Couchbase

Maroš Kopec, Bibiána Ťureková, Nikoleta Češeková, Iuliia Gurianova

Agenda

- Evolution
- Main features
- Core principles
- Architecture
- Comparison with MongoDB
- Demo

Couchbase

membase

- Memcached technology
- Sharding
- Replication
- Persistence

- Document oriented model based on JSON

Main Features

- **Flexible data model**
  - Json documents with no fixed schema
- **Easy scalability**
  - Grow cluster without downtime or application changes
  - Automatic sharding and rebalancing
- **High performance**
  - consistently high throughput
  - sub-millisecond response times
- **Always on**
  - No downtime for upgrades, maintenance...

Developer integration

- Provide libraries for:
  - Net
  - PHP
  - Ruby
  - Python
  - C
  - Node. js
  - Java
  - GO

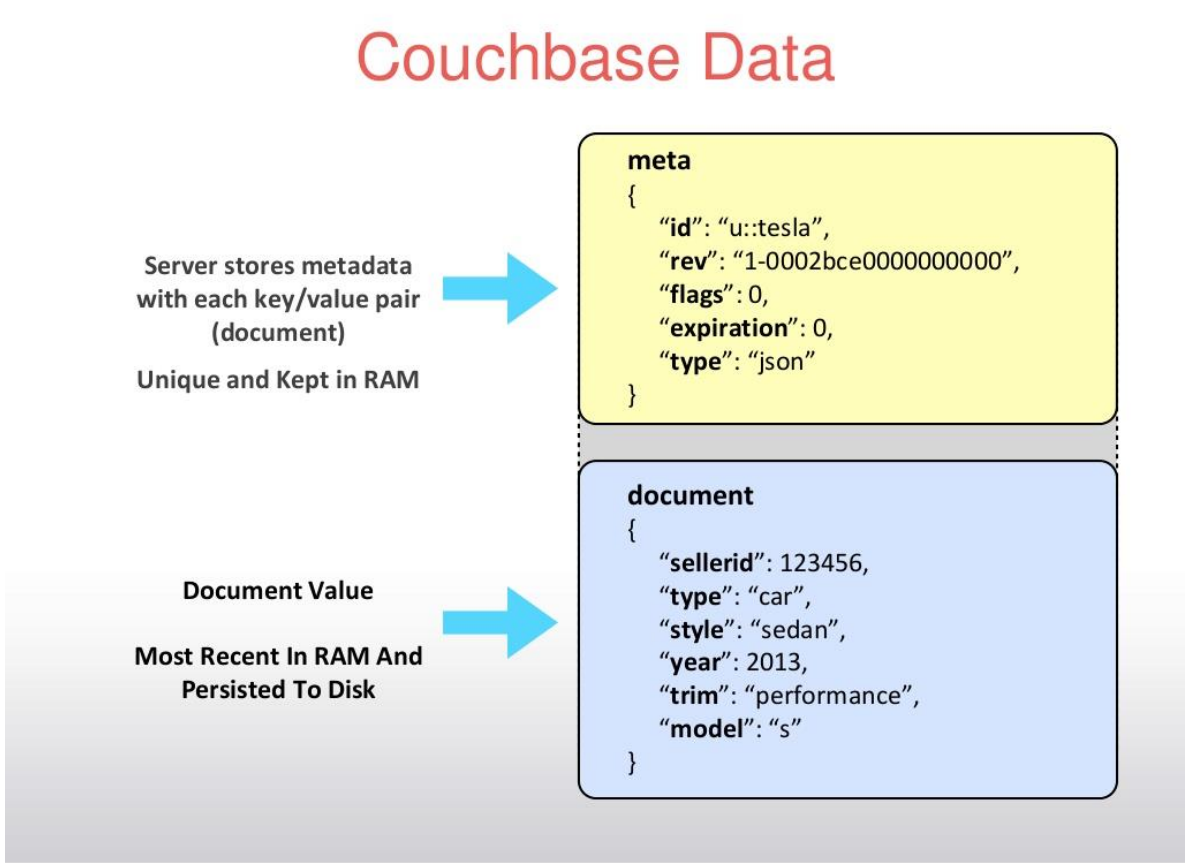
Key customers

Real use cases

- Real-time user activity
- User preferences
- Caching
- Promotions tracking
- Users targeting

