Improve Mobile Cloud Computing Speed and Data Security Using Multi Objective Task Scheduling Approaches

Dr. GURURAJ A NAGALIKAR
Assistant professor Department of Computer Science,
GOVERNMENT FIRST GRADE COLLEGE SHORAPUR, DT. YADGIR

Abstract:
The creation of MCC has made it possible for future generations to gain access to a lot of knowledge on the computing concept. Providing mobile users with cheap, Pay-As-You-Go access to a shared pool of virtualized cloud resources including storage, data processing, network, and s/w apps over the internet is what we call mobile cloud computing (MCC). Mobile computing mixed with cloud computing (MCC) enhances mobile device performance via offloading techniques. To make up for mobile devices' flaws—such as limited storage, short battery life, inadequate sensing capacities, and slow CPU speed—mobile applications tackle a wide range of problems, such as Quality of Service (QoS), mobility management, energy management, resource utilization, and issues related to data privacy and security. The MCC addresses these issues by moving the resource-intensive parts of an app to the cloud and by balancing the workload between the mobile device and the cloud in such a way that the task priority requirements are met. Because the cloud stores and processes so much information, its performance suffers if jobs are poorly scheduled. That's why MCC puts so much faith in its time-management system.

Keywords: *Mobile Cloud Computing (MCC), Quality of Service (QoS), Data privacy & security, offloading techniques.*

DOI Number: 10.48047/NQ.2022.20.3.NQ22973 **NeuroQuantology2022;20(3): 1115-1121**

Introduction

As a widely adopted paradigm, cloud computing promises to meet the on-demand service needs of consumers and enterprises alike. It has altered how people approach their work and how they learn about and manage computing services. Cloud computing refers to the sharing of computing resources such as data storage, network infrastructure, platform software, and client access to applications across an online network. Client devices may include smartphones, tablets, notebooks, desktops, and the like[1][2]. Because of its potential for expansion in meeting consumers' evolving service needs, cloud computing has gained widespread recognition as both a computing platform and an acceptable business model. In a "Pay as You Go" model, cloud computing delivers IT infrastructure, platforms, and applications as online services. The services are a la carte, so customers may pick and choose what they need. Cloud service providers are the ones in charge of the underlying infrastructure, and their customers could be either businesses or individuals. Virtualization, grid technologies, service oriented architecture (SOA), metering tools, etc., all contribute to the growth of this on-demand system. The National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (such as networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Figure 1 shows how cloud computing may be broken down into two distinct sets of models, the deployment model and the service model, depending on factors like deployment location and cloud service type.

