Choosing the Right NoSQL Database for the Job

Zhixiong Cai, Xumeng Lyu, Edward Han, Ningwei Chu.

Catalog

- Overview & Background
- Research Design
- Evaluated Databases
- Software Quality Attributes
- Results
- Critique

Overview & Background

SQL vs NoSQL

SQL: databases with structured query language: mySQL

ACID for Atomicity, Consistency, Isolation, Durability in transactions

NoSQL: include key-value, document, column and graph stores

Less support for ACID, more availability and scalability

BASE for Basically available, Soft state, Eventually consistent

Related works

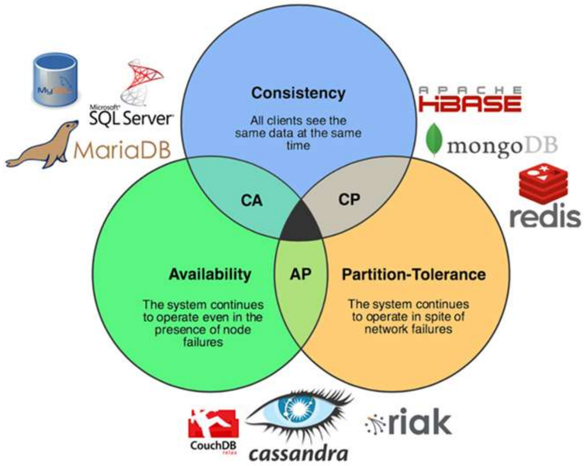
- NoSQL introduced for distributed databases, for sharing and management of distributed data, and flexibility towards unstructured data
- Performance evaluation using YCSB for read/write, latency and elasticity
- Comparison of NoSQL and Relational Database Management System
- Applicability research, which NoSQL database applies to which situation

No quality attributes evaluation and how NoSQL database fits these attributes

Main Contributions

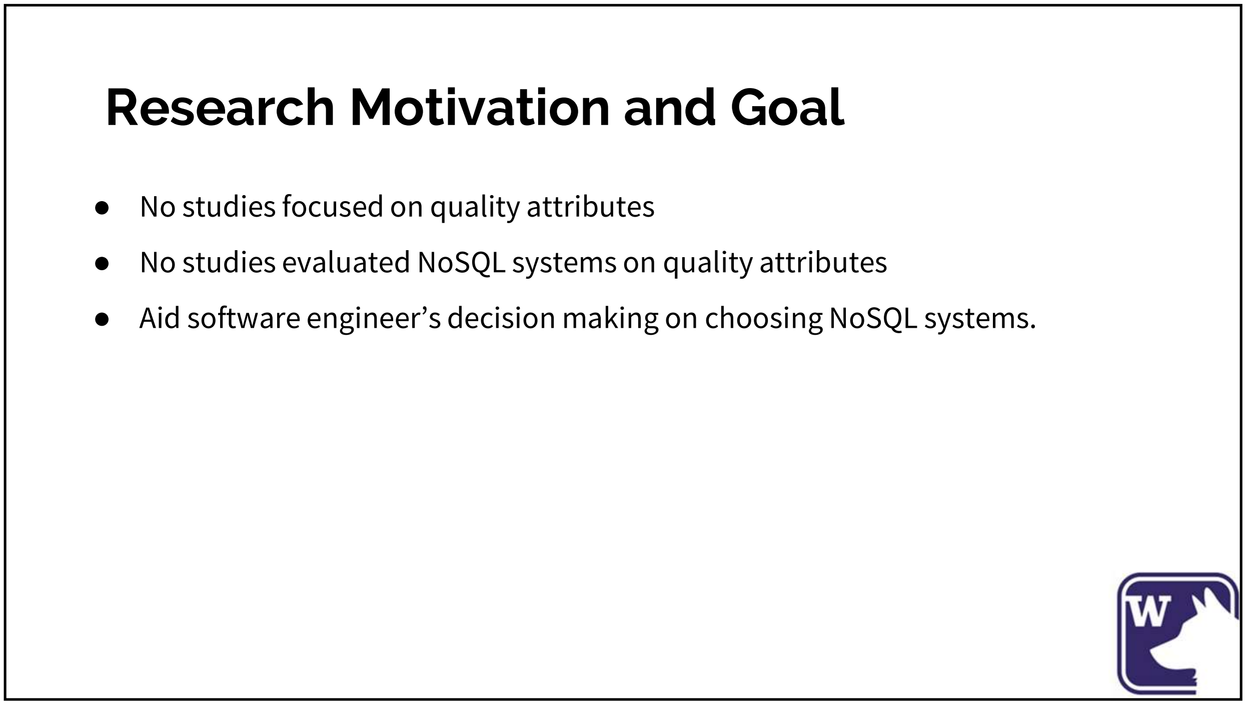
Quality-attribute oriented evaluation of NoSQL databases, including availability, consistency, durability, maintainability, performance, reliability, robustness, scalability, stabilization time & recovery time

