Introduction to BI Architecture

Final Exam Paper

Cloud and Big Data Technologies Recommendations

Student Name: Carlos Timoteo

Student id: 040905599

2018-03-03

Table of Contents

[Introduction 3](#_Toc507844986)

[Motivation 3](#_Toc507844987)

[Objective 4](#_Toc507844988)

[Theoretical Reference 4](#_Toc507844989)

[Massively Parallel Processing (MPP) 4](#_Toc507844990)

[Shared-nothing architecture 4](#_Toc507844991)

[Apache Spark 5](#_Toc507844992)

[Google Cloud Dataflow 6](#_Toc507844993)

[Greenplum 6](#_Toc507844994)

[Hadoop 7](#_Toc507844995)

[MapReduce 7](#_Toc507844996)

[Hadoop Distributed File System (HDFS) 8](#_Toc507844997)

[Pig 9](#_Toc507844998)

[Hive 10](#_Toc507844999)

[HBase 10](#_Toc507845000)

[Mahout 11](#_Toc507845001)

[Hadoop With Query (HAWQ) 11](#_Toc507845002)

[Shark 12](#_Toc507845003)

[Impala 13](#_Toc507845004)

[Presto 13](#_Toc507845005)

[Drill 14](#_Toc507845006)

[Newer database features (improved since SQL Server 2008R2) 14](#_Toc507845007)

[Graph Database 14](#_Toc507845008)

[Not only Structured Query Language (NoSQL) 15](#_Toc507845009)

[Cloud Services 16](#_Toc507845010)

[Amazon AWS 16](#_Toc507845011)

[Microsoft Azure 17](#_Toc507845012)

[Hortonworks 18](#_Toc507845013)

[Databricks 18](#_Toc507845014)

[Recommendations 19](#_Toc507845015)

[Conclusion 20](#_Toc507845016)

[References 20](#_Toc507845017)

# Introduction

Big Data, Analytics and Cloud computing are topics firmly embedded in business dialog. The amount of data being generated today is astonish. Besides that, data has also evolved in type, volume, velocity with the digitisation of business process. Data enables future strategies and immediate change, thanks to the power of predictive and prescriptive analytics [1].

By using Big Data technologies, organizations can ask questions in seconds rather than days. It also enables companies to establish new relations with its clients via social networks, where feedbacks are rapidly assessed.

Artificial Intelligence (AI) meets Big Data technologies by providing revolutionary business. The machine is the new reader, machine learning algorithms can scrap huge amount of data from the internet and process it in a question of minutes. For example, IBM Watson is able to provide you answers for right questions instead of using Google to search for results.

These technologies require heavy computing power, storage, security and IT infrastructure. The Cloud computing is key to provide all these services based on a company’s individual requirements. The main benefits of Cloud computing is to enable flexibility, scalability, cost-effectiveness and rapid-prototyping.

## Motivation

The motivation for this paper is that a government agency needs new technologies to be integrated into an existing BI architecture to help manage the growing data collection and demands for data analysis.

## Objective

The objective of this paper is to explain some of the new technologies, how they can be integrated into agency existing architecture and how the agency may be able to leverage them.The technologies to be covered are Massive Parallel (MPP), Hadoop, modern database features and cloud services.

# Theoretical Reference

## Massively Parallel Processing (MPP)

### Shared-nothing architecture

In Shared-Nothing environment each system has its own private memory and one or more disks. The clustered processors communicate by passing messages through a network that interconnects the computers. In addition, requests from clients are automatically routed to the system that owns the resource. Only one of the clustered systems can “own” and access a particular resource at a time. Of course, in the event of a failure, resource ownership may be dynamically transferred to another system in the cluster. Shared-nothing databases use the local disk, while shared-disk databases rely on storage that is shared and accessible via the network. Because of this, shared-nothing databases has an inherent performance advantage.

