جــامــعــة الــشــرق الأوســط
MIDDLE EAST UNIVERSITY

Task Scheduling Using Best-Level-Job-First on Private Cloud Computing

جدولة المهام على الحوسبة السحابية الخاصة باستخدام أفضل مستوى عمل أولا"

By

Mohammed Ibrahim Aridah

401410063

Supervisor

Dr. Mohammed A. F. Al -Husainy

A thesis Submitted in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Department of Computer Science

Faculty of Information Technology

Middle East University

August, 2016

AUTHORIZATION STATEMENT

I am, Mohammed Ibrahim Ismail Aridah, authorize the Middle East University to supply a copy of my thesis to libraries, establishments or individuals upon their request.

Name: Mohammed Ibrahim Ismail Aridah

Data: Tue. 16 August , 2016

Signature:

اقرار تفويض

أنا محمد ابراهيم اسماعيل عريضة ، أفوض جامعة الشرق الاوسط للدراسات العليا بتزويد نسخ من رسالتي ورقيا و الكترونيا للمكتبات أو المنظمات أو الهيئات و المؤسسات المعنية بالابحاث و الدراسات العلمية عند طلبها.

الاسم : محمد ابراهيم اسماعيل عريضة

التاريخ : 16/8/2015

التوقيع :

Examination Committee Decision

This is to certify that the thesis entitled "Task Scheduling Using Best-Level-Job-First on Private Cloud Computing" was successfully defended and approved on, 2016

Examination Committee Members **Signature**

(Head of Committee)
Dr.Ahmed Al-Hmouz
Assistant Professor
Faculty of Information Technology
Middle East University

(Internal Committee Member)
Dr. Mohammed A. F. Al –Husainy
Assistant Professor
Faculty of Information Technology
Middle East University

(External Committee Member)
Dr. Jehad Al-Saadi
Assistant Professor
Faculty of Information Technology
Arab Open University

Acknowledgment

Foremost, great thank for allah, then I would like to thank my thesis advisor Dr. Mohammad Fadhil Al Husainy for the continuous support of me, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Many sincere thanks also go to Information Technology Faculty members at Middle East University for their insightful conversations and hard questions, thank you for teaching me how to be a dedicated researcher. Million thanks go to my fellow colleagues for their support and encouragement, and kindness during my study.

I am forever indebted to my family who supported me during this academic journey and my whole life, and all of my friends specially Awad Al Sabbah, Ali Jawdat, Mohammed Al-dyria.

Thank you very much.

Mohammed Ibrahim Aridah

August/2016

Dedication

قال تعالى:(وإذ تأذن ربكم لئن شكرتم لأزيدنكم)

I dedicate this work to my parents, my brothers, my sister, my sons Yahiya and Ibrahim, my relatives, my friends, and all those who taught, helped, and supported me.

I lovingly dedicate this thesis to my wife, who supported me during all steps of the way.

Table of Contents

Chapter Two

Background and Literature Review

Chapter Five

Conclusion and future work

List of Tables

List of Figures

List of Abbreviations

Abbreviations	Meaning
ACO	Ant Colony Optimization
ART	Average Response Time
AWT	Average Waiting Time
BJF	Best-Job-First
BLA	Bees Life Algorithm
BLJF	Best-Level-Job-First
BVCF	Broker Virtual Machine Communication Framework
CIS	Cloud Information Services
CPU	Central Processing Unit
ECS	Elastic Container Service
ELB	Elastic Load Balancing
FCFS	First Come First Serve Scheduling Algorithm
GA	Genetic Algorithm
IaaS	Infrastructure as a Service
IT	Information Technology
JDK	Java Development Kit
LoU	Level-of-User
MIPS	Million Instructions Per Second
MPDSA	Modified Prioritized Deadline based Scheduling Algorithm
NIST	National Institute of Standards and Technology
NIST	National Institute of Standards and Technology

Q	Queue
PaaS	Platform as a Service
PJSA	Priority based Job Scheduling algorithm in Cloud
PSJN	Pre-emptive Shortest Job Next
QoS	Quality of services
RAM	Random Access Memory
RR	Round-Robin
SaaS	Software as a Service
SJF	Shortest Job First
SLA	Service Level Agreement
SOA	Service Oriented Architecture
VM	Virtual Machine
W	weight

Abstract

Task Scheduling Using Best-Level-Job-First on Private Cloud Computing

By: Mohammad Ibrahim Aridah

Supervisor: Dr. Mohammed A. Fadhil Al Husainy

Cloud computing is one of the latest IT technologies, and consisting of a set of virtualized resources that serve the users via the Internet.

Task Scheduler is one of the most important cloud computing problems. And that cannot be scheduled using single criteria, but according too many of the criteria's and rules. These rules and criteria's to be agreed upon by the service provider and the user.

There is a real gap in the design task scheduling algorithm and the level of service provided to users of the scheduling algorithm. How can making a priority of the user with the highest level to get the best service in private cloud computing that use the same network speed and the same storage capacity and the same processor speed.

The proposed algorithm named "Best Level Job First," based on four criteria: First, Level of User: it is the rank of the user, Time: time is expected to carry out the work, Cost: Here, we consider it the amount of energy consumption. Load: load on the system.

The proposed algorithm bothers significantly to the level of the users in the process of implementing their request for an instruction to build "the actual level of the user. And change the behavior of the algorithm is practically in queue.

By comparing the proposed algorithm with the algorithm Short Job First (SJF) the result given by proposed algorithm is the lower waiting time for users in the highest level. And by comparing the proposed algorithm with the algorithm Round Robin (RR) the result given by the proposed algorithm is the fast response time for users in the highest level.

Keywords:

Best-Level-Job-First, Quality of Service, CloudSim, weights, scheduling algorithm, Levels of user

الخــلاصة

جدولة المهام على الحوسبة السحابية الخاصة باستخدام أفضل مستوى عمل أولا"

اعداد : محمد ابراهيم عريضة

اشراف : د. محمد فاضل الحسيني

ملخص:

الحوسبة السحابية هي البيئة الجديدة لتكنولوجيا المعلومات، توفر المتطلبات الضخمة للبنية التحتية والموارد. إنها بركة افتراضية من الموارد التي يتم توفيرها للمستخدمين عبر الإنترنت. واحدة من القضايا الأساسية في هذه البيئة يرتبط بجدولة المهام. جدولة المهام السحابية تصنف كمشكلة غير منتظمة. و تلعب دورا رئيسيا في أنظمة الحوسبة السحابية.

لا يمكن أن تتم جدولة المهام على أساس معيار واحد ولكن وفقا لكثير من القواعد والأنظمة، توصف كاتفاق بين مستخدمي ومقدمي الخدمة. في الحوسبة السحابية الخاصة، هناك فجوة كبيرة بين معظم خوارزميات جدولة المهام التي تم تصميمها والمستوى الفعلي للمستخدمين. هنالك صعوبات في ترتيب مهام المستخدمين لتنفيذها. كيف يمكن جدولة أولويات المهام للمستخدمين على أساس تصنيف المستخدمين في المؤسسة التي تستخدم نفس الخدمات السحابية الخاصة. اقترحنا خوارزمية جدولة المهام تسمى خوارزمية أفضل–مستوى– وظيفي –أولا" (BLJF)، الذي يأخذ بعين الاعتبار مستوى المستخدمين، واعتبرت المعايير الأربعة: الوقت يشير إلى وقت الانتهاء، التكلفة تشير إلى سعر الموارد المستخدمة(استهلاك الطاقة)، ويشير التحميل إلى العبء على النظام، والمستوى يشير إلى مستويات المستخدمين.

الخوارزمية المقترحة تهتم بمستوى المستخدمين عند اختيار المهام لجدولتها، ولها القدرة على تغيير السلوك من خلال ترتيب أوزان المهام في قائمة الانتظار بشكل ديناميكي .

بمقارنة الخوارزمية المقترحة (BLJF) مع خوارزمية (round robin) و (SJF) كانت افضل في وقت الاستجابة و ضمن المعدل لوقت الانتظار.

الكلمات المفتاحيه: افضل–مستوى –عمل – اولا"، جودة الخدمة، CloudSim، الوزن ، جدولة الخوارزمية ، رتبة المستخدم.

Chapter One
Introduction

1.1 Overview

Cloud Computing issue refers to the resources and services that presented over the Internet, these services presented in the cloud environment to response for user requirement. The expanding of cloud resources or services and increasing used by people with a various background leads to the need for emphasis on the management of cloud users and their requirements.

With cloud computing, the main objective is a combine between multiple resources (hardware and software) that is available through the internet, the customers get all or some of these resources according to the used cloud system. Recently, the cloud computing has evolved rapidly due to the use of advanced equipment and virtualization technology and the use of distributed systems and so on (Navimipour & Milani, 2015, and Al-Mughrabi, 2013).

A private cloud is one of the deployment models of cloud computing that operated by a single organization. It is characterized that the resources are provided by the cloud vendor for internal uses, and determined the organization which have access to these resources. Thus, the utilizing of private cloud can be much more secure and data control than other deployment models of the cloud (Beal, 2016).

The rapid growth of cloud computing services and increasing numbers of users requires managing all those cloud users and their requests. one of the most important point to maintain the user requests in the cloud environment is Task Scheduling, it is using for increasing system performance and for decreasing task performance time. Task scheduler technique considered as a basic requirement for the provision of effective service in the cloud computing. It is implemented by mapping job requirements for the available resources to the users and minimizing total response time.

The scheduling technique that implemented the job in smallest execution time is called Shortest Job First (SJF) Scheduling Algorithm or Min-Min Algorithm. While the Scheduling technique that implemented all processes running in a circular queue in the smallest unit of time is called Round-Robin (RR) Scheduling Algorithm (Shimpy & Sidhu, 2014).

Based on the (Suakanto , & et al., 2012), the user utilizing two parameters for measuring the qualities received, these parameters represent in the average response time and waiting time. This thesis, the researcher focuses on two points. The first point is showing a new way of task scheduling algorithm that called Best-Level-Job-First (BLJF) on the private cloud. The second point is comparing this algorithm with a RR and preemptive SJF.

1.2 Problem Statement

In private cloud computing, the user level isn’t equal, and they don't provide the equivalent level of services, based on the user's position in the institution, some tasks are generated by very significant users and others are less significant. The significance of request relies upon the position level of who demand this request .When a big number of tasks becomes perform from the low level of the user versus the high level of user inquired task then the problem of task scheduling in a private cloud (which have the same Quality of Service (QoS)) increased, when a large number of tasks becomes to execute from the low level of the user in front of the high level of user requested task.

There is a large gap between the actual level of users and most tasks scheduling algorithms in the private cloud computing .They essentially used same criteria’s for scheduling tasks regardless of the various scheduling algorithms suggested for the cloud environment, such as:

- Execution time of the task
- Cost (price) of used resources by the task
- Load of the system
- The levels of users do not take into consideration in scheduling users' tasks.

Due to using the same networks and cloud server, then the given QoS for all users in an institution (head manager, coordinators, and regular users) are the same and accordingly all the tasks are equal. In the end, the problem is how the scheduling process can define which task is more significant than others to give it a priority to execute before others.

1.3 Research Questions

In this thesis, there are many questions that need to be answered, as below:

1- How can the system select the task which has priority, when the user used the same QoS?

2- How can the system determine the task which has a higher priority from another task?

3- How should the system select the tasks which request priority, although the different level of QoS?

4- How can the system select the task which is an important, when the QoS are equal?

1.4 Limitations and Scope

In this thesis, some limitations have noted below:

- The algorithm applied to independent tasks.
- The user level doesn't make any effect on ordering the queue that has the same user level.
- When BLJF work as SJF behavior, the task of a low-level user associated with many tasks of a high-level user. This task will be put the end of the queue and it waits for a long time till receiving the first response from the cloud system.

1.5 Objective

The main goal has proposed an algorithm for optimizing priority the task scheduling, through focused on giving classification for each user with different level to prioritize their tasks during arranging the tasks in the task queue. While the QoS in the private cloud based mainly on investigated good scheduling for the tasks that minimizes the completion time, cost, system load and it doesn't neglect the level of the users in the institution that use the cloud server. Therefore, the proposed algorithm should have an ability to change its behavior in ordering tasks in queue dynamically.

1.6 Contribution

This thesis contributes the following issues:

- Detecting the paramount factor (Level of user) that users who have higher level can relish better service.
- Explaining the four criteria of choosing priority task depend on the user level, time, cost, load.
- Explaining mainly there are groups of users, some of them request a more consequential task and others are request a customary task.

1.7 Motivation

The Scheduling on cloud computing can be relegated into both user and system level but cloud computing providers offer computer resources to users on a pay-per-use base.

The different users' needs require offering different level of QoS in the same time user priority must be considered. In a private cloud, there are mainly a multiple groups of user, the first one is important because its involving with controlling and paramount function of the organization and other who are mainly involved with the quotidian routine task. Scheduling in the private cloud is responsible for arranging tasks in the task queue in a congruous sequence through taking into consideration all the obligatory parameters to give each task opportune QoS. Most of the traditional scheduling approaches largely ignore user-priority issue, User-priority must be considered during task scheduling with the assurance that users who have higher level can relish better service, where it is largely affected on the QoS that gives to the users which they have different level.

The researcher is utilizing task scheduling proposed algorithm based on the level of users, when the tasks are determined by a scheduler and dynamically transmuting the tasks weight according to the level of users. In the private cloud environment is investigating a congruous QoS for all tasks, it should be required integrating incipient criteria to relegate users in different levels; this will give the tasks submitted from different user's different weights.

1.8 Methodology

This thesis is proposed tasks scheduling algorithm that called BLJF to simulate private cloud environment.

The BLJF selects the level of a user through adding a new parameter called Level-of-user to choose the level of the user's tasks. BLJF gives each user an appropriate QoS, therefore, it used to submit the best scheduling private cloud server and apply for tasks .BLJF represented four major parameters as following:

- Level-of-user refers to the rank of user's tasks.
- Time refers to the finished time of the task.
- Cost refers to the price of used resource by the task.
- Load refers to the private cloud system load.

1.9 Thesis Outline

This thesis contains five chapters:

- Chapter one presents the general idea of cloud computing, then discusses the main problem in this thesis that presented in scheduling tasks in private cloud and how providing appropriate solution based on proposed model.

- Chapter two explains the main concepts of cloud computing through the background which includes scheduling cloud tasks, techniques on it and private cloud benefits. In addition to, it presents samples of related work with this thesis

- Chapter three discussed the proposed methodology in details.
- Chapter four declares the experimental settings and results, also makes comparisons between previous techniques (RR, SJF) and the proposed one.
- Chapter five summarizes the conclusion of this thesis and proposes some ideas for future work.

Chapter Two

Background and Literature Review

2.1 Background

Cloud computing issue defined as a new business model for providing services and resources through the Internet. The cloud computing is a combination of several concepts from virtualization, distributed application design, grid computing, utility computing and clustering (Liu & Yang, 2013 and Mustafa, et al., 2014).

The users using the cloud for moving different tasks of preparing a minimum level of hardware and software infrastructure and focusing on their business by accessing the cloud from anywhere and anytime (Buyya,et al., 2009).

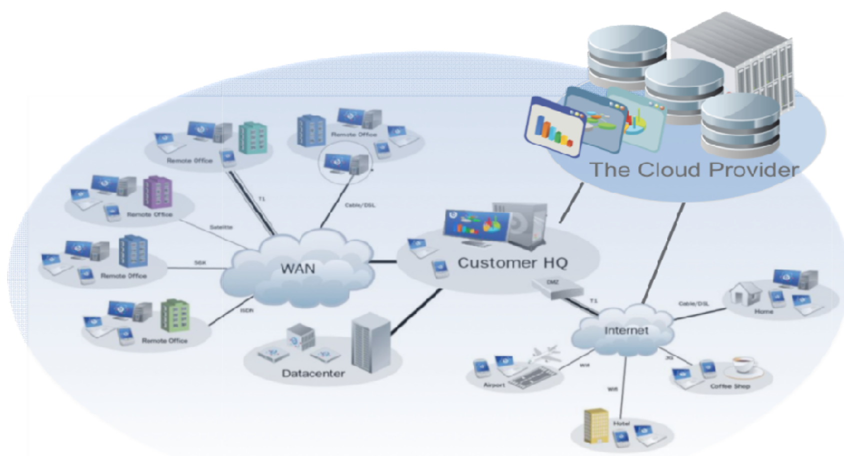


Figure (2-1): Cloud computing

There are different definitions of cloud computing, the most important definition that presented by National Institute of Standards and Technology (NIST), it is defined the cloud is a type of parallel and distributed systems, it includes several virtualized computer that presented the services and resources as a single computer based on the agreement between service provider and customer. Recently, several environments depending on the cloud such as Microsoft Azure, Amazon EC2, Google App Engine, and Aneka (Mustafa, et al., 2014 and Buyya et al. , 2009).

