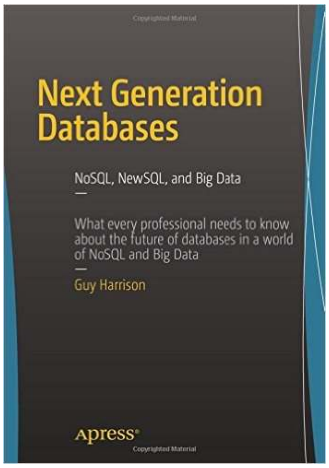
Next Generation Databases and Oracle

Guy Harrison,
CTO dbKoda

dbKoda

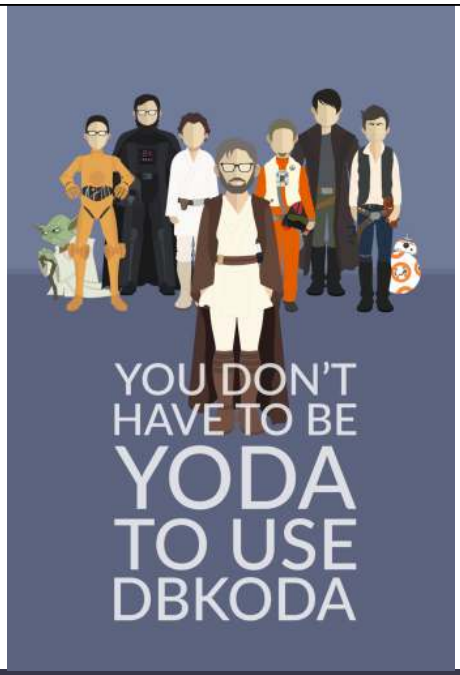
dbKoda

Introductions



Web: guyharrison.net
Email: guy.a.harrison@gmail.com
Twitter: @guyharrison

TOBA CAPITAL

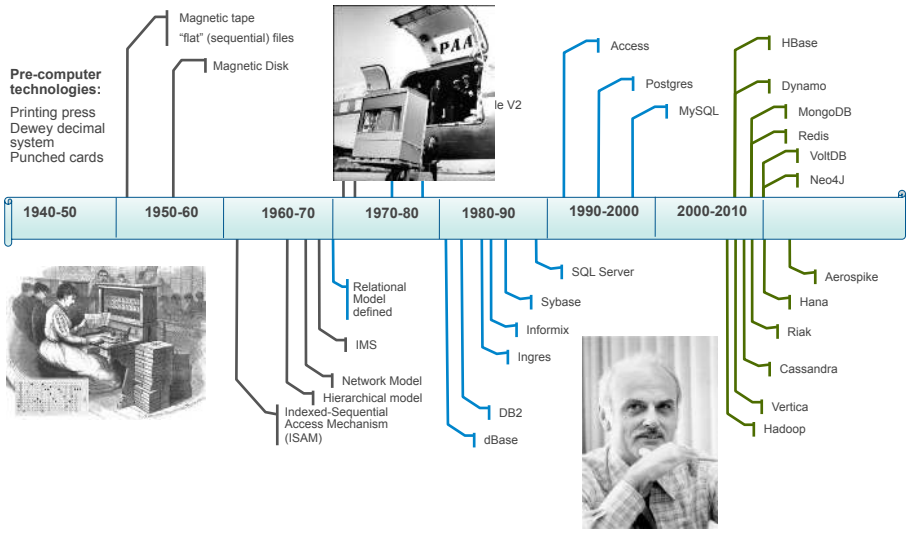


dbKoda

3 Database revolutions

dbKoda

History of databases

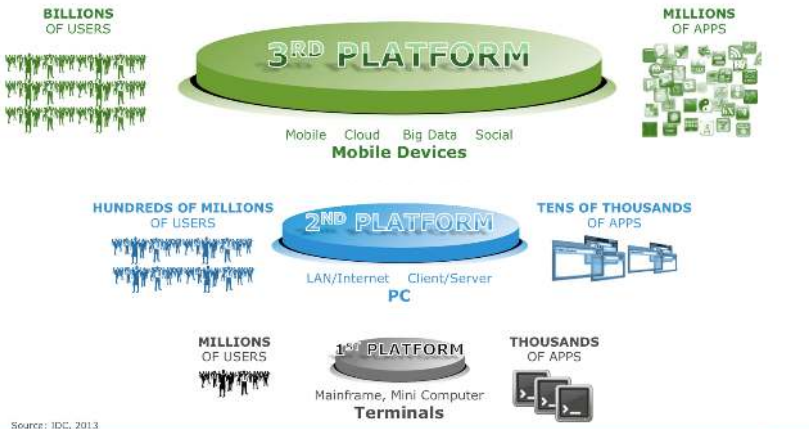


Why?

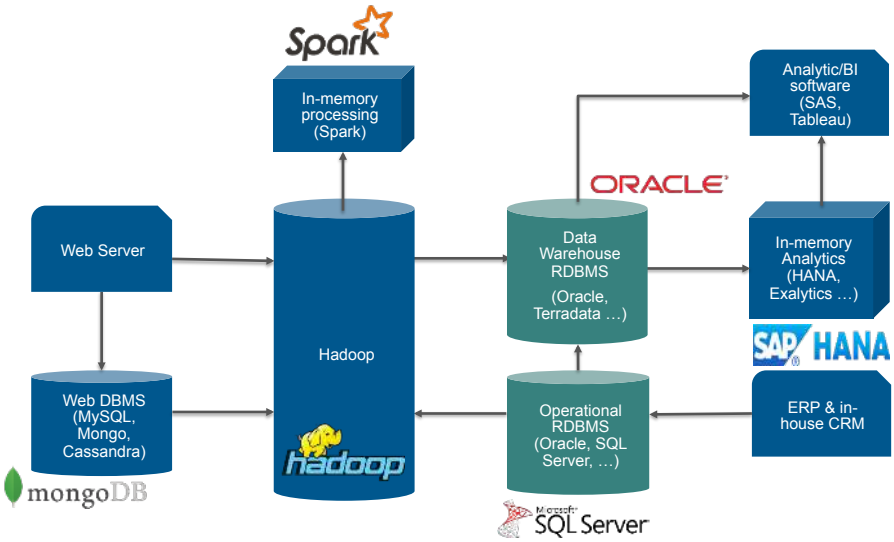
3rd Platform drives new demands on the database:

- Global high availability
- Data volumes
- Unstructured data
- Transaction rates
- Latency

A single architecture cannot meet all those demands.



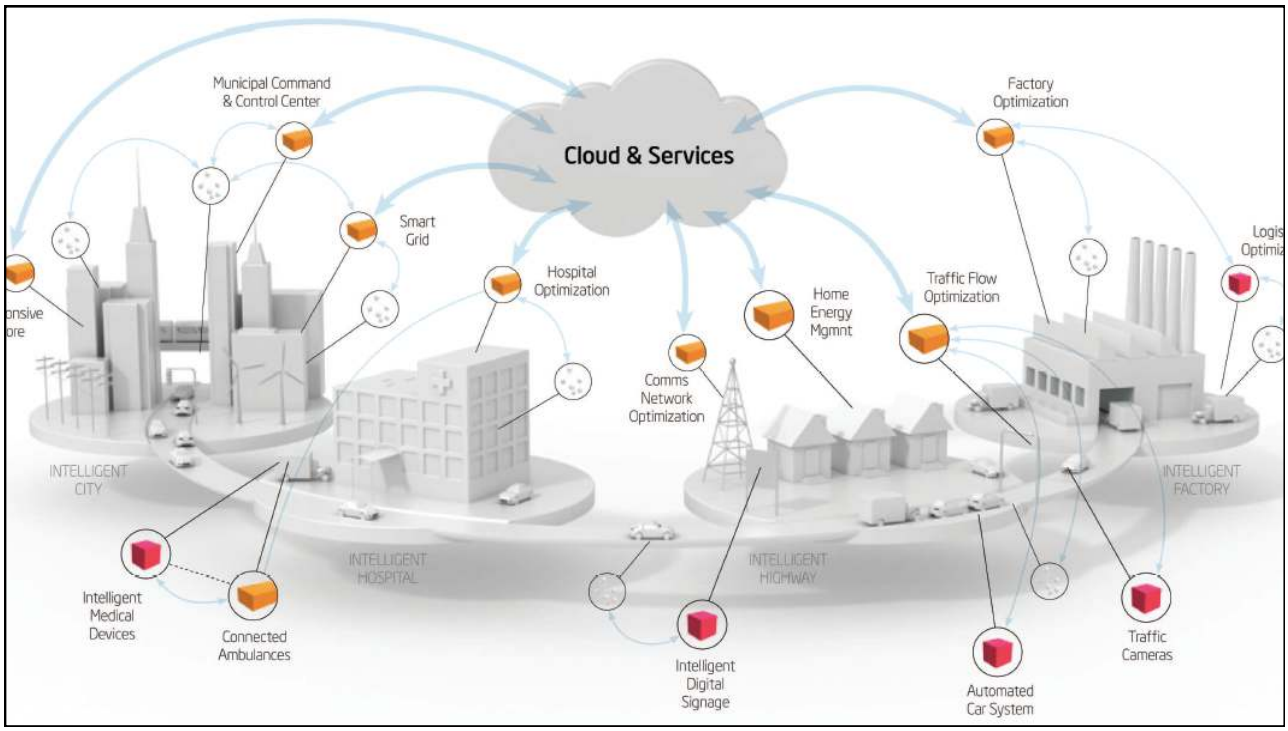
It takes all sorts.



