Real-world consistency explained

A short introduction

Uwe Friedrichsen (codecentric AG) – EA Connect Day – Berlin, 6. October 2016

@ufried

Uwe Friedrichsen | uwe.friedrichsen@codecentric.de | http://slideshare.net/ufried | http://ufried.tumblr.com

About this talk ...


Past

RDBMS

ACID

RDBMS

- "One database to rule them all"
- Good all-rounder
  - Rich schema
  - Rich access patterns
- Designed for scarce resources
  - Storage, CPU, Backup are expensive
  - Network is slow
- Shared database
  - Replication was expensive
  - Licenses were expensive
  - Operations were expensive
  - Easy integration model
  - "Strange attractor"
  - Accidental central integration hub
  - Data spaghetti

ACID

- Atomicity
- Consistency
- Isolation
- Durability

- Great programming model
  - No temporal inconsistencies
  - No anomalies
  - Easy to reason about

- But reality often is different!
  - ACID does not necessarily mean "serializability"
  - Databases often run at lower consistency levels
  - Anomalies happen
  - Most developers are not aware of it

ANSI SQL

Anomalies

- Dirty write (P0): w1[x]…w2[x]…(c1 or a1)
- Dirty read (P1): w1[x]…r2[x]…(c1 or a1)
- Fuzzy read (P2): r1[x]…w2[x]…(c1 or a1)
- Phantom read (P3): r1[P]…w2[y in P]…(c1 or a1)

Isolation levels

	Dirty write	*Dirty read*	*Fuzzy read*	*Phantom read*
Read uncommitted	Not possible	Possible	Possible	Possible
Read committed	Not possible	Not possible	Possible	Possible
Repeatable read	Not possible	Not possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible	Not possible

See [Ber+1995]

Extended anomaly model

- Dirty write (P0): w1[x]...w2[x]...(c1 or a1)
- Dirty read (P1): w1[x]...r2[x]...(c1 or a1)
- Lost update (P4): r1[x]...w2[x]...w1[x]...c1
- Lost cursor u. (P4C): rc1[x]...w2[x]...wc1[x]...c1.
- Fuzzy read (P2): r1[x]...w2[x]...(c1 or a1)
- Phantom read (P3): r1[P]...w2[y in P]...(c1 or a1)
- Read skew (A5A): r1[x]...w2[x]...w2[y]...c2...r1[y]...(c1 or a1)
- Write skew (A5B): r1[x]...r2[y]...w1[y]...w2[x]...(c1 and c2 occur)

see [Ber+1995]

Extended isolation level model

Isolation level	*Dirty write*	*Dirty read*	*Cursor lost update*	*Lost update*	*Fuzzy read*	*Phantom read*	*Read skew*	*Write skew*
Read uncommitted	Not possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible
Read committed	Not possible	Not possible	Possible	Possible	Possible	Possible	Possible	Possible
Cursor stability	Not possible	Not possible	Not possible	Sometimes possible	Sometimes possible	Possible	Possible	Sometimes possible
Repeatable read	Not possible	Not possible	Not possible	Not possible	Not possible	Possible	Not possible	Not possible
Snapshot	Not possible	Not possible	Not possible	Not possible	Not possible	Sometimes possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible	Not possible	Not possible	Not possible	Not possible	Not possible

See [Ber+1995]

Default & maximum isolation levels

Database	Default	Maximum
Actian Ingres 10.0/10S [1]	S	S
Aerospike [2]	RC	RC
Akiban Persistit [3]	SI	SI
Clustrix CLX 4100 [4]	RR	RR
Greenplum 4.1 [8]	RC	S
IBM DB2 10 for z/OS [5]	CS	S
IBM Informix 11.50 [9]	Depends	S
MySQL 5.6 [12]	RR	S
MemSQL 1b [10]	RC	RC
MS SQL Server 2012 [11]	RC	S
NuoDB [13]	CR	CR
Oracle 11g [14]	RC	SI
Oracle Berkeley DB [7]	S	S
Oracle Berkeley DB JE [6]	RR	S
Postgres 9.2.2 [15]	RC	S
SAP HANA [16]	RC	SI
ScaleDB 1.02 [17]	RC	RC
VoltDB [18]	S	S
RC: read committed, RR: repeatable read, SI: snapshot isolation, S: serializability, CS: cursor stability, CR: consistent read		

Table 1: Default and maximum isolation levels for ACID and NewSQL databases as of January 2013.

