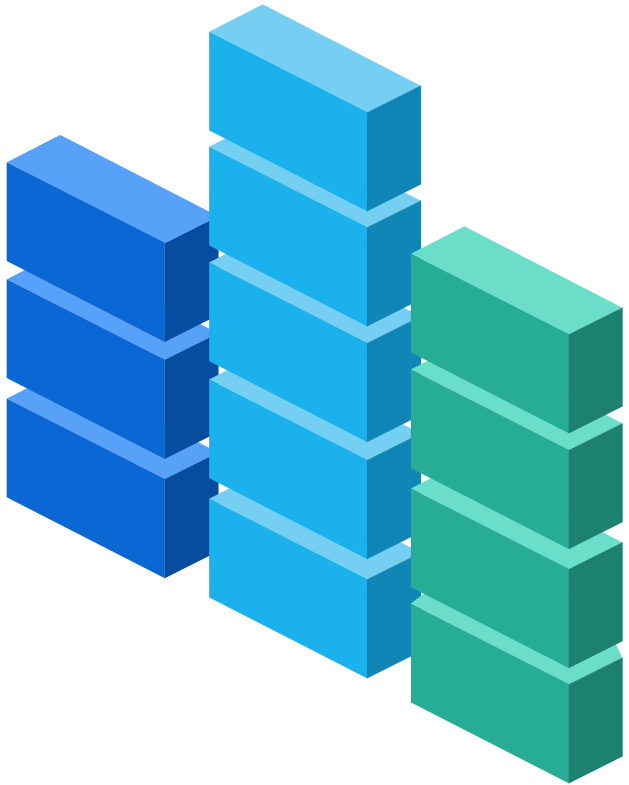
ULB

INFO-H415
ADVANCED DATABASES

13/12/22



Document Models:
Arango
MarkLogic

Prepared by: Rishika, Ahmad, Chidiebere, Mirwise

Supervised by: Prof. Dr. Esteban Zimanyi

OBJECTIVE

Identify a suitable application, and perform a benchmark comparison between two proposed DB models under the same technology that are expected to specialize well with the requirements of the application

AGENDA

Document Model Introduction 01

Brief Introduction to Arango & MarkLogic 02

Proposed Application & Dataset: Yelp 03

Relational Model Comparison 04

Query Use Cases 05

Benchmark Results & Analysis 06

Conclusion 07

Document Model: Introduction

- NoSQL database stores unstructured data as documents (JSON, XML, etc.)
- Not restricted to a schema like relational models
- Allows additional data types/structures
- Easy to scale horizontally and store large amounts of data
- Makes use of index to improve performance
- Groups similar documents into collections (similar to concept of table)


```
{
    "year" : 2013,
    "title" : "Turn It Down, Or Else!",
    "info" : {
        "directors" : [ "Alice Smith", "Bob Jones"],
        "release_date" : "2013-01-18T00:00:00Z",
        "rating" : 6.2,
        "genres" : ["Comedy", "Drama"],
        "image_url" : "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@._V1_SX400_.jpg",
        "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
        "actors" : ["David Matthewman", "Jonathan G. Neff"]
    }
}
```

Document Model: Introduction

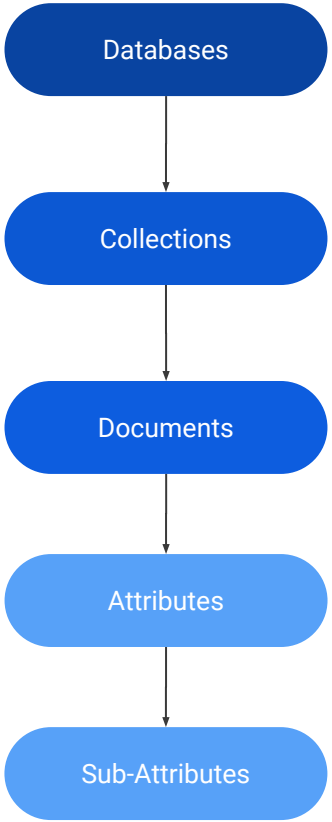


Image source: https://images.app.goo.gl/KVT8VHmJHYY9V8id8

MarkLogic® Introduction

- Native XML database that is multi-model
  - Documents
  - RDF stores (semantic)
  - Search engine
- Supports ACID transactions
- Easy to communicate with, example:
  - REST API
  - Node.js Client API

MarkLogic

Introduction

- Supports a few different languages to query XML documents:
  - XQuery
  - SPARQL
  - JavaScript
- Models JSON documents as a tree of nodes, which allows us to traverse the document similar to XML using XQuery.
- Uses indexes to improve performance, such as
  - Element range index
  - Inverted index

Introduction

- Native multi-model database
  - Documents
  - Key-value pairs
  - Graphs
  - Search engine
- Allows to integrate different data models and perform queries uniformly
- Supports ACID transactions
- Supports DML however it does not support DDL and DCL.

