

Data Warehouse Design for Big Data in Academia

Alex Rudniy*

Department of Computing Sciences, University of Scranton, Scranton, 18510, PA, USA

*Corresponding Author: Alex Rudniy. Email: rudniy@cs.scranton.edu

Received: 08 January 2021; Accepted: 06 September 2021

Abstract: This paper describes the process of design and construction of a data warehouse (“DW”) for an online learning platform using three prominent technologies, Microsoft SQL Server, MongoDB and Apache Hive. The three systems are evaluated for *corpus* construction and descriptive analytics. The case also demonstrates the value of evidence-centered design principles for data warehouse design that is sustainable enough to adapt to the demands of handling big data in a variety of contexts. Additionally, the paper addresses maintainability-performance tradeoff, storage considerations and accessibility of big data corpora. In this NSF-sponsored work, the data were processed, transformed, and stored in the three versions of a data warehouse in search for a better performing and more suitable platform. The data warehouse engines—a relational database, a No-SQL database, and a big data technology for parallel computations—were subjected to principled analysis. Design, construction and evaluation of a data warehouse were scrutinized to find improved ways of storing, organizing and extracting information. The work also examines building corpora, performing ad-hoc extractions, and ensuring confidentiality. It was found that Apache Hive demonstrated the best processing time followed by SQL Server and MongoDB. In the aspect of analytical queries, the SQL Server was a top performer followed by MongoDB and Hive. This paper also discusses a novel process for render students anonymity complying with Family Educational Rights and Privacy Act regulations. Five phases for DW design are recommended: 1) Establishing goals at the outset based on Evidence-Centered Design principles; 2) Recognizing the unique demands of student data and use; 3) Adopting a model that integrates cost with technical considerations; 4) Designing a comparative database and 5) Planning for a DW design that is sustainable. Recommendations for future research include attempting DW design in contexts involving larger data sets, more refined operations, and ensuring attention is paid to sustainability of operations.

Keywords: Big data; data warehouse; MongoDB; Apache hive; SQL server



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Students and educators are rapidly taking coursework from the classroom and paper to online settings where they can engage in peer-review exchanges and interactive e-learning. The MyReviewers platform was implemented at USF in 2009 [1] and by the fall 2015 approximately 500 instructors had used it providing 115,118 reviews of documents. In turn, students conducted 118,919 peer reviews and posted 5,962 revision plans. As the online platform expanded its number of users, it outgrew its original design, demanding for a better performing and more suitable corpora storage system.

Initial data warehouse (“DW”) structure did not satisfy growing needs of data consumption and processing with the addition of other participating universities and multiple dynamic rubrics. Finding improved ways for information retrieval and storage as well as providing insight into the process of DW design are the subject of this work [2]. This information can be applied to other learning systems and facilitate the selection of optimal DW platforms for a given educational context.

The paper is organized as follows: following the Introduction, Background focuses on the importance of the current study within the context of the literature on data warehouse design. The Methods section describes how this case was studied, including the technology used, its application and evaluation. The Results section describes the object of study, the challenges, constraints, and the solutions developed over the course of the research. It also addresses data de-identification and cyber security issues. Finally, the Conclusions section considers the findings and obstacles in the context of an online learning system as well as in the broader context of data warehouse design.

2 Background

Recent technological advances make digital storage and electronic processing facilities capable of accommodating big data corpora that remains accessible to a wide audience. Only a decade ago, a common approach was to build and maintain computational hardware on premises or outsource this tedious and time-consuming process to a third party for a fee. Currently, the growth of cloud technologies has changed the paradigm for computing. Cloud computing is a standardized IT capability (services, software, or infrastructure) delivered via Internet technologies in a pay-per-use, self-service way [3].

Cloud computing allows designers in multiple geographical regions in a few minutes to setup and start using a scalable virtual machine. A virtual machine (“VM”) is as an entire machine virtualization defined as a software computer that, as in the case with a physical computer, runs an operating system and applications and has virtual devices providing the same functionality as physical hardware [4].

Large public cloud providers—such as Amazon Web Services, Google Cloud Platform, Microsoft Azure and others—offer a number of pre-built virtual machines varying in processing power, operating systems, and preinstalled software with an option to design a custom virtual machine.

2.1 Literature on Data Warehouse Design

This work addresses the need for a data warehouse for efficient use in peer-review, e-learning setting. The literature that informs this work consists of case studies of data warehouse design now used to support industries ranging from accounting, e-commerce, education, insurance,

healthcare, transportation and others [5]. A number of other studies in education and related fields stress the importance of tasks completed in this work, such as collecting texts, constructing corpora with additional attributes extracted or assigned to each document, and storing data for subsequent analysis and processing [6].

Several publications compare the performance of data warehousing platforms such as MongoDB and SQL Server [7], or MongoDB and HBase [8–10]. Unlike those, this work brings three DW platforms for a more comprehensive research.