See [Bai+2013a]

Wrap-up – Past

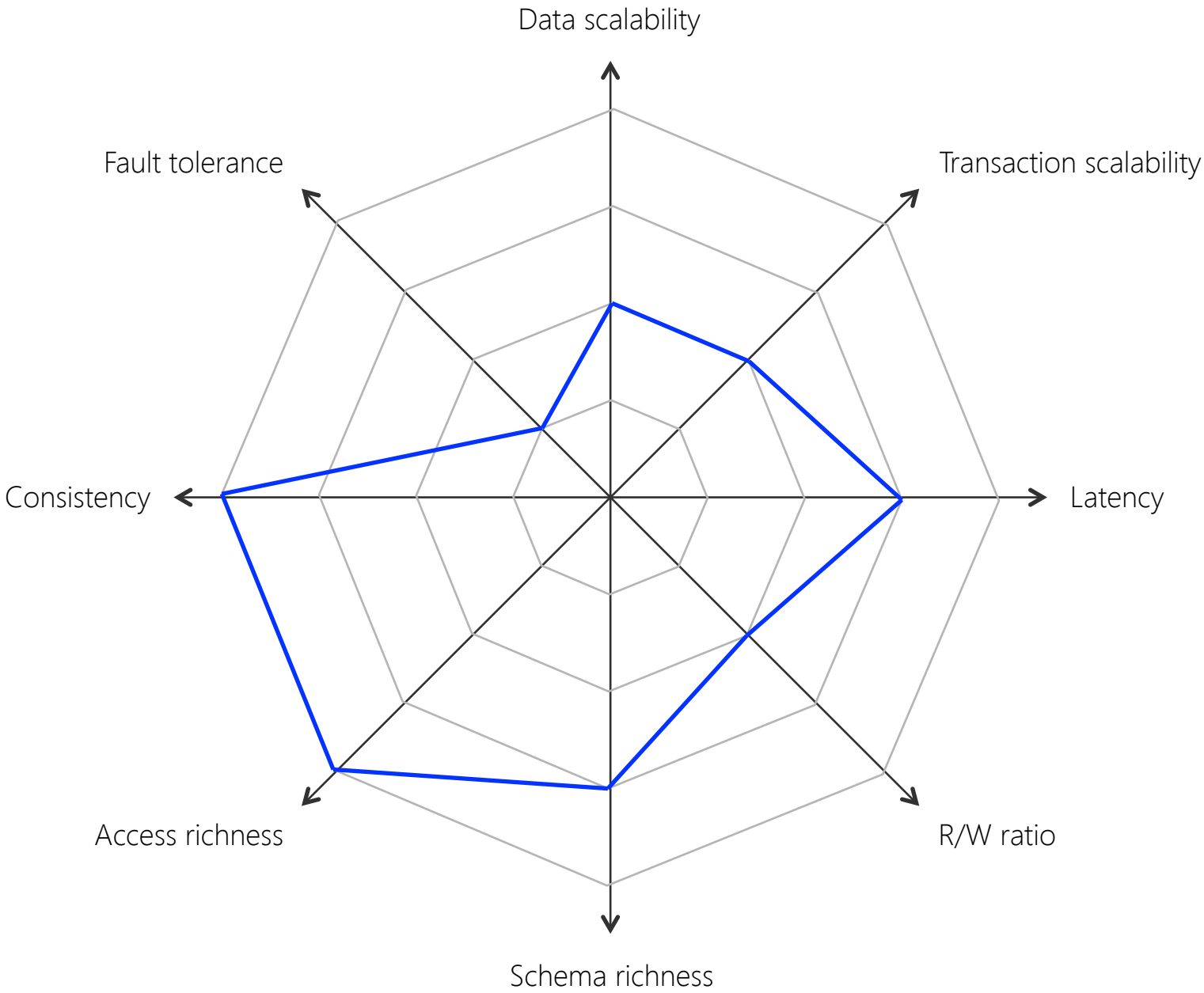
- The relational model is a good tradeoff
- ACID makes a developer's life easy
- Yet, we often live (unknowingly) with less than serializability

