

Task Scheduling in Cloud Computing: Review

Dr. Ranjan Kumar Mondal

Assistant Professor, Department of CSE, Swami
Vivekananda University, Kolkata, WB

ranjankm@svu.ac.in

Abstract

Task scheduling plays a crucial role in cloud computer networks. Task scheduling is governed by several guidelines and agreements that we may refer to as a contract between cloud service providers and consumers rather than a single set of criteria. This agreement represents the level of service quality that the user expects from the providers. Delivering high-quality services to consumers by the deal is crucial since providers simultaneously manage several other responsibilities.

Finding or exploring the best mapping or assignment of a collection of subtasks of various tasks across the available set of resources (processors/computer machines) to accomplish the intended objectives for functions may be seen as a task scheduling issue. In this work, we conduct a comparative analysis of several algorithms to determine their applicability, practicality, and adaptability in a cloud situation. We then attempt to suggest a hybrid strategy that may be used to improve the current platform further and enable cloud providers to provide higher-quality services.

Keywords: GA, PSO, Task Scheduling, Scheduling Types, Cloud Computing, Cloud Architecture

Overview

Although there isn't a precise definition for the term "cloud," we may use many approaches and methods to describe it. Cloud computing is a supercomputing method linked to the Internet. It is a shared infrastructure that only uses

distributed computing, virtualization, and other techniques to join the massive system pools. Users may upload data and access processing power using their computers. The cloud computing environment over the Internet provides them with a range of networking, storage, and computing resources (Kaur & Verma, 2012).

“Cloud is a parallel and distributed computing system that consists of a collection of interconnected and virtualized computers that are provisioned dynamically and presented as one or more unified computing resources based on service-level agreement (SLA) established through negotiation between the cloud service providers and users,” according to R. Buyya’s definition of the term (Kaur & Verma, 2012).

Cloud computing is a distributed computing architecture that operates on a vast scale and is dependent on the economic size of an abstract, virtualized, dynamic cloud operator. Its primary function is managing storage, processing power, and other platforms and services.

They are given on demand over the Internet to outside consumers. Cloud computing is a rapidly developing computational paradigm that aims to relieve cloud users of the maintenance of hardware, software, networks, and data resources and assign these responsibilities to cloud service providers (Gu et al., 2012).

Numerous resources are available via clouds, such as computing platforms, data centers, storage, networks, firewalls, and software in the form of services. Additionally, it offers methods for controlling these resources so that cloud users may use them without experiencing any performance-related issues.

Based on the amount of abstraction and the service model used by the providers, cloud computing services are classified into three classes: (1) Infrastructure as a Service,

Software as a Service, Platform as a Service, and (3) Both.

The three fundamental components of cloud computing are virtualization, elasticity, and distribution. Virtualization is one of the critical components of the cloud. The majority of

hardware and software support virtualization. Numerous components, including operating systems, storage, software, and hardware, may be virtualized and managed on cloud platforms.

Related Work

One of the task scheduling issues that has been researched in the past is the NP-hard [3] problem. Due to scheduling's NP-hardness, it is mainly controlled by various well-known heuristic approaches that only partially solve problem cases. A heuristic algorithm guarantees an approximation of the optimum solution finding in a time less than that of a polynomial. To find a route within the specified solution space, it disregards and eliminates any other choices that look reasonable. The following categories also apply to heuristic-based algorithms: 1. Organizing within groups 2. using lists for scheduling.

Clustering techniques [5, 6, 7] assume that many processors are available to do the job for the subtasks. They utilize as many processors as feasible to shorten the creates schedules' makespan since they believe many processors are available for work.

The list scheduling class includes the majority of heuristics. There are two stages to the list scheduling algorithm's searching process: in the first, each subtask is assigned a priority based on a set of criteria, and after establishing.

When a job reaches a lower priority, it is added to the list of tasks that must wait for its turn. In the second phase, the highest priority subtask is chosen from the list and given to the most appropriate processor when the various processors become available to handle new tasks. Therefore, using additional greedy heuristics, the search is constrained to smaller areas of the solution space, improving the makespan. Its inability to provide consistent outcomes for diverse computer systems is a downside.

Meta-heuristics-based algorithms use a combinatorics procedure and these heuristic-based methods to locate the answer. These algorithms have shown optimal performance

across many scheduling problems. However, they need an adequate sample of solutions inside the specified search space. It has proven possible to use Particle Swarm Optimization [8], Simulated Annealing [9], Genetic Algorithms [10], and Ant Colony Optimization [11] for a variety of scheduling issues efficiently and successfully. The main disadvantage of these algorithms over heuristic-based ones is that they are less effective and have more considerable computing costs.

The Cloud Architecture

The front and back end are the two most important parts of cloud computing's architecture. The portion of the cloud users can see is called the front end. This covers the software and hardware users use to reach the cloud. All that remains of cloud computing's back end is the cloud itself, mostly made up of computers and storage devices.