Big Data and Hadoop

dbKoda

The industrial revolution of data



The instrumented human

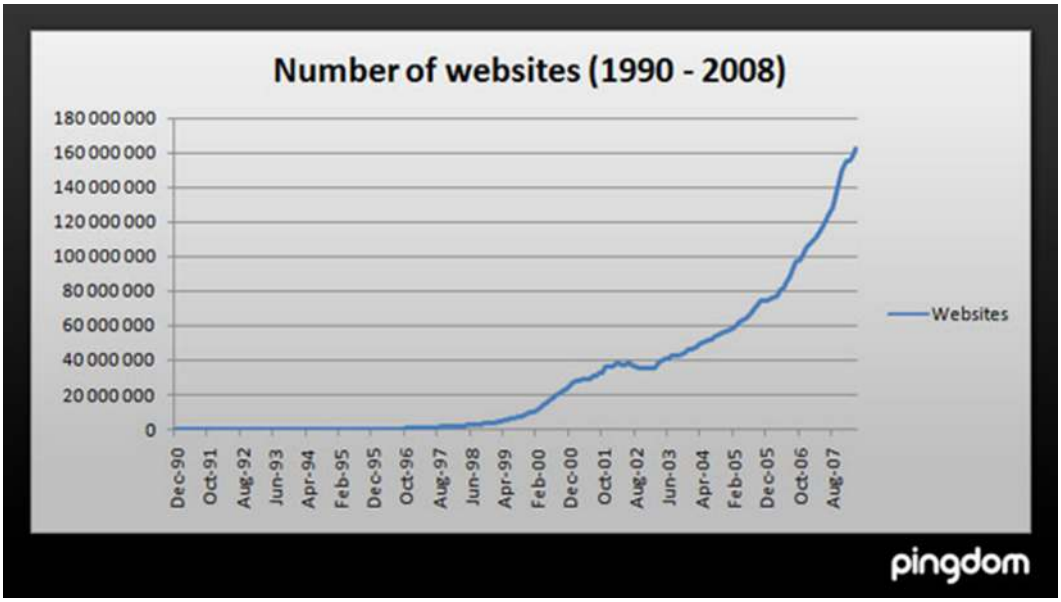
- Compass
- Camera
- Mike/earphones
- Heads up display
- Emotion/Attention monitor

- Bluetooth Personal Area Network
- 3G/WiFi Wide Area Network
- GPS
- Storage

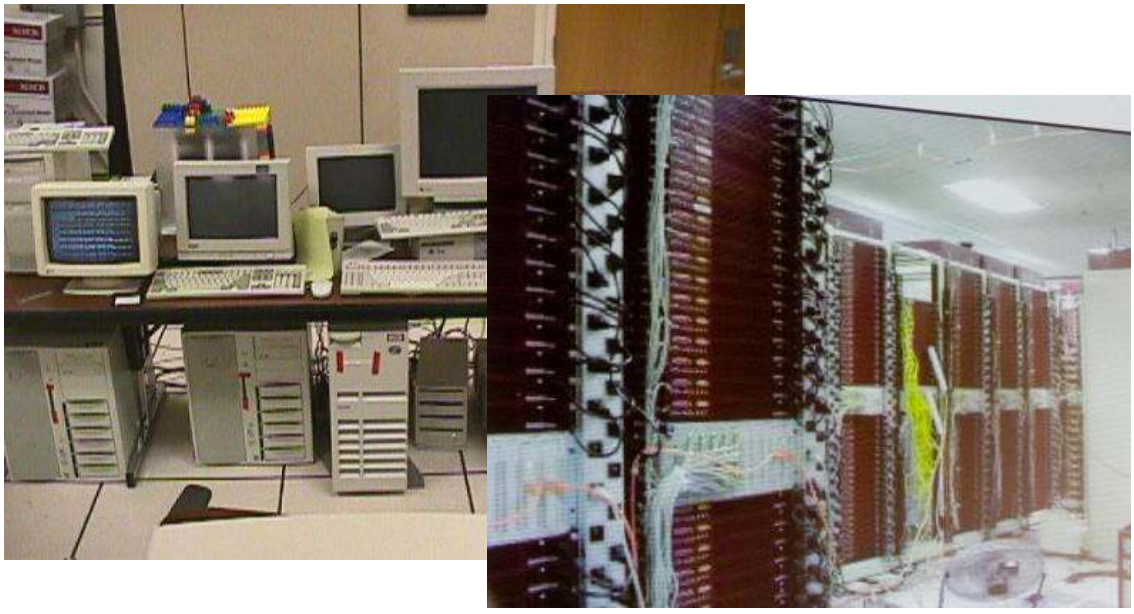
- Pulse, temp monitor
- Silent alarms
- Pedometer, sleep monitoring

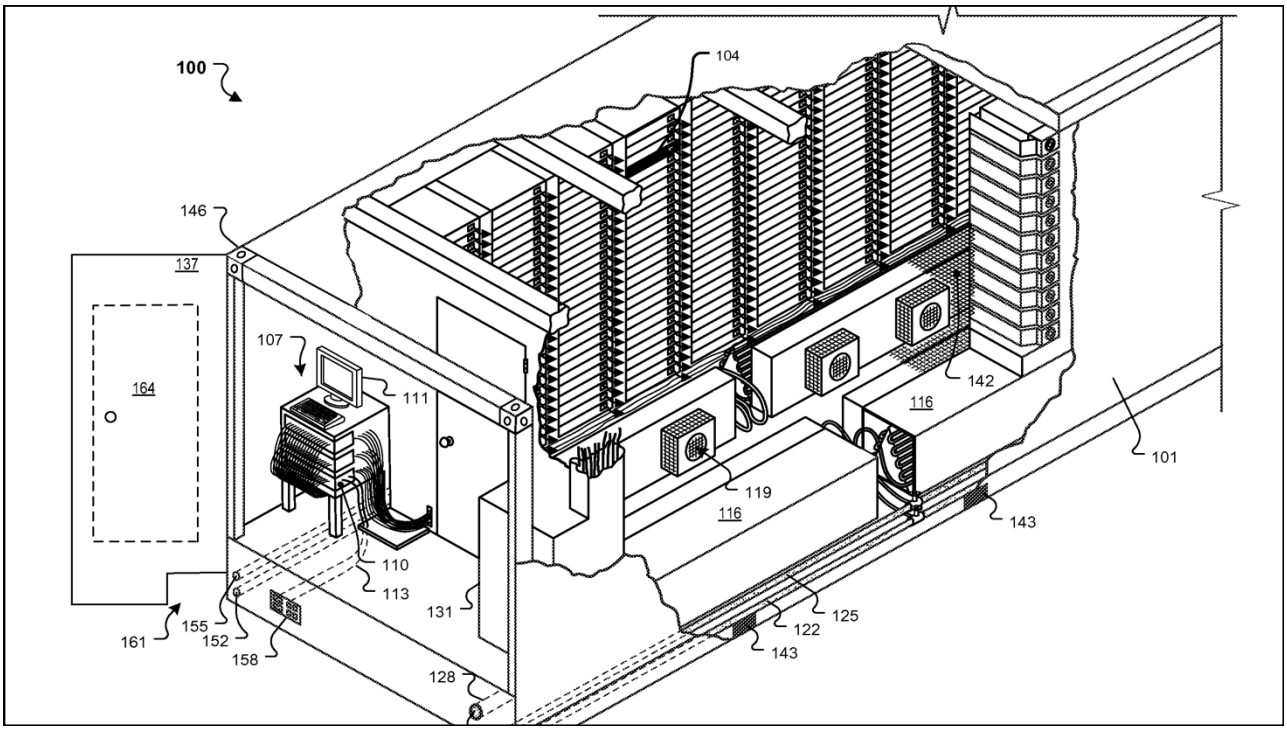
dbKoda

Pioneers of big data

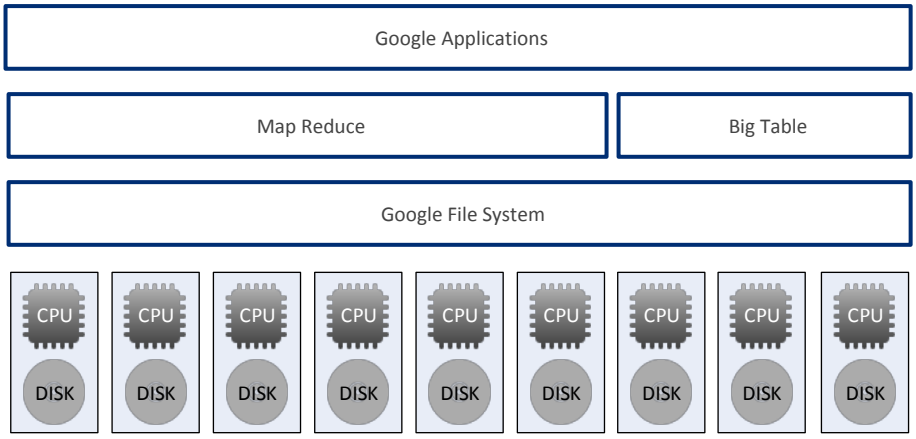


dbKoda




100
104
146
137
107
111
164
142
116
101
119
116
143
110
113
131
125
122
143
155
152
161
158
128

