International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conferernce on Industrial Automation and Computing (ICIAC- 12th & 13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

MOBILE CLOUD MULTIMEDIA SERVICES USING ENHANCE BLIND ONLINE SCHEDULING ALGORITHM

Saiyad Sharik Kaji, Prof.M.B.Chandak
Department of Computer Science, Nagpur University, Nagpur-441111
sharik.kaji@gmail.com
chandakmb@gmail.com

ABSTRACT –
Mobile cloud is a new emerging technology which can be used to enable users to enjoy abundant multimedia applications in a pervasive computing environment. Therefore, the scheduling of massive multimedia flows with heterogeneous QoS guarantees becomes an important issue for the mobile cloud. Generally, the predominant popular cloud-based scheduling algorithms assume that the request rate and service time, are available for the system operator. However, this assumption can hardly be maintained in many practical scenarios, especially for the large-scale mobile cloud. In this article, we consider the scheduling problem for a practical mobile cloud in which the above parameters are unavailable and unknown. Taking into account the performance of the users and the impartial free time among the servers, the highlight of this article lies in proposing a blind online scheduling algorithm (BOSA). Specifically, we assign available multimedia servers based on the last timeslot information of the users' requests, and route all the multimedia flows according to the first come- first-served rule. Moreover, we design detailed steps to apply the BOSA to a content recommendation system, and show that the proposed BOSA can achieve asymptotic optimality.
Keywords **-** Scheduling of massive multimedia flows with heterogeneous Quality of Service.

I. INTRODUCTION

Currently, a large number of mobile cloud platforms have been proposed to provide a pervasive computing environment. In most of them, each mobile device is linked with a system- level service in the cloud infrastructure. Moreover, with the rapid development of wireless communication technologies, users are expected to use more multimedia services in the mobile cloud to avoid the installation of the software in mobile devices. Informally, it is called Mobile Cloud Multimedia Services (MCMS).

Clearly, when all the multimedia services move to the cloud, MCMS becomes larger and more complicated, thus it is necessary to design an efficient scheduling scheme that dynamically allocates appropriate user service requests to available multimedia servers without the help of a centralized controller. Basically, since various service requests usually come from different users and the scheduling policies for MCMS are typically delay-sensitive, intuitively, the users are allocated to the servers with less service time to reduce transmission and waiting delays. However, in this case, the faster servers naturally become busy since they possess less free time than the slower ones, and hence, this leads to significantly raise the energy consumption.

More recently, green multimedia services attracted so much attention and became an irreversible trend. As a result, it is necessary to enable all the servers to have the same (or similar) free time when assigning heterogeneous services. This encourages us to design an exquisite scheduling policy for MCMS with the following goals: minimizing the delay of the service, and achieving impartial free time among the servers regardless of their service time. Although substantial scheduling schemes have jointly taken into account the delay and energy in various cloud environments, a key point assumed in most existing works is that average service time and request rate, are known for the system operator. Obviously, this is helpful for simplifying the underlying scheduling problem and constructing easy service models. Nevertheless, this assumption cannot be always satisfied in practical MCMS.

It should be noted that even though the operator utilizes a powerful prediction technology with a sufficient former data, the above two parameters, in particular for the mobile cloud, are unknown as well. As a result, it is meaningful to study the scheduling problem when the important parameters are unavailable. Specifically, we informally define this case as blind scheduling. Furthermore, since the multimedia services sometimes are real-time, the scheduling policies should be implemented online for practical uses. Essentially, blind online scheduling results in some critical difficulties:

• Designing the user routing according to the availability and ability of the servers determining the server assignment without knowing the service request and time

• Implementing the scheduling in a distributed way for online operation

These three problems are interacted with each other, and hence they should be resolved jointly do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or rather than separately. Our objective in this work is to jointly consider the above three problems by designing a total blind online scheduling algorithm for the general MCMS. Specifically, we consider that new users are routed to the server whose free time is the largest for achieving the energy efficiency. Then, the available servers are assigned based on a separated method to get the delay as small as possible. Moreover, the allocation can be set easily for the online implementation. The rest of this article is organized. We first overview the related work on scheduling in cloud environments then provide the system model and formulate the problem. We design afterwards a blind online scheduling scheme and show its main characteristics, and apply the proposed scheduling scheme to a cloud-based program recommendation system. Subsequently, we conduct numerical simulations to demonstrate the efficiency of the proposed scheme. Finally, we conclude the article with a summary.