The main important case in cloud computing is the user pays only for the actual used without needing any details about the process. For appropriate adoption to the cloud, the users utilizing services cloud based on two points: the first point is QoS. The second point is Service Level Agreements (SLAs) which means a contract between the user and the service provider after negotiations and acceptance, this contract containing details of QoS and level, and penalties for violating the expectations (Lovesums, Krishnamoorthy,& Prince, 2014 and Buyya, et al., 2009).

2.1.1 Types of Cloud Computing Services

Cloud computing is divided into three types of services offerings as shown in figure (2-2) that explained the layered design of service-oriented in cloud computing architecture. These services are (Calheiros, et al., 2011):

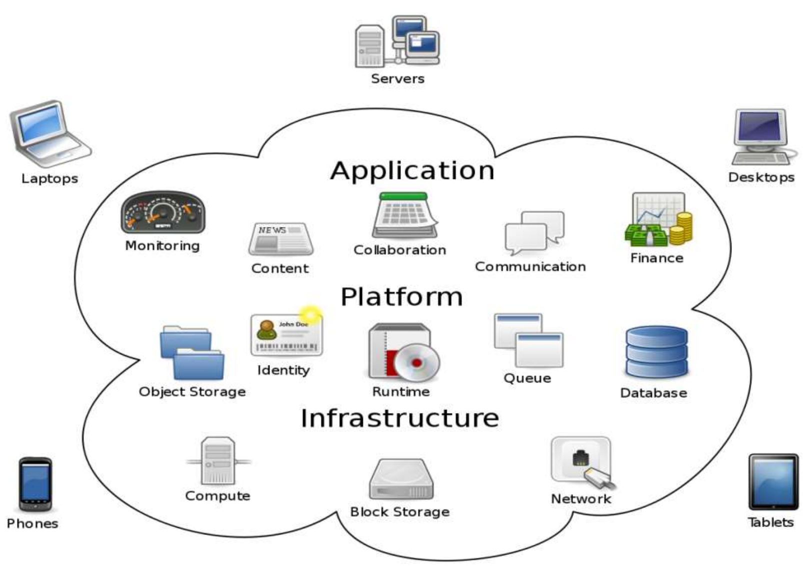


Figure (2-2): Cloud computing architecture (Buyya, et al., 2009).

i. **Infrastructure as a Service (IaaS):**It is the delivered hardware by the service provider (such as server, storage and network) and the customer is regulated associated software (such as operating systems virtualization technology, file system).

ii. **Platform as a Service (PaaS):**It is using when the service provider presenting the customer middleware capabilities to use their platform and IT solution, and proving the email and database as a solutions to use.

iii. **Software as a Service (SaaS):**It is presenting the applications as a service to the customer that runs on a cloud infrastructure; it is offering the applications to be less expensive than buying, installing and maintaining it.

2.1.2 Cloud classification

The cloud computing resources are deployed over different types of delivery models, these models classified based on the characteristics and purpose of each them as listed below (Katyal ,& Mishra , 2013 and Goyal, 2014):-

- **Private cloud**

The private cloud is provisioned for exclusive use and managed or operated by the company's own Information Technology (IT) organization for their internal use. Thus, it gives the organization high level of security.

This type is utilizing to store or manage Big-Data of an organization, at the same time; it is utilizing to provide appropriate resources according to requirements of employees or clients, an example of a private cloud (OpenStack, VMware).

- **Public cloud**

The public cloud is provisioned for general organizations or individual users using based on their requirements. It is characterized more vulnerable than private clouds, and presented the highest level of efficiency to share resources and confidentiality in the major security issue. An example of the public clouds (Amazon web services, Google Compute Engine, Microsoft).

- **Community cloud**

The community cloud is provisioned for exclusive use and controlled or managed by several organizations and users. Thus, this type integrates between characteristics of public and private clouds such as (mission, security requirements, policy, and compliance considerations).

The community cloud is similar to a private cloud, but the infrastructure and computing resources are utilizing between two or more organizations rather than a single organization, these organizations have same general issues like privacy, security, and regulatory considerations.

- **Hybrid cloud**

The hybrid cloud is a combine of two or more of cloud deployment models like public, private and community. It allows organizations to manage some resources internally and externally.

2.1.3 Benefits of cloud computing

Cloud computing has several benefits to makes the user more adaptive with the cloud, the most important benefit is reducing the cost for using it in IT infrastructure installation and management because the cloud users only need a terminal to connect to the cloud without buying any spatial hardware for using cloud (Sasikala, 2011).

In addition to, it characterized by faster implementations, lower up front and ongoing costs, Pay only for the services you need, guaranteed SLA, Predictable spending, using cloud from anywhere, lower capital expenditures, Focus on business not technology, and so on (Al-Bahadili, et al., 2013).

2.1.4 Private Cloud

The private cloud is sharing the resource within a single organization. The private cloud considered as a suitable option for many organizations, because it is including data privacy, trust, and implementing within the corporate firewall, under the control of the IT department.

The private cloud is differently on other cloud models when it affords to compute power as a service within a virtualized environment using an underlying pool of physical computing resource. It's an infrastructure for presenting the services to the user based on-demand through a self-service portal; these demands are controlled and managed by the owner organization (Subramanian, 2011).

2.1.4.1 Features and benefits of private clouds

There are many features of private clouds; the important features are (Mandel, 2016):

1. **High level of security and privacy**: Despite the development of cloud computing, the organizations still worried from adapting to the cloud due to security issues. The private cloud offered access restricted to connections made from behind one organization's firewall, and dedicated using resources in internal hosting without attention the attackers.
2. **Higher control**; the user concerns persist about loss of control over cloud resources. The private cloud has the ability to configure and manage resources in a single organization to achieve a tailored network solution.

3. **Efficiency Cost and energy**: The private cloud produces the resources in anytime and from anywhere based on user demand, or organization requirements
4. **Reliability**: The resources in the private cloud are hosted internally and providing monitoring and management component. Thus, it considered more reliability of physical infrastructure.
5. **Cloud is bursting:** The weakness of burst between internal cloud and external cloud platforms is considered a problem in current cloud solutions. The private cloud offering employ cloud bursting to allow the provider to switch certain non-sensitive functions to public cloud, it is providing more space in the private cloud for the sensitive functions.

2.1.4.2 Private Cloud Risks

Although the private cloud has many benefits, but it is exposed to several risks such as:

- The private cloud implementation considered as an investment hurdle, in addition to purchases of new hardware and software.
- The private cloud mostly needed a new operational process for relevant implementing.

2.1.5 Task Scheduling

Scheduling is a process to determine the order resource which is mapped to be executed. The main advantage of scheduling algorithm represented a high performance (Lovesum, et al., 2014)

2.1.5.1 Task Scheduling Algorithms

The main examples of scheduling algorithms are First Come First Serve Scheduling Algorithm, Priority Scheduling Algorithm and Genetic Algorithm (Shimpy & Sidhu, 2014), as shown in the following:

1. **First Come First Serve Scheduling Algorithm (FCFS):** is considered an easy method in scheduling algorithms, where the processes are ordered according to the time of arrival and submitting to the CPU.
2. **Priority Scheduling Algorithm:** This scheduling algorithm is preemptive, which are based on the priority in this type of scheduling algorithm. For example**:**

A. Shortest Job First Scheduling Algorithm (SJF) or (Min-Min algorithm):

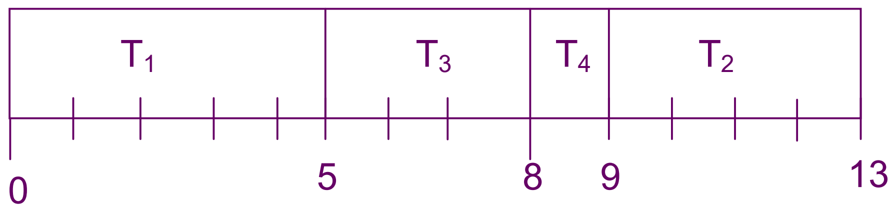
- It gives the minimum average waiting time by moving a short task time before a long one the waiting time. It may be either preemptive or non-preemptive. For example, a new task arrives has the shortest burst time at the ready queue, while a previous task is executing has the longest burst time. When is executing a new task process has the shortest burst time. A preemptive SJF algorithm will preempt the currently executing process, whereas a non-preemptive SJF algorithm allowing the currently running task to finish its Central Processing Unit (CPU) burst.

- Example of Non-Preemptive and Preemptive SJF:

Table (2.1): Example of task parameters -1

Task	Arrival Time	Burst Time
T1	0	5
T2	3	4
T3	5	3
T4	6	1

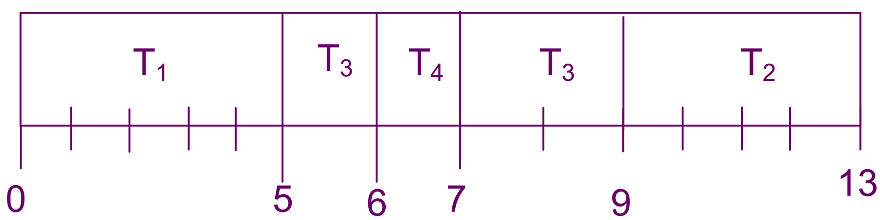
- SJF (non-preemptive)



Waiting time for T1 = 0; T2 = 6; T3 = 0; T4 = 2

Average waiting time = (0 + 6 + 0 + 2)/4 = 2

- SJF (preemptive)



Waiting time for T1 = 0; T2 = 6; T3 = 1; T4 = 0

Average waiting time = (0 + 6 + 0 + 1)/4 = 1.75

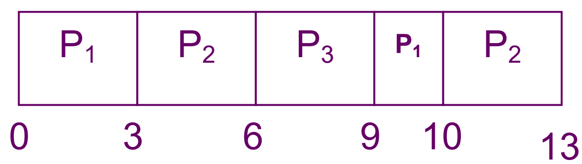
B. Round-Robin Scheduling Algorithm (RR): another type of priority scheduling technique. This type is a simplest, fairest and the most widely used of scheduling algorithms. All processes are running in a circular queue with the smallest unit of time which is called time slices or quantum.

☒ Considered the following set of processes that arrive at time 0, with the length of the CPU-burst time given in milliseconds:

Table (2.2): Example of task parameters-2

Process	Burst Time
P1	**4**
P2	**6**
P3	**3**

- If we use a time quantum of 3 milliseconds. The Gantt chart is:



C. **Max-Min algorithm:** It is the inverse of SJF by selecting the bigger tasks to be executed first.

3. **Genetic Algorithm (GA):** It is a method to solving a problem which uses the genetics model. GA is considered a search technique to find an optimized solution (Kaur & Kinger, 2014).

2.1.5.2 Task Scheduling Algorithms on Cloud Computing

Scheduling of tasks in cloud computing can define an order the jobs, where is the balance between improving the QoS, and at the same time maintaining the efficiency and fairness among the jobs by choosing the best suitable resource available for execution of tasks or to allocate computer machines to tasks in such a manner that the completion time is minimized as possible (Mohialdeen, 2013, and Singh, et al., 2014).

The main goals of task scheduling algorithm in cloud computing are reducing the response time and enhancing resource utilization (Mustafa, et al., 2014).

A very important role played in how to meet requirements cloud users of QoS and used the resources of cloud efficiently which is effective cost way.

The optimization of usage resources is represented the main reason for scheduling by improving the completed task at the minimum cost and the same time, where the user is has owned the same of QoS (Lovesum, et al., 2014).

The scheduling processes of the cloud are divided into three stages namely (Kaur, & Kinger, 2014)**:**

- **The resource is discovering and filtering :**the resources presented in the network system and collected status information about it by Data center broker.
- **Resource selection**: the resource selected target based on certain requirements of task and resource.
- **Task allocation**: Task is allocated to select the resource.

2.1.5.3 Problems with Task Scheduling

The scheduling classified as(Singh, & Ahmed, 2014):

- User level: the problem is raised by the providers and customers due to service provisioning.

- System level: the problem is raised by the resource management and data center.

2.1.5.4 Taxonomy of scheduling algorithms in Cloud

- **The tasks can be classified as** (Annette, et al., 2013)**:**
  - **Independent**: Not require any communication between tasks.
  - **Dependent:** The tasks have a type of order to be followed during the scheduling process
- **Generally two categories of the scheduling algorithm** (Singh, et al., 2014)**:**
  - **Static scheduling**: the schedule of tasks knows the environment and estimates of task execution/running time with has the information about the complete structure of tasks and mapping of resources before execution.
  - **Dynamic scheduling**: Depends on the current status of system, computer machines and the submitted tasks to cloud environment for making scheduling decisions.
- **Static strategy applied in two fashions** (Annette, at al., 2013):
  - **The heuristics based class of algorithms**: By making realistic assumptions about a priori knowledge concerning process and system loading characteristics but cannot give an optimal answer. It only requires the most reasonable amount of cost and other system resources to perform their function.
  - **The guided random search based algorithms**: GA is an example of this type, which is searched for a near-optimal solution in spaces.

The guided random has a close resemblance to the phenomenon existing in nature and also called as “nature’s heuristics”. It makes random choosing and guiding them through the problem space.

- **Dynamic strategy applied in two fashions** (Patel,& Bhoi, 2013)**:**
  - **Immediate /Online Mode:** The hot execution job, the scheduler schedules any recently arriving job as soon as it arrives with no waits for the next time interval on available resources at that moment.
  - **Batch / Offline Mode:** The scheduler stores are arriving jobs (in a queue for example) and solving the execution process over successive time intervals, so that it is better mapped a job for suitable resources depending on its characteristics.

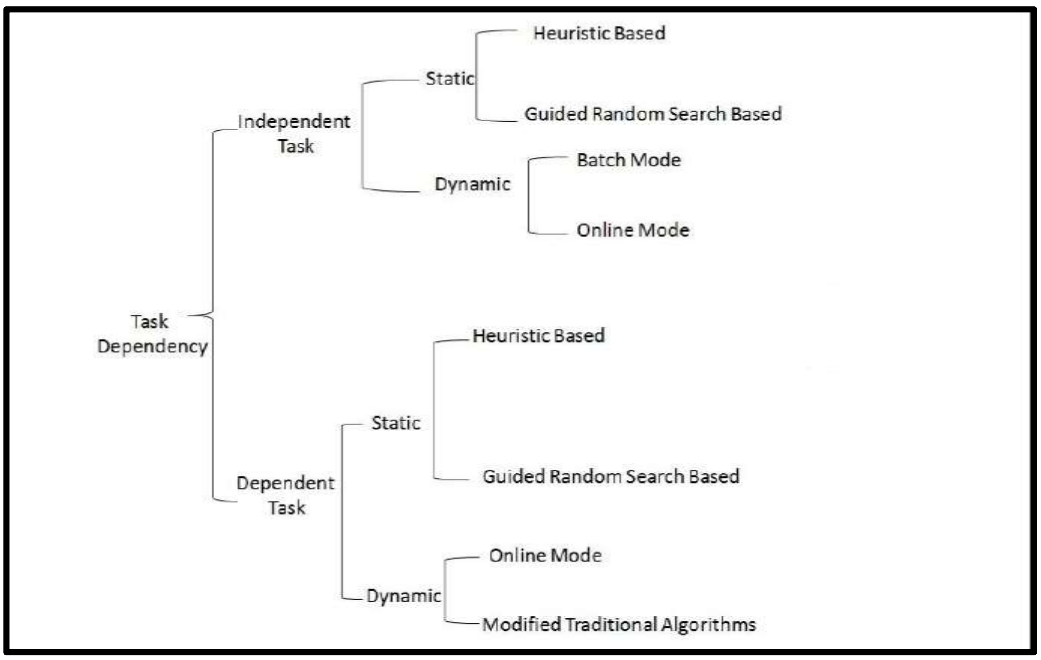


Figure (2.3): Taxonomy of scheduling algorithms in Cloud based on Task dependency (Annette, et al., 2013)

2.1.5.5 Main Parameters Task Scheduling on Cloud Computing

The main parameters are that user's concerns QoS. It is the collective effort of service performance, which determines the degree of the satisfaction of the services user. It is an extending vector; it can be described from many aspects such as complete time, cost, extension, throughput, etc. Some users wished the execution time of their application to be shorter, Where is chosen high-quality resources to serve. So it can short the execution time and finish the task as soon as possible. Other preferring the cost of execution, sometimes the cost is more expensive and hoped the price of using cloud resource is as low as possible.

