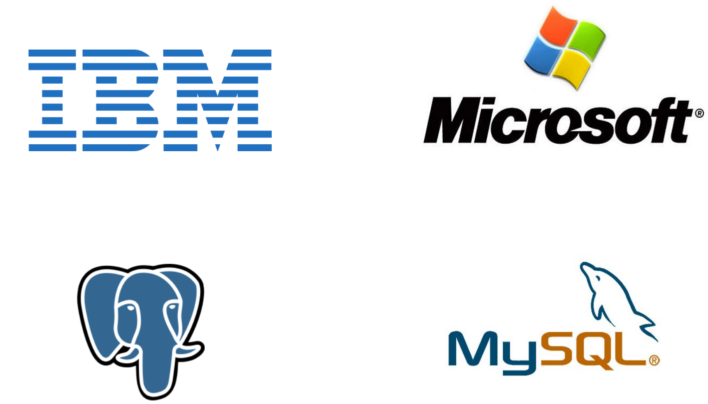
Welcome

It used to be **easy**...

they all looked pretty much alike



NoSQL	BigData	MapReduce	Graph	Document
BigTable	Shared Nothing	Column Oriented	CAP	Eventual Consistency
ACID	BASE	Mongo	Coudera	Hadoop
Voldemort	Cassandra	Dynamo	Marklogic	Redis
Velocity	Hbase	Hypertable	Riak	BDB

Now it’s downright

cOnfuZ1nG!

What Happened?

we changed scale

we changed tack

so where does

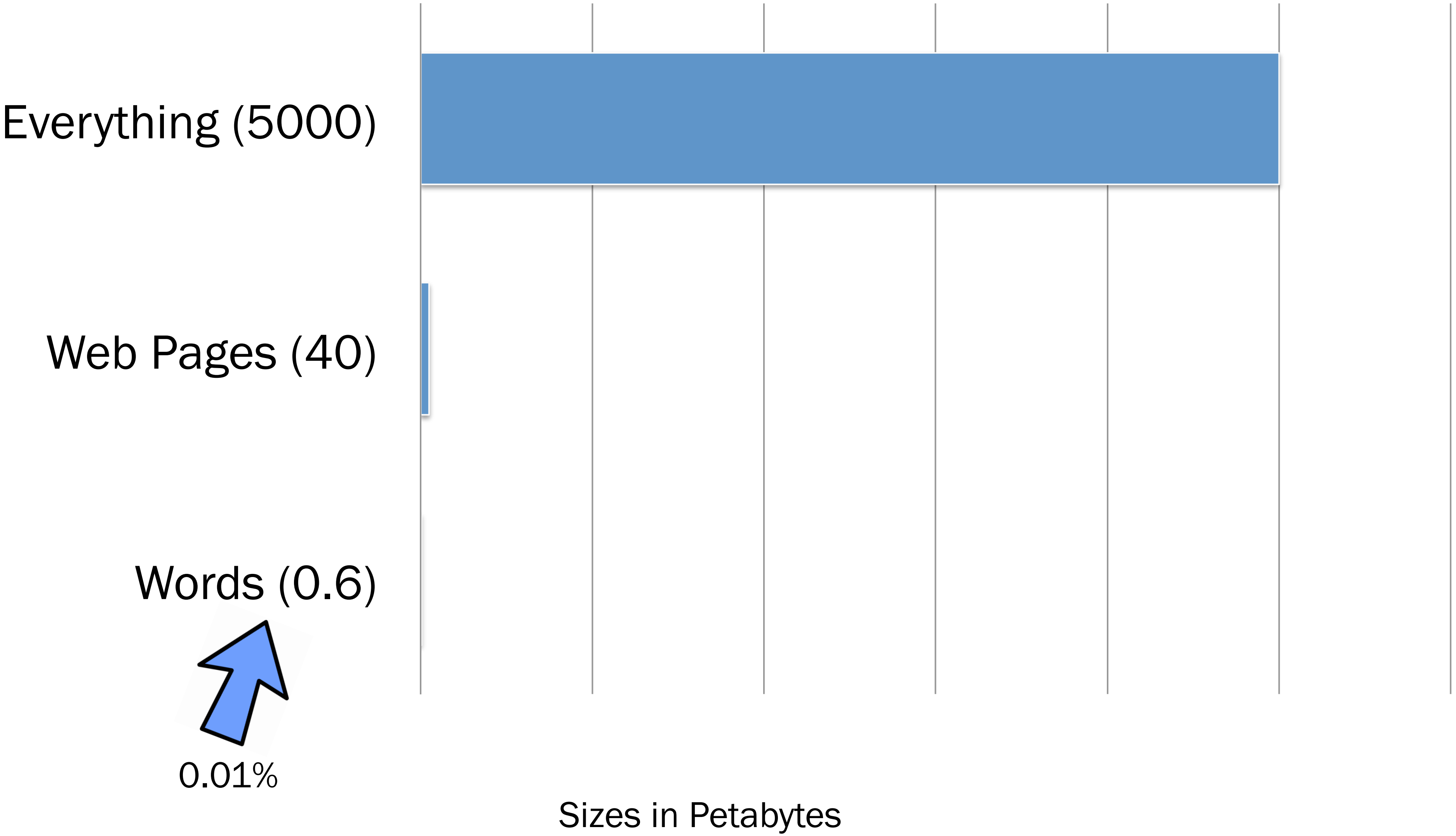
big data meet

big database?