2.2 Platforms and Technologies

Recent studies that seek to improve data warehouse efficiency explore the use of various platforms and technologies, including the three used in our research: Apache Hive, MongoDB and Microsoft SQL Server. These three platforms were used to populate instances of the data warehouse and perform benchmarking: first, Microsoft SQL Server 2016 supporting in-memory processing and in-database analytics; second, MongoDB, a NoSQL document-oriented database; and third, Apache Hive, a big data warehouse with its SQL-like domain specific language, which is translated into sequence(s) of MapReduce jobs for a parallel execution on an Apache Hadoop cluster. These frameworks were evaluated for the tasks of constructing a *corpus* and producing descriptive statistics using the same hardware configurations.

Apache Hive facilitates reading, writing, and managing large datasets residing in distributed storage and supports SQL queries [11,12]. Hive data warehouse platform, commonly used in conjunction with Apache Scoop for data ingestion [13], works on top of the Hadoop big data framework for the distributed processing of large data sets across clusters of computers [12,14].

HiveQL, a dialect of SQL used for data processing, helps users transition more easily into the use of big data technologies. Queries written in HiveQL are translated into MapReduce and executed using Hadoop, which by design lacks the expressiveness of SQL [12]. Record Columnar File format was introduced in Hive for faster execution of MapReduce [15].

Like Hive, MongoDB is recognized as a promising new technology for data warehousing [16]. MongoDB is a NoSQL document-oriented database program, which is distributed free of charge with open source code. It supports high availability, horizontal scaling, geographic distribution, *ad hoc* queries, indexing and real-time aggregation [17]. NoSQL (not only a structured query language) is a term used to describe high-performance, non-relational databases. NoSQL databases utilize a variety of data models, including document, graph, key-value, and columnar [18]. An extended discussion of NoSQL use for implementing data warehouses is presented by Yanguia et al. [19].

This paper assesses the value of these two as well as the SQL Server, a relational database management system featuring high scalability, performance, and availability. The latest versions of SQL Server include capabilities of in-memory and non-relational databases as well as built-in analytics and machine learning tools [16].

3 Methods

University students and instructors in science, technology, engineering and mathematics known as STEM, as well as in other majors benefit from using online learning management systems (“LMS”) to facilitate classroom assignments through the exchange of peer reviews as well as teacher feedback and instruction. This work focuses on design of a data warehouse for MyReviewers LMS, which focuses on peer-reviews and team-projects; assessment of student

learning at the class, student, and program level; and identification of students at risk. Its features include document markup, peer review, e-portfolios, team assignments, pre-built comments, learning analytics, student and instructor surveys, and other teaching tools. Due to the large amounts and complexity of accumulated data, a need for effective data storage, processing, and extraction arose. This requirement was addressed by a call to design a data warehouse, as a repository for historical, integrated and consistent data, commonly equipped with tools for data extraction [20].

3.1 Data Warehouse Design

Several major requirements were imposed on the DW design: hold, manipulate, and extract datasets and corpora from operational databases holding data for at least three years; and store student writings, reviews, and surveys accumulated by the LMS. The data included rubric scores, reviewers' comments, course and project information, survey responses, and more. In general, such DW would also be useful for other LMS and academic electronic systems to effectively organize, store, manipulate and subset accumulated data.

The researchers established five major research directions:

- (1) to demonstrate ways the assessment community can use big data, real-time assessment tools to create valid measures of writing development and knowledge adaption,
- (2) to provide quantitative evidence regarding the likely effects of particular commenting and scoring patterns on particular cohorts of students,
- (3) to inform faculty regarding the efficacy of particular high impact practices, especially peer review,
- (4) to provide a domain map to help us better understand non-cognitive competencies and student success in the curriculum, and
- (5) to gather the evidence necessary to build interactive assessment loops and algorithms to provide more helpful feedback and assessments.

To accommodate these goals, multiple resources stored in the LMS operational databases had to be extracted, decoded, filtered, merged, and stored in a DW for further on-demand customized extraction. In particular, there was a need to subset data consistently and create corpora in several dimensions, such as by university, by semester, by course, by student group, by project, and by rubric. Additionally, multiple dimensions related to a student essay had to be preserved and extracted in a user-friendly fashion. These dimensions included reviews written by instructors and peer students, corresponding rubric criteria, revision plans, decoded and transformed responses of surveys, and the student writing itself.

To successfully analyze big data generated from digital environments, information must be organized in formats amenable to confidentiality, data mining, and analytics tools. Seven goals were therefore set for the data warehouse as listed below:

- (1) to gather evidence related to foundational measurement categories of fairness, validity, and reliability;
- (2) to protect student confidentiality by multiple means, including the ability to anonymize writing samples;
- (3) to assemble disparate information within a single storage;
- (4) to achieve automated data extraction;
- (5) to process data request in a timeframe acceptable to stakeholders;
- (6) to execute ad-hoc queries in a timely manner;
- (7) to ensure sustainability for newly generated data for semesters to come.

Generally, designing a data warehouse involves extraction of relevant data from enterprise information systems, data transformation and integration, removal of flaws and inconsistencies, placement into a data warehouse, and providing end users with the capability to carry out analyses and forecasting studies [20].