A cloud environment consists of software programs that may be accessed online and used as services whenever needed. Cloud-based applications are designed so that the underlying computing infrastructure is only used when necessary, drawing resources only when requested, carrying out specific tasks, then giving up unnecessary resources and frequently disposing of them once the task is finished. The applications dynamically scale up or down in response to resource requirements while they are in use.

Cloud architectures tackle significant challenges primarily associated with handling massive volumes of data. In

When using a conventional data processing method, acquiring the number of computers required for an application to run well might be challenging. Obtaining machines when one wants them is one of the hardest things to do. It is difficult to distribute and coordinate a large-scale job over many computers, run processes on them, and set up a backup system if one fails during the operation. Additionally, it is difficult to scale up and down in response to shifting workloads automatically. Getting rid of all that equipment is not easy when the work is done.

Applications developed using cloud architectures operate in the cloud, where cloud providers choose where the

infrastructure is physically located. They profit from the straightforward APIs of web-based services that may grow on demand. In contrast, the intricate logic behind their dependability and scalability is executed and concealed within the massive cloud.

Figure 2. Cloud Service Layers

IaaS, or infrastructure as a service, is the lowest tier of cloud computing systems. It essentially offers virtualized resources, such as storage, communication, and processing power, that are accessible whenever needed. PaaS facilitates simple cloud programmability. One word for the software delivery paradigm is SaaS.

Synopsis: Work Scheduling

Cloud computing comprises many resources that vary in a few ways. Because cloud computing services depend on these resources, scheduling tasks in the cloud differs from traditional scheduling methods because cloud computing tasks are more expensive to complete using cloud computing resources.

To increase the adaptability and dependability of cloud systems, task scheduling is essential. The primary motivation for allocating tasks to the resources in compliance with the time limit is determining the most efficient and comprehensive order in which different activities may be completed to provide the user with the most optimal and gratifying outcome. On cloud

Computers and resources (such as networks, firewalls, and cups) are always dynamically assigned according to the order and specifications of the job and its subtasks. Work scheduling in the cloud becomes a dynamic challenge, meaning no previously established sequence may be helpful when processing the work. The unpredictable task flow motivates the dynamic scheduling approach, leading to uncertain execution routes. Additionally, several tasks sharing resources concurrently create uncertainty about available resources.

Task scheduling in the cloud refers to selecting the most appropriate resource for the job at hand or allocating computers to projects in a way that minimizes the amount of time needed to complete them. In scheduling algorithms, a list of tasks is generated by assigning a priority to each job individually. The parameters used to determine the priority of each work might vary. Then, based on their priority, tasks are selected and allocated to processors and computers that meet predetermined target functions as they become available [12].

Scheduling Categories

Static scheduling: This method arranges work in a predetermined context, meaning it already knows about the task structure, resource mapping before execution, and task execution/running time estimations.

When making scheduling decisions, dynamic scheduling must consider the status of computer systems and the tasks submitted to the cloud environment.

Cloud computing leverages virtualization techniques to execute user tasks and map cloud resources to the virtual machine layer. As a result, task scheduling in the cloud computing environment is accomplished at both the applications and virtual resource layers. Scheduling entails mapping activities and resources per specific rules to achieve a particular objective. The cloud computing paradigm makes it easier to connect tasks to resources. Since the necessary resources are all virtual machines (VMs), finding the right resource package is as easy as looking through the different VMs.

Analyzation Of Current Work Scheduling Algorithms

Cost-Time Optimization Algorithm based on Budget Distribution and Deadline [13]

This method considered two limitations: money and the job completion date. This study introduced the (DBD-CTO) scheduling method to plan jobs in a cloud computing environment. This method completes jobs ahead of schedule to meet its objective while minimizing computing costs incurred during operation.

Enhanced job scheduling in cloud computing environments using a cost-based algorithm[14]

The paper's author suggested the method known as the enhanced cost-based scheduling algorithm. His work's primary goal is to arrange tasks on cloud computing platforms with resources that vary in computation speed and cost. When jobs are grouped, communication between jobs and resources maximizes the computation/communication ratio. The computing performance and resource cost were assessed using this algorithm. Combining several tasks during execution also enhanced the ratio of task execution to data transmission across tasks. Combining tasks is often done after evaluating the processing and capabilities of various available resources. The simulation was run using CloudSim, and its inputs were the tasks' average MI, the tasks' granularity size, the total number of tasks, and the task overhead time. The outcome of his study demonstrates that, for this specific method, the time required to finish tasks after grouping is much shorter than the time needed to complete tasks without grouping.