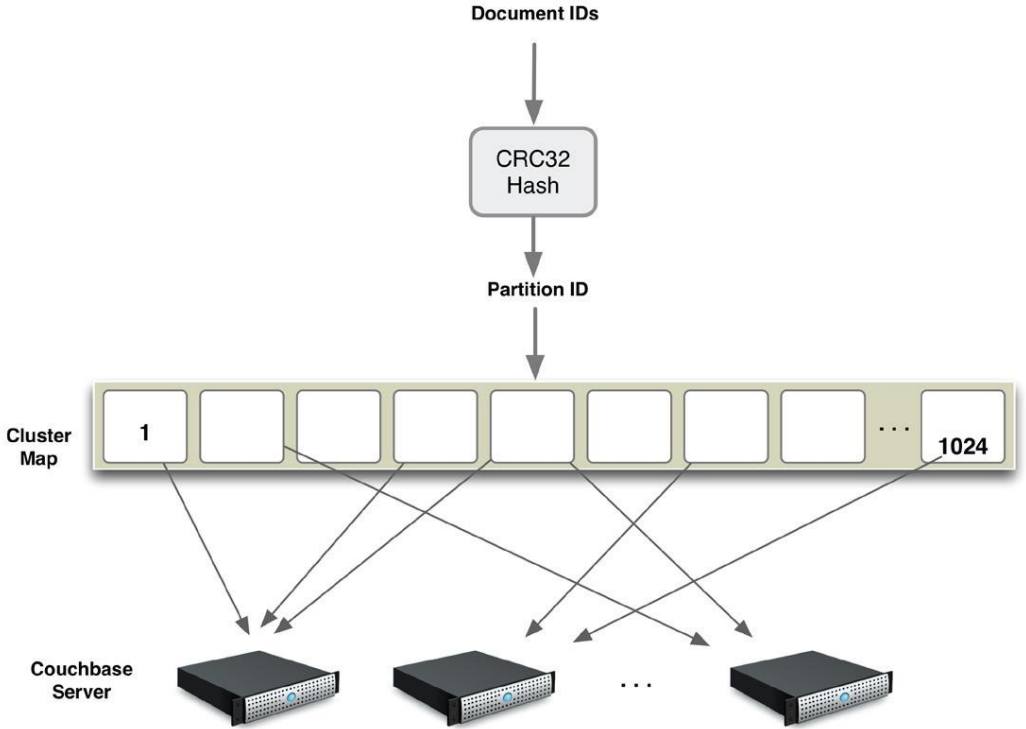
Core principles

- Documents stored in **buckets** (“db”)
- **vbuckets**
  - Owner of keys
  - Responsible for distribution and replication
- Document = key + value



