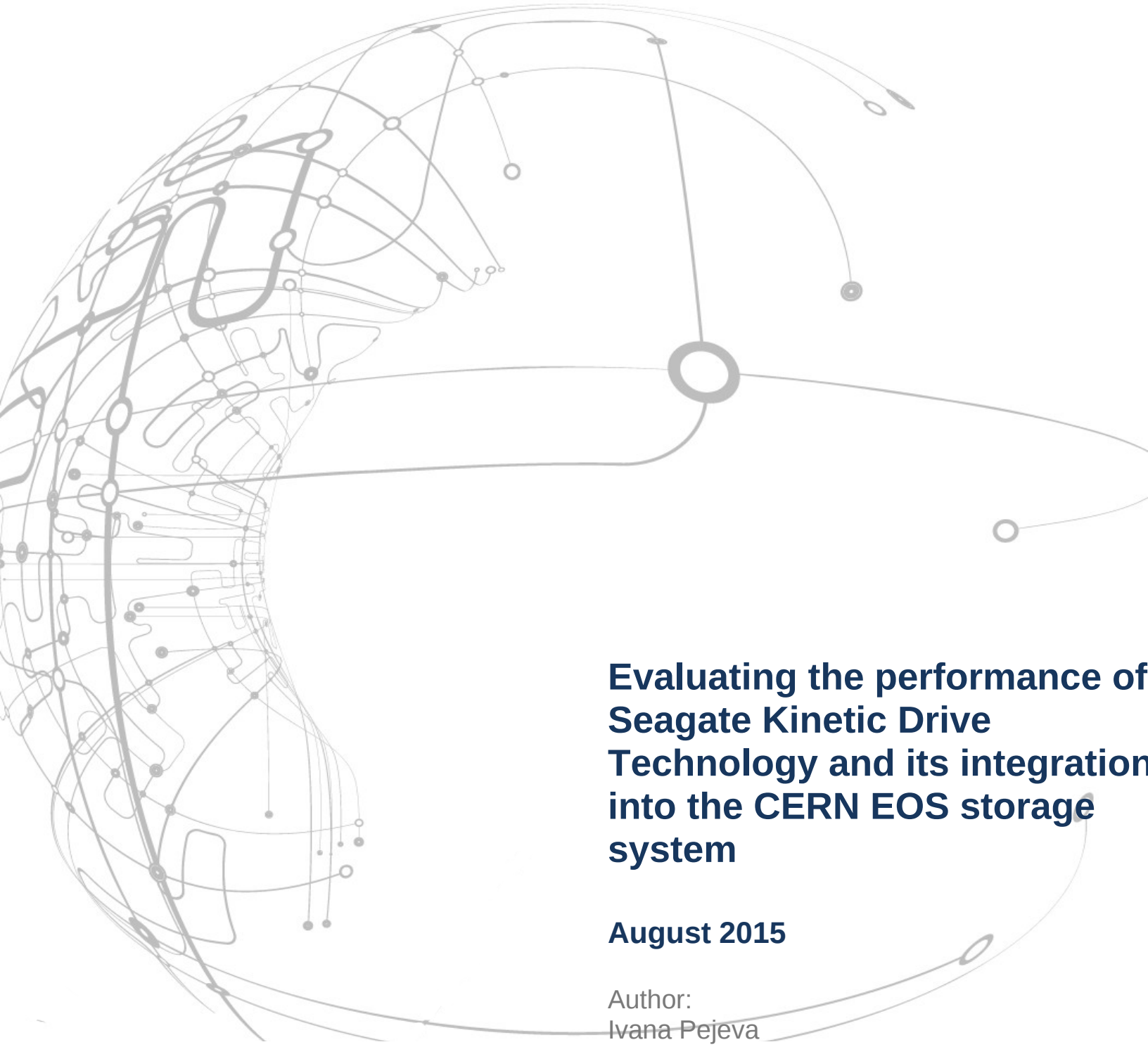


# Evaluating the performance of Seagate Kinetic Drive Technology and its integration into the CERN EOS storage system

**August 2015**

Author:
Ivana Pejeva

Supervisor:
Andreas J. Peters

**CERN openlab Summer Student Report 2015**

CERN openlab

# Abstract

The big amount of data produced by CERN experiments at the Large Hadron Collider (LHC) needs to be efficently stored and analyzed. Because of the constant increasing data volume the essential function at CERN is archiving the vast quantities of data.