Introduction

- Easy to interact with using HTTP API (extendable with JavaScript)
- Supports a unified query language AQL (ArangoDB Query Language)
- AQL is known to be a declarative language having similarities with SQL
- Uses indexes to improve performance, such as:
    - Persistent index
    - Inverted index

Tools Comparison

TOOL FEATURES	ARANGO	MARKLOGIC
Model Types	Document, key-value, Graph, Search	Document, RDF, Search
XML Support	No	Yes
SQL Support	No	Yes
APIs	Less options	More options
Supported Programming Languages	More options	Less options
Triggers	No	Yes
MapReduce	No	Yes
Foreign Keys	Yes	No
Transactions	ACID	ACID
Concurrency	Yes	Yes

Proposed Application: Yelp

- Founded in 2004 by former PayPal employees (Russel Simmons and Jeremy Stoppelman)
- Connects businesses with customers through crowd-funding reviews.

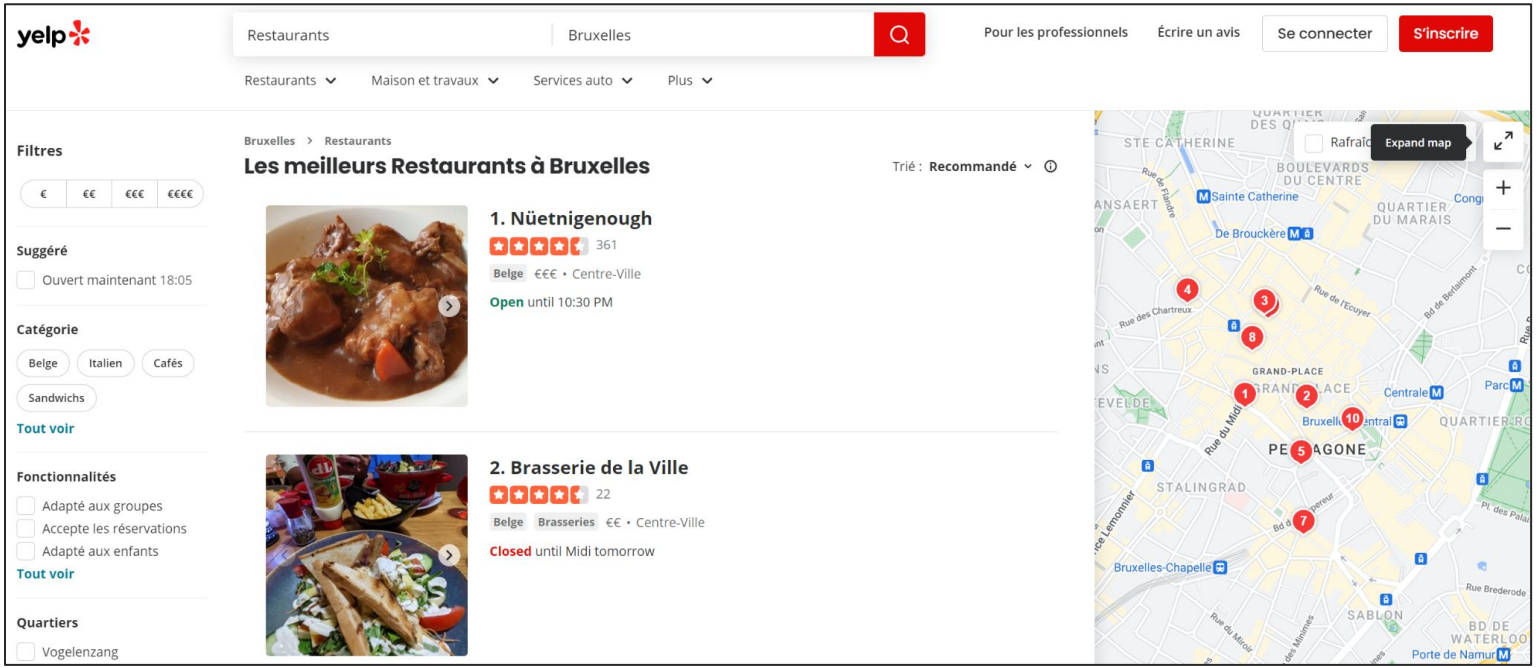
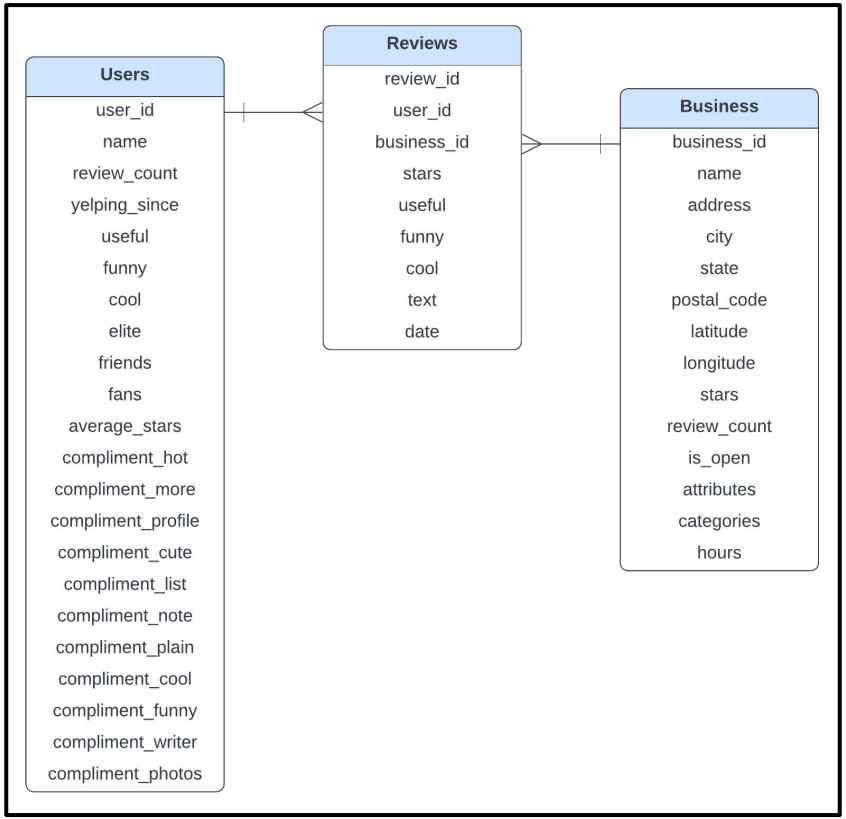


