Big Data

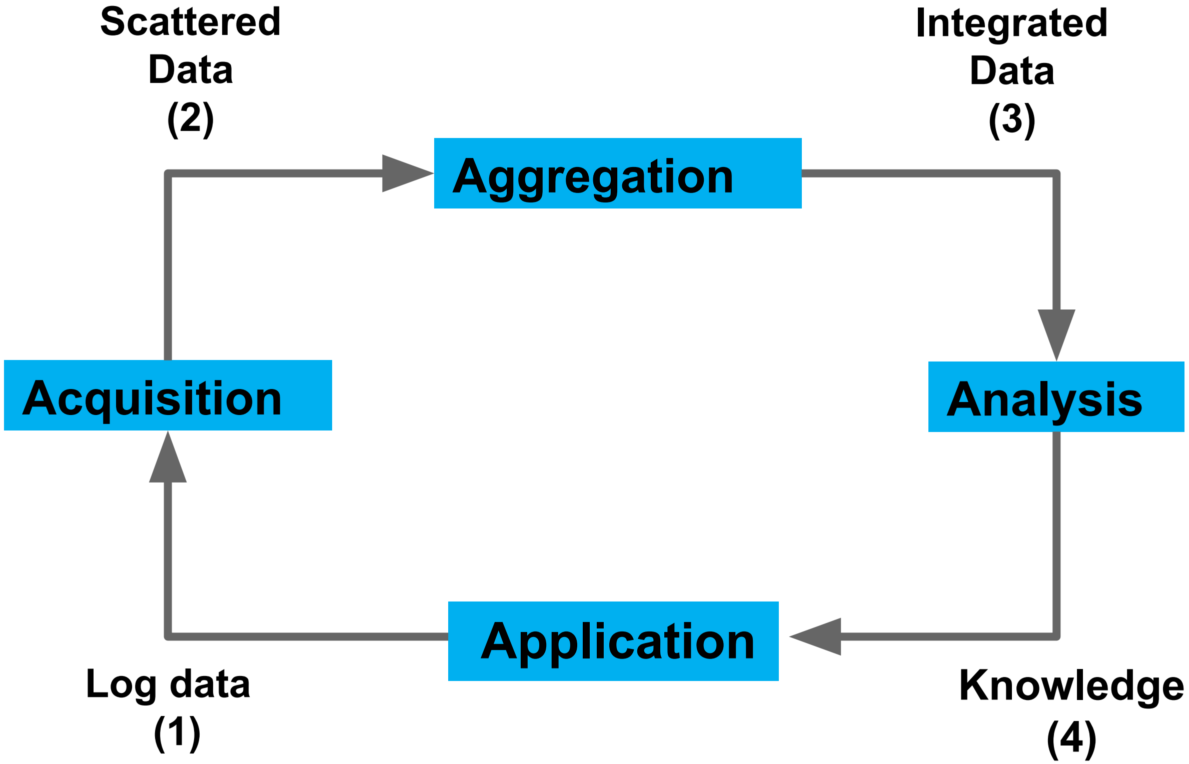
Introduction to Big Data
Lecture 2

Overview

- Data and Analytics
- Traditional Analytics
- SQL Databases
- Big Data
- Big Data Analytics
- NoSQL Databases

Data and Analytics

Data Lifecycle



Analytics = Discovery

- Novelty Discovery
  - Finding new, rare, one-in-a-[million / billion / trillion/ etc.] objects and events
- Class Discovery
  - Finding new classes of objects and behaviors
  - Learning the rules that constrain class boundaries
- Association Discovery
  - Finding unusual (improbable) co-occurring associations
- Correlation Discovery
  - Finding patterns and dependencies, which reveal new natural laws or new scientific principles

From: Kirk Borne, Dynamic Events in Massive Data Streams, GMU

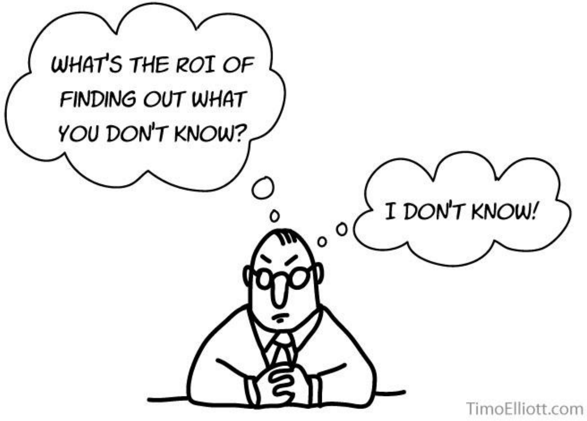
Goals of Data Analytics

From sensors (data collection, measurement, observation, …)

to Monitoring and Alerting

to Sense making (Data and Analytics Science)

to Cents-Making (Getting to ROI!!)



