T.C.

ALTINBAS UNIVERSITY

Graduate School of Sciences and Engineering

Electrical and Computer Engineering

A COMPARISON OF DEVELOPED SCHEDULING IN APACHE FLINK FRAMEWORK

Hayder Majeed Jaafar FULFULI

Master of Science

Supervisor

Asst. Prof. Abdullahi Abdu IBRAHIM

Istanbul, 2021

A COMPARISON OF DEVELOPED SCHEDULING IN APACHE FLINK FRAMEWORK /DISSERTATION

by

Hayder Majeed Jaafar Fulfuli

Electrical and Computer Engineering

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “A COMPARISON OF DEVELOPED SCHEDULING IN APACHE FLINK FRAMEWORK” prepared and presented by “Hayder Majeed Jaafar FULFULI” was accepted as a Master of Science Thesis in Electrical and Computer Engineering.

Asst. Prof. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Çağatay AYDIN	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Abdullahi Abdu IBRAHIM	School of Engineering and Natural Sciences, Altinbas University	_____
Assoc. Prof. Adil Deniz DURU	Faculty of sports sciences, Marmara University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Hayder Majeed Jaafar

Fulfuli

Signature

ACKNOWLEDGEMENTS

Above all, I would like to express my sincere gratitude to the consultants of "asst. Proof. Dr. Abdullahi Abdu Ibrahim" for the continuous support of my studies and major research, for his patience, motivation, enthusiasm, and momentary knowledge. He helped me guide him all the time researching and writing this thesis. I never imagined i'd have a better advisor and mentor for my masters study.

I also dedicate this work to my father, may god have mercy on him, who wished to see me in this degree.

Finally, i express my sincere gratitude to my family for all their prayers and blessings. My dear wife and daughters, this journey would not have started without your advice and encouragement, and it was very difficult to continue and accomplish the journey without your support. Thank you for everything.

ABSTRACT

A COMPARISON OF DEVELOPED SCHEDULING IN APACHE FLINK FRAMEWORK/ DISSERTATION

Fulfuli, Hayder Majeed Jaafar,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Abdullahi Abdu Ibrahim

Date:19/ 03/2021

Pages: (68)

Big data applications have matured to be one of the fundamental components in the information technology sector nowadays, a chance for executives has been furnished to accomplish perfect results, for business and economics. Nevertheless, the occurrence of data grouping dispatch differs in administration, storage, and handling, in all common database systems unable to handle such functions as enormous data gathering. The administration of resource and function planning performs a material role in Big Data processing. Variable classifications are emerging of the schedulers, which are dependent on their activity, efficiency, and attribute. As a result, in this thesis, we will assort, examine, and approach a comparison particular information that is linked with certain of schedulers could be used in Big Data frameworks. Furthermore, this thesis will recognize the shortcoming and durability of the distinct utilization of these schedulers. Besides, the study investigates certain schemes for the competence of distinct utilization for defining in

each case the individual scheduler has some shortcoming or worthless. So, we will cover these issues in this thesis that are not examined in the existing researches.

Keywords: Job Scheduling, Scheduler Algorithms In Big Data, Apache Hadoop Job Scheduler, Big Data Job Scheduler, Apache Flink Scheduling

ÖZ

APACHE FLINK ÇERÇEVESİNDE GELİŞTİRİLMİŞ PLANLAMANIN BİR KARŞILAŞTIRMASI/ DAĞITIM

Fulfuli, Hayder Majeed Jaafar,

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi,

Danışman: DR. Abdullahi Abdu İbrahim

Tarih: 19/ 03/2021

Sayfalar: (68)

Büyük veri uygulamaları günümüzde bilişim sektörünün temel bileşenlerinden biri olmak için olgunlaşmış, yöneticilere iş ve ekonomi için mükemmel sonuçlar elde etme şansı verilmiştir.Bununla birlikte, muazzam veri toplama gibi bu tür işlevleri yerine getiremeyen tüm yaygın veritabanı sistemlerinde, veri gruplama gönderiminin oluşumu, yönetim, depolama ve işlemede farklılık gösterir. Kaynak ve işlev planlamasının yönetimi, Büyük Veri işlemede önemli bir rol oynar. Programlayıcıların faaliyetlerine, verimliliğine ve özniteliğine bağlı olan değişken sınıflandırmalar ortaya çıkmaktadır. Sonuç olarak, bu tezde, Büyük Veri çerçevelerinde kullanılabilecek belirli programlayıcılarla bağlantılı belirli bir bilgiyi sınıflandıracak, inceleyecek ve karşılaştıracağız. Ayrıca, bu tez, bu programlayıcıların farklı kullanımının eksikliğini ve dayanıklılığını da tanıyacaktır. Ayrıca, çalışma, her bir programlayıcının bazı eksiklikleri veya değersizleri olduğu her durumda tanımlamak için farklı kullanım yeterliliği için belirli planları araştırmaktadır. Dolayısıyla mevcut araştırmalarda incelenmeyen bu konuları bu tezde ele

alacağız.

Anahtar Kelimeler: İş Zamanlaması, Büyük Veri'de Zamanlayıcı Algoritmaları, Apache Hadoop İş Zamanlayıcısı, Büyük Veri İş Zamanlayıcısı, Apache Flink Zamanlayıcı

TABLE OF CONTENTS

Pages

page

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF CHARTS

Pages

LLIST OF ABBREVIATIONS

API	:	Application Programming Interface
CPU	:	Central Processing Unit
DAG	:	Directed Acyclic Graphs.
FIFO	:	First-In, First-Out
HDFS	:	Hadoop Distributed File System
I/O	:	Input and Output
IoT	:	Internet of Things
IT	:	Information Technology
VM	::	Java Virtual Machine
ATE	::	Longest Approximate Time to End
NIST	::	The National Institute of Standards and Technology
NKS	::	Next-k Scheduling
RDD	::	Resilient Distributed Dataset
SLR	:	Systematic Literature Review

1. INTRODUCTION

The technology world newly has emerged a growth dramatically to produce data. Basically, a large amount of data is being generated every second throughout the distinct processes accomplished in companies, in different sectors such as social media, healthcare, astronomy, oceanology, banking, industrialization, and many other sectors. Researchers use “Big Data” as a popular expression to characterize the sustainable increase and information accessibility (built, semi-built, or unbuilt) [1].

Big Data applications have matured to be one of the fundamental components in the present information technology field, a chance for executives has been furnished to accomplish perfect results, for business and economics. Though, the dispatch of such data-collecting differs in administration, storage, and handling in all common database systems unable to handle such functions as enormous data-collecting [2].The administration of resource and function planning performs a material role in Big Data processing. The purpose of the administration of resources is to assure that all loops are used rationally and shared appropriately [1].

Furthermore, function scheduling is one of the significant attributes of huge datasets processing and it is more liable for tasks organizing that could be set to the possible resources. There are several algorithms for scheduling function, where each one able to use certain technology for this objective. The primary function of the scheduler is to decrease estimation time and increase the use of the resource. For the processing and analyzing data sets, there are various effective means available at once, for example, Apache Flink [2]. These frameworks provide different types of scheduling features for practical workloads and applications. From the literature, the primary features of these means could be defined as follows:

- **Apache Flink:** Is another open-source framework for distributed stream processing. Flink able to process two types of data (stream and batch) [3], [4]. Flink has layered architectures, where each component is a part of the specific layer. Flink is designed to run on a local cluster (single JVM) or can be deployed to another cluster manager such as (YARN, Mesos, and Standalone) [3].

  Flink uses master (Job Manager) to handle job submission and worker (Task Managers) to execute different operations [5].

The Task manager executes the tasks that are received from the job manager, which in turn can to interchange information between them when the other worker needs information. Also, the task manager has many slots for the process in the cluster and is used for executing assignments in the parallel model. The assignments scheduling in the above-mentioned frameworks are of ultimate necessity since they make an impact on time estimation and resource utilization. Hence, the purpose of the current research is to match in thorough variable schedulers being used in Big Data frameworks along and to define the advantages and disadvantages of each in different use case schemes.

1.1 OBJECTIVE

This study will critically classify, match, and examine particular information that is linked with certain schedulers that could be used in Big Data frameworks. Besides, it will define the shortcomings and strengths associated with each scheduler in different utilization. A thorough discussion of these issues will be exited from certain points of view. Aiming to our objective, a systematic literature review methodology is used to gather information about schedulers and classify them from a different perspective. Consequently, the thesis is premeditated to be an extensive review of the possible schedulers along with a variable classification of schedulers depending on their variable activity, attributes, and efficiency.

1.2 METHODOLOGY

The objective of this study is to classify, match, and examine particular information that is linked with certain schedulers that could be used in Big Data frameworks. Besides, it will define the shortcomings and strengths associated with each scheduler in different utilization. Considering that, we will adopt the systematic literature review methodology, which is interpreting and assessing research related to any specific subject. A systematic literature review is completely depending on subsidiary researches, such as some published theses, scholarly articles, and investigation papers. The major cause for this choice due to the massive amount of information about Big Data scheduling that requires to be gathered, analyzed, and finally interpreted in a way as to give compulsory outcomes for the current job.

1.3 RESEARCH APPROACH CHOICE

In fact, there are various types of researches in the review of literature that can be

classified into two types: namely deductive and inductive. A deductive is completely dependent on the previous concepts and assumptions; while an inductive depends on non-statistical procedures. It is a reviewing process to develop innovative prototypes and concepts. Consequently, we adopted the logical method to get possible insights about scheduling in Big Data [6], [7].

1.4 RESEARCH METHODS

Even with the existence of several research methods of classification, the quantitative and non-qualitative methods are common ones, by the variance generating from the essential precepts in the research method. The non-qualitative method is used to collect non-numerical data of a matter, and also the prime indication is to make sure from all collected data through the support of the mechanisms of literal analysis. Furthermore, a quantitative method contains collecting numerical data and checks these data compiled by using statistical mechanisms. Intentionally, this study prefers, the non-quantitative method because it is the most suitable one for compiling secondary data (published theses, scholarly articles, and investigation papers) and by thematic analysis check [7], [9].

