International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Exploration of Location based Services using AI for Mobile Cloud Services

K. Sandhya, Y.Sunanda

Abstract: ***Mobile devices have several sensors, including GPS that can capture information about the location of a mobile user. The use of certain devices will, therefore, simplify services and make it simpler for operators to respond to the demands of mobile users. The main aim of this analysis is to incorporate middleware to pick suitable cloud services that leverage from mobile device position and cost preferences. If the number of small activities within a meta feature exceeds the number of major work, the Max min algorithm device operations are conducted in addition to big tasks, where the design of the process is dependent on how many functions it does. The model is wide since tasks cannot be conducted simultaneously. A new amendment to the computation system is used to overcome the drawbacks of the Max-Min algorithm. It encompasses the positives of Max-Min and eliminates drawbacks. This study focuses specifically on the number of resources and incidents. The work can be further expanded with the algorithm suggested for the cloud system and several other parameters such as scalability, performance, reliability, and others can be taken into account.***

Keywords: Cloud Computing, Meta Task Scheduling, Max-min Algorithm, mobile cloud computing

I. INTRODUCTION

AI is a technique that aims to make machines humane. This will step up the digital transition in sectors. Whether it is human beings, livestock, trees, vehicles, machinery, water, bricks, oceans, houses or anything you can believe of, the environment can become an independent place to bring them together and make intelligent decisions. We need to have a machine learning (ML) [1], a framework emulating human cognition and analysis of data (DA) [2] in the program, to make the world and its physical objects autonomous. ML should build strategies for learning in different network components/devices to automatically make them autonomous, while DA would assess/analyze all of the data generated over time to detect past trends and become more reliable/beneficial in the long term. Location-based services (LBS) are services offered via mobile phones that take the geographical location of the user into account.

LBS usually provides entertainment or information. Since LBS depends heavily on the position of the mobile user, the main aim of the service provider's program is to decide where the customer is. Several methods are necessary to do this.

Local news, instructions, points of interest, directory service, fleet management, urgencies, asset tracking, position-sensitive building, and community advertising are some of the most popular LBS applications.

The convergence around cloud computing and mobile computing should transform how distributed resources are used. Mobile devices can enable the cloud repository and use unrestricted computing services at any moment, although they are not identical to classical platforms. Even though, they have sensing skills to collect mobile device background information. Context is usually where individuals, communities and machines and physical artifacts are situated, defined and in their environment. Mobile background information can, therefore, enable agencies to provide consumers with appropriate, reliable and context-aware programs that provide sufficient care in compliance with their actual situation. The above detail must, therefore, be used to enhance mobile cloud services as shown in Figure 1.

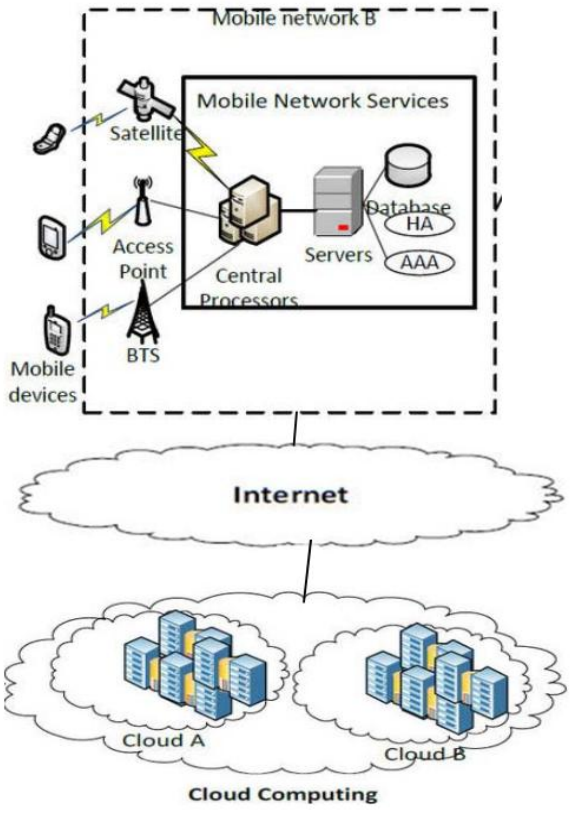

Fig .1: Overview of Mobile cloud computing construction

The goal of this paper is to add middleware that gathers mobile device background data such as position. It then chooses a service based on the cloud services and product criteria that are applicable. It also uses the chosen product fee that provides the most detailed information for the better selection of the service provider.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
K. Sandhya*, Assistant professor, Department of CSE GITAM University, Hyderabad, Telangana, India.
Y.Sunanda, Assistant professor, Department of CSE, Malla Reddy Institute of Technology & Science, Hyderabad, Telangana, India.

