Big Data Analysis: Trends & Challenges

Big Data Principles, Architectures & Applications (BDAA 2014)

HPCS 2014, Bologna, 22 July 2014

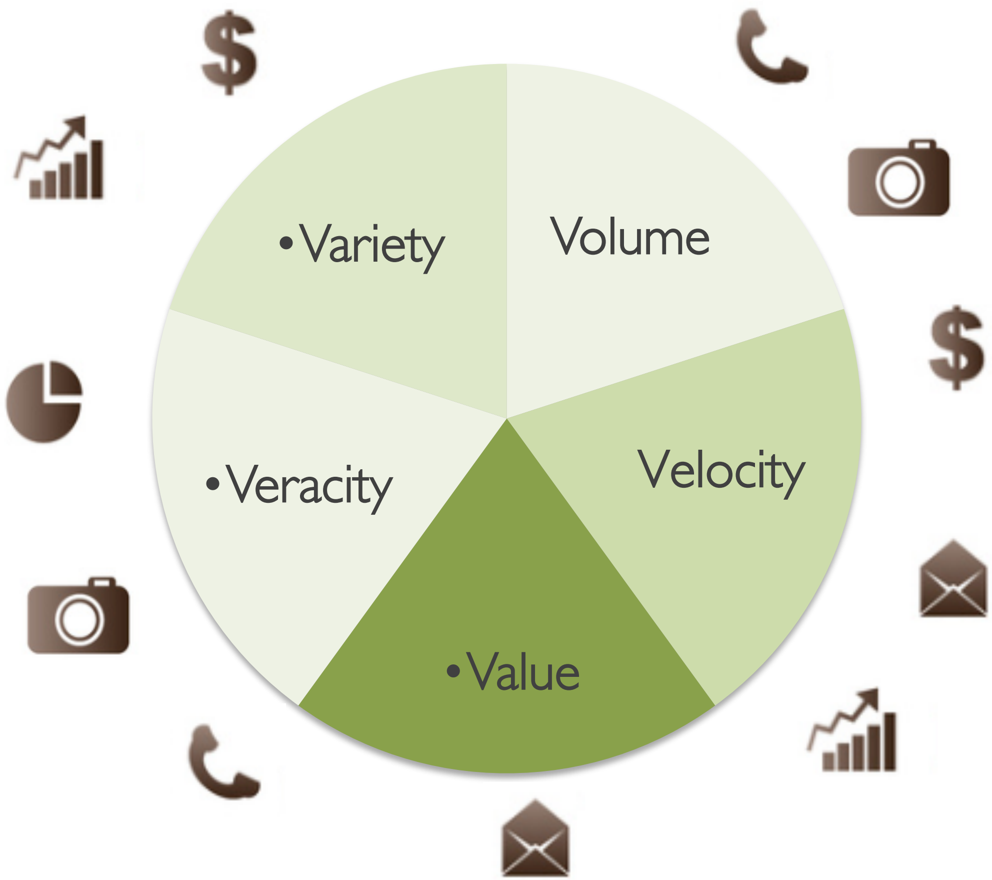
Prof. Sonia Bergamaschi

Department of Engineering "Enzo Ferrari"

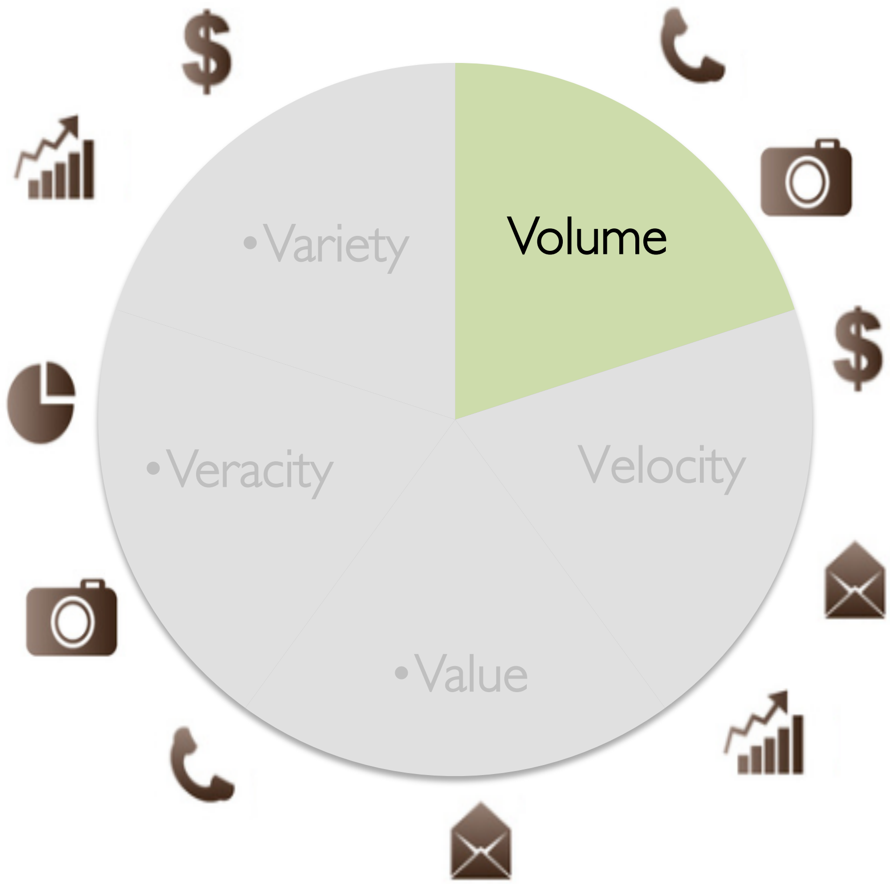
Università di Modena e Reggio Emilia

DB Group UNIMORE

Big Data is often described using Five Vs



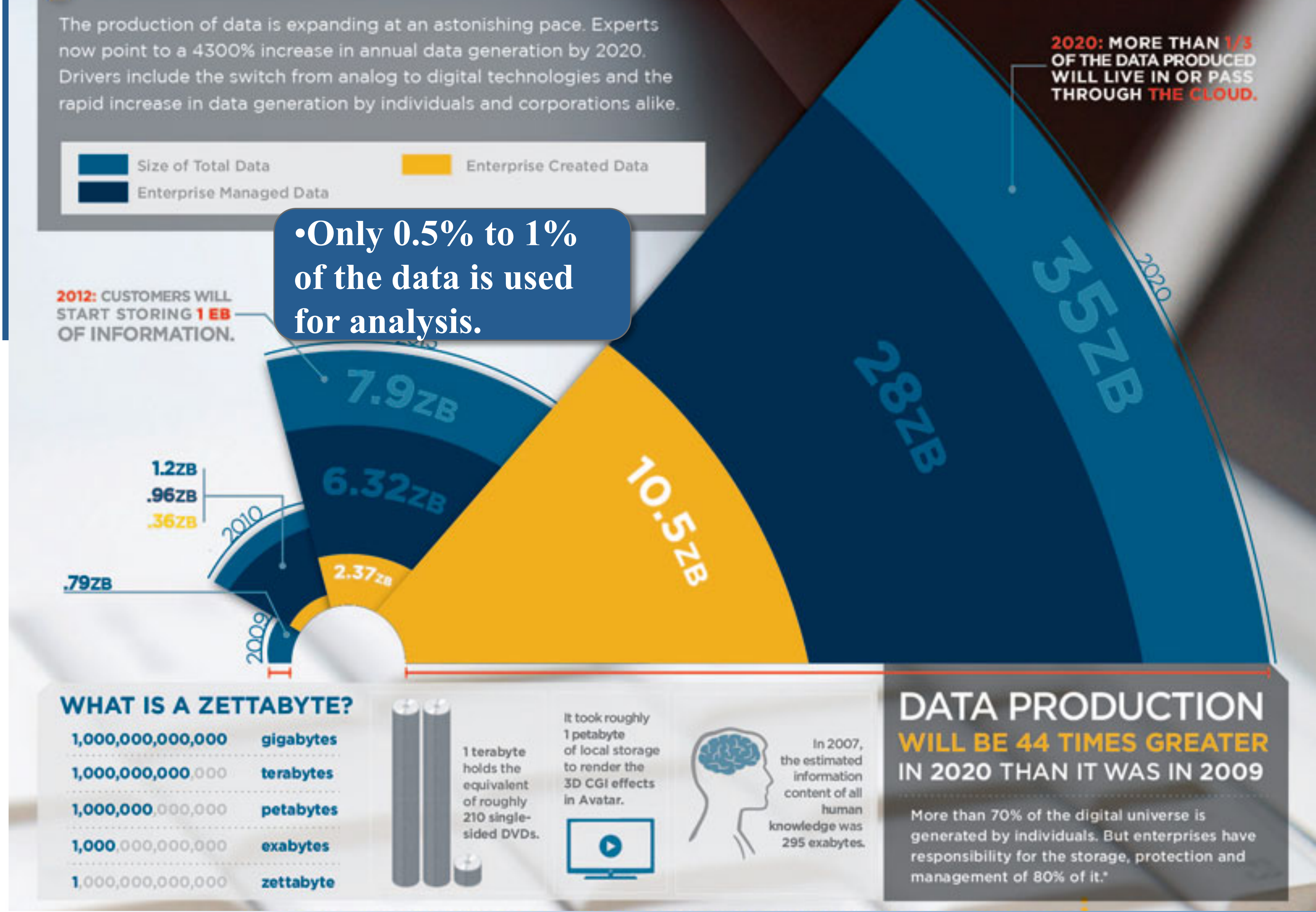
Increasing volumes of data, that grow at exponential rates



The increase in data volume is due to many factors:

- transaction based data stored through the years
- text data constantly streaming in from social media
- increasing amounts of sensor data being collected, etc.

