Seagate is building hard disk drives with a direct Ethernet interface and object-style API access for scalable object stores, a plan which - if it works - would destroy much of the existing, typical storage stack.

Drives would become native key/value stores that manage their own space mapping with accessing applications simply dealing at the object level with gets and puts instead of using file abstractions.
Application
Clustering
Management
Interconnect
Storage

LibKinetic

App
App
App
App

D

C++, Java, Python, Erlang, DIY

ProtoBuf
TCP/IP/GbE

Proprietary
to System Vendor

GPL
Standard

Proprietary
to Seagate
SAS versus Kinetic Open Storage

- Standard form factor
- 2 SAS ports
- SCSI command set
  - data = read (LBA, count)
  - write (LBA, count, data)
  - LBA :: [0, max]
  - data :: count * 512 bytes
  - CRC on cmd and PI on block

- Standard form factor
- 2 Ethernet ports (same connector)
- Kinetic key/value API
  - value = get (key)
  - put (key, value)
  - delete (key)
  - key :: 1 byte to 4 KiB
  - value :: 0 bytes to 1 MiB
  - HMAC on cmd and SHA on value
Typical HA High Density

Intel server
  - Double Socket
  - 48GB Ram
  - 1000w

SAS tray
  - Connected to the server
Low cost HA Configuration

Each drive talks to both switches
Each switch has 2 by 10Gb/s Ethernet
Kinetic Tray talks directly to ToR
No servers
System Hardware

Typical JBOD architecture

- Does not require a server, just JBODs to the ToR Switch
- $10 \times 60 \times 4TB = 2.4PB/Rack$
Kinetic Drive

Provides RPC to Key/Value database
  • Data is pre-indexed
P2P (Drive to Drive) copy of key ranges
Communicate using existing Data Center Plumbing (TCP/IP)
Multiple masters - Data sharing between machines
Configurable caching per command
  • Async, Sync, Flush
Local space management
Kinetic Systems

Clustering (performance, reliability, management)
Compatibility with large scale applications (S3, etc.)
Centralized Management
  • Reliability, availability, durability
Goals of API

Data movement
  • Get/put/delete/getnext/getprevious
  • Versioned (== for success), options

Range operations

Multiple masters
  • Authentication/Integrity/Authorization

Cluster-able
  • Simple cluster configuration version enforcement

3rd party copy

Management
Existing Traffic Flow
Kinetic Traffic Flow
Cumulative operations ordered by length

- 92% of the operations
- 0.5% of the data

Cumulative percentage

Length (KB)
Map of Operations
Performance Metrics

Same normal performance expectations

- Sequential Write: 50MB/s
- Random Write: 50MB/s
- Sequential Read: 50MB/s
- Random Read: 1.2x slower than traditional drives
Write Performance Results

1MB values put rate (MB/s) vs. 1KB values put rate

- MB/s
  - Y-axis: 0 to 120
  - X-axis: 0 to 8
- Puts/s
  - Y-axis: 0 to 6000
  - X-axis: 0 to 8
Bootstrapping devices
Kinetic Security Deep Dive

Kinetic Protocol

Transports

Drive Security
Kinetic Protocol

Authentication
  Identity of Client

Integrity
  Command and data
  Requests and responses

Roles
  Get/put/management/security

Replay prevention
  Messages inside a session
  Messages between sessions
Transports

Cleartext (Port 8123)
  • Normal Client (not recommended for configuration)

TLS (Port 8444)
  • Admin Client or normal client
Drive Security

ISE
• Erase all customer information and configuration
• quick return factory “remanufacture”

SED
• Pin Unlock at power on
• Over the TLS port
Conclusion

Next Generation Storage Devices

• Disaggregates storage from compute
• Enable innovation in hardware and software ecosystem
• Lower TCO

Integration with:

• Swift
• HDFS
• Scality
• Basho Riak
• Ceph
More information

- https://developers.seagate.com
- http://github.com/Seagate