Scale-out Storage Architectures in the NVM Era

"Evolution or Revolution?"

Sandeep Uttamchandani

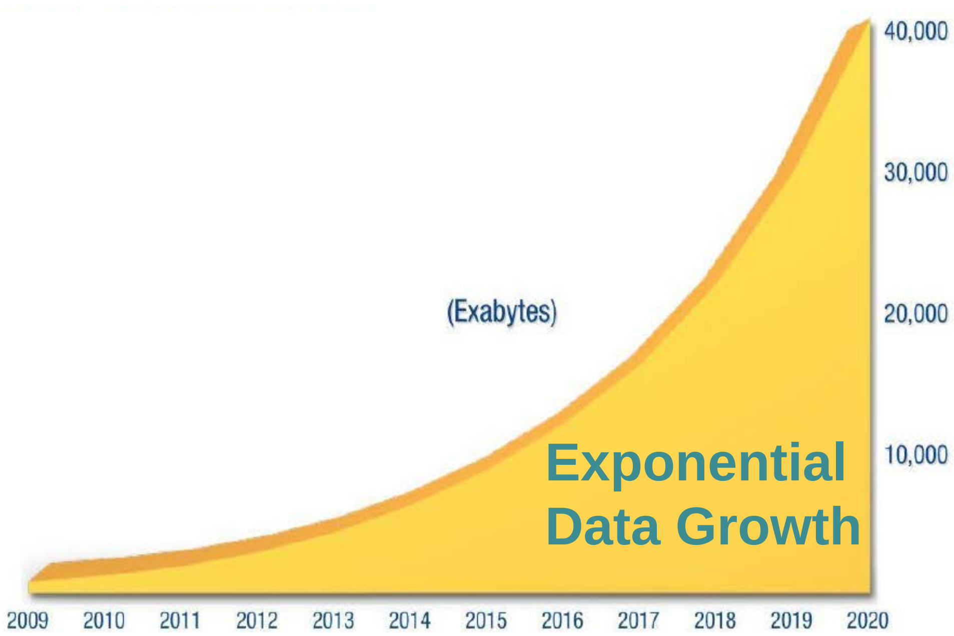
Chief Architect, Cloud Storage

VMWare Inc.

Agenda: 3 WHYs

- **Why** Enterprise Storage → Scale-out Architectures
- **Why** NVM != <*Yet Another Storage Tier*>
- **Why** New Scale-out Design != Clean slate

Exponential Data Growth in Enterprises

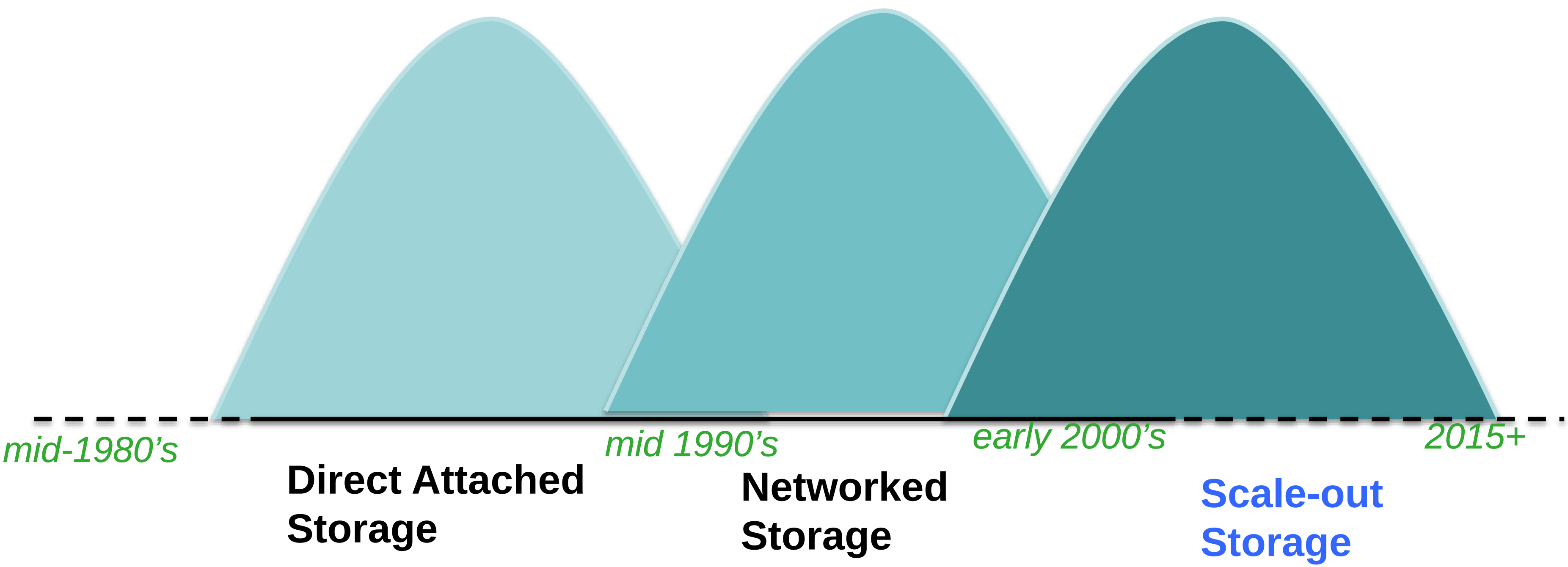


Source: IDC's Digital Universe Study, 2012

"There were 5 Exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days."

