

An Empirical Study of NoSQL Databases by Using MongoDB Databases

Wen-Chen Hu
Department of Computer Science
University of North Dakota
Grand Forks, ND 58202
wenchen@cs.und.edu

Naima Kaabouch
Department of Electrical
Engineering
University of North Dakota
Grand Forks, ND 58202
naima.kaabouch@engr.und.edu

Hung-Jen Yang
Department of Industrial Technology Education
National Kaohsiung Normal University
Kaohsiung City, Taiwan
hjyang@nknucc.nknu.edu.tw

Abstract

Relational databases have dominated the database markets for decades because they perform extremely well for traditional applications like electronic commerce and inventory systems. However, the relational databases do not suit some of the contemporary applications such as big data and cloud computing well because of various reasons like their low scalability and unable to handle a high volume of data. NoSQL (not only SQL) databases are part of the solution for developing those newer applications. The approach they use is different from the one used by relational databases. This paper discusses NoSQL databases by using an empirical instead of theoretical approach. Other than introducing the types and features of generic NoSQL databases, practical NoSQL database programming and usage are shown by using MongoDB, a NoSQL database. A summary about NoSQL databases is given at the end of this paper.

1 Introduction

Much of current data processing requires horizontal scaling, faster speed, and processing different kinds of data and relational databases are not able to meet the requirements adequately. A NoSQL database is to meet the needs by providing a simple and efficient mechanism for data storage and retrieval. The approach it uses is different from the one used by a relational database. An RDBMS (relational database management system) is a general-purpose data store, whereas a NoSQL database is normally a key-value store for simple insertion and retrieval operations. This paper discusses the NoSQL databases by using an empirical approach. A NoSQL database, MongoDB, is used to show how to program and use NoSQL databases.

1.1 Features of NoSQL Databases

Many of contemporary applications require the features of being generic, distributed, open-source, and scalable. The features are also some of the benefits of NoSQL databases including (Han, E, Le, & Du, 2011):

- NoSQL databases are able to handle structured, semi-structured, and unstructured data, not just semi-structured data for relational databases,
- Compared to relational databases, NoSQL databases are simple and therefore are faster.
- Most NoSQL databases allow object-oriented programming, which is the most popular programming paradigm.
- NoSQL databases include the feature of horizontal scalability, which is the ability to connect multiple NoSQL databases so that they work as a single logical unit.

Relational databases can no longer meet the requirements of some of the mobile-age applications, which desire simple, efficient, and scalable databases. The emergence of NoSQL databases is to meet the needs of newer applications. However, from the history of databases, none of other databases has successfully replaced relational databases, which have the greatest advantages of easy-to-use. NoSQL databases do not intend to replace relational databases. Instead, they are a great complement to relational databases.

1.2 Types of NoSQL Databases

Unlike relational databases using tables, NoSQL databases do not use a standard data structure. There are several types of NoSQL databases including (Zaki, 2014):

- *Document databases* (e.g., *MongoDB*), which pair each key with a document (a document could be collections, tags, metadata, or directory hierarchies).
- *Graph stores* (e.g., *Neo4J* and *HyperGraphDB*), which are best for storing network (such as social networks) information.

- *Key-value stores (e.g., Riak and Oracle NoSQL Database)*, which are the simplest NoSQL databases, where each item includes a pair, key and value, like associative arrays.
- *Wide-column stores (e.g., Cassandra and HBase)*, which store columns (instead of rows) of data (the size of the data is usually higher than the size of the relational databases' row data).

1.3 Paper Organization

This paper discusses the next generation databases, NoSQL databases, by using MongoDB databases. The rest of this paper is organized as follows. The next section discusses how to download and install a MongoDB database. MongoDB shell interface is explained in Section 3. Sections 4 and 5 discuss how to connect Java and ASP.NET to MongoDB databases, respectively. A summary of this research is given in the final section.

2 MongoDB Database Download and Installation

Several NoSQL databases are available on the market and each one is not significantly different from the others. This study picks the MongoDB databases (MongoDB, Inc., n.d.) because of their simplicity and high efficiency. This section discusses the steps of how to download and install a MongoDB database as follows:

- Download a MongoDB database.* Download the NoSQL database MongoDB at MongoDB Downloads (<http://www.mongodb.org/downloads>). The one the authors downloaded is Windows 64-bit `mongodb-win32-x86_64-2008plus-2.4.8.zip`.
- Install the MongoDB database.* Unzip the zip file `mongodb-win32-x86_64-2008plus-2.4.8.zip` and it will create a folder `mongodb-win32-x86_64-2008plus-2.4.8`, which again includes one folder: `bin`, and three files: (i) `README`, (ii) `THIRD-PARTY-NOTICES`, and (iii) `GNU-AGPL-3.0`.
- Download a Java SE (Java Platform, Standard Edition).* Download a Java SE at Java SE Downloads (<http://www.oracle.com/technetwork/java/javase/downloads>). The one the authors downloaded is Java SE Development Kit 7u45 for Windows x86 `jdk-7u45-windows-i586.exe`.
- Install the Java SE.* Doubly click the file `jdk-7u45-windows-i586.exe` and it will start installing the Java SE for you. After installation, it creates a folder such as `C:\Program File (x86)\Java`, which again includes two folders: (i) `jdk1.7.0_45`, where the JDK (Java Development Kit) includes a complete JRE plus tools for developing, debugging, and monitoring Java applications, and (ii) `jre7`, where the JRE (Java Runtime Environment) contains everything required to run Java applications on your system.

- (e) *Download an Eclipse IDE for Java Developers.* Download an Eclipse IDE for Java Developers at Eclipse IDE for Java Developers. The one the authors downloaded is `eclipse-java-europa-winter-win32.zip`.
- (f) *Install the Eclipse IDE for Java Developers.* Unzip the zip file `eclipse-java-europa-winter-win32.zip` and it will create a folder `eclipse`, which again includes four folders: (i) configuration, (ii) features, (iii) plugins, and (iv) readme.
- (g) *Download a Mongo Java Driver.* Download a Mongo Java driver at Mongo Java driver. The one the authors downloaded is `mongo-2.10.1.jar`, which is put into the folder of Mongo DB such as `C:\MongoDB-SDK\bin`.

3 The MongoDB Shell Interface

The MongoDB shell is an interface for systems administrators as well as a way for developers to test queries and operations directly with the MongoDB databases. This section discusses various MongoDB shell commands (Seguin, 2013).

3.1 Starting up a MongoDB Database Server

In order to use a MongoDB database, a single MongoDB database server must be activated by using the following three steps:

- (a) Create a folder `C:\data\db`,
- (b) `cd` (change directory) into the `bin` folder of the Mongo DB directory such as `C:\MongoDB\bin` to start the MongoDB database server, and
- (c) Enter the command `mongod`, which is used to start up a MongoDB database server.

Figure 1 shows a screenshot of starting up a MongoDB database server.

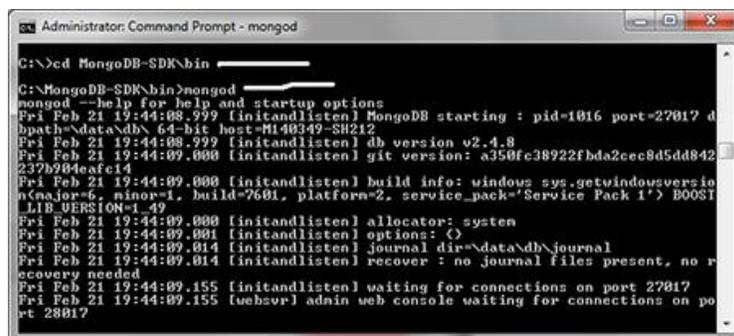


Figure 1: A Screenshot Showing Starting up a MongoDB Database Server.

3.2 Activating mongo and Entering MongoDB Commands

From another system prompt, start the MongoDB by issuing the `mongo` command. `mongo` is an interactive JavaScript shell interface to MongoDB database. It allows the system administrators and developers to communicate with the database directly. Figure 2 shows a screenshot of a list of MongoDB (shell) commands entered. The MongoDB commands are

```
> db.createCollection( "persons" )
> db.persons.insert( { personID: "P1", name: "Poke Mon" } )
> db.persons.insert( { personID: "P2", name: "Digi Mon" } )
> db.persons.insert( { personID: "P3", name: "Sponge Bob" } )
>
> db.createCollection( "passwords" )
> db.passwords.insert( { pID: "P1", password: "secret" } )
> db.passwords.insert( { pID: "P2", password: "hush" } )
> db.passwords.insert( { pID: "P3", password: "secret" } )
```

which create the collections `persons` and `passwords` and enter three documents for each.