From: Kirk Borne, Dynamic Events in Massive Data Streams, GMU

Challenge of Data Analytics

- Challenges
    - Find the/a key pattern that indicates a situational change:
        - single event
        - sequence of events
    - Have we seen this pattern before?
        - Determine its characteristics, not just that it exists
    - Predict what event occurs next because this/these event(s) occurred in the pattern
    - How to identify relevant fragments of data easily from a multitude of data sources?
    - Difficult to determine what the right answer is in advance
- Issues
    - The needle hasn’t grown as fast as the haystack!!
    - We need new analytics methods to deal with larger, more complex data and problems!!

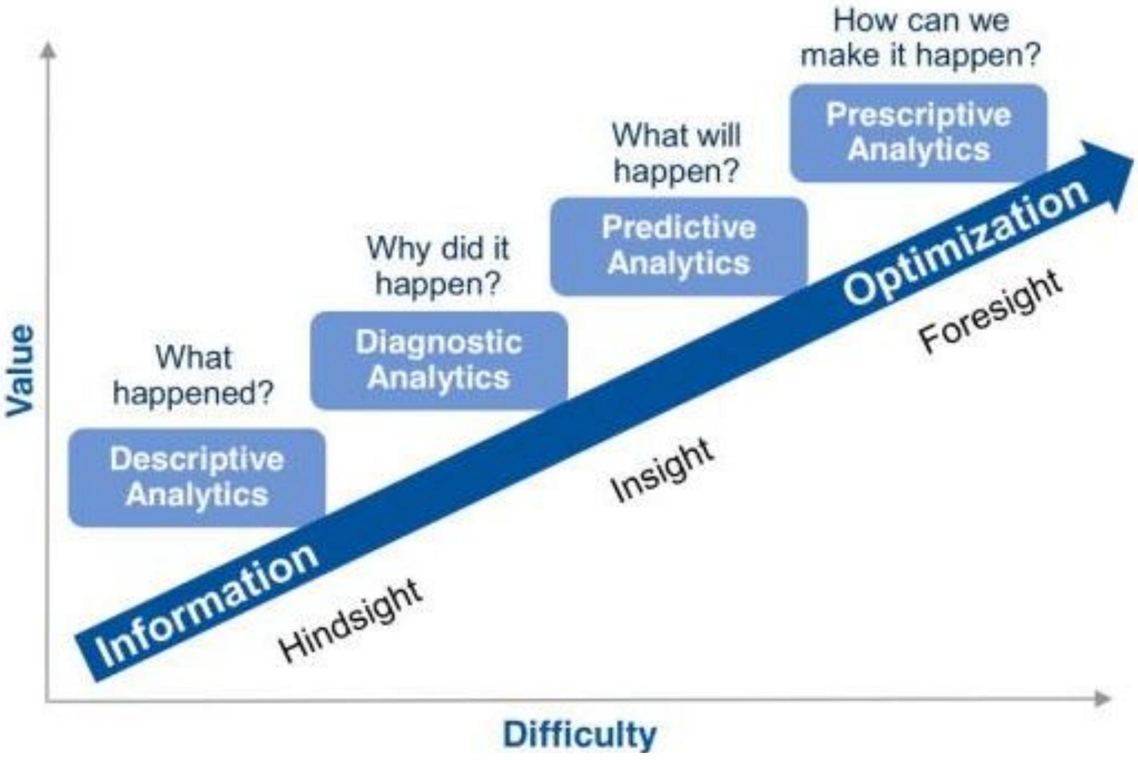
Traditional Data Analytics

Types of “Traditional” Analytics

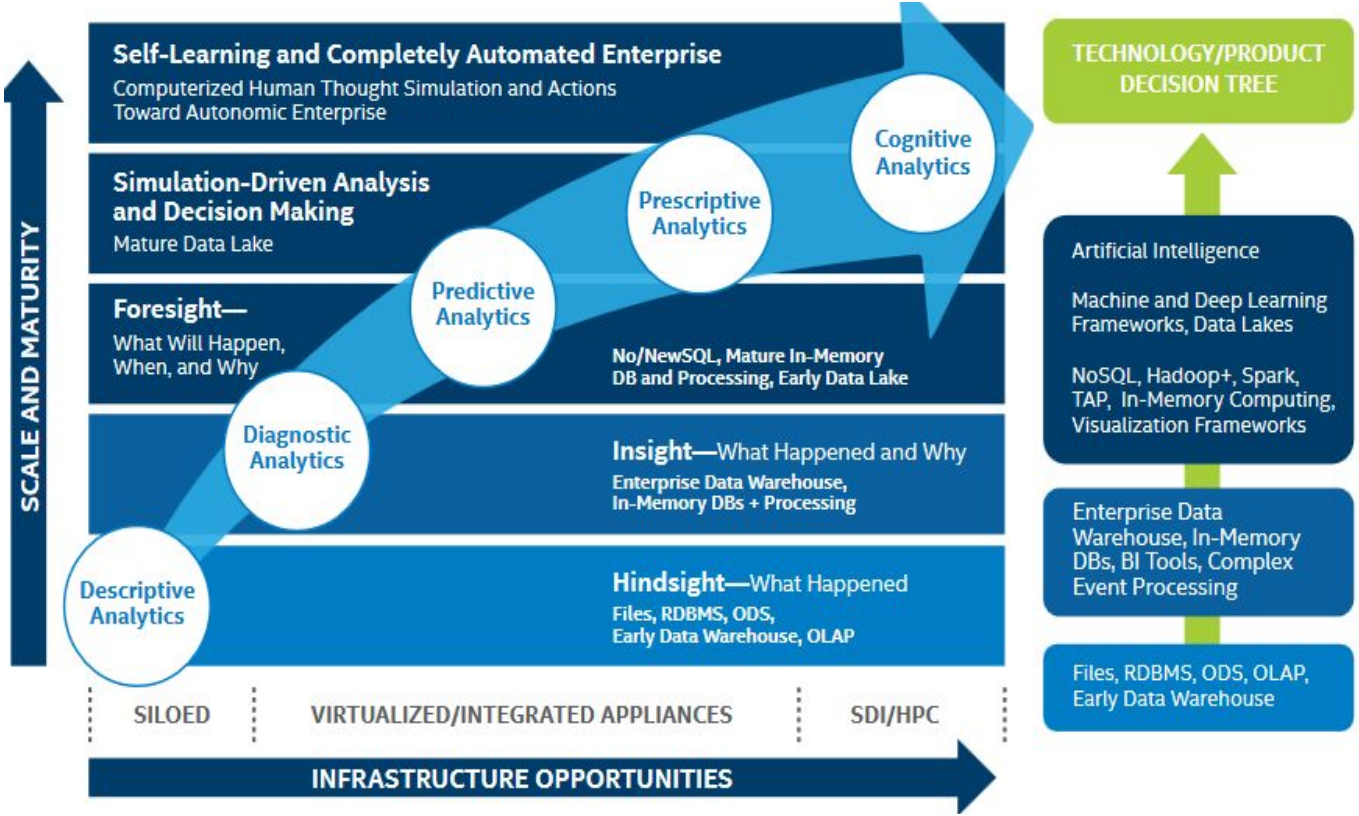
- **Descriptive**: A set of techniques for reviewing and examining the data set(s) to understand the data and analyze business performance
- **Diagnostic**: A set of techniques for determine what has happened and why
- **Predictive**: A set of techniques that analyse current and historical data to determine what is most likely to (not) happen
- **Prescriptive**: A set of techniques for computationally developing and analyzing alternatives that can become courses of action – either tactical or strategic – that may discover the unexpected
