Kinetic drive

Bingzhe Li
Consumption has changed

- It’s an object storage world, unprecedented growth and scale
- In total, a complete redefinition of the storage stack

What’s Kinetic drive

• Kinetic drive is the world’s first Ethernet-connected HDD with an open source object API designed specifically for hyperscale and scale-out environments.
• Seagate Kinetic HDD simplifies storage hardware and software architectures to reduce Total Cost of Ownership (TCO) and enable rapid response to a growing cloud storage infrastructure.

Kinetic drive features

• New class of Ethernet drives
• 2 Ethernet ports instead of 2 SAS ports
• Key/Value interface instead of SCSI
• Standard form-factor
• Open source key/value API and libraries
HDD Form Factor and Connector

Kinetic chassis conform to dimensions for the HDD drive form factor.

Kinetic drives repurposes the standard SAS HDD connector. The connector is defined by and conforms to the dimensions and specifications of SFF-8482 Rev 2.3. Its placement in the drive form factor is identical to SAS drives.

https://developers.seagate.com/display/KV/Kinetic+Open+Storage+Documentation+Wiki
The first Kinetic drive is a 4TB, 5900 rpm, 3.5" hard disk drive (HDD).

Compared to its conventional sister drive, the Kinetic drive implements the Kinetic API that enables key-value object storage. The Kinetic drive replaces the Serial Advance Technology Attachment (SATA) or Serial Attached SCSI (SAS) interface connections with the two 1-Gbps SGMII Ethernet ports, which enables direct network attached connectivity. The Ethernet interface allows communication between drives and direct communication to the datacenter, eliminating the need for Storage Servers for datacenter storage racks.
The Seagate Kinetic Open Storage Data Center vs. the Traditional Model

Basic architecture

Advantages of kinetic drive

- **Performance**
- **Scalability**
- **Simplicity**
- **TCO** (Total cost of ownership)
Libraries, API enable Applications

- Application
- Clustering
- Management

LibKinetic

- Interconnect
  - ProtoBuf
  - TCP/IP/GbE

- Storage

C++, Java, Python, Erlang, DIY

2014 Data Storage Innovation Conference. © Insert Your Company Name. All Rights Reserved.

From Ali Fenn’s and James Hugher’s presentation in SNIA’s DSI Conference - Santa Clara - April 2014
Multiple Masters

- Application
- Clustering
- Management

Application

LibKinetic

Interconnect

ProtoBuf
TCP/IP/GbE

Storage

Proprietary to Vendor

Proprietary to Seagate

LGPL Standard

2014 Data Storage Innovation Conference. © Insert Your Company Name. All Rights Reserved.
P2P operation example

**copydrive (see src/copydrive.cc)**
This example uses the P2P push functionality to copy an entire keyspace from one drive to one or more other drives. For example, to run a pipeline that looks like A -> B -> C you could call it like so:
```
./copydrive A 8123 B 8123 C 4444
```
Goals of the kinetic API

• Data movement
  • get/put/delete/getnext/getprevious
• Multiple masters
• Cluster-able
• 3rd party copy
• Management
Software applications

• OpenStack Swift
• Distributed Hash Tables
• Hadoop Distributed File system
• Basho Riak CS
OpenStack—Kinetic drive

https://developers.seagate.com/display/KV/OpenStack+Swift
OpenStack Swift

OpenStack Swift enables an object storage system that is highly scalable and durable. Swift allows the ability to store, retrieve, and delete objects in containers using a RESTful HTTP API.