CAP theorem


MySQL
Microsoft SQL Server
MariaDB
Consistency
All clients see the same data at the same time
APACHE HBASE
mongoDB
redis
CA
CP
AP
Availability
The system continues to operate even in the presence of node failures
Partition-Tolerance
The system continues to operate in spite of network failures
CouchDB
cassandra
riak

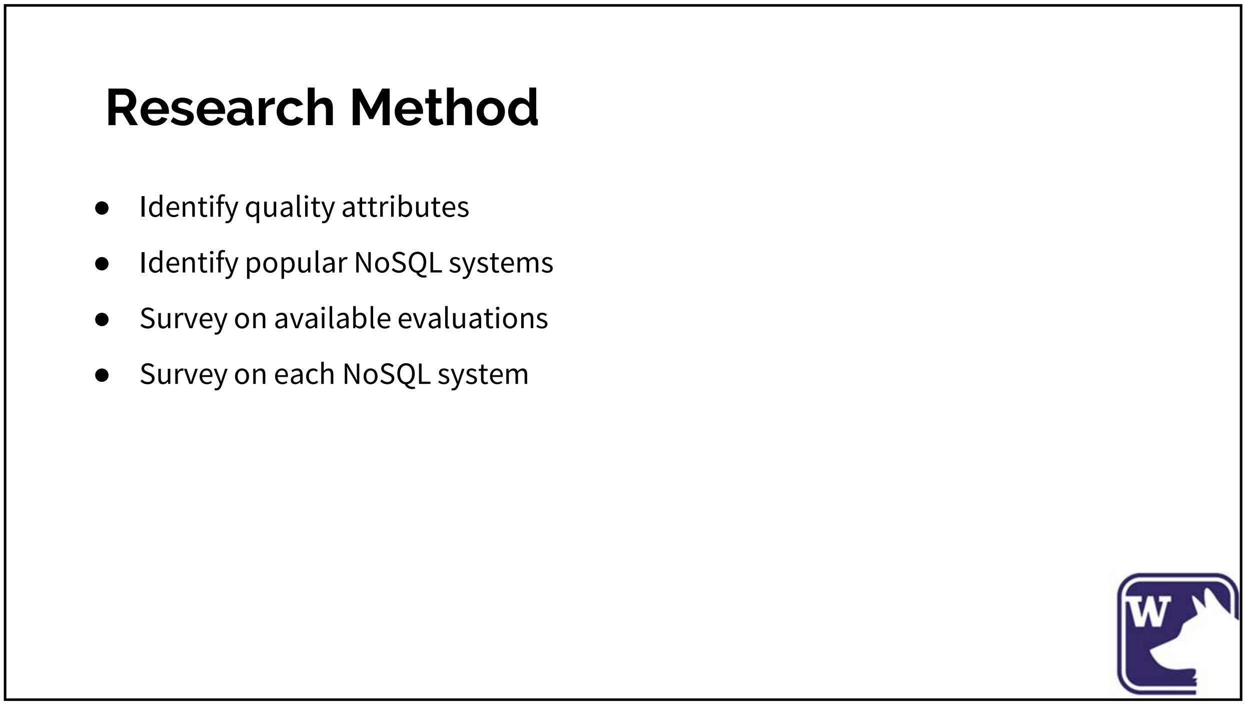
Consistency, Availability and Partition-Tolerance can't be simultaneously guaranteed in distributed systems.

Research design & Evaluated Databases



Research Motivation and Goal

- No studies focused on quality attributes
- No studies evaluated NoSQL systems on quality attributes
- Aid software engineer’s decision making on choosing NoSQL systems.