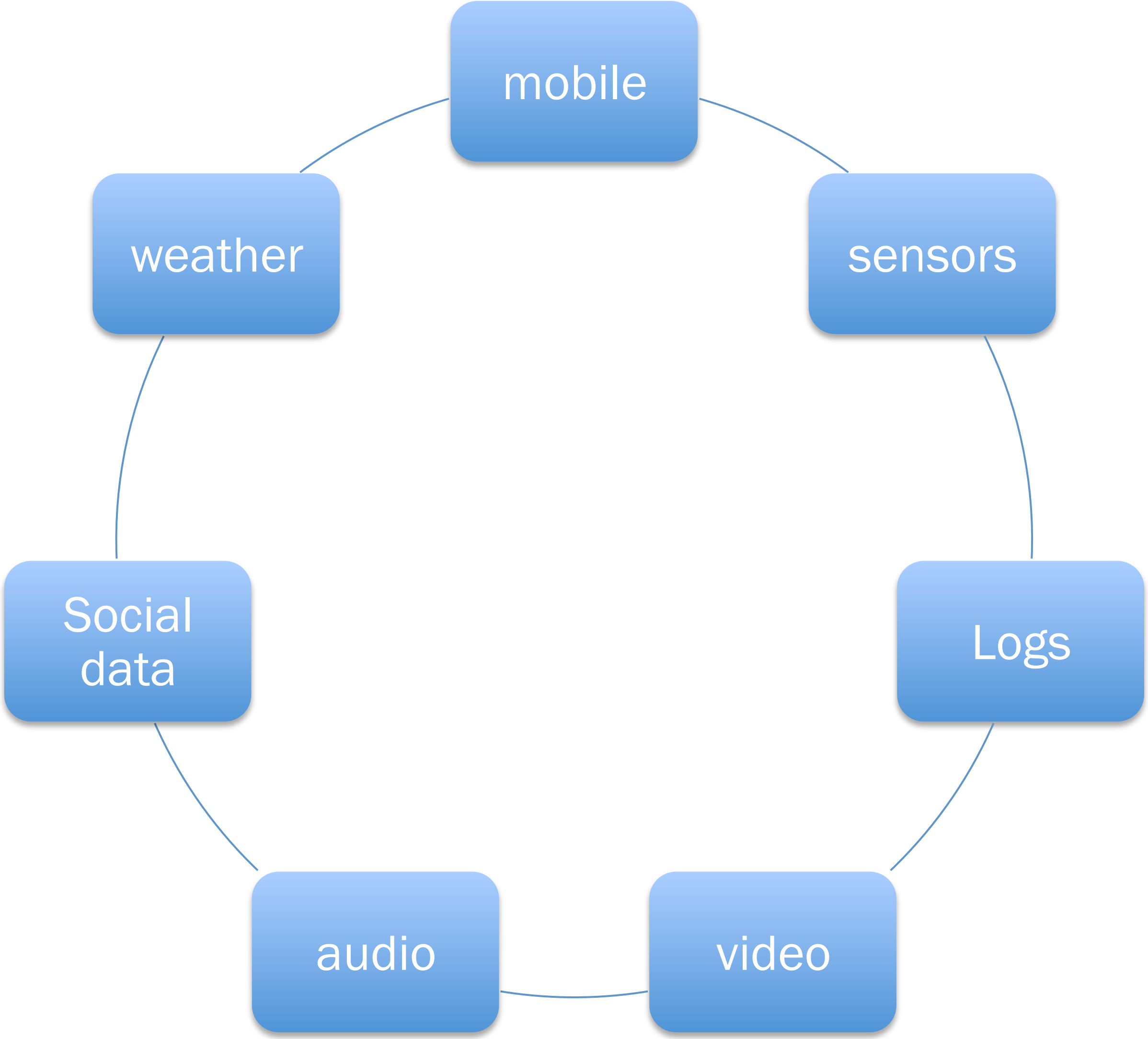
The world’s largest NoSQL database?

The Internet

So how Big is Big?



Many more Big Sources



mobile

sensors

Logs

video

audio

Social data

weather

But it is pretty useful

- Marketing
- Fraud detection
- Tax Evasion
- Intelligence
- Advertising
- Scientific research

Gartner

80% of business is conducted on unstructured information

Big Data is now a new class of economic asset*

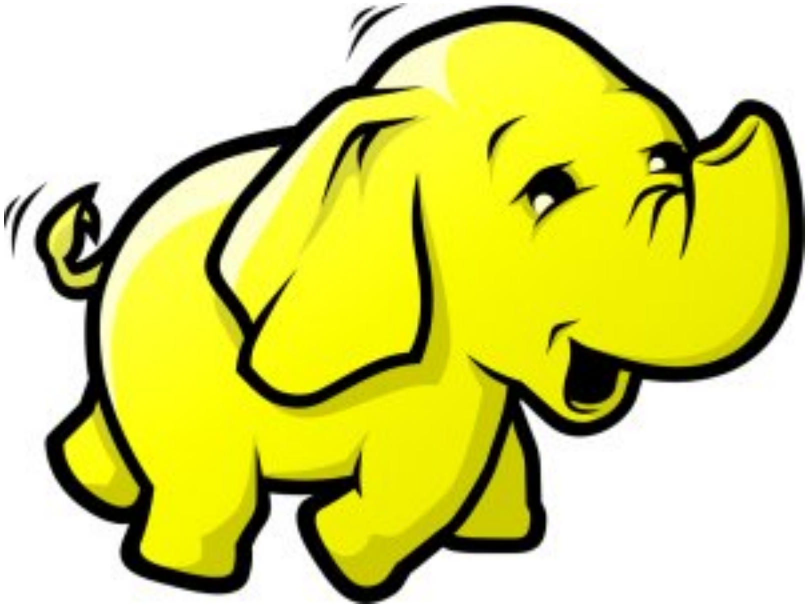
*World economic forum 2012

Yet 80% Enterprise Databases < 1TB

Along came the Big Data Movement

MapReduce (2004)

- Large, distributed, ordered map
- Fault-tolerant file system
- Petabyte scaling



Disruptive

Simple

Pragmatic

Solved an insoluble problem

Unencumbered by tradition (good & bad)

Hacker rather than Enterprise culture

A Different Focus

Tradition

- Global consistency
- Schema driven
- Reliable Network
- Highly Structured

The new wave

- Local consistency
- Schemaless / Last
- Unreliable Network
- Semi-structured/ Unstructured

Novel?

Possibly better put as:

A timely and elegant combination of existing ideas, placed together to solve a previously unsolved problem.

Backlash (2009)

Not novel (dates back to the 80's)

Physical level not the logical level (messy?)

Incompatible with tooling

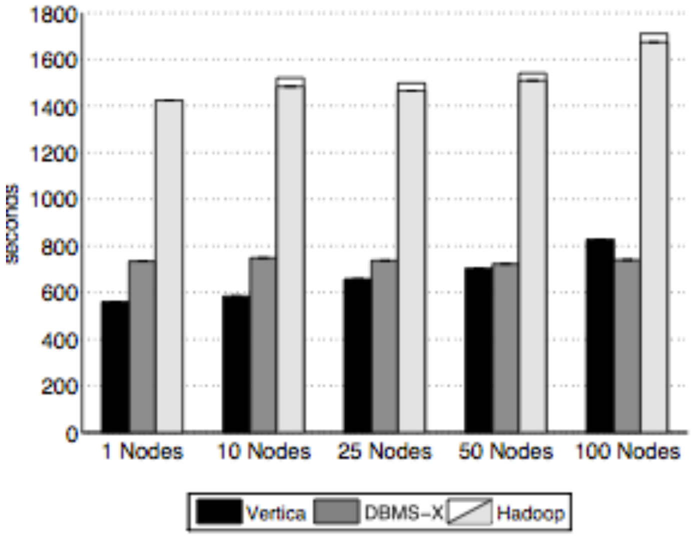
Lack of integrity (referential) & ACID

MR is brute force ignoring indexing, scew

All points are reasonable

And they proved it too!

"A comparison of Approaches to Large Scale Data Analysis" – Sigmod 2009



7: Aggregation Task Results (2.5 million Groups)

- Vertica vs. DBMSX vs. Hadoop
- Vertica up to 7 x faster than Hadoop over benchmarks

→ Databases faster than Hadoop

But possibly missed the point?

Was MapReduce was not supposed to be a Data Warehousing tool?

If you need more, layer it on top

For example Tensing & Magastore
@ Google

So MapReduce represents a bottom-up approach to accessing very large data sets that is unencumbered by the past.

…and the Database Field knew it had Problems

We Lose: Joe Hellerstein (Berkeley) 2001

“Databases are commoditised and cornered to slow-moving, evolving, structure intensive, applications that require schema evolution.“ ...

“The internet companies are lost and we will remain in the doldrums of the enterprise space.” ...