1.5 RESEARCH SCHEME

A research scheme is a proposed strategy that presents the path to hard works, ideas, facilitating the performance of the researchers analytically. Therefore, the literature review, need to apply the following phases:

- Research question conception.
- Search keywords to be manipulated.
- Inclusion or exclusion standards to be tuned.
- Paper's quality estimation to be set up.
- Methodology Implementation.

1.5.1 Research Question Conception

Research questions would replicate the main aim of the systematic literature review in order to that facets of the distinct objectives are included. We can build-up these questions as follows:

- What is the type of scheduling techniques are mostly applied in the Big Data framework?
- What is the reason behind using the above scheduling techniques?
- The appropriate scheduling algorithms evaluation in variable use?
- Strengths vs. Weaknesses of noticeable schedulers on Big Data frameworks?
- Scheduling improvements that supply in literature?

1.5.2 Research Keywords to Be Manipulated

For manipulation of logical literature review, we should examine the papers that are related to the associated subject. Thus, the selection of keyword is a substantial stage in research because the outcomes feature is straightly associated with the keywords which have been selected at the first spot. The wisely nominated conditions and the keywords that were used for directing the research in this study are:

- Apache Flink scheduling.
- Scheduler algorithms in Big Data.
- Scheduling of job.
- Job scheduler of Big Data.

In Table 1.1, we nominated the database which is used for searching for defining the related articles:

Table 1.1: Selected databases

Database	Location
Acm Digital Library	*Http://Dl.Acm.Org/*
Citeseerx	Http://Citeseerx.Ist.Psu.Edu/Index
Sciencedirect	*Http://Www.Sciencedirect.Com/*
Ieee Xplore	*Http://Ieeexplore.Ieee.Org/Xplore/Home.Jsp*
Google Scholar	*Https://Scholar.Google.Com.Tr/*

1.5.3 Inclusion or Exclusion Standards to Be Tuned

Exclusion and inclusion standards make filtering possible for the papers retrieved from databases to find the most relevant ones. Based on the research questions, the exclusion and inclusion standards in Table 1.2 are used to the particular papers [6].

The search in well-defined sources in this stage should be performed and the achieved researches need to be assessed according to the proven standards. Implementing the said search on the obvious sources thereafter, we acquired a group of 335 outcomes categorized by the inclusion standards. The studies were classified in advance to put in 79 principal suggestions.

Table 1.2: Inclusion and exclusion standards

Inclusion Standards
1. The Studies Scheduling Algorithm In Big Data. 2. The Journal And/Or Conference Papers. 3. The Studies That Describe The Scheduling Algorithm In The Big Data Frameworks. 4. The Main Or Subordinate Researches.
Exclusion Standards
1. The Studies Are Not Manageable In The Complete Text. 2. The Studies Which Are Not Associated With The Scheduling Algorithm In Big Data. 3. The Studies Not Submitted In English. 4. The Research Questions Are Not Being Responded By The Related Studies. 5. Reproduced Studies Such As Those Issued In Other Papers. In Such A Case; We Incorporated The Latest One. 6. Prefaces, Slides, Panels, Editorials, Or Tutorials.

1.5.4 Papers Quality Estimation to Be Set Up

The paper quality estimation acts as a dynamic part in SLR whereas it intends to assure the significant papers with regards to accurately qualifying to address the associated questions that are exercised in the study. For each paper incorporated in this study, these standards are to be set up [6, 8]:

- The research objectives are obviously identified or not?
- Are the researchers designate the search approach comprehensively?
- Is the search approach including a sufficient figure of years or not?
- Are the researchers submitting adequate data that corroborate their explanations and assumptions?
- Does the approach interpret suitably and sufficiently?

1.5.5 Methodology Implementation

The researchers implement at this point a thematic analysis for obtaining awareness and intuition from the gathered articles along with an issued thesis, academic articles, and examination papers. The familiar components of these papers consist of scheduling in Big Data and which scheduling is commonly applied within this framework. Besides, the advantage vs. Disadvantage of each scheduling algorithm is considered alongside schemes in order to make the process is appropriate cases for identifying which case scheduler will be gaunt or unworkable. The prime goal of such thematic analysis for directing the researchers to accomplish their aims for the study.

2. BACKGROUND INFORMATION

2.1 BACKGROUND OF BIG DATA

Along with the conception of IOT, besides an enormous growth in data size, Big Data research comes to be very crucial in Data Processing. Basically, Big Data refers to data that is unable to be manipulated in an old-fashioned database management settings [10],[12].

It is a common term used to describe the intensive rise and obtainability of data, either the controlled or uncontrolled one. The management tackles old-style data that has no capacity of handling, investigating the massive size of data. Thus, a different group means will be used normally entitled "Big Data Framework". For treating the visualization, analysis, storage, job scheduling, and error acceptance as challenges [13], [14].

The scheduler is one of the essential and vital elements of such a framework because it carries out its actions through related cooperation with the resource manager. Making the handling quicker and reduce the response period as much as possible are the fundamental goals of scheduling [13].

The scheduling activity in the mentioned framework has its impact on accomplishment duration, affects the finishing interval, boosts productivity, and dominates the obtainable resources.

2.2 BIG DATA FEATURES

Big Data becomes more complicated, particularly when the size of data turns into prolonged from numerous data sources and is intended to be connected, joined, and interrelated in order to insert the necessary information for assistance to conclude decisions perfectly. Big Data is a group of massive and complex data sets and sizes that consist of immediate data and common media analyses. Big Data has various features commonly mentioned by 7V's [15], [16].

The group of V's are the features of the Big Data and gathered from various researchers [17].

Firstly, Big Data was categorized with generally used three V's, but it has protracted by other V's. The Fourth "V" (Veracity) [21] was found by IBM. Oracle also found one

more dimension that was characterized as Value [22].

The National Institute of Standards and Technology (NIST) also added a new one, called Variability [23].

However, another dimension, Visualization was found by some scholars for Big Data [24]. As in Fig.2.1.

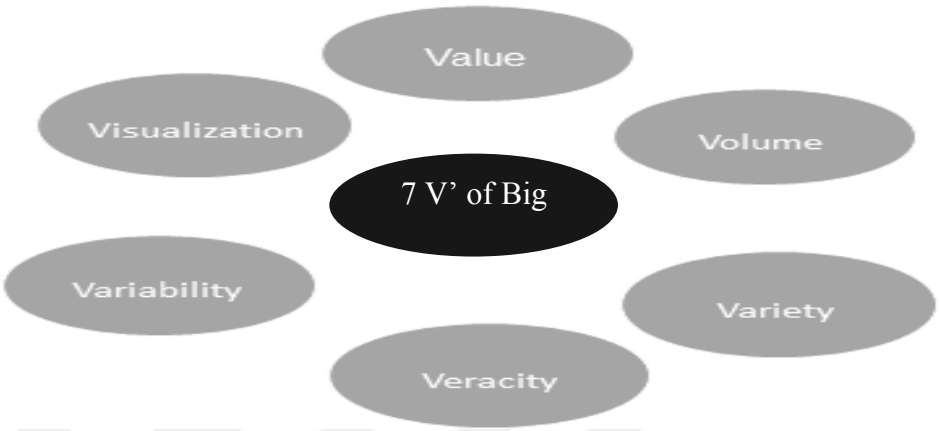


Figure 2.1: 7V's of Big Data

These seven features are pointed out briefly in the consequence.

2.2.1 Volume

The 'volume' simply means how much we have of data which needs analysis [18].

In Big Data researches, the volume is one of the most main characteristics which require to be well-thought-out, since the data rises in large volumes every day, but actually every single minute [19].

2.2.2 Velocity

Data velocity indicates the rapidity of new data is being created or investigated [15]. Because of the existing rapidly increasing data level, it is very essential to handle the information in an effective and speedy way to reach satisfactory outcomes [16].

2.2.3 Variety

Variety refers to collecting different types of data from numerous varieties of data sources like smartphones, community networks, or any other data sources [20], [21]. These distinct data can be classified as semi-built, built, and unbuilt [22].

The diversity issue indicates that it is effortless to order the offered information in a significant manner, particularly when this data is varying rapidly [22].

2.2.4 Veracity

Veracity indicates assuring that the acquired data is precise. But if it is not precise, thus the final result would not be accurate, whereas decisions relay on dependable and true data [23].

2.2.5 Variability

Variability states to the meaning of data which is regularly and quickly exchanging [19].

It is completely different from the variety. For instance, in a giant company like Google or Facebook, the depot generates and stores some types of data. Also, if one of these various types of data is sent to be used for mining and to make sense, but different meanings provided by the data at every time [8], [19].

2.2.6 Visualization

Visualization represents the demonstration of complex data which could be logical and reasonable. There are certain means in Big Data sectors, which are applied for data visualizing and support the meaning to be understandable of different data. Because of the range of different data formats. built, semi-built, and unbuilt so it is not possible to scheme operative visualization means in Big Data[19].

2.2.7 Value

It is one of the critical features of Big Data. There are massive data sizes, as essential information is unseen surrounded by a bigger body of new data. Hence, the struggle is to distinguish and recognize which is the valuable thing in this data [16], [19].

2.3 COMMON OVERVIEW OF BIG DATA FRAMEWORKS ARCHITECTURE

A framework is a software offering a basic functionality which could be selectively replaced by the user [24].

The principle of the framework in Big Data likes the traditional data handling

frameworks, but traditional data technologies do not act that can treat the storage and analysis of massive and developing sizes of various data any more today [20].