RELATED WORK - To improve the resource utilization of the mobile cloud, various scheduling mechanisms have been developed in recent years. In general, they can be broadly divided into two categories:

• Implementing the scheduling with predictable cloud parameters

• Considering scheduling with unknown user request rate and server service rate.

Specifically, for the first one, the majority of the current literature is based on approximations of user request by complete trial and error or by manual settings. In particular, some studies use the servers' approximation technique to achieve the near optimal solution, focus on heterogeneous multimedia services with different QoS requests by separating the user request and service ability. In addition, variable approximate techniques are applied to the scheduling scheme with multiple goal functions to achieve the tradeoff between various performance metrics, e.g., transmission delay, power consumption, etc. Recently, optimal scheduling in heavy traffic regimes has received considerable attention.

Interestingly, the correlation among the request rate, the service time, and the energy consumption is so small and even ignorable in this case. As a result, these metrics can be treated independently and the joint algorithm can be designed individually [6]. In fact, the uncertainties of the request rate and service rate further complicate the scheduling problem. Specifically, first derives the relationship between the service request rate and the user waiting time. Subsequently a vast volume of papers concentrated on the performance analysis with suboptimal performance on the service time or the energy consumption. Note that the majority of existing blind scheduling schemes are based on the stochastic optimization, which typically undergoes tremendous computation complexity.

Therefore, these scheduling schemes can hardly be implemented online. Most recently, utilizes a stochastic model to predict the service request rate, but its complexity is high for the online operation. The study of heterogeneous users with different QoS requirements and unknown request rates is conducted in a sequence of papers, in which the underlying relationship between the above two parameters are clearly formulated and how the parameters impact the performance is well investigated

.

I will further extend these works by proposing a simple blind scheduling scheme for online operation, and at the same time, we inherit the properties of the previous work. It is important to note that although the current work is derived from, there are two distinguished differences:

• The scheduling algorithm in this work is much simpler, which facilitates the online operation in the mobile cloud, and

• The condition to achieve the asymptotical optimality is more critical in this work, but it can also be held in the heavy traffic regime.

MOBILE CLOUD MULTIMEDIA SERVICE ARCHITECTURE - In this section, we describe a general MCMS architecture based on a pervasive multimedia service framework. Inspired by the concept of the separated function, MCMS is represented by four parts. Specifically, mobile users request heterogeneous multimedia services to the mobile clouds via base stations, e.g., access point, transceiver station, etc., which construct the user

interfaces between the mobile cloud and the users. Mobile users' state information, e.g., ID and location are also sent to the process unit to decide whether to agree or decline the request. Note that in the architecture of MCMS, each multimedia service is encoded independently with the utility computing, visualization, and service-oriented link (e.g., heterogeneous multimedia applications, database servers, etc.). Basically, the process unit plays the role of accessing control, and it usually consists of eight elements: access mode, service provider information, user request information, unified interface, energy consumption, flow scheduler, resource allocation, and user state information.

In particular, the energy consumption, flow scheduler, and resource allocation strategy can be integrated into a general scheduling middleware which can be embedded into the process unit. In this work, we will concentrate on this part by designing a simple scheduling scheme for online operation. After that, the service requests are delivered to virtual machines which act as the bridge between the service requests and available resource. In particular, the virtual machines first find appropriate servers which can provide the corresponding multimedia services, then assign the available server with a Determining the server assignment without knowing the service request and time .These three problems are interacted with each other, and hence they should be resolved jointly rather than separately. Our objective in this work is to jointly consider the above three problems by designing a total blind online scheduling algorithm for the general MCMS. Specifically, we consider that new users are routed to the server whose free time is the largest for achieving the energy efficiency. Then, the available servers are assigned based on a separated method to get the delay as small as possible. Moreover, the allocation can be set easily for the online implementation. The rest of this article is organized as follows. We first overview the related work on scheduling in cloud environments we then provide the system model and formulate the problem.

BLIND SCHEDULING STRATEGY - Based on the above discussion, I can design a simple blind online scheduling scheme by jointly considering routing and assignment but separately developing them. Specifically, we assign available multimedia servers based on the last time-slot information of the users' requests, and route the heterogeneous multimedia flows according to the first-come-first served rule. We propose a blind online scheduling algorithm (BOSA) outlined in Table 1. It is obvious that the proposed blind scheduling algorithm is suitable for practical use in the sense that it does not require the knowledge of the system parameters such as the request rates, service rates, etc. Instead, at each time-slot, one only needs to calculate s* and u* with computational complexity O(1) which can be implemented online in a distributed way. The separated approach is attractive because the computation of s* and u* that are required for a reasonable approximation grows only linearly with the dimension of users. Moreover, the results can be used to derive estimates on the system parameters with a desired level of accuracy, and this point is consistent with our probability-constrained formulation.