- **Decisive**: A set of techniques for visualizing information and recommending courses of action to facilitate human decision-making when presented with a set of alternatives

	Passive	Active
Deductive	Descriptive	Diagnostic
Inductive	Predictive	Prescriptive

Types of “Traditional” Analytics



Types of “Traditional” Analytics


Self-Learning and Completely Automated Enterprise
Computerized Human Thought Simulation and Actions Toward Autonomic Enterprise
Simulation-Driven Analysis and Decision Making
Mature Data Lake
Foresight—
What Will Happen, When, and Why
No/NewSQL, Mature In-Memory DB and Processing, Early Data Lake
Insight—What Happened and Why
Enterprise Data Warehouse, In-Memory DBs + Processing
Hindsight—What Happened
Files, RDBMS, ODS, Early Data Warehouse, OLAP
Descriptive Analytics
Diagnostic Analytics
Predictive Analytics
Prescriptive Analytics
Cognitive Analytics
SCALE AND MATURITY
SILOED
VIRTUALIZED/INTEGRATED APPLIANCES
SDI/HPC
INFRASTRUCTURE OPPORTUNITIES
TECHNOLOGY/PRODUCT DECISION TREE
Artificial Intelligence
Machine and Deep Learning Frameworks, Data Lakes
NoSQL, Hadoop+, Spark, TAP, In-Memory Computing, Visualization Frameworks
Enterprise Data Warehouse, In-Memory DBs, BI Tools, Complex Event Processing
Files, RDBMS, ODS, OLAP, Early Data Warehouse

