Prototype Big Data Archive in a Public Cloud

Group 56: Pathfinder of Big Data

Zhi Jiang, Isaac T Chan, Zhaoheng Wang

CS 461: Senior Capstone Fall 2016

Oregon State University

Abstract

OSU campuses generate data constantly from multiples sources, including computer labs, wireless usage, student devices, and many others. This quantity of data, also known as big data, can effectively represent all kinds of behaviors of students for information technology. Currently, the data is very difficult to manage because it is collected from multiple sources and is impossible to analyze. Therefore, this project requires the use of various technologies for support. There are nine pieces of technologies required:1) Methods to measure performance metrics of database functionality. 2) Methods of database security. 3) Methods of user interaction with the system. 4) The framework and storage of processing unprocessed data 5) The ingestion and parsing for unprocessed data. 6) The operation for formatted and cleaned data in data storage 7) The storage way for dealing with processed data. 8) The programming language for achieving database functionality. 9) The visualization tool use to display the data. For each piece, we will provide the best three technology options. Our goal for this paper is to analyze each technology option and determine the optimal technology that we will implement.

Contents

1 Introduction

The purpose of this document is identify the best way to solve problem we stated before. So in this document, we separate the whole product into nine distinct pieces and each member will discuss three pieces individually. Isaac Chan is in charge of these pieces: methods to measure performance metrics of database functionality, methods of database security, and methods of user interaction with the system. Zhi Jiang is in charge of these three pieces: the framework and storage of processing unprocessed data, the ingestion and parsing for unprocessed data and the operation for formatted, and cleaned data in data storage. Zhaoheng Wang is in charge of the storage way for dealing with processed data, programming language for achieving database functionality, and the visualization tool to display the data.

2 Methods to measure performance metrics of database functionality

The following are the three best options for benchmarking and measuring performance of our implemented NoSQL database: YCSB, AWS Cloudwatch, and TPC-H. YCSB, or Yahoo! Cloud Serving Benchmark, is an open source framework for evaluating and comparing the performance of multiple types of data-serving systems[1]. AWS Cloudwatch is a built-in utility of AWS and can be used to collect and track metrics. TPC-H benchmark consists of a suite of business oriented ad-hoc queries and concurrent data modifications[2]. With the benchmarking and performance measurement utility, we hope to obtain a baseline for our database performance and examine how various data and query loads compare to the baseline. We will be evaluating operation speed of operations such as database inserts, updates, and reads.

	Inserts	**Updates**	**Reads**	**Visualization**	**Extra Notes**
YCSB	Yes	Yes	Yes	Raw data, can be plotted	Open-source utility means we can customize tests to fit our use-case
AWS Cloudwatch	Yes	Yes	Yes	On AWS UI and also provides raw data	Built-in utility eliminates complexity of implementation
TPC-H	Yes	Yes	Yes	Raw data	Lack of customizable tests

YCSB is a very customizable, open-source utility that can produce relevant and informational metrics for our database. It has a wide user-base and should be easy to implement.

AWS Cloudwatch is a built-in tool that can deliver relevant metrics and should work well with our database on AWS. It also has customizable metrics, which we would implement using AWS CLI (command line interface).

Finally, TPC-H is an enterprise-grade option, mostly used for server-production companies to measure how their products compare to alternatives. There is a lack of customization and a lack of a community for troubleshooting. Documentation is minimal. Implementing TPC-H is likely to be a challenge.

We will select AWS Cloudwatch as the best option. Cloudwatch can also use customized metrics tests which is important in order to know metrics to fit our use case. Although YCSB may be more easily customizable, the lack of installation makes Cloudwatch the best option.

3 Methods of database security

Traditionally, NoSQL databases have minimal security. In most implementations, the only security is to allow access to the database via trusted machines. However, relying solely on the network is almost certainly an invitation for a breach to sensitive information. NoSQL databases also cannot use external encryption tools, such as LDAP or Kerberos. Our best options for database security is AWSs user authentication policies, encrypting sensitive data fields, and using sufficient input validation to avoid injection attacks.With methods of database security, we hope to improve security of our database by restricting access and preventing malicious utilization of the data. We will evaluate database security by ensuring only approved user access, whether or not sensitive data is encrypted, and resistance to injection attacks.

