INDEPENDENT AND DEPENDENT JOB SCHEDULING ALGORITHMS BASED ON WEIGHTING MODEL FOR GRID ENVIRONMENT

HAZEM M. AL-NAJJAR

UNIVERSITI SAINS MALAYSIA

2018

INDEPENDENT AND DEPENDENT JOB SCHEDULING ALGORITHMS BASED ON WEIGHTING MODEL FOR GRID ENVIRONMENT

by

HAZEM M. AL-NAJJAR

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

Jun 2018

ACKNOWLEDGEMENT

Although this thesis represents an achievement that bears my name, it would not have been possible without help and encourage of others whom I would like to thank. First, and for most, I thank Allah for all His blessings and guidance.

I would like to express my gratitude to my beloved wife Nadia Al-Rousan, my parents Dr. Mohammad Al-Najjar and Mrs. Radwah Mahafza, my daughters Logain and Leen, my brothers and my sisters that continuously supporting me in my endeavor to complete my Ph.D. study.

I would like to express my sincere thanks and deepest gratefulness to my supervisor Dr. Syed Sahal Nazli Alhady and my co-supervisor Assoc. Prof. Dr. Junita Mohamad-Saleh for their patience, motivation, and immense knowledge. Their guidance helped me at all during the current time of this research and in writing of this thesis. I could not have imagined having better advisors and mentors for my Ph.D study.

TABLE OF CONTENTS

APPENDICES

LIST OF PUBLICATIONS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
ADJS	Adaptive Deadline Based Dependent Job Scheduling
Alg_1	Job Scheduling Algorithm Using Weighting Value
Alg_2	Job Scheduling Algorithm Using Weighting Value and the CPU Requirements
Alg_3	Job Scheduling Algorithm Using Weighting Value and Ranking of Jobs
APC	Analytic Hierarchy Process
BIC	Bayesian Criterion
CPU	Central Processing Unit
Dep_Alg_1	Dependent Job Scheduling Algorithm Dependent Value
Dep_Alg_2	Dependent Job Scheduling Algorithm Using Dependent Value And Job's Position
EDF	Earliest Deadline First
FCFS	Fist Come First Serve
FCM	Fuzzy C-Mean
FIFO	First-In-First-Out
GG	Gath-Geva Clustering Algorithm
GK	Gustafson-Kessel Clustering Algorithm
HTML	Hypertext Markup Language
JRB	Job Ranking Backfilling
LJF	Longest Job First
LWF	Largest Weight First

MILP	Mixed Integer Linear Programming
MLP	Multi-Layer Perceptron
NP	Non-Deterministic Polynomial-Time
NS	Number of Sequence
PSO	Practical Swarm Optimization
RAM	Random-Access Memory
SJF	Shortest Job First
SJF	Shortest Job First
SOM	Self-Organization Mapping Algorithm
SPSS	Statistical Package for the Social Sciences
TCP/IP	Transmission Control Protocol/Internet Protocol
TNS	Total Number of Sequences

LIST OF SYMBOLS

F1	Factor multiple with number of CPU
F2	Factor multiple with memory size
F3	Factor multiple with average CPU time
F4	Factor used to describe the behavior of categorical variables in raking model
F5	Factor uses to weight the dependent jobs equation multiple with job weight
F6	Factor uses to weight the dependent jobs equation multiple with job ID
β	Beta value used for discrete uniform distribution
λ	Lamda value used for Poisson distribution
k	Degree of freedom value uses for chi-squared distribution

ALGORITMA PENJADUALAN KERJA TIDAK BERGANTUNGAN DAN BERGANTUNGAN BERDASARKAN MODEL PEMBERAT UNTUK PERSEKITARAN GRID

ABSTRAK

Sistem Grid telah digunakan untuk menyelesaikan masalah kompleks yang memerlukan bertahun-tahun untuk dilaksanakan. Salah satu isu dalam meningkatkan prestasi sistem grid ialah bagaimana menjadualkan kerja yang dikemukakan dengan cekap, pantas dan boleh dipercayai. Tambahan pula, penjadualan kerja dianggap sebagai masalah *NP-hard*. Oleh itu mencari urutan yang optimum untuk meningkatkan prestasi jumlah masa perlaksanaan dan purata masa menunggu akan menjadi sukar dan akan menggunakan banyak sumber pengiraan. Hasilnya, penyelidik-penyelidik telah cuba memperbaiki sistem penjadualan kerja dengan menggunakan pelbagai algoritma. Walau bagaimanapun, algoritma terdahulu adalah rumit dan memerlukan sumber pengiraan yang banyak. Tambahan pula, kerja-kerja mereka tidak mempertimbang penggunaan pemboleh ubah kategori kerja dalam pelaksanaan kerja sebagai parameter domain. Tesis ini membentangkan model pemberat menggunakan kluster *Twostep* untuk memberian pembolehubah kategori dan berterusan bagi kerja ke dalam kelas untuk kedua-dua penjadualan kerja tidak bergantungan dan bergantungan. Selepas itu, persamaan kedudukan digunakan untuk mengatur kelas yang dihasilkan daripada paling ringan hingga paling berat. Tambahan pula, model regresi linear bersama kelas kedudukan yang dihasilkan digunakan ke dalam model pemberat yang dicadangkan. Model yang terhasil kemudiannya digunakan ke dalam algorithma-algorithma penjadualan kerja tidak bergantungan dan bergantungan untuk mengesahkan keupayaan model penjadualan kerja yang dicadangkan dalam persekitaran yang sebenar. Untuk mengesahkan keupayaan model pemberat yang bermatlamat untuk meningkatkan prestasi penjadualan kerja, gabungan antara nilai pemberat kerja dan kedudukan penyusunan semula kerja telah diambil kira. Sebaliknya, pemberat kerja digabungkan dengan graf kerja tanggungan untuk meningkatkan penjadualan kerja bergantungan. Melalui simulasi, algoritma penjadualan kerja bebas menunjukkan peningkatan dalam jumlah masa pelaksanaan dan masa menunggu purata yang bersamaan dengan 1.13 dan 7.12 kali, masing-masing. Untuk algoritma yang bergantung, hasilnya mengatasi algorithma sebelumnya perlaksanaan dan purata masa menunggu masing masing bersamaan 1.13 dan 7.12 gandaan. Bagi algoritma yang bersandaran, secara purata Algoritma 4 mengatasi prestasi algoritma sebelumnya dalam jumlah masa pelaksanaan dan purata masa menunggu, dengan penambahbaikan masing-masing adalah 1.31 dan 3.05 gandaan. Keputusan ini telah menunjukkan bahawa pembolehubah kerja kategori dan berterusan boleh digunakan untuk memperbaiki jumlah masa pelaksanaan dan purata masa menunggu oleh algoritma penjadualan kerja dengan kurang bebanan dalam persekitaran yang sebenar.