“As databases are black boxes which require a lot of coaxing to get maximum performance”

Yet they do some very cool stuff

Statistically based optimisers, Compression, indexing structures, distributed optimisers, their own declarative language

love
hate

They are an Awesome Tool

They Don’t talk our Language

They Default to Constraint

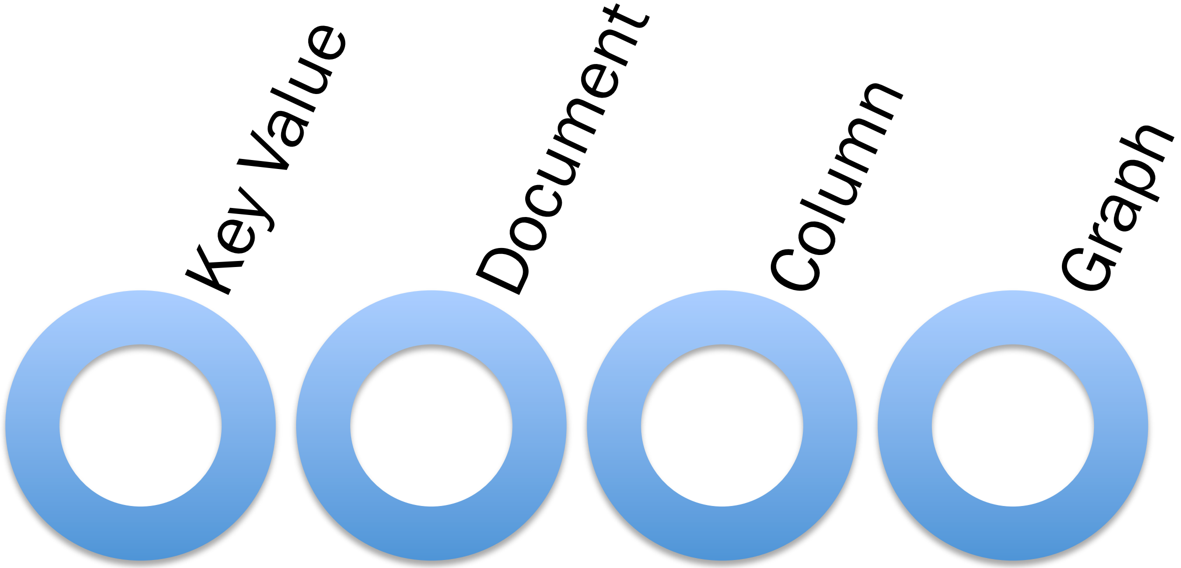
So NoSurprise with NoSQL then

- Simpler Contract
- Shared nothing
- No joins / ACID
- No impedance mismatch
- No slow schema evolution
- Simple code paths
- Just works

The NoSQL Approach

Simple, flexible storage over a diverse range of data structures that will scale almost indefinitely.

