***Why Tez Performs faster in compare to MapReduce***

*Sifat Ibtisum*

*M.S in computer science*

*North Dakota State University*

***Abstract***: Big data is a popular topic on cloud computing research. The main characteristics of big data are volume, velocity and variety. However, these characteristics are difficult to handle by using traditional software and methods. Hadoop is open source framework for handling several domains of big data problems. MapReduce is the main engine of Hadoop cluster and widely used nowadays. A batch oriented process is involved in it. Nevertheless, Apache has an alternative engine called ‘Tez’ which supports interactive query and doesn’t store data in HDFS. In this paper, a comparative studies were made based on the performance on MapReduce and Tez. And the result showed that Tez performs lot faster than MapReduce i.e execution time for MapReduce is lot longer than tez.

***Keywords: Hadoop, Tez, MapReduce, HDFS, Compression.***

***Introduction***:

The name Hadoop has become synonymous with big data. Hadoop is an open-source software-framework for distributed storage of large datasets on computer clusters. In easy words, this means that the ability to scale any size of data without the worry of hardware failures. Hadoop provides large amounts of storage for all sorts of data along with the ability to handle virtually limitless concurrent jobs or tasks.

It has the ability to implement on single or multiple cluster. An example of big data analytics cloud be, social media information namely Facebook, twitter, Instagram are used to predict person’s life style and taste. Then based on that information, products are manufactured to meet customer’s requirement.

Hadoop is the answer for all big data solution. Its component includes HBase, Hive, Sqoop, Mahout, Oozie, ZooKeeper. They work on Hadoop distributed file system (HDFS) and MapReduce version 2, which is called ‘YARN’. HDFS is a logical disk over physical directories in each data node of Hadoop cluster. It communicates with TCP protocol port 22 such as secure shell (SSH) on each node in cluster. HDFS disk is a high fault-tolerant with a number of replicas in HDFS configuration. YARN can access and process only on it. The replication number is a direct effect to HDFS storage. Hortonworks and cloudera which is the organization that provides Hadoop platform. They use data compression algorithm with Hadoop platform which can reduce disk storage and bandwidth network between each node on cluster.

YARN is the prerequisite for Hadoop, providing resource management and a central platform to deliver consistent operations, security, and data governance tools across Hadoop clusters. It has two frameworks to process data which are MapReduce and Tez.

MapReduce is a framework that supports batch processing and it is default framework of Hadoop cluster. On the other side, Tez supports interactive processing.

This paper focuses on the response time and extreme throughput at petabyte scale by analyzing data on mapreduce and Tez. Higher-level data processing applications like Hive and Pig need an execution framework that can express their complex query logic in an efficient manner and then execute it with high performance.

Moreover, this paper organizes including Background review and related works, Methodology, Evaluation and conclusion. This session related work and background information (Hadoop and Mapreduce and Tez). Both MapReduce and Tez framework with compression methods are described in this paper. Lastly, this paper describes the flaws and strength of each of the frameworks and suggests which one is better for faster processing and why.