Big Data

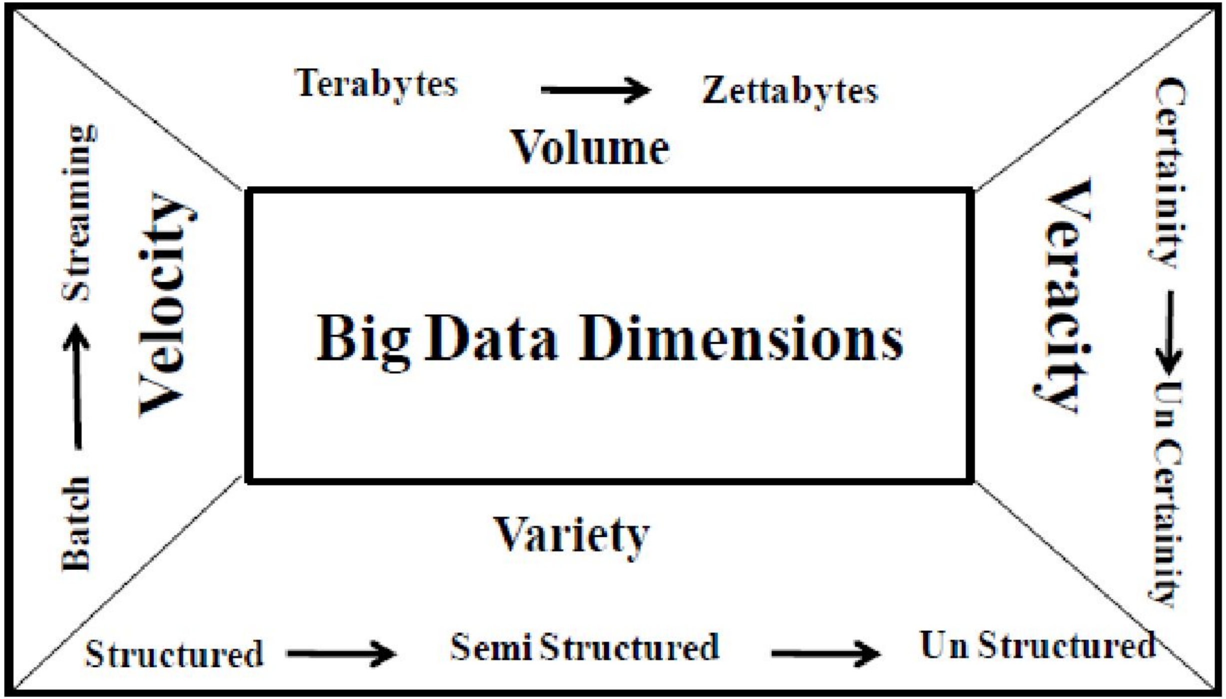
Big Data

- the amount of data just beyond technology’s capability to store, manage and process efficiently.
- data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it...

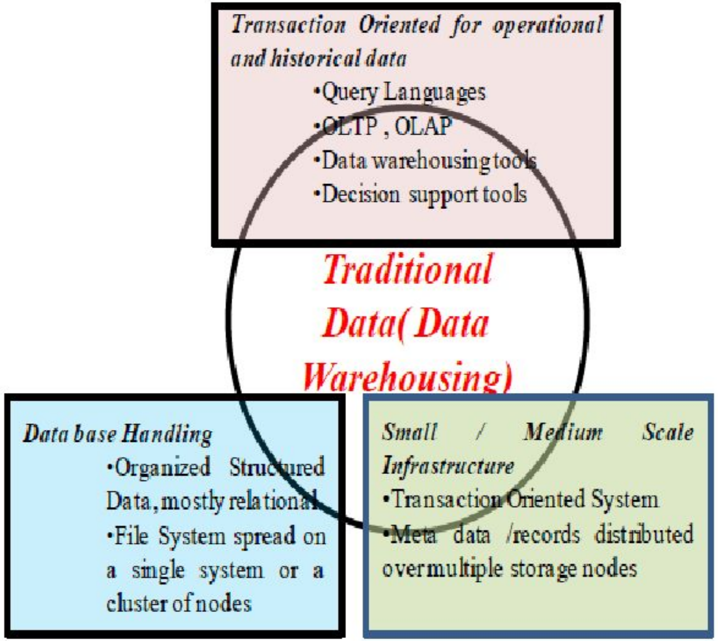
Data vs Big Data

Standard Data (OLTP / OLAP)		Big Data
Structured / Processed	Data	Structured / Semi Structured / Unstructured
Schema on Write	Processing / Querying	Schema on read
Less agile – more up front development	Agility	More agile – allows for dynamic changes
Business professionals / applications	Users	Data Scientists / BI professionals

Data vs Big Data



Data vs Big Data



Traditional Data(Data Warehousing)

Transaction Oriented for operational and historical data

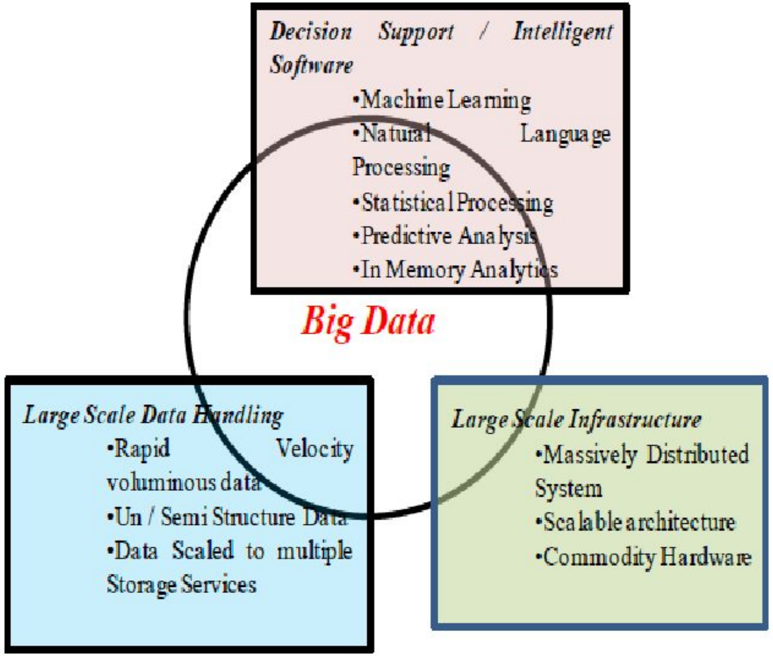
- Query Languages
- OLTP , OLAP
- Data warehousing tools
- Decision support tools