Next, I employ the above scheduling scheme in a practical mobile cloud environment. In general, there are two distinct types: one is to add the algorithm in the virtualization part which attains an approximate solution of the goal function within the cloud, and the other one is to implement it in the task division and mapping part which act as a middleware between the cloud virtualization and multimedia integration. For the former, it focuses on the cloud resource, providing QoS in the cloud infrastructure to support heterogeneous multimedia services with unknown user demand and available server. For the latter, it aims at improving cloud QoS in the upper layers, such as QoS in the application layer and QoS mapping between the cloud infrastructure and the QoS request. For ease of implementation, we employ the first one in our simulation. Note that, in fact, the results of the second one are able to achieve the same performance as that of the first type.

In the framework of multimedia services in the mobile cloud, content-based program recommendation techniques are used widely. There are lots of works on designing diverse program recommendation algorithms for different goals. Most notably, designs a content recommendation system (CPRS) for multimedia service platforms to drop the cloud's workload. Specifically, the proposed CPRS structure can consist of two parts. The first one is the digital television client (DTC), which is used by all users to obtain a multimedia service. If the DTC is new, the service fashions do not exist in cloud-based rating sharing servers (CRSS), which can be viewed as the task division and mapping shown in Fig. 2. After DTC is used to watch the programs, CRSS automatically records the program information which is selected by the DTC. The second part focuses on the criteria of CRSS by operating the map programming framework.

Note that multiple servers are connected as an independent cloud. In this section, we apply BOSA to CPRS based on the map-reduce algorithm. Typically, there are several steps in the map reduce algorithm: starting, recording, setting order, and recommendation.

More precisely, in the starting step, m (m > 1) services are randomly selected as initial multimedia service samples which are stored in the virtualization. The next step is recording, which can be further divided into two parts: computation and record. In the computation part, distance calculation at each server is implemented. In particular, the average distance between the current server and m services is calculated. The service with the largest distance can be given the highest score, and the service with the smallest distance is given the lowest score. In the third step, the scores for each service are collected and ordered. Usually, the highest score has the highest order.

Moreover, there are also two parts in this step:

Researching and connect. The first step aims at researching the highest order in existing multimedia service and the second step is to connect the service to the most appropriate service. More precisely, the server ID, service ID are linked together in this step. In addition, another task of the connecting part is to set the values of the priority and assign the most appropriate server based on BOSA. The final step is to make a recommendation to the service with the highest priority. Next, we introduce how to include the map reduce algorithm into BOSA. Note that the advanced hierarchy procedure (AHP) is one of the most popular methods for solving problems related to content recommendation. AHP is a complex goal decision making approach that simplifies NP-hard problems by arranging the decision factors in a specific order structure. There are mainly three steps:

• Using AHP to get the highest service order in each server;

• Among the service with the high order in each server, comparing the free time of their connecting servers. The service in the highest server's free time gets the highest order in the whole system;

• Re-map the connection between the service and server only for the service with the lowest order, and thus reduce the whole computation complexity.

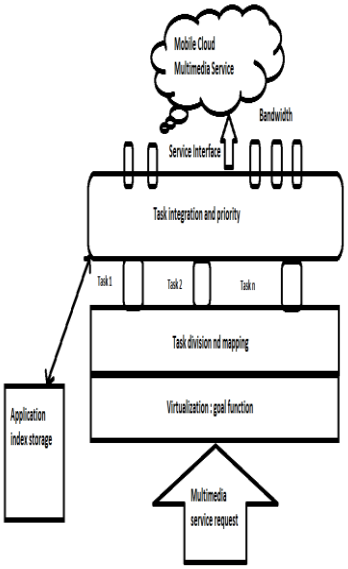

Fig1 .Illustration of the proposed blind online scheduling scheme in a mobile cloud

APPLICATION -In the framework of multimedia services in the mobile cloud, content-based program recommendation techniques are used widely. There are lots of works on designing diverse program recommendation algorithms for different goals. Most notably, [10] designs a content recommendation system (CPRS) for multimedia service platforms to drop the cloud's workload. Specifically, the proposed CPRS structure can consist of two parts. The first one is the digital television client (DTC), which is used by all users to obtain a multimedia service. If the DTC is new, the service fashions do not exist in cloud-based rating sharing servers (CRSS), which can be viewed as the task division and mapping. After DTC is used to watch the programs, CRSS automatically records the program information which is selected by the DTC. Theples which are stored in the virtualization part, the next step is recording, which can be further divided into two parts: computation and record. In the computation part, distance calculation at each server is implemented. In particular, the average distance between the current server and m services is calculated. The service with the largest distance can be given the highest score, and the service with the smallest distance is given the lowest score. In the third step, the scores for each service are collected and ordered. Usually, the highest score has the highest order.