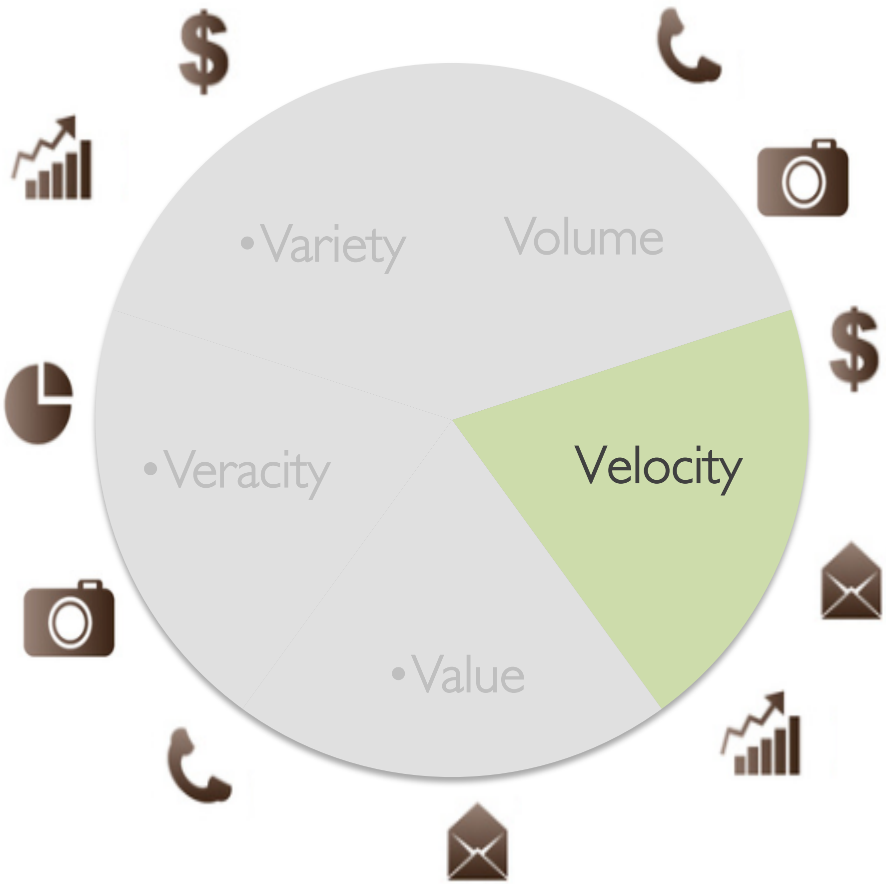
In the past, excessive data volume created a storage issue, but with today's decreasing storage costs, other issues emerge, including how to determine *relevance* amidst the large volumes of data and how to create *value* from data that is relevant



- Only 0.5% to 1% of the data is used for analysis.

http://www.csc.com/insights/flxwd/78931-big_data_growth_just_beginning_to_explode
http://www.guardian.co.uk/news/datablog/2012/dec/19/big-data-study-digital-universe-global-volume

Increasing velocity at which data changes, travels or increases



- According to Gartner, velocity means both:
- how fast data is being produced
- how fast the data must be processed to meet demand

Reacting quickly enough to deal with velocity is a challenge to most organizations

Velocity

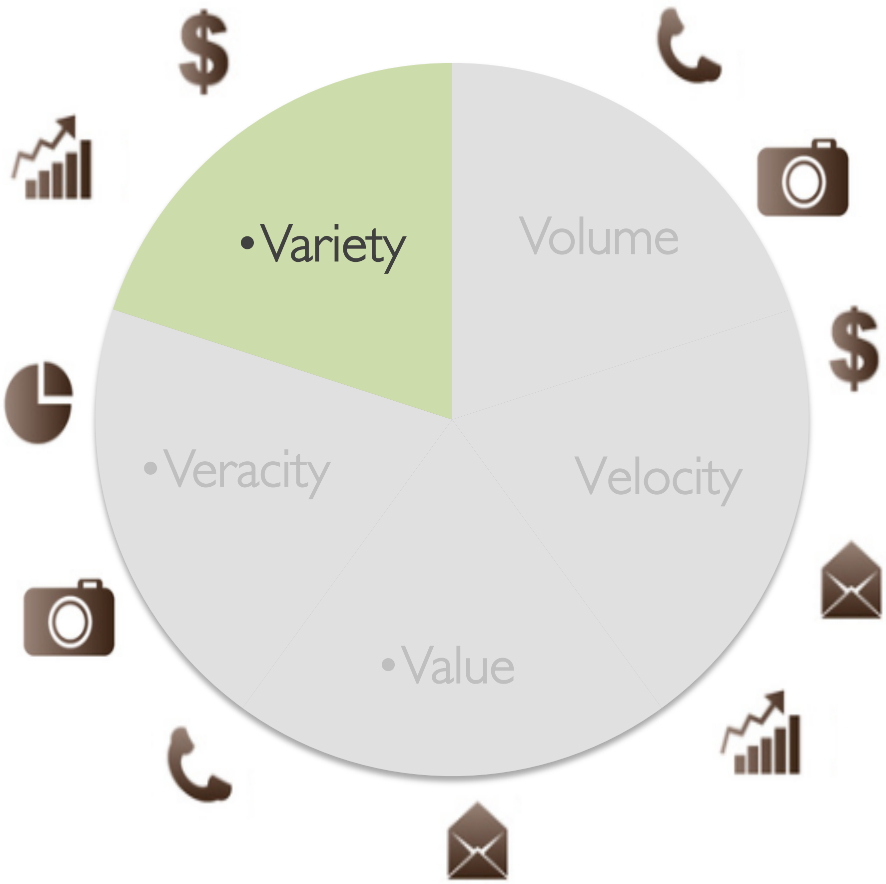
Fast Data

Rapid Changes

Real-Time/Stream Analysis

Current application examples: financial services, stock brokerage, weather tracking, movies/entertainment and online retail

Increasing Variety of data types

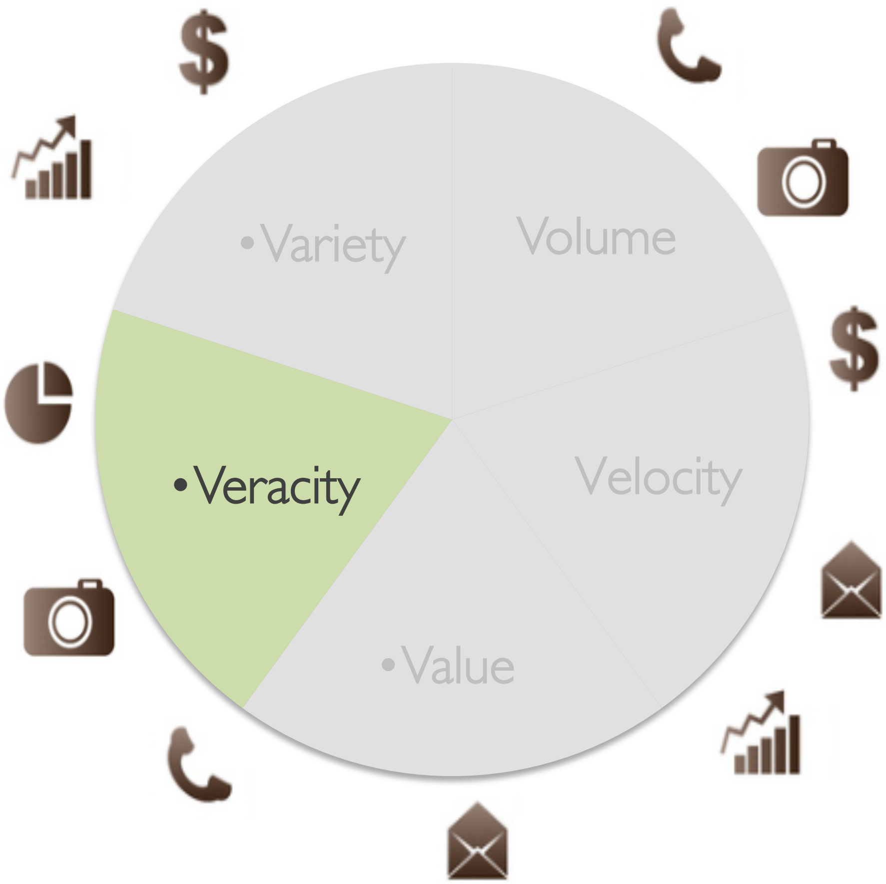


Data today comes in all types of formats:

•from traditional databases to RDF data stores created by end users and OLAP systems

•to text documents, email, meter-collected data, video, audio, stock ticker data and financial transactions.

We see increasing veracity (or accuracy) of data

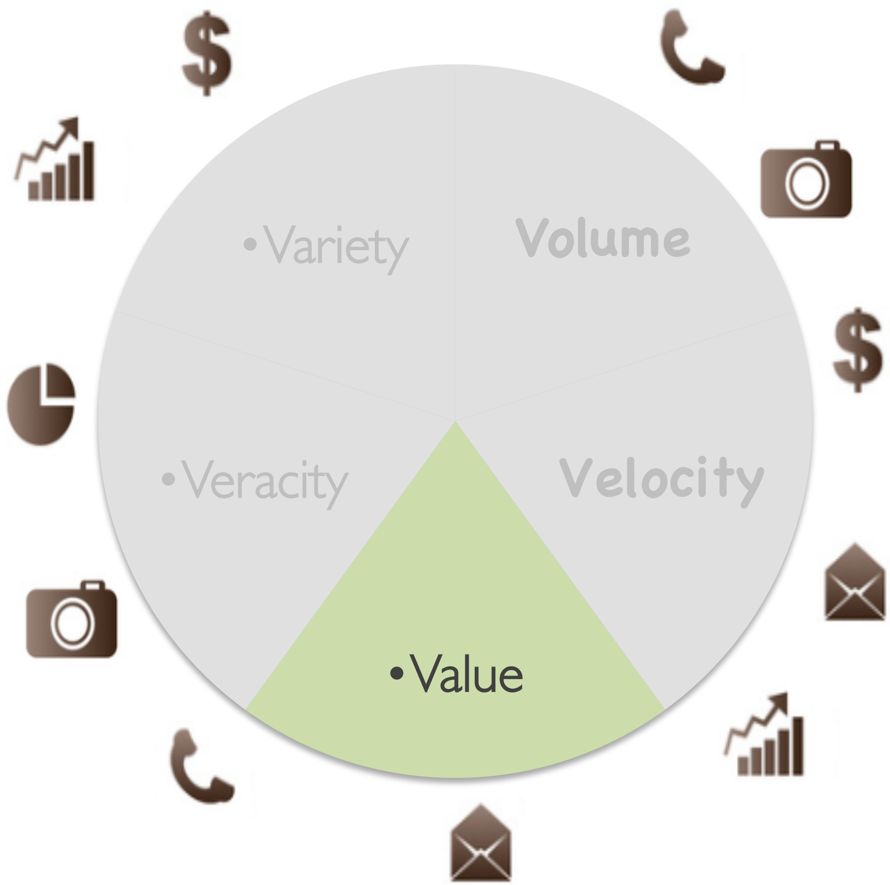


Refers to the ***messiness*** or ***trustworthiness*** of the data. With many forms of big data *quality* and *accuracy* are *less controllable*

(just think of Twitter posts with hash tags, abbreviations, typos and colloquial speech as well as the reliability and accuracy of content)

but technology now allows us to work with this type of data.