Eric Schmidt, Google 2010 Convention

Enterprise Storage Evolution



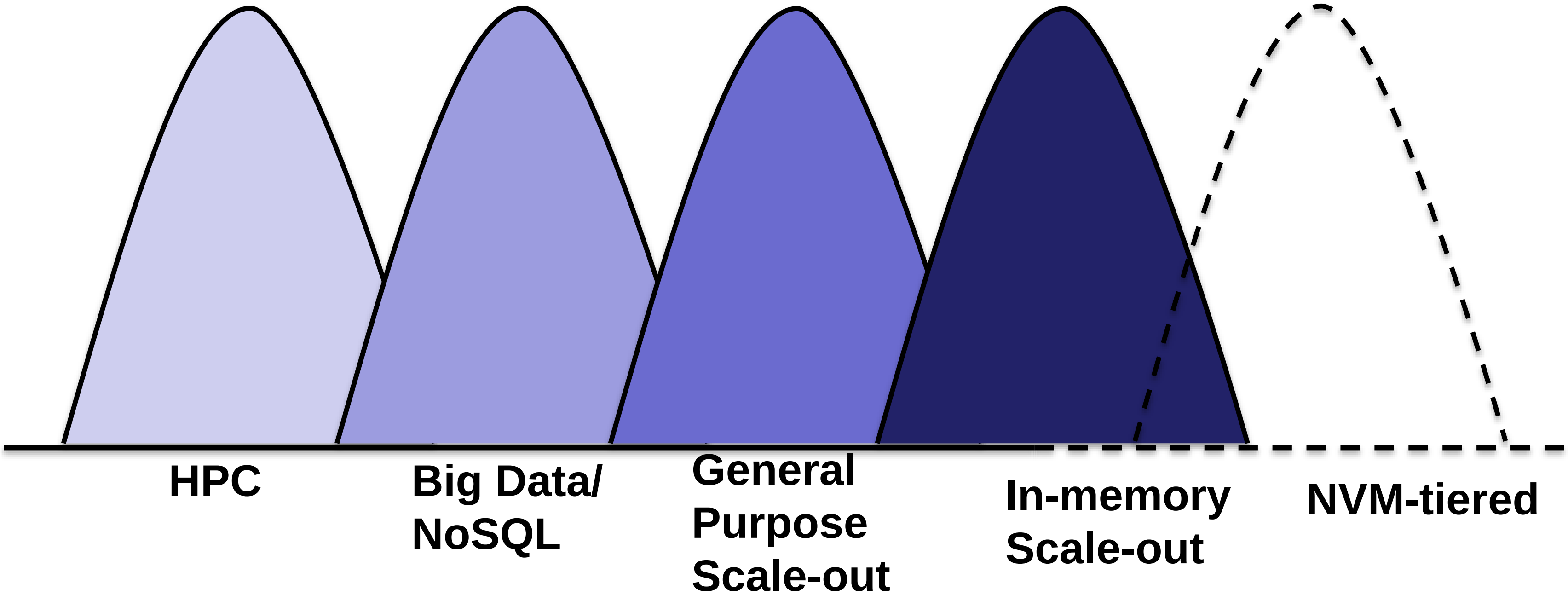
Scale-out Storage Evolution

- Lustre
- IBM GPFS
- VMWare VMFS
- Veritas CFS
- …

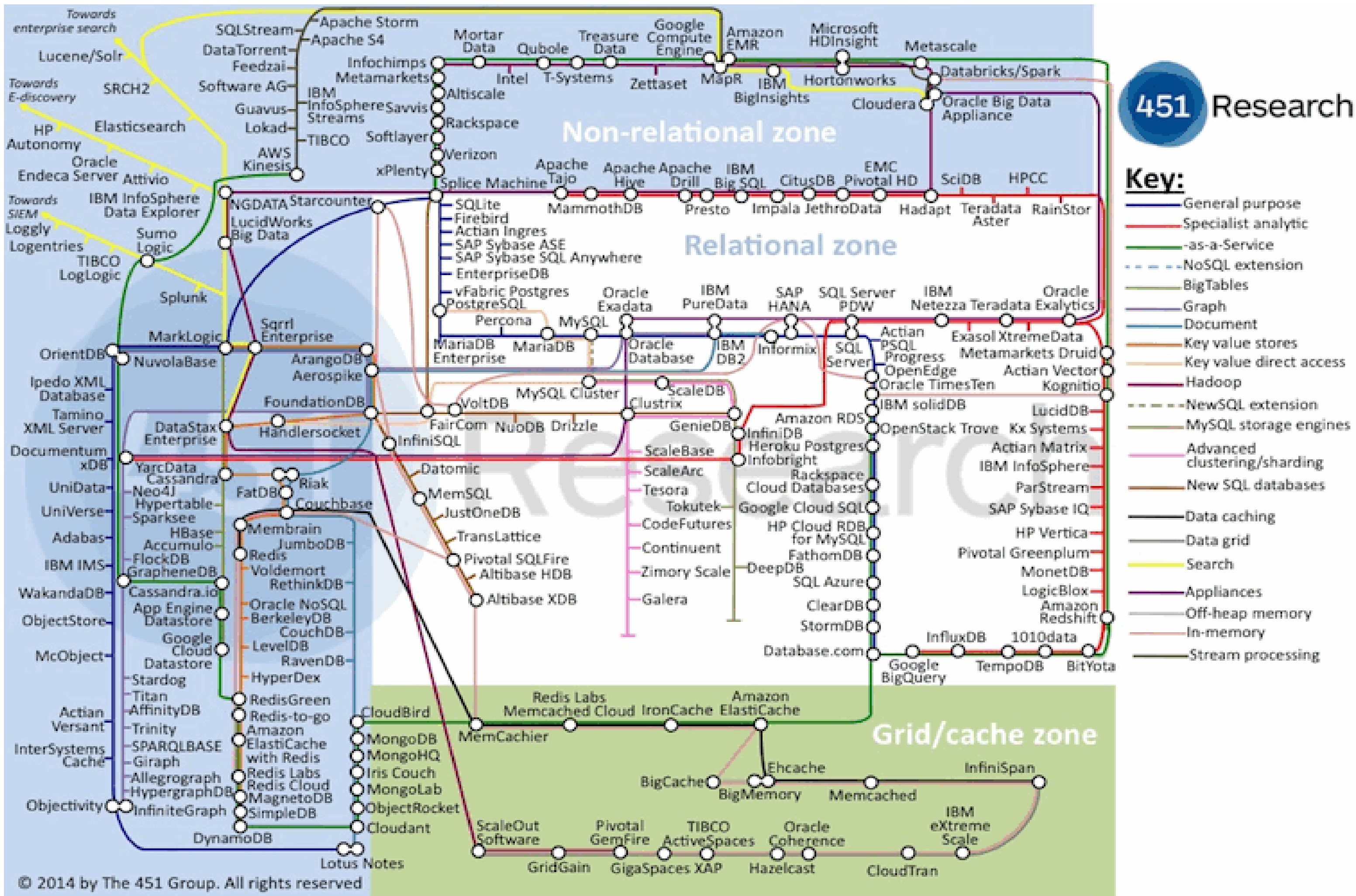
- Cassandra
- HDFS
- MongoDB
- HBase
- …