And if you go NoSQL …

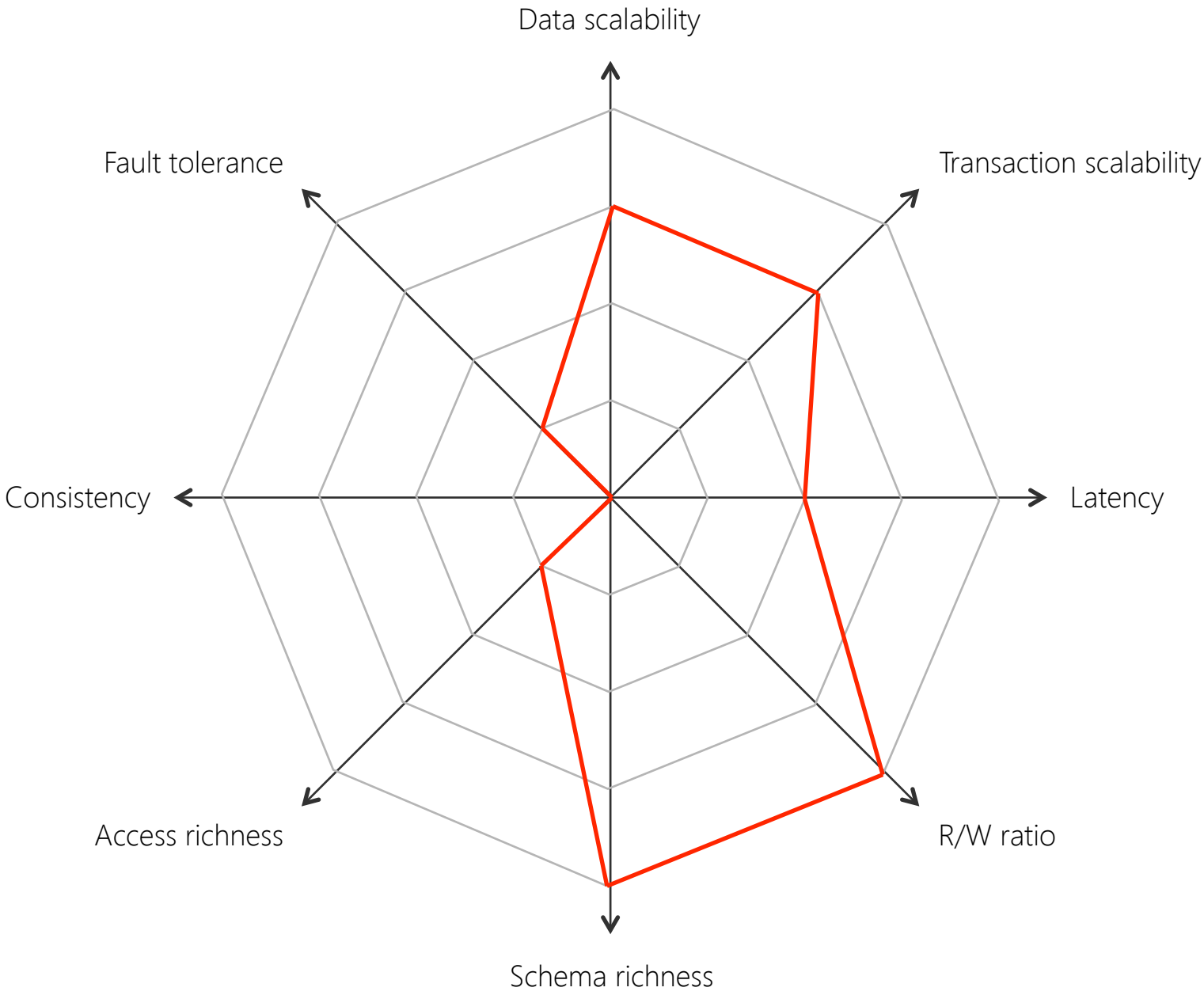
The 8 dimensions of storage

- Data Scalability (amount of data)
- Transaction Scalability (access rate)
- Latency (response time considering scalability)
- Read/Write Ratio (variability of r/w mix considering scalability)
- Schema Richness (variability of data model)
- Access Richness (variability of access patterns)
- Consistency (data consistency guarantees)
- Fault Tolerance (ability to handle failures gracefully)

