Biostatistics and Biometrics
Open Access Journal
ISSN: 2573-2633

Juniper Publishers
key to the Researchers

Review Article
Volume 5 Issue 4 - February 2018
DOI: 10.19080/BBOAJ.2018.05.555666

Biostat Biometrics Open Acc J

Review of Designing a Big Data Platform in Healthcare

Dillon Chrimes[1]*, Hamid Zamani[2], Belaid Moa[3] and Alex Kuo[2]

[1]*Database Integration and Management, Vancouver Island Health Authority, Canada*

[2]*Department of Human and Social Development, University of Victoria, Canada*

[3]*Compute Canada/West Grid/University Systems, University of Victoria, Canada*

Submission: January 27, 2018; **Published:** February 28, 2018

***Corresponding author:** Dillon Chrimes, Database Integration and Management, Vancouver Island Health Authority, V8R1J8, Victoria, BC, Canada, Email: dillon.chrimes@viha.ca

Abstract

In many hospital systems, new technologies that influence patient data require extensive technical testing before implementation in to production. Therefore, to implement, an existing High Performance Computing (HPC) Linux node clusters via WestGrid were used, and simulation of patient data benchmarked and cross-referenced with current metadata profiles in operational hospital systems at the Vancouver Island Health Authority (VIHA), Victoria, Canada. Over the tested cross platform, the data were generated, indexed and stored over a Hadoop Distributed File System (HDFS) to no SQL database (HBase) that represented three billion patient records. The study objective to establish an interactive Big Data Platform (BDA) was successful in that Hadoop/Map Reduce technologies formed the framework of the platform distributed with HBase (key-value No SQL database storage) and generated desired hospital-specific metadata at very large volumes. In fact, the framework over generated HBase data files took a week or a month for one billion (10TB) and three billion (30TB), respectively. Further performance tests retrieved results from simulated patient records with Apache tools in Hadoop's ecosystem. At optimized iteration, HDFS ingestion with HBase exhibited sustained database integrity over hundreds of iterations; however, to complete the bulk loading via Map Reduce to HBase required a month. Inconsistencies of Map Reduce limited the capacity to generate/replicate data to HBase efficiently. Hospital system based on patient encounter database was very difficult and data profiles were fully representative of complex patient-to-hospital relationships. Our work is important to lead discovery of big data technologies useful across platforms of hospital systems.

Keywords : Analytics; Big data; Cross platform; Distributed data; Distributed filing system; Healthcare; High performance computing; Hospital system; Interactive query; Relational database

Abbreviations : HPC: High Performance Computing; VIHA: Vancouver Island Health Authority; HDFS: Hadoop Distributed File System; BDA: Big Data Platform; GFT: Google Flu Trends; CMH: Children's Mercy Hospital; HER: Electronic Health Records; CPOE: Computerized Physician Order Entry; PACS: Picture Archiving and Communication Systems; CDSS: Clinical Decision Support Systems; PLIS: Provincial Laboratory Information Systems; BDA: Big Data Analytics; HDFS: Hadoop Distributed File System; UAT: Uer Acceptance Tests; ADT: Admission, Discharge and Transfer; DM: Deployment Manager; PBS: Portable Batch System; RM: Resource Manager; LDAP: lightweight directory access protocol; HPC: High Performance Computing; JDBC: Java Database Connectivity; DAD: Discharge Abstract Database; CIHI: Canadian Institute for Health Information's; BDA: Big Data Analytics; IB: Infinite Band; KV: key Value

Introduction

Big Data is a collection of data that is large, complex, distributed, and growing fast (or 5Vs- volume, variety, velocity, veracity, and Value) [1,2]. It has high potential for unlocking new sources of economic values, providing fresh insights into medical sciences and industrial systems while assisting on policy making [3]. Several published studies have asserted that Big Data managed efficiently can improve care delivery while reducing healthcare costs [4]. A McKinsey Global Institute study suggests, "If US healthcare were to use big data creatively and effectively to drive efficiency and quality, the sector could create more than $300 billion in value every year". A number of published articles also reported using Big Data to improve population health with better policy decision making. For example, Dugas et al.[5] collected 21 months (2009 Jan - 2010 Oct) of data at an urban academic hospital in Baltimore, Maryland US to assess the correlation of city Google Flu Trends (GFT) using big data technologies to find data trends over thousands of searched influenza cases. The study found that GFT had high correlation with the number of identified influenza results.

Similarly, in 2010, during a major Haitian cholera outbreak, daily reported case data for all departments from the Haiti Ministry of Health and daily volume of Twitter posts containing

the word "cholera" has similar large analytical patterns. The Twitter data (via Apache Storm - a big data technology) provided earlier disease outbreak information [6]. Similarly, in a study by Twist et al. [7] a platform Constellation (using Apache Storm and Hadoop in real-time) was successfully deployed at the Children's Mercy Hospital (CMH) in Kansas City (Missouri, US) to match patients' clinical data to their genome sequencing, thereby facilitating treatments and faster time for analytical results from 50 to 26 hours [8,9]. Moreover, Big Data in healthcare includes nationally standardized data collection schema, internationally accepted medical classification and terminology (e.g. ICD, SNOMED CT), semi-structured data for data transportation (e.g. HL7 messages), unstructured clinical notes (physicians' progress notes, medical images (e.g. MRI, X-rays), genetic lab data, and other types of data (e.g. public health and mental or behavioral health). Huge volumes of very heterogeneous raw data are generated daily by a variety of hospital systems such as Electronic Health Records (EHR), Computerized Physician Order Entry (CPOE), Picture Archiving and Communication Systems (PACS), Clinical Decision Support Systems (CDSS), and Provincial Laboratory Information Systems (PLIS).

In this study, we described our practical application among collaborations with Vancouver Island Health Authority (VIHA) funded research project "Design and Implement a Big Data Analytics Framework for Health Applications". The main objective of this project was to collaborate with the VIHA staff to develop and establish a BDA platform for application that applies to large sets of data collected from discharged patient records. A Hadoop/Map Reduce framework formed the platform with no SQL database called HBase representing real hospital-specific metadata and file ingestion. The modeled data produced through technological framework and processes, formed three billion of emulated patient data. This was generated and cross-referenced with inpatient profiles based on the metadata dictionaries provided through consultation and meeting with the VIHA's staff.

Literature review

Big Data Analytics (BDA) designed and engineered in many industrial systems is developed to extract knowledge via data mining processes from sets of Big Data [10]. Wang Lee, et al. [11] further described the extraction of useful knowledge from Big Data in terms of a processing pipeline that transfers, stores, and analyses data for whole systems. According to Chrimes et al. [12] review the process of achieving full Big Data utilization involves five distinct configuration stages; each with specific challenges, as follows:

Data aggregation: Copy/transfer data to a storage drive is a commonly used method of aggregating and migrating large quantities of data. Big Data research projects usually involve multiple organizations, geographic locations, and research IT teams; therefore, generating large datasets from replication away from production systems at hospitals removes any ongoing network resource consumption and database resources that could render the system in operable. Further, exchange of data between groups and databases is very difficult to coordinate; hence, a second database for big data should be carried out to ease its maintenance and slightly separate operational issues. Furthermore, transferring vast amounts of data over the network requires a significant amount of bandwidth over a consistent long duration. Additionally, to replicate data from the sources and generate it iteratively across instances and multiple nodes, as Hadoop Distributed File System (HDFS) can accomplish, does require batch processes or file block process [13-15].