Google Software Architecture (circa 2005)



Hadoop: 1.0: Open Source Map-Reduce Stack

Hadoop at Yahoo

2010 (biggest cluster)

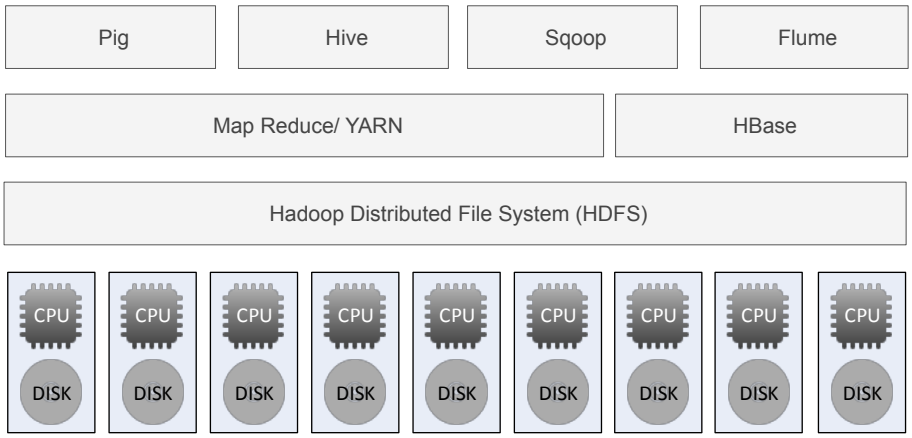
- 4000 nodes 16PB disk
- 64 TB of RAM
- 32,000 Cores

2014:

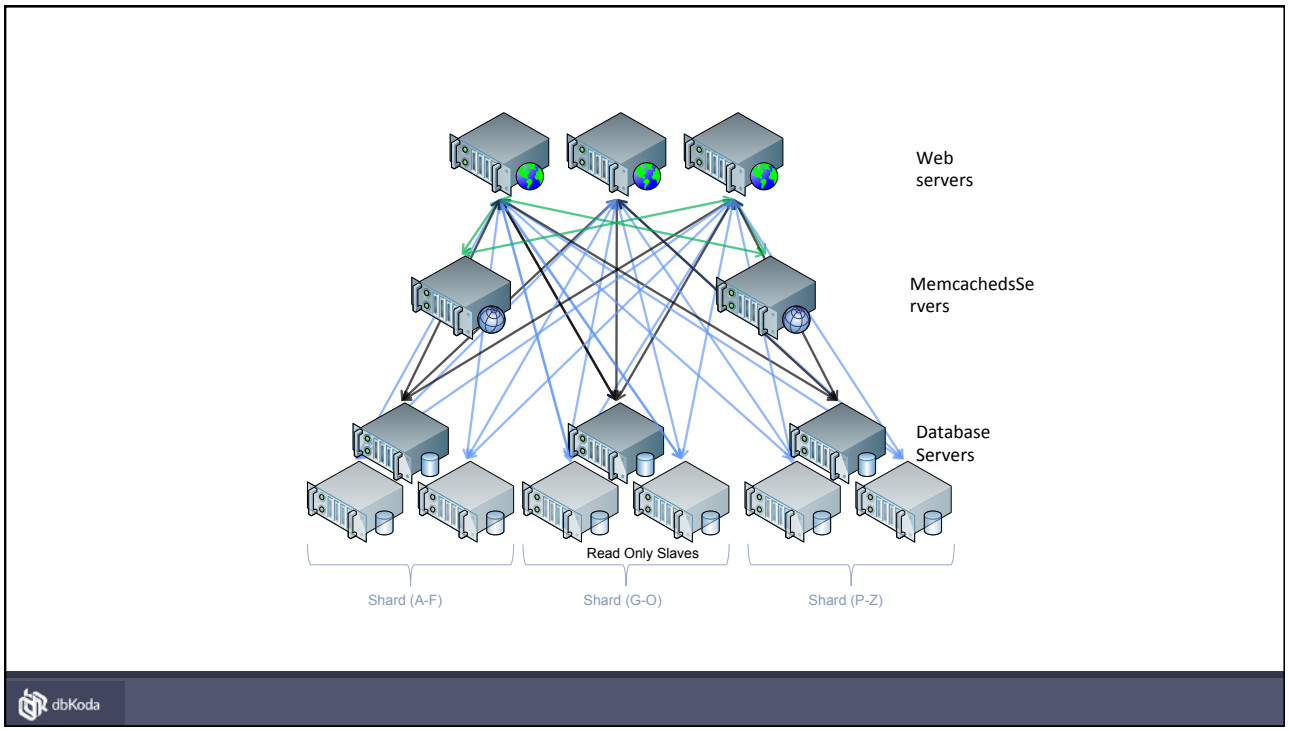
- 16 Clusters
- 32,500 nodes

dbKoda

Hadoop family

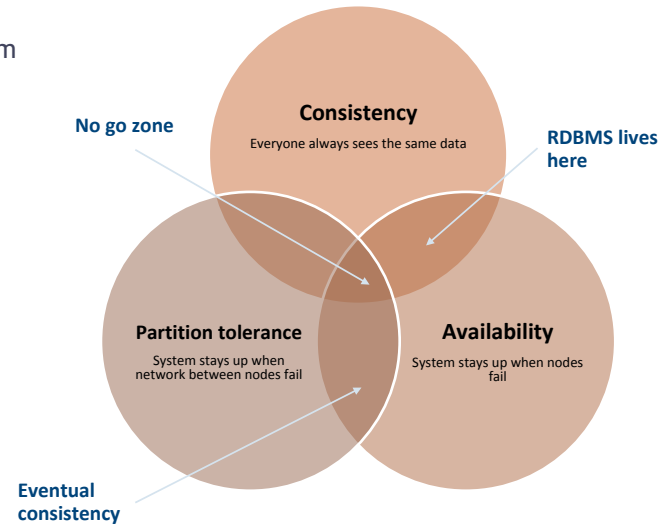


NoSQL

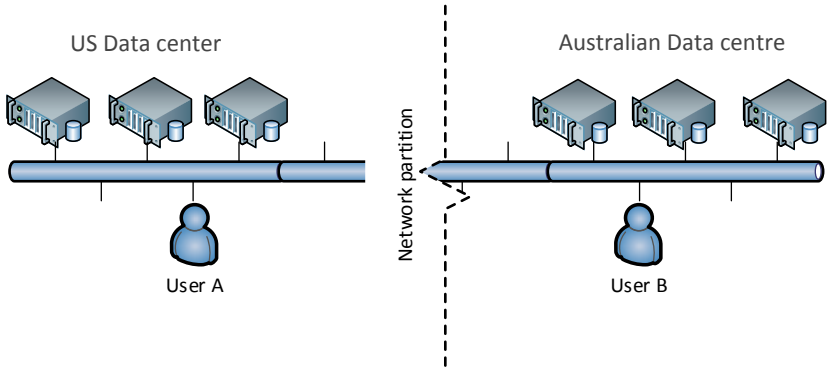


CAP Theorem says something has to give.

CAP (Brewer's) Theorem says you can only have two out of three of Consistency, Partition Tolerance, Availability