	User restricted access	**Sensitive data encryption**	**Resistance to injection attacks**
AWS user authentication	Yes	No	Yes
Sensitive data encryption	No	Yes	No
Input validation	No	No	Yes

AWS offers a useful utility for user authentication, where users of the system can be granted different levels of access. Naturally, with authentication there is a built-in resistance to injection attacks because hopefully authenticated users are less prone to malicious intent.

There will be sensitive data in our database, most notably user-identification information, such as student IDs. This will be encrypted prior to given to us, and will most likely remain encrypted as the database is implemented and real data is inserted.

Finally, input validation is a minimal concern if we decide to implement NoSQL using AWS DynamoDB. DynamoDB does not support multiple actions with a single command, removing the risk of injection attacks. If we choose to implement the database using another tool, then there will need to be test cases to identify and ignore injection attacks.

As shown in the comparison table above, no one method can cover the security of the table. We will need to implement all three methods. Due to the lack of external tool support by NoSQL databases, we must resort to using modular security methods.

4 Methods of user interaction with the system

There will only be one type of user that interacts with the system - OSU staff that are able to manage and analyze the data. There are many analysis tools that can be used in conjunction with AWS. Most notably, Amazon Machine Learning (AML), Amazon EMR, and Amazon QuickSight. AML is a service that provides easy-to-use machine learning technology. Amazon EMR is a comprehensive utility that allows users to interact with databases, data warehouses, and customize their analysis. Amazon QuickSight is a fast business intelligence service that allows users to visualize data and provide responsive analysis. Our goal is to shape our implementation database to work with a utility that allows users to interact with our system. The ideal utility would provide tools to manage the data, provide analysis, machine learning support, and visualization.

	Data Management	Analysis	Machine Learning	Visualization
AML	No	No	Yes	No
EMR	Yes	Yes	Yes	No
QuickSight	No	Yes	No	Yes

AML is a very specific utility that only really offers machine learning analysis in a simple interface. It does contain much else within the tool, but the user can visualize the results from machine learning models with Amazon Cloudwatch.

Amazon EMR is a comprehensive tool that can manage data and provide analysis, through conventional queries or machine learning technology. It is more difficult to use, and does not provide native visualization.

Lastly, Amazon QuickSight is a business intelligence tool that can deliver fast analysis and boasts a very attractive visualization tool. However, more in-depth analysis and methods of data management are unavailable.

Amazon EMR is the technology that we will primarily be considering. After the conclusion of our project, maintaining the database, as well as analysis, is critical to the survival of our prototype. Our chosen technology must support all of these, and Amazon EMR is the most comprehensive tool. However, in the end, all of these technologies can be used in conjunction with our final big data prototype, but it is important to have a primary tool in order to ensure database maintenance and analysis is continuable and extensible after the Expo.

5 The framework and storage of processing unprocessed data

In the entire large system, the client will collect data for us firstly, afterward we should do some analyzing and parsing for these unprocessed data. Meanwhile, we need to provide enough space to store these unprocessed data. In this step, the primary we must consider is to find a proper framework to build storage and then perform more operation like parsing data, so we will discuss framework and storage in this section.

According to clients requirement, we should use Amazon Web Service to complete these tasks. Although we have talked about advantages of EMR above, we also would like to choose Amazon EMR (Elastic MapReduce) in this part and we focus on Hadoop framework of Amazon ERM. Specifically, Hadoop is software framework that perform distributed processing a large amount of data across hundreds of inexpensive servers[3]. The framework contains two kinds of tool. One is storage, and which is used store unprocessed data due to data cannot be stored in database directly. Another tool is used parse data. The advantage of Hadoop is obvious, because Hadoop can provide a high level of durability and availability while still being able to process computational analytical workloads in parallel. The combination of availability, durability, and scalability of processing makes Hadoop a natural fit for big data workloads[4]. In Hadoop, there are many kinds of tools, so according to our researches, we find two effective tools which are used to store unprocessed data as a storage, and they can also interact with Hadoop framework.