Making the DW output comprehensible by non-technical audience mostly comprised by writing program administrators and their colleagues was among the main goals of the work. DW output had to render complex data from unstructured student writings into a format that would fit into a table with each line combining scores, comments and other features of reviews and survey responses. The search for a solution led to a spreadsheet format for structured data, shown in Fig. 1, with hyperlinks referencing text documents with student writing. A column with hyperlinks referencing corresponding text documents allowed to interactively open them in a text editor with a click of a mouse.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Click to pop up a text file	Class_De scriptic	term_cod e	org_cod e	class_cod e	student_ user_id	project_ id	project_n ame	draft_i_ draft_id	Draft_Na me	Grader_U ser_id	Grader_R ole	Rubric_Sc ore	Final_Gra de	Rubric	DocID	129.605 Goals of the As (wgt=25,	129.606 Figures and pag (wgt=15,
2	147717.txt	Cell Struct	201603	Dmouth	BIOL12	7342	581	Writing A	1222	Draft	9809	Instructor	0.00		129	147717	0.00	0.00
3	147720.txt	Cell Struct	201603	Dmouth	BIOL12	8358	581	Writing A	1222	Draft	9809	Instructor	0.00		129	147720	0.00	0.00
4	147722.txt	Cell Struct	201603	Dmouth	BIOL12	7927	581	Writing A	1222	Draft	9809	Instructor	0.00		129	147722	0.00	0.00
5	147727.txt	Cell Struct	201603	Dmouth	BIOL12	7225	581	Writing A	1222	Draft	9809	Instructor	0.00		129	147727	0.00	0.00
6	147732.txt	Cell Struct	201603	Dmouth	BIOL12	8368	581	Writing A	1222	Draft	9809	Instructor	0.00		129	147732	0.00	0.00
7	147734.txt	Cell Struct	201603	Dmouth	BIOL12	8620	581	Writing A	1222	Draft	9809	Instructor	0.00		129	147734	0.00	0.00

Figure 1: Partial spreadsheet contents

As Fig. 1 illustrates, the data categories were complex, yet sorting had to be easily achieved. Specified sets of rubric scores, for example, had to be accessed across multiple classes—along with associated information (such as student background and survey responses). Additionally to output in the spreadsheet format, DW produced identical output in plain text.

Finally, it was necessary to choose an appropriate programming language for coding not only extract-transform-load operations from the LMS into the DW but also user interfaces for integration with MyReviewers LMS. This kind of planning would allow the creating of a sustainable system that could be used for information retrieval, at a granular level, across time and circumstance. Putting together all the constraints and challenges, it was decided to use Microsoft Visual Studio with C# for developing data migration tool and user interface, since this was the technology employed by the MyReviewers LMS.

Microsoft Excel spreadsheets were chosen for the output format while plain text was used for storing student writings. Since Excel was known and readily available to project participants, it allowed to avoid additional training.

We used Gartner Report and its Magic Quadrant charts in order to assess the applicability of available technologies. In Gartner Magic Quadrant for Data Management Solutions for Analytics, Hadoop distributions were represented by at least three companies, Hortonworks (classified as a niche player), MapR (a niche player) and Cloudera (a challenger); MongoDB was placed among the niche players in 2017, while Microsoft technology stayed among the leaders [16].

3.2 Microsoft SQL Server

An initial version of the DW was assembled on a server in a University of South Florida network protected and maintained by the university Information Technologies. Later on, a similar environment was reconstructed in a Microsoft Azure cloud on a prefabricated virtual machine with similar hardware. SQL Server has a number of native tools for data exchange, such as whole database backup and restore, Integration Services, a Bulk Copy utility, which were used to migrate DW to Azure and to import de-identified data.

The SQL Server version of DW followed the Star schema (see Fig. 2) introduced by Kimball in 1997 to the broad audience however invented in 1960s along with dimensions and facts [21].

```

[[
  {
    "org_code" : "USF",
    "term_code" : 201601,
    "project_name" : "Project 1",
    "Draft_Name" : "Early",
    "Grader_Role" : "Instructor",
    "Class_Description" : "ENC 1101",
    "N_Writers" : 39,
    "N_Graders" : 4,
    "Avg_Score" : 0.42
  }, {
    "org_code" : "USF",
    "term_code" : 201601,
    "project_name" : "Project 1",
    "Draft_Name" : "Intermediate",
    "Grader_Role" : "Instructor",
    "Class_Description" : "ENC 1101",
    "N_Writers" : 324,
    "N_Graders" : 11,
    "Avg_Score" : 2.72
  }, {
    "org_code" : "USF",
    "term_code" : 201601,
    "project_name" : "Project 1",
    "Draft_Name" : "Final",
    "Grader_Role" : "Instructor",
    "Class_Description" : "ENC 1101",
    "N_Writers" : 315,
    "N_Graders" : 11,
    "Avg_Score" : 2.78
  }
]]

```

Figure 2: Sample analytical query results in JSON format

The DW schema had a single fact table with more than a hundred attributes and rubric identifiers corresponding to N rubrics (Fig. 2). Each rubric table was a dimension according to the Star schema terminology. A single row in the fact table referenced no more than one rubric.