Network partition



dbKoda

Major influences on non-relational

Amazon Dynamo

- Eventually consistent transaction model
- Consistent hashing

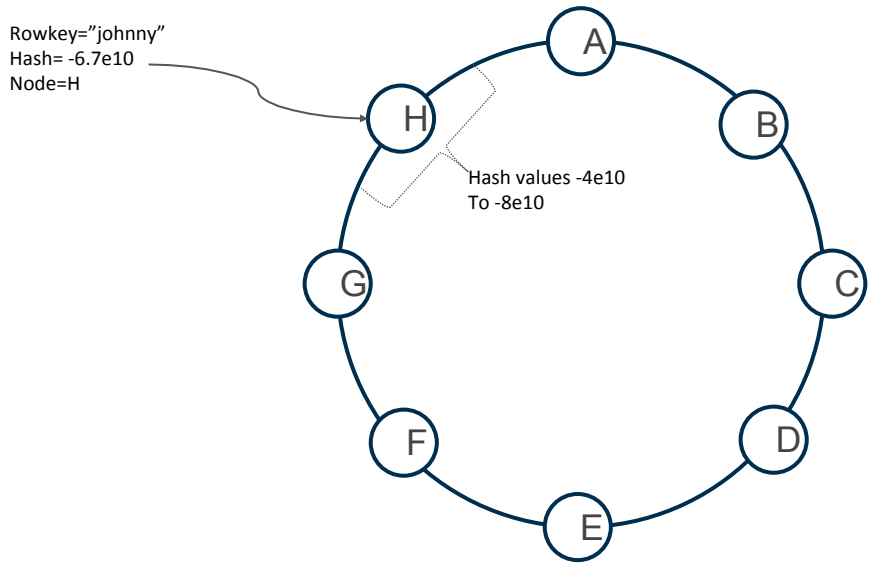
Google BigTable

- Column Family model for sparse distributed columnar data

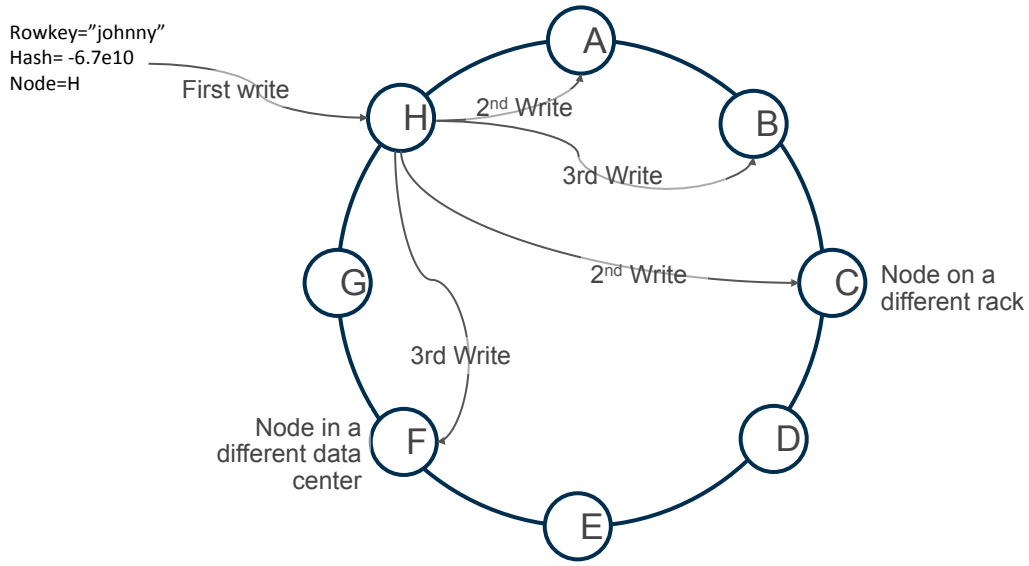
OODBMS and XML DBs

- Paved the way for the document database

Dynamo Consistent Hashing



Dynamo Consistent Hashing

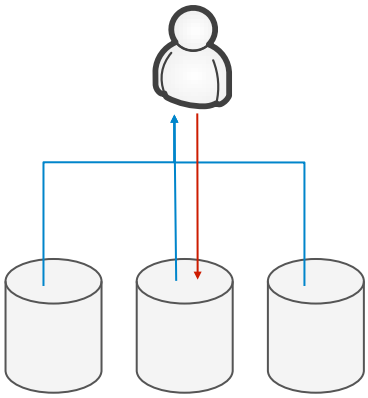


NWR – tunable consistency

N = No of copies of data

W= No of copies that must be written *synchronously* before returning control to program

R = No of copies to Read



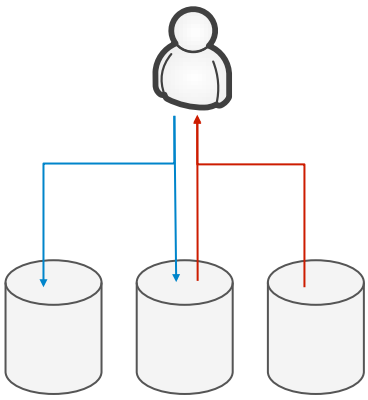
N=3 W=1 R=N
Fastest write, slow but consistent reads
There will be 3 copies of the data.
A write request returns once the first copy is written – the other two can happen later.
A read request reads all copies to make sure it gets the latest version.
Data might be lost if a node fails before the second write.

NWR – tunable consistency

N = No of copies of data

W= No of copies that must be written *synchronously* before returning control to program

R = No of copies to Read



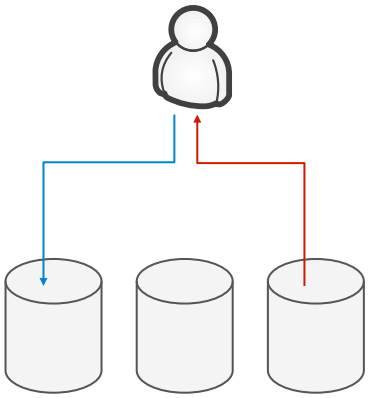
N=3 W=2 R=2
Faster writes, still consistent (quorum assembly)
There will be 3 copies of the data.
A write request returns when 2 copies are written – the other can happen later.
A read request reads two copies to make sure it has the latest version.

NWR – tunable consistency

N = No of copies of data

W= No of copies that must be written *synchronously* before returning control to program

R = No of copies to Read



N=3 W=1 R=1
Fast, but not consistent
There will be 3 copies of the data.
A write request returns once the first copy is written – the other two can happen later.
A read request reads a single version only: it might not get the latest copy.
Data might be lost if a node fails before the second write.

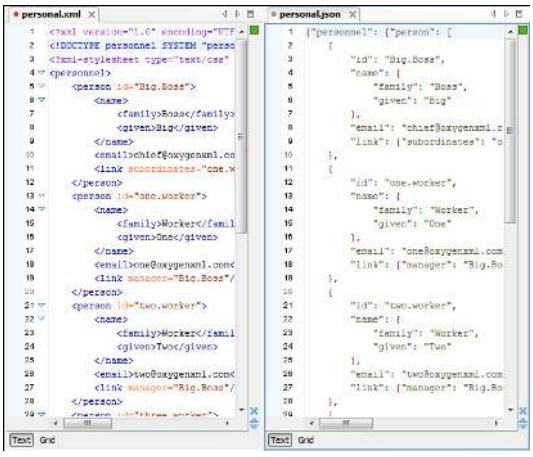
Revenge of the Object Nerds – Document databases

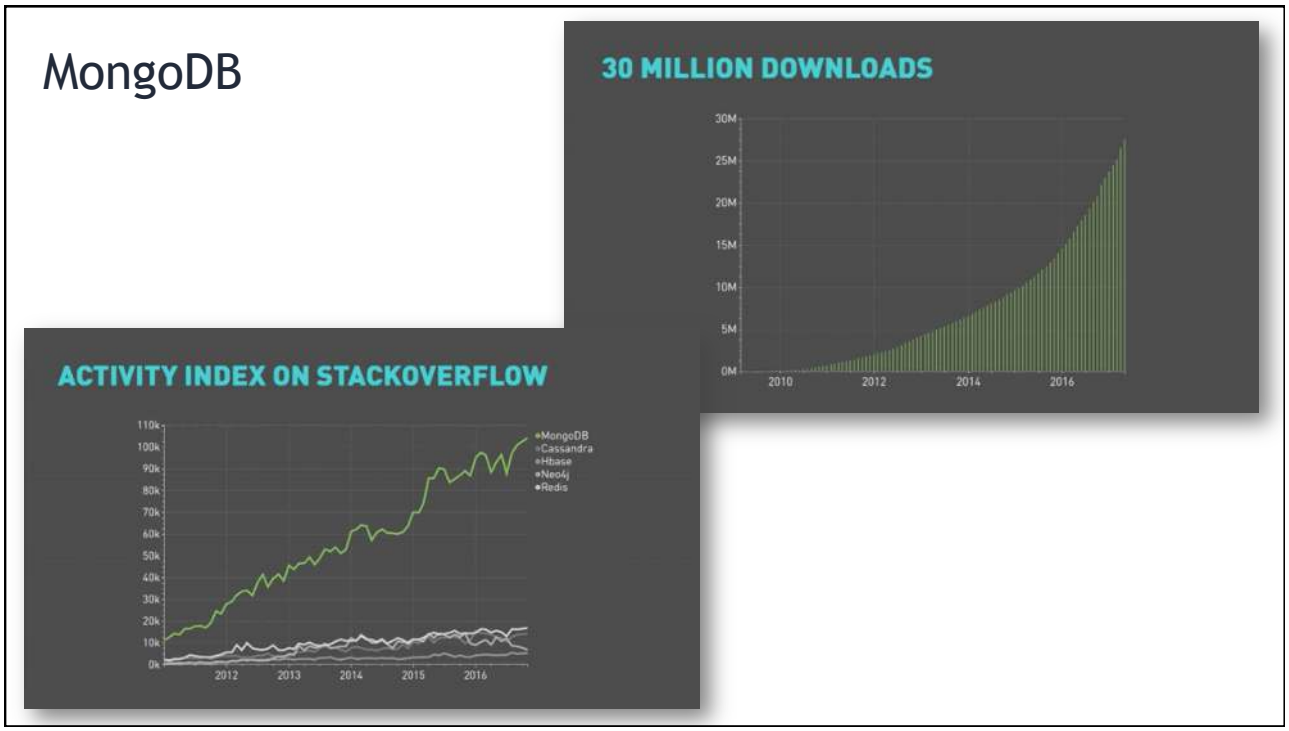
Structured documents – XML and JSON (JavaScript Object Notation) become more prevalent within applications

