Data 650 Summer 2019

Assignment 1

HBase

James Nicholson

Email: Nicholsonjb3@gmail.com

University of Maryland University College

# Part 1 Questions And Answers

1. Discuss the differences between HBase and relational database tables. When would you use each?
   1. There are several differences between a relational base table and HBase tables. Relational base tables are row-based, while HBase tables are column based. Relation base tables (RBT) require SQL (structured query language) verses HBase that uses NO SQL. RBT tables are based on a fixed schema and are static, while HBase tables are flexible and dynamic. RBT follows the ACID (Atomicity, Consistency, Isolation, and Durability) property, and HBase uses CAP (Consistency, Availability, Partition-tolerance) theorem. RBT is very good at handling structured data that is is not sparse while HBase can process structured, unstructured as well as semi-structure data as well as sparse data. RBT is not scalable while HBase can scale very quickly. RBT is used whenever a user wants to manage data, its associations, and when the user cares about data governance and security at a massive scale. HBase is made for large amounts of structured data, and unstructured data is such as email or poorly designed log files, twitter feeds and web pages. Also, when the data size or type is such that one is unable to save it in an RBT, then one should select HBase as an option. A great example of this would be a product catalog.
2. Discuss the differences between HBase and Hive. When would you use each?
   1. HBase and Hive are two Hadoop big data technologies that serve different functions. Hive is a SQL engine for Hadoop that is designed for SQL savvy people to run MapReduce jobs through SQL Style queries. Hive allows for developers to create logical relational schemas on various file formats and physical storage, which line in or outside of the Hadoop cluster. Hive is best used when to map HDFS files to Hive tables and query the data. Hive should also be used for data warehousing requirements and when developers do not want to write complex MapReduce code. Hive is a query engine that, while HBase, is data storage predominantly for unstructured data. Hive is mostly used for batch, for example, Online analytical processing (OLAP), but HBase is widely used for transactional processing where the response time of the query is not highly interactive, as seen with Online Transactional Processing (OLTP). Contrastingly Hive, processes in HBase are run in real-time on the database instead of transforming into MapReduce jobs. HBase is to real-time querying, and Hive is to analytical queries.
3. What are the differences between the delete and drop HBase Shell commands?
   1. The drop command is used to delete a table in Hbase, but the table must be disabled first, while the Delete will remove values from a row or column.
4. When would you use the deleteall HBase Shell command?
   1. Deleteall will remove all cells in a given row or column.
5. When would you use the drop\_all HBase Shell command?
   1. Drop\_all will remove or delete all tables with a matching regex expression in HBase.
6. What is data versioning in HBase, and how is it implemented? What are the pros and cons of versioning?
   1. In HBase, cells are a combination of the row, column family, and version contains a value of a timestamp that displays the column family version. A version is a timestamp value that is placed next to each row value. Since HBase also uses HDFS, it is not easy to update data. To enable that feature, HBase creates a version of the cells being updated. Maintain versioning allows for how many records to be stored in an HBase table when a new record is updated. Having too many records may cause difficulty when attempting to retrieve data from the HBase table.
7. What is the row key in HBase?
   1. A row key is an indexed value in an HBase table that is sorted lexicographically by this row key
8. Define the column family. Is it possible to create an HBase table without any column families? Why or why not?
   1. A column family is a grouping of similar data in Hadoop. It is identical to a table in a Relational Database Management System. When creating a table in HBase, it is required to define or more column families. Column families will remain fixed throughout the lifetime of the HBase and should be strongly considered when creating a table; however, new column families can be added in the future if needed.

# Part 2: Retail Bookstore Case Study

# Introduction

A retail store requires a new updated inventory management system. The store's massive inventory is currently stored on various physical storage devices ranging from CD to high volume hard disks. Improvements to productivity for the store, a Big Data solution is needed to help update the current database system as well as handle future inventory management. This use case study will examine a possible Big Data System solution using Apache Hadoop as well as provide sample code on how to manage the new system using the HBase Shell.

# Problem Statement

**Problem** **Statement**: Intensive manual search process and inventory management due to physical storage devices.

**Description**: Inventory for a retail book store contain Ascii-based text files stored on traditional physical storage devices ranging from CD’s, DVDs, Optical disk, and high-volume hard disks.

**Risk**: The physical storage of inventory creates a highly manual process for inventory management and searchability and promotes the risk of inadequate organizational response to business requests and lost time that should be spent carrying out core business activities.

# Design

The idea behind the database design is to create a searchable database that can store existing Ascii-based text files as well as additional text files in the future. The goals of the database are straightforward:

1. Improved inventory management within the store
2. Reduce the total time spent searching for books
3. Increase revenue per visit and transaction.