First of all, the Amazon Simple Storage Service (Amazon S3) is one of best choices for us. Our main part of product database is built on AWS cloud platform, thus one advantage of S3 is that it has high interactivity with our database.

In the other words, it is effortless to build connection and transform data among them. In addition, according to clients description, our product should be able deal with many kinds of data such as log file and stream, hence S3 is scalable and it can satisfy as much needs as our data. The cost of entire product is also an important criterion we need to consider, because client wants to compare cost between cloud product and local hardware, so one crucial benefit of choosing this service is its cost is low.

Second technology about storage we find is Hadoop Distributed File System (HDFS). Compare with S3, the scalability of HDFS is not better than the former. The main difference is that HDFS depends on local storage, so it has to add larger hard drives or more machines to the cluster when it needs to expand storage space for more data[5]. Meanwhile this weakness will cause that cost of it will be increased obviously. As for size limitation, any size of files can be allowed to store in HDFS, but the maximum size of single data element is only up to 5GB. We have not yet known all information about sample data client will provide, thus it is unclear whether this limitation of size on S3 will affect our product. Overall, S3 is better than HDFS because it can maximally decrease cost and ensure effective connection with database.

6 The ingestion and parsing for unprocessed data

Our product must be able to ingest and parse these unprocessed data such as format conversion, thus we need to find proper tool base on Hadoop framework for corresponding type of data. In section, we will compare two tools for parsing log file firstly, and then we will discuss a tool deal with stream data.

Dealing with log files is indispensable to our product, so Apache Spark is a remarkable tool for parsing log file. Apache Spark, as source processing engine for a large-scale data, can be easily used to parse log files. Specifically, it supports multiple programming language to write application of data analyze such as Java, Scala and R, so which means we have many choices to develop application quickly. On the other hand, the processing speed is very important to Big Data, so Apache Spark still has high speed of processing data for this aspect. According to 6 Sparkling Features of Apache Spark written by Lijin Joseji, Spark enables applications in Hadoop clusters to run up to 100x faster in memory, and 10x faster even when running on disk. Spark makes it possible by reducing number of read/write to disc[6].

MapReduce is one component of Hadoop and it is also used to process and generate large data sets. The obvious restriction of MapReduce is that it only provides two kinds of operations: Map and Reduce. Because the core concept of MapReduce processing a data is that it will separate the data into a series key/value pairs by Map function firstly, and then using Reduce function to sort each key/value. But in fact, many calculating for data cannot fit this kind of operation model. On contrary, Apache Spark can provide more operations to deal with data. As for programming language, MapReduce only supports Java, so these attributes make writing program more complicated for developers. On the other hand, all of data need to be store disk while MapReduce is processing them and Apache Spark process data in memory. Although Apache Spark is faster than MapReduce, it also means Apache Spark needs a lot of memory[7]. If size of data we want to process is not large, probably we do not need to provide more memory for Apache Spark. Overall, Apache Spark has more advantages than MapReduce on many aspects like speed of processing data and methods of operating data, so we would like to choose Apache Spark to parse log files.

The stream is another important type of data we will process, so we would like to choose an effective tool to analysis streaming data specially. Amazon Kinesis is also provided by AWS, therefore it has can commendably interact with other AWS products we choose such as data storage S3. Amazon Kinesis Streams, as one function of Amazon Kinesis, can support developers to build custom applications that process or analyze streaming data for specialized needs[8]. Another advantage is Amazon Kinesis Streams supports real-time data processing. Actually we are not sure this benefit is necessary for us because client will provide sample data rather than real-time data, but we will meet with client and determine whether the client needs function later. Amazon Kinesis API can be used in Amazon Web Services SDKs, and Amazon Web Services SDKs contains multiple programming language such as Java and PHP, so developers could use familiar programming language to complete tasks in this part.