The main advantage of shared-nothing clustering is scalability. In theory, a shared-nothing multiprocessor can scale up to thousands of processors because they do not interfere with one another, nothing is shared. For this reason shared-nothing is generally preferable to other forms of clustering. Furthermore, the scalability of shared-nothing clustering makes it ideal for read intensive analytical processing typical of data warehouses.

Because it lacks dynamic load balancing, shared-nothing requires that each server accommodate the peak load for the data it owns. This causes shared-nothing implementations to invest more in servers using either a scale-up or a scale-out approach. Because of latency issues between geographically distributed databases, you should partition data based upon geo-location to minimize or eliminate function- and data-shipping between disparate locations [14].

The key points to choose a shared-nothing architecture are: it can exploit in simpler and cheaper hardware; it works well in a high-volume, read-write environment; it needs fixed load balancing based upon the partitioning scheme; it provides unlimited scalability; data is partitioned across the cluster and since it depends on partitioning, data shipping can kill scalability. Apache Spark, Google Cloud Dataflow and Greenplum implement shared-nothing architecture.

## Apache Spark

Spark is a powerful framework for parallel computations which are mapped into multiple servers with no need of dealing with low level operations such as scheduling, data partitioning, communication and recovery. A simple paradigm of parallel computation supported by Spark is “Map and Reduce” which has been made popular by Google.

In the framework a set of keywords is mapped into a number of workers (e.g. parallel servers available for computation) and the results are then reduced (e.g. collected) by applying a “reduce” operator. Spark can however support additional forms of parallel computation by taking inspiration on Microsoft’s Cosmos. [3]

Spark has a dedicated API for Python called PySpark. PySpark APIs reflect the Scala APIs in which Spark is developed but the former typically take a Python function and return a Python collection. Python also supports an interactive shell “./bin/pyspark” for performing data science using the command line. Spark compilers allocate the computation on the underlying network of server with no explicit need of dealing with low level details and cluster configuration for data scientists. Spark supports programs written in Python, Java, R, Julia and Scala and its packages and libraries, in addition, it provides its own spark packages and libraries making it a very flexible framework.

## Google Cloud Dataflow

The Dataflow programming model is designed to simplify the mechanics of large-scale data processing. When you program with a Dataflow SDK, you are essentially creating a data processing job to be executed by one of the Cloud Dataflow runner services. This model lets you concentrate on the logical composition of your data processing job, rather than the physical orchestration of parallel processing. When you think about data processing with Dataflow, you can think in terms of Pipelines, PCollections, Transforms, I/O Sources and Sinks. A pipeline encapsulates an entire series of computations that accepts some input data from external sources, transforms that data to provide some useful intelligence, and produces some output data. A PCollection represents a set of data in your pipeline. The Dataflow PCollection classes are specialized container classes that can represent data sets of virtually unlimited size. A transform is a data processing operation, or a step, in your pipeline. A transform takes one or more PCollections as input, performs a processing function that you provide on the elements of that PCollection, and produces an output PCollection. The Dataflow SDKs provide data source and data sink APIs for pipeline I/O. You use the source APIs to read data into your pipeline, and the sink APIs to write output data from your pipeline. [13]

## Greenplum

Greenplum is such a solution built to support the next generation of data warehousing and large-scale analytics processing. It is a massively parallel processing (MPP) database which is based on a shared nothing architecture. Some additional features provided by Greenplum are complex query optimization, parallel data loading, fault tolerance, workload management, administration and monitoring.

The architecture of Greenplum can be divided into several layers. Layer 1 can be considered as an application layer which comprises of applications responsible for loading data into segments, queries which retrieve the results from the segments or applications to check the state of Greenplum. Layer 2 consists of the master segment which contains the status of all the segments and is responsible for distribution data across segments. Layer 3 is the network interface through which Greenplum communicates, called GNet Software Interconnect. Layer 4 contains all the segments which are connected to GNet Software Interconnect. Layer 5 is provided for loading data in parallel from external sources. [14]