INDEPENDENT AND DEPENDENT JOB SCHEDULING ALGORITHMS BASED ON WEIGHTING MODEL FOR GRID ENVIRONMENT

ABSTRACT

Grid system has been used to solve complex problems that need years to be executed. One of the issues in improving the performance of a grid system is how to schedule the submitted jobs in an efficient, fast and reliable way. In addition, job scheduling is considered as a NP-hard problem. Therefore, finding the optimal sequence to improve the total execution time and average waiting time will be very difficult and will consume a lot of computational resources. As a result, researchers have tried to improve job scheduling system using multiple algorithms. However, the previous algorithms are complicated and needed a lot of computational resources. Besides that, their works have not considered using job categorical variables in serving jobs as a dominant parameter. This thesis presents job weighting model using a *Twostep* clustering to assign the categorical and continuous variables of jobs into classes for both independent and dependent job scheduling. After that, ranking equation is used to arrange the generated classes from lightest to heaviest. Moreover, linear regression model with the generated ranking classes is employed into the proposed weighting model. The resulting model is then applied onto the independent and dependent job scheduling algorithms to verify the capability of proposed job scheduling model in a real environment. To validate the capability of the proposed weighting model which aims to improve independent job scheduling, a combination between job weight value and job ranking backfilling is considered. On the other hand, a job weight is combined with a graph of dependent jobs to improve dependent job scheduling. By simulation, independent job scheduling algorithm showed improvement in total execution time and average waiting time which is equal to 1.13

and 7.12 times, respectively. For the dependent algorithm, the results outperform the previous algorithms in total execution time and average waiting time, the improvement is 1.31 and 3.05 times, respectively. The results have demonstrated that the categorical and continuous variables of jobs can be used to improve the total execution time and average waiting time of job scheduling algorithms with less overhead in a real environment.

CHAPTER ONE

INTRODUCTION

1.1 Overview

Grid system is defined as a technology to share computing, application, data, power and storage across multiple organizations (i.e. research institutes and universities) that belong to different administration domains to handle large sets of data and to solve problems in different sciences such as finding prime numbers for very huge set (Yan et al., 2016). Besides that, grid system is used to solve complicated and difficult chemistry, physics and medical problems in fast, reliable and robust way (Foster et al., 2008; Mohsen et al., 2012).

The grid resources are computers distributed all over the world and linked together through different network connections. The resources receive jobs (submitted problem) through special server in the network called scheduler, when a scheduler receives jobs from different users, it serves them based on scheduling algorithm (Yan et al., 2016).

The scheduler accepts many independent jobs (i.e., jobs which can execute immediately without having to depend on other jobs) or/and dependent jobs (i.e., jobs which need to wait for other jobs depending on certain conditions prior to its execution). These submitted jobs will increase the demand for job processing on a grid system and the scheduler needs to serve these jobs as fast as possible without degrading the performance of the system (Patel, 2014; Hashemi et al., 2012; Jenaris and Periyasamy, 2016). Hence, job scheduling algorithm is required to manage the flow of coming jobs in order to improve the performance of a grid system in the most effective and cheap way to improve a grid performance without changing the internal architectures of grid system itself (Bellavista et al., 2017). As a result, improving job scheduling algorithm will enhance grid system performance metrics such as total execution time, average waiting time, energy consumption, bandwidth utilization, resources utilization and so on.

There has been an increase in the use of a grid system by researchers, this is because of the advantages provided by a grid system such as save money for not having to buy servers, do not have a single point of failure and high data transfer (Xia et al., 2010; Mohsen et al. 2012). Besides that, increasing the complexity of job scheduling problems has triggered many researchers to improve the performance of job scheduling in a grid system by proposing new systems.

Unfortunately, the previous job scheduling algorithms have been built based on continuous job variables only without considering categorical jobs variables. Also, they have been built without taking into account the factor on how to arrange the jobs' priorities using proper ranking equation. Therefore, more efficient job scheduling algorithm is required.

1.2 Problem Statement

Job in a queue has been defined by many researchers using different variables and techniques. As a result, the researchers have suggested different models to calculate the smallest job to be used in serving the independent and dependent jobs in a grid system as employed in Rodero et al. (2009), Al-Najjar et al. (2012), Shin et al. (2015), Rubab et al. (2016), Harounabadi (2016), Punia and Gulati (2016), Kaur, (2017), Ghosh et al. (2017), Idris et al. (2017), Tovar et al. (2018), Shabanov et al. (2018). Unfortunately, many suggested job scheduling models in the literature faced many problems which mainly can be summarized as follow: calculating job's weight value is complicated and needs lot of resources and if a job's weight value is large, it needs a large memory space to save the value. On the other hand, finding the most suitable ranking equation for job in a grid system has been discussed by many researchers. Rodero et al. (2009) proposed an experimental equation to serve jobs, where Toporkov et. al. (2015) suggested a mathematical ranking equation to define the job's weight. However, the effectiveness of ranking equation has not proven in the previous works as to the best knowledge to the authors. Besides that, no statistical analysis found to calculate the rank equation based on the job trace.