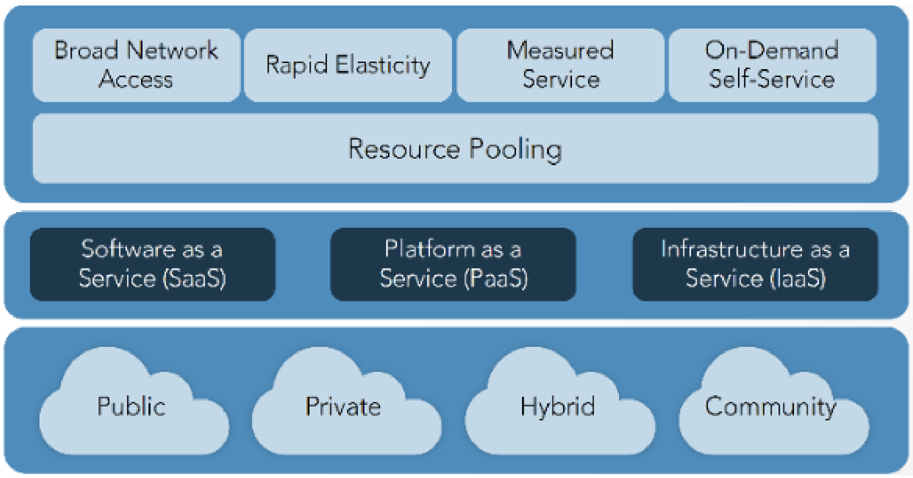

Figure 1: The NIST Model of Cloud Computing

The cloud's purpose is established by its deployment methodology, which is dependent on the cloud's physical location. Private clouds, public clouds, hybrid clouds, and community clouds are the four primary cloud deployment models[3].[4]. The private cloud is developed for a company's exclusive internal use. The private cloud's software and hardware must be purchased and maintained by the company itself. Even though a company relies on infrastructure built by another entity, it may retain full control over its operation and use of the resulting data. The site could be on or off campus. Safety and privacy are prioritized because threats are mitigated here. Cloud service companies deploy the public cloud for general consumption. While the user isn't worried about allocating resources, security and privacy issues with the data are always present. When you combine private and public clouds, you have a hybrid cloud. These clouds are a cohesive group that also manages to keep its individuality. Public cloud is used to serve end consumers, whereas private cloud is used to safeguard an organization's internal operations. There is still cause for alarm regarding organizational reliance, the integration of several cloud security platforms, and the allocation of workload [5]. A community cloud is designed to meet the demands of a specific set of users who share similar concerns about the cloud's goal, policies, regulatory compliance requirements, and security. Third parties or, more typically, the participating businesses and organizations set up and operate the infrastructure. By pooling resources, knowledge, and expertise, teams can maximize output, decrease response times, and save money. The Stack Model's three services are essential to the service model architecture, which are as follows: Cloud computing may be broken down into three categories: infrastructure, platforms, and applications. With IaaS, clients don't have to worry about the underlying hardware (such as servers, networks, and storage devices). Google's distributed File System and Amazon's Database as a Service (SimpleDB) are two examples of dedicated infrastructure services. Developers take advantage of the cloud services provided by the PaaS. This encompasses anything from computer programs to control systems and development frameworks. These services allow programmers to write and deploy their applications on the cloud service provider's infrastructure [6]. Facebook's Platform is an environment for Facebook applications, while Google's App Engine is an application runtime for the web, and Microsoft's Azure is a platform for developing and running Windows programs. The end user is the primary focus of software as a service (SaaS) programs, which eliminate the need for local software installation and make all necessary tools available online. Online office suites like Google Docs and Microsoft's Window Live are just a few examples. Others include Google's Maps API for integrating maps and geographic data and Adobe's Photoshop Express for online image editing. The umbrella term for the aforementioned three cloud service models is the Software Platform Infrastructure (SPI) model[7]. Human-as-a-Service (HuaaS) (C. Baun et al., 2011) is an alternative model that treats people like a resource and treats the services they provide as a commodity. Crowdsourcing by Amazon (an Amazon Mechanical Task) is a common example of this, as it allows for the accommodation of fine-grained jobs by engaged and concerned human resources in exchange for an equally modest remuneration. Some of the most notable characteristics of cloud computing include its virtualization, dependability, usability, extensibility, and autonomy. Mobile Cloud Computing (MCC) is the outcome of developments in the cloud service model; under MCC, users can access cloud-based services from their mobile devices[8]. The convergence of cloud and mobile computing not only enables ubiquitous access to cloud services but also alleviates the limitations of resource-constrained mobile devices.

Literature Survey

To improve the functionality of smart mobile phones with limited hardware, B. Chun and colleagues (2011) suggested using a clone cloud architecture. A clone was characterized as a copy of a smartphone that was hosted on a cloud-based VM. The core idea was to leverage virtual machine technology to seamlessly or substantially replicate an application's execution environment on smart mobile devices and then extend that execution to a distributed setting.

A computational offloading architecture, Phone2Cloud, was presented by F. Xia and coworkers in 2014. The design goal was to increase application performance while simultaneously minimizing power consumption in smart mobile devices (SMDs). Unlike previous frameworks, this one used application and scenario trials to do a quantitative study of the system's energy savings. This was done to see how much power could be conserved.