Data maintenance: Since Big Data involves large volumes; it is very difficult to maintain access to all the data for ongoing queries. Moreover, time and cost can prohibit small organizations and technical system development and integration departments from managing large amounts of data. Another challenge, in healthcare, is that real patient data, metadata, and data profiles need to constantly be updated with clinical events table; otherwise, the analytics is rendered useless. There are many solutions available to provide maintenance including cloud computing [16], grid computing [17], No SQL/New SQL and other storage systems (e.g., MongoDB, HBase, Voldemort DB, Cassandra, Hadoop Distributed File System (HDFS) and Google's Big Table [18,19]. Legality and ethics is also a major issue in data maintenance. Security, confidentiality and privacy are all mandated by legislation with strict regulations. For example, the Health Insurance Portability and Accountability Act (HIPPA) require the removal of 18 types of identifiers, including any residual information that could identify patients. Privacy concerns can be addressed using new technologies, such as key-value storage services, but advanced configuration and technical knowledge is needed. For example, Pattuk et al. [20] proposed a framework for secure Big Data management involving an HBase database called Big Secret, which securely outsources and processes encrypted data over public key-value stores. Although hospitals house their data in server racks in a highly secure buildings and the vendors commonly are not allowed to use cloud services, especially when there is no control of the location.

Data integration: Data integration and interoperability processes involve combining and transforming data into an appropriate format for analysis. Since Big Data in healthcare are extremely large, distributed at different locations, unstructured and heterogeneous, the management of data integration over time is very time consuming [21]. Numerous solutions have been proposed for raw Big Data integration [22]. The problem with these methods is they are problem-oriented, i.e., the method is only applied to specific data sets or aggregates. Very few generic approaches exist across integrated unstructured data. 4. Data analysis: Data analysis or analytics is highly important for a successful BDA [23]. BDA complexity involves analytic algorithms to be programmed in that the computing time increases dramatically even with small increases in data

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

volume. However, it is very difficult to efficiently analyze the data interactively using traditional analytical software, such as IBM SPSS, Microsoft Excel or Math Works MATLAB because Big Data is too large, too heterogeneous and highly distributed over many locations in the healthcare continuum. It can take several days, and most likely months, to obtain a result over a very large data set (in terabytes and beyond). Moreover, for complex analyses, the computing time increases exponentially even with incremental growth in the data size. For example, Bayesian Network are a popular algorithm for modeling knowledge in computational biology and bioinformatics, and the computing time required to find the best network increases exponentially as the number of records rises [24]. Even for simple data analysis, it can take several days; even months, to obtain a result when databases are very large and SQL-like "joins" are executed.

Many studies suggest parallelization of computing model for high performance over analytical platforms to reduce computationally intense problems [25-30]. In addition, using the traditional analysis methods, the error rate related to analyzing large amount of data may add a new dimension to the challenge of analyzing large data sets. Whereas, in BDA, the large sets are frequently analyzed without any mention of the error dimension. To address the analytical challenges, many recently published studies have suggested that using High Performance Computing (HPC), and parallelization of computing model can efficiently increase analysis performance for the computationally intense problems.

Pattern interpretation: Knowledge representation is an absolute must to achieve for any data mining and BDA platforms. Further, BDA is of little value if decision-makers do not understand the patterns. Additionally, given the complex nature of the data and how big data technologies can increase users' interactive queries and utilization, representations of trends and individualized results will not be comprehensible to non-experts. Moreover, many people instinctively believe that bigger data means better information. But agile data science cannot protect us from inaccuracies and faulty assumptions. Many reporters are often fooled into thinking that correlations have true significance (or lack thereof).

Methodology

The basic premise of a BDA platform in healthcare was to construct the platform capable of compiling large heterogeneous clinical data from diverse sources while querying large volumes quickly and efficiently. We must create a simulated technical and operational environment with the cross platform over billions of patient records under performance of valuable data queries. Furthermore, it needs to be proven possible that it is beneficial to implement as a bolt-on solution to the current hospital system. Also, the applications were required to ensure patient data security and privacy with legislation, confidentiality laws and regulations. It must cover the complex relationships in the clinical event table of a patient in the hospital system with possible multi-encounters that are interrelated, for example chronic severe mental health, diabetes, and substance abuse as shown in Figure 1. As well as movement of patients in the hospital with medical services and health outcomes must be incorporated in the data schema to query interactively.

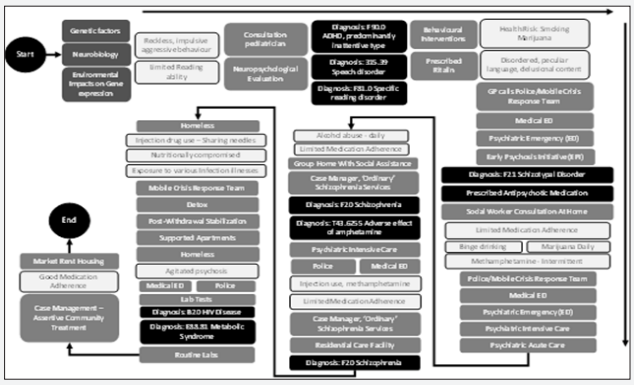

Figure 1: Patient can have multiple encounters and complex relationships in the patient record of the hospital system.

Platform architecture

The BDA platform harnesses the technical power and advanced programming to produce accessible front-end tools to end users that allow for analysis of large quantities of back-end data in an interactive enriching manner. All this must be accomplished cost effectively under rigorous and varied usability user acceptance tests (UATs) to be deployed to production. Based on the design philosophy of simulation with multiple interactive components for end users to query data at extreme volumes within seconds, we constructed a dynamic platform with interfaced backend applications, such as Apache Phoenix, Spark, and Drill, linked to backend HBase over Hadoop Distributed File System (HDFS). With the Hadoop/MapReduce framework, the platform allowed users to easily analyze and visualize health Big Data. The overall platform included four main components (Figure 2):

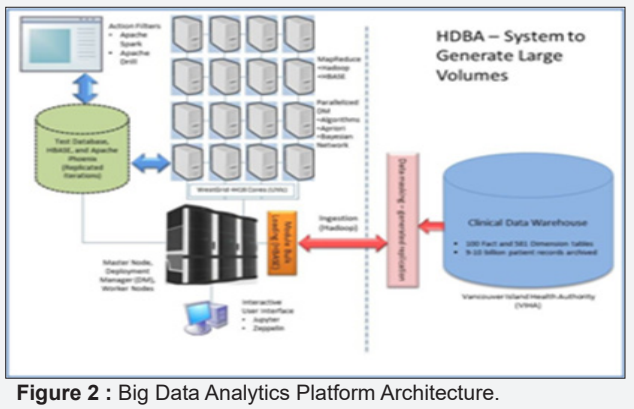

Figure 2 : Big Data Analytics Platform Architecture.

o A clinical data warehouse stores healthcare data. Currently at VIHA there are over 1000 tables in its Admission,

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

Discharge and Transfer (ADT) data from hospital system, and annually ca. one million patient encounters add to 50+ years archive (500 million at VIHA and 10 billion provincially).

o High performance Linux clusters (West Grid University System) were used to install software, build configurations, and run simulation queries (Hadoop ecosystem, Apache Phoenix, Spark and Drill).

o HBase no SQL database was used to store data from VIHA clinical data warehouse. HDFS distributes the data to indexed storage across the West Grid clusters with backup, high availability and redundancy.

o A master deployment manager (DM) was used to access the clusters from sponsored accounts over the Portable Batch System (PBS) of the Resource Manager (RM). The access to the DM is controlled by lightweight directory access protocol (LDAP) while accessing worker nodes was restricted to only the user running the job. This architecture permitted an agile and stabilized access with system administrator that could be launched from any terminal for each PBS job.

High performance computing (HPC) infrastructure

In this study, as described above, we relied on West Grid's existing architecture as the computing infrastructure. West Grid is a nationally Canadian funded program since 2003, mainly used in western Canada while EastGrid and Ontario and Quebec grids are available. WestGrid installation at the University of Victoria (UVic) started in July 2010. The WestGrid computing facilities at UVic have 2 main clusters called Hermes and Nestor. Hermes is a capacity cluster geared towards serial jobs with 84 nodes having 8 cores each and 120 nodes with 12 cores each, which gives a total of 2112 cores. Nestor is a large cluster consisting of 288 nodes (2304 cores) geared towards large parallel jobs. The computing system of these two clusters share infrastructure such as resource management, job scheduling, networked storage, and service and interactive nodes. In this study, we use five dedicated worker nodes and one head node from Hermes cluster.