Different Flavours



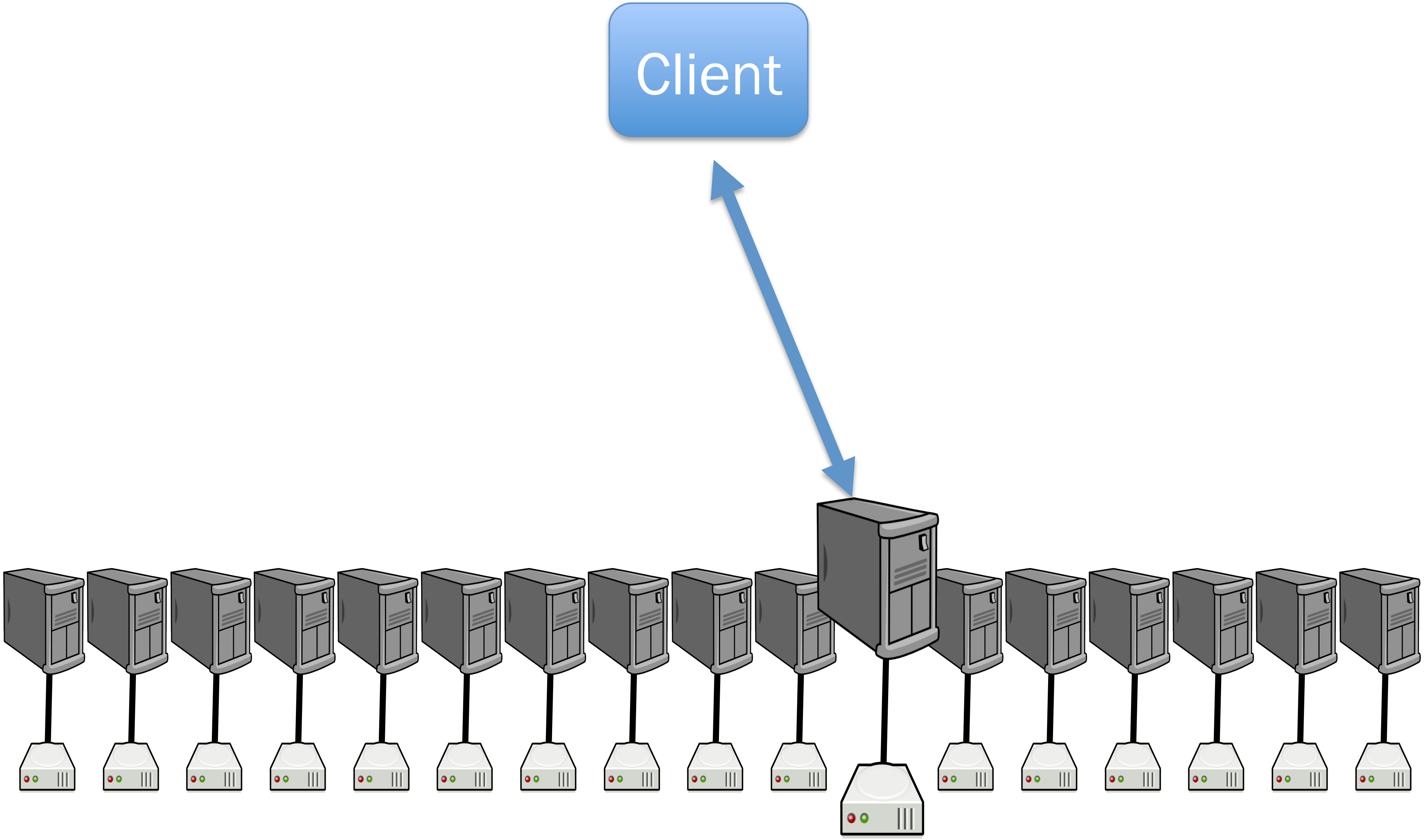
Key Value

Document

Column

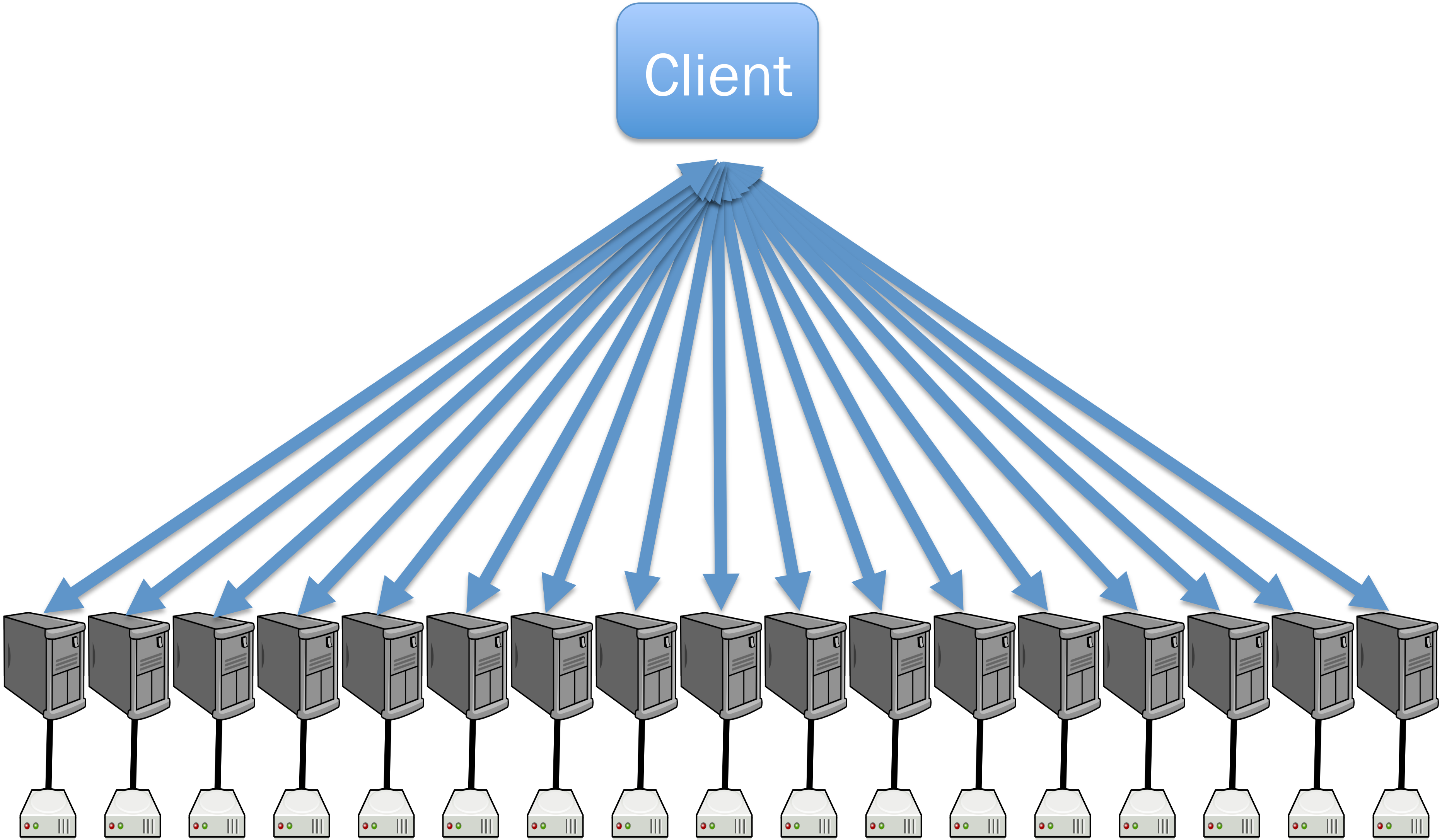
Graph

Two Ways In: Key Based Access



Client

Two Ways In: Broadcast to Every Node



So..

A simple bottom up approach to data storage that scales almost indefinitely.

- No relations
- No joins
- No SQL
- No Transactions
- No sluggish schema evolution

The Relational Database

The ‘Relational Camp’ had been busy too

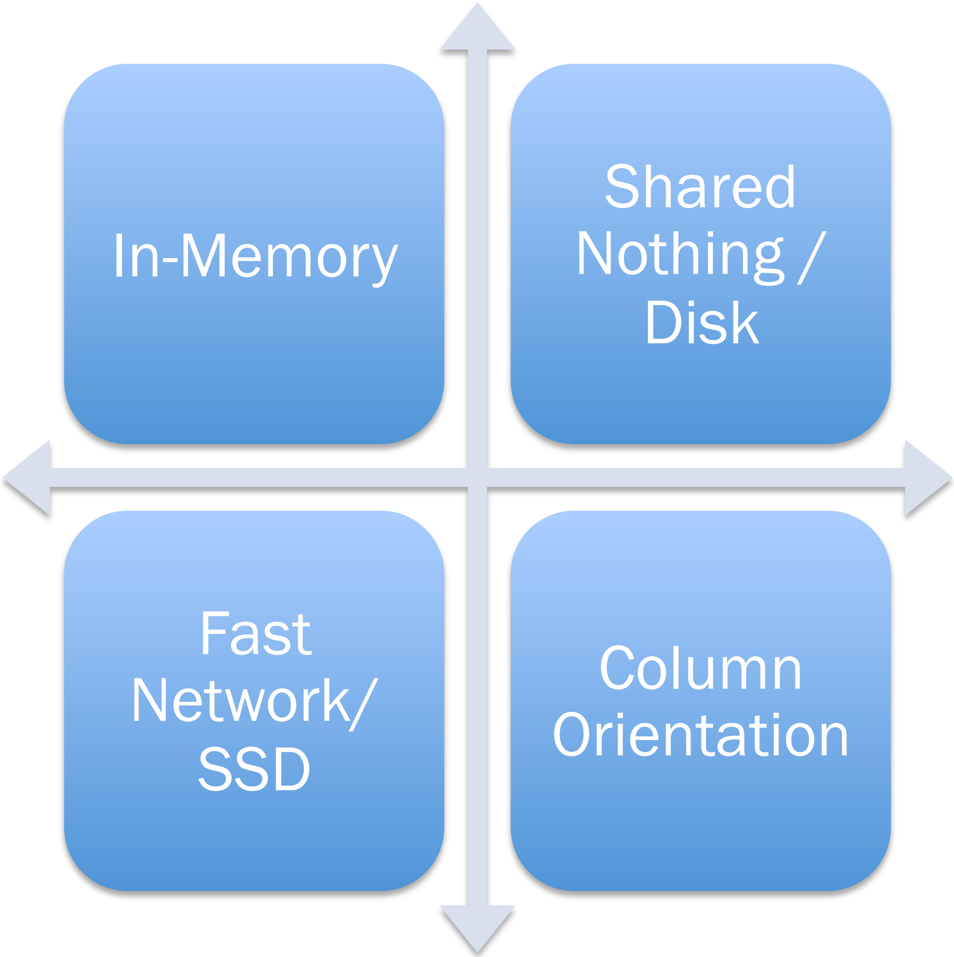
Realisation that the traditional architecture was insufficient for various modern workloads

End of an Era Paper - 2007

“Because RDBMSs can be beaten by more than an order of magnitude on the standard OLTP benchmark, then there is no market where they are competitive. As such, they should be considered as legacy technology more than a quarter of a century in age, for which a complete redesign and re-architecting is the appropriate next step.” – Michael Stonebraker