It is widely agreed that relational databases are not effective for processing unstructured or semi-structured data. Therefore, this work evaluated two alternatives: the MongoDB no-SQL document database and Hive data warehouse for efficient reading, writing, and managing large datasets.

3.3 MongoDB

MongoDB was chosen because of its broad set of technical resources, its strong position in the market [16], its advertised ability to handle corpora, and its widespread use among companies

for data processing [22]. MongoDB was installed on an Azure virtual machine with the same configuration as the SQL Server. Its built-in utility was applied for data ingestion and conversion to the BSON data interchange format used by the Mongo data model. The database was configured, secured and verified.

3.4 Apache Hive

Apache Hive is well known for its effective processing of big data (with Apache Scoop for larger workloads), and its efficient data ingestion in CSV format for smaller datasets. Hive, a member of the Apache Hadoop ecosystem, uses distributed data storage and parallel data processing when deployed on a cluster of servers. Hive inherits features of Hadoop, which is a framework for processing big data in distributed mode on low-cost hardware with efficient programming models and the ability to handle data redundancy, detect hardware failure and resolve problems at the application level [14].

Hive was set up on an Azure virtual machine in the same region with the same hardware configuration for evaluation validity. The latest Hortonworks Hadoop suite with the Ambari utility for installation, maintenance, querying and automation and the Scoop utility for Hive data ingestion were employed at this step. To illustrate the difference in data representation between tabular and JSON formats, results of the same analytical query are shown in Tab. 1 for the tabular form used by Hive and SQL Server and in Fig. 3 for the JSON format, which is a text version of the MongoDB BSON.

Table 1: Sample analytical query results in tabular format

Org code	Term code	Class	Project name	Draft name	Grader role	N writers	N graders	Average score
USF	201601	ENC 1101	Project 1	Early	Instructor	39	4	0.42
USF	201601	ENC 1101	Project 1	Intermediate	Instructor	324	11	2.72
USF	201601	ENC 1101	Project 1	Final	Instructor	315	11	2.78

It is possible to see that the tabular view requires same columns for all rows in a single data view while JSON is more flexible. BSON, a binary version of JSON, an open data exchange format, known as a simplified version of the eXtensible Markup Language (XML), is used by MongoDB and other document databases as an internal data storage format [23].

3.5 Funding Constraints

The budget of the project did not accommodate for commercial software licenses or technical support, which led to a search for free software. Microsoft SQL Server Developer (“MSSD”) Edition provides access to all the features of the commercial version with the restriction for use in non-production environments. Due to the budget limitations, MSSD had to be replaced with another DW chassis. Albeit the other two tools considered in the study—MongoDB and Apache Hive—are distributed under a free-of-charge license, their technical support and maintenance subscriptions come with a cost. Therefore, low maintenance operation was another constraint to be considered.

3.6 Technical Constraints

From the technical point of view, several requirements had to be considered when designing the DW: capability of data import from the LMS, data transformation, merging, and efficient

processing and extraction according to envisioned scenarios. On the other hand, the DW software had to be capable of complying with security policies of universities involved in the study.

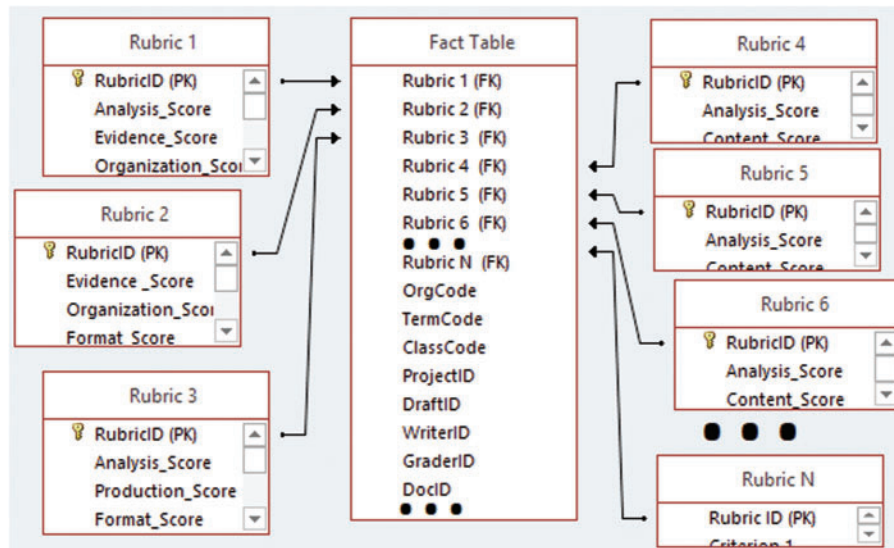


Figure 3: Data warehouse Star schema

4 Results

4.1 Data Sources

As Figs. 1 and 3 demonstrate, information stored in MyReviewers LMS employs a variety of layouts. Reviews, for example, come in several forms, with rubric-oriented feedback being the most informative, where an instructor selects either an existing rubric or creates a new one, and assigns it to a project. Each rubric consists of several criteria, usually with four or five traits. A reviewer, either an instructor or a student, leaves textual feedback and sets a numerical score for each criterion. After a review is complete, overall numerical score is calculated using a formula as defined in a corresponding rubric. Then the numeric score is converted into a letter grade. Additionally, to ensure that feedback is informative, a reviewer is encouraged to use Floating Comments, which can be added by highlighting a line of text or adding a remark in a floating bubble.

