Test of Oracle JSON support in the view of CMS JSON data

Sartaj Singh Baveja

Supervisors : Katarzyna Maria Dziedziniewicz-Wójcik (CERN IT)
Valentin Kuznetsov (CMS)

Who am I?

- Alumnus of NSIT

- Will join Lawrence Berkeley Labs as a Research scholar

- Bhangra freak

Importance of JSON

- Used in web services responses
- Easy to parse which makes it easier to use

```
json = {
    "firstName": "John",
    "lastName": "Smith",
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        "212 555-1234",
        "646 555-4567"
    ]
}
```

CMS JSON data

- 200k-300k documents are generated everyday
- Size of each document : ~12kB i.e. 3GB/day or 1-2TB/year
- Each document
  - Deep Nested Structure
  - Represents CMS Framework Job Report

```
# CMS FWJR data structure
{"meta_data": {"agent_ver": "1.0.14.pre5",
               "fwjr_id": "1-0",
               "host": "a.b.com",
               "jobtype": "Processing",
               "jobstate": "success",
               "ts": 1456500229},
 "LFNArray": ["/store/file1.root",
              "/store/file2.root",
              "/lfn/fallbackfile.root", "/lfn/skipedfile.root"],
 "LFNArrayRef": ["fallbackFiles",
                 "outputLFNs",
                 "lfn",
                 "skippedFiles",
                 "inputLFNs"],
 "PFNArray": ["root://file1.root",
              "root://file2.root",
              ],
 "PFNArrayRef": ["inputPFNs", "outputPFNs", "pfn"],
 "steps": [{"name": "cmsRun1",
            "analysis": {},
            "cleanup": {},
            "logs": {},
            "errors": [
                  {
                      "details": "An exception",
                      "type": "Fatal Exception",
                      "exitCode": 8001
                  }
              ],
            "input": [{"catalog": "",
                       "events": 6893,
```

Existing Solution | MongoDB

- Used by CMS as a buffer before putting docs to HDFS.
- Open-source NoSQL database
- Provides high performance, high availability, and automatic scaling
- Avoids table based relational database structure
- Schema Less

JSON in Oracle Database

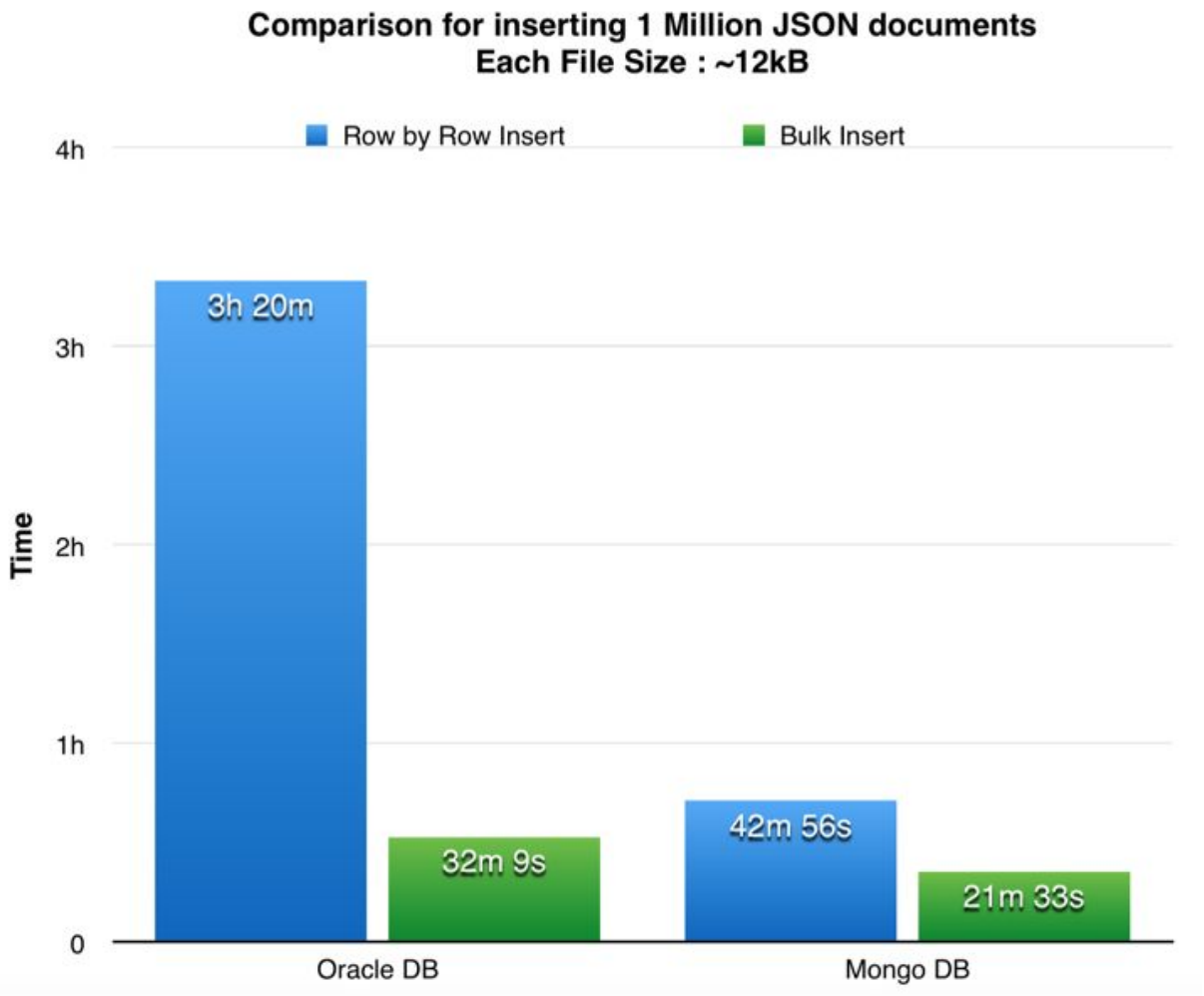
- Recently introduced JSON support with relational database features such as
  - Commits / Rollbacks
  - Indexing
  - Declarative Querying (join JSON data with relational data, project data relationally)
  - Views