The Data and Storage Service Group in the IT department at CERN is operating and evaluating different cloud storage technologies to ensure that all incoming data from experiments can be stored reliably in a cost effective way.

One of the main storage systems used and developed at CERN is EOS [1], a multi-petabyte disk storage. A recent R&D project aims to integrate the Seagate Kinetic drive technology [2] as a promising storage solution for the future. Seagate Kinetic offsers ethernet enabled disk drives with an object storage API.

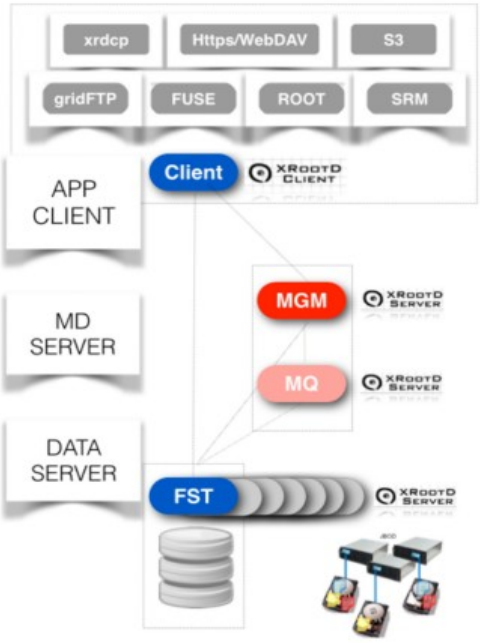
The main goal of this project is a performance evaluation of Seagate Kinetic drive technology and its integration into the CERN EOS storage system.

# Table of Contents

# 1 Introduction

### EOS Storage System

*EOS* is a storage system developed in CERN that is used for storing physics analysis data produced at the LHC. The main goal for EOS is to provide fast and reliable disk only storage technology for CERN LHC use cases (see [3]).




EOS is based on three components: management server (MGM), message queue (MQ) and file storage services (FST). All components are implemented as plug-ins for the *XRootD* storage server.

Core of the implementation is the *XRootD* framework providing a feature-rich remote access protocol. A fundamental concept of EOS is to use a set of single disks (JBOD) as storage media without the need to build local RAID arrays. Default mode of operation is to store files with two replicas. Files can be accessed via native *XRootD* protocol, a POSIX-like *FUSE* client or *HTTP(S) & WebDav* protocol.

### Deployment

Meta data services are deployed as six active-passive pairs (one for each EOS instance) with real-time failover capabilities on high-memory nodes at the Meyrin center. File storage services are deployed on approx. 1.400 server nodes with attached storage (up to 50 disks per node) distributed in both computer centers (the CERN Center in Meyrin/Switzerland and the WIGNER Center in Budapest/ Hungary).

### Directly Attached Storage (DAS) vs Ethernet Drive Technology

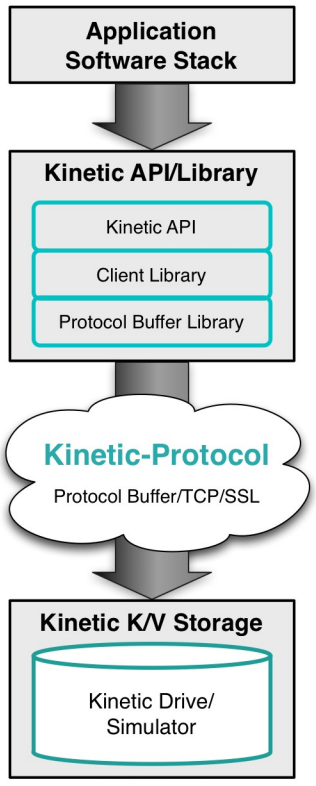
A key difference between DAS and Ethernet Drive Technology is scalability. Disadvantage of DAS architecture is its limited scalability, a Host Bus Adaptor can only support a limited number of drives. At the same time the ethernet drive technology allows servers and storage to be scaled independently, cloud data centers can add servers and storage at entirely different rates.

With Direct Attached Storage architecture, clients must connect directly to the server that contains the storage in order to access the data. On the other hand the ethernet drive architecture is eliminating the need for storage servers, allowing storage applications to talk directly to a big number of hard drives over Ethernet.

