Object Storage 201

Understanding Architectural Trade-offs in Object Storage Technologies

SNIA™

Cloud Storage Initiative

SNIA Legal Notice

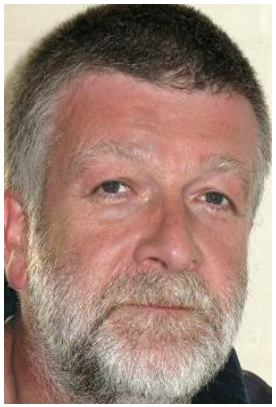
- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Today's Presenters

SNIA™
Cloud Storage Initiative

Alex McDonald, SNIA – CSI
Cloud Storage Initiative Chair - NetApp



Vishnu Vardhan,
Sr Mgr, Product Management,
NetApp
Twitter: @vardhanv

Paul S. Levy
System's Engineer & Architect
Intel Storage Division

Agenda

SNIA™
Cloud Storage Initiative

- **Object store market requirements**
- How do object stores scale?
- Architectural choices for object stores
- Conclusions

Object store market requirements

SNIA Cloud Storage Initiative

- Expected Scale
  - 100s of Billions of objects
- Performance Requirements
  - Latency (Low latency vs Medium latency workloads)
  - Throughput (High throughput vs Low throughput workloads)
- Key Features
  - Multiple sites
  - Data protection policies
  - Business logic policies
  - Erasure coding (leading to an object count explosion)
  - Multiple / complex failure domains (disk, node, rack, data center, business unit)
- Operations and Management Requirements
  - Adding nodes & sites
  - Changing capacity of existing nodes and related balancing / re-balancing
  - System upgrades and Data Migration
  - Long lived systems, with topology changes and required rebalancing

*Sources: Google: Out Mobile Planet (26 apps/phone), Intel est.

Agenda

SNIA™
Cloud Storage Initiative

- Object store market requirements
- **How do object stores scale?**
- Architectural choices for object stores
- Conclusions