- Dot notation to access fields of the embedded document

```
SELECT po.po_document.Requestor FROM j_purchaseorder po;
```

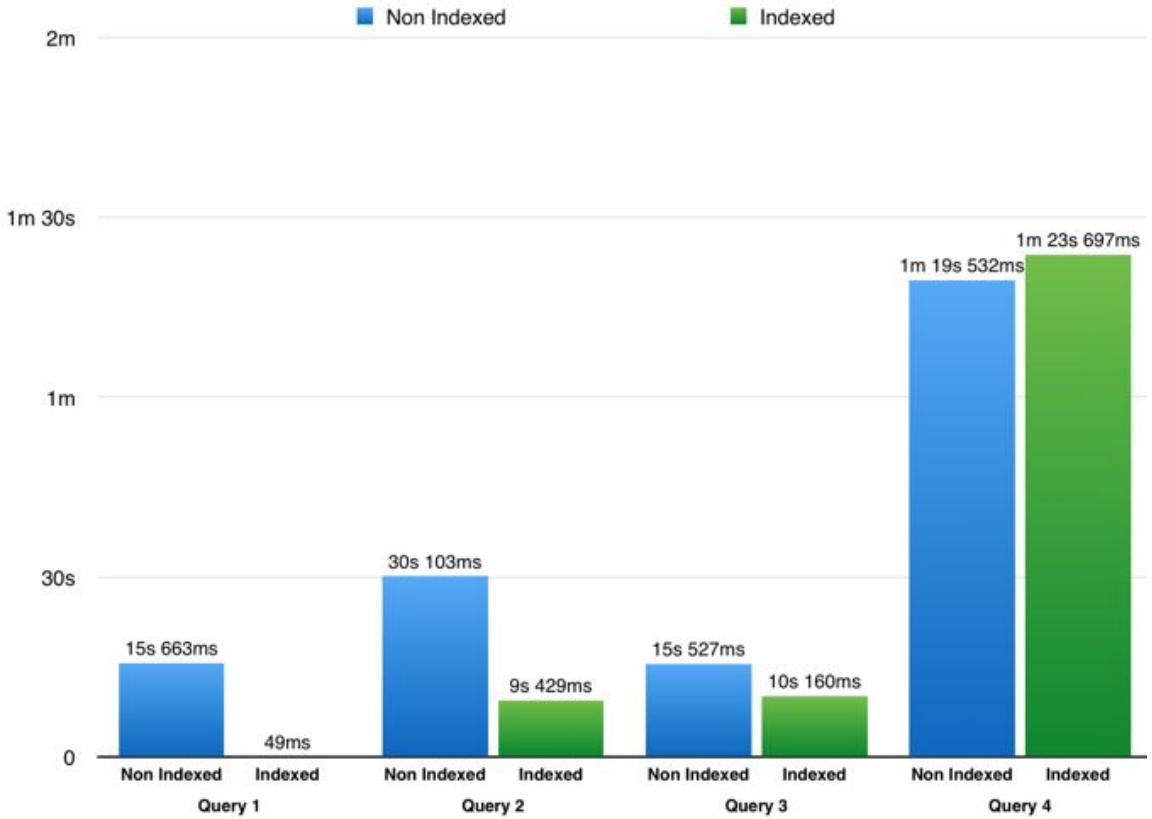
RESULTS

Injection Rate



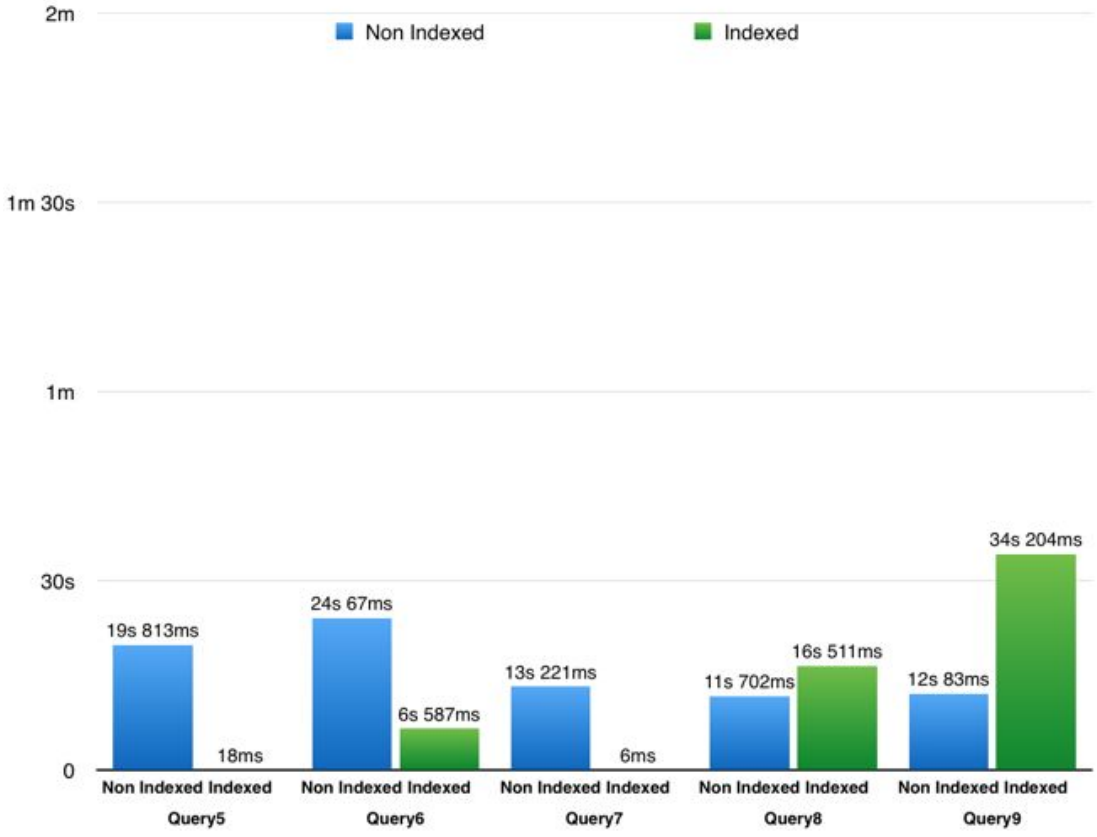
MongoDB Indexed vs Non Indexed Queries

Query 1/2/3 : Search for specific string
Query 4 : Aggregate data

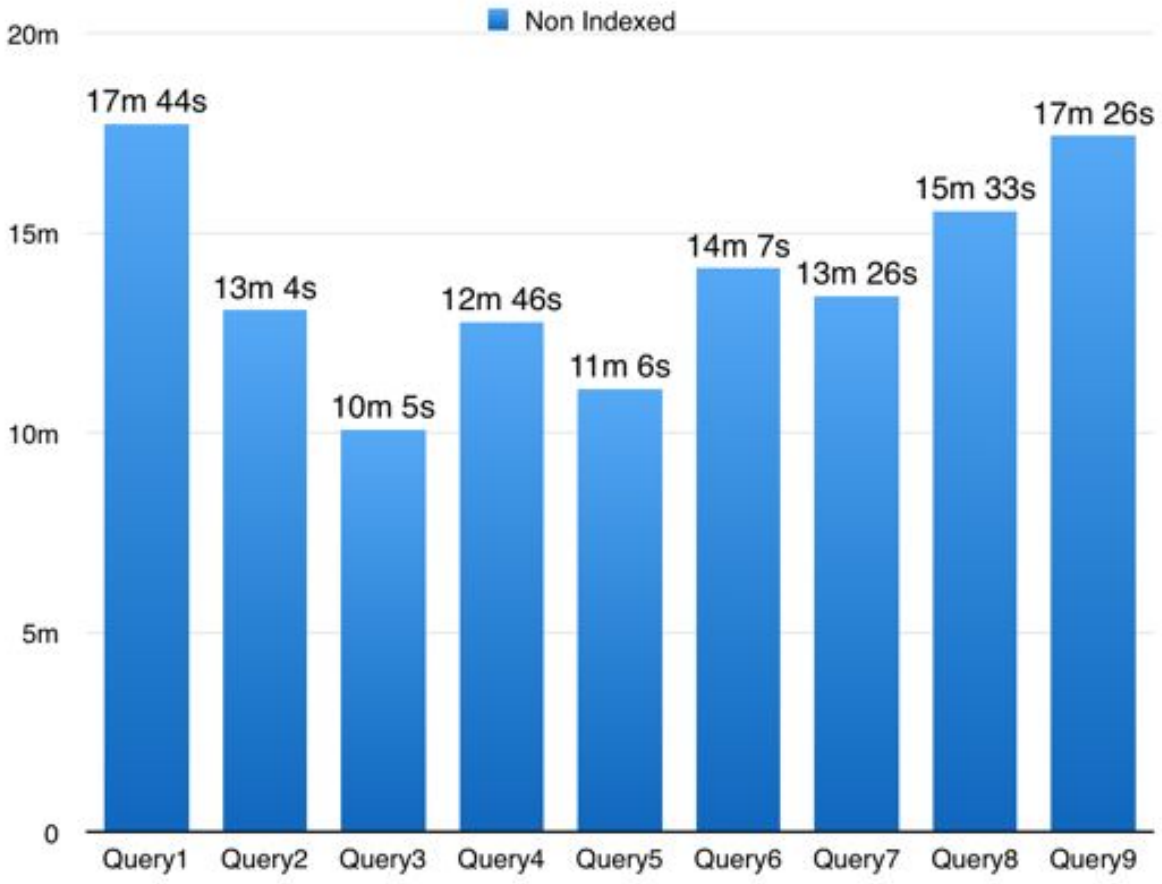


MongoDB Indexed vs Non Indexed Queries

Query 5 : Find a string
Query 6/7 : Find records based on provided pattern
Query 8 : Logical Query Operators
Query 9 : Comparison Query Operators



Oracle Non Indexed Queries



MongoDB vs Oracle JSON Library

Higher Data Size 1.3 GB for 1M documents	**Lower** Data Size 54MB for 1M documents
223 MB Index Size	**1.14 GB** Index Size
Queries are **not** atomic	Queries **are** atomic
Lower Insert Time	**Higher** Insert Time
Indexing **works** perfectly	Indexing on arrays **doesn't work**
No such errors	**Internal errors** such as "No Data to be read from socket" occur frequently

Future Work

- Compare the performance with other databases
  - Postgres
  - MySQL
- Indexing with Oracle JSON Library
- CPU / RAM metrics

Thank You!

sartajcorp@gmail.com

www.sartajsingh.in

sartaj10