cloudscaling

Cloud Storage Futures (previously: Designing Private & Public Clouds)

May 22nd, 2012

Randy Bias, CTO & Co-founder

@randybias

Part 1:

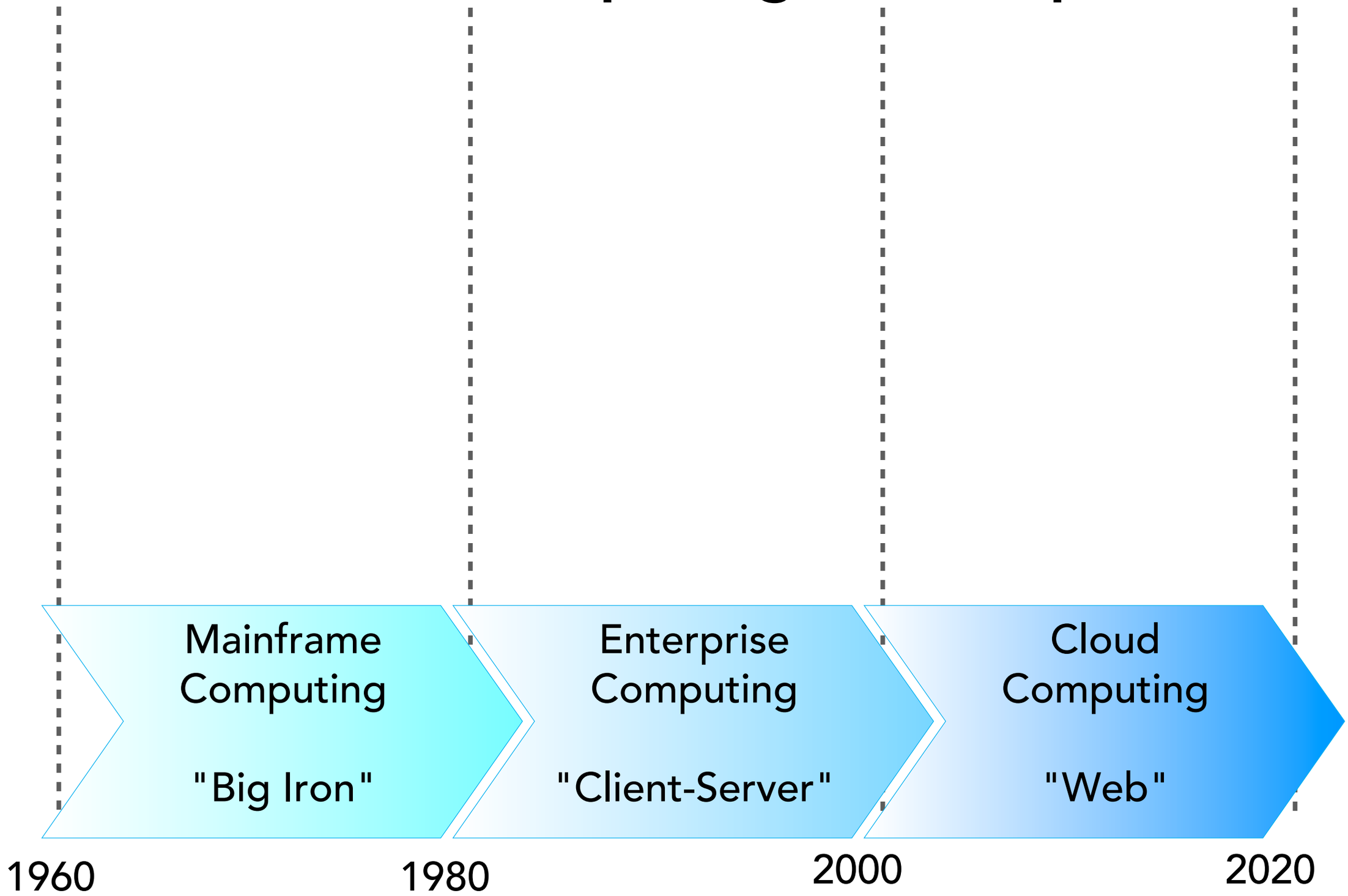
The Two Cloud Architectures

A Story of Two Clouds

... Driven by Two App Types

cloudscaling

Cloud Computing ... Disrupts



cloudscaling

Cloud Computing ... Disrupts

IBM

TANDEM

CONTROL DATA

Mainframe Computing

"Big Iron"

Enterprise Computing

"Client-Server"

Cloud Computing

"Web"

1960 1980 2000 2020

Disruption

Cloud Computing ... Disrupts

IBM

TANDEM

CONTROL DATA

IBM

EMC²

CISCO

Microsoft

Mainframe Computing

"Big Iron"

Enterprise Computing

"Client-Server"

Cloud Computing

"Web"

1960

1980

2000

2020

Disruption

Disruption

Cloud Computing ... Disrupts

IBM

TANDEM

CONTROL DATA

IBM

EMC²

CISCO

Microsoft

Google Apps

amazon web services™

nexenta
Enterprise class storage for everyone

Cassandra

openstack

hadoop

Mainframe Computing

"Big Iron"

Enterprise Computing

"Client-Server"

Cloud Computing

"Web"

1960 — 1980 — 2000 — 2020

Disruption — Disruption — Disruption

IT – Evolution of Computing Models

SLA

Scaling

Hardware

HA Type

Software

Consumption

IT – Evolution of Computing Models

SLA	99.999
Scaling	Vertical
Hardware	Custom
HA Type	Hardware
Software	Centralized
Consumption	Centralized Service

IT – Evolution of Computing Models

SLA	99.999	99.9
Scaling	Vertical	Horizontal
Hardware	Custom	Enterprise
HA Type	Hardware	Software
Software	Centralized	Decentralized
Consumption	Centralized Service	Shared Service

Mainframe "big-iron" → Enterprise "client/server" → Cloud "scale-out"

IT – Evolution of Computing Models

SLA	99.999	99.9	Always On
Scaling	Vertical	Horizontal	
Hardware	Custom	Enterprise	Commodity
HA Type	Hardware	Software	
Software	Centralized	Decentralized	Distributed
Consumption	Centralized Service	Shared Service	Self-service

Mainframe "big-iron" → Enterprise "client/server" → Cloud "scale-out"

Enterprise Computing
(existing apps built in silos)

Cloud Computing
(new elastic apps)