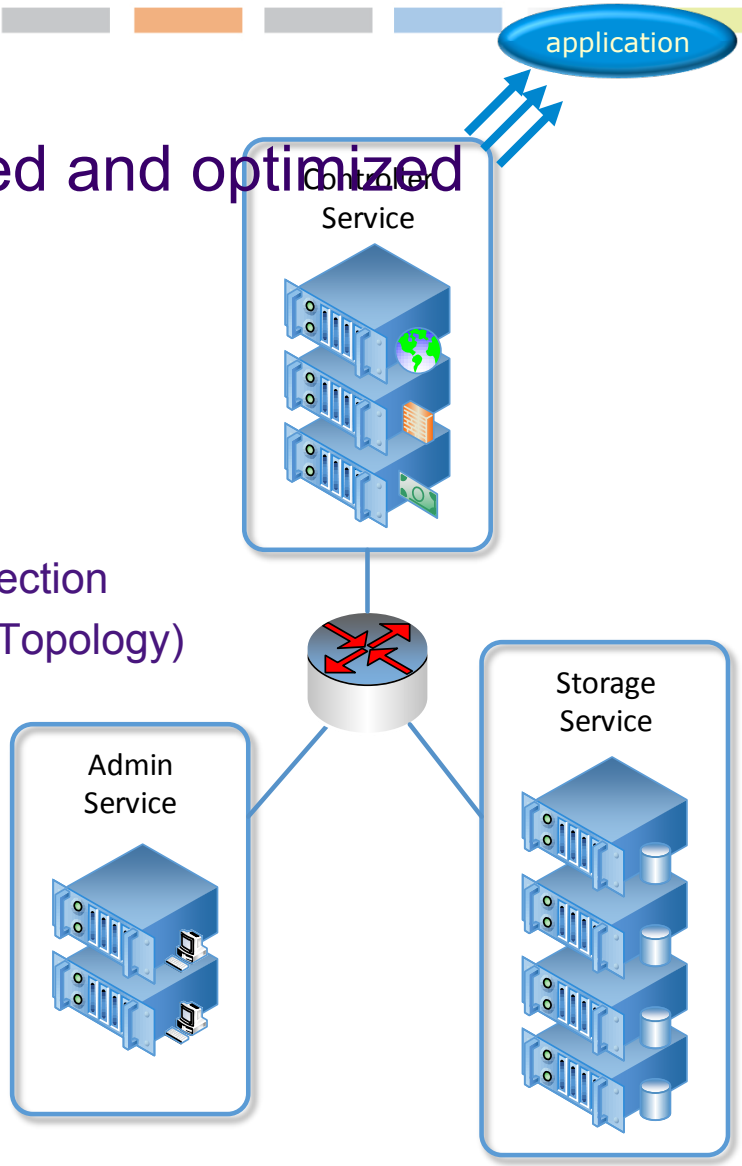
Scale-Out Properties

SNIA Cloud Storage Initiative

- Shared nothing architecture
- Ubiquitous Ethernet to tie things together
- Horizontally scaling of Independent services
  - Load Balancing – Load balancing across a large number of nodes
  - Placement Service – Determines location of where data needs to be placed
  - Storage Service – Basic storage service
  - Tiering and Policy management – Execute tiering & policy actions
  - Access Control, audit & security – Provide authentication, audit and security services
  - Administration and Maintenance – Handle topology changes (disks, nodes, racks, sites), expansions, contractions, hardware refreshes, system upgrades

How do Object Stores Scale? Functional Decomposition

- Essential functions are segmented and optimized for horizontal scaling
  - Controller
    - Gateway – Client interface
    - Load Balancing
    - Data Placement
    - Tiering, Policy Management & Data protection
    - System metadata Mapping information (Topology)
    - Object Metadata
  - Admin & Maintenance
    - Admin & Maintenance
  - Storage Servers
    - Information Repository

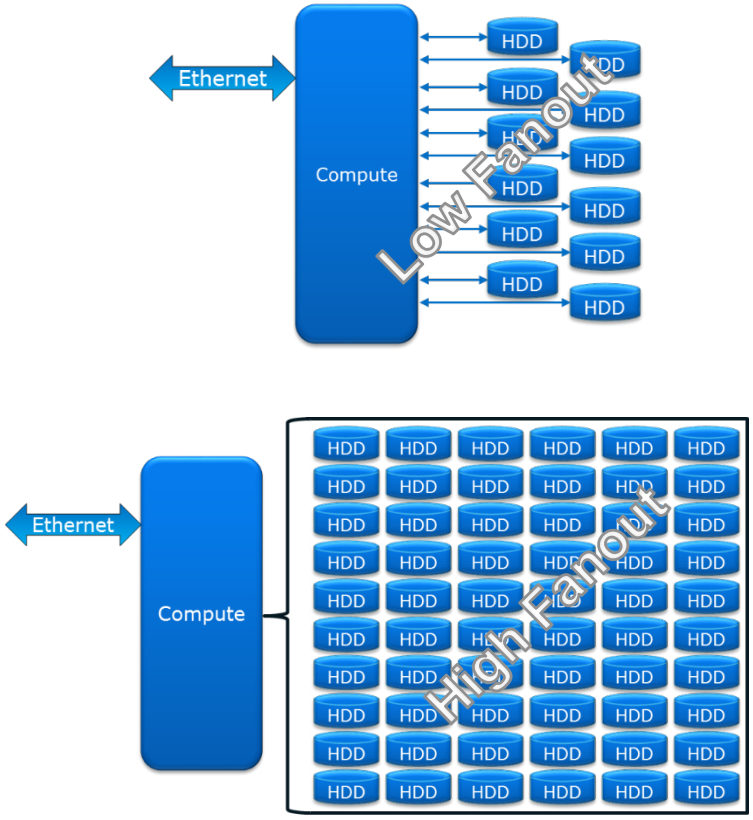


Storage Nodes - Typical

Storage Nodes store information.

Functionally they:

- Store Objects
- Support the Private Network
- Support Maintenance activities
- Defines Blast Radius