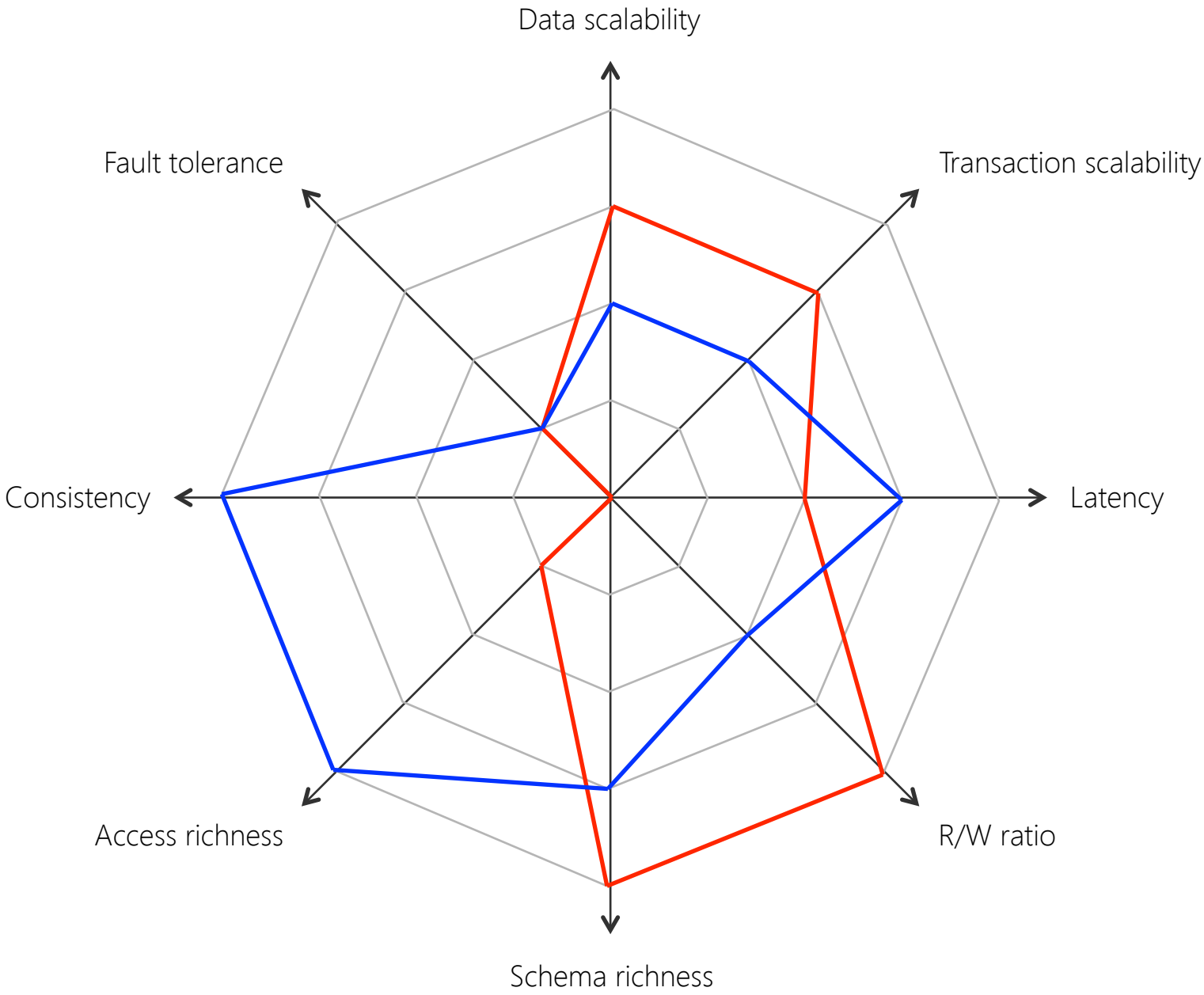
Relational database



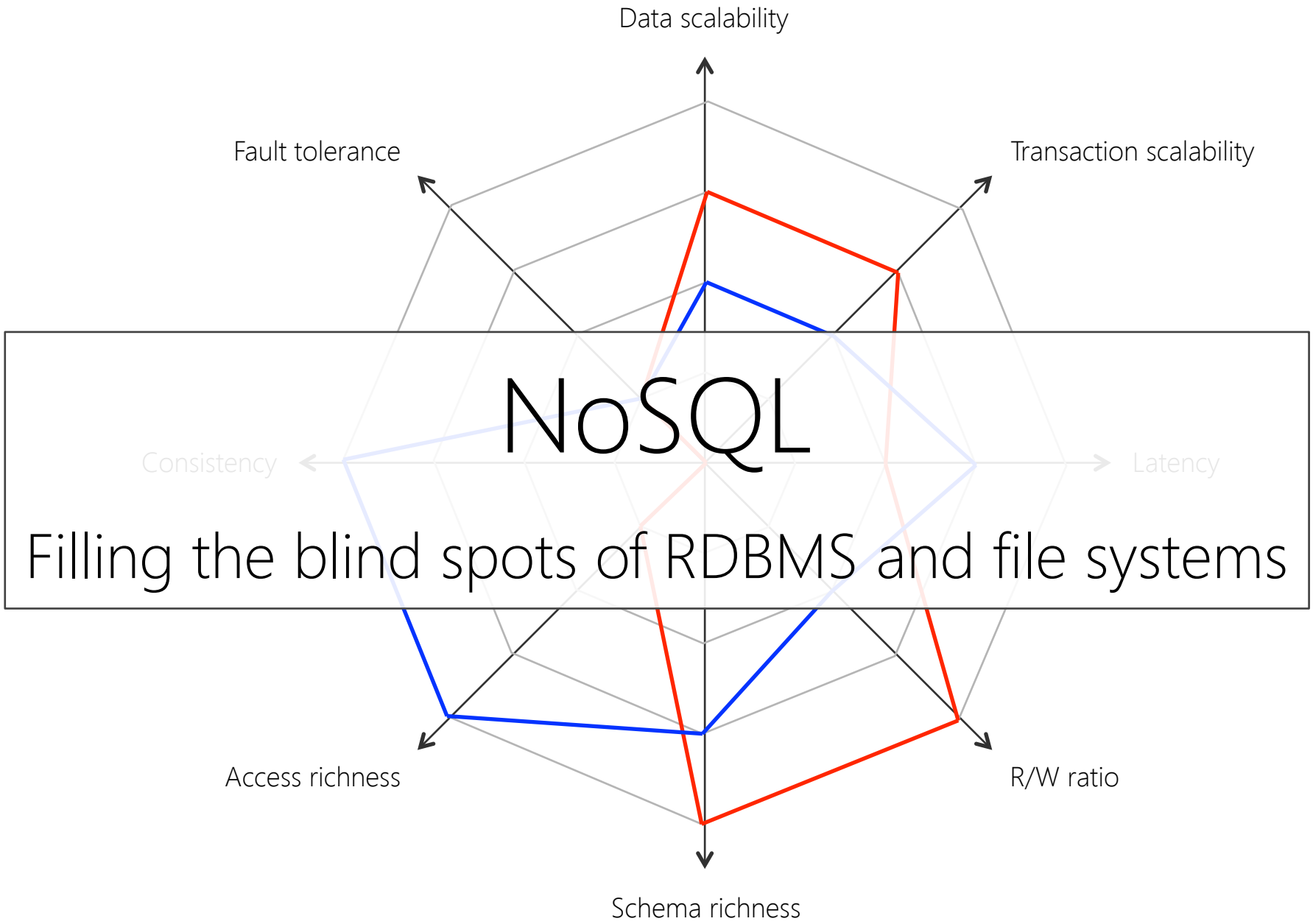
File system


Data scalability
Transaction scalability
Latency
R/W ratio
Schema richness
Access richness
Consistency
Fault tolerance