- VMWare VSAN
- Ceph
- GlusterFS
- Swift
- …

- Redis Cluster
- Stanford RamCloud
- Spark/Tachyon
- VoltDB/H-Store
- …



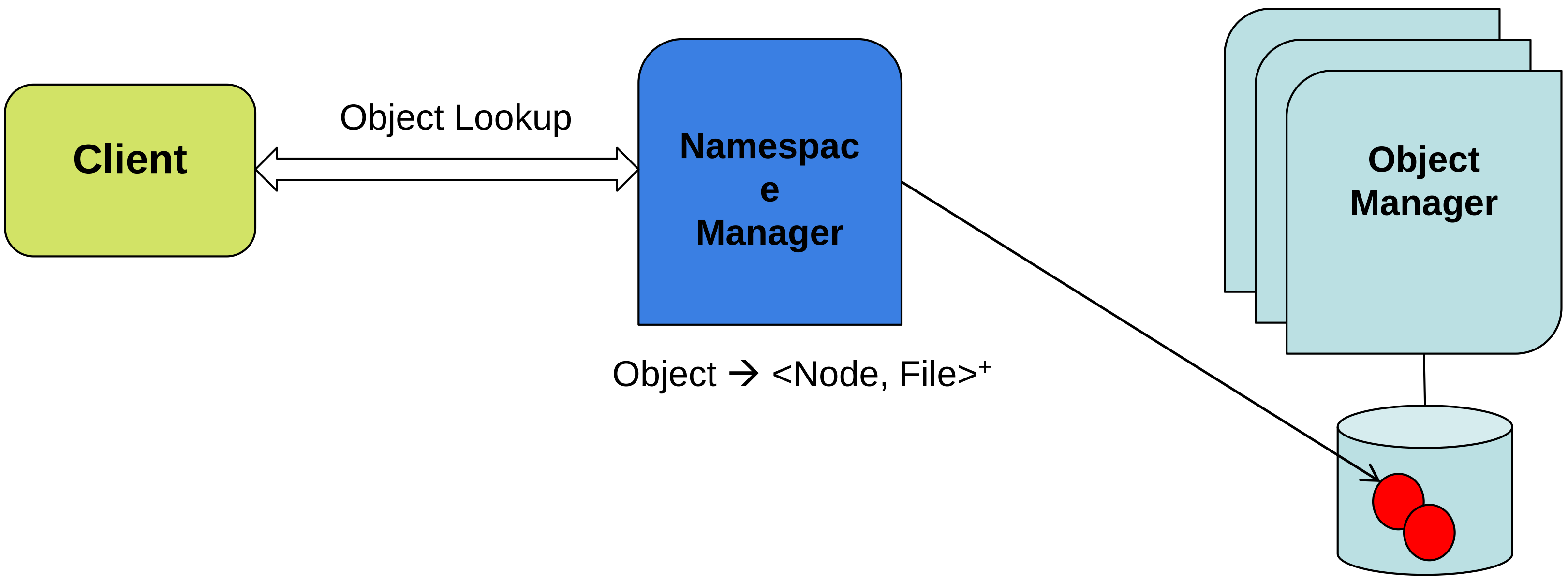
Scale-out Architectures: End of *one-size-fits-all*