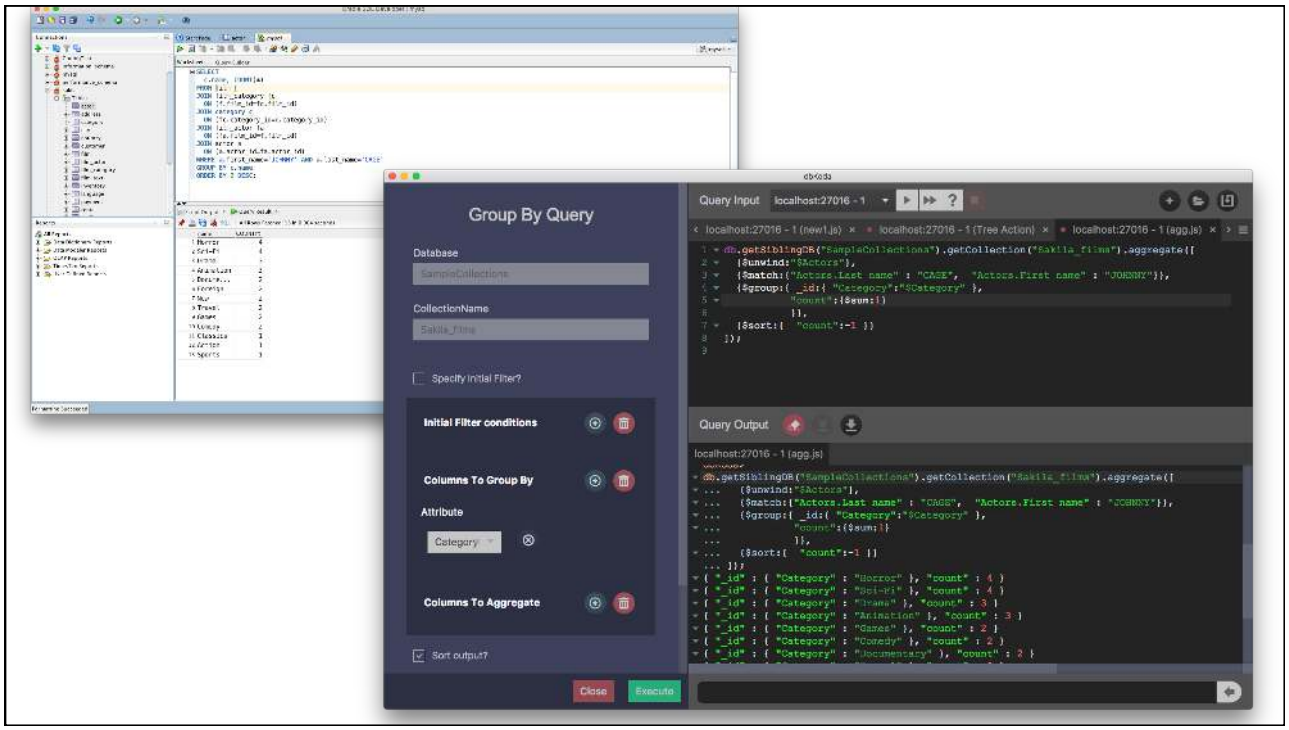
Web programmers start storing these in BLOBS in MySQL

Emergence of XML and JSON databases

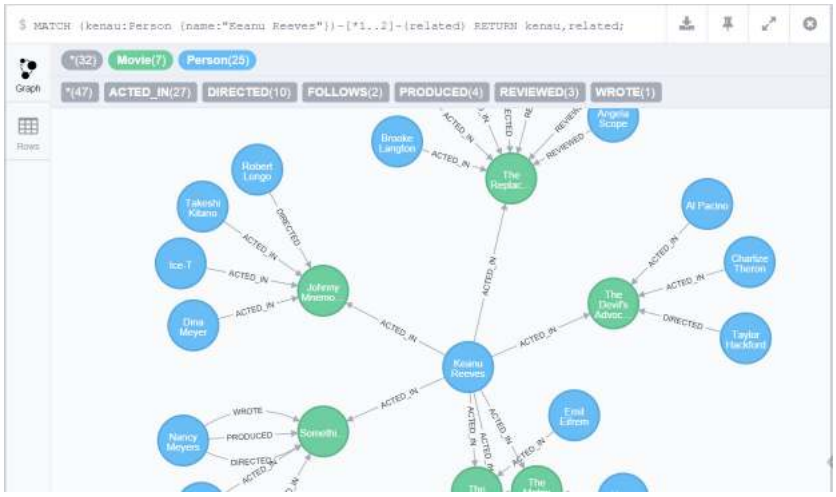
Flexible schemas supported agile development and continuous deployment in a way RDBMS ALTER TABLE could not




MongoDB
30 MILLION DOWNLOADS
30M
25M
20M
15M
10M
5M
0M
2010
2012
2014
2016
ACTIVITY INDEX ON STACKOVERFLOW
110k
100k
90k
80k
70k
60k
50k
40k
30k
20k
10k
0k
2012
2013
2014
2015
2016
MongoDB
Cassandra
Hbase
Neo4j
Redis


Group By Query
Database
SampleCollections
CollectionName
Sakila_Films
Specify Initial Filter?
Initial Filter conditions
Columns To Group By
Attribute
Category
Columns To Aggregate
Sort output?
Close
Execute
Query Input
Query Output

Graph databases



No means yes!

Spark SQL

SHARK

HIVE

N1QL – JSON + SQL

{SQL}

Not Only SQL

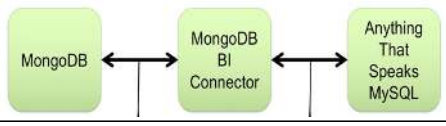
APACHE DRILL

ORACLE BIG DATA

Oracle Big Data SQL

TERADATA

The MongoDB BI Connector: A "SQL Bridge"



cloudera IMPALA

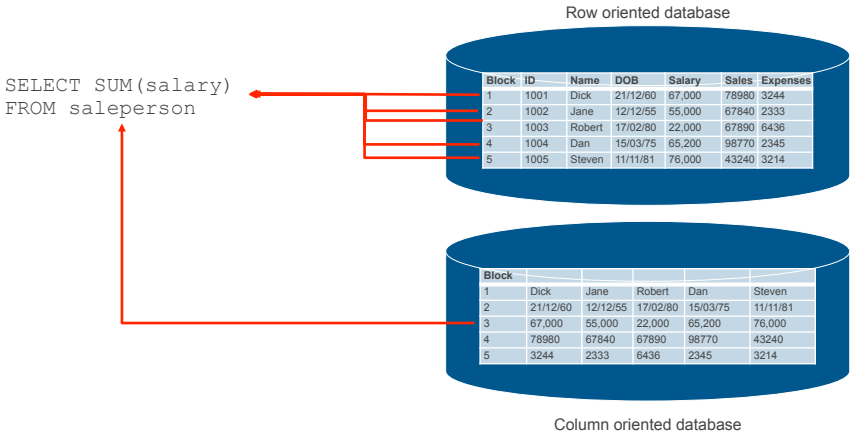
presto

Column-oriented Databases

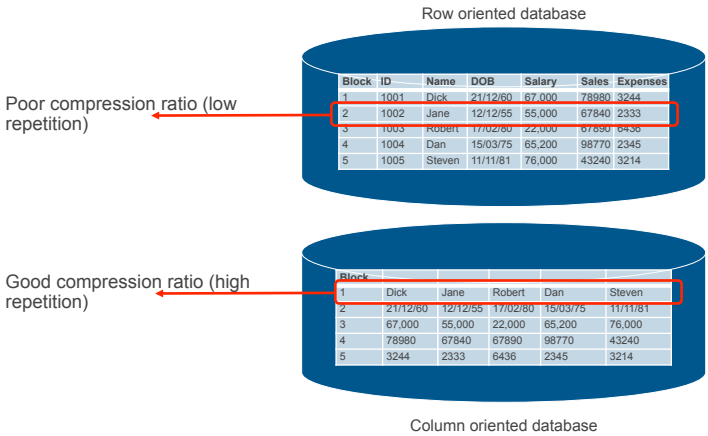
Row orientation vs column orientation

ID	Name	DOB	Salary	Sales	Expenses
1001	Dick	21/12/60	67,000	78980	3244
1002	Jane	12/12/55	55,000	67840	2333
1003	Robert	17/02/80	22,000	67890	6436
1004	Dan	15/03/75	65,200	98770	2345
1005	Steven	11/11/81	76,000	43240	3214

