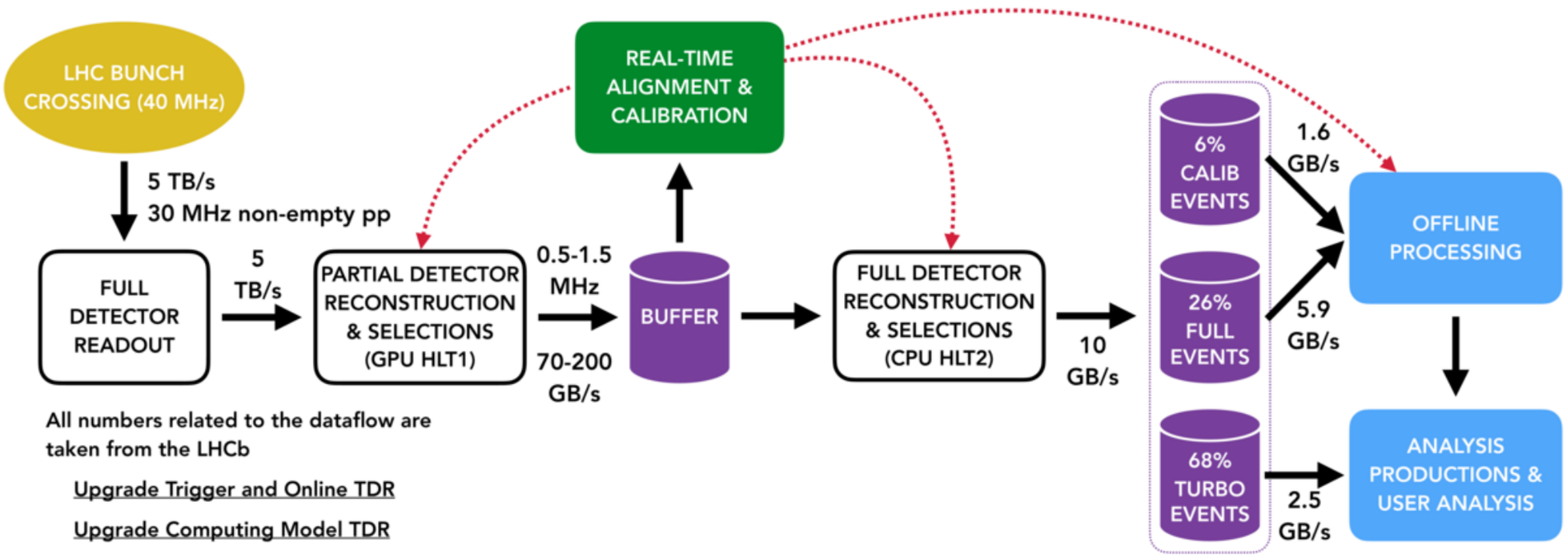
23rd Virtual IEEE Real Time Conference, Aug 1-6 2022

The LHCb Experiment



The LHCb High Level Trigger system

- The detector produces data at a rate of ~ 32 Tb/s. It will be reduced by a purely software-based multi-stage trigger system run on GPGPUs and CPUs
- In the first step data rate is reduced from 5 TB/s to 70-200 GB/s and stored temporarily on the HLT1 buffer
- In the second step ~ 4000 nodes are responsible for reprocessing and reconstruct the data to produce offline like quality files for future physics analysis and store them into the HLT2 buffer at a rate of 10 GB/s
- The last step is to read back those files from the HLT2 buffer, pack them in the correct physics stream order and move them permanently to the long term tape-storage area



HLT2 storage, what do we need?

- Aggregated throughput of 20 GB/s
- I/O pattern mostly sequential (10 GB/s in and 10 GB/s out)
- Fault-tolerant system (redundancy a the level of disks and performance effectiveness on degraded mode)
- Enough disk space to cover several days of data acquisition
- POSIX-compliant file system (preferably)
- Scalability
- Limited budget

Candidate solution

Among the several options and tests performed, taking into account the cost aspect of a commercial products, a software defined storage on commercial off the shelf hardware is the most suitable one

- Ceph 17.2.0 (Quincy)
- Open source
- Reliability
- Scalability
- Monitoring
- Large support given by the community

CEPH in nutshell

- CEPH is based on RADOS. Everything is an object
- Data can be accessed via LIBRADOS, block device, Rados Gateway or CephFS
- Possibility for erasure code or replica, depending on data effectiveness goal
- Object storage daemons (OSD): they collect and store all data on the physical devices (RocksDB back-end)
- Metadata servers (MDS): they manage the namespace and ensure security, consistency, and coherence (CephFS deployment)
- CRUSH: A data distribution algorithm used to store object replicas according to a configured replication policy and failure domain out of the box
- Monitoring with Grafana and Prometheus, disk failure-prediction

Hardware used

The system consists of 12 servers and 12 disk enclosure with the following specifications