The resource scheduling considered QoS constraints of the aspect user and the system load of balance mode both of them to be satisfied (Liu, et al., 2013).

Job scheduling in cloud computing needs the following:

1. **Cost:** The source scouting of cloud computing is widely distributed throughout the world. Each organization has their own management policies to provide relevant services. In which the memory usage has played a major role in calculating the cost (Lovesums, et al., 2014).
2. **Best running time**: One of the user's goals is to have the best running time on each task and can be divided into different categories according its needs (Lovesums, et al., 2014).
3. **Load Balance**: The load balance has become another important measure in the cloud. The task scheduling algorithm can maintain it (Lovesums, et al., 2014).
4. **The throughput of the system**: It is mainly used for the cloud computing systems; throughput measures the system task scheduling optimizing performance, (Kaur,&Kinger, 2014).

   The QoS is expressed to measure the completion time, latency, execution price, packet loss rate, throughput and reliability (Patel, &Mer, 2013).

2.1.5.6 Task Scheduling criteria working Cloud Computing

Scheduling algorithms criteria working on Cloud Computing are different; all algorithms are efficient in one way or another. These existing algorithms showed enhanced quality of service, consistency, maximum resource utilization, effective implementation, fairness among tasks, load balancing, minimized make span, energy efficiency, high profits and bandwidth utilization over cloud but not all at the same time (Awan, & Shah, 2015).

Figure (2.4): Scheduling algorithms criteria

FCFS, SJF, greedy method, prioritization is the traditional scheduling strategy used for task scheduling. The performance is a prime concern in any scheduling strategy (Saxena, & Chauhan, 2014).

2.2 Literature Review

There is much research for developing methods the task scheduling in a private cloud computing. In this part introduced some of these methods and discuss others.

2.2.1 Task Scheduling Algorithms in Private Cloud

Singh, et al., 2014 tested the throughput in private cloud by using SJF algorithms. Then focusing on several problems and proposed solving for it, the first problem is starvation; they solve it by using bounded waiting. The second problem is load balancing, they solve it by monitored the load and dispatch the job to the least loaded VM. The researchers assumed two groups of users in a private cloud environment, group one who are involved with controlling and important function of an organization, while the other group who are mainly involved with the daily routine task (Singh, et al., 2014).

Sanghani, et al., 2013 proposed an efficient algorithm to solve fast execution of tasks that assigned by the user. When taking scheduling decisions, this algorithm handled process requirement of the Job execution and time limit of the resource. The researchers presented the effective communication framework between broker and virtual machine for indicating the task and fetching the results in optimal time and cost using Broker Virtual Machine Communication Framework (BVCF).

The researchers implemented the proposed algorithm over Cloud reports under Virtual Machine (VM) scheduling policies at host level to enhance resource utilization. This paper is analyzed execution of cloudlets over Round Robin and proposed algorithm for Pre-emptive Shortest Job Next (PSJN). The experiment results of this paper is reducing the total execution time, making the cost low as possible and improving the CPU utilization efficiency (Sanghani, et al., 2013).

2.2.2 Power Consumption

Electric energy consumption defined as energy demand made on existing electricity supply, it measured into watt/hours.

The power consumption is used to calculate the task cost by using a CPU utilization model in CloudSim. (Dhingra, & Paul, 2014) using Linear Power Model to calculate power consumption as shown below:

P (U)= Pidle + (Pbusy- Pidle)* U

P: estimated power consumption at a given instant of time.

Pbusy: power consumed when the server is fully utilized.

Pidle: power consumed by the idle server.

U: CPU utilization may change over time due to variability of the workload.

Akijian, & Kayed, 2015 proposed a new formula to calculate power consumptions by using two factors: current CPU utilization and Random Access Memory (RAM) size. Where, all the VMs will be sorted in a descending order according to their current CPU utilization and RAM sizes. This thesis aimed to investigate power consumptions throughout different scenarios, and finding the weight of each parameter to formulate this relation and use it in the enhanced algorithm. It designs a suitable broker model to distribute the VMs and the tasks with minimum power consumption; to avoid CO_2 emissions in the environment (Akijian, & Kayed, 2015).

Parikh & Sinha, 2013 focused on priority based task scheduling optimization in cloud computing, the priority is sorted by the ratio of task's cost to its profit. The researchers aggregate the tasks into grouped according to the basis of data and requested resources by the task and prioritized. The task selection of priority formula is considered as an optimal method to gives better results over sequential scheduling, this method is selecting the resource based on cost and turnaround time by using greedy approach.

The experimental results are minimizing the turnaround time and cost of each job, therefore it minimizing the average turnaround time and cost of all submitted tasks in a time slot respectively. All these experiments improved cost and completion time of tasks as compared to Sequential Assignment (Parikh, &Sinha, 2013).

2.2.3 Priority on cloud computing

The general definition of priority issue is the arrangement of jobs priority to reduce service response time and improving performance. The priority considered as an important issue in Job scheduling environment, each Job has a priority associated with it. The aimed of the priority scheduling is arrangement of jobs implemented based on the priority; the job with high priority is served before the job with low priority. Although the benefits of priority issue, there are several problems that related to priority based Job Scheduling Algorithm such as complexity, consistency and finish time.

Patel, & Bhoi, 2013 declared the jobs that occurred in the same task can be described by several parameters (process_id, burst_time, arrival_time and deadline). The executions of Jobs are preemptive based on time quantum and if a job completes its execution before time quantum, that job is removed from queue. The jobs implementation is depending on the jobs priority. Some of the jobs have the same time delay, and then FCFS algorithm is used for scheduling jobs to implement. Other jobs have the different time delay, then the jobs having minimum time delay is selected for execution. This paper proposed Modified Prioritized Deadline Based Scheduling Algorithm (MPDSA) is proposed to execute jobs with the lowest deadline time delay in a cyclic manner using quantum time. This algorithm satisfies system requirements and supports scalability under heavy workloads (Patel, & Bhoi, 2013).

Singh and Patra, 2013 focused on the cloud workflows, the goal of Cloud workflow scheduling scheme is to make sure that the appropriate activities are executed by the correct service at the suitable time. Also, they explained the execution of cloud workflows is exposed to random factors in allocating and scheduling workload. This paper proposed an important step, in workflow scheduling scheme, is to provide an efficient workflow allocation model based on the client's requirements. The workflow scheduling model in this method will schedule jobs based on executing taking minimum possible time. The proposed algorithm works on following three steps (Singh, & Patra, 2013):

Step 1: From number of requesting jobs, cluster these jobs on the basis of their certain attributes.

Step 2: Within each cluster apply some priority algorithm to prioritize jobs on basis of certain attributes. This job attribute for prioritizing may be different from the attribute used for clustering.

Step 3: Now assign these clusters the computing environment which is capable of performing execution of jobs within cluster taking least time.

Ghanbari & Othmana , 2012 proposed a new decision-making scheduling algorithm in cloud computing environment that depends on multiple criteria. The proposed algorithm is Priority based Job Scheduling algorithm in Cloud (PJSC); it is divided into three levels of priorities:

scheduling level (objective level), resources level (attribute level) and job level (alternative level). In PJSC algorithm, priority vectors are calculated in a cloud environment for all resources and jobs according to an allocation of resource. The priority for all jobs are compared with other jobs separately and determining the resource that has higher priority than others based on decision maker(s) (Ghanbari, & Othmana, 2012).

Salot, and Gandhi, 2013 proposed a new scheduling algorithm based on the priority of resource and job ratio, the requested job of resources determines this priority. The priority of all jobs is compared with other jobs separately; it is creating comparison matrixes of jobs according to the priority of resource accessibilities. These comparison matrixes are utilizing to compute a priority vector (vector of weights). After that, they are creating a comparison matrix for resources according to priorities. This matrix determines the resource that has higher priority than others based on decision maker(s). The next step of the proposed algorithm is to calculate PVS; it is denoted as priority vector of scheduling jobs. Finally, they choose the maximum element of PVS, and selected the appropriate corresponding element in order to allocate a suitable resource (Salot, & Purnima, 2013).

2.2.6 **Classification Criteria on Task Scheduling Algorithms for Cloud Computing:**

A large number of studies in task scheduling algorithm worked on cloud computing. The task scheduling algorithms in cloud environment classified a category depends on some criteria as the following:

2.2.6.1 Priority Based Task Scheduling

Liu, et al., 2013 proposed in this paper a priority-based method to balance the parallel workloads in the cloud. They are using virtualization technologies to partition the computing capacity of each node into two levels. The first level foregrounds VM with high CPU priority, while the second level is background VM with low CPU priority. This scheduling algorithm used parallel jobs way to make efficient use of the two levels VMs to improve the responsiveness of these jobs. The result shows that the parallel scheduling algorithm significantly outperforms commonly used algorithms such as the extensible Argonne scheduling system in a data center setting. The method is more effective for consolidating parallel workload in data centers (Liu, et al., 2013).

Ghanbari, & Othman, 2012 classified three levels of priority:

- Scheduling level (objective level)
- Resource level (attribute level)
- Job level (alternative level)

Each job priority is separately compared with other jobs and determined how one have a higher than other to be executed the first. The drawback of this algorithm is complexity issue and makespan (finish time) (Ghanbari, & Othman, 2012).

2.2.6.2 Reduced Makespan Based Task Scheduling

Bhoi, et al., 2013 represented an algorithm to reduce makespan and the load balanced across resources. The improved way approached by changing max-min to make the task has lowest execution time which it has the largest resource. While is the task has the highest execution time because it has the lowest resource. Finally, selected the nearest greater or equal average task to executed resource with minimum time (Bhoi, et al., 2013).

Choudhary, & Peddoju, 2012 presented an algorithm that gives improved cost and better completion timing of tasks. They merge of three different approached (Task Grouping, Prioritization, and Greedy Allocation) .It worked on two constraint groups:

1) Deadline constrained tasks.
2) Cost based tasks.

When the tasks are received a task broker groups according to above constraint categories, priorities are assigned to each task based on their group type (Choudhary, & Peddoju, 2012).

Zhao, et al., 2009 defined a GA model. It is based on Service Oriented Architecture (SOA) Model. GA took into account both time utilization and resource utilization factors to achieve better results. The result shows optimal resources utilization, through this algorithm showed advantageous results (Zhao, et al., 2009).

2.2.6.3 Energy Efficient Task Scheduling

Bitam, 2012 used an algorithm for neighborhood search like bees in nature. The optimizing Bees Life Algorithm (BLA) is applied by two genetic operators (Crossover and Mutation). The results are shown the efficiency of the algorithm, but it only focused on makespan which could result in unfairness among tasks (Bitam, 2012).

Liu, et al., 2013 used an algorithm which focused on minimize energy consumption and maximize the profit of service providers under the constraint of deadlines. This algorithm obtained the high profits and consumed less energy. But for the higher arrival rates, it gave the worse results, failed the number of applications and increased the arrival to get higher rates (Liu, et al., 2013).

2.2.6.4 Improved Cost Based Task Scheduling

Selvarani, & Sadhasivam, 2010 worked to provide the algorithm. It is divided the tasks into three different lists based on their priority level. The available resources are allocated according to the priority lists and grouping. The algorithm is applied the schedule tasks to each list, where It is aiming to minimize the total tasks time and cost (Selvarani, & Sadhasivam, 2010).

2.2.6.5 QoS-Based Task Scheduling

Han, et al., 2013 suggested the algorithm, which is mainly considered the QoS requirements because it has a resource heterogeneity properties and executed the low complexity in the cloud environment. The algorithm is worked on improved a performance in context by shortening makespan (completion time), and it worked on the tasks scheduled for groups on the resources manner with high QoS capability of tasks is providing executed first. The drawback, the tasks a higher QoS requests is be needed to be executed first this may result in the delay of the tasks that needed the low QoS requests (Han, et al., 2013).

Ning, et al., 2013 suggested the algorithm. It focused on investigation the higher QoS by the lower cost of service. It also considered the factors affecting the cloud overall performance, including the cloud cost of service, node's load rate, bandwidth utilization, and network delay (Ning, et al., 2013).

2.2.6.6 Improved Consistency Based Task Scheduling

Ergu, et al., 2013 focused on the weight for each task because it is considered the original storage size. Therefore, they represented the algorithm according to work the principle of assigning cloud storage and cloud resources of the reciprocal tasks (Ergu, et al., 2013).

2.2.6.7 Enhanced Load Balancing Based Task Scheduling Techniques in Cloud

Fang, et al., 2010 suggested the technique for scheduling task. It is showed results by utilizing the higher resource and improving the load balance in the cloud (Fang, et al., 2010).

Li, et al., 2011 suggested in this scheduling technique utilized the characteristic of Ant Colony Optimization (ACO). It worked on decreasing computation time for the tasks execution, managed workload on each VM in the cloud and minimized the makespan and load balance of the entire system (Li, et al., 2011).

2.2.6.8 Improved Fairness Based Task Scheduling Techniques in Cloud

Xu, et al., 2011 focused on the interpretation of distributive justice to the dynamic tasks requests to different users various services in the cloud environment. User tasks are classified and depended on QoS parameters which are completion time and bandwidth. Based on idea of Berger Model two fairness constraints are defined:

1) Task justice

2) System justice (Xu, et al., 2011).

Mustafa, et al., 2014 declared ordering the jobs according to Million Instructions Per Second(MIPS), memory size, and bandwidth of the resource. This technique reduced the execution time of jobs, utilized grid resources sufficiently, delayed network to schedule and executed jobs. But, the algorithm doesn't parallel schedule resource. The tasks are sorted according to their priority and they are put into three different lists based on three levels of priority. They are a high, medium and low priority. The algorithm is divided into two parts. The first part is called a task schedule algorithm is the classification of task depended on service type of task. The second part is called a group of task and mapping with a computing resource (Mustafa, et al., 2014).

2.5 Task scheduling in Private Cloud

Shinde, & Kadam, 2014 presented a private cloud which is a type of cloud computing platform within a unique community. It is implemented in the corporate firewall under the control of the IT department. That means all users of the organization who owns this cloud just can use the resources provided by the private cloud and access it. Amazon Elastic Container Service (ECS) provides the service schedule for long-running tasks and applications. ECS provided optimistic concurrency controls so multiple schedulers can be operating at the same time; the cluster manager can confirm that the resource is available and commit it to the scheduler. When a task is started on a container instance, it can pass through several states before it finishes on its own or stops manually. Some tasks are running as batch jobs that naturally progress through from pending to running then stops. Other tasks continue running indefinitely which can be part of a service (Shinde, & Kadam, 2014).

Barclay, 2016 described the scheduler can listen for events from the cluster manager and take action, such as maintaining the availability of your applications, or interact with other resources like Elastic Load Balancing (ELB) (Barclay, 2016).