The technology introduced by H. Qian and coworkers in 2015 was dubbed "Jade." This system monitored the application and the device's health to determine where the work could be accomplished. Its primary goal was to reduce the time and effort required to create mobile apps that take use of computational offloading and are energy mindful, while simultaneously boosting these benefits for end users.

Hongyou L. and coworkers (2013) presented a model that drew from multiple approaches. Specifically, an energy-aware scheduling algorithm that employs the workload-aware consolidation technique (ESWCT) and an energy-aware live migration approach that employs the workload-aware consolidation technique (ELMWCT).

Xue Lin and colleagues (Lin et al., 2015) investigated the work scheduling difficulties in MC and proposed a minimal-delay technique that reduced power usage by moving the jobs away from the local mobile cores and into the cloud. The local mobile cores were used to transfer the duties. The energy consumption of the subsequent phases of this algorithm is minimized by employing a technique called Dynamic Voltage and Frequency Scaling (DVFS). For the sole aim of relocating jobs, a linear-time rescheduling technique was developed. However, order O (N3K) computational complexity prevents the method from being practical.

Hosseinimotlagh S. et al. (2015) introduced a virtual machine scheduling method. The QoS requirements were met thanks to the algorithm's determination of an ideal amount of energy use. By adjusting the power provided to a virtual machine on a host, you can get it running at peak efficiency. Only when the host's CPU utilization was higher than the optimal value did the algorithm's performance improve slightly.

Multi-objective task scheduling with autoregressive dragonfly optimization (ADO-MTS)

Smartphones, tablets, and laptops have quickly become the most popular computers because of their portability and ease of use. Data and processing are offloaded from mobile devices and onto an external server in the cloud while utilizing a mobile cloud application. This reduces the need for processing power, saves local storage, and lengthens battery life. By (i) selectively shifting processes of an application to the cloud, and (ii) carefully arranging things to be done in the mobile device and in the cloud with the task-precedence demands in mind, MCC seeks to increase mobile device performance and energy efficiency. One method by which this will be accomplished is by (i) executing a subset of an application in the cloud. One uses a scheduling algorithm to plan and carry out the jobs so as to optimize the computed gain or benefits [9].[10]. Cloud performance can be negatively impacted by the huge amount of data being handled if jobs are not properly arranged. Therefore, the method used to schedule jobs is an integral aspect of MCC. A data center's scheduling approach should take into account CPU, connectivity, memory, and other resources. In order to schedule resources effectively and efficiently, this is required. Therefore, the (ADO-MTS) approach was created to schedule work in the cloud while considering energy usage.. The ADO-MTS optimizes in order to meet three goals: energy efficiency, makespan, and resource utilization [11]. Whereas in an MC setting energy estimates are made based on the frequency of operation of the cores during task scheduling, in a cloud setting they are developed based on the energy requirements of individual applications and their dynamic executions. Makespan gives details on how long it takes to complete each cloud-based job. When developing the ADO-MTS method for scheduling cloud-based jobs, researchers found that combining the CAViaR strategy with DA resulted in the most efficient scheduling. Resource consumption, task duration, and energy output are used to calculate the efficacy of the suggested method [12].

The shorter lifespan of batteries used in mobile devices is contributing to the power constraint that has resulted from the increasing demand for batteries and the advancements in their energy density. Considerable energy consumption must be taken into account while scheduling interdependent tasks. Traditional approaches to task scheduling for shared platforms like clusters, grids, and clouds prioritize reducing the time required to execute the job rather than the energy required to do so. Time to completion and energy consumption may fluctuate significantly over the course of a run due to physical defects, process instability, or variations in voltage or frequency. Therefore, scheduling is seen as a challenge in cloud computing. This is owing to the fact that it requires taking into account not just various energy issues, but also the functional characteristics of resources. The current scheduling methods focus mostly on fairly allocating the available resources. In reality, however, the many resource-intensive apps running on the VMs also have a major effect on the system's overall performance and energy consumption. Therefore, it is crucial to consider all of these aspects when planning the activities.