In terms of technologies, customers and employees will need the following:

1. Computers with inventory information tied to an author
2. Data collection and decision points for real-time interactions and analytics
3. Storage and processing services for batch-oriented analytics.

In terms of data sets, the store will have at least the following rules to consider

1. A topic may have many sub-topics as well as subtopic assignment may change with Reprint editions.
2. If a book belongs to a series, the series name and number are required.
3. A book can have multiple authors
4. For some authors, their affiliation information may be available, and the author may have numerous affiliations.
5. If a book has multiple reprinted editions, the book store might only carry a few of them at a given time.
6. Books may also have multiple IBSN numbers.

For the system, the ACID (Atomicity, Consistency, Isolation, and Durability) versus the CAP (Consistency, Availability, Partition-tolerance) theorem. Since transactions in the system need to be the same before and after a transaction is completed, ACID would be best suited for the network while the CAP theorem does allow for the availability of the system in that every request will return a response. The CAP would not be optimal if the method returned incorrect information. HBase does follow the ACID properties in which is the main reason for this decision.

# Implementation Methods

Implementation of the intimal database will use HBase Shell to create and manage the database. However, before HBase tables are used to create the database, several steps should be taken beforehand, the first step is the identification of the topic of the Author. For this use case, is that the user will use a computer to look up the authors’ information. When this occurs, the database will trigger looks in an author profile in the database. The goal of the database is to use an indexed structure to do fast and efficient lookups of information. Once the author’s info is located, the data is sent back to the user in the system. The models created in the system will use a custom-built or Commercial-off-the-shelf (COTS) software to evaluate requests and the author’s profile to ensure the item is in stock in real-time. As mentioned before, to build an accurate model, a batch-oriented processing farm will be used. This database will be leveraged using Apache Hadoop and the Apache Hadoop Distributed File System (HDFS) to create a model of inventory management. The system will contain a database (or Data warehouse) that will hold all of the previously existing text files and any new incoming text files. After data linking and integration are completed, the business will be able to handle the hundreds of millions of records that exist now and in the future.

After implementation is completed, the data will flow through the ecosystem via the following sections.

Step 1: Collect Data

To look up data, gather it, and make decisions on the data. A system will be implemented that is distributed. Since the business receives new books every day, the data will need to be loaded without any delays. The data flows into the Hadoop cluster and loaded using Flume or Scribe, which is a fault-tolerant ingest system for the Apache Hadoop ecosystem.

Step 2: Organize and Move the Data

The next step is to add data (Topics, Subtopics, Title, Series, Series Title, etc.) and start collating, interpreting, and understanding the data. This step is completed using Apache Hadoop MapReduce. Once the data is moved, and author profiles are created from the NoSQL database through the Hadoop InputFormat interface and then added to the MapReduce data sets. This data will be combined with the inventory management system to be used by the business.

Step 3: Analyze the Data

The last step is to analyze the data to ensure inventory levels are correct for users. This information and also be used to create profiles of which books are purchased more frequently or which an author will most often write styles of books.

# Pseudocode

Now that the design and implantation have been discussed. Several examples of pseudocode are provided below to show how the database is created and using the HBase Shell if the proposed solutions are unattainable.

Using the HBase shell to create the schema of the table, the create() command is used, as seen in Table 1.

create ‘books’,’ Topic’, ‘Sub-topic’, ‘Title’, ‘Series’,’ Series Title’, ‘Series Number’, ‘Author’, ‘Author Affiliation’, ‘Publisher’, ‘Reprint edition’, ‘Language’, ‘Paperback’, ‘ISBN’



Verification of the table being created in HBase or to check it the books table is done using the list command. The user can also see information about the recently created table using the describe command.

describe ‘books.’

The describe() command displays information about column families, associated filters, and the version of the table.

Once the table is created, the records provided as examples from the business are loaded into the table using the put() command.

Record 1:

put ‘books’, ‘00001’,’ Topic’, ‘Fiction’

put ‘books’, ‘00001’,’Sub-topic’, ‘Young-adult’

put ‘books’, ‘00001’,’Sub-topic’, ‘Mystery’

put ‘books’, ‘00001’,’Sub-topic’, ‘Adventure’,

put ‘books’, ‘00001’,’Title’,’ A study in Charlotte’

put ‘books’, ‘00001’,’Series’,’Yes’

put ‘books’, ‘00001’,’Series Title’, ‘Charlotte Holmes Series’

put ‘books’, ‘00001’,’Series Number’,’1’

put ‘books’, ‘00001’,’Author’, ‘Author 1: Brittany Cavallaro’,’Author’, ‘Author 2:’,’Author’,’Author 4:’

put ‘books’, ‘00001’,’Publisher’,’ Katherine Tegen Books’

put ‘books’, ‘00001’,’Reprint edition’, ‘January 3, 2017’

put ‘books’, ‘00001’,’Language’,’ English’

put ‘books’, ‘00001’,’Paperback’, ‘352 pages’,

put ‘books’, ‘00001’,’ISBN’, ‘ISBN-10: 0062398911’,’ISBN’, ‘ISBN-13: 978-0062398918’

put ‘books’, ‘00001’,’Product Dimensions’,’5.3 x 0.8 x 8 inches’