Krisragh, 2015 allowed the actions for declaratively describing which run in the cloud. It created, maintained, and invoked to scheduled work. The scheduler doesn't host any workloads or run any code. It only invoked code hosted elsewhere in Azure, on-premises, or with another provider. The scheduler allows you to create, update, delete, view, manage jobs and collections job programmatically by using scripts in the portal (Krisragh, 2015).

Chapter Three

The Proposed BLJF Scheduling Algorithm

3.1 Brief

The cloud computing is considered as a new technology during this decade. Still, the users are poorly dealing with this technology which is enabling the users to store their data in the optimal time.

In this thesis, the problems summarized as needs to select a procedure for executing the process depend on level-of-user, time, cost and load system.

3.2 Software Tools Used in the Research

3.2.1 The Java Development Kit (JDK)

It is produced by Oracle Corporation in the form of a binary product. JDK aimed Java developers on Solaris, Linux, Mac OS X or Windows. The JDK contains a private JVM and a few other resources to finish the development of the Java Application.

3.2.2 CloudSim

It can be used to check the correctness of proposed algorithm. It used to verify the task grouping and scheduling in a simulation cloud computing environment, this simulation based on java language, supported both system and modeling of cloud computing system components (Parikh, & Sinha, 2013, and Mustafa, et al., 2014).

CloudSim simulator is utilizing for enabling modulation and simulation of Cloud computing environments. It is offering classes for describing data centers, virtual machines, applications, users, computational resources, and policies for management of diverse parts of the system (e.g., scheduling and provisioning).

The benefits of Cloudsim are (Buyya, et al., 2009):

(i) Free-using (no cost) is testing services in the repeatable and controllable environment.

(ii) The performance is testing the bottlenecks before deploying on actual cloud.

As shown in figure (3-1), the layered is designed of the CloudSim software framework and architectural components, where is the top layer to the user code that exposes basic entities for hosts (number of machines, their specification, and so on), applications (number of tasks and their requirements), VMs, number of users and its application types, and broker scheduling policies (Calheiros, et al., 2011).

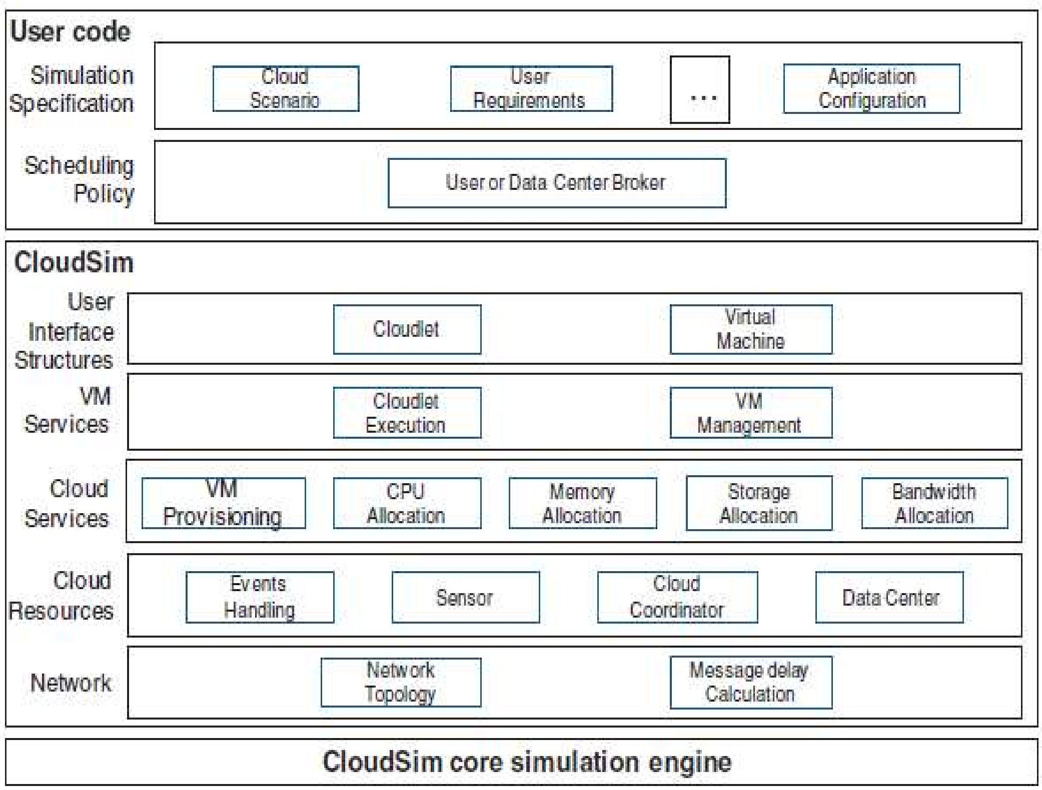


Figure (3-1) … Layered CloudSim architectural (Calheiros, et al., 2011).

- **A Datacenter** is containing a set of hosts, which are responsible for managing VMs during their life cycles .It is offering the providers resource in a cloud computing environment (memory, cores, capacity, and storage).This class model the core infrastructure level services (hardware, software) . The component of datacenter is the instantiation a generalized resource provisioning component that implements a set of policies for allocating bandwidth, memory, and storage devices, as shown in figure (3-2).

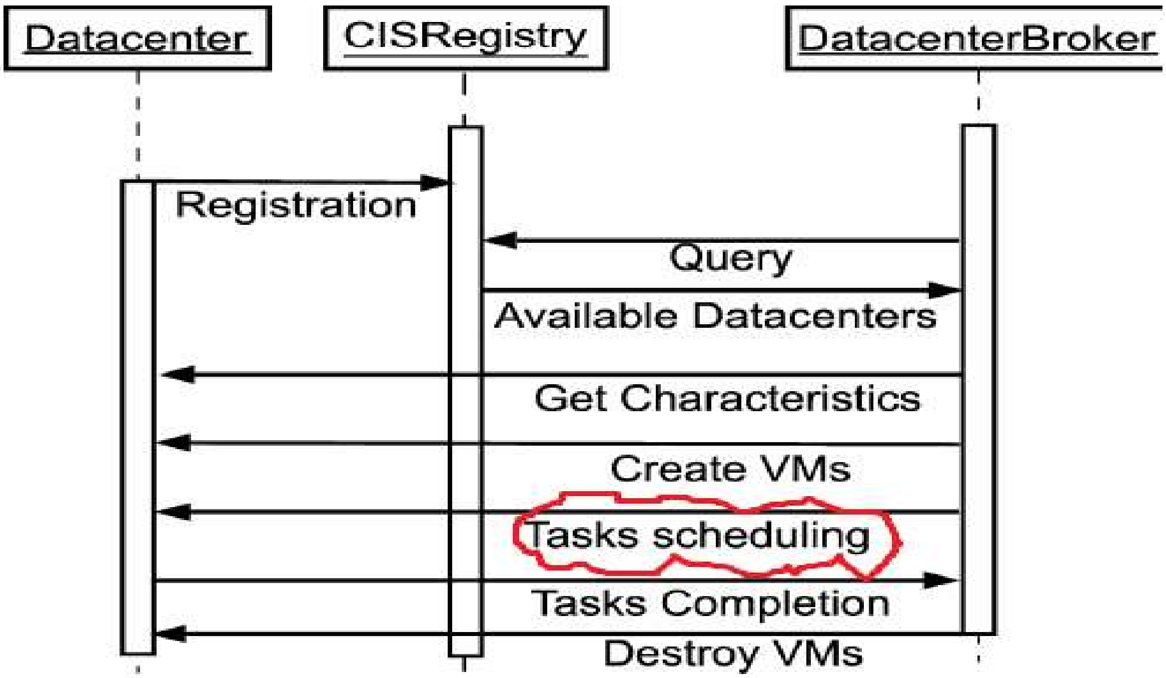


Figure (3-2): Simulation Data Flow (Calheiros ,et al., 2011).

- **The cloud information services (CIS)** is cloud computing service to provide the information about all the registration of computing resource. The resource information of cloud computing collected by CIS, such as operating system (windows, Linux), management policy (time share, space share), resource index and processing capability MIPS. Addition to CIS provides information to the user on the availability of the resources.

The Information collected by the collector from the CIS. It is collected available of the computing resource and get its characteristic such as processing capability MIPS and memory size to each available of a resource through CIS.

The Information collects by collector from cloud information service. It is collected available of the computing resource and get its characteristic such as processing capability MIPS and memory size to each available of resource through cloud information service (Mustafa, et al., 2014).

- **Datacenter Broker**: it is a class models responsible for transferring between users and service providers, depending on user's QoS requirements and deploys service tasks across Clouds. Cloud broker sends requested to the cloud service provider for the QoS, these requested and the list of VM is created on the VM server. Here tasks priority is assigned according to the QoS value of task (Buyya , et al., 2009 and Mathukiya, & Gohel , 2015).

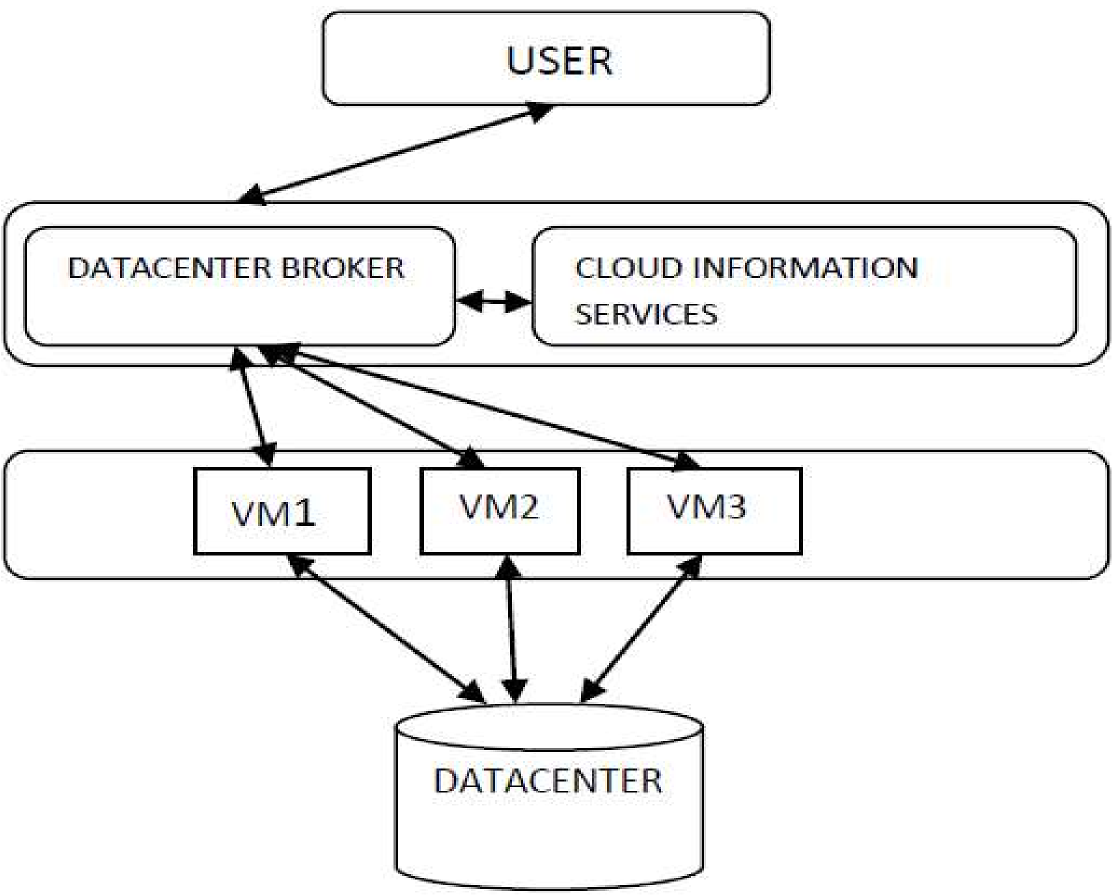


Figure (3-3): Scheduling Tasks in Cloud Computing (Shimpy , &Sidhu , 2014)

- **Host:** The host represents a physical computing node in the cloud. It is created a pre-configured processing ability (expressed in MIPS), memory, storage, and a scheduling policy for allocating processing cores to virtual machines (Buyya, et al., 2009).

Depending on basic functionalities by CloudSim, the researcher is using the CloudSim as a framework for modeling and to simulation the cloud computing infrastructures and services. It is built primarily on the Cloud Computing and Distributed Systems Laboratory. The researcher would be able to test its scenarios and configurations, then allowing the development of best practices in all the critical side related to Cloud Computing

3.2.3 Net-Beans

NetBeans a Platform allows the developer to develop its applications by using modules. The Platform is written in Java and having a set of tools. Net-Beans began in 1996 as a student project called Xelfi, which had the goal of writing a "Delphi-like" Java IDE in Java. Later Sun Microsystems decided to make it open source in 2000.

Today, the Net-Beans software is developed by a community of volunteers in collaboration with employees at Sun Microsystems(Kærvang, et al., 2013). It provides built-in support for developing in Java, C, C++, XML, and HTML. Net-Beans can run on any platform with a Java Virtual Machine (Schreckmann, 2014).

A current version is called (Net-Beans IDE 8.0). It is an open source IDE that is backed and supported by a vast and robust community of developers who makes sure that the product is constantly updated and that you can get help and expert tutorial at all times. It's allowed create websites and applications for desktop and mobile.

3.3 Parameters used in BLJF

Private cloud resources are shared by different users in the same organization, which is considered as a big challenge to scheduling tasks efficiently in the private cloud systems. The private cloud uses different scenarios for scheduling task to achieve users' requests. However, it still needs a new task scheduling technique to become more efficiency and adaptable because there is a different level of users and the highest level of users must enjoy in special services. This thesis implements a new tasks scheduling algorithm called BLJF, where it simulates in private cloud environment.

BLJF takes into consideration the level of a user, by adding a new parameter named level-of-user to determine the class of the user's tasks that submitted to the private cloud server and apply the best scheduling for submitted tasks which gives each user an appropriate QoS.

BLJF considers the following four parameters:

- **Level-of-User (LoU):** refers to the rank of the user (user's tasks) in the institution. Which having the main effect in the performance of the proposed scheduling algorithm in the study that implement in the private cloud.
- **Time:** refers to the completion time of the task.
- **Cost:** refers to the price of used resource by the task.
- **Load:** refers to the private cloud system load.

BLJF algorithm works in the private cloud that serves a number of levels of users: (**1…N**), where level **1** represents the highest level user and level **N** represents the lowest level user. The user level parameter has been given higher weight than all other parameters that are used in the proposed tasks scheduling algorithm in the private cloud.

3.3.1 Steps of BLJF

A clear explanation is giving about the main operations that are doing in the proposed tasks scheduling algorithm (BLJF), the steps of the proposed tasks scheduling algorithm, with numerical examples, are given below:

Step1: Determine the values of the four parameters for each submitted task (LoU, Time, Cost, and Load). (See appendix C).

Step2: Create the parameters vector for each submitted task taking into consideration to put the LoU parameter first:

LoU	**Time** or **Cost** or **Load**	**Time** or **Cost** or **Load**	**Time** or **Cost** or **Load**

Step3: Calculate the weight for each submitted task from the values in the parameters vector of the task. The weight value is the concatenation of the parameters values in the vector.

Example: Assume we have the following ranges of values (for ease of explanation) for the four parameters used:

- LoU= (1 to 5), where 1 represents the high level and 5 represents the low level
- Time = (1 to 10) msec

- Load=(1 to 100) MB
- Cost=(1 to 100) $

Suppose the order of the parameters values in the vector as:

LoU	Time	Cost	Load

And the submitted tasks with their parameters values are listed in Table (3-1). The values of weights, in the last column of the table, are calculated by concatenating the values of the four parameters for each task based on the order of the parameters in the vector above.

Table (3-1): Values of parameters and the weights for the submitted tasks

Task name	LoU	Time	Cost	Load	Weight
T_1	3	7	30	52	373052
T_2	1	5	60	23	156023
T_3	2	3	15	23	231523
T_4	1	8	33	48	183348
T_5	2	9	41	23	294123

Step4: Order the submitted tasks in the scheduling Queue (Q) (in ascending order) based on the calculated weights for the tasks in Step 3.

