ENHANCED LATE-STRAGGLER ALGORITHM WITH ON-DEMAND ETL FOR BIG DATA RETRIEVAL

ANWAR HUSSEIN ZAKARIA KATRAWI

UNIVERSITI SAINS MALAYSIA

2022

ENHANCED LATE-STRAGGLER ALGORITHM WITH ON-DEMAND ETL FOR BIG DATA RETRIEVAL

by

ANWAR HUSSEIN ZAKARIA KATRAWI

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

November 2022

ACKNOWLEDGEMENT

In the name of Allah (SWT), the most gracious and merciful, I am thankful to my Creator, who blessed me with the abilities to complete my Doctorate.

I would like to express my heartfelt gratitude to **Prof. Dr. ROSNI ABDULLAH** for her significant contributions through the supervision of this thesis. She made invaluable inputs into the research through her guidance from the start of the research proposal to the ultimate completion of the thesis. She challenged me many times during writing the thesis and helped me refine and shape my ideas to fit the parameters of the research I was doing. She gave me valuable but detailed feedbacks. On top of that, she passed onto me valuable pieces of literature she found related to my research topic and constantly reminded me of the importance of reflecting critically on my work. Through her guidance, I learned valuable research experience and skills to stay with me throughout my life.

I pay my profound gratitude and recognition to my co-supervisor **Dr. MOHAMMED ANBAR**, who led my research activities in fixing the real research gaps. His precious time, advice, and concerns helped me immensely in improving and completing this thesis.

I dedicate this special document of my education to my parents **Mr. HUSSEIN ZAKARIA** (father) and Madam **FAYZA AHMAD** (mother), and my lovely wife, **GHADA AHMAD**, and my beautiful daughters.

Lastly, to all my friends who inspired, encouraged, assisted, and guided me in one way or the other but whose name could not be mentioned here, thank you, and may God richly bless us all.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

ATE	Average Execution Time for node
BS	Block Size
CPUU	CPU utilization
DB	Database
DB	Database
DBMS	Database Management System
DBMS	Database Management System
DFD	Data Flow Diagram
DFD	Data Flow Diagram
DML	Data Manipulation Language
DML	Data Manipulation Language
DS	Data Size
DW	Data Warehouse
DW	Data Warehouse
ETL	Extract-Transform-Load or Extraction-Transformation-Loading
GB	Gigabyte
GB	Gigabyte
HDFS	Hadoop Distributed File System
HDFS	Hadoop Distributed File System
HDP	Hadoop
HDP	Hortonworks Data Platform
HDP	Hortonworks Data Platform
HI_LATE	LATE with historical information
Hive	SQL Hive Query Language

HiveQL	Hive Query Language
HPL/SQL	Hybrid Procedural SQL
HPL/SQL	Hybrid Procedural SQL
HwB	Hadoop without Backup
JDBC	Java Database Connectivity
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
JVM	Java Virtual Machine
LATE	Longest Approximate Time to End
MB	Megabyte
MB	Megabyte
MPP	Massively Parallel Processing
MPP	Massively Parallel Processing
MR	Map-Reduce
MR	Map-Reduce
NoSQL	Not Only Structured Query Language
NoSQL	Not Only Structured Query Language
ODBC	Open Database Connectivity
ODBC	Open Database Connectivity
OLAP	Online Analytical Processing
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OLTP	Online Transaction Processing
DCS	Data Capturing Schema
PL/SQL	Procedural Language/Structured Query Language
RDBMS	Relational Database Management System
SQ	Structured Query Language

TNDB	total no. of Data blocks
Tnew	Normal runtime
SA	Staging Area
Trem	Expected remaining time
UM	Utilized Memory

PENINGKATAN ALGORITMA PELENTANG-LATE BERSAMA ETL ATAS-PERMINTAAN UNTUK PENCAPAIAN DATA RAYA

ABSTRAK

Pertumbuhan informasi digital adalah luar biasa. Dokumen digital menguasai hampir setiap aspek perniagaan sehingga sukar untuk dibayangkan tanpanya. Dengan potensi yang belum pernah berlaku sebelum ini, revolusi informasi digital yang sedang berjalan juga memberikan risiko dan cabaran, terutamanya apabila berurusan dengan pengekstrakan dan analisis data digital. Kaedah konvensional ETL pemprosesan *Big Data* terdiri daripada Pengekstrakan, Transformasi dan Pemuatan yang disepadukan ke dalam gudang. Penggunaan kaedah ini tanpa sebarang pengoptimuman selalunya membawa kepada kelewatan dalam pengambilan data, yang dikenali sebagai masalah *straggler*, iaitu situasi yang timbul apabila tugasan ditangguhkan kerana pemprosesan yang rendah pada beberapa nod. Masalah *straggler* dianggap sebagai masalah utama, terutamanya apabila sumber data adalah penting dan jika sumber ini digunakan secara tidak cekap. Oleh itu, mengesan dan, oleh itu, menghapuskan masalah straggler lebih awal adalah penting untuk meningkatkan prestasi ETL. Kerja ini mencadangkan pendekatan ETL hibrid yang menggabungkan *MapReduce* dengan Pemintaan *Extraction Transformation Loading* (ETL). Kajian menunjukkan bahawa *Longest Approximate Time to End* (LATE) muncul sebagai Algoritma Pengendalian dalam *MapReduce*, kerana ia mengatasi prestasi *Hadoop* penjadual sebenar *Mantri*, *MonTool* dan algoritma *Dolly*. *Combinatory Late-Machine* (CLM) yang dicadangkan menggabungkan algoritma prestasi CPU dengan algoritma LATE dan mengurangkan masa pelaksanaan sebanyak kira-kira 33% kepada Penjadual LATE dan 42% kepada

Hadoop sebenar berdasarkan metrik yang digunakan. Menggunakan pendekatan H-ETL yang dicadangkan, peningkatan dalam mempercepatkan masa yang diperlukan untuk pemprosesan dicapai apabila bilangan tugasan meningkat dalam Pemetaan. Pada masa yang sama, bilangan *Mapper* berkurangan. Penjimatan masa yang ketara dicapai, di mana penjimatan masa ialah 6 minit (8%), 29 minit (24%), dan 35 minit (35%) apabila bilangan *Mapper* meningkat daripada 25 kepada 30, 30 kepada 38 dan 25 kepada 38, masing-masing. Oleh itu, pendekatan H-ETL yang dicadangkan adalah lebih cekap daripada pendekatan sedia ada, seperti ETL tradisional dan pendekatan Pengagihan Berbutir halus yang dicadangkan oleh Mahfoud Bala.