Conceptual analytics framework

Under the umbrella of the hospital system and its end users, the framework and the applications would allow users to query, visualize, interpret, and modify outputs of the data. The overall purpose was to make Big Data capabilities accessible to stakeholders, including UVic researchers, VIHA physicians and database administrators, and other healthcare practitioners. Our analytics framework on the platform includes nine technical integrated parts of the system:

A clinical data warehouse was part of the main goal to achieve billions of patient records to represent Big Data in healthcare application. Currently, the data warehouse at VIHA has over 1000 relational tables of its hospital system that encompass ten billion records archived over a period of ~50 years. Health data are continuously generated and added to the warehouse at a rate which has grown exponentially to over one million encounters annually at VIHA.

Table 1: Data schema of patient data.

Create Table If Not Exists Dads1
EncounterID BIGINT NOT NULL,
Admit_by_Ambulance VARCHAR,
Admit_Category VARCHAR,
Admission_Date VARCHAR,
Admission_Time VARCHAR,
Age INTEGER,
Anesthestic_Grade VARCHAR ,
Anesthetist_ID VARCHAR,
Anesthetistic_Technique INTEGER,
Arrival_Date_in_ER VARCHAR NOT NULL,
Arrival_Time_in_ER VARCHAR NOT NULL,
Date_Patient_Left_ED VARCHAR ,
Date_of_Transfer_In VARCHAR,
Days_in_Unit VARCHAR,
Discharge_Date VARCHAR NOT NULL,
Discharge_Disposition VARCHAR NOT NULL,
Discharge_Site VARCHAR NOT NULL,
Discharge_Time VARCHAR NOT NULL,
Birth_Date VARCHAR,
Diagnosis_Cluster VARCHAR,

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

Diagnosis_Code VARCHAR NOT NULL,
Diagnosis_Occurrence INTEGER,
Diagnosis_Prefix VARCHAR,
Diagnosis_Type VARCHAR,
Entry_Code VARCHAR,
Episode_Duration INTEGER,
First_Name VARCHAR,
Glasgo_Coma_Scale VARCHAR,
Gender VARCHAR,
Health_Care_Number VARCHAR NOT NULL,
HCN_Prov VARCHAR,
Institute_From INTEGER,
Institution_To INTEGER,
Interven_Attribute_Extent VARCHAR,
Interven_Attribute_Location VARCHAR,
Interven_Attribute_Status VARCHAR,
Interven_Code VARCHAR,
Interven_Episode_Start_Date VARCHAR,
Interven_Episode_St_Date VARCHAR,
Interven_Location VARCHAR,
Interven_Occurrence INTEGER,
Interven_Options VARCHAR,
Interven_Pr_Number INTEGER,
Interven_Preadmit_Flag VARCHAR,
Interven_provider_service INTEGER,
Interven_Start_Time_unknown VARCHAR,
Interven_St_T_unknown VARCHAR,
Last_Name VARCHAR,
LOS INTEGER NOT NULL,
Middle_Name VARCHAR,
Most_Responsible_Site VARCHAR,
MRN VARCHAR,
Out_of_Hospital_ooh_indicator VARCHAR,
Out_of_hospital_ooh_number INTEGER,
PHN INTEGER,
Postal_Code VARCHAR,
Pdr_Number INTEGER,
Provider_Occurrence INTEGER,
Provider_Service VARCHAR,
Provider_Type VARCHAR,
Patient_Service INTEGER,
Patient_Service_Days INTEGER,
Patient_Service_Occurrence INTEGER,
Patient_Service_Type VARCHAR,
Readmit_Code INTEGER,
Reporting_Prov INTEGER,
Residence_Code INTEGER,

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

Responsibility_for_payment INTEGER,
Service_Nursing_Area VARCHAR,
Time_Patient_Left_ED VARCHAR,
Time_Pt_left_ED_unknown VARCHAR,
Transfer_Hours VARCHAR,
Transfer_Nursing_Unit VARCHAR,
Transfer_In_Date VARCHAR,
Transfer_Out_Date VARCHAR,
Unit_Transfer_Occurrence INTEGER,
Unplanned_return_to_OR VARCHAR,
Wait_Time_in_ED VARCHAR,
FIN INTEGER NOT NULL,
Encounter_Number INTEGER NOT NULL,
Admit_Source VARCHAR,
Encounter_Type VARCHAR,
Medical_Services VARCHAR,
MostResponProvider VARCHAR,
Address VARCHAR,
Family_Physician VARCHAR,
Location_Building VARCHAR NOT NULL,
Location_unit VARCHAR NOT NULL,
Location_Room VARCHAR NOT NULL,
Location_Bed VARCHAR NOT NULL,
CONSTRAINT PK PRIMARY KEY (EncounterID, Arrival_Date_in_ER, Arrival_Time_in_ER, Discharge_Date, Discharge_Disposition, Discharge_Site, Discharge_Time, Diagnosis_Code, Health_Care_Number, OS, FIN, Encounter_Number, Location_unit), Salt Buckets =5.

A back-end No SQL database of HBase with indexing rows to columns uniquely and key-stores that encrypts the data. In the emulation of the database, each row represented encounter-based patient data as a Big Integer, with diagnoses, interventions, and procedures specific to that patient, which the current ADT system has in its database schema linked to a bigger data warehouse, which includes DAD (Table 1). This patient-specific structure in the database allowed for active updates to add to the data generation while maintaining accurate patient querying over the platform. Patient-specific rows across the columns according to the existing abstraction were essential part of the emulation; HBase established a wide range of indexes for each unique row, and each row contained a key value that was linked to the family of qualifiers and primary keys (columns). HBase was chosen due to its No SQL services, especially linear to modular scalability to document architecture. In addition, it allows for SQL-like layer of Apache Phoenix to be configured on top of HBase big-tables. The HBase operations were specific to family qualifiers at each iteration; therefore, the data was patient-centric combined with certain DAD data (from different sources of metadata) in the rows and columns, such that summary of diagnosis or medical services as a whole could be queried. The BDA platform was built on top of the available open-source software (HBase). HBase (No SQL version 0.98.11) is a No SQL database composed of the main deployment master (DM) node and its fail-over master, the Region Servers holding HBASE data, and Zoo Keeper, which contained services to allocate data locality (Zookeeper), of three nodes, that orchestrated that ensemble. The xml configuration files were HBase-site.xml and the HBase-env.sh were adjusted to improve the performance of HBase.

The construction and build of the framework of HBase (No SQL) across Hadoop (HDFS)\Map Reduce established the BDA platform. This construct coincided with and was enforced by the existing architecture of the West Grid clusters at UVic (secure login via LDAP directory to deployment database nodes and restricted accounts to dedicated nodes). It was initially running the architecture of the platform with five worker nodes and one master node (each with 12 cores). The data were distributed in parallel on the nodes via a balanced allocation to each local disk with running part of the batch jobs with set metadata and columns for each row up to 50 million in a serial computing process that generated replications.

A HPC clusters with a total of 4412 cores with batch processing and parallelized nodes. Hermes has 2112-core capacity cluster(s) geared towards serial jobs that can be distributed. It consists of 84 nodes having eight cores each and 120 nodes with 12 cores each. The systems are designed for high-performance

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

and advanced-research computing and include clusters with fast interconnection and shared memory systems. Serial programs run on one CPU or core on a computing cluster node. Parallel programs, on the other hand, may have multiple processes or threads running at the same time, so that installations need to communicate to carry out their tasks. This study used both types to send job to ingest file and also to run the Hadoop/Map Reduce framework or parallel program process. It also utilized the batch serial process to access and start jobs over the Hadoop top-down head node to slave architecture.

