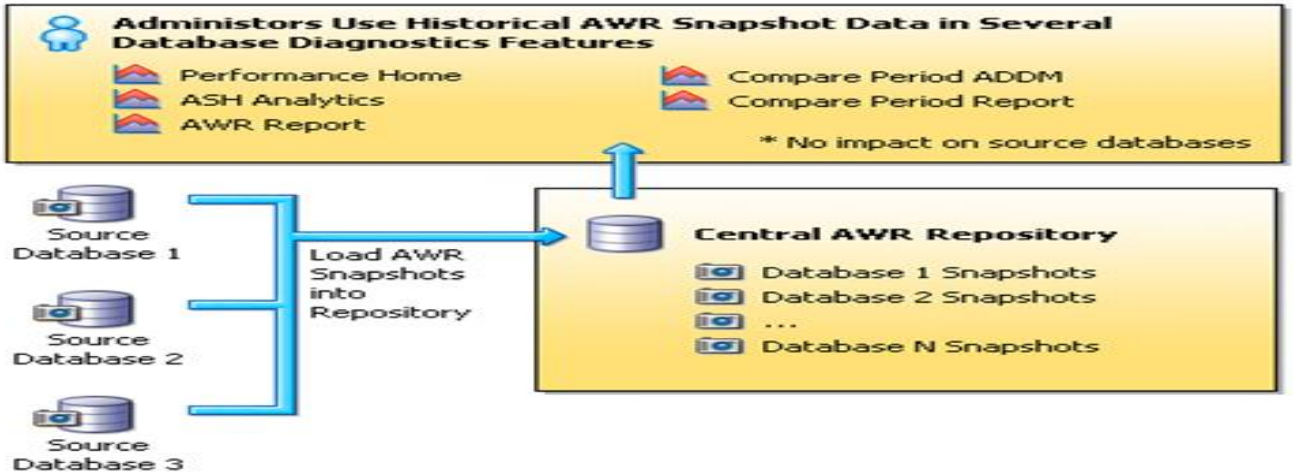
ORACLE®

General announcements

- In-Memory is available next month
- http://www.oracle.com/us/corporate/events/dbim/index.html
- X4-2 Exadata announced (well actually around Jan 1)
- OEM/Grid control 12c R4 just released
  - **AWR Warehouse in EM12c Rel. 4 !!!!**

AWR Architecture



bscvisitor/BSCunyoug

Tanel Podor

- low value-density data moves slowly towards hadoop, high value-density data towards in-memory DB

ORACLE®

NoSQL databases

Bryan Grenn
Principal Engineered systems sales consultant

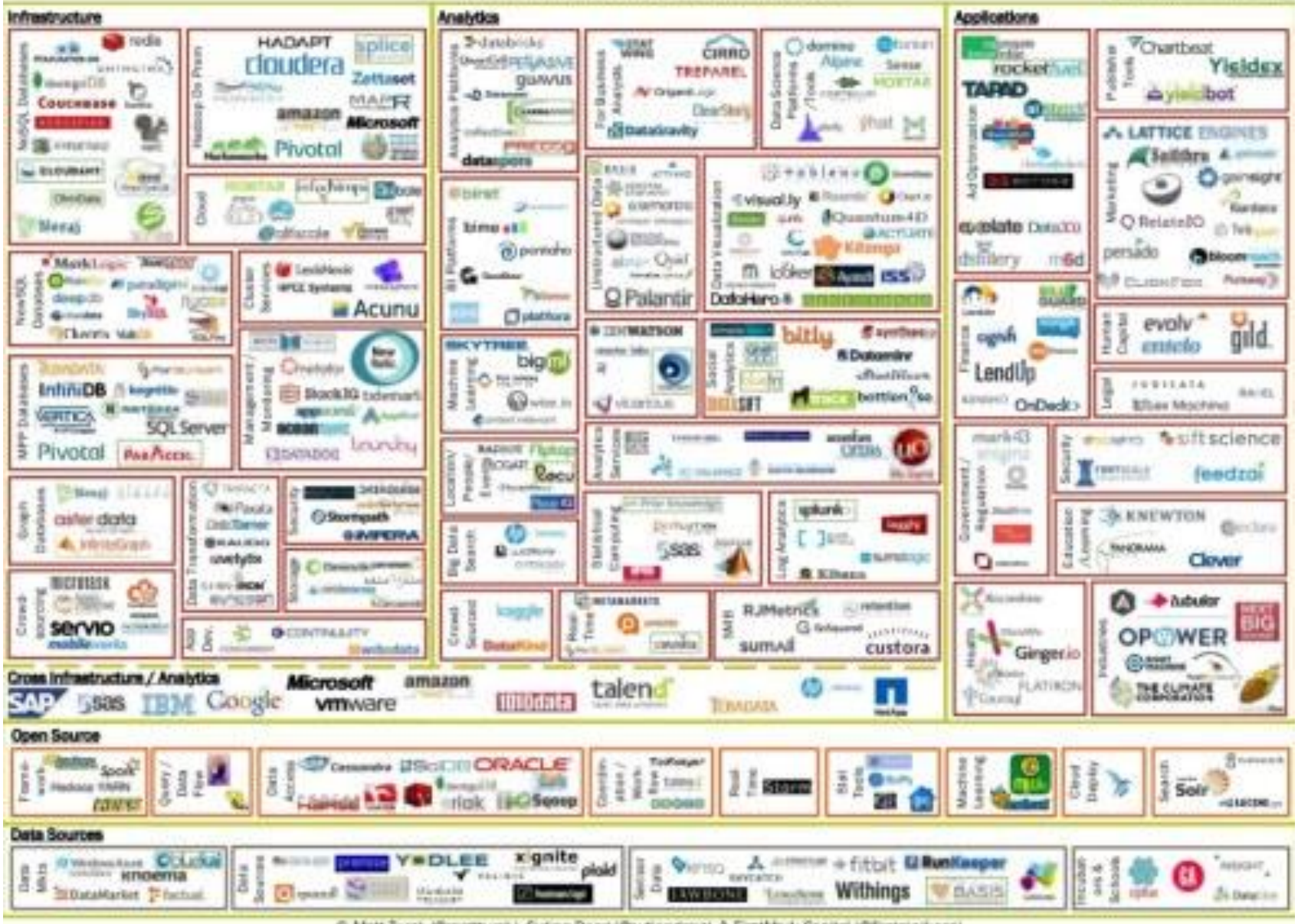
ORACLE® PARTNERNETWORK

Specialized. Recognized by Oracle.
Preferred by Customers.

Agenda

- Big data landscape
- Hadoop vs NoSQL
- NoSQL background
- NoSQL architecture
- Why noSQL
- What is NoSQL
- Use cases
- RDF and Graph databases
- Why you should know this

BIG DATA LANDSCAPE, VERSION 3.0

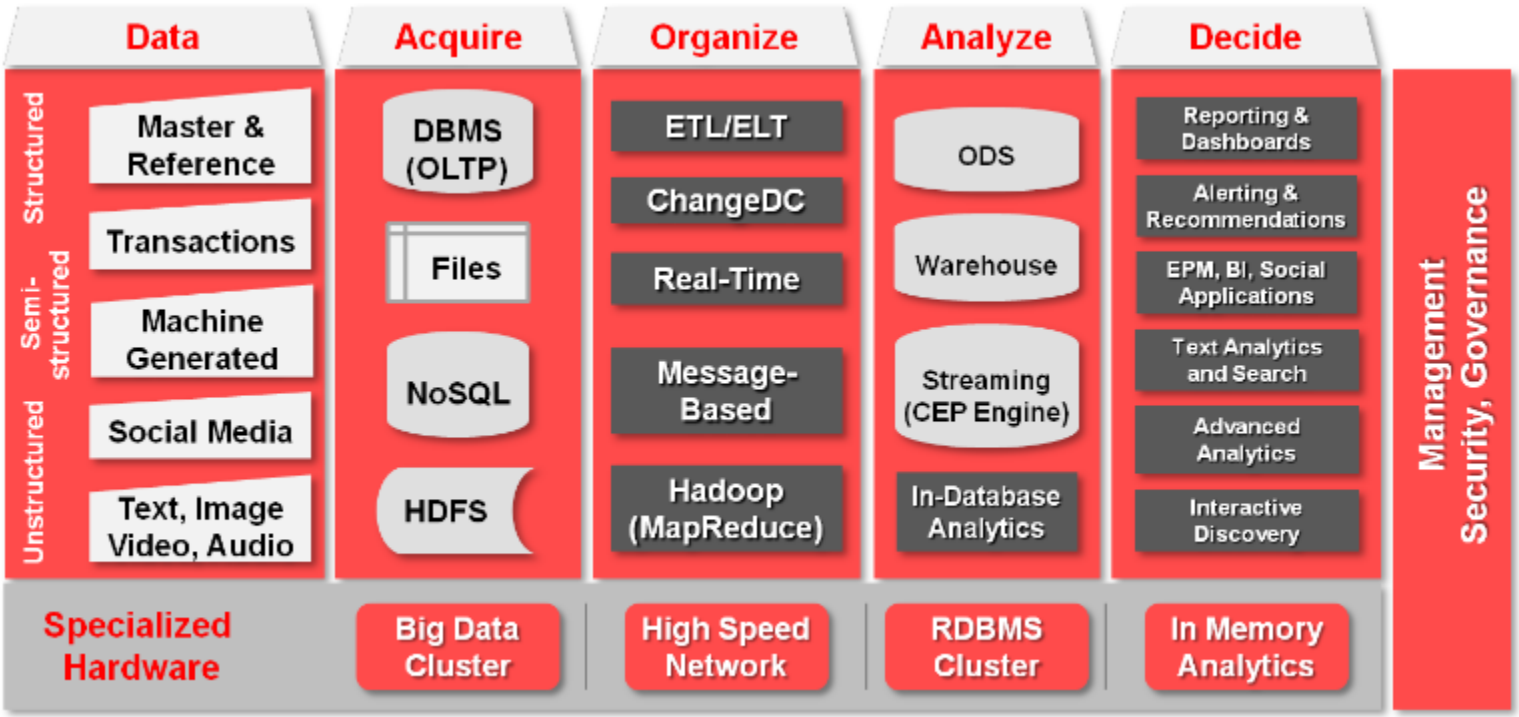


© Matt Turck (@mattturck), Susan Dong (@susandong) & FirstMark Capital (@firstmarkcap)

What is Big Data

4 V of Big data

- Volume
- Velocity
- Variety (no defined structure)
- Value (valuable as an aggregate)


Data
Acquire
Organize
Analyze
Decide
Structured
Semi-structured
Unstructured
Master & Reference
Transactions
Machine Generated
Social Media
Text, Image Video, Audio
DBMS (OLTP)
Files
NoSQL
HDFS
ETL/ELT
ChangeDC
Real-Time
Message-Based
Hadoop (MapReduce)
ODS
Warehouse
Streaming (CEP Engine)
In-Database Analytics
Reporting & Dashboards
Alerting & Recommendations
EPM, BI, Social Applications
Text Analytics and Search
Advanced Analytics
Interactive Discovery
Management Security, Governance
Specialized Hardware
Big Data Cluster
High Speed Network
RDBMS Cluster
In Memory Analytics