## Hadoop

Apache Hadoop implements MapReduce paradigm and coordinate several activities like: MapReduce jobs need to be scheduled based on the workload; Jobs are managed and controlled to ensure that any encountered errors are properly handled and guaranteeing that the job still continues to execute; Input data needs to be spread across the cluster to avoid losing any information; Map and reduce steps needs to be conducted across the distributed system, preferably on the same machines where the data resides. [4]

### MapReduce

MapReduce paradigm provides the means to break a large task into smaller ones, run them in parallel, and consolidate the outputs of the individual tasks into the final output. It involves two steps a Map, which applies a function to a piece of data and Reduce, which consolidates the intermediate outputs from the Map step into one. Each step uses key/value pairs as input and output. Although this concept exists for decades, Google led a well-known work using this concept in its search engine to craw the web.

The advantage of using this paradigm is that it is able to distribute the workload over a cluster of computing nodes and run the task in parallel. Apache Hadoop implements that paradigm, it seems easy to implement but it is not, especially in a distributed system. [4]

### Hadoop Distributed File System (HDFS)

HDFS is a file system that provides the capability to distribute data across a cluster that implements MapReduce. HDFS is not an alternative to common file systems, such as ext4, ntfs, fat32. On contrary, HDFS depends on each disk drive’s file system to manage the data. Usually, a filesystem breaks a file into blocks of 4k, but HDFS breaks into 64MB blocks and stores it across the cluster. Another feature is that whenever possible, HDFS attempts to store the blocks for a file on different machines so that the map step can operate on each block in parallel. Also, by default, HDFS creates three copies of each block across the cluster to provide fault tolerance. Instead, if a machine fails, HDFS replicates an accessible copy of the relevant data blocks to another available machine. By providing, three copies of each block allow Hadoop some flexibility in determining which machine to use for the map step.

HDFS manages the data access by using three java daemons called NameNode, DatNode and Secondary NameNode. The NameNode daemon determines and tracks where the various blocks of a data file are stored. The DataNode daemon manages the data stored on each machine and periodically sends a report about the blocks stored on the DataNodes and sends to the NameNode. The Secondary NameNode provides the capability to perform some of the NameNode tasks, then it is not a demand.

A typical MaReduce program in Java consists of three classes: the driver, the mapper and the reducer. The driver provides details such as input file locations, the provisions for adding the input file to the map task, the names of mapper and reducer classes, and the location of the reduce task output. The mapper provides the logic to be processed on each data block. Next, the key/value pairs are processed by shuffle and sort functionality based on the number of reducers to be executed. The reducer processes the values for each key and emits a key/value pair as defined by the reduce logic. The output is then stored in HDFS like any other file in 64MB blocks replicated three times across the nodes. Additionally, Java files can be written for the combiner or the custom partitioner, if applicable. Then, the Java code is compiled and stored as a Java Archive (JAR) file. This JAR file is executed against the specified HDFS input files.

If necessary, Python, C or Ruby code can be written to be used with Hadoop Streaming API, which allows the user to write and run Hadoop jobs with no direct knowledge of Java. There is a version 2.0 of MapReduce, called Yet Another Resource Negotiator (YARN) that separates the resource management of the cluster from the scheduling and monitoring of jobs running on the cluster. Hadoop’s popularity has spawned proprietary and open source tools to make it easier to use and provide additional functionality and features. [4]

### Pig

Apache Pig project consists of a data flow language, Pig Latin, and an environment to execute the Pig code. The main benefit of using it is to simplify the task of developing and executing a MapReduce job, it is an abstraction layer over Hadoop. Pig instructions are translated into one or more MapReduce jobs. Pig also provides several common data manipulation functions like inner and outer joins between files. [4]

### Hive

Similar to Pig, Apache Hive enables users to process data without explicitly writing MapReduce code. One key difference is that Hive Query Language (HiveQL) resembles SQL rather than scripting language. Hive is used to apply some structure to the unstructured data over HDFS. Additionally, a user may consider using Hive if the user has experience with SQL and the data is already in HDFS. Although Hive’s performance may be better in certain applications than a conventional SQL database, Hive is not intended for real-time querying. Instead, HBase may be a better choice for real-time query. [4]

### HBase

Unlike Pig and Hive, which are intended for batch applications, Apache HBase is capable of providing real-time read and write access to datasets with multiple rows and columns. HBase design is based on Google Bigtable. By the end of 2007, HBase was included as part of a Hadoop distribution. Later in 2010, Facebook began to use HBase for its user messaging infrastructure.