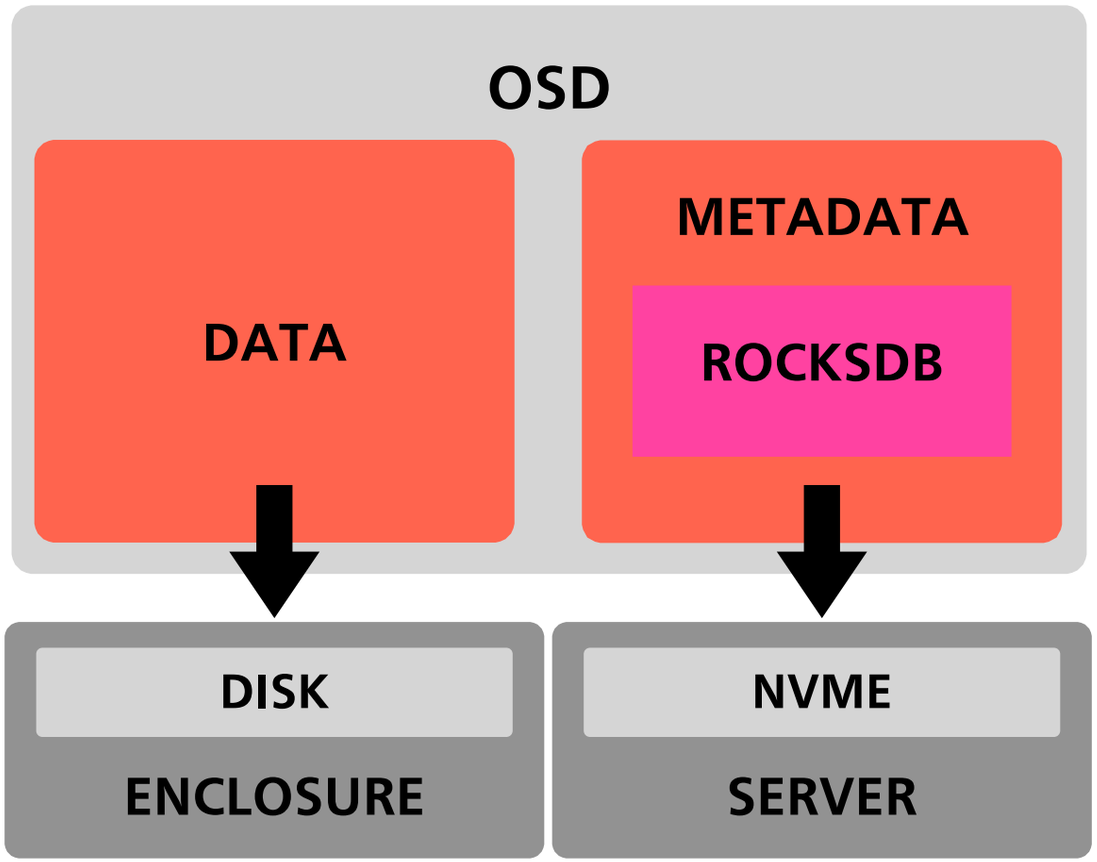
SERVERS SPECIFICATIONS	
Server	Supermicro SYS-220BT-HNTR
CPU	Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz
Memory	16 x 32 GB DDR4-3200 ECC
Disk	4 x 1.9 TB NVMe Samsung + 1 TB SSD Samsung EVO
OS	RHEL 8, Linux 4.18
Network	Mellanox MT28800 ConnectX-5 Ex, PCIeGen4 x16
Storage	Broadcom LSI SAS38xx Fusion-MPT 12GSAS

DISKS ENCLOSURE SPECIFICATIONS	
Enclosure	Western Digital Ultrastar Data 102, 2 Disk controllers
Disks	60 x Ultrastar DC HC550 18 TB

- 96 Gb/s SAS full duplex node to disk enclosure connection
- 200 Gb/s full duplex network connections per node

OSD Deployment

- OSDs are deployed using a combination of slow and fast devices
- RocksDB allow us to split data and metadata among different devices
- Data is placed on the hard disks
- Metadata are placed over NVME partitions



CEPH File System

- The 4000 clients access the buffer through 400 Gb/s aggregated network using CephFS Kernel module
- The LHCb file system it is configured with two RADOS pools, one for data and one for metadata
    - The bookkeeping is managed by metadata server which expose a POSIX-like interface to the clients

Data Pool

The data pool is an erasure code pool

- Small overhead needed to ensure data protection
- More CPU power needed in order to compute the encoding chunks

Erasure Coding profile configuration

- The erasure code configuration is 8+2 (8 data chunks, 2 parity chunks) using the Reed Solomon technique. The usable space decrease from 11 PB to 8.8 PB
- 1 stream per client, object size 64 MB
- Data and coding chunks are computed in the CPU on packets of 16 kB size each
- The failure domain is at the level of disks and nodes (10 chunks over 12 servers for increased redundancy)

Metadata pool

- The metadata pool is used for managing file metadata in a Ceph File System (inodes, dentries, file system hierarchy...)
- The metadata pool space need is very low (12 TB raw space, 3 TB usable) for keeping track of a large amount of objects, but what is important is the I/O capabilities
- The pool is replicated (x3)
- This is placed on the SSD devices of the servers
- 1 metadata server active, 4 in standby