This is an example of the ADO-MTS approach described for scheduling cloud-based operations. Block diagram of the proposed ADO-MTS method is shown in Figure 1. The task buffer and the task manager coordinate all of the work that users have asked to be done in the cloud. There are many resources in the cloud, and they all need to be powered. The task manager gathers data on how much power, time, and materials are being used. In order to allocate tasks to the available cloud computing resources, an efficient task scheduling method is required. Therefore, the ADO scheduling method includes the multi-objective model as a component of the fitness function. So, jobs are sent to the most appropriate cloud resources considering energy usage, Makespan, and resource utilization.

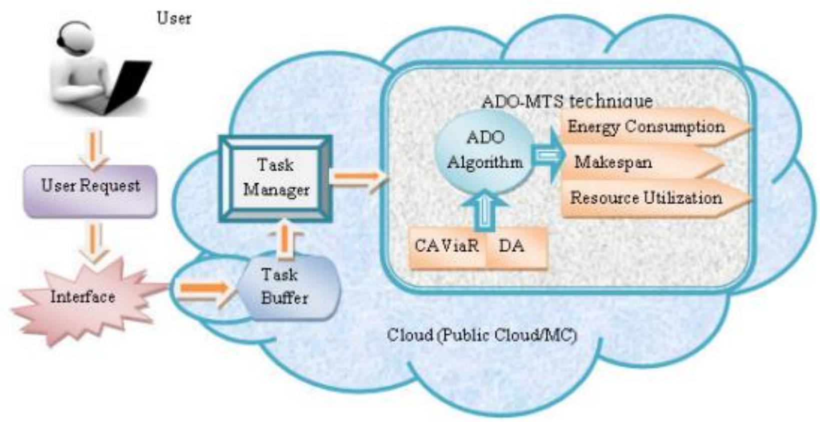

Figure 2.Schematic Representation of the ADO-MTS Method

However, there are concerns about data privacy and security that must be taken into mind while implementing cloud storage-related services despite the many advantages of MCC. Hiding any sensitive information while transforming raw data into sanitized data is a laborious operation. Furthermore, keeping sensitive data safe requires keeping and making use of data, so that the data given to the mobile user is useful. The MCC model used by the system to safeguard individual records is shown in Figure 2.

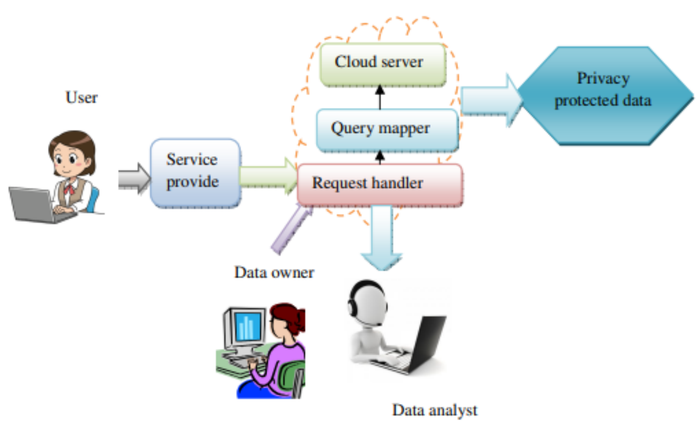

Figure 3: Model of MCC in Privacy Preservation System

The model is comprised of three essential components, namely the data owner, the cloud server, and the cloud user. Cloud users are individuals that are tasked with the responsibility of accessing information that is kept on a cloud server. It is the responsibility of the cloud user to make a request for a service, which must then be coordinated with the help of a query mapper and a request handler in order to get the required data. The cloud server is made up of individual servers, each of which may be either virtual or physical. If the owners want to keep the data stored in the cloud and communicate it to cloud users or end users, they will need to employ a method that effectively protects users' privacy. This work proposes a Self-Adaptive Autoregressive Dragonfly Optimization (SADO) technique by adapting a matrix product based model and by incorporating the self-adaptive concept in ADO algorithm. The goal of this technique is to reveal only the required details while hiding the sensitive information and to provide secure transmission among mobile users whose data are stored in the cloud platform. The privacy preservation rate and the utility missing rate are used to develop a new objective function, which is then applied to the evaluation of the optimization process in order to identify the ideal fractional order derivative coefficients, which are then put to use in the generation of the fractional matrix. Using a private key, the raw data is transformed into secure encrypted data and then stored here. After that, the created key is used to decrypt the data in order to recover the real source data that was encrypted.

