

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

Volume 5 Issue 2 July - December 2015

International Manuscript ID : 22308849072015-03

EFFECTIVE APPROACHES FOR PROCESSING OF NOSQL DATABASES IN BIG DATA ENVIRONMENT

Manu Bansal

Assistant Professor

Department of IT

University Institute of Engineering & Technology

Panjab University, Chandigarh

ABSTRACT

From a last few years, the world of web technologies are associated with assorted programming languages and scripts. The domain of web development is not confined to a specific programming language or library by which the web applications and portals are under development. Now days, thousands of toolkits, programming paradigms, scripts, databases, application programming interfaces (APIs) are in use for multiple services and applications. Now the days are gone when only static website is needed using classical hypertext markup language (HTML). The corporate world is using different tools to design, develop and launch the application with maximum user interaction as well as effective graphical user interface (GUI). To implement and launch multiple types of services in the web portals, different technologies are used which are Web 1.0, Web 2.0 and Web 3.0. These evolutions or terms or simply technology transitions represents the type of web application that is being used.

Keywords – NoSQL Database, Big Data, Unstructured Data Processing

INTRODUCTION

Web 1.0 is the phrase that relates the World Wide Web with plain and read only static pages. Web 1.0 is the term that refers to an early phase of the World Wide Web's evolution. In Web 1.0 based applications, there is only restricted or limited user interaction in the websites. Web 1.0 is considered as simply an information portal in which the web users passively get information without being given the functions or modules to post feedback, reviews and comments. In case of Web 1.0, the content is fetched and displayed to the user using file system rather than database connectivity.

Web 2.0 is the most prominent and usable phrase of the World Wide Web that includes the interactive data. Unlike Web 1.0, the Web 2.0 technologies facilitate communication between users and websites, so it allows users to interact more freely with the information exchange. Web 2.0 encourages participation, collaboration, and information sharing. Web 2.0 applications include are LinkedIn, Youtube, Wiki, Flickr, Twitter, Facebook, and many others. Web 2.0 sites integrate the features by which the users interact with the web application more than just to retrieve. In this technology, the user is invited to post the comment on published write-ups, or create a user account or profile on the site, which may enable increased participation. By growing emphasis on these already-extant capabilities, they encourage the user to rely more on their browser for user interface, application software and file storage facilities. This type of computing is known as "network as platform". The key features of Web 2.0 include social media, social networking, podcasting, web content voting, curating with RSS, polling, mashups, user created web applications and blogs, self-publishing platforms, tagging as well as social bookmarking.

Web 3.0 is related with the Semantic Web that integrates the World Wide Web with interactive services, dynamic applications, and machine-to-machine interaction. Web 3.0 is simply a

semantic web technology in which the computers can interpret information like humans and intelligently generate and distribute useful content tailored to the needs of users. The example of Web 3.0 is Tivo DVR (Digital Video Recorder) in which the recording program search the web and read that it finds based on the user preferences. As another example, suppose I want to watch a good movie in the nearest theatre and willing to have some south Indian food at that time. In the classical approach, we search the movies and South Indian food separately. Then we try to relate both of these searches depending upon the minimum distance between movie theatre and South Indian restaurant. It takes lots of time. Using Web 3.0 technology, we use Web 3.0 Browser and we search a complex sentence “I want to watch a famous movie and willing to have South Indian Food”. The Web 3.0 browser will search my preferences and display the results of movie theatres with latest famous movie and nearby South Indian Restaurant.