Tests

The system was tested in different scenarios and under different conditions

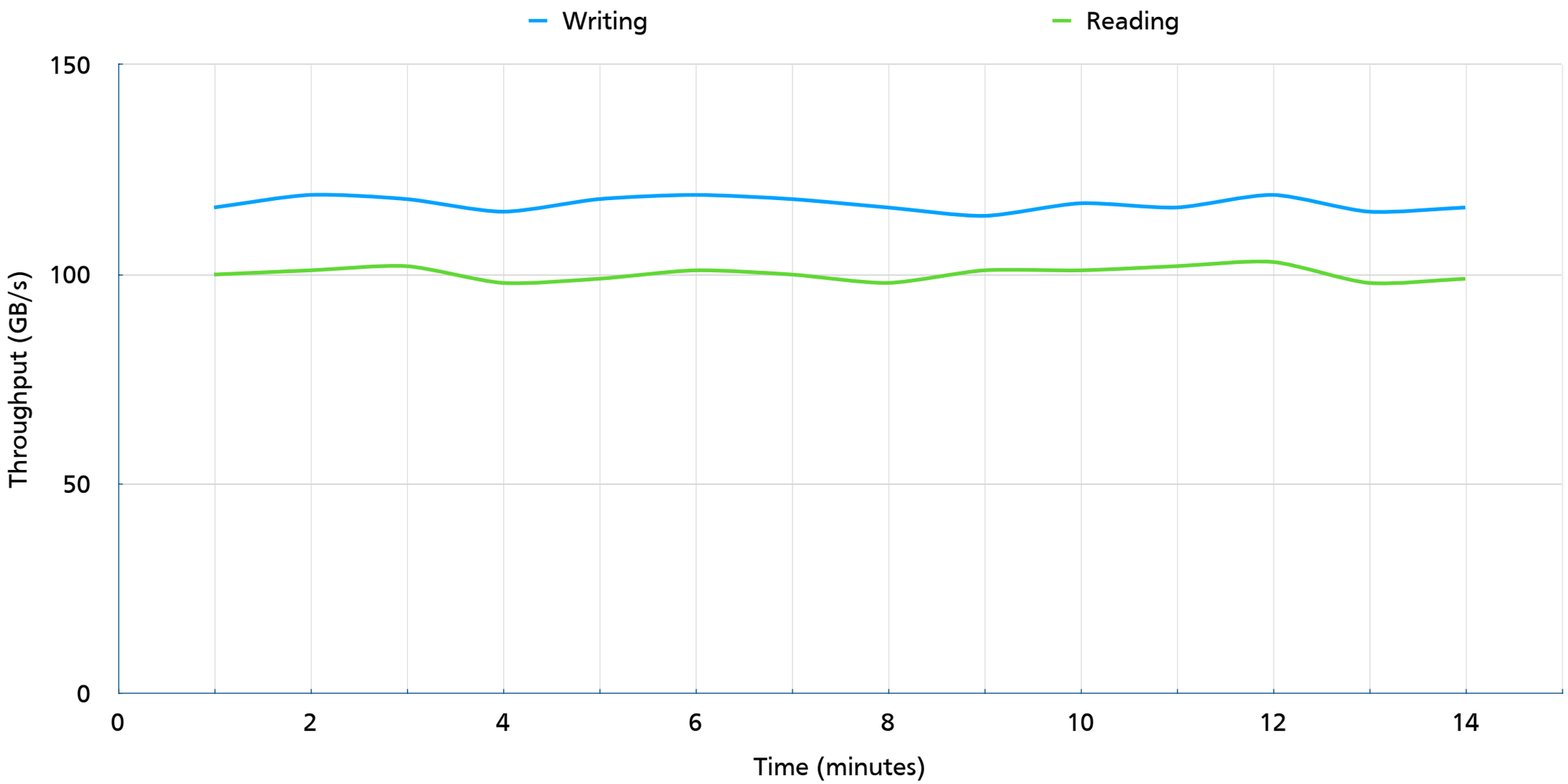
- I/O operation were performed both from a small and a large cluster of nodes
- The tests were performed in 100 % writing, 100 % reading and in a mixed scenario with 50 % writing and 50 % reading at the same time
- Degraded mode performance
- Performance vs occupancy vs defragmentation

Tests were performed writing and reading files with ***dd*** command-line utility

- Writing: *dd if=/dev/zero of=/HLT2/filename-xyzw bs=64M*
- Reading: *dd if=/HLT2/filename-xyzw of=/dev/null bs=64M*

