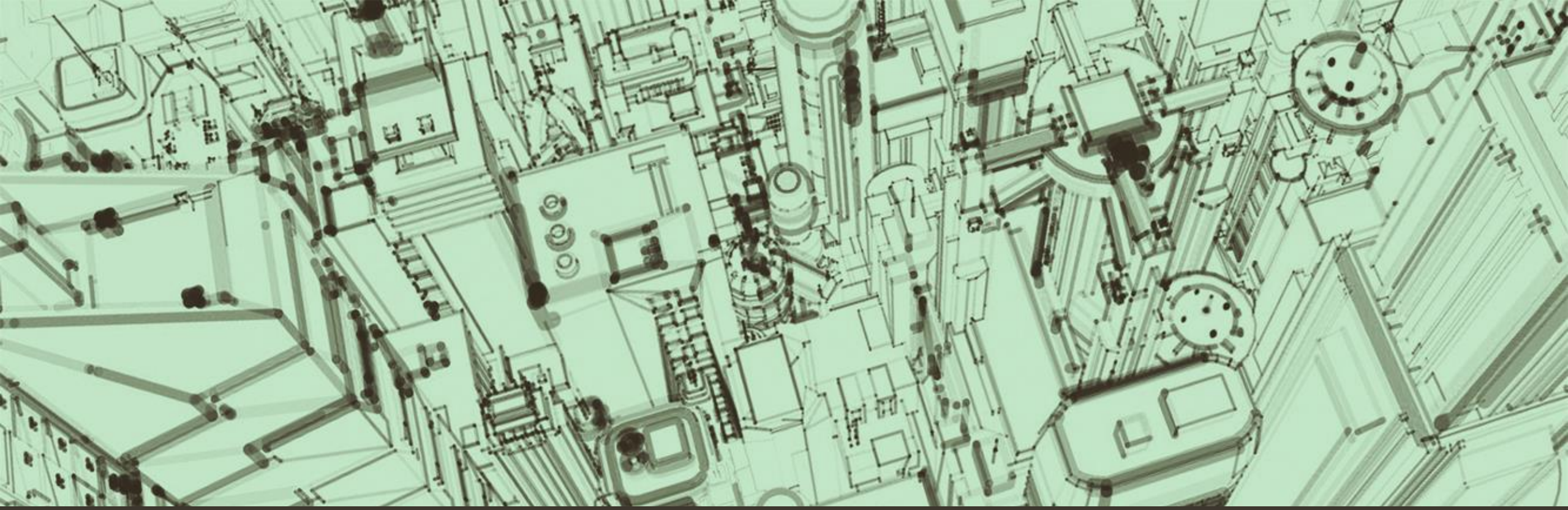


NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison

International Journal of Database Theory and Application. Vol. 6, No. 4. 2013

Introduction

- NoSQL, for ―Not Only SQL,‖ refers to an eclectic and increasingly familiar group of non-relational data management systems; where databases are not built primarily on tables, and generally do not use SQL for data manipulation.

- NoSQL systems are distributed, non-relational databases designed for large-scale data storage and for massively-parallel data processing across a large number of commodity servers.

Introduction

- NoSQL database systems arose alongside major Internet companies, such as Google, Amazon, and Facebook; which had challenges in dealing with huge quantities of data with conventional RDBMS solutions could not cope.

- There are two trends that bringing these problems)related to RDBMS) to the attention of the international software community:
  - 1. The exponential growth of the volume of data generated by users, systems and sensors, further accelerated by the concentration of large part of this volume on big distributed systems.
  - 2. The increasing interdependency and complexity of data accelerated by the Internet, Web2.0, social networks and open and standardized access to data sources from a large number of different systems.