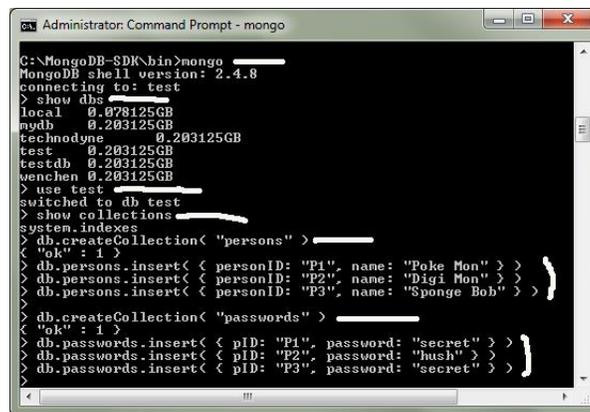


Figure 2: A Screenshot Showing a List of MongoDB (Shell) Commands and System Responses.

After connecting to the database server, which runs as `mongod`, the developers can start using the `mongo` shell to manage their databases. Table 1 shows some common MongoDB shell commands.

Command	Description
<code>db</code>	Reports the name of the current database.
<code>db.help()</code>	Help on db methods
<code>db.mycoll.help()</code>	Help on collection methods
<code>DBQuery.shellBatchSize = x</code>	Sets default number of items to display on shell.

<code>exit</code>	Quits the mongo shell.
<code>help admin</code>	Administrative help
<code>help connect</code>	Connecting to a db help
<code>help keys</code>	Key shortcuts
<code>help misc</code>	Miscellaneous things to know
<code>help mr</code>	Mapreduce
<code>it</code>	Result of the last line evaluated; use to further iterate
<code>rs.help()</code>	Replica set helpers
<code>sh.help()</code>	Sharding helpers
<code>show collections</code>	Shows collections in current database.
<code>show dbs</code>	Shows database names.
<code>show logs</code>	Shows the accessible logger names.
<code>show profile</code>	Shows most recent <code>system.profile</code> entries with time ≥ 1 ms.
<code>show users</code>	Shows users in current database.
<code>use <db_name></code>	Sets current database.

Table 1: Some MongoDB Shell Commands and Their Explanations (Mongo, Inc., n.d.).

4 Connecting Java to MongoDB

Other than managing the MongoDB databases via the shell interface, developers can also use various languages to manage them. This section discusses the steps of connecting a Java program to a MongoDB database.

4.1 Creating a Java Project

Select the following options in the Eclipse IDE:

`File` \Rightarrow `New` \Rightarrow `Java Project`

and enter the project name such as `com.userid.mongo`.

4.2 Adding the Mongo Java Driver Library

Take the following steps to add the Mongo Java driver library:

- (a) Go to the Libraries tab and click the button Add Libraries.
- (b) Select the option User Library and click the button Next.
- (c) Click the button User Libraries.

- (d) Click the button New, enter a user library name such as java-driver, and click the button Add JARs....
- (e) Add the JAR such as C:\MongoDB\bin\mongo-2.10.1.jar and click the button OK:

4.3 Starting Developing the Application

Take the following steps on the Eclipse IDE. Point to the project in the left pane of Package Explorer, right click the mouse, select the options New and Class, enter the class name such as App, and click the button Finish. Complete the class such as the one shown in Program 1.

```
package com.userid.mongo;

import java.net.UnknownHostException;
import java.util.Date;
import com.mongodb.BasicDBObject;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.MongoClient;
import com.mongodb.MongoException;

/**
 * Java + MongoDB Hello World Example
 *
 */
public class App {
    public static void main( String[] args ) {
        try {
            /*** Connect to MongoDB ***/
            // Since 2.10.0, uses MongoClient
            MongoClient mongo = new MongoClient( "localhost", 27017 );

            /*** Get database ***/
            // If database doesn't exist, MongoDB will create it for you.
            DB db = mongo.getDB( "testdb" );

            /*** Get collection / table from 'testdb' ***/
            // If collection doesn't exist, MongoDB will create it for you.
            DBCollection table = db.getCollection( "user" );

            /*** Insert ***/
            // Create a document to store key and value.
            BasicDBObject document = new BasicDBObject( );
            document.put( "name", "Pokemon" );
            document.put( "age", 30 );
            document.put( "createdDate", new Date( ) );
            table.insert( document );

            /*** Find and display ***/
            BasicDBObject searchQuery = new BasicDBObject( );
            searchQuery.put( "name", "Pokemon" );

            DBCursor cursor = table.find( searchQuery );

            while ( cursor.hasNext( ) )
                System.out.println( cursor.next( ) );

            /*** Update ***/
```

```

// Search document where name="Pokemon" and update it with new values.
BasicDBObject query = new BasicDBObject( );
query.put( "name", "Pokemon");

BasicDBObject newDocument = new BasicDBObject( );
newDocument.put( "name", "Digimon" );

BasicDBObject updateObj = new BasicDBObject( );
updateObj.put( "$set", newDocument );

table.update( query, updateObj );

/**** Find and display ****/
BasicDBObject searchQuery2
    = new BasicDBObject( ).append( "name", "Digimon" );

DBCursor cursor2 = table.find( searchQuery2 );

while ( cursor2.hasNext( ) )
    System.out.println( cursor2.next( ) );

/**** Done ****/
System.out.println( "Done" );
}
catch ( UnknownHostException e ) {
    e.printStackTrace( );
}
catch ( MongoException e ) {
    e.printStackTrace( );
}
}
}