HBase is a data store that is intended to be distributed across a cluster of nodes. HBase is built upon HDFS and achieves its real-time access speeds by sharing the workload over a large number of nodes in a distributed cluster. An HBase table also has a third dimension, called version, to maintain the different values of a row and column intersection over time.

HBase uses a key/value structure to store the contents of an HBase table. Each key consists of the following elements: Row length, Row, Column family length, Column family, Column qualifier, Version, Key type. The row is used as the primary attribute to access the contents of a table. The structure or layout of the row has to be specifically designed based on how the data will be accessed.

A common use case for a data store such as HBase is to store the results from a web crawler. HBase provides some benefits like it is possible to get to a billion rows and millions of columns in a HBase table, the row needs to be defined based on how the data will be accesses, finally, it may be advantages to use the column qualifiers to actually store the data of interest, rather than simply storing it in a cell. A second use case is the storage and search of messages, this implementation provided Facebook the auto-complete capability in the search box and to return the results of the query quickly, with the most recent messages at the top. [4]

### Mahout

Apache Mahout is a powerful, scalable machine-learning library that runs on top of Hadoop. With Hadoop and Mahout, data scientists can write MapReduce jobs that reference a number of predefined algorithms to build machine-learning algorithms easily. Some algorithms exposed by Mahout are Collaborative filtering, Naïve Bayes classifier, Random Forest classifier, K-Means Clustering, Principal Component Analysis, Frequent Pattern Matching, etc.

The next major version, Mahout 1.0, will include Scala as a programming language to users write jobs and users can choose between MapReduce, Spark and H2O.ai to run jobs, resulting in significant performance increase. [5]

### Hadoop With Query (HAWQ)

Apache HAWQ combines exceptional MPP-based analytics performance, robust ANSI SQL compliance, Hadoop ecosystem integration and manageability, and flexible data-store format support all natively in Hadoop.

HAWQ adds SQL’s expressive power to Hadoop to accelerate data analytics projects, simplify development while increasing productivity, expand Hadoop’s capabilities, and cut costs. HAWQ can help render Hadoop queries faster than any Hadoop-based query interface on the market by adding rich, proven, parallel SQL processing facilities. HAWQ leverages existing business intelligence and analytics products and a workforce’s existing SQL skills to bring more than 100 times performance improvement to a wide range of query types and workloads.

HAWQ uses YARN to its integration and management, Ambari for provision capabilities, interfaces with HCatalog, supports Parquet, AVRO, HBase and others file formats. Plus, HAWQ uses Apache MADlib machine learning library to execute advanced analytics for data-driven digital transformation. [6]

### Shark

Shark is an open source Hadoop project that uses the Apache Spark advanced execution engine to accelerate SQL-like queries. Shark makes use of Hive’s language, its metadata, and its interfaces, so like Hive it offers a simple way to apply structure to large amounts of unstructured data, and then perform batch SQL-like queries on that data. In addition to full compatibility with Hive, Shark offers speed, spark integration and scalability. Shark is now integrated into Apache Spark project as a module and renamed to Spark SQL. [7]

Like Spark, Shark allows for data sets to be held in memory. When tables are created in Hive, users can access a simple Shark API to indicate that the table be held in memory. In-memory table queries are up to 100x faster than standard Hive queries. Shark allows Hive tables to be queried from within a Spark job, allowing for it to be combined with the logic exposed by Spark libraries like MLlib. Shark is great for quickly returning results for simple queries. However, sometimes users need to do batch processing, like executing a complex query on a multi-petabyte table. Shark is ideal for these jobs as well, supporting mid-query fault tolerance to ensure the whole job completes as quickly as possible even if nodes fail. [8]