ENHANCED LATE-STRAGGLER ALGORITHM WITH ON-DEMAND ETL FOR BIG DATA RETRIEVAL

ABSTRACT

The growth of digital information is phenomenal. Digital documents dominate nearly every aspect of doing business to the point that it is hard to imagine doing without them. With an unprecedented potential lurking in its depths, the ongoing digital information revolution also presents risks and challenges, mainly when dealing with the extraction and analysis of digital data. The conventional method ETL of Big Data processing consists of Extraction, Transformation, and Loading integrated into a warehouse. Using this method without any optimization often leads to a delay in data retrieval, known as the straggler problem, which is a situation that arises when tasks are delayed due to low processing on some nodes. The straggler problem is considered by many as a major problem, especially when the data resources are important and if these resources are inefficiently used. Hence, detecting and, therefore, eliminating the straggler problem early is crucial to enhancing the ETL performance. This work proposes a hybrid ETL approach that merges MapReduce with Extraction Transformation Loading (ETL) on demand. Investigations show that Longest Approximate Time to End (LATE) Scheduler pops up as a Handling Algorithm in MapReduce, as it outperforms Hadoop native scheduler Mantri, MonTool, and Dolly algorithms. The proposed Combinatory Late-Machine (CLM) combined CPU performance algorithm with the LATE algorithm and decreased the execution time by about 33% to LATE Scheduler and 42% to Hadoop native based on the used metrics. Using the proposed approach H-ETL, an improvement in the speed-up of the time required for processing is achieved when the number of tasks increases in Mapping.

At the same time, the number of Mappers is decreased. A significant timesaving is achieved, where the time saving is 6 minutes (8%), 29 minutes (24%), and 35 minutes (35%) when the number of Mappers is increased from 25 to 30, 30 to 38 and 25 to 38, respectively. Therefore, the proposed H-ETL approach is more efficient than the existing approaches, such as the traditional ETL and the fine-Grained Distribution approach proposed by Mahfoud Bala.

CHAPTER 1

INTRODUCTION

1.1 Overview

In the last decades, data would stay and be consumed in one place when there were no interconnected systems. At the beginning of Internet technology, the ability and demand for exchanging and transforming data are urgent. It conducts the utilization of Extraction Transformation Loading (ETL) (Patel, 2018). ETL helps in transforming, reloading, and reusing the data. For this purpose, the companies made an essential investment in ETL infrastructure, both data warehousing and Big Data hardware and software (Marr, 2015). Big data analytics hurdles include rapid growth of data, organizational resistance, data validation, integration of disparate sources of data, and timely generation of insights from data collected from populations of interest (Arulmurugan, 2019). These hurdles involve data capturing, curation, storage, searching, sharing, transfer, analysis, and presentation (Verma, 2017). ETL provides the solution for these challenges with efficient Extraction (E) of data coming from heterogeneous sources, Transformation (T), and Loading (L) into a designated database (Kaushik, 2017). MapReduce is a framework and attempts to accomplish the task by detecting and managing positions of stragglers through a set of algorithms, a situation that arises when tasks are delayed due to reduced processing in some nodes. Straggler is a big challenge while implementing MapReduce due to parallelism and distributed processing of data. The EA algorithm is one of the algorithms that work to detect stragglers tasks (Bhandare et al., 2016).

The researchers propose an enhancement in the big data retrieval method by combining enhanced LATE-STRAGGLER ALGORITHM WITH ON-DEMAND ETL FOR BIG DATA RETRIEVAL.

1.1.1 Big Data

One of the core areas in concurrent research and practice is Big data analytics, which refers to collecting, organizing, and analyzing huge datasets (Debortoli et al., 2013). Data analysis is being transformed into wide-ranging approaches based on data, including huge data access, and providing better chances in science, commerce, and digital applications (Lee et al., 2014; Shamsi et al., 2013). These applications are perfectly parallel and suitable for MapReduce programming that enables users to do large-scale data analysis such as handling the task scheduling by the application, system architecture, and data partitioning (Jain et al., 2013).

Data analysis has been the backbone of any big data in all enterprises, and it will continue moving forward. According to the statistics provided by the Hadoop and Big Data Market, there will be a market growth from $17.1B in 2017 to $99.31B till 2022 and result in 28.5% growth, as illustrated by Figure 1-1.