Value – The most important V of all!



- Then there is another V to take into account when looking at Big *Data: Value!*
- Having access to big data is no good unless we can turn it into value.
- Companies are starting to generate amazing value from their big data.

Questions typically asked on Big Data

- What if your data volume gets so large and varied you don't know how to deal with it?
- Do you store all your data?
- Do you analyze it all?
- What is coverage, skew, quality?
- How can you find out which data points are really important?
- How can you use it to your best advantage?

[4]

Current Focus on Big Data

- Focus on verticals
  advertising, social media, retail, financial services, telecom and healthcare
  - Aggregate data, focused on transactions, limited integration (limited complexity), analytics to find (simple) patterns
  - Emphasis on technologies to handle volume/scale, and to lesser extent velocity: Hadoop, NoSQL, MPP (Massive Parallel Processing) for data warehouse: DWA (Data Warehousing Appliance), ….
  - Full faith in the power of data (no hypothesis), bottom up analysis

Full faith in the power of data

The quest for knowledge used to begin with grand theories.

Now it begins with massive amounts of data. **Welcome to the Petabyte Age**!

Technologies for Big Data

- Managing Big Data
- Analyzing Big Data

Technologies for Big Data

- Managing Big Data
- Analyzing Big Data

Database Philosophy

God made integers,

all else is the work of man.

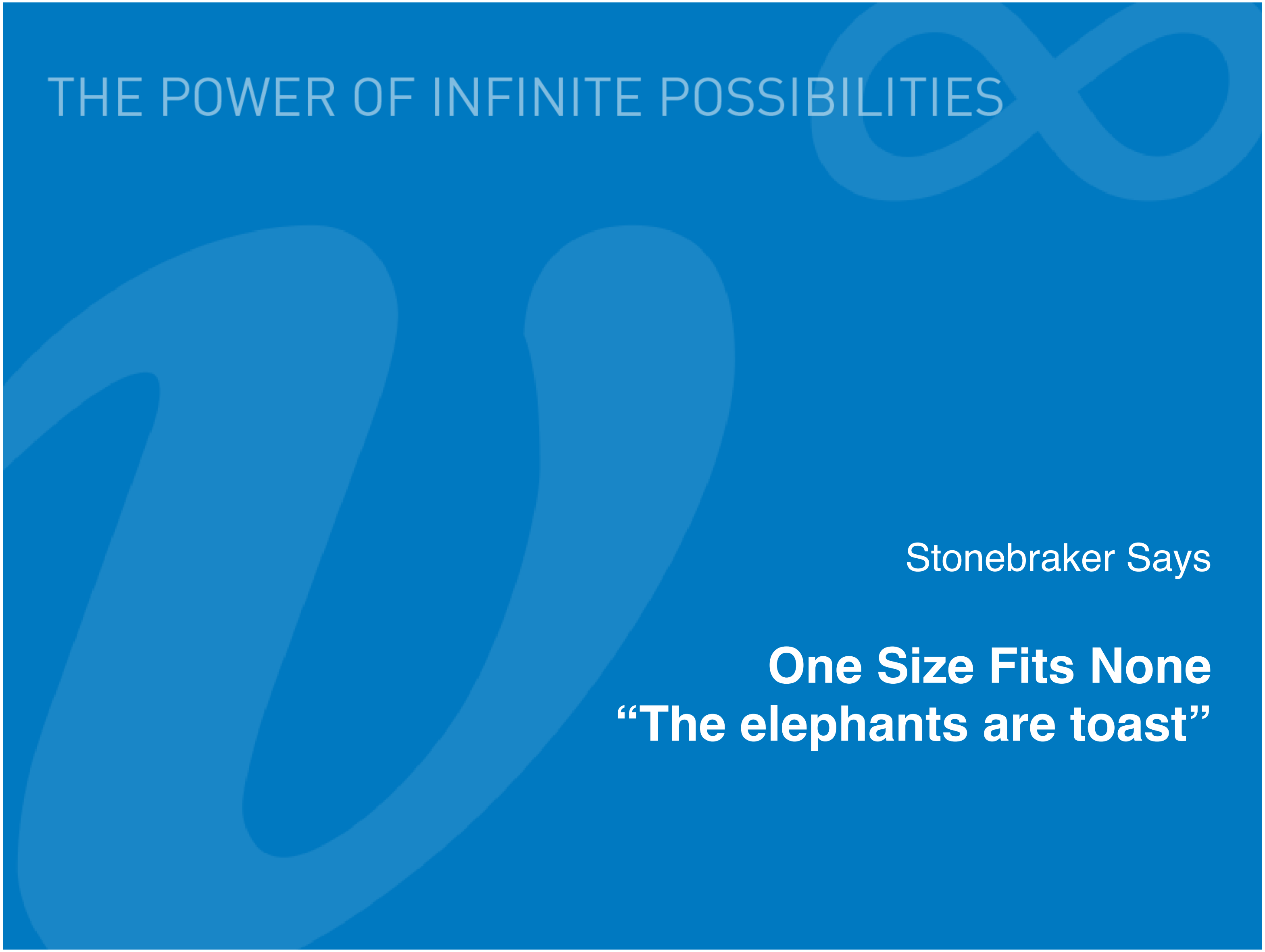
(Leopold Kronecker, 19th Century Mathematician)

Codd made relations,

all else is the work of man.

(Raghu Ramakrishan, DB text book author)

Traditional RDBMS



[7] [Stonebraker05]

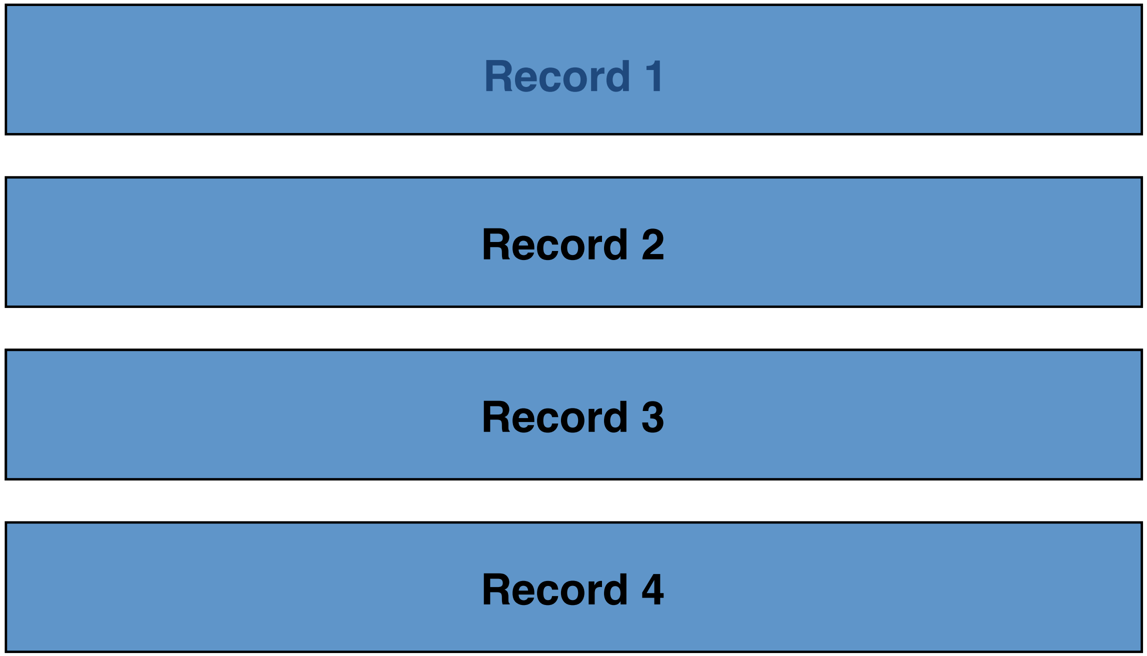
Traditional RDBMS: The Elephants

- Sell code lines that date from the 1970's
  - Legacy code
  - Built for very different hardware configurations
  - And some cannot adapt to grids….
- That was designed for business data processing (OLTP)
  - Only market back then
  - Now warehouses, science, real time, embedded, ..

Current DBMS Gold Standard

- Store fields in one record contiguously on disk
- Use B-tree indexing
- Use small (e.g., 4K) disk blocks; heavily encoded
- Align fields on byte or word boundaries
- Conventional (row-oriented) query optimizer and executor
- Write-ahead log
- Row-level dynamic locking

Terminology -- “Row Store”