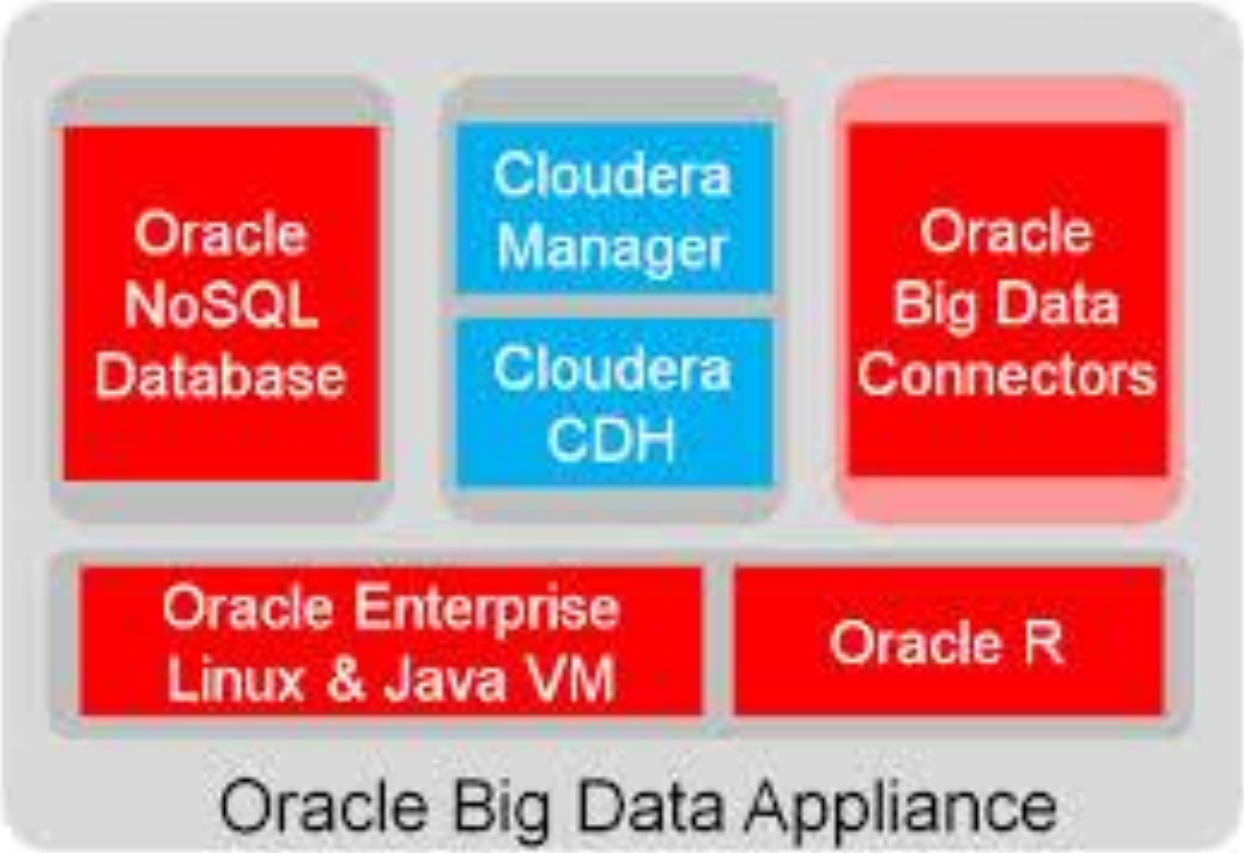
Big Data Appliance

Big Data Appliance X4-2 Hardware		
Full Rack	**Starter Rack**	**In-Rack Expansion**
18 x compute/storage nodes	6 x compute/storage nodes	6 x compute/storage nodes

Per Node:

- 2 x Eight-Core Intel ® Xeon ® E5-2650 V2 Processors
- 64 GB Memory (individual nodes can be expanded to 512 GB)
- Disk Controller HBA with 512MB Battery backed write cache
- 12 x 4TB 7,200 RPM High Capacity SAS Disks
- 2 x QDR (40Gb/s) Ports
- 4 x 10 Gb Ethernet Ports
- 1 x ILOM Ethernet Port

Oracle Big Data Appliance



Hadoop in a nutshell

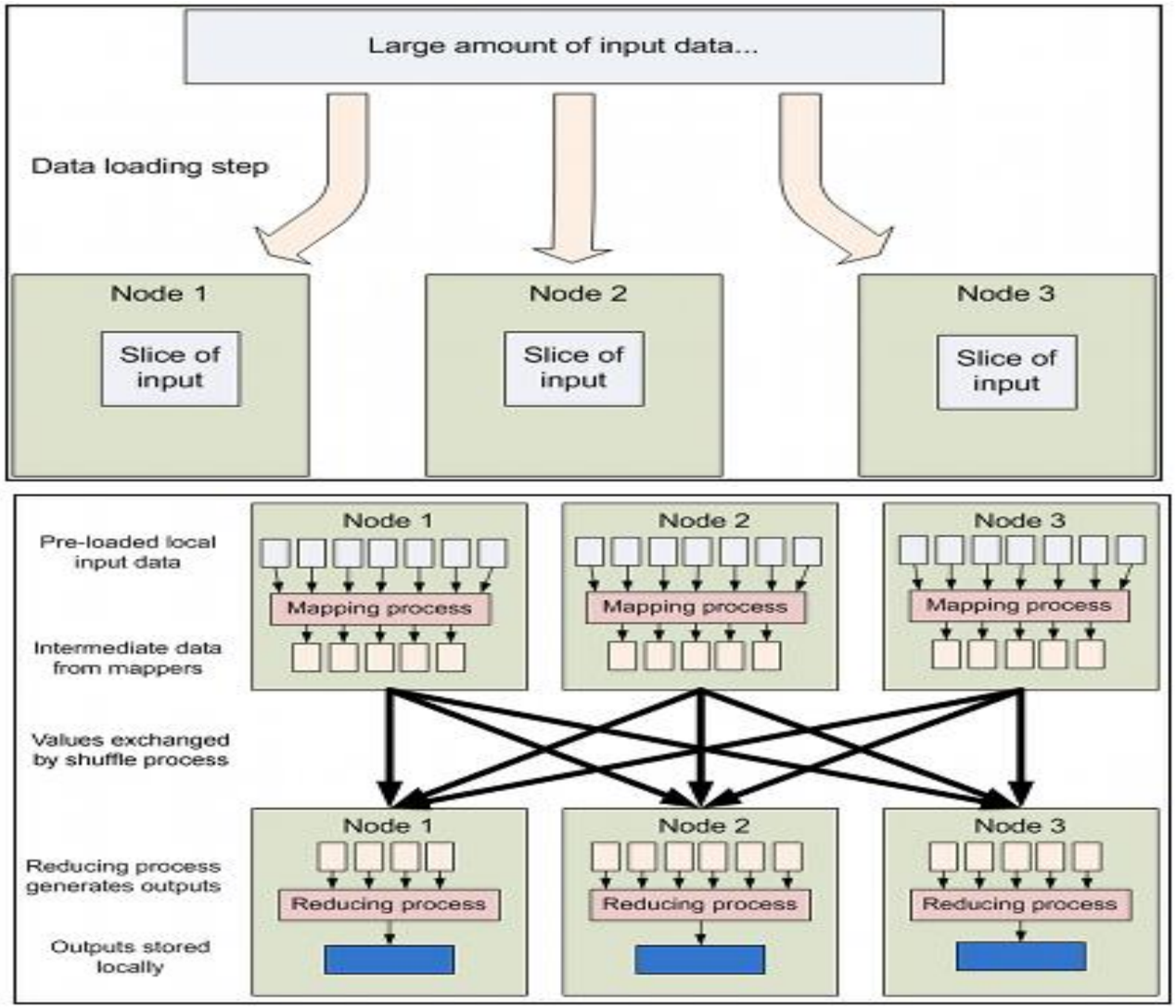
Hadoop is a file system

Hadoop is used to store large amounts of data

Hadoop is searched as a whole for data, or to summarize data using map-reduce jobs

Hadoop does not use indexes.

HDFS (***Hadoop Distributed File System)***



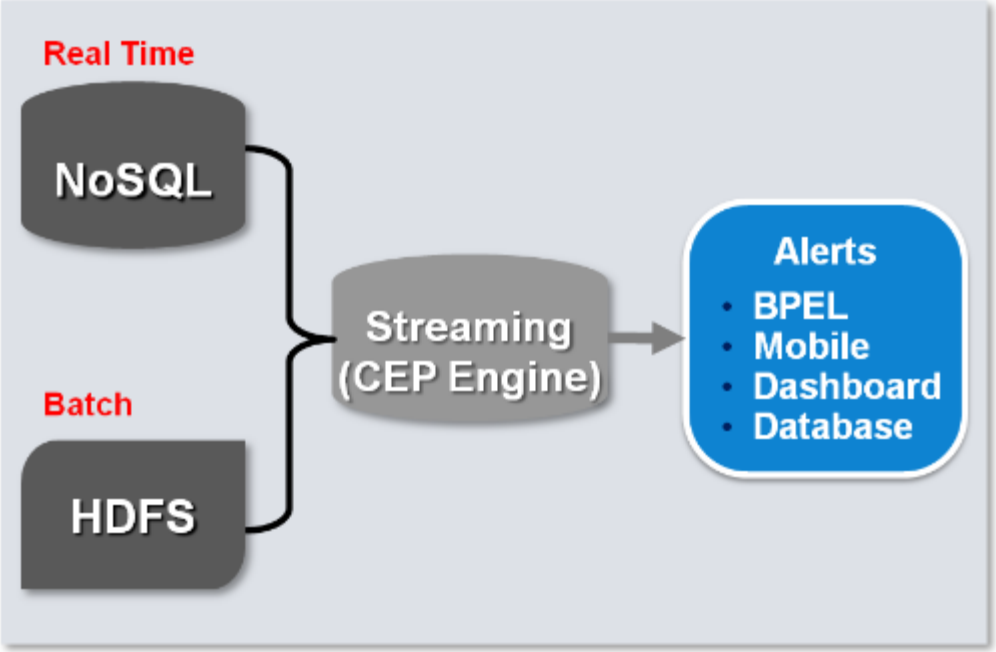
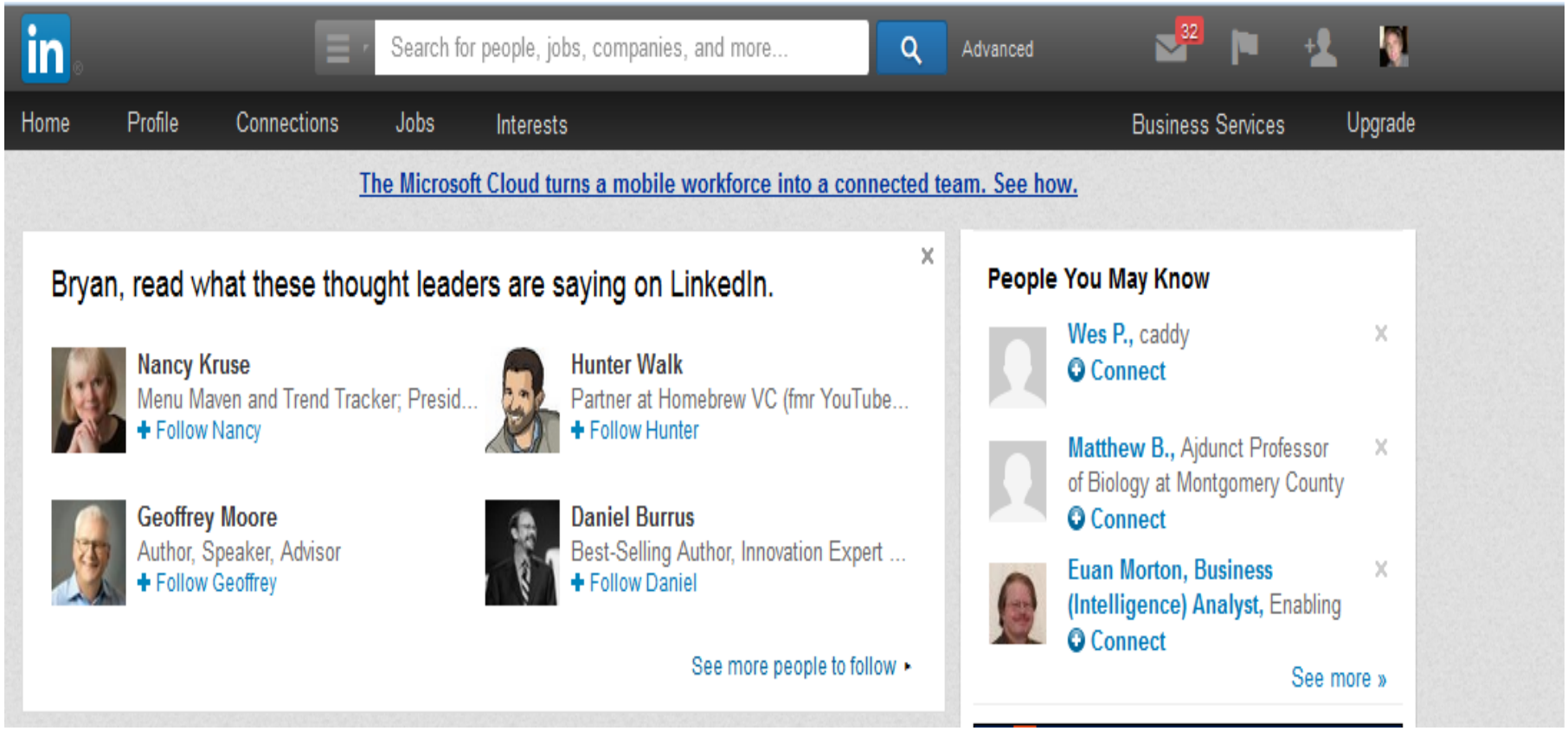