Scale-out apps require elastic infrastructure

	Traditional apps	Elastic cloud-ready apps
APPS	SAP, ORACLE, Windows, VB.net, Microsoft .net	HTML5, JS, PHP, BIG DATA, iOS, Android, Chrome
INFRA	nexenta Enterprise class storage for everyone, vmware, SAVVIS, EMC², terremark	amazon web services™, rackspace, Google

Scale-out Cloud Technology

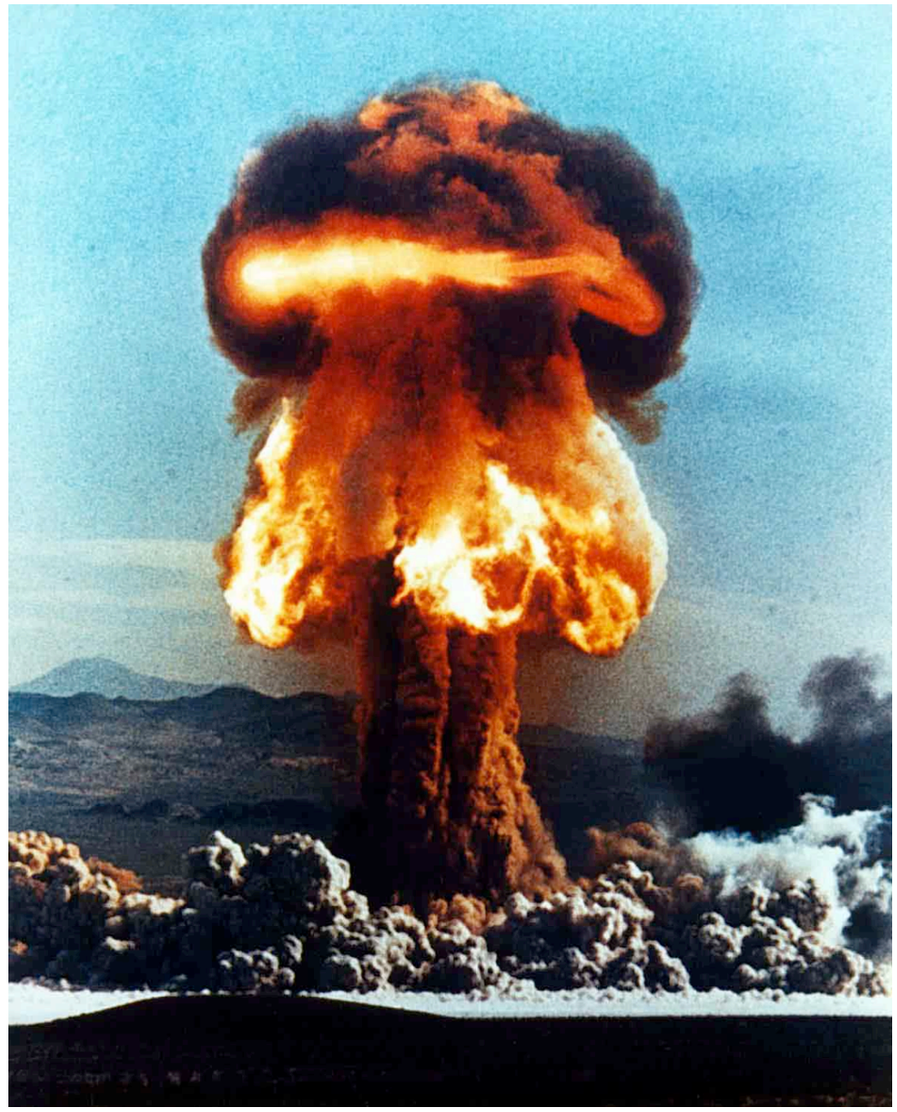
	Traditional apps	Elastic cloud-ready apps
APPS	SAP, ORACLE, Windows, VB.net, Microsoft .net	HTML, JS, PHP, BIG DATA, iOS
INFRA	nexenta Enterprise class storage for everyone, vmware, SAVVIS, EMC², terremark	openstack™, nexenta Enterprise class storage for everyone, ARISTA, EUCALYPTUS

Scale-out Principles

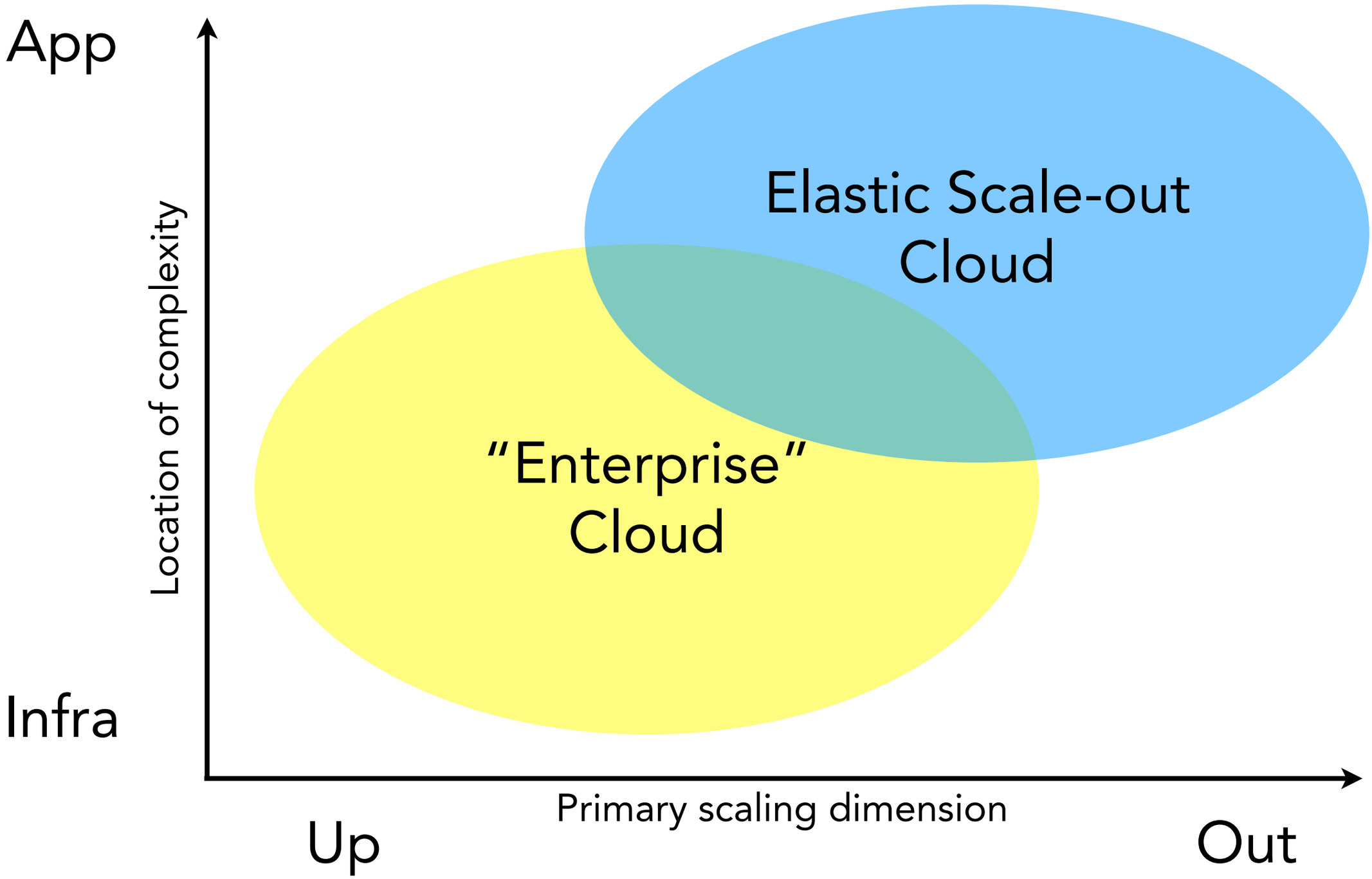
- Small failure domains
- Risk acceptance vs. risk mitigation
- More boxes for throughput & redundancy
- Assume app manages complexity:
  - Data replication
  - Assumes infrastructure is unreliable:
    - Server & data redundancy
  - Geo-distribution
  - Auto-scaling