7 The operation for formatted and cleaned data in data storage

After data are formatted and cleaned by corresponding tools, we should be able to do some operations for these data. For example, we need to transfer these data from data storage to database. In section, we will discuss differences and functions of three tools operating data.

The main purpose of Hive is control data in storage on Hadoop framework such as HDFS or S3. After data is formatted and cleaned by corresponding tool, they will be stored in data storage again, and then next step is move these data from data storage to database. In general, the purpose of Amazon ERM Hive is to make connection between storage and database. The developer can write appropriate Hive command or Hive script to operate data in storage. For instance, developer can use Hive to make table for data on storage after a log is pursed, and then this table can be imported to external database. On the contrary, Hive can also perform the same operation for data from external database to data storage, so the interoperability of data is improved between these tools. In addition, it is worth nothing that Hive scripts use an SQL-like language called Hive QL[9], so it can help developers who are familiar with SQL to complete corresponding tasks quickly.

Impala, as real-time interactive SQL query tool, has the similar functions with Hive in Amazon ERM. There is difference between methods to execute SQL queries for them. The way of Impala executing SQL queries is using a massively parallel processing (MPP) engine. On the other hand, Hive executes SQL queries using MapReduce. Hence Impala does need to create MapReduce jobs and then it will spend faster query times than Hive[10]. The advantage of Impala can help developer to implement quickly some ideas about operation of data, so we will consider use these two tools together in this part.

Amazon EMR also supports Apache Pig, and Apache Pig is used to operate data on top of Hadoop as well. Firstly, Pig is different from SQL, so the developer need to speed more time learning it. Secondly, Apache Pig, as a dataflow language, can control and optimize each step while it processing data. In fact, Apache Pig is unfitting for this product, because the purpose of this part is to operate data between data storage and database, nevertheless the Apache Pig is unable to interact with external database. We will consider use advantages of Apache Pig and Hive together, for instance, Apache Pig processes data and then Hive transfers data between data storage and database. Overall, these three tools have the respective advantages, so the best way is to combine advantage of each to operate data in this part.

8 THE STORAGE WAY FOR DEALING WITH PROCESSED DATA

The goal for this part is to figure which is the optimal option of storage way for dealing with processed data.our product requires to use the NoSQL database however it may not be the optimal option for storage. As a result, we will first compare other storage way with database to check whether database is the optimal choice. If the database is the optimal choice, we will compare the Sql with NoSQL databases. After that, it is necessary to figure out which type of SQL or NoSQL is the best option.

There are many ways for storing data such as Database, Files, Cookies and other ways. we would like to start with cookie storage. Usually, cookies are used to store tiny bits of data. These data are very small only below 4 kilobytes per domain[11]. Besides. the cookies are pass the data by request which is not suitable for our product. Therefore, the cookie storing is not a good choice for our product. Another option is using Files such as XML to store the data. The data will be very large quality in our product thus there will be a lots of XML files for storing. Because of this, the management will be complex by using XML files. So the file storing is also not the suitable option for storage. The next option is using database. The database is used to store the data. The data could be managing, retrieving and organizing in the database[12].The database gives promote accessibility for the data and the data could be easily accessed by using query. Besides, it makes the data more security and reducing the cost for data insert, storing[13]. As a result, the database is the optimal choice for our product.

The database could divide into two types in general: the Relationship database and Non-relational database.The Relational database is usually represent the data base on the table however the Non-relational database use dynamic schema for data. When dealing with multiple type of data, the Relationship database is not the optimal choice because it is not the optimal choice if the data is storing in hierarchical. However, the Non-relational database is the optimal choice for dealing large quantity of data which stores in hierarchical. For scalability, the Relationship database could increase its scalability by promoting the power of hardware however the Non- relational database will increase its scalability by reducing the load. After comparing with the Relationship database and Non-relational database, we find that the Non-relational database Is the optimal choice for our product because the Non-relational database is suitable for dealing with large quantity of data which stores in hierarchical[14].