From the information technology point of view, MyReviewers LMS is a web-based application, with a frontend written in Microsoft .Net served by the Internet Information Services platform and Microsoft SQL Server databases in the backend. Due to a data separation requirement posed by an institutional policy, two database instances with identical schema but different content coexist: one database for the USF university and another one for other institutions. Two additional databases were employed for essay uploading, display, and storage, including a separate storage for all essay drafts uploaded in the PDF format.

4.2 Data Transformation

The four databases comprising the backend of MyReviewers are optimized for fast, anomaly-free read, write and update operations. These are assured by several well-established techniques,

including database normalization. As a consequence, data is split into multiple interrelated “narrow” tables, with a small number of columns, usually up to a few dozen, and large number of rows, reaching hundreds of millions and beyond.

These narrow tables are passed to an Operational Data Store (“ODS”), where data are de-normalized by joining multiple columns of “narrow” tables into a smaller number of “wide” tables preserving relationships and applying necessary processing. Since MyReviewers backend is built on the SQL Server, it was also used for ODS.

Data tables were exported into flat files preserving the original structure. These datasets had more than 120 columns. Different number of criteria in different rubrics commanded variable number of columns in datasets.

The primary key uniquely identifying each row consisted of term code, school code, class code, project ID, draft ID, writer ID and reviewer ID. Writer and reviewer IDs commonly contained email addresses or institutional IDs, which in some cases were a mix of a last name, letters and numbers. To avoid re-identification of student records, writer and reviewer IDs were substituted with artificial IDs. A separately stored secure lookup table preserved original to artificial ID mapping.

The ODS tables described above were subsequently integrated into the Star schema depicted in Fig. 2 for storage optimization. The process of data extraction from operational databases into ODS, and to the Star schema was automated via a number of SQL queries incorporated into stored procedures.

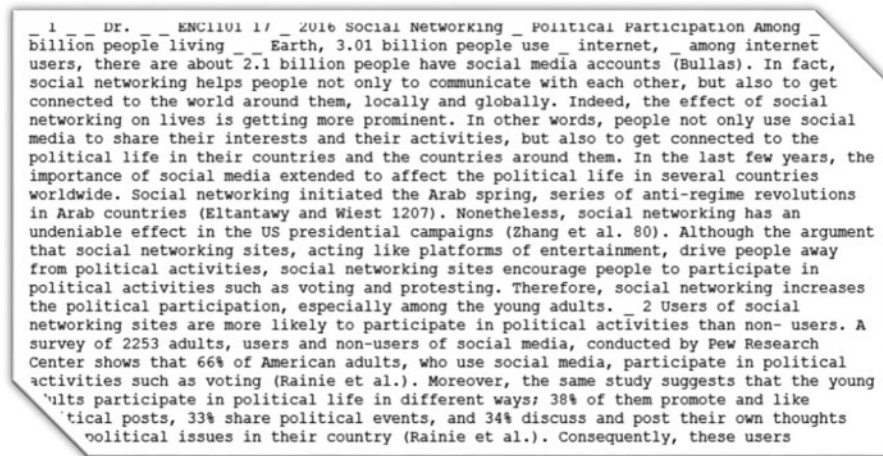
4.3 Data De-Identification and Cybersecurity

Research studies on student essays fall into the category of research on human subjects, which are susceptible to the Family Educational Rights and Privacy Act. The Office for Human Research Protections describes unanticipated problems, which may arise during the course of research in the “Guidance on Reviewing and Reporting Unanticipated Problems Involving Risks to Subjects or Others and Adverse Events.” In many cases, Institutional Review Board (“IRB”) approval and additional ethics training are required before research can begin. The Common Rule states that when reviewing research proposals, IRBs must determine if investigators have made adequate provisions for protecting the privacy of subjects and maintaining the confidentiality of the data [24,25].

Fig. 4 shows a sample de-identified text. Underscore symbols were used to replace personal information. In certain cases, false positives—words that match by accident—were wiped out. In order to avoid the loss of confidentiality, a red flag for IRBs, researchers should conduct data de-identification. This task requires thorough processing since ineffectively anonymized data can be subsequently re-identified using indirect markers and identifiers.

In Fig. 4, writers’ and reviewers’ names were retrieved from a user table and removed from student writings using a straightforward brute-force search. Another more advanced way to de-identify texts was designed using named entity recognition methods, which were commonly used in the natural language processing. Both approaches will be described in detail in a separate publication.

Security features for the de-identified data included regular installation of recent operating system updates, employment of an anti-malware program, use of storage drive encryption; and the use of private networks and tunneling for encrypting connections to the data server [26–30].