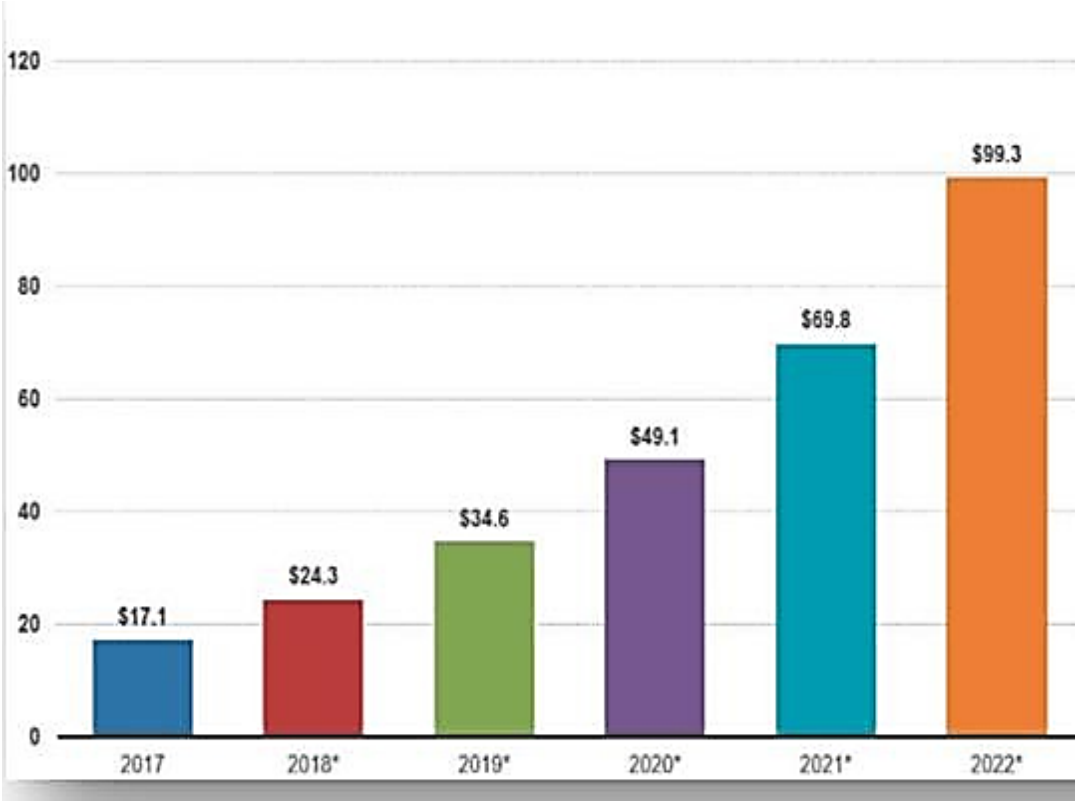

Figure 1.1 Big Data and Hadoop Market Worldwide From 2017 to 2022

1.1.2 Hadoop

Hadoop is a distributed system that runs on commodity hardware at a low cost. The input data is divided into chunks and dispersed among several nodes. The Node Manager in slave nodes communicates with the master node by sending a heartbeat message. Hadoop features a master-slave design. The master node distributes the work to slave nodes to process large-scale data based on the metadata and heartbeat messages. Hadoop comprises four primary modules: Hadoop common, YARN (yet another resource negotiator), HDFS (Hadoop Distributed File System), and MR. Hadoop common includes Hadoop configuration tools, libraries, and support functions for other modules. YARN uses the scheduler and application manager to manage cluster resources such as slave node resources, task scheduling, and data node monitoring.

Hadoop represents a programming framework that enables large data sets to be processed in a distributed computing environment. Apache's Hadoop ecosystem consists of the Hadoop kernel, MapReduce, Hadoop Distributed File System (HDFS) and several different components like Apache Hive, Base and Zookeeper (S. Lee et al., 2019). As its name says, Hadoop Distributed File System (HDFS) is a file system whose design is based on being distributed and handles large data sets running on commodity hardware in the Hadoop framework, so Hadoop is a suitable environment for ETL implementation.

1.1.3 MapReduce in Extract, Transform and Load (ETL)

MapReduce is a technique for simultaneously processing huge datasets over several cores. The scheduler first creates various containers in the slave nodes to divide the job into multiple tasks. A container is a YARN Java virtual machine process linked to a set of physical resources, such as a CPU core, disc space, and memory.

MapReduce is a Java-based computing model that manages the processing on distributed servers by converting input data into another set of data and summary methods. MapReduce generates enormous information. MapReduce is utilized in big information applications in big companies such as Yahoo, Cloudera, Amazon, and Google, among several others (Xu et al., 2017).

MapReduce architecture has two functions: Map and Reduce. The data input process occurs by dividing them by the Hadoop Distributed File system (HDFS) in the form of fixed-size blocks. In contrast, the process is done in a parallel way. The Map task work by acquiring the data in the form of key-value pair. After that temporary

(intermediate) result is produced using the same form. Since it's intermediate, it's used as input for the Reduce task that shares the results.

Despite a lot of ETL and MapReduce compatibility," data partitioning" is a missing aspect of the classic ETL process. To distribute and parallelize the ETL process, we must add a "data partitioning phase" after the data extraction (E). The transforming phase (T), which consists of cleansing, filtering, merging, conforming, and aggregating data, must be adapted according to the MapReduce model. In the ETL scheme, the T phase will be divided into two phases: (i)" transforming phase" (cleansing/standardizing) and (ii)" merging/aggregating phase". Thus, the ETL can handle very large data and tasks (El Akkaoui et al., 2019). The similarity between the ETL and the MapReduce process makes it easy to adapt the classic ETL scheme quickly. The MR process on Hadoop can be broken into small stages as shown in Figure 1.2.