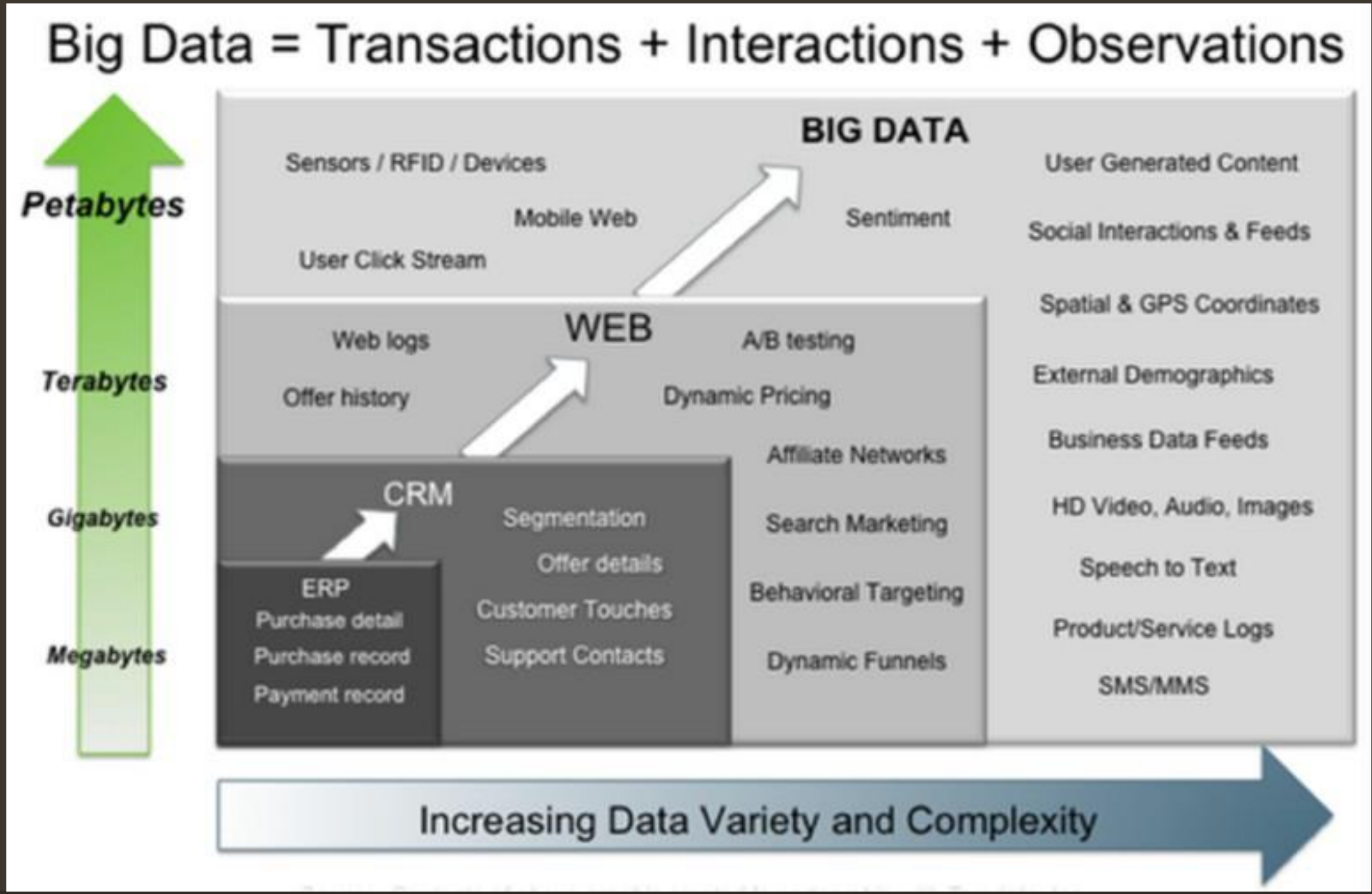


Figure 1: Big Data Transactions with Interactions and Observations.

Classification of NoSQL Databases

- (1) Key-Value stores;
- (2) Document databases (or stores);
- (3) Wide-Column (or Column-Family) stores;
- (4) Graph databases.