The previous independent and dependent job scheduling algorithms that have been discussed by many researchers. All of the works used only continuous variables of the jobs (i.e memory size) without considering the effect of the job's categorical variables such as Berger model (Xu. et al., 2012), particle swarm optimization (Pande et al., 2010), energy optimization model (Liu, 2016), firefly algorithm (Esa and Yousif, 2016), bees swarm (Bitam et al., 2017) and other intelligent models such as swarm intelligence algorithm (Effatparvar et al., 2016) and Bee Colony (Bouyer, 2016; Zhou and Yao, 2017). However, those categorical variables of the jobs could be important to define a particular job to the scheduler to make the job unique. Such variables can be the group ID of user and the application ID that describes the type of the submitted jobs (i.e. if the job is a Unix job, HTML code, JavaScript code). as explained by Janiak and Krysiak (2012), Falzon and Li (2012), Vegda and Prajapati (2014), Selvi and Manimegalai (2017). Therefore, there may be lack of information regarding a certain job which may be needed to improve the entire grid system performance. Overall, this problem relates to the question of "*How to serve an/a independent/dependent job in a fast way without testing all the possible solutions and by using a simple value (presented using not more 8 bits) that can describe the job behavior?*".

This can be done by choosing the best clustering and prediction models based on continuous and categorical jobs' variables to design a weighting model that will be used in designing independent and dependent algorithms in a real and simulated environment.

1.3 Aim and Research Objectives

The aim of this research is to develop job scheduling algorithms for dependent and independent jobs in a grid system using continuous and categorical job variables, to improve the total execution time and average waiting time of a scheduling system. The objectives set forth to support the aim of the research are:

i. To develop job weighting model based on jobs' continuous and categorical variables to define the smallest jobs in the job scheduling system.

ii. To develop job ranking equations using continuous job variables based on enhanced Rodero job ranking equation to improve the smallest job selection in a grid system.

iii. To design, implement and simulate independent/dependent job scheduling algorithms based on job weighting, to improve total execution time and average waiting time in a grid system.

1.4 Research Scope and Limitations

This research is bounded by the following scopes:

i. This research focuses on job scheduling models and how to advance jobs from the queue to maximize the performance metrics. To make the research specific, two metrics are chosen to validate the performance of the proposed algorithms, which are total execution time and average waiting time.

ii. This research is bounded into predefined categorical and continues job variables only. In other words, a new user, group and application are not considered in the job scheduling model and only predefined information for the above variables are considered.

iii. The resources have an equal probability to be selected to execute jobs that assigned by the scheduler in a grid system. Besides that, grid system will allocate the resources that needed for each job using virtual machines and virtualization technology. Besides that, the resources are uniformly chosen by the scheduler.

iv. The proposed job scheduler is described as a global, dynamic, suboptimal, approximate and centric scheduler as will be discussed in Chapter 2.

v. The scheduler do not have any information about the run time of the submitted job and the completion time of the job on specific virtual machine is unknown.

vi. Ranking the job's cluster values are based on predefined ranking equation.

vii. The ranking equation is based on the predefined the dataset.

1.5 Thesis Organization

This thesis is organized into six chapters as follows.

Chapter 1 explains the overview of job scheduling model, then the problem statement of the job scheduling, the research objectives, scope of the research and the outlines of the thesis are explained.

Chapter 2 defines grid system and the issues involved in the field. This chapter then explains job scheduling in a grid system and how different research works solved the scheduling using different models and techniques. Finally, the job scheduling analysis is discussed and explained by discussing the output and input of the scheduling model in grid system.

Chapter 3 discusses the proposed job weighting model to solve the problem of independent and dependent jobs in a grid system.

Chapter 4 proposes three independent and two dependent job scheduling algorithms and discusses and analyzes the merits and demerits of the proposed job scheduling algorithms.

Chapter 5 discusses the analysis of clustering and ranking algorithms that used to build the job weighting model. This chapter explains the results of the simulation program for dependent and independent job scheduling algorithms. After that, simulated results of the algorithms are analyzed and discussed to explore the improvement over the previous algorithms.

Finally, the conclusions of the research are drawn in Chapter 6. The chapter explains the findings and the achieved objectives and highlights the future researches.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter is dedicated to provide the relevant background information required to understand the principle concept of job scheduling algorithms in a grid system. It starts with Section 2.2 which explains about grid system, the issues of grid system and highlights main problems that face grid system. Then, Section 2.3 presents a discussion on job scheduling in grid system highlighting independent and dependent jobs concept and explains categorical and continuous job's variables. It also presents job scheduling selection strategy and how to select the resources to execute the jobs. Besides that, clustering and prediction models in a grid system are discussed in the same section. Moreover, in order to understand the related works of job scheduling, Section 2.4 highlights some of the related works in the field. Section 2.5 explains about job scheduling analysis and how to understand the performance parameters and variables in grid system. Finally, a summary is presented in Section 2.6.

2.2 Grid System

Grid system is defined as a technology used to coordinate and share computing applications, data and storage space or network resources across multiple organizations and administration environments. Indirectly it is able to tackle complex computational problems (i.e. analyzing huge images) in an easy and fast manner (Bashir et al., 2016; Mishra et al., 2017; Khan et al., 2017).

Grid system is a suitable solution for many researchers who target to improve and solve their mathematical, physical, chemistry problems and various other problems (Hashemi et al., 2012; Zissis et al., 2012; Hwang et al., 2016; Le-Khac et al., 2017; Yousafzai et al, 2017). Moreover, grid systems have many characteristics that make it a suitable solution for many problems, which can be described as follows:

- Large scale and multiple administration: Grid system can cover a large areas across multiple administrative domains, to allow all computers from different sites to work together (Hwang et al., 2016; Le-Khac et al., 2017).
- Scalability: Grid might grow from a few integrated resources to millions of resources. Therefore, the problem of latency and bandwidth in grid system should be solved from the beginning (Hwang et al., 2016; Le-Khac et al., 2017).
- Resource sharing: Gird uses the technology of the resource sharing to serve the submitting jobs (Hwang et al., 2016; Le-Khac et al., 2017).