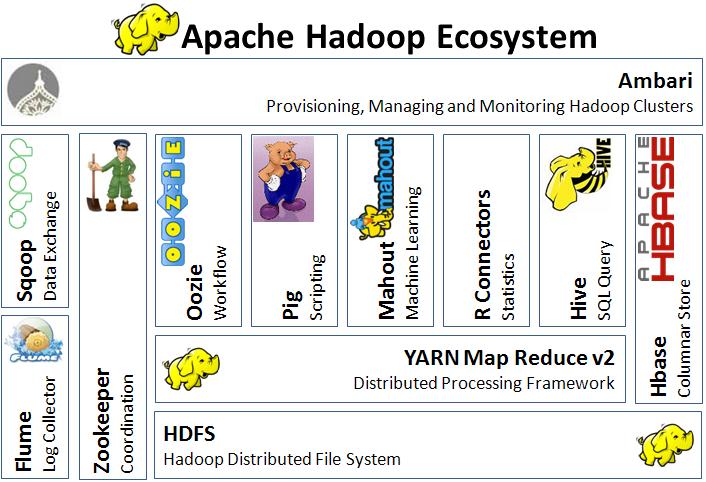
***Related works:***

Presently, big Data is the most popular issue and Hadoop is the core of that. And there are lots of research works related in improving and optimizing the Hadoop cluster. And this paper focuses on performance comparison of those Hadoop frameworks namely MapReduce and Tez with Pig scripts which are pre structure of data. And the result of this research implies that for big data analysis Tez performs better than MapReduce. In the Hadoop compression algorithms, Andre Wenas used GZIP, LZJB and ZLE Compression techniques for data Warehouse and his results show the best performance on ZLE. Yanpei Chen’s research tries to select compress or decompress MapReduce output file for reduced power Consumption. His results shows decrease in energy Consumption more than 50%. Bhavin J. Mathiya use more Compression algorithm like DEFLATE, LZ4, Bzip and Gzip with word-count benchmark both with Map output and reduced output. He presented Bzip2 with higher compression ratio upto 85% than other techniques.

***Concept***:

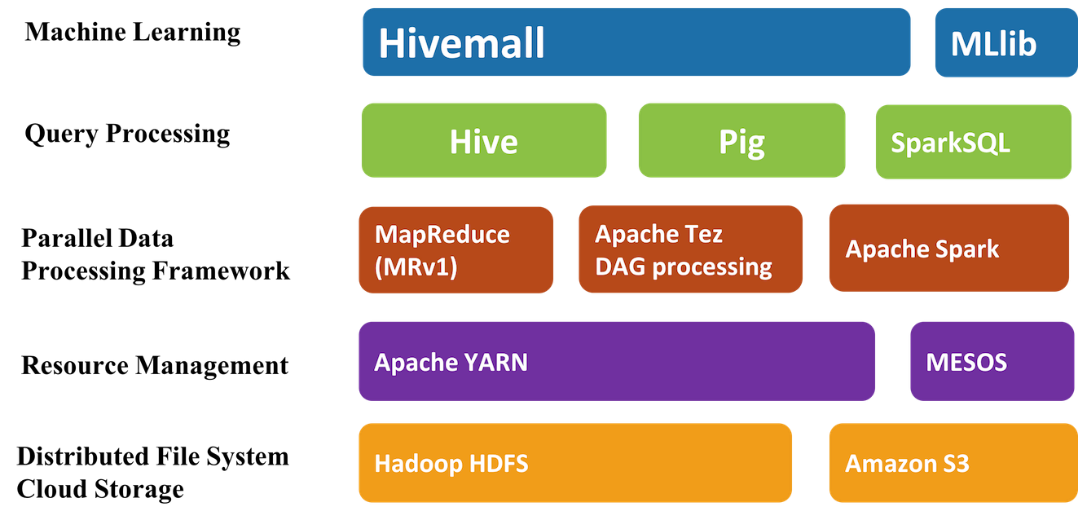
***Hadoop***:

Hadoop was initially introduced by two scientist Doug Cutting and Mike Cafarella in the year 2005. By definition, Apache Hadoop is an open source software platform for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware.  Hadoop services provide for data storage, data processing, data access, data governance, security, and operations. Hadoop offers the privilege of scalability, reliability and flexibility which are pretty unique in comparison to relational database. Like, distributed processing of data local to each node in a cluster enables Hadoop to store, manage, process and analyze data at petabyte scale, large computing clusters are prone to failure of individual nodes in the cluster. Hadoop is fundamentally resilient – when a node fails processing is re-directed to the remaining nodes in the cluster and data is automatically re-replicated in preparation for future node failures and lastly, unlike traditional relational database management systems, you don’t have to create structured schemas before storing data. You can store data in any format, including semi-structured or unstructured formats, and then parse and apply schema to the data when read.



***Hadoop Architecture:***

Let’s, breakdown the architecture and explain in detail about this.



*Fig: Hadoop Architecture*

***Hadoop HDFS:***

This is a cloud storage used as distributed file system. The Hadoop Distributed File System (HDFS) is the primary data storage system used by Hadoopapplications. It employs a NameNode and DataNode architecture to implement a distributed file system that provides high-performance access to data across highly scalable Hadoop clusters.