The Non-relational database we want to evaluating are: SimpleDB, MongoDB, DynamoDB, Cloud Datastore. we generate several criteria for evaluating these NoSQL databases. These criteria will evaluating the NoSQL database in various features such as cost,platform support, security,loading speed and many other features.

Tool	Platform	Security for the data	Speed for loading data	Features	cost for the tool
SimpleDB	Amazon web services	Yes	Fast	High availability and simple to use	Low

MongoDB	Cross-platform	Yes	Fast	Document database, high performance and high availability	Low
DynamoDB	Amazon web services	Yes	Fast	Support key-value model,High availability, free-text search,flexible database schema	25GB for free and the cost is low
Cloud Datastore	Google cloud platform	Does not mention on the product page	Fast	High availability and high scalability	Low

SimpleDB is offered by Amazon web services. It is security for the data and the speed for loading is fast. Besides, the cost for it is low.

MongoDB is supported by cross-platform. It is also security for the data and the speed for loading is fast. It has the high availability and high performance. The cost for it is also low.

DynamoDB is supported by Amazon web services. It is more security for the data than others and the speed for loading is fast. It has many features for example, the free-text search will make the information searching more convenient[15]. Besides, it has high availability and flexible database schema.

Cloud Datastore is supported by Google cloud platform.it is fast and it has high availability and high scalability.

We will select the DynamoDB for storing because our client requires to use the Amazon web services as the platform. Besides, the DynamoDB has more features which is suitable for our product. For example, the free-text search allows to search the information easier and flexible schema make the schema easy to development. Furthermore, the cost for DynamoDB is very low.

9 Programming language for achieving Database functionality

There are many options for programming language such as java, python, php. Each programming language will have different features. Our goal for this part is to figure out the suitable language for our product and avoid using many different languages. Because the more languages we use the more mistake we will might have.we generate several criteria for evaluating different language such as APIs, testability,security and tool support .

language	API for inserting	API for updating	API for listing table	Testability	Security	DynamoDB support
Java	Yes	Yes	Yes	Testable	Secure	Yes
php	Yes	Yes	Yes	Testable	Normal	Yes
python	No	No	No	Testable	Secure	Yes

Java is the optimal choice for achieving Database functionality. Here are the reason as following. The document of Amazon DynamoDB provide the APIs for basic functionality in java such as inserting data, updating data and listing table. These APIs make the database functionality achieving more easily. Besides, Java is testable and more secure than the php. Another point is most of tool for our product is using Java.If we use the language other than java, the process will become complex because we also need to figure out the translation way between java and that language. This will also make the test process become much more complex. therefore , java is the suitable language for achieving Database functionality.

10 The visualization tool use to display the data

There are various visualization tools could be using to make the data visualization. Each visualization tool has different features. Our goal for this part is to figure out the optimal choice of visualization tool for displaying the data. The visualization tool we choose to evaluate are Tableau,QuickSight and FusionChart. we generate several criteria for evaluating them such as platform, speed, features and cost.

Visualization Tool	Platform	Speed	Features	Cost
Tableau	Tableau Online	Fast	Allows cross database, beautiful design	Low
QuickSight	Amazon Web Services	Fast	High accessibility, Get answer fast, Easy share business insight, Smart Visualizations	Low
FusionChart	No platform	Fast	Controllable for chart	Normal

Tableau could generate the graph fast and the cost of it is not high. The QuickSight is supported by Amazon web services. It has high accessibility which allows access data from multiple source. Besides, it could share the business insight in security way.Another important feature for QuickSight is it could generate visualizations very fast for very large quantity of data.Furthermore, the cost of it is low[16]. FusionChart does not require the platform to support and it could generate the graph fast. The cost of Fusionchart is expensive than Tableau and QuickSight.

Comparing with Tableau and FusionChart, the QuickSight is the optimal choice for our product. The QuickSight is supported by Amazon web services which fits the requirement of platform. Besides, it has strong accessibility which allows to communicate with other data service on Amazon web services such as Amazon DynamoDB easily.Furthermore, it has fast speed for generate the large data set which fits our purpose. Therefore, the QuickSight will be the optimal option as the visualization tool use to display the data.