### Impala

Apache Impala is the open source, native analytic database for Apache Hadoop. It provides fast, interactive SQL queries directly on your Apache Hadoop data stored in HDFS, HBase, or the Amazon Simple Storage Service (S3). In addition to using the same unified storage platform, Impala also uses the same metadata, SQL syntax (Hive SQL), ODBC driver, and user interface (Impala query UI in Hue) as Apache Hive. This provides a familiar and unified platform for real-time or batch-oriented queries. Impala does not replace the batch processing frameworks built on MapReduce such as Hive. Hive and other frameworks built on MapReduce are best suited for long running batch jobs, such as those involving batch processing of Extract, Transform, and Load (ETL) type jobs.

The Impala solution is composed of clients, which are entities including Hue, ODBC clients, JDBC clients, and the Impala Shell can all interact with Impala. These interfaces are typically used to issue queries or complete administrative tasks such as connecting to Impala; Hive metastore which stores information about the data available to Impala; Impala process that runs on DataNodes, coordinates and executes queries, these queries are distributed among Impala nodes and these nodes, then, act as workers, executing parallel query fragments; HBase and HDFS storage for data to be queried. [9]

### Presto

Presto is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes. It is not a general-purpose relational database and is not a replacement for databases like MySQL, PostgreSQL or Oracle. In addition, Presto was not designed to handle Online Transaction Processing (OLTP).

Presto allows querying data where it lives, including Hive, Cassandra, relational databases or even proprietary data stores. A single Presto query can combine data from multiple sources, allowing for analytics across your entire organization. When Presto parses a statement, it converts it into a query and creates a distributed query plan which is then realized as a series of interconnected stages running on Presto workers. When you retrieve information about a query in Presto, you receive a snapshot of every component that is involved in producing a result set in response to a statement. [10]

### Drill

Drill is an Apache open-source SQL query engine for Big Data exploration. It is designed from the ground up to support high-performance analysis on the semi-structured and rapidly evolving data coming from modern Big Data applications, while still providing the familiarity and ecosystem of ANSI SQL. Drill provides plug-and-play integration with existing Apache Hive and Apache HBase deployments. The key features of Apache Drill are low-latency SQL queries, dynamic queries on files in different formats and HBase tables, without requiring metadata definitions in the Hive metastore, nested data support, integration with Apache Hive and BI/SQL tool integration using standard JDBC/ODBC drivers. In addition, Drill drivers does not depend on Hadoop or Spark, that way you can install in any environment that you need, even Windows environments. [11]

## Newer database features

### Graph Database

They are intended for use cases such as networks, where there are items and relationships between these items. While it is possible to store graphs such as trees in a relational database, it often becomes cumbersome to navigate, scale, and add new relationships. Graph databases help to overcome these possible obstacles and can be optimized to quickly traverse a graph.

In-database analytics

In-database analytics is the integration of data analytics with data warehousing functionality. It eliminates the movement of data by embedding analytical functionality directly into the database. It is similar to data mining to some extent since the database is mined for required data using descriptive and predictive models for discovering meaningful pattern in the database.

Together with performance and scalability advantages stemming from database platforms with parallelized, shared-nothing Massive Parallel Processing (MPP) architectures, the database-embedded calculations are capable of respond to growing demand for high-throughput, substantial elimination of inaccuracies and exploitation of DBMS parallelization. [16]

SAP HANA is the platform supporting both transactional and analytical workloads, i.e., mixed workload. IBM is having multiple In-database options for its DB2 and Netezza databases. Emerging In-database analytics exploits benefits from Vertica, Teradata, Netezza, Greenplum and Aster Data Systems. Leading In-database analytical software modules includes Python, R, SAS, SPSS Modeler and MADlib.