On the other hand, grid system faces many problems that need to be taken into account by researchers to enhance and improve a grid system which could be summarized as follows:

- **Job scheduling**: Job scheduling defines as changing the flow of coming jobs to improve the performance of a grid system, which considers as one of the most effective and cheap way to improve a grid performance without changing the internal architectures of a grid system (Venkatarama, 2016; Bellavista et al., 2017).
- **Resources selection:** A method of assigning jobs to salves in a grid system, to improve the performance metrics (Ezugwu et al., 2016; Nik et al., 2017).

- **Limited local knowledge:** Having a knowledge about a grid system and clients' job should be able to increase the speed of a grid system. This is because the system can be implemented to decide suitable resources to handle particular job's requirements without degrading the performance of a grid system (Bensalem et al., 2010; Kovendan et al., 2017).
- **Routing Problem:** The resources can be distributed in any place all over the world. Hence, routing the data from one resource to another within a grid system is an important issue that needs to be solved by researchers (Singh et al., 2016; Cao et al., 2017).
- **Scalability**: A grid system should be designed to handle increasing number of users in the network. Hence, even if the number of requests increases, the system should be able to serve all the incoming jobs (Van et al., 2012; Rittinghouse et al., 2016).

From the listed issues, it is clear that the most important problem to be considered by many researchers is job scheduling. This is because improving the scheduling system will surely improve the performance of a grid system and enhance the resource management (Venkatarama, 2016; Bellavista et al., 2017). This does not mean that other problems are not important but job scheduling is considered as the main component that will direct the flow of jobs in a grid system. On the other hand, job scheduling faces many problems that need to be taken into account by researchers to enhance and improve a grid system which can be summarized as follows:

- Independent jobs can be starved if a grid system serves a light job first without considering the arrival time of a job (Van et al., 2012; Mishra et al., 2014).

- The complexity of serving the dependent jobs may degrade the performance of a grid system, if no procedure is followed to serve the dependent jobs (Taheri et al., 2013; Venkatarama, 2016).
- The total execution time and average waiting time of independent or dependent jobs may increase because scheduler has limited knowledge about jobs behavior and job's weight (Van et al., 2012; Mishra et al., 2014). Job weight can be defined as a value that describes the complexity of a job in a grid system.
- No standard job ranking model is used to define the job's weight and instead of having standard model, researchers have tried to suggest mathematical and experimental models to be used (Van et al., 2012; Mishra et al., 2014).

Finally, to understand the main components in designing any job scheduling model in a grid system, the next section will highlight the main techniques used in previous works.

2.3 Review of Job Scheduling in Grid System

Job scheduling is a method of assigning independent/dependent jobs to slaves in a grid system to improve the performance metrics of grid system. Many researchers in the literature have addressed independent/dependent job scheduling in a grid system as shown in Wang et al. (2008), Kwon et al. (2008), Yu. et al., (2008), Al-Najjar et al. (2012), Zhang. et al. (2014) and Hajikano et al. (2016). Moreover, job scheduling is classified based on the classification factors as shown in Figure 2.1. The taxonomy of job scheduling is as follows:

i. Local/Global

Local scheduling determines how the process will execute and allocate in a single processor as presented by Dash et al. (2016) and Rawal et al. (2016), where global scheduling is defined as policies used by a grid system to allocate jobs to different CPUs that distributed into different places all over the world. Therefore, using previous definitions, a grid system could be described as a global scheduling as studied by Bouyer et al. (2016).

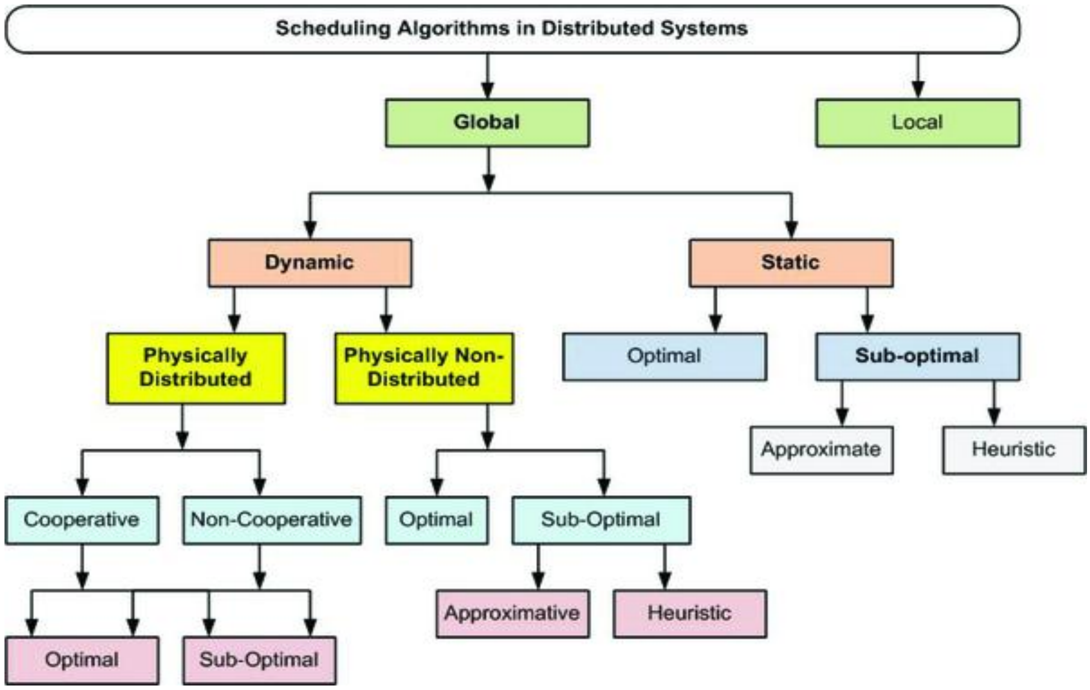

Figure 2.1:Taxonomy of scheduling algorithms (Preve, 2011).

ii. Static/Dynamic

In static scheduling each job knows the duration to finish its execution, so the allocation is depending on the time required to finish a job, where in the dynamic scheduling, system will not have a previous information about the time and when the job will complete, only jobs' requirements are available (Dash et al., 2016; Rawal et al., 2016).