One characteristic of Max-Min's approach is that it first selects big tasks, which in effect postpone small tasks for a long time. And from the other hand, Min-min is suitable for smaller than broad jobs, which is the opposite of Max-min. The fundamental idea of an enhanced version of Max-min assigns resource output assignments with maximum time achieves the lowest total time rather than the initial Max-min assignment function of full execution time.

The main contributions and organization of this paper are summarized as follows: In section 2 we describe background details of QoS in mobile cloud services. Section 3 discusses the methods and methodology of work. Section 4 deliberates results and discussions. Finally, in section 5, we concluded the paper.

II. BACKGROUND WORKS

In [3], the algorithm suggested by the authors is based on the original Min-min algorithm [5]. It is called Min-min-guided QoS which assigns high bandwidth roles before others. QoS behaves like Min-min whenever tasks have the same bandwidth open, so it would choose to use QoS controlled Min-min if tasks have a major bandwidth. The QoS-guided Min-min requires more effort during that minute.

In [4] the authors suggest a dynamic verification framework that, owing to the implementation of mobile devices, is increasingly important. The extensive use of devices validates the program that can be applied in various ways and secures validity using the user activity of your software. Text verification schemes add security layers when using password-based confirmation methods. In some cases, they can also bypass traditional testing approaches. In this document, scholars suggest a realistic and expensive scheme of authenticity. Environmental data is only sent when a customer needs an identifier that makes the program more easily and gently.

In [5], the writers suggest a present, somewhat static, Monolithic robotic control system, with application data configured until the actual process, starts Additional real-time control loops are needed for process control and more efficient managing of robotics for more complex activities. Consequently, this paper represents a real-time and personal communication robot control system that can create locksmiths in real-time and large-scale cloud computing systems. The different complex configurations of the robot and controller are simulated elements in the proof of the principle. The stability of the real-time design is measured in comparison to the un-validated variant of the same specification.

In [6] Smart PFP dependent computation model was developed by the authors using the hierarchical algorithms. The work suggested lets researchers conserve electricity and improve efficiency by generating beneficial results. Studies have also focused on indoor orientation and position lately perceived to be main utilities. Wireless technologies such as Finger-Printing (FP) are the leading and commonly employed methods of placing and enhancing performance. Investigators keep comparing Wi-Fi Received Signal Strength (RSS) to the fingerprint radio database. The new suggested P-FP algorithm decreases the code sophistication of the standard FP solution, without further disruption to accuracy.

In [7], the authors were inspired by the simulation issue of the planning algorithm across cloud data centers for the assessment of the different output parameters. CloudSched has been developed for resource planning in cloud data centers dynamically. This work provided a cloud computing modeling framework with a variety of resource optimization algorithms and policies. IaaS (Infrastructure as a Service) was based on the application design framework.

III. METHODS AND METHODOLOGY

The software is an integrated system of cooperation intended to communicate with mobile device clouds. The middleware acts as a converter to monitor the situations of mobile devices, such as battery consumption, CPU usage, network speed, and sensor data, etc. Depending on this contextual information, the middleware tries to change every service order, as shown in Figure 2.