What's a failure domain?

- "Blast radius" during a failure
- What is impacted?
- Public SAN failures:
  - FlexiScale SAN failure in 2007
  - UOL Brazil in 2011:
    - http://goo.gl/8ct9n
  - There are many more
- Enterprise HA 'pairs' typically support BIG failure domains



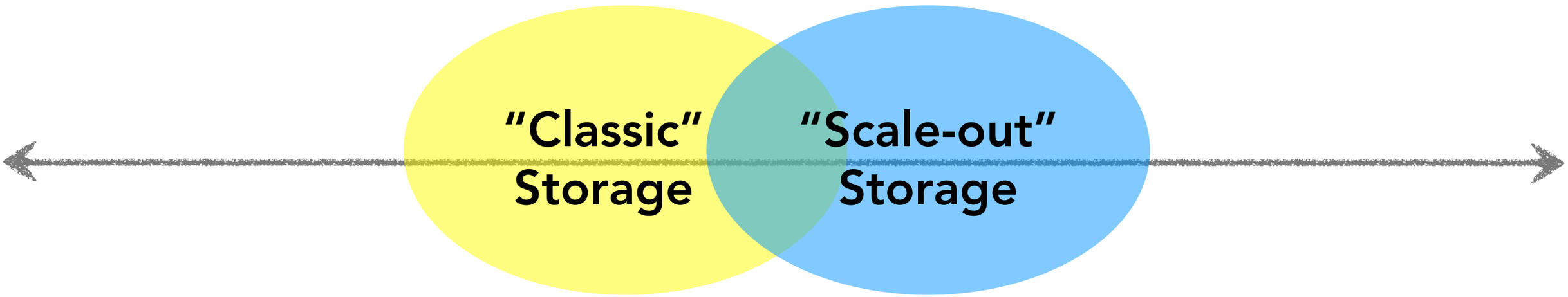
Two Diff Arches for Two Kinds of Apps



Part 2:

Storage Architectures Are Changing

Two Diff Storages for Two Kinds of Clouds



Uptime in Infra
Every part is redundant
Data mgmt in Infra
Bigger SAN/NAS/DFS

Uptime in apps
Minimal h/w redundancy
Data mgmt in apps
Smaller failure domains

Difference in Tiers

Tier	$	Purpose	Classic	Scale-out
1	$$$$	Mission Critical	•SAN, then NAS •10-15K RPM •SSD	•On-demand SAN (EBS) •DynamoDB (AWS) •Variable service levels
2	$$	Important	•NAS then SAN •7.2K RPM	•DAS •App / DFS to scale out
3	$	Archive & Backups	•Tape •Nearline 5.4K	•Object Storage

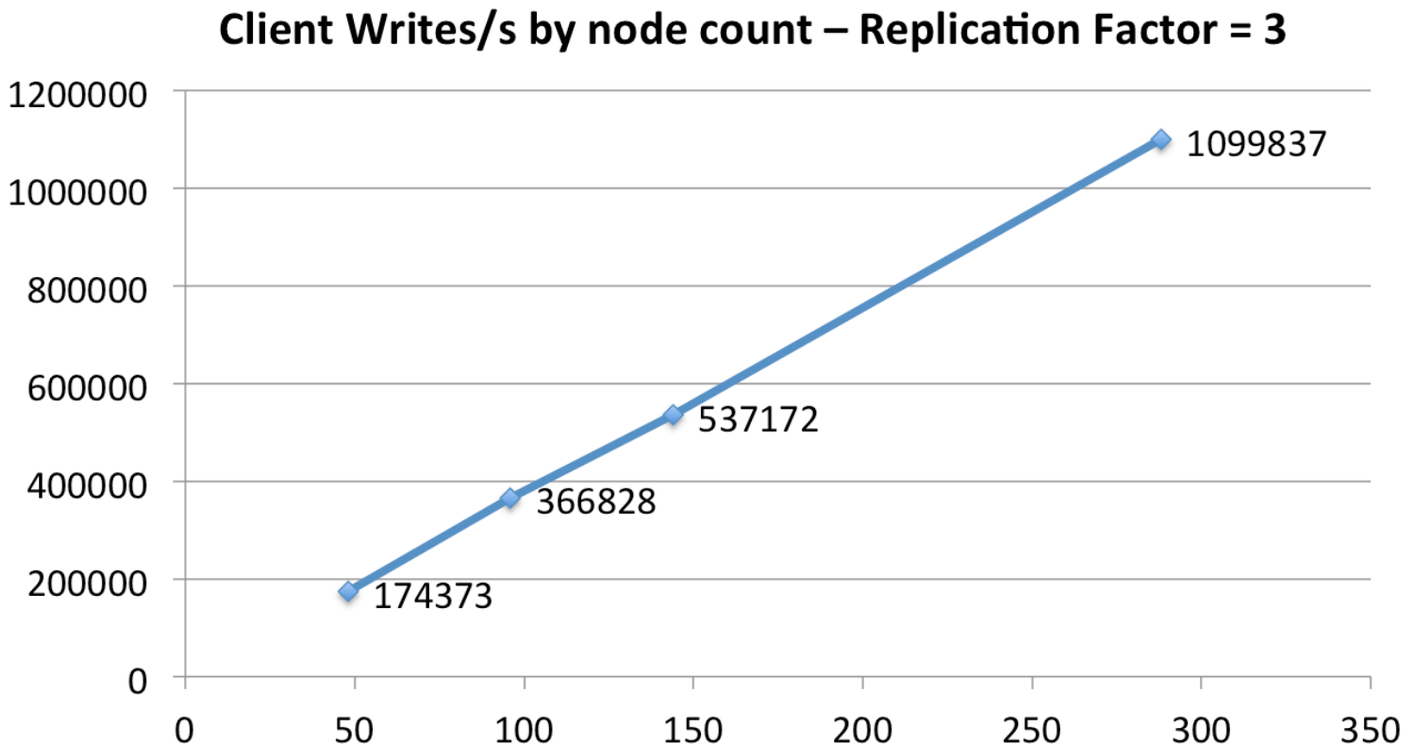
The Biggest Difference is in Where Data Management Resides