Data base Handling

- Organized Structured Data, mostly relational
- File System spread on a single system or a cluster of nodes

Small / Medium Scale Infrastructure

- Transaction Oriented System
- Meta data /records distributed over multiple storage nodes



Big Data

Decision Support / Intelligent Software

- Machine Learning
- Natural Language Processing
- Statistical Processing
- Predictive Analysis
- In Memory Analytics

Large Scale Data Handling

- Rapid Velocity voluminous data
- Un / Semi Structure Data
- Data Scaled to multiple Storage Services

Large Scale Infrastructure

- Massively Distributed System
- Scalable architecture
- Commodity Hardware

SQL

SQL Characteristics

- Data stored in columns and tables
- Relationships represented by data
- Data Manipulation Language
- Data Definition Language
- Transactions
- Abstraction from physical layer

SQL Physical Layer Abstraction

- Applications specify what, not how
- Query optimization engine
- Physical layer can change without modifying applications
  - Create indexes to support queries
  - In Memory databases

Data Manipulation Language (DML)

- Data manipulated with Select, Insert, Update, & Delete statements

  ```
  Select T1.Column1, T2.Column2 …
  From Table1, Table2 …
  Where T1.Column1 = T2.Column1 …
  ```

- Data Aggregation
- Compound statements
- Functions and Procedures
- Explicit transaction control

Data Definition Language

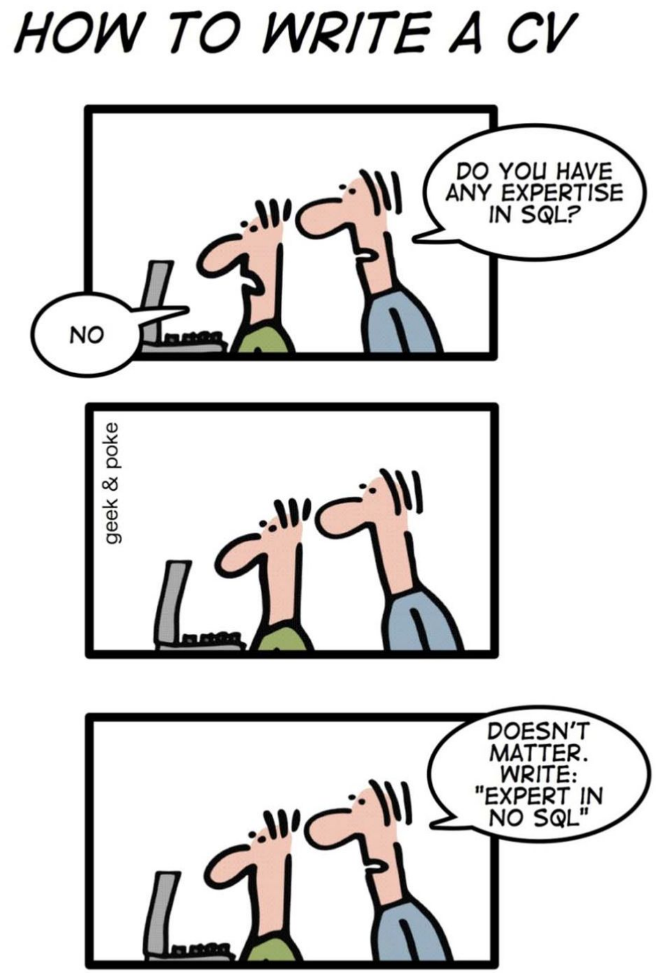
- Schema defined at the start

  Create Table (Column1 Datatype1, Column2 Datatype 2, …)

- Constraints to define and enforce relationships
  - Primary Key
  - Foreign Key
  - Etc.
- Triggers to respond to Insert, Update & Delete
  - Stored Modules
  - Alter…
  - Drop…
  - Security and Access Control

SQL Transactions – ACID Properties

- **Atomic** – All of the work in a transaction completes (commit) or none of it completes
- **Consistent** – A transaction transforms the database from one consistent state to another consistent state. Consistency is defined in terms of constraints.
- **Isolated** – The results of any changes made during a transaction are not visible until the transaction has committed.
- **Durable** – The results of a committed transaction survive failures


HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"

Leverage the NoSQL boom

Why is the name NoSQL ?

Why is the name NoSQL ?