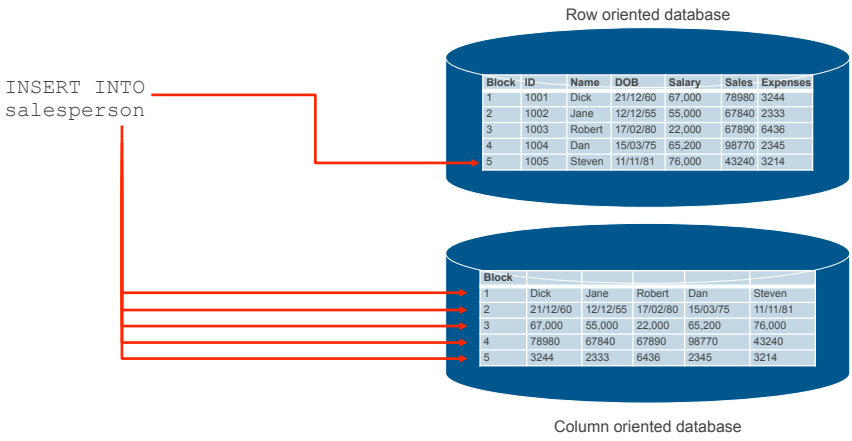
Analytical queries



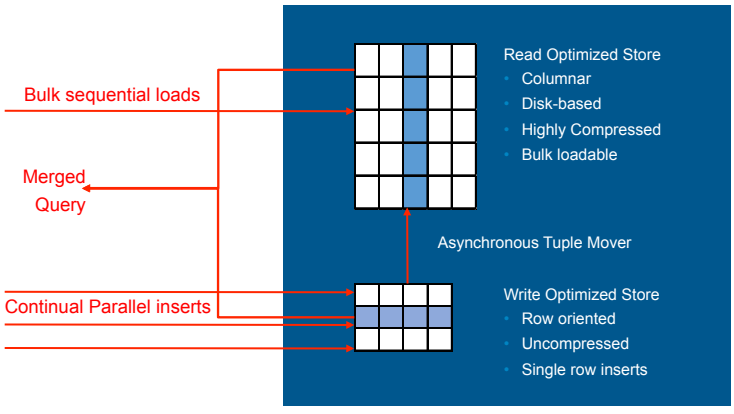
Compression



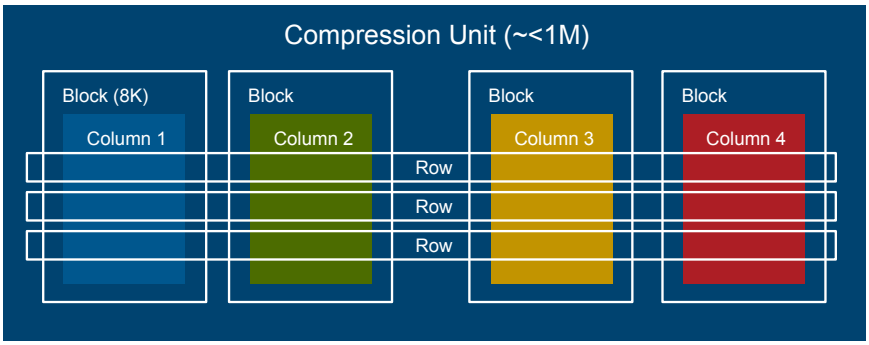
Inserts



C-Store (Vertica) solution for inserts



Exadata Hybrid Columnar Compression (EHCC)

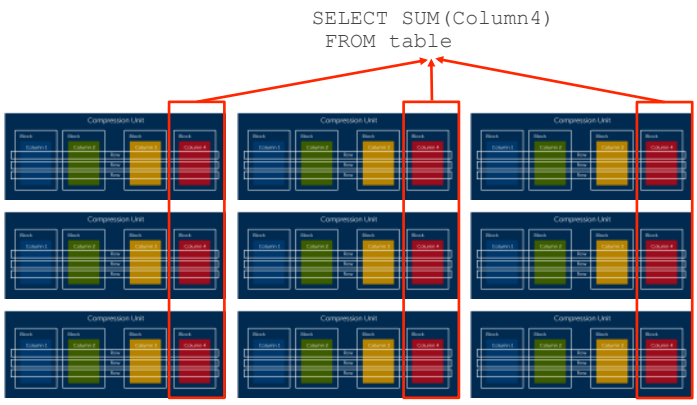


Exadata Hybrid Columnar Compression

Provides high compression ratio

Manageable impact on row read/write operations

Some optimization of analytic queries

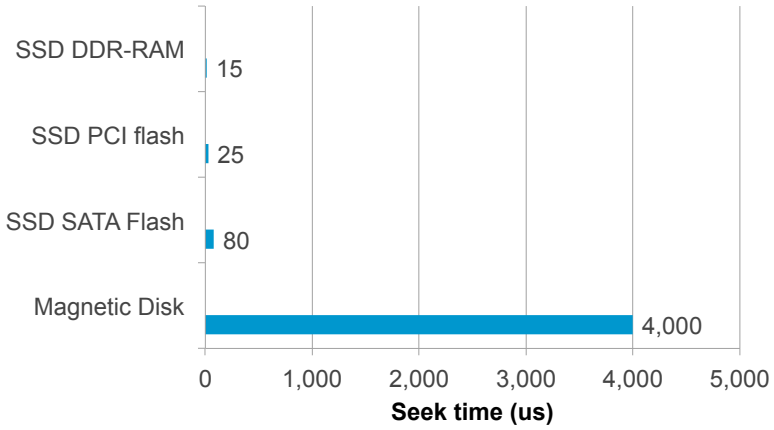


SSD and in-memory databases

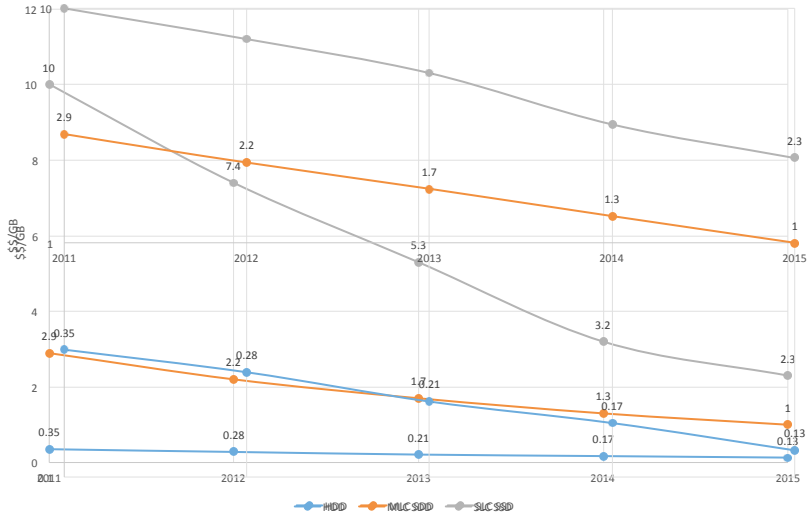
5MB HDD circa 1956

The more that things change....

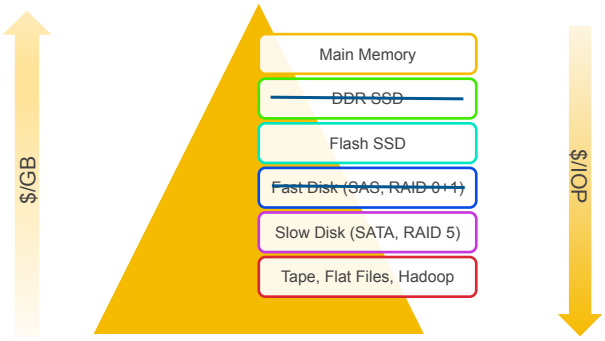
Cheaper by the IO



But not by the GB



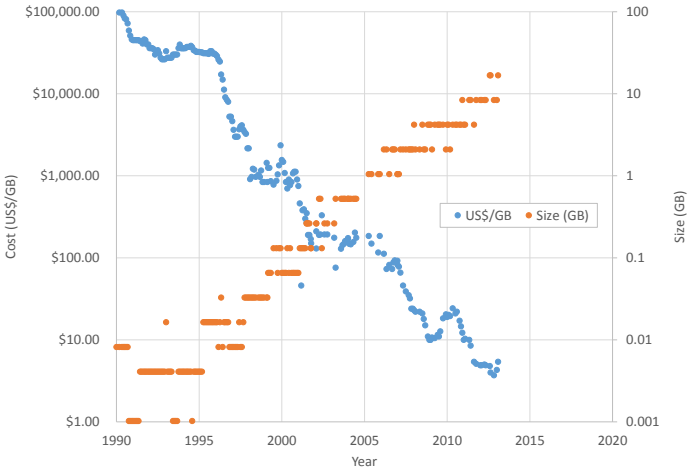
Tiered storage management



In-memory databases

Cost of RAM falling 50% each 18 months.

Some databases can fit entirely within the RAM of a single server or cluster of servers



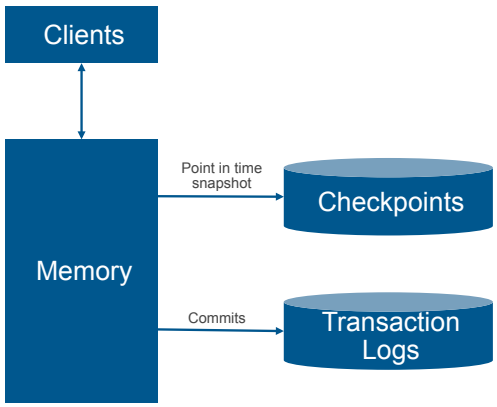
Oracle times ten

In-memory transactional database

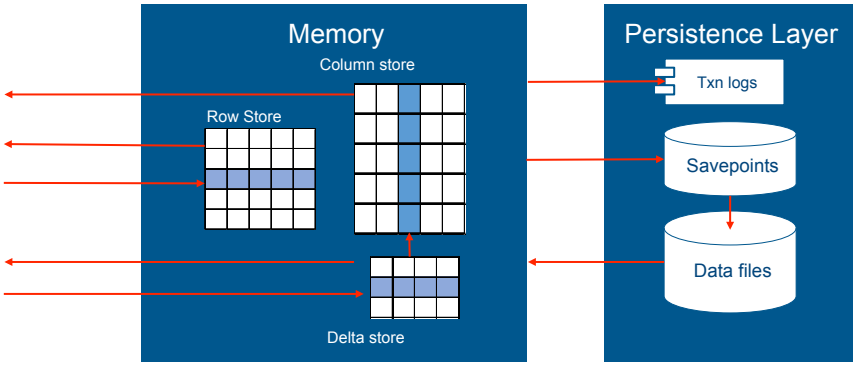
Disk-based Checkpoints and disk-based logging

By default, COMMITs are not durable (writes to the transaction log are asynchronous).