- In scale-out systems, apps are managing the data:
  - Riak / Scale-out distributed data store
  - Hadoop+HDFS / Scale-out distributed computation systems
  - Cassandra / Scale-out distributed columnar database

Cassandra / Netflix use case

- 3 x Replication
- Linearly scaling performance
  - 50 - 300 nodes
- > 1M writes/second

- When is this perfect?
  - data size unknown
  - growth unknown
  - lots of elastic dynamism



Cassandra / Netflix use case

- DAS ('ephemeral store')
- Per node perf is constant
  - disk
  - CPU
  - network
- Client write times constant
- Nothing special here

Per Node Activity

Per Node	48 Nodes	96 Nodes	144 Nodes	288 Nodes
Per Server Writes/s	10,900 w/s	11,460 w/s	11,900 w/s	11,456 w/s
Mean Server Latency	0.0117 ms	0.0134 ms	0.0148 ms	0.0139 ms
Mean CPU %Busy	74.4 %	75.4 %	72.5 %	81.5 %
Disk Read	5,600 KB/s	4,590 KB/s	4,060 KB/s	4,280 KB/s
Disk Write	12,800 KB/s	11,590 KB/s	10,380 KB/s	10,080 KB/s
Network Read	22,460 KB/s	23,610 KB/s	21,390 KB/s	23,640 KB/s
Network Write	18,600 KB/s	19,600 KB/s	17,810 KB/s	19,770 KB/s

Node specification – Xen Virtual Images, AWS US East, three zones

- Cassandra 0.8.6, CentOS, SunJDK6
- AWS EC2 m1 Extra Large – Standard price $ 0.68/Hour
- 15 GB RAM, 4 Cores, 1Gbit network
- 4 internal disks (total 1.6TB, striped together, md, XFS)

Cassandra / Netflix use case

- On-demand & app-managed
- Cost per GB/hr: $.006
- Cost per GB/mo: $4.14
- Includes: storage, DB, storage admin, network, network admin, etc. etc. etc.

Time is Money

	48 nodes	96 nodes	144 nodes	288 nodes
Writes Capacity	174373 w/s	366828 w/s	537172 w/s	1,099,837 w/s
Storage Capacity	12.8 TB	25.6 TB	38.4 TB	76.8 TB
Nodes Cost/hr	$32.64	$65.28	$97.92	$195.84
Test Driver Instances	10	20	30	60
Test Driver Cost/hr	$20.00	$40.00	$60.00	$120.00
Cross AZ Traffic	5 TB/hr	10 TB/hr	15 TB/hr	30[1] TB/hr
Traffic Cost/10min	$8.33	$16.66	$25.00	$50.00
Setup Duration	15 minutes	22 minutes	31 minutes	66[2] minutes
AWS Billed Duration	1hr	1hr	1 hr	2 hr
Total Test Cost	$60.97	$121.94	$182.92	$561.68

[1] Estimate two thirds of total network traffic
[2] Workaround for a tooling bug slowed setup

Part 3:

Scale-out Storage ... Now & Future

Only Change is Certain

There are a few basic approaches being taken ...

Scale-out SAN

DAS+DFS

Block-devices-as-a-Service (EBS)

DAS+BigData

DAS+Database

Scale-out SAN

- In-rack SAN == faster, bigger DAS w/ better stat-muxing
- Accept normal DAS failure rates
- Assume app handles data replication
- Like AWS ‘ephemeral storage’
- KT architecture
- *Customers didn’t “get it”*
  - “Ephemeral SAN” not well understood