Agenda: 3 WHYs

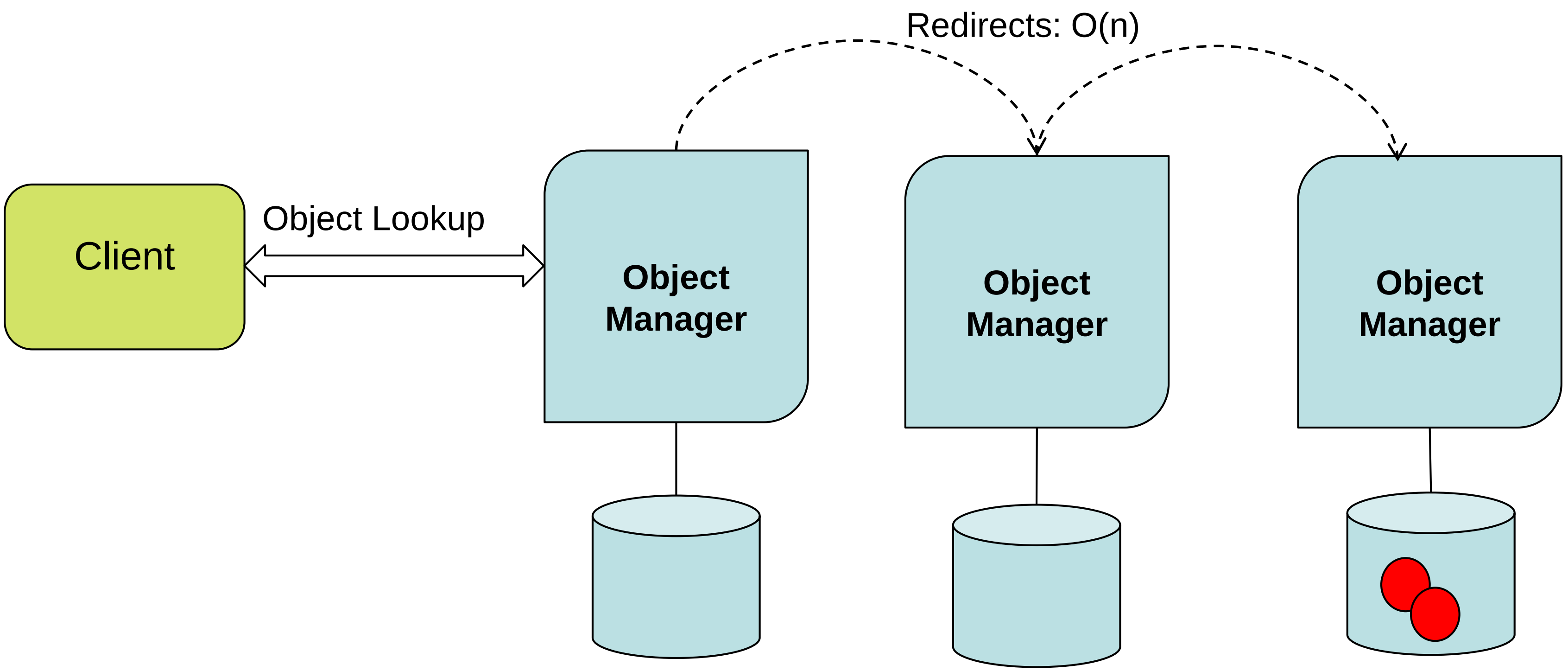
- Why Enterprise Storage → Scale-out Architectures
- Why NVM != <Yet Another Storage Tier>
- Why New Scale-out Design != Clean slate

