Comparison and Evaluation of Cloud Processing Models in Cloud-Based Networks

Ghaida Muttashar Abdulsahib
Department of Computer Engineering
University of Technology
Baghdad, Iraq.
E-mail: gh961@yahoo.com

Osamah Ibrahim Khalaf
College of Information Engineering
AI-Nahrain University
Baghdad, Iraq.
E-mail: usama81818@yahoo.com

Abstract - A cloud network refers to a computer network that is inside or part of a cloud computing infrastructure. In this paper we introduce, evaluate and compare models and algorithms that are used to schedule processing in cloud-based networks. The results show that the Buffer-Pool-based scheduling model applies integrated management on all computational tasks as well as the bottom layer of the resource associated with them. In practice, this model solves the management weakness resulting from online parallel processing by using the SaaS (Software-as-a-Service) structure which is defined as a software authorizing model to give permission to use the software to authorized users. This software is hosted on external servers as opposed to on servers situated in-house, which improves the SaaS structure quality service and eliminates the extra energy associated with duplicate computing.

Keywords: cloud-based networks, cloud processing, cloud network

I. INTRODUCTION

The cloud network refers to a computer network that is inside or part of a cloud computing infrastructure. In fact, it is a computer network that establishes a network connection between cloud-based services, cloud services and solutions. The cloud network can be a cloud-based network or cloudy network. Its related components and external users, software, and services to communicate with each other are all part of the cloud network. Typically, the cloud network resembles a standard computer network, but its components, the devices and operations that are performed, are on cloud computing [1]. The continuous growth of graphs on the one hand and the use of companies from localized data centers on the other hand, companies have been thinking about using a cloud-based network to solve the problem of reducing performance. Cloud processing enables them to use unlimited computing resources to increase their efficiency [2]. This growing need has increased the importance of cloud networks and cloud computing. But the most important principle in a cloud-based network is to process information and communicate in the fastest time possible, also called scheduling models. This paper is dedicated to the introduction, evaluation and comparison of models and algorithms that are used to schedule cloud processing in cloud-based networks.

II. CLOUD PROCESSING FEATURES

Cloud processing uses almost all of the attributes that other computational contain, but the correct utilization of these attributes has created it more important to prioritize the use of this network over others. Cloud processing has six important features as follows [3]:

A. Wide access to the system: Access to cloud assets is accessible over the system and utilization institutionalized strategies to get to clients to the system
B. Providing demand-based services: Users can get to their assets and programming without expecting to interface with cloud handling specialist co-ops..
C. Calculation service (pay as much as use): One of the most important features of cloud processing is the calculation framework dependent on the utilization of administrations and assets.
D. Resource variety: In cloud processing there is a huge amount of resources that have become independent from virtualization using their physical location.
E. Multi-user (tenant) (resource sharing): Centralizing, increasing the use of useless resources, and allowing users to share resources.
F. Fast development: The assets in the cloud ought to be extended quickly and be open to boundless clients whenever.

III. SCHEDULING

Scheduling in the cloud takes place in three steps, first we need to identify existing resources and gather information about them. The following stage is the determination organize. At this stage, the assets are chosen dependent on a progression of primary parameters of assets and errands, lastly the chose work is assigned to the chosen source. Figure 1 demonstrates the planning stages in the cloud condition [4].
A. Scheduling Levels

From a general point of view, the scheduling are divided into two levels [5]:

A1. User Level Scheduling: At this level, the problem of providing service and customer service is dealt with. This mainly refer to the economic issues such as the demand and balance of supply, rivalry among clients, and the relative decrease of client costs.

A2. Framework Level Scheduling: This level tends to the issues of assets administration inside the server farms. From the client's perspective, server farm is an incorporated framework that gives administrations to clients however in certainty focuses of information are a mix of the expansive number of physical machine that serve to coordinate. Subsequent to getting a great deal of work from various clients and relegating them to physical machines, the execution of focus information is influenced. With the end goal to build the efficiency of the framework and control, a few things, for example, the concurrent execution of procedures, asset sharing, blunders, and so on, ought to be considered. Framework level booking is done in a static and dynamic way. Static strategies are utilized for the time when the attributes of the arrangement of assignments to be planned, for example, errand handling time, correspondences, information reliance, and synchronization needs preceding execution, are utilized. Dynamic strategies are utilized when there are not very many suspicions previously execution, and errands are entered on-line.

IV. SCHEDULING MODEL BASED ON BUFFER-POOL FACTOR

A. Model Framework

This model performs the process of analyzing and making process for online information step by step. The end of each stage is marked with a mark called milestone. Each step produces a sequence of results, which ultimately results in the optimal outcome as the data goes to the next stage. After processing all the steps, the last result is obtained. If the client is not accepted the final result, he will perform (undo) operation one-by-one and selects and executes other methods for processing. Therefore, knowing the pool object and the technology of data caching is very important to create collaborative methods between data extracted and the data in the database, which improves data processing efficiency [6].
the information in the buffer pool and evaluates the scheduling and then sends the information to the resources storage layer through the computing resources broker (CCRB).