```

Program 1: src/com/wenchen/mongo/App.java

4.4 Building and Running the Project

Select the following options of the Eclipse IDE to build the project:

Project ⇒ Build Project

and then select the following options to run the project:

Run ⇒ Run

The output is then displayed in the Console tab of the Eclipse as shown in Figure 3.

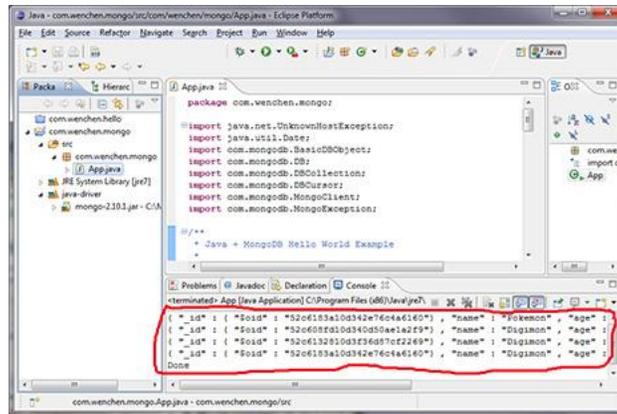


Figure 3: A Screenshot Showing the Results from Program 1.

5 Connecting ASP.NET to MongoDB

The previous section shows how to run a MongoDB database by using Java in an Eclipse environment. Nevertheless, these days many databases are connected to the Web. This section discusses how to connect ASP.NET (Microsoft, n.d.) to a MongoDB database step by step (Shovan, 2013).

5.1 Starting the Visual Studio

Select the following Windows options to start the Microsoft Visual Studio:

```

Start ⇒ All Programs ⇒ Visual Studio 2013
      ⇒ Visual Studio 2013

```

5.2 Creating a New Project

The new project is an ASP.NET Web Application, instead of Website, in C#. Add `mongoctxsharpdriver` from nuget using package manager console by selecting the following options:

```

TOOLS ⇒ NuGet Package Manager
      ⇒ Package Manager Console

```

and enter the following command:

```

PM> Install-Package mongoctxsharpdriver

```

The connection string for MongoDB server has to be defined. Because the MongoDB database server runs on the port 27017 by default, add the following code to the `web.config` file inside `<configuration>` `</configuration>`:

```

<appSettings>
  <add key="connectionString" value="Server=localhost:27017" />
</appSettings>

```

5.3 Starting the ASP.NET Programming

This simple ASP.NET application contains `create.aspx` page with a textbox, a button, and a label. The application displays the users' passwords based on the name entered from the Web. Figure 4 shows the web interfaces before and after entering a query.

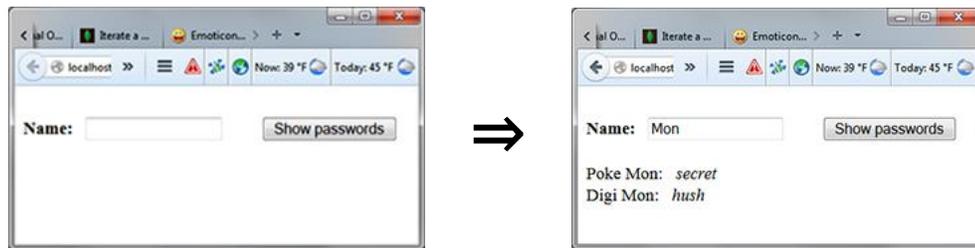


Figure 4: Two Screenshots Showing the System Interfaces Before and After Entering the Query.

The corresponding code is shown in Programs 2, 3, and 4. Program 2 is the code for web interface, Program 3 is the C# code for implementing the web interface, and two classes are defined in Program 4.