Image source: https://fr.yelp.be/bruxelles

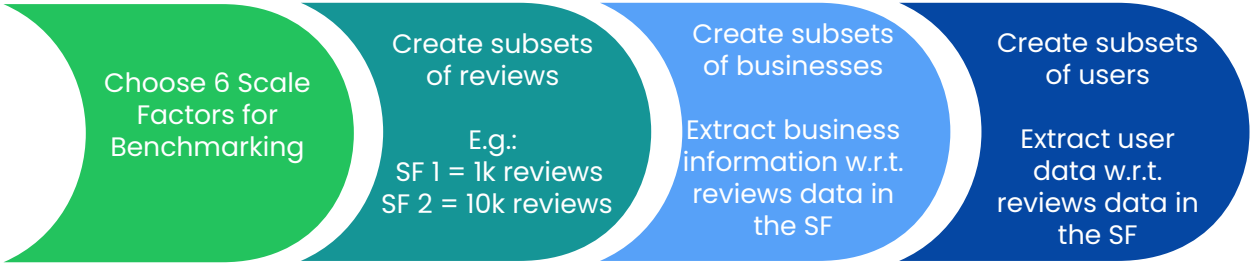
Yelp Dataset

- Dataset provided from Yelp Inc. themselves, and retrieved from Kaggle
- Contains json files for businesses, reviews, users, photos, checking and tips
- Collections used for the project:
  - Business (150k documents)
  - Review (7 mil documents)
  - User (200k documents)



