# Concept Identification Using Lookup Approach

Mohammed Asif     Praveen Kumar     Senthil Madhappan

## Problem Statement

- Design an engine to tag the concepts for the given text using lookup approach. The design should be scalable and update-friendly.
- List of all valid concepts reside on a HBase / NoSQL database and the initial set would be 10 million concepts.
- Benchmark the performance for tagging the document of size 250 words with concepts.

## Motivation

Concept identification is becoming an area of great interest and has been extensively used in various applications, notably

- Information Retrieval & Extraction
  - ➢ Concept Based Indexing
  - ➢ Summarization
  - ➢ Question Answering
- Data Mining
  - ➢ Document Classification & Categorization
  - ➢ Ontology Engineering
  - ➢ Literature Based Knowledge Discovery

## High-level Solution Approach

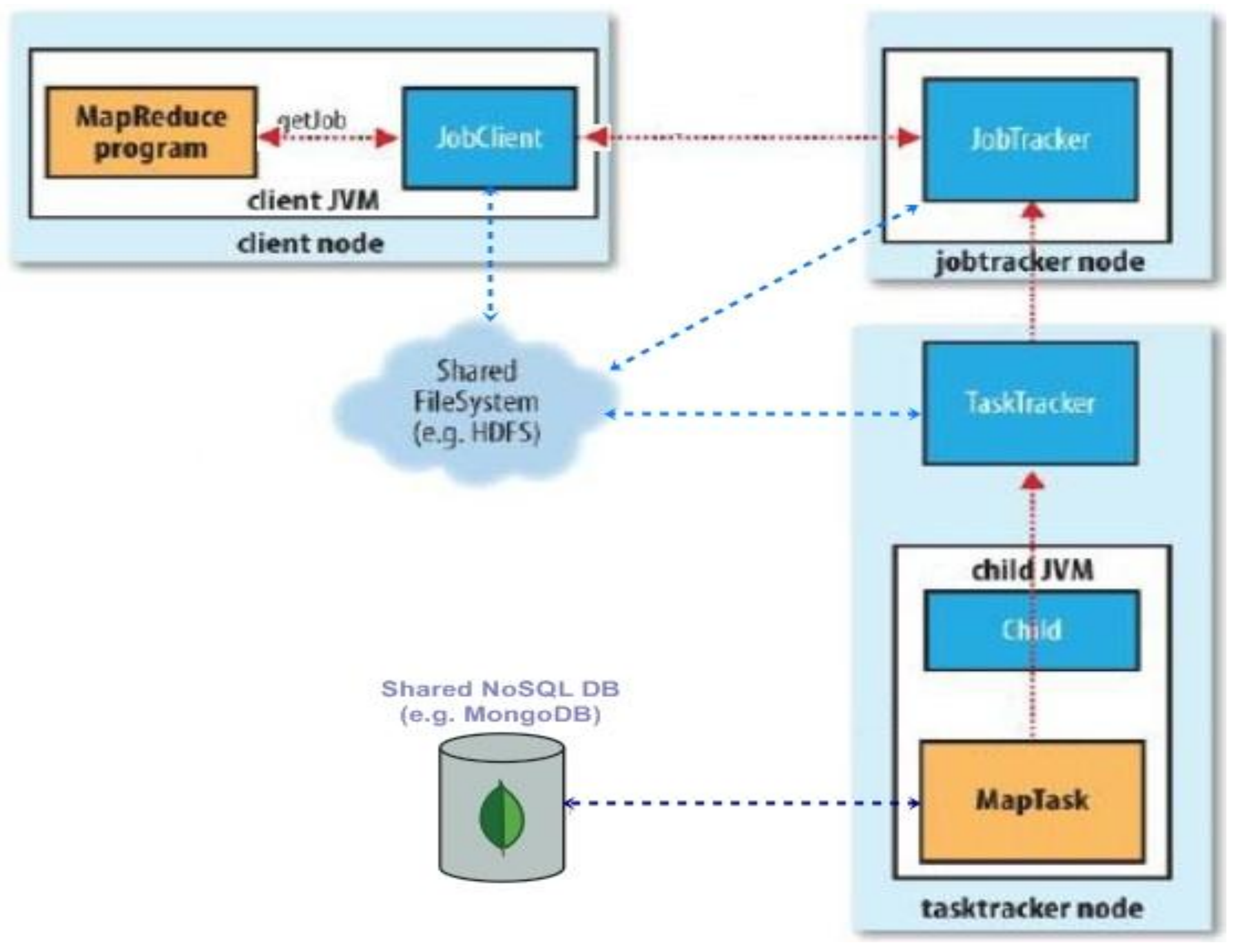
**Issues:**

- Loading the complete known concepts into memory is expensive in terms of memory foot print that gets added to the JVM heap.
- Lookup cost to check for a concept from known concepts.
- Number of lookups required to find an existing concept.