Key-Value stores

- **Primary Use:** The simplicity of Key-Value Stores makes them ideally suited to lightning-fast, highly-scalable retrieval of the values needed for application tasks like managing user profiles or sessions or retrieving product names.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

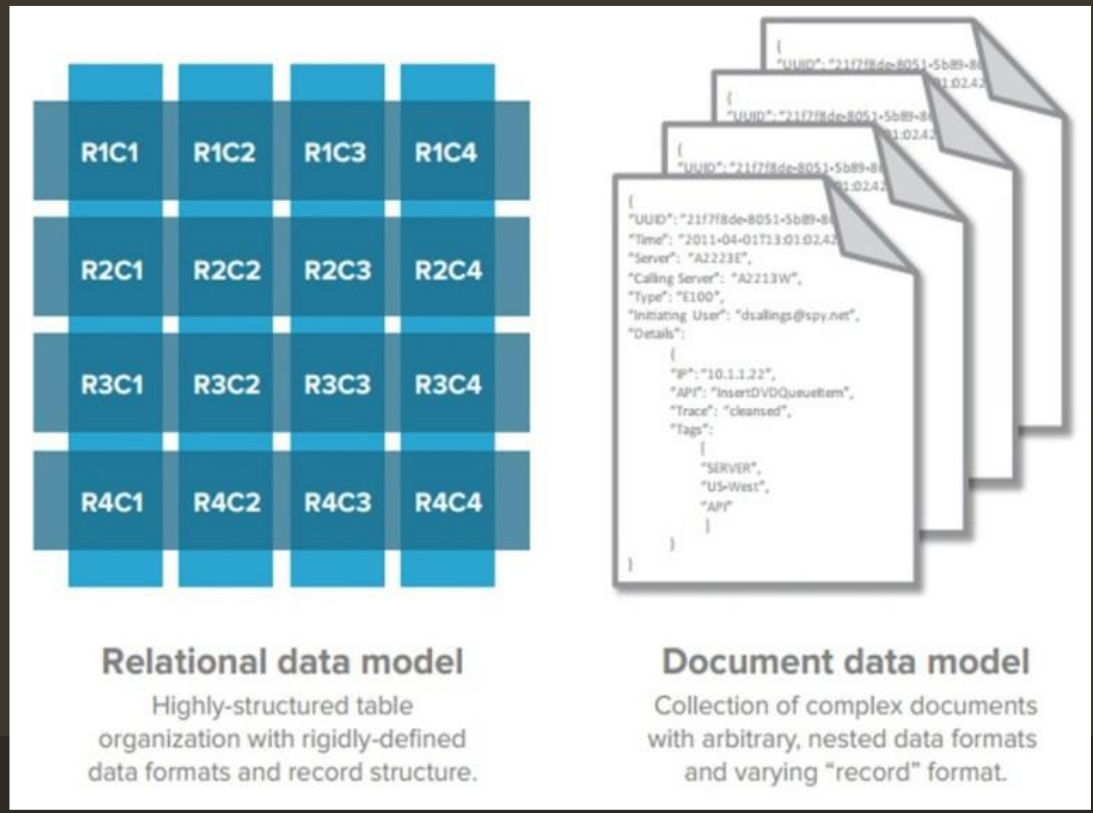
- Examples: Key-Value Stores- Dynamo (Amazon);

Voldemort (LinkedIn); Redis; BerkeleyDB; Riak.

Document databases (or stores)