A master DM is the portable batch serial login that was configured as head node to worker nodes. Deployment of the Hadoop environment on the nodes carried out via a sequence of setup scripts that the user calls after loading the modules and setup additional configuration to the head node with YARN and ZooKeeper as allocators of various deployments. Setup scripts created an initial configuration depending on the number of nodes chosen when launching the job. The user can adjust those configurations to match the needs of the job and its performance.

Making the BDA platform InfiniBand-enabled was challenging, as most of the Hadoop environment services rely on the hostname to get the IP address of the machine. Since the hostnames on a cluster are usually assigned to their management network, the setup scripts and the configuration files required adjustment. The InfiniBand was used because it offers low latency (in us) and high bandwidth (~40Gb/s) connectivity between the nodes. YARN, Hadoop's resource and job manager, un-fortunately still partly used the Gig-Ethernet interface when orchestrating between the nodes, but the data transfer was carried out on the InfiniBand. Yarn was the resource manager of Hadoop and services of scheduling incongruent to running the Hadoop jobs. In addition to the Map Reduce component, Yarn and HDFS constitute the main components.

The queries via Apache Phoenix (version 4.3.0) resided as a thin SQL-like layer on HBase. This allowed ingested data to form structured schema-based data in the No SQL database. Phoenix can run SQL-like queries against the HBase data. Similar to the HBase shell, Phoenix is equipped with a python interface to run SQL statements and it utilizes a .csv file bulk loader tool to ingest a large flat file using Map Reduce. The load balancing between the Region Servers (e.g., "salt bucket") was set to the number of slaves or worker nodes that allowed ingested data to be balanced and distributed evenly. Apache Spark was also built from source and installed to use on HBase and the Hadoop cluster. Spark utilizes Yarn and HDFS architecture and is known to scale and perform well in the data space (over distributed files over multiple nodes). Inspired by Google's big query engine Dremel, Drill supports a wide range of data sources, including .csv, JSON, HBase, etc...[31]. By (re)compiling and optimizing each of the queries while it interacts with the distributed data sets via the so-called drillbit service, Drill showed capacity of the query with performance at a low latency SQL query.

Developing tools

Several of the open-source software were installed and configured to form the analytics platform. The software stack formed is shown in Figure 3. Apache Zeppelin 0.6.0 is a web-based notebook that enables interactive data analytics via local host and Spark-SQL. It has many built-in visualization features to support knowledge presentation. Similarly, Jupyter 4.0.6 (formerly known as iPython Notebook) is an interactive notebook that supports users to interact with data in various programming languages and combine code with markdown text to perform visualizations. Apache Spark (1.3.0 to 1.5.2) is an open-source parallel processing framework to utilize Yarn (Hadoop's resource manager) and less use of Map Reduce for running large-scale data analytics applications across computer clusters. It has its own SQL-like queries built in. Apache Phoenix 4.3.0 is an open-source skin on HBase that provides a Java Database Connectivity (JDBC) driver and SQL-like access. Phoenix enables Online transaction Processing (OLTP) and operational analytics over Hadoop's foundation for low latency applications. It compiles SQL-like query into a series of HBase scans, and it orchestrates the running of those scans to produce regular JDBC result sets. Phoenix is fully integrated with other Hadoop products such as Spark, Hive, Pig, Flume, and Map Reduce.

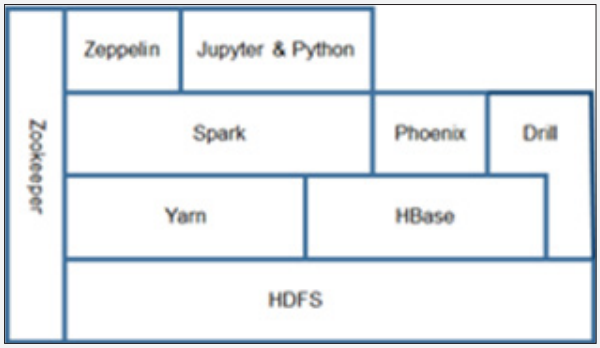

Figure 3 : Software stack.

Drill 1.3.0 is also Apache open-source software (established 2015) that supports data-intensive distributed large-scale datasets using SQL ANSI:2003 query types. It offers allow latency query engine and utilizes ZooKeeper. As opposed to the master/slave architecture of Spark in which a driver is handling the execution of the Directed Acyclic Graph (DAG) on a given set of executors, the drillbits are loosely coupled and each can accept a query from the client (Chrimes, Moa, Kuo, and Kushniruk). To run Drill over a distributed mode, the user will need a ZooKeeper cluster continuously running. Drill 1.3.0 and ZooKeeper 3.4.6 were installed and configured on the framework of the platform over a port with a local host. The receiving drillbit becomes the driver for the query, parsing, and optimization over a generated efficient, distributed, and multiphase execution plan; it also gathers the results back when the scheduled execution is done [32,33]. At the foundation of the stack is HDFS. It is the most

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

important foundational component of the platform. Yarn is there source manager of Hadoop and services of scheduling incongruently running the jobs. In addition to Map Reduce component, Yarn and HDFS constitute the main components of Hadoop 2.6.0 version.

Data privacy protection

Ensuring patient data security and privacy was an important requirement in this study. The established platform used the three following methods to protect data security and privacy:

Data masking: Typically this is carried out by database administrators thru rules and regulations set by business/security analysts based on current legislations of BC Ministry of Health. The goal was to generate a comprehensive list of sensitive elements specific to the organization and associated tables, columns, and relationships across the data warehouse and encryption of indexed key stores provided by HBase.

Data replication: We worked in conjunction with Business Intelligence and Data warehouse, Clinical reporting, Application Platform Services, Database Administrators, and Physicians/Nurses groups to identify the masking or encryption required and optimal techniques to de-identify and restrict access to patient data. Once the data form distributed HBase data sets across working nodes, it was queried via Apache Phoenix, Spark and Drill only thru PBS held by West Grid.

Using HBase over security/Privacy mechanisms: HBase provided comprehensive security/privacy support thru its qualifiers and key-stores of data ingested. The access control to data stored in HBase was at table level, column family level and column level. HBase supports Kerberos authentication, Remote Procedure Call (RPC) and at-rest privacy protection. Data could not be queried without West Grid for authentication.

Findings/Results

Data emulation and modeling

Over the span of twelve months in 2014-2015, several interviews were conducted with business intelligence data warehouse, clinical reporting, application platform, and health informatics architecture teams employed at VIHA (Table 2). During these interviews, an emulated health Big Data was generated from hospital admissions (based on encounter types) and a discharge system (based on diagnoses and procedures). In it, data profiles (including dependencies) and the importance of the metadata for the clinical reporting were confirmed and verified. Furthermore, current reporting limitations of the different combinations of the DAD and ADT data were recorded to form accurate simulation of the existing and future queries. To test the feasibility of the BDA platform and its performance, the emulated patient data had 90 columns that combined DAD and ADT metadata profiles.

Table 2: Use cases and patient encounter scenarios related to metadata of the patient visit and its placement in the database related to query output.