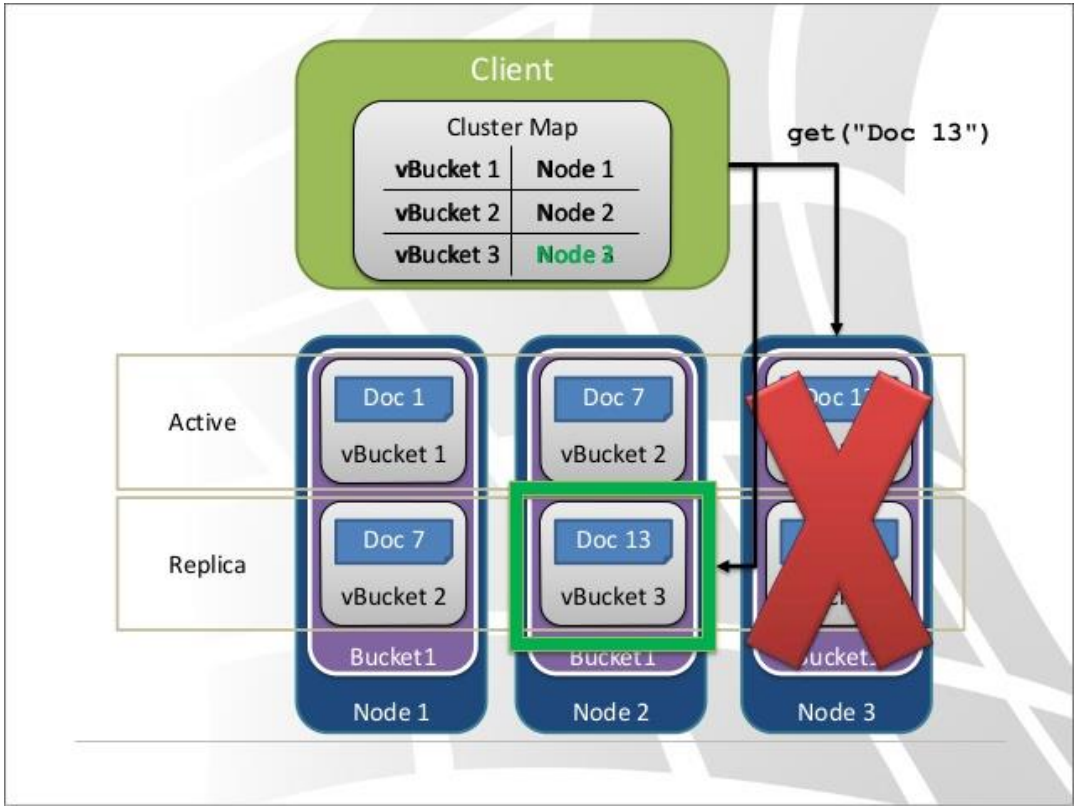
Architecture

- Highly distributed architecture
- Horizontal scaling by hash sharding (by document ID)
- Servers are equal
- Server
  - Data manager
  - Cluster manager



Replication

- documents in a bucket are isolated
- configurable number of replicas (up to three)
- if a server crashes, the system will locate the replicas of the documents, and promote them to active status



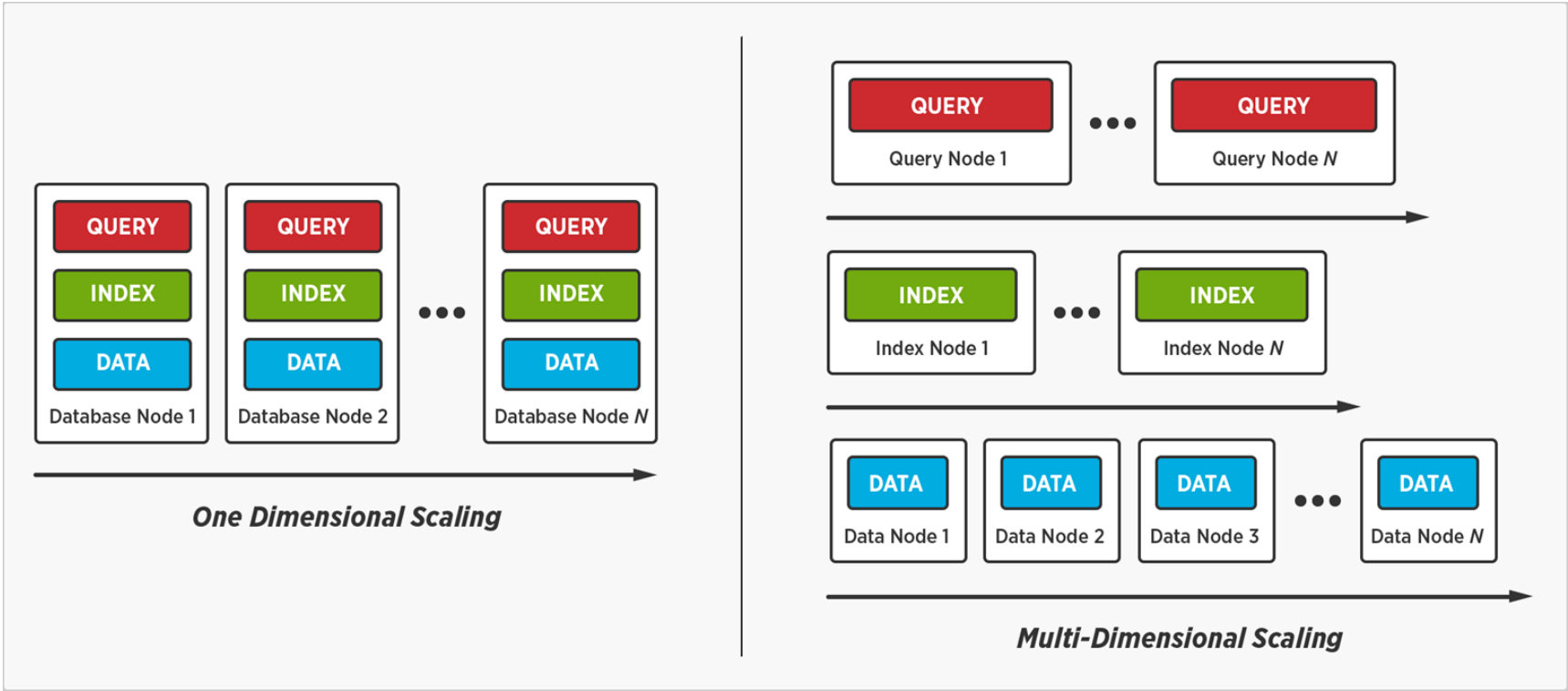
MongoDB

Couchbase

- master-slave; Single-node and sharded environments
- third-party caches
- unique proprietary language
- CLI tools for maintenance