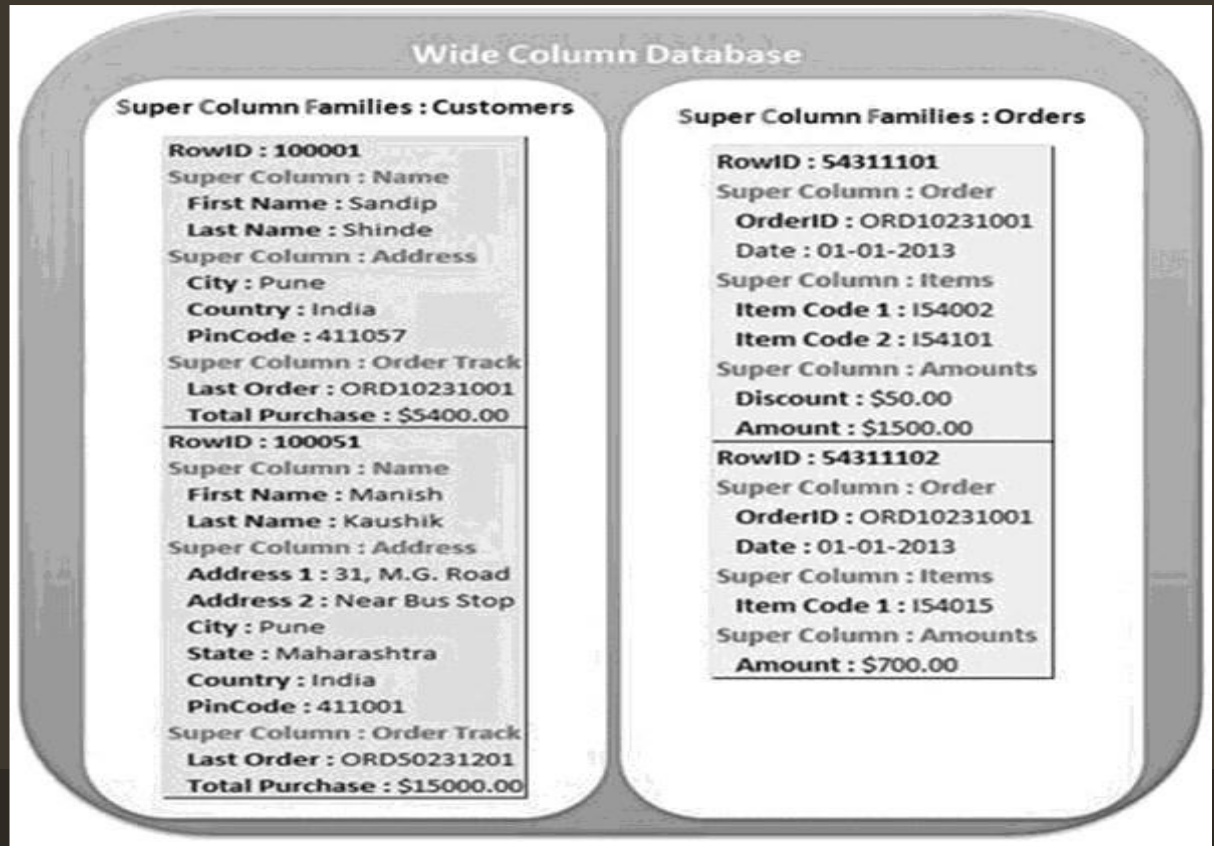
- **Primary Use:** Document databases are good for storing and managing Big Data-size collections of literal documents, like text documents, email messages, and XML documents, as well as conceptual documents like de-normalized representations of a database entity such as a product or customer.

- Examples: CouchDB (JSON); MongoDB (BSON).
- MongoDB and CouchDB are open source and they are document oriented and schema free.



Wide-Column (or Column-Family) Stores

- Primary Uses: This type of DMS is great for:
- (1) Distributed data storage, especially versioned data because of time-stamping functions.
- (2) Large-scale, batch-oriented data processing: sorting, parsing, conversion, algorithmic crunching, etc.
- (3) Exploratory and predictive analytics.



