

ASSIGNMENT 2

CLOUD COMPUTING

MONGODB VS COUCHDB

Submitted by:
Rashmi Gulhane
Akansha Gupta

Mongodb

Steps we followed with mongodb

1. Created cluster as specified below:
Cluster configuration:
3 config server
2 masters (query servers)
2 shard servers
2. CRUD operations using mongo shell
3. CRUD operations using java edition programme
4. Performance evaluation with multiple scenarios
5. Consistency measurement with different scenarios

Tools used for benchmarking

Eclipse - IDE

Java - programming language

Using Multiple masters:

Tried to evaluate mongodb by using multiple query processors for processing common query:

1. Inserted 10k of records by using single master(query processor), it took 11 seconds
2. Used two masters and inserted 5K-5K records from each at the same time then both took 5.6 seconds around.

So with increasing masters query processing time got halved. That means in mongodb multiple threads can operate data at the same time until those are dealing with different documents.

CouchDB

Steps followed for CouchDB

1. 3 Nodes are configured as following:
CouchDB 1.6 database
Bigcouch 0.4.2-1
Number of Shards for each db:2
Copies of each shards:3
2. CURL Restful API Implementation using JAVA
3. Performance evaluation with multiple scenarios
4. Consistency measurement with different scenarios

Tools used for benchmarking

Eclipse - IDE
Java - programming language
Curl Restful API

Mongodb vs CouchDB

Data sets & Configuration:

OS: Ubuntu 14.04 LTS-64 Bit

CPU: Intel® Core™ i5-4570 CPU @ 3.20GHz

Database File sizes used: 2.4 MB, 12.7 MB, 127MB

No of Documents each of the above file contains: 1K, 10K, 100K, 1000K

Scenarios used for performance measurement:

Used three scenarios for performance measurement:

1. Insertion of Documents

One by one document insertion:

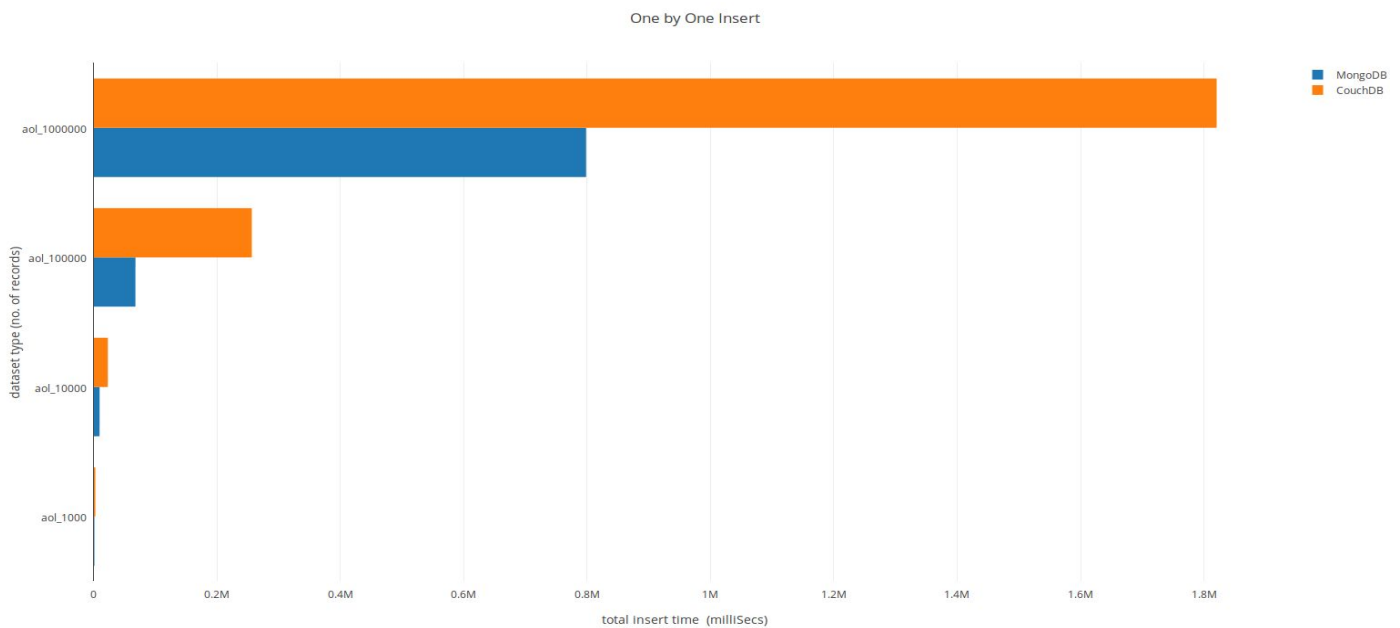
- **Creating records In memory:** Used a java for loop to generate random records to insert and evaluated by varying loop counter range.
- **Reading records from file:** Also tried to read records from a file that contains json documents to insert and evaluated by using different size of files.