Queue (Q):

T_2	T_4	T_3	T_5	T_1	………..

Step5: Execute the next desired task from the scheduling Queue (Q) (either as Round Robin or SJF algorithm).

Step6: If there are new submitted tasks go to Step1, otherwise go to Step 5.

3.3.2 Flowchart of BLJF

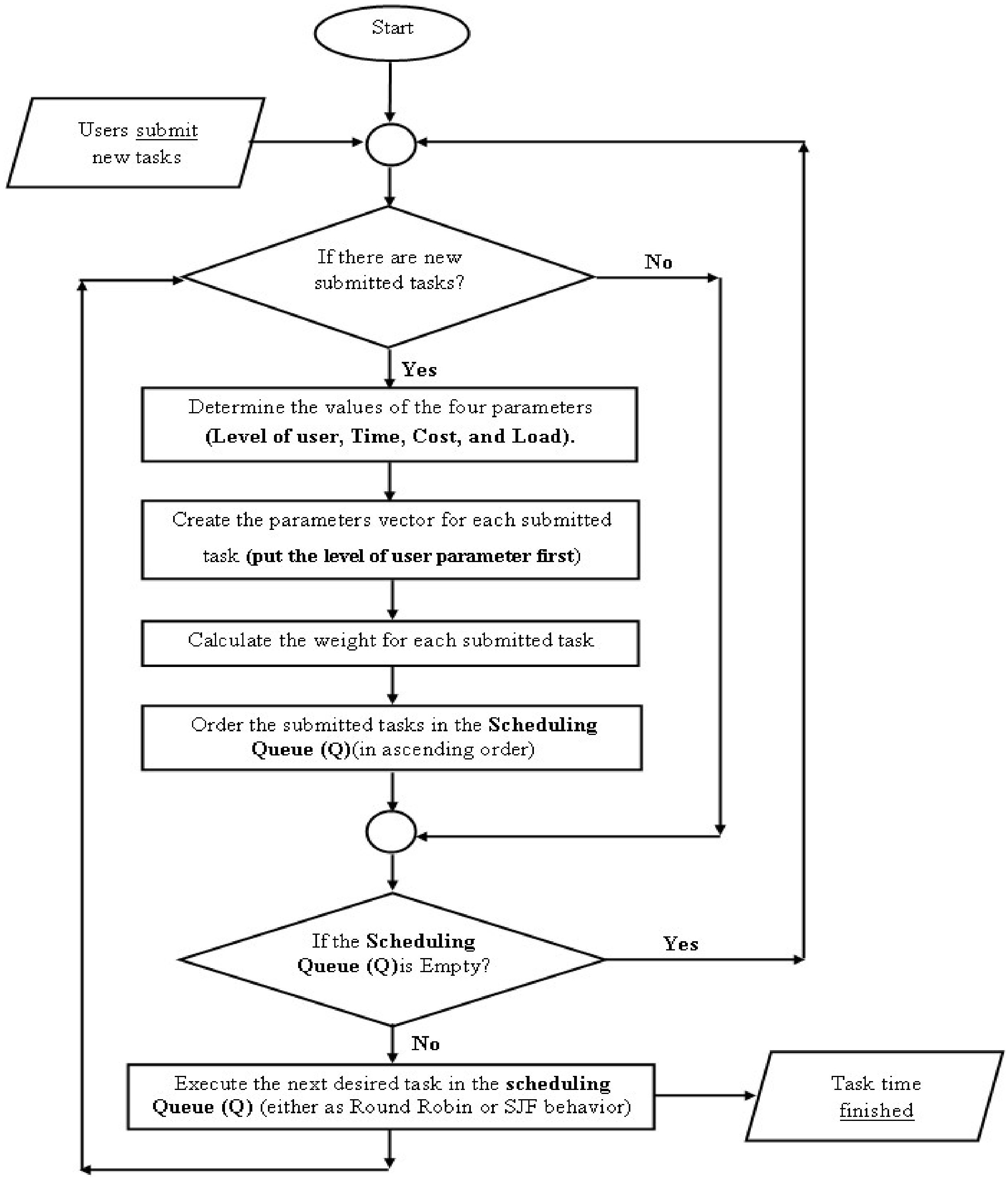


Figure (3-4): Best- Level-Job-First Diagram (BLJF)

3.4 Implementation of BLJF

The proposed task scheduling algorithm have been tested and implemented in two behaviors: (RR and SJF)

3.4.1 BLJF as RR:

The researcher supposed the parameters values (LoU, time, cost, and load) of the submitted tasks are:

Table (3-2): Submitted tasks with their parameters values

Task Number	Time	Cost	Load	LoU
0	16	306	2881	4
1	8	157	924	5
2	36	681	2601	2
3	24	448	3239	5
4	26	493	1429	3
5	56	1033	2233	2
6	3	56	2386	2
7	2	33	59	3
8	16	298	1133	1
9	52	970	2663	1
10	7	122	2269	3

The parameters vector is created for each submitted task (Level-Of-User first).

LoU	Time	Cost	Load

The weight is calculated for each submitted task.

Table (3-3): Weights of the submitted tasks in Table (3-2)

Task Number	LoU	Time	Cost	Load	Weight
0	4	16	306	2881	4 0016 0306 2880
1	5	8	157	924	5 0008 0157 0924
2	2	36	681	2601	2 0036 0681 2601
3	5	24	448	3239	5 0024 0448 3239
4	3	26	493	1429	3 0026 0493 1429
5	2	56	1033	2233	2 0055 1032 2233
6	2	3	56	2386	2 0003 0056 2386
7	3	2	33	59	3 0002 0033 0059
8	1	16	298	1133	1 0016 0298 1133
9	1	52	970	2663	1 0052 0970 2663
10	3	7	122	2269	3 0007 0122 2269

Order the submitted tasks in the Queue:

Table (3-4): Order of Tasks Based on the Calculated Weights

Task Number	LoU	Time	Cost	Load	Weight
8	1	16	298	1133	1 0016 0298 1133
9	1	52	970	2663	1 0052 0970 2663
6	2	3	56	2386	2 0003 0056 2386
2	2	36	681	2601	2 0036 0681 2601
5	2	56	1033	2233	2 0055 1032 2233
7	3	2	33	59	3 0002 0033 0059
10	3	7	122	2269	3 0007 0122 2269
4	3	26	493	1429	3 0026 0493 1429
0	4	16	306	2881	4 0016 0306 2880
1	5	8	157	924	5 0008 0157 0924
3	5	24	448	3239	5 0024 0448 3239

Quantum time is calculated based on the time values of the submitted task in the queue Q. The quantum time is the median of the tasks time values for the tasks in the queue Q: Median =0.164 msec.

The next task desired is getting from the Queue (Q) and it is executing (for the 0.164 msec period) using Round Robin method based on the calculated quantum time (0.164).

T8	T9	T6	T2	T5	T7	T10	T4	T0	T1	T3	T9	T2	T5	T4	T0	T3	*T9*	*T2*	*T5*	*T9*	*T5*
Round 1											**Round 2**						**Round 3**				

3.4.2 BLJF as SJF

The researcher supposed the parameters values (levels of users, time, cost, and load) of the submitted tasks are:

Table (3-5): Submitted Tasks With Their Parameters Values

Task Number	Time	Cost	Load	LoU
0	16	306	2881	4
1	8	157	924	5
2	36	681	2601	2
3	24	448	3239	5
4	26	493	1429	3
5	56	1033	2233	2
6	3	56	2386	2
7	2	33	59	3
8	16	298	1133	1
9	52	970	2663	1
10	7	122	2269	3

The parameters vector createdfor each submitted task (Level-of-User first).

Level of User(LoU)	Time	Cost	Load

The weight calculated for each submitted task.

Table (3-6): Weights of the Submitted Tasks in Table (3-5)

Task Number	LoU	Time	Cost	Load	Weight
0	4	16	306	2881	4 0016 0306 2880
1	5	8	157	924	5 0008 0157 0924
2	2	36	681	2601	2 0036 0681 2601
3	5	24	448	3239	5 0024 0448 3239
4	3	26	493	1429	3 0026 0493 1429
5	2	56	1033	2233	2 0055 1032 2233
6	2	3	56	2386	2 0003 0056 2386
7	3	2	33	59	3 0002 0033 0059
8	1	16	298	1133	1 0016 0298 1133
9	1	52	970	2663	1 0052 0970 2663
10	3	7	122	2269	3 0007 0122 2269

Order the submitted tasks in the Queue:

Table (3-7): Order of Tasks Based on the Calculated Weights

Task Number	LoU	Time	Cost	Load	Weight
8	1	16	298	1133	1 0016 0298 1133
9	1	52	970	2663	1 0052 0970 2663
6	2	3	56	2386	2 0003 0056 2386
2	2	36	681	2601	2 0036 0681 2601
5	2	56	1033	2233	2 0055 1032 2233
7	3	2	33	59	3 0002 0033 0059
10	3	7	122	2269	3 0007 0122 2269
4	3	26	493	1429	3 0026 0493 1429
0	4	16	306	2881	4 0016 0306 2880
1	5	8	157	924	5 0008 0157 0924
3	5	24	448	3239	5 0024 0448 3239

The next desired task is getting from the Queue (Q) and it is executing used SJF.

T8	T9	T6	T2	T5	T7	T10	T4	T0	T1	T3

Chapter Four

The Experimental Results

4.1 Dataset Used in The Experiments

CloudSim tools used to implement different tasks of parameters values in thesis experiments (dataset). The dataset was generated by CloudSim (about 120 tasks) randomly (see appendix A), and the ranges of values for the tasks parameters used in the experiments were:

- LoU = (1 to 10), where 1 represents the high level and 10 represents the low level
- Time = (1 to 100) msec
- Load = (1 to 50) MB
- Cost = (1 to 20) $

The order of values in parameter vector used in the experiments as:

LoU	Time	Cost	Load

The implementation code of the proposed task scheduling algorithm BLJF has been written by using VB.net (2010), this code is implemented to extract results that used to evaluate the performance of the proposed task scheduling algorithm. The computer system specification used to implement the algorithm is:

- Processor: Intel® core™ i3-5005u CPU @ 2.00GHz 2.00 GHz
- Installed memory (RAM): 4GB DDR3l
- Hard disk drive: 500 GB
- Operating System: Win10 Home

4.2 Measurements used to evaluate scheduling algorithm

In this thesis, the researcher used two measurements to evaluate the performance of any tasks scheduling algorithm. The researcher has been used Average Response Time (ART) and Average Waiting Time (AWT) measurements to evaluate the performance of the BLJF.

Appropriate tasks scheduling algorithm is investigating the lowest values for both ART and AWT. In the following two subsections, the first comparison is performed between Round-Robin and BLJF based on calculated quantum time and the second comparison is performed between SJF and BLJF without using the quantum time.

4.2.1 The comparison between RR and BLJF

BLJF and Round-Robin algorithms have been implemented on the same dataset, and the values of AWT and ART of these algorithms have been registered in Table (4-1) and Table (4-2) respectively.

Table (4-1): BLJF (as RR behavior)

Best Level Job First (BLJF)		
LoU	**AWT**	**ART**
1	**22.91**	**1.88**
2	**15.08**	**5.00**
3	**22.89**	**7.90**
4	**26.65**	**10.92**
5	**22.98**	**14.70**
6	**31.07**	**19.31**
7	**30.38**	**23.95**
8	**34.61**	**27.60**
9	**34.90**	**30.37**
10	**37.88**	**32.83**
Total Average	**27.94**	**17.45**

Table (4-1) represents the average waiting time and the average response time for assumed 10 levels of users by using proposed algorithm (BLJF).and Table (4-2) represent the average waiting time and the average response time for assumed 10 level of users by using Round Robin algorithm (RR).

Table (4-2): Round Robin

RR		
LoU	**AWT**	**ART**
1	**32.62**	**18.26**
2	**29.85**	**21.49**
3	**27.67**	**15.04**
4	**17.13**	**9.95**
5	**29.86**	**20.71**
6	**28.27**	**16.42**
7	**26.25**	**16.81**
8	**28.17**	**20.84**
9	**27.33**	**17.37**
10	**20.11**	**11.37**
Total Average	**26.72**	**16.83**

By comparing the two tables (Tables 4-1 and 4-2) , It is clearly defined that the total average (AWT and ART) for the two algorithms (BLJF and RR) is close, but by using proposed algorithm(BLJF) the tasks which have the high level of user have least ART than RR algorithm.

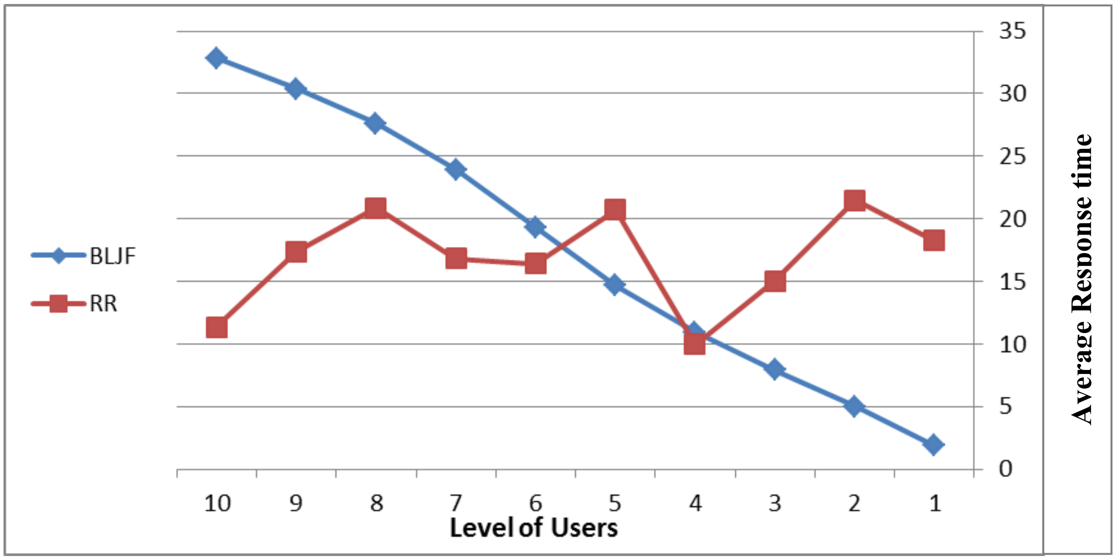


Figure (4-1): ART of RR and BLJF

Figures (4-1 and 4-2) show the comparison between the recorded values of ART and AWT (Tables 4-1 and 4-2) in both BLJF and RR algorithms. These results shows the proposed algorithm BLJF succeeded to achieve minimum response and waiting time for the highest level users compared to the response and waiting time for the lowest level of users.