- **Examples: Bigtable (Google); Hypertable;**

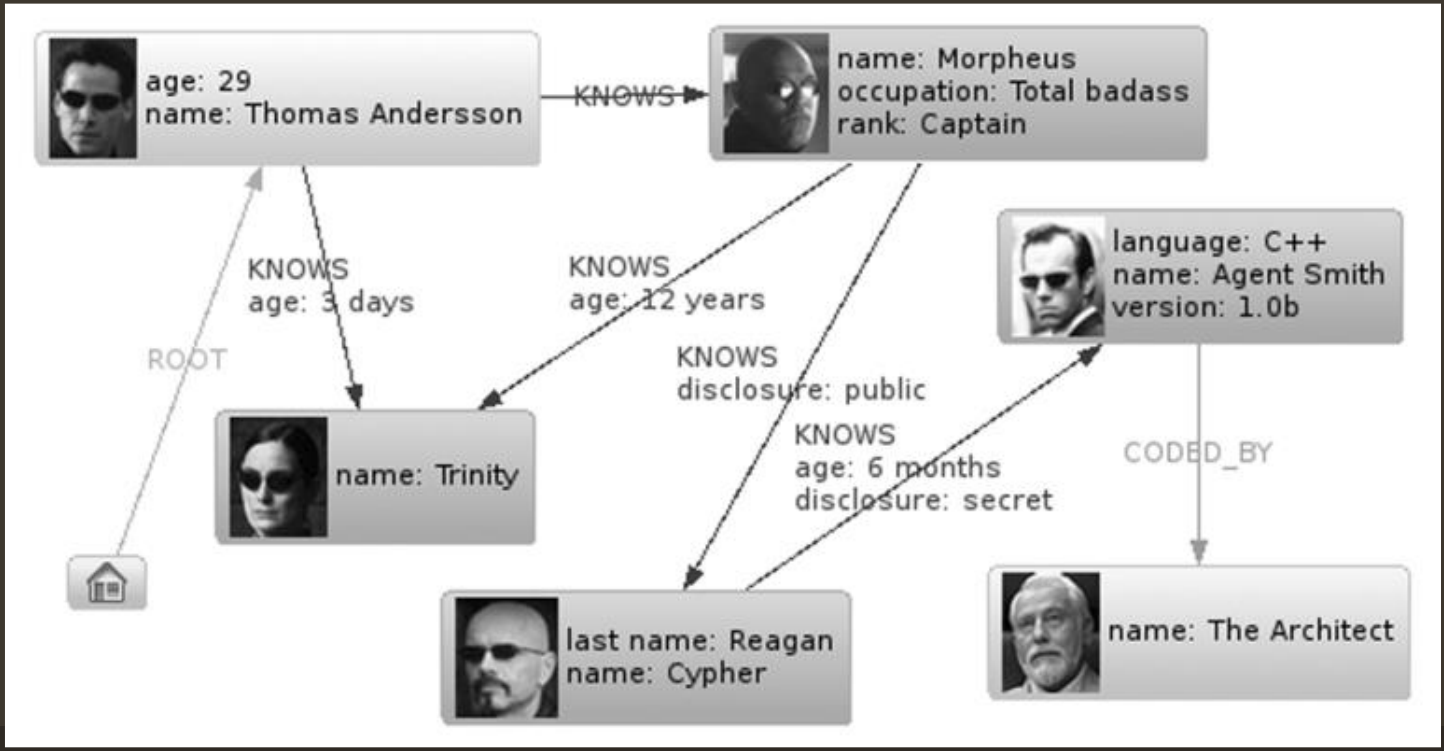
Cassandra (Facebook; used by Digg, Twitter);

