



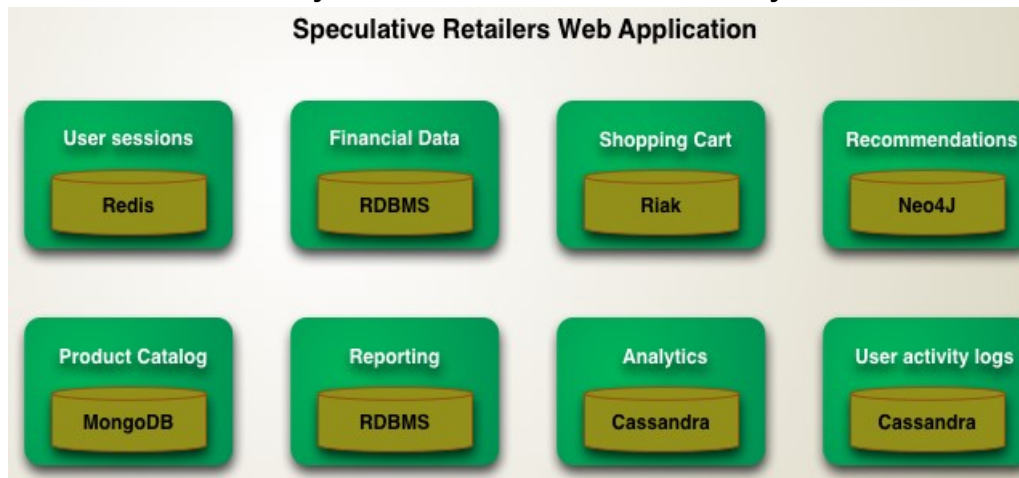
Parallele og distribuerte databaser – del IV

Today's topics:

- Multi-model databases overview
- Example benchmark
- ArangoDB
- Demo / Hands-on

Polyglot Persistence

- Polyglot persistence: a variety of different database systems for different kinds of data



Picture taken from <https://martinfowler.com/bliki/PolyglotPersistence.html>

- Complexity cost
 - Each data storage mechanism introduces a new interface to be learned for each new data storage mechanism
 - Storage is usually a performance bottleneck
 - Multiple data silos
 - More complicated deployment, more frequent upgrades
 - Data consistency and duplication issues

Multi-model databases

- A database that consists of different data storage mechanisms (e.g. relational, document, key/value, graph database):
 - All in one database engine
 - With a unifying query language and API
 - That cover all data models and even allow for mixing them in a single query
- Next evolution of NoSQL technologies
- Multi-model vs Multi-modal
 - Multi-model: relational, key-value, document, graph, tree, etc.
 - Multi-modal: video, audio, image, text, etc.

Examples

- ArangoDB – document (JSON), graph, key-value
- CouchBase – relational (SQL), document
- CrateDB – relational (SQL), document (Lucene)
- MarkLogic – document (XML and JSON), graph (RDF with OWL/RDFS), text, geospatial, binary, SQL
- OrientDB – document (JSON), graph, key-value, text, geospatial, binary, reactive, SQL
- Datastax – key-value, tabular, graph
- Virtuoso – RDF, XML, relational
- ...

Hot topics in multi-model databases

- Benchmarking
- Extensions of existing query languages
- Cross-model schema languages and evolution
- Query processing
 - Cross-model complex joins
 - New index structures
- Model mapping
- Cross-model transaction and consistency

Example benchmark

- Based on ArangoDB blog post <https://www.arangodb.com/2015/10/benchmark-postgresql-mongodb-arangodb/>
- Focus on:



the *multi-model* NoSQL DB

- Multidatastore (document, graph, og key-value)
- Cluster distribusjon
- AQL spørrespråk
- ACID



- Gir og muligheter som en documentstore base
- Arving
- Meget likt spørre språk som «normal» SQL
- Støtte for typesetting

Comparison criteria

Based on ArangoDB blog post <https://www.arangodb.com/2015/10/benchmark-postgresql-mongodb-arangodb/>

- Single read: single document read of profiles (100.000 documents)
- Single write: single document writes of profile (100.000 documents)
- Aggregation: ad-hoc aggregation over a single collection (1,632,803 documents)
- Neighbors: finding (distinct) direct neighbors plus the neighbors of the neighbors, returning IDs (for 1,000 vertices)
- Neighbors with data: finding (distinct) direct neighbors plus the neighbors of the neighbors and return their profiles (for 100 vertices)

Comparison criteria (cont')

Based on ArangoDB blog post <https://www.arangodb.com/2015/10/benchmark-postgresql-mongodb-arangodb/>