### Not only Structured Query Language (NoSQL)

NoSQL are data stores that are applied to unstructured data. In general, the power of this data store is that as the size of the data grows, the implemented solution can scale by simply adding additional machines to the distributed system.

Key/value stores contain data that can be simply accessed by a given identifier. As described in the MapReduce discussion, the values can be complex. In a key/value store, there is no stored structure of how to use the data; the client that reads and writes to a key/value store needs to maintain and utilize the logic of how to meaningfully extract the useful elements from the key and the value.

Document stores are useful when the value of the key/value pair is a file and the file itself is self-describing (for example, JSON or XML). The underlying structure of the documents can be used to query and customize the display of the documents’ content. For example, a document store may provide the ability to create indexes to speed the searching of the documents. Document stores may be useful for content management of web pages, web analytics of stored log data.

Column family stores are useful for sparse datasets, records with thousands of columns but only a few columns have entries. The key/value concept still applies, but I this case a key is associated with a collection of columns. In this collection, related columns are grouped into column families. Column family data stores are useful to store and render blog entries, tags, and viewers’ feedback, to store and update various web page metrics and counters.

Some examples of NoSQL databases are Redis, MongoDB, Cassandra, HBase and Neo4j. [4]

## Cloud Services

### Amazon AWS

Amazon Web Services (AWS) provides a broad platform of managed services to help you build, secure, and seamlessly scale end-to-end big data applications quickly and with ease. Accordingly to the analytics needs and demand, you can easily resize your environment (horizontally or vertically) without having to wait for additional hardware or being required to over invest to provision enough capacity.

In addition, you get flexible computing on a global infrastructure with access to the many different geographic regions that AWS offers, along with the ability to use other scalable services that augment to build sophisticated big data applications. These other services include Amazon Simple Storage Service (Amazon S3) to store data and AWS Data Pipeline to orchestrate jobs to move and transform that data easily. AWS IoT, which lets connected devices interact with cloud applications and other connected devices.

AWS has many options to help get data into the cloud, including secure devices like AWS Import/Export Snowball to accelerate petabyte-scale data transfers, Amazon Kinesis Firehose to load streaming data, and scalable private connections through AWS Direct Connect. As mobile continues to rapidly grow in usage, you can use the suite of services within the AWS Mobile Hub to collect and measure app usage and data or export that data to another service for further custom analysis.

The following services are described in order from collecting, processing, storing, and analyzing big data: Amazon Kinesis Streams, AWS Lambda, Amazon Elastic MapReduce, Amazon Machine Learning, Amazon DynamoDB, Amazon Redshift, Amazon Elasticsearch Service, and Amazon QuickSight. In addition, Amazon EC2 instances are available for self-managed big data applications. [17]

### Microsoft Azure

Microsoft launched the Azure Machine Learning cloud platform—Azure ML. Azure ML provides an easy-to-use and powerful set of cloud-based data transformation and machine learning tools. A few of the benefits Azure ML provides for machine learning solutions: solutions can be quickly deployed as web services; models run in a highly scalable cloud environment; code and data are maintained in a secure cloud environment; available algorithms and data transformations are extendable using the R language for solution-specific functionality. Using Machine Learning Studio, data scientists and developers can quickly build, test, and develop predictive models using state-of-the art machine learning algorithms. [18]

### Hortonworks

Hortonworks is an industry leading innovator that creates, distributes and supports enterprise-ready open data platforms and modern data applications that deliver actionable intelligence from all data: data-in-motion and data-at-rest. Hortonworks is focused on driving innovation in open source communities such as Apache Hadoop, NiFi and Spark. Along with its 1800+ partners, Hortonworks provides the expertise, training and services that allow customers to unlock transformational value for their organizations across any line of business.

The Hortonworks Data Platform is a massively scalable and open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, HDFS, HCatalog, Pig, Hive, HBase, Zookeeper and Ambari. [19]

### Databricks

Databricks is a company founded by the creators of Apache Spark, which aims to help clients with cloud-based big data processing using Spark. Databricks grew out of the AMPLab project at University of California, Berkeley that was involved in making Apache Spark, a distributed computing framework built atop Scala. Databricks develops a web-based platform for working with Spark, which provides automated cluster management and IPython-style notebooks.