***Apache YARN:***

YARN is one of the core components in open source Apache Hadoop distributed processing frameworks which helps in job scheduling of various applications and resource management in the cluster. YARN was initially called MapReduce 2 since it took the original MapReduce to another level by giving new and better approaches for decoupling the MapReduce resource management for the scheduling capabilities from the data processing unit. YARN is being extensively used for writing applications by Hadoop developers. It lets people create applications and work with huge amounts of data and manipulate it in an efficient manner. YARN is much more effective and versatile than Hadoop MapReduce and this is exactly what is required in a world inundated with big data and there is a perennial search for the next most valuable tool for working in the ever-increasing and challenging environment of Big Data Hadoop.

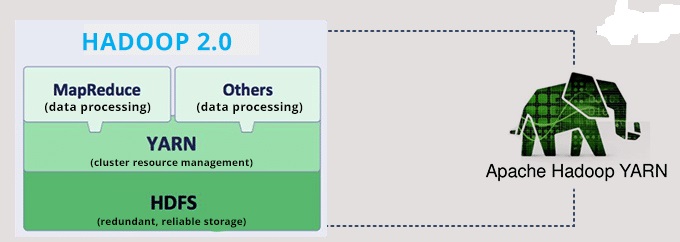


Fig: How does Apache Hadoop YARN work?

***MapReduce:***

The term ‘MapReduce’ actually refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job. Another definitation could be, Hadoop MapReduce (Hadoop Map/Reduce) is a software framework for distributed processing of large data sets on compute clusters of commodity hardware. It is a sub-project of the Apache Hadoop project. The framework takes care of scheduling tasks, monitoring them and re-executing any failed tasks.

***Apache Tez:***

Apache Tez is an extensible framework for building high performance batch and interactive data processing applications, coordinated by YARN in Apache Hadoop. Tez improves the MapReduce paradigm by dramatically improving its speed, while maintaining MapReduce’s ability to scale to petabytes of data. Important Hadoop ecosystem projects like Apache Hive and Apache Pig use Apache Tez, as do a growing number of third party data access applications developed for the broader Hadoop ecosystem.

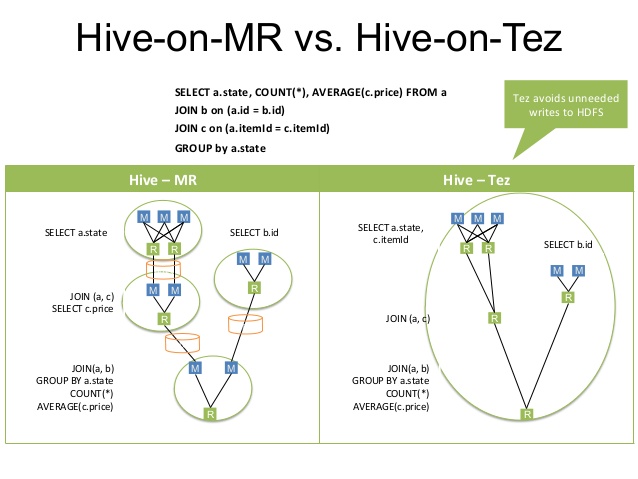


Fig: comparison between MR and Tez for running the same query.

***Apache Spark:***

Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets. With Spark running on Apache Hadoop YARN, developers everywhere can now create applications to exploit Spark’s power, derive insights, and enrich their data science workloads within a single, shared dataset in Hadoop. Spark promises a performance that is up to 100 times faster than Hadoop MapReduce. However, Tez also does the same kind of work but the basic difference between these two are, Spark can run as a standalone or on top of Hadoop YARN, while Tez can only run on top of YARN.

**Query Processing:**

**Hive:**

Hadoop was built to organize and store massive amounts of data of all shapes, sizes and formats. Because of Hadoop’s “schema on read” architecture, a Hadoop cluster is a perfect reservoir of heterogeneous data—structured and unstructured—from a multitude of sources. Hive provides the necessary SQL abstraction to integrate SQL-like queries ([HiveQL](https://en.wikipedia.org/wiki/Apache_Hive#HiveQL)) into the underlying Java without the need to implement queries in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop.[[3]](https://en.wikipedia.org/wiki/Apache_Hive#cite_note-:3-3) While initially developed by [Facebook](https://en.wikipedia.org/wiki/Facebook), Apache Hive is used and developed by other companies such as [Netflix](https://en.wikipedia.org/wiki/Netflix) and the [Financial Industry Regulatory Authority](https://en.wikipedia.org/wiki/Financial_Industry_Regulatory_Authority)(FINRA).[[4]](https://en.wikipedia.org/wiki/Apache_Hive#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Apache_Hive#cite_note-5) Amazon maintains a software fork of Apache Hive included in [Amazon Elastic MapReduce](https://en.wikipedia.org/wiki/Apache_Hadoop#On_Amazon_Elastic_MapReduce) on [Amazon Web Services](https://en.wikipedia.org/wiki/Amazon_Web_Services).