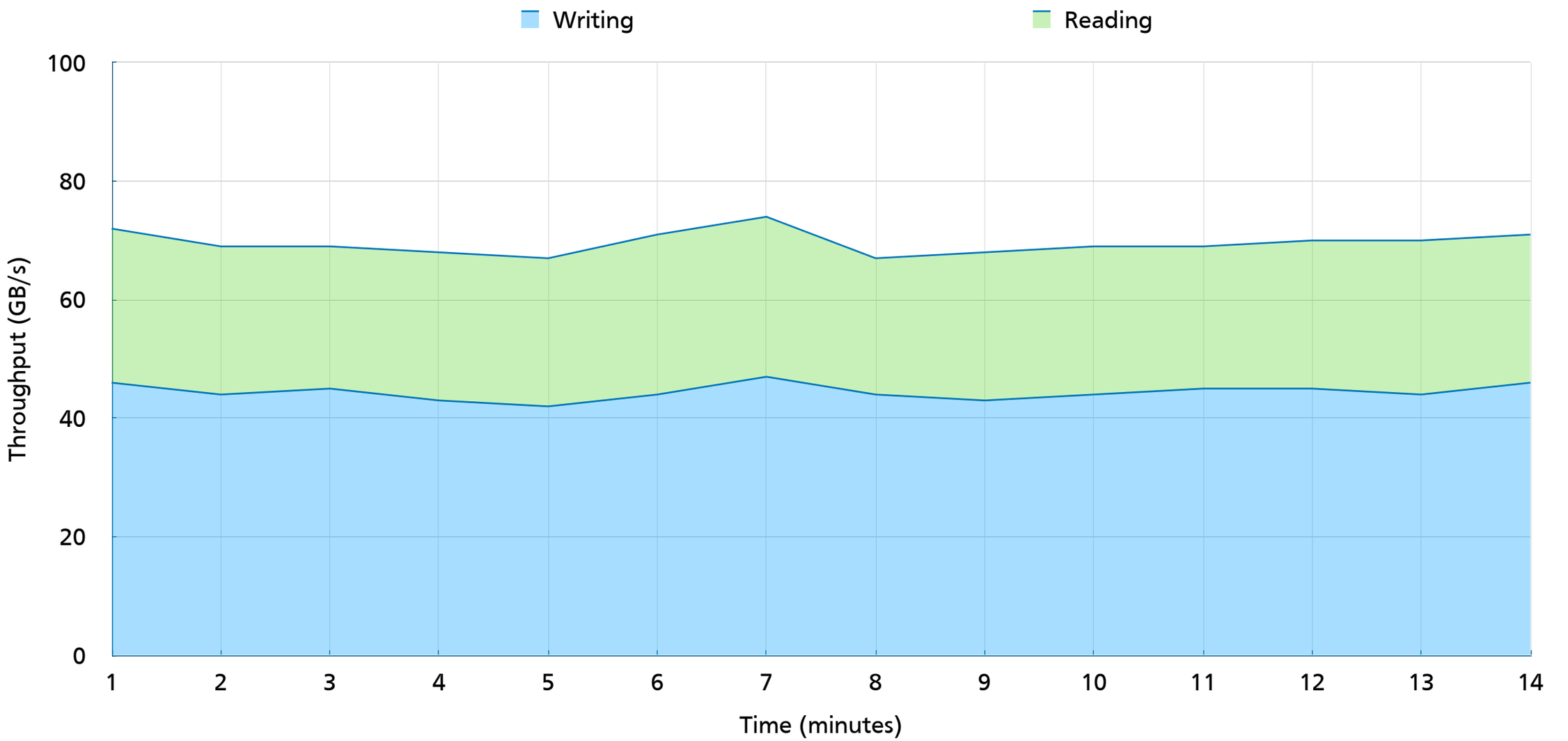
Results

- Uncontrolled I/O, 100 % write and 100 % read, 20 % occupancy
- Writing and read with 16 streams per server from 12 servers