Research Method

- Identify quality attributes
- Identify popular NoSQL systems
- Survey on available evaluations
- Survey on each NoSQL system

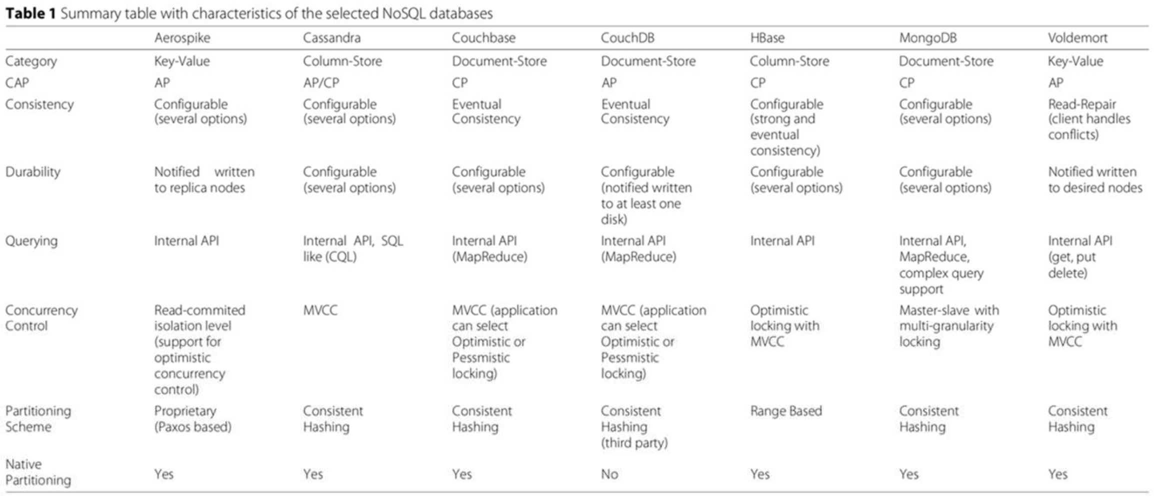
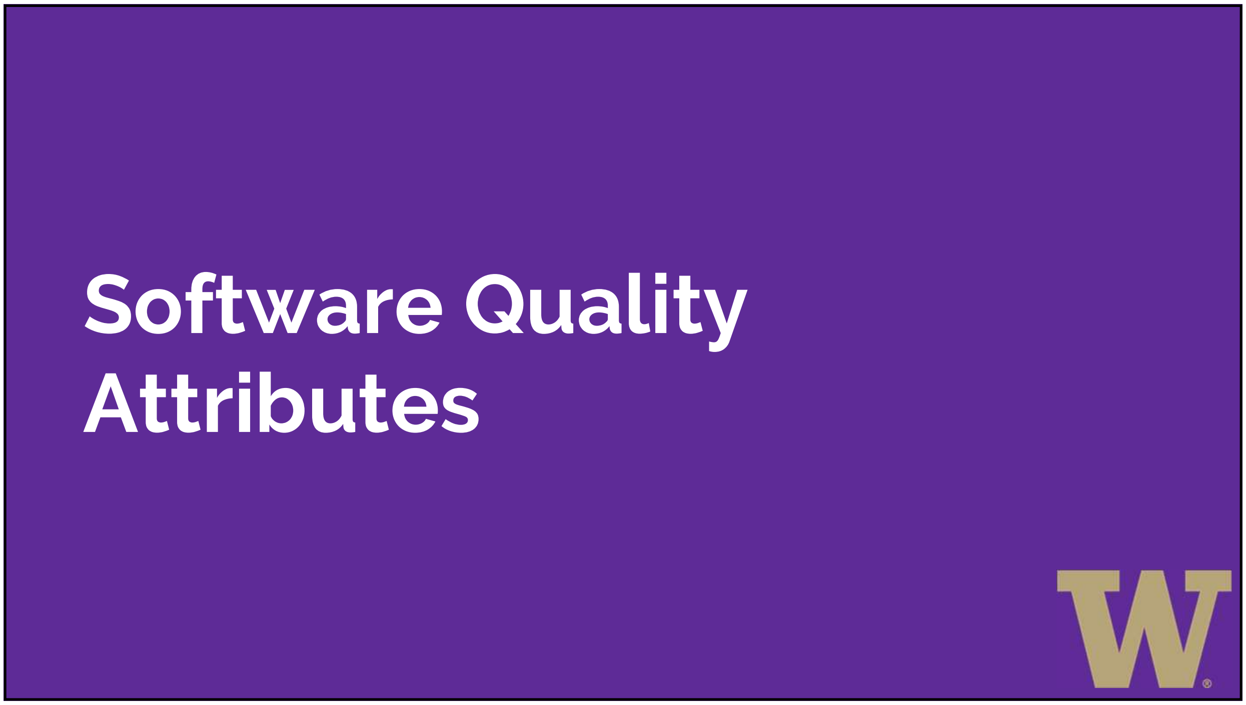