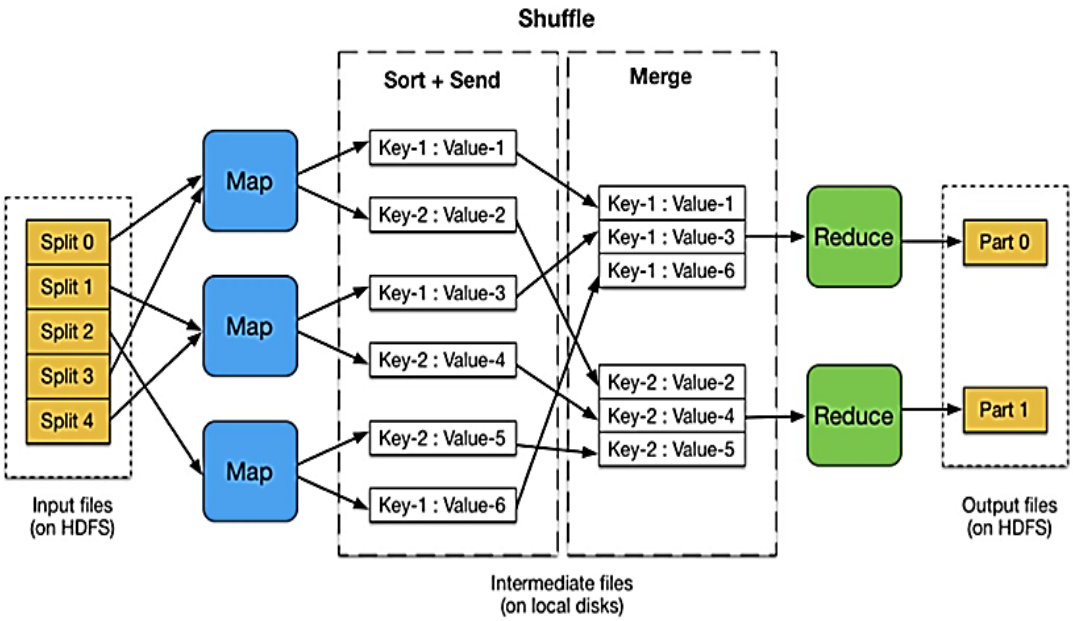

Figure 1.2 Use the Home tab to apply 0 to the text that you want to appear here. MapReduce stages

1.1.4 Straggler Detection Algorithm

For issues like Web indexing, data sorting, data searching, and other issues, MapReduce, a fundamental model for preparing and creating massive data sets, offers a straightforward, ad hoc solution. MapReduce (Dean & Ghemawat, 2004) has been utilized by organizations like Google, Facebook, and Yahoo as a component of many Big Data Applications structures.

Differences in network traffic and CPU accessibility frequently result in stragglers.

Even a few stragglers can greatly reduce the overall response time for the job because a job in the MapReduce Framework does not finish until all map and reduce tasks are finished. Therefore, it is essential to identify stragglers early and move them to other, equally speedier frameworks. Better will be the overall response time for the employment before the straggler is found.

Innocently, one could assume that providing straggler care would be a straightforward process of double slower tasks.

It is actually a complicated problem for a number of reasons. First off, speculative assignments are not free; they require certain resources, such a system with other active tasks (Isard & Budiu, 2007). Second, choosing the node on which to conduct speculative tasks is just as important as choosing the task itself. Third, it may be difficult to identify stragglers and nods that are just a little bit slower than the norm in heterogeneous environments (Ananthanarayanan et al., 2008). Finally, Stragglers should be acknowledged for arriving as promptly as was predicted.

Numerous stragglers regulating approaches have lately been developed, and The Problem of Stragglers has recently received substantial discussion (Ananthanarayanan et al., 2013). These tactics could be fully considered. Speculative execution and blacklisting.

Examples of Stragglers detection and mitigation tools include the Hadoop native scheduler, LATE, Mantri, MonTool, and Dolly.

1.2 Research Motivation

With the advent of digital technology and smart devices, a massive amount of digital data is being generated daily. Systems based on digital sensors and their remote connectivity have a communication addendum as another massive amount of data capturing valuable information for enterprises (Zdravevski et al., 2019). Big data is difficult to process using traditional technologies and necessitates massive parallel processing (Chen et al., 2020). Technologies that can store and process terabytes, petabytes, zettabytes of data without raising the data warehousing cost are the need of this time (Kozielski & Wrembel, 2009). The aptitude for drawing insights from this massive data can transform how people live, think and work. Big data open-source technologies have gained quite a bit of transaction due to the demonstrated ability to process large amounts of data in parallel. Both parallel processing and the technique of bringing effective computation of data have made it possible to process large datasets at high speeds (Zdravevski et al., 2019). These key features and the ability to process vast data have been a great motivation to investigate the architecture of the industry-leading big data processing framework by Apache, Hadoop. Distributed computing accomplished broad appropriation because of consequently parallelizing and transparently executing tasks in distributed environments. Straggling tasks is an

essential test confronted by all Big Data Processing Frameworks for example MapReduce, Dryad, and Spark. Stragglers are the assignments that run much slower than different tasks and since a job completes just when it's last undertaking completions, stragglers postponement work fruition.. Understanding how big data storage and analysis is achieved and experimenting with ETL vs Big Data environments can provide a great insight into the much talked about technology.

1.3 Research Problem

The extensive increase in the amount of digital data has dramatically changed the way organizations' functionalities impact our lives directly or indirectly.

ETL can revisit the emergence of new paradigms such as MapReduce to improve the ETL performance because with MapReduce the data is divided into partitions in a parallel manner (Yoo et al., 2019). Stragglers are jobs that take significantly longer to complete than comparable ones. There are various reasons for the task to take longer, including broken equipment, hardware heterogeneity, the amount of data to analyze, system obstruction, and competition for current assets.

When stragglers occur, it will result in an increased time required for task execution, which means that the performance of the entire job will suffer due to several reasons such as imperfect machines, the proportion of information to process, framework blockage, heterogeneity among equipment, and communication bandwidth (Gill et al., 2020) (Karagiannis et al., 2013). Existing approaches based on cloud computing such as prophet, quasar, bubble-up and others (Chen & Guo, 2017), NoSQL models (Yoo et al., 2019) showed that stragglers problem still appear when using

models such as Dolly, the Hadoop native scheduler, MonTool, LATE, and Mantri (Arpitha et al., 2017).

