SQL vs. NoSQL

Things to consider for your next database choice

Jens Wilke
@cruftex
https://cruftex.net

headissue
software works.

cache2k
https://cache2k.org

My DB Background

- “NoSql” Java Database engine for setop box in 2000
- Two ORM Mappers / persistence layers
- Database related bachelor thesis with Prof. Bayer (inventor of the B-Tree)

Why this presentation?

Why NoSQL?

(Web) Scale

Special Purpose

and….

SQL … S***!

http://blog.schauderhaft.de/2010/02/15/why-sql-sucks/

Comments on the blog post:

“SQL is fun to learn and use, just get the basics down of Set Theory / Domain Relational Algebra, normalization and start with ANSI 92 SQL statements”

“I hate SQL. […] I can write HTML5, CSS3, JavaScript, jQuery, PHP, & MD5 encryption all day, but SQL…it's like someone decided to give a genetically-engineered gorilla the opportunity to write code.”

and….

Impedance Mismatch

SQL Relational Data Model
(tables, primary keys, foreign keys, data types)

vs.

- Objects
- JSON
- XML
- Graph/RDF

I say: SQL is Cool!

- Declarative style vs. imperative style:
  I tell what I want, the database finds out how to get it

- Well, I know, there is some legacy in its syntax, so there is in Unix `dd`

No SQL, Really?

New Databases speak SQL, too! Examples:

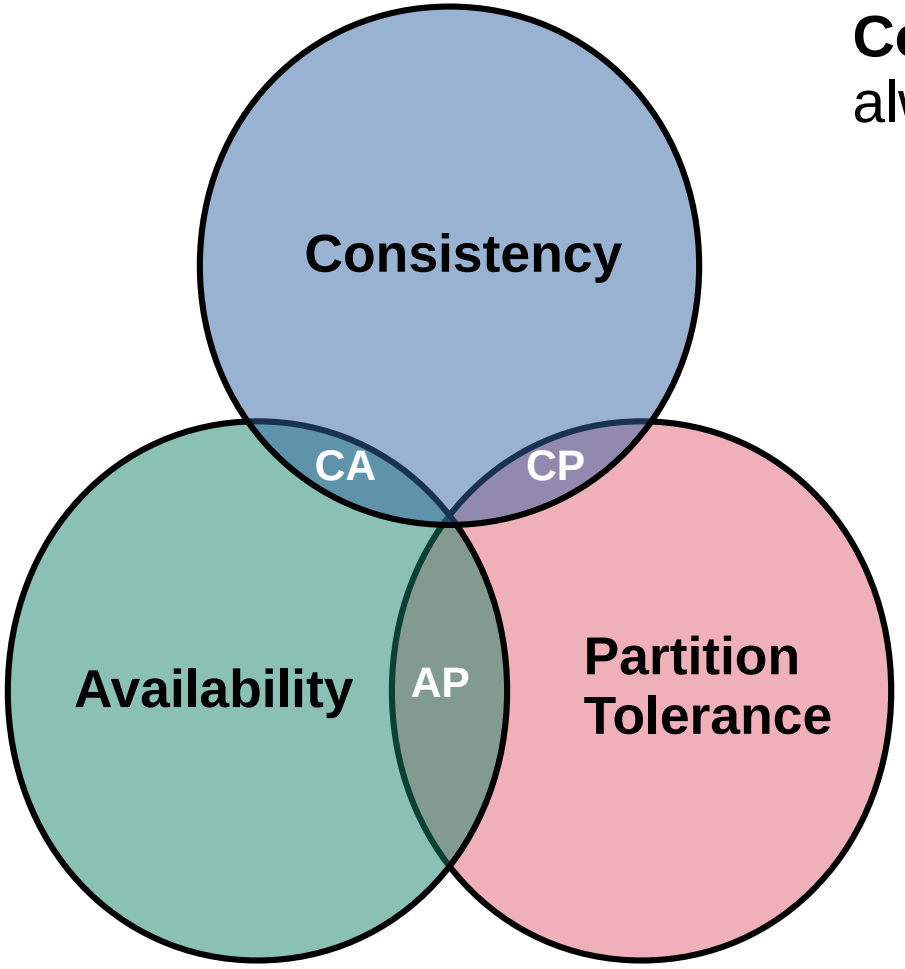
- CockroachDB
- Apache Hive
- Apache Spark

Specialized Query Languages look a lot like SQL:

- CouchDB N1QL

Fault Tolerance, Scaling
→ Distributed Systems

The CAP Theorem

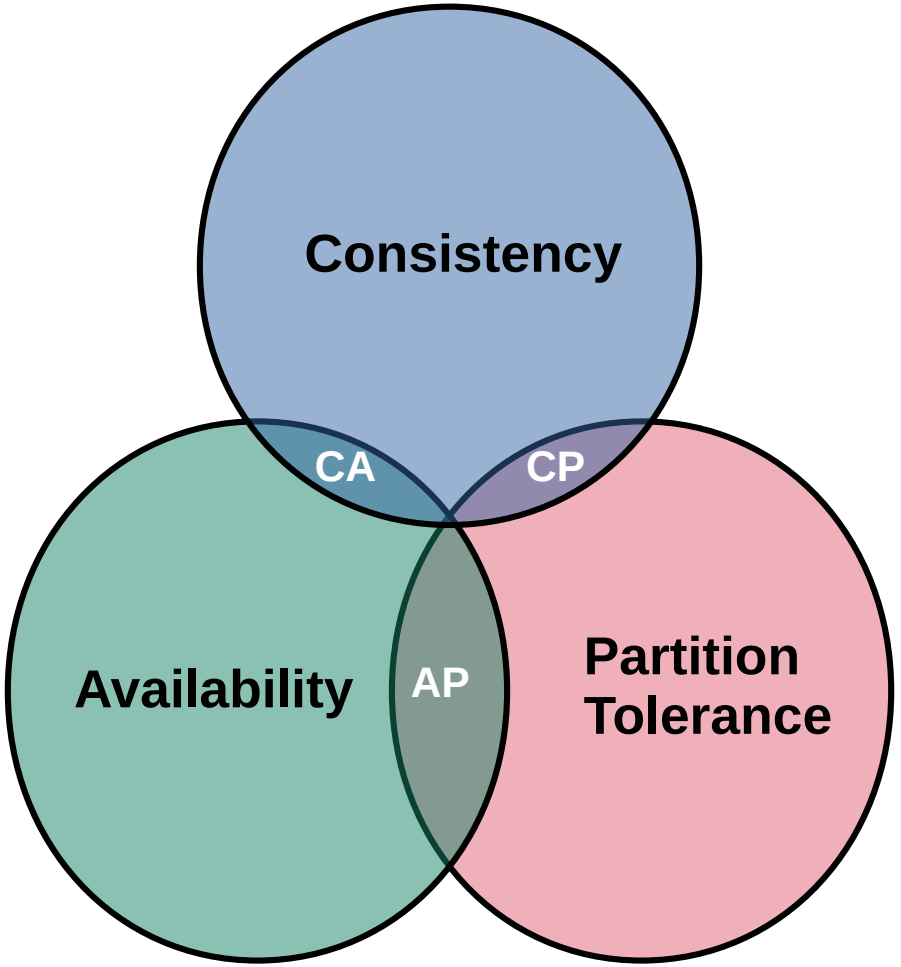


Consistency: All clients have always the same view of the data

Availability: Each client can always read and write

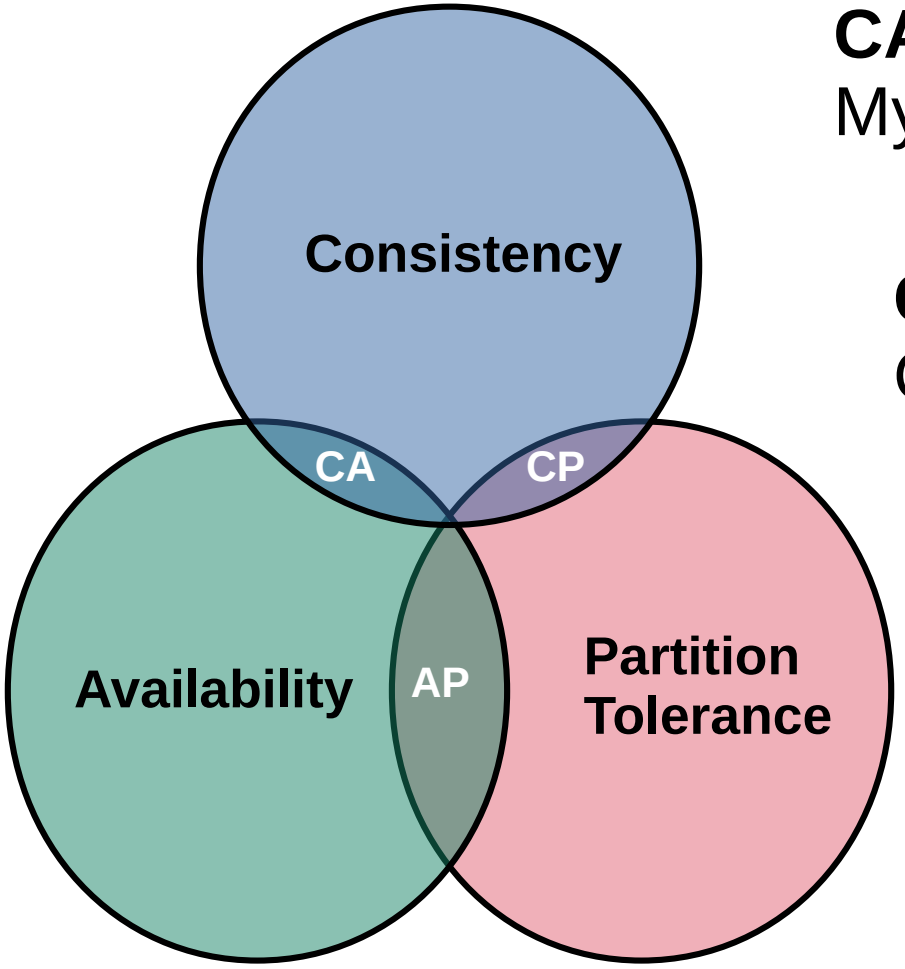
Partition Tolerance: The total system works despite network partitions