RDBMS & file system



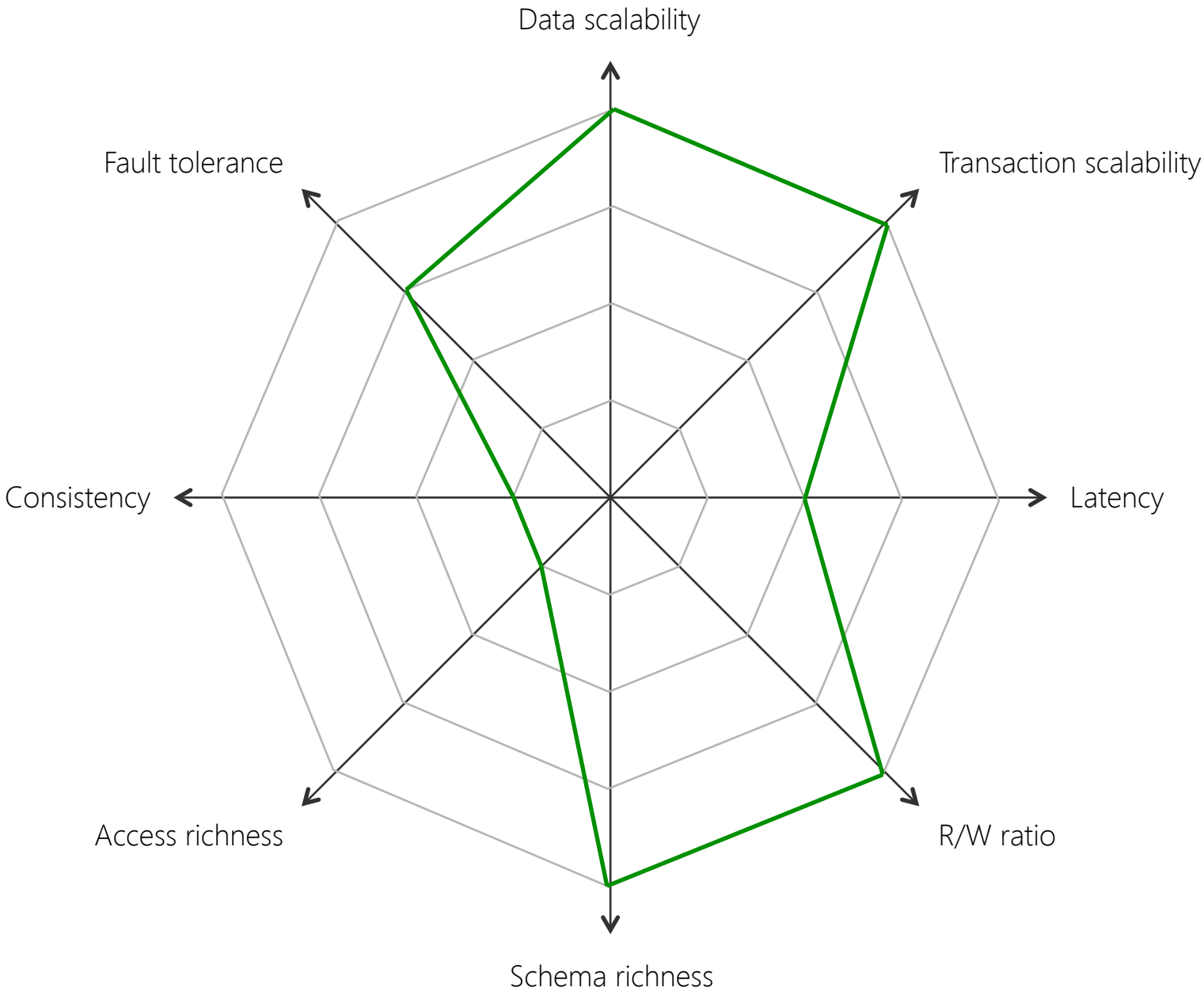
RDBMS & file system



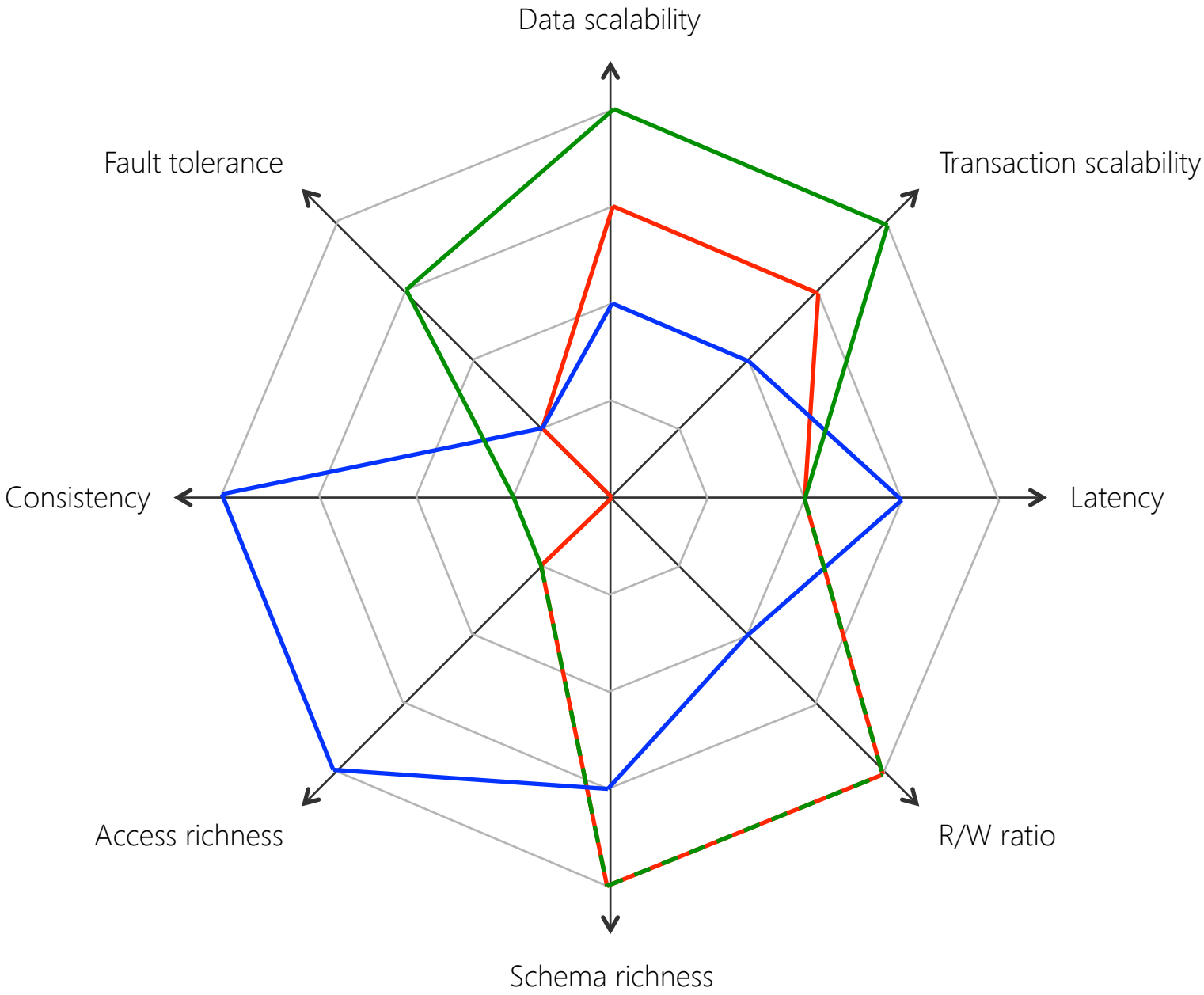
NoSQL

Filling the blind spots of RDBMS and file systems