Dedicated Storage SW

9K Jumbo Frames

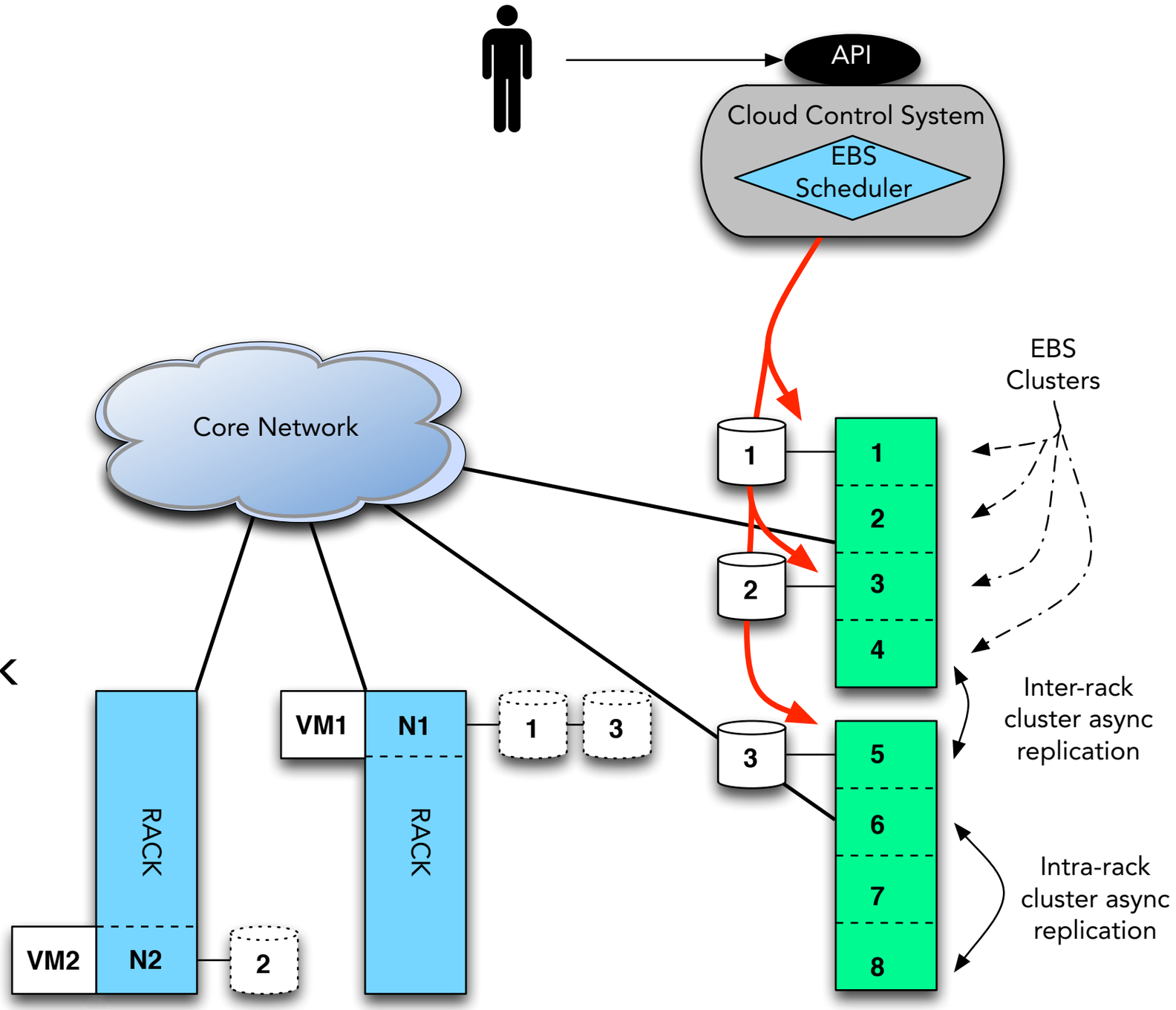
SSD caches (ZIL/L2ARC)

No replication

Max HW redundancy

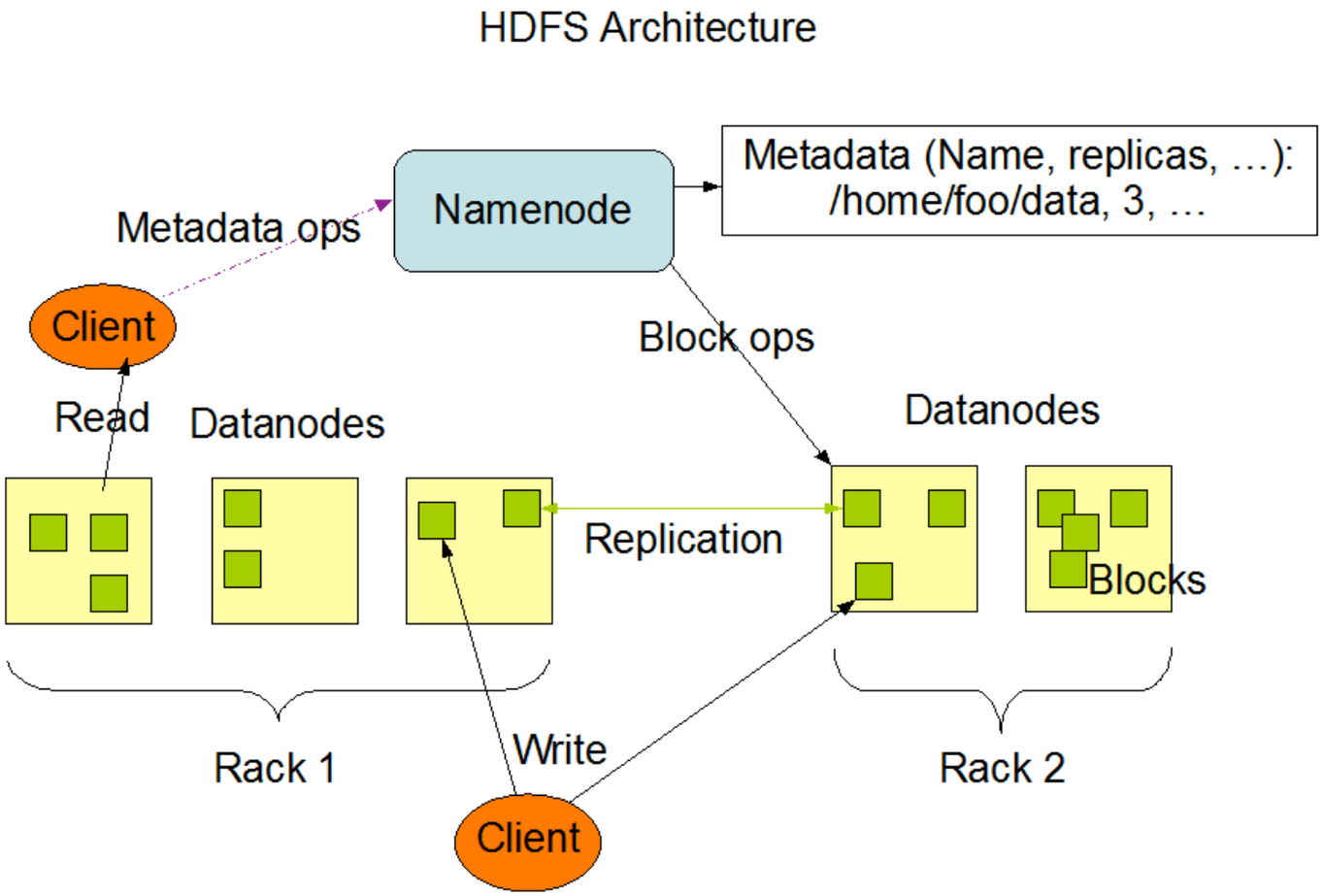
AWS EBS - “Block-devices-as-a-Service”

- Scale-out SAN (sort of)
- Block scheduler
- Async replication
  - Some failure tolerance
- Scheduler:
  - Allocates customer block devices across many failure domains
- Customer run RAID inside VMs to increase redundancy