Several frameworks in Big Data such as Apache Flink, Apache Hadoop, Apache Storm, Apache Spark, etc. These frameworks have developed to be common in the Big Data extent since they can make assessments in a circulated approach over a large number of machines and give user permits for estimations with a simple API (Application Programming Interface). Certainly, these frameworks can manage some significant features such as agility, flexibility, security, efficiency, and scalability. Accordingly, the framework in Big Data should include all these structures necessary to accomplish estimations with superlative outcomes. Aforementioned, there are many frameworks used in Big Data, whereas each one has a different structural design. Some of them are designed to resolve particular cases, for example, in a health ministry there is a framework named Smart Health [25].

It is used to help in design depending on the specific features of the health sector. Likewise, there are some of the other frameworks, such as Apache Samza [26], Apache Tez [27], Apache Calcite [28]; depend on general-purpose.

In spite of, the distinct resolution frameworks that do not build upon general-purpose frameworks require to undertake different sorts of Big Data capacity. Usually, frameworks in Big Data are created on some solid ideas such as spread storage, considerable parallel processing, and fault-tolerant and accessible systems. Therefore; it is essential to refer to a framework that contains different service layers for data generation, analytics, processing, storage, and acquisition. Fig.2.2 shows the common architecture of Big Data frameworks.

Figure 2.2: Common architecture of Big Data frameworks

2.3.1. Layer of Data Formation

The formation of data represents the first Layer in a Big Data framework. Data formation is a very miscellaneous, wide-range, and complex datasets created by

lengthwise and circulated data sources such as videotapes, clickstreams, sensors, and several data sources [29].

This Layer is the first phase for the data passing from different sources to start its trip. This Layer is the first step for the data coming from various sources in order to begin its trip. The prioritization and classification of data here enable the flow of data to move to another Layer easily.

2.3.2. Layer of Data Obtainment

The data passing from a different source called Data Obtainments. This Layer conducts with the automation of data absorption [30].

This Layer obtains every kind of data, as well as social information, log file, and Web data. The data that has been collected can contain considerable excessive or inadequate data, that enlarge storage size unreasonably and impact on the analyses of succeeding data. Similarly, the great increase is familiar among datasets gathered by sensors for environment monitoring and this Layer requires processing with data, whether this data is real-time or as a batch.

2.3.3. Layer of Data Storage

The data storage Layer manages computer technologies for continuous data storage in Big Data structures. It offers perfect support for dissimilar data, containing unbuilt, built, and semi-built data. Because of the involvement and variety of existing data, the structure of the storage Layers needs to use technologies to keep all existing data. MongoDB and Cassandra are the best examples of databases that can comply with the requirements of design for both storage services and administration.

2.3.4. Layer of Data Management

The data management Layer contains computer technologies which are involving in data management. Besides, a particular framework in Big Data, that could carry out data processing relying on either the data is in a stream or batch format. The mechanism for batch processing distributes tasks into minor ones and drives them to particular computing resources for processing, this approach is used in Apache Hadoop’s Map Reduce. Additionally, Stream processing needs instant processing of the information and requires the system to respond to new information [31].

2.3.5. Layer of Data Analytics

The main role of this Layer is to derive the significant information which is already created by the previous layer that contains batch processing and stream processing. The analytics layer is the process of inspecting enormous and different groups of data in order to detect the invisible outlines, anonymous relationships, and other appropriate information which can support the organizations in making applicable decisions for several objectives. This Layer can be distinguished between accessible and inaccessible. Accessible analytics is used for real-time situations which need a low response for outcomes, but inaccessible analytics regularly uses batch processing for change and absorption. The key purpose of this Layer is supplying an easy method to help the user to get on the value from the difficult information.

2.4 SCHEDULING OVERVIEW

We can clarify Scheduling as a method to give tasks to a certain resource, entitling one of the essential fragments of data management. The key goals of scheduling in Big Data platforms are decreasing the time of completion, boosting the output, also it dominates on existing resources of a circulated application, this is can be done by assigning jobs perfectly to the CPUs/PCs [32].

The scheduling algorithms are formulated and applied for sustaining the applications in environments of Big Data processing. Many tools are employed in Big Data for the distribution of resources, wherever everyone has different architectural features. Worthwhile to mention, that the scheduler is able to be functional based on the workload type and the bunch. Thus, these jobs need to be managed in an adequate manner which involves the lowest job-emptiness and highest advantage from the resource. The scheduler that applied in this technique for scheduling job subordinates on the used algorithm techniques [33].

As a result, many scheduling algorithms have been applied in Big Data platforms, like first in first out scheduling (FIFO), fair scheduling, capacity scheduling, Longest Approximate Time to End (LATE) scheduling, deadline constraint scheduling, and adaptive scheduling. These algorithms have different structures and techniques for scheduling jobs. Actually, the scheduler owns common features which could be definitely influenced on the performance of certain features such as scalability and elasticity, wide-ranging objective, dynamicity, objectivity, time proficiency, and

capacity adjusting. For a more effective scheduler, these characteristics should be observed. Nevertheless, that will be difficult to discover the perfect manner for scheduling.

2.4.1 Scheduling Essentials of Big Data

Indeed, the mutual requirements for scheduling of Big Data frameworks outline the practical and non-practical conditions for a scheduling function. The general requirements are defined below:

2.4.1.1. Flexibility and scalability

Scheduling needs to observe a lot of processors that are participating in assignments managing. The scheduler must be responsive to the modifications in the implementation setting that used to be adjustable to the capacity modifications through supplying or de-supplying the resources.

2.4.1.2 All-Purpose

It needs to be concerned about the scheduling, whereas there are certain constraints to several types of applications which can be implemented. The scheduler must sustain different types of jobs with perfect activity. For instance, a non-interactive batch job demanding great productivity that might prioritize time-sharing scheduling; like, a real-time job demanding short-time reactions prefers space-sharing scheduling.

2.4.1.3 Functionality

The scheduling algorithm has to employ the entire existing resources and must be able to adjust its behavior to manage if there are a lot of processing functions. Thus, the scheduler requires to adjust itself to resource availability changes always.

2.4.1.4 Authenticity

The resource sharing process by different users in a rational manner which assures the accessibility of each user to the existing resources based on call.

2.4.1.5 Proficiency of duration

It is so necessary for the scheduler to develop the activity of programmed tasks relatively. The scheduler needs to apply multi-task methods to handle several groups of data that are related to several involved users concurrently, whereas the jobs planning

and resources have to be enhanced in their application.

2.4.1.6 Value of cost

The scheduler needs to decrease the complete value of the cost of implementation through reducing the total of the applied resources. Anyway, the implementation boost for the determined number of jobs using a great activity of queuing structure also the approximation and communication costs are decreased accordingly.

2.4.1.7 Adjustment of capacity

This mechanism is applied to balance the capacity among the exited resources. It can be regarded as an inspiring condition when certain resources are not coinciding with job features. Either old-style approaches such as round-robin scheduling, or, new methods have been suggested to overcome on a significant and various systems; these are the smallest connection, slow start time, and factor-based adjustment that adopted [34], [35].

2.4.1.8 Status of location

The status of location is regarded as the dimension between the job knot and input knot. The transmission average of data is based on the status of the location. The transmission period of data will be concise if the computational knot is close to the input knot. Generally, the scheduling guidelines attempt to appointing assignments, that need to be closer to the input data knot in order to keep the cost of the network at the same time minimize the communication cost.

3. COMPOSITION EXAMINATION

Big Data is an extensive research field so far in which several investigations have occurred. Thus, the objective of the current work, a systematic examination of the composition needs to be taken place. We will present a summary of the further major reviews which have been accompanied by the job scheduling approach. The aforementioned examination will be carried out on various reviews that concentrate on job scheduling in Big Data. The next section will point out on the comparison among scheduling algorithms, while Section 3.2 will examine the scheduling algorithm enhancement in frameworks.

3.1 COMPARISON OF COMPOSITION ANALYSIS

There are many studies concerning Big Data, however, for the intention of comparison or assessment of several types of scheduling, there is only a limited number is applicable. A revision is prepared by Gautam and his classmates. The writers had completed an examination on the scheduling algorithm for managing the Big Data by matching algorithms which are applied in Apache Hadoop also submitting the concerns which affect job scheduling. Moreover, they finished a categorization to be manipulated based on various constraints such as resources, precedence, and duration. The writers examined many mechanisms which are applied to control the restrictions of Map Reduce additionally, how monitoring means or frameworks of some resource can be supportive in completing optimum outcomes. Contrarily, some studies on scheduling algorithms which are applied in Hadoop Map Reduce. Further research achieved by Pakize [36] that shows an outline of the Hadoop Map Reduce and examines related matters that influence on schedule. The writer gathered 15 categories of general scheduler algorithms, examining and sorting these algorithms.

Furthermore, the main objective of this classification in order to be familiar with the advantages and disadvantages of each scheduler algorithm. For researching the concerns that impact Big Data scheduling, Yoo, and Sim [37] finalized an assessment of the common group of issues in Map Reduce such as, objectivity, synchronization, and status of the location. The writers accomplished a study to conclude which schedulers are approaching the related issues.

Likewise, they completed a comparison of these scheduling techniques and showed the variances between them based on the strengths and shortcomings of each. One more

study by Bone and Wade [38] finalizes an assessment on scheduling which is applicable in Hadoop Map Reduce. They explained the main problems which are influencing using 3 kinds of standards specifically, 1) examining the advantage, 2) examining the disadvantages, and 3) contrast.

Further study is done by Tragi and Sharma [39] which examined the scheduling algorithms that applied in Hadoop additionally, different other methods are used for the escalation of the scheduler in Map Reduce. Also, a created sorting for different methods and schedulers is applied. There are other relative studies on various scheduling algorithms in Hadoop which have been completed by Sharma and Malhotra [40].