As the number of severs increases, the complexity of managing storage on multiple servers can escalate rapidly and increases costs. Ethernet Drive Technology has the potential to cut costs by eliminating servers and staff admin time.

# 2 The Seagate Kinetic Open Storage Platform

Seagate Kinetic hard drives speak Ethernet rather than SATA, SAS or fibre channel. This enables the HDDs to talk directly to other devices and other components in the system, rather than going through intermediary devices, controllers or other compute nodes.




*Object storage*

Another important characteristic of object storage is a key/value storage interface enabling clients to communicate objects to the devices, rather than blocks. Object based storage organizes data into flexible-sized data containers, with the approach of addressing and manipulating discrete units of storage called objects. The key semantics for object storage are PUT, GET, and DELETE.

Scale-out storage systems use objects because objects enable systems to scale - and with Seagate Kinetic Drives, objects are available at the drive level.

*Kinetic API*

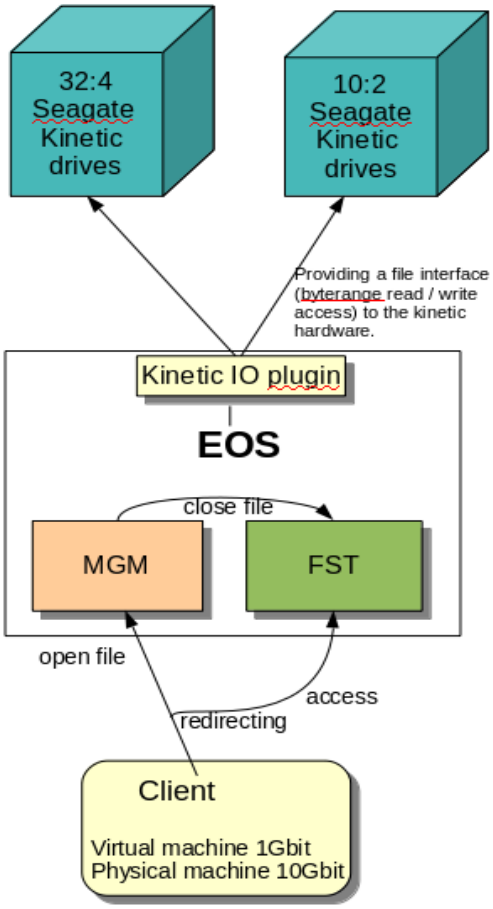
Access to a Kinetic drive by a client is through an application programming interface (API). It consists of a set of methods that behave as remote procedure calls (RPC). There are specific management methods, some of which change the drive state such as access constraints.

*Protocol*

Kinetic drives use an open-source object-based storage protocol, which melds meta data with data, allowing scale-out network-attached storage and big data file applications to access data regardless of its location in a storage pool.

Kinetic Client applications can communicate with a Kinetic Device by sending messages over a network using TCP. Each individual message is called a “Kinetic Protocol Data Unit” (Kinetic PDU) and represents an individual request or response. For example, a Kinetic Client may send a message requesting the value associated with a particular key to a Kinetic Device. The device would respond with a message containing the value.

# 3 Kinetic Integration with EOS




EOS storage servers (FST) provide a plug-in mechanism to replace IO to local attached disks with remote protocols. For the comunication between the EOS FST and the Kinetic Drives, a Kinetic IO plug-in has been developed.

The Kinetic IO plug-in abstracts a whole cluster of disks as a single device - the same way a RAID system abstracts an disk array as a single virtual block device.

Files are chunked into a set of objects. The plug-in allows to use an arbitrary Reed-Solomon code to split and encode objects over $m+k$ drives to provide fault taulerance and improve single object performance using striping over $m$ data chunks and $k$ parity chunks.

The implementation is based on the Intel ISA erasure encoding library [4] with a cauchy matrix. For each object a CRC32 block checksum is computed and stored as meta data.

CRC32 is currently the natively supported Kinetic object checksum. In the future this might be replaced with a hardware accelerated CRC32C checksum.

Each Kinetic cluster can contain more than $m+k$ disks.