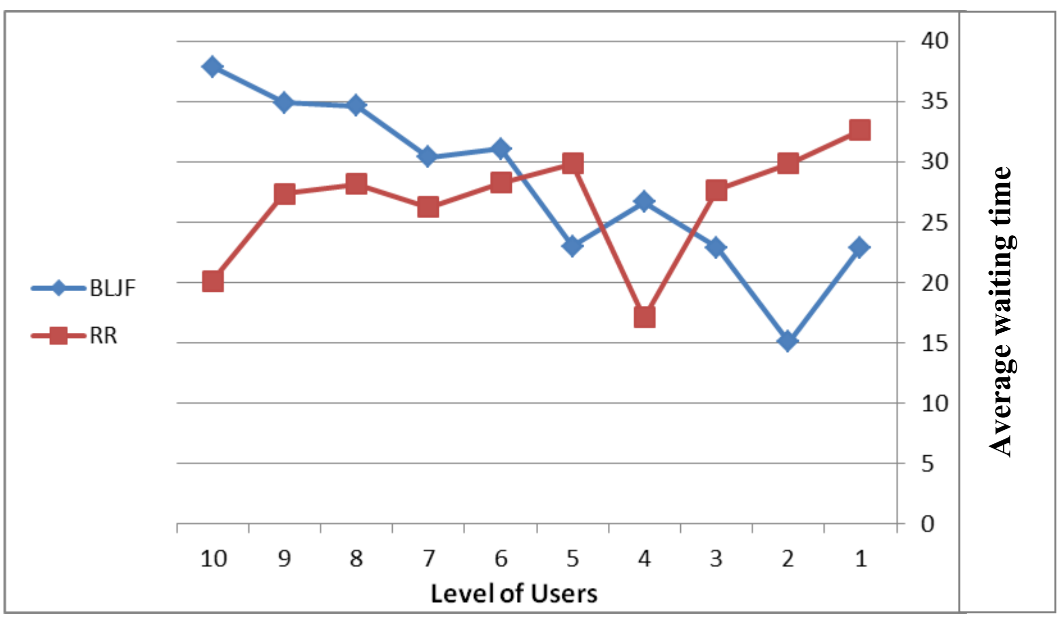


Figure (4-2): AWT of RR and BLJF

Above experiments conclude that the BLJF has been achieved the highest level of users better QoS. This experiment has been implemented by decreasing waiting time, which is spending in the scheduling queue, and shorter time to response for these users by the private cloud server. This has been done with simple changes in the total average waiting and response time.

4.2.2 The comparison between SJF and BLJF

BLJF and SJF algorithms have been implemented on the same dataset, and the values of AWT and ART of these algorithms have been registered in Table (4-3) and Table (4-4) respectively.

Table (4-3): BLJF (as SJF behavior)

Best Level Job First (BLJF)		
LoU	**AWT**	**ART**
1	2.58	2.58
2	6.50	6.50
3	10.29	10.29
4	13.28	13.28
5	17.64	17.64
6	23.78	23.78
7	30.19	30.19
8	34.27	34.27
9	37.96	37.96
10	41.25	41.25
Total Average	21.77	21.77

Table (4-3) represent the average waiting time and the average response time for assumed 10 level of users by using proposed algorithm (BLJF) when changing its behaviors (as short job first) . And Table (4-4) represents the average waiting time and the average response time for assumed 10 levels of users by using Short job first algorithm (SJF).

Table (4-4): SJF

Short Job First (SJF)		
LoU	**AWT**	**ART**
1	**20.21**	**20.21**
2	**11.03**	**11.03**
3	**19.75**	**19.75**
4	**8.20**	**8.20**
5	**15.00**	**15.00**
6	**16.87**	**16.87**
7	**13.90**	**13.90**
8	**13.09**	**13.09**
9	**14.61**	**14.61**
10	**9.55**	**9.55**
Total Average	**14.22**	**14.22**

By comparing the pervious tables (Tables 4-3 and Table 4-4), It is clearly defined that the proposed algorithm (BLJF) offering to the tasks which have high level of user less AWT than SJF algorithm. Against the lower level of user which have a high AWT than SJF algorithm

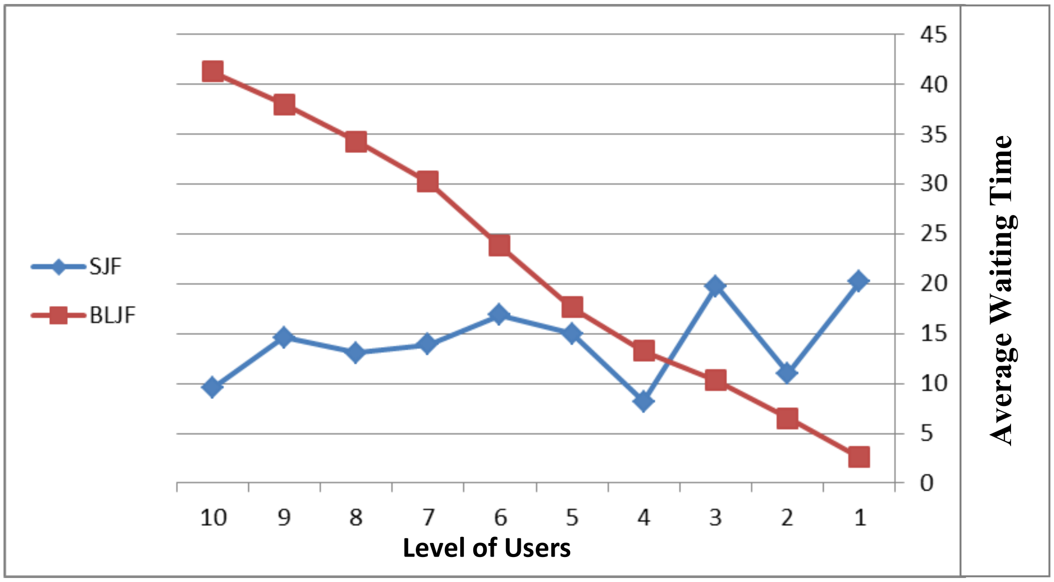


Figure (4-3): AWT of SJF and BLJF

Figures (4-3 and 4-4) show the comparison between the recorded values of ART and AWT (in Tables 4-3 and 4-4) in both BLJF and SJF algorithms. These results show the proposed algorithm (BLJF) succeeded to achieve minimum response and waiting time for the highest level of users compared to the response and waiting time for the lowest level users.

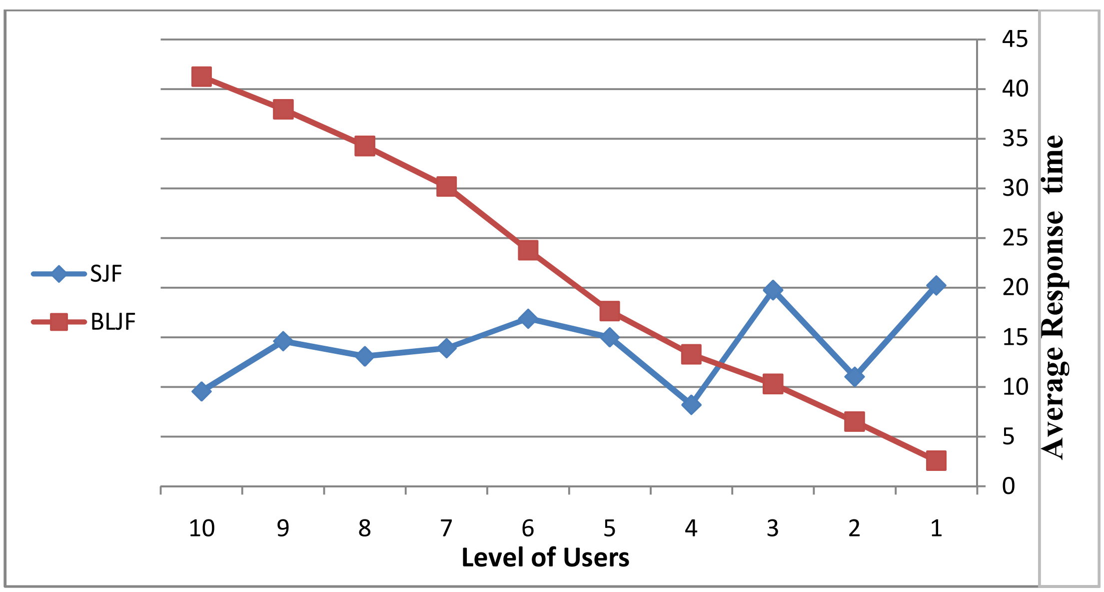


Figure (4-4): ART of SJF and BLJF

Above experiments conclude that the BLJF has been achieved the highest level of users better QoS. This experiment has been implemented by decreasing waiting time, which is spending in the scheduling queue, and shorter time to response for these users by the private cloud server. But in this case, the total average waiting and response time have been increased more than the previous case (between Round-Robin and BLJF).

4.3 Adaptation characteristic of BLJF

The proposed algorithm has a special characteristic; it can change behavior by reordering the tasks based on the priority vectors. The two sections below will be clarifying the proposed algorithm work as the Short Job First (SJF), and as the Round Robin algorithm (RR).

Another dataset has been generated randomly using CloudSim (see appendix B). The ranges of values for the tasks parameters used in the experiments were:

- LoU = (1 to 5), where 1 represents the high level and 5 represents the low level
- Time = (1 to 100) msec
- Load = (1 to 50) MB
- Cost = (1 to 20) $

4.3.1 The BLJF works as the Short Job First(SJF)algorithm

The proposed algorithms can change behavior by ordering the priority vector as:

Time	Level Of User(LoU)	Cost	Load

After generating a new number of tasks which have five levels of users, Table (4-5) represents the AWT and ART for BLJF algorithm when change its priority vector as (Time|| LoU || Cost || Load). And Table (4-6) below represents the AWT and ART for SJF algorithm.

Table (4-5): BLJF works as SJF (change behavior)

BLJF –SJF		
LoU	**AWT**	**ART**
1	**2.94**	**0.68**
2	**3.56**	**0.84**
3	**0.52**	**0.24**
4	**0.35**	**0.35**
5	**0.66**	**0.31**
Total Average	**1.61**	**0.49**

By comparing the results between Table (4-5) and Table (4-6) the total average (AWT and ART) is close. That means the proposed algorithm success to change its behaviors as SJF.

Table (4-6): SJF

SJF		
LoU	**AWT**	**ART**
1	**2.10**	**0.68**
2	**2.68**	**0.79**
3	**0.52**	**0.24**
4	**0.35**	**0.35**
5	**0.66**	**0.31**
Total Average	**1.26**	**0.48**

Figures (4-5 and 4-6) below depicted the results for the above Tables (4-5 and 4-6). The proposed algorithm produced average waiting time and average response time closely to short job first. This means that BLJF algorithm has an ability to adapt to work as SJF.

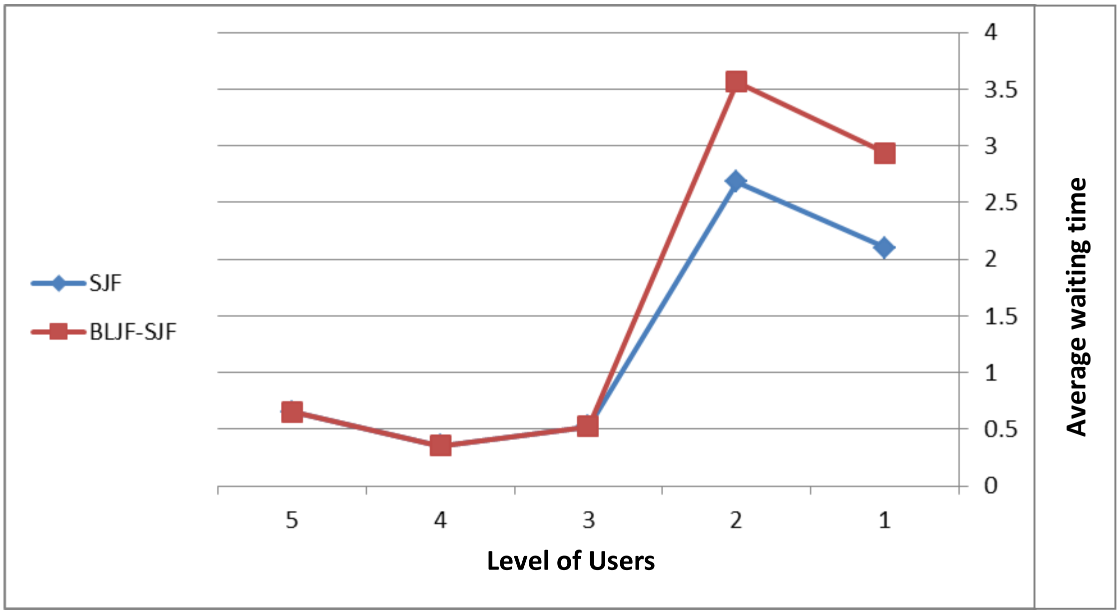


Figure (4-5): AWT of BLJF and SJF

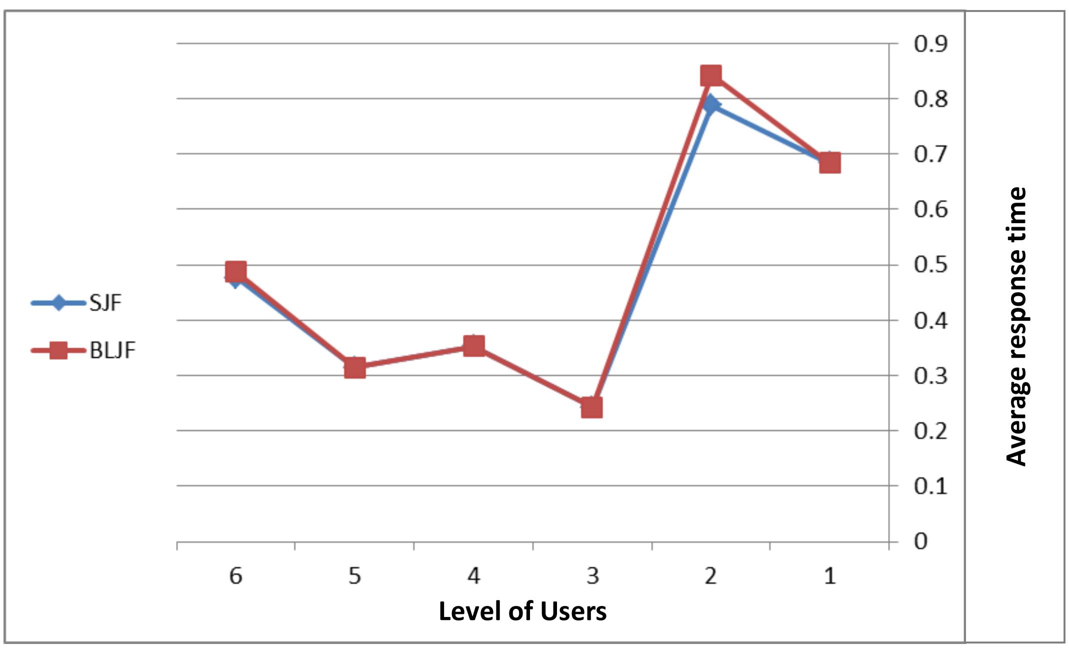


Figure (4-6): ART of BLJF and SJF

4.3.2 The BLJF works as Round Robin algorithm

Using the same dataset, and order the submitted tasks in the queue only based on the arrival time of tasks (without taking into consideration the parameters vector), the proposed algorithm BLJF has been worked as RR algorithm. Table (4-7) represents the algorithm BLJF results AWT and ART.

Table (4-7): BL JF works as RR (change behavior)

BLJF		
LoU	**AWT**	**ART**
1	4.41	1.52
2	2.41	0.85
3	1.72	1.42
4	0.00	0.00
5	0.74	0.37
Total Average	**1.86**	**0.83**

Table (4-8) represents the AWT and ART for RR algorithm. By comparing the results between Table (4-7) and Table (4-8) the total average (AWT and ART) is close. That means the proposed algorithm success to change its behaviors as RR algorithm.

Table (4-8): RR

RR		
LoU	**AWT**	**ART**
1	**3.04**	**1.36**
2	**1.89**	**0.79**
3	**1.55**	**1.25**
4	**0.00**	**0.00**
5	**0.74**	**0.37**
Total Average	**1.44**	**0.75**

Figures (4-7 and 4-8) has been explained the results for the above Tables (4-7 and 4-8). The proposed algorithm produced AWT and ART closely to RR. This means that BLJF algorithm has an ability to adapt for working as RR. This feature (changing behavior as RR and SJF) given by proposed algorithm (BLJF) make it more flexibility and adaptation.