For developing the web applications, different technologies are used depending upon the requirements and functions to be integrated. Web Development and Programming can be fragmented into number of areas that includes different types of and a typical and basic web development hierarchy might consist of:

Client Side Scripting

- Ajax
- JavaScript
- jQuery
- AngularJS
- BackboneJS
- EmberJS
- ReactJS
- Microsoft Silverlight
- HTML5

- CSS3
- Scalable Vector Graphics (SVG)
- WebGL

Server Side Scripting

- ASP.NET and ASP.NET MVC Frameworks
- ColdFusion
- CGI
- Erlang
- Groovy with Grails framework
- Java
- Lotus Domino
- Node.js
- Perl
- PHP
- Python
- Ruby
- Scala
- SSJS Server-Side JavaScript

Integration Toolkits for Client Side Scripting and Server Side Scripting

- Google Web Toolkit – To create and maintain complex JavaScript front-end applications in Java.
- Dart provides – To create and maintain complex JavaScript front-end applications as well as supporting server-side code in Dart.
- Opa – A high-level language in which both the client and the server parts are implemented.

- Pyjamas – A tool and framework for developing Ajax applications and Rich Internet Applications in Python.
- Tersus – A platform for the development of rich web applications by visually defining user interface, client side behavior and server side processing.
- Ruby On Rails
- LAMP (Linux / Apache / MySQL / PHP)
- LAPR (Linux / Apache / PostgreSQL / Ruby on Rails)

SQL, NEWSQL AND NOSQL DATABASES

Whenever there is need of Web 2.0 portal, the database oriented applications are required as the back-end database keeps and maintains the records required for the appropriate functioning of the modules. For example, the guestbook messages, posts, blogs, e-mail messages, chat messages, comments are stored in the back-end databases so that these can be retrieved, processed or edited at any instance of time.

Classically, the RDBMS packages are used the database applications which includes MySQL, Oracle, Apache Derby, IBM DB2, IBM Notes, Microsoft SQL Server, PostgreSQL, SQLite, Sybase and many others. These are known as Traditional SQL Databases which are ACID Properties Compliant.

NewSQL is a contemporary relational database management systems providing the same scalable performance of NoSQL systems for online transaction processing (OLTP) read-write workloads maintaining the ACID guarantees of a classical database system.

Now days, the web applications are using the data in heterogeneous formats that includes audio, video, text, streaming, signals, images, pixels and many others. In each pattern of file, there are

number of file formats. As in video, there are number of file formats including MPEG, MP4, AVI, 3GP, WMV, OGG, FLV and others. In the same manner, image or graphics file format includes GIF, PNG, JPEG, PCX, BMP, TIFF and lots of others.

Now the major issue is the compatibility of web application with all these file formats in different domains. At this point, the concept and implementation of NoSQL Databases comes to the scenario. In NoSQL database, any type of file format can be processed and integrated in the web applications.

NoSQL (Not Only SQL) database provides the system for storage and retrieval of data that is modeled in the methodology other than the tabular relations used in relational databases. The data structure in NoSQL databases is entirely different from the classical RDBMS. NoSQL databases are used rapidly in big data and real-time web applications.

There have been various approaches to classify NoSQL databases, each with different categories and subcategories. Because of the variety of approaches and overlaps it is difficult to get and maintain an overview of non-relational databases.

Column	Accumulo, Cassandra, Druid, HBase
Document	Clusterpoint, Apache CouchDB, Couchbase, MarkLogic, MongoDB
Key-Value	Dynamo, FoundationDB, MemcacheDB, Redis, Riak, FairCom c-treeACE, Aerospike
Graph	Allegro, Neo4J, InfiniteGraph, OrientDB, Virtuoso, Stardog

MongoDB one of the prominent cross-platform document-oriented NoSQL Database released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is free and open-source software.

As per the Media Reports, the database technology at the back-end for the implementation of Aadhaar Cards in India was MongoDB. Aadhaar implementation enrolled thousands of Indians each day and adds terabytes of data to the Data Repository. MongoDB and other data management and analytics software providers continued to produce insights that aid Aadhaar and the lives of millions of India's citizens.

MongoDB's customers include companies such as Disney, the New York Times, Cisco, MTV, Forbes, Craigslist, ADP, AstraZeneca, FourSquare, IBM, Intuit, Microsoft, McAfee, Under Armour, and eBay.