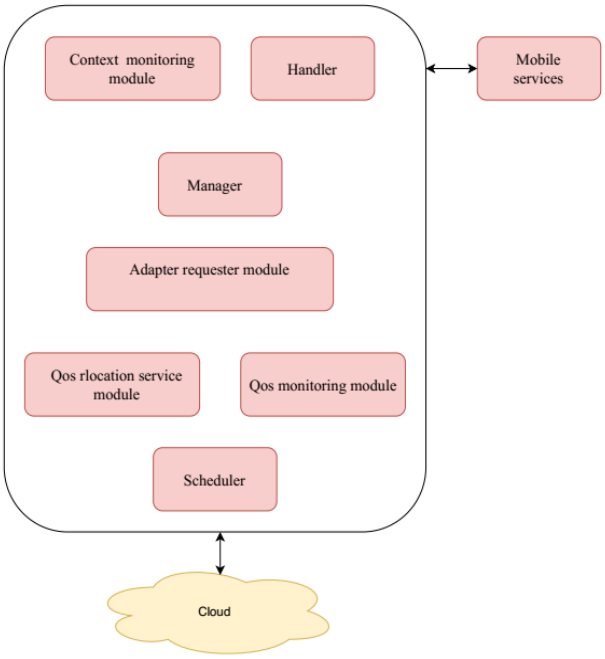

Fig. 2: Overview of Middleware architecture with cloud services

When a mobile application passes a request feature to the operator, the dispatcher will provide a query as well as the context management mechanism intercepts the current context of the machine user. The task is managed by the vendor, who initially sets up a session with a unique ID for several requests and for returning alerts whenever the cloud process has started. The manager then sends the code to the converter request module to broaden the traditional and upgraded Max-Min algorithms. This algorithm selects the function with the longest total time and assigns it to the quicker tool to make it run less. The Adapter Request module manages the service requests in compliance with the current mobile device background and QoS level given by the context management framework. The QoS level is chosen from the QoS-location services module which comprises the cloud resource provider lists and it's QoS at several locations and updates this module concerning mobile user mobility. The adapter request module then takes advantage of the position of the device to pick the closest cloud option. First, for each cloud tool, it uses the approximate completion time to choose the correct QoS level.

If the mobile user defines cost, the adapter selects the service based on the cost of the device; we display the algorithm (1) of the middleware portion.

The QoS module controls the QoS stage. The service provider or the information conveyed by the mobile device must render some divergence clear at QoS levels of service; then the background surveillance application or the QoS control module can announce the interface request module to reassess and re-connect to a stronger functional and QoS level service. The adapter request is then sent to the programmer who delivers the request to the cloud provider.

The method calculates the completion time for each required item. The function with either the expected average completion time is then allocated to a tool with the estimated average total time. Eventually, this configured function is eliminated from meta-tasks and all measured times are modified and reviewed until every requested request is carried out. It would be designed to reduce the combined size of the total time in highly clustered settings, such as the Cloud World, whereas simultaneously providing available resources to achieve a load balancing in smaller distributed systems. The suggested algorithm produces mapping schemes that are similar to other methods for such efficient analysis and reduces the time necessary to complete all operations.

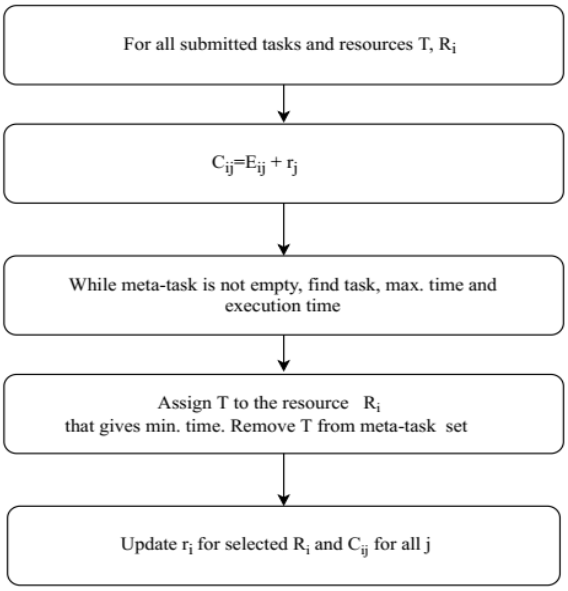

Fig. 3: The Improved Max-Min Algorithm

The option of work with a set duration is the highest mission. We go for the most gradually possible resource when choosing the resource for minimal execution. The implementation of the longest resource requires high-speed infrastructure to be used simultaneously for many other simple tasks. By the availability of resources, they often achieve the smallest possible amount of planned activities. Not as original Max-min, but only when the work becomes heterogeneous between completion and implementation because large jobs and tiny tasks are visible. Enhanced Max-min pseudo coding is shown in Figure 3. We refer to the anticipated time frame as Cj, which is specified as r_j, which reflects the ready time of resource R_j and E_{ij}, which is the execution time of Ti's R_j property. Figure 4 is a planned software flowchart. Initial Max-min identical to Max-min improved exactly except in Figure 4, phase (1).

