Introduction to NoSQL

NoSQL Seminar 2012 @ TUT

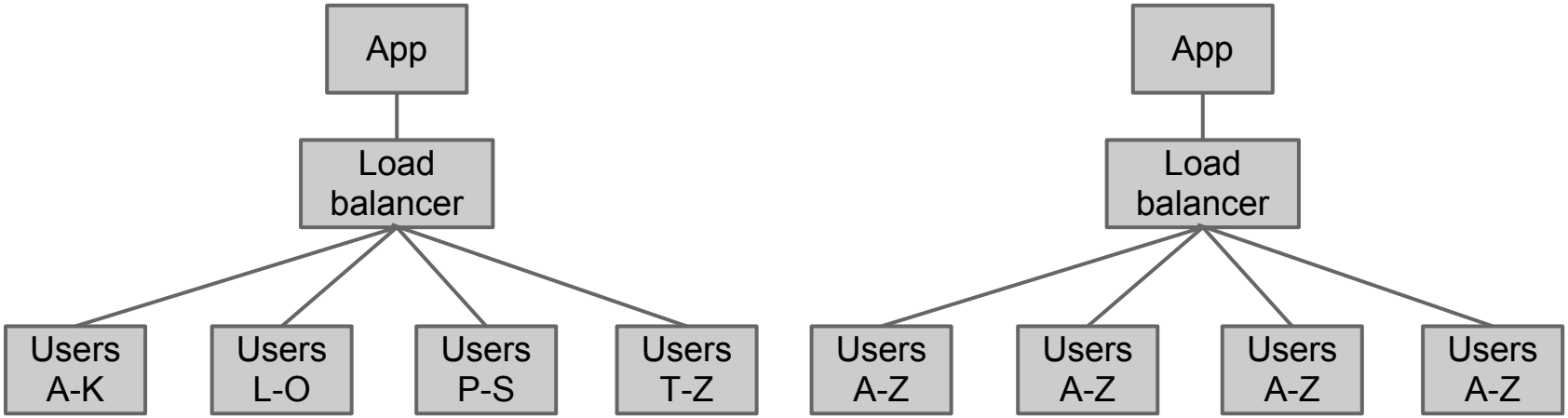
Arto Salminen

What is NoSQL?

- Class of database management systems (DBMS)
- "*Not only SQL*"
  - Does not use SQL as querying language
  - Distributed, fault-tolerant architecture
  - No fixed schema (formally described structure)
  - No joins (typical in databases operated with SQL)
    - Expensive operation for combining records from two or more tables into one set
    - Joins require strong consistency and fixed schemas
      - Lack of these makes NoSQL databases more flexible
- It's not a replacement for a RDBMS but compliments it