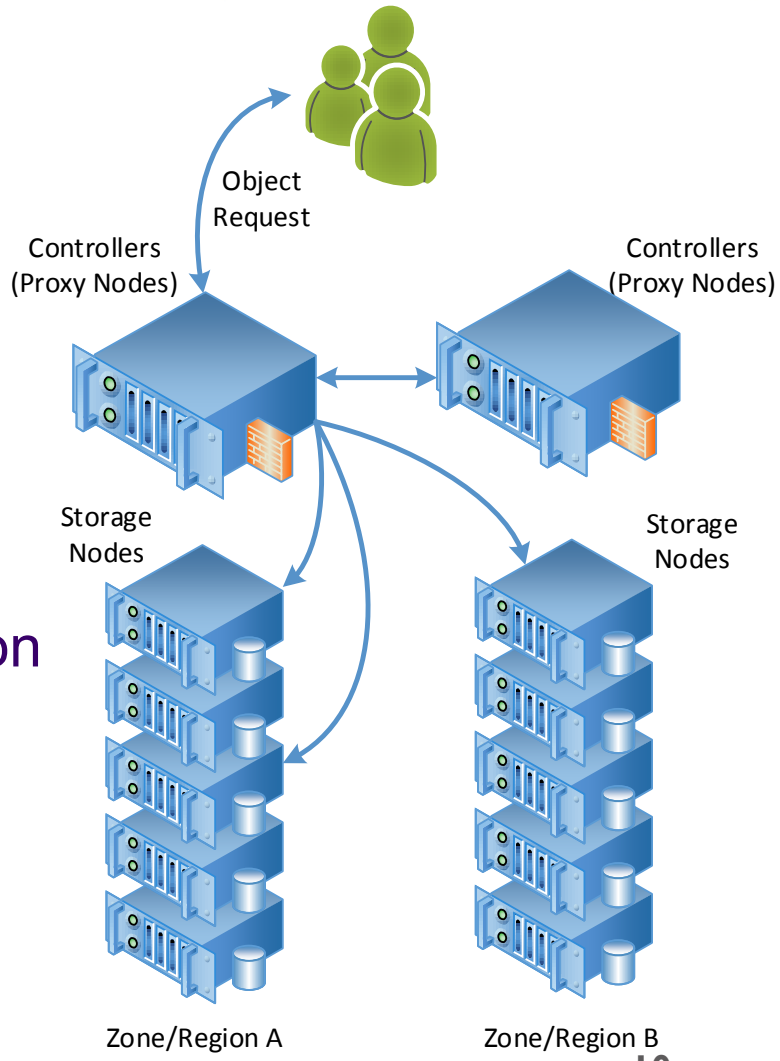


Controllers (Proxy Nodes) - Typical

Controller Nodes are the brains of the Object Store.

Functionally they:

- Provide a gateway – Client interface
- Load Balancing
- Data Placement
- System metadata Mapping information (Topology)
- Object Metadata
- Tiering, Policy Management & Data protection



Agenda

SNIA™
Cloud Storage Initiative

- Object store market requirements
- How do object stores scale?
- **Architectural choices for object stores**
- Conclusions

Architectural choices

SNIA™ Cloud Storage Initiative

Controller Nodes

- Provide a gateway – Client interface
- Load Balancing
- **Data Placement**
- **System metadata Mapping information (Topology)**
- **Object Metadata**
- **Tiering, Policy Management & Data protection**

Admin Nodes

- **Admin and Maintenance procedures**

Storage Nodes

- **Information repository**

TODAY'S DISCUSSION

Architectural choices

SNIA Cloud Storage Initiative

Controller Nodes

- Provide a gateway – Client interface
- Load Balancing
- **Data Placement**
- **System metadata Mapping information (Topology)**
- **Object Metadata**
- **Tiering, Policy Management & Data protection**

Admin Nodes

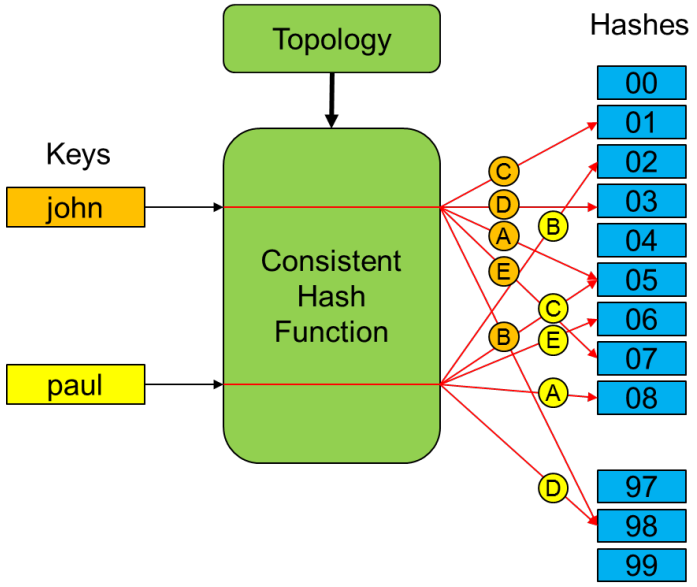
- **Admin and Maintenance procedures**

Storage Nodes

- **Information repository**

Consistent Hash

- Generates a repeatable sequence of hash values based on the input key and topology
- State Less
- Every element in the system knows data location given the Key and Topology