Scale-out Design: Object Lookup



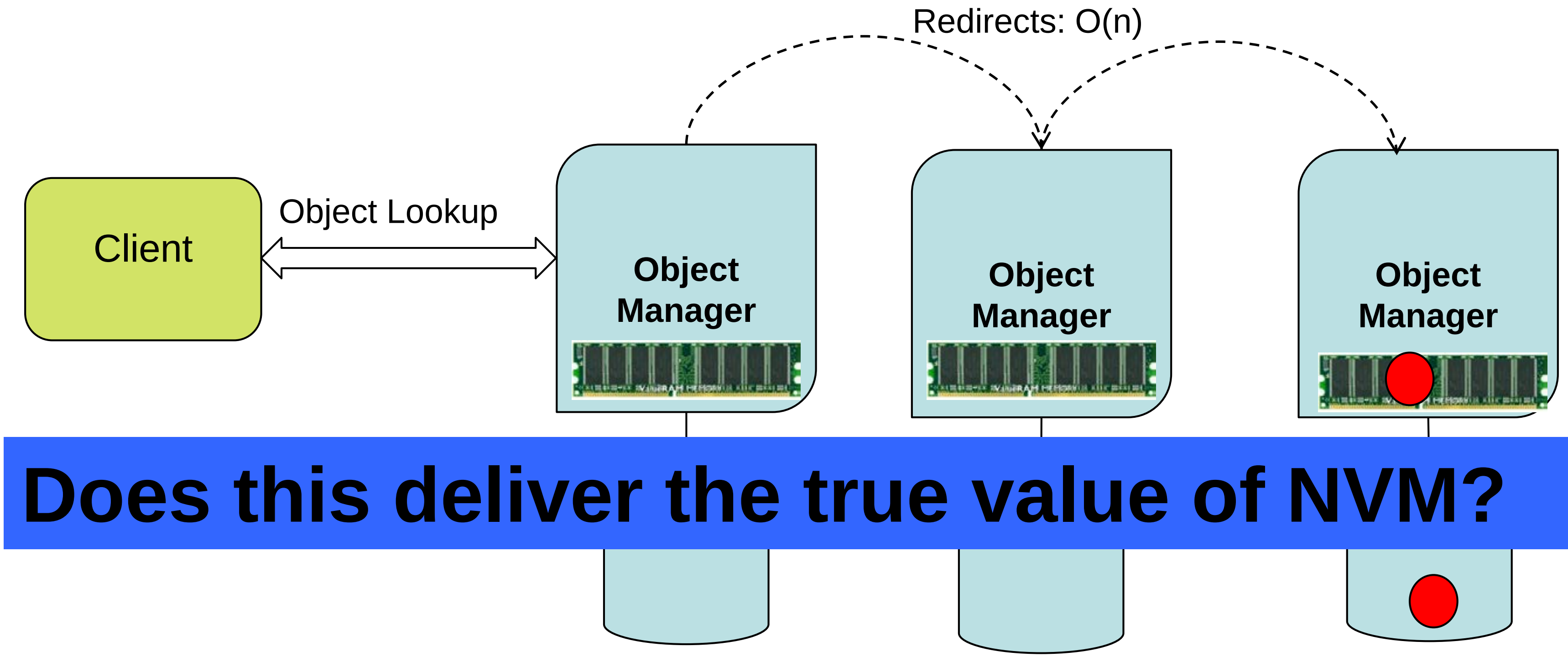
Directory Lookup Design Pattern

Scale-out Design: Object Lookup



Range-based Lookup Design Pattern

Scale-out Design: Object Lookup



Does this deliver the true value of NVM?

Range-based Lookup Pattern

Network is the new bottleneck

	1983	2005	2016+
NW RTT	2.54 ms*	150-300 μs	1-2 μs
Best IO Latency	60-80 ms	4-6 ms	100s of ns

**S. Rumble, et. al. Its time for low Latency. HotOS 2011*

Namespace Sharding

Object Lookup: ★ Client-aware ✪ Client-opaque

Directory Lookup	GPFS Lustre	HDFS MongoDB	EMC OneFS	
Key-based Range Routing		Cassandra Riak	Swift	Redis
Lookup + Key-based Routing		HBase BigTable	Ceph	RamCloud

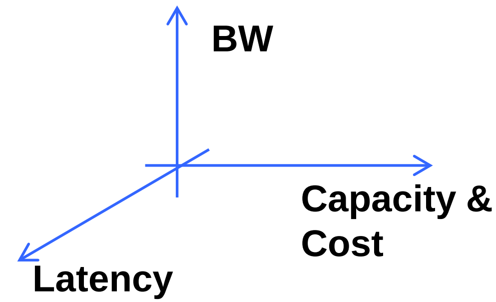
Shared Disk ↔ **Big Data/ NoSQL** ↔ **General Purpose Scale-out** ↔ **In-mem Scale-out** ↔ **NVM-tiered (Future)**

Namespace Sharding

Object Lookup: ★ Client-aware ✪ Client-opaque

	Shared Disk	Big Data/ NoSQL	General Purpose Scale-out	In-mem	NVM-tiered (Future)
Directory Lookup	GPFS ✪ Lustre ★	HDFS ★ MongoDB ✪	EMC OneFS ✪		
Key-based Range Routing		Cassandra ✪ Riak ✪	Swift ✪	Redis ★✪	
Lookup + Range-based		HBase ★ BigTable ★	Ceph ★	RamCloud ★	★