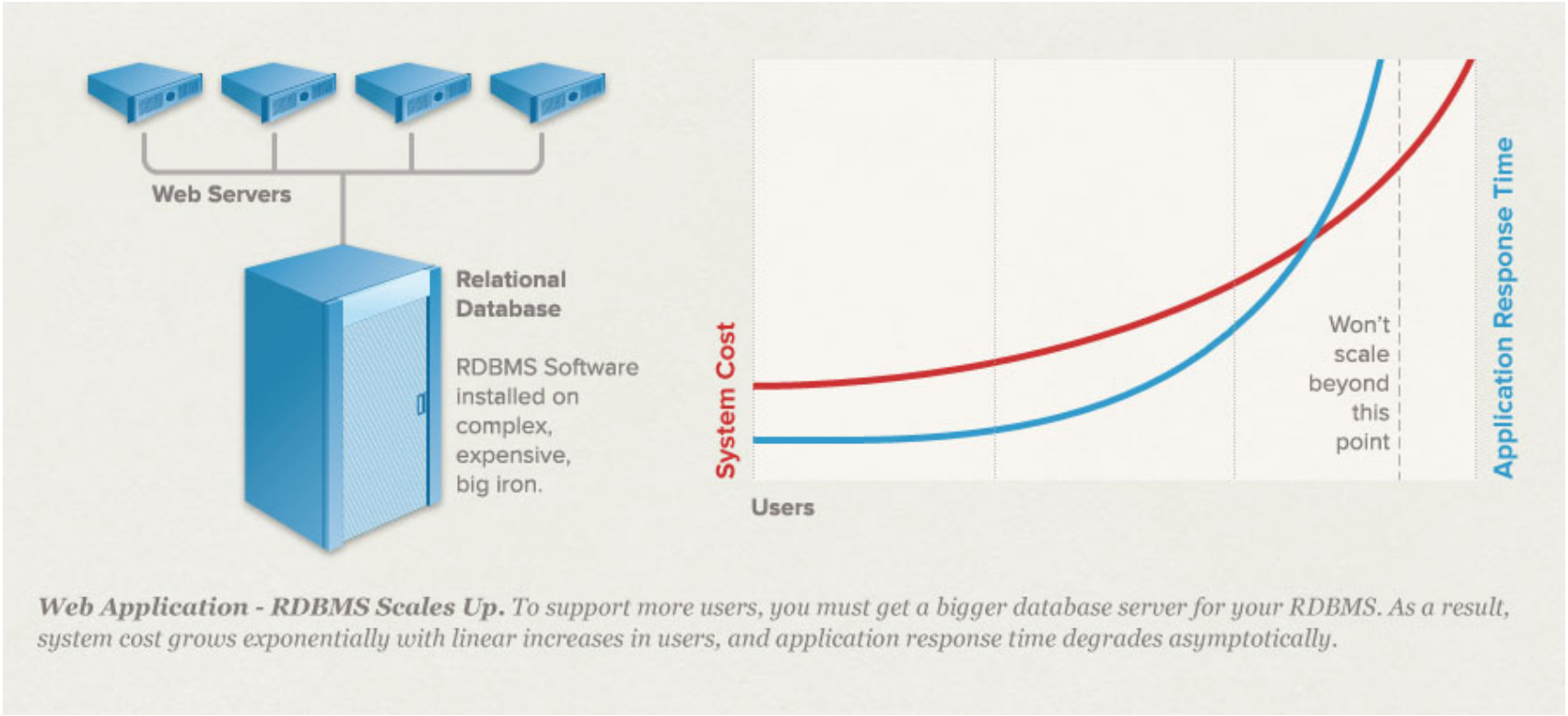
Database Scaling

- RDBMS are "scaled up" by adding hardware processing power
- NoSQL is "scaled out" by spreading the load
  - Partitioning (sharding) / replication



Relational DB Scaling

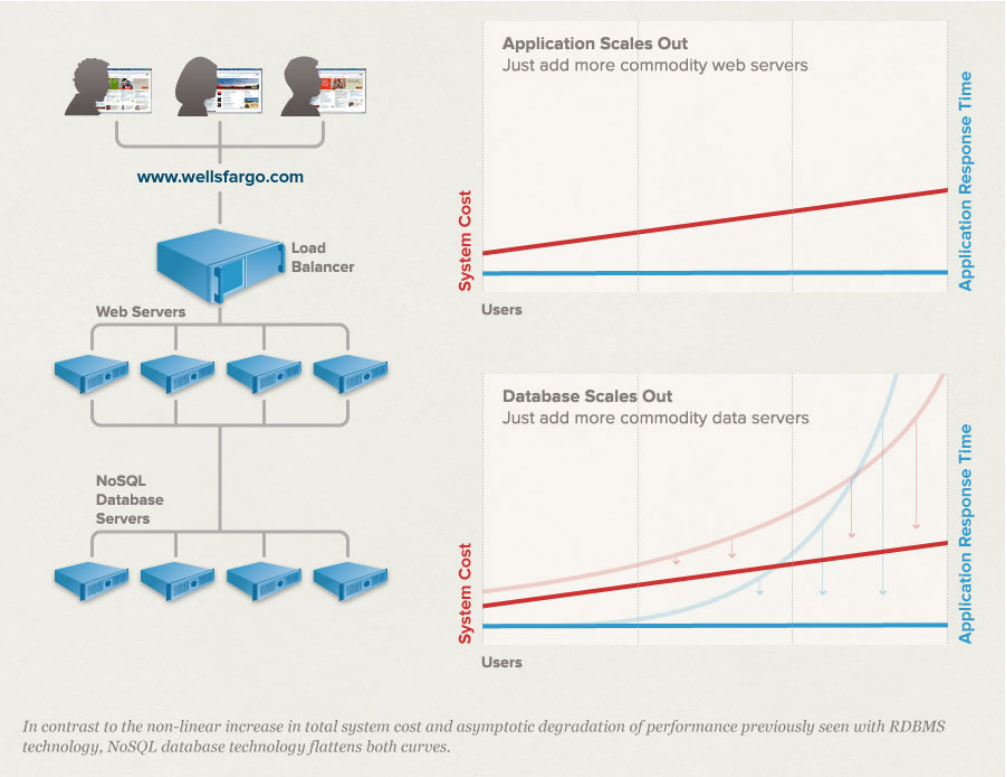
- At certain point relational database won't scale



