

Big Data System

Final Project

Elaine Pei-Ling Chong

[Company Name]

# Table of Contents

[1 Table of Contents 2](#_Toc501135760)

[2 Introduction 3](#_Toc501135761)

[3 Methodology 3](#_Toc501135762)

[4 Dataset 4](#_Toc501135763)

[4.1 MongoDB 5](#_Toc501135764)

[4.2 Redis 5](#_Toc501135765)

[4.3 Neo4j 6](#_Toc501135766)

[5 Flexibility 6](#_Toc501135767)

[6 Scalability 7](#_Toc501135768)

[7 Availability 9](#_Toc501135769)

[8 Conclusion 9](#_Toc501135770)

[9 References 10](#_Toc501135771)

# Introduction

With the rise in databases, different databases are used to fit the needs of an individual application to store different kind of data.

This report aimed to select the best database model to deal with data from different colleges in the word, which may have varying numbers of continuous elements but still the same overall structure for continuous assessment and final exam with the consideration of the flexibility, scalability and availability.

NoSql (Not-only SQL) databases are chose because they are schemaless databases which provide higher flexibility compares to SQL databases which are relational databases. Shard and replication are carried out for selected databases to offer high scalability and high availability.

# Methodology

The approach I used for the solution is polyglot persistence. Polyglot persistence means using different type of data storage technologies in different ways and take advantage of the databases to enhance the application. I have chosen MongoDB, Redis and Neo4j as my solution.

# Dataset

The dataset used to store to the databases is given by lecture and can be found in Moodle (<https://2018-moodle.dkit.ie/pluginfile.php/415016/mod_resource/content/1/FinalProjectData1718.csv)>

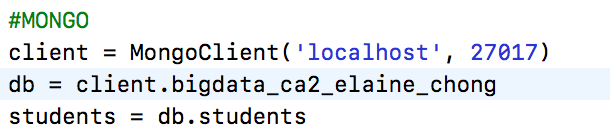
The data file contains multiple columns which describe student’s result.

1. studentid: The Id of student
2. Lab 1: The result of lab 1
3. Christmas Test: The result of Christmas test
4. Lab 2: The result of lab 2
5. Easter Test: The result of Easter test
6. Lab3: The result of lab 3
7. parttimejob: 1 means have part-time job; 0 means do not have part-time job
8. Exam Grade: The result of exam grade

To store the data inside the CSV file to the database, I have read the file and convert into JSON list and save to a JSON file. reader.\_\_next\_\_() is used to skip the first line which is the header of the file.



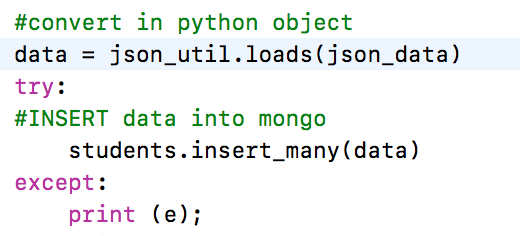
Different type of data storages has a different way to upload to the databases, but they are all start with connecting to the database in python file. To connect to the databases, I downloaded packages such as pymongo for MongoDB, Redis for Redis and py2neo for Ne4j.

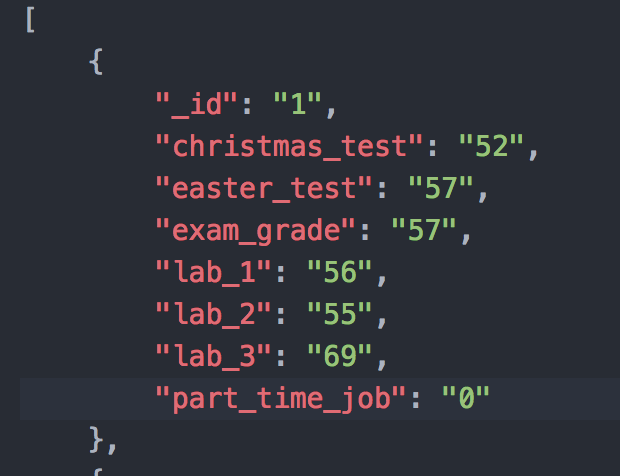


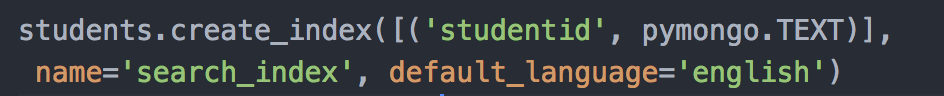
 

## MongoDB

After deleted all the records to prevent duplicate, I have uploaded the data to the databases. The data needed to convert into python object before uploading to mongo, in mongo, it allows the user to insert many records at the same time. The format for JSON shown. Indexing in Mongo can improve the query performance. Thus I created the index for field studentId.