No Longer a One-Size-Fits-All

Architecting for Different Non-Functionals



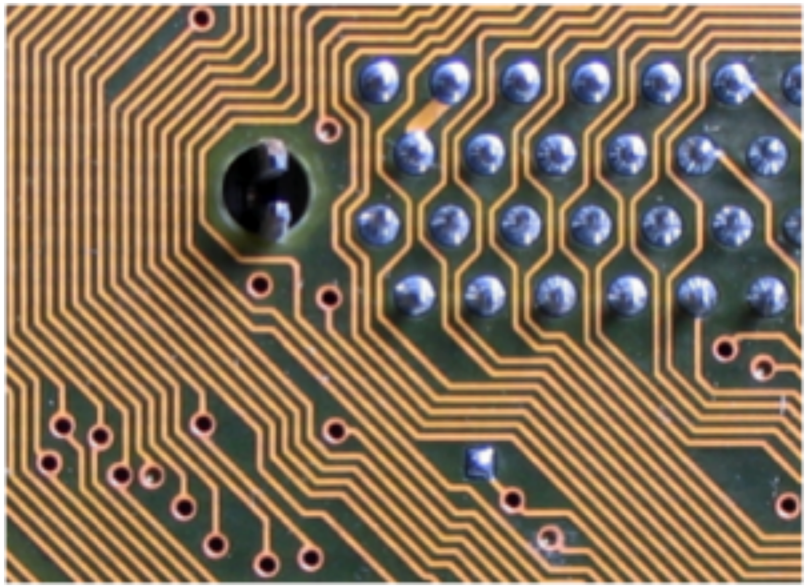
In-Memory

Shared Nothing / Disk

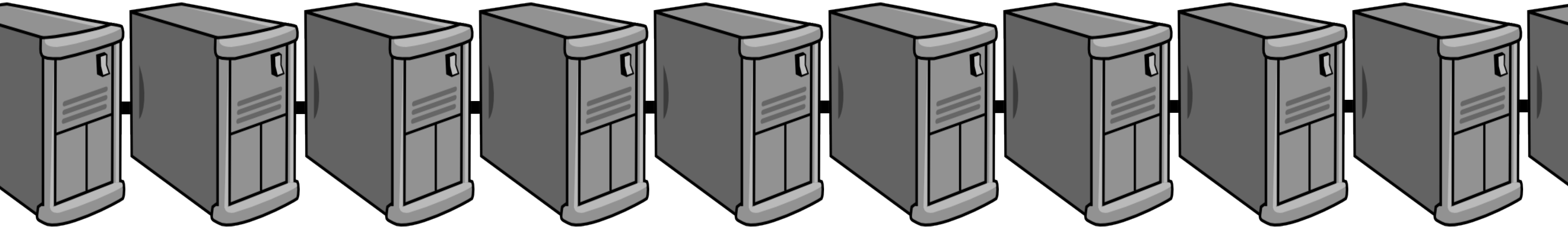
Fast Network/ SSD

Column Orientation

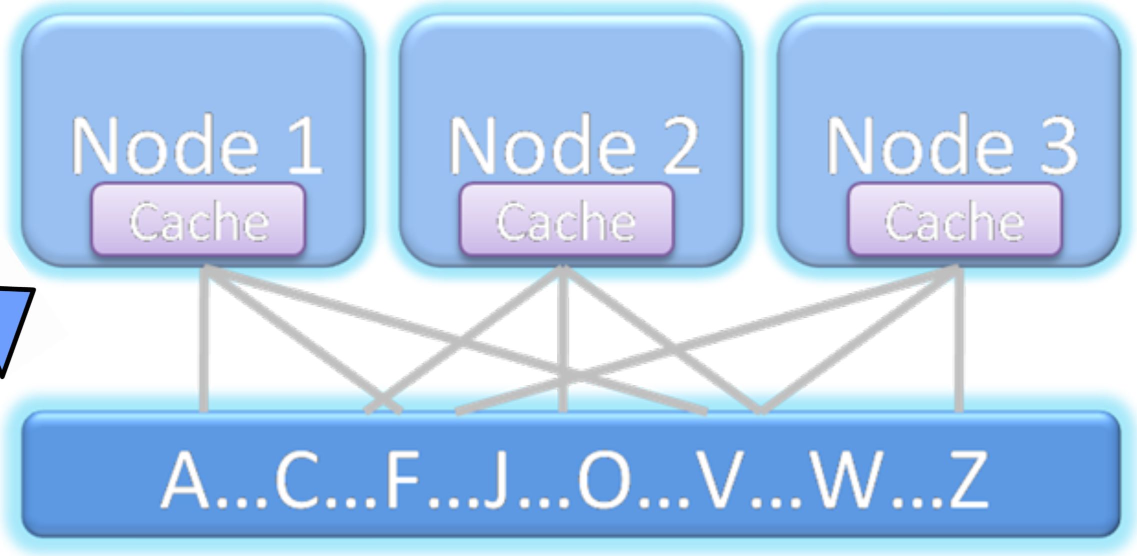
In-Memory



Distributed In-Memory



Shared Disk Architecture



Cache sits above whole dataset

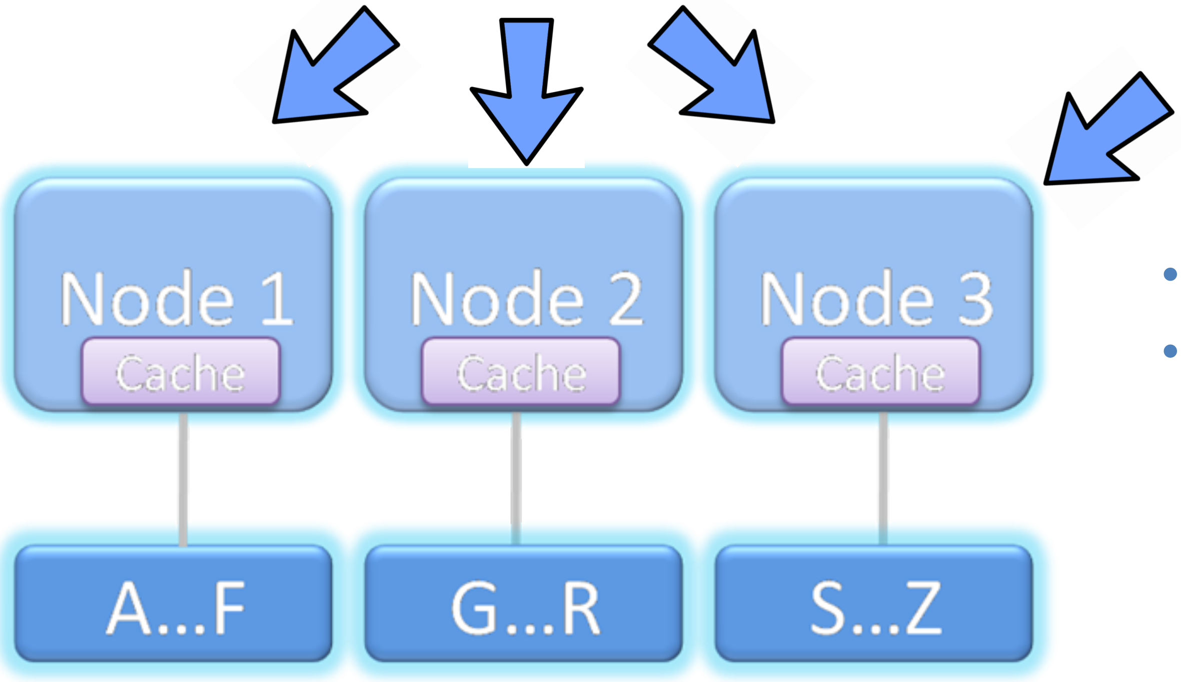
Single node can handle any query

All machines see all data

Shared Nothing Architecture

Queries hit every node

Cache
over just
the shard



- Autonomy over a shard
- Divide and conqueror (non-key hit every node)