iii. Optimal/Suboptimal

Job scheduling is considered as NP-hard problem, so the solution will take a polynomial time to be found. Therefore, finding the optimal solution for job scheduling, to serve a job in zero time and to make performance of grid system as fast as possible is a very difficult task and practically is impossible. As a result, researchers have tried to find the suboptimal solution to solve job scheduling (Dash et al., 2016; Rawal et al., 2016).

iv. Approximate/ Heuristic

The approximate job scheduling is used to find a solution that is sufficiently good or suitable to the scheduling problem. The goodness of the solution can be described by using one or more performance metrics such as resource utilization, memory usage, execution time and so on. On the other hand, some problems cannot be solved in tradition models or the models are very slow, therefore heuristic model will take a place in finding a solution as mentioned in Bansal et al. (2016).

v. Distributed/Centralized

In grid system, the scheduler is either distributed or centralized. In the centralized model, one centralized scheduler will receive the requests and allocate resources based on job's requirements. In the distributed model, the scheduler is distributed in many places (Preve, 2011).

Finally, job scheduling has been used to serve independent and dependent jobs in grid system by changing the flow of the incoming jobs to a grid system. To understand job scheduling system the following subsections cover the main terminologies and techniques that could be found in job scheduling field.

2.3.1 Independent and Dependent Job in Grid System

The following is a definition of job scheduling based on the optimal definition which can be extracted from the previous works.

"Job scheduling model can be defined as finding trusted users submitting trusted jobs

to trusted scheduler to serve dependent/independent jobs based on smallest execution time and minimum resources requirement or/and jobs' priority taking into consideration the waiting time of other jobs to avoid starvation and serve the jobs by using trusted resources via trusted secured channels."

Based on the previous job scheduling definition, job scheduling is divided into two parts as follow:

1. Trusted users/scheduler/jobs/resources/channels: Grid system should have trusted users that submit trusted jobs (real jobs) to trusted scheduler. The trusted scheduler should be existed in a grid system to ensure that all the submitted jobs can be served. The resources that run jobs must be evaluated from any viruses or/and malware that cause a performance degradation. Moreover, the channels that used in sending and receiving jobs should be protected from attackers and any malicious activities that will break the connections (Demchenko et al., 2017; Tan et al., 2017).
2. Independent/dependent job: Independent job means job that can be served directly, if there are enough resources, without waiting other jobs to finish. The dependent job means that job must wait other jobs to complete to start execution after that, sometimes this type of jobs can block a grid system as proved in Al-Najjar et al. (2012) and Gim (2017).

The differences between independent and dependent jobs in a grid system is found in Figure 2.2. The figure shows four trees that have one, zero, five and zero child(ern). The job that have zero child is defined as independent jobs, while others with one or more children are defined as dependent jobs. Returning to the example, jobs 1 to 4 and 5 to 10 are considered as independent and dependent jobs, respectively. Finally, the problem of job scheduling has been discussed in many research works and solved by using job's variables. The job's variables are divided into continuous and categorical variables as will be discussed in the next sub-section.

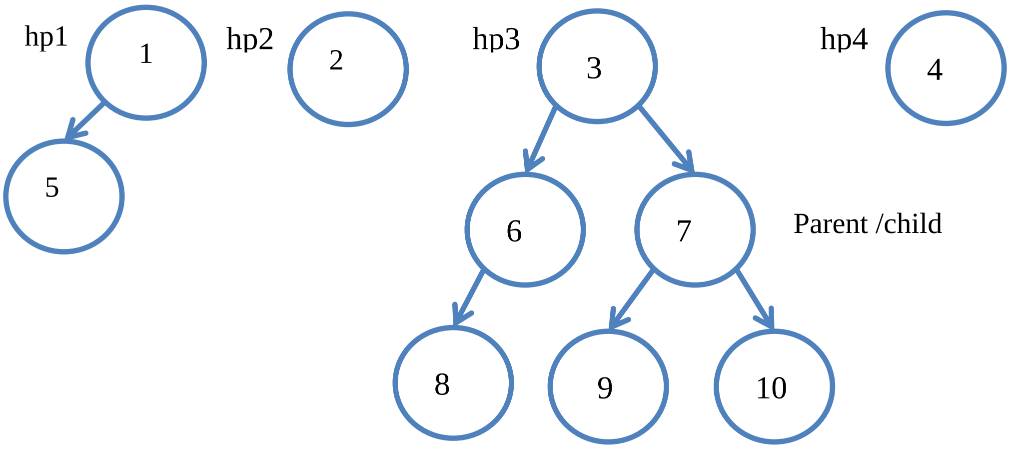

Figure 2.2: Differences between independent and dependent jobs, where hp is the head parent.

2.3.2 Categorical and Continuous Job Variables in Grid System

Jobs in grid system contain a set of variables which can be described as quantitative and/or qualitative data (Glaser, 2017). In job's quantitative data, all the variables can be described as a scale or measure called continuous variables where job's qualitative variables defined as a discrete or categorical variables (Spinner et al., 2017). Jobs in a grid system contain many variables such as number of CPUs, group ID, user ID, executable ID, memory size and memory speed. The job's categorical variables are divided into three types as follows (Bernard et al., 2016; Jha et al., 2017):

i. **Nominal job's variables** are variables related to the nominal scale, where jobs are grouped based on unordered data such as group ID (group 1, group 2, group 3), user ID (user 1, user 2, user 3).

ii. **Dichotomous job's variables** are considered as a special case of nominal variables, in which there are only two categories or levels (i.e. "Yes" or "No" question's answer).

iii. **Ordinal job's variables** are considered as a special case of nominal variables in which categories are divided into two or more levels and can be ordered or ranked. For example, job's weight can be ranked as follows: light, moderate and heavy.

On the other hand, job's continuous variables are defined as quantitative variables, which divided into two types, either interval or ratio variables as follow (Mertler et al., 2016; Patten et al., 2017):

- **Interval job's variables:** are variables that can be described as interval such as memory size from 1 Kbyte to 4 Gbyte, where zero is considered as a base value for a memory size.
- **Ratio job's variables**: are variables that can be described as ratio such as size, speed. For example, memory size for job 1 is twice heavier than memory size for job 2.

