ADVANCE MAP REDUCE TASK SCHEDULING ALGORITHM USING MOBILE CLOUD MULTIMEDIA SERVICES ARCHITECTURE

S.Saravanan, Research Scholar
M.Kumarasamy College of Engg-Karur
jeyasaraa@gmail.com(9944059288)

Dr.V.Venkatachalam, Principal
The Kavery Engineering College-Mecheri.
vv01062007@hotmail.com

ABSTRACT - There is a massive improvement in the computer technology which leads to infinite number of resources in all over the world stored in the computer. Computing devices have several uses and are necessary for businesses, scientists, governments, engineers. These have to generate data that comes from anywhere. Sensors gathering climate data, a person posting to a social media site, or a cell phone. Furthermore, the data itself may be too large to store on a single machine. In order to reduce the time it takes to process the data, and to have the storage space to store the data, we introduce a technique called map reduce programming model. In this programming method, it has to divide the workload among computers in a network. As a result, the performance of Map Reduce strongly depends on how evenly it distributes this workload among the computer. In Map Reduce, workload distribution depends on the algorithm that partitions the data. To avoid the problems of uneven distribution of data we use data sampling. By using the partitioning mechanism, the partitioned distributes the data depends on how large and representative the sample is and on how well the samples are analyzed. Due to this improves load balancing and memory consumption of the computers. In addition to that we use micro-partitioning methods to divide the workload into small tasks that are dynamically scheduled at runtime. This approach is only effective in systems with high-throughput, low-latency task schedulers and efficient data materialization. To enhance the accuracy in scheduling we propose an innovative method called Map Reduce Task Scheduling algorithm for Deadline constraints. This method allows user specify a job's deadline and tries to make the job be finished before the deadline. Through measuring the node's computing capacity, a node classification algorithm is proposed in MTSD. This algorithm classifies the nodes into several levels in heterogeneous clusters. Under this algorithm, we firstly illuminate a novel data distribution model which distributes data according to the node's capacity level respectively.
Keywords: MTSD, Map Reduce,MCMS.

INTRODUCTION

Currently, a large number of mobile cloud platforms have been proposed to provide a pervasive computing environment [1, 2, 5, 8]. In most of them, each mobile device is linked with a sys-tem-level service in the cloud infrastructure. Moreover, with the rapid development of wire-less communication technologies, users are expected to use more multimedia services in the mobile cloud to avoid the installation of the soft-ware 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 [1-5].

Basically, since various service requests usual-ly 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 [6]. 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 ser-vice, 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 [6]. 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. Further-more, 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 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. First overview the related works on scheduling in cloud environments. I then provide the sys-tem model and formulate the problem. I 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, 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 [1–3]; [4–5] 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, [7] 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 ser-vice 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 [6]. Most recently, [8] 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 [9–11]. In which the underlying relationship between the above two parameters are clearly formulated and how the parameters impact the performance is well investigated. In this work, we 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 [6].

It is important to note that although the cur-rent work is derived from [6], 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. Details of this will be discussed later.

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 which are shown in Fig. 1. Specifi-cally, 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 [7]. 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 multi-media 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 middle-ware 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 ser-vices, then assign the available server with a specific goal function (e.g., the minimum energy cost, the minimum waiting time, etc.).

Based on the above analysis, the architecture of MCMS is a classical large-scale distributed system. From the function perspective, the architecture of MCMS can also be divided into four layers:

Server Maintain (SM) Layer: This layer provides the hardware facility and infrastructure, in which multiple

ervers are connected with hybrid networks to provide multimedia services. Typically, these servers are placed in deal locations with plentiful of power supply and low venture of breakdown.
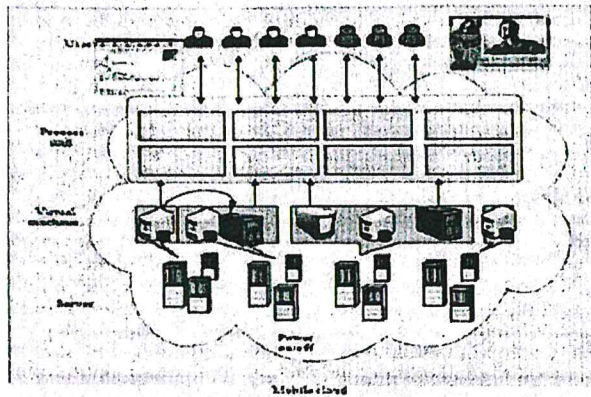