Not
Only SQL

Brewer’s CAP Theorem (E. Brewer, N. Lynch)

A distributed system can support only two of the following characteristics:

- Consistency
- Availability
- Partition tolerance

Brewer’s CAP Theorem …

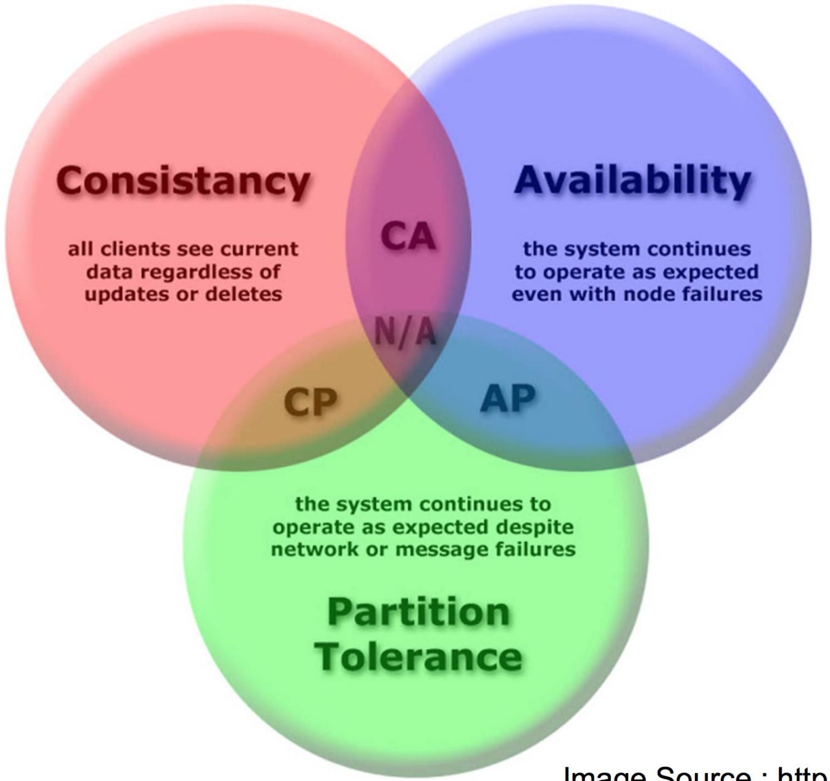

Consistancy
all clients see current data regardless of updates or deletes
CA
N/A
Availability
the system continues to operate as expected even with node failures
CP
AP
the system continues to operate as expected despite network or message failures
Partition Tolerance

Image Source : http://blog.nosqltips.com

CAP Theorem

CAP theorem states that any networked shared-data system can provide only two out of the following three properties mentioned below.

- **Consistency**: similar to the consistency property of ACID (Atomicity, Consistency, Isolation, Durability) , the data is synchronized across all cluster nodes, and all the nodes would see the similar data at the same time.
- **Availability**: guaranteed that every request receives a response however, the request is successful/failed in receiving the data which has been requested would not be known.
- **Partition tolerance**: single node failure should not cause the entire system to fail and the system should continue to function even under circumstances of arbitrary message loss or partial failure of the system.

BASE Transactions

Acronym contrived to be the opposite of ACID

- **B**asically **A**vailable
  - This constraint states that the system guarantees the availability of the data as regards to the CAP Theorem
- **S**oft state
  - The state of the system could change over time, and called as ‘soft’ state
- **E**ventually Consistent
  - The system will eventually become consistent once it stops receiving input. Here the data will propagate everywhere sooner or later, but if the system continues to receive input it will not check for the consistency of every transaction before it moves onto the next transaction.

NoSQL Characteristics

- Weak consistency – stale data OK
- Availability first
- Best effort
- Approximate answers OK
- Aggressive (optimistic)
- Simpler and faster

NoSQL Database Types

Discussing NoSQL databases is complicated because there are a variety of types:

- Column Store – Each storage block contains data from only one column
- Document Store – stores documents made up of tagged elements
- Key-Value Store – Hash table of keys
- XML Databases
- Graph Databases
- Codasyl Databases
- Object Oriented Databases
- Etc...

Column Store

Each storage block contains data from only one column

- Example
  - Hadoop / Hbase
    - http://hadoop.apache.org/
  - Ingres VectorWise (Column Store integrated with an SQL database)
  - Google BigTable
- More efficient than row (or document) store if:
  - Multiple row/record/documents are inserted at the same time, so updates of column blocks can be done together
  - Retrievals access only some of the columns in a row/record/document