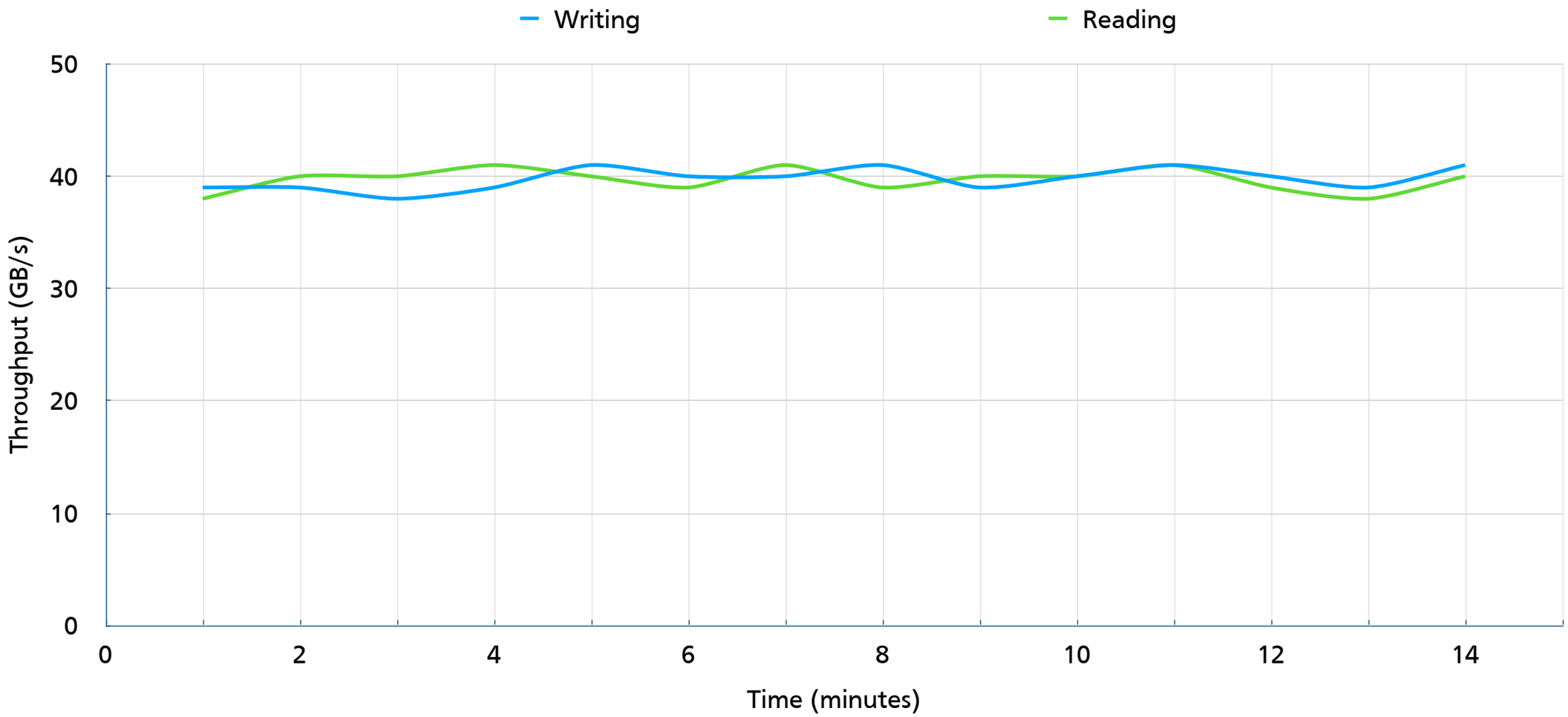
Results

- Uncontrolled I/O, writing and reading concurrently, 20 % occupancy
- Writing with 16 streams per server from 6 servers and reading with 16 streams per server from 6 servers



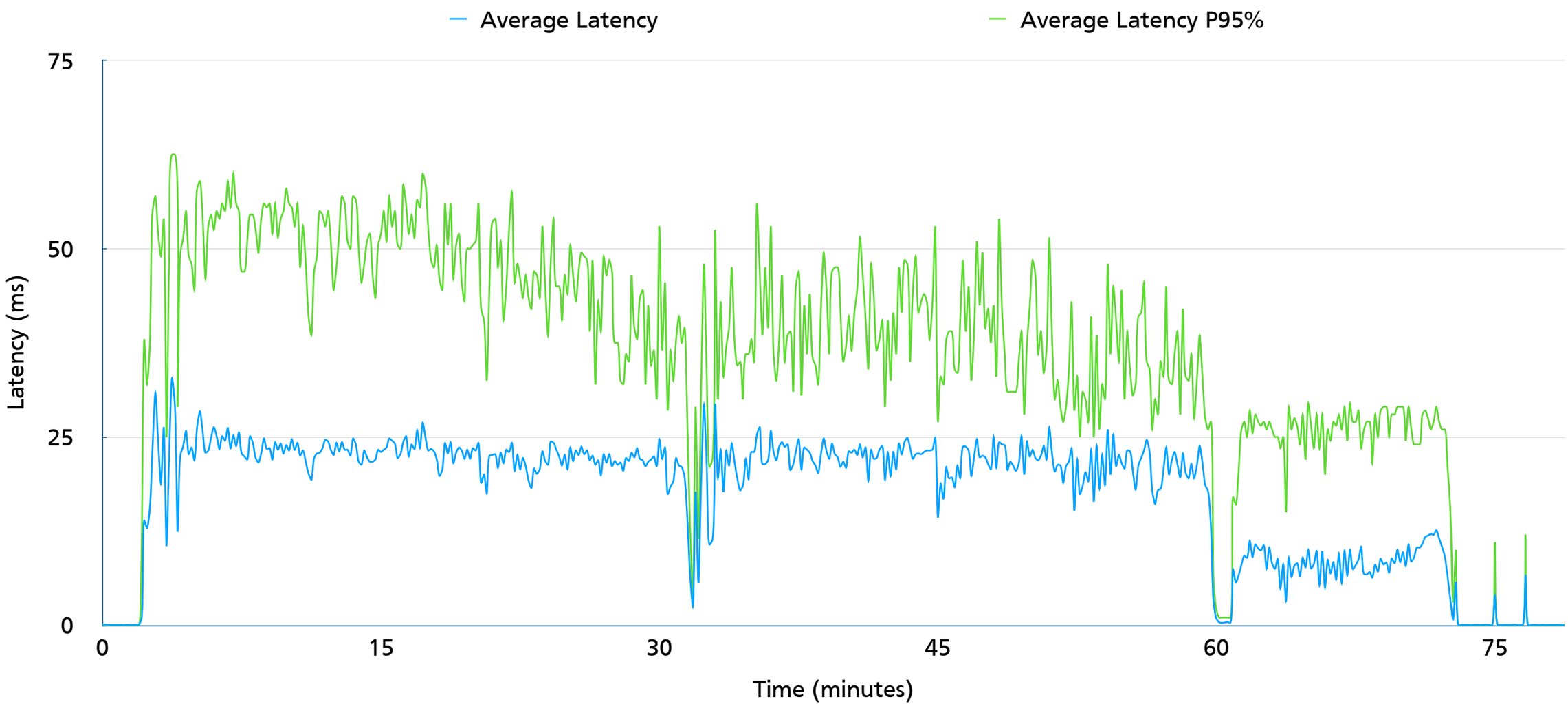
Results

- Controlled I/O from ~ 2000 nodes, 100 % write and 100 % read, 20 % occupancy
- 1 stream per client limited to 32 MB/s
- Network connectivity limited to 40 GB/s