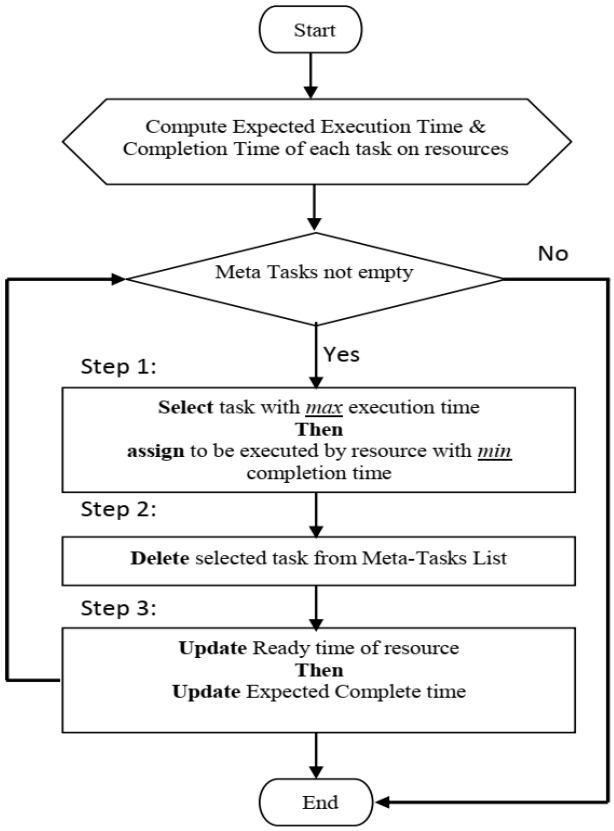

Fig. 4: Flow chart of Improved Max-Min

The Max-min algorithm has a comparable structure to the initial Max-min $O(mn^2)$, Min-min and RASA with m as the number of resources and n as the number of tasks. Even though the same implementation period is comparable to Max-min, Min-min, and hybrid, it provides a more robust system. Updated Max-min facilitates the load balancing of available resources and allows requested activities to be completed concurrently with a higher probability that original Max-min.

IV. RESULTS AND DISCUSSION

Cloud Analyst estimates the response time on each user base. Nonetheless, the details of the simulation determined by the cloud analyst are shown in Table 1. Upon running the simulation, we will note that it offers greater response time as a data center close to the customer. The key results indicate that getting the business closer to consumers increases service quality (in this instance, the response time). It is predicted that consumers are less impacted by Internet problems (high latency, poor bandwidth) while near to the application server geographically. That is why we deliver in our solution module the option of the cloud provider by position for the mobile user to reduce delegate response times while optimizing mobile device existence.

Table 1. Response Time by Region

Userbase	Avg (ms)	Min (ms)	Max(ms)
US1	52,23	46,56	67,56
US2	128,23	172,67	231,88
US3	49,23	46,78	64,23
US4	204,23	221,12	298,12

Table 2. Resources Specification

Resource	Processing Speed (MIPS)	Bandwidth
RS1	160	350
RS2	350	25

To illustrate our algorithm, assume that we have four T1, T2, T3, and T4 tasks and that the scheduler has two RS1 and RS2 tools like the 2 problems. In Table 2, the speed and the bandwidth of contact links are listed for each tool while Table 3 displays the number of instructions and data in T1 to T4 activities. To measure the estimated completion time and implementation time of activities on each resource using the data provided in Table 1 and Table 2.

Table 3. Meta-Tasks Specification

Task	Volume if Instr	Data volume
T1	226	83
T2	45	41
T3	314	92
T4	180	378

Figure 5 shows the different resources namely US1, US2, US3 and US4 is plotted on X-axis and time of execution on Y-axis. It is evident that RS1 have more processing speed and bandwidth as compared to the RS2.

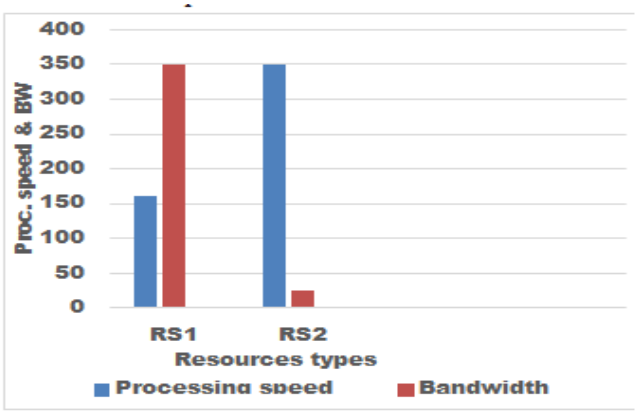

Fig.5: Comparison of processing speed of different resources

Figure 6 shows the different usebases namely RS1 and RS2 is plotted on X-axis and processing speed and bandwidth on Y-axis. It is evident that US4 have more time of execution as compared to the other userbases.