## Redis

In Redis, I set pipeline to speed up the query and use hash multiple sets (hmset) to store the data.

The format for hmset in Redis is

hmset key field1 value 1 field 2 value2

Example:

hmset student:1 lab\_1 56 christmas\_test 52

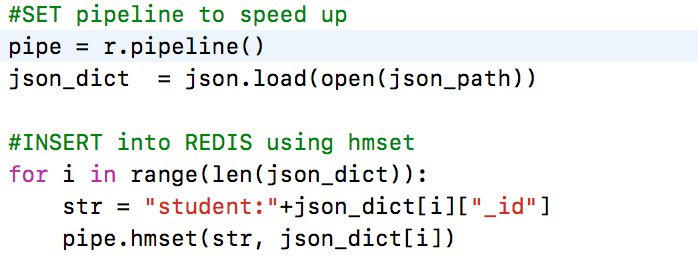
The format for hmset in redis (Python) is

r.hmset(key, {field1: value1, field2:value2})

Example

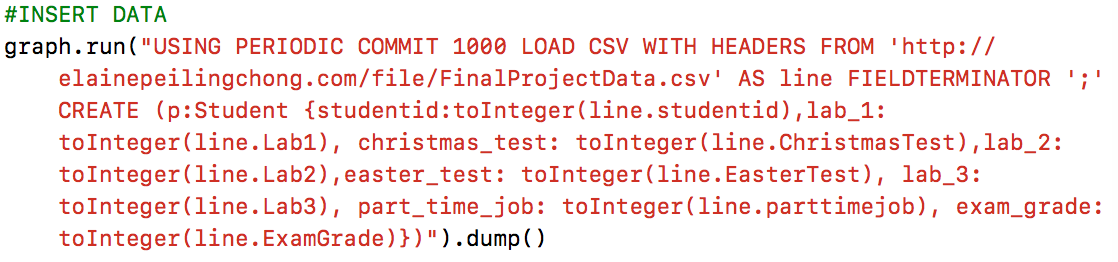
r.hmset(student:1, {lab\_1: 56, christmas\_test:52})

It is easy to get the information of student with Id=1 in Redis by executing r.hgetall(“student:1”), it will return all the information about the student.



## Neo4j

For Neo4j, first I upload the CSV file to my host and insert the data by access the URL. I tried to access the URL in Moodle, but it needed authentication so I uploaded to my server. I have created the label for every student as Student and converted all the data into an integer.



# Flexibility

MongoDB is a document database. The data stored in a specific format such as JSON. It allows saving information without creating a table with a fixed structure.

Different colleges might have different structure to store their information no matter in module or the college itself, thus, MongoDB is recommended to store all the information because it is a schema-less database, it can store data in flexible, the data structure can be changed any time which means that the users are allowed to add any new data even the databases is already create and the collection is set up.

For example, if college A wants to add an additional field to store their second campus's address, the college can add the record themselves without having any impact on other colleges' database structure.

Redis is a key-value data store, and it is used for session cache to keep user’s information even if he connects to a different server.

If I want to know if A student is the top student in the class and will the friends of A student have the high grade as well, Neo4j is used to do it. Neo4j is a graph database which is suitable recommendation system and to analyse the relationship for further investigation.

# Scalability

Every college can store their information on their machine rather than to the machine which is owned by the developer. Sharding can be performed in order to enable colleges to save their information and to improve the scalability. Sharding means rather than store the whole dataset in 1 server, it is split into different servers to increase the speed and performance of the entire application. Sharding can be carried out in all three databases chosen.

In Redis, Redis Cluster is used to provide automatically sharded across multiple servers. Neo4j uses High Availability(HA) mode as scalability package.

To remain available even when one or more shards are unavailable, other shards can still read or write from the available shards. When the master is failing and not possible to communicate, Neo4j High Availability and Redis's Redis Cluster provide a master-slave model where every cluster has 1 to N replicas. New master (primary member) will be elected from the slave (secondary member) when the master fails.

Method to do sharding in MongoDB shown below.

1. Create folder in the machine

mkdir ./mongo4

mkdir ./mongo5

mkdir ./mongoShard

1. Start the servers

mongod --configsvr --replSet configRepl --dbpath ./mongoShard --port 27016

mongod --shardsvr --replSet shard27014Repl --dbpath ./mongo4 --port 27014

mongod --shardsvr --replSet shard27015Repl --dbpath ./mongo5 --port 27015

1. Start the client

mongo localhost:27014

mongo localhost:27015

mongo localhost:27016

1. Run this command

In mongo client 27016

rs.initiate({

\_id: "configRepl",

configsvr:true,

members: [

{\_id: 0, host: "localhost:27016"}]

})

In mongo client 27014

rs.initiate({

\_id: "shard27014Repl",

members: [

{\_id: 0, host: "localhost:27014"}]