Yelp Dataset: Scale Factors

Data Modeling Flow



Scale Factor Summary

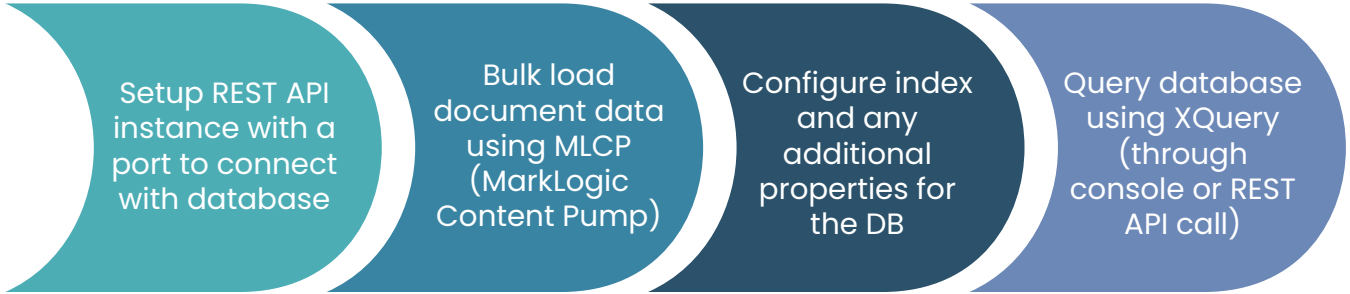
	SF 1	SF 2	SF 3	SF 4	SF 5	SF 6
Reviews	1,000	10,000	100,000	1,000,000	2,000,000	6,990,280
Businesses	823	3,930	9,973	27,095	45,055	150,346
Users	994	9,472	79,345	541,999	895,924	1,987,897

Relational Model: Discussion


```
{
 "business_id": "EjzWyITI75CKSKb6RGCG_Q",
 "name": "Sonny's BBQ",
 "address": "8106 W Hillsborough Avenue",
 "city": "Tampa",
 "state": "FL",
 "postal_code": "33615",
 "latitude": 27.9958415,
 "longitude": -82.5722699,
 "stars": 3,
 "review_count": 11,
 "is_open": 0,
 "attributes":{
  "GoodForKids": "False",
  "HasTV": "None",
  "NoiseLevel": "u'average'",
  "RestaurantsDelivery": "False",
  "WiFi": "u'no'",
  "RestaurantsPriceRange2": "2",
  "RestaurantsReservations": "False",
  "OutdoorSeating": "False",
  "RestaurantsTakeOut": "True",
  "RestaurantsGoodForGroups": "False",
  "RestaurantsAttire": "u'casual'",
  "Ambience": "{'romantic': False, 'intimate': False, 'touristy': False, 'hipster': False, 'divey': False, 'classy': False, 'trendy': False, 'upscale': False, 'casual': True}",
  "BusinessParking": "{'garage': False, 'street': False, 'validated': False, 'lot': False, 'valet': False}",
  "BusinessAcceptsCreditCards": "True",
  "GoodForMeal": "{'dessert': False, 'latenight': False, 'lunch': False, 'dinner': False, 'brunch': False, 'breakfast': False}"
 },
 "categories": "Barbeque, Restaurants, Event Planning & Services, Caterers",
 "hours":{
  "Monday": "11:0-21:0",
  "Tuesday": "11:0-21:0",
  "Wednesday": "11:0-21:0",
  "Thursday": "11:0-21:0",
  "Friday": "11:0-21:30",
  "Saturday": "11:0-21:30",
  "Sunday": "11:0-21:0"
 }
}
```


- For element ‘attributes’, it has a nested object with many children and nested children, which would require several hundred columns to transform.
- Element ‘categories’ sometimes have the data type as a list of strings, while in other cases it’s in a nested json object form, which adds more complexity for conversion.
- Cannot directly add child {“Cuisine Type” : “True”} to “attributes”, if new data is added (without changing schema)
- Due to these issues, data maintenance will have more overhead cost if stored as relational model

MarkLogic DB Setup Flow



MLCP Import Example

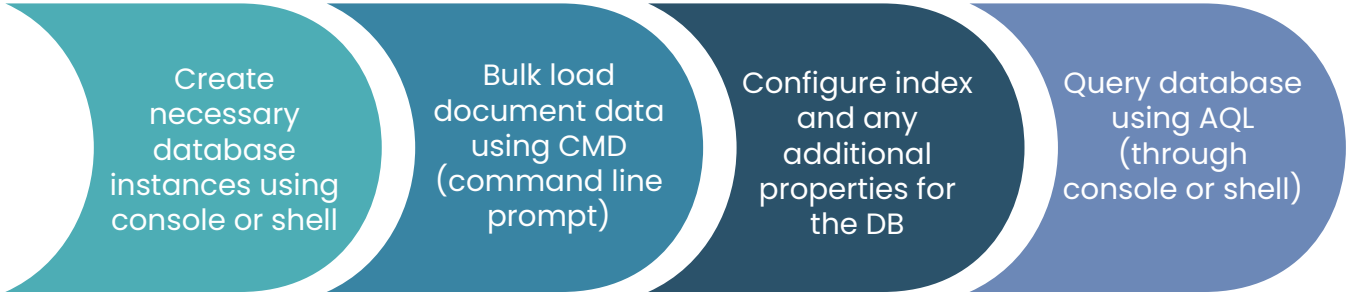
```
mlcp.bat import -host localhost -port 8021 -username admin -password admin -input_file_path ^
"C:\sf_data\reviews_1.json" -mode local -input_file_type delimited_json -output_collections "reviews"
```