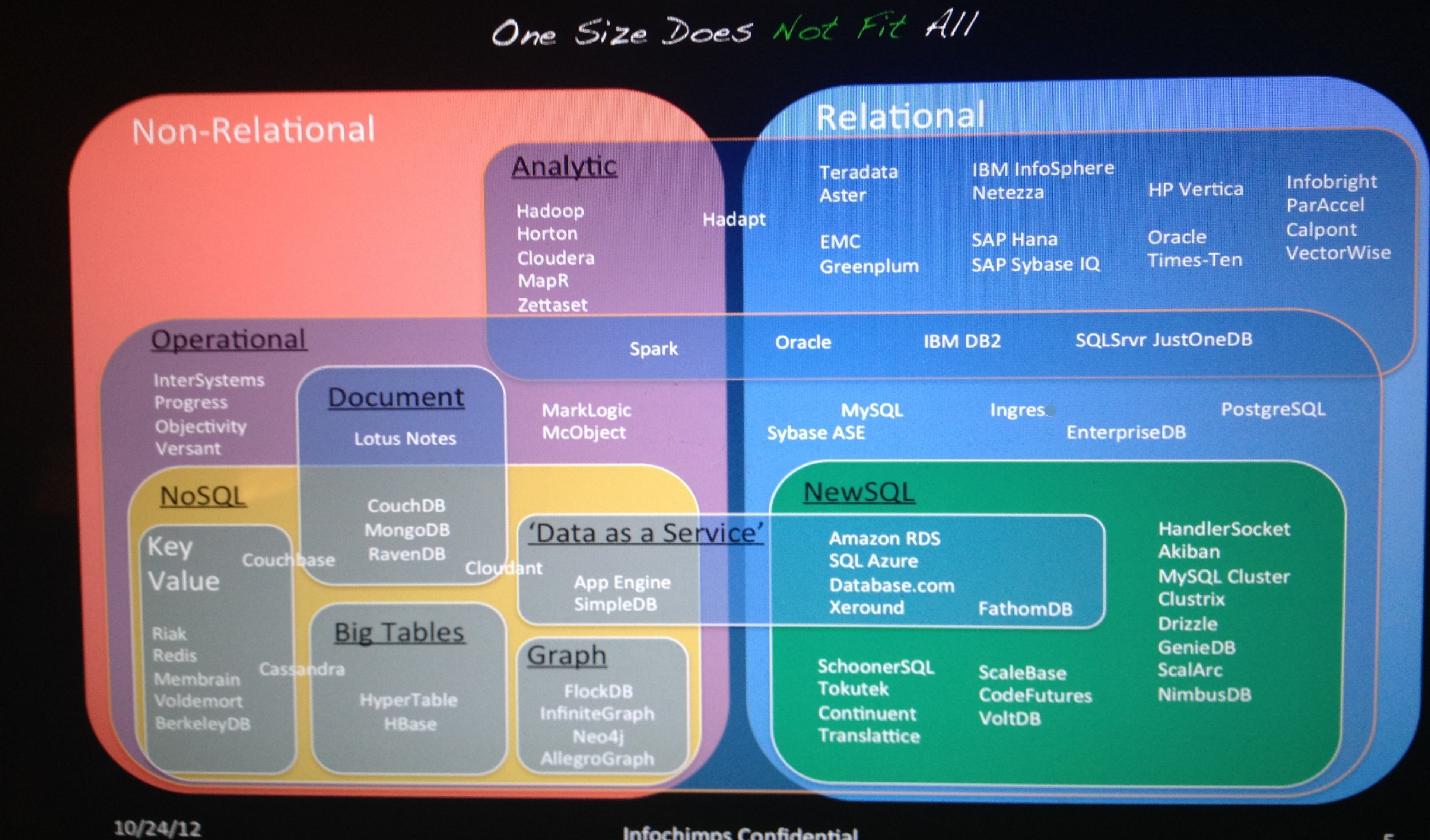
E.g. DB2, Oracle, Sybase, SQLServer, Postgres, MySQL, Netezza, Teradata,…

At This Point, RDBMS is “long in the tooth”

There are at least 6 (non trivial) markets where a row store can be clobbered by a specialized architecture

- Warehouse (Vertica, Red Shift, Sybase IQ, DW Appliances)
- OLTP (VoltDB, HANA, Hekaton)
- RDF (Vertica, et. al.)
- Text (Google, Yahoo, …)
- Scientific data (R, MatLab, SciDB)
- Data Streaming (Storm, Spark Streaming, InfoSphere)

Variety of Data Analytics Enablers

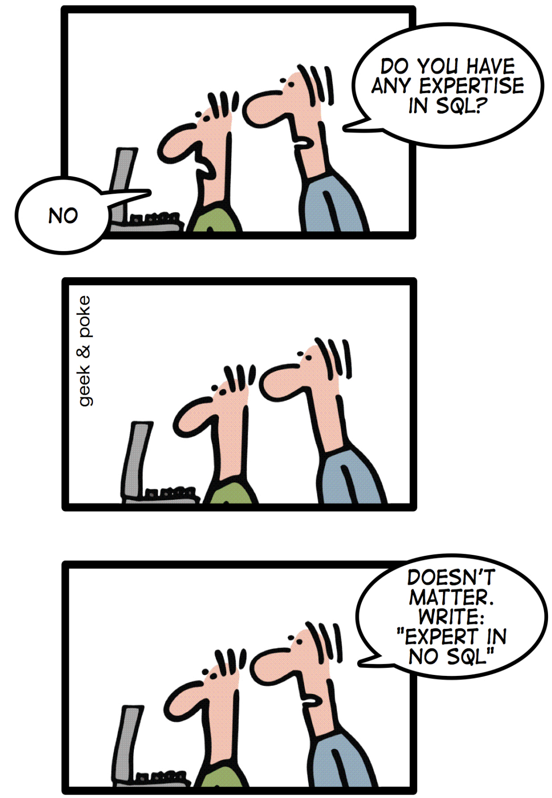


http://techcrunch.com/2012/10/27/big-data-right-now-five-trendy-open-source-technologies/

An emerging "movement" around non-relational software for Big Data

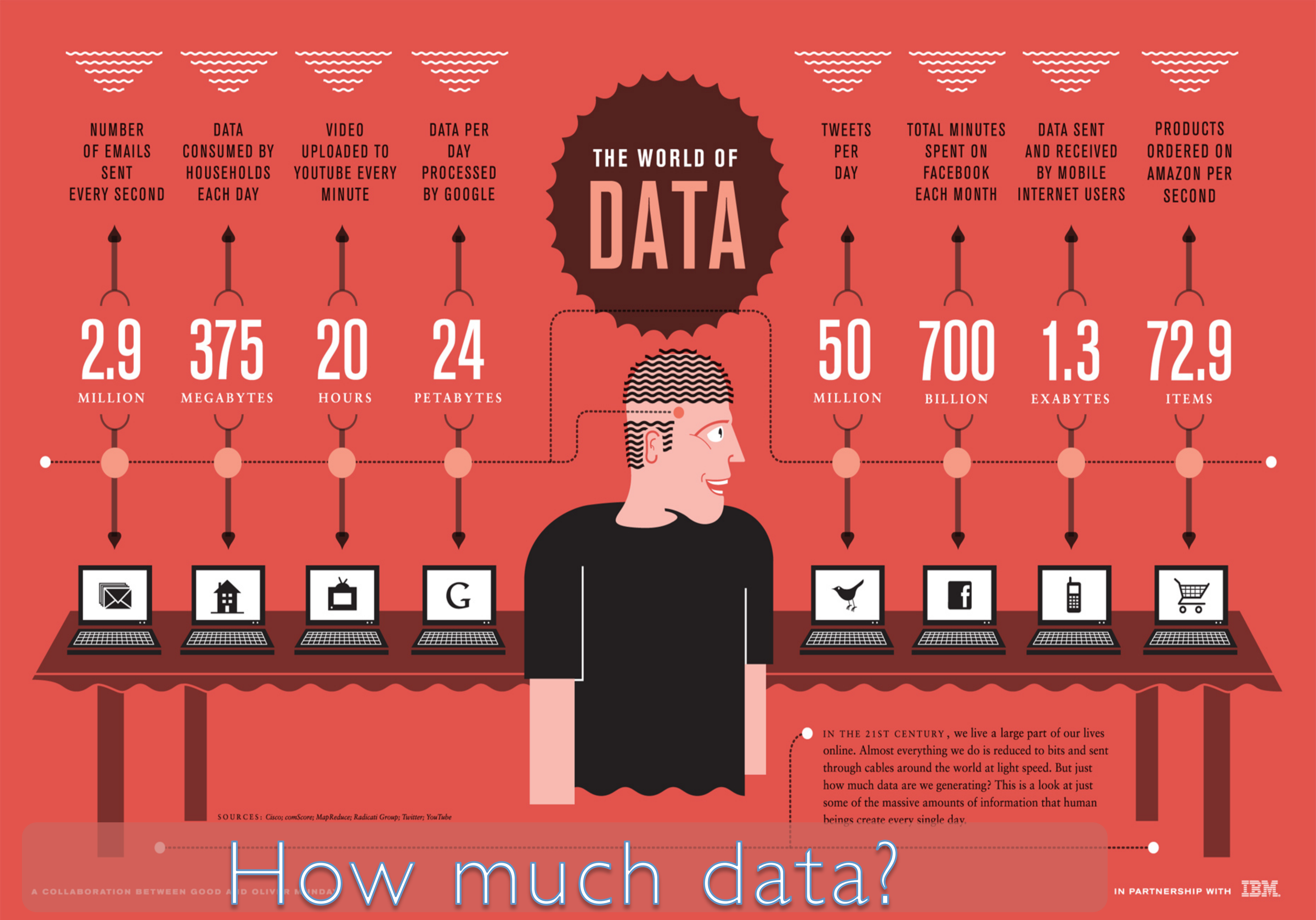
- NOSQL stands for "Not Only SQL" where SQL doesn't really mean the query language, but instead it denotes relational DBMS.
- Google, Facebook, Linkedin, eBay, Amazon, etc. did not use 'traditional' RDBMS for Big Data. They need:
  - To perform a massive number of Simple Operations very quickly
- They inspired many NOSQL systems:
  - Memcached demonstrated that in-memory indexes can be highly scalable, distributing and replicating objects over multiple nodes
  - Dynamo (Amazon) pioneered the idea of *eventual consistency* as a way to achieve higher availability and scalability [10]
  - BigTable, HDFS (Google), demonstrated that persistent record storage could be scaled to thousands of nodes [9]
  - Map-Reduce (Google) paradigm for parallel processing

HOW TO WRITE A CV



Leverage the NoSQL boom

How much data?



http://www.knowledgeinfusion.com/blog/2011/11/get-your-head-out-of-the-clouds-and-into-big-data/

Why NOSQL Big-Data Technology?