1 DR. ENCL101 1 / 2016 Social Networking Political Participation Among
 billion people living Earth, 3.01 billion people use internet, among internet
 users, there are about 2.1 billion people have social media accounts (Bullas). In fact,
 social networking helps people not only to communicate with each other, but also to get
 connected to the world around them, locally and globally. Indeed, the effect of social
 networking on lives is getting more prominent. In other words, people not only use social
 media to share their interests and their activities, but also to get connected to the
 political life in their countries and the countries around them. In the last few years, the
 importance of social media extended to affect the political life in several countries
 worldwide. Social networking initiated the Arab spring, series of anti-regime revolutions
 in Arab countries (Eltantawy and Wiest 1207). Nonetheless, social networking has an
 undeniable effect in the US presidential campaigns (Zhang et al. 80). Although the argument
 that social networking sites, acting like platforms of entertainment, drive people away
 from political activities, social networking sites encourage people to participate in
 political activities such as voting and protesting. Therefore, social networking increases
 the political participation, especially among the young adults. 2 Users of social
 networking sites are more likely to participate in political activities than non- users. A
 survey of 2253 adults, users and non-users of social media, conducted by Pew Research
 Center shows that 66% of American adults, who use social media, participate in political
 activities such as voting (Rainie et al.). Moreover, the same study suggests that the young
 adults participate in political life in different ways; 38% of them promote and like
 political posts, 33% share political events, and 34% discuss and post their own thoughts
 political issues in their country (Rainie et al.). Consequently, these users

Figure 4: Sample de-identified text

4.4 Challenges and Solutions

Our research encountered four major challenges that are commonly related to working with big data (known as “the four Vs” of big data [31]): (1) lack of documentation caused Veracity, (2) database table size limitation is related to Volume, (3) high latency of in-database processing of unstructured textual data falls into Velocity and (4) data complexity is attributed to Variety.

The first obstacle was related to a lack of detailed documentation of the existing databases. MyReviewers program was rapidly developed by a team of skilled programmers who did not always have time to scrupulously document all the changes in the database since it was not their focus. Thus, it took numerous meetings and clarifications to fetch all the entities and their relationships needed for DW data loading process commonly referred to as Extract, Transform and Load.

Second, the initial goal for merging rubric score data into a single sparse table and all rubric comments into another sparse table faced an unexpected limitation of the SQL Server related to big data. The newly constructed tables had multiple columns per single row. However, writing and STEM projects accumulated in MyReviewers between 2016 and 2017 utilized more than 5,000 criteria total, which exceeded the limit of 1,024 columns for nonwide tables. An apparent solution was to switch to a wide table with its maximum of 30,000 columns. However, this potential solution faced a limit in maximum row size of 8,060 bytes [32].

In order to accommodate more columns and rows than allowed by SQL Server, the “divide and conquer” principle was employed. The ODS was switched to multiple tables with smaller number of columns where each table reported one rubric with data for a single semester at a single university.

A third challenge was related to processing students’ writings stored within a column of a database table. An initial version of the DW design suggested storing student texts as a data type allowing up to 2 gigabytes, which exceeded lengths of the longest texts. After loading, student writings had to be de-identified by removing names, email addresses and other personal information. Surprisingly, this process took too much time—from one to two weeks per university. This timing was not acceptable since multiple runs had to be made for evaluation and correction purposes.

Multi-threading and other programming approaches failed to speed up the processing. Database input/output, therefore, was determined to be the bottleneck. The problem was resolved by placing writings out of the DW as external text files in a hard drive while preserving paths within the DW and employing parallel processing with multi-threading. Later, deidentified texts were programmatically placed back into DW tables.

A fourth challenge was caused by the complexity of the data. In addition to student writings, the dataset included related scores and comments, grader, project and class information, and responses to four different surveys. The surveys' flow depended on students' answers. For example, if a student answered No to question 1, the survey would omit questions 1a and 1b and go directly to question 2. The complexity of data and applicable scenarios resulted in several versions of data dictionary, shown in [Tab. 2](#). The final version of the data dictionary included graphical illustrations and extended descriptions simplifying user understanding of the complex information.

Table 2: Excerpt from a data dictionary

Field name	Description
Clickable file name	Sample: 147717.txt
Class description	Sample: cell structure and function
Term code	Sample: 201603
Org code	Sample: Dmouth
Class code	Sample: BIOL12
Student user ID	Artificial ID of a student. Sample: 7342
Project ID	A unique number assigned to a writing project. Sample: 581
Project name	Sample: writing assignment 1
Draft ID	A unique number assigned to a draft type. For example, an initial draft type for writing assignment 1 has draft ID 1222; while a revision draft for the same assignment had draft ID 1227.

[Tab. 2](#) shows nine rows extracted from the DW data dictionary describing in detail the format of data extraction. The Field Name lists all the columns in data extracts, whereas the Description is filled with sample values and more detailed explanations. The data dictionary was created to facilitate users' work with data and explain the meaning of variables included.

Another technical challenge faced while working with MongoDB was the lack of experience with NoSQL technology, which was common to the field of big data analytics. This limitation was solved by employing an SQL to NoSQL translation tool [\[33\]](#).