Holistic Shifts in Design Constraints

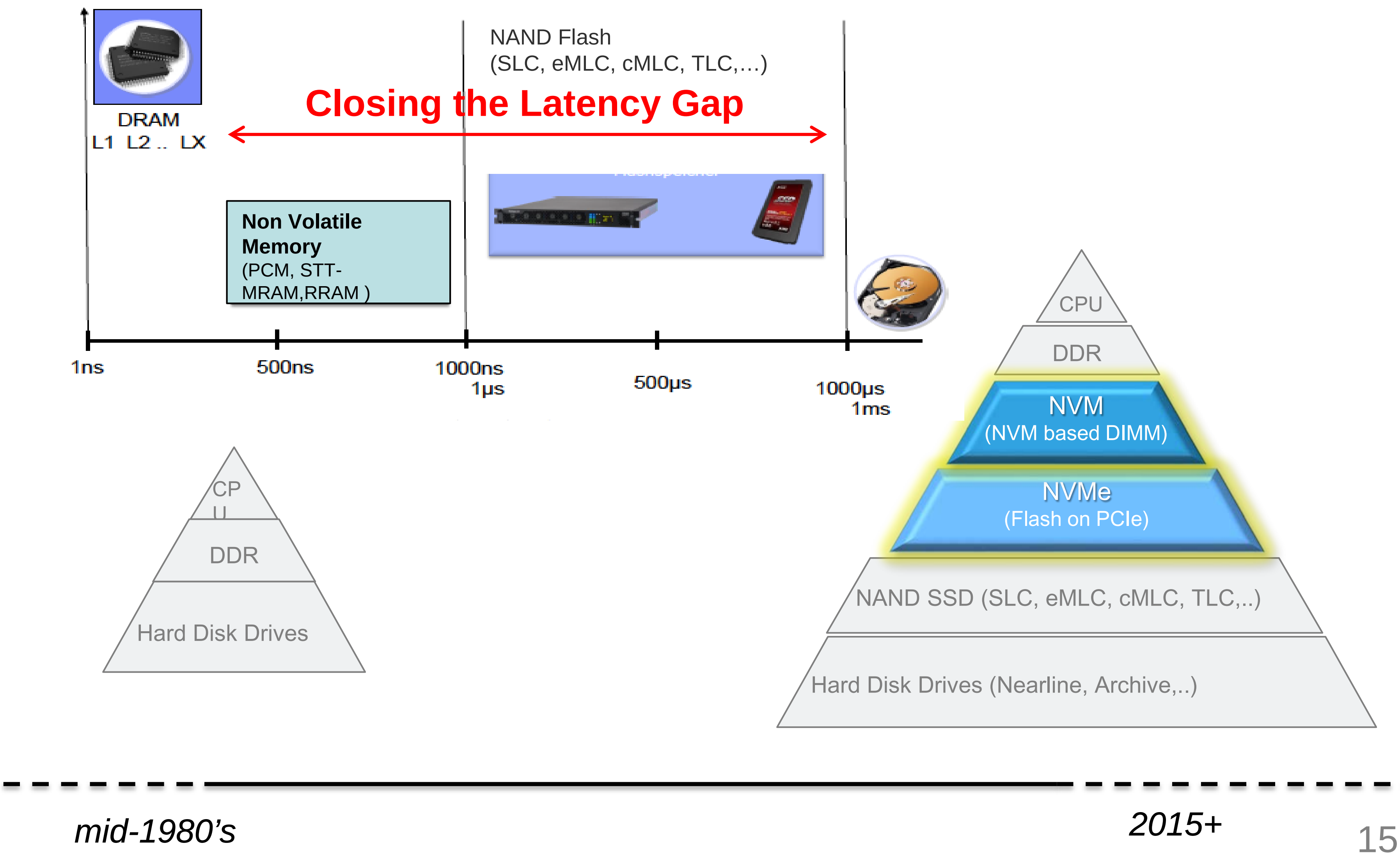


Application	POSIX/ACID Batch-oriented	Non-POSIX/BASE Real-time	• *Beyond traditional block/file POSIX applications*
CPU	1x10MHz	16x3GHz	• *Slow-down of Moore's law*
Memory	< 2 MB	>16GB	• *Bigger & Cheaper*
Network	3Mbps	10Gbps	• *Network is becoming the new latency bottleneck*
Storage	<30MB	>4TB	• *Emergence of distinct Capacity and Performance Storage Tiers*