Can configure synchronous replication or synchronous log writes to avoid data loss



SAP HANA



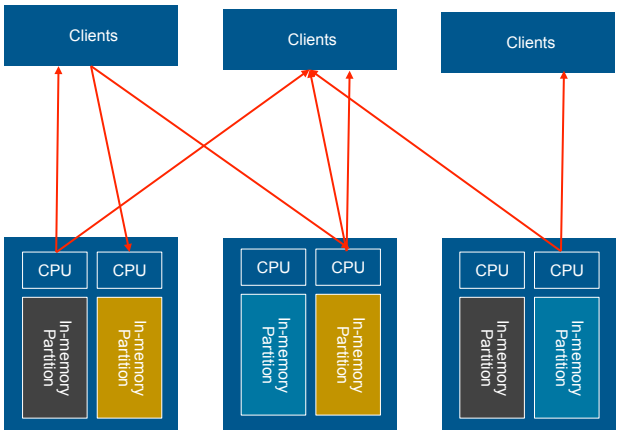
Note: Table must be either row or column – not both

VoltDB

Single threaded access to memory: no latch/mutex waits

Transactions in self-contained stored procedures: minimal locking

K-Safety for COMMIT: No sync waits



Spark (sort of) in-memory Hadoop

In Memory compute

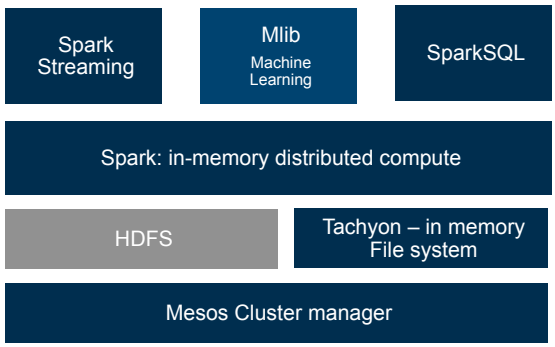
HDFS compatible

Libraries for data processing, machine learning, streaming, SQL, etc

Python and Scala interfaces

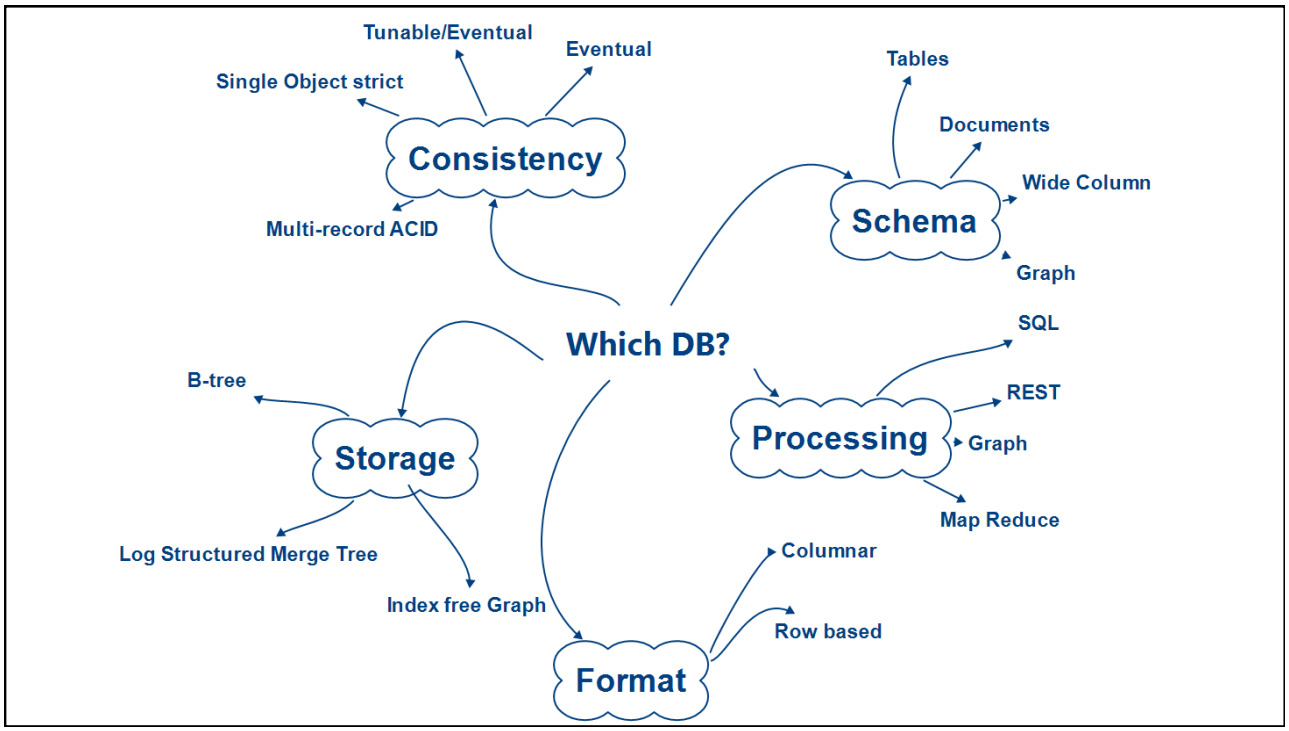
Part of the Berkeley Data Analytic Stack

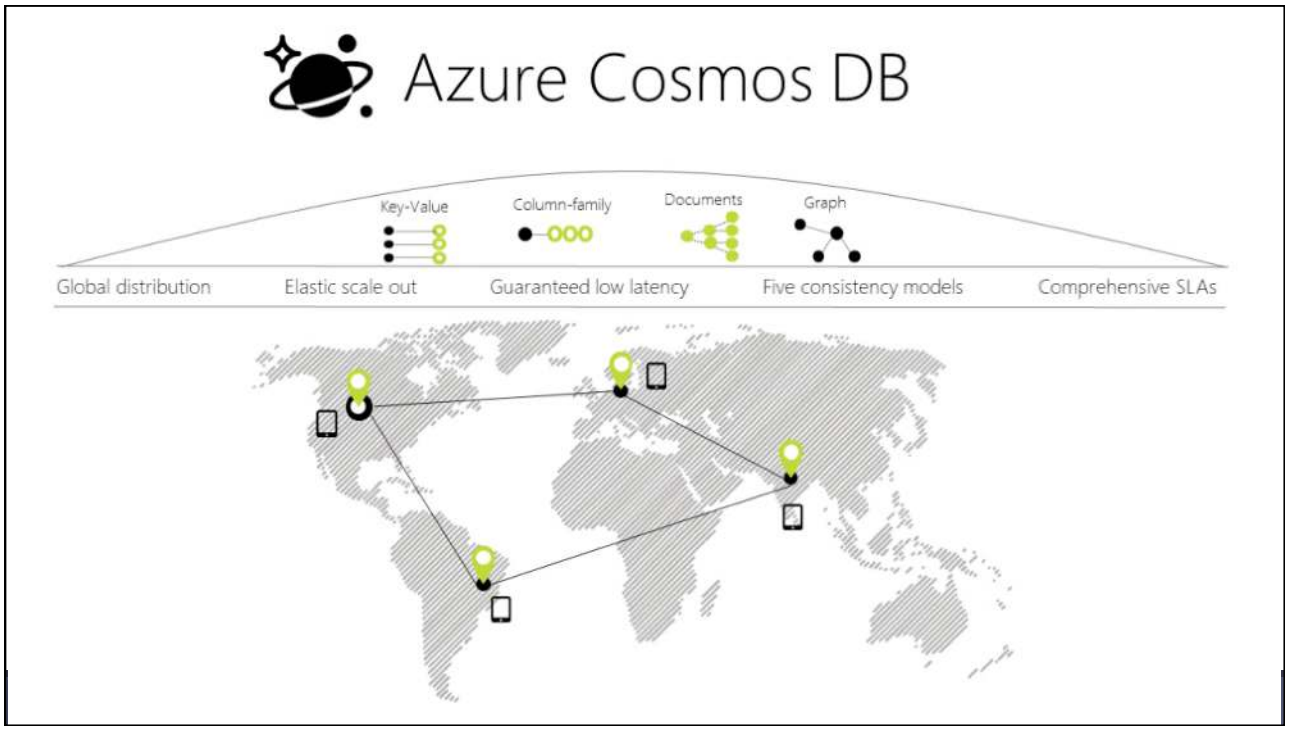
Integrating into all Hadoop distributions (and Cassandra)



What will the database of the future look like?