Following a common practice for evaluating DW solutions, this study measured performance for conducting regular task of *corpus* construction and for producing descriptive analytics [\[19\]](#). It also measured actual execution time in order to account for flaws due to varying environments, varying operating system and network loads, or other interferences. To mitigate these risks, the experiment was repeated ten times at different time of day, and the minimum execution time was recorded for each DW version as depicted in [Fig. 5](#).

The charts depicted in [Fig. 5](#) demonstrate that Hive had the most inconsistent performance, with the shortest time for *corpus* retrieval but the longest time for the analytical query. This is explained by the overhead required to translate Hive queries into Hadoop map-reduce jobs

performing lower level data processing, while a straightforward data retrieval commonly avoids such additional work. It should also be noted that Hive was developed to perform batch processing rather than real-time computations. However, the growth of the DW should make up for this overhead. Consistently showing the second-best time was MongoDB, while SQL server produced the best numbers for the analytical query.

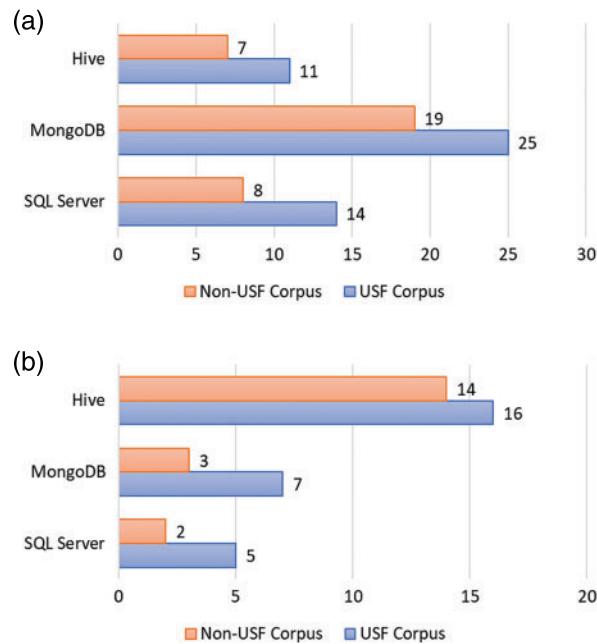


Figure 5: Execution time (a) *Corpus* query execution time, minutes (b) Analytical query execution time, seconds

5 Conclusions

This work demonstrated a novel way to construct a data warehouse providing research environments with efficient means for *corpus* and dataset design and management. During the course of the project, three versions of the data warehouse design were built and evaluated.

Our first Research Question asks which DW technology—SQL Server, MongoDB or Hive—performed best to store, process, query, subset, extract and analyze data for users. We found that the SQL Server was the most suitable tool for the MyReviewers DW. A limitation, however, was that its license cost and unsatisfactory large text processing formed presented an insurmountable barrier. Instead, we employed Hive as an SQL-friendly distributed DW, which produced good results throughout the study.

Our evaluations showed mixed results for Research Question 2 (Fig. 5), “Which framework requires less maintenance and performs faster?” Hive’s scalability, use of commodity hardware and the simplified programming model weighted the decision in its favor.

Funding Statement: This work was sponsored by NSF grant # 1544135 “Collaborative Research: The Role of Instructor and Peer Feedback in Improving the Cognitive, Interpersonal, and

Intrapersonal Competencies of Student Writers in STEM Courses”, PI Rudniy. NSF URL https://www.nsf.gov/awardsearch/showAward?AWD_ID=1544135.

Conflicts of Interest: The author declares that he has no conflicts of interest to report regarding the present study.