Case No.	Case Description	Column (Metadata) Used for Analysis	Database Build	Query Output
1	Uncontrolled Type 2 diabetes & Complex comorbidities	ICD10-CA, MRN, PHN and LOS, Discharge	DAD with Diagnosis Codes, patient IDs and Discharge in Columns	ICD10-CA codes with counts, frequencies or max values for patient encounters
2	TB of the lung & uncontrolled DM 2	ICD10-CA, MRN, PHN, Inpatient Encounter, Location, Unit Transfer	DAD and ADT columns	ICD10-CA and encounter type codes with counts, frequencies or max values for patient encounters
3	A on C Renal Failure, Fracture, Heart Failure to CCU and stable DM 2	ICD10-CA, MRN, PHN, Intervention (CCI), Episode, Unit Transfer, Bed Location, CCU codes, Discharge	DAD and ADT columns	ICD10-CA, CCI and encounter types and unit transfer and bed location codes with counts, frequencies or max values for patient encounters
4	Multi-location Cancer patient on Palliative	ICD10-CA, MRN, PHN, Intervention (CCI), Surgery, Episode, Bed Location, Transfer to ALC Unit, Medical Services and Patient Services, Discharge	DAD and ADT columns	ICD10-CA, CCI and encounter types and unit transfer and bed location and medical service codes with counts, frequencies or max values for patient encounters
5	1 cardiac with complications	ICD10-CA, MRN, PHN, Intervention (CCI), Surgery, Episode, Bed Location, Transfer, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, CCI and encounter types and transfer codes with counts, frequencies or max values for patient encounters
6	1 ER to surgical, Fracture, re-admit category 7 days and some complication after	ICD10-CA, MRN, PHN, Intervention (CCI), Surgery, Episode, Bed Location, Medical Services, Progress Notes, Discharge, Re-Admission	DAD and ADT columns	ICD10-CA, CCI and medical services and re-admit codes with counts, frequencies or max values for patient encounters

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

7	1 Simple Day-Surg. with complication, so got admitted to Inpatient (Allergy to medication)	ICD10-CA, MRN, PHN, Intervention (CCI), Surgery, Bed Location, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, CCI and medical services codes with counts, frequencies or max values for patient encounters
8	1 cardiac with complications and Death	ICD10-CA, MRN, PHN, Intervention (CCI), Episode, Bed Location, Transfer, Medical Services, Discharge Disposition	DAD and ADT columns	ICD10-CA, CCI and medical services, discharge disposition and transfer codes with counts, frequencies or max values for patient encounters
9	1 Normal birth with postpartum hemorrhage complication	ICD10-CA, MRN, PHN, Intervention (CCI), Surgery, Episode, Bed Location, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, CCI and medical services and discharge codes with counts, frequencies or max values for patient encounters
10	1 HIV/AIDS patient treats for underlying factor (an infection)	ICD10-CA, MRN, PHN, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, and medical services codes with counts, frequencies or max values for patient encounters
11	Strep A infection	ICD10-CA, MRN, PHN, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, and medical services codes with counts, frequencies or max values for patient encounters
12	Cold but Negative Strep A. Child with throat culture	ICD10-CA, MRN, PHN, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, and medical services codes with counts, frequencies or max values for patient encounters
13	Adult patient with Strep A. positive and physical exam	ICD10-CA, MRN, PHN, Medical Services, Patient Services, Discharge	DAD and ADT columns	ICD10-CA, patient and medical services codes with counts, frequencies or max values for patient encounters
14	Severe Pharyngitis with physical exam	ICD10-CA, MRN, PHN, Medical Services, Patient Services, Discharge	DAD and ADT columns	ICD10-CA, patient and medical services codes with counts, frequencies or max values for patient encounters
15	Child, moderate Pharyngitis, throat culture negative, physical exam	ICD10-CA, MRN, PHN, Medical Services, Discharge	DAD and ADT columns	ICD10-CA, and medical services codes with counts, frequencies or max values for patient encounters
16	Adult, history of heart disease, Positive culture for Strep A.	ICD10-CA, MRN, PHN, Medical Services, Patient Services, Discharge	DAD and ADT columns	ICD10-CA, patient and medical services codes with counts, frequencies or max values for patient encounters
17	Adult, physical exam, moderate pharyngitis, positive for strep A. culture and positive second time, re-admit	ICD10-CA, MRN, PHN, Medical Services, Patient Services, Discharge	DAD and ADT columns	ICD10-CA, patient and medical services codes with counts, frequencies or max values for patient, readmit encounters

We successfully benchmarked the performance of the BDA platform with clinical data warehouse utilization processes. Within the archive of data warehouse, two of the largest data sets are the Admission, Discharge, Transfer (ADT) and the Discharge Abstract Database (DAD). ADT has over 1000 tables with 75 columns containing individual patient bed-tracking information, while the DAD is set by a data dictionary (hundreds of data elements) of 28 columns contains Canadian Institute for Health Information's (CIHI) diagnostic codes and discharge abstract metadata. These data sets are not system linked to form an all-encompassing database. In a hospital system, the capacity to record patient data efficiently in the ADT is crucial to timely patient care and quality patient-care deliverables. Thus, the ADT system is often referred to as the source of truth for reporting operations of inpatient to outpatient and discharged. A suitable analysis of ADT and DAD integrated data in this study shows many benefits of using big data technologies to produce high volumes while interactively applying new ways to query the data to find unknown correlations and trends. Figure 4 shows the overall industrial design of the platform crossover and bolt-on with the current production system.

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

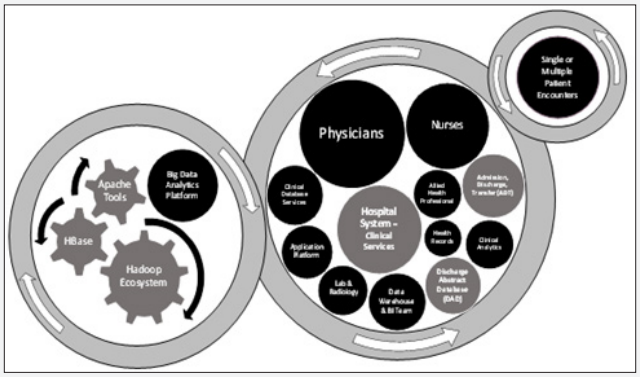

Figure 4 : Industrial design of the Big Data Analytics (BDA) platform as bolt-on solution to hospital system services involving physicians, nurses, health professionals, health records, data warehouse, database, analytics and application platform services maintained with patient encounters into system.

Data ingestion and query performance

The pathway to running ingestions and queries over the BDA platform includes nine user-to-system steps:

- Generating .csv flat files
- Apache Phoenix Module Load
- HDFS Module and Ingestion of H Files
- Bulk loading H Files to HBase
- HBase Compression
- Phoenix SQL-like Queries
- Apache Spark and Drill Module Loads
- Notebook and Python/Pypark Module Loads
- Spark and Drill SQL-like Queries

Thus this sequence, the Phoenix module loaded after Hadoop and HBase SQL code was directed and then iteratively run to ingest three billion rows to the existing HBase. Phoenix can run SQL-like queries against the HBase data. It was utilized to index and place schema over each .csv file bulk loaded to ingest using Map Reduce. The queries via Apache Phoenix resided as a thin SQL-like layer on HBase. This allowed ingested data where the batch loads were 50 million each via the index and schema between H Base's Region Servers thru a functional SQL-like code of "salt bucket" that set the number of worker nodes in the cluster to five evenly distributed data. This additional code was deemed necessary as HDFS did not automatically distribute data evenly and queried unbalanced data showed slow performance. Performance was measured with three main processes: HDFS ingestions, bulk loads to HBase, and query times. Three flat files (.csv) with different number of rows (50 million, 1 and 3 billon) were ingested to HDFS for testing (Table 3). At an optimized iteration, HDFS ingestion required three seconds but HBase required four to twelve hours to complete the Reducer of Map Reduce. H Base bulk loads took a week for one billion and over two months for three billion (Table 4). A SQL script containing all the queries was written and ran using Phoenix sqlline.py. The total number of queries that were used was 22: two simple queries with wildcard column selection; ten simple queries that did not involve more than three columns in the primary keys (family qualifiers); and, ten complex queries that had >3columns selected (Table 5). All queries run on Zeppelin, Jupyter, Spark-terminal and Pyspark, as well as Drill took approximately the time of 50-120 seconds to load the data and query. Spark was configured to run on specialized Yarn-client with 10 executors, four cores with 8 GB of RAM each; therefore, each node had two executors with a total of eight cores and 16 GB memory. However, Drill was faster with its configuration involving inherent ZooKeeper allocations via its drill bit components. For performance benchmarking, three metric measures were used: HDFS ingestion(s), bulk loads to HBase, and query times via Phoenix. We computed the ingestion efficiency (IE) and query efficiency (QE) of one billion compared to 50 million records using the formula in equation (1):

$$\text{IE, QE} = \frac{1(3)\text{B} \times \text{Ti}(50\text{M})}{50\text{M} \times \text{Til}(3)\text{B}} \qquad (1)$$

Table 3: HDFS data ingestions.