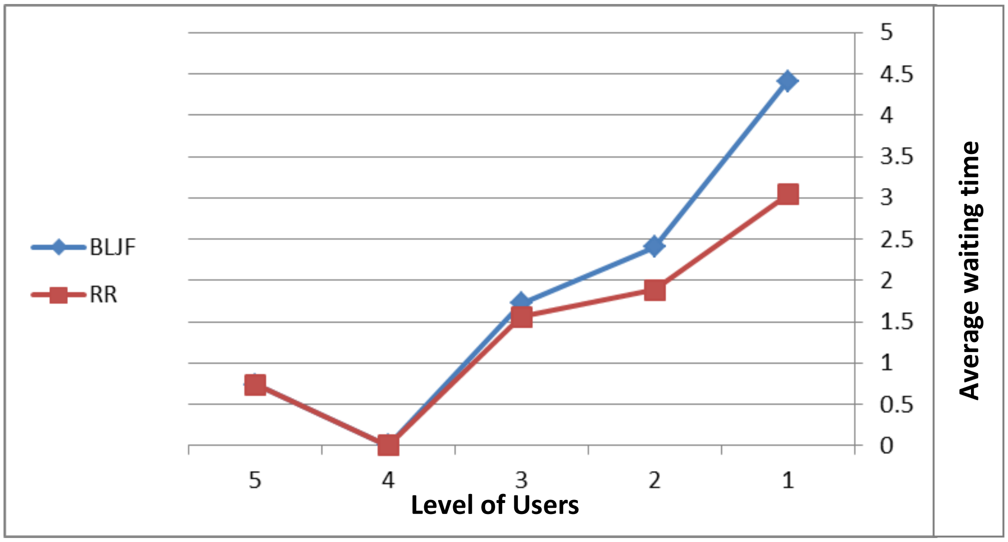


Figure (4-7): AWT of BLJF and RR

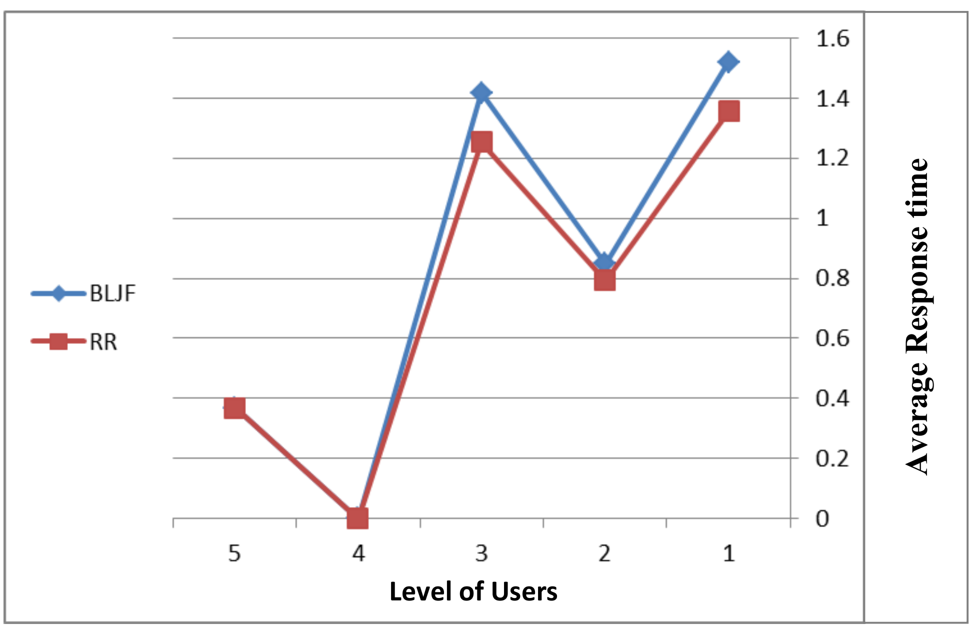


Figure (4-8): ART of BLJF and RR

4.4 Analysis and Discussion

BLJF algorithm has been focused on the level of user by adding a new parameter called level-of-user. The high level of user task must have a special treatment over other tasks, when it's arrived, no way to still it waiting for a long time and must be executed as fast as possible.

The proposed algorithm BLJF treats the higher level of users which have important tasks that must be responded in fast time compare with other lower level users that having daily traditional tasks. BLJF gave the highest level users as small as possible waiting and response time.

The researcher used two measurements to evaluate the qualities of service that received by the users: the average response time and waiting time. When comparing the proposed algorithm with the SJF algorithm, the results produced by the proposed algorithm is the lower waiting time for highest level users. And when comparing the proposed algorithm with the RR algorithm, the recorded results produced by the proposed algorithm is the fast response time for highest level users.

Depending on the parameters vector of the task, submitted tasks in the queue can be ordered in different ways. And by re-arrange the parameters values in the vector, the algorithm can adapt and control over the way of how the tasks will be executed.

When gave Time the first order in the parameters vector, the proposed algorithm produced average waiting time and response time closely to SJF algorithm. Otherwise, when ignoring the parameters values in the vector and based only on the arrival time of the submitted tasks, the proposed algorithm can work closely to RR algorithm.

4.4.1 Relation between level of user and the proposed algorithm (BLJF)

The highest level users must receive special cloud services because they have important tasks that must be executed very fast. The proposed scheduling algorithm BLJF has been found the solution by adding a new parameter (LoU), it focused on the level of the user for each submitted task when it is ordering tasks in the queue for execution.

By calculating the weight for each submitted task to the private cloud based on the parameters vector, it consists of four values (LoU, Time, cost, and Load) and by giving the LoU the first priority in calculating the task's weight. This helped to produce tasks scheduling algorithm BLJF that succeeded to give high priority to the high-level tasks and less priority to the low-level tasks.

4.4.2 Flexibility of implementing the proposed algorithm (BLJF)

When a large number of tasks from the different level of users are submitted to be executed in the private cloud, the determination of the task's priority value becomes the necessity to give each task the service that must enjoy.

The proposed BLJF algorithm has an ability to change easily the way of ordering tasks in the queue for execution. This is done by:

- Changing the order of parameters in the parameters vector of the tasks, will give different weight value for tasks and this will lead to change the order of the tasks in the queue.
- Neglect one or more of the four parameters in the parameters vector of the tasks will give the proposed algorithm an adaptation capability to work like other scheduling algorithms such as (SJF and RR).

Chapter Five

Conclusion and Future Work

5.1 Conclusion

The proposed scheduling algorithm has been implemented to achieve user's requirements, in addition to, realized high resources utilization (time, cost, and load). Scheduling of tasks cannot be done based on individual criteria, but it depending on several rules and regulations that can be described as agreement between users and providers of the cloud. In the private cloud computing, there is a large gap between the most tasks scheduling algorithms that are designed and the actual level of users.

There are many algorithms to use the priority scheduler. The proposed algorithm (BLJF) used priority to give each user in the private cloud the desired service and maintain the system performance in an acceptable ratio.

Depending on four parameters: Time refers to completion time, Cost refers to the price of used resources (power consumption), Load refers to the system load, and Level refers to the level of the user. BLJF mainly concerned in the levels of users. BLJF achieved many goals:

- ✓ Arrange users' tasks based on the Level-Of-User parameter, in the parameters vector, to give each user's task the desired service from the cloud server.
- ✓ Calculate weight for each submitted task depending on the vector, it consists of four parameters that take into consideration the parameters used in the cloud service centers.
- ✓ Having an ability to change the policy for scheduling tasks in the queue, it has been achieved that through canceling some parameters or re-ordering the parameters in the parameters vector that lead to change the calculated weight for each task. This change of the policy supports the proposed algorithm BLJF capability to work like other scheduling algorithms SJF and RR.

- ✓ The performance comparison between the proposed algorithm BLJF and the two scheduling algorithms SJF and RR showed that BLJF achieves fast response time and less waiting time for the high-level users when it compared with the response time and waiting time for low-level users. This represents the big challenge in scheduling tasks in any private cloud.

5.2 Future Work

During this research, some points have been registered to be suggestions for future work:

- ✓ In cloud computing environment, the energy efficiency scheduling is a more concern. The researchers can use another strategy in the private cloud by considering other performance factors, such as the operation cost, the energy efficiency, and the execution time.
- ✓ The proposed scheduling algorithm can be tested on depending tasks (interconnected tasks) which will certainly affect the performance of the algorithm and may produce different results.
- ✓ The proposed algorithms tested on one cloud server and certainly will produce different results if it is testing on two or more cloud servers having different characteristics.
- ✓ The proposed algorithm arranges tasks, of different users levels, in a single queue. It can be tested after distributing users on two or more queues, each queue designated for a specific level of users.

Reference

- Al-Mughrabi, A.A ,(2013). A Framework Model for Database Replication within the Availability Zones International Journal of Computer Science Issues,10(2), 76-90
- Akijian, T.,&Kayed, A. (2015). Resource Allocation Technique to Obtain Energy Efficient Cloud, ACM Digital Library,978(1).
- Al-Bahadili, H., Al-Sabbah,H. ,&Abu Arqoub, M.(2013). Modeling and Analysis of Cloud Collaborative Commerce, International Journal on Cloud Computing: Services and Architecture ,3(1)
- Annette, J, R., Banu,W, A., &Shriram, S. (2013). A Taxonomy And Survey Of Scheduling Algorithms In Cloud: Based On Task Dependency. International Journal of Computer Applications, 82(15), 20-26.
- Awan, M., & Shah, M. A.(2015). A Survey on Task Scheduling Algorithms in Cloud Computing Environment.International Journal of Computer and Information Technology, 441-448
- Barclay, C. (2016). How to create a custom scheduler for Amazon ECS,Amazon Web Services,(On-Line): available:https://aws.amazon.com/blogs/compute/how-to-create-a-custom-scheduler-for-amazon-ecs/
- Beal,V.(2016).PrivateCloud,Webopedia,(On-Line),available: http://www.webopedia.com/TERM/P/private_cloud.html
- Bhoi, U., &Ramanuj, P. N. (2013). Enhanced Max-Min Task Scheduling Algorithm In Cloud Computing. International Journal of Application or Innovation in Engineering and Management (IJAIEM), 259-264.

- Bitam, S. (2012, February). Bees life algorithm for job scheduling in cloud computing. In Proceedings of The Third International Conference on Communications and Information Technology (pp. 186-191).
- Buyya, R., Pandey, S., &Vecchiola, C. (2009).Cloudbus Toolkit For Market-Oriented Cloud Computing. In Cloud Computing (pp. 24-44). Springer Berlin Heidelberg.
- Buyya, R., Ranjan, R., &Calheiros, R. N. (2009). Modeling and Simulation of Scalable Cloud Computing Environments and The Cloudsim Toolkit: Challenges And Opportunities. In High Performance Computing & Simulation, 2009.HPCS'09. International Conference on (pp. 1-11). IEEE.
- Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., &Buyya, R. (2011).Cloudsim: A Toolkit For Modeling And Simulation Of Cloud Computing Environments And Evaluation Of Resource Provisioning Algorithms. Software: Practice and Experience, 41(1), 23-50.
- Choudhary, M., &Peddoju, S. K. (2012). A Dynamic Optimization Algorithm For Task Scheduling In Cloud Environment. International Journal of Engineering Research and Applications (IJERA), 2(3), 2564-2568.
- Dhingra, A.,&Paul , S. (2014).Green Cloud: Smart Resource Allocation and Optimization using Simulated Annealing Technique,Indian Journal of Computer Science and Engineering (IJCSE), 41-49
- Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2013). The Analytic Hierarchy Process: Task Scheduling and Resource Allocation in Cloud Computing Environment. The Journal of Supercomputing, 64(3), 835-848.

- Fang, Y., Wang, F., &Ge, J. (2010). A Task Scheduling Algorithm Based On Load Balancing In Cloud Computing. In Web Information Systems and Mining(pp. 271-277). Springer Berlin Heidelberg.
- Ghanbari, S., & Othman, M. (2012). A Priority Based Job Scheduling Algorithm In Cloud Computing. Procedia Engineering, 50, 778-785.
- Goyal, S. (2014). Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review,Computer Network and Information Security, 3(20)
- Han, H., Deyui, Q., Zheng, W., & Bin, F. (2013). A Qos Guided task Scheduling Model in cloud computing environment. In Emerging Intelligent Data and Web Technologies (EIDWT), 2013Fourth International Conferenceon(pp. 72-76). IEEE.
- Junwei,G. &Yongsheng,Y.,(2013). Research of cloud computing task scheduling algorithm based on improved genetic algorithm, ICCSEE
- Kærvang, D., Billestrup, J., & Jensen, L. S.(2013). The Practice of the Netbeans Community. (On-line), available :http://vbn.aau.dk/ws/files/61074114/1231156153.pdf
- Katyal, M., & Mishra, A. (2013).A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. arXiv preprint arXiv:1403.6918.
- Kaur, R., &Kinger, S. (2014). Analysis of Job Scheduling Algorithms in Cloud Computing. International Journal of Computer Trends and Technology (IJCTT),9(7), 379-386.

- Krisragh, D. (2015). What is Azure Scheduler, Microsoft Azure,(On-line),available: https://azure.microsoft.com/en-us/documentation/articles/scheduler-intro/
- Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011).Cloud Task Scheduling Based On Load Balancing Ant Colony Optimization.In Chinagrid Conference (ChinaGrid), 2011 Sixth Annual (pp. 3-9). IEEE.
- Liu, G., & Yang, C. (2013). Scheduling Research Based On Genetic Algorithm And Qos Constraints Of Cloud Computing Resources. Journal of Theoretical & Applied Information Technology, 51(1).
- Liu, J., Luo, X. G., Zhang, X. M., Zhang, F., & Li, B. N. (2013). Job Scheduling Model for Cloud Computing Based On Multi-Objective Genetic Algorithm. IJCSI International Journal of Computer Science Issues, 10(1), 134-139.
- Liu, X., Wang, c., Zhou, B., Chen, J., Yang, T.,&Zomaya, A. (2013). Priority-Based Consolidation ofParallel Workloads in the Cloud, IEEE Transactions On Parallel And Distributed Systems,24(9), 1874-1882
- Lovesum, S. J., Krishnamoorthy, K., & Prince, P. (2014). An Optimized Qos Based Cost Effective Resource Scheduling In Cloud. Journal of Theoretical & Applied Information Technology, 66(1).
- Mandel, R.(2016). What is a Private Cloud, Interoute Communications Limited,(on-line), available: http://www.interoute.com/cloud-article/what-private-cloud
- Mathukiya, E., &Gohel, p. (2015).Efficient Qos Based Tasks Scheduling using Multi-Objective optimization for Cloud Computing. International Journal of Innovative Research in Computer and Communication Engineering,3(8), 7169-7173

- Mohialdeen, I. A. (2013).Comparative Study of Scheduling Algorithms in Cloud Computing Environment. Journal of Computer Science, 9(2), 252.
- Mustafa, L. M., Elmahy, M. K., &Haggag, M. H. (2014). Improve Scheduling Task based Task Grouping in Cloud Computing System. International Journal of Computer Applications, 93(8) ,0975 – 8887
- Navimipour, N. J., &Milani, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. International Journal of Modeling and Optimization, 5(1), 44.
- Ning, W., Yang, Y., Kun, M., Yu, C., &Hao, D. (2013). A Task Scheduling Algorithm Based On Qos And Complexity-Aware Optimization In Cloud Computing. In Information and Communications Technology 2013, National Doctoral Academic Forum on (pp. 1-8). IET.
- Parikh, S., &Sinha, R. (2013). Double Level Priority based Optimization Algorithm for Task Scheduling in Cloud Computing. International Journal of Computer Applications, 62(20), 34-37
- Patel, R., &Mer, H. (2013). A Survey of Various Qos-Based Task Scheduling Algorithm in Cloud Computing Environment.International Journal Of Scientific & Technology Research2(11),196-112.
- Patel, S., &Bhoi, U. (2013). Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review. International Journal of Scientific &Technology Research, 2, 147-152.
- Salot, P., & Gandhi, P. (2013). Job Resource Ratio Based Priority Driven Scheduling in Cloud Computing ,International Journal for Scientific Research & Development,1(2), 373-377

