ArangoDB

A native multi-model NoSQL database

Fall 2019

Bala Vignesh B

COMP 630: Issues in Database Management

Instructor: Jisheng Pang

Presentation

- What is Arango and why it is called Native Multi-Modal
- Key Features
- ArangoDB Ranking
- Platform Availability
- ArangoDB Query Language (AQL)
- SQL VS AQL
- Graph Basics
- Performance
- JSON REST API - Foxx
- Drivers Available

ArangoDB

ArangoDB is a native multi-model database which means it provides the capabilities of key-value store , graph database, document database. ArangoDB is *native*, because we can use and combine all data models, even in a single query.

Key Features

- Open Source
- Simple Installation on various platform
- Flexible data modeling
- Powerful query language (AQL)
- Replication and Sharding
- JavaScript for all
- Join between two Collections

DB-Engine.com Ranking

Comparing with 355 Database, Yes it is actually 355 different database available.

Overall	61
Document stores	11
Graph DBMS	4
Key-value stores	10
Search engines	7

Platforms


docker


kubernetes


CentOS


fedora


redhat.
LINUX


debian


ubuntu


openSUSE


suse


archlinux.


Windows Azure


amazon
web services

AQL

ArangoDB Query Language (AQL) is a simple human readable language similar to Structured Query Language (SQL). AQL is a data manipulation language (DML), not a data definition language (DDL).

Terminology

SQL	AQL
database	database
table	collection
row	document
column	attribute
table joins	collection joins
primary key	primary key (automatically present on _key attribute)
index	index

CRUD

INSERT

SQL

```
INSERT INTO Characters (name, age)
    VALUES ("Jaime", 36);
```

AQL

```
INSERT { name: "Jaime", age: 36 }
INTO Characters
```

UPDATE

SQL

```
UPDATE Characters SET age = 37
    WHERE id = 102;
```

AQL

```
UPDATE { _key: "102" } WITH { age: 37 }
    IN Characters
```

SELECT

SQL

```
SELECT *  FROM Characters;
```

AQL

```
FOR character IN Characters
    RETURN character
```

DELETE

SQL

```
DELETE FROM Characters
    WHERE name=”Ned”;
```

AQL

```
FOR character IN Characters
    FILTER character.name == "Ned"
    REMOVE character IN Characters
```

Advanced

GROUP BY ORDER BY

SQL

```
SELECT age, COUNT(*) AS number
FROM Characters
    GROUP BY age ORDER BY age DESC;
```

AQL

```
FOR character IN Characters
    COLLECT age = character.age
    WITH COUNT INTO number
    SORT age DESC
RETURN {
    age: age,
    count: number
}
```

JOIN

SQL

```
SELECT * from  Characters character, Traits traits
    WHERE character. traits = traits.id
    and traits.lang=”en";
```

AQL

```
FOR character IN Characters
    RETURN MERGE(character,
    { traits: DOCUMENT ("Traits",
character.traits)[*].en
    } )
```

Graph

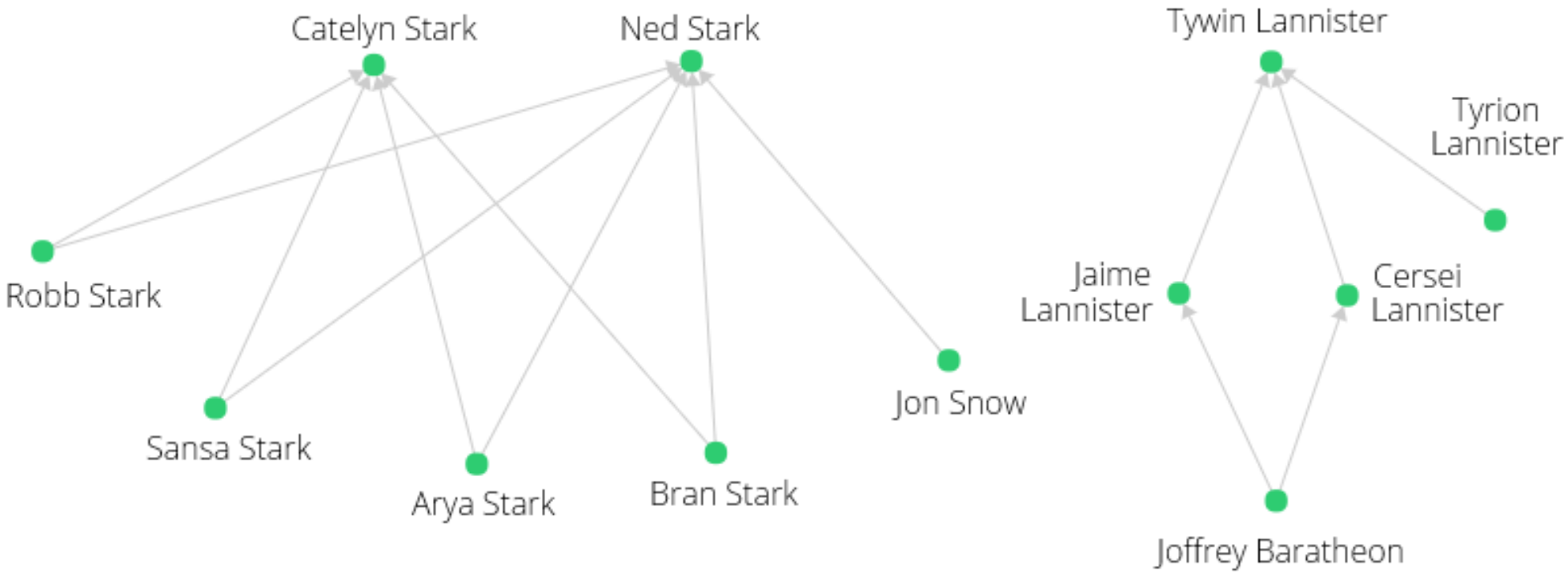
What is a graph? a graph is defined as set of **vertices** and **edges**. In ArangoDB, each edge has a single direction, it can't point both ways at once. This model is also known as oriented graph. There are two types of Graph traversal strategies they are depth-first and breadth-first.