- masterless, clustered, and replicated distributed database; Multi-Dimensional Scaling (MDS)
- caching layer for both data and indexes
- SQL-based query language
- Web UI, CLI, REST API

Multi-Dimensional Scaling



Queries

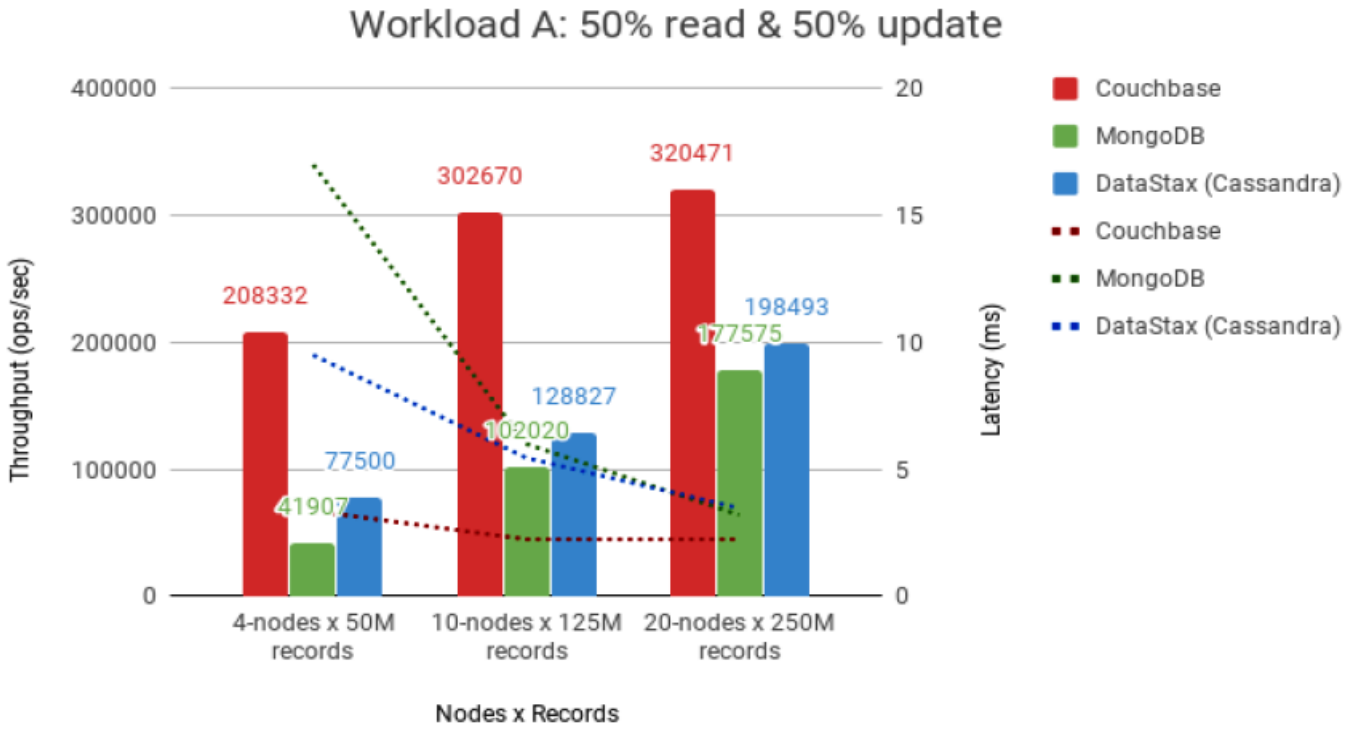
MongoDB™ Query

```
db.stocks.aggregate([
      { "$match": {
            "$and": [
              {"symbol": {
                    "$in": [
                      "AAPL",
                      "GOOG"]}},
              { "value": {
                    "$gt": 0 }}]}},
      { "$group": {
            "_id": {
              "symbol": "$symbol" },
            "sum(value * volume)": {
              "$sum": {
                "$multiply": [
                  "$value",
                  "$volume"]}}}},
      { "$project": {
            "_id": 0,
            "sum(value * volume)": "$sum(value * volume)",
            "symbol": "$_id.symbol"}}
      { "$sort": {
            "sum(value * volume)": -1,
            "symbol": 1 }}])
```