Whereas the writers display the essential elements of Hadoop along with that comparing three kinds of scheduling, called FIFO, Fair and Capacity scheduler in order to estimate their activity. Well along, the writers suggest constraints which could develop the activity of Hadoop. Besides that, further study is accomplished by Karambir and Rani [41].

anywhere the study is presented as to the Hadoop part and other parts possibly integrated with Hadoop through a comparison on the scheduling algorithms which are involved. Several studies in the composition that inspect scheduling algorithms, but there are few classify these algorithms. According to Singh and Agrawal [42], schedulers are classified into five major types such as makespan-aware schedulers performance-manager schedulers, resource contention-aware schedulers, energy, and speculative execution-based schedulers, and data locality-aware schedulers. These schedules have been created based on the requirement to apply some scheduling metrics such as job response duration, job execution, output, power, workload, and time. Furthermore, Nidhi Tiwari et al. [43] sort Map Reduce algorithms based on features like the algorithm's flexibility to the worktime environment, the entity scheduled, and the quality attribute. Moreover, an analysis of scheduling algorithms is conducted as intended for different quality characteristics by sorting them depending on the identical objects that they schedule. The grouping suggested by the aforesaid guides forthcoming research to develop improved algorithms. Arpitha and Shoney [44], examined the Hadoop scheduling and its primary characteristics which influences on the tasks of the scheduler. The study discusses various characteristics of scheduling algorithms, that called fair scheduler, deadline constraint scheduler, FIFO scheduler,

LATE scheduler, dynamic priority scheduler, delay scheduler, capacity scheduler, taxonomy, and escalation depending on the scheduler for miscellaneous Hadoop (COSHH) Scheduler. They prepared an evaluation to find out the differences in these scheduling methods based on both the miscellaneous and identical setting, as well as revealing the strengths and weaknesses of each scheduling algorithm. By Mohamed and Hong [45] efforts, they finalize a review of Hadoop Map Reduce job scheduling algorithms however the concentration was on the HDFS and Map Reduce. They consider the Hadoop component and define the aspects that influence the scheduling jobs. Moreover, they submit justifications for algorithms so far which are not incorporated in earlier comparative studies.

Likewise, Nagina and Dhingra [46] showed Hadoop scheduling algorithms in two classifications: inbuilt and user-defined and prepared a study and of each type as applied in Hadoop. They designated the techniques which can be employed for these scheduling algorithms. Additionally, Patel [47] focuses on the related information on Hadoop Map Reduce and what is the suitable design to be used. He explained that there are different scheduler algorithms used in Hadoop, whereas each mechanism has problems which are so important to be resolved. Also, a comparison is finalized about the algorithms which are using different features. There are particular categorizations of schedulers that established on their contracting features, effectiveness, performance, etc., as aforesaid. Still, we sort, match and examine the in-depth information related to many schedulers being used in Big Data Apache Flink. Furthermore, this review will define the strengths and weaknesses of these frameworks according to their applications. Besides, the study looks at scenarios for the convenience of issues for defining which scheduler will be unworkable or inadequate. Thus, such issues we will try to include them in this thesis are not researched in the said examinations.

3.2 SCHEDULERS ENHANCEMENT

Essentially, Hadoop has 3 kinds of schedulers: Capacity, FIFO, and Fair, that cannot fulfill all of the specific organizations. Thus, some of the scholars have created certain developments on schedulers by recommending notable approaches. Job Schedulers represent a substantial section of Hadoop, and scholars have proposed several scheduling techniques under certain frameworks to develop scheduling jobs. Concerning the original schedulers, Tao, and his colleagues [48] suggest a technique to develop the Fair scheduling algorithm through data locality and job character. This

technique permits the tasks to be absolute for new users and could be convenient for numerous policies for CPU-bound and I/O bound systems. The tries of papers to expand both implementation time and transmission of data. Correspondingly, Santhosh and Hemanth [49] recommend task tracker-aware scheduling, that supports the user's capacity to design and carry out re-run tasks in another slot in case of insufficiency. Besides, the users can design maximum loads for every task tracker. The last consequences offer resolutions for scheduling algorithms and pass the restrictions by getting on more control job execution by the user. Further issues regarding schedulers take account of the diversity of the hardware within clusters as deliberated by some writers. Zaharia et al. [50] had proposed LATE scheduler. Essentially, this technique highlights on an intellectual carrying out of tasks. As a result, some cases may fail or slowly handled. LATE scheduler discovers the task which is working in slow motion and creates another task for it as an alternative task. The key indication of this scheduler will enable the user to completed his assignment as fast as possible. What is more, there are also other reviews on scheduling in different settings. Rasoolia and Down [51] recommended COSHH, that represents an abbreviation for Classification and Optimization-based Scheduler for Heterogeneous Hadoop Systems. In this technique, the scheduling choice is created as applicable to the scheduling method for application and based on existing resources and the number of assignments. The key indication of the COSHH scheduler is using the framework data to accomplish well scheduling in order to develop Hadoop's scheduling activity. Definitely, the COSHH consists of two main processes, where every process is triggered by receiving one of these messages. Upon receiving a new job, the scheduler performs the queuing process to store the incoming job in an appropriate queue. Upon receiving a heartbeat message, the scheduler triggers the routing process to assign a job to the existing free resource. Besides, The COSHH scheduler algorithm is fully adjustable to any difference in the constraints in the framework. The sorting portion identifies variations and adjusts the classes depending on the different constraints.

For the purpose of resolving the location of the data issue, scholars have suggested other options. The location of data consequence in decreasing the network traffic which boosts the scheduling activity is one of these improvements. Zaharia et al [52] had recommended new techniques for boosting the location of data and equality for the Fair scheduler algorithm. They had deliberated, in their analyses, Delay scheduling whereas a discrepancy between the location of data and fair scheduling. This is proven by

tentative examinations, and an algorithm is proposed for approaching and determining this problem. By using this technique, before an assignment is designated to the data, it will be reviewed if the data is appropriate to assign the assignment. If not, task assignment will be postponed until there will be a free slot is available for the required data. To reduce non-local task carrying out, Xiaohong et al. [53] recommended the Next-k scheduling (NKS) algorithm and created and employed a new technique for elaborating the location of data in MapReduce for the different computing setting along with the load of the network on the cluster. The technique is not only elaborating on the location of data, the load of the network on the cluster. On the contrary to these deeds, He and his classmates [54] submit a procedure of scheduling that concentrates on elaborating the location of data and typical reaction period. The core aim in this scheduling is providing all enthralled knots an identical attempt to snatch the resident assignment beforehand. One more suggestion by Kondikoppa et al. [55] structures network-aware scheduling planned and implemented for Hadoop. This scheduling gives control to the administrator in order to find out which hub has the information needed for each task. In this scheduler, up to a certain extent, the location of data issues can be resolved. When tasks are assigned to the task tracker, the information needed to finish this task is reviewed to be adequate for the task. One more study by Kc and Anyanwu [56] includes deadline-restrictions schedulers and the possibility of developing them in terms of a framework by processing the data and arranging the deadline requirement. Thus, data processing is done by cost pattern job implementation and Hadoop scheduler with restrictions. For the cost pattern, there are many constraints such as the information size, data allocation, etc. Besides, Hadoop scheduler with restriction has a deadline part, which is different from the information. When the job is provided to the Hadoop scheduler, it verifies if the assignment could be completed based on the period set by the deadline or not. This scheduler focuses on collective system use. For boosting capacity scheduling, some studies are completed by Thakur et al. [57], in which the writers deliberated Dynamic Capacity Scheduling in Hadoop and suggest a new method to increase capacity schedulers through presenting what are the issues that make schedulers more effective for implementing tasks in less time. Besides, they deal with a pipeline and queue management in their research for improving the activity of Hadoop. Additional work, such as Sandholm and Lai [58], offers dynamic Priority Scheduler, whereas the algorithm provides the capability to manipulate the assigned capability. The essential technique provides an ability to prioritize then it supports the users with tools to modify and improve their distribution

in order to be appropriate for the requirements of their tasks.

4. SCHEDULING IN FRAMEWOR

4.1 APACHE FLINK

Apache Flink is an open-source framework used for distribution and processing and provides data distribution, communication, and fault tolerance for distributed estimations over the data stream [3].

Apache Flink is a hybrid and has the ability to process both data streams (stream) and batch data (dataset). The main core of Apache Flink is a stream, but the batch processing is a special class of this application. The Flink cluster consists of two important categories job manager and job tracker. The job manager, or master, plays an interface with the user application. It arranges the tasks received and sends them to the workers. The job manager orders the coordination of resource management and job scheduling. The real data processing happens by the task managers (worker). The task manager has one or more slots used in parallel for executing jobs. The job manager is the coordinator to distribute the implementation of the data stream. It can notice that in Fig 4.1.

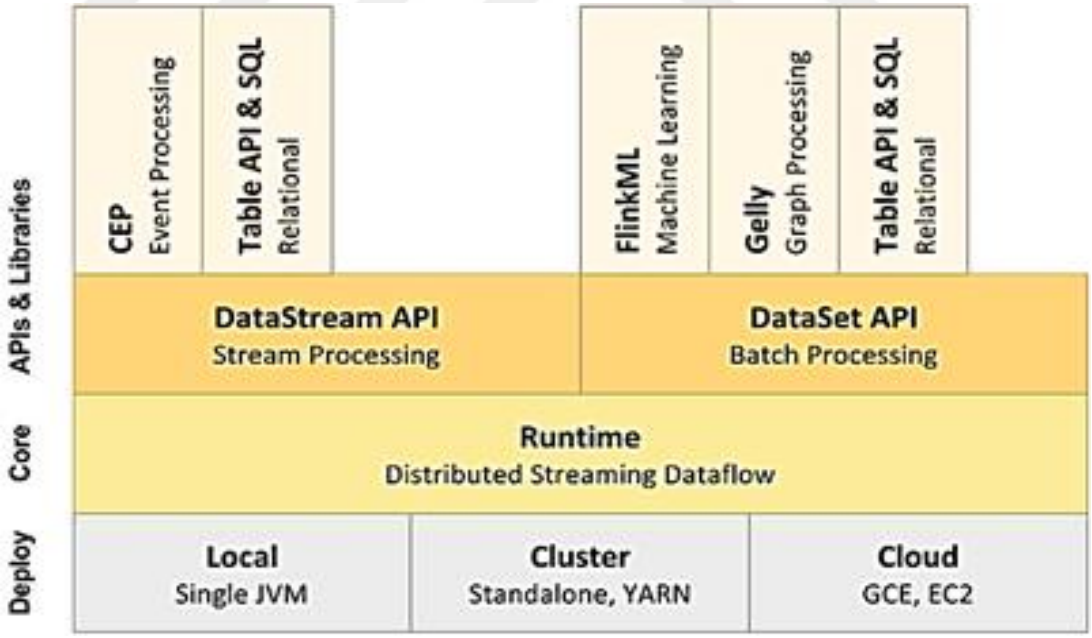


Figure 4.1: Apache Flink stack [59]

The job manager is able to manage and observe the progress for the operator as well as the capability to schedule new operators and establish checkpoints and recovery. In great accessibility, the job manager continues a minimal group of the metadata for each checkpoint for tasking tolerant storage, like the standby of job manager capable of

restoration the dataflow implementation from there. The real data processing is executed by the task managers, which implement many operators and send the report of condition to the job manager [60].