Sample Use Cases are:

1) Dependency Management
2) Network Infrastructure
3) Recommendation Engine
4) Social Media Management

Many more uses cases present in real life in which Relational DB will be complex.

Graph Tree


Catelyn Stark
Ned Stark
Robb Stark
Sansa Stark
Arya Stark
Bran Stark
Jon Snow
Tywin Lannister
Tyrion Lannister
Jaime Lannister
Cersei Lannister
Joffrey Baratheon

Graph Queries

FIRST DEGREE Nodes – Children

AQL

```
FOR c IN Characters
      FILTER c.name == "Ned"
      FOR v IN 1..1 INBOUND c ChildOf
            RETURN v.name
```

For Identifying Parent Node use - OUTBOUND

SECOND DEGREE Nodes – Grand Children

AQL

```
FOR c IN Characters
    FILTER c.name == "Tywin"
    FOR v IN 2..2 INBOUND c ChildOf
        RETURN v.name
```

NoSQL Performance Benchmark 2018

ArangoDB, MongoDB, Neo4j, OrientDB and PostgreSQL

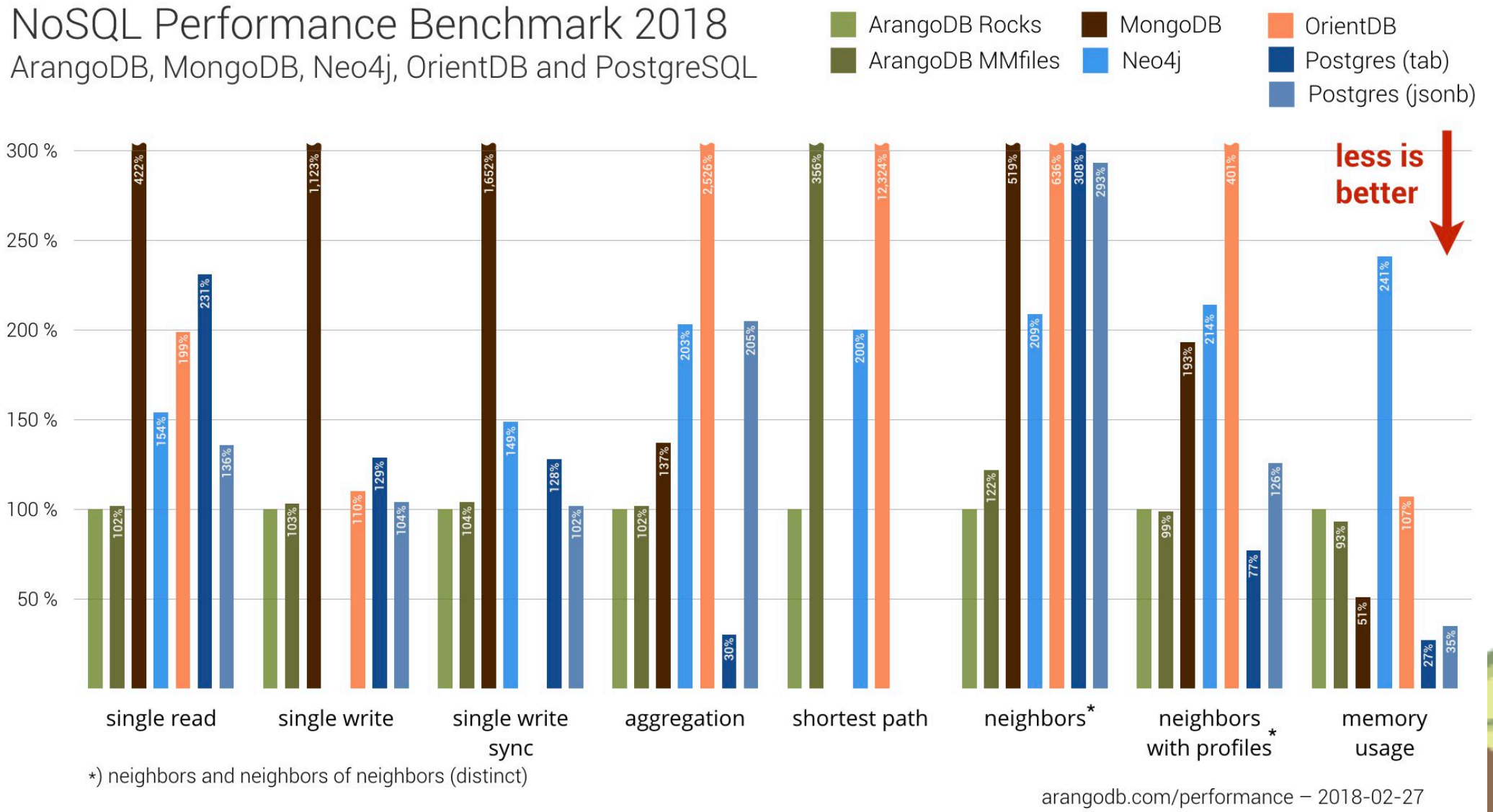


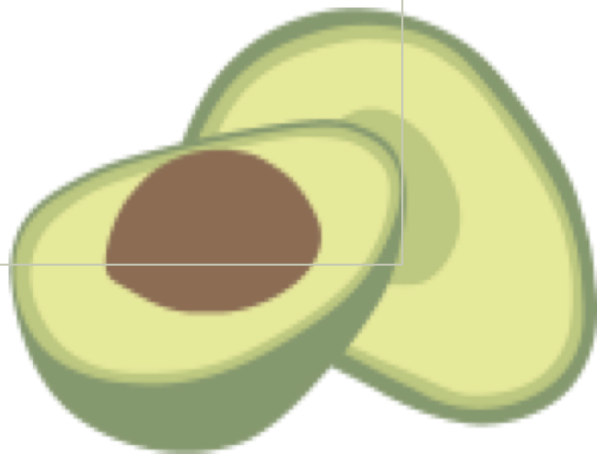
Figure from ("Benchmark: MongoDB, PostgreSQL, OrientDB, Neo4j and ArangoDB", 2019)

ArangoDB Foxx

Foxx services consist of JavaScript code running in the V8 JavaScript runtime embedded inside ArangoDB.

Few Advantages are

- Standardizing your data access and storage logic
- Running inside of ArangoDB
- Authentication and Authorization
- Swagger
- infinite extensibility and integration with external services



Sample Foxx