Figure 6: Use Case #2: Big Data for Complex Event Processing

Hadoop and NoSQL

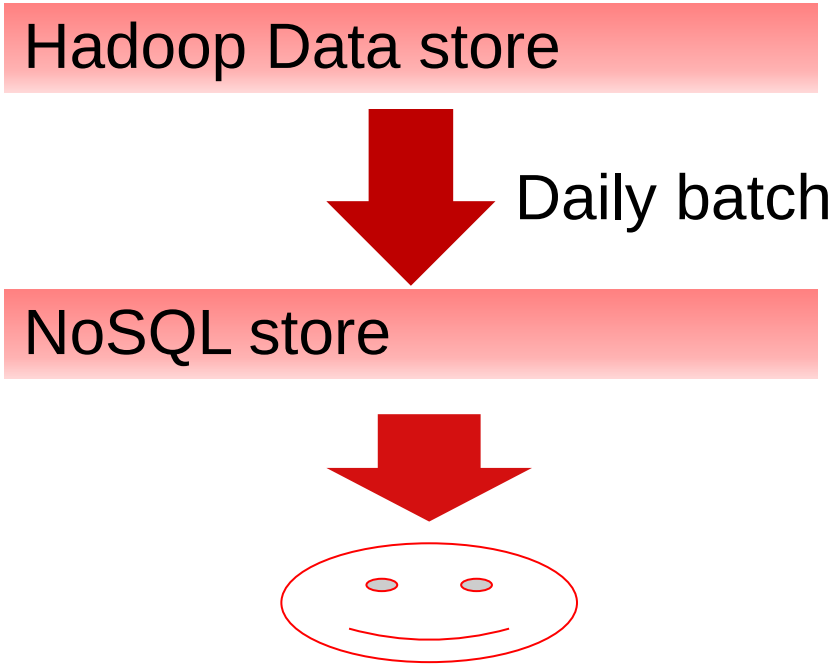
Hadoop and NoSQL are complementary technologies



For Linkedin

Hadoop - All data is stored is stored in Hadoop

NoSQL – Recommendations are calculated daily from Hadoop and loaded in NoSQL



Agenda

- NoSQL Background

Where did the name NoSQL come from ?

2009

Used to **organize an event to discuss open-source distributed databases.**

Doesn't mean anything specific

Sometimes means No SQL

Sometimes "not only SQL"

What is a NoSQL database

Uses one of the following data Models

- Key-Value
- Columnar
- Document
- Graph

Examples of noSQL database

- Oracle NoSQL
- Cassandra
- Voldemort
- MongoDB

Characteristics of NoSQL

- Database
- Simple Data Structure
- Real-time
- Delivers a service
- Fast access to specific records
- Read, write, delete, update

CAP Theorem

- Consistency
- Availability
- Partitioning

Pick Only 2.

Visual Guide to NoSQL Systems

Availability:
Each client can always read and write.

A

Data Models

- Relational (comparison)
- Key-Value
- Column-Oriented/Tabular
- Document-Oriented

CA

RDBMSs (MySQL, Postgres, etc)

Aster Data
Greenplum
Vertica

AP

Dynamo
Voldemort
Tokyo Cabinet
KAI

Cassandra
SimpleDB
CouchDB
Riak

Pick Two

C — P

Consistency:
All clients always have the same view of the data.

CP

BigTable
Hypertable
Hbase

MongoDB
Terrastore
Scalaris

Berkeley DB
MemcacheDB
Redis

Partition Tolerance:
The system works well despite physical network partitions.

When to use a RDBMS

- High-value, High-density, complex data
- Complex data relationships
- Joins
- Designed to scale up
- Well Defined

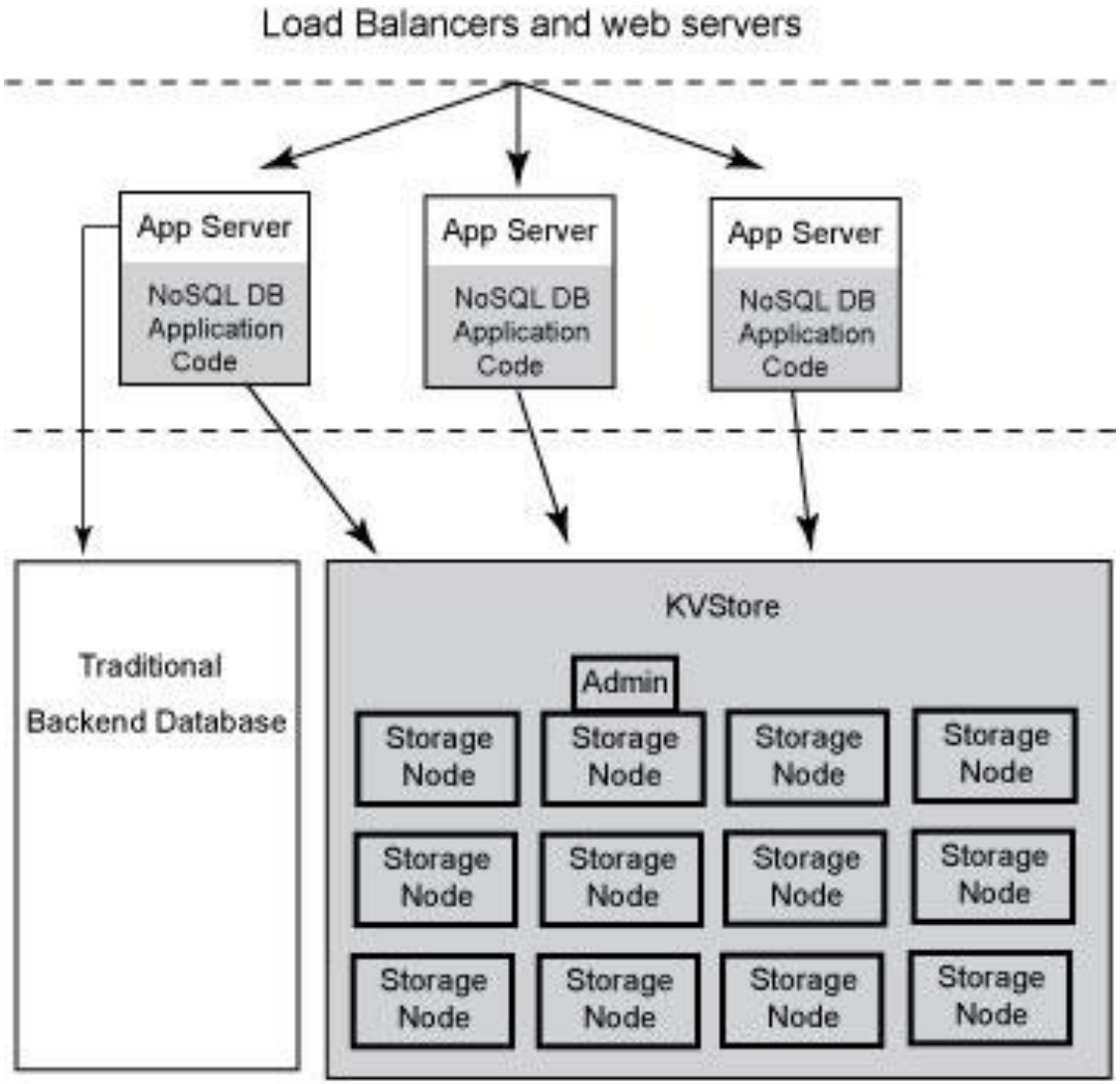
When to use NoSQL

- Low-value, Low-density, Simple data
- Very Simple relationships
- Avoids Joins
- Distributed storage and processing
- Schema-free
- Application Centric
- No logging
- Limited security

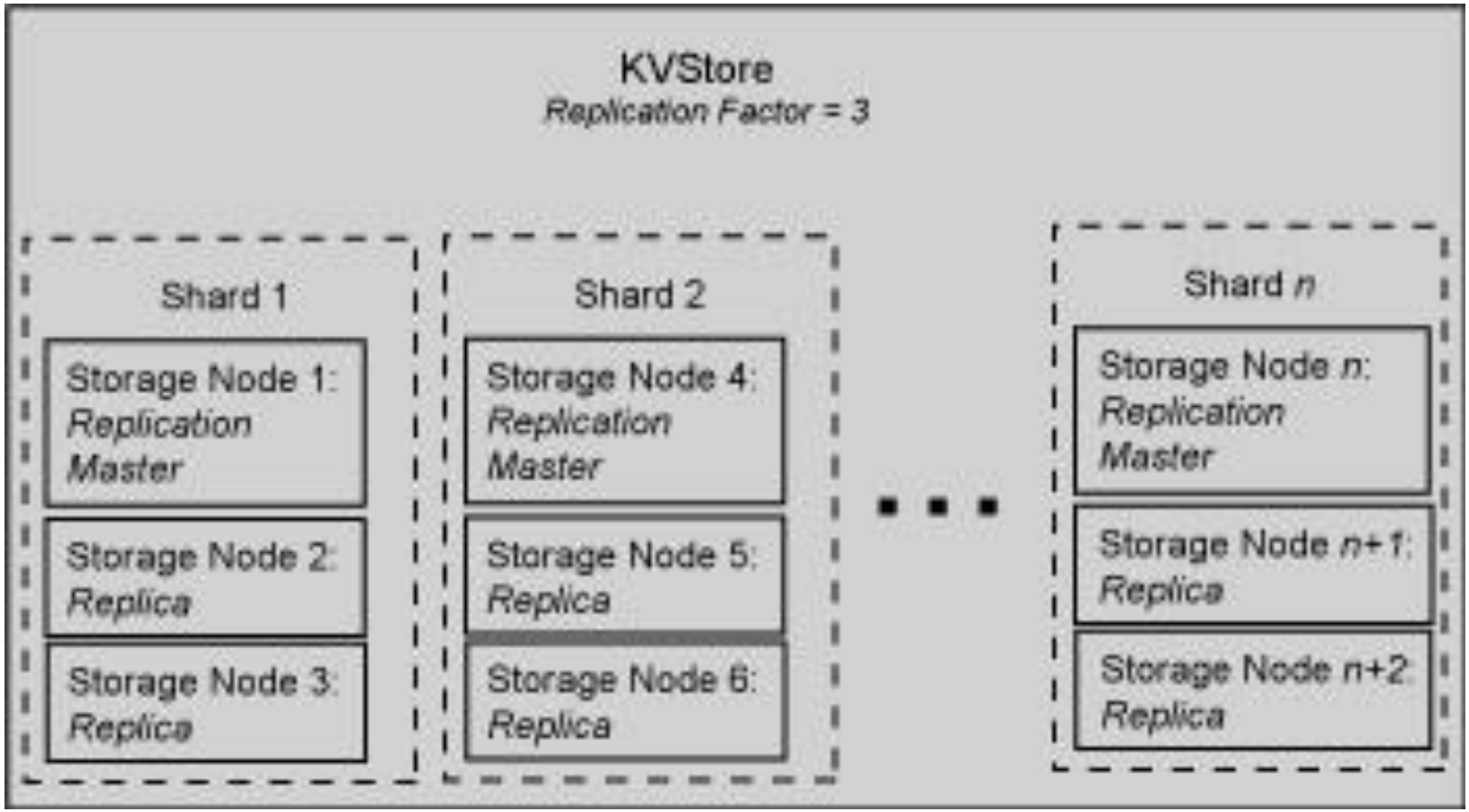
Oracle NoSQL

- A Key-Value database
- Written in Java
- Accessible using Java APIs
- Built on Berkeley DB
- The Oracle solution to acquiring Big Data