- Challenges of Traditional Data Warehouse Technology:
  - ➢ *Could not scale*
  - ➢ *Not suited for compute-intensive deep analytics*
  - ➢ *Price-performance challenge*
  - – *(or why did Google, Yahoo! and Facebook need to invent a new stack?)*
    1. Fault-tolerance at scale
    2. Variety of data types
    3. Manage data volumes without archiving
    4. Parallelism was an add-on
- Main innovations: Map-Reduce on Distributed File-Systems
- Main message: *different approaches to data-processing*
- Caveats:
  - – Many databases innovations remain unique to the traditional stack
    - *Variety of indexing, complex query optimization, storage optimization*
  - – All of these are being re-discovered and re-invented for big-data

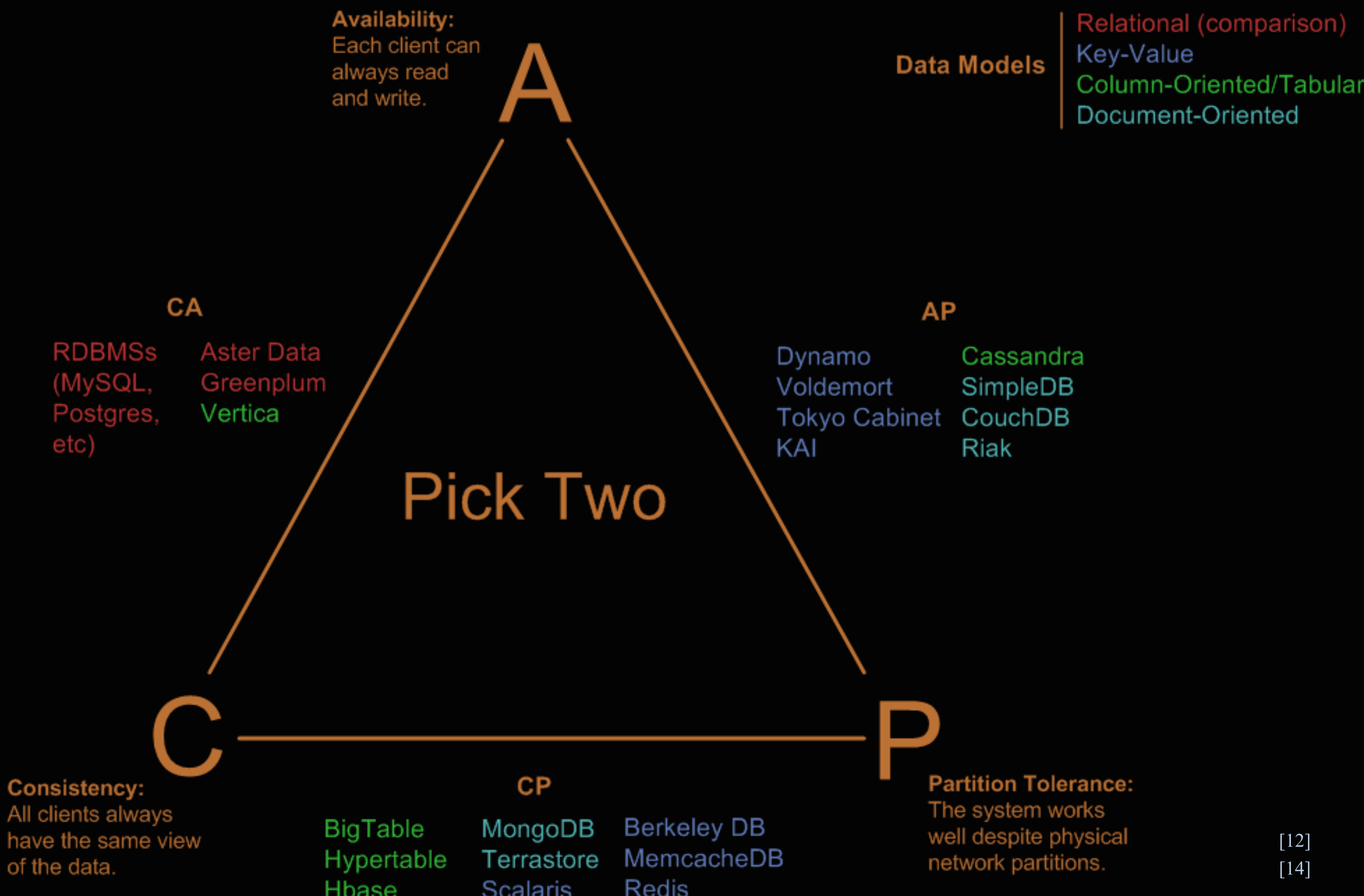
NOSQL Solutions

- High scalability for simple operations (SO) on multiple nodes
  - key lookups (reads and writes of one/ small number of records). Fine for Web 2.0 sites where millions of users may both read and write data
  - This is in contrast to complex queries or joins, read-mostly access, or other application loads
  - SO are highly parallelizable.
- Shared-nothing architectures, Data partitioning (sharding) and replication on multiple nodes
  scale until the network bandwidth is exhausted if: data objects are
    - partitioned and distributed across the nodes in the system in a manner that balances the load
    - The application is able to make the majority of transactions "single-shared" (local to one node)
  - Adopted by NOSQL systems, but also DW and DB systems
  - Relaxed consistency therefore higher performance and availability
- Flexibility on the data structure

But with some sacrifice:

- Interface far more easy then SQL.
  - more low level programming
- Transaction management less rigorous
  - Relaxed consistency
- Queries that span multiple shards are very inefficient or impossible

Visual Guide to NoSQL Systems



[12]

[14]

http://blog.nahurst.com/visual-guide-to-nosql-systems

Technologies for Big Data

- Managing Big Data
- Analyzing Big Data

Moving the computation near the data

- Moore's Law has held firm for over 40 years
  - processing power doubles every two years
  - Processing speed is no longer the problem

- Getting the data to the processor becomes the bottleneck

- Quick calculation:
  - Typical disk data transfer rate: 75MB/sec
  - Time taken to transfer "only" 100GB of data to the processor: ~ 22minutes !
    - Actual time will be worse, if servers have less than 100GB of RAM available

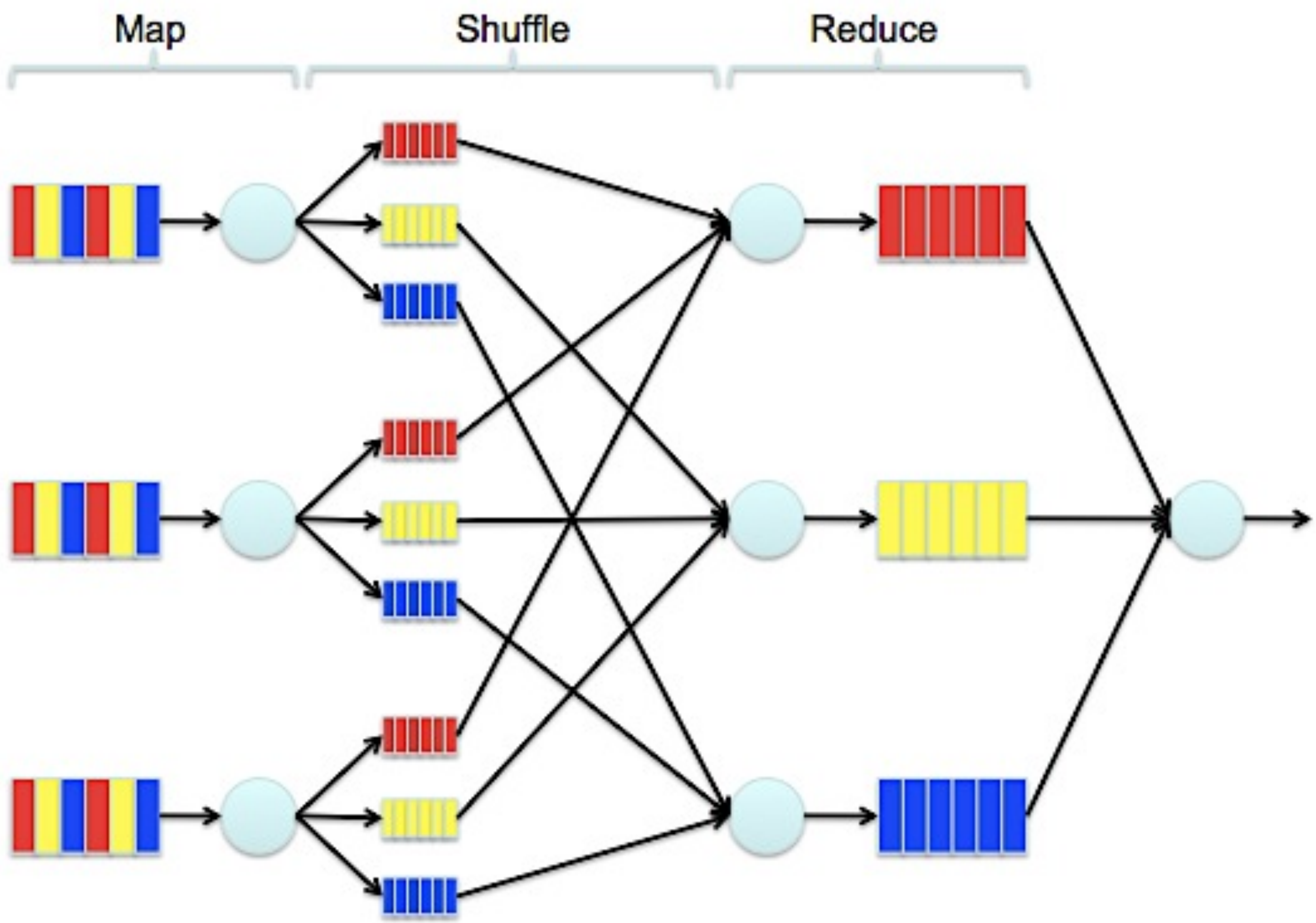
- MapReduce solution: move the computation near the data, instead of moving the data:
  - note that often the data transfer over the network is still the bottleneck!

MapReduce vs. RDBMS

- RDMBS is good when you have a Gigabytes of structured data, which read and write often and need high integrity.
- Hadoop is good when you have a Petabytes of semi-structured or unstructured (though fit for structure too)
- Hadoop is good for analyzing the whole dataset (batch query), whereas RDBMS is good for point queries or updates.

	Traditional RDBMS	MapReduce
Access	Interactive and batch	Batch
Updates	Read and write many times	Write once, read many times
Structure	Static schema	Dynamic schema
Scaling	Nonlinear	Linear