Apache Flink has its own local cluster (single JVM) that can procedure the data also or can be set up to another cluster manager such as (YARN, Mesos, and Standalone) along with, it is able to operate on the cloud (Amazon or Google cloud). Diagram 4.8 shows the Apache Flink structural design.

4.1.1 Standalone- Apache Flink

Apache Flink can deploy different cluster managers with a simple independent, which is a cluster manager. The default scheduler algorithm used in standalone is FIFO and works by processing jobs as a queue, the job that approaches first will be managed first, and so on. This scheduler algorithm is careless about the size or any other precedence, also when the job comes to queue for managing, so only this job is permitted and the other jobs postpone until the first one is completed the implementation. FIFO scheduling is used when the order of job implementation is insignificant. Furthermore, each application procedure all available knots, whose number bases on applications and the setting configuration. A standalone cluster is capable to launch by hand through beginning the master and worker manually, or by using the launch scripts that provided.

Strengths

- FIFO scheduling technique is easy to execute.
- The Job implements in the same order that has been presented.

Weaknesses

- Jobs is suspending until the managing ends for the jobs implement which waits for all jobs if it is long-running operation.
- FIFO Scheduler is a non-adjustable resource between jobs whether it is a long or short job.
- FIFO scheduler is not covering the conception of the priority or size of the job.

Suitability

- It is adequate for performing jobs in order based on the coming significance.

- It is sufficient for a single type of job because of the using FIFO scheduler and the well-arranged processing.
- This Cluster Manager meets to assure that there is no single point of failure and programs are capable to progress as soon as a standby Job Manager has occupied management.

4.1.2 Apache Flink on YARN

The most essential characteristic of the Apache Flink is it capable of incorporating with Hadoop YARN. When the Flink runs on YARN, resource management and scheduling are organized by YARN and the scheduler is connectable. The YARN has three key scheduler plug-ins, called FIFO, capacity scheduler, and fair scheduler. These schedulers manage Flink like any other application running in YARN and allocate CPU and memory according to their reasoning.

4.1.2.1 FIFO Scheduler

The default scheduler in Hadoop is FIFO [32], [36]-[61],[64]; that treats jobs on a function basis. Otherwise, whichever task is on the front is implemented, it is followed by the rest coming to their functions and ongoing to the point no task is left unfinished [65].

As shown in Fig.4.2. All that is important for jobs to derive to function, be processed and the scheduler proceeds with the next. By doing so, it turns into perfect that priority and size or order based on other elements have no principle in the FIFO scheduler [13].

FIFO scheduling is used when order implementation of jobs has insignificance.

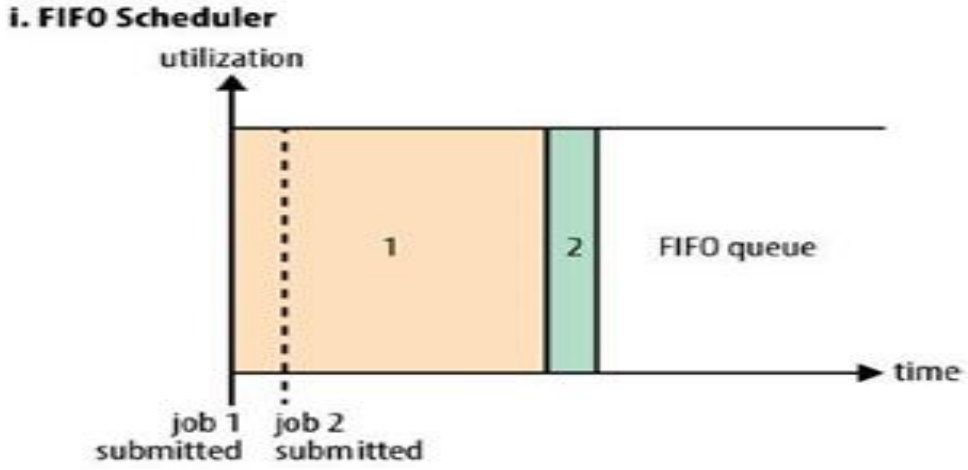


Figure 4.2: FIFO scheduler work

Strengths

Simplicity in practice and practicability in all schedulers.

The Job implements in the same order that has been given.

Weaknesses

- FIFO scheduler is not preventative.
- Notions such as size and jobs length are not essential in FIFO scheduler.
- Unsteadiness in resource assignment.
- It works like one by one task basis, thus requiring extended processing time should some tasks delay for too much time.

Suitability

- Appropriate for implementing orderly tasks depending on the onset priority.
- Satisfactory for one kind of job.

4.1.2.2 Capability scheduler

This was initially introduced by the Yahoo company [32, 66] in conditions that immense clusters are to be used by different enterprises [67].

The key objective of this scheduler is to improve output thereby assure a collection of users have entirely controlled access to resources [68].

This scheduler uses a queue technique, whereas each is allocated with necessary resources to a company. Given the characteristics of security is first and foremost, only the selected organizations can access only one queue, that make sure the queue will never be accessed by another one. Simultaneously, reliable control procedures are adopted task operations to allocated resources mannerly. If there are any new jobs reach the queue, resources are allocated back to the former queue after completing the present running jobs. The configuration of precedence is critical in this scheduler. And more, graded as hierarchic queues are implemented so that sub-queues receive resources before other recently arriving ones that might remove these resources [67].

As shown in Fig.4.3.

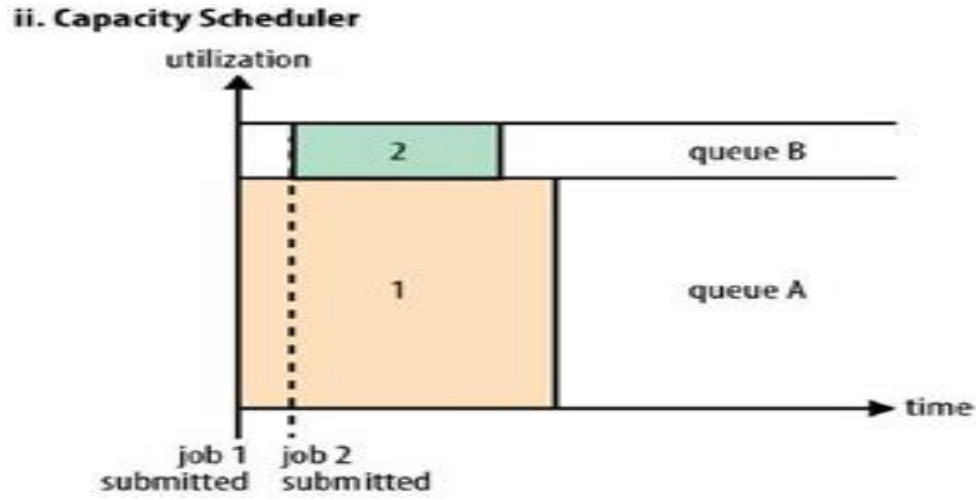


Figure 4.3: Capacity scheduler work

Strengths

- Competency of scheduler assures re-employ of the ability otherwise it is inapplicable by the queues.
- A decrease in expenses.
- Resources usage in cluster environments is increased.
- Capability scheduler can sustain several feathers such as security, multi-occupation, hierarchal order, and resistance.

Weaknesses

- It is hard to select suitable queues.
- Capability scheduler is very hard to compare with other schedulers.
- Capability scheduler has some restrictions on pending jobs to guarantee equality of cluster from a queue, stability, and single user.

Suitability

- Proper when priority is vital.
- The capacity scheduler is fit for many concerned users. Thus, it is demanding the distribution of resources.
- It is suitable for similar settings.

4.1.2.3 Fair Scheduler

This scheduler is established by Facebook [32, 67, 69], it is depending on assigning regular extents of resources among the jobs [69, 70].

Fair scheduler arranges jobs into groups and assurances fairness in sharing resources among the jobs in groups [42].

In addition to that groups manage job configuration features and configuration attributes decide the group in which the job is located. As a result, all users have their own groups with the minimum share assigned to each one, and the resources are shared in reasonable extents among the groups [71].

In the case of open slots, they can be used by other groups [72, 73].

Should there be a user or group that sends many jobs excessively, algorithm obstructions, and shifts them non-runnable [13].

If there is a single job in the process, it obtains on the complete cluster [74].

Furthermore, a reasonable scheduler facilitates many other properties for example scales on queues and ultimate shares. As in Fig.4.4 below.