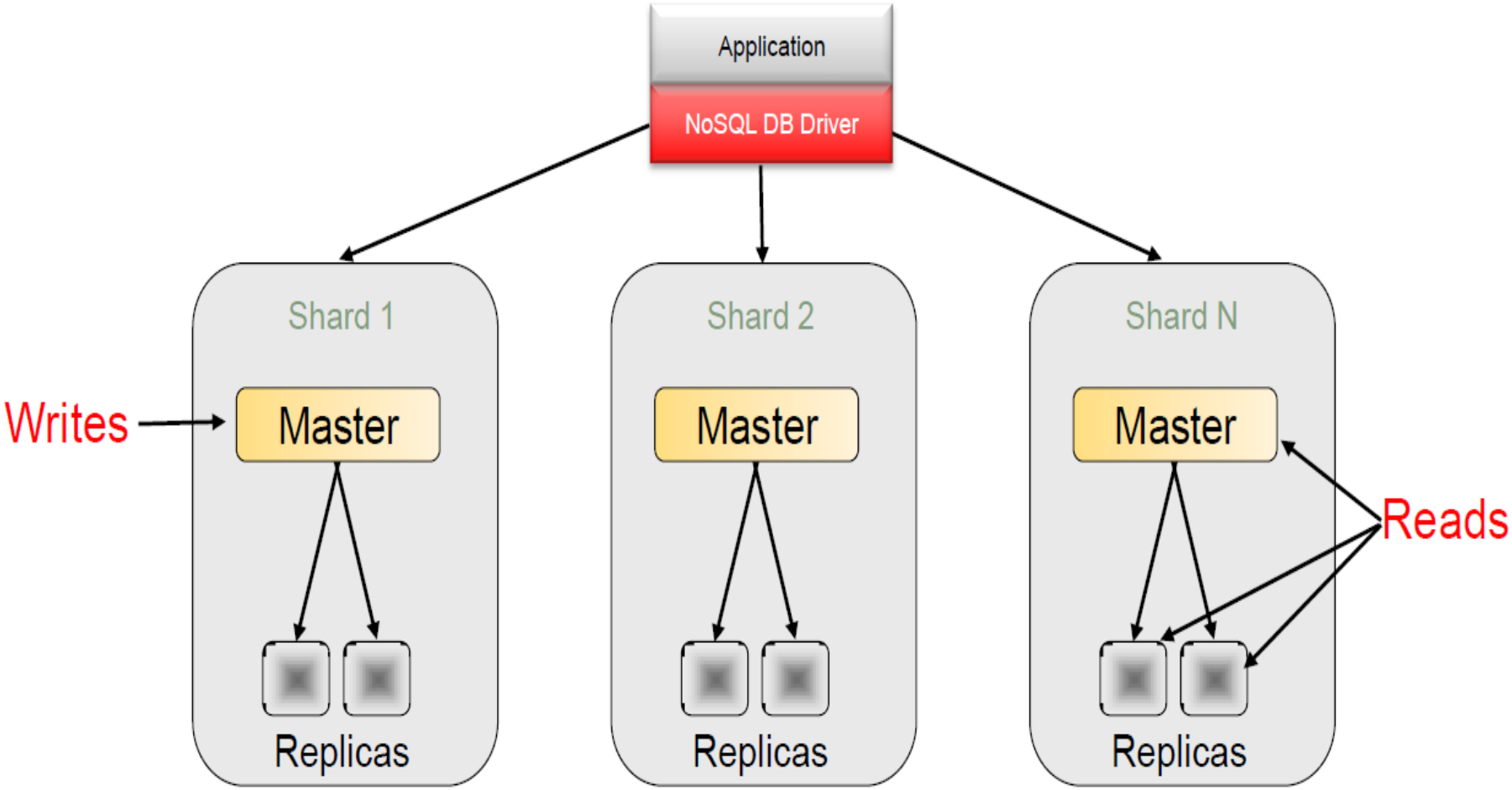
What is the architecture ?



Replication (shards)



Logical Architecture – Application's view

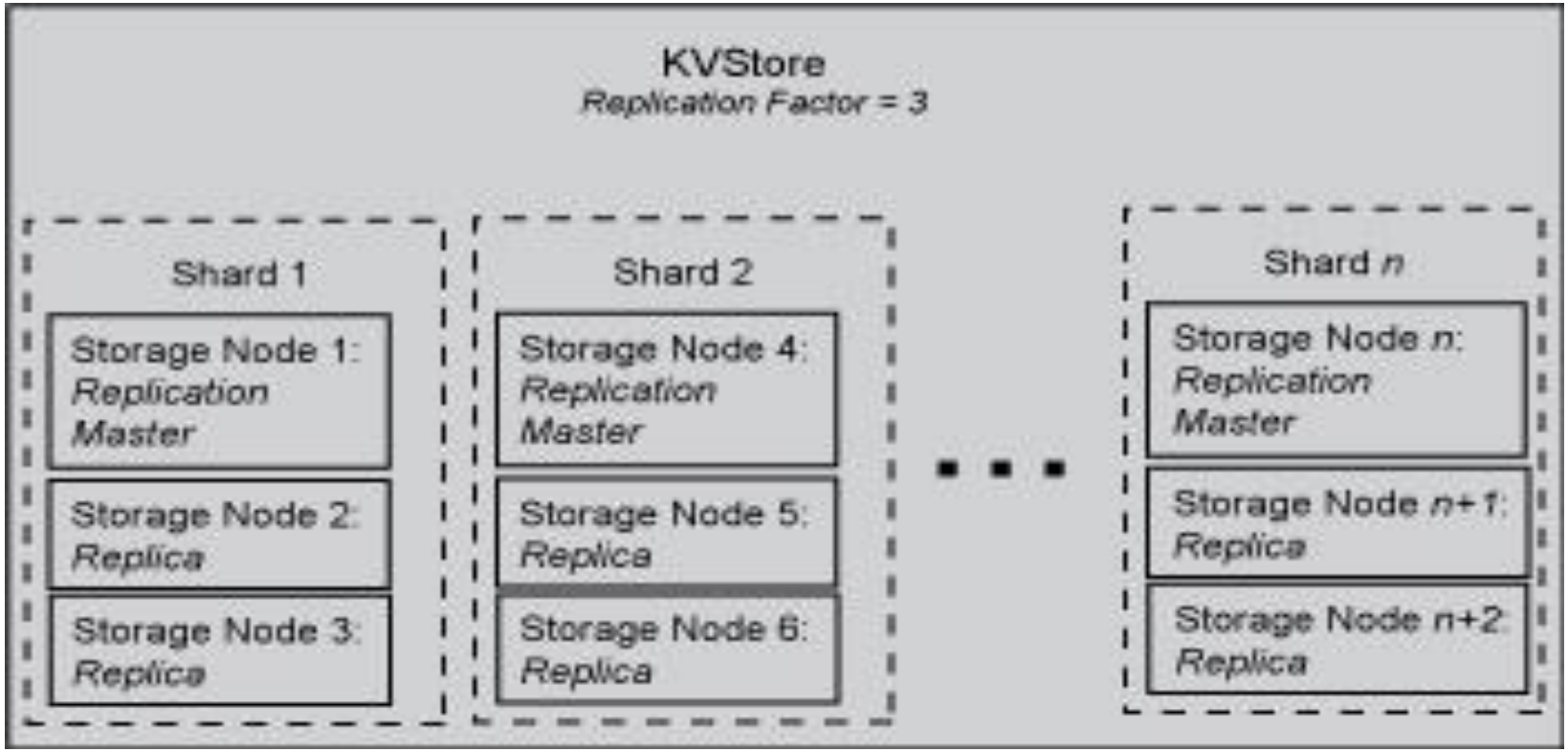


Partitions (hash)

Each shard contains one or more *partitions*. Table rows (or key-value pairs) in the store are accessed by the data's key. Keys, in turn, are assigned to a partition. Once a key is placed in a partition, it cannot be moved to a different partition. Oracle NoSQL Database spreads records evenly across all available partitions by hashing each record's key.

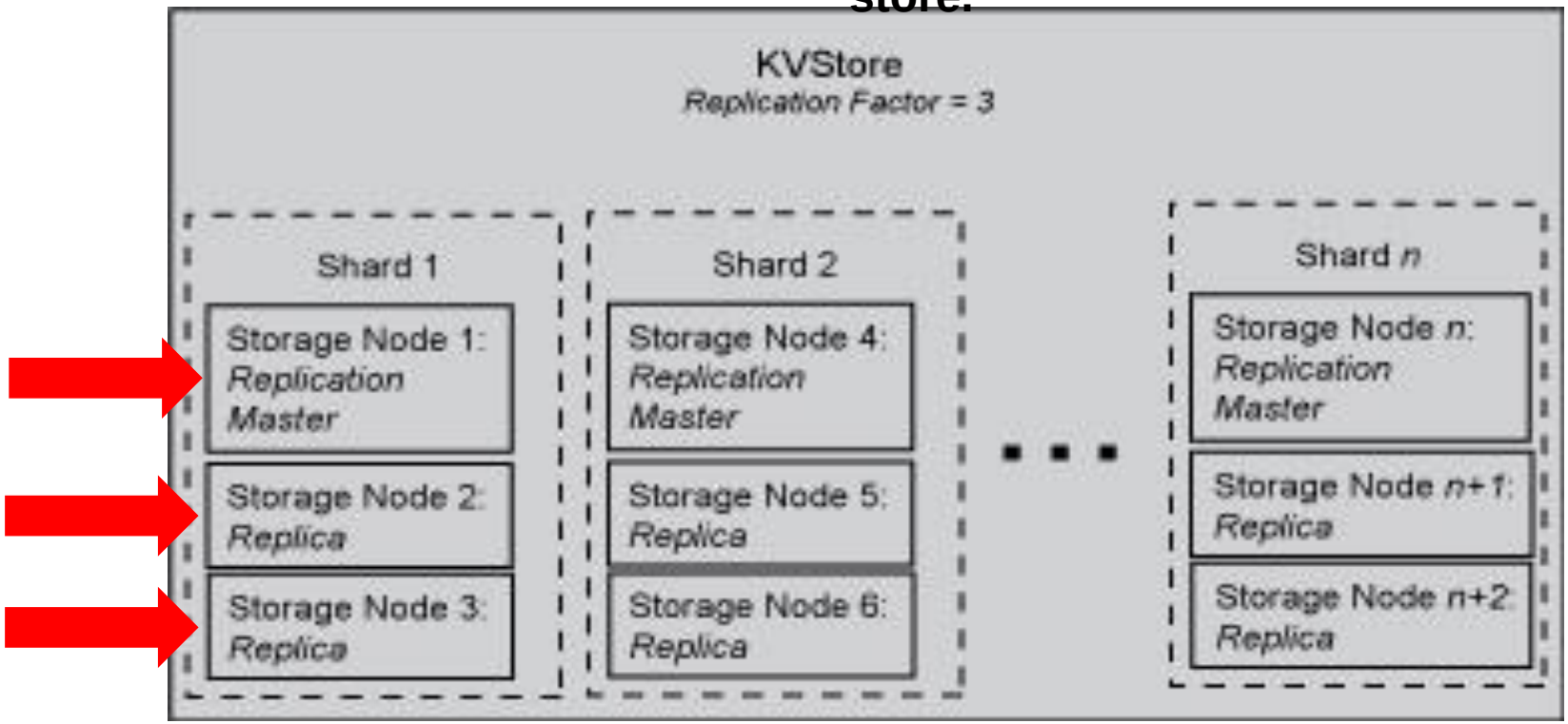
Durability policy

When setting an acknowledgment-based durability policy, you can require acknowledgment from different levels:



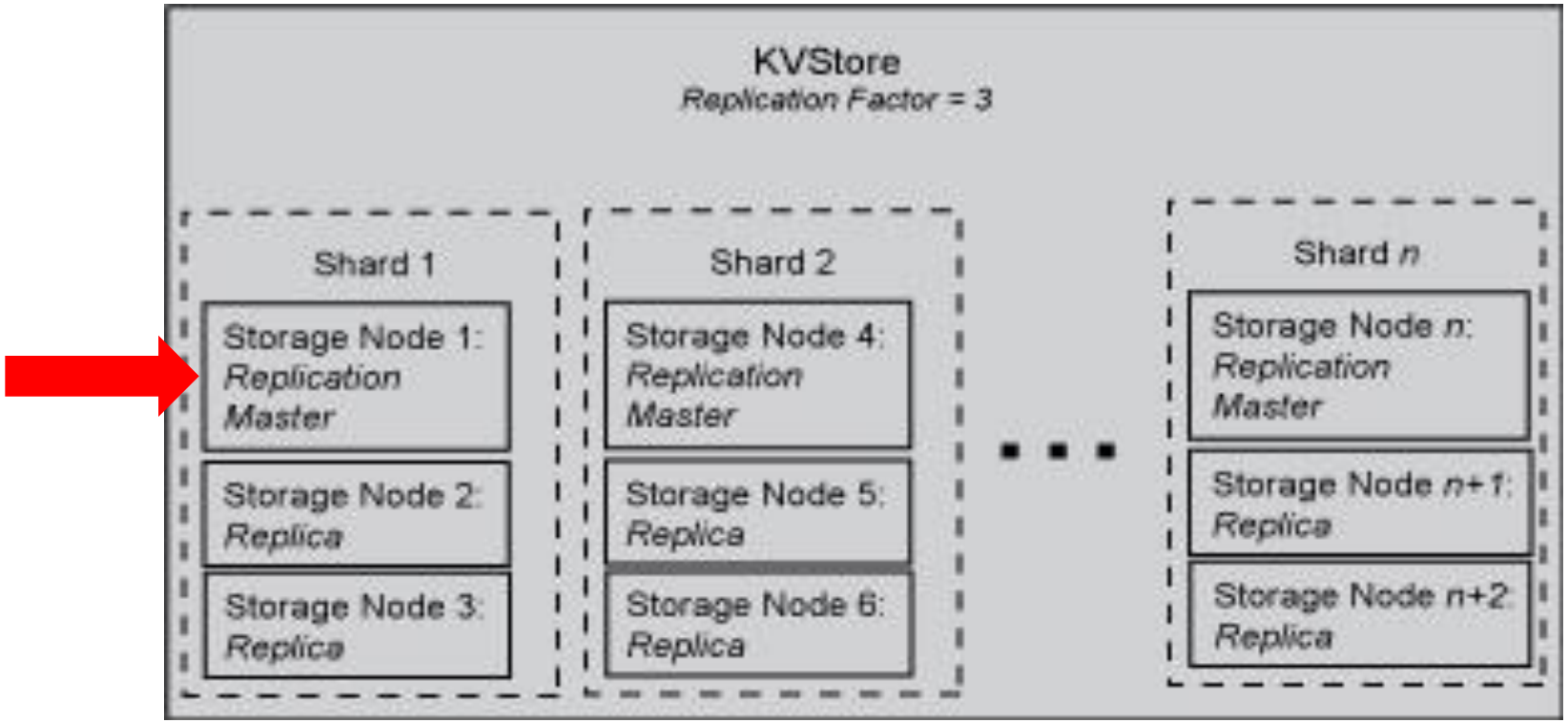
Durability policy (Cont)

- **All replicas. That is, all of the replica nodes in the shard that reside in a primary zone. Remember that your store has more than one shard, so the master node is not waiting for acknowledgments from every machine in the store.**



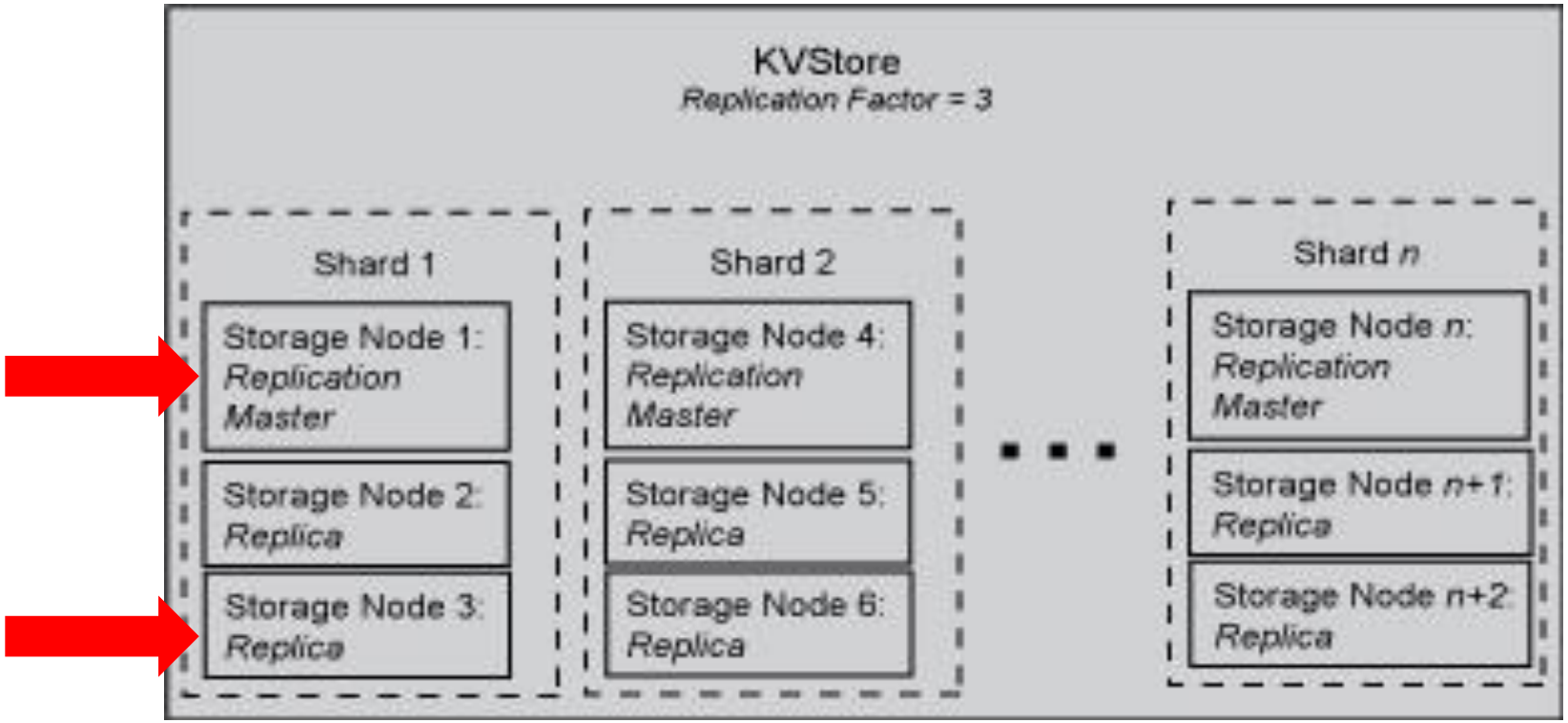
Durability policy (cont)

- **No replicas. In this case, the master returns with normal status from the write operation as soon as it has met its synchronization-based durability policy.**



Durability policy (cont)

- **A simple majority of replicas in primary zones. That is, if the shard has 5 replica nodes residing in primary zones, then the master will wait for acknowledgments from 3 nodes.**



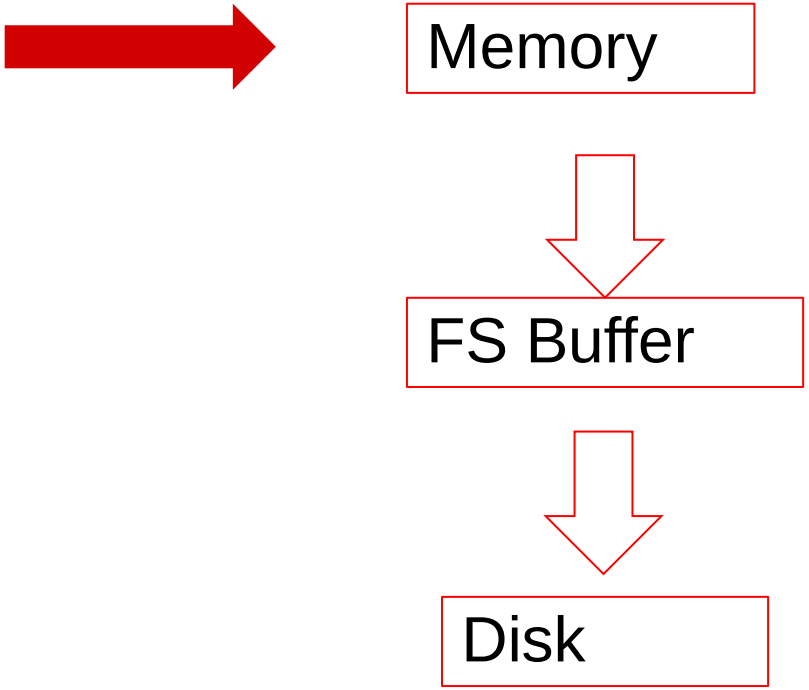
Synchronization-Based Durability

You can control how much of this process the master node will wait to complete before it returns from the write operation with a normal status. There are three different levels of synchronization durability that you can require:

Synchronization-Based Durability (cont)