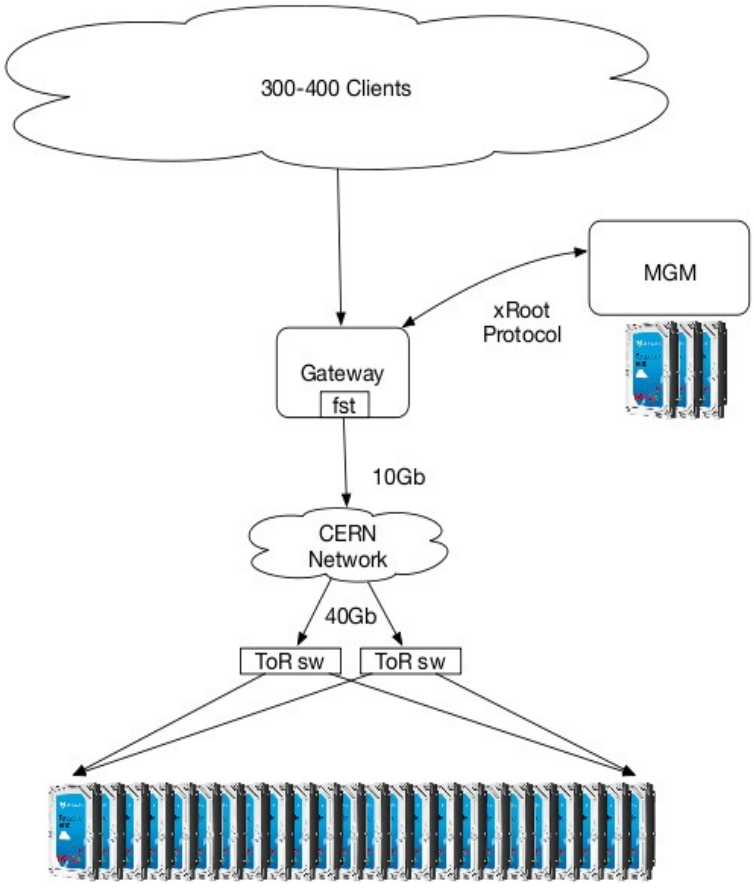
The implementation currently uses an round-robin algorithm to select a subcluster of m+$k$ disks. This might be optimized in the future.

# 4 Test Deployment

For the Kinetic Drive Cluster two *m+k* configurations were used :

- 32:4 32+4 encoding : one can lose 4 drives in a cluster of 36 drives without data loss, the remaining 32 drives allow to reconstruct each object - the space overhead is 12,5 %
- 10:2 10+2 encoding : one can lose 2 drives in a cluster of 12 drives without data loss, the remaining 10 drives allow to reconstruct each object without data loss – the space overhead is 20 %

For comparisons a conventional EOS configuration (EOS DEV) with directly attached disks was tested storing two replicas on two individual FSTs. All FSTs were connected via 10GE.
As clients one virtual machine with 1GE and a physical machine with 10GE have been used for comparison – the reason is that many virtual batch and desktop nodes at CERN are connected with 1GE.