Web Application - RDBMS Scales Up. *To support more users, you must get a bigger database server for your RDBMS. As a result, system cost grows exponentially with linear increases in users, and application response time degrades asymptotically.*

NoSQL DB Scaling

- Scaling horizontally is possible with NoSQL
- Scaling up / down is easy
  - Supports rapid production-ready prototyping
- Better handling of traffic spikes



Where NoSQL Is Used?

- Google (BigTable, LevelDB)
- LinkedIn (Voldemort)
- Facebook (Cassandra)
- Twitter (Hadoop/Hbase, FlockDB, Cassandra)
- Netflix (SimpleDB, Hadoop/HBase, Cassandra)
- CERN (CouchDB)

History of NoSQL

- MultiValue databases at TRW in **1965**.
- DBM is released by AT&T in **1979**.
- Lotus Domino released in **1989**.
- Carlo Strozzi used the term NoSQL in **1998** to name his lightweight, open-source relational database that did not expose the standard SQL interface.
- Graph database Neo4j is started in **2000**.
- Google BigTable is started in **2004**. Paper published in 2006.
- CouchDB is started in **2005**.
- The research paper on Amazon Dynamo is released in **2007**.
- The document database MongoDB is started in **2007** as a part of a open source cloud computing stack and first standalone release in 2009.
- Facebooks open sources the Cassandra project in **2008**.
- Project Voldemort started in **2008**.
- The term NoSQL was reintroduced in early **2009**.
- Some NoSQL conferences
  NoSQL Matters, NoSQL Now!, INOSA

CAP Theorem 1/2

- It is impossible for a distributed computer system to simultaneously provide all three of the following guarantees:
  - Consistency (all nodes see the same data at the same time)
  - Availability (a guarantee that every request receives a response about whether it was successful or failed)
  - Partition tolerance (the system continues to operate despite arbitrary message loss or failure of part of the system)

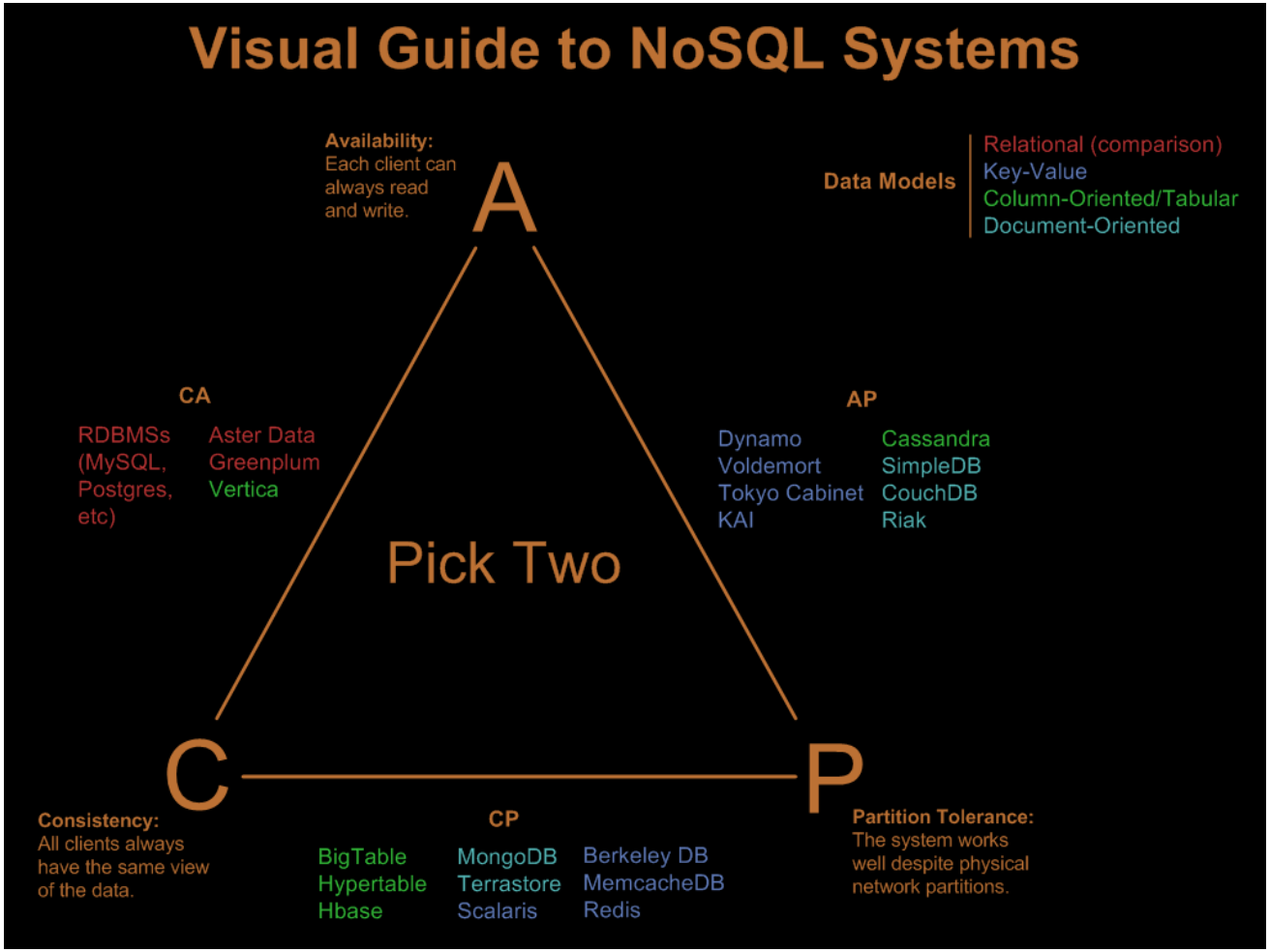
A distributed system can satisfy any two of these guarantees at the same time, but not all three.