Couchbase N1QL

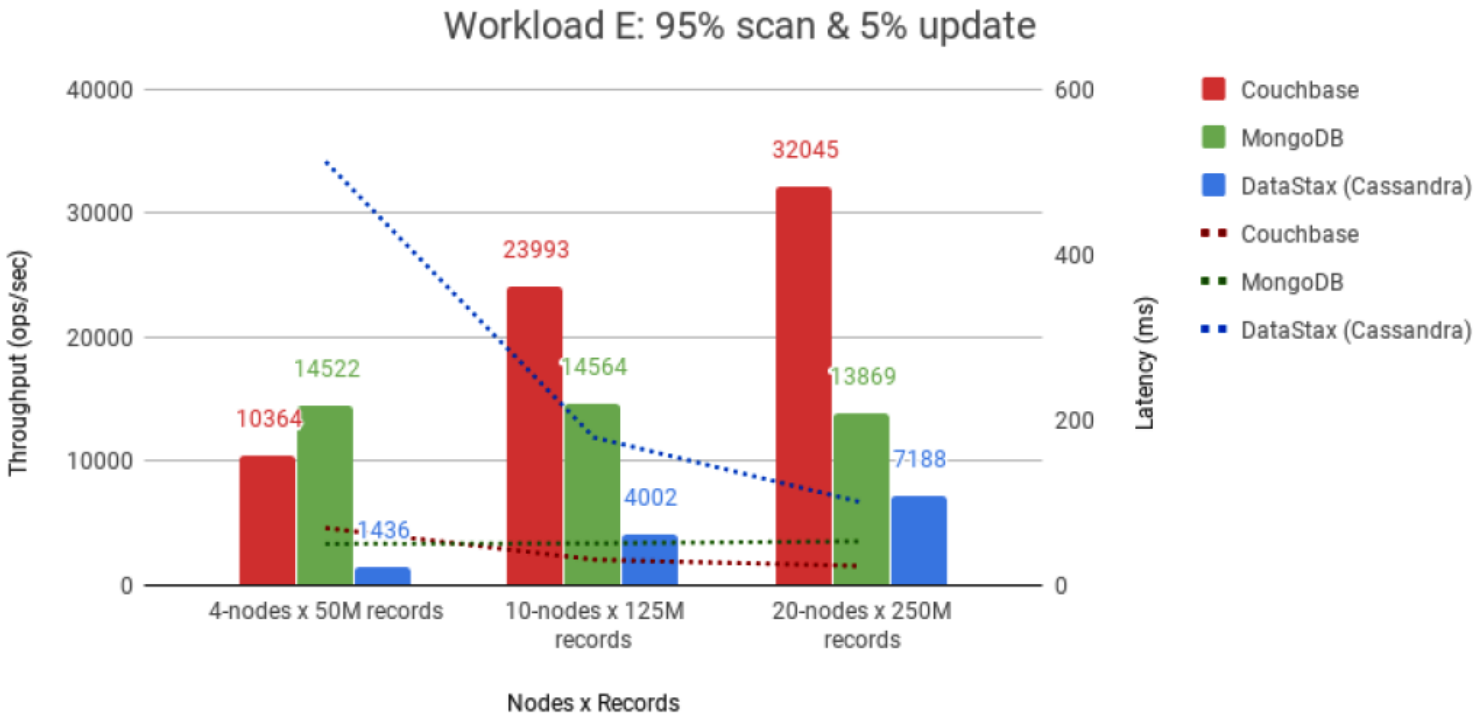
```
SELECT SUM(value * volume) AS val, symbol
FROM   db.stocks
WHERE  symbol IN ( "AAPL", "GOOG" ) AND value > 0
GROUP  BY symbol
ORDER  BY val DESC, symbol ASC
```