# 5 Performance Benchmarks

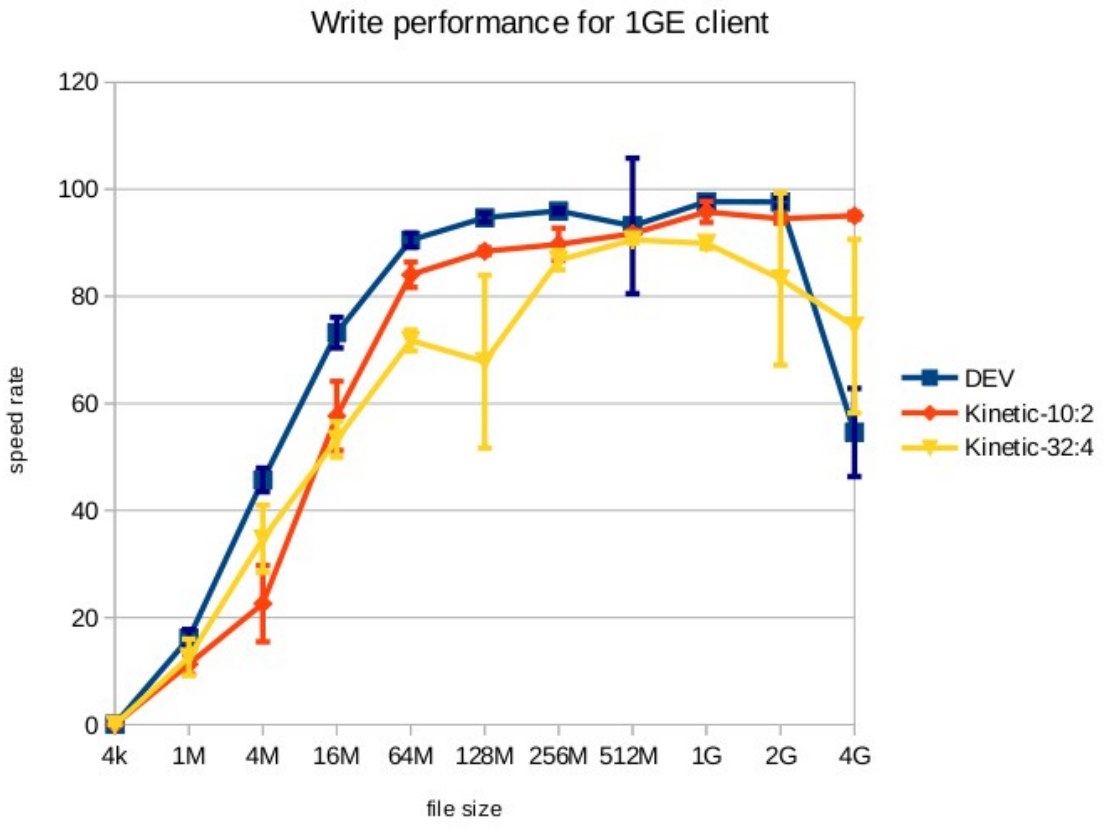
The aim of these tests is to study the behaviour of data read and write performance with typical access patterns. The main motivation is a scalability evaluation of the Kinetic Drives Technology with the interest to add storage as the need for capacity grows for minimal cost trading some performance parameters for scalability.

The tests reproduce the major data access patterns observed at CERN:

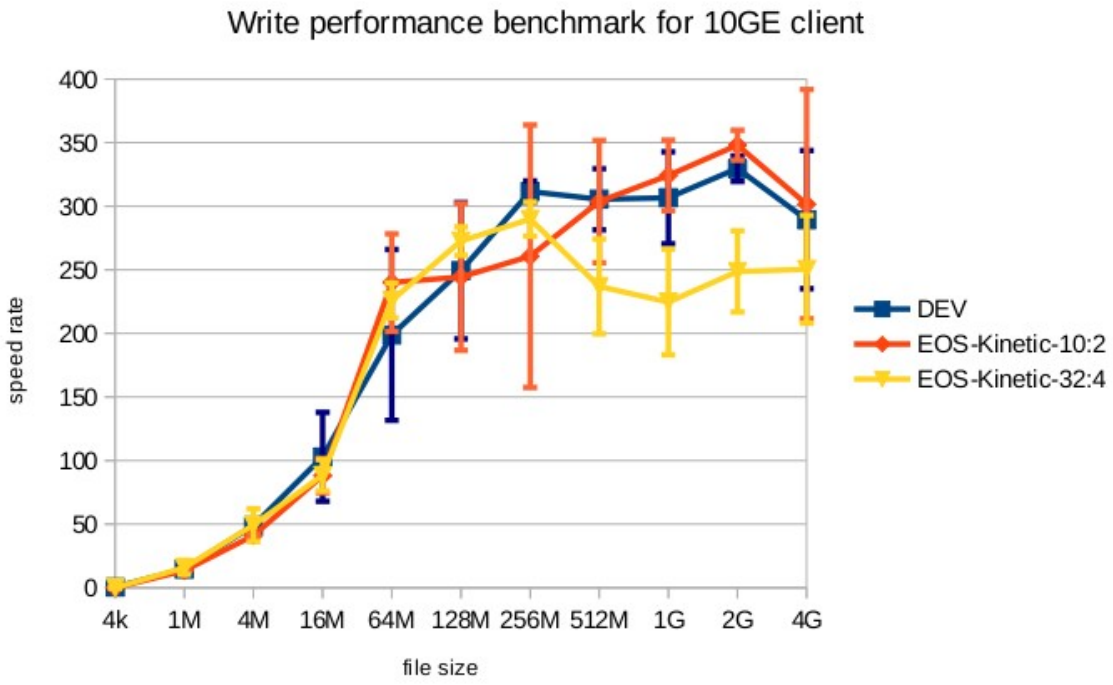
- data creation – sequential file writing
- data analysis – byte-range file reading