Gilbert and Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, 2002

CAP Theorem 2/2

- In other words, CAP can be expressed as "If the network is broken, your database won't work"
  - "won't work" = down OR inconsistent
- In RDBMS we do not have P (network partitions)
  - Consistency and Availability are achieved
- In NoSQL we want to have P
  - Need to select either C or A
  - Drop A -> Accept waiting until data is consistent
  - Drop C -> Accept getting inconsistent data sometimes

NoSQL Systems and CAP



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

ACID vs BASE

Scalability and better performance of NoSQL is achieved by sacrificing ACID compatibility.

Atomic, Consistent, Isolated, Durable

NoSQL is having BASE compatibility instead.

Basically Available, Soft state,
Eventual consistency

ACID -- Requirement for SQL DBs

Atomicity. All of the operations in the transaction will complete, or none will.

Consistency. Transactions never observe or result in inconsistent data.

Isolation. The transaction will behave as if it is the only operation being performed upon the database (i.e. uncommitted transactions are isolated)

Durability. Upon completion of the transaction, the operation will not be reversed (i.e. committed transactions are permanent)

BASE -- Basically Available

- Use replication and sharding to reduce the likelihood of data unavailability and use sharding, or partitioning the data among many different storage servers, to make any remaining failures partial.
- The result is a system that is always available, even if subsets of the data become unavailable for short periods of time.

BASE and Availability

- The availability of BASE is achieved through supporting partial failures without total system failure.
- Example. If users are partitioned across five database servers, BASE design encourages crafting operations in such a way that a user database failure impacts only the 20 percent of the users on that particular host.
  - This leads to higher perceived availability of the system. Even though a single node is failing, the interface is still operational.

BASE -- Eventually Consistent

- Although applications must deal with instantaneous consistency, NoSQL systems ensure that at some future point in time the data assumes a consistent state.
- In contrast to ACID systems that enforce consistency at transaction commit, NoSQL guarantees consistency only at some undefined future time.
  - Where ACID is pessimistic and forces consistency at the end of every operation, BASE is optimistic and accepts that the database consistency will be in a state of flux.

BASE and Consistency

- As DB nodes are added while scaling up, need for synchronization arises
- If absolute consistency is required, nodes need to communicate when read/write operations are performed on a node
  - Consistency over availability -> bottleneck
- As a trade-off, "eventual consistency" is used
- Consistency is maintained later
  - Numerous approaches for keeping up "distributed consistency" are available
    - Amazon Dynamo - consistent hashing
    - CouchDB - asynchronous master-master replication
    - MongoDB - auto-sharding+replication cluster with a master server

BASE -- Soft State

- While ACID systems assume that data consistency is a hard requirement, NoSQL systems allow data to be inconsistent and relegate designing around such inconsistencies to application developers.
- In other words, soft state indicates that the state of the system may change over time, even without input.
  - This is because of the eventual consistency model (the acronym is a bit contrived).