Figure 1. Multimedia transfer architecture in mobile cloud

Data Mapping (DM) Layer: The mechanism of data mapping is built on top of the SM layer. DM acts as the connection between the physical server and QoS requests, and it involves the pro-vision of servers, content update, and networking components. The users should pay for the ser-vice time and service quality, and the DM layer can considerably reduce the waiting time among the huge number of servers. Thus, users can save cost as the payment strongly depends on how much the server has been utilized. In particular, in the DM layer, the mapping fashion can be expanded or reduced dynamically according to the practical need [1, 5].

Virtual Machine (VM) Layer: VM performs as a service platform which offers an advanced integrated environment for building, testing, and deploying user services. Currently, the VM techniques are quite mature, and almost all the clouds employ it to improve the QoS [2].

Access Control (AC) Layer: In this layer, users can request multimedia services and access the mobile cloud via the different kinds of net-works, and they should pay for this access. For example, Microsoft's Live Mesh also allows shar-ing multimedia services across numerous mobile devices simultaneously.

Furthermore, combing the architecture of MCMS with the above four function layers, it is obvious that the MCMS is an open and integrated service platform, which pr ides different interfaces for different multimedia ap, cations. In particular, MCMS has the following addition-al advantages compared to traditional mobile clouds:

Dynamic maintain: Dynamic on-demand multimedia service maintain is an efficient and flexible way for multimedia service providers and mobile users to run their applications in a dynamic environment without additional resource requests.

Scalability service: The deployment of mobile multimedia applications can be adapted and scaled to meet the unpredictable user QoS requests due to variable available resources. Multimedia application providers can easily involve an unexpected service without or with little extra resource usage.

Information sharing: Multimedia application providers can share the resources and costs, in the open architecture of MCMS, to support numerous services and large number of users.

Information integration: Service requests from a variety of users and available services from different multimedia application providers can be integrated easily in the framework of MCMS by employing dynamic maintenance, scalability service, and information integration [9].

BLIND ONLINE MULTIMEDIA SCHEDULING

In this section, we first briefly describe the sys-tem model. Next, we propose a blind scheduling scheme by fully making use of the blind system information and large-scale cloud in heavy traffic. Additionally, we apply our blind scheduling scheme to a cloud-based program recommendation system.

SYSTEM MODEL

We consider a classical mobile cloud environment. Specifically, $U = \{1, ..., U\}$ represents user classes and $S = \{1, ..., S\}$ denotes server classes. Users request multimedia services according to an independent Poisson processes P_u, $u \square U$. Assume that service time of each server satisfies an exponential function, and the server from the same class s S has the same average service time. Moreover, let be a spe-cific scheduling scheme and $t \geq 0$ be a time slot, and W_u $(t; \ )$ denotes the waiting time of class u U user that has the longest waiting time [6].

By jointly considering waiting-time and energy consumption, an efficient scheduling should have both low delay for users and an impartial task division among the servers. To that end, we introduce Y_s $(t; \ )$ to show the minimum free time of server class s, and define an average factor

$$f_s = \frac{Y_s(t;\square)}{\Sigma_x Y_s(t; \ )}$$

Note that the average factor defined in this work is different from that of [6] in which an average free time of server is used. There are two main reasons that we replace the previous one:

- The average value is difficult to obtain in a dynamic environment
- The average value sometimes cannot guarantee the server's fairness

Let $T > 0$ be a given time period, which can be viewed as the service time of a specific multimedia application, e.g. $T = 90$min for a football match. As we know, the average factor can be set directly with perfectly predictable system parameters, however, for blind scheduling, the average factor itself is an unknown variable depending on the realization of the request rates. To circumvent this difficulty, we employ a probability-constrained formulation in which the specific constraint is set by some pre-specified fraction of the parameters. Usually, the constrained probability is artificially set by the sys-tem operator, who determines the system perfor-mance on the tradeoff between the system performance and complexity.