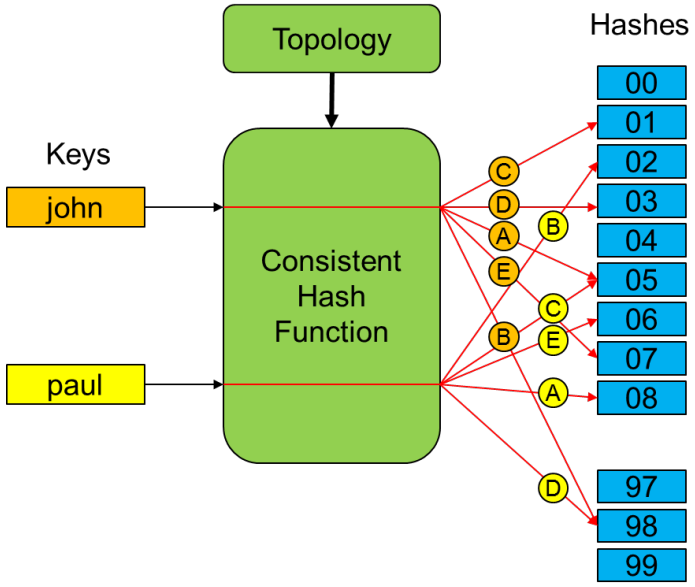
john's hash sequence:
05,98,01,03,07

paul's hash sequence:
08,02,05,98,06

Extreme Performance, Stateless, Consistent (implementations vary)

Consistent Hash

- Pros
  - State Less
  - Scales to trillions of Objects
  - Fast
- Cons
  - Consistent
  - Topology Dependent
  - Collision avoidance to guarantee uniform distribution
  - Data movement required for adding storage (Topology Change)
  - Should replace failed nodes for best performance