})

In mongo client 27015

rs.initiate({

\_id: "shard27015Repl",

members: [

{\_id: 0, host: "localhost:27015"}]

})

1. Start the configuration database
   1. sever

mongos --configdb configRepl/localhost:27016 --port 27020

* 1. client

mongo localhost:27020

1. Run the commend below in client 27020
   1. Use the database admin

use admin

* 1. Add the shard

db.runCommand({addshard : "shard27014Repl/localhost:27014"})

db.runCommand({addshard : "shard27015Repl/localhost:27015"})

db.runCommand({enablesharding : "bigdata\_ca2\_elaine\_chong"})

db.runCommand({shardcollection : "bigdata\_ca2\_elaine\_chong.students", key : {\_id : 1}})

* 1. Check how the data split

db.students.getShardDistribution()

# Availability

MongoDB, Redis and Neo4j support master-slave replication which enable data from one database server to be copied to be to other servers. If the master downs, the automatic failover will allow a secondary member to become the primary member. In MongoDB and Redis, only the master has the permission to write, but in Neo4j, both master and slave can write.

In order to do Replication in Mongo, the step below is carried out.

1. Create file

mkdir ./mongo1

mkdir ./mongo2

mkdir ./mongo3

1. Start the servers with different port number

mongod --replSet bigdata\_ca2\_elaine\_chong -dbpath ./mongo1 -port 27011

mongod --replSet bigdata\_ca2\_elaine\_chong -dbpath ./mongo2 -port 27012

mongod --replSet bigdata\_ca2\_elaine\_chong -dbpath ./mongo3 -port 27013

1. Start the clients

mongo localhost:27011

mongo localhost:27012

mongo localhost:27013

1. In port 27011, initiates a replica set

rs.initiate({

\_id: bigdata\_ca2\_elaine\_chong,

members: [

{\_id: 1, host: 'localhost:27011'},

{\_id: 2, host: 'localhost:27012'},

{\_id: 3, host: 'localhost:27013'}]})

1. In port 27012 and port 27013, run the command below to make sure they are agreed to be the slave and read the data

rs.slaveOk(true)

In Redis, it is easier to do replication.

1. Copy the redis.conf and set it to another name such as redis-s1.conf.
2. Modify the redis-s1.conf

Port 6380

Slaveof 127.0.0.1 6379

1. Start the servers with different configuration file
2. Start the clients with corresponding port

It is very easy to backup the data in Neo4j because it has a simple backup tool called neo-4j. Neo4j Enterprise edition provides more function than the community edition such as High Availability for replicate and sharding. Since I was using the community edition, I was not allowed to do that.

# Conclusion

This report shows the solution for a company to choose the appropriate databases to store the data. MongoDB, Redis and Neo4j are chosen to perform different functions. Indexing in MongoDB and pipelining in Redis enable to improve the performance of the databases. Use sharding with replication can provide high scalability and high availability.

# References

Fowler, M. and Sadalage, P. (2012). *NoSQL distilled*. Boston, Mass.: Addison-Wesley.

Serra, J. (2015). *What is Polyglot Persistence?*. [online] James Serra's Blog. Available at: http://www.jamesserra.com/archive/2015/07/what-is-polyglot-persistence/ [Accessed 5 Dec. 2017].

Branson, T. (n.d.). *Two main approaches to database scalability*. [online] Infosecurity Magazine. Available at: https://www.infosecurity-magazine.com/opinions/a-glimpse-into-database-scalability/ [Accessed 6 Dec. 2017].

cases, T. (2017). *Top 5 Redis use cases*. [online] ObjectRocket. Available at: https://www.objectrocket.com/blog/how-to/top-5-redis-use-cases [Accessed 9 Dec. 2017].

Quabase.sei.cmu.edu. (n.d.). *Replication for Backup*. [online] Available at: https://quabase.sei.cmu.edu/mediawiki/index.php/Replication\_for\_Backup [Accessed 9 Dec. 2017].

Redis.io. (n.d.). *Replication – Redis*. [online] Available at: https://redis.io/topics/replication [Accessed 10 Dec. 2017].

Groups.google.com. (2012). *Google Groups*. [online] Available at: https://groups.google.com/forum/#!topic/redis-db/cT\_1G7D2wdA [Accessed 12 Dec. 2017].

Redis.io. (n.d.). *Redis cluster tutorial – Redis*. [online] Available at: https://redis.io/topics/cluster-tutorial [Accessed 12 Dec. 2017].

Montag, D. (2013). Understanding Neo4j Scalability. *Neo Technology*. [online] Available at: https://go.neo4j.com/rs/710-RRC-335/images/Understanding%20Neo4j%20Scalability%282%29.pdf?\_ga=2.268609649.1643455312.1513352634-656938311.1512670522 [Accessed 14 Dec. 2017].