Some breeds of NoSQL solutions

- Key-Value Stores
- Column Family Stores
- Document Databases
- Graph Databases
- In addition: Object and RDF databases as well as Tuple stores

Key-Value Stores

- Dynamo, Voldemort, Rhino DHT ...
  - DeCandia et al. "Dynamo: Amazon's Highly Available Key-value Store", 2007
- Key-Value is based on a hash table where there is a unique key and a pointer to a particular item of data.
- Mappings are usually accompanied by cache mechanisms to maximize performance.
- API is typically simple -- implementation is often complex.

Column Family Stores

- BigTable, Cassandra, HBase, Hadoop ...
  - Chang et al. "Bigtable: A Distributed Storage System for Structured Data", 2006
- Store and process very large amounts of data distributed over many machines.
  - "Petabytes of data across thousands of servers"
- Keys point to multiple columns

jim_87	**Age**	**Name**	**Gender**	**Phone**
	25	Jim	M	123456
jill_90	**Age**	**Name**	**Gender**	**Phone**
	22	Jill	F	654321

- Cassandra example

Document Databases (Stores)

- CouchDB, MongoDB, Lotus Notes, Redis ...
- *Documents* are addressed in the database via a unique key that represents that document.
- Semi-structured documents can be XML or JSON formatted, for instance.
- In addition to the key, documents can be retrieved with queries.
- Redis is sometimes referred to as *data structure server* since keys can contain strings, hashes, lists, sets and sorted sets.

Graph Databases

- Neo4J, FlockDB, GraphBase, InfoGrip, ...
- Graph Databases are built with nodes, relationships between nodes (edges) and the properties of nodes.
  - Nodes represent entities (e.g. "Bob" or "Alice").
    - Similar in nature to the objects as in object-oriented programming.
  - Properties are pertinent information related to nodes (e. g. age: 18).
  - Edges connect nodes to nodes or nodes to properties.
    - Represent the relationship between the two.
- Scaling graph DBs is problematic
  - Neo4J: cache sharding, sharding strategy heuristics

Some NoSQL Challenges

- Lack of maturity -- numerous solutions still in their beta stages
- Lack of commercial support for enterprise users
- Lack of support for data analysis
- Maintenance efforts and skills are required -- experts are hard to find

References and Material

Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos, Nabil Hachem, and Pat Helland. 2007. The end of an architectural era: (it's time for a complete rewrite). In*Proceedings of the 33rd international conference on Very large data bases* (VLDB '07). VLDB Endowment 1150-1160.

Werner Vogels. 2008. Eventually Consistent. *Queue* 6, 6 (October 2008), 14-19. DOI=10.1145/1466443.1466448

Dan Pritchett. 2008. BASE: An Acid Alternative. *Queue* 6, 3 (May 2008), 48-55. DOI=10.1145/1394127.1394128

Seth Gilbert and Nancy Lynch. 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News* 33, 2 (June 2002), 51-59. DOI=10.1145/564585.564601

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. 2006. Bigtable: a distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7* (OSDI '06), Vol. 7. USENIX Association, Berkeley, CA, USA, 15-15.

Avinash Lakshman and Prashant Malik. 2010. Cassandra: a decentralized structured storage system. *SIGOPS Oper. Syst. Rev.* 44, 2 (April 2010), 35-40. DOI=10.1145/1773912.1773922

Couch DB - The Definitive Guide, http://guide.couchdb.org/index.html

HP whitepaper: There is no free lunch with distributed data

Long list of NoSQL papers: http://nosqlsummer.org/papers

Possible Presentation Topics

NoSQL architectures

- Key-Value Store
- Graph
- Big Table (Columnar)
- Document Store
- ...

Implementations

- MongoDB
- HBase
- Cassandra
- CouchDB
- Google App Engine Datastore
- Hadoop
- BigData
- Redis
- Riak
- Neo4j
- …

Hosted services

- Freebase
- OpenLink Virtuoso
- Datastore on Google Appengine
- Amazon DynamoDB
- Cloudant Data Layer (CouchDB)
- **...**

Technologies / misc

- MapReduce
- Fault tolerance
- Taxonomy
- Challenges / Limitations
- Tools
- Use Cases
- Distributed databases
- Parallel systems
- ...