NO_SYNC

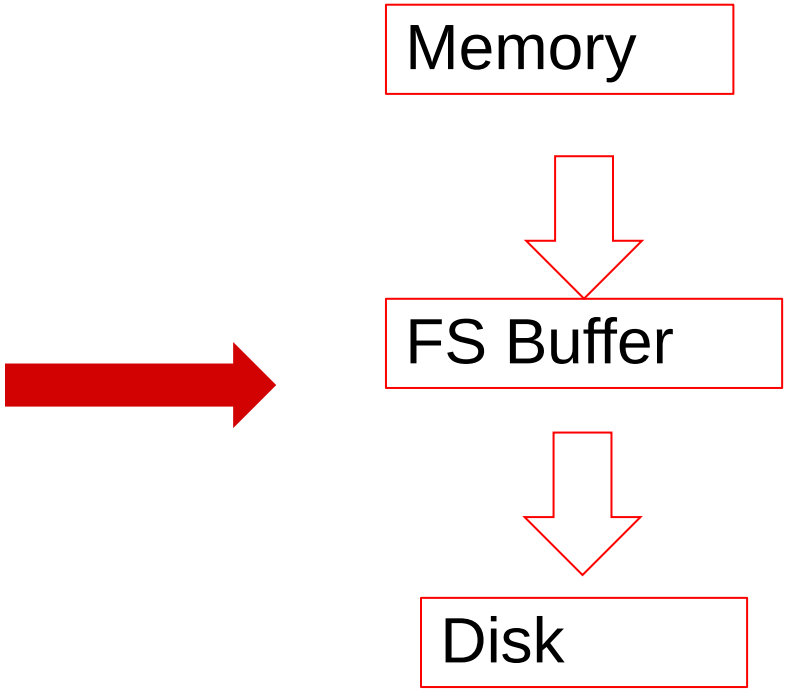
The data is written to the host's in-memory cache, but the master node does not wait for the data to be written to the file system's data buffers, or for the data to be physically transferred to stable storage. This is the fastest, but least durable, synchronization policy.



Synchronization-Based Durability (Cont)

WRITE_NO_SYNC

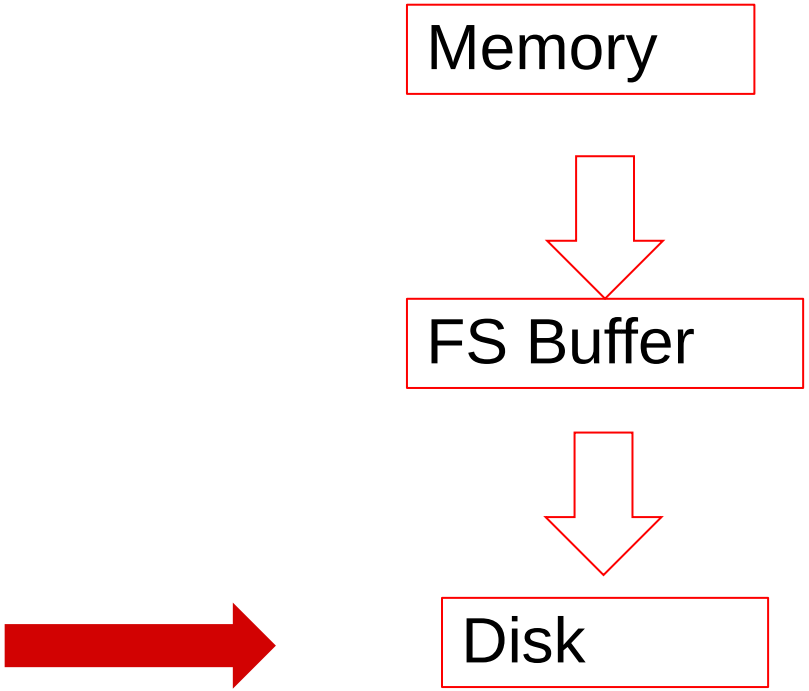
The data is written to the in-memory cache, and then written to the file system's data buffers, but the data is not necessarily transferred to stable storage before the operation completes normally.



Synchronization-Based Durability (cont)

SYNC

The data is written to the in-memory cache, then transferred to the file system's data buffers, and then synchronized to stable storage before the write operation completes normally. This is the slowest, but most durable, synchronization policy.



the fastest possible write performance can be achieved through a durability policy that requires:

No acknowledgments.

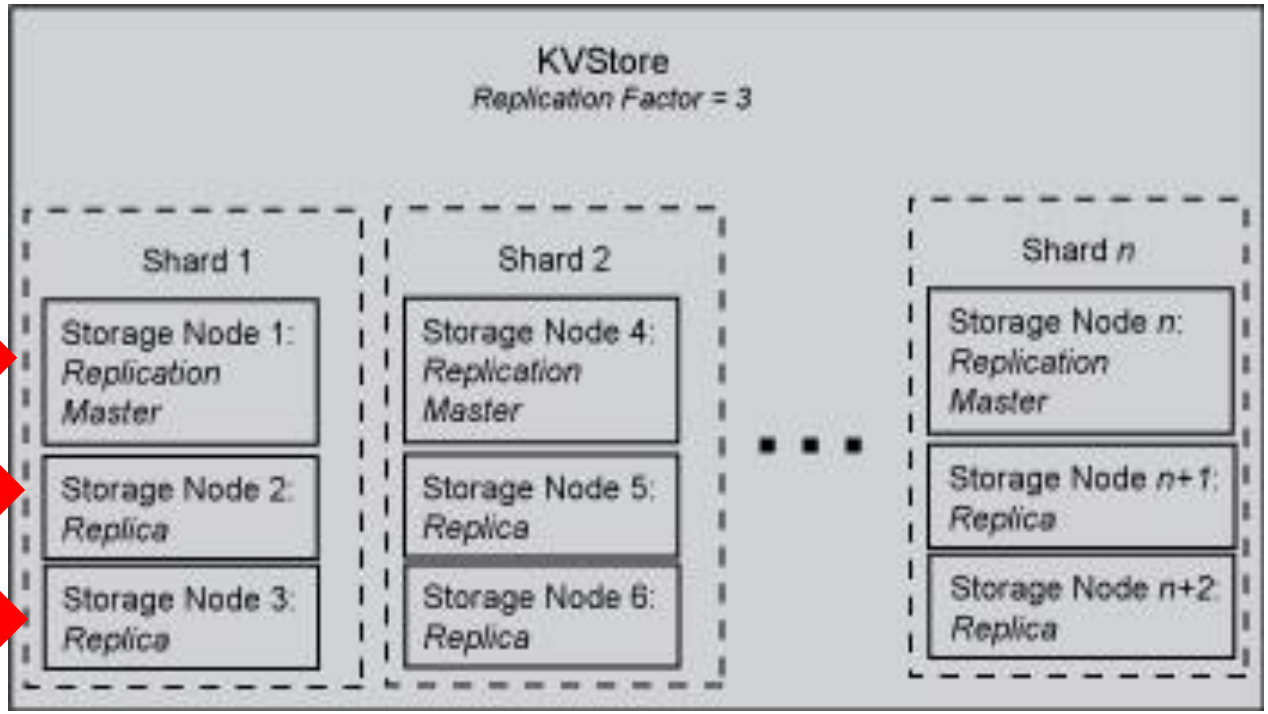
NO_SYNC at the master.

NO_SYNC at the replicas.

NO_SYNC

NO_SYNC

NO_SYNC



Highest possible durability guarantee, you can use:

All replicas must acknowledge the write operation.

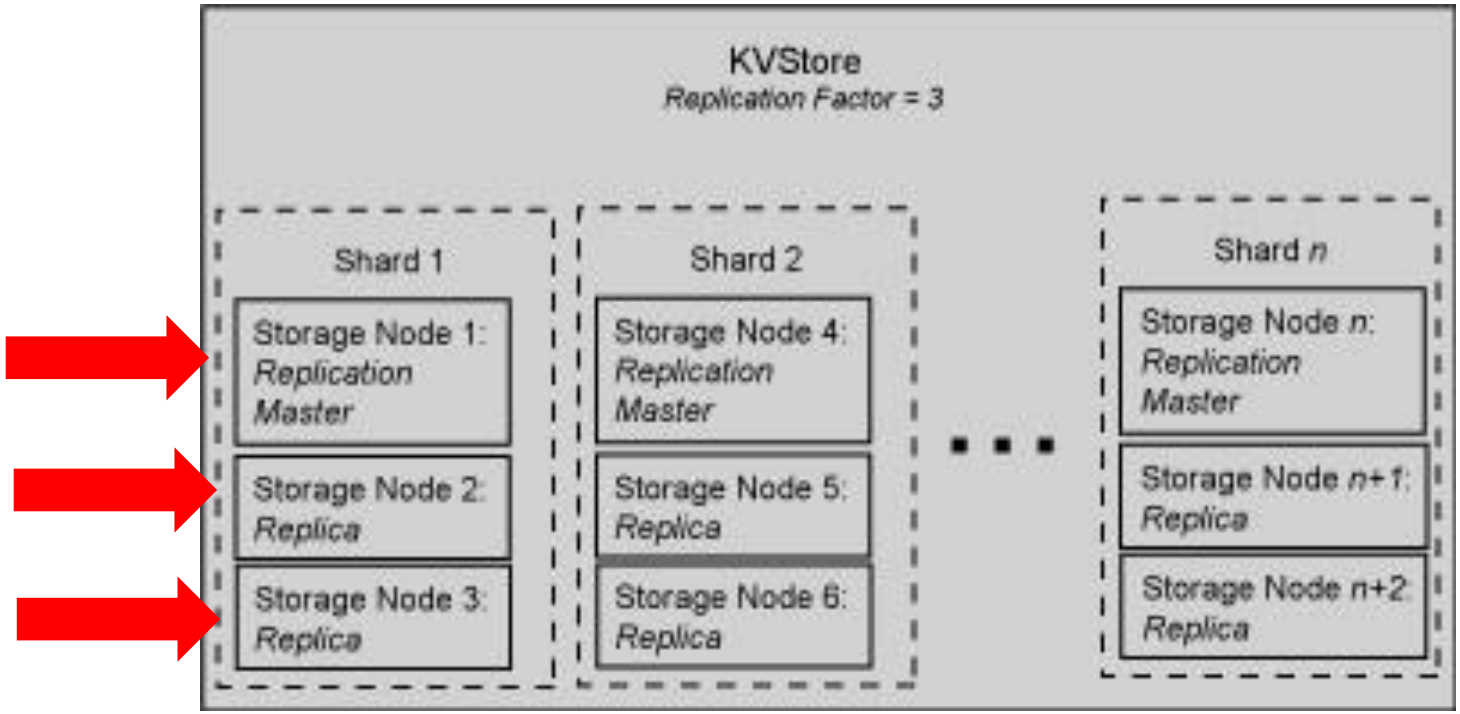
SYNC at the master.

SYNC at the replicas.

SYNC

SYNC

SYNC



Most commonly, durability policies attempt to strike a balance between write performance and data durability guarantees. For example:

Simple majority of replicas must acknowledge the write.