Databricks Unified Analytics Platform (UAP) is a unified approach to data analytics at scale. It provides a platform that accelerates innovation by unifying data science, engineering, and business. Some benefits of adopting Databricks UAP are accelerate performance by 5x with Databricks Runtime, increase productivity by 4-5x through interactive data science, streamline processes from ETL to production, ensure enterprise security and compliance. [20]

# Recommendations

The Big Data and Cloud technologies presented in this paper can be adopted by the government agency to empower the data analytics and processing of new data sources. The first recommendation is to avoid the premature adoption of Apache Hadoop and MapReduce technologies based on HDFS, because it only makes sense if you are dealing with hundreds of terabytes or petabytes of data and losing any data is sensitive. The motivation for the development of Apache Hadoop is to avoid losing sensitive information when you are dealing with big amount of data and data format. Therefore, instead of using Hadoop, I suggest using Massive Parallel Processing technologies like Apache Spark or Greenplum. In addition, you can use Apache Presto or Drill to integrate multiple data sources and deal with them in a structured way.

Since MPP technologies are recommended or if you still need Hadoop, the second recommendation is to use Analytics as a Service solutions like Hortonworks and Databricks. Since they provide you a completed configured environment and administration capabilities to avoid issues in the Hadoop or Spark environment setup, but those services are offered at a cost that may not be affordable for you. If the budget is a concern, I suggest Amazon AWS services that will provide some preconfigured solutions and infrastructure at an affordable cost.

The last recommendation is to adopt Greenplum, SAP HANA or another In-database analytics solution since there is less environment configuration concerns and most backend developers are familiar with database and data warehouse technologies.

# Conclusion

This paper presents some Big Data and Cloud technologies to be adopted by the government agency to satisfy their needs of new solutions to be integrated into an existing BI architecture to help manage the growing data collection and demands for data analysis. As justified in the previous section, Hadoop should be avoided, MPP should be considered instead, Analytics as a Service solutions are preferable and new database technologies should be considered as well.

# References

[1] <https://www.itproportal.com/features/the-importance-of-big-data-and-analytics-in-the-era-of-digital-transformation/>

[2] <https://www.newgenapps.com/blog/benefits-role-of-cloud-computing-in-digital-transformation>

[3] Gulli, A. (2015). A collection of Data Science Interview Questions Solved in Python and Spark Hands-on Big Data and Machine Learning, Volume I. CreateSpace Independent Publishing Platform.

[4] EMC Education Services (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons.

[5] <https://mapr.com/products/product-overview/apache-mahout/>

[6] <http://hawq.incubator.apache.org/>

[7] <https://spark.apache.org/sql/>

[8] <https://mapr.com/products/product-overview/shark/>

[9] <https://impala.apache.org/impala-docs.html>

[10] <https://prestodb.io/docs/current/>

[11] <https://drill.apache.org/docs/>

[12] Lee, S. (2011). Shared-nothing vs. shared-disk cloud database architecture. International Journal of Energy, Information and Communications, 2(4).

[13] <https://cloud.google.com/dataflow/model/programming-model>

[14] <http://www.emc.com/collateral/hardware/white-papers/h8072-greenplum-database-wp.pdf>

[15] <http://madlib.apache.org/>

[16] Pethuru Raj, Anupama Raman, Dhivya Nagaraj, Siddhartha Duggirala (2015). High-Performance Big-Data Analytics: Computing Systems and Approaches. Springer International Publishing.

[17] <https://aws.amazon.com/pt/big-data/>

[18] Stephen F. Elston (2015). Data Science in the Cloud: with Microsoft Azure Machine Learning and R. O'Reilly Media

[19] <https://hortonworks.com/>

[20] <https://databricks.com/wp-content/uploads/2017/11/DB-UAP-Solution-Sheet-170926.pdf>