Finally, serving jobs in a grid system is an important issue defined as job scheduling, where deciding the resources that should execute the jobs is defined as job selection strategy. The selection strategy gives the scheduler more freedom to improve the resource utilization in a grid system.

2.3.3 Job Selection Strategy in Grid System

Selecting the resource to run a job is a challenging issue in a grid system, especially if resources are affected by malware. Therefore, to select suitable resources, many models have been presented in the literature (Ahmad et al., 2017 and Tychalas et al., 2017) to solve this issue as follows:

- **Biggest free**: The machines that have highest free resources will be chosen to run the selected job.
- **Random**: The machines that will execute the selected job will be chosen randomly without taking any parameters into consideration.
- **Best Fit**: The machine that leaves the least free resource will be chosen to execute the selected job.
- **Equal Utilization**: The lower utilization machine will be used to serve the selected job.

The selection strategies are divided into two main parts, which are selecting based on the jobs variables (i.e best fit) and selecting the resources information (i.e. biggest free). Moreover, the jobs in a grid system contain many variables, researchers have suggested many methods to combine jobs' variables in one single value either using mathematical equations or using clustering methods.

2.3.4 Job Clustering Models

Job clustering is a process used to divide the jobs into groups, so each group will contain jobs that have approximately the same characteristics based on the similarity definition (Berkhin, 2006). In this process, the cluster number is the value that will be used to decide the action on the jobs. For example, assume dataset contains 50 jobs, each job has the following information: number of CPU, memory size, group ID and user ID. If the administrator wants to divide the jobs into groups, to define the weight of each jobs, the administrator can divide the jobs into a number of groups and then order the groups from light to heavy based on the weight of each group.

Moreover, to clarify the clustering concept, the main algorithms in the field of job's clustering are described as follows:

- **Hierarchical methods**: Jobs are divided into clusters by building a tree of clusters. In this method, the jobs are organized into hierarchical way such as hierarchical clustering (Aggarwal and Reddy, 2013).
- **Partitioning relocation methods**: Jobs are clustered into different sub sets by using greedy heuristics model, such as K-Means and K-Medoids (Berkhin, 2006).
- **Density-based partitioning**: The data is divided into groups based on the density area of the data within the dataset values such as Gustafson-Kessel (GK) Clustering and Gath-Geva (GG) Clustering (Mota et al., 2017).
- **Methods based on co-occurrence of categorical data:** Method used for clustering categorical data (Brusco et al., 2017).
- **Artificial intelligence models:** In this approach, artificial intelligent models will be used to cluster jobs such as Fuzzy C-Mean (FCM) clustering (Roy and Datta, 2017).

To clarify the concept of jobs clustering, the main algorithms in the field are explained as follow:

- **K-Means Clustering**: K-means algorithm that suggested by MacQueen in 1967 is a clustering algorithm that divides the jobs into predefined number of clusters assume K. The main idea is to define the center for each cluster and use the distance between the jobs set and the cluster centroid to assign the jobs object to its nearest cluster.
- **Fuzzy C-Means Clustering**: Fuzzy C-Mean (FCM) algorithm is developed

by Dunn in 1973 and improved by Bezdek in 1981 to be used in the pattern recognition systems.

- **Self-Organization Mapping (SOM)-Training phase**: is divided into two phases: training phase and mapping phase. Training phase builds the map using input, while mapping phase automatically classifies a new job. Dividing jobs into groups is done, without considering the number of clusters on advance and only the size of the map should be defined. SOM could be described as visualization technique to describe a high dimensional data by using self-organizing neural networks to reduce the dimensions of data (Kohonen and Somervuo, 1998; Bullinaria, 2004).
- **TwoStep Clustering (IBM, 2012)**: TwoStep Clustering is a tool designed to handle the naturally of data which defines the categorical and continuous variables of data without converting the categorical variable's into another form. The main differences between this clustering method and other clustering methods are summarized in the following points:
    1. Can use categorical and continuous variables.
    2. Can automatically choose the number of clusters.
    3. The algorithm can work well with huge dataset.

All the used clustering algorithms accept continuous variables only, while Twosetp clustering can accept continuous and categorical variables. Therefore, to cluster jobs using the clustering algorithms that cannot consider categorical variables, a code conversion should be used first by converting a categorical variables to dummy variables (i.e one and zero) (Coussement et al., 2017; Mitzenmacher and Upfal, 2017).

To find the suitable number of clusters theoretically many methods are suggested such as elbow method, average silhouette method, gap statistic method and other models (Halkidi et al., 2001; Amorim et al., 2015; Landa et al., 2017). On the other hand, all previous clustering algorithms in the literature review can divide the jobs into groups but cannot define which group is the best. Therefore, it is the responsibility of the researcher to arrange them based on the importance of the cluster. Researchers have used many ways to arrange the clusters such as using mathematical models, self-decide based on variables, or using statistical analysis of the generated clusters. This process is defined as ranking the clusters. Finding ranked clustered jobs could be done using many ways but finding the most suitable way is depending on the job's variables, the size of the dataset and other variables. Using the ranked clustered value in a grid system could improve the performance but it could not be used online because this will degrade the performance of a grid system. To solve this issue, many techniques can be used such as prediction and probability models.

2.3.5 Job Prediction Models

Job prediction models are used to predict the future behavior of job using the knowledge from previous submitted jobs to the scheduler. Prediction method is divided into two phases, training phase and prediction phase. In the training phase, previous jobs are used to train the prediction model. In the prediction phase, the model is used to predict the future behavior of the jobs. For example, in previous job's example, the administrator can build a prediction model to predict the jobs weight by using previous jobs information. Besides that, the prediction model contains two types of variables which are independent variables (input variables) and dependent variables (output variables). To build a prediction model by using the dependent and independent variables, many models could be used. They are discussed in the following subsections.