The key terminologies used for Swift are:
• **Proxy Server** - Scalable API request handler, determines storage node distribution of objects based on URL
• **Partitions** - A complete and non-overlapping set of key ranges such that each object, container and account is a member of exactly one partition as per the value of its key
• **Ring** - Maps each partition to a set of devices
• **Objects** - Key-value entries in the object store
• **Containers** - Groups of objects
• **Accounts** - Groups of containers
• **Object Server** - Service that provides access to object methods
• **Container Server** - Service that provides access to container methods
• **Account Server** - Service that provides access to account methods

https://developers.seagate.com/display/KV/OpenStack+Swift
Swift clusters using Kinetic drives allow access to any drives and, thus, any object. For the current Kinetic integration, a fraction of Object Server commands (Object Daemon) are embedded within the Proxy Server acting as a logical construct.
Heterogeneous Clusters

Keeping Object Server operations within the Proxy Server conservatively integrates the Kinetic ecosystem with the current Swift ecosystem and potentially provides backward compatibility with existing Swift deployments. The additional value gained by this implementation is allowing the Proxy Servers to use existing protocol, while reducing the amount of I/O from the Object Servers. A possible heterogeneous cluster with Kinetic and Non-Kinetic Zones are shown below.

https://developers.seagate.com/display/KV/OpenStack+Swift
Future possibilities

Current Kinetic replication process involves P2P Copy, where the Kinetic drives can copy objects from other Kinetic drives. Future potential goals for complete Kinetic integration would be to eliminate the Proxy Server restrictions of Zone accesses, such that the Proxy Servers can access all drives from any Zone. In this situation, the Client can make a single connection to a Proxy Server and obtain access to objects from all Zones.

https://developers.seagate.com/display/KV/OpenStack+Swift
## TCO Estimate for 1 PB Usable Swift Cluster

<table>
<thead>
<tr>
<th></th>
<th>With Conventional Drives &amp; Nodes</th>
<th>With Kinetic Drives &amp; Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Annual Costs:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware:</td>
<td>$132,015</td>
<td>$83,209</td>
</tr>
<tr>
<td>Power and cooling:</td>
<td>$36,735</td>
<td>$25,614</td>
</tr>
<tr>
<td>Replacement drives:</td>
<td>$9,375</td>
<td>$5,357</td>
</tr>
<tr>
<td><strong>TOTAL ANNUAL COST:</strong></td>
<td>$178,125</td>
<td>$114,180</td>
</tr>
<tr>
<td>Cost per usable TB / Year:</td>
<td>$178</td>
<td>$114</td>
</tr>
<tr>
<td>Cost per usable GB / Month:</td>
<td>$0.015</td>
<td>$0.010</td>
</tr>
<tr>
<td><strong>Estimated Savings Over Conventional Drives:</strong></td>
<td>-36%</td>
<td></td>
</tr>
</tbody>
</table>

**Kinetic drives enables much higher:**

- Ratio of server node CPU / drive
- Drive density in chassis

**Hardware is amortized over 3 years**

**Model doesn’t include SwiftStack license and support fees**

*Seagate Confidential – Under Embargo Until November 3, 2014 @ 8:01 a.m. ET*

---

*From Ali Fenn's presentation in OpenStack Summit - Paris - November 2014 RaliQuest, York Street Strategy*
Distributed Hash Tables
Distributed Hash Tables (DHT) provides a lookup service by storing key-value pairs in a decentralized distributed system.

The benefits of DHTs are:
- Highly scalable by automatically distributing loads to new nodes
- Distributed storage of objects with known names
- Self-organizing with no need of a central server
- Robust against node failure except for bootstrap nodes
  - Bootstrap node provides initial configuration information to newly joining nodes so that they may successfully join the overlay network
  - Data is automatically sent away from failed nodes

The disadvantages to DHTs are:
- Searching feature based on hash algorithm, where very similar data values can be at totally different nodes.
- Security problems with secure routing and difficulty in verifying data integrity.

https://developers.seagate.com/display/KV/Distributed+Hash+Tables
Node failure/removal

In the case of a node failure or removal, the data must be replicated on to the next successor node for survival.
Node Lookup

Since node 14 does not exist in the ring, the data stored for node 14 will be in the next successor, which in this case is node 15.
Kinetic Open Storage Implementation

DHT can be implemented with Kinetic drives with the use of an integrated Kinetic API that can directly store the key/value objects onto the drives. The use of Kinetic drives eliminates the need for servers to communicate with the drive.