Essentially, the outline of the probability-con-strained formulation is as follows: the system operator first subjectively chooses a venture level η, which represents the maximum violated value of the average factor. As we know, the advantage of the probability-constrained formulation lies in elastically adapting to the venture-level that the scheduler

can tolerate. In general, blind scheduling aims to solve the following stochastic probability optimization problem, where η (0.1) is the venture level, $P_u(T)$ shows the recorded user request over time period T, and $W_{u,k}$ represents the waiting time up to T by the kth class u that has requested the service at the beginning of time.

In fact, the core problem of resolving the goal function is how to control the match error given a tolerable limit from a long-time scale, which can yield the desired asymptotical optimality.

$$\min_{\Upsilon} \sum_{u=1}^{U} \sum_{k=1}^{P_u(T)} \frac{W_{u,k}(T)}{P_u(T)}$$

$$s.t. \ \mathbb{P}\left(\frac{\Upsilon_s(t,\tau)}{\Sigma_{s'}\Upsilon(t,\tau)}\right) \geq 1-\eta.$$

Indeed, this mission is possible in the heavy traffic regime in which the routing and assignment are independent of each other [10]. To be more specific, the routing algorithm first takes charge of the service time, then controls the practical waiting time for a user requesting a specific multimedia service, while the assignment algorithm is responsible for distributing the servers into different classes according to different multimedia service requests.

Moreover, we are interested in asymptotical optimality criteria in which determines the queue length and waiting time, rather than the system service state (e.g., service time). If the system service time is set as the asymptotical optimality criteria, then, from the mathematical perspective, any blind policy, e.g., first-come-first-served, etc., cannot resolve the goal function due to the definitions of P_u and $W_{u,k}$. Moreover, the formulation of the the goal function concentrates on an important relationship between the service time and average factor. In particular, each server has its own queue, and the decision on which user should be allocated to a queue is made as soon as the user arrival. In this regard, this strategy implies that a well-defined separated policy is a possible solution because in this case the sys-tem operation area belongs to the heavy traffic regime. It is important to note that the asymptotically optimal allocation should grow linearly as the increase of the user request rate. This based on the following two considerations:
• Linear function is easy to calculate in an online fashion
• Linear function can well fit the user request rate in a heavy traffic [4]

Our previous work [6] has demonstrated that the upper and lower allocation bounds can be used to achieve an asymptotic optimality. In this work, we extend [6] to an online implementation. To this end, one should calculate the allocation values in a very simple way.

BLIND SCHEDULING STRATEGY

Based on the above discussion, we 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 multi-media 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, we employ the above scheduling scheme in a practical mobile cloud environment, which is shown in Fig. 2. 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 middle-ware between the cloud virtualization and multi-media integration. For the former, it focuses on the cloud resource, providing QoS in the cloud infrastructure to support heterogeneous multi-media 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.