- Neighbors with data: finding (distinct) direct neighbors plus the neighbors of the neighbors and return their profiles (for 100 vertices)
- Shortes path: finding 40 shortest paths (in a highly connected social graph)
 - This answers the question how close to each other two people are in the social network

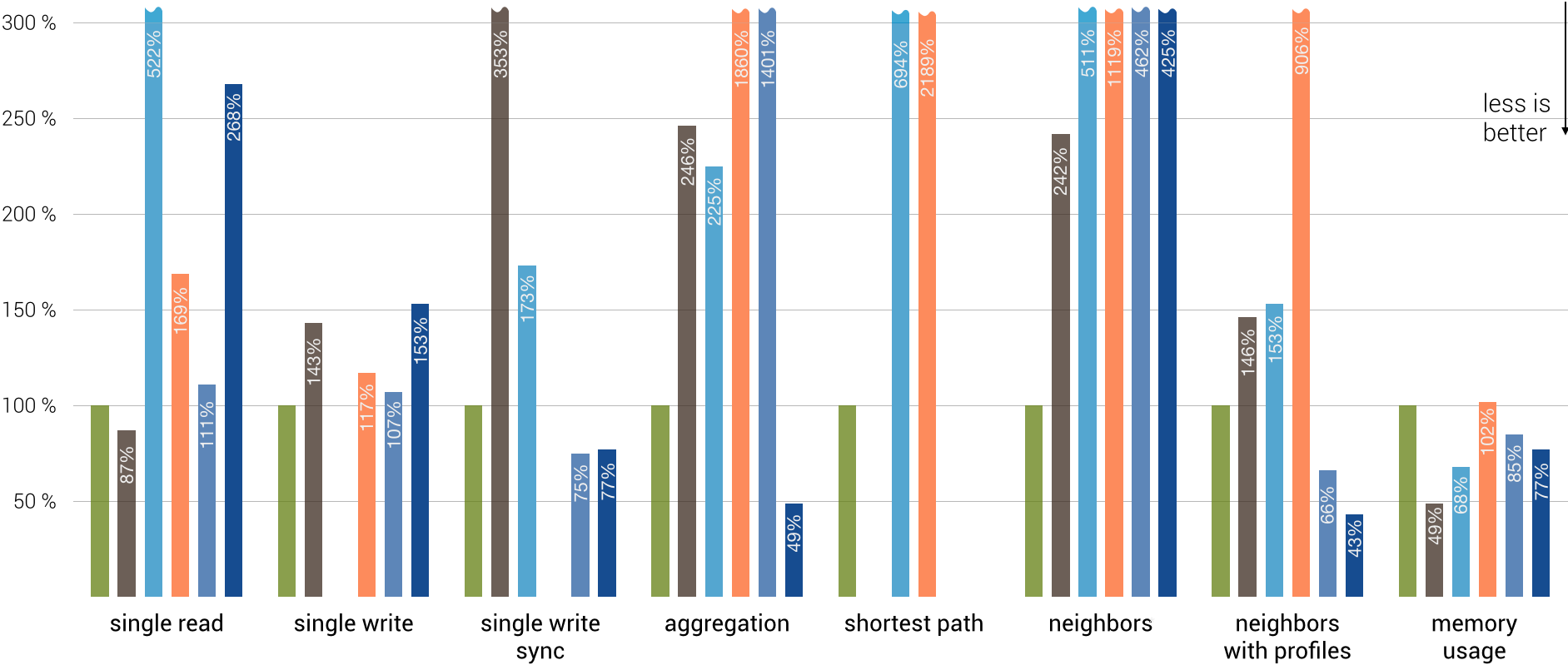
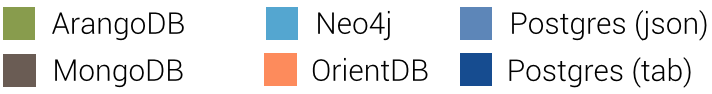
Benchmarking tests

Based on ArangoDB blog post <https://www.arangodb.com/2015/10/benchmark-postgresql-mongodb-arangodb/>

- For the tests run the workloads 5 times, averaging the results
- Each test starts with an individual warm-up phase that allows databases to load data in memory and every test iteration starts from scratch to prevent a cache comparison test

NoSQL Performance Test

ArangoDB, Postgres, MongoDB, Neo4j and OrientDB



*) neighbors and neighbors of neighbors (distinct)

Database versions: ArangoDB 2.7 RC2, OrientDB 2.2 alpha, MongoDB 3.0.6, Neo4J 2.3 M3, PostgreSQL 9.4.4