**Pig**:

With YARN as the architectural center of ApacheTM Hadoop, multiple data access engines such as Apache Pig interact with data stored in the cluster. Apache Pig allows Apache Hadoop users to write complex MapReduce transformations using a simple scripting language called Pig Latin. Pig translates the Pig Latin script into MapReduce so that it can be executed within YARN for access to a single dataset stored in the Hadoop Distributed File System (HDFS).

Since Pig Latin scripts can be graphs (instead of requiring a single output) it is possible to build complex data flows involving multiple inputs, transforms, and outputs. Users can extend Pig Latin by writing their own functions, using Java, Python, Ruby, or other scripting languages. Pig Latin is sometimes extended using UDFs (User Defined Functions), which the user can write in any of those languages and then call directly from the Pig Latin.

**Spark SQL**:

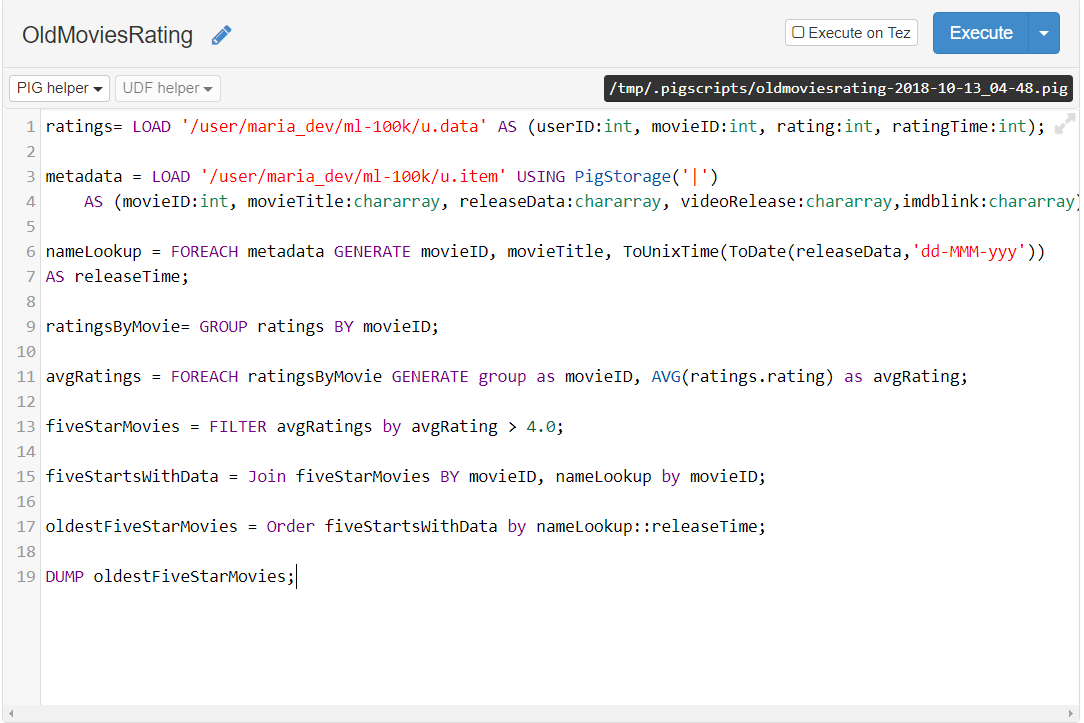
Spark SQL brings native support for SQL to Spark and streamlines the process of querying data stored both in [RDDs](https://databricks.com/glossary/what-is-rdd) (Spark’s distributed [datasets](https://databricks.com/glossary/what-are-datasets)) and in external sources. Spark SQL conveniently blurs the lines between RDDs and relational tables. Unifying these powerful abstractions makes it easy for developers to intermix SQL commands querying external data with complex analytics, all within in a single application. Concretely, Spark SQL will allow developers to:

* Import relational data from [Parquet](https://databricks.com/glossary/what-is-parquet) files and Hive tables
* Run SQL queries over imported data and existing RDDs
* Easily write RDDs out to Hive tables or Parquet files

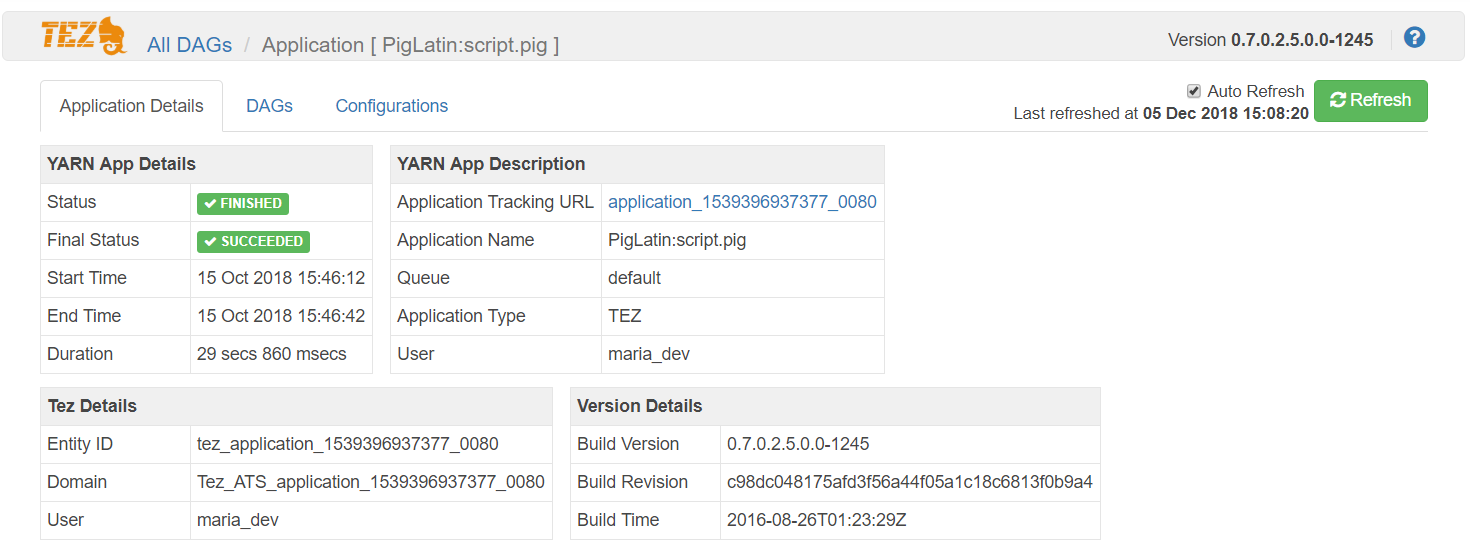
Spark SQL also includes a cost-based optimizer, columnar storage, and code generation to make queries fast. At the same time, it scales to thousands of nodes and multi-hour queries using the Spark engine, which provides full mid-query fault tolerance, without having to worry about using a different engine for historical data.