https://developers.seagate.com/display/KV/Distributed+Hash+Tables
Hadoop Distributed File System
https://developers.seagate.com/display/KV/Hadoop+Distributed+File+System
Kinetic Open Storage Integration

HDFS is integrated with Kinetic drives by providing a plug-in for the HDFS Client to communicate with the Kinetic drives. The integration of HDFS with Kinetic drives are independent of the Hadoop ecosystem and is designed to allow the client to choose whether to use the Default File System or the Kinetic File System for the HDFS application. With the Kinetic File System plug-in, the HDFS client no longer needs to communicate with the Datanode in order to communicate with the storage devices. When needed, the HDFS client will communicate with the Namenode to obtain information about the file system's namespace and client's accessibility, before directly connecting to the Kinetic drives. The image on the right shows the architecture for the HDFS Kinetic integration.

https://developers.seagate.com/display/KV/Hadoop+Distributed+File+System
With the use current HDFS architecture and conventional drives, when Datanodes fail, the drives attached to the nodes also fail and become no longer accessible. Having no Datanodes with the use of Kinetic drives, this allows failures to happen only at the drive level, which reduces chances of inaccessible data in a large scale environment. The comparison between conventional and Kinetic drives in HDFS are shown below.

https://developers.seagate.com/display/KV/Hadoop+Distributed+File+System
Basho Riak+Kinetic drive

Riak is a distributed NoSQL key-value data store with extra query capabilities that offers high availability, fault tolerance, operational simplicity, and scalability.

https://developers.seagate.com/display/KV/Basho+Riak+CS
Basho Riak Users Today

http://basho.com/about/customers/
• Riak CS (Cloud Storage) is a highly scalable, open source distributed key/value store built on top of Basho Riak, designed to be used for public or private clouds, and for applications and services with high reliability demands.
• Riak CS API allow RESTful Get, Put and Delete operations for objects and buckets, along with a Map-Reduce function.
• The Riak CS API shards large object sizes into 1MB chunks of key/value objects, which is sent to the Riak cluster to be streamed, stored, and replicated in the Ring.

https://developers.seagate.com/display/KV/Basho+Riak+CS
Kinetic Implementation

Replacing the drives in the Riak Node within The Ring with Kinetic drives allow access to any drives, thus any object. Integrating Kinetic drives to the Riak CS ecosystem allows the elimination of the object servers in the Riak Nodes, such any Riak CS API Server can access all the drives in any node.

In the Riak Ring, the Object Server takes responsibility of a limited number of drives within the Riak Node, which restricts other Object Servers from accessing those drives for service client requests. The elimination of Object Servers can potentially allow any Riak CS Server Node to communicate with all Riak Nodes, thus access to all the drives. Data replication can potentially be performed using P2P Copy, such that Kinetic drives can communicate with each other independent of their Riak Node locations, with the absence of the Object Servers.

https://developers.seagate.com/display/KV/Basho+Riak+CS
Evaluating the performance of Seagate Kinetic Drive Technology and its integration into the CERN EOS storage system

Ivana Pejeva
August 2015
CERN openlab Summer Student Report 2015
Goals of the project

- Goal: Performance evaluation of Seagate Kinetic drive technology and its integration into the CERN EOS storage system.

- CERN openlab is a unique public-private partnership that accelerates the development of cutting-edge solutions for the worldwide LHC community and wider scientific research.

- EOS[1], a multi-petabyte disk storage

[1]"Exabyte Scale Storage at CERN", Andreas J. Peters, Lukasz Janyst
EOS Storage System

Three components:
• Management server (MGM)
• Message queue (MQ)
• File storage services (FST)

DAS vs. Ethernet Drive tech:
• Scalability
  • DAS only support limited number of drives
  • Ethernet is scaled independently
Kinetic Integration with EOS

- 32+4 encoding
- 10+2 encoding

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.
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.
Thanks!