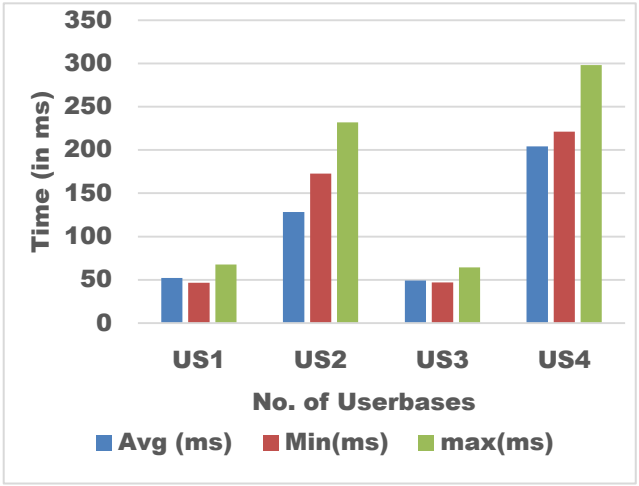

Fig.6: Comparison of duration of different usersbases

V. CONCLUSION

Mobile devices have many sensors, including GPS, capable of collecting data on a smart user's surroundings. The use of certain tools can, therefore, automate apps and allow operators to react more easily to mobile users ' needs. The principal goal of this analysis is to integrate middleware that chooses suitable cloud services based on the location and costs requirements of mobile devices. If the number of major tasks in a meta-function is greater than the number of big jobs, Max-min's application framework operations rely on as to how many and execute small tasks with large tasks concurrently. When functions cannot be done at the same time, the composition is high. The Max-Min programming algorithm is being revised lately to resolve these Max-Min algorithm pitfalls. It provides the strengths of Max-Min and removes the disadvantages. This work mainly reports on the number of resources and events. The study could be further extended by using the theoretical method for the actual cloud computing network and taking account of several other criteria like scalability, availability, stability, and other factors.

REFERENCES

1. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Machine Learning: An Artificial Intelligence Approach. Springer Science & Business Media, 2013.
2. I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2016.
3. X. He, X-He Sun, and G. V. Laszewski, "QoS Guided Min-min Heuristic for Grid Task Scheduling," Journal of Computer Science and Technology, Vol. 18, pp. 442-451, 2003.
4.] Benzekki, K., El Fergougui, A. and ElAlaoui, A.E., 2018. A Context-Aware Authentication System for Mobile Cloud Computing. Procedia Computer Science, 127,379-387.
5. Hinze, C., Tomzik, D.A., Lechler, A., Xu, X.W. Verl, A., 2019. Control Architecture for Industrial Robotics based on Container Virtualization. In Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter (pp. 64-73). Springer Vieweg, Berlin, Heidelberg.
6. Sciarrone, C. Fiandrino, I. Bisio, F. Lavagetto, D. Kliazovich and P. Bouvry, 2016. Smart Probabilistic Fingerprinting for Indoor Localization over Fog Computing Platforms. 5th IEEE International Conference on Cloud Networking, 3-5 October 2016, Pisa, Italy, DOI:10.1109/CloudNet.2016.43.
7. Tian, W., Zhao, Y., Xu, M., Zhong, Y, Sun, X. 2015. A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center, IEEE Trans. on Autom. Sci. and Engine. 12(1), 153-161.

AUTHOR PROFILES

K. Sandhya received her M. Tech in Computer Science and Engineering from JNTUH in 2012 and B.Tech Degree in Computer Science and Engineering from JNTU in 2006 and currently working as Assistant professor in GITAM University, Hyderabad, in the Dept. of CSE and has 8 years of teaching experience. Her area of interest in Artificial Intelligence, Machine learning and Cloud computing.

Y.Sunanda received her M. Tech in Computer Science and Engineering from University of JNTUH in 2012 and B.Tech Degree in Computer Science and Engineering from University of JNTUK in 2009 and currently working as Assistant professor in Malla Reddy Institute of Technology & Science, Hyderabad, in the Dept. of CSE and has 7 years of teaching experience. Her area of interest in Information Security, Artificial Intelligence, Machine learning and Cloud computing.