Cassandra

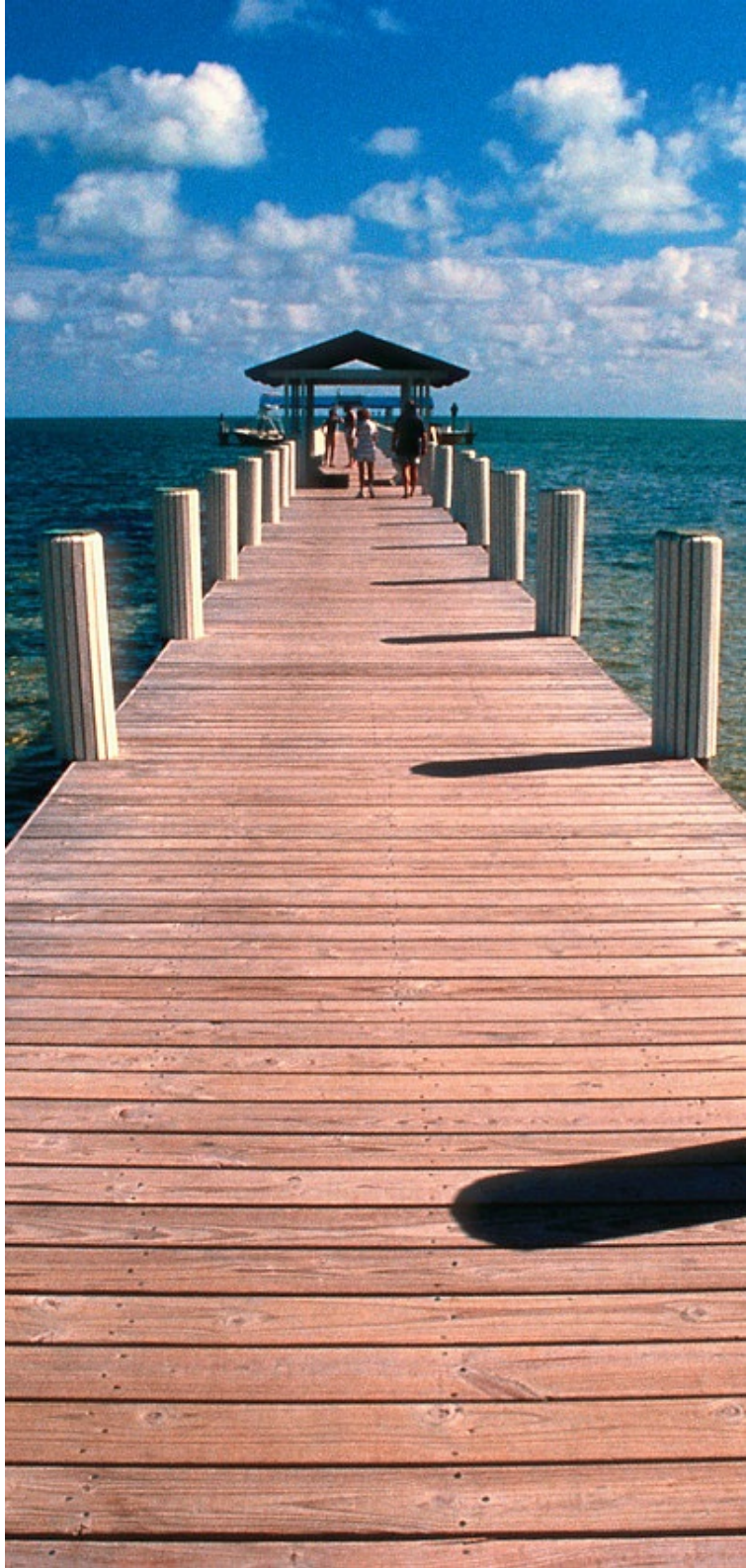

Data scalability
Transaction scalability
Latency
R/W ratio
Schema richness
Access richness
Consistency
Fault tolerance