Vendors polarise over this issue

Shared Nothing

- TerraData (Aster Data)
- Netezza (IBM)
- ParAccel
- Vertica
- Greenplumb

Shared Everything

- Oracle RAC/Exadata
- IBM purescale
- Sybase IQ
- Microsoft SQL Server

(there is some blurring)

Column Oriented Storage

Columns laid contiguously

2-10x compression typical

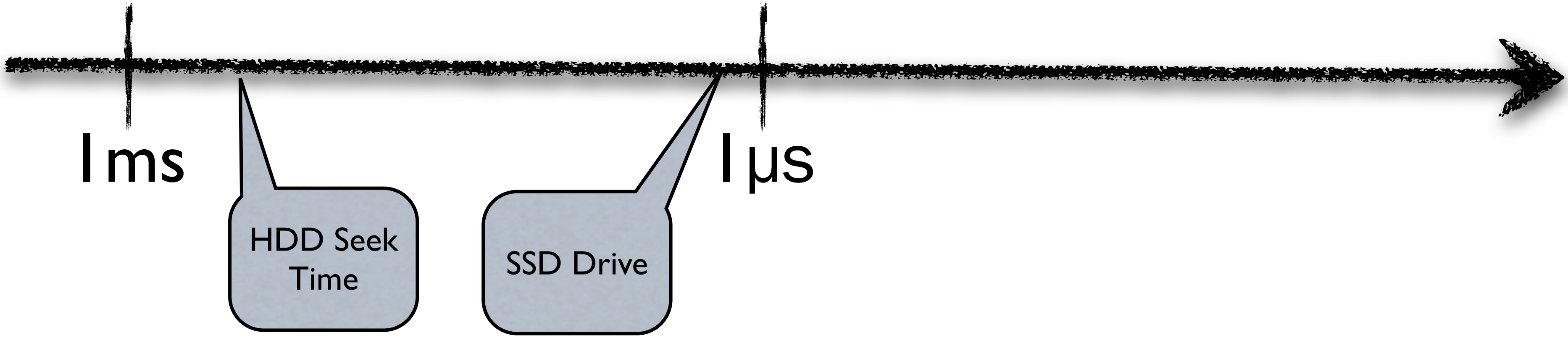
Indexing becomes less important.

Pinpoint I/O slow (tuple construction)

Bulk read/write faster

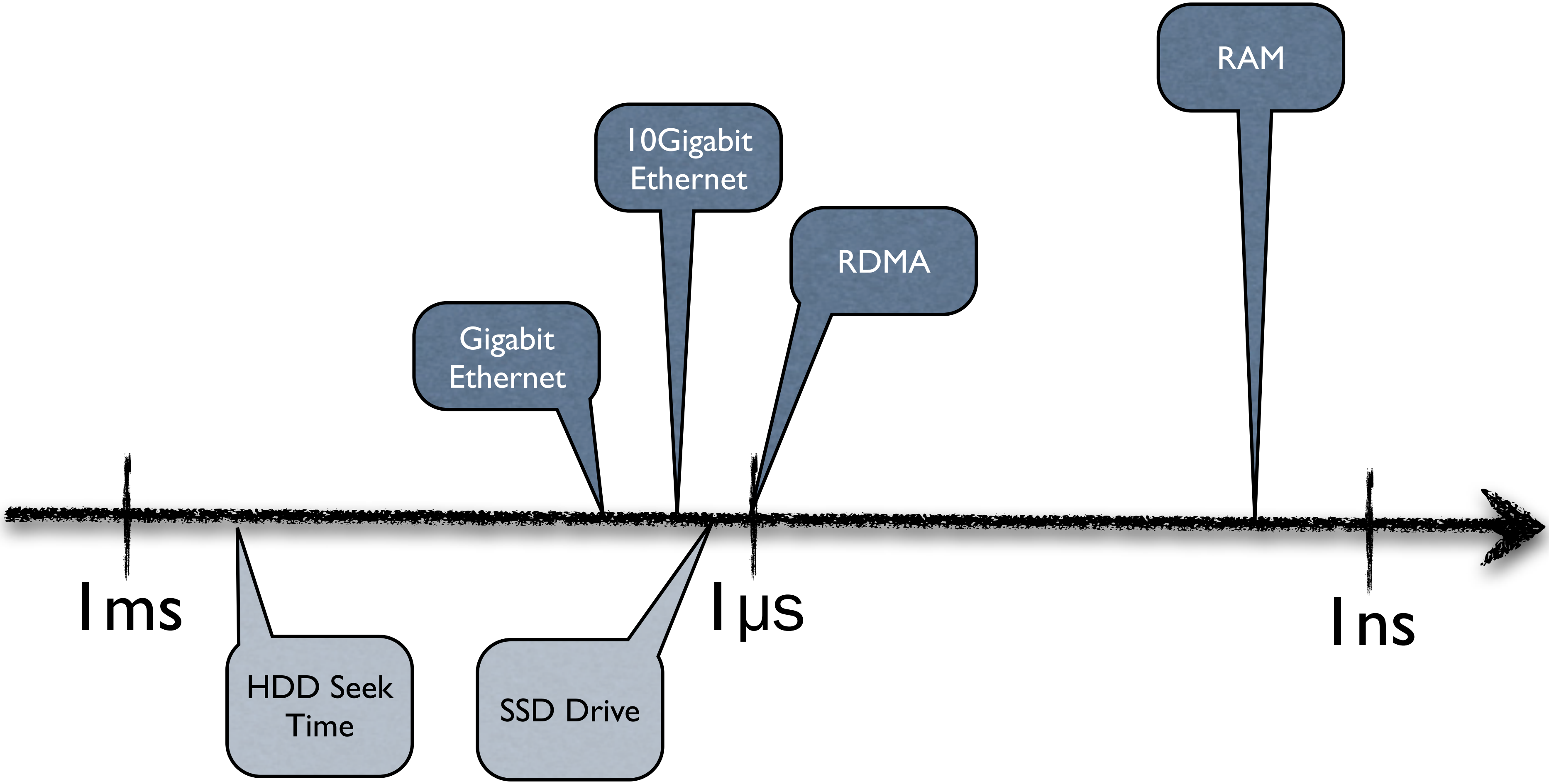
Compression >> row-based alternatives

Solid State Drives



- Traditional databases are designed for sequential access over magnetic drives, not random access over SSD.
- Weakens the columnar/row argument

Faster Networking



The best technologies of the moment are leveraging many of these factors

There is a new and impressive breed

- Products < 5 years old
- Shared nothing with SSD's over shards
- Large address spaces (256GB+)
- No indexes (column oriented)
- No referential integrity
- Surprisingly quick for big queries when compared with incumbent technologies.