The CAP Theorem



Pick Two!

The CAP Theorem



CA:
MySQL, PostgreSQL, Oracle, …

CP:
Cassandra, CouchDB, Riak, ...

AP:
HBase, MongoDB (2010!),
Redis, ...

As of 2010 from this blog post:
http://blog.nahurst.com/visual-guide-to-nosql-systems

Details more Details and Limitations

Smart Algorithms and Theories

- CAP
- Consistent Hashing
- Paxos
- Raft
- Vector Clocks
- Calvin
- ...

Eventual Consistency

- Casual Consistency
- Read-your-write Consistency
- Session Consistency
- Monotonic Read Consistency
- Monotonic Write Consistency

ACM 2008 / 1466448 Werner Vogels (CTO Amazon)

Consistent?!

→Jepsen Reports

https://jepsen.io/

MongoDB 3.4.0-rc3 Feb2017

The Jepsen Report says:

*"[…] In this Jepsen analysis, we develop new tests which show the MongoDB v0 replication protocol is **intrinsically unsafe**, allowing the **loss** of majority-committed documents. […]"*

Good! Improvements:
→ Defined consistency guarantees (Casual Consistency) and rigorous tests

Jepsen Reports Wrap Up

- You will find scary things about any modern and distributed database
- If you don’t find anything it’s even more scary

Rumble in NoSQL Paradise

…. oops, we gotta earn money!

Riak

- Distributed Key/Value store
- Major driver: Basho Technologies
- August 2009: Initial release
- Mid 2017: Basho runs out of money
- April 2018: First community release
- License: Apache

MongoDB

- Distributed Key/Value store
- Major driver: MongoDB Inc
- Feb 2009: Initial release
- Oct 2017: MongoDB IPO
- 2018: Total Funding: $309,8M, Revenue $154.5M, Negative cash flow: $47.0M
- Oct 2018: License Change:
  Server Side Public License (SSPL)
- MongoDB drops out of Debian, Red Hat, Ubuntu dirstros, since not Open Source (OSI approved) any more

My Take on the Licensing

No matter whether OSI compliant or not: More restrictive licensing, kills the contributor community around the (core) product

Less adoption

Less thinkers, less innovation

There is a chance that the product will disappear, since a fork or community take over cannot and/or will not happen

What about the Old Guys?

PostgreSQL

- 1996: Initial Release
- Diverse Community
- JSON Support
- XML Support
- Solutions for: High Availability, Scaling

  But:

  Fragmentation, Many Add-Ons,
  No relevant Jepsen Test

  But No But: Diversity → Fragmentation

MySQL


NoSQL + SQL = MySQL 8.0
Document Store
JSON Functions
UTF8 Performance
CTEs & Window Functions
GIS Support
Atomic DDLs
Transactional Data Dictionary

Wrap Up

Directions of DBs

- Universal / Relational / Traditional / SQL (PostgreSQL, MySQL) or specialized:
- Wide Column Stores (Hbase, Cassandra)
- Document Stores (CouchDB. MongoDB)
- Key/Value Stores (Redis, memcache)
- In Memory Data Grids (hazelcast, SAP Hana)
- Time Series (OpenTSDB, influxDB)
- Graph / Triplet Stores (neo4j, OrientDB, ArangoDB)
- Search Engines (Apache SOLR, elasticsearch)
- Stream (RethinkDB)
- ....

There is no Single Reason....

- Managed Offerings / Cloud
- Speed
- Operating Cost
- Developer Productivity
- SQL
- Jepsen
- Graph
- Map/Reduce
- ...
- Community
- Maturity
- Support
- License
- Robustness
- Scalability of Processing
- Scalability of Data
- High Availability
- ....

Take Away

- Know SQL and have at least one traditional/universal/relational database in your toolbox
- If needed you can tune this database to a large extend to specific scenario (analytics, cluster/scaling, high availability, ….)
- Databases exist that can serve specific requirements better, but know there limitations and consistency model (suggestion: read and understand the latest Jepsen report on it)