## 5.1 Sequential Write performance benchmark

Normal performance expectations for sequential read and write for the kinetic drives are currently 50MB/s. For conventional attached disk drives this is approx. doubled (~100MB/s). While reading typically involves synchronous IO, writining uses in both implementations a write-back cache without forced flushing on *close*. Therefore we expect a higher write performance than the nominal drive rate when files fit into the write-back cache. The bottleneck in this measurement is due to network and buffer configurations – not dominated by single drive performance.



Files were sequentialy stored on the three configurations: Kinetic-32:4, Kinetic-10:2 and the conventional DEV setup (two replicas). The file size was varied from 4KB to 4GB and each file was uploaded 10 times using XRootD protocol and the *xrdcopy* command. The test series was run from the 1GE and the 10GE client.




The graph for the 1 GE client shows that a plateau of upload rate (~90-100 MB/s) is reached for files larger than 64MB. There is a systematic few percent decrease of performance of Kinetic-10:2 vs. the conventional disk layout DEV. And a second few percent effect when changing from Kinetic-10:2 to Kinetic-32:4. These effects are only visible for file sizes larger than 64MB.

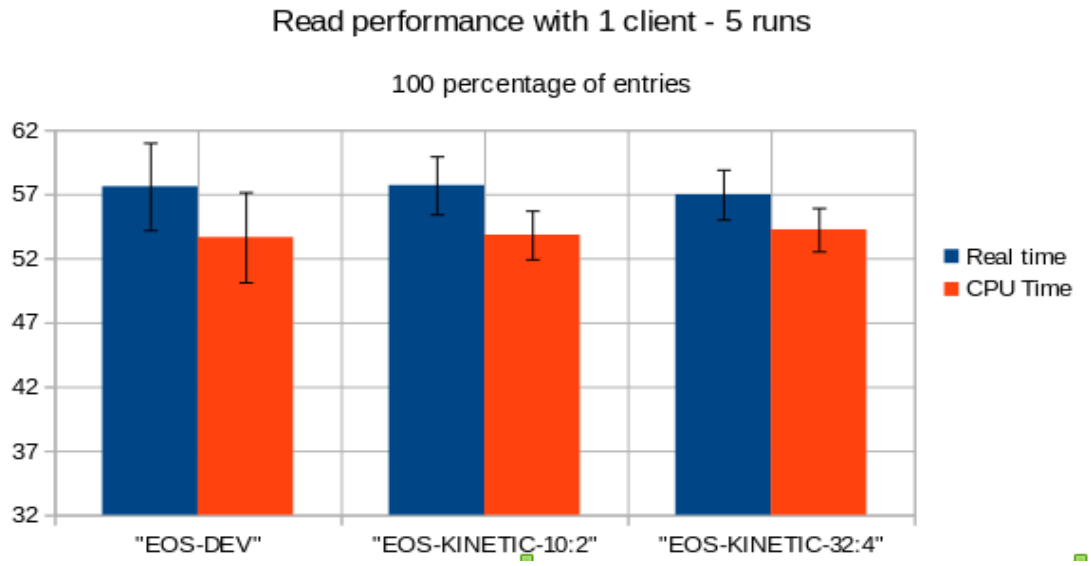
For a 10 GE client a plateau is reached with files larger than 256MB (~250-350 MB/s). There is no visible difference between the DEV and Kinetic-10:2 configuration, while the Kinetic-32:4 configuration involves a systematic performance loss around 10% for large files.

It should be mentioned that background network activity on the switches where client and server are connected is unpredictable. This can be responsible for a systematic shift in one or several of the test series. Taking that into account the test results are essentially compatible for all three configurations and a small effect due to the additional encoding and checksumming of the data for Kinetic configurations is expected.