In addition, those approaches are not considered node resources (CPU and memory) because they focus on task problem only. Therefore, a concentrated effort is vitally needed to effectively detect and mitigate strugglers' effects (Bhandare et al., 2016; Liya Thomas, 2014; Brahmwar et al., 2014).

1.3.1 ETL Performance

Meanwhile, another reason for the degradation in the performance of ETL, is the dealing with a variety of Big Data sources, while Traditional ETL works well with clean and consistent data; however, it fails to deal with the variety of Big Data sources effectively (El Akkaoui et al., 2019; Karagiannis et al., 2013).

1.3.2 ETL Synchronization

ETL should be performed at off-peak hours, which means analysis and operational activities must be stopped (Muddasir & Raghuveer, 2017; Naeem et al., 2008; Naik et al., 2019; Zuters, 2011). This condition will be not suitable for the systems that are running 24/7 (Muddasir & Raghuveer, 2017).

1.3.3 ETL Extraction Problem

In the ETL process before performing any user query operations on them. In some cases, this may be unsuitable since the data is not offered by the provider for free or become difficult due to their large size, which means the whole data would be fetched and that is translated into consumed time and hence increased cost (Baldacci

et al., 2017; Nesrine et al., 2018) on the other hand to this, direct execution of query on the data source limits the ability for data to be updated which by time leads to inadequate results and thus high costs (Machado et al., 2019).

1.3.4 Research Problem is summarized as follows:

i. Existing approaches lacks in overcoming the straggler problems which suffers from low performance because of an unexpected slow task execution time.

ii. Some algorithms such as Hadoop native scheduler, MonTool, LATE, and Mantri are not considered node resources because they focus on task problem only.

1.3.5 Research Problems related to ETL

i. Existing approaches face a big challenge to synchronize the changes or updates that happen in the data source with the data centre or warehouse.

ii. Existing approaches focus on the data extraction without identifying what data to be queried and retrieved and thus return a huge amount of data that incur more time and more costs.

1.4 Research Objectives

The main goal of this research is to propose a hybrid approach for big data retrieval based on Enhanced Transform and Load (ETL) and MapReduce. The following objectives are set to achieve the main research goal:

i. To propose a straggler detection algorithm based on improvement of LATE algorithms to enhance MapReduce performance in terms of time execution.

ii. To enhance ETL by introducing on demand usage model for improvement of big data retrieval using MapReduce.

iii. To propose a hybridization based enhanced ETL algorithm & enhanced MapReduce mechanism for the big data retrieval.

1.5 Scope and Limitation

Our research focuses on two Big Data analytical tools, namely, ETL and MapReduce, whereby the limitations of the techniques have been identified and discussed. It has been noted that inefficiencies, particularly of stragglers, result in the high cost of data processing and a waste of resources such as time, storage, and memory. The data used in this experiment was derived from the Stack Overflow posts of questions and answers along with their metadata dataset. The limitations of this research are the type of data which is text, and the size of data, 200 GB.

Table 1.1 Scope and Limitation

Data Set	Stackoverflow question
Evaluate matrix	Sort, Grep and WordCount
Type of data	TEXT
Data Set Size	200GB

1.6 Research Contributions

We will study 5 straggler algorithms used in HADOOP, choosing the best one among them and working on improving it.

i. An enhancement of MapReduce performance by using an improved straggler detection algorithm.
ii. An enhancement of the ETL approach through the utilization of ETL on-demand process.
iii. An enhancement in big data retrieval method by combining enhanced ETL and MapReduce mechanism.

The first contribution will be combined with the second to get faster data processing and retrieval data in this research work. In this contribution, we will improve the ETL on demand by solving the problem of updating the source data.

1.7 Thesis Organization

This thesis is structured into the following six chapters:

Chapter 2 discusses the background of the research and related studies. This chapter critically reviews the existing approaches for the straggler detection algorithm, ETL and ETL approach presents their advantages and limitations. Finally, this chapter comprehensively discusses the gaps in the existing approaches.

Chapter 3 explains the methodology phases of the proposed approach for the Hybrid Approach for big data retravel. Additionally, it describes the integrated phases of the proposed approach.

Chapter 4 presents the design and tools used for the proposed approach. This chapter contains the HETL, and rules designs of the proposed approach. This chapter also explains the implementation of the phases in detail.

Chapter 5 reports the experiments and their results. It also presents a comprehensive analysis of the results achieved using the proposed approach. In addition, this chapter evaluates the performance of the proposed approach in comparison with existing approaches.

Chapter 6 presents the conclusions drawn from our work and suggests possible directions for future research.