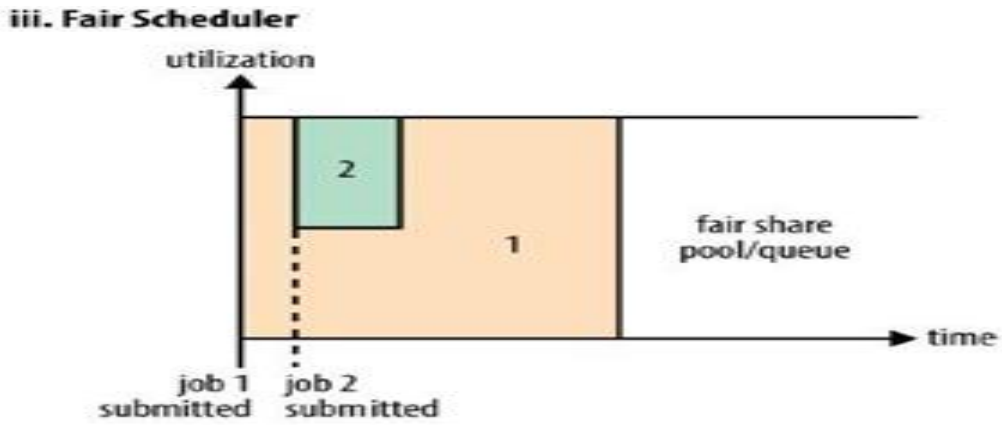


Figure 4.4: Fair scheduler works

Strengths

- It is appropriate for short jobs instead of larger ones.
- The scheduler makes sure from reasonable allocation of resources.
- It is less complicated.

- It is can control synchronous jobs among all users and groups.

Weaknesses

- It is running only a limited number of tasks simultaneously.
- It has many problematic configurations.
- It disregards the job scales for each knot, leading to unequal operation at the knot level.

Suitability

- The reasonable scheduler is sufficient to ensure that emptiness will not occur.
- It is adequate for sharing resource clusters among users.
- It is sufficient to administrate users to introduce a lot of jobs.
- Fair scheduler is adequate for an equivalent setting.

5. DISCUSSION

The essential objective of this study is to analogize and examine the comprehensive information related to some schedulers being used in Big Data frameworks. What's more, it will pinpoint the weaknesses and strengths relevant to each scheduler in several applications. There will be a detailed discussion on these issues from a different perspective. This study represented four of the Big Data frameworks, specifically, Apache Flink, jointly analogized and with regards to different job scheduling settings since each framework uses different mechanisms and constructions for scheduling jobs.

5.1 DISCUSSION AND RESULTS

Aiming to our objective, five-research queries were worked out in UNIT 1 and, far along, answered in CHAPTER 4. Therefore; we use the systematic literature review (SLR) methodology, which is the clarification and estimation of research related to any particular topic.

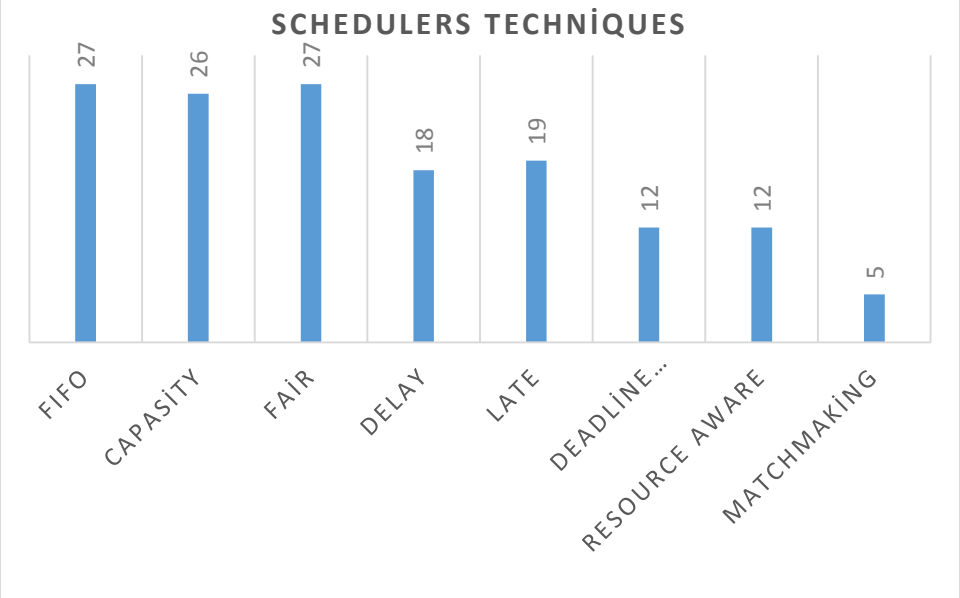
SLR is entirely depending on supplementary studies, such as scholarly articles, research papers, and published theses. Here, a summary is provided as to the results and the analysis of the answers to the research questions.

- **RQ1. What is the most scheduling technique used in big data? And why it has the most important compare to the others?**

  Task scheduling is a significant feature of processing great data groups and is responsible for organizing jobs to be allocated to the available resources. There are many algorithms for task scheduling, where each one applies a different technique for the purpose. The key task for the scheduler is to decrease assessment time and maximize the use of the resources. For scheduling jobs in Big Data, there are distinct cluster managers that are employing the scheduling algorithms. The most standard one being FIFO that has applications in different frameworks as a default scheduler algorithm, such as in Apache Flink and Standalone. This scheduling technique is also simple and effective when compared to other schedulers available in the Big Data area, regarding its effortlessness to configure and implement. As shown in Chart 5.1 which explaines the occurrence of studies on the scheduler in Big Data frameworks. Chart 5.1 shows the studies in which scheduling algorithms have been applied

the best in Big Data frameworks.

Chart 5.1: The occurrence of studies on the scheduler in Big Data frameworks



- **RQ2. Which type of scheduling techniques are applied in different Big Data frameworks?**

  There are various frameworks in Big Data, each with its own techniques for scheduling jobs. In CHAPTER 4, we clarified the algorithms used in different frameworks. As already mentioned, there are some algorithms applied in these frameworks with an exclusive technique to schedule jobs; for example, FIFO runs by processing jobs as a queue, in other words, whichever job comes first will be processed first, and the cycle continues.

On the other hand, Fair scheduler approach arranges jobs into groups and assures objectivity in sharing resources between the jobs in groups. Similarly, each framework practices different techniques and structures for scheduling jobs, which must be processed in such a way that causes minimum job-emptiness and extreme advantage from the resource. In this technique, the scheduler is employing for scheduling job dependents on the technique of the algorithm used. Oppositely, Capability scheduler applies a graded as hierarchic queues technique. Thus, each queue is assigned to an enterprise, and resources are allocated among these queues. Similar to the FIFO scheduling techniques, Delay scheduler uses a queue technique for scheduling jobs to designate tasks to the knot; the delay algorithm begins the

search at the first job in the queue for a resident task. If not effective, the scheduler postpones the job's implementation and searches for a resident task from subsequent jobs. If a job for a long time, then the scheduler will be allocated it to nonresident tasks. As well, Deadline Constraint scheduler uses queues for scheduling jobs for implementation and has a priority queue of the jobs organized by their deadlines. Tasks scheduling is first tried for jobs in front of the priority queue. The scheduler is checked if this job can be done within the time indicated by the deadline. If so, the job can be scheduled, if not, it will fail. Matchmaking scheduler applies a queue technique, moreover, it gives the same way to all knots to yield a resident task. It is used to mark knots and confirm that each one gets a reasonable opportunity to snatch resident tasks. When a knot collapses to find a resident task in the queue for the first time, so the nonresident task will be designated to the knot. Uncertainty the knot collapses to find the resident task again, the scheduler will allocate a nonresident task to prevent losing out the resources. Contrariwise, Resources-Aware scheduling focuses on using many resources such as input and output, memory, CPU network, and disk usage.

There are two approaches for Resource-Aware scheduling, called slot filtering and dynamic free-slot advertisement, wherein the latter case, because of having a stable number of obtainable estimation slots configured on every knot, this number is considered dynamically using resources metrics got from every knot. For the second approach, Free Slot Priorities/Filtering, the supervisors of the cluster will configure an extreme number of computed slots per knot at the configuration duration.

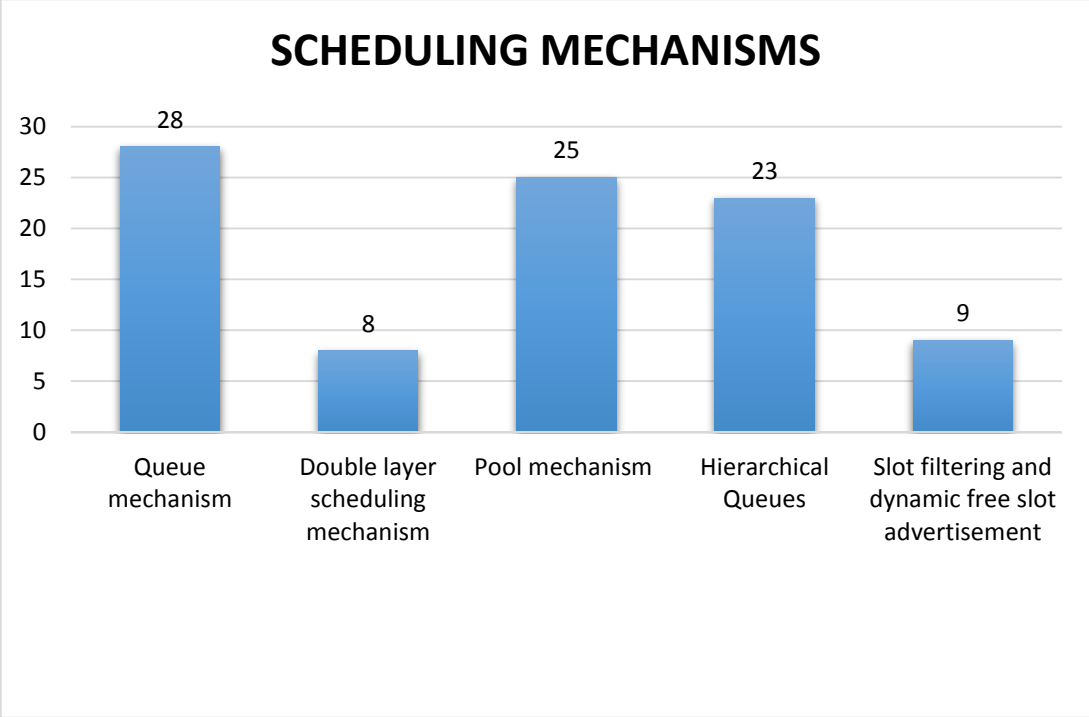
In this method, the order in which the free slots are announced is ascertained according to their resource accessibility. In an environment where small tasks usually pick up a quite long-time to complete, this method achieves the main advantages in terms of operation. Likewise, LATE scheduler focuses on enhancing the hardiness and reduces the expenses of tasks in the cluster. The key objective of LATE scheduler algorithm is to develop the operation of the jobs and diminish the reaction time. LATE discovers which procedures are operating lazily in a cluster and, then, makes a correspondent process in the background as a backup. The common scheduler algorithm applied in these frameworks is deliberated in detail in CHAPTER 4. In this part, these scheduler techniques are listed below.