Initially developed by 10gen (now MongoDB Inc.) in October 2007 as a component of a planned platform as a service product, the organization shifted to an open source development model in 2009, with 10gen offering commercial support and other services. Since then, MongoDB has been adopted as the major technology by a number of major websites and services.

WEB APPLICATIONS DEVELOPING USING PHP AND NOSQL DATABASES

For using mongodb with php, we need to use mongodb php driver.

Download the driver from the url [Download PHP Driver](#).

Now unzip the archive and put `php_mongo.dll` in your PHP extension directory ("ext" by default) and add the following line to your `php.ini` file:

```
extension=php_mongo.dll
```

Database Connection and Select database

To make a connection, we need to specify database name, if database doesn't exist then mongodb creates it automatically.

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
?>
```

When program is executed, it will give the following result:

```
Connection to database successfully
Database mydb selected
```

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->createCollection("mycol");
echo "Collection created successfully";
```


?>

After running this program, it will give the following result:

Connection to database successfully

Database mydb selected

Collection created successfully

To insert a document into mongodb, insert() method is used.

```
<?php
```

```
    // connect to mongodb
```

```
    $m = new MongoClient();
```

```
    echo "Connection to database successfully";
```

```
    // select a database
```

```
    $db = $m->mydb;
```

```
    echo "Database mydb selected";
```

```
    $collection = $db->mycol;
```

```
    echo "Collection selected successfully";
```

```
    $document = array(
```

```
        "title" => "MongoDB",
```

```
        "description" => "database",
```

```
        "likes" => 100,
```

```
        "url" => "http://www.mynosql.com/mongodb/",
```

```
        "by", "My NoSQL Implementation"
```

```
    );
```

```
    $collection->insert($document);
```

```
    echo "Document inserted successfully";
```

```
?>
```

This program will give the following result:

Connection to database successfully

Database mydb selected

Collection selected successfully

Document inserted successfully

To select all documents from the collection, find() method is used.

```
<?php
    // connect to mongodb
    $m = new MongoClient();
    echo "Connection to database successfully";
    // select a database
    $db = $m->mydb;
    echo "Database mydb selected";
    $collection = $db->mycol;
    echo "Collection selected successfully";

    $cursor = $collection->find();
    // iterate cursor to display title of documents
    foreach ($cursor as $document) {
        echo $document["title"] . "\n";
    }
?>
```

When program is executed, it will display the following result:

Connection to database successfully

Database mydb selected

Collection selected successfully

```
{  
  "title": "MongoDB"  
}
```

To update a document , you need to use update() method.

```
<?php
```

```
  // connect to mongodb
```

```
  $m = new MongoClient();
```

```
  echo "Connection to database successfully";
```

```
  // select a database
```

```
  $db = $m->mydb;
```

```
  echo "Database mydb selected";
```

```
  $collection = $db->mycol;
```

```
  echo "Collection selected successfully";
```

```
  // now update the document
```

```
  $collection->update(array("title"=>"MongoDB"), array('$set'=>array("title"=>"MongoDB  
Implementation")));
```

```
  echo "Document updated successfully";
```

```
  // now display the updated document
```

```
  $cursor = $collection->find();
```

```
  // iterate cursor to display title of documents
```

```
  echo "Updated document";
```

```
  foreach ($cursor as $document) {
```

```
    echo $document["title"] . "\n";  
  }  
?>
```

When the program is executed, it will give the result:

Connection to database successfully

Database mydb selected

Collection selected successfully

Document updated successfully

Updated document

```
{  
  "title": "MongoDB Implementation"  
}
```

To delete a document, you need to use remove() method.

```
<?php  
  // connect to mongodb  
  $m = new MongoClient();  
  echo "Connection to database successfully";  
  // select a database  
  $db = $m->mydb;  
  echo "Database mydb selected";  
  $collection = $db->mycol;  
  echo "Collection selected successfully";  
  
  // now remove the document  
  $collection->remove(array("title"=>"MongoDB Implementation"),false);
```

```
echo "Documents deleted successfully";

// now display the available documents
$cursor = $collection->find();
// iterate cursor to display title of documents
echo "Updated document";
foreach ($cursor as $document) {
    echo $document["title"] . "\n";
}
?>
```