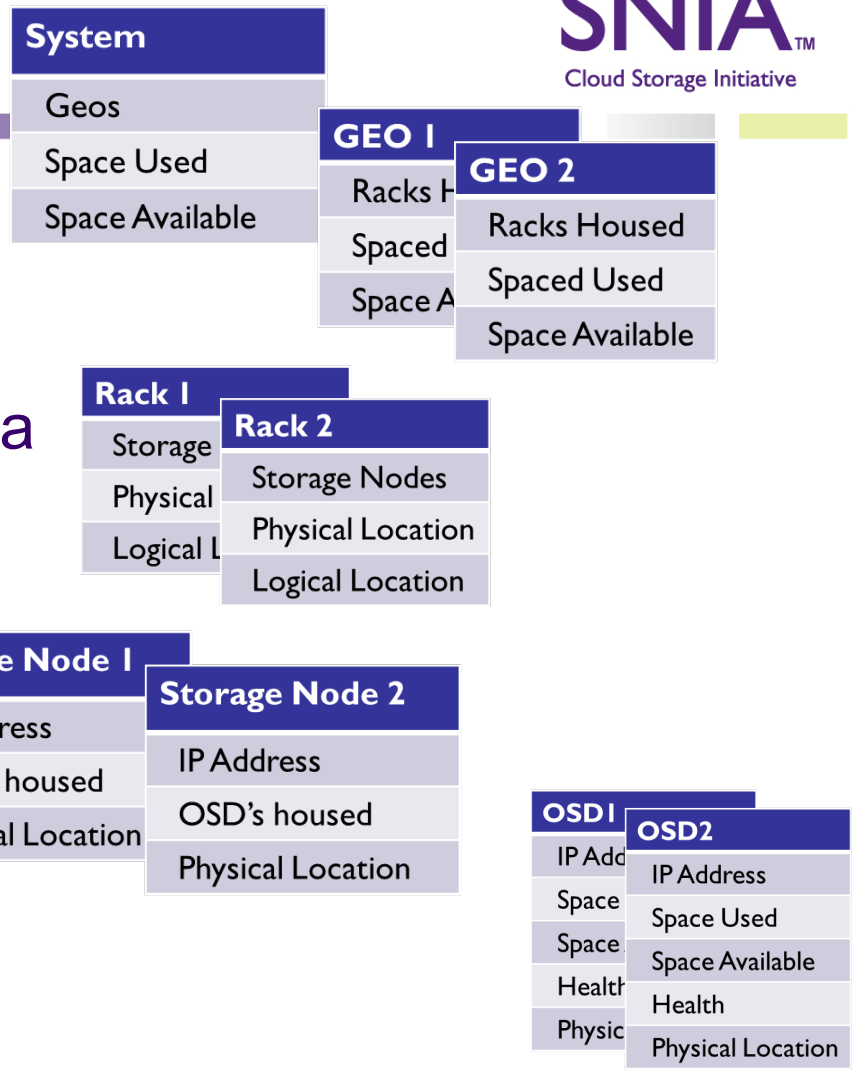


john's hash sequence:
05,98,01,03,07

paul's hash sequence:
08,02,05,98,06

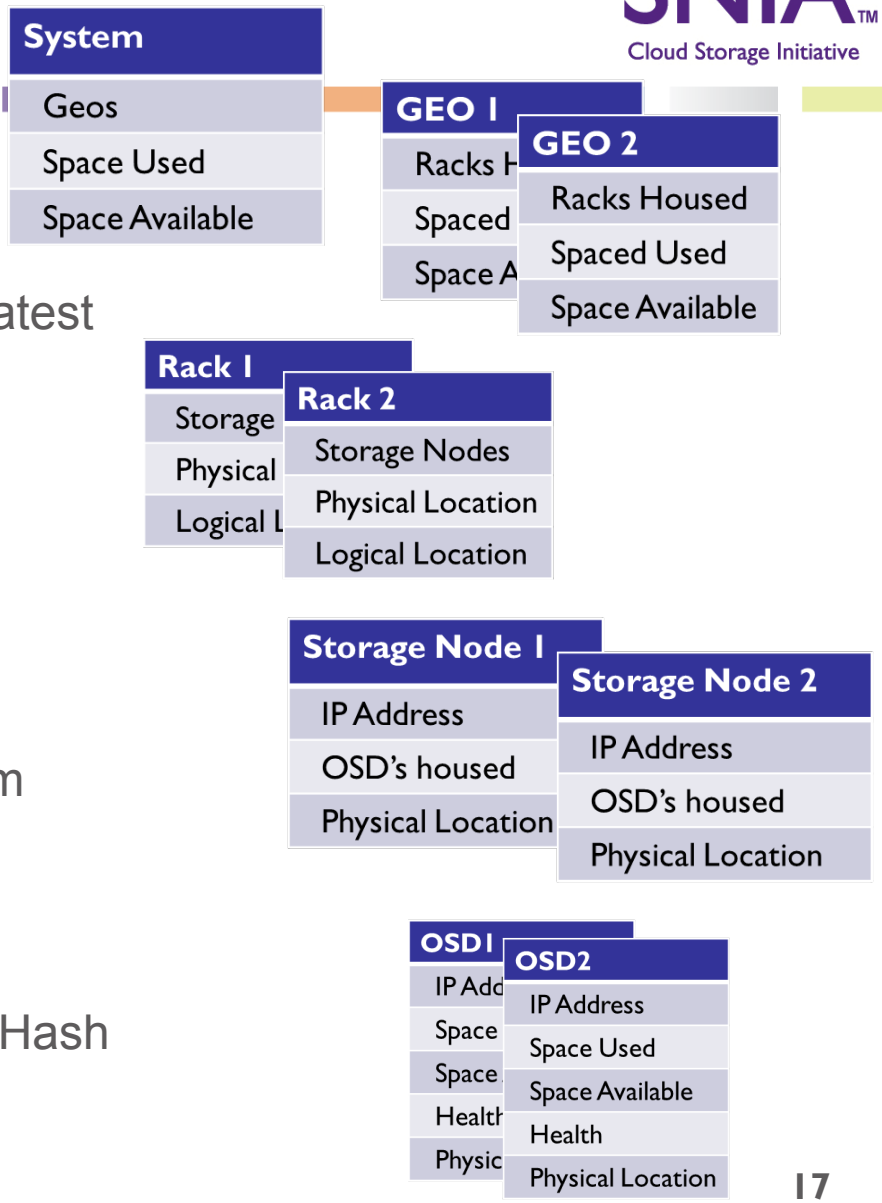
Centralized Database

- Organized collection data in structured sets that form tabular relationships between a key and a value.
- Stateful
- Located in one place



Moderately Performant, Stateful, Consistent

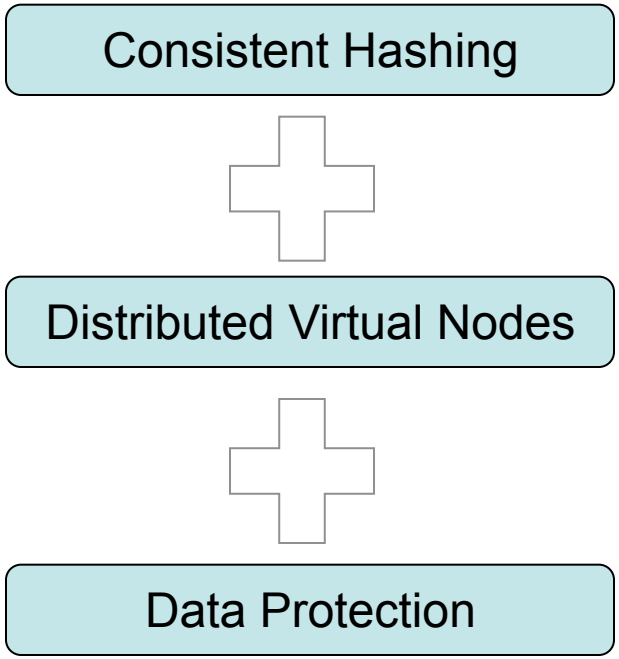
Centralized Database



- Pros
  - Consistent – always returns the latest copy
  - Flexible Placement
  - Scales to Billions of Objects
  - Maintenance and Expansion has manageable impact
- Cons
  - State Full – Needs protection from Single point of failure
  - Centralized – Bottleneck must be managed
  - Not as performant as Consistent Hash or scalable as No-SQL