This layer has five sections:

1. Manager Agent: This agent uses a specific task which enables it to arrange online information. (Figure 3).

2. Sub Tasks Distribute Agent: takes task scheduler data from the manager and selects the appropriate Internet hop for calculating, and then transfers the result to the Buffer-pool agent. (Figure 4).

3. Buffer-Pool Agent: It is the core of the structure. It processes all computational tasks, processes and pre-selected methods in each step. It also provides a repository for storing mid-term results and scheduling resources that process parallel data.

4. Mission Assistance Agent: This section compares the results of whole algorithms that are data in the Buffer-Pool, with quality control criteria and arranges them in sequence. (Figure 5).

5. Cloud Computing Resources Broker (CCRB): It implements Buffer-pool and manages the resource scheduling data. It also makes optimal use of the source and when the agent is idle, the source selects the next task and thereby reduces the processing of scheduling.

A3. Source Storage Layer: This layer is designed to store data and ensures that access to the data source is safe and high-speed. This layer provides data for the kernel layer (layer 2) and storage service outside the cloud. This layer shares a lot of administrative solutions for data storage; it uses parallel database and main memory database to store data permanently in physical memory and put backup copy of the data on the disk which uses high and increasing efficiency [6].

V. AVAILABLE SCHEDULING ALGORITHMS

Today, increasing the efficiency of the cloud is a vital issue. Legitimate and ideal scheduling results in increased cloud performance and performance. The dynamic and varied nature of cloud processing and different user requests has caused the complexity of the scheduling problem in the cloud environment. Optimum scheduling is one of the most necessary difficulties for computer networks, especially cloud processing. The problem of scheduling work in cloud processing is to do N work on the M source. In this case, it is necessary to make the best possible operation of this N work on the machine M, in such a way that the final completion time of all tasks is reduced. The scheduling benefit in cloud processing is to switch to processes that are in standby mode and one of them to be selected to run. The choice of which method makes it possible to create scheduling war.. all algorithms have a specific way for selection. The purpose of most of the scheduling process is to reduce the execution period of work flows [7]. Typically, cloud processing algorithms can be classified into the following categories:

A. Exploratory Scheduling Algorithms

In the light of a straightforward characterization, work booking calculations can be arranged into two fundamental gatherings in cloud environments, the batch mode exploration scheduling algorithm and Immedia's exploratory scheduling algorithm [8]. The batch mode of the exploratory scheduler before the time is defined, the tasks are defined on a many factors before they begin to run, such as the Min-min and Max-min Exploitation Scheduling, Immedia exploration.
scheduling, mapping of tasks in the cloud resources upon entering the scheduler. Such as the FIFO service algorithm [9]. Exploratory algorithms include the following:

**B. Min-Min Algorithm**

This algorithm was proposed by Braun and his colleagues in 2001 for use in distributed systems, grids, and cloud processing to minimize resource allocation time for completing requested tasks [10]. The purpose of this algorithm is to reduce completion time of task by assigning resources to small tasks at an early stage. In this algorithm, at initial, an arrangement of least occasions for each work is figured on the source, at that point a work that has a shorter finish time in its accumulation is chosen and allotted to its proposed machine. The chose work is expelled from the arrangement of undertakings and by adding the runtime to whatever remains of the work on the chose source, we refresh the execution time of the other work.

**C. Min-Max Algorithm**

This calculation was proposed by Ibarra in 1977. In this calculation, initially an arrangement of least for each errand is computed on the resource, at that point a work is chosen that has the greatest consummation time in its gathering and is assigned to its planned machine. The chosen work is expelled from the arrangement of undertakings and, by adding the runtime to whatever is left of the work on the chosen resource, we refresh the execution time of the other works.

**D. Random Algorithm**

The possibility of irregular calculation is haphazardly to relegate chosen work to the source. In this calculation, no consideration is paid to whether the heap volume is light or heavy [11]. Hence, the results of the selection may be a source under heavy load, in which case tasks must wait before the service is dedicated. The unpredictability of calculation is low and the undertakings do not have overhead and preprocessing.

**E. Most Fit Task**

It is responsible for finding the most appropriate work for the available server. This search is from the initializing of the queue. It is known as the most commonly used method for categorizing the second type.

**F. Serving Algorithm in Order of Entry**

The idea of the service algorithm was introduced in the distributed networks first in 1989 by Brent, and in 1998, by Schweighelzln where it was used in grid and scheduling networks for parallel work in cloud processing. This calculation relegates errands to the source arranged by info. The determination foundation in this strategy is the entry time, indeed, the administrations are served by the cloud server in a similar request they enter the framework.

**G. Round-Robin Scheduling Algorithm**

This calculation is a non-selective scheduler. The scheduler doles out to each procedure a consistent time unit that instructs it to hinder and afterward hovers between them. The mix of this calculation and the administration calculation, individually, is utilized to enter the work scheduler in a cloud domain, actually, by coming to the interruptible, the running assignment enters the line, and the following occupation is chosen dependent on the administration calculation arranged by passage.