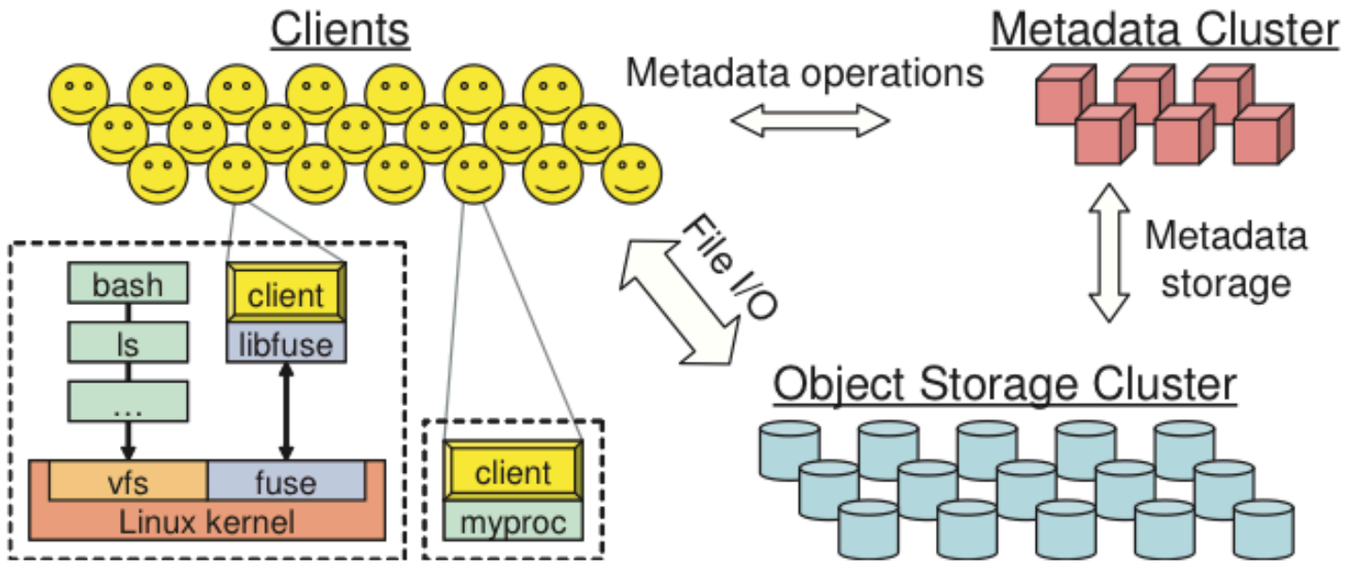
DAS + Big Data
(Storage + Compute + DFS)

- Storage capability:
  - Replication
  - Disk & server failure
  - Data rebalancing
  - Data locality
    - rack awareness
  - Checksums (basic)
- Also:
  - Built in computation



Distributed File Systems (DFS) over DAS

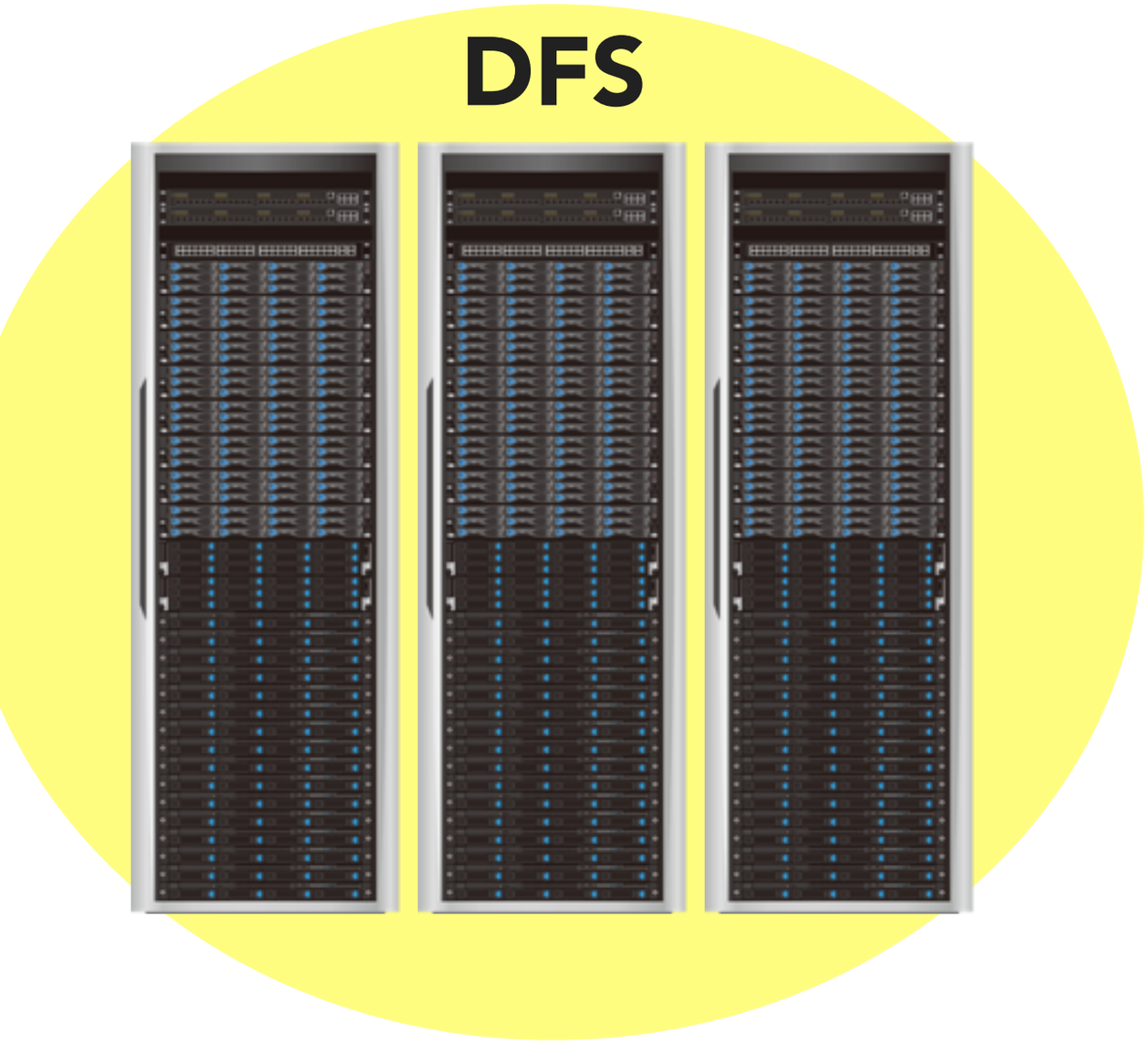
- Storage capability:
  - Replication
  - Disk & server failure
  - Data rebalancing
  - Checksums (w/ btrfs)
  - Block devices
- Also:
  - No computation