NoSQL

- Databases that contain relationships other than structured tabular sets. (e.g. Key-Value, Graph, Document)
- Distributed across multiple systems
- No single point of failure

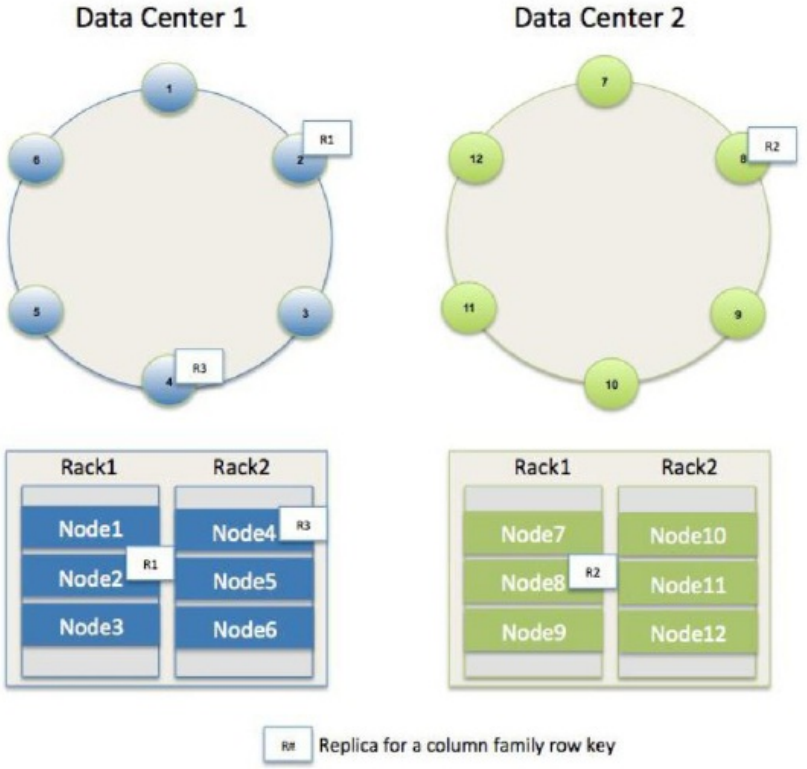


High Performance, De-Centralized, State-full, Eventually Consistent

No-SQL

SNIA™
Cloud Storage Initiative

- Pros
  - Not Centralized
  - Flexible Placement
  - Scales to Trillions of Objects
  - Distributed across multiple sites
  - Partition tolerant
  - Highly performant @ scale
- Cons
  - Eventually Consistent



Data Placement Tradeoff's
Performance and scalability – Comparison of options

SNIA Cloud Storage Initiative

Attributes	Consistent hashing	Centralized DB's	NoSQL DB's
Lookup Latency	Real time / fixed time	DB size dependent	200[1] us
Throughput	Network topology & system design	Network topology & system design	Network topology & system design
Scalability	No limit	DB size dependent	100's of Billions[2]
Consistency	Strong	Strong	Eventual
Examples	Ceph, Swift	MySql, Hadoop Name Node	MongoDB, Cassandra, HBASE

1. Assumes Cassandra with a mostly read workload with 8 shards. (http://planetcassandra.org/nosql-performance-benchmarks/) Converting Ops / Sec into latency
2. http://www.mongodb.com/scale

Architectural choices

SNIA™ Cloud Storage Initiative

Controller Nodes

- Provide a gateway – Client interface
- Load Balancing
- **Data Placement**
- **System metadata Mapping information (Topology)**
- **Object Metadata**
- **Tiering, Policy Management & Data protection**