Data Size	Ingestion Time
50 Million records (23GB)	~3-6min
1 Billion records (451GB)	~60-120min
3 Billion records (10TB)	~180-360min

Table 4: H Base distributed data durations.

Data Size	Completion Time
50 Million records (0.5TB)	3-12hrs
1 Billion records (10TB)	60-240hrs
3 Billion records (30TB)	300-480hrs

Table 5: Performance of queries over 50 million (M), 1 and 3 billion (B) patient records.

Query Questions		Response Time (sec)		
		50M	1B	3B
1.	Wildcard Selection of Encounter Data	1.84	1.87	3.05
2.	Wildcard Selection of EncounterID with Ambulance Encounter	1.83	1.77	1.65
3.	Diagnosis (Dx) with LOS	1.14	1.47	2.11
4.	Frequency of Diagnosis (Dx) with LOS	1.64	1.68	2.32
5.	Dx with Discharge Date and Time	0.83	0.77	1.02
6.	Dx with Unit Transfer Occurrence	1.15	1.22	1.67
7.	Dx with Location of Building, Unit, Room, and Bed and Discharge Disposition	1.16	0.84	0.98

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

8.	Dx with Encounter Type & LOS	0.69	0.68	0.98
9.	Dx with Medical Services & LOS	0.44	0.38	1.02
10.	Provider Service with Dx	1.35	1.44	1.92
11.	Highest LOS for MRNs with Admit_ Date	1.45	1.46	1.62
12.	Frequency (or number) of Admit_ Category with Discharge_Date	1.86	1.76	1.89
13.	Admitted by Ambulance, Interventions, and Medical Services with Dx	1.79	1.87	1.89
14.	Intervention and Location with Admit_Dateand Admit_Time	1.64	1.75	1.87
15.	Medical Services with Unit Transfer Occurrences	1.83	1.75	1.92
16.	Admit Category and Discharge with Transfer	1.64	1.75	1.75
17.	Encounter Types with Discharge and Transfer	1.85	1.76	1.53
18.	Medical Services with Days in Unit	1.82	1.9	2.34
19.	Admission, Transfer with Intervention and Encounter	1.97	1.88	3.02
20.	Frequency (or number) of Admit_ Category with Patient Services	1.85	1.79	2.61
21.	Provider Occurrence with Nurse Services	1.62	1.65	1.7
22.	Provider with Dx and Intervention	1.85	1.75	1.71

Where, is the time it takes to ingest N records to either HDFS or HBase. Figure 5 shows the fluxes of the IE for all 60 iterations to three billion. Furthermore, QE performance of Apache Spark and Apache Drill on the exact same data (Figure 6).

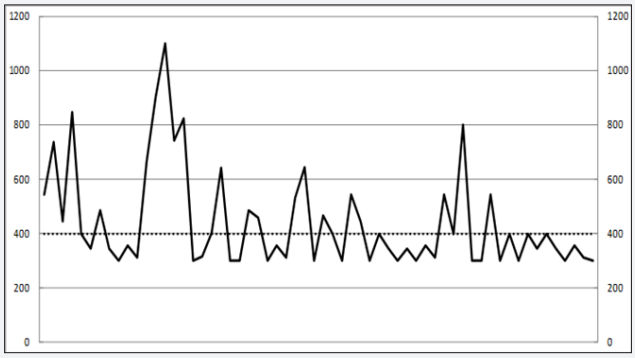

Figure 5 : Sixty iterations of H File to bulk loading via Map Reduce to HBase distributed database. Solid line is actual minutes to complete all job tasks and the dotted line is average duration.

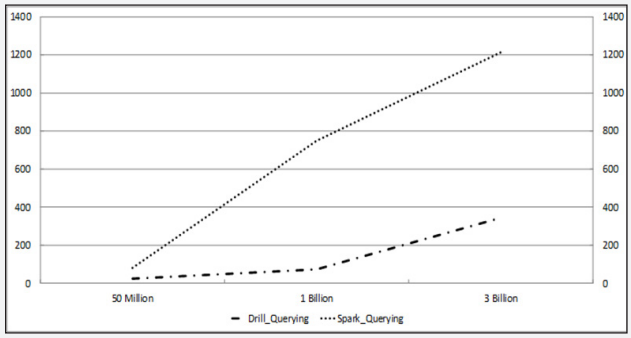

Figure 6 : Drill and Spark query performance.

Discussion

The Big Data Analytics (BDA) platform with Hadoop/Map Reduce framework over HBase was successfully implemented. The primary objective of established proof-of-concept of interactive platform with high performance was achieved. Further evaluation of the process of how to configure to efficiently operate and analyze patient data over a distributed computing system in operational hospital system was accomplished. Nevertheless, there were a few challenges to maintaining the platform operationally: major undertaking to manually run batches of file ingestions to accurately update the database without errors. Furthermore, ongoing update of software versions and integration with Hadoop's ecosystem requires constant maintenance of operating system. Despite having a comprehensive data security/privacy covered by access, authentication and key-store data encryption of HBase, we still faced similar difficulties to obtain real patient data for research. Nonetheless, we used simulated data of real schemas of hospitalization data of VIHA's data warehouse for further change control and development of net new technologies utilizing patient data in non-production environments to test for production.

The bulk of the methodology of the replication of operational generations of the emulated data and queries with Hadoop configurations and other software, i.e., HBase, Spark and Drill, was completed over industrial design and integration of technical Big Data components. Most of the configurations were somewhat out-of-the-box installations of the distributed filing system and Map Reduce components in Java, Python and Scale to perform as expected. Therefore, Hadoop-Map Reduce configurations established the platform. It produced the bulk loads of 50 million rows in iteration to one billion were slow. Ingestions to three billion were even slower. It was found that these slow performances were caused by Reducer in placing data to map of database (Mapper was fast ~1-6min). Additionally, each iteration of 50 million rows to form key-valued no SQL database took longer because the HBase had to be compacted (and re-indexed) after each batch before running the next iteration. Number of failed ingestions increased as the file

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

system grew more than expected and compression of HBase had to run more frequently.

If we were to re-run the iteration without running compression, one or all the servers would go "Dead" resulting in system crash and cleanup of the data that was distributed partially in that iteration. At times, a full clean-up was required, so after running HBase for too long, removing error messages required clearing out content on the cluster, re-starting the modules, and even re-loading Hadoop and HBase. Thus, the process of reaching to 3 billion iterations took about a month but these times were as fast as or faster than current data migrations (of a similar size) estimated. Finally, since we did not use real patient data, advanced health patterns or trends were only confirmed to be randomized replicated data clusters.

The most impactful technology of the Big Data components in this study was Map Reduce (and Java code performance therein). Map Reduce methodology is inherently complex as it has separate Map and Reduce task and steps in its defaulted programming framework, as this study discovered. This study's platform was highly dependent on the efficiency of Map Reduce in ingesting files over the six nodes, using this workflow: input → map → copy/sort → reduce → output similar to a study by Chen, et al. [34]. Once configurations in Yarn, Zoo Keeper and others, the Reducer were optimized with iterations of 50 million rows with data, integrity of the desired clinical event model was established via SQL-like in Apache Phoenix. According to blogs with technical resolutions, enabling or disabling services or xml settings over the platform as expected to be carried because the system relied heavily on InfiniBand (IB) bandwidth. Also there are known issue with using Map Reduce over HBase with slow performance after additional indexing of data and its store [35-38].