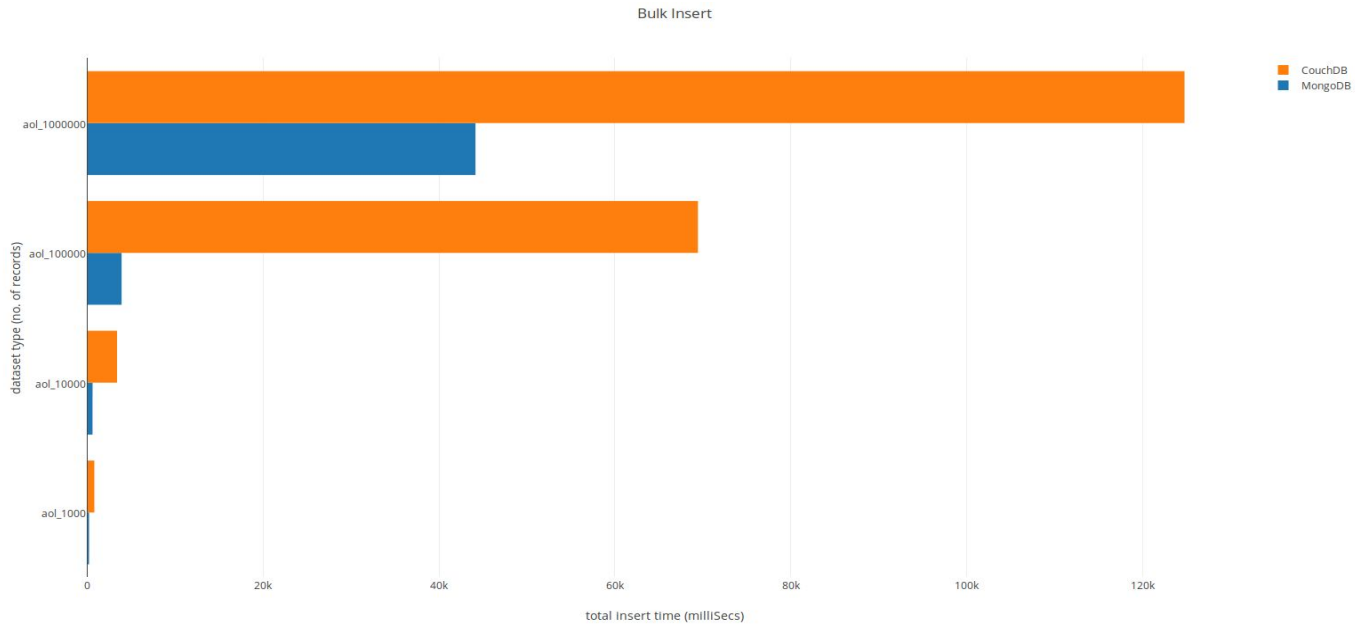
Bulk insertion:

1. **Creating records In memory:** Created a list of random generated records by for loop then inserted into db by single insert statement.

2. **Reading records from file:** Read all json documents from file, created a list then inserted by single insert statement.

| | MongoDB | | CouchDB | |
|-------|-------------------------------------|---|-------------------------------------|---|
| | Document Insertion (in millisec) | Document Insertion in Bulk (in millisec) | Document Insertion (in millisec) | Document Insertion in Bulk (in millisec) |
| 1K | 1312 | 149 | 2320 | 805 |
| 10K | 9928 | 601 | 23292 | 3389 |
| 100K | 68054 | 3905 | 256668 | 69449 |
| 1000K | 798552 | 44151 | 1820575 | 124786 |