RDBMS-FS-Cassandra



Wrap-up

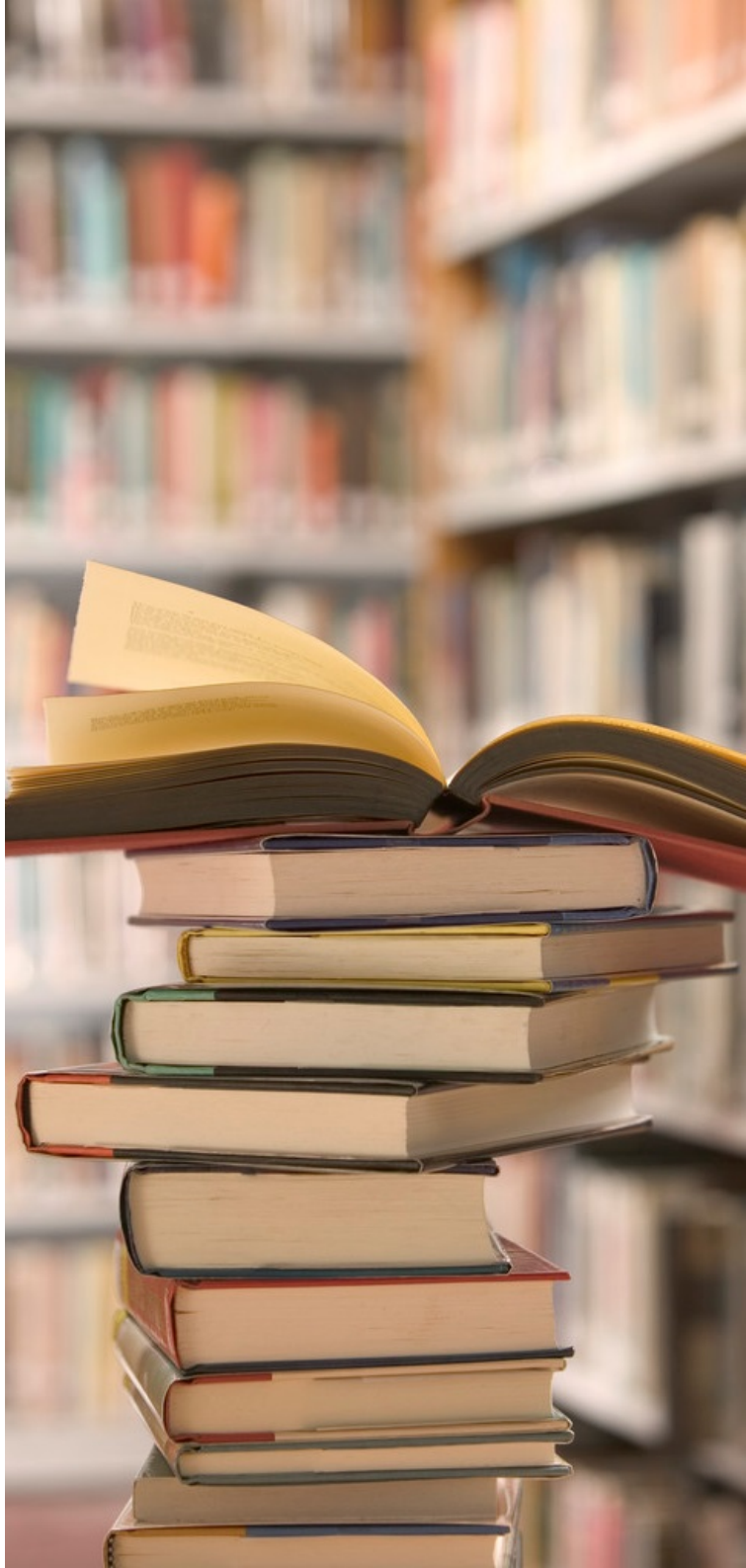
- ACID offers a great programming model
- But ACID often does not mean serializability
- Most databases do not use serializability
- Understand the trade-offs of NoSQL DBs



And if you want the whole nine yards …

The full story

- On YouTube
  - https://www.youtube.com/watch?v=WG3xKyldSK0
- On slideshare
  - http://www.slideshare.net/ufried/realworld-consistency-explained



References

[Bai+2013a] Peter Bailis, Alan Fekete, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica, "HAT, not CAP: Towards Highly Available Transactions", HotOS 2013

[Ber+1995] Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, Patrick O'Neil, "A Critique of ANSI SQL Isolation Levels", Microsoft Research, Technical Report MSR-TR-95-51, June 1995

@ufried

Uwe Friedrichsen | uwe.friedrichsen@codecentric.de | http://slideshare.net/ufried | http://ufried.tumblr.com