- Sanghani, N., Khimani,R., Sutaria,K., Vasani,P. (2013). Pre-Emptable Shortest Job Next Scheduling In Private Cloud Computing, Journal Of Information, Knowledge And Research In Computer Engineering,2(2),385-388
- Sasikala, P. (2011). Cloud computing: present status and future implications.International Journal of Cloud Computing,1(1), 23–36.
- Saxena,D., & Chauhan, S. (2014). A Review On Dynamic Fair Priority Task Scheduling Algorithm In Cloud Computing,International Journal of Science, Environment and Technology, 3(3), 997-1003.
- Schreckmann,D.(2014).An Introduction to Java Development with NetBeans IDE,(On-line),available:http://www.javaranch.com/journal/2003/10/NetBeans_IDE_Project_Basics_Printable.pdf
- Selvarani, S., &Sadhasivam, G. S. (2010). Improved cost-based algorithm for task scheduling in cloud computing. In Computational intelligence and computing research (iccic), 2010 IEEE international conference on (pp. 1-5). IEEE.
- Shimpy, E., &Sidhu, M. J. (2014). Different Scheduling Algorithms In Different Cloud Environment. Algorithms, International Journal of Advanced Research in Computer and Communication Engineering, 3(9),8003-8006.
- Shinde,M. S., & Kadam, A. (2014). Survey of Recent Task Scheduling Strategies in Cloud Computing, International Journal of Advanced Research in Computer Science and Software Engineering,4(11),830-834
- Singh, L., Ahmed, J.(2014). A Greedy Algorithm For Task Scheduling & Resource Allocation Problems In Cloud Computing,International Journal Of Research & Development In Technology And Management Science,21(1)

- Singh, R. ,&Patra, P. ,(2013), Prioritized Workflow Scheduling in Cloud
- Singh, R. M., Paul,S., & Kumar, A. (2014). Task Scheduling in Cloud Computing: Review ,International Journal of Computer Science and Information Technologies, 5(6), 7940-7944
- Singh,A., Sahu, S., Gautam, K.,&Tiwari, M. (2014). Private Cloud Scheduling with SJF, Bound Waiting, Priority and Load Balancing, International Journal of Advanced Research in Computer Science and Software Engineering,4(1), 367-372
- Suakanto, S., Supangkat, S., Suhardi, S., &Saragih, R. (2012). Performance Measurement Of Cloud Computing Services, International Journal on Cloud Computing: Services and Architecture(IJCCSA),Vol2(2).
- Subramanian,k. (2011).Private Clouds,Trend Micro Inc ,(On line), available :http://www.trendmicro.com/cloud-content/us/pdfs/business/white-papers/wp_private-clouds-analyst-ksubramanian.pdf
- Xu, B., Zhao, C., Hu, E., & Hu, B. (2011). Job Scheduling Algorithm Based On Berger Model In Cloud Environment. Advances in Engineering Software, 42(7), 419-425.
- Zhao, C., Zhang, S., Liu, Q., Xie, J., & Hu, J. (2009). Independent Tasks Scheduling Based On Genetic Algorithm In Cloud Computing. In Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on (pp. 1-4). IEEE.

Appendix - A

TaskName	Time	Cost	Load	User-Level
0	0.454423465	8.678378749	21.02864583	8
1	0.169558401	3.249146307	25.61848958	10
2	0.103084623	1.951118101	21.38671875	7
3	0.056905807	1.067389549	32.35677083	7
4	0.273635792	5.106006462	6.0546875	7
5	0.366821898	7.03810858	31.34765625	10
6	0.568084706	10.67866103	24.86979167	3
7	0.056742229	1.067080841	23.76302083	10
8	0.431553646	8.07721765	23.50260417	2
9	0.307191291	5.734637421	3.678385417	6
10	0.194929723	3.63578199	11.29557292	4
11	0.257826542	4.844444904	14.12760417	8
12	0.340362254	6.530434837	19.56380208	6
13	0.611104895	11.73995761	10.3515625	3
14	0.349470311	6.651368785	24.57682292	4
15	0.604059702	11.18354008	18.61979167	7
16	0.525373881	9.926898445	18.42447917	4
17	0.254926866	4.758966768	12.01171875	4
18	0.335911835	6.333414274	27.37630208	6
19	0.572608018	10.63373351	7.51953125	1
20	0.106660009	2.022248781	17.02473958	10
21	0.198209292	3.700164876	17.83854167	7
22	0.548612694	10.44603572	20.99609375	5
23	0.229143833	4.244218182	20.27994792	10
24	0.3866407	7.336884868	25.09765625	3
25	0.382536362	7.196988314	15.55989583	4
26	0.011089565	0.210704976	25.78125	2
27	0.435918168	8.282572901	29.78515625	9
28	0.481130779	9.250819896	5.95703125	6
29	0.414450546	7.935149478	18.32682292	9
30	0.489729717	9.090655705	2.799479167	7
31	0.025165382	0.465911006	29.32942708	6
32	0.30634324	5.765984079	27.44140625	3
33	0.49496075	9.396375472	16.50390625	5
34	0.619710394	11.5034347	26.59505208	6
35	0.250339976	4.744358149	32.421875	3
36	0.609349544	11.64696676	21.12630208	1
37	0.40813263	7.817493552	13.60677083	6
38	0.355625094	6.635915636	8.984375	7
39	0.577685281	11.14007394	2.24609375	1
40	0.354497079	6.73554835	11.78385417	6

41	0.006260832	0.118906895	32.2265625	7
42	0.434317086	8.128939845	31.640625	7
43	0.505634245	9.545722755	19.69401042	1
44	0.528283515	9.883390072	20.21484375	2
45	0.25412328	4.873732109	8.7890625	9
46	0.287419768	5.472708149	2.018229167	9
47	0.224542288	4.195388452	22.72135417	5
48	0.300188302	5.630670684	13.31380208	9
49	0.144817537	2.679972978	16.50390625	4
50	0.341104173	6.306896232	15.78776042	5
51	0.429252145	8.152437226	12.40234375	5
52	0.226961099	4.36015765	15.55989583	8
53	0.424679154	8.027721656	18.84765625	10
54	0.265533441	4.989254065	9.765625	3
55	0.346990725	6.677301109	25.42317708	5
56	0.312867361	5.959787135	7.51953125	2
57	0.064782068	1.216175785	16.53645833	4
58	0.361524389	6.687389183	24.05598958	5
59	0.543098042	10.42467145	19.88932292	3
60	0.113390812	2.166406755	18.39192708	6
61	0.166521048	3.096462146	18.03385417	10
62	0.146916001	2.786683772	14.90885417	9
63	0.192088152	3.629490625	16.04817708	5
64	0.56989307	10.54174177	5.208333333	9
65	0.613776114	11.60222667	12.59765625	3
66	0.080145102	1.531225389	16.37369792	1
67	0.376542134	7.249024417	19.95442708	1
68	0.38773024	7.285277028	24.15364583	2
69	0.409159074	7.847103651	27.734375	6
70	0.483528924	9.124483564	15.49479167	7
71	0.601333075	11.38163782	6.73828125	1
72	0.395326525	7.597836081	25.48828125	6
73	0.131842024	2.482595617	26.23697917	9
74	0.478754765	9.046033887	13.4765625	6
75	0.085683136	1.642589224	12.33723958	5
76	0.571704972	10.73744567	5.76171875	2
77	0.579467887	11.09460308	17.48046875	6
78	0.37207673	7.09067788	28.28776042	7
79	0.395552607	7.608593458	24.51171875	9
80	0.054975066	1.052117545	11.71875	5
81	0.406643014	7.802145146	8.919270833	10
82	0.301719375	5.715637489	22.94921875	7
83	0.353860997	6.714857988	14.90885417	8

84	0.444277553	8.448605825	12.72786458	6
85	0.572511821	10.93357805	8.7890625	7
86	0.459988494	8.654162436	2.799479167	1
87	0.416040202	7.705925877	30.46875	9
88	0.207686197	3.962932529	16.08072917	1
89	0.284913383	5.427293889	31.60807292	1
90	0.077588903	1.491192031	6.184895833	3
91	0.573419607	10.68598797	3.125	9
92	0.508566471	9.749477505	17.87109375	8
93	0.378620163	7.227651853	24.90234375	5
94	0.482649224	9.049201604	3.743489583	5
95	0.221943347	4.193602195	9.47265625	8
96	0.262891556	4.916174795	1.5625	7
97	0.065158677	1.215323855	26.66015625	2
98	0.046953739	0.903552255	16.17838542	2
99	0.308719469	5.89328584	20.34505208	9
100	0.115015014	2.200234971	0.162760417	3
101	0.491712278	9.1633272	15.85286458	5
102	0.081881299	1.53187597	11.03515625	4
103	0.568901243	10.91995935	3.352864583	7
104	0.01940084	0.370508674	10.38411458	6
105	0.42160589	8.04822588	26.46484375	6
106	0.121551155	2.299655267	17.64322917	5
107	0.290400397	5.381170407	3.22265625	7
108	0.115810284	2.186348881	3.22265625	6
109	0.598872842	11.1360522	19.66145833	5
110	0.405675971	7.757285205	20.99609375	7
111	0.418932495	8.02435738	21.71223958	8
112	0.210445806	4.03435296	14.58333333	2
113	0.234403675	4.46323826	19.36848958	2
114	0.624621726	11.73129878	8.7890625	6
115	0.440894218	8.237746795	5.859375	8
116	0.050552112	0.94124676	6.380208333	5
117	0.301177034	5.759069598	6.0546875	7
118	0.239174771	4.433889169	25.32552083	7
119	0.356572827	6.708513808	26.00911458	2
120	0.079974116	1.492306063	3.190104167	1
121	0.201072527	3.79599011	8.951822917	2
122	0.062515799	1.179204255	20.44270833	8
123	0.556187474	10.60379023	9.08203125	5
124	0.488172019	9.291238841	32.03125	10
125	0.61576571	11.49509503	21.32161458	3

Appendix - B

Task Number	Time	Cost	Load	User-Level	ArrivalTime
0	0.16413	3.063967	28.80859	4	0.0015
1	0.081469	1.569065	9.244792	5	0.0115
2	0.361301	6.812103	26.00911	2	0.0215
3	0.242088	4.482006	32.38932	5	0.0315
4	0.256014	4.930742	14.29036	3	0.0415
5	0.556857	10.32769	22.33073	2	0.0515
6	0.030081	0.55691	23.86068	2	0.0615
7	0.017094	0.325489	0.585938	3	0.0715
8	0.158134	2.975104	11.32813	1	0.0815
9	0.518965	9.704852	26.6276	1	0.0915
10	0.06568	1.221847	22.6888	3	0.1015

Appendix - C

- **Function to calculate Time**

```
static double Time(double DATA, double RLOR , double OSRU , double RAMS,
double GBIT, double CRUO, double FSBS){
doubleti = 0.0 ;
if((DATA + RLOR + OSRU) > (RAMS * 1024) ){
ti = Math.max(Math.max(((GBIT * 1000 / 2)/2400),(GBIT * 1000 / CRUO) ),
      GBIT * 1000 / FSBS);
  }
else{
if(CRUO > FSBS){
ti = Math.max((GBIT * 1000/ CRUO ),(GBIT * 1000/ FSBS) );
        }
else{
ti = Math.min((GBIT * 1000/ CRUO ),(GBIT * 1000/ FSBS));
    }
  }
returnti ;   }
```

- **Function to calculate Cost (Power consumption)**

```
static double Tpower(double DATA, double RLOR , double OSRU , double RAMS,
double GBIT, double CRUO, double FSBS, double CTDS, double CPUC,
double CPUS, double OCCUR){
  double Po=Power(DATA,RLOR,OSRU,RAMS,GBIT,CRUO,FSBS,CTDS,CPUC,CPUS,
OCCUR);
double Ti = Time(DATA,RLOR ,OSRU,RAMS,GBIT,CRUO,FSBS) ;
doubleTp = Po * Ti;
returnTp ;
  }
```

```
static double Power(double DATA, double RLOR , double OSRU , double RAMS,
double GBIT, double CRUO, double FSBS, double CTDS, double CPUC,
double CPUS, double OCCUR){
   double L=Usage(DATA,RLOR,OSRU,RAMS,GBIT,CRUO,FSBS,CTDS,CPUC,CPUS,OCCUR);
double Po = 0.0 ;
if((RAMS - ((DATA + RLOR + OSRU)/1024))> 0){
       Po = (L * 3 * 12) + ((RAMS - ((DATA + RLOR + OSRU)/1024)) * 1.7 )
              + (((DATA + RLOR + OSRU)/1024) * 10 ) ;
}else{
       Po = (L * 3 * 12) + ( RAMS * 10 ) ;
   }
return Po ;
   }
static double Current(double DATA, double RLOR , double OSRU , double RAMS,
double GBIT, double CRUO, double FSBS, double CTDS, double CPUC,
double CPUS, double OCCUR){
   double L =
Usage(DATA,RLOR,OSRU,RAMS,GBIT,CRUO,FSBS,CTDS,CPUC,CPUS,OCCUR) ;
double Cu = 0.0 ;
if(L * 3 > 3){
        Cu = 3;
}else{
       Cu = L * 3 ;
   }
return Cu;
   }
static double Usage(double DATA, double RLOR , double OSRU , double RAMS,
double GBIT, double CRUO, double FSBS, double CTDS, double CPUC,
```

```
double CPUS, double OCCUR){
double K = Time(DATA,RLOR ,OSRU,RAMS,GBIT,CRUO,FSBS) ;
double us = 0.0;
if((RAMS/(CTDS * CPUC))< 3){
us = (((GBIT / (CPUS * CPUC * (RAMS / CTDS)))/K)+
        (OCCUR/(CPUS * CPUC * 1000)))*(10/6.0);
   }
else
   {
us = (((GBIT / (CPUS * CPUC * 3 ))/K)+ (OCCUR/(CPUS * CPUC * 1000)))*(10/6);
   }
return us ;
   }
```

- **Function to calculate Load (TaskInfo)**

```
static double LoadTask(double W1,double W2,double W3,double W4,double MNUL,
doubleRLOR,doubleOSRU,double RAMS1,double FSBS,double CTDS1,
double CPUC1,double CPUS1,double OCCUR,double RAMS2,double CTDS2,
double CPUC2,double CPUS2, double MNTK)[][]{
int user ;
doublearrivalT;
int CRUO ;
double GBIT ;
int DATA ;
// defin to start task load
int i ;
int j = (int) Math.round(((Math.random()* MNTK/3)* 3) + 20);
```

```
doubleTaskInfo[][]= new double [j][15] ;
//
for(i = 0 ; i < j ; i++ ){
arrivalT = System.currentTimeMillis();
user = (int) Math.round( (MNUL-1) * Math.random() + 1);
CRUO = (int) Math.round(Math.random() * 1000);
DATA = (int) Math.round(Math.random() * 100);
GBIT = Math.random();
// chose best server for do the task with low power .
// and return back" best server , time , cost , load
doubleGStask[] = Gserver( DATA, RLOR, OSRU, RAMS1,GBIT, CRUO, FSBS, CTDS1,
    CPUC1,CPUS1, OCCUR,RAMS2, CTDS2, CPUC2,CPUS2);
// calculate task weight and inputs is w 1...4, time , cost , load , user
// TaskWeight[1...4][0 Ws,1 Order,2 cal]
double TW[][]= TaskWeight(W1,W2,W3,W4,GStask[1],GStask[2],GStask[3],user);
//
TaskInfo[i][0] = i ; // name
TaskInfo[i][1] = CRUO;// Task MHz
TaskInfo[i][2] = DATA;// Task MB size
TaskInfo[i][3] = GBIT;// Task GB excution
TaskInfo[i][4] = GStask[0];//best server
TaskInfo[i][5] = GStask[1];//time
TaskInfo[i][6] = GStask[2];//cost
TaskInfo[i][7] = GStask[3];//load
TaskInfo[i][8] = user; // user level
TaskInfo[i][9] = Math.round(TW[0][2]);// weight 1
TaskInfo[i][10] = Math.round(TW[1][2]);// weight 2
```

```
TaskInfo[i][11] = Math.round(TW[2][2]);// weight 3
TaskInfo[i][12] = Math.round(TW[3][2]);// weight 4
TaskInfo[i][13] = arrivalT;// arrival time
}
returnTaskInfo;
}
```