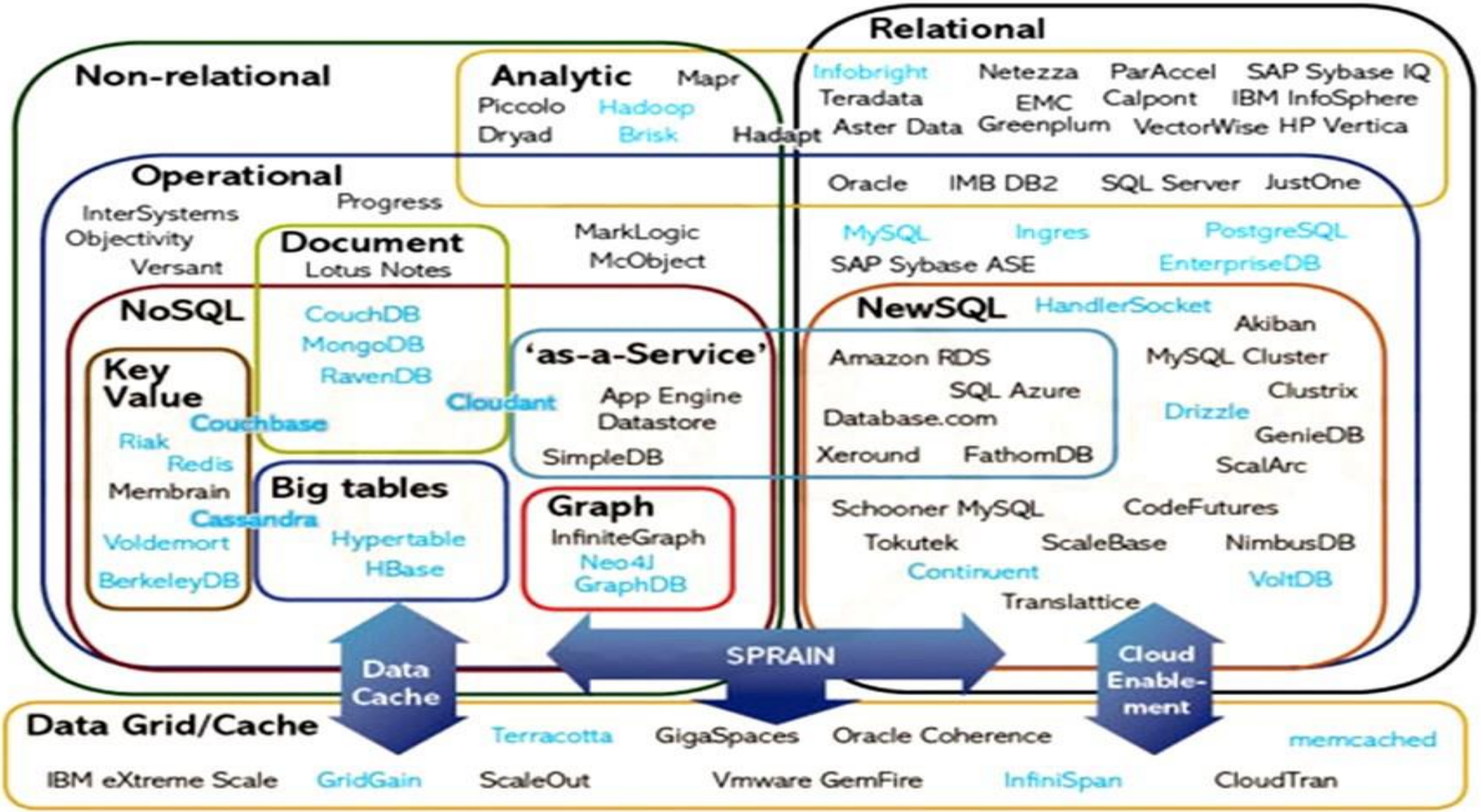
SimpleDB (Amazon); DynamoDB.

Graph Databases

- Primary Uses: In general, graph databases are useful when you are more interested in relationships between data than in the data itself: for example, in representing and traversing social networks, generating recommendations or conducting forensic investigations.

- Graph Database Examples: Neo4j; InfoGrid; Sones GraphDB; AllegroGraph; InfiniteGraph.




Relational
Non-relational
Analytic
Mapr
Piccolo
Hadoop
Dryad
Brisk
Hadapt
Infobright
Netezza
ParAccel
SAP Sybase IQ
Teradata
EMC
Calpont
IBM InfoSphere
Aster Data
Greenplum
VectorWise
HP Vertica
Operational
Oracle
IMB DB2
SQL Server
JustOne
Progress
InterSystems
Objectivity
Versant
Document
Lotus Notes
MarkLogic
McObject
MySQL
Ingres
PostgreSQL
SAP Sybase ASE
EnterpriseDB
NoSQL
CouchDB
MongoDB
RavenDB
Cloudant
Key Value
Couchbase
Riak
Redis
Membrain
Cassandra
Voldemort
BerkeleyDB
'as-a-Service'
App Engine Datastore
SimpleDB
Big tables
Hypertable
HBase
Graph
InfiniteGraph
Neo4J
GraphDB
NewSQL
HandlerSocket
Akiban
Amazon RDS
MySQL Cluster
SQL Azure
Clustrix
Drizzle
Database.com
GenieDB
Xeround
FathomDB
ScalArc
Schooner MySQL
CodeFutures
Tokutek
ScaleBase
NimbusDB
Continuent
VoltDB
Translattice
Data Cache
SPRAIN
Cloud Enable-ment
Data Grid/Cache
Terracotta
GigaSpaces
Oracle Coherence
memcached
IBM eXtreme Scale
GridGain
ScaleOut
Vmware GemFire
InfiniSpan
CloudTran

