Mongo and Cassandra Database Project Report by Tendai Mataranyika-200935933

A) Requirements Specification

Introduction

Relational database management systems have been in use for a while now. However, new trends in databases are working towards non-SQL approaches.These include, column families, document-oriented databases, key-value tuple stores, etc. The main goal of this report to reduced the complexity of the relational model, especially for applications where other data models are more suitable. A quick overview of these trends and the frameworks that support them is discussed at http://nosql-database.org. Note that these new trends are championed by big players of the day (e.g., Google, Amazon, Twitter, Facebook, etc) In this project, we followed the same trend. We used two frameworks to support the modeling and design of the database systems for two small applications. These two frameworks are: mongodb and cassandra. mongodb (http://www.mongodb.org) is a schema-less document store, while cassandra belongs to the column family category.

The applications

For each database management framework, we focused on one particular application that needs a set of use cases.

1. Blogging application

The first application we considered in this project is a blogging application. This application manages a collection of blog entries. A blog entry has an author (the writer), a post (with a title and the content of the post), a set of tags to support faster search and comments made by readers. Each comment should indicate the commenter's name and his/her actual comment. The blog application supports the insertion and deletion of blogs. Furthermore, it supports following use cases:

_ view all the posts written by a given author. For each post, it shows the title, the content of the post and the date of its writeup.
_ view all the tags attached to a post
_ view all the comments readers submitted for a post
_ add new tags to a post
_ add new comments to a post

For this application, the database management system to used is mongodb.

2. Message management system

The second application we considered in this project is a message management system. It supports users sending messages to each other. The message management system application supports the following use cases:

_ compose messages and send them to another user
_ view incoming and outgoing messages
_ View detailed message thread

_ search, insert and delete messages

The database management system used for this application is Cassandra

3 Recommendations & Milestones
PHP was used as the server side scripting language since it integrates well with Mongo and Cassandra PHP drivers.

B) Design

A web based solution was made for both the two Cassandra and Mongo database based systems. A Linux platform was used to provide the backend for Mongo and Cassandra.

C) Implementation

Blogging application (Mongo)
Thisapplication manages a collection of blog entries. A blog entry has an author(the writer), a post (with a title and the content of the post), a set of tags tosupport faster search and comments made by readers. Each comment indicates the commenter's name and his/her actual comment. The *Insertcommentfrm .php* file is responsible for the insertion of a new comment. The *insertcommentscript.php f*ile take as input the contents of the *insertcommentfrm.php* and saves the comments into the Mongo database. The *viewcommentsfrm.php* and *viewcommentscript.php* are for viewing user comments.

The blogging application supports the insertion of blogs by utilizing the *insertblogfrm.php* file and the *insertblogscript.php* file.Furthermore, it also fulfills addition of new tags and viewing of all the tags attached to a post by utilizing the *inserttagfrm.php* and *inserttagscript.php* and *viewtagfrm.php* sand the *viewtagscript.php* files

The *connection.php* file is responsible for the establishment of a connection to the Mongo database.

Message Management System (Cassandra Database)

This application is a message management system. It supports users sending messages to each other. The messagemanagement system is responsible for compose

messages and send them to another user through the utilization of *login.php* which allows a user to login or register for an account, class.php which is the main class for the database object template, *compose.php* for creation of new messages and d*efault.php* for various actions, *outgoing.php* for the send items and *search.php* for searching of emails. Thrift files are also important since thrift is the PHP driver for Cassandra.

D) Testing

The two systems where tested via the web interface as can be evidenced by the Cassandra application which is still accessible via mail.namibiaadverts.com/cassandra