Moreover, there are also two parts in this step: researching and connect. The first step aims at researching the highest order in existing multimedia service and the second step is to connect the service

to the most appropriate service. More precisely, the server ID, service ID are linked together in this step. In addition, another task of the connecting part is to set the values of the priority and assign the most appropriate server based on BOSA. The final step is to make a recommendation to the service with the highest priority. Next, we introduce how to include the mapreduce algorithm into BOSA. Note that the advanced hierarchy procedure (AHP) is one of the most popular methods for solving problems related to content recommendation. AHP is a complex goal decision making approach that simplifies NP-hard problems by arranging the decision factors in a specific order structure.

There are mainly three steps:
• Using AHP to get the highest service order in each server;
• Among the service with the high order in each server, comparing the free time of their connecting servers. The service in the highest server's free time gets the highest order in the whole system;
• Re-map the connection between the service and server only for the service with the lowest order, and thus reduce the whole computation complexity.

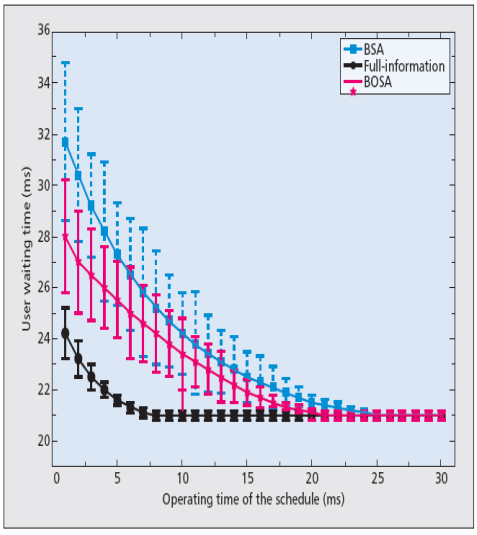

Fig2. The performance comparison in terms of user waiting time

Therefore, we can progressively use AHP to embed the map-reduce algorithm into BOSA, which is shown in Fig.2. second part focuses on the criteria of CRSS by operating the map programming framework. Note that multiple servers are connected as an independent cloud [10]. In this section, we apply BOSA to CPRS based on the map-reduce algorithm.. Typically, there are several steps in the mapreduce algorithm: starting, recording, setting order, and recommendation. More precisely, in the starting step, m (m > 1) services are randomly selected as initial multimedia service sam-ples which are stored in the virtualization part. The next step is recording, which can be further divided into two parts: computation and
record. In the computation part, distance calculation at each server is implemented. In particular, the average distance between the current server and m services is calculated. The service with the largest distance can be given the highest score, and the service with the smallest distance is given the lowest score. In the third step, the scores for each service are collected and ordered. Usually, the highest score has the highest order. Moreover, there are also two parts in this step: researching and connect. The first step aims at researching the highest order in existing multimedia service and the second step is to connect the service to the most appropriate service.
More precisely, the server ID, service ID are linked together in this step. In addition, another task of the connecting part is to set the values of the priority and assign the most appropriate server based on BOSA. The final step is to make a recommendation to the service with the highest priority.

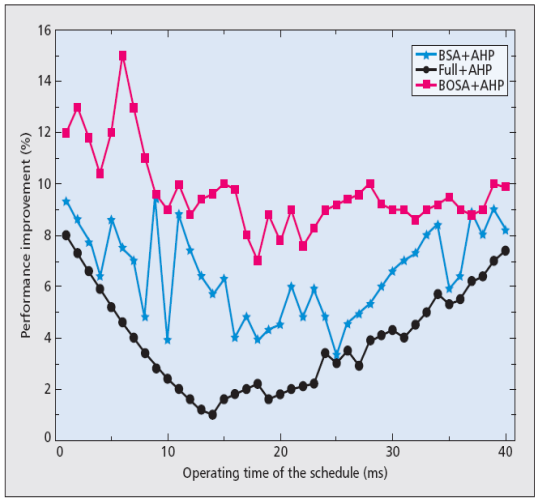

Fig3. Performance improvement when employing the content-based program recommendation technique

Next, we introduce how to include the mapreduce algorithm into BOSA. Note that the advanced hierarchy procedure (AHP) is one of the most popular methods for solving problems related to content recommendation. AHP is a complex goal decision making approach that simplifies NP-hard problems by arranging the decision factors in a specific order structure.
There are mainly three steps:

• Using AHP to get the highest service order in each server;

• Among the service with the high order in each server, comparing the free time of their connecting servers. The service in the highest server's free time gets the highest order in the whole system;
• Re-map the connection between the service and server only for the service with the lowest order, and thus reduce the whole computation complexity.