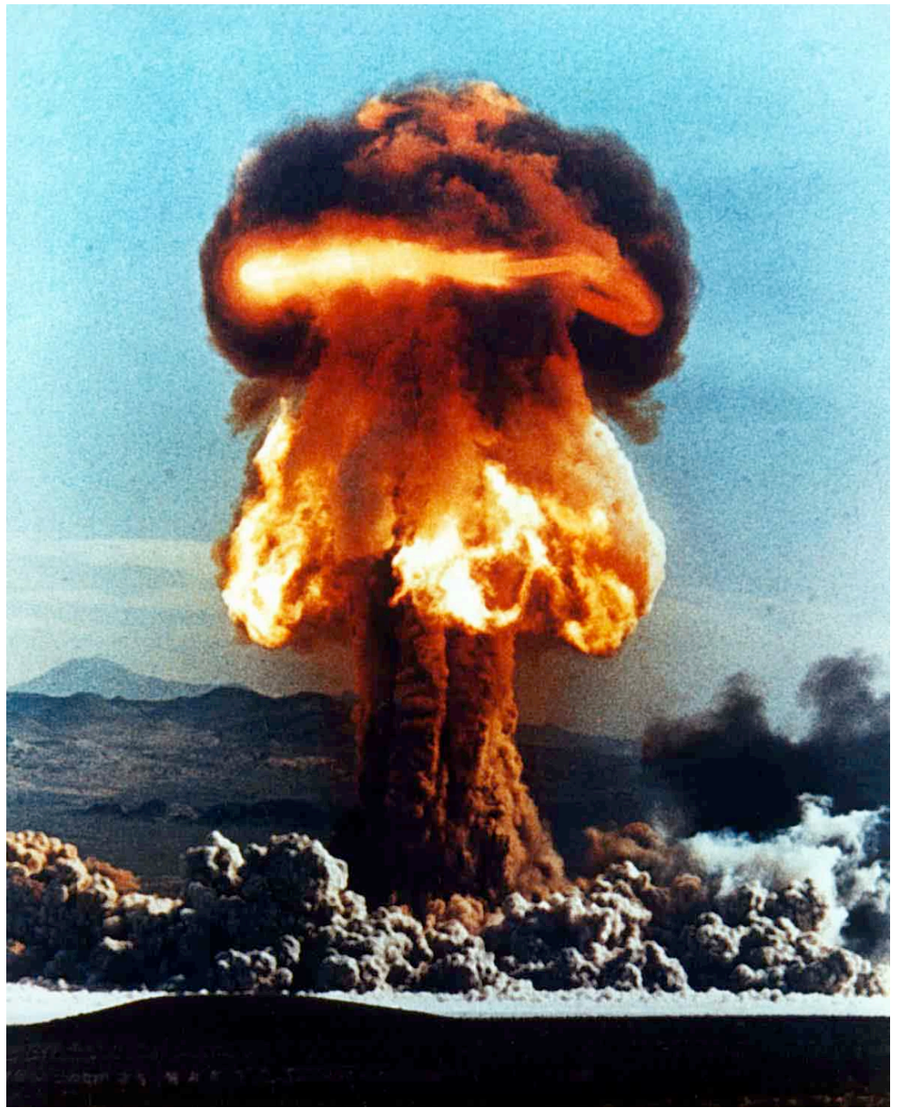
ceph architecture✴

✴ Looks familiar doesn't it?

Why is DFS at the Physical Layer Dangerous for Scale-out?