XQuery Example

Find reviews that include the word 'love' (case insensitive) and were reviewed by a user who had elite status on the year of the review

```
(
  for $r in collection('reviews')[matches(lower-case(text), 'love')],
  $u in collection('users')
  where $r/user_id = $u/user_id
  and matches($u/elite, format-dateTime(xs:dateTime($r/date), "[Y0001]"))
  return $r
)[position() = 1 to 1000]
```

DB Setup Flow



Bulk Import Example

```
arangoimport --server.database "yelp_sf_1" --server.username "test_adb"
--server.password "adb" --file "sf_1/reviews_1.json" --type json --collection reviews
```

AQL Example

Find reviews that include the word 'love' (case insensitive) and were reviewed by a user who had elite status on the year of the review

```
FOR r in reviews
FILTER LIKE(LOWER(r.text),'%love%',true)

FOR u in users
LET years = TO_BOOL(TO_STRING(DATE_YEAR(r.date)) IN SPLIT(u.elite,','))

FILTER years == true
and r.user_id == u.user_id

LIMIT 1000

RETURN r
```

Use of Index: Arango & MarkLogic

PERSISTENT INDEX

- General purpose index type
- Used for equality lookups, lookups based on a leftmost prefix of the index attributes, range queries and for sorting
- Operations that use persistent index have a logarithmic complexity

MarkLogic®

ELEMENT RANGE INDEX

- Accelerates queries that involve comparisons within specific data type
- Keeps tab on different values that appear within the XML elements/JSON properties using a given name, type and collation set

Benchmark: Method

- A python script was developed to run all queries against each DB
  - Arango connector: pyArango
  - MarkLogic connector: REST API
- Total of eighteen queries:
  - 5 CRUD (Create, Read, Update, Delete)
  - 13 OLAP (Filters, Joins, Aggregations, Word search)
- For each scale factor, all queries are run in sequence and iterated 3 times to get an average run time for the respective queries
- Results are then integrated together before final analysis

Query Use Cases: CRUD

DELETE

Remove the additional 1 million reviews that were added to the collection

UPDATE

Update 1: Add an additional attribute (sentiment_score) which is the total score from useful, funny & cool attributes.
Update 2: Removing the extra attribute