- Group technique

- ❖ Queue technique
- ❖ Double layer scheduling technique
- ❖ Hierarchic queues
- ❖ Slot filtering and dynamic free slot announcement.

Chart 5.2 displays the number of studies that mentioned the scheduling techniques which are applied in various Big Data frameworks.

Chart 5.2: Frequency of studies on scheduling mechanisms are being used in different Big Data frameworks



RQ3. What is the most case that will scheduling algorithms are being appropriate?

This study recognizes the weaknesses and strengths of different usage of these frameworks. Besides, the study inspects scenarios for the appropriateness of cases so as to decide which case scheduler will be appropriate. These issues are included in this study which is not revealed in the said studies in CHAPTER 3. Each scheduling algorithm is discussed separately and in full in CHAPTER 4. In Table 5.1 summarizes the most suitable cases of scheduling algorithms.

Table 5.1: Appropriateness cases in different scheduler algorithms

Appropriate	FIFO	Fair	Capacity
Small Clusters	✓	✓	
Identical Setting	✓	✓	✓
Sharing Resource		✓	✓
Seniority			✓
Large Cluster		✓	✓
Quick Response Time		✓	
Increases The Resource Usage			✓
Separation			✓
User Management		✓	
Productivity			✓

- **RQ4. What is the major strengths and weaknesses of different types of schedulers Algorithms in Apache Hadoop and Apache Flink?**

The main objectives of scheduling in Big Data platforms are to diminish the completing time, as well as magnify the productivity, manage existing resources of an assigned application by properly distributing jobs to the CPUs or computers. The scheduling algorithms are created and executed to support applications in Big Data processing environments. Many techniques are used in Big Data for resource assignment, whereas each one has different structural appearances. A scheduler can be effective based on the type of workload and the cluster. The strengths and weakness are entirely deliberated for each scheduling algorithm in CHAPTER 4, but in this part, we will outline more frequently strengths and weaknesses of the scheduler is magnifying the productivity, support separation containers, guarantee a reasonable assignment of resources, moderate overhead, enhancing the status of the location of data, ensure there is no fault in the task, also grant the user's security. On the contrary, weakness

of the scheduler is hard configurations and executions, slow progress for the tasks, unable to support resource-balancing, and the possibility of the emptiness. In Table 5 summarizes the strengths and weaknesses of the scheduler in Big Data frameworks.

Table 5.2: Strengths and weaknesses of the scheduler in Big Data frameworks.

	Cases	FIFO-Hadoop	Fair-Hadoop	Capacity-Hadoop	Default scheduler
Strengths	Figures out locality problems	✓		✓	
	Easiness of scheduling.	✓	✓		✓
	Reactive to small jobs matching with the big jobs		✓		
	Flexible to new modifications		✓	✓	
	Figures out the slow work of the extended jobs in a cluster		✓		✓
	Equips high cluster employment			✓	
	Cluster allocation between institutions.			✓	
	Regular implementation for the submitted order	✓			✓
	Stretch the security			✓	
weakness	non-preventative	✓		✓	
	Implementation for one task while other tasks are pending	✓			✓
	Nonequivalence of resources	✓			✓
	The volume and the precedence of the job are not prioritized	✓			✓
	Problematic in configurations		✓	✓	
	The slowness in fine-grained mode than in the coarse-grained mode	✓	✓		
	Only action is taken on suitable slow tasks.	✓			
	Consumed time in locking at identical data	✓			✓
	nonadoptive with recent modifications		✓		
	Non-fullness	✓			

6. CONCLUSION AND ANTICIPATED GUIDELINES

6.1 CONCLUSION

This thesis is attempted to introduce scheduling jobs in four frameworks, called Apache Hadoop, Apache Flink, and Apache Storm. Due to the examination of the papers from the SLR, we can say that Big Data obtain various scheduling algorithms within different frameworks. In this thesis, 79 papers were gone through to pinpoint the most significant elements influencing on the carrying out of the suggested schedulers. Indeed, schedulers have common features which could definitely have an impact on the productivity of certain features such as adaptively and flexibility, wide-ranging resolution, dynamicity, objectivity, time effectiveness, and balancing of workload. These schedulers can affect on the framework's performance. The key goals of scheduling in Big Data platforms are decreasing the completing time, magnifying the productivity, and manage the existing resources of an allocated application by well-assigning jobs to the CPUs/PCs. According to the 27 studies and analyses, the most common scheduler mechanism used in Big Bata frameworks is the FIFO scheduler algorithm. This scheduler provides different features such as effortlessness, functionality, and so on. What's more, it is used as a default scheduler in the most popular frameworks, called Apache Hadoop and Standalone, that's why this scheduler controls others in terms of popular use in Big Data frameworks. Besides, many other techniques are applied in Big Data for resource assignment, and each one has different structural appearances such as queue technique, group technique, slot filtering, and so on. The scheduling technique impacts on the managing method of the jobs, on the other side, the framework's performance is influenced.

These jobs need to be managed in such a manner that minimizes job emptiness and improves the advantage of the resource. Therefore, the most common technique used in Big Data scheduling is found to be the queue technique according to different schedulers such as FIFO, Capability, Delay, and many other schedulers. Oppositely, this thesis offers a complete indication of the proper cases for scheduling jobs in Big Data. We examined cases to represent which one is the most sufficient. As per the final results, different cases are adequate for each scheduling category. In particular, a scheduler may be suitable for priority, productivity, separation, resource usage, long-operating tasks, management of users, full work and balancing of load work, and so

on.

These cases make an influence on the functionality of the framework. Formerly a scheduler is most suitable for the case to be applied, it will decrease the estimation time and improve the resource's usage. In conclusion, this also deliberated the strengths and weaknesses of each scheduler as well as what the composition suggests to resolve the issues in each specific scheduler. The outcomes show that none of these scheduler algorithms can satisfy our needs, and there are different schedulers proposed in the composition to debug these deficiencies.

6.2 ANTICIPATED GUIDELINES

The following can be considered as potential research that can extend this research:

- The number of scheduler algorithm used in comparison need to be increased.
- Studying and investigating on the challenges that make an effect on the scheduling jobs.
- Conducting experimental research on scheduling in a Big Data framework.

REFERENCES

[1]Pop, F., J. Kołodziej, and B. Di Martino, *Resource Management for Big DataPlatforms: Algorithms, Modelling, and High-Performance Computing Techniques*. 2016: Springer.

[2]Storm, a. *Apache Software Foundation*. 2017; Available from https://www.apache.org/.

[3]flink. *Apache Flink* 2017; Available from https://flink.apache.org/.

[4]Friedman, E. and K. Tzoumas, *Introduction to Apache Flink: Stream Processing for Real-Time and Beyond.* 2016.

[5]Deshpande, T., *Learning Apache Flink*. 2017: Packt Publishing Ltd.

[6]Kitchenham, B., *Procedures for performing systematic reviews.* Keele, UK, Keele University, 2004. 33(2004): p. 1-26.

[7]Neuman, W.L., *Social research methods: Qualitative and quantitative approaches*. 2013: Pearson education.

[8]RASHED, A.H.R., *BENEFITS AND CHALLENGES OF BIG DATAON CLOUD FOR SMES AND GOVERNMENT AGENCIES*. 2018. p. 108.

[9]Creswell, J.W. and J.D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. 2017: Sage publications.

[10] Anagnostopoulos, I., S. Zeadally, and E. Exposito, *Handling Big Data: research challenges and future directions.* The Journal of Supercomputing, 2016. 72(4): p. 1494-1516.

[11] Tole, A.A., *Big Datachallenges.* Database Systems Journal, 2013. 4(3): p. 31-40.

[12] Salehnia, A., *Comparisons of Relational Databases with Large data: a Teaching Approach.*

[13] Usama, M., M. Liu, and M. Chen, *Job schedulers for Big Dataprocessing in Hadoop environment: Testing real-life schedule with benchmark programs.* Digital Communications and Networks, 2017.

[14] Chen, C.P. and C.-Y. Zhang, *Data-intensive applications, challenges, techniques and technologies: A survey on Big Data.* Information Sciences, 2014. 275: p. 314-347.

[15] Anuradha, J., *A brief introduction on Big Data5Vs characteristics and Hadoop Technology.* Procedia computer science, 2015. 48: p. 319-324.

[16] Uddin, M.F. and N. Gupta. *Seven V's of Big Dataunderstanding Big Datato extract value*. in *American Society for Engineering Education (ASEE Zone 1), 2014 Zone 1 Conference of the*. 2014. IEEE.

[17] Lee, I., *Big Data: Dimensions, evolution, impacts, and challenges.* Business Horizons, 2017. 60(3): p. 293-303.