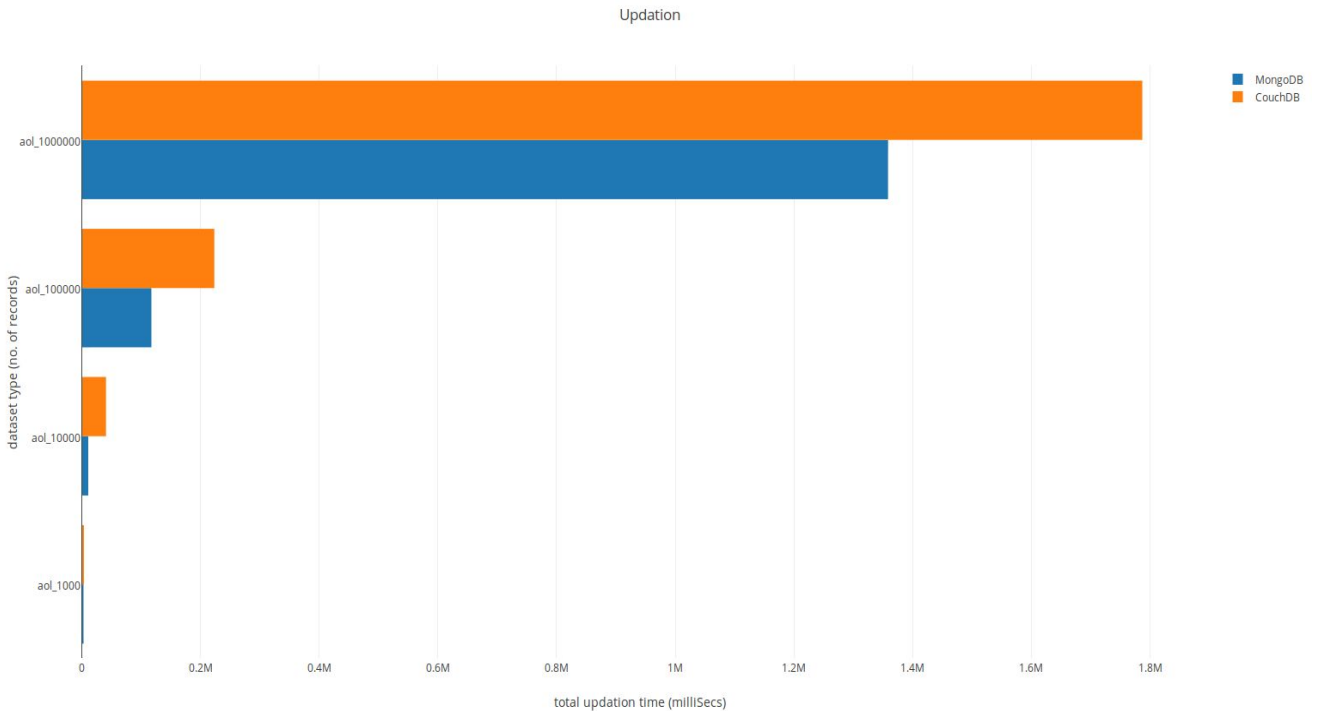




2. Updation of Documents

The following readings are noted down while updating each document in the database.

| No of Documents | MongoDB (in millisec) | CouchDB (in millisec) |
|-----------------|--------------------------|--------------------------|
| 1K | 1577 | 3560 |
| 10K | 10853 | 40728 |
| 100K | 117261 | 223256 |
| 1000K | 1358505 | 1786911 |



Scenarios used for consistency:

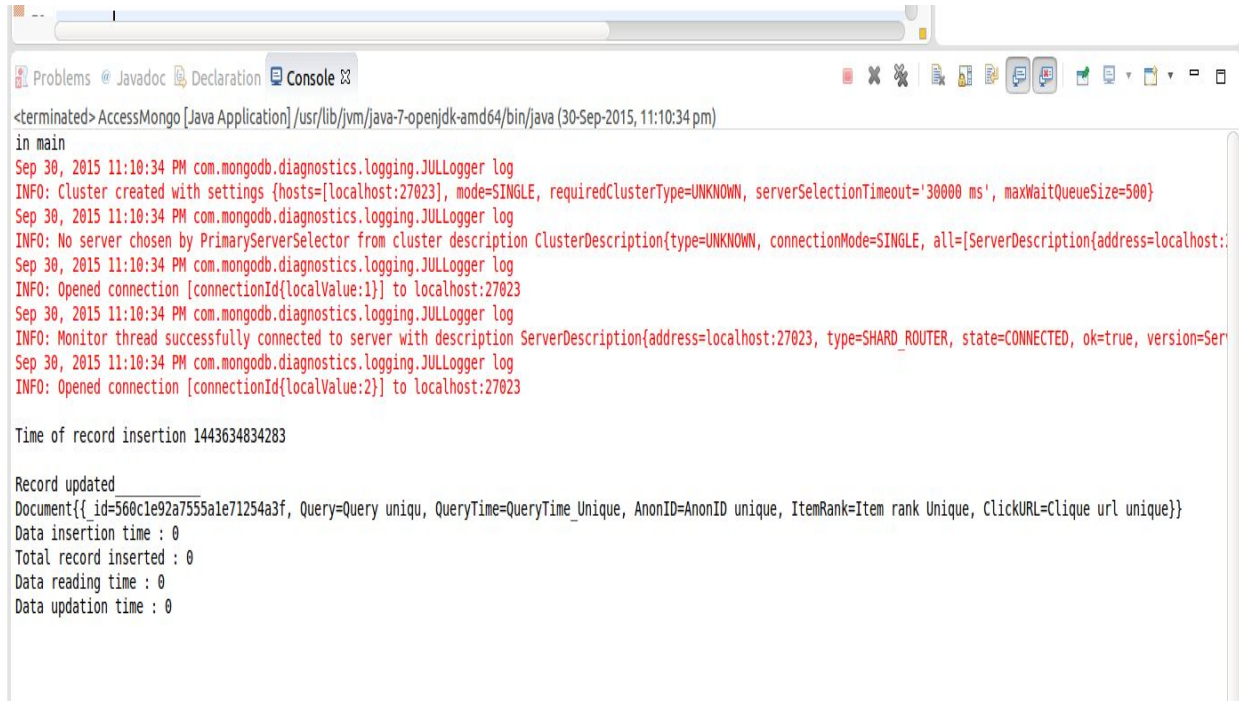
Updated one record from a query processor. Another query processor was set to check that record using a loop. Printed the Timestamp for each of the above operations

Consistency Check for MongoDB:

We used two query processor nodes for consistency measurement. Below are scenarios:

1. Updated one record from a query processor. Another query processor was set to check that record using a loop. It reported for updated record by time difference of **30 milliseconds**.

Insert command from one query processor:

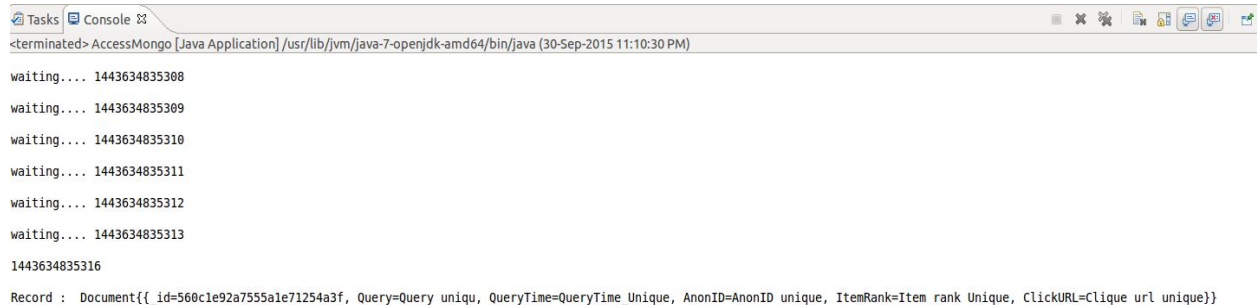


```
<terminated> AccessMongo [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (30-Sep-2015, 11:10:34 pm)
in main
Sep 30, 2015 11:10:34 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27023], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Sep 30, 2015 11:10:34 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: No server chosen by PrimaryServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, all=[ServerDescription{address=localhost:
Sep 30, 2015 11:10:34 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1}] to localhost:27023
Sep 30, 2015 11:10:34 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27023, type=SHARD_ROUTER, state=CONNECTED, ok=true, version=Ser
Sep 30, 2015 11:10:34 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2}] to localhost:27023

Time of record insertion 1443634834283

Record updated
Document{{_id=560c1e92a7555a1e71254a3f, Query=Query unique, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, ClickURL=Clique url unique}}
Data insertion time : 0
Total record inserted : 0
Data reading time : 0
Data updation time : 0
```