**H. Greedy-R Algorithm**

In this algorithm, tasks with the fastest running time are allocated to the strongest available cloud source to maximize system response time.

**I. Greedy-P Algorithm**

In this algorithm, tasks with the first fast execution time are allocated to the weakest available cloud source to maximize the response time of the system.

**J. Awareness-Source Scheduling Algorithm**

This calculation is a half breed calculation, Saeed Parsa and Reza Maleki, in 2009, proposed a calculation that assessed the dispersion and adaptability of disseminated frameworks, for example, matrix and cloud handling. This calculation is made by breaking down the two min to min and max to min calculations and the benefit of min to min and max to min calculations and covering their impediments. The strategy of this calculation is to think about an arrangement of errands, if the quantity of assignments in the set is combined, the Max-min technique is utilized to allocate assets to the work, and if the quantity of undertakings is individual, the Min-min technique is utilized.

**K. Distributed Distribution Scheduling Algorithm**

In 2012, Mehdi Javanmard and his colleagues presented a reliable distribution scheduling algorithm in cloud computing of the environment. In this calculation, an enhanced algorithm has been created with an adaptive method and with the classification and taking into account the period spent on work in the qualification function. By scrutinizing and evaluating previous algorithms, work scheduling has been performed with parameters that are combined with a failure rate. For this reason, in the suggested algorithm, in addition to the previously utilized parameters, other parameter
parameters are used which we can obtain based on them with different scheduling. This work is accompanied by the algorithm. A big work is divided into small works. To balance the work, their shuttle time is calculated individually. Then, the scheduling of each single operation, taking into account the return time in the form of a collaborative work, ultimately improves the productivity and efficiency of the system, as well as its actual time presented with the previous algorithm and, ultimately, with the mechanism of presentation by the proposed algorithm, the total processing time in cloud computing is optimized relative to previous algorithms.

VI. ALGORITHMS AND CRITERIA FOR COMPARISON

In this section, a comparison is made between the CCSH and the HEFT algorithm. Simulation has been used due to hardware constraints and high cost of renting cloud resources. The two factors of the average cost of money and the total time scheduling are compared together. Cloud configurations are performed to simulate 222 homogeneous virtual machines, and it is assumed that the communication network is strong. The speed of the virtual machine is determined by uniform probability of the distribution. Fastest ever virtual machine is constantly 4.2 GHz and the fastest slowest virtual machine ever to be 1.33 GHz.

A. Assessing the Cost-Conscious Cost Factor $\delta$

To evaluate the factor $\delta$, we consider a graph with 42 task and $\delta = 00102, ..., 0910$. The results in Fig. 6 show that, as long as the total length of the scheduling is minimal, the appropriate of the value of $\delta$ is effective in optimizing monetary cost. The values of the y axis indicate the cost of money saving (CR) and the rate of reduction in runtime (ER) between HEFT and CCSH. The greater the cost savings, the lower the runtime.

B. Random Generated Graphs

Here we use a large graph with 24242, 241241241, 224424 functions. The efficiency of CCSH and HEFT were compared in these of the large graph, respectively. The results of comparing these large of the graphs are shown in Fig. 7. As you can see, we are able to store nearly 33% to 32% of the total amount of money in overall cases when the total scheduling is a less than 14%.

VII. CONCLUSION

The most important challenge in cloud processing is scheduling to provide the best possible service to users. In cloud topology the scattering of assets experience issues of designation which cannot be satisfied with current asset apportioning methods. We still face problems to choose productive and suitable scheduling algorithms for cloud processing. Considering the nature of scheduling and its role in cloud processing efficiency, we investigated various categorizations of scheduling algorithms and common scheduling techniques in cloud topology. We made
comparisons between two scheduling main algorithms: CCSH and HEFT. The use of simulation was outlined to overcome hardware limitations and avoid the cost of renting cloud devices. The two parameters of average cost and total period scheduling were used for the comparison. Various cloud configurations were used to simulate 222 homogeneous virtual machines, with the assumption of a fast enough communication network unaffected by the number of the machines. The speed of the overall virtual machine is limited by the probability of distribution. We found that static and dynamic batch algorithms have the fastest processing time, in terms of efficiency of task processing and service quality. The Buffer-Pool scheduling model employed an integrated management role of all computational tasks and the bottom layer of the source related to them. In practice this model solves: i) the management weakness of online parallel processing of the data under SaaS platform, ii) enhances SaaS structure quality of service and iii) reduces the lost power resulting from duplicate computing. Until now common scheduling algorithms were not intended to save money. Awareness-cost scheduling showed that the CCSH algorithm is able to reduce the payload time, in proportion to monetary cost, to increase the time spent on performing the task. This will help to improve the saving of monetary costs and improves maintenance efficiency. Future work will focus on the integrity and comprehensiveness of E-Commerce in the existing cloud infrastructure and to use the concept of the Pareto Optimality, named after the Italian Vilfredo Pareto and defined as: a state of allocation of resources from which it is impossible to reallocate so as to make any one individual or preference, see [12].

REFERENCES