Avg. Node Configuration

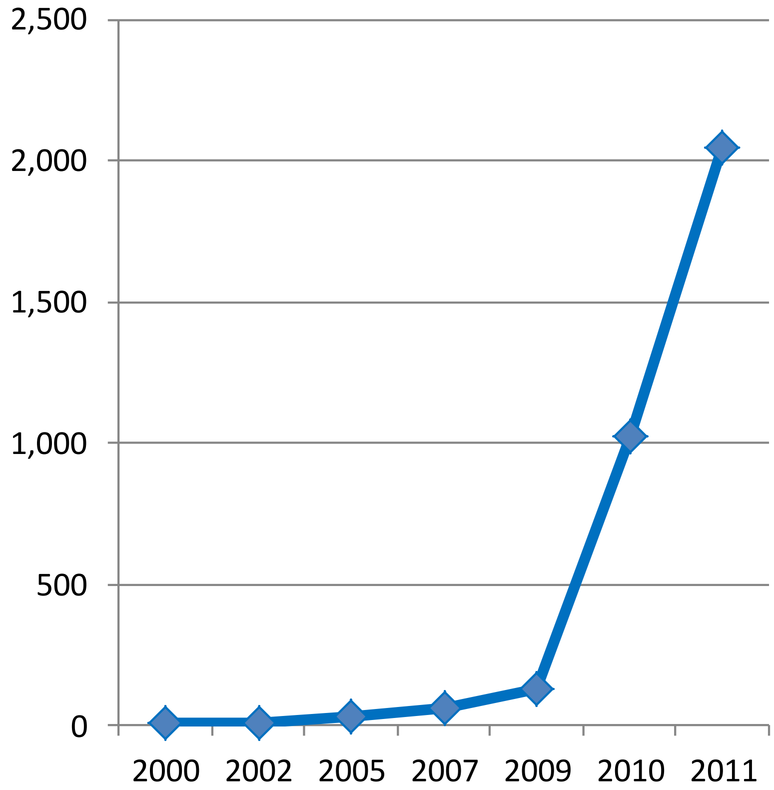
1984 ***2012***

The New Storage Hierarchy



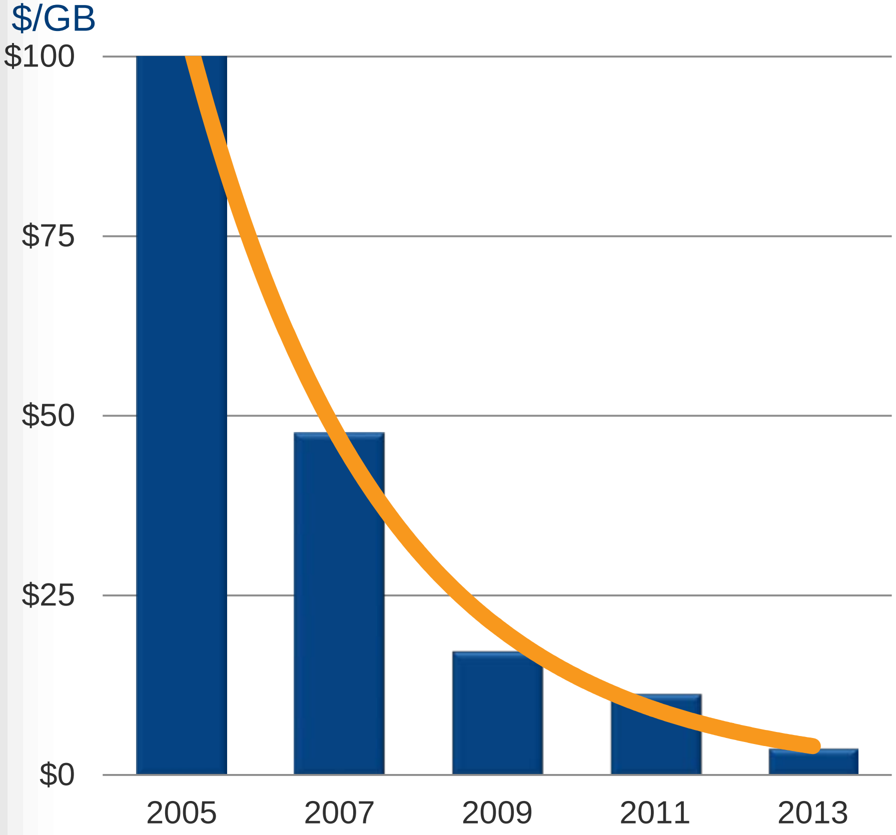
Bigger and Cheaper

Maximum DRAM in GB



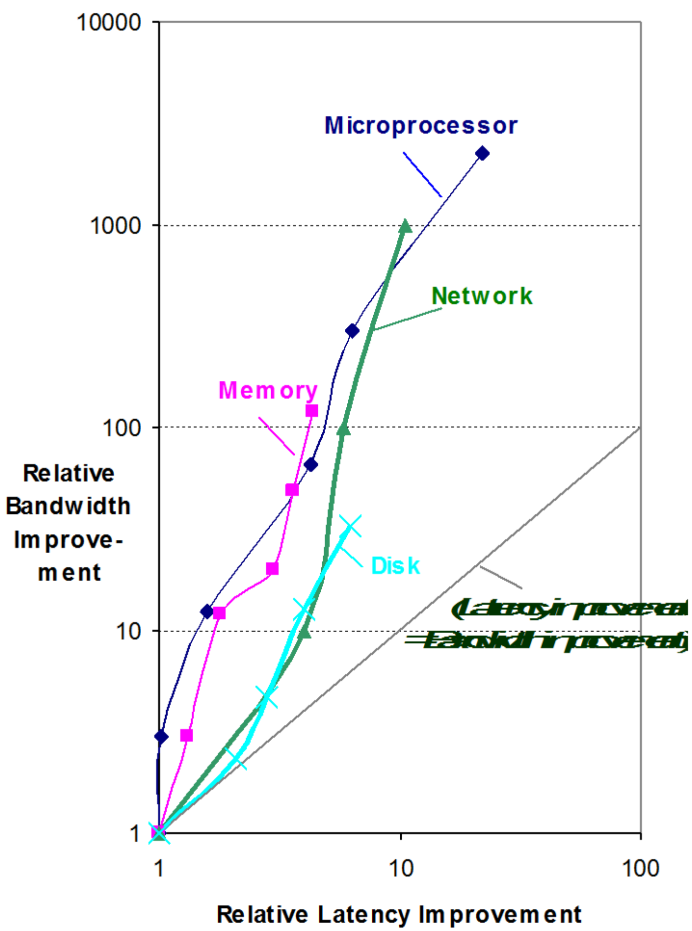
Source: Forrester Research, *The x86 Server Grows Up And Out* (October 8, 2010)

DRAM



Source: Gartner Dataquest, *Forecast: DRAM Market Statistics* (1Q11)

Latency lags Bandwidth*



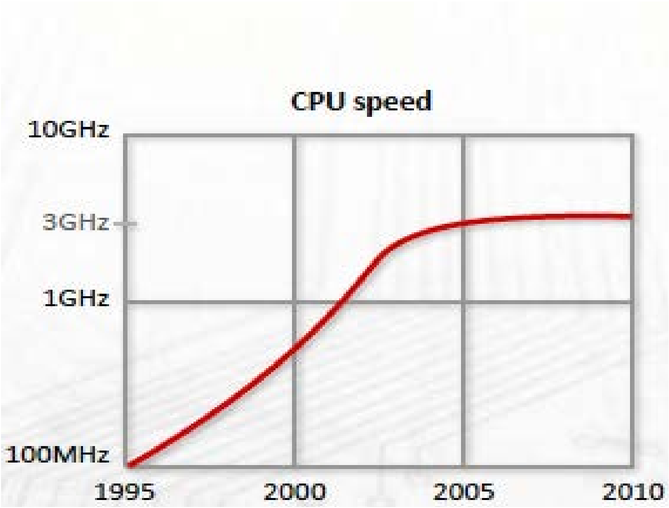
	Mid-1980s	2009	Change
Disk capacity	**30 MB**	**500 GB**	**16667x**
Max. transfer rate	**2 MB/s**	**100 MB/s**	**50x**
Latency (seek & rotate)	**20 ms**	**10 ms**	**2x**

Source: Stanford RamCloud Talk, Feb 2010

**David Patterson, Latency Lags Bandwidth, CACM, 2004*

CPU Scaling: Slow-down of Moore's Law

- CPU Scaling is not exponential anymore
  - 3000% increase from 1995-2004
  - 30-40% increase since 2004