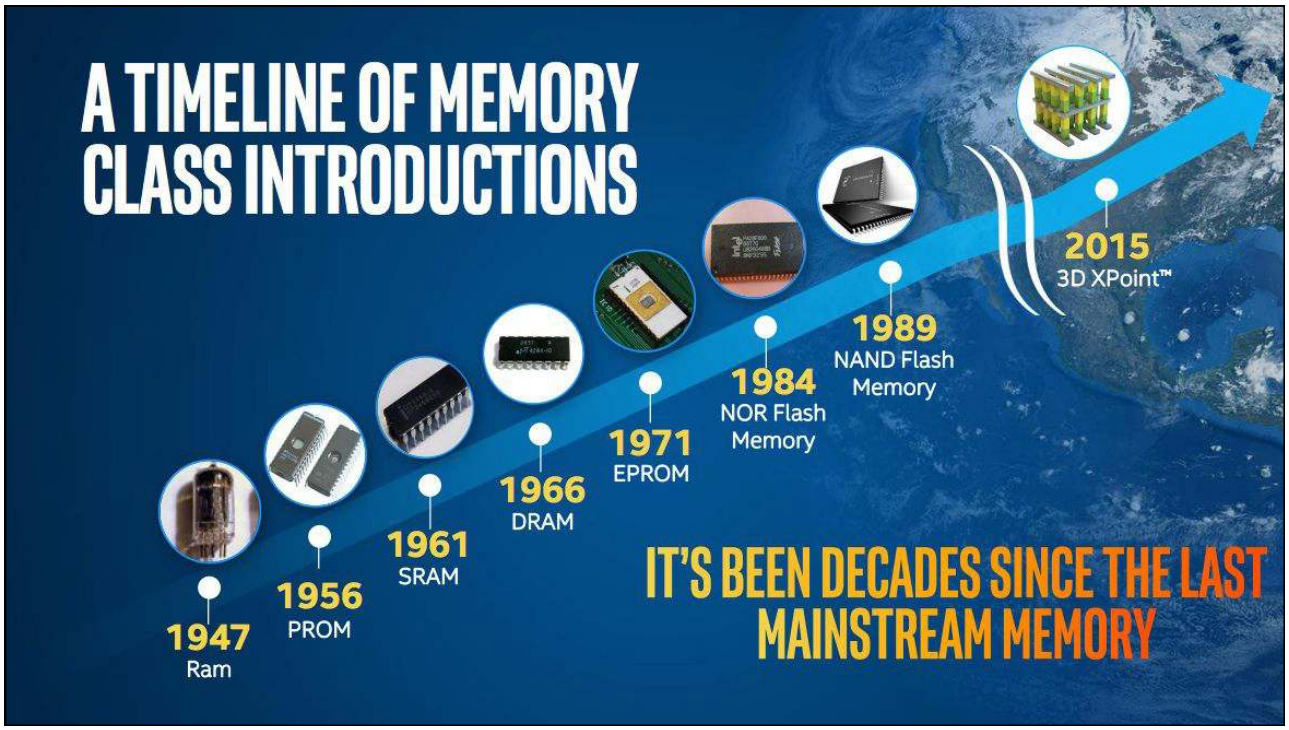
dbKoda


Tunable/Eventual
Eventual
Single Object strict
Consistency
Multi-record ACID
Tables
Documents
Wide Column
Schema
Graph
Which DB?
SQL
B-tree
REST
Storage
Processing
Graph
Map Reduce
Log Structured Merge Tree
Index free Graph
Columnar
Row based
Format

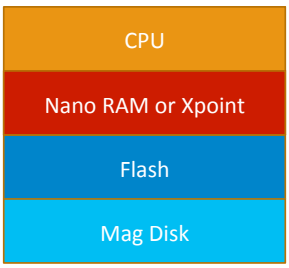

Azure Cosmos DB
Key-Value
Column-family
Documents
Graph
Global distribution
Elastic scale out
Guaranteed low latency
Five consistency models
Comprehensive SLAs

Disruptive Technologies

dbKoda



Universal Memory

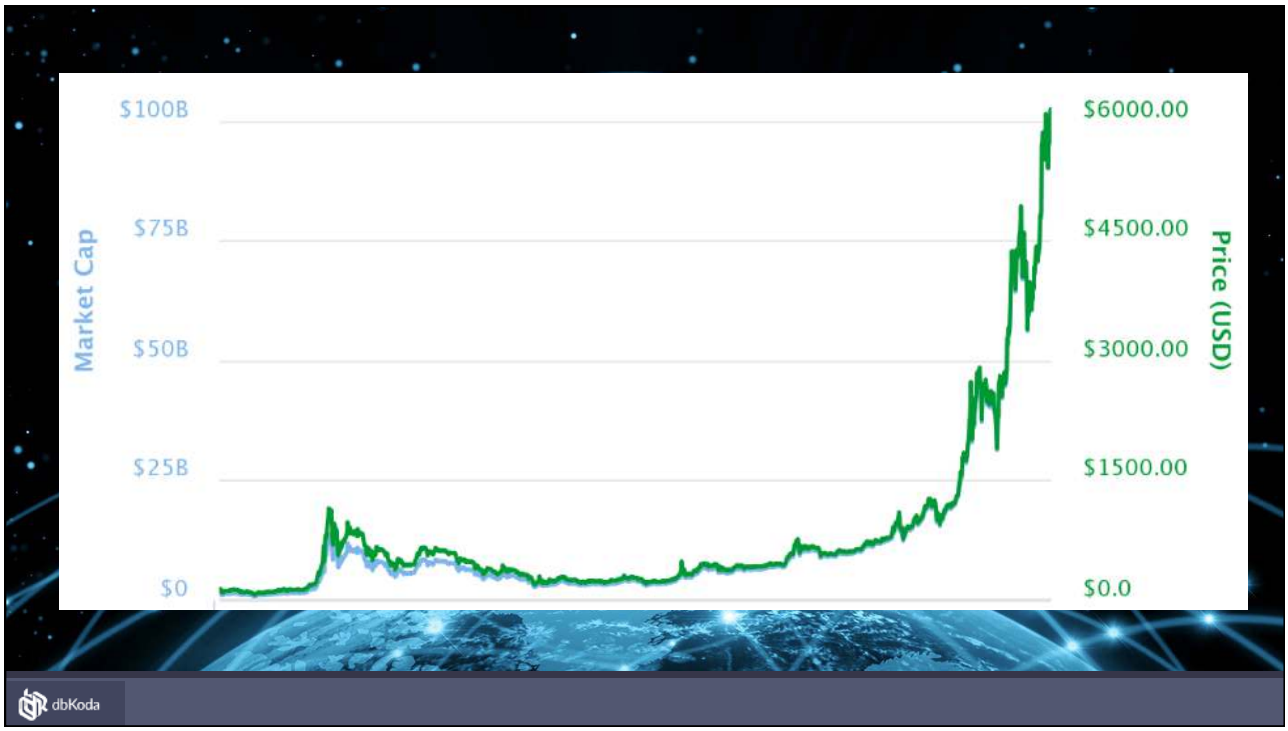


A single technology that provided performance & economics at all layers of storage would eliminate the rationale behind almost all existing database architectures

dbKoda

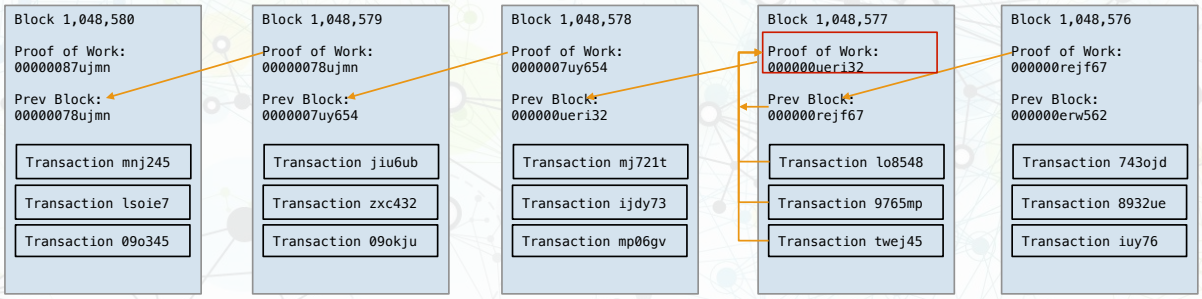
Blockchain

dbKoda



Blockchain basics

Proof of work (nonce) is dependent on transactions in the current block and signature of previous block



Changing a transaction on the blockchain requires rebuilding the block and all subsequent blocks faster than the all of the current nodes on the blockchain network. This becomes effectively impossible the deeper the block is on the chain

Calculating the nonce is computationally expensive. First node on the blockchain network to correctly calculate the nonce is rewarded with bitcoins

dbKoda

Blockchains and databases

The Blockchain represents a new type of database

- Public
- Distributed
- Tamper proof
- Immutable history

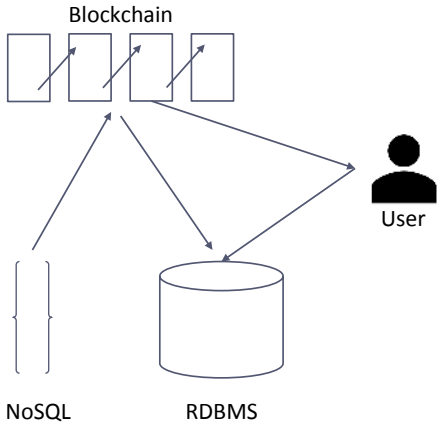
But doesn't provide traditional database features:

- Low throughput
- High latency
- No structure
- Limited volumes
- Excessively transparent

dbKoda

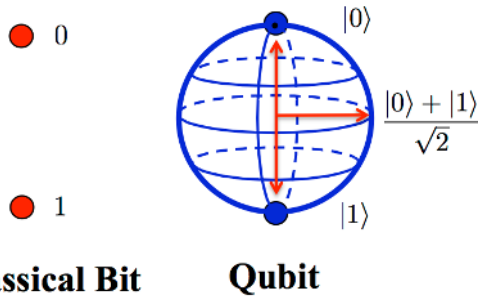
Databases of the future will leverage blockchain:

- Mediate changes to private data on public databases
  - Maintain the integrity of your personal data on a public database
- Facilitate transactions between disparate databases
  - Enable transactions between Oracle (shudder!) and MongoDB
- Assert proof of state
  - "Seal" data within a database



dbKoda

Quantum Computing



dbKoda

Thanks!

guy@southbanksofware.com
@guy_harrison
http://guy_harrison.net

dbKoda