Key-Value Store

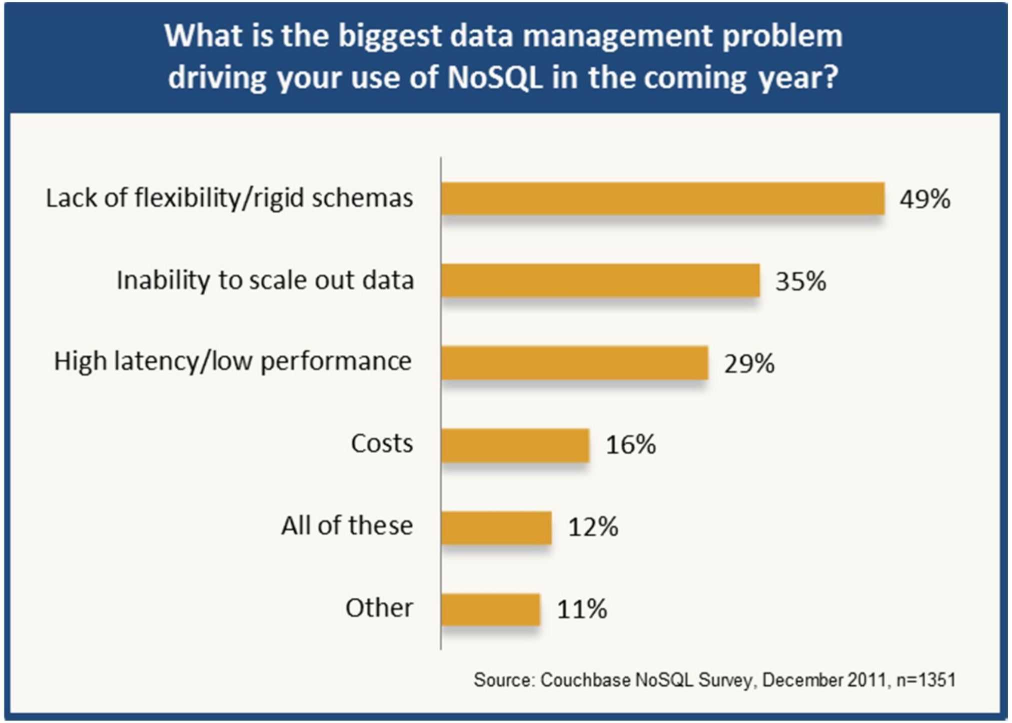
- Store items as alphanumeric identifiers (keys) with associated values in simple, standalone tables (Hash table)
- The values can be simple text strings or more complex lists and sets.
- Query can usually only be performed against keys and it limited to exact matches
- Fast access to small data values
- Example
    - Project-Voldemort (Linkedin)
    - MemCacheDB
    - Backend storage is Berkeley-DB
    - Amazon Dynamo (P2P key-value store)

Document Store

- Schema-free, document oriented database
- Uses JSON – JavaScript Object Notation
- Example
  - CouchDB
  - MongoDB

Adoption of NoSQL Database

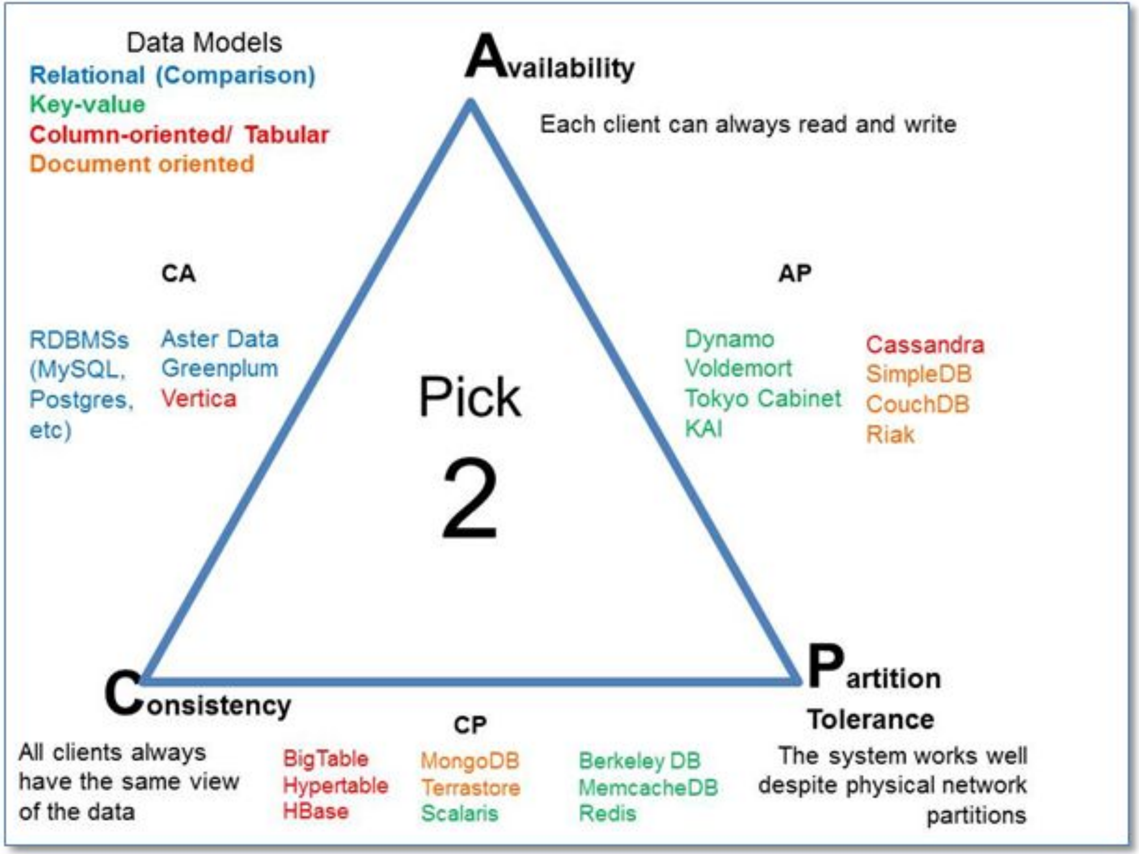
Couchbase survey was conducted in the 2012.