Conclusion

The efficiency of a cloud computing system depends in large part on the effectiveness of its scheduling algorithms. This study provides a quick overview and analysis of the current job scheduling techniques available. We provide the first effort to jointly schedule tasks on the mobile device's local cores, wireless communication channels, and the cloud in order to achieve the lowest possible energy consumption while still meeting the task graph's hard completion time constraint in an MCC context. The majority of scheduling algorithms only care about a handful of inputs. By incorporating additional measures into existing approaches, a better scheduling algorithm can be constructed, with the potential for superior performance and outputs that can be deployed in a cloud environment in the future.

References

1. B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti (2011), “clone cloud: Elastic execution between mobile device and cloud”, in Proc. of the 6th Conf. on computer systems. ACM New York, pp. 301-314, April 2011.
2. F. Xia, F. Ding, J. Li, X. Kong, L. T. Yang, J. Ma (2014), “ Phone2cloud: exploiting computation offloading for energy saving on smartphones in mobile cloud computing”, In form. Syst. Front. 16 (1), pp. 95-111, 2014.
3. H. Qian, D. Andresen (2015), “Jade: reducing energy consumption of android app”, Int. J. Network. Distrib. Comput (IJNDC), 3(3), pp. 150-158, 2015
4. Hongyou L., Jiangyong W., Jian P., Junfeng W and Tang L. (2013), “ EnergyAware Scheduling Scheme Using Workload-Aware Consolidation Technique in Cloud Data Centres”, China Communications, 10(12), pp. 114-124, December 2013
5. Lin X., Wang Y., Xie Q and Pedram M. (2015), “Task Scheduling with Dynamic Voltage and Frequency Scaling for Energy Minimization in the Mobile Cloud Computing Environment”, IEEE Transactions on Services Computing, 8(2), pp. 175- 186, March-April 1 2015.
6. Hosseinimotlagh S., Khunjush F and Samadzadeh, R. (2015), “SEATS: smart energy-aware task scheduling in real-time cloud computing”, The Journal of Supercomputing, 71(1), pp. 45-66, January 2015.
7. Ismaila L and Fardoun A. (2016), “EATS: Energy-Aware Tasks Scheduling in Cloud Computing Systems”, Procedia Computer Science, 83, pp. 870-877, 2016.

8. Jadhav R.H. (2018), "Distributed Bottom up Approach for Data Anonymization using MapReduce framework on Cloud," International Journal of Advance Scientific Research And Engineering Trends, 3(6), pp. 109-113, June 2018.
9. Dr. G. Anandharaj, K. Suganthi (2019), "Energy-Efficient And Improved Qos - Driven Task Scheduling Algorithm In Mcc Environment Using Cloudsim Simulator", International Journal Of Scientific & Technology Research, 8(11), pp. 1098-1105, November 2019
10. Wang Z., Pang X., Chen Y., Shao H., Wang Q., Wu L., Chen H. and Qi H. (2019), "Privacy-preserving Crowd-sourced Statistical Data Publishing with an Untrusted Server", IEEE Transactions on Mobile Computing, 18(6), pp. 1356-1367, 2019
11. Xiong Y., Huang S., Wu M., She J and Jiang K. (2019), "A Johnson's-Rule-Based Genetic Algorithm for Two-Stage-Task Scheduling Problem in Data-Centers of Cloud Computing", IEEE Transactions on Cloud Computing, 7(3), pp. 597-610, 2019
12. Ashok George and A. Sumathi (2018), "Dyadic product and crow lion algorithm based coefficient generation for privacy protection on cloud", Cluster Computing, special issue 1/2019, pp. 1-12, 2018.