2.3.5(a) Multinomial Logistic Regression Model

Multinomial logistic regression is a general logistic regression model that builds a regression model using categorical variable as an output and multiple variables as an input (Mertler and Reinhart, 2016). The multinomial model is used to predict the probabilities of the dependent variable that contains different categorical output (more than two levels), by using a mixed/ non mixed independent variables.

Multinomial logistic regression is used mainly when more than two types of categorical variables existed in the dependent variable such as a blood samples O+, O-, B+, B-, AB+, A+, names in the voting systems or country names.

2.3.5(b) Linear Regression Model

Linear regression is a model that used for building a relationship between dependent variable and independent variables. In case, more than one independent variables are used, the model called a multivariate linear regression. In the two cases, the dependent variable is numerical value (Lane, 2012).

The linear regression model accepts only numeric values as dependent variable, therefore categorical variables should be converted to dummy variables.

2.3.5(c) Discriminant Function Analysis

Discriminant function analysis is a method used to find a relationship between dependent variable and independent variables, in which the dependent variable should be categorical variable and independent variables should be numerical variable. Moreover, in designing a discriminant function analysis one question should be answered, which is "*how many dimensions are needed to express the relationship between input and output*" (Bian, 2012).

2.4 Job Scheduling Algorithms

This section explains the previous works and how the problem of job scheduling has been solved. As a first step, the previous and most popular models are explained and then more complicated models are discussed. Section 2.4.1 presents four models which are FCFS, LJF, SJF and JR-Backfilling, where Section 2.4.2 presents complicated, hybrid and complex job scheduling techniques. The section divided the job scheduling models depending on the techniques used in designing job scheduling model and year of the model. The techniques are divided as follows: priority of the jobs, fairness between the jobs, energy optimization, optimal sequence, firefly and/or BAT optimization, Practical Swarm Optimization (PSO), genetic model, Multi-Layer Perceptron (MLP) model and other techniques in the literatures.

2.4.1 A Comparison of Job Scheduling Models

To serve the incoming jobs, the scheduler will use one of two policies which are preemptive or non-preemptive policies (von et al., 2017). In preemptive policy, such as First Come First Serve (FCFS), Job Ranking Backfilling (JR-Backfilling) Longest Job First (LJF), Shortest Job First (SJF), Min-Min, Max-Min and Min-Max, the queued jobs will not be advanced until the resources are free to be used and no working jobs are located in the resources, where in the non-preemptive, such as round robin, the queued jobs are forced to stop running and one of the queued jobs will take the place until specific time. Therefore, advancing the job is done by stopping the running jobs and replacing them with another jobs in a grid system until the last job completed/terminated/stopped. Many approaches are used to schedule the jobs in a grid system. In this section four previous and most popular algorithms are discussed.

2.4.1(a) First Come First Serve

FCFS serves the jobs based on its position in the queue. Therefore, the job that is in the head of the queue will be served first- if there are available resources to serve the job- without considering the job's weight or the requirements of the job. This procedure in serving jobs will make the approach more robust in serving jobs, since no blocking in the incoming jobs will be allowed and the approach considered a deterministic value for the waiting time. The disadvantage of FCFS can be noted, if the light jobs arrived in the last and the resource can serve them but a grid system cannot serve, because the job must wait its turn or/and the heavy jobs are in the head of the queue and the resources are not enough to serve them. As shown in Figure 2.3, the incoming jobs will be stored at the end of the queue and the algorithm advances the jobs in the head of the queue to be served first (Thakur et al., 2017; Umar and Pujiyanta, 2017). FCFS is simple, simple in implementation and no starvation could be happened.

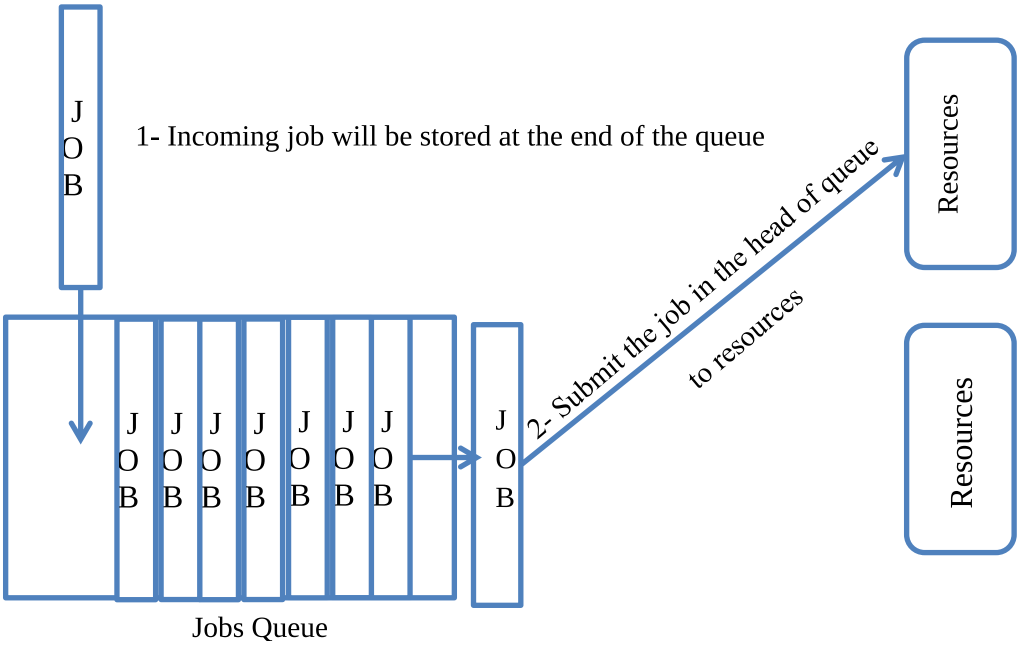

Figure 2.3: First Come First Serve procedure.

2.4.1(b) Longest Job First (LJF)/Shortest Job First (SJF)

Longest Job First (LJF) focuses on serving the longest jobs that have higher requirements first as shown in Figure 2.4. Thus, the algorithm will search on the longest job to be advanced first. On the other hand, shortest job first serves the shortest job first. The complexity of LJF and SJF algorithms is O (*n*), where *n* is the size of the queue (Thakur et al., 2017; Umar and Pujiyanta, 2017).