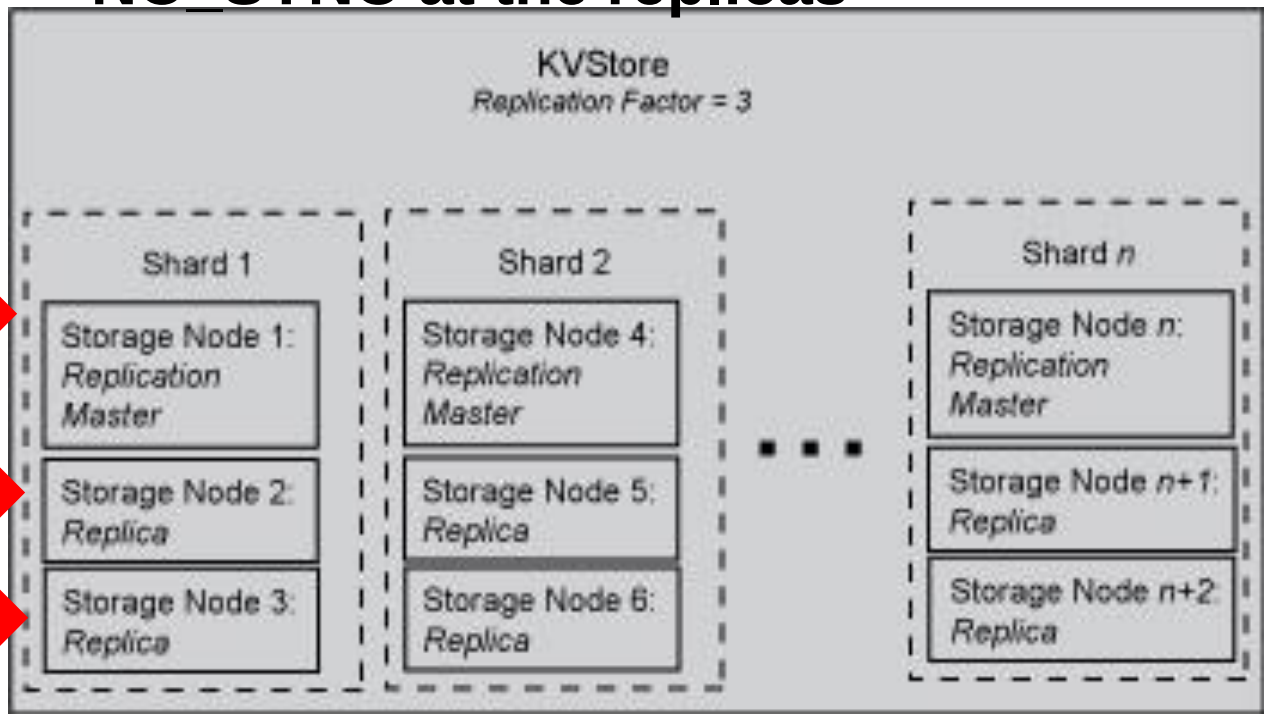
SYNC at the master.

NO_SYNC at the replicas

SYNC

NO_SYNC

NO_SYNC



Read consistency

None - Read anywhere regardless of the state of the master and replicats

Time based – Read anywhere, but throw an exception if a replicat lags longer than a set time.

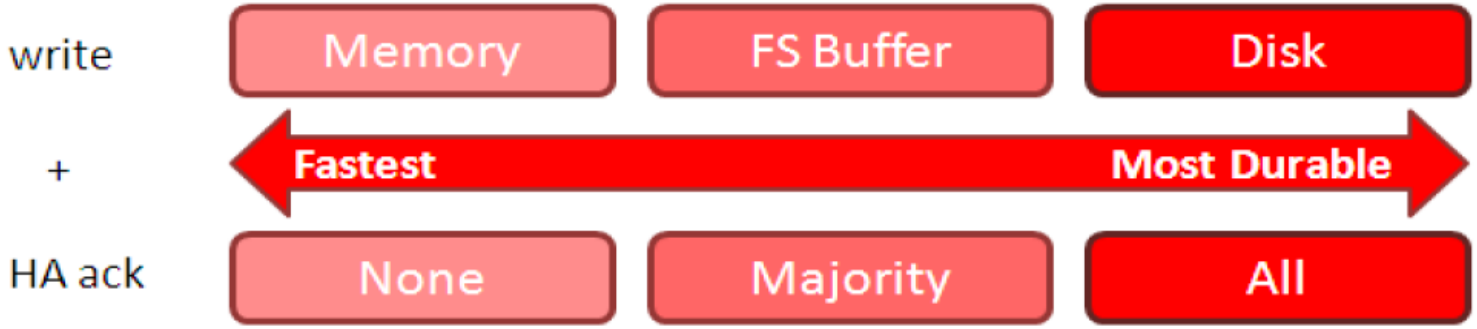
Version-based – Versions information can be kept and transferred. Data is fetched if it matches a specified version.

Absolute - Only go after master to ensure data is current

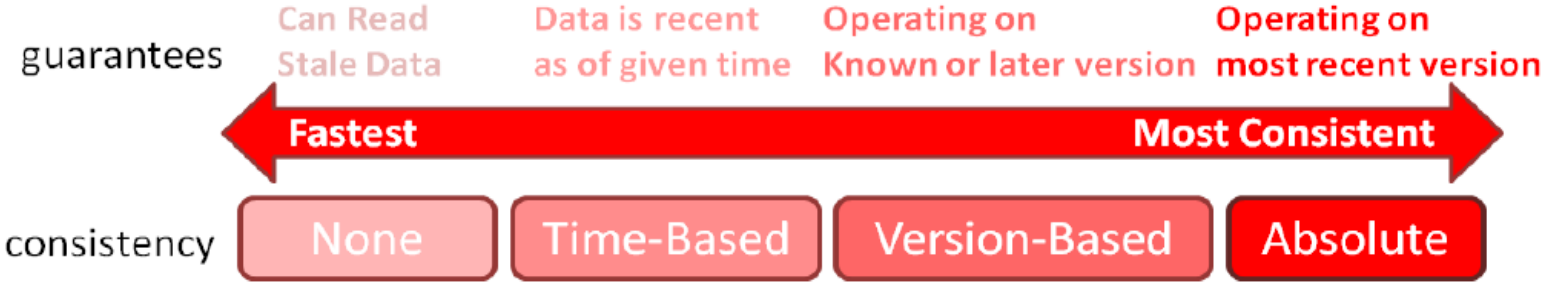
Transaction Durability and Read consistency

ACID Transactions – Configurability

- Configurable **D**urability Policy



- Configurable **C**onsistency Policy



Security

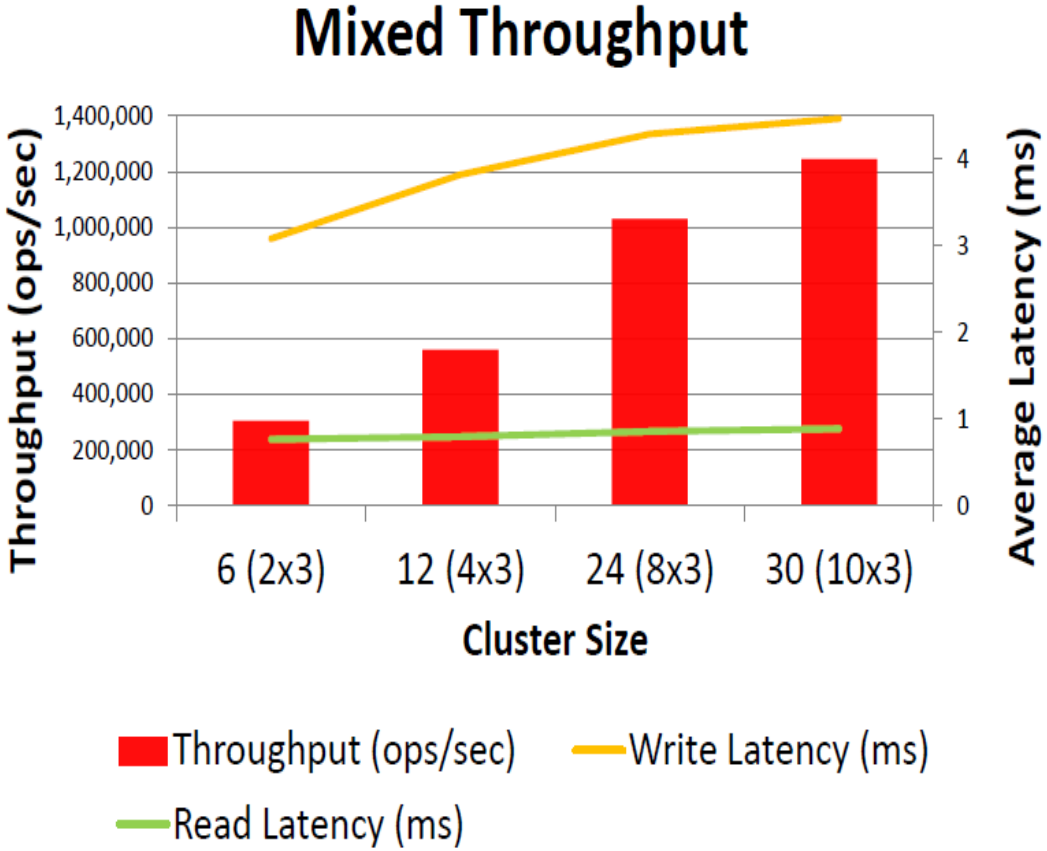
Oracle NoSQL Database is intended to be installed in a secure location where physical and network access to the store is restricted to trusted users. For this reason, at this time Oracle NoSQL Database's security model is designed to prevent accidental access to the data. It is *not* designed to prevent malicious access or denial-of-service attacks.

Security

- ID/Password
- Wallet support
- Encryption support
- SSL Certificate support
- Invalid logon attempt lock to prevent Denial of Service attacks
- No Individual object, row level, auditing etc. etc.

Latest YCSB Benchmark Results

- 1.25M ops/sec
- 2 billion records
- 2 TB of data
- 95% read, 5% update
- Low latency
- High Scalability



Oracle database benchmark

Oracle produced a world record SPECjEnterprise2010 benchmark result of

57,422.17 SPECjEnterprise2010 EjOPS

using Oracle's SPARC T5-8 server in the application tier and another SPARC T5-8 server for the database tier.

Items of interest

- Can be Highly distributed
- Commodity hardware and local storage
- Low value data
- Some level of replication but usually is not completely synchronous
- Writes are to the Master only (then replicated)
- Reads can be controlled and different consistency levels can be used

What is a KV Store ?

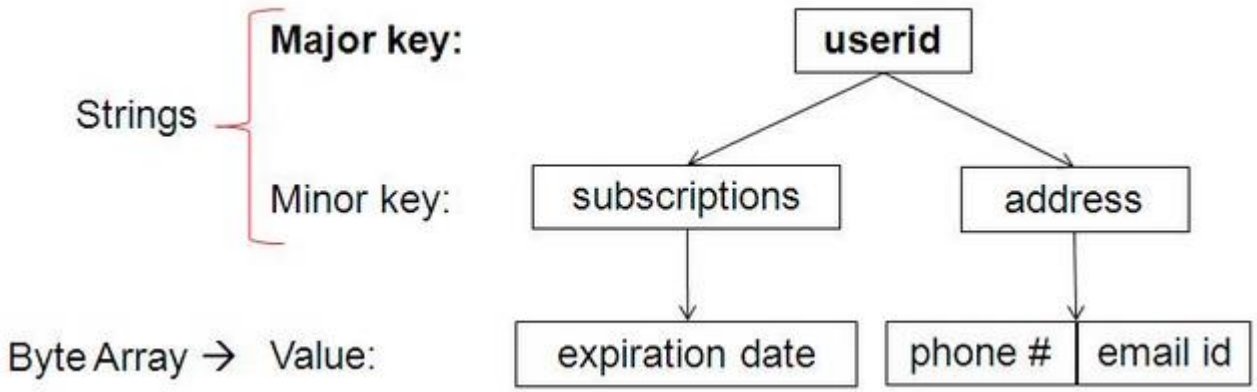
So what is a Key value store ?

Simply a record is made up of 2 pieces.

Key - May be multiple Layers

Value – Value that the key points to. This value type is user defined.

Major keys/Minor Keys



A key is the concatenation of a Major Key Path and a Minor Key Path, both of which are specified by the application. All records sharing a Major Key Path are co-located to achieve data locality.