**Experiment**:

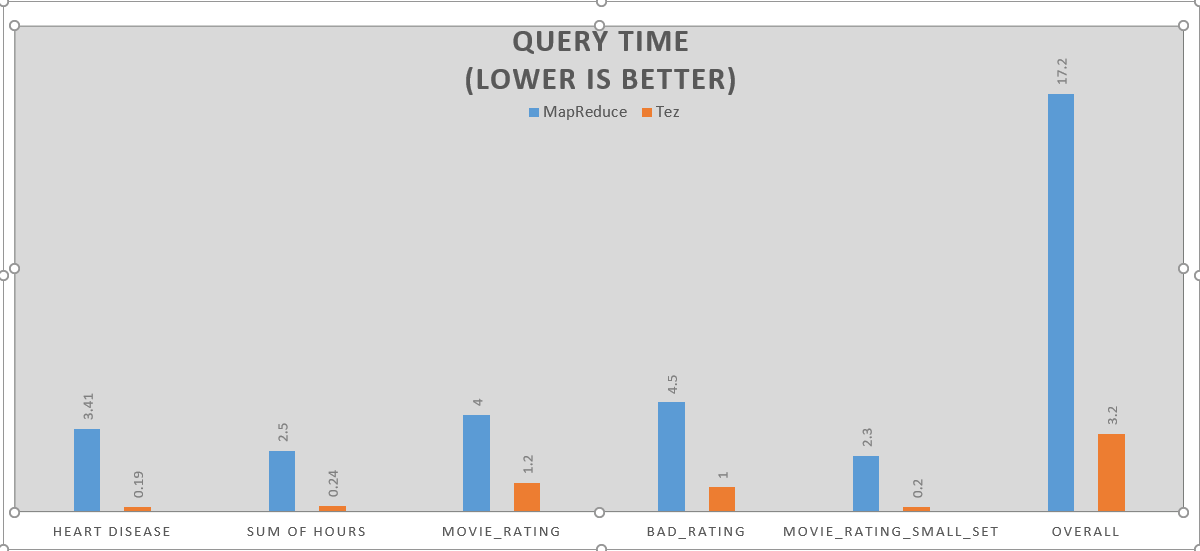
For a convincing result, we ran five test result with various length of dataset and calling different query. Most of our datasets are obtained from [www.kaggle.com](http://www.kaggle.com). Where the length of the dataset varied from 10mb to 100mb. Then various queries were run to get the inside of the dataset.



Here a movie dataset was used where I managed to get the rating < 4.00 for past 100 years. In the output file, it displayed the movie detail with its rating which must be <=4.

****

So, we conducted another four excrement using both MapReduce (MR) and Tez. From both the frameworks, it was possible to obtain identical same output. But using Tez the query time was lot smaller than MR. Now, we can find the query time variation for all the five datasets, it’s just because different datasets have various length.



**Decision**:

Apache Tez represents an alternative to the traditional MapReduce that allows for jobs to meet demands for fast response times and extreme throughput at petabyte scale. MR stores every temporary data into the HDFS, so for doing this back and forth it take time. MR uses lots of mapper for splitting the dataset as a result it takes time for the reducer to get all the output from each of the clusters. And every after these steps, it stores data into HDFS. While, on the other hand Tez takes a holistic approach towards job. It keeps the job into few number of clusters hence the number of mapper is also less. Moreover, it doesn’t store the temporary data into HDFS. Hence, it doesn’t have to spend any extra time for storing and fetching data from HDFS.

These are the reasons I found after my research, why Tez is faster in data processing than MapReduce.

**Conclusion**:

Now, we can easily understand why Tez performs faster than MapReduce. That’s the reason why Hadoop cluster has become so widely used with Apache Tez. On the other hand, MapReduce being on the slower side, it is losing its popularity.

***Reference:***

1. MapReduce and Apache Tez Data Compression Techniques. (<http://ijsrst.com/paper/2932.pdf>).
2. Hive on Tez-hive Integration with Tez. (<http://hadooptutorial.info/hive-on-tez/#tc-comment-title>).
3. Apache Pig (<https://hortonworks.com/apache/pig/#section_2>).
4. Difference between Pig & Hive-the two key components of Hadoop ecosystem (<https://www.dezyre.com/article/difference-between-pig-and-hive-the-two-key-components-of-hadoop-ecosystem/79>).
5. Apache Hive (<https://hortonworks.com/apache/hive/>).
6. Hadoop Ecosystem components and its Architecture (<https://www.dezyre.com/article/hadoop-ecosystem-components-and-its-architecture/114>).
7. Apache Hadoop YARN (<https://hortonworks.com/apache/yarn/>).
8. Hadoop ecosystem overview (http://thebigdatablog.weebly.com/blog/the-hadoop-ecosystem-overview).
9. Hive+Tez: A performance deep dive (<https://www.slideshare.net/Hadoop_Summit/w-235phall1pandey>).
10. Hadoop-MapReduce (<https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm>).
11. Apache Tez (<http://tez.apache.org/>).
12. Apache Spark (<https://hortonworks.com/apache/spark/>).
13. Hadoop MapReduce (<https://www.webopedia.com/TERM/H/hadoop_mapreduce.html>).
14. Spark SQL (<https://databricks.com/glossary/what-is-spark-sql>).
15. Apache Spark SQL (<https://spark.apache.org/sql/>).