The data used in this study consisted of diagnosis codes, personal health numbers, medical record numbers, dates of birth, and location mnemonics (to mention only a few of the 90 columns), as these codes are standardized for hospital systems and, compared to genetic data, easier to replicate in metadata in large volumes. The use of groups of events allows the definition of a phenotype to go beyond diagnosis as coded using the International Classification of Disease, version 9, codes (ICD-9) and potentially allows assessment of the accuracy of assigned codes [39,40]. In healthcare, the complexity of Big Data storage and querying increase with unstructured sets of data and/or images. Images take up lots of storage capacity and are difficult to process and next to impossible to query in large volumes. The growth in the volume of medical images produced on a daily basis in modern hospitals has forced a move away from traditional medical image analysis and indexing approaches towards scalable solutions [41]. Map Reduce has been used to speed up and make possible three large-scale medical image processing use-cases:

I. Parameter optimization for lung texture classification using support vector machines (SVM),

II. Content-based medical image indexing/retrieval, and

III. Dimensional directional wavelet analysis for solid texture classification [42].

In their study, as in our study, a default cluster of heterogeneous computing nodes was set up using the Hadoop platform, allowing for a maximum of 42 concurrent Map tasks. The present study did not test the amount and efficiency of concurrent Map tasks of Map Reduce to process the data to HBase ingestions; this is something to be investigated further as using real hospital data that is highly unstructured and rich in images might require this enhancement. Moreover, this study ran up against limitations in the ability of the Reducer component of Map Reduce more than Map tasks to form the bulk loads of HBase and its No SQL schema, and, therefore, the Reducer improvements should be investigated before Map tasks.

The complex nature of HBase means that it is difficult to test the robustness of the data in emulations based on real data. Several steps were required to prepare the DAD database alone for statistical rendering before it was sent to CIHI. The actual columns used in this study are the ones used by VIHA to derive the information accurately in a relational database, which ensures the data is in alias pools and not duplicated for any of the encounters. The DAD data also makes calculations (which in the reporting workflow adds columns), which is what the reporting and data warehouse teams also do to form their databases. Adding columns to a No SQL database is much easier than adding columns to a SQL relational database, and von et al. [43] showed good performance of multi-term keyword searches. Therefore, it is an advantage to have a large database with row keys and column families already set up; this is supported by Xu et al. [44], as their middleware ZQL could easily convert relational to non-relational data. However, the indexing of HBase proved to decrease the time to ingest the data accurately (so that queries produced accurate information).

If the indexing was not reiterated, with the addition of new columns or rows, then the data cannot be queried at all. Spark and Drill performed well with the larger volumes, thus offering an alternative to HBase (without the indexing); however, the data cannot be fully represented without indexing if real patient data is to be used. Not all columns are known, but mimicking an index with family columns does enable simulation of the data model of the hospital. This is a significant drawback, and more data modeling of the relational to the non-relational database is required. Essentially this study is proposing a row-column key-value (KV) model to the data distributed over a customized BDA platform for healthcare application. Wang, et al. [45] support this study's claim in their statement that non-relational data models, such as the KV model implemented in No SQL

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

databases. Wang, et al. [46] further stated that No SQL provided high performance solutions for healthcare, being better suited for high-dimensional data storage and querying, optimized for database scalability and performance. A KV pair data model supports faster queries of large-scale microarray data and can be implemented using HBase (an implementation of Google's Big Table storage system). The new KV data model implemented on HBase exhibited an average 5.24-fold increase in high-dimensional biological data query performance compared to the relational model implemented on MySQL Cluster and an average 6.47-fold increase on query performance on Mongo DB. Their performance evaluation found that the new KV data model, in particular its implementation in HBase, outperforms the relational model currently implemented; therefore, it supports this study's proposed No SQL technology for large-scale data management over operational BDA platform.

Nelson and Staggers [47] have stressed the importance of patient data modeling with Big Data platforms in healthcare, indicating that a lack of BDA ecosystems is one of the reasons why healthcare is behind other sectors in utilizing current technologies to harness Big Data. Nelson & Staggers [47] noted that nursing informatics and data from nurse progress notes are underutilized in hospital systems. Wang et al. [11] also compare bioinformatics with healthcare and Big Data applications. Bioinformatics can match extremely large libraries of genetic data to libraries of medications or treatments; however, such matching in real-time over large hospital systems of patient-centric frameworks in Canada is difficult due to current traditional data warehousing practices of storing relational data. Chawla & Davis [48] and Kuo et al. [49] argue that even structured data lack interoperability among hospital systems, so that no solutions could possibly link all data. At VIHA, for example, it is difficult to link the DAD and ADT data on encounters, because the DAD data on diagnosis and intervention are not stored together or integrated or have relational dependencies in an all in one data warehouse, while the ADT automatically links the data to encounters. Therefore, more validation is required to match corresponding medical services in ADT to patient diagnosis from the DAD.

Srirama, et al. [50] indicated that Hadoop is suitable for simple iterative algorithms where they can be expressed as a sequential execution of constant Map Reduce models (that could also be configured to be representative of the clinical event model of hospital systems). It is not well suited for complex statistical analysis or iterative problems. To amend the Hadoop's ecosystem weaknesses, we plan to engineer "R" to work over Hadoop (e.g. RHadoop). R and Hadoop complement each other very well in BDA and in data visualizations [51-53].

Conclusion

In this study, Hadoop/Map Reduce framework was proposed to implement the data-intensive distributed computing platform. Few studies have tested Big Data tools in Hadoop's ecosystem in healthcare. And even fewer studies have established a simulation of 3 billion patient records. Therefore, this study achieved the top three V's that define Big Data: high performance (or velocity) over its generator of detailed data (or variety) that formed extremely large quantities (or volume).Our future work will involve user acceptance testing under different simulations and clinical event models.

Acknowledgement

This study was financially supported by the University of Victoria – Island Health Authority collaborative research grant, Victoria, BC, Canada.

References

1. Hansen MM, Miron-Shatz T, Lau AY, Paton C (2014) Big Data in Science and Healthcare: A Review of Recent Literature and Perspectives, Yearbook of Med Inform 9: 21-26.
2. Manyika J, Chui M, Bughin J, Brown B, Dobbs R, et al. (2014) Big data: The next frontier for innovation, competition, and productivity.
3. Canada Health Infoway (2013) Big Data Analytics in Health - White Paper.
4. Raghupathi W, Raghupathi V (2014) Big data analytics in healthcare: promise and potential. Health Inf Sci Syst 2: 3.
5. Dugas AF, Hsieh YH, Levin SR, Pines JM, Mareiniss DP, et al. (2012) Google Flu Trends: correlation with emergency department influenza rates and crowding metrics. Clin Infect Dis 54(4): 463-469.
6. Chunara R, Andrews JR, Brownstein JS (2012) Social and news media enable estimation of epidemiological patterns early in the 2010 Haitian cholera outbreak Am J Trop Med Hyg 86(1): 39-45.
7. Twist GP, Gaedigk A, Miller NA, Farrow EG, Willig LK, et al. (2016) Constellation: a tool for rapid, automated phenotype assignment of a highly polymorphic pharmacogene, CYP2D6, from whole-genome sequences. NPJ Genomic Med 1: 15007.
8. Miller NA, Farrow EG, Gibson M, Willig LK, Twist G, et al. (2015) A 26-hour system of highly sensitive whole genome sequencing for emergency management of genetic diseases. Genome Med 7(1): 100.
9. Saunders CJ, Miller NA, Soden SE, Dinwiddie DL, Noll A, et al. (2012) Rapid Whole-Genome Sequencing for Genetic Disease Diagnosis in Neonatal Intensive Care Units. Sci Trans Med 4(154): 154ra135.
10. Baro E, Degoul S, Beuscart R, Chazard E (2015) Toward a literature-drive definition of big data in healthcare. Biomed Res Int doi: 10.1155/2015/639021.
11. Wang B, Li R, Perrizo W (2014) Big Data Analytics in Bioinformatics and Healthcare, Medical Information Science Reference.
12. Chrimes D, Kuo MH, Moa B, Hu W (2017) Towards a Real-time Big Data Analytics Platform for Health Applications. International Journal of Big Data Intelligence 4: 2.
13. Grover M, Malaska T, Seidman J, Shapira G (2015) Hadoop Application Architectures: Designing Real-World Big Data Applications. O'Reilly Publishing. San Francisco, California, USA.
14. Lai WK, Chen YC, Wu TY, Obaidat MS (2014) Towards a framework for large-scale multimedia data storage and processing on Hadoop platform. J Supercomput 68: 488-507.
15. White T (2015) Hadoop-The Definitive Guide: Storage and analysis at internet scale. O'Reilly Publishing. San Francisco, California.