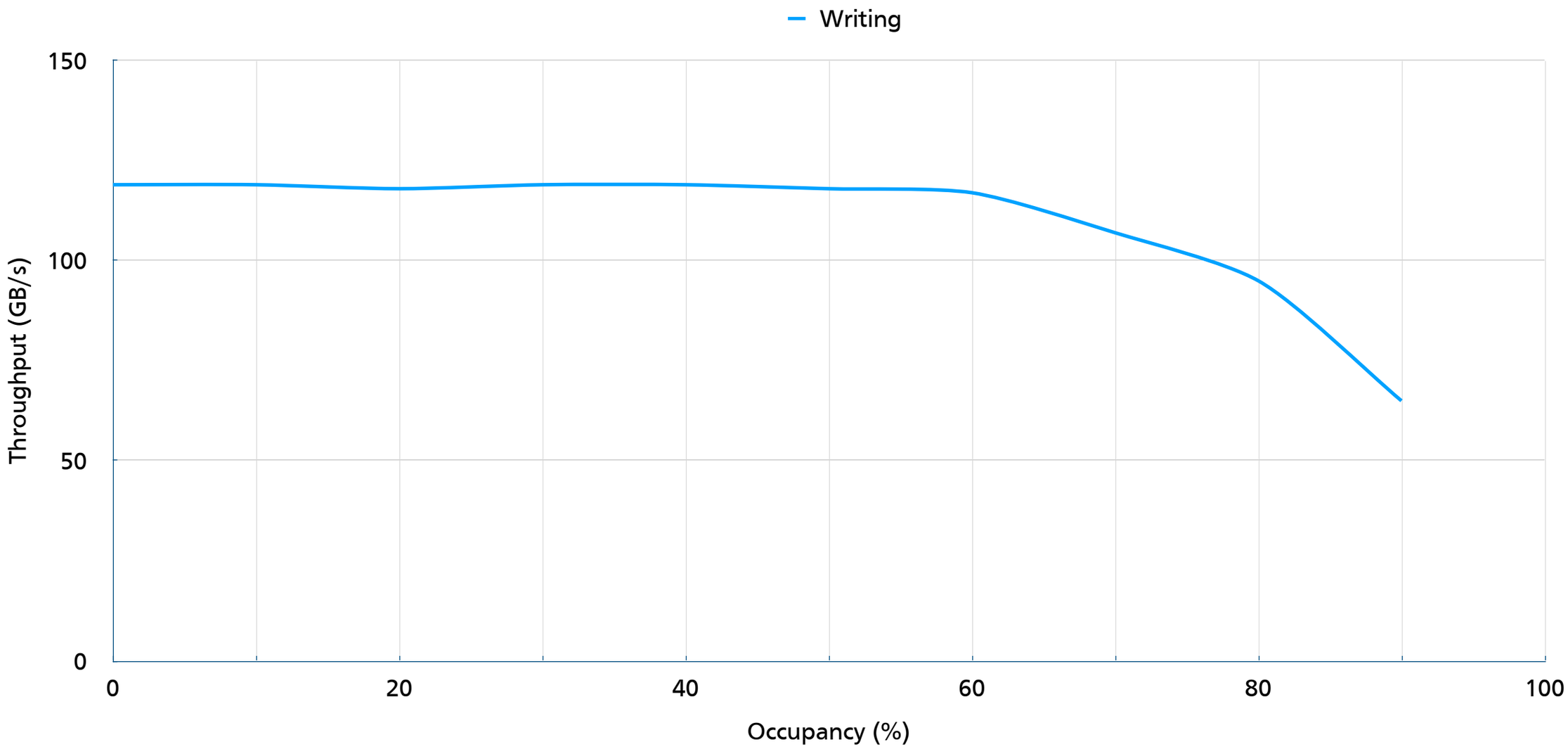
Results

- OSDs average latency and 95 % percentile latency, 20 % occupancy
- Writing and reading with 16 streams per server from 12 server, throughput limited to 40 GB/s