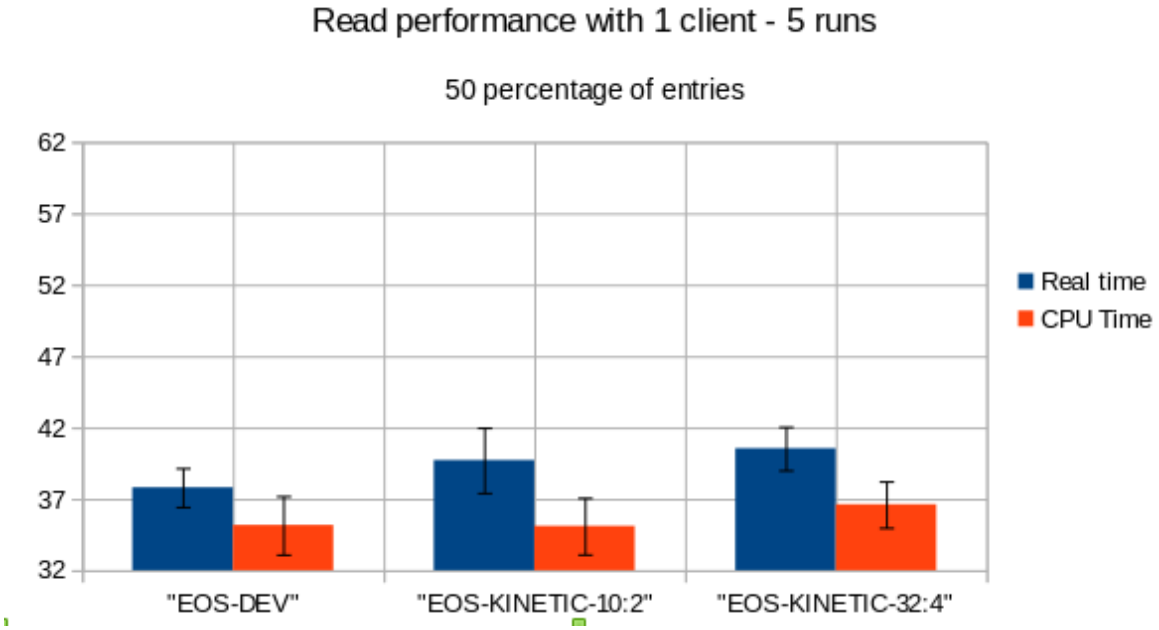
## 5.2 Read performance benchmark using ROOT

ROOT [5] is an Object-Oriented data analysis framework for HEP data processing that offers a common set of features and tools, mostly used by physicist for analyzing the data of their experiments. An existing multi-client ROOT benchmark [6] is used and modified for evaluating the read performance of Seagate Kinetic Drives. This benchmark offers a way to change the configuration in order to use it to evaluate the performance of the Kinetic Drives Technology. It allows to change the read volume size, the percentage of entries or number of ROOT files that are accessed. The benchmark offers the usage of numerous physical and/or virtual machines with different features in order to measure the aggregated throughput until a given deadline. For this read performance benchmark, a file from the ATLAS experiment is used, which is accessed via XRootD protocol by all client hosts.