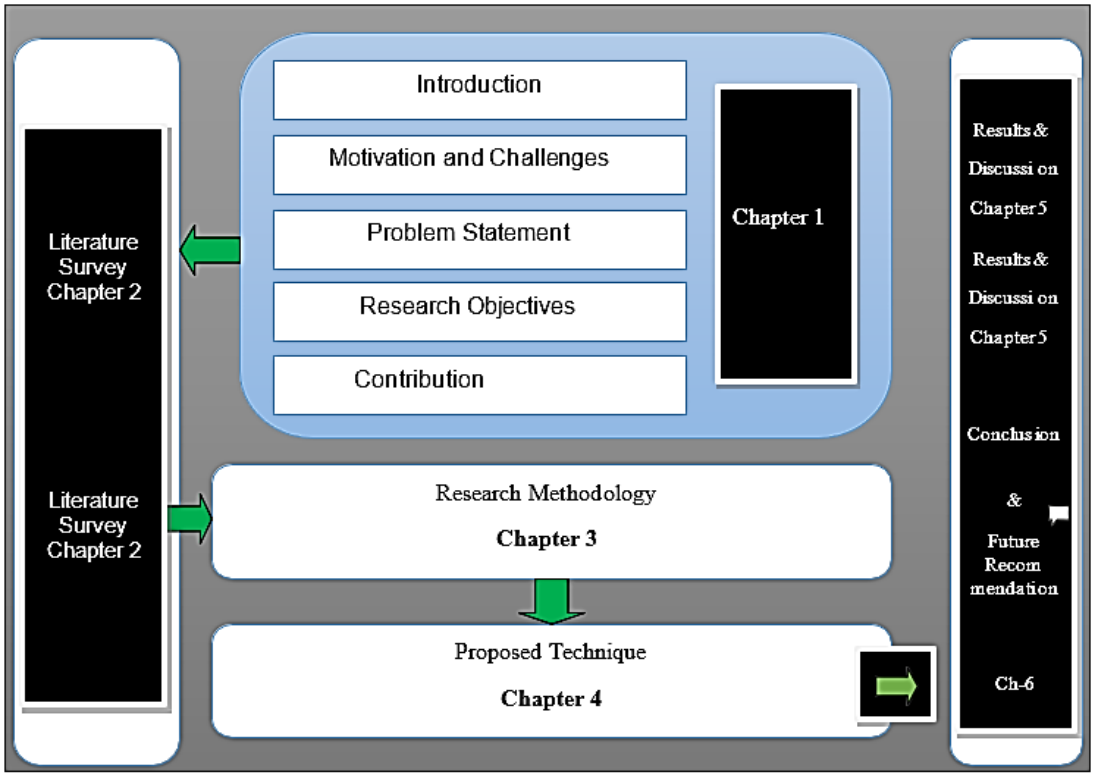

Figure 1.3 Thesis Organization

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents literature survey of Big Data and its various tools & techniques which are used for the Big Data analysis. ETL on Demand, MapReduce, and Stragglers Detection are all discussed in detail in Chapter 2. This chapter is divided into four sections: section 2.1 presents the background, section 2.2 discusses ETL architecture, section 2.3 discusses MapReduce design, section 2.4 discusses Related Work, Section 2.5 discusses about Research gap analysis and section 2.6 concludes with a summary.

2.2 Big Data

The researchers have proposed a strategy for 'blending' large and dense analytical insights to address this underlying issue. They establish a methodological framework based on the cognitivist linguistics term "blending." (Bornakke & Due, 2018).

Beyond standard transactional data, the growth of data kinds from numerous sources such as social media, mobile devices, etc., provides a tremendous degree of diversity. As a result, the data are no longer in easy-to-consume net structures but rather in many structures, such as organized, unstructured, and semi-structured data. Big Data encompasses all these data kinds. The architecture level in the data sources layer is the starting point for all subsequent Big Data processing. This layer directly connects to the Ingestion layer, which is responsible for validating, cleaning,

transforming, reducing, and integrating data for the Hadoop ecosystem to utilize its eventual sources and Big Data Ingestion layers. The researcher feels it is vital to point out that their work continues the first two comparison studies on the main five Hadoop Big Data distributions. They have used a Model-Driven Engineering "MDE" methodologies to propose a universal Meta-modelling for data sources and Big Data Ingestion layers (Erraissi et al., 2018).

The authors have investigated data integration options because one of the primary benefits of having many data types available is the capacity to fuse information. They also investigate data quality issues and time-related factors like recency and change frequency. Most of the data is structured and published in standard forms that are straightforward to analyze; there is sufficient possibility to combine diverse data sets, and the volume of data is continually expanding. They have identified several issues that must be addressed for these data to be effectively utilized (Barbosa et al., 2014).

Semi-structured data, on the other hand, is made up of both structured and unstructured forms of information (Suchitra 2017). This type of data is usually structured but the form in which it is organized remains unknown.

The researchers have created a competency taxonomy for big data and business intelligence by evaluating and interpreting the LSA's statistical results. Their main findings are that: 1) Business knowledge is just as important as technical skills for working successfully on BI and BD initiatives. 2) BI competency is defined by skills related to commercial products from large software vendors, whereas BD jobs demand strong software development and statistical skills. 3) The demand for BI competencies is still far greater than BD competencies. 4) BD initiatives are currently much more

human-capital-intensive. Individual professionals, businesses, and academic institutions can use their findings to measure and improve their BD and BI skills (Debortoli et al., 2014).

The authors examine huge data that spans years, from the past to the present and into the future. Map Reduce with Hadoop distributed File System (HDFS) is also discussed to handle the problem space of unstructured analytics. Some technologies and strategies are available to analyze terabytes of data efficiently daily and the challenges, issues, and benefits of big data (Devakunchari, 2014).

2.2.1 Applications of Big Data

By delivering analytics and predictive methodologies, big data analytics assists businesses and entrepreneurs in making better-educated business decisions. Big Data is almost everywhere. Big data analytics could be used in any industry, such as health care or general living standards. Big data is a field that we can use in any industry to use a large amount of data to one's benefit. The most common uses for big data are given below (Misra et al., 2014).

Healthcare

Electronic health records have resulted in a vast amount of data. Clinic data, patient data, and machine-generated/sensor data are the three forms of data created in a hospital or clinic.

The Third Eye-Data idea

It is becoming increasingly important to businesses around the world. Big Data analytics is a one-stop-shop for practically any business; it aids in predicting client purchase patterns and detecting fraud and abuse.

Banking

The utilization of user data may exacerbate privacy concerns. Big Data Analytics may be able to reveal sensitive personal data by finding hidden connections between seemingly unmistakable bits of data. According to research, 62 percent of financiers are cautious in using Big Data due to isolation difficulties (Mukherjee et al., 2016).