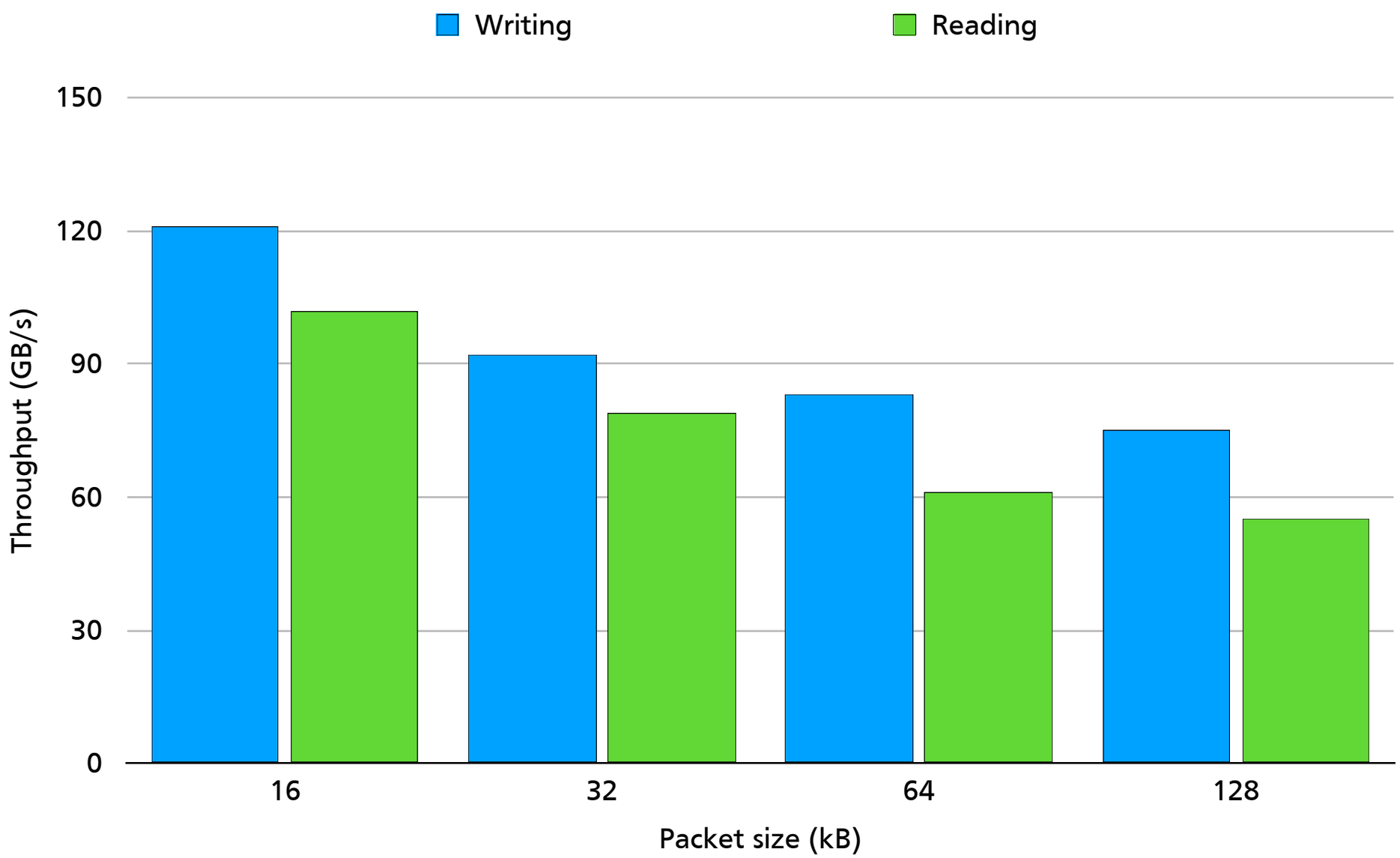
Results

- Performance versus occupancy, uncontrolled I/O
- Writing with 16 streams per server from 12 servers