READ

Display x reviews from the collection

CREATE

Insert an additional 1 million reviews into the review collection

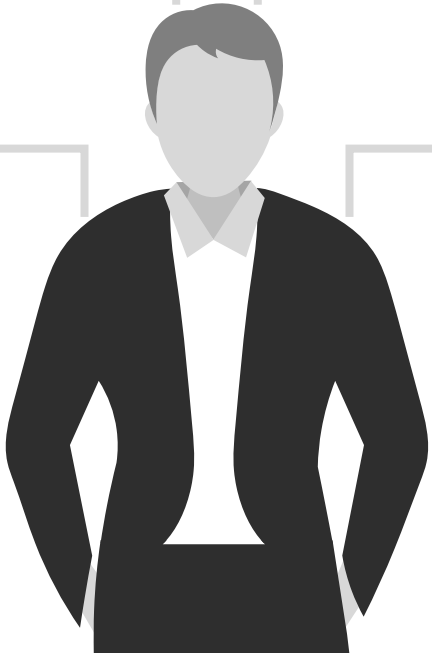
Query Use Cases: OLAP

Query 14

Find the reviews of restaurants that have free wifi

Query 13

Find top rated restaurants in the state of Florida (FL) with a rating of 4+



Query 12

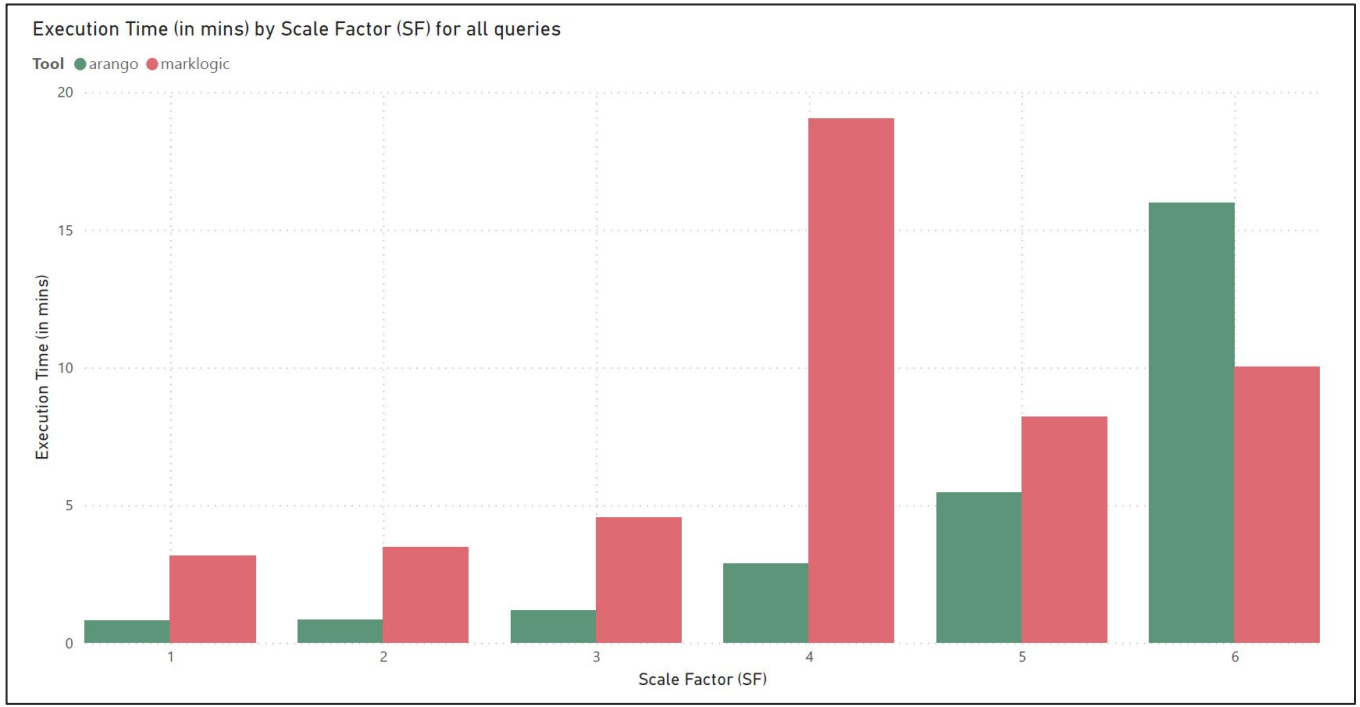
Find the reviews of businesses that are open on both days of the weekend until 10pm

Query 09

Find the reviews of users who have at least 100 fans

Note: These are just a few examples of our OLAP query set. More can be found in the report