Table 1 Summary table with characteristics of the selected NoSQL databases

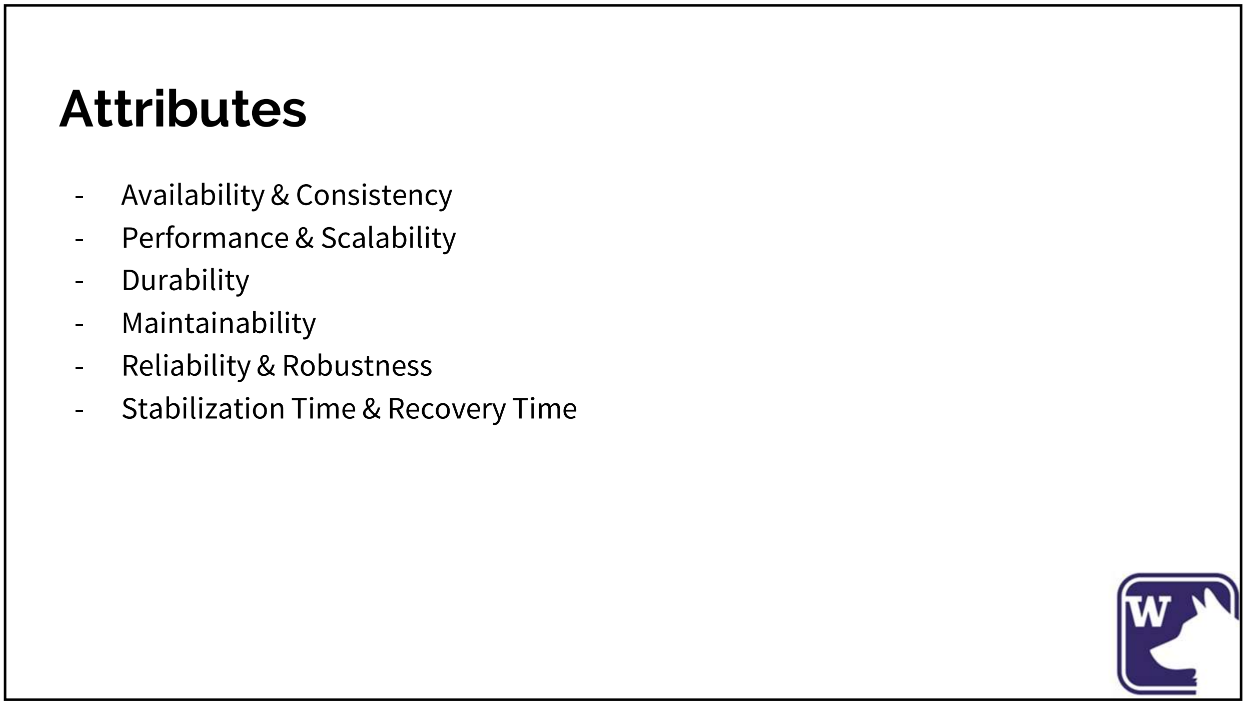
	Aerospike	Cassandra	Couchbase	CouchDB	HBase	MongoDB	Voldemort
Category	Key-Value	Column-Store	Document-Store	Document-Store	Column-Store	Document-Store	Key-Value
CAP	AP	AP/CP	CP	AP	CP	CP	AP
Consistency	Configurable (several options)	Configurable (several options)	Eventual Consistency	Eventual Consistency	Configurable (strong and eventual consistency)	Configurable (several options)	Read-Repair (client handles conflicts)
Durability	Notified written to replica nodes	Configurable (several options)	Configurable (several options)	Configurable (notified written to at least one disk)	Configurable (several options)	Configurable (several options)	Notified written to desired nodes
Querying	Internal API	Internal API, SQL like (CQL)	Internal API (MapReduce)	Internal API (MapReduce)	Internal API	Internal API, MapReduce, complex query support	Internal API (get, put delete)
Concurrency Control	Read-commited isolation level (support for optimistic concurrency control)	MVCC	MVCC (application can select Optimistic or Pessmistic locking)	MVCC (application can select Optimistic or Pessmistic locking)	Optimistic locking with MVCC	Master-slave with multi-granularity locking	Optimistic locking with MVCC
Partitioning Scheme	Proprietary (Paxos based)	Consistent Hashing	Consistent Hashing	Consistent Hashing (third party)	Range Based	Consistent Hashing	Consistent Hashing
Native Partitioning	Yes	Yes	Yes	No	Yes	Yes	Yes