Therefore, we can progressively use AHP to embed the map-reduce algorithm into BOSA, which is shown. NUMERICAL RESULTS In this section, we evaluate the performance of the blind online scheduling algorithm by conducting simulation experiments in a practical environment, which consist of three kinds of users and servers: audio, file, and video (U = S = 3). We compare our blind online scheme with a full-information case and the blind scheduling algorithm (BSA). For the full-information case, the scheduler knows the user request rate and server service time. For comparison purposes, we set N = 90, Ns = 30, fs = 1/S for s OE [1, S], and h = 90 percent. In order to capture the heavy traffic regime, we set the average request rate for each user to 9000/second. We first test the performance of the BOSA. Figure 4 shows the performance of our proposed BOSA versus those of full-information case and
BSA. From the given results, we can observe that when the waiting time goes large, the performance gap between BOSA and full-information is negligible. That is to say, BOSA achieves asymptotically optimal in the heavy traffic regime, which is consistent with our theoretical analysis. Moreover, Fig. 4 also verifies that when the system traffic belongs to a heavy traffic regime, the processes of routing and assignment can be separately designed. In addition, from the given results, it is clear that the performance of BOSA is better than that of BSA. That is because the computational complexity of BOSA is much lower than that of BSA, and the scheduling scheme costs fewer handling time. As a result, the average waiting time of the user is lower. Next, we examine the performance improvement when employing the content-based program recommendation technique. We denote the three scheduling schemes by "Full+AHP", "BSA+AHP", and "BOSA+AHP" respectively exhibits the performance of each scheme when the operating time of the scheduling varies from 1 to 40 ms. Obviously, "BOSA+AHP" achieves the best performance.
That is because AHP utilizes the complete comparison enabling the server to determine the trade-offs among criteria, and this mechanism is in accordance with the structure of the BSA and BOSA. That is, AHP can be embedded naturally and seamlessly in BSA and BOSA. Moreover, since BOSA just needs the information of Ys(t) and Wu(t) at each time slot, AHP can be implemented at each server as well. That is the reason why BOSA also outperforms BSA in terms of the user waiting time.

CONCLUSIONS - Cloud environment where the user waiting time and server service time are unknown. Our main contribution is to design a blind online scheduling scheme by jointly considering delay and energy among the servers. Specifically, we assign available multimedia servers based on the last time-slot information of the users' requests, and route the heterogeneous multimedia flows according to the first-come-first-served rule. Furthermore, we apply the blind scheduling scheme to a content recommendation system, and provide the detailed implementation steps. Extensive simulation results indicate that the proposed scheme can efficiently schedule heterogeneous multimedia flows to satisfy dynamic QoS requirements in a practical mobile cloud.

REFERENCES

[1] G. Q. Hu, W. P. Tay, and Y. G. Wen, "Cloud Robotics: Architecture, Challenges and Applications," IEEE Network, vol. 26, no. 3, 2012, pp. 21–28.
[2] Y. G. Wen, W. W. Zhang, and H. Y. Luo, "Energy-Optimal Mobile Application Execution: Taming Resource- Poor Mobile Devices with Cloud Clones," Proc. IEEE INFOCOM 2012.
[3] M. Chen, S. Gonzalez, and V. Leung, "Applications and Design Issues of Mobile Agents in Wireless Sensor Networks", IEEE Wireless Commun. Mag., vol. 14, no. 6, 2007, pp. 20–26.
[4] M. Chen et al., "Software Agent-based Intelligence for Code-Centric RFID Systems", IEEE Intelligent Systems, vol. 25, no. 2, 2010, pp. 12–19.
[5] W. Zhu et al., "Multimedia Cloud Computing," IEEE Signal Proc. Mag., vol. 28, no. 3, 2011, pp. 59–69.
[6] L. Zhou and H. Wang, "Toward Blind Scheduling in Mobile Media Cloud: Fairness, Simplicity, and Asymptotic Optimality," to appear in IEEE Trans. Multimedia.
[7] J. Rodrigues, L. Zhou, L. Mendes, K. Lin, and J. Lloret, "Distributed Media-Aware Flow Scheduling in Cloud Computing Environment", Computer Communications,