Attributes		NoSQL Databases								
Database model		Document-Stored		Wide-Column Stored				Key-Value Stored		Graph-oriented
	Features	MongoDB	CouchDB	DynamoBD	HBase	Cassandra	Accumulo	Redis	Riak	Neo4j
Design & Features	Data storage	Volatile memory File System	Volatile memory File System	SSD	HDFS		Hadoop	Volatile memory File System	Bitcask LevelDB Volatile memory	File System Volatile memory
	Query language	Volatile memory File System	JavaScript Memcached-protocol	API calls	API calls REST XML Thrift	API calls CQL Thrift		API calls	HTTP JavaScript REST Erlang	API calls REST SparQL Cypher Tinkerpop Gremlin
	Protocol	Custom, binary (BSON)	HTTP, REST	–	HTTP/REST Thrift	Thrift & custom binary CQL3	Thrift	Telnet-like	HTTP, REST	HTTP/REST Tembedding in Java
	Conditional entry updates	Yes	Yes	Yes	Yes	No	Yes	No	No	
	MapReduce	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
	Unicode	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	TTL for Entries	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	
	Compression	Yes	Yes	–	Yes	Yes	Yes	Yes	Yes	
Integrity	Integrity model	BASE	MVCC	ASID	Log Replication	BASE	MVCC	–	BASE	ASID
	Atomicity	Conditional	Yes	Yes	Yes	Yes	Conditional	Yes	No	Yes
	Consistency	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
	Isolation	No	Yes	Yes	No	No	–	Yes	Yes	Yes
	Durability (data storage)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes
	Transactions	No	No	No	Yes	No	Yes	Yes	No	Yes
	Referential integrity	No	No	No	No	No	No	Yes	No	Yes
	Revision control	No	Yes	Yes	Yes	No	Yes	No	Yes	No
Indexing	Secondary Indexes	Yes	Yes	No	Yes	Yes	Yes	–	Yes	–
	Composite keys	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes	–
	Full text search	No	No	No	No	No	Yes	No	Yes	Yes
	Geospatial Indexes	Yes	No	No	No	No	Yes	–	–	Yes
	Graph support	No	No	No	No	No	Yes	No	Yes	Yes
Distribution	Horizontal scalable	Yes	Yes	Yes	Yes	Yes	Yes		Yes	No
	Replication	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes
	Replication mode	Master-Slave-Replica Replication	Master-Slave Replication	–	Master-Slave Replication	Master-Slave Replication	–	Master-Slave Replication	Multi-master replication	–
	Sharding	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
	Shared nothing architecture	Yes	Yes	Yes	Yes	Yes	–	–	Yes	–
System	Value size max.	16MB	20MB	64KB	2TB	2GB	1EB	–	64MB	
	Operating system	Cross-platform	Ubuntu Red Hat Windows Mac OS X	Cross-platform	Cross-platform	Cross-platform	NIX 32 entries Operating system	Linux *NIX Mac OS X Windows	Cross-platform	Cross-platform
	Programming language	C++	Erlang C++ C Python	Java	Java	Java	Java	C C++	Erlang	Java

Conclusions

- Computational and storage requirements of applications such as for Big Data Analytics, Business Intelligence and social networking over peta-byte datasets have pushed sql-like centralized databases to their limits.

- NoSQL is a large and expanding field, for the purposes of this paper:
  - characteristics;
  - classification;
  - comparison and evaluation of different types of NoSQL databases;
  - current state of adoption of NoSQL databases.

References

- Moniruzzaman, A.B.M. , Hossain, S.A. (2013). 'NoSQL database: new era of databases for big data analytics – classification, characteristics and comparison'. **International Journal of Database Theory and Application**. Vol. 6, No. 4, p. 1-14.
- http://techielicous.com/2011/11/02/nosql-in-the-real-world
- http://blog.neo4j.org/2010/02/top-10-ways-to-get-to-know-neo4j.html
- http://bi-bigdata.com/2013/01/13/what-is-wide-column-stores/
- http://gigaom.com/2011/07/29/couchbase-2-0-unql-sql-nosql/
- http://www.readwritewebcomimages.com
- http://hortonworks.com/blog/7-key-drivers-for-the-big-data-market/