Admin Nodes

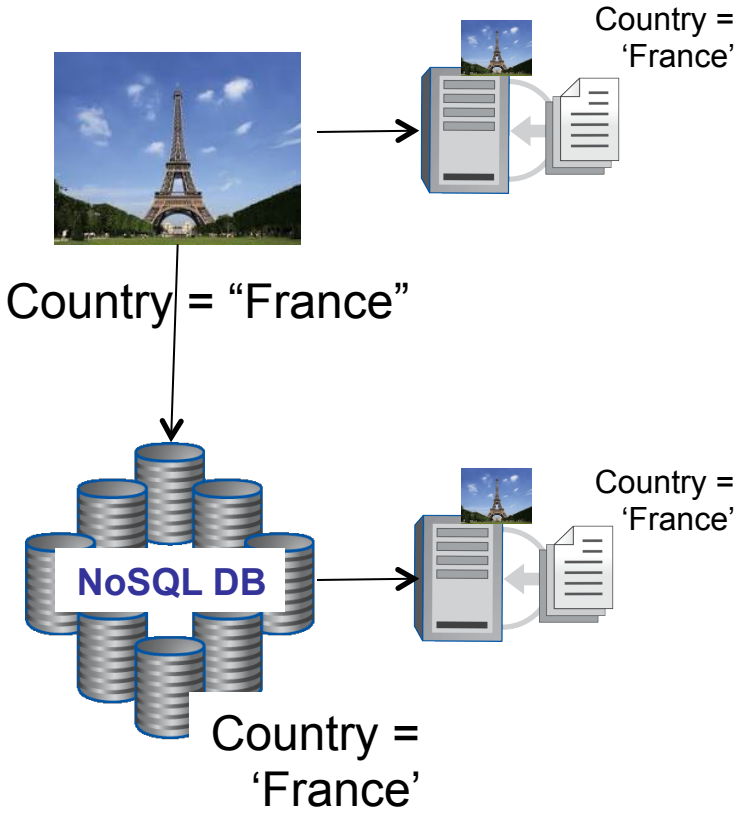
- **Admin and Maintenance procedures**

Storage Nodes

- **Information repository**

Object Meta-Data storage

- Store with the data (e.g in the XATTR of the file-system)
  - Infinitely scalable
  - Limited usability apart from serving the meta-data back to the application
- Store in a centralized database
  - Single point of failure and scalability challenges
- Store in a NoSql database
  - Scale to 100's of billions of objects
  - Policy based on business logic and application metadata
  - Analytics on the metadata



Tiering, policy management and data protection

SNIA Cloud Storage Initiative

- Consistent Hashing – Simplistic policy
  - Keep 3 copies in Rack 1, Use erasure coding instead of replication
  - No per-object policies
  - Ingest only policy
- NoSQL & Centralized DB approaches
  - Keep 3 copies in Rack 1
  - Keep 3 copies in Rack 1 for 10 days, then keep 2 copies for 30 days, then keep 1 copy (erasure coded)
  - If Object metadata = 'CEO's email' keep 5 copies
  - If object metadata ="Europe" stay within London datacenter
  - Per object policies

Architectural choices

SNIA Cloud Storage Initiative

Controller Nodes

- Provide a gateway – Client interface
- Load Balancing
- **Data Placement**
- **System metadata Mapping information (Topology)**
- **Object Metadata**
- **Tiering, Policy Management & Data protection**

Admin Nodes

- **Admin and Maintenance procedures**

Storage Nodes

- **Information repository**

Operations Maintenance and Growth

- Scenarios to consider
  - Adding a disk, node, rack, site
  - Reducing capacity of a node (Rebalance)
  - Disk, node, rack, site failure (Rebuild)
  - Hardware refresh (and migrating data to a new node)
  - Upgrade
- Hash based systems must maintain hash order when topology changes. (DIAG)
- DB Based system can re-balance over time
- Long lived systems, with topology changes and required rebalancing
  - Systems live for many generations, topology changes are common

Architectural choices

SNIA Cloud Storage Initiative

Controller Nodes

- Provide a gateway – Client interface
- Load Balancing
- **Data Placement**
- **System metadata Mapping information (Topology)**
- **Object Metadata**
- **Tiering, Policy Management & Data protection**

Admin Nodes

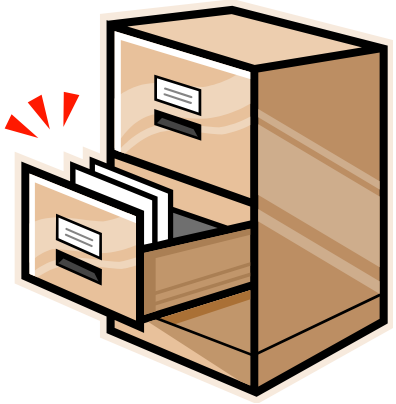
- **Admin and Maintenance procedures**

Storage Nodes

- **Information repository**

Existing file-systems vs custom file-systems

- Use existing filesystems
  - Leverage years of file-system hardening
  - Limited by / Advantages of file-system scalability at a per node level
    - EXT4 vs XFS vs BTRFS