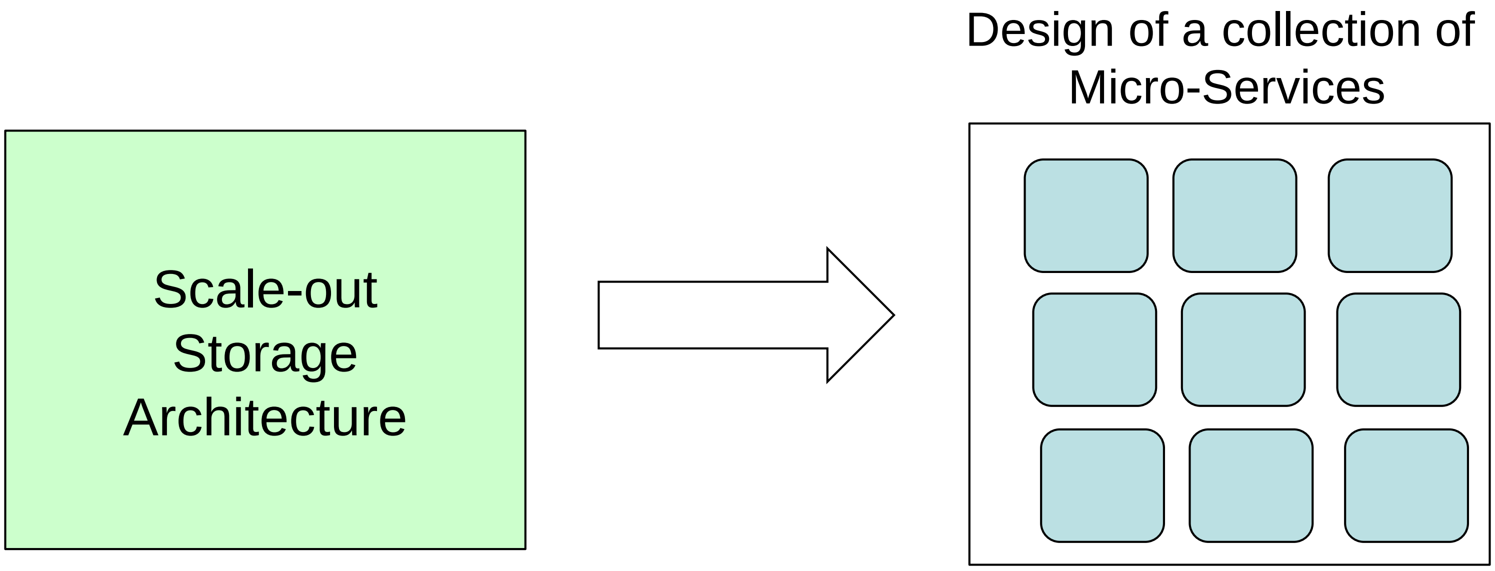
- Multi-core scaling is linear
  - NUMA, locking, sharing latencies, programming models…

- Compute per unit of data is decreasing

Agenda: 3 WHYs

- Why Enterprise Storage → Scale-out Architectures
- Why NVM != <Yet Another Storage Tier>
- Why New Scale-out Design != Clean slate Re-design

What does re-design mean?



Blueprint of a Scale-out Storage Architecture

Object Manager
- Write Buffering
- Distributed Caching
- Lookup
- Update Coordination
- Persistence Model

Namespace Manager
- Resource Mgt
- Sharding/ Placement
- Consistency Primitives
- Data Durability
- HA/ Failover

Cluster Manager
- Group Membership
- Centralized State Mgr.
- Leader Election
- Transport Middleware

Concurrency Control

	Shared Disk	Big Data/ NoSQL	General Purpose Scale-out	In-mem	NVM-tiered (Future)
Pessimistic	GPFS (Distributed LM) VMFS	HDFS	Ceph		
Optimistic w/ retry		Cassandra (CAS) CouchDB (MVCC)		Redis (CAS)	
Optimistic w/ heuristics		Cassandra (Merged Updates)	Swift	Redis (LWW)	

High Availability

***Data Replication*:** ■ **State-based** O **Ops-based** ⊙ **Lineage**

	Shared Disk	Big Data/ NoSQL	General Purpose Scale-out	In-mem	NVM-tiered (Future)
Active-Active		CouchDB ■			
Active-Passive	Lustre (NA) GPFS (NA)		Ceph ■ Swift ■	Redis O	
Active-Backup		Cassandra ■ HBase ■		RamCloud ■ Tachyon ⊙	

Persistence Model

	Shared Disk	Big Data/ NoSQL	General Purpose Scale-out	In-mem	NVM-tiered (Future)
In-place	GPFS Lustre VMFS	MongoDB		Redis	
Out-of-place		HDFS CouchDB	Ceph VMware VSAN	RamCloud	
Versioned		Cassandra HBase	Swift Scality		

Summary

- Why Enterprise Storage → Scale-out Architectures
  - Web 2.0 model to handle exponential data growth
- Why NVM != <*Yet Another Storage Tier*>
  - Holistic shifts across compute, network, memory, storage
- Why New Scale-out Design != Clean slate
  - Piecemeal evolution of micro-services within a Scale-out Architecture.

Questions?

Sandeep Uttamchandani
suttamchandani@vmware.com

Blog: https://blogs.vmware.com/cto/author/suttamchandani/