Benchmarking Results: Overall

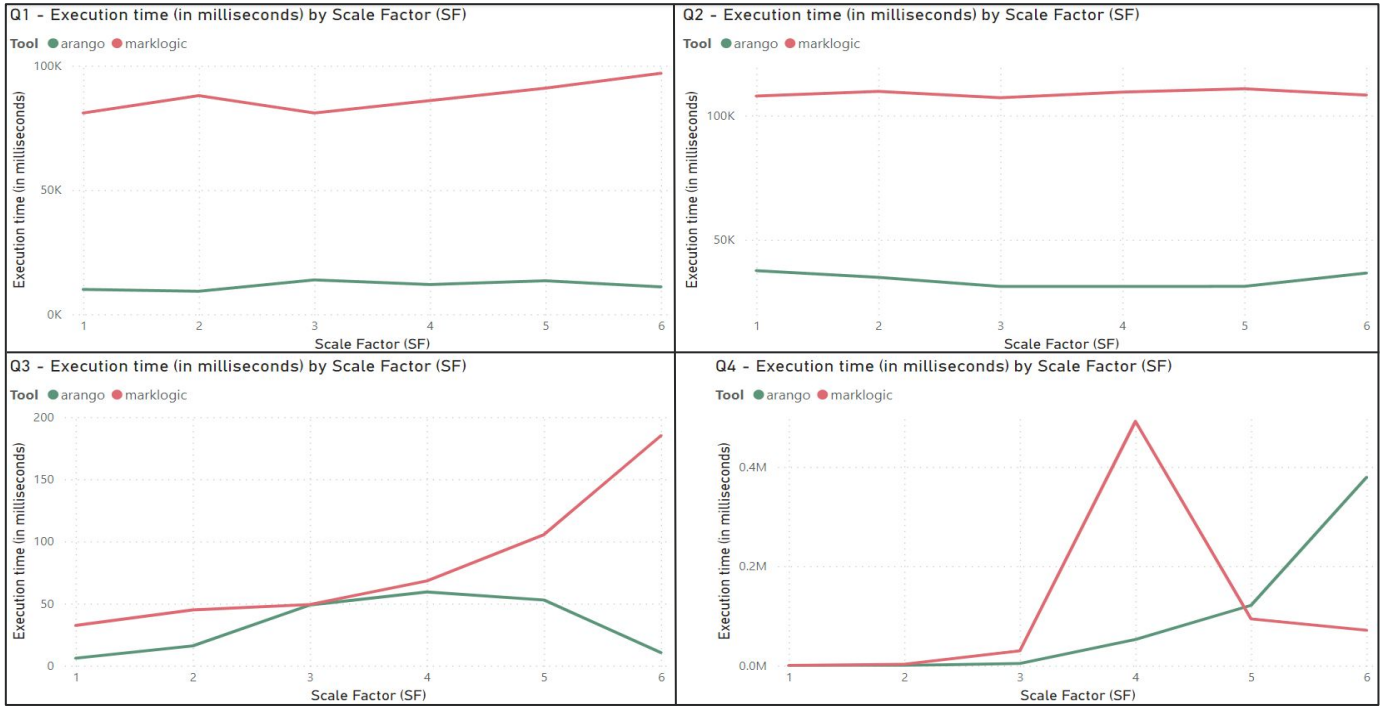


- Interestingly Arango starts off much better with the smaller SFs, performing better, but the gap closes in from SF 5, and Arango shows poorer performance in the last SF, which may suggest that MarkLogic works better with larger volumes of data

Benchmarking Results: Overall CRUD and OLAP

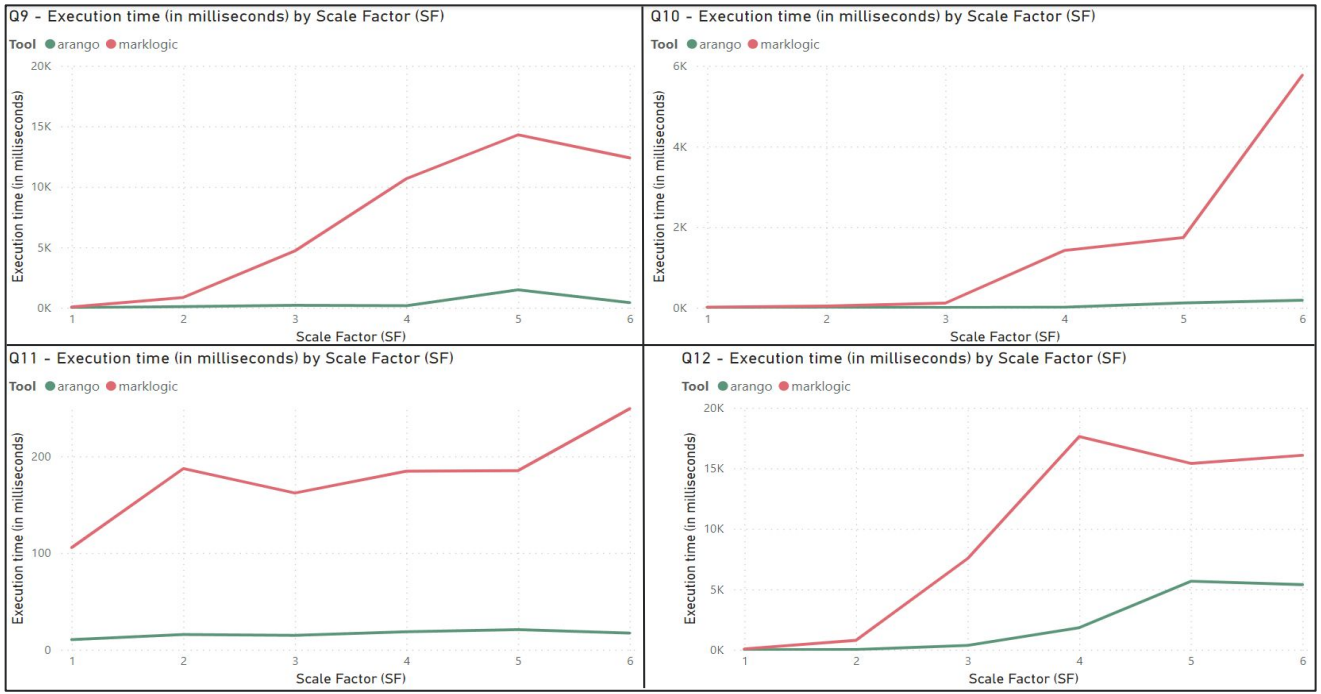
- Splitting the overall run time to CRUD and OLAP queries, we still see a similar pattern as from the previous slide.