TPC-H Benchmarks

Several new contenders with good scores:

- Exasol
- ParAccel
- Vectorwise

TPC-H Benchmarks

- Exasol has 100GB -> 10TB benchmarks
- Up to 20x faster than nearest rivals

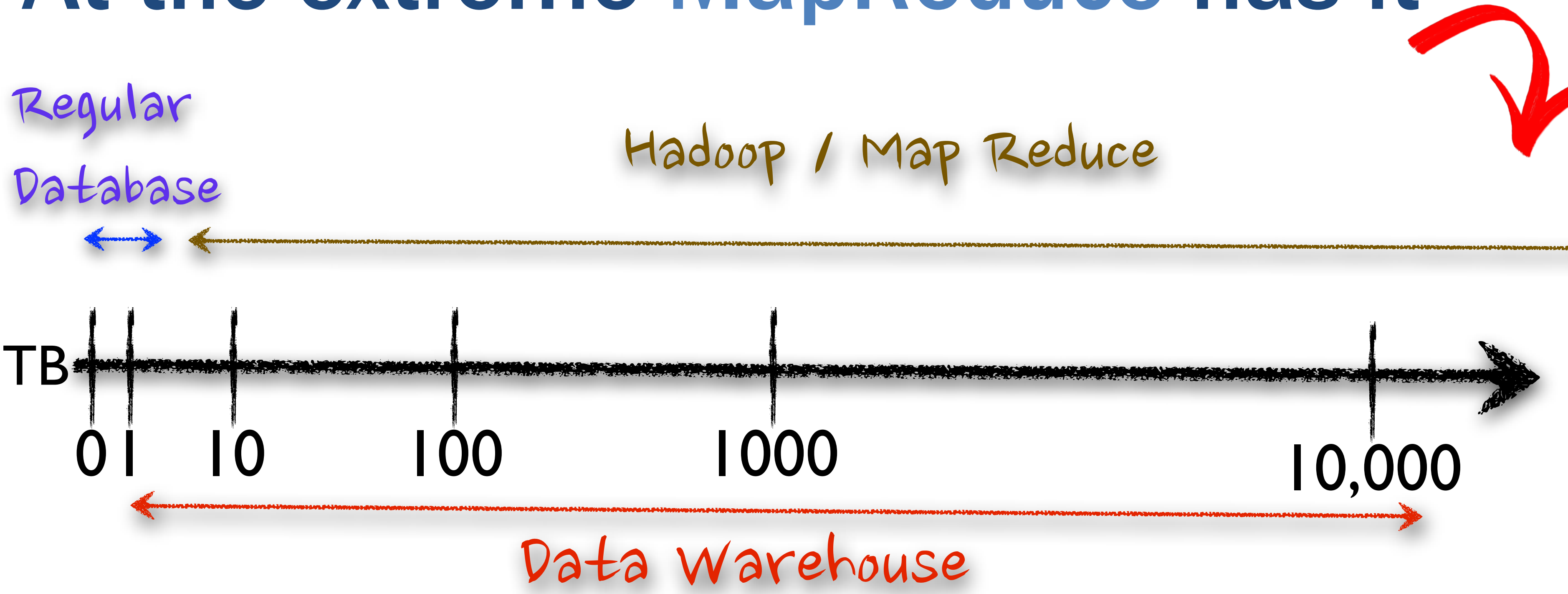
(But take benchmarks with a pinch of salt)

Relational Approach

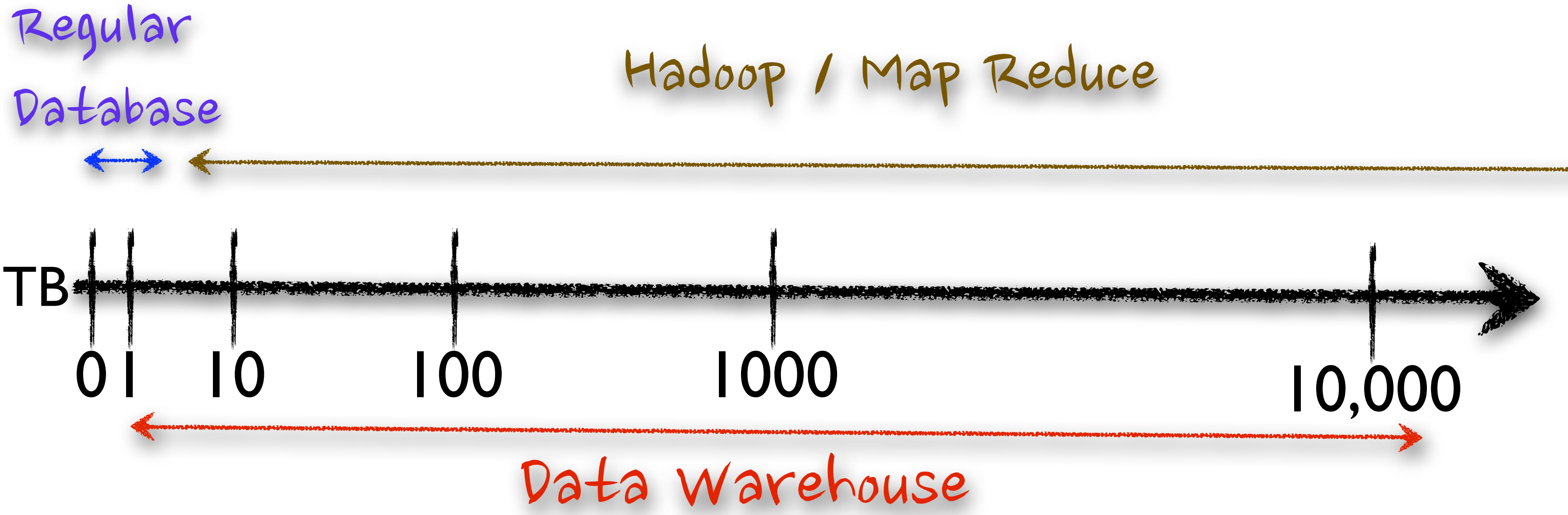
Solid data from every angle, bounded in terms of scale, but with a boundary that is rapidly expanding.