MapReduce

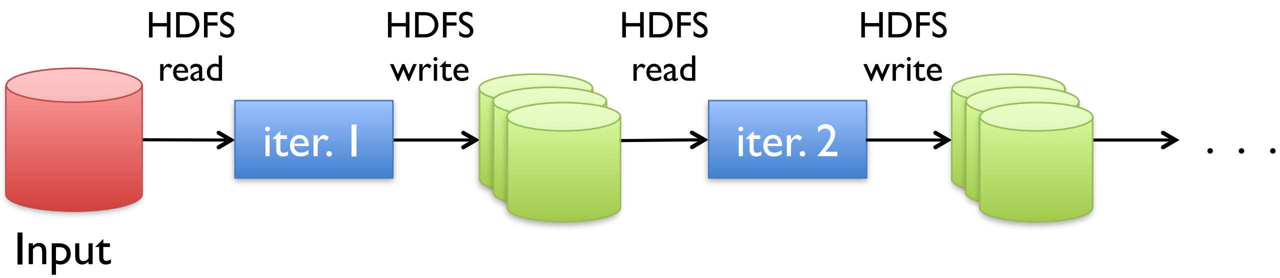


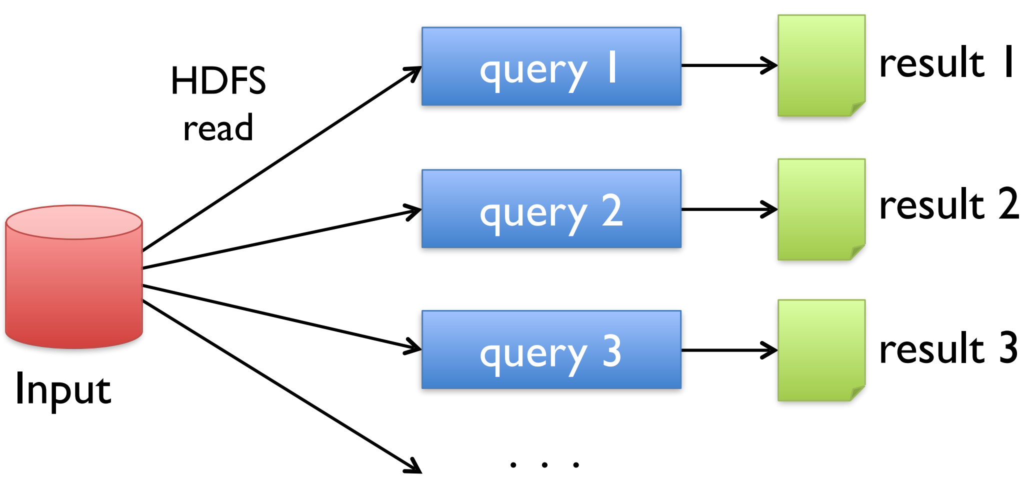
http://www.slideshare.net/lynnlangit/hadoop-mapreduce-fundamentals-21427224

MapReduce (Hadoop) Problems (1)

A typical MapReduce program consists of a chain (or dataflow) of MapReduce jobs:

- Complex, multi-stage applications (e.g. interactive graph algorithms and machine learning - logistic regression, k-means…)
- Files are stored in HDFS (Hadoop Distributed File System) at each iteration (access time too low)



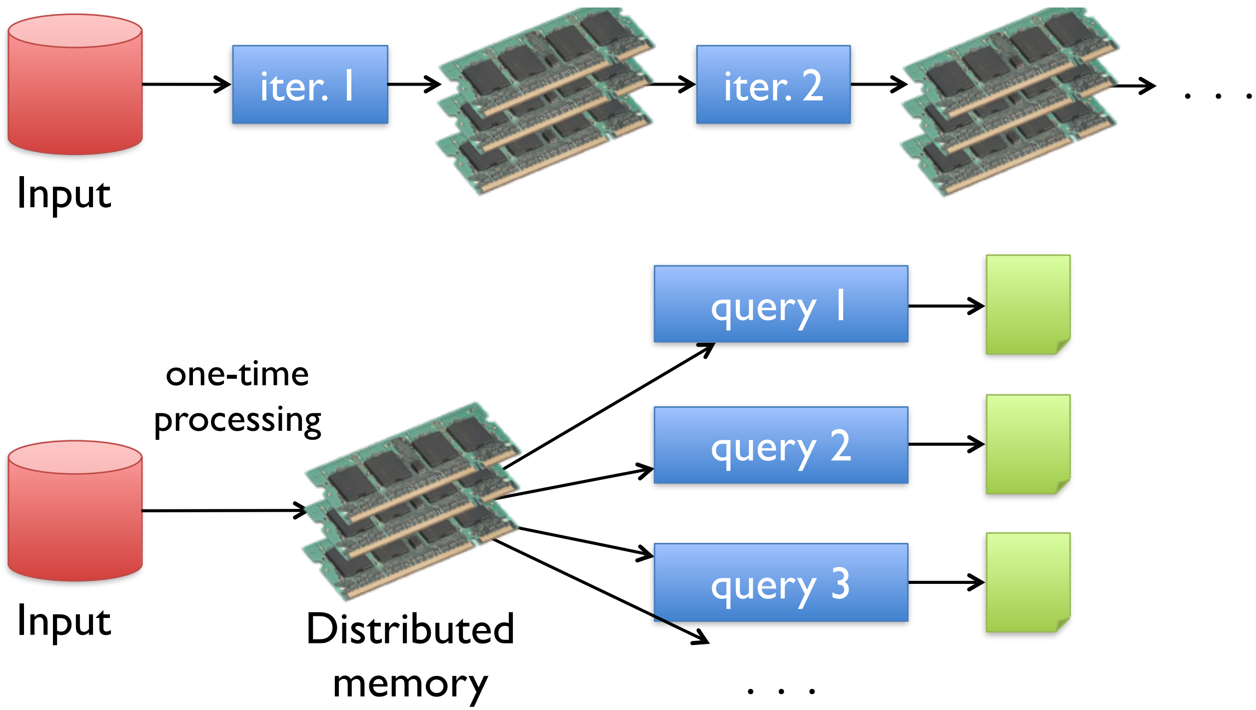


Images partially taken from Robert Metzger @ Big Data Beers Meetup, Nov. 19th, 2013

Solving MapReduce (Hadoop) Problems (1): a proposal

Solved in **Spark** [1] [3]@ amplab – UC BERKLEY

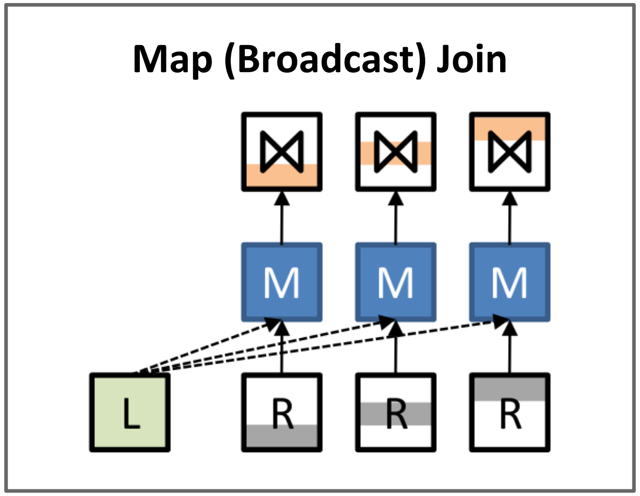
- cashing data in the main memory – in memory processing
- key idea: *Resilient Distributed Datasets (RDD)*
  - distributed collections of objects that can be cached in memory
  - manipulated through various parallel operations
  - automatically rebuilt on failure (RDD track their transformation – logs and checkpoints)

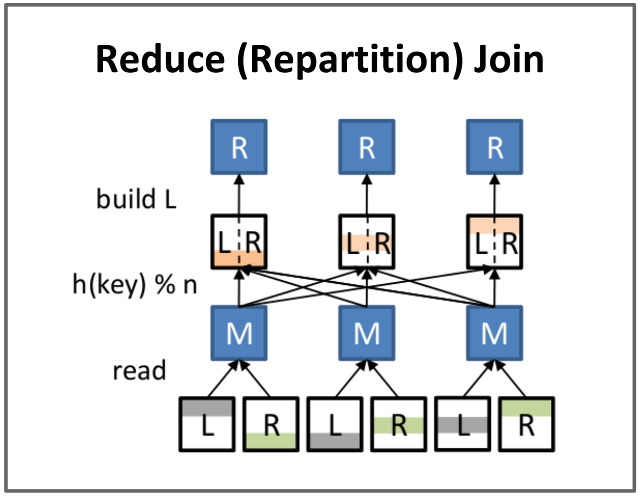


[1]

MapReduce Problems (2)

Join in Hadoop:
which strategy to choose? How to configure it?





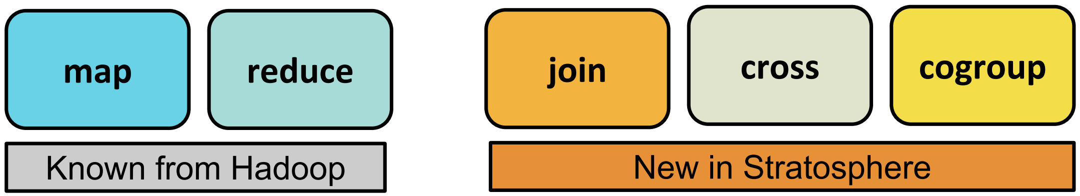
- Joins do not naturally fit MapReduce
- Very time consuming to implement
- Hand optimization necessary

Image from Robert Metzger's speech – "Stratosphere: System Overview" – Big Data Beers Meetup, Nov. 19th 2013

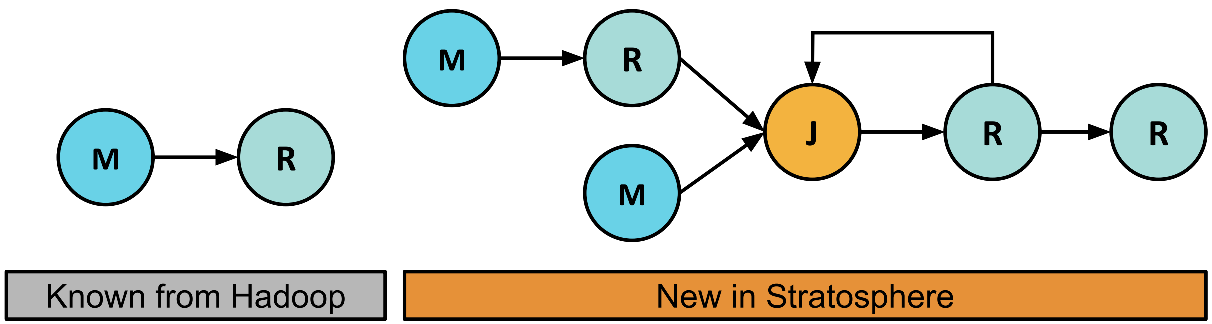
Solving MapReduce Problem (2): a proposal

Solved in **Stratosphere** [2] Project Leader: Prof. Volker Markl - TU Berlin

- Extends MapReduce with more operators



- Support for advanced data flow graphs



- Only write to disk if necessary, otherwise in-memory
- Natively implemented JOINS into the system
  - Optimizer decides join strategy (e.g. Hybrid Hash Join starts in-memory and gracefully degrade)

Image from Robert Metzger's speech – "Stratosphere: System Overview" – Big Data Beers Meetup, Nov. 19th 2013

BIG VENDORS: Oracle, IBM, Teradata, Sap, HP, Microsoft,…

Data Warehouse Appliance (DWA)

A new category of computer architecture for data warehousing (DW) specifically targeted for Big Data Analytics and Discovery that is:

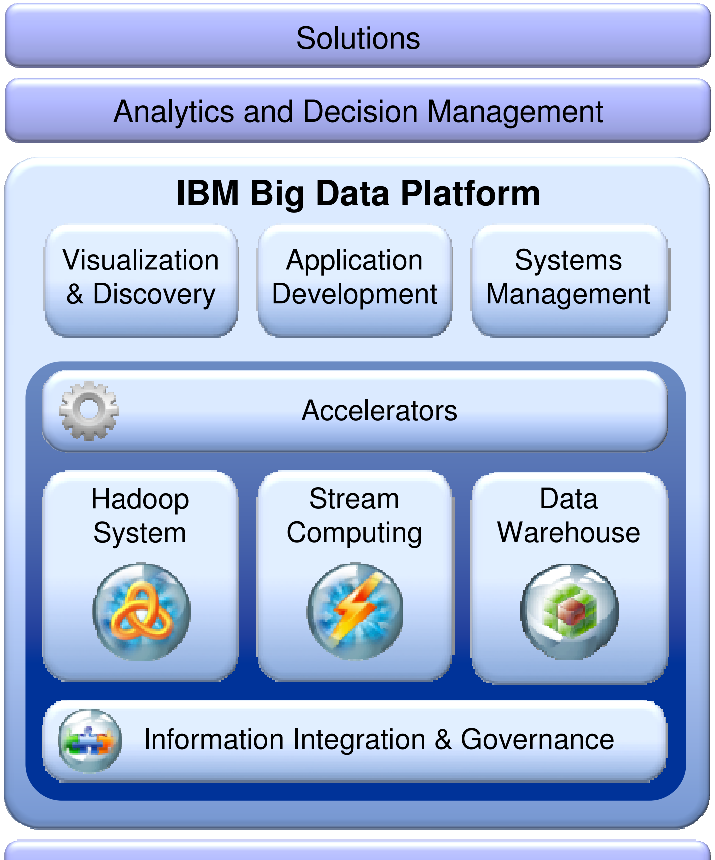
- simple to use (not a pre-configuration) and very high performance for this workload.
- A DWA includes an integrated set of servers, storage, operating system(s), and DBMS.
- New Database Solutions (based on: exploiting main memory, combined row and column databases, enforcing MPP)

IBM Big Data Infrastructure

•Appliance: Netezza

The IBM Big Data Platform

- Process any type of data
  - Structured, unstructured, in-motion, at-rest
- Built-for-purpose engines
  - Designed to handle different requirements

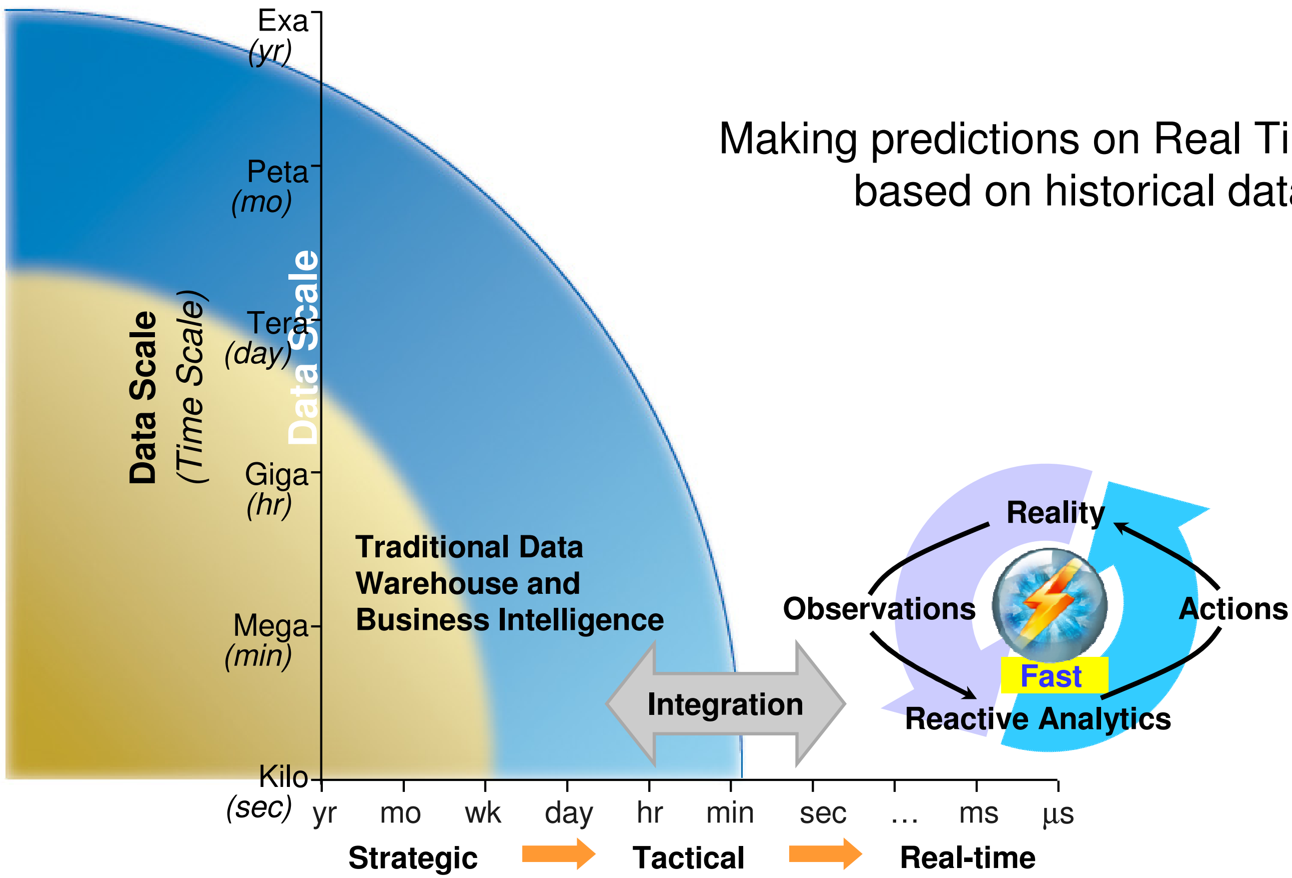


- Analyze data in motion
- Manage and govern data in the ecosystem
- Enterprise data integration
- Grow and evolve on current infrastructure

IBM Combining deep and reactive Analytics

Combining Deep and Reactive Analytics

Making predictions on Real Time data based on historical data

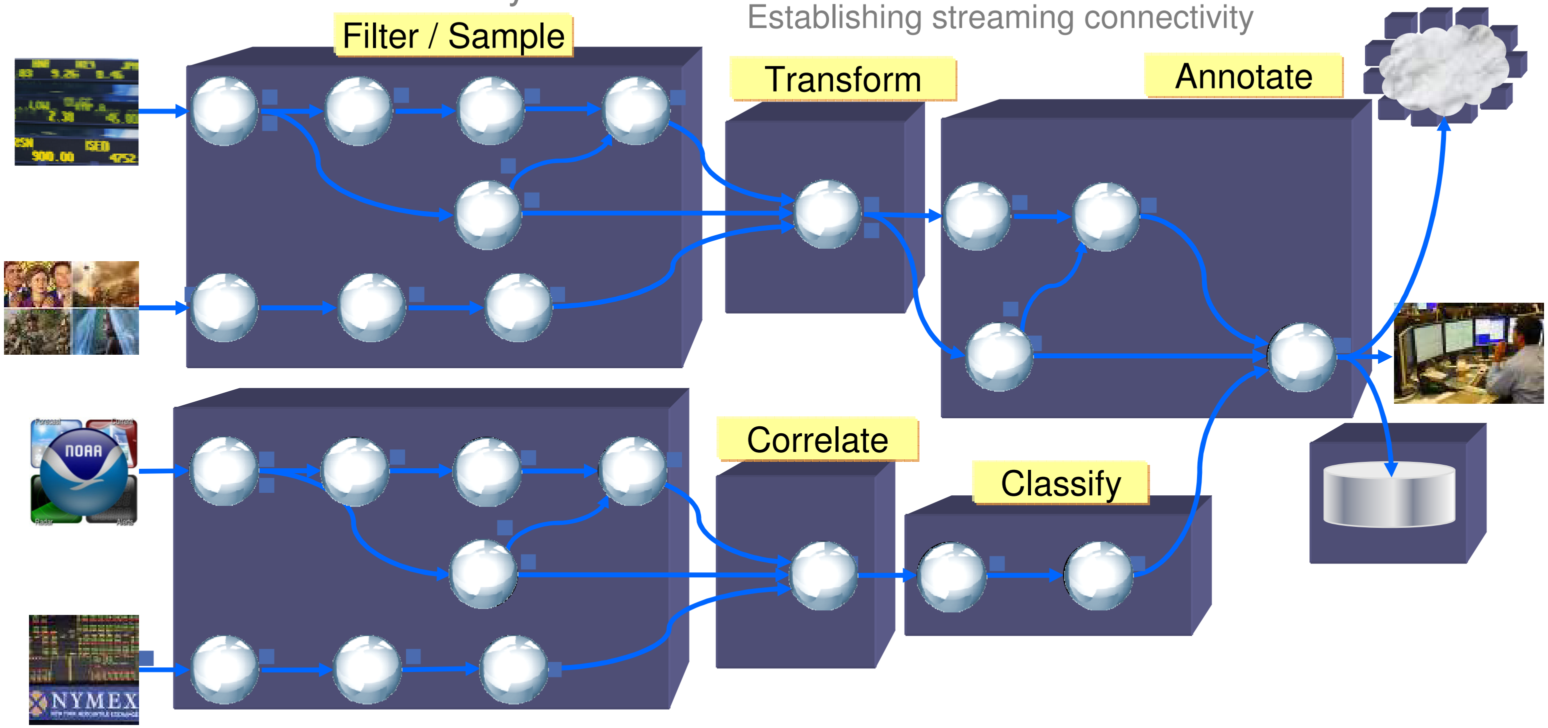


IBM how streams works

How Streams Works

→ Continuous ingestion
→ Continuous analysis

Infrastructure provides services for
Scheduling analytics across hardware hosts,
Establishing streaming connectivity



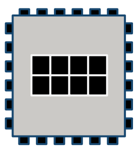
Achieve scale:
By partitioning applications into software components

Where appropriate:
Elements can be *fused* together

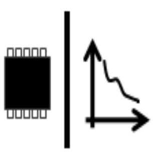
In-Memory Computing – The Time is NOW

The elements of In-Memory computing are not new. However, dramatically improved hardware economics and technology innovations in software has now made it possible for SAP to deliver on its vision of the Real-Time Enterprise with In-Memory business applications