Evaluated NoSQL databases

- Aerospike
- Cassandra
- CouchDB
- Couchbase
- HBase
- MongoDB
- Voldemort



Software Quality Attributes



Attributes

- Availability & Consistency
- Performance & Scalability
- Durability
- Maintainability
- Reliability & Robustness
- Stabilization Time & Recovery Time

Availability

- the percentage a system is operating correctly
- More emphasis availability instead of consistency

Consistency

- all nodes see the same data at the same time
- **Trade-offs** between availability and consistency
- some NoSQL DB solutions allow fine-tuning.

Performance

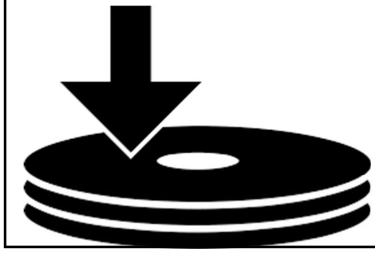
- **Write**>> Key-Value stores or Column Store databases perform better in writing
- **Read**<< document based databases are more read-oriented

Scalability

- be defined as the change in performance when new nodes are added(horizon), or hardware is improved(vertical)
- NoSQL databases have been developed specifically to target scenarios where scalability is very important.

Durability

- Durability refers to the requirement that data be valid and committed to disk after a successful transaction
- Some are inherently lack of durability (redis) 👎
- Some have good durability due to their inherent properties (MongoDB) 👍



Maintainability

- Easy to maintain?
- NoSQL systems **offer limited maintainability** when compared with traditional RDBMSs
- maintainability is moved more into the application layer and less into the database layer

Reliability

- Reliability concerns the system's **probability** of operating without failures for a given period of time
- It describe the chance, rather than proportion of a period of time, a NoSQL DB system is offline comparing to availability

Robustness

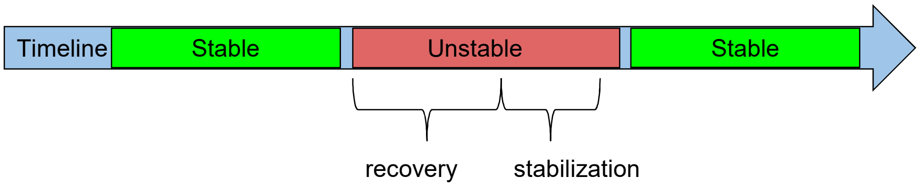
- Robustness is concerned with the ability of the database to cope with errors during execution
- crashes are "faded out" by appropriate replication and consensus algorithms 👍
- lack of code maturity and extensiv👎esting