Comparisons

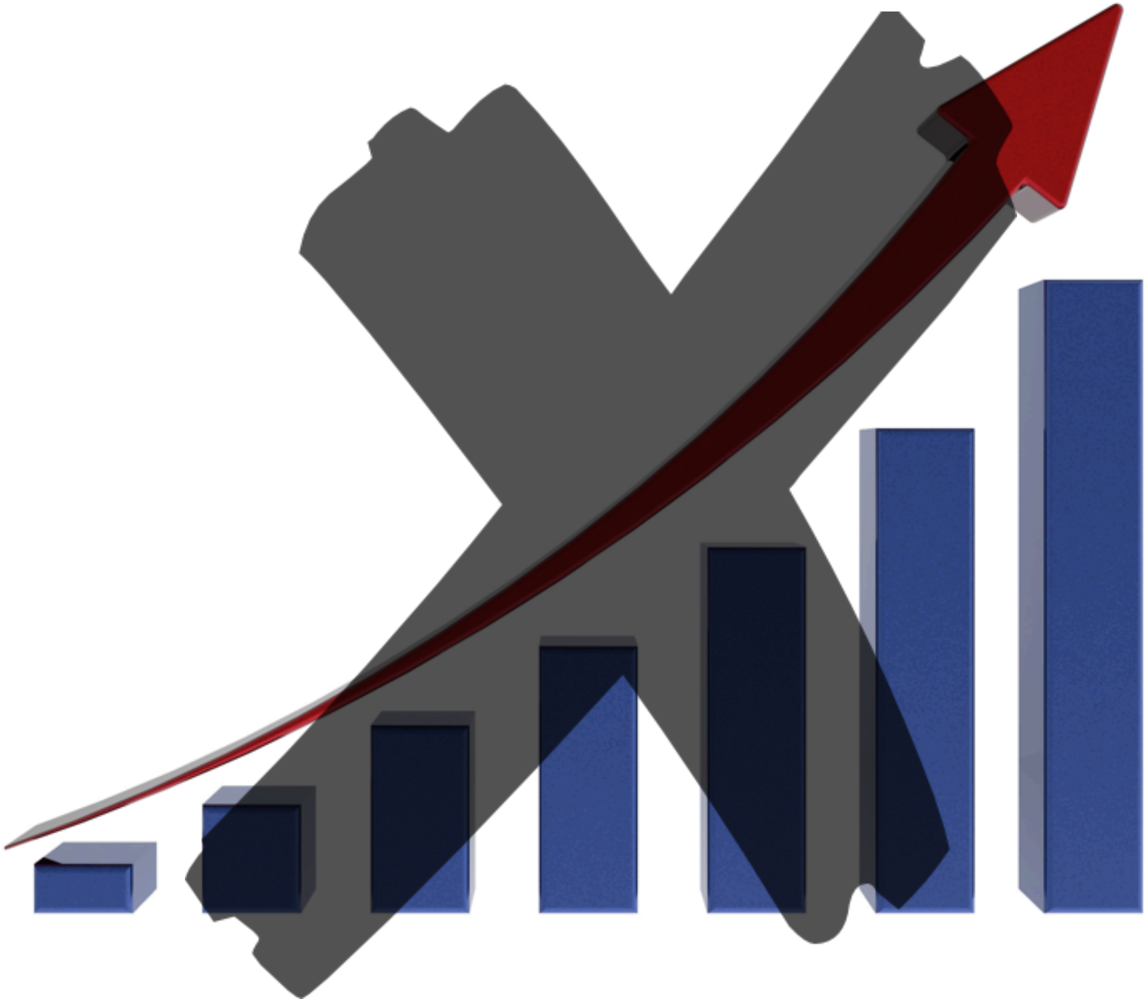
At the extreme MapReduce has it



But there is massive overlap



It’s not just data volume/velocity



The Dimensions of Data

- Volume (pure physical size)
- Velocity (rate of change)
- Variety (number of different types of data, formats and sources)
- Static & Dynamic Complexity

Consider the characteristics of data to be integrated, and how that equates to cost

Ability to model data is much more of a gating factor than raw size, particularly when considering new forms of data

Dave Campbell
(Microsoft – VLDB Keynote)

It becomes about your data and you want to do with it

Do you need to more than just SQL to process your data?

Does your data change rapidly?

Are you ok with some degree of eventual consistency?

Do isolation and consistency matter

Do you need to answer questions absolutely or within a tolerance?

Do you want to keep your data in its natural form?

Do you prefer to work bottom up or top down?

How risk averse are you?

Are you willing to pay big vendor prices?

Composite Offerings

Hadoop has Pig & Hbase

Mongo offers Query Language, atomaticity & MR

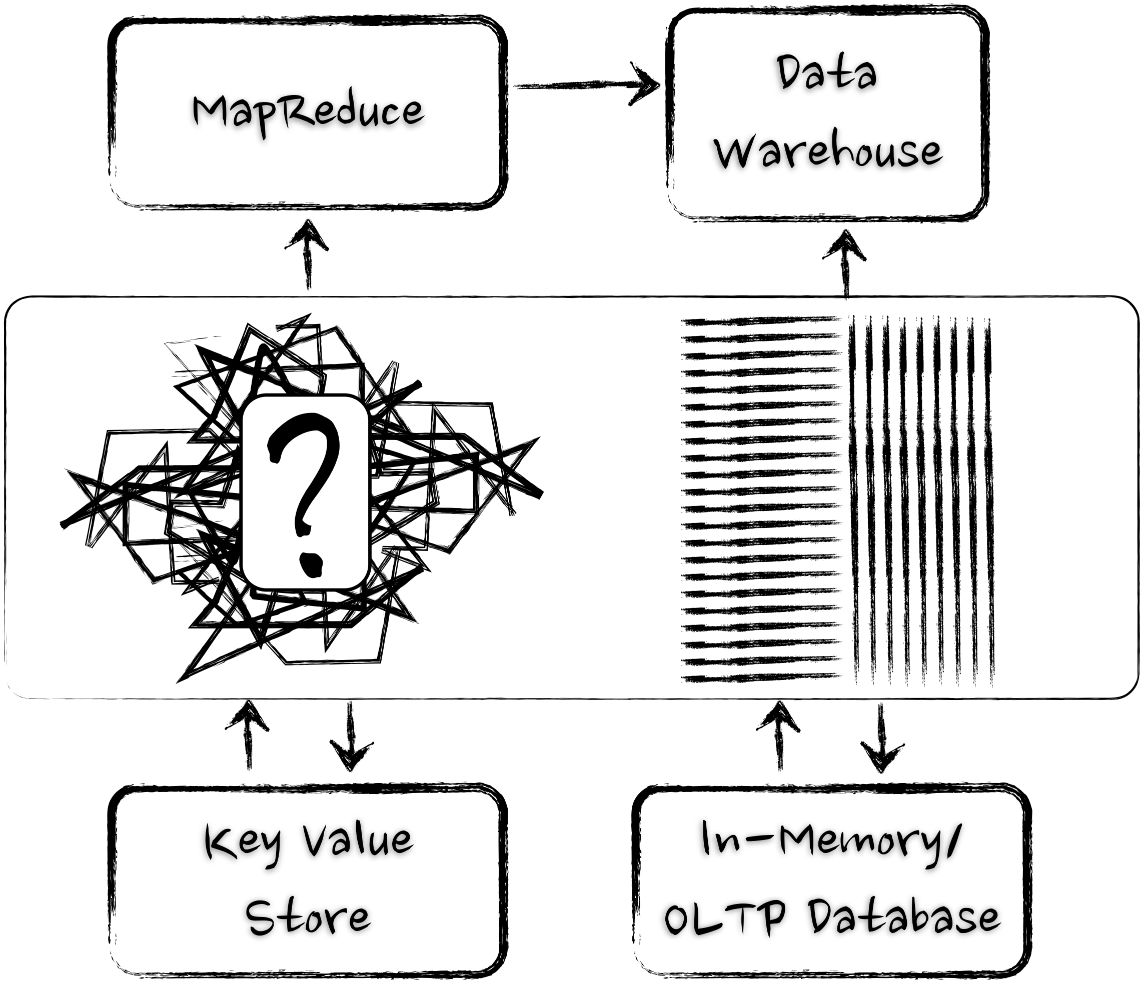
Oracle have BigData appliance with Cloudera

IBM have a Map Reduce offering

Sybase (now part of SAP) provides MR natively

EMC acquired Greenplum which has MR support

Complementary Solutions



Relational world has focused on keeping data consistent and well structured so it can be sliced and diced at will

Big data technologies focus on executing code next to data, where that data is held in a more natural form.

So

- NoSQL has disrupted the database market, questioning the need for constraint and highlighting the power of simple solutions.
- DB startups are providing some surprisingly fast solutions that drop some traditional database tenets and cleverly leverage new hardware advances.
- Your problem (and budget) is likely a better guide than the size of the data
- The market is converging on both sides towards a middle ground and integrated suites of complementary tools.

The right tool for the job

“Attempting to force one technology or tool to satisfy a particular need for which another tool is more effective and efficient is like attempting to drive a screw into a wall with a hammer when a screwdriver is at hand: the screw may eventually enter the wall but at what cost?”

E.F. Codd, 1993

Thanks

RBS
The Royal Bank of Scotland

http://www.benstopford.com