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

16. Dai L, Gao X, Guo Y, Xiao J, Zhang Z (2012) Bioinformatics clouds for big data manipulation. Biol Direct 7(43): 1-7.

17. Mohammed EA, Far BH, Naugler C (2014) Applications of the MapReduce programming framework to clinical big data analysis: current landscape and future trends. BioData Min 7: 22.

18. Lith A, Mattson J (2010) Investigating storage solutions for large data: a comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data. Master of Science Thesis, Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden.

19. Moniruzzaman ABM, Hossain SA (2013) NoSQL Database: new era of databases for Big Data Analytics - Classification, Characteristics and Comparison. International Journal of Database Theory and Application, 6(4): 1-14.

20. Pattuk E, Kantarcioglu M, Khadilkar V, Ulusoy H, Mehrotra S (2013) Big Secret: A secure data management framework for key-value stores. Tech. Rep.

21. Martin Sanchez F, Verspoor K (2014) Big Data in Medicine Is Driving Big Changes. Yearbook of Medical Informatics 9(1): 14-20.

22. Chen J, Chen Y, Du X, Li C, Lu J, et al. (2012) Big data challenge: a data management perspective. Frontiers of Computer Science 7(2): 157-164.

23. Chrimes D, Moa B, Kuo MH, Kushniruk A (2017) Operational Efficiencies and Simulated Performance of Big Data Analytics Platform over Billions of Patient Records of a Hospital System. Advances in Science, Technology and Engineering Systems Journal 2(1): 23-41.

24. Schadt EE, Linderman MD, Sorenson J, Lee L, Nolan GP (2010) Computational solutions to large-scale data management and analysis. Nature Reviews 11(9): 647-657.

25. Deepthi K, Anuradha K (2016) Big data Mining Using Very-Large-Scale Data Processing Platforms, International journal of engineering research and applications 6(2): 39-45.

26. Marozzo F, Talia D, Trunfio P (2012) P2P-MapReduce: Parallel data processing in dynamic Cloud environments. Journal of Computer and System Sciences 78(5): 1382-1402.

27. Taylor RC (2010) An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC Bioinformatics 11(12): S1.

28. Vaidya DP, Deshpande SP (2015) Parallel Data Mining Architecture for Big Data, International journal of electronics, communication and soft computing science and engineering 5: 208-213.

29. Wu X, Zhu X, Wu GQ, Ding W (2014) Data Mining with Big Data, IEEE Transactions on Knowledge and Data Engineering 26(1): 97-107.

30. Zhang YP (2015) i2 MapReduce: Incremental MapReduce for Mining Evolving Big Data, IEEE Transactions on Knowledge and Data Engineering, 27(7): 1906-1919.

31. Sitto K, Presser M (2015) Field Guide to Hadoop - An Introduction to Hadoop, Its Ecosystem, and Aligned Technologies, O'Reilly Media, San Francisco, USA.

32. Dunning T, Friedman E, Shiran T, Nadeau J (2016) Apache-Drill. O'Reilly Media Publications, USA.

33. Jurney R (2013) Agile data science: building data analytics applications with Hadoop. O'Reilly Publications. San Francisco, California.

34. Chen Y, Alspaugh S, Katz R (2012) Interactive Analytical Processing in Big Data Systems: A Cross-Industry Study of MapReduce Workloads. Proceedings of the VLDB Endowment 5(12): 1802-1813.

35. Greeshma AL, Pradeepini G (2016) Input split frequent pattern tree using MapReduce paradigm in Hadoop. Journal of Theoretical and Applied Information Technology 84(2): 260-271.

36. Maier M (2013) Towards a Big Data Reference Architecture. Master's Thesis. Eindhoven University of Technology, Department of Mathematics and Computer Science.

37. Sakr S, Elgammal A (2016) Towards a comprehensive data analytics framework for smart healthcare services. Big Data Research 4: 44-58.

38. Yu SC, Kao QL, Lee CR (2016) Performance Optimization of the SSVD Collaborative Filtering Algorithm on MapReduce Architectures. 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, 1(3): 612-619.

39. Freire SM, Teodoro D, Wei-Kleiner F, Sundsvall E, Karlsson D, et al. (2016) Comparing the Performance of NoSQL Approaches for Managing Archetype-Based Electronic Health Record Data. PLoS ONE 11(3): e0150069.

40. Hripcsak G, Albers DJ (2013) Next-generation phenotyping of electronic health records. J Am Med Inform Assoc 20(1): 117-121.

41. Wang F, Lee R, Liu Q, Aji A, Zhang X, et al. (2011) Hadoop-GIS: A high performance query system for analytical medical imaging with MapReduce. In: Atlanta - USA: Technical Report, Emory University, p. 1-13.

42. Markonis D, Schaer R, Eggel I, Müller H, Depeursinge A (2014) Using MapReduce for large-scale medical image analysis, Healthcare Informatics, Imaging and Systems Biology (HISB) IEEE Second International Conf. La Jolla, CA, USA, 1(10): 27-28.

43. Christian W, Datta A (2012) Multi-term Keyword Search in NoSQL Systems. IEEE Internet Computing, 16(1): 34-43.

44. Xu J, Shi M, Chen C, Zhang Z, Fu J, et al. (2016) ZQL: A unified middleware bridging both relational and NoSQL databases. 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, pp. 730-737.

45. Wang Y, Goh W, Wong L, Montana G (2013) Random forests on Hadoop for genome-wide association studies of multivariate neuroimaging phenotypes. BMC Bioinformatics 14(16): 1-15.

46. Wang S, Pandis I, Wu C, He S, Johnson D, et al. (2014) High dimensional biological data retrieval optimization with NoSQL technology. BMC Genomics 15(8): S3.

47. Nelson R, Staggers N (2014) Health Informatics: an inter professional approach. Mosby, an imprint of Elsevier Inc. Saint Louis.

48. Chawla NV, Davis DA (2013) Bringing Big Data to Personalized Healthcare: A Patient-Centered Framework. J Gen Intern Med 28(3): S660-S665.

49. Kuo MH, Kushniruk A, Borycki E (2011) A Comparison of National Health Data Interoperability Approaches in Taiwan, Denmark and Canada. Electronic Healthcare 10(2): 14-25.

50. Srirama SN, Jakovits P, Vainikko E (2012) Adapting scientific computing problems to clouds using MapReduce. Future Generation Computer Systems 28(1): 184-192.

51. Das S, Sismanis Y, Beyer KS (2010) Ricardo: Integrating R and Hadoop, Proceedings of the 2010 ACM SIGMOD/PODS Conference (SIGMOD'10) Pp. 987-998.

52. Dufresne Y, Jeram S, Pelletier A (2014) The True North Strong and Free Healthcare? Nationalism and Attitudes Towards Private Healthcare Options in Canada. Canadian Journal of Political Science 47(3): 569-595.

53. Kuo MH, Saham, T, Kushniruk AW, Borycki EM, Grunwell D (2014) Health Big Data Analytics: Current Perspectives, Challenges and Potential Solutions. International Journal of Big Data Intelligence 1(4): 114-126.

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666

DOI: 10.19080/BBOAJ.2018.05.555666

How to cite this article: Dillon C, Hamid Z, Belaid M, Alex K. Review of Designing a Big Data Platform in Healthcare. Biostat Biometrics Open Acc J. 2018; 5(4): 555666. DOI: 10.19080/BBOAJ.2018.05.555666