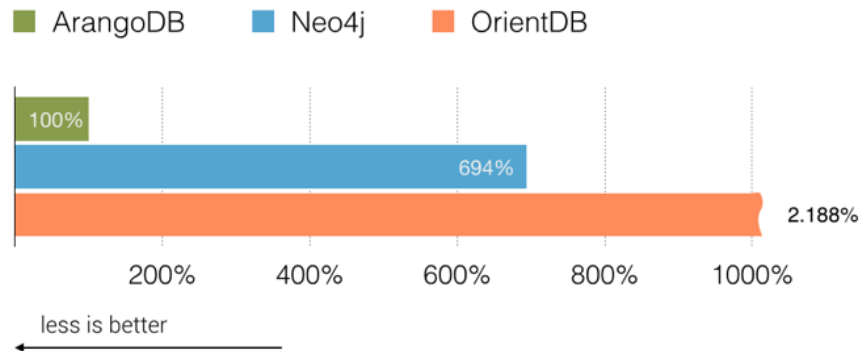
Weinberger 2015-10-13 (r207)

Based on ArangoDB blog post <https://www.arangodb.com/2015/10/benchmark-postgresql-mongodb-arangodb/>

	single read	single write	single write sync	aggregation (ad-hoc query)	shortest path	neighbors* distinct 1st+2nd deg	neighbors with profile data	memory usage
ArangoDB	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
2.7.0 RC2	16.962 sec.	20.530 sec.	91.816 sec.	1.250 sec.	0.061 sec.	0.464 sec.	4.327 sec.	13.142 GB
MongoDB	86.76 %	143.11 %	352.83 %	246.08 %		242.26 %	146.32 %	49.20 %
3.0.6 - WiredTiger	14.716	29.380	323.953	3.077		1.124	6.331	6.466 GB
Neo4j	522.31 %		173.02 %	225.37 %	694.41 %	511.43 %	152.59 %	68.47 %
2.3 M3	88.593		158.860	2.818	0.422	2.372	6.602	8.998 GB
OrientDB	168.55 %	116.96 %		1860.12 %	2188.82 %	1119.4 %	905.72 %	102.06 %
2.2 alpha	28.590	24.011		23.259	1.331	5.192	39.187	13.413 GB
PostgreSQL (json)	111.07 %	106.92 %	75.12 %	1401.22 %		461.91 %	66.21 %	85.21 %
9.4.4 - json	18.840	21.951	68.972	17.521		2.142	2.865	11.199 GB
PostgreSQL (tab.)	267.93 %	153.27 %	76.87 %	48.77 %		425.08 %	43.34 %	77.40 %
9.4.4 - tabular	45.445	31.465	70.581	0.61		1.972	1.875	10.172 GB

*) distance 1 and the distance 2 neighbors, each of them once.

Weinberger 2015-10-13 (r207)



Weinberger 2015-10-13 (r207)

ArangoDB

<https://www.arangodb.com/>

Hva er ArangoDB?

- Multi-model database
- Document store
- Key / value store
- Graph



Hvilke fordeler?

- Skrevet i C++
- Singel
- Cluser
- Mixed
- **CAP** - CP
- Behandle forskjellige data
- Beste fra de 3 NoSQL løsningene
- Distrubusjon
- eComerse
- BigData

Innebygd funksjonalitet

- Async
- Foxx js framework
- Arangosh
- AQL – spørrespråket

Importtyper

- Data import
- Data export
- JSON, csv, tab separerte filer
- JSON-array, JSON object per linje
- Evt. bruk "*--separator*" for å bestemme csv separator

Lagring i ArangoDB

- Collections
 - Lagrer json objekter med <key, value>
- Edge collections
 - Json object med ”_from / __to” key
 - Kan og inneholde verdier
- Mulig å lagre RDF data

AQL – ArangoDB “SQL”

<https://docs.arangodb.com/latest/AQL/index.html>

- Et språk for både graf, dokument, og key / value
- FOR – FILTER – RETURN
- LET – COLLECT

Nye funksjoner

(i nye release 3.2)

- Pregel computing model
 - Supersteps
- Pregel algoritmer (graf algoritmer)

Graf funksjoner

- PageRank
- Weakly Connected Components
- Strongly Connected Components
- HITS (hubs and authorities)
- Single-Source Shortest Path
- Community Detection via Label Propagation
- Vertex Centrality measures
 - Closeness Centrality via Effective Closeness
 - Betweenness Centrality via LineRank

DEMO

- Docker
- Import
- GUI
- Collections / edge collections
- Grafvisning
- Query

Referanser

- <https://www.arangodb.com>
- <https://www.arangodb.com/2017/03/alpha3-arangodb-3-2-support-distributed-graph-processing/>
- <https://www.arangodb.com/arangodb-white-papers/>