Recovery Time

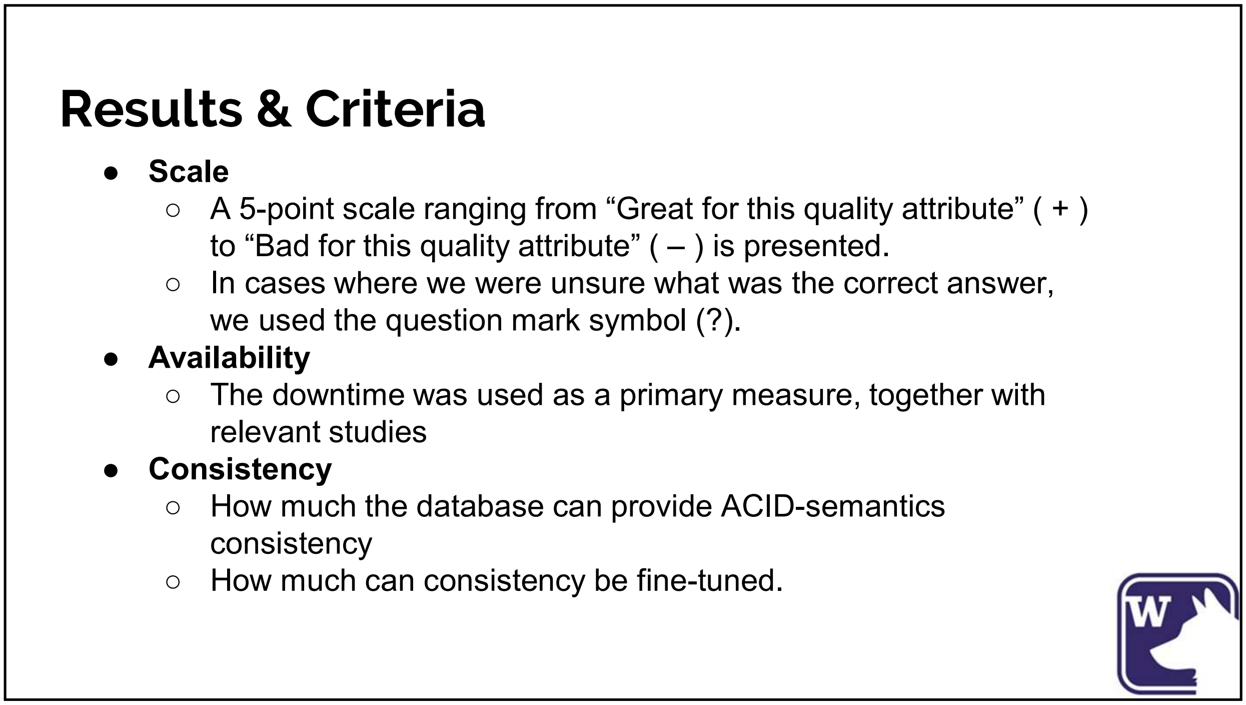
- time it takes for several NoSQL systems to recover from a node failure