What is the biggest data management problem driving your use of NoSQL in the coming year?

Source: Couchbase NoSQL Survey, December 2011, n=1351

Databases Landscape with respect to CAP Theorem



Data Models
Relational (Comparison)
Key-value
Column-oriented/ Tabular
Document oriented

Availability

Each client can always read and write

Pick 2

CA

RDBMSs (MySQL, Postgres, etc)

Aster Data
Greenplum
Vertica

AP

Dynamo
Voldemort
Tokyo Cabinet
KAI

Cassandra
SimpleDB
CouchDB
Riak

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

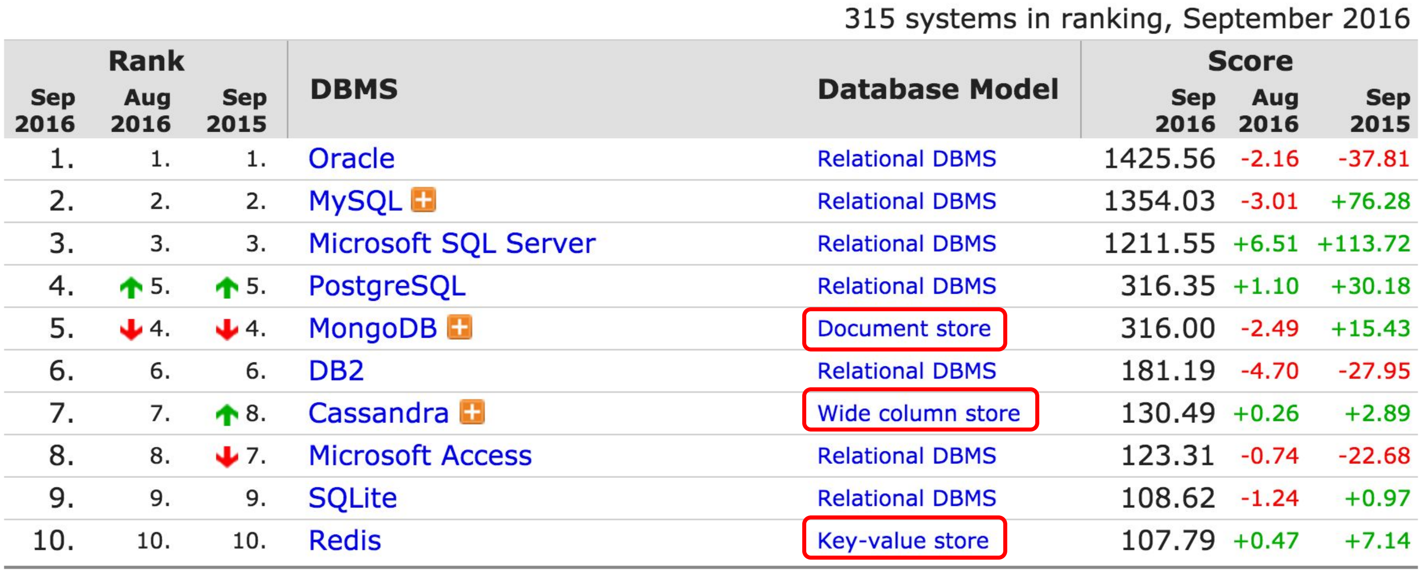
http://blog.flux7.com/blogs/nosql/cap-theorem-why-does-it-matter

NoSQL Summary

NoSQL databases:

- Rejects overhead of ACID transactions
    - “Complexity” of SQL
    - Burden of up-front schema design
    - Declarative query expression
- Programmer responsible for
    - Step-by-step procedural language
    - Navigating access path

Database Ranking



315 systems in ranking, September 2016

Rank			DBMS	Database Model	Score		
Sep 2016	Aug 2016	Sep 2015			Sep 2016	Aug 2016	Sep 2015
1.	1.	1.	Oracle	Relational DBMS	1425.56	-2.16	-37.81
2.	2.	2.	MySQL ⊞	Relational DBMS	1354.03	-3.01	+76.28
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1211.55	+6.51	+113.72
4.	↑ 5.	↑ 5.	PostgreSQL	Relational DBMS	316.35	+1.10	+30.18
5.	↓ 4.	↓ 4.	MongoDB ⊞	Document store	316.00	-2.49	+15.43
6.	6.	6.	DB2	Relational DBMS	181.19	-4.70	-27.95
7.	7.	↑ 8.	Cassandra ⊞	Wide column store	130.49	+0.26	+2.89
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	123.31	-0.74	-22.68
9.	9.	9.	SQLite	Relational DBMS	108.62	-1.24	+0.97
10.	10.	10.	Redis	Key-value store	107.79	+0.47	+7.14

From : http://db-engines.com/en/ranking

NOTE: Most of the major Relational DB Vendors have included NoSQL components to their solutions to stay ahead of the competition.

What we covered

- SQL Databases
  - SQL Standard
  - SQL Characteristics
- NoSQL Databases
  - What’s NoSQL?
  - CAP Theorem
  - BASE Transactions
  - General Characteristics
  - NoSQL Database Types
- In-Memory Databases