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Create.aspx.cs"
Inherits="AspWithMongo.Create" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>ASP.NET with Mongo</title>
</head>
<body>
  <form id="form1" runat="server">
    <asp:TextBox ID="name1" runat="server" Width="120px">
    </asp:TextBox>
    <asp:Button ID="showButton" runat="server" Text="Show passwords"
      OnClick="showButton_Click" />
    <asp:Label ID="pLabel" runat="server"></asp:Label>
  </form>
</body>
</html>

```

Program 2: C:\AspWithMongo\Create.aspx

```

using MongoDB;
using MongoDB.Driver;
using MongoDB.Driver.Builders;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace AspWithMongo {
    public partial class Create : System.Web.UI.Page {
        string pwords = "";
        protected void Page_Load( object sender, EventArgs e ) {
        }

        protected void showButton Click( object sender, EventArgs e ) {
            MongoServer server = MongoServer.Create(
                ConfigurationManager.AppSettings["connectionString"] );
            MongoDB myDB = server.GetDatabase( "test" );
            MongoCollection<Info> Persons = myDB.GetCollection<Info>( "persons" );
            MongoCollection<Info1> Passwords = myDB.GetCollection<Info1>( "passwords" );
            foreach ( Info Aperson in Persons.Find(
                Query.Matches( "name", name1.Text ) ) ) {
                foreach ( Info1 Apword in Passwords.Find(
                    Query.EQ( "pID", Aperson.personID ) ) )
                    pwords = pwords + " " + Aperson.name + ": " + Apword.password ;
                pLabel.Text = pwords;
            }
        }
    }
}

```

Program 3: C:\AspWithMongo\Create.aspx.cs

```

using MongoDB.Bson;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace AspWithMongo {
    public class Info {
        public ObjectId _id { get; set; }
        public string personId { get; set; }
        public string name { get; set; }
    }
    public class Info1 {
        public ObjectId _id { get; set; }
        public string pID { get; set; }
        public string password { get; set; }
    }
}

```

Program 4: C:\AspWithMongo\Info.cs

6 Summary

Though relational databases have its limitations, they have dominated the database markets for decades, during which several databases like object-oriented databases tried to replace them, but failed. However, the emergence of new applications such as big data and cloud computing makes the shortcomings of relational databases like low scalability and agility become unbearable (Khan, 2014). The creation of NoSQL databases is to mitigate the problems of relational databases and facilitate the development of contemporary applications. This paper discusses the NoSQL databases by using an empirical approach, which uses the MongoDB databases as practical examples. Readers learn how to program for the MongoDB, a NoSQL database, by studying this article. Future studies include applying NoSQL databases to the applications of big data and cloud computing.

References

- Apache Software Foundation. (n.d.a). *Cassandra*. Retrieved January 28, 2015, from <http://cassandra.apache.org/>
- Apache Software Foundation. (n.d.b). *HBase*. Retrieved February 08, 2015, from <http://hbase.apache.org/>
- Basho Technologies. (n.d.). *Riak*. Retrieved February 12, 2015, from <http://basho.com/riak/>
- Han, J., E, H., Le, G., & Du, J. (2011, October 26-28). Survey on NoSQL database. *The 6th International Conference on Pervasive Computing and Applications (ICPCA 2011)*, pp. 363-366.
- Khan, H. (2014, December). NoSQL: A database for cloud computing. *International Journal of Computer Science and Network (IJCSN)*, 3(6):498-501.
- Kobrix Software. (n.d.). *HyperGraphDB*. Retrieved February 14, 2015, from <http://hypergraphdb.org/index>
- Microsoft. (2015). *ASP.NET*. Retrieved February 13, 2015, from <http://www.asp.net>
- mongoDB, Inc. (n.d.). *mongoDB*. Retrieved March 07, 2015, from <http://www.mongodb.org/>
- Neo Technology, Inc. (n.d.). *Neo4j*. Retrieved January 22, 2015, from <http://neo4j.com/download/>
- Oracle Corporation. (n.d.). *Oracle NoSQL Database*. Retrieved from March 03, 2015, from <http://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html>
- Seguin, K. (2013). *The Little MongoDB Book*. Retrieved January 08, 2015, from <http://openmymind.net/mongodb.pdf>
- Shovan, P. (2013, October 23). *Connecting MongoDB with ASP.NET*. Retrieved March 03, 2015, from <http://www.codeproject.com/Articles/656093/Connecting-MongoDB-with-ASP-NET>
- Zaki, A. K. (2014, May). NoSQL databases: new millennium database for big data, big users, cloud computing and its security challenges. *International Journal of Research in Engineering and Technology (IJRET)*, 3(3):403-409.