References

- [1] D. Raths, “Innovator awards. Bringing peer review tech to classroom,” 2016. [Online]. Available: <https://campustechnology.com/articles/2016/07/13/bringing-peer-review-tech-to-the-classroom.aspx>.
- [2] J. Moxley and D. Eubanks, “On keeping score: instructors’ vs. students’ rubric ratings of 46,689 essays,” *Journal of the Council of Writing Program Administrators*, vol. 39, no. 2, pp. 53–80, 2016.
- [3] M. Hamdaqa and T. Ladan Tahvildari, “Cloud computing uncovered: A research landscape,” *Advances in Computers*, vol. 86, pp. 41–85, 2012.
- [4] VMware vSphere 5.1 Documentation Center, *VMware vSphere Documentation*, 2021. [Online]. Available: https://pubs.vmware.com/vsphere-51/topic/com.vmware.vsphere.vm_admin.doc/GUID-CEFF6D89-8C19-4143-8C26-4B6D6734D2CB.html.
- [5] R. Kimball and M. Rodd, *The Data Warehouse Toolkit. The Complete Guide to Dimensional Modelling*, 2nd ed., New York, NY: Wiley Computer Publishing, 2002.
- [6] R. Boettger and S. Wulff, “Using authentic language data to teach discipline-specific writing patterns to STEM students,” in *Proc. of IEEE ProComm Int. Professional Communication Conf.*, Austin, TX, 2016.
- [7] P. Raj, A. Raman, D. Nagaraj and S. Duggirala, *High-Performance Integrated Systems, Databases, and Warehouses for Big and Fast Data Analytics, in High-Performance Big-Data Analytics*. Switzerland: Springer International Publishing, pp. 25–66, 2015.
- [8] M. Chavalier, M. El Malki, A. Kopliku, O. Teste and R. Tournier, “Document-oriented data warehouses: Models and extended cuboids, extended cuboids in oriented document,” in *Proc. of IEEE 10th Int. Conf. on Research Challenges in Information Science, Grenoble*, pp. 1–11, 2016.
- [9] M. Chevalier, M. El Malki, A. Kopliku, O. Teste and R. Tournier, “Benchmark for OLAP on NoSQL technologies comparing NoSQL multidimensional data warehousing solutions,” in *Proc. of IEEE 9th Int. Conf. on Research Challenges in Information Science, Athens*, pp. 480–485, 2015.
- [10] P. O’Neil, E. O’Neil, X. Chen and S. Revilak, “The Star schema benchmark and augmented fact table indexing,” in *Performance Evaluation and Benchmarking*, R. Nambiar, M. Poess (Eds.), vol. 5895. Berlin: Springer, 2009.
- [11] Apache Hive, 2020. [Online]. Available: <https://cwiki.apache.org/confluence/display/Hive/Home>.
- [12] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka *et al.*, “Hive—A petabyte scale data warehouse using Hadoop,” in *Proc. of IEEE 26th Int. Conf. on Data Engineering*, Long Beach, CA, pp. 996–1005, 2010.
- [13] A. Bondarev, D. Zakirov and D. Zakirov, “Data warehouse on Hadoop platform for decision support systems in education,” in *Proc. of Twelve Int. Conf. on Electronics Computer and Computation (ICECCO)*, Almaty, pp. 1–4, 2015.
- [14] Apache Hadoop, [Online]. Available: <http://hadoop.apache.org>.
- [15] Y. He, R. Lee, Y. Huai, Z. Shao, N. Jain *et al.*, “RCFile: A fast and space-efficient data placement structure in MapReduce-based warehouse systems,” in *Proc. of IEEE 27th Int. Conf. on Data Engineering*, Hannover, pp. 1199–1208, 2011.
- [16] R. Edjlali, A. Ronthal, R. Greenwald, M. Beyer and D. Feinberg, Gartner magic quadrant for data management solutions for analytics, 2017. [Online]. Available: <https://www.gartner.com/en/documents/3614317/magic-quadrant-for-data-management-solutions-for-analyti>.
- [17] MongoDB. What is MongoDB, 2021. [Online]. Available: <https://www.mongodb.com/what-is-mongodb>.
- [18] AWS. What is NoSQL, 2021. [Online]. Available: <https://aws.amazon.com/nosql>.

- [19] R. Yanguia, A. Nablib and F. Gargouria, “Automatic transformation of data warehouse schema to NoSQL data base: Comparative study,” *Proceedings of 20th Int. Conf. on Knowledge-Based and Intelligent Information & Engineering Systems*, vol. 96, pp. 255–264, 2016.
- [20] M. Golfarelli and S. Rizzi, *Modern Principles and Methodologies*. New Delhi: Tata McGraw Hill Education, 2009.
- [21] L. Yessad and A. Labiod, “Comparative study of data warehouses modeling approaches: Inmon, kimball and data vault,” in *Proc. of Int. Conf. on System Reliability and Science*, Paris, France, pp. 95–99, 2016.
- [22] MongoDB. Customer success stories, 2021. [Online]. Available: <https://www.mongodb.com/who-uses-mongodb>.
- [23] MongoDB. JSON and BSON, 2021. [Online]. Available: <https://www.mongodb.com/json-and-bson>.
- [24] J. Menikoff, J. Kaneshiro and I. Pritchard, “The common rule updated,” *New England Journal of Medicine*, vol. 82, pp. 613–615, 2017.
- [25] Office for Human Research Protections. Federal Policy for the Protection of Human Subjects (‘Common Rule’), 2016. [Online]. Available: <https://www.hhs.gov/ohrp/regulations-and-policy/regulations/common-rule/index.html>.
- [26] Open VPN. Move your network to the cloud, 2021. [Online]. Available: <https://openvpn.net>.
- [27] SSH.COM. SSH Protocol—Secure Remote Login and File Transfer, 2021. [Online]. Available: <https://www.ssh.com/ssh/protocol/>.
- [28] FileZilla, The Free FTP Solution, 2021. [Online]. Available: <https://filezilla-project.org>.
- [29] WinSCP, Free SFTP Client for Windows, 2021. [Online]. Available: https://winscp.net/eng/docs/free_sftp_client_for_windows.
- [30] Cygwin. This is the home of the Cygwin project, 2021. [Online]. Available: <http://www.cygwin.com>.
- [31] IBM Big Data And Analytics Hub. Four Vs of Big Data, 2015. [Online]. Available: <https://cloud-computing-today.com/2015/09/25/1073736/>.
- [32] Microsoft. Maximum Capacity Specifications for SQL Server, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server/maximum-capacity-specifications-for-sql-server>.
- [33] Query Translator. Convert MySQL queries to MongoDB syntax, 2021. [Online]. Available: <http://www.querymongo.com>.