Benchmarking Results: CRUD Queries



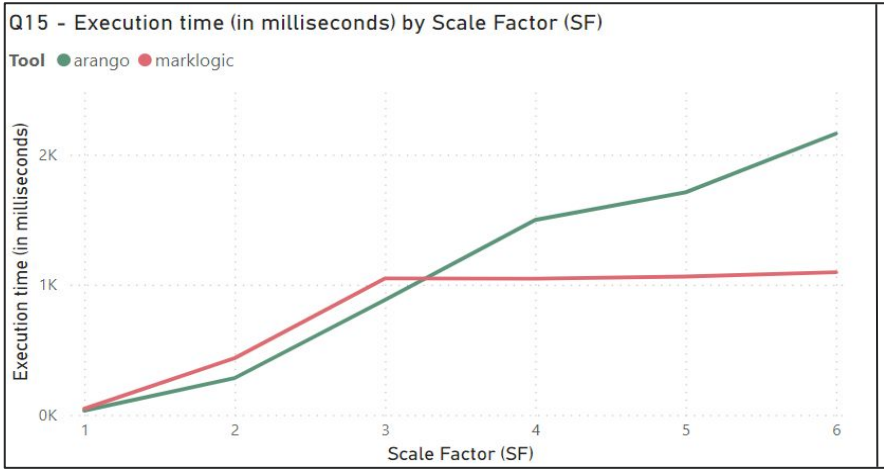
- Arango manages to perform both large inserts and deletes at a much faster pace as compared to MarkLogic, and similar observation can be seen with read operations
- Interestingly, with update operations (Q4-5), MarkLogic starts to perform better from SF5 onwards

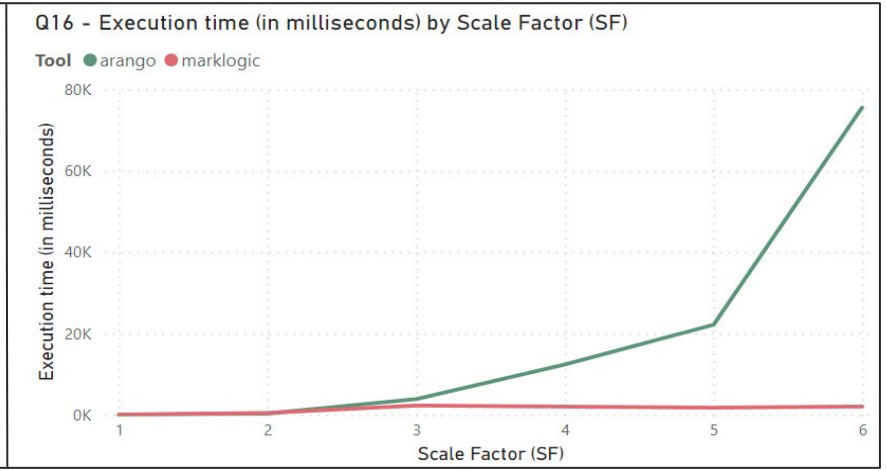
Benchmarking Results: Queries 09-12



- Arango outperforms MarkLogic by a large margin, with queries that involve normal filters, joins and aggregation

Benchmarking Results: Queries 15-16





- MarkLogic starts to shine when it comes to queries that involve word searches.
  - For example Q15: identifying key words from the review text
- Furthermore, Arango does seem to struggle a bit when it comes to nested sub-queries (Q16)

Conclusion

- Both tools have allowed us to learn the benefits of document models, in terms of flexibility, usage and performance
- Arango seems to outshine MarkLogic in majority of the scenarios within the use cases we have tested in terms of CRUD and OLAP queries
- MarkLogic does show better performance in some specific cases, such as word searches and update operations on large data volumes

Q&A

Image source: https://i0.wp.com/www.videooutcomes.com/wp-content/uploads/2021/06/Video-Outcomes-questions-for-your-corporate-video-production-company.jpg?fit=768%2C512&ssl=1