On insert from another query processor



```
Tasks Console
<terminated> AccessMongo [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (30-Sep-2015 11:10:30 PM)

waiting.... 1443634835308
waiting.... 1443634835309
waiting.... 1443634835310
waiting.... 1443634835311
waiting.... 1443634835312
waiting.... 1443634835313
1443634835316

Record : Document{{_id=560c1e92a7555a1e71254a3f, Query=Query unique, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, ClickURL=Clique url unique}}
```


Update command from one query processor:

```
Problems Javadoc Declaration Console
<terminated> AccessMongo [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (30-Sep-2015, 11:15:06 pm)
in main
Sep 30, 2015 11:15:06 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27023], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Sep 30, 2015 11:15:06 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: No server chosen by PrimaryServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, all=[ServerDescription{address=localhost
Sep 30, 2015 11:15:06 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1}] to localhost:27023
Sep 30, 2015 11:15:06 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27023, type=SHARD_ROUTER, state=CONNECTED, ok=true, version=Se
Sep 30, 2015 11:15:06 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2}] to localhost:27023

Time of record updation 1443635106974

Record updated
Document{{_id=560c1f97a7555a1fa1c12e54, Query=_____ updated Query_____, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, Clic
Data insertion time : 0
Total record inserted : 0
Data reading time : 0
Data updation time : 44
```

Update receive on another query processor

```
Record : Document{{_id=560c1f97a7555a1fa1c12e54, Query=Query unique, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, ClickURL=Clique url unique}}
1443635108002

Record : Document{{_id=560c1f97a7555a1fa1c12e54, Query=Query unique, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, ClickURL=Clique url unique}}
1443635108004

Record : Document{{_id=560c1f97a7555a1fa1c12e54, Query=Query unique, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, ClickURL=Clique url unique}}
1443635108007

Record : Document{{_id=560c1f97a7555a1fa1c12e54, Query=_____ updated Query_____, QueryTime=QueryTime_Unique, AnonID=AnonID unique, ItemRank=Item rank Unique, ClickURL=Clique url
```

Consistency Check for CouchDB:

On Document Updater side:

```
Problems Javadoc Declaration Console
<terminated> UpdaterConst [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (30-Sep-2015, 8:52:48 pm)
current time TimeStamp:1443626568634 : {"_id":"1","_rev":"8-4fd6e33716d69a07b43900f44fed89a8","name":"foo"}

The value is updated at time 1443626568634
```

On Document Reader Side:

```
44
TimeStamp:1443626569124 : {"_id":"1","_rev":"8-4fd6e33716d69a07b43900f44fed89a8","name":"foo"}

45
TimeStamp:1443626569133 : {"_id":"1","_rev":"8-4fd6e33716d69a07b43900f44fed89a8","name":"foo"}

46
TimeStamp:1443626569143 : {"_id":"1","_rev":"8-4fd6e33716d69a07b43900f44fed89a8","name":"foo"}

47
TimeStamp:1443626569417 : {"_id":"1","_rev":"9-c265fce091531a818e822eb5ef0e9273","name":"zzzzz"}

48
TimeStamp:1443626569517 : {"_id":"1","_rev":"9-c265fce091531a818e822eb5ef0e9273","name":"zzzzz"}

49
TimeStamp:1443626569526 : {"_id":"1","_rev":"9-c265fce091531a818e822eb5ef0e9273","name":"zzzzz"}
```

Timestamp Difference Obtained from update and read operation on two different Nodes:
Timestamp Difference= 1443626569417-1443626568634
= 783 milliseconds

Conclusion:

| MongoDB | CouchDB |
|--|---|
| Master/slave configuration (auto failover with replica sets) | Master-Master Configuration |
| Use when required high Consistency | Use when required high Availability |
| Better update-in-place than CouchDB | Update with conflict detection |
| Sharding built-in | Only CouchDB 2.0 Developer preview supports Sharding. Stable version is not available |
| Use when data changes too much | Use when data changes occasionally |

Final Conclusion: MongoDB is faster out of the box; CouchDB is safer out of the box.

Note: High quality images are given in separate folder attached with submission.