**Elizabeth Sayler, Sukhmani Thukral, Sean Bruce**

**Quiz - in class**

**ISIT 420 Spring 2020**

Compare and contrast the advantages and disadvantages of the four storage and   
query systems for solving the following problems:

**[A] Storing real time events streaming in at very high rates (e.g. all the point of sale data from 100’s of Target department stores, twitter replies and forwards from a Taylor Swift tweet) and posting real time statistics gathered from that data.**

Online transaction processing can be done for various kinds of data. However, it becomes complex when we need to process real time data. Hadoop offers features that can be used to process data in batches. It takes care of the 3 Vs, that is, volume, velocity, and variety. Velocity refers to real-time or near real-time data. Hadoop Distributed File System is designed to use data sets. The architecture includes the Name Node and a Data Node. The name node has information regarding the file-details, whereas, the data nodes store the data.

Hadoop is required when data volume is high, else, we can use SQL. SQL stores data in a logical form. SQL (Structured Query Language) is a relational database management system. Furthermore, while Hadoop can work with any datatype, SQL is designed to work with structured data. In other words, it can only be used to query a relational database management system. Additionally, Hadoop focuses on Batch processing and SQL supports real time data processing.

When comparing Hadoop to Mongo DB, we realize that MongoDB cannot manage unclustered data, just like SQL. MongoDB is a NoSQL database. However, MongoDB can also be used for real-time data processing but for smaller subsets of data. The main components of MongoDB include: mongod, mongos, mongo. Mongod is the core database. Mongos is the controller, and mongo is the interactive shell.

While, Hadoop, Mongo and SQL can all be used for real-time data processing. The processing time in MongoDB is measured in milliseconds, while for hadoop it will be minutes and hours, and for SQL it will be seconds.

MDX (Multi dimensional Expression) helps refer multiple dimensions unlike SQL, and Hadoop can be used with MDX to gather data.

Example: to get data from Twitter we would use Hadoop. We will need to generate the keys, like, Consumer, ConsumerSecret, Access Token, and Access Token Secret. We will also need the necessary Jar files: core and stream. Thereafter, we will add these files to our project, and run the code.

If we are looking at data from target we would want it to be atomic and consistent and hence we might want to use SQL.

**[B] Guaranteeing the accuracy and completeness of 1000’s of transactions (bank ATM events, stock trades)**

I would choose a RDBMS solution for this problem. The problems that come along with RDBMS solutions include cost, management large volumes of data and limits on field lengths. While these sound like limiting issues for this application, the security risks involved with financial transactions should be paramount.

For that reason, the user access options and privileges available in RDBMS make it the clear winner. It has the speed needed for heavy traffic and provided routes for secure access to the data. Data Cubes are too slow and both mongoDB and Hadoop simply do not offer the security layer necessary for this type of application.

**[C] Analyzing the last weeks sales from 100’s of Target department stores and deciding which items should have their prices raised or lowered.**

In this case the data is already present but needs to be analyzed. Relatively speaking time is also less of a factor for processing the data. However, there is bound to be quite a lot of data that needs aggregating and analyzing. Not to mention that the question being asked will require multiple complicated queries.

Data Cubes excels in business intelligence applications. With them, the user can analyze the aggregated data and produce detailed records to help answer complicated questions. In short, Data Cubes can be used to take large amounts of data and easily drill down to the specific info needed to answer such questions.

While RDBMS would do this, the tools sets for analytics are simply not as robust as those in Data Cubes. And neither mongoDB or hadoop have analytic tools geared towards large stores of data.

**[D] Allowing an Amazon customer to look up the details and status of any currently outstanding orders they have.**

Amazon is a massive company with records likely up into the millions and thus the biggest concern would be storage space. However making sure the data is secure and redundant in case of failure points is not something to ignore either.

Data cubes take a more analytical approach and don’t update often enough to be useful for status updates, Hadoop has a function where you can set up redundancies but it’s not baked into the system and thus is also a less than ideal choice.

The data is relational in nature so i would suggest using a system with that already in it; like an RDBMS.

**[E] Storing the responses to an online survey posted by a major league soccer team with 25 questions that the form does NOT require the user to answer every question and with the expectation that at least 500,000 fans will fill in the survey the day it is posted.**

Storing the responses doesn't require the updates that RDBMS can provide (and while technically it could be stored in a relational database, the data isn’t particularly relational dependent). Data Cubes are good for any future processing and analyzing that might need to be done, but to just store the data it is extremely overkill.

MongoDB is flexible in how it stores data - perfect for unanswered questions, but to store it in Hadoop you can easily put them in as a null. Both systems can handle the amount of data easily. Personally i would suggest using mongo as it is a more self-contained system. (and a little more user friendly to beginners, compared to Hadoop)