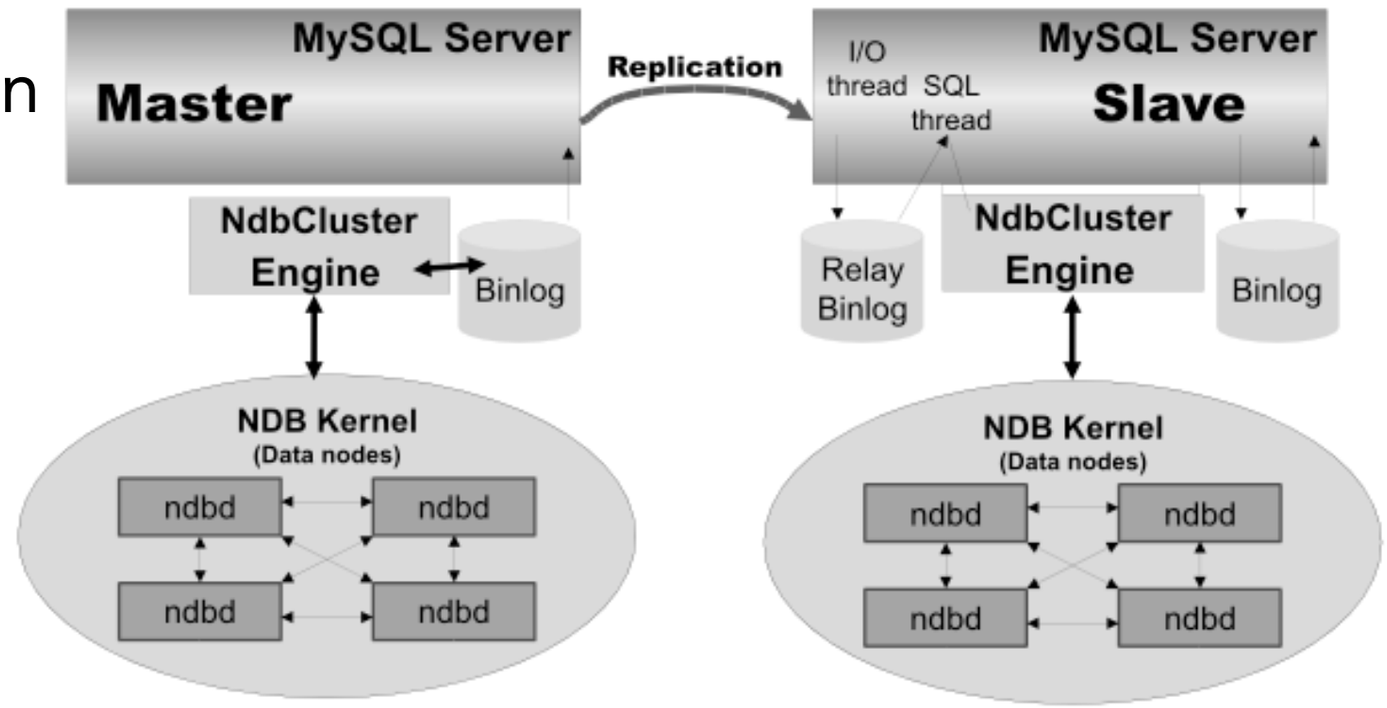


==



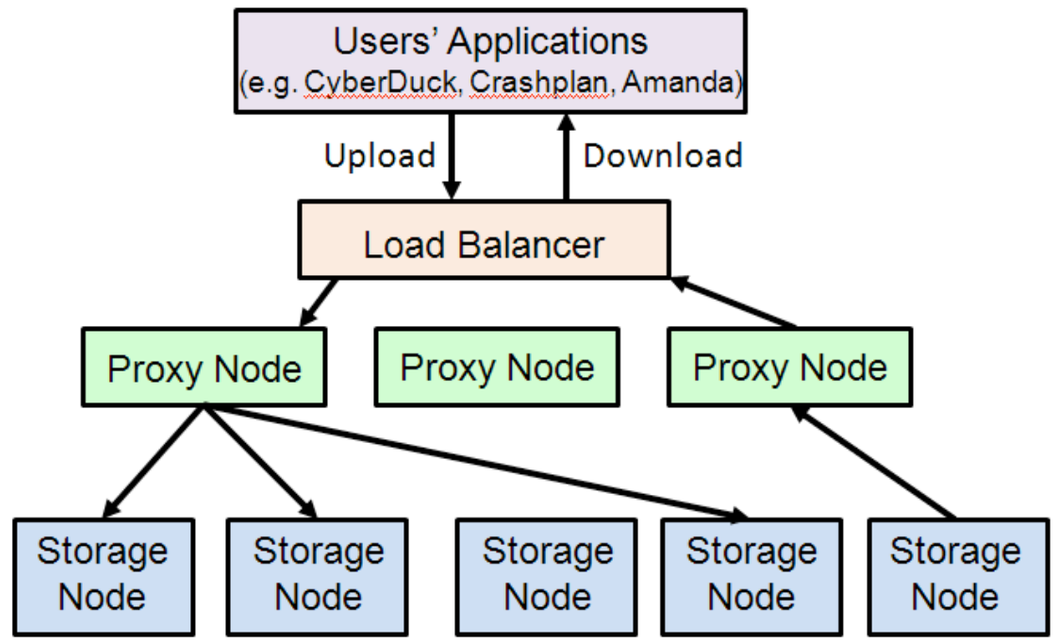
DAS + Database Replication / Scaling

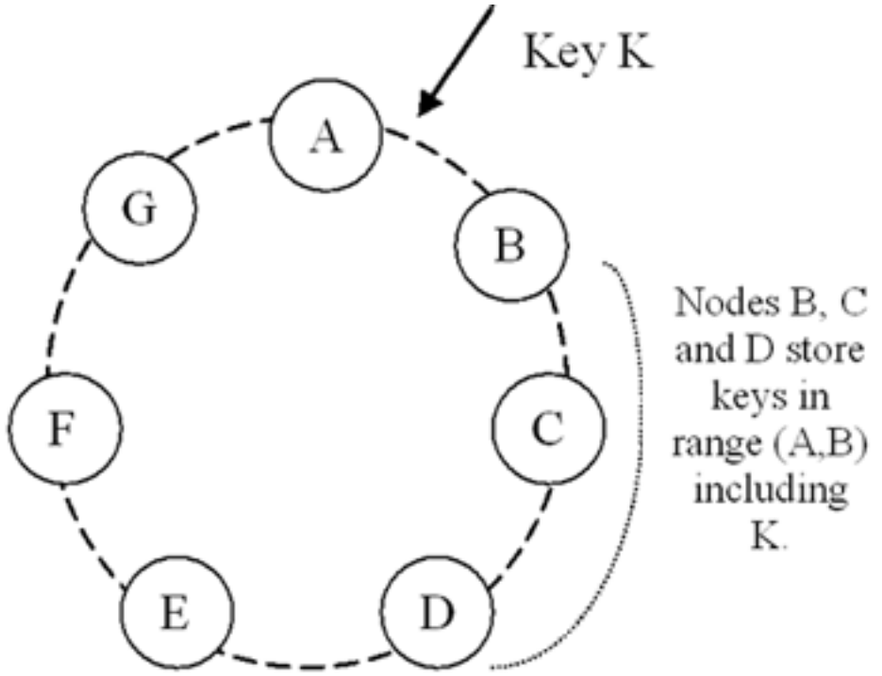
- Storage capability:
  - Async/Sync Replication
  - Server failure
  - Checksums (sort of)
- Also:
  - Std RDBMS
  - SQL i/f
  - Well understood



Object Storage

- Storage capability:
  - Replication
  - Disk & server failure
  - Data rebalancing
  - Checksums (sometimes)
- Also:
  - Looks like a big web app
  - Uses a DHT/CHT to ‘index’ blobs
  - Very simple





Where does OpenStorage Fit?

Scale-out Solution	Purpose / Tier	Virtual or Physical?	Fit
Scale-out SAN	Tier-1/2	Physical	In-rack SAN
EBS	Tier-1	Physical	EBS Clusters (scale-out SAN)
DAS+BigData	Tier-2	Virtual	Reliable, bit-rot resistant DAS
DAS+DFS	Tier-2	Physical / Virtual	Reliable, bit-rot resistant DAS (unproven)
DAS+DB	Tier-2	Virtual	In-VM reliable DAS
Object Storage	Tier-3	Physical	Reliable, bit-rot resistant DAS

Summarizing ZFS Value in Scale-out

- Data integrity & bit rot an issue that few solve today
- Most SAN/NAS solutions don't 'scale down'
- Commodity x86 servers are winning
- There are two scale-out places ZFS wins:
  - Small SAN clusters
  - Best DAS management

Summary

Conclusions / Speculations

- Build the right cloud
- Which means the right storage for *that* cloud
- A single cloud ***might*** support both ...
- Open storage can be used for both ...
- ... WITH the appropriate design/forethought

cloudscaling

Q&A

@randybias