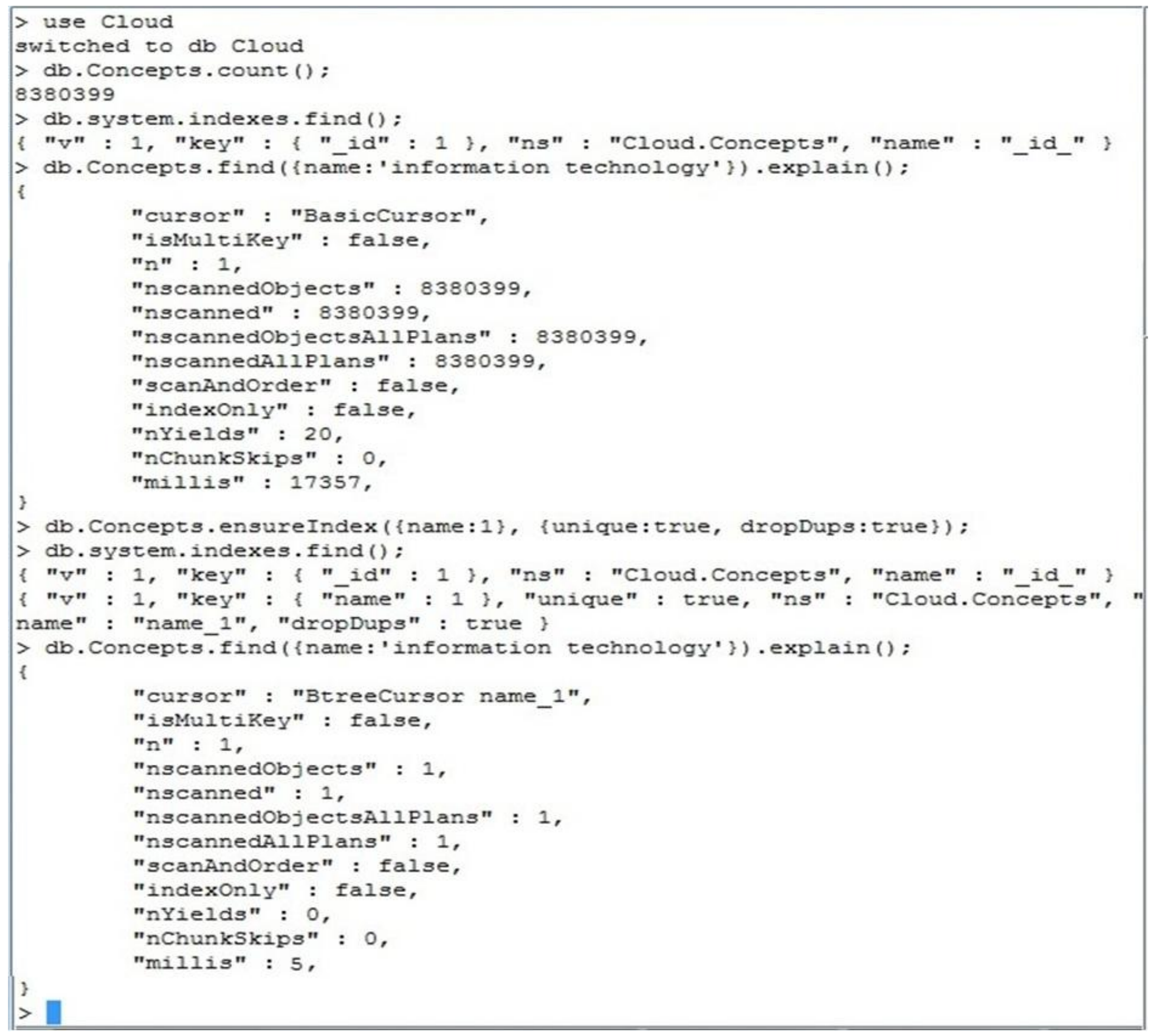
**Solution Approach:**

- Store the concepts on a central storage off the JVM heap.
  - ➢ The possible options would be HBase (or) NoSQL database like MongoDB
- Time required for the lookup operation can be reduced by
  - ➢ Using local/distributed caches between the Hadoop task and the storage.
  - ➢ Identifying the words that will never form a concept and avoid lookup for those words e.g. stop words.