NUMERICAL RESULTS

In this section, we evaluate the performance of the blind online scheduling algorithm by con-ducting 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) in [6]. For the full-information case, the scheduler knows the user request rate and server service time. For comparison purposes, we set $N = 90$, $N_s = 30$, $f_s = 1/S$ for s [1, S], and $\eta = 90$ percent. In order to capture the heavy traffic regime, we set the average request rate for each user to 9000/second.
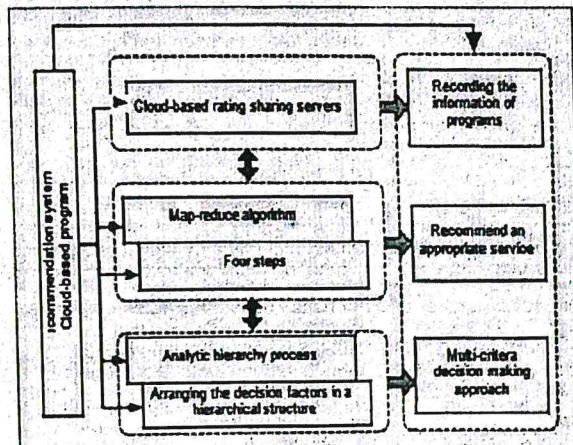

Figure 3. The mechanism of including the map-reduce algorithm into BOSA.

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.
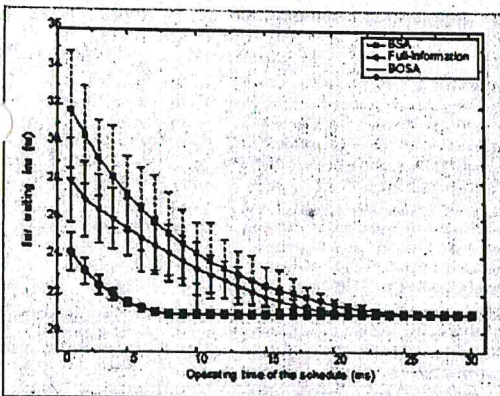

Figure 4. *The performance comparison in terms of user waiting time.*

Next, we examine the performance improve-ment when employing the content-based pro-gram recommendation technique. We denote the three scheduling schemes by "Full+AHP", "BSA+AHP", and "BOSA+AHP" respectively. Figure 5 exhibits the performance of each scheme when the
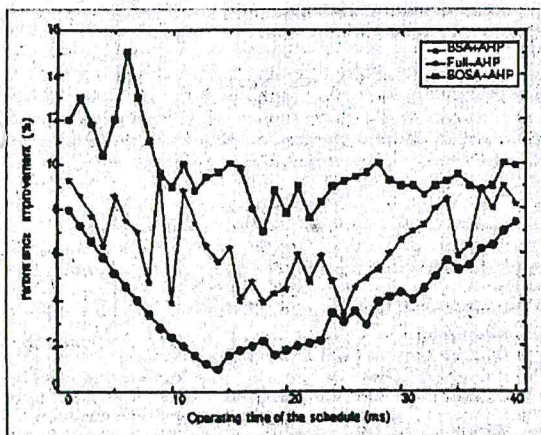

Figure 5. *Performance improvement when employing the content-based pro-gram recommendation technique.*

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 $Y_s(t)$ and $W_d(t)$ 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

In this article, we consider a practical mobile 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 pro-vide 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] A. Iosup et al.,"Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," IEEE Trans. Parallel and Distributed Systems., vol. 22, no. 6, 2011, pp. 931–45.

[2] 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.

[3] J. Rodrigues, L. Zhou, L. Mendes, K. Lin, and J. Lloret, "Distributed Media-Aware Flow Scheduling in Cloud Computing Environment", Computer Communications, vol. 35, no. 1, 2012, pp. 1819–27.

[4] L. Zhou and H. Wang, "Toward Blind Scheduling in Mobile Media Cloud: Fairness, Simplicity, and Asymptotic Optimality," to appear in IEEE Trans. Multimedia

[5] 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.

[6] M. Chen et al., "Software Agent-based Intelligence for Code-Centric RFID Systems", IEEE Intelligent Systems, vol. 25, no. 2, 2010, pp. 12–19.

[7] S. Meng, L. Liu, and T. Wang, "State Monitoring in Cloud Datacenters," IEEE Trans. Knowledge and Data Engineering, vol. 23, no. 9, pp. 1328–44, 2011.

[8] W. Zhu et al., "Multimedia Cloud Computing," IEEE Signal Proc. Mag., vol. 28, no. 3, 2011, pp. 59–69.

[9] 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.