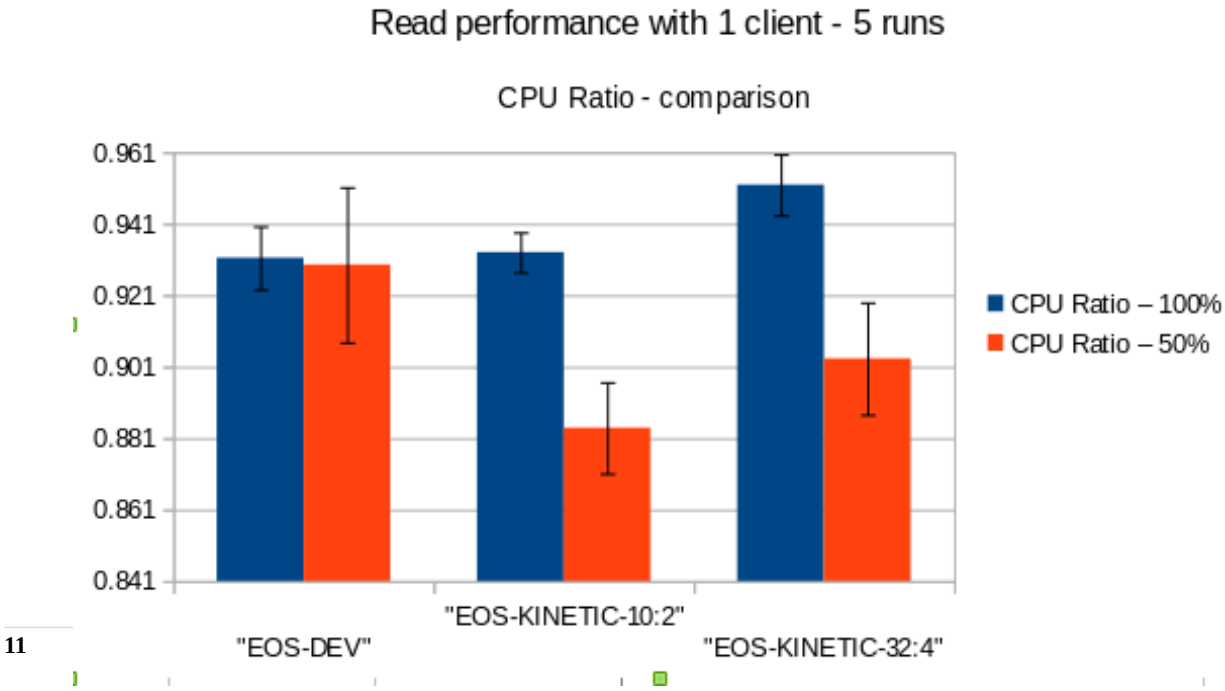
For the read performance benchmark one single 1GE client is used with the following parameter settings: 100 percentage of entries, 100 percentage volume, 30MB tree cache size. This test was performed 5 times and the results are plotted below.




The same test was performed for sparse access with 50 percentage of entries, accessing the ATLAS file with a single client . The others parameters were identical to the previous test.




The results for reading 100% entries are similar for all three test configurations and the analysis jobs run at ~92-95% CPU/Real time efficiency. When only 50% of entries are accessed there is a CPU/Real time efficiency drop of 2-4% visible for the Kinetic configurations. This can be accounted to the block chunking of the Kinetic plug-in where complete blocks of 10 or 32 MB have to be fetched once or several times instead of the few bytes of the required byte range – and additionally the drives synchronization/coupling when reading.



# 6 Summary and Outlook

It was demonstrated that the usage of Ethernet drive technology with Seagate Kinetic as a remote storage back-end in EOS has very little impact on the performance of the two most frequent use cases of data production and ROOT analysis used in LHC experiments. Due to the additional drive clustering done by the Kinetic plug-in into a single virtual FST mount point the scalability in terms of disk drives can be easily scaled-up by a factor of 100 allowing EOS to handle more than 1M disks per instance.

The same tests can now be continued with a multi-client & single-server set-up to evaluate client scalability behaviour. Each connected client involves to cache at least one complete block in the FST gateway (10 MB and 32 MB for the test configurations) – the available 64 GB of memory set an upper limit for the number of concurrent clients per FST for Seagate Kinetic configurations.
The scaling behaviour with respect to the the number of clients and the total throughput per FST is of particular interest here.

The last and most important point for future work is to compare the operational effort and total cost of ownership of direct attached disks and Ethernet disks to draw final conclusions.

# 7 Acknowledgements

I want to thank Seagate Technology as a member of the CERN *openlab* partnership program for their support and the opportunity to evaluate the Seagate Kinetic open storage solutions as a storage technology for LHC.
Finally I want to thank Seagate Technology, CERN IT and the DSS group for the possibility to participate in the CERN *openlab* summerstudent program 2015.

# 8 References

[1] "Exabyte Scale Storage at CERN", Andreas J. Peters, Lukasz Janyst
[2] Seagate Kinetic Disk Technology, *http://www.seagate.com/*
[3] "EOS as present and future solution for data storage at CERN", Andreas J. Peters, Elvin A. Sindrilaru, Geoffray Adde
[4] Intel Storage Acceleration Library, https://software.intel.com/en-us/articles/optimizing-storage-solutions-using-the-intel-intelligent-storage-acceleration-library
[5] ROOT Data Analysis Framework , *https://root.cern.ch/*
[6] "Using S3 cloud storage with ROOT and CvmFS" Maria Arsuaga-Rios, Seppo S. Heikkila, Dirk Duellmann, Rene Meusel, Jakob Blomer, Ben Couturier