Agriculture

A biotechnology company uses sensor data to improve procurement efficiency. It collects and runs reenactments to discover how plants react to various environmental changes. Its information environment adapts to changes in the quality of data, such as temperature, water levels, soil arrangement, development, yield, and quality sequencing of each plant in the proving ground. These games allow you to find the best ecological conditions for the best quality types.

2.2.2 Extract, Transform and Load (ETL)

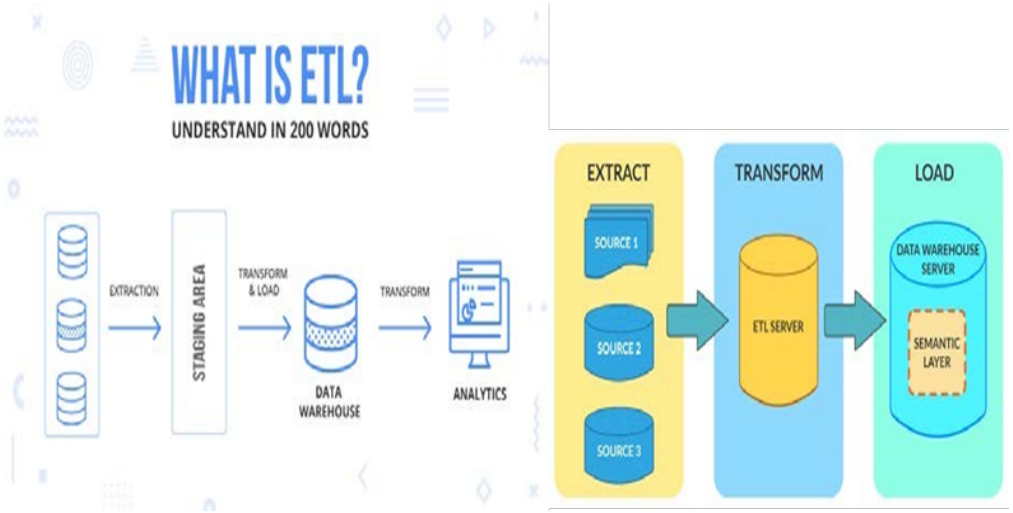

Figure 2.1 ETL Stages

https://www.spec-india.com/tech-in-200-words/what-is-etl)

The Extract-Transform-and-Load (ETL) method is devoted into the extraction of required source data, which is then prepared for transformation, followed by cleansing, then standardization and conforming, and finally loading in the warehouse, which provides important information for analysis and decision-making, as shown in Figure 2.1 and corresponding discussion is done further in this section.

Step 1: Extract

Data extraction from many source systems is the initial step in the ETL process. Most data storage systems combine data from many sources. Each design can organize data differently or utilize alternative formats. The data is converted to a form that we may use to begin the transformation process during the extraction procedure. The extraction process includes analyzing the extracted data, which leads to a check to see if the data matches the predicted pattern or structure. If this is not the case, we will discard the data. The extraction task must meet several key requirements: it has as little influence on the source system as possible. Extraction procedures in large systems are frequently scheduled during times or days when the impact is zero or minor.

During data extraction, raw data is exported or transferred from source sites to a staging area. Information can be retrieved from a variety of structured and unstructured data sources by data management teams. SQL or NoSQL servers, CRM and ERP systems, flat files, email, and web pages are just a few of these.

Step 2: Transform

The extraction data is transformed into data that we will load during the transformation step. Some data sources will necessitate a little amount of data

processing. In other circumstances, transformations may be required, for example, to merge data from several sources, generate totals from many rows of data, divide one column into several, or reject the entire record of incorrect data.

Data processing is done on the raw data at the staging area. In this step, the data is transformed and consolidated to get it ready for its intended analytical use case. We may include the following tasks in this phase:

i. Data filtration, cleaning, removing duplicity, validation, and authentication of the data.
ii. Carrying out computations, translations, or summaries using the raw data. It may involve modifying text strings, changing row and column headings for consistency, converting money or other units of measurement, and more.
iii. Ensuring data quality and compliance via audits.
iv. Removing, encrypting, or protecting data governed by industry or governmental regulators
v. Data Formatting into tables or join tables for matching the schema of the target data warehouse.

Step 3: Load

The data from the previous phase is loaded into the destination system during the loading phase. Old data gets erased with new data in some databases. The data warehouse keeps track of the records to be audited, and we may trace a value's whole history across time. When it is desired to maintain many levels of granularity, the rolling process is used.

This last step involves moving the modified data from the staging area to the desired data warehouse. A full load of all data is typically the first step, followed by periodic incremental data updates and, less frequently, full refreshes to completely replace all the data in the warehouse. Most businesses that employ ETL have an automated, clearly defined, batch-driven process. ETL is often carried out off-peak hours when there is little to no traffic on the data warehouse and the source systems.

At the Internet-scale, scientific study necessitates access to, analysis of, and exchange of data dispersed across numerous heterogeneous data sources. As a first phase, an eager ETL process creates an integrated data repository, combining and loading data from all data sources. This approach' bootstrapping is inefficient for scientific research that necessitates access to data from very big and often scattered data sources. The metadata is loaded in a lazy ETL operation, but it is still eagerly. In terms of bootstrapping, lazy ETL is faster. However, queries on eager ETL's integrated data repository run faster since all the data is available ahead of time. The authors suggest a novel ETL approach for scientific data integration in this research, a combination of eager and lazy ETL approaches that can apply to both data and metadata (Kathiravelu et al., 2018).

Large organizations began to aggregate and store information from many sources with diverse data types, such as payroll systems, sales records, inventory systems, and so on, in the 1970s, and ETL was born. The need to combine this data arose naturally, opening the route for ETL to emerge.