When program is executed, it will produce the following result:

Connection to database successfully

Database mydb selected

Collection selected successfully

Documents deleted successfully

PERFORMANCE ANALYSIS

In this section, the performance analysis of BaseSQL (MySQL) is done with NoSQL (MongoDB). The Queries are formed and processed so that the execution time can be store and compared. In the results, it is found that the NoSQL Database Query Processing is better than classical BaseSQL in terms of execution time.

Simulation Attempt	BaseSQL Database	NoSQL Database
1	0.49823	0.2312412
2	0.32523	0.24494
3	0.842949	0.6839383
4	0.4794747	0.39587558

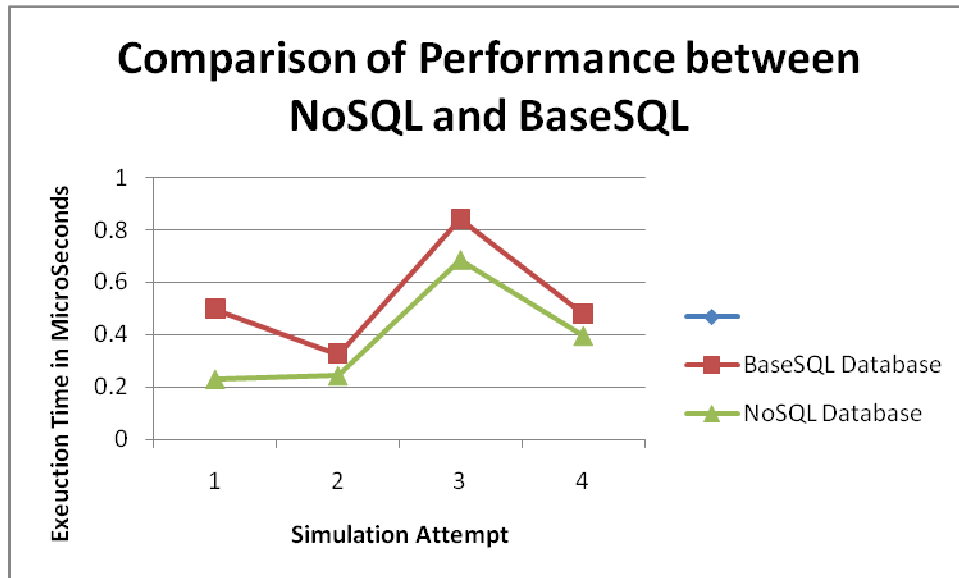


Figure 1 – Comparison of Performance of NoSQL and BaseSQL in Query Processing

CONCLUSION

NoSQL envelops a wide mixed bag of diverse database advances that were created because of an ascent in the volume of information put away about clients, articles and items, the recurrence in which this information is gotten to, and execution and handling needs. Social databases, then again, were not intended to adapt to the scale and dexterity challenges that face present day applications, nor were they constructed to exploit the shabby stockpiling and handling force accessible today.

REFERENCES

- [1] Banker, K. (2011). *MongoDB in action*. Manning Publications Co..

- [2] Wei-ping, Z., Ming-Xin, L., & Huan, C. (2011, May). Using MongoDB to implement textbook management system instead of MySQL. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on* (pp. 303-305). IEEE.
- [3] Mohan, C. (2013, March). History repeats itself: sensible and NonsenSQL aspects of the NoSQL hoopla. In *Proceedings of the 16th International Conference on Extending Database Technology* (pp. 11-16). ACM.
- [4] Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., & Abramov, J. (2011, November). Security issues in nosql databases. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on* (pp. 541-547). IEEE.
- [5] Leavitt, N. (2010). Will NoSQL databases live up to their promise?. *Computer*, 43(2), 12-14.
- [6] Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4), 12-27.
- [7] Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4), 10-11.