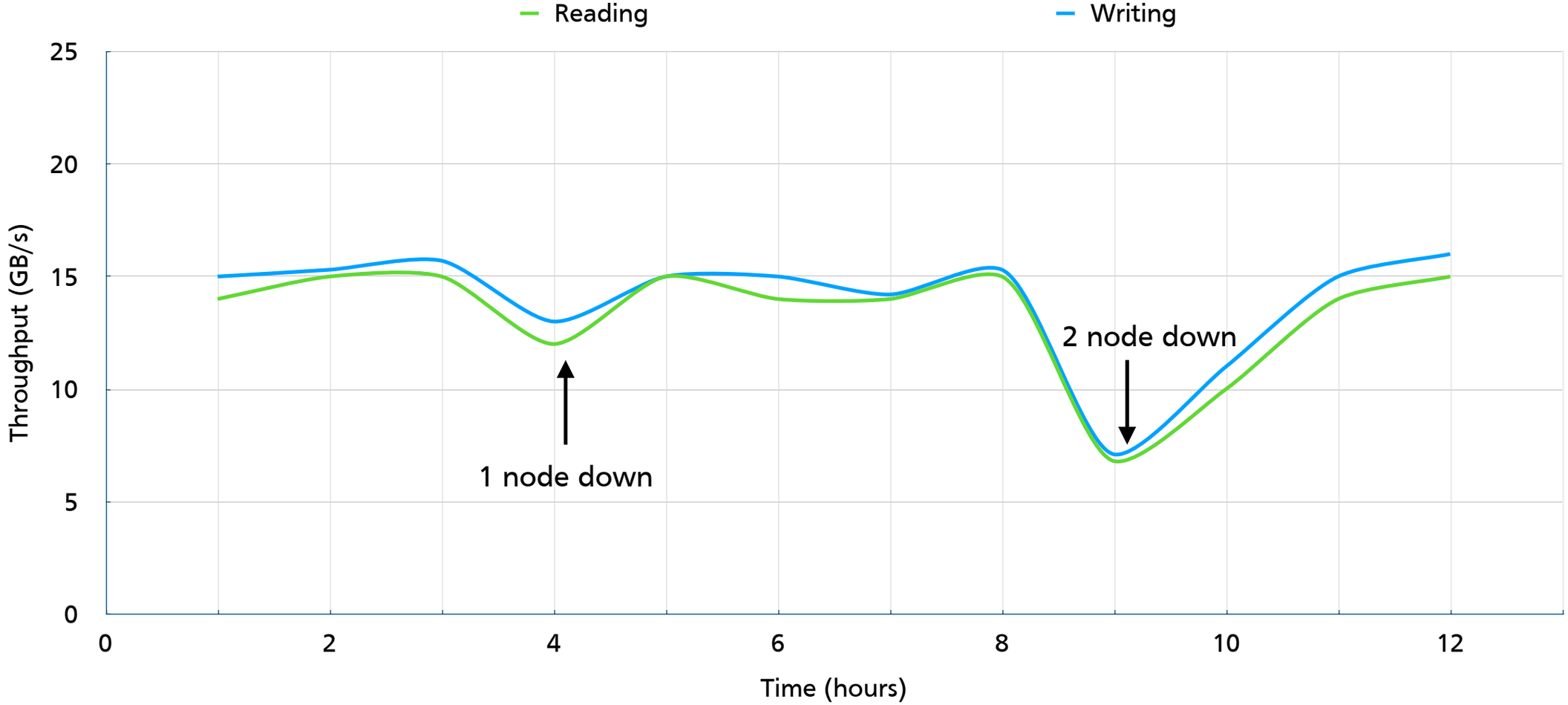
Results

- Max throughput measured according to the erasure code packet size
- Writing and reading with 16 streams per server from 12 servers, 20 % occupancy



Real case/disaster scenario

- The system exceeds requirements, what about degraded mode?
- Reading and writing concurrently with 16 streams per server from 12 servers, 50 % occupancy
- I/O rate limited in order to simulate the data taking scenario



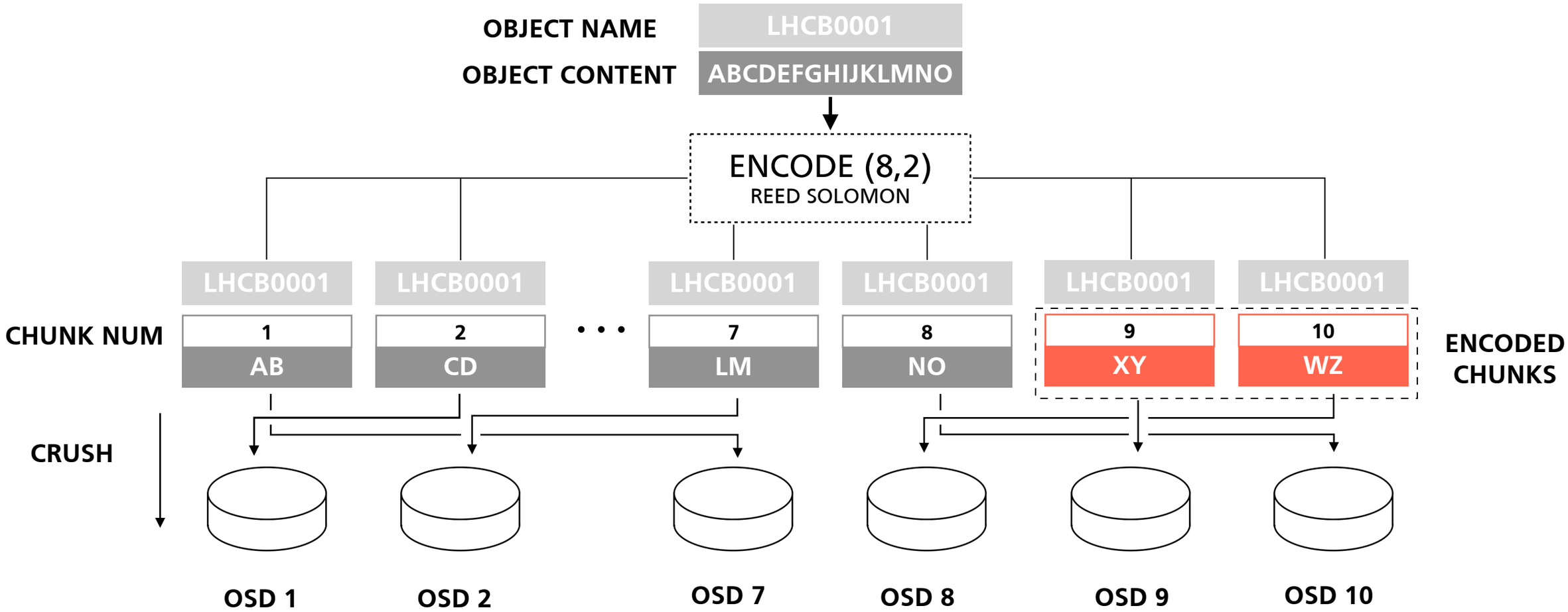
Conclusion

- The initial requirement of an aggregated throughput of 20 GB/s was achieved and exceeded
- The infrastructure shows a very good scalability by serving thousands of clients concurrently
- Robustness and availability are fully satisfying the performance level required even during degraded mode
- In the current configuration the limiting bottleneck can be identified in the number of spindles per server or the amount of CPU power consumed during the writing operation. For increased performance it would be necessary to add more resources but by design the network connectivity is capped at 40 GB/s
- The cost per usable TB is 45 USD

THANK YOU FOR YOUR ATTENTION

BACKUP

Erasure coding



The full infrastructure