Customer database Key Value Example

Store information about a user “Bob Smith”

Surname -- Smith

Familiar Name – Bob

Values

- Contact (phone number, address. Etc)
- Image file
- Voice Greeting
- Public key (for encryption)

Contact information

***/surname/familiar name/-/*contact**

The value for this key is a user defined record that contains all of the small user properties (name, phone number, address, and so forth).

Ex. /Smith/Bob/-/contact

Public key information

***/surname/familiar name/-/*publickeys**

The value for this key is an user defined record that contains the user's public keys.

Ex. /Smith/Bob/-/publickeys

Image File

/image.lob/-/*surname/familiar name*

The value for this key is an image file, saved using Oracle NoSQL Database's large object support.

Ex. /image.lob/-/smith/bob

Voice Greeting

/audio.lob/-/voicegreeting/*surname/familiar name*

The value for this key is an mp3 file, also saved using the large object support.

Ex. /audio.lob/-/smith/bob

Example Key Value information (summary)

- /*surname*/*familiar name*/-/contact
- /*surname*/*familiar name*/-/publickeys
- /image.lob/-/*surname/familiar name*
- /audio.lob/-/voicegreeting/*surname/familiar name*

Craigs list

- Repository of 2.5+ billion archived postings
- growing and growing and growing
- 3 shards across 3 node replica sets
- duplicate config in 2nd data center
- ~6TB of data, sized up to 12TB

Ex. */Rochester*/itemsforsale/-/carsandtrucks/byowner

Amazon

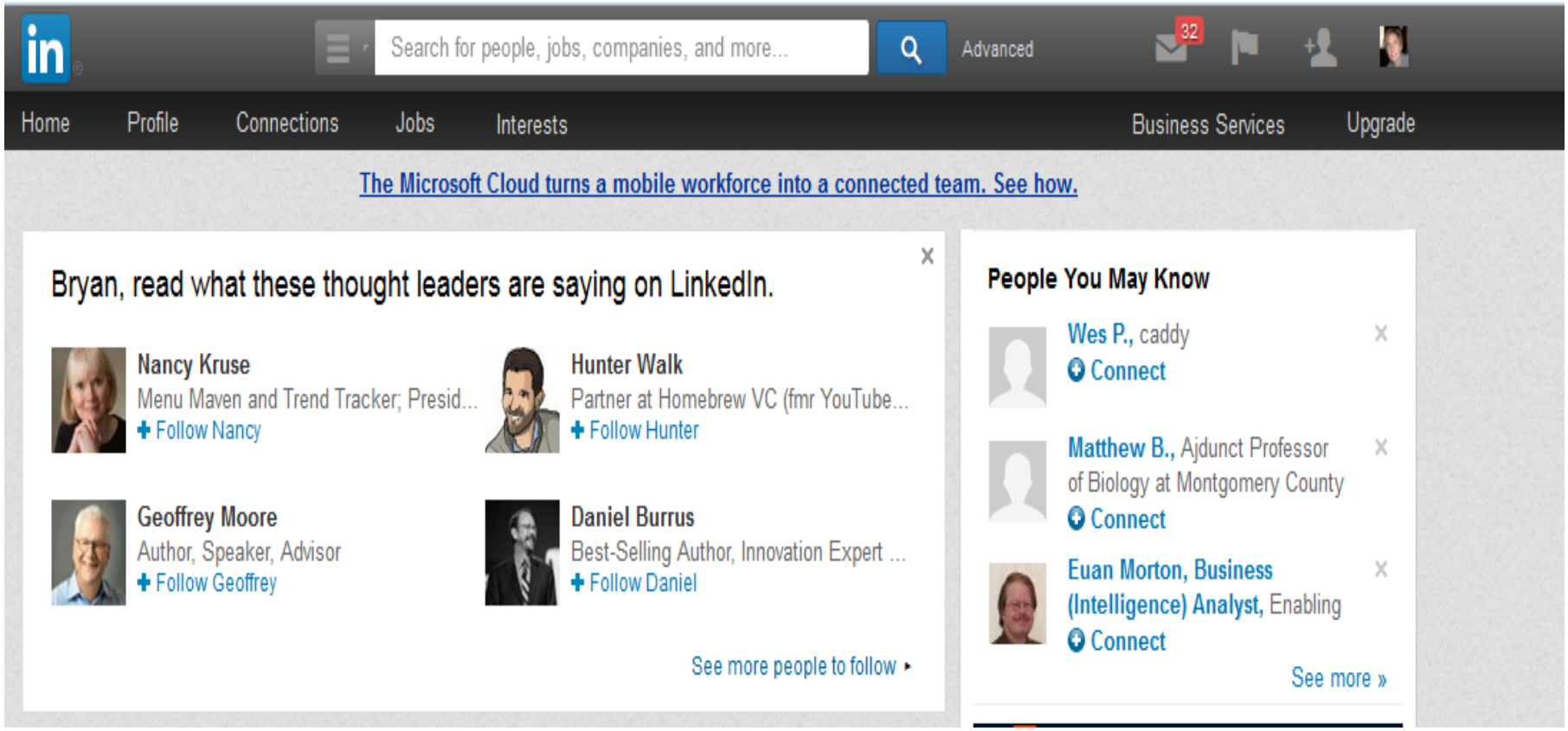
- Shopping cart
- 3 million checkouts in a single day (2007)
- Contains ACID (Atomicity, Consistency, Isolation, Durability) Properties but these can be controlled
- Weaker “C” allows for higher availability

- Groundbreaking paper in 2007

http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

and NoSQL

Hadoop and NoSQL are complementary technologies



Other Example of uses for KVM

- User information (like this example)
- Email Application: This sample application uses table based data modeling concepts
  to create a fully functioning email client/server application.
- **Kvitter Sample Application -** Kvitter is a Twitter-like microblog sample application. A user is uniquely identified by his/her user name. A user signs in using his/her password. A blog is uniquely identified by the blog Id. A blog is created by a user. A user can follow other users.
- Inventory data

Reading records

- Retrieve a single record
- Multiple records with same major key
- There is the concept of Parent child relations
  - This is useful for Purchase history type information
  - Line items on an order

Get multiple records

Craigs list /Buffalo/forsale/-/cars+trucks/

Amazon /loginid/-/shoppingcart/items

- There can be secondary indexes

Again Why NoSQL, Why a KV ?

- Simpler data model that is more flexible
- Can be split across multiple nodes for availability
- Faster than an RDBMS
- Durability and consistency can be controlled as needed

Other uses for a KV Store

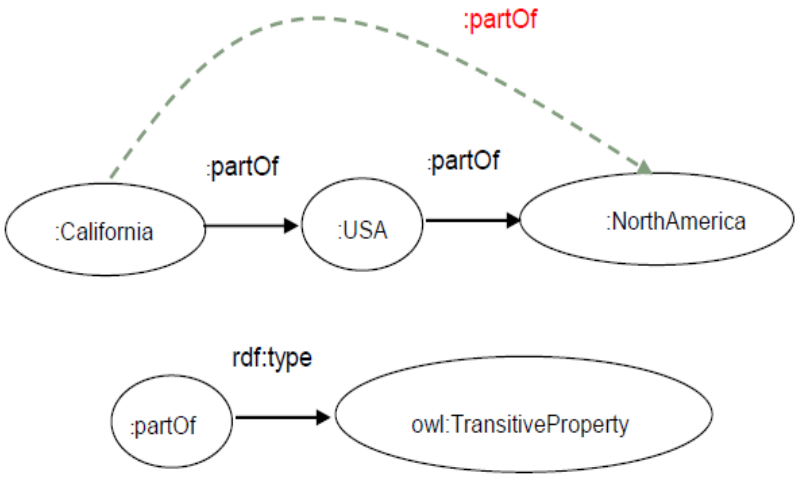
- RDF (resource description framework), Graphing Database
- Semantic store

Relationship information (Metadata) on data will become even more important.

Why a Graph Database?

Graph Database

- Model data in terms of relationships
- Flexible schema evolves easily by adding new relationships
- Supports querying and discovery by graph patterns and traversal
- Enables graph analytics such as reachability, connectivity, transitivity, same as, proximity, centrality…



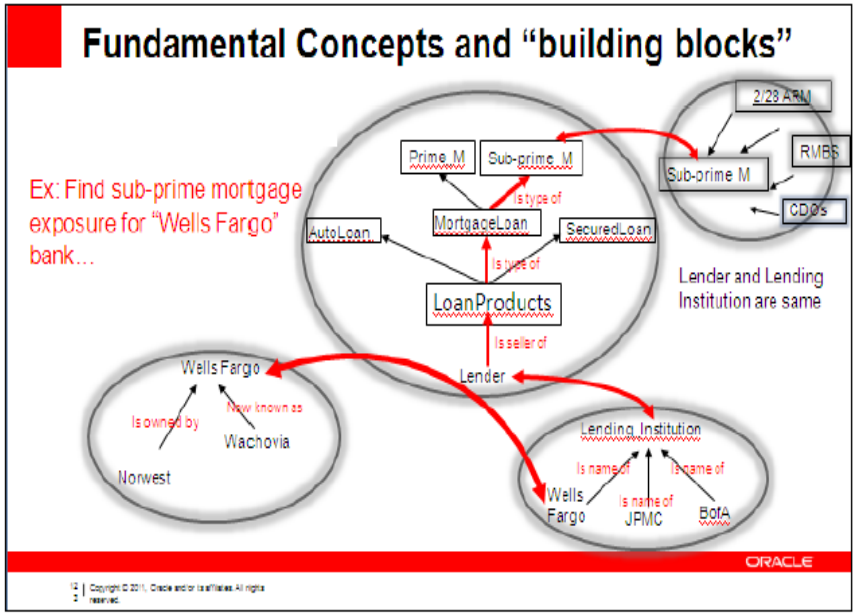

```
Query:  SELECT ?x ?y
        FROM …
        WHERE {  ?x :partOf ?y }
```

RDF Graph For NoSQL Database EE

For horizontal scalability, lower query latency/cost, ease of install & management

RDF Graph Feature for NoSQL

- Simple high volume queries
- Queries aggregating over most of the graph (e.g. what are the hobbies of the 100 most popular people in the network)
- Frequent, large-scale updates
- Open Linked Data applications



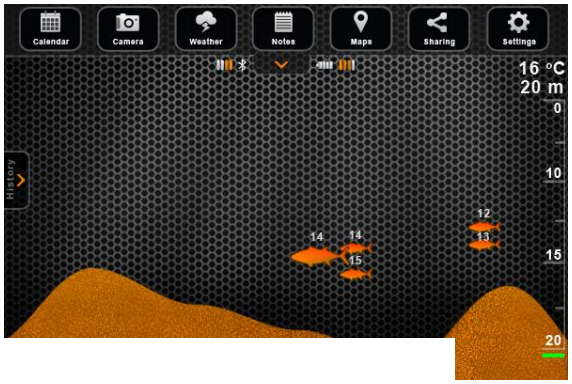
Why am I talking about this ?

For RDBMS DBA’s everything’s nail

For Oracle DBA's (regardless of database type)

- HDFS (cloudera partnering with Oracle)
- NoSQL (oracle NoSQL or Oracle Berkely DB)
- RDBMS (traditional relational)
- Times Ten

Data lake



Hadoop is like radar scanning for data

NoSQL is like GPS, identifying a specific location.

ISAM makes a comeback ?

Resource

- Big data appliance Virtualbox image

http://www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html

- Oracle nosql download
- http://www.oracle.com/technetwork/database/database-technologies/nosqldb/downloads/index.html

Contact me

Email - Bryan.grenn@oracle.com
Twitter - @bryangrenn
Blog http://bryangrenn.blogspot.com