Get all Keys from Collection:

```
router.get '/entries' function (req,
res) {
const keys = db._query(aql
` FOR entry IN ${foxxCollection}
RETURN entry._key `);
res.send(keys);
});
```

Get Value of a Key from Collection:

```
router.get '/entries/:key' function (req, res){
const data =
${foxxCollection}.document(req.pathParams.key);
res.send(data);
});
```

Driver Support

- Go
- Java
- JavaScript
- PHP
- Python
- Ruby
- Scala
- .NET

Summary

ArangoDB is a open source with Polyglot Persistence database which has a powerful query language and a Web Interface. With it's Foxx, we can relay on a Microservice based architecture with direct interaction from user interface. Easy to install and scale in real time cloud platforms.

References

1. ArangoDB Query Language (AQL) Tutorial | ArangoDB Documentation. (2019). Retrieved 11 October 2019, from https://www.arangodb.com/docs/stable/aql/tutorial.html
2. ArangoDB System Properties. (2019). Retrieved 11 October 2019, from https://db-engines.com/en/system/ ArangoDB
3. Benchmark: MongoDB, PostgreSQL, OrientDB, Neo4j and ArangoDB. (2019). Retrieved 11 October 2019, from https://www.arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/
4. SQL (Structured Query Language) and AQL (ArangoDB Query Language). (2019). Retrieved 10 October 2019, from https://www.arangodb.com/why-arangodb/sql-aql-comparison/

Thank You