Stabilization Time

- time it takes for the system to stabilize when that node rejoins the cluster

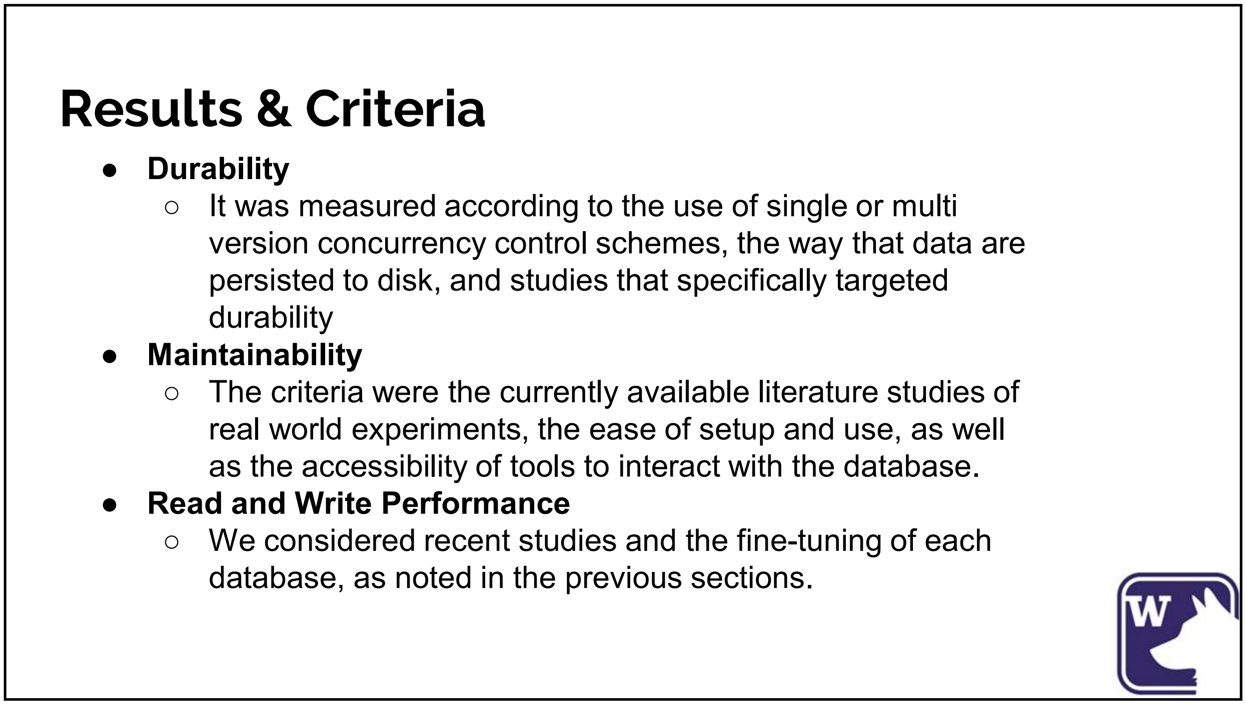


Results & Critique



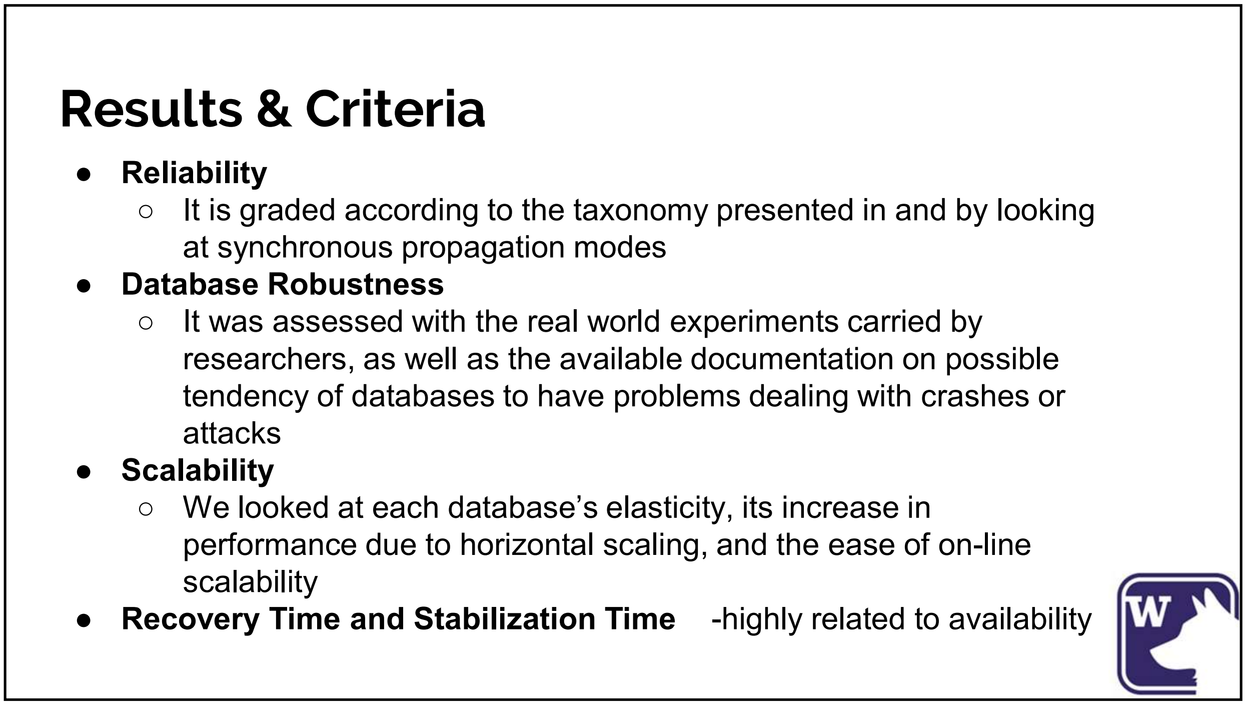
Results & Criteria

- **Scale**
  - A 5-point scale ranging from “Great for this quality attribute” (+) to “Bad for this quality attribute” (–) is presented.
  - In cases where we were unsure what was the correct answer, we used the question mark symbol (?).
- **Availability**
  - The downtime was used as a primary measure, together with relevant studies
- **Consistency**
  - How much the database can provide ACID-semantics consistency
  - How much can consistency be fine-tuned.



Results & Criteria

- **Durability**
  - It was measured according to the use of single or multi version concurrency control schemes, the way that data are persisted to disk, and studies that specifically targeted durability
- **Maintainability**
  - The criteria were the currently available literature studies of real world experiments, the ease of setup and use, as well as the accessibility of tools to interact with the database.
- **Read and Write Performance**
  - We considered recent studies and the fine-tuning of each database, as noted in the previous sections.