• HW Technology Innovations

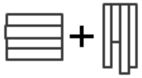


- Multi-Core Architecture (8 x 8core CPU per blade)
- Massive parallel scaling with many blades

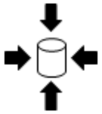


- 64bit address space – 2TB in current servers
- 100GB/s data throughput
- Dramatic decline in price/ performance

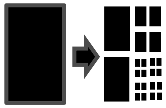
• SAP SW Technology Innovations



- Row and Column Store

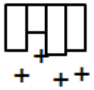


- Compression



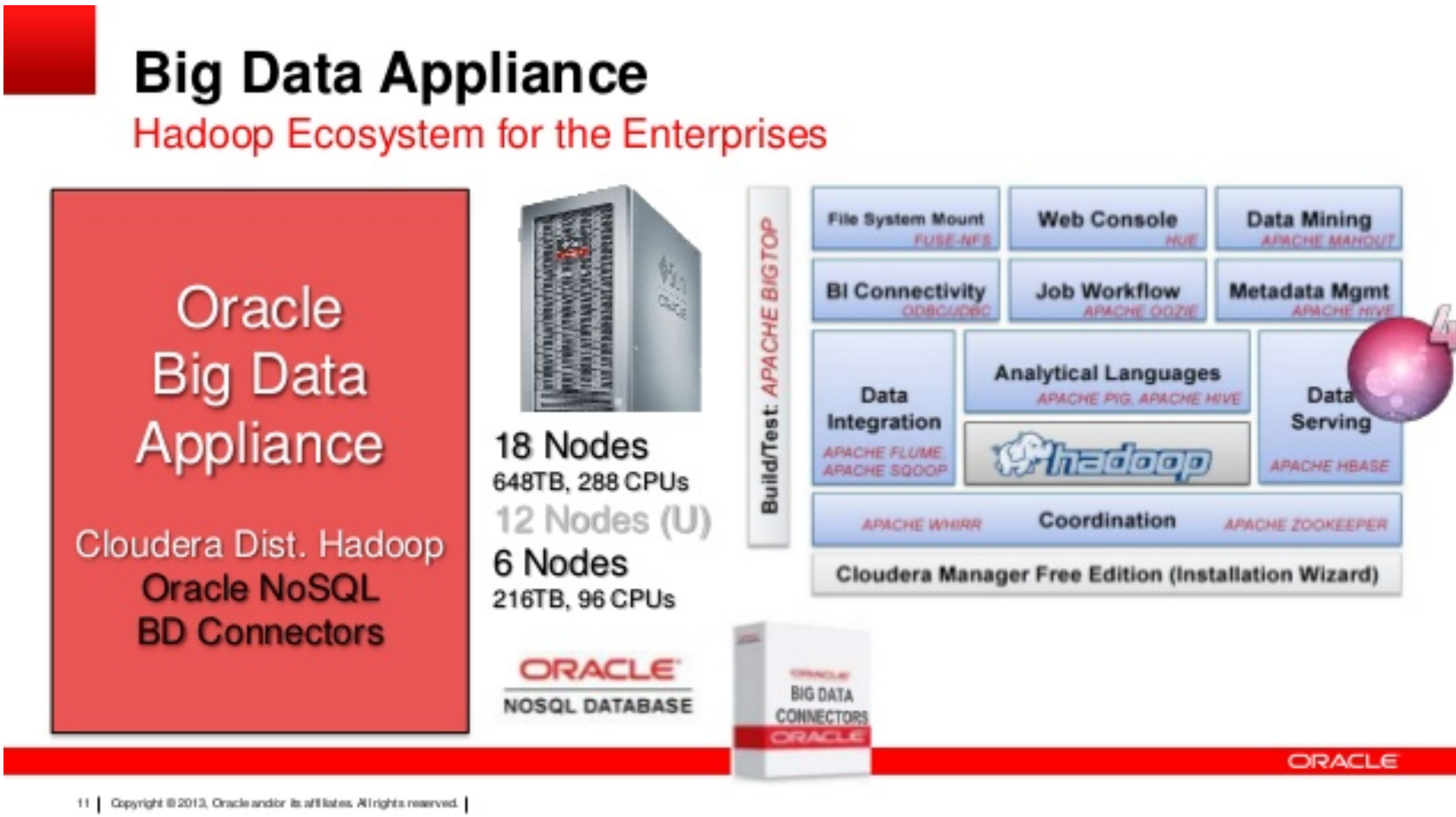
- Partitioning

- No Aggregate Tables



- Real-Time Data Capture
- Insert Only on Delta

- Appliance: ORACLE EXADATA



Oracle

Specific Costs for Build versus Buy Comparison

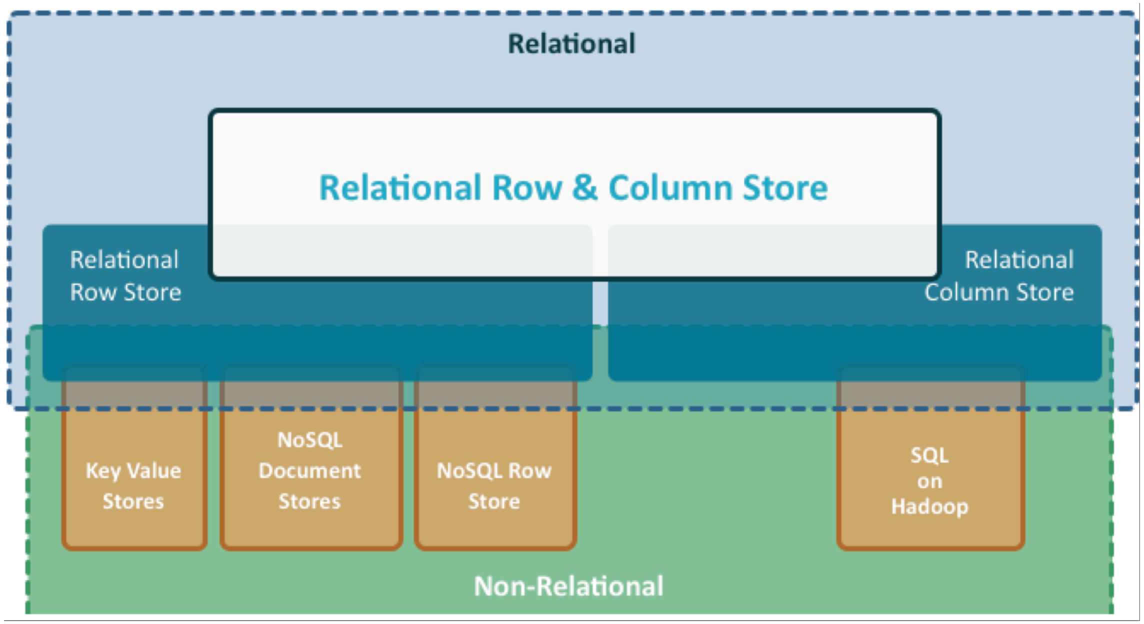
Table 1 lists those project items where ESG believes there is a pricing choice between build and buy. The table reflects estimated pricing for the "build" consumption option only.

Table 1. Medium Big Data Project Three-year TCO – Summary of Buy Cost Items

Item	Cost	Notes
Build Versus Buy Elements (Using Build Pricing)		
Servers	$410,500	18 @ $22.8k each; enterprise class with dual power supplies, 48TB of serial-attached SCSI (SAS) storage, 48-64 gigabytes memory, 1 rack
Networking	$40,000	3 @ estimated $6k for InfiniBand, 1 @ $11k for admin switch, 18 @ $0.6k for cables, looms, patch panels, etc.
Hardware support (three years)	$67,600	@15% of list cost
Hadoop licensing	$129,600	Cloudera: 18 nodes @ estimated $7.2k each
Installation	$14,000	Licenses and dedicated hardware
Build Project Costs	***$920,900***	Those project items where a "buy" option exists

http://www.slideshare.net/SwissHUG/how-to-explore-big-data-lmc-swlnk

The Database Landscape



Advances in in-memory computing technology are making "hybrid transactional and analytical processing"(HTAP) a reality. HTAP is performing transactional and analytical operations in a single database of record, often doing time-sensitive analysis of streaming data.

- MemSQL
  - In-Memory Storage
  - Access to Real-Time and Historical Data
  - Code Generation and Compiled Query Execution Plans
  - Lock-Free Data Structures and Multiversion Concurrency Control
  - Fault Tolerance and ACID Compliance

Bibliography

[1] Zaharia, Matei, et al. "Spark: cluster computing with working sets."Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. 2010.

[2] Battré, Dominic, et al. "Nephele/PACTs: a programming model and execution framework for web-scale analytical processing." Proceedings of the 1st ACM symposium on Cloud computing. ACM, 2010.

[3] amplab.cs.berkeley.edu/benchmark

[4] Amit Seth: "Transforming Big Data into Smart Data " - www.slideshare.net/apsheth/transforming-big-data-into-smart-data-deriving-value-via-harnessing-volume-variety-and-velocity-using-semantic-techniques-and-technologies

[5] AnHai Doan, Alon Y. Halevy, Zachary G. Ives: Principles of Data Integration. Morgan Kaufmann 2012

[6] X.L.Dong and Divesh Srivastava. Big data integration. Tutorial in VLDB'13

[7] http://voltdb.com/stonebraker-says

[8] Michael Stonebraker, Ugur Çetintemel: "One Size Fits All: An Idea Whose Time Has Come and Gone", ICDE 2005: pp. 2-11.

[9] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber: "Bigtable: A Distributed Storage System for Structured Data", OSDI 2006.

Bibliography

[10] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels, "Dynamo: Amazon's Highly Available Key-value Store", SOSP 2007.

[11] http://blog.nahurst.com/visual-guide-to-nosql-systems

[12] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, and partition-tolerant web services", ACM SIGACT News 33, 2, pp 51-59, 2002

[13] David J. DeWitt, Samuel Madden, Michael Stonebraker, "How to Build a High-Performance Data Warehouse", http://db.csail.mit.edu/madden/high_perf.pdf

[14] Eric A. Brewer, "Towards robust distributed systems". (Invited Talk) Principles of Distributed Computing, Portland, Oregon, 2000.

[15] Stadford University CS 345a, "MapReduce", 2010. http://www.stanford.edu/class/cs345a/slides/02-mapreduce.pdf

[16] Neil Conway "Bloom and CALM", 2012 http://basho.com/blog/technical/2012/07/16/Neil-Conway-Basho-Chats-CALM-Bloom/

Thank You