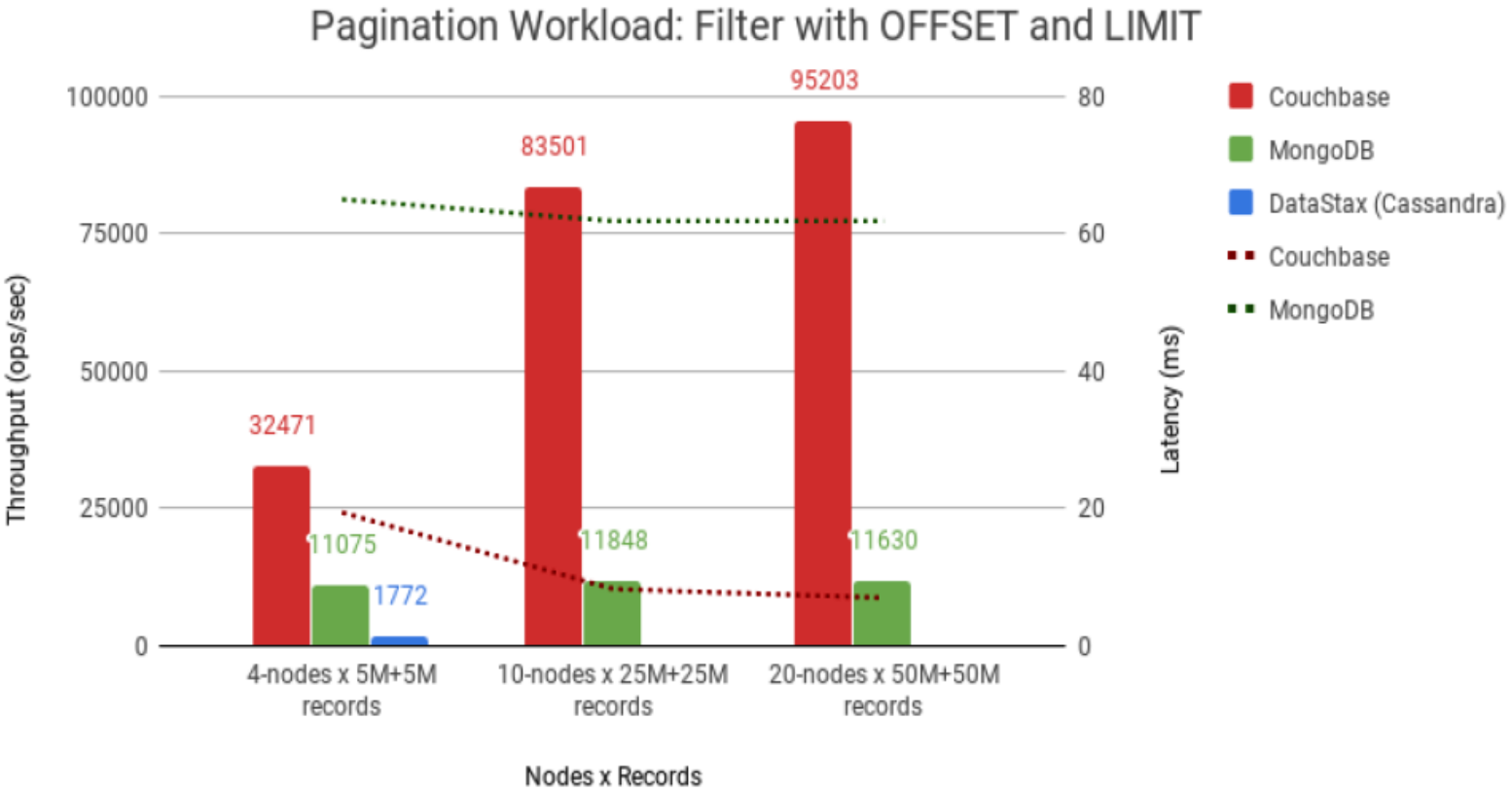
Update workload

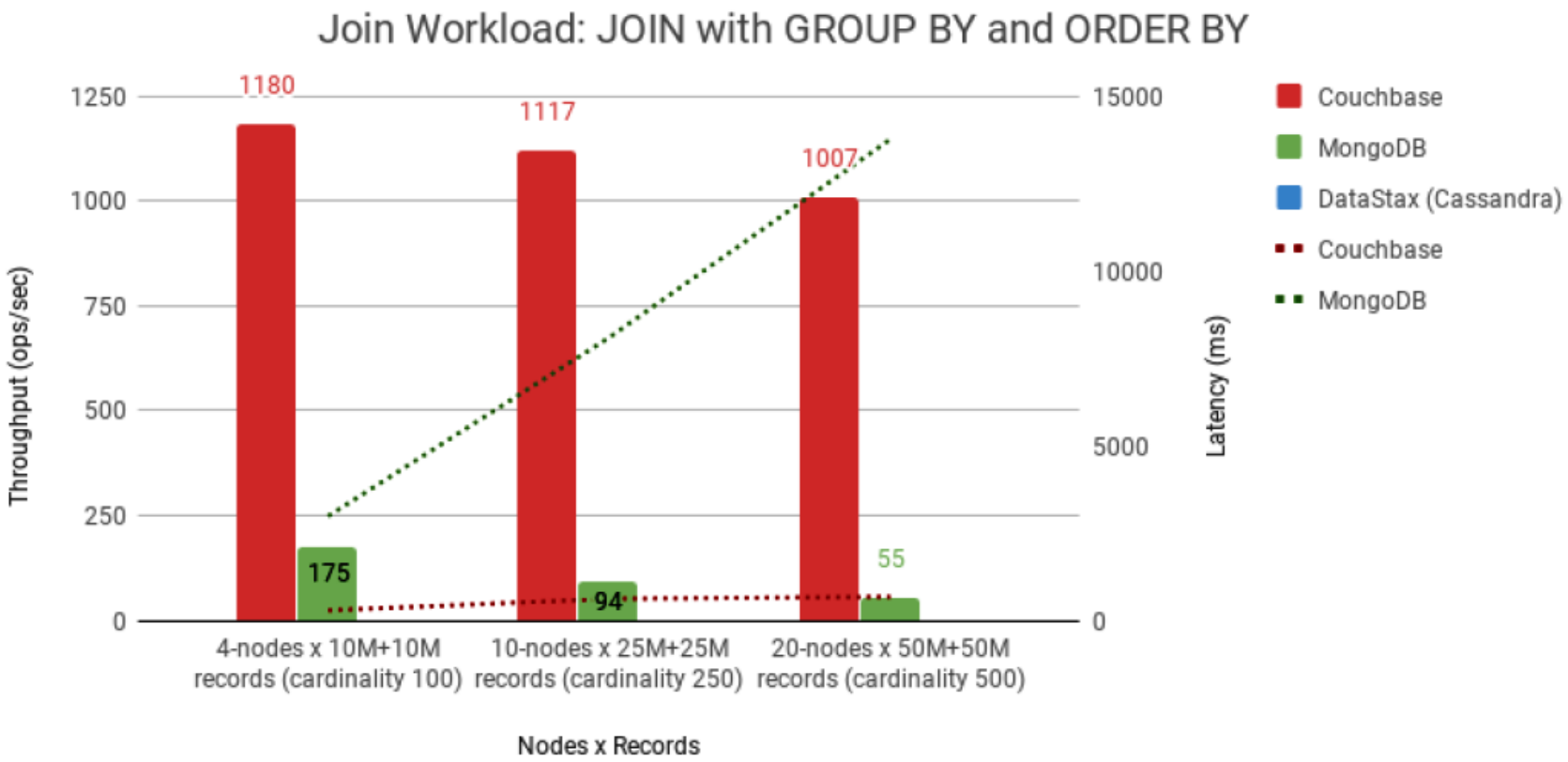


https://www.datanami.com/2019/01/16/benchmarking-nosql-databases/

Short-range scan workload




Pagination Workload: Filter with OFFSET and LIMIT
Throughput (ops/sec)
Latency (ms)
100000
75000
50000
25000
0
80
60
40
20
0
95203
83501
32471
11075
1772
11848
11630
4-nodes x 5M+5M records
10-nodes x 25M+25M records
20-nodes x 50M+50M records
Nodes x Records
Couchbase
MongoDB
DataStax (Cassandra)
Couchbase
MongoDB


Join Workload: JOIN with GROUP BY and ORDER BY
Throughput (ops/sec)
Latency (ms)
1250
1000
750
500
250
0
15000
10000
5000
0
1180
1117
1007
175
94
55
4-nodes x 10M+10M records (cardinality 100)
10-nodes x 25M+25M records (cardinality 250)
20-nodes x 50M+50M records (cardinality 500)
Nodes x Records
Couchbase
MongoDB
DataStax (Cassandra)
Couchbase
MongoDB

DEMO

Questions