Results & Criteria

- **Reliability**
  - It is graded according to the taxonomy presented in and by looking at synchronous propagation modes
- **Database Robustness**
  - It was assessed with the real world experiments carried by researchers, as well as the available documentation on possible tendency of databases to have problems dealing with crashes or attacks
- **Scalability**
  - We looked at each database’s elasticity, its increase in performance due to horizontal scaling, and the ease of on-line scalability
- **Recovery Time and Stabilization Time** -highly related to availability

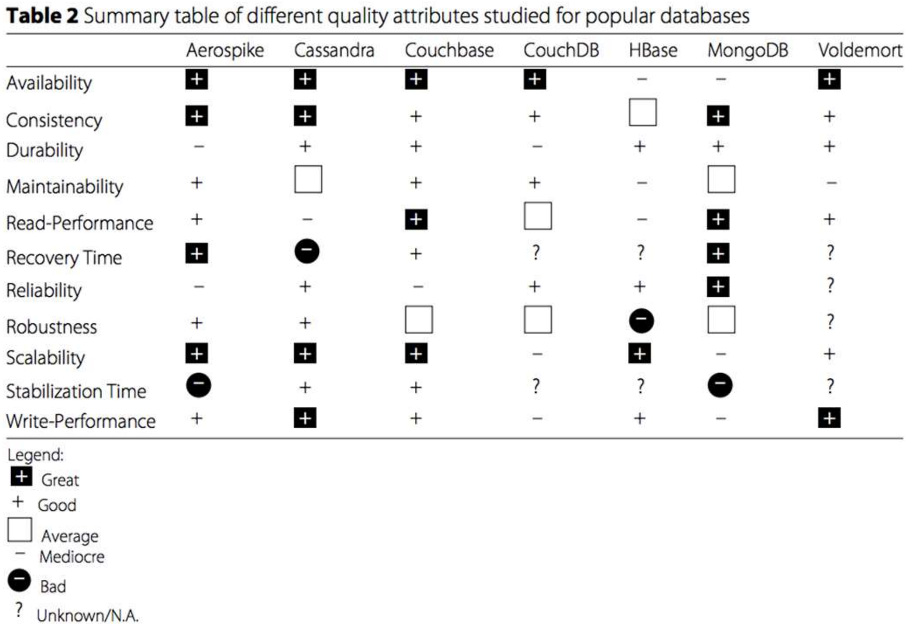


Table 2 Summary table of different quality attributes studied for popular databases

	Aerospike	Cassandra	Couchbase	CouchDB	HBase	MongoDB	Voldemort
Availability	⊞	⊞	⊞	⊞	−	−	⊞
Consistency	⊞	⊞	+	+	□	⊞	+
Durability	−	+	+	−	+	+	+
Maintainability	+	□	+	+	−	□	−
Read-Performance	+	−	⊞	□	−	⊞	+
Recovery Time	⊞	⊖	+	?	?	⊞	?
Reliability	−	+	−	+	+	⊞	?
Robustness	+	+	□	□	⊖	□	?
Scalability	⊞	⊞	⊞	−	⊞	−	+
Stabilization Time	⊖	+	+	?	?	⊖	?
Write-Performance	+	⊞	+	−	+	−	⊞

Legend:

⊞ Great

+ Good

□ Average

− Mediocre

⊖ Bad

? Unknown/N.A.

Conclusion & Critique

- **Time-based Perspective to the Evolution of NoSQL Research**
  - four clearly distinct periods:
  - 1) Database type characterization (where NoSQL was in its infancy and researchers tried to categorize databases into different sets);
  - 2) Performance evaluations, with the advent of YCSB and a surge in NoSQL popularity;
  - 3) Real-world scenarios and criticism to some interpretations of the CAP theorem;
  - 4) An even bigger focus on applicability and a reinvigorated focus on the validation of benchmarking software.
- There is still not enough information to verify how suited each nonrelational database is in a specific scenario or system.

Conclusion & Critique

- **NoSQL is still an in-development field, with many questions and a shortage of definite answers.**
    - There is also a lack of studies which focus on use-case oriented scenarios or software engineering quality attributes.
    - Its technology is ever-increasing and ever-changing, rendering even recent benchmarks and performance evaluations obsolete.
    - All of these reasons make it difficult to find the best pick for each of the quality attributes we chose in this work, as well as others.
- The summary table we presented makes it clear that there is a current need for a broad study of quality attributes in order to better understand the NoSQL ecosystem, and it would be interesting to conduct research in this domain.

Thanks!