Record 2:

put ‘books’, ‘00002’,’ Topic’, ‘Non-Fiction.’

put ‘books’, ‘00002’,’Sub-topic’, ‘Adult’

put ‘books’, ‘00002’,’Sub-topic’, ‘Young-Adult’

put ‘books’, ‘00002’,’Sub-topic’, ‘Self Help’

put ‘books’, ‘00002’,’Sub-topic’, ‘Professional training’

put ‘books’, ‘00002’,’Sub-topic’, ‘Comedy’

put ‘books’, ‘00002’,’Title’,’ How to be a comedian’

put ‘books’, ‘00002’,’Series’ ‘no’

put ‘books’, ‘00002’,’Author’, ‘Author 1: George Carlin’, ‘Author 2:David Letterman’, ‘Author 3: Jay Leno’, ‘Author 4: Conan O’Brian’, ‘Author 5: Jimmy Kimmel’, ‘Author 5: George Lopez’, ‘Author 6: Eddie Murphy’, ‘Author 7: Ellen Degeneres’, ‘Author 8: Richard Pryor’, ‘Author 9: Margaret Cho’

put ‘books’, ‘00002’,’Author Affiliation,’ ‘Author 2, 12 Second Street, NY 10005’, ‘Author 5, 91, Hollywood Street, Los Angeles, CA 90100’

put ‘books’, ‘00002’,’ Publisher’, ‘McGraw Hill Academic Press’

put ‘books’, ‘00002’,’ Reprint edition’,’8’

put ‘books’, ‘00002’,’Language’,’ English’

put ‘books’, ‘00002’,’Paperback’, ‘666 pages’

put ‘books’, ‘00002’,’ISBN’,’ISBN-10: 0087738911’, ‘00002’,’ISBN’,’ISBN-13: 942-006711918’

The following commands are used to verify information about the newly created table and two records have been added and explain their functions.

* exists ‘books.’
  + checks to see of books table exists
* count ‘books’
  + count the number of records in the table
* get ‘books’,’00001’
  + read record 00001 from books table
* scan ‘books’
  + scans data in HTable books and returns table data.

If the user wanted to delete a record from the books table, they would use the command

deleteall ‘books,’ ‘00001’.

The book’s table has been created but if the business wanted to update the table. The alter command is used. Below is an example of using the alter command:

Alter ‘books’, {Name => ‘issue date’}

The above command will add a new column family to the Hbase table books. If a column needs to be removed the alter command is used as seen below:

Alter ‘books’, {Name => ‘issue date’, Method => ‘delete’}

The alter command can also be used to update the maximum number of cells of a column family as seen below:

alter ‘books’, NAME ⇒ ‘ author’, VERSIONS ⇒ 5

There are several different commands to consider as well for security purposes. There are three commands used for this function: grant, revoke, and user\_permission.

The grant command gives specific user rights such as read, write, execute, and admin on a table for a user. The example below grants all privileges to the user ‘DATA650’

Grant ‘DATA650’, ‘RWXCA’ [‘books’]

The following code revokes all permission from the use named ‘John Smith’

Revoke ‘DATA650’

To see which permission a user has the for a table the user\_permission command is used as seen below:

User\_permission ‘DATA650’

The mention above code can be used to set up an HBase Table for the retail store. An automated solution for populating tables would be best suited for future implementations.

# Conclusions and takeaways

Big data allows users to leverage incredible amounts of data and processing resources to arrive at precise models. The Apache Hadoop is designed to handle structured, unstructured, and semi-structured data the business will receive. The retail store will need to process hundreds of millions of books in multiple languages, which Hadoop will help with their Big Data need. It also allows the business to determine different needs that were not expected, which will create more accurate models and also allow for new ideas and new business opportunities in the future. A suggestion for improving the Hadoop clusters in the future would be to add Apache Hive, which is a data warehouse tool built on top of Hadoop for data analysis, summarization, and querying. Hive uses a SQL-like interface to query data sored in various data structures and file systems.