## Architecture



## Performance Tuning

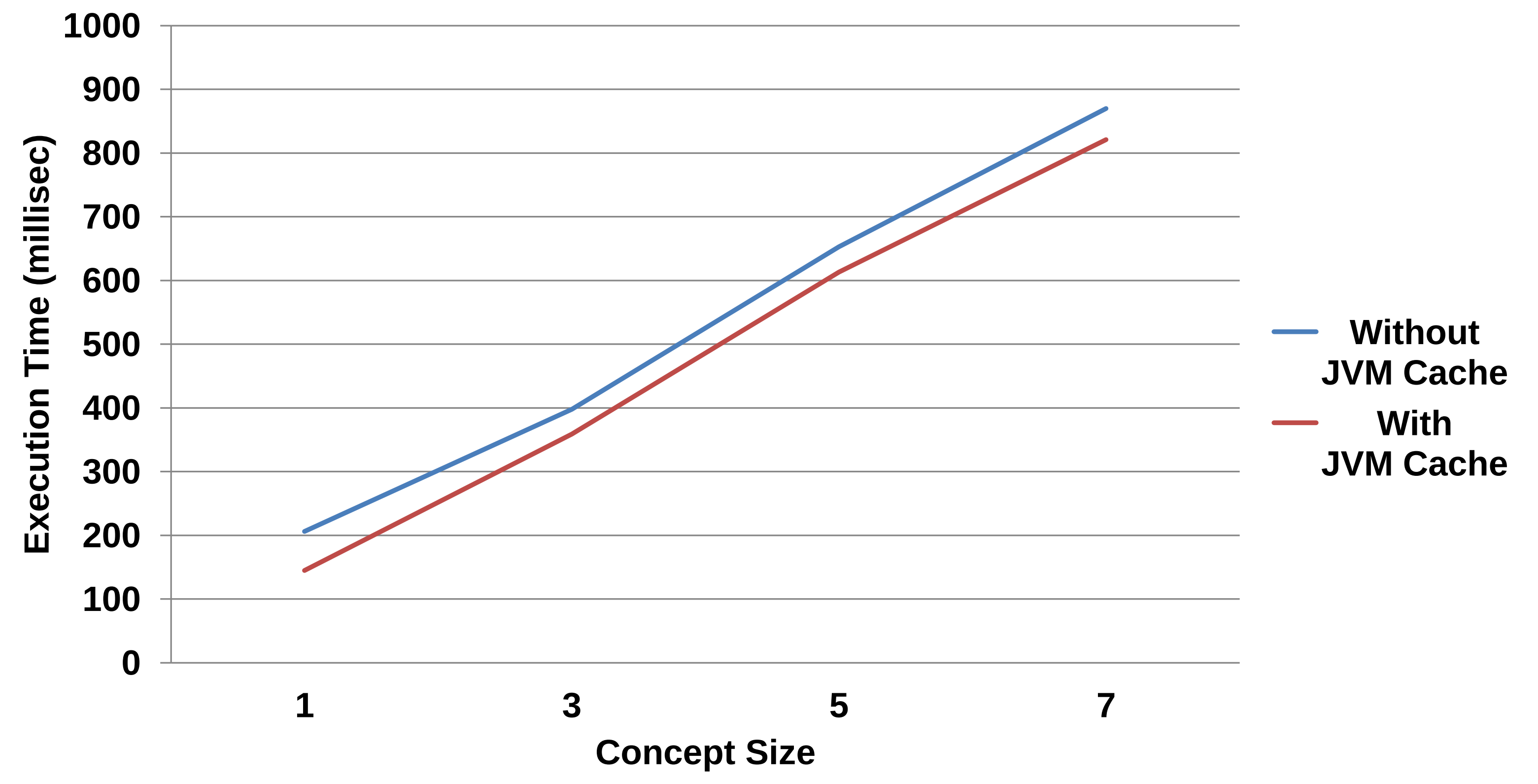


```
> use Cloud
switched to db Cloud
> db.Concepts.count();
8380399
> db.system.indexes.find();
{ "v" : 1, "key" : { "_id" : 1 }, "ns" : "Cloud.Concepts", "name" : "_id_" }
> db.Concepts.find({name:'information technology'}).explain();
{
        "cursor" : "BasicCursor",
        "isMultiKey" : false,
        "n" : 1,
        "nscannedObjects" : 8380399,
        "nscanned" : 8380399,
        "nscannedObjectsAllPlans" : 8380399,
        "nscannedAllPlans" : 8380399,
        "scanAndOrder" : false,
        "indexOnly" : false,
        "nYields" : 20,
        "nChunkSkips" : 0,
        "millis" : 17357,
}
> db.Concepts.ensureIndex({name:1}, {unique:true, dropDups:true});
> db.system.indexes.find();
{ "v" : 1, "key" : { "_id" : 1 }, "ns" : "Cloud.Concepts", "name" : "_id_" }
{ "v" : 1, "key" : { "name" : 1 }, "unique" : true, "ns" : "Cloud.Concepts", "
name" : "name_1", "dropDups" : true }
> db.Concepts.find({name:'information technology'}).explain();
{
        "cursor" : "BtreeCursor name_1",
        "isMultiKey" : false,
        "n" : 1,
        "nscannedObjects" : 1,
        "nscanned" : 1,
        "nscannedObjectsAllPlans" : 1,
        "nscannedAllPlans" : 1,
        "scanAndOrder" : false,
        "indexOnly" : false,
        "nYields" : 0,
        "nChunkSkips" : 0,
        "millis" : 5,
}
>
```

## Hardware/Software Configuration

| | | |
|---|---|---|
| Virtual Machine | Host OS | Windows 7 (64bit) |
| | Guest OS | Ubuntu 12.04 (64bit) |
| | RAM | 1.5GB |
| | Processor | 2 |
| | Network | Bridged Adapter |
| Hadoop | | Pseudo-Distributed |
| Mongo DB | | Single Centralized Server |

## Results



**Dataset Used:**

- Seeded Concepts – 10 Million Wikipedia Titles
- Lookup Documents – 100 Gynanidhi Documents (English)

## Future Work

- Experiment with MongoDB replica set configuration
  - ➢ Use local database server instead of single centralized server over network.
- Experiment with various database tuning strategies
  - ➢ Create additional index and/or change database schema.
  - ➢ Tuning database cache parameters.
- Experiment with fully-distributed Hadoop cluster environment.
- Experiment with distributed JVM cache.