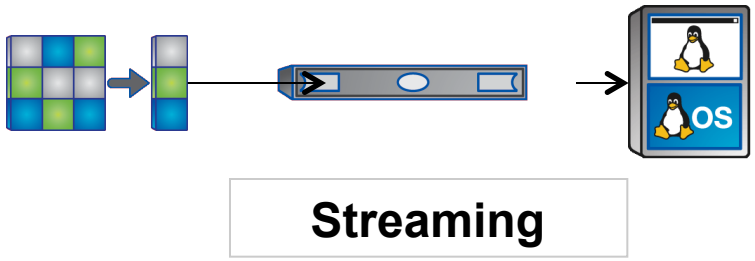


- Custom filesystem
  - Optimize for specific use cases by reducing unnecessary file-system overhead
    - Minimize storage overhead
    - Reduce operating system latency
    - LevelDB/RocksDB

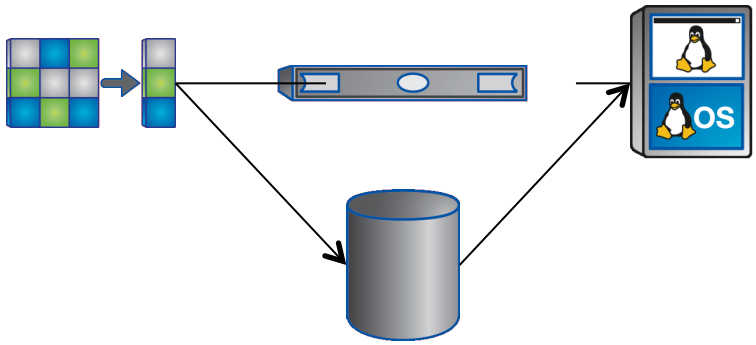
Streaming vs Store & forward architectures

SNIA Cloud Storage Initiative

- Streaming vs Store & Forward
  - Write - Do not wait for a full object to be written, before you make placement decisions – make them on the fly
  - Read - Do not wait for a full object to be assembled before you respond to a read
- Streaming architectures
  - Lower time to first byte
  - Enabling scaling to very large object sizes
  - Compress and encrypt data on the fly
  - Can restore lost fragments instead of whole objects
  - Ability to support random reads
- Store and forward architectures
  - Lower complexity



Streaming



Store & Forward

Agenda

SNIA™
Cloud Storage Initiative

- Object store market requirements
- How do object stores scale ?
- Architectural choices for object stores
- **Conclusions**

Performance and scalability in the real world

- Real world performance
  - Most object storage latencies in the 10's of ms
  - Limited by WAN & LAN latencies / throughput
  - At scale - Object sizes matter, data protection schemes matter
- Real world scalability
  - Ability to handle topology changes (disks, nodes, racks, sites), expansions, contractions, hardware refreshes, system upgrades
  - Real world scale – trillions of objects vs 100's of billions, single name-space versus 5 namespaces ?
- Real world consistency
  - Use case specific decision – Simultaneous modification of same object needs strong consistency properties

Architecture tradeoffs

- Simple to deploy
- Scalable
  - How the systems scale beyond a few PBs,
- Manageable
  - How will they be managed across multiple years and deployments
- Flexible
  - Ability to incorporate new features and capabilities

Object storage
"Market Requirements" versus "Your requirements"

- Expected Scale
  - 100s of Billions of objects
- Performance Requirements
  - Latency (Low latency vs Medium latency workloads)
  - Throughput (High throughput vs Low throughput workloads)
- Key Features
  - Multiple sites
  - Data protection policies
  - Business logic policies
  - Erasure coding – (leading to an object count explosion)
  - Multiple / complex failure domains (disk, node, rack, data center, business unit)
- Operations and Management Requirements
  - Adding nodes & sites
  - Changing capacity of existing nodes and related balancing / re-balancing
  - System upgrades and Data Migration
  - Long lived systems, with topology changes and required rebalancing

After This Webcast

SNIA Cloud Storage Initiative

- This webcast and a copy of the slides will be posted to the SNIA Cloud Storage Initiative (CSI) website and available on-demand
  - http://www.snia.org/forum/csi/knowledge/webcasts
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-CSI blog
  - http://www.sniacloud.com/
- The original “Object Storage 101” webcast is available on-demand at the SNIA-ESF website http://www.snia.org/forums/esf/knowledge/webcasts

Conclusion

SNIA™
Cloud Storage Initiative

Thank You