[18] Data, T.V.s.o.B. *The 7 V's of Big Data*. 2017; Available from: https://www.impactradius.com/blog/7-vs-big-data/.

[19] Sivarajah, U., et al., *Critical analysis of Big Datachallenges and analytical methods.* Journal of Business Research, 2017. 70: p. 263-286.

[20] Karakaya, Z. *Software engineering issues in Big Dataapplication development*. in *Computer Science and Engineering (UBMK), 2017 International Conference on*. 2017. IEEE.

[21] Hashem, I.A.T., et al., *The rise of "Big Data" on cloud computing: Review and open research issues.* Information Systems, 2015. 47: p. 98-115.

[22] Erevelles, S., N. Fukawa, and L. Swayne, *Big Dataconsumer analytics and the transformation of marketing.* Journal of Business Research, 2016. 69(2): p. 897-904.

[23] Gandomi, A. and M. Haider, *Beyond the hype: Big Dataconcepts, methods, and analytics.* International Journal of Information Management, 2015. 35(2): p. 137-144.

[24] Tekiner, F. and J.A. Keane. *Big Dataframework*. in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. 2013. IEEE.

[25] Sakr, S. and A. Elgammal, *Towards a comprehensive data analytics framework for smart healthcare services.* Big DataResearch, 2016. 4: p. 44-58.

[26] Apache. *Apache Samza*. 2018; Available from: http://samza.apache.org/.

[27] TEZ, A. *ApacheTez*. 2018; Available from: https://tez.apache.org/

[28] Apache. *Apache Calcite*. 2018; Available from: https://calcite.apache.org/.

[29] Chen, M., et al., *Big Data: related technologies, challenges and future prospects*. 2014: Springer.

[30] Lehmann, D., D. Fekete, and G. Vossen, *Technology selection for Big Dataand analytical applications*. 2016, Working Papers, ERCIS-European Research Center for Information Systems.

[31] Karakaya, Z., A. Yazici, and M. Alayyoub. *A Comparison of Stream Processing Frameworks*. in *Computer and Applications (ICCA), 2017 International Conference on*. 2017. IEEE.

[32] Gautam, J.V., et al. *A survey on job scheduling algorithms in Big Dataprocessing*. in *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*. 2015. IEEE.

[33] Bhavsar Nikhil, B.R., Patil Balu,Tad Mukesh, *A SURVEY ON SCHEDULING IN HADOOP FOR BIGDATA PROCESSING.* Multidisciplinary Journal of Research in Engineering and Technology, 2015: p. 5.

[34] Li, K.-C., et al., *Big Data: Algorithms, analytics, and applications*. 2015: CRC Press.

[35] Gaur, M., B. Minocha, and S.K. Muttoo, *A Study of Factors Affecting MapReduce Scheduling*, in *Big DataAnalytics*. 2018, Springer. p. 275-281.

[36] Pakize, S.R., *A comprehensive view of Hadoop MapReduce scheduling algorithms.* International Journal of Computer Networks & Communications Security, 2014. 2(9): p. 308-317.

[37] Yoo, D. and K.M. Sim. *A comparative review of job scheduling for MapReduce*. in *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*. 2011. IEEE.

[38] A.M.Wade, P.K.B.a., *Survey on Hadoop MapReduce Scheduling Algorithms.* Grenze Int. J. of Engineering and Technology, 2015.

[39] Adhishtha Tyagi, S.S., *A brief review of scheduling algorithms of Map Reduce model using Hadoop.* International Journal of Engineering Trends and Technology (IJETT), 2017.

[40] Anuradha Sharma, E.S.M., *A Comparative Review on Different Scheduling Algorithms in Hadoop.* International Journal of Innovative Research in Computer and Communication Engineering, 2016.

[41] Karambir, A.R., *Review Of Fair Job Scheduling Approaches For Map Reduce.* International Research Journal of Engineering and Technology (IRJET), 2016. 03(05).

[42] Singh, N. and S. Agrawal. *A review of research on MapReduce scheduling algorithms in Hadoop*. in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. 2015. IEEE.

[43] Tiwari, N., et al., *Classification framework of MapReduce scheduling algorithms.* ACM Computing Surveys (CSUR), 2015. 47(3): p. 49.

[44] Arpitha, H. and S. Sebastian, *Comparative study of Job Schedulers in Hadoop Environment.* International Journal of Advanced Research in Computer Science, 2017. 8(3).

[45] Mohamed, E. and Z. Hong. *Hadoop-MapReduce Job Scheduling Algorithms Survey*. in *Cloud Computing and Big Data(CCBD), 2016 7th International Conference on*. 2016. IEEE.

[46] Nagina, D. and S. Dhingra, *Scheduling Algorithms in Big Data: A Survey.* Int. J. Eng. Comput. Sci, 2016. 5(8): p. 17737-17743.

[47] Patel, H.M., *A comparative analysis of MapReduce scheduling algorithms for Hadoop.* Int. J. Innov. Emerg. Res. Eng, 2015. 2.

[48] Tao, Y., et al. *Job scheduling optimization for multi-user MapReduce clusters*. in *Parallel Architectures, Algorithms and Programming (PAAP), 2011 Fourth International Symposium on*. 2011. IEEE.

[49] Mr. Santhosh S, M.H.K.G., *IMPROVED FAIR SCHEDULING ALGORITHM FOR.*

TASKTRACKER IN HADOOP MAP-REDUCE. International Journal of Advanced Technology in Engineering and Science, 2015. 03(01).

[50] Zaharia, M., et al. *Improving MapReduce performance in heterogeneous environments*. in *Osdi*. 2008.

[51] Rasooli, A. and D.G. Down, *COSHH: A classification and optimization based scheduler for heterogeneous Hadoop systems.* Future generation computer systems, 2014. 36: p. 1-15.

[52] Zaharia, M., et al. *Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling*. in *Proceedings of the 5th European conference on Computer systems*. 2010. ACM.

[53] Zhang, X., et al. *Improving data locality of mapreduce by scheduling in homogeneous computing environments*. in *Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on*. 2011. IEEE.

[54] He, C., Y. Lu, and D. Swanson. *Matchmaking: A new mapreduce scheduling technique*. in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. 2011. IEEE.

[55] Kondikoppa, P., et al. *Network-aware scheduling of mapreduce framework ondistributed clusters over high speed networks*. in *Proceedings of the 2012 workshop on Cloud services, federation, and the 8th open cirrus summit*. 2012. ACM.

[56] Kc, K. and K. Anyanwu. *Scheduling hadoop jobs to meet deadlines*. in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. 2010. IEEE.

[57] Thakur, S., R. Singh, and S. Sharma, *Dynamic Capacity Scheduling in Hadoop.* International Journal of Computer Applications, 2015. 125(15).

[58] Sandholm, T. and K. Lai. *Dynamic proportional share scheduling in hadoop*. in *Job scheduling strategies for parallel processing*. 2010. Springer.

[59] Flink, A. *Apache Flink Stack*. 2018; Available from:

https://ci.apache.org/projects/flink/flink-docs-release-1.4/internals/components.html.

[60] Carbone, P., et al., *Apache flink: Stream and batch processing in a single engine.* Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2015. 36(4).

[61] Ashwani Sheoran1, D.M., K. Senthil Kumar3, *Mapreduce Scheduler: A Bird Eye View*, in *International Conference on Electronics, Communication and Aerospace Technology*. 2017.

[62] Anusha Itnal1, S.U., *Review of Map Reduce Scheduling Algorithms*, in *International Journal of Computer Techniques*. 2016.

[63] Perwej, D.Y., *THE AMBIENT SCRUTINIZE OF SCHEDULING ALGORITHMS IN BIG DATATERRITORY*. International Journal of Advanced Research (IJAR), 2018.

[64] Raoot, P., *Analysis of Job Scheduling Algorithms for Dynamic Job Ordering Optimization.*

[65] Manjaly, J.S. and E.A. Chinnu, *A relative study on task schedulers in Hadoop MapReduce.* International Journal of Advanced Research in Computer Science and Software Engineering, 2013. 3(5): p. 744-747.

[66] Hadoop, A. *Capacity scheduler*. 2017; Available from: http://hadoop.apache.org/docs/r1.2.1/capacity_scheduler.html.

[67] Divya S1., K.R.R., Rini Mary Nithila I3., Vinothini M4, *Big DataAnalysis and Its Scheduling Policy – Hadoop.* IOSR Journal of Computer Engineering (IOSR-JCE), Jan – Feb. 2015.

[68] Chen, J., D. Wang, and W. Zhao, *A task scheduling algorithm for Hadoop platform.* Journal of Computers, 2013. 8(4): p. 929-936.

[69] Hadoop, a. *Fair scheduler*. 2017; Available from: http://hadoop.apache.org/docs/r1.2.1/fair_scheduler.html.

[70] Patil, S. and S. Deshmukh, *Survey on task assignment techniques in Hadoop.* International Journal of Computer Applications, 2012. 59(14).

[71] Binu, B.P.A.a.A., *Survey on Job Schedulers in Hadoop Cluster*. IOSR Journal of Computer Engineering, vol. 15, Issue 1,, 2013. vol(15): p. Pp 46-50.

[72] Narkhede, V. and S. Khandare, *Fair scheduling algorithm with Dynamic load Balancing using In grid computing.* International Journal of Engineering and Scienc e, 2013. 2(10).

[73] Rao, B.T. and L. Reddy, *Survey on improved scheduling in Hadoop MapReduce in cloud environments.* arXiv preprint arXiv:1207.0780, 2012.

[74] G, S.S.a.H.K., *IMPROVED FAIR SCHEDULING ALGORITHM FOR TASKTRACKER IN HADOOP MAPREDUCE.* International Journal of Advanced Technology in Engineering and Science, 2015. vol.03(01).

[75] Mesos, A. *Apache Mesos*. 2047; Available from: http://mesos.apache.org/documentation/latest/.

[76] Hindman, B., et al. *Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center*. in *NSDI*. 2011.