The authors focus on the issue of "Volume" to assure high performance for Extracting-Transforming-Loading (ETL) operations, which is one of the so-called "4Vs" (volume, velocity, variety, and veracity) that characterize the complexity of Big

Data. They present a new fine-grained parallelization/distribution strategy for populating the Data Warehouse. Unlike previous techniques, which only distributed the ETL at the coarse-grained level of processing, their methodology allows for several levels of parallelization/distribution at the process, functionality, and elementary function levels. They outlined the ETL process in terms of basic capabilities that may be run on a cluster of computers using our approach's MapReduce (MR) paradigm. This novel technique allows the ETL process to be distributed at three levels: coarse-grained distribution at the "process" level, fine-grained distribution at the "functionality" and "elementary functions" levels. Their findings showed that using 25 to 38 parallel tasks allows the unique approach to speed up the ETL process by up to 33% while maintaining a linear improvement rate (Mahfoud Bala et al., 2017). Designing an extract, transform, and load (ETL) process is tough because of the ambiguity of user needs and the complexity of data integration and transformation; designing an extract, transform, and load (ETL) process is tough. Current research has looked towards using an ontology-based method to solve these restrictions by harmonizing the semantics of user requirements within the ETL process design to make the ETL process specification easier to generate. The authors used the Requirement Analysis Method for ETL Processes (RAMEPs) to create the ontology for ETL process activities. They gathered from the organization's perspectives, decision-maker, and developer. As a result, the ETL process specification for the student affairs data warehouse (DW) system is generated using ontology. They verified the accuracy of the ontology model. Furthermore, the case study's ontology creation process is described, demonstrating how the ontology-based approach successfully executed the design and developed the ETL process specification (Simitsis, 2007).

Enterprises' expectations for lower data processing delays and real-time requirements are accompanied by the rapid development of data warehouse technology and its applications. Unfortunately, most present systems are unable to provide these essential functionalities. The authors presented a real-time Extract, Transform, and Load (ETL) solution based on the Enterprise Service Bus (ESB). The ETL functionality was implemented as a component on the ESB platform. To load real-time records, they constructed a real-time partition. Experiments show that this design method successfully achieves the real-time property while maintaining a high modularity and extensibility level (GAO, 2008).

ETL management is essentially the metadata management of ETL procedures & for the construction of a DW. A well-designed metadata management system can greatly improve ETL efficiency. However, the huge number and extensive spread of metadata in ETL procedures lead to metadata mismanagement, for which there is currently no acceptable solution. The authors recommend that ETL be managed intensively through a metadata repository to address this issue. Metadata repository can display metadata to DBAs in a clear, simple, and focused manner, making metadata easier to comprehend. As a result, metadata management becomes more direct, simple, and focused. ETL methods based on metadata repositories provide a considerably better optimization effect than traditional ones (Li, 2010).

The data generated by social media platforms such as Facebook, Twitter, and YouTube introduce additional problems, and we must address challenges for the decisional support system (DSS) & Keeping in mind, the authors suggest a novel technique for ETL (Extract-Transform-Load) development termed as BigDimETL (Big Dimensional ETL). Their method integrates Big Data using the MapReduce

paradigm with the consideration of the Multi-Dimensional Structure (MDS) (Mallek et al., 2017).

Extraction-Transformation-Loading (ETL) tools are pieces of software that extract data from various sources, cleanse it, customize it, and load it into a data warehouse. The authors investigated the logical optimization of ETL operations in this study, representing it as a state-space search issue. They treat each ETL workflow as a state, and the state constructed space using a collection of proper state transitions. Furthermore, they present an exhaustive approach and two heuristic techniques for reducing the execution cost of an ETL Workflow. The heuristic method with greedy characteristics greatly beats the other two algorithms for a vast range of experimental scenarios (Simitsis et al., 2005).

2.2.3 Traditional vs On-Demand ETL

Traditional ETL approaches, which typically operate on a single machine (ETL server), are incapable of handling large data volumes (at the terabytes and petabytes scale). On the other hand, after the introduction of new techniques such as Cloud Computing MapReduce and NoSQL, the ETL can be considered (Biswas et al., 2019).

The datastore and data processing are receiving more and more attention in extracting crucial information as data exploration has risen fast in recent years. Finding a scalable way to process large-scale data in either the relational database system or the developing NoSQL database is a significant issue. MapReduce is appealing for processing big data in parallel because of Hadoop's inherent scalability and fault tolerance. Most past studies have focused on integrating the Hadoop distributed file system with SQL or SQL-like queries translators. However, it may not be easy to

update data regularly in such a file system. As a result, we require a flexible datastore like HBase to store data on a scale-out storage system and handle dynamic data transparently. The authors have proposed the JackHare framework, which includes a SQL query compiler, JDBC driver, and a systematically way for processing unstructured data in HBase using the MapReduce architecture. After importing the JDBC driver into a SQL client GUI, they use HBase as the underlying data store to run the ANSI- SQL queries. The results of our experiments suggest that our methods can perform well in terms of efficiency and scalability. In the following scenarios, the ETL process method could be deemed centralized: (a) the ETL process is done on a single machine (an ETL server), (b) in a single instance (one execution per time), and (c) the data size is medium (Chung et al., 2014).

The authors aim to present a survey on NOSQL Models, particularly a column-oriented NoSQL database, to provide the user with the benefits of utilizing NoSQL databases rather than relational databases to overcome the relational database's limitations (Gajendran, 1998).

The author examines the NewSQL data management system and contrast it with the NoSQL and classic database systems. He discusses the architecture, properties, and classification of NewSQL databases for Big data management using online transaction processing (OLTP). In separate category tables, it also lists popular NoSQL and NewSQL databases. He further compares SQL-based RDBMS, NoSQL, and NewSQL databases using a set of metrics and various NoSQL and NewSQL research challenges (Moniruzzaman, (2014)).