A PSO-based Heuristic for Cloud Computing Environment Workflow Application Scheduling [15]

The study introduces us to an approach based on particle swarm optimization. Here, the cost of calculation and data transfer are considered while scheduling jobs. The author's primary goal is to reduce the overall cost of running the apps on cloud service providers' resources. The work completed for this article is split into two sections: 1. creation of a task-resource mapping model to reduce execution costs overall. 2. creating a heuristic based on the model above that solves task-resource mappings using PSO. The suggested method considers telecommunication costs for Every job, considering the different connections among them. In light of this, the PSO-based algorithm outperformed the "Best Resource Selection" (BRS) method in terms of cost savings, saving three times as much money and achieving a better workload allocation across resources. By increasing the particle size and resource quantity, the algorithms suggested in this study may be used for a wide range of jobs.

Particle Swarm Optimization for Multi-Objective Task Assignment in Cloud Computing [16]

The author of this article schedules jobs here to lower the overall cost of transferring and executing and the overall time required for transmitting and executing. The task scheduling issue in this work is allocating all task data to all resources in a cloud computing environment to minimize overall processing and communication costs. Task scheduling may be the mapping that associates each task with a Directed Acyclic Graph (DAG) $G(V, E)$. Every node has a weight, which primarily indicates how much data needs to be processed by a job on a particular node. The weights assigned to each edge indicate how much information is transferred from one activity that has to be handled to another. This algorithm's goal is not the only one; it also considers the cost, time, and expense of processing and sending data. The experiment's outcome demonstrates how effectively this algorithm reduces all costs and times in a cloud computing environment.

Particle Swarm Optimization Workflow Scheduling Algorithm for Cloud (BPSO): A Bi-Criteria Priority Approach [17]

This article presents a method for allocating workflow tasks to cloud resources to minimize execution time and cost within the specified deadline and budget. The workflow activities in BPSO are carried out according to their priority, determined by utilizing the bottom level, which is the same as what is specified in HEFT [20]. The PSO is then initialized using the provided priority. Following the priority assignment, the tasks are arranged in decreasing order of importance.

Degree. Afterward, the tasks are sent to various processors according to how they should be completed to finish the workflow application. The author created Java simulation software with a data center that included six resources with varying processing speeds for the experiment. The results of the experiment indicate that this method performs comparably to PSO.

Genetic algorithm-based scheduling of independent jobs [18]

The author of this work presents GA in task scheduling to accommodate memory limitations and demands for high performance in cloud environments. The author assumed several things about the tasks: 1. Tasks come in waves. 2. Work is not preemptive. 3. There are two methods for a process unit to access a task: shared and exclusive. The central scheduling technique is also assumed in this study [21, 22]. The method created in this research takes time consumption and resource use into account synthetically, yielding satisfactory results.

Task Scheduling Using Genetic Simulated Annealing Algorithm [19]

The study created an algorithm while considering the benefits of both genetic algorithms and simulated annealing. The algorithm considered the QOS needs of various job types based on the kinds of tasks users do in the cloud. QOS is only a measure of how satisfied customers are with the services. This study argues that since the parameters have multiple orders of magnitude and different dimensions, it is necessary to treat the parameters dimensionless when evaluating how activities are implemented in cloud resources. Various weights of factors might be assigned based on the kind of task to identify resources that meet the requirements and the quality of the work. The algorithm initially executes the GA steps, and then annealing occurs, enhancing the GA's local search capability. This method effectively concludes the cloud computing resource search and allocation procedure.

Future Scope

Since task scheduling is one of the most well-known issues in cloud computing, there is always a possibility that work in this specific area that has already been accomplished may be modified. The researchers carried out their work by their area of expertise, and eventually, other individuals completed their jobs. They had considered several approaches and applied limitations when scheduling, but they could not capture every aspect simultaneously because cloud computing is so large. However, they did note that there is a possibility that

algorithms will change and that part of the system may need to be modified.

Conclusion

Task scheduling remains a fundamental and challenging problem in cloud computing due to the heterogeneous, dynamic, and large-scale nature of cloud resources. This review has examined various scheduling strategies, ranging from traditional heuristic and list-based approaches to advanced meta-heuristic techniques such as Genetic Algorithms, Particle Swarm Optimization, Simulated Annealing, and their hybrids. The analysis shows that while heuristic methods offer lower computational overhead and faster decision-making, they often lack adaptability and optimality in diverse cloud environments. In contrast, meta-heuristic and hybrid algorithms demonstrate superior performance in optimizing key objectives such as execution time, cost, resource utilization, and Quality of Service, albeit at the expense of higher computational complexity. Therefore, no single scheduling algorithm is universally optimal for all cloud scenarios. Future task scheduling solutions should focus on hybrid, adaptive, and multi-objective approaches that can dynamically respond to changing workloads, resource availability, and user constraints, enabling cloud service providers to deliver more efficient, reliable, and cost-effective services.