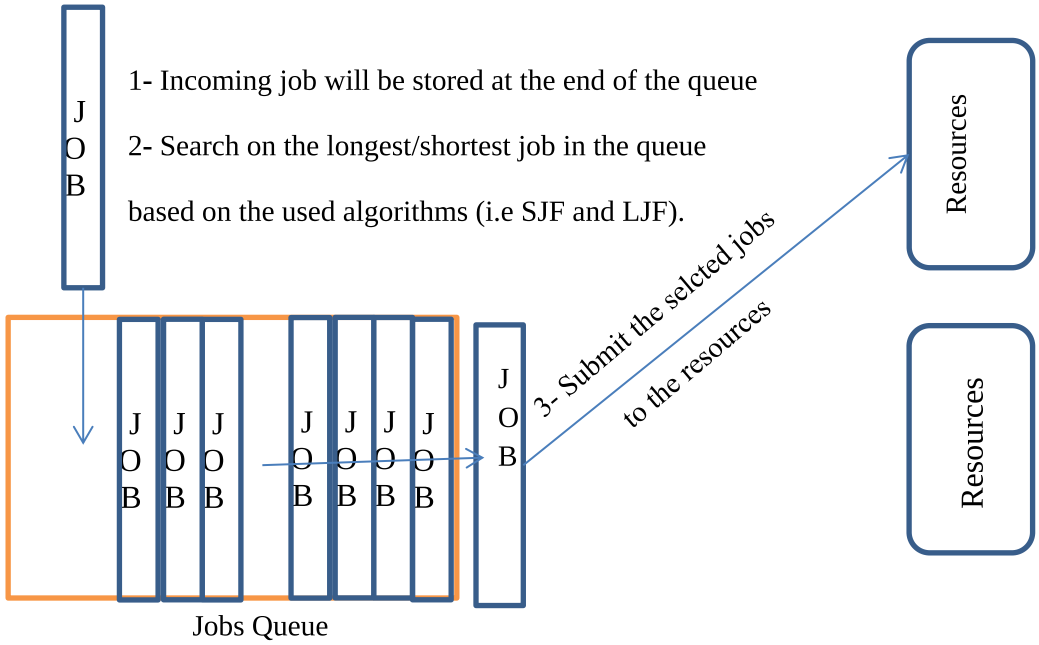

Figure 2.4: LJF/SJF procedure.

2.4.1(c) Job Ranking (JR) Backfilling

The drawbacks of the FCFS, SJF and LJF algorithms are solved in job ranking backfilling model. The backfilling model aims to advance small jobs to be executed first before large jobs at the same time considering the waiting time of the jobs to avoid starvation (Yousafzai et al., 2017). Moreover, defining small job is considered a challengeable issue in job scheduling, so researchers defined many job serving methods, such as user-defined method, where users will define the expected time for job or/and jobs requirement, or prediction method, in which the system will predict on the weight of job after submitting it. Redero et al. (2009) defined a small job based on its resource requirements, in order to advance small jobs for execution. The JR-backfilling police works as shown in Figure 2.5 and as follow.

- Update the information of the available resources in a grid system.
- Assign the rank value for each job by calculating the rank using predefined equation, then finds the smallest jobs between the current ones, if the job requirements were not matched, then its rank value is -1.
- Advance the job that has smallest positive rank value to be executed in the resources.

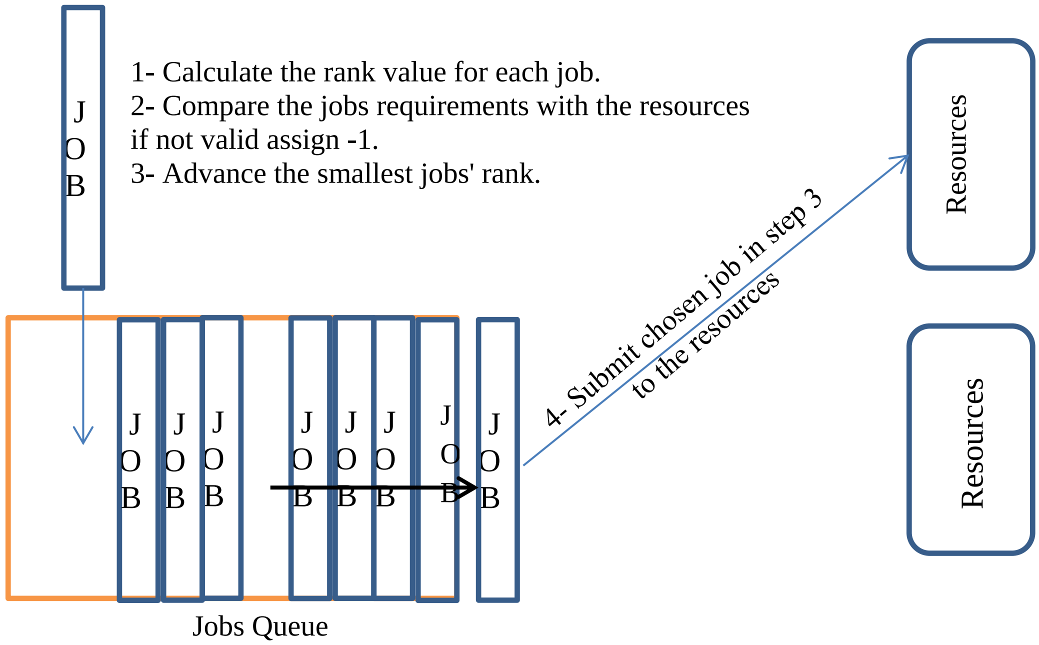

Figure 2.5: Job Ranking (JR) Backfilling procedure.

2.4.2 Related Works on Job Scheduling Models

This section is divided in two subsections, in section 2.4.2(a) the previous independent job scheduling algorithms are presented and in the section 2.4.2(b) the previous dependent job scheduling algorithms are presented.