Dynamic Resource Allocation of Smart Home Workloads in the Cloud

Shahin Vakilinia
ETS
Synchromedia Laboratory
Montreal, Canada

Mohamed Cheriet
ETS
Montreal, Canada

Jananjoy Rajkumar
ETS
Montreal, Canada

***Abstract*—Cloud computing offers provision for elastic and scalable infrastructure resource allocation across the network that allows deployment of services for controlling home devices and appliances. Data generated from heterogeneous smart home devices are processed in different application services deployed in the cloud data center. The primary challenge of smart home service provider's is to optimize the cloud resource allocation while satisfying the Quality of Service(QoS) constraints of the application services. Service execution time is one of the most vital QoS parameters. In this paper, a queuing theoretic approach is proposed to model the smart home workload. First, $M/M/c$ queue model is applied to find the response time of smart home tasks with light variation over the arrival rate. Then, Markovian Modulated Poisson Process (MMPP) is used to extend the model to a more advanced type of smart home processing workloads. Next, the optimal number of Virtual Machines(VMs) required deploying the application servers that can satisfy the execution time constraint of incoming workloads is calculated. Finally, total service time of a smart home application is calculated considering into account the possible level of concurrency and dependency among tasks of an application service. In the end, some numerical and simulation examples are provided to validate our findings.**

I. Introduction

The utilization of cloud computing technologies in the smart home system provides an opportunity for scalable and dynamic resource allocation in controlling the home devices and appliances without upfront capital investment. Technological advancement had pushed forward to migrate most of the application to the cloud including smart home application servers to facilitate the benefits of the infrastructure resource management. Moreover, home users can access the cloud services through Internet for controlling and monitoring the connected home devices using smart-phone or display panel. Fig. 1. shows the cloud-based smart home architecture. As depicted, data generated from heterogeneous home devices are sent to the cloud and are handled in various application servers in the form of Virtual Machines (VMs). Service applications run on the cloud which takes the results of the processed data of devices over multiple VMs and provide a service for smart home scenarios. This paper mainly focuses on tackling how dynamically allocate the minimum number of VM resources for controlling home devices satisfying QoS constraints of the application services.The execution time delay is considered as the main QoS factor. Thus, the main problem addressed here is to find the optimal number of VMs to satisfy the execution time constraints of applications running on the cloud. Due to the dynamicity, heterogeneity and rapid variation of the smart home workload in the scale of the smart city, stochastic process analysis should be taken into the account, and the main methodology of this paper goes through the analysis of the randomness in some generated tasks and processing time. In this paper, processing workload of the Smart home scenarios is initially modeled at the cloud computing data center by a Poisson process. Then, using $M/M/C$ queues the response time of the task execution time is calculated which used in determining the minimum number of application servers in the form of VMs. Next to address time variation and heterogeneity of the smart residence workload, Markovian Modulated Poisson Process (MMPP) is applied and the same analysis for the response time is applied to find the minimum number of the VMs required to serve the tasks while satisfying the execution constraint. After finding the execution delay of each task of the smart home device, the final contribution of this paper is to estimate the execution time of an application service as a collection of sequential and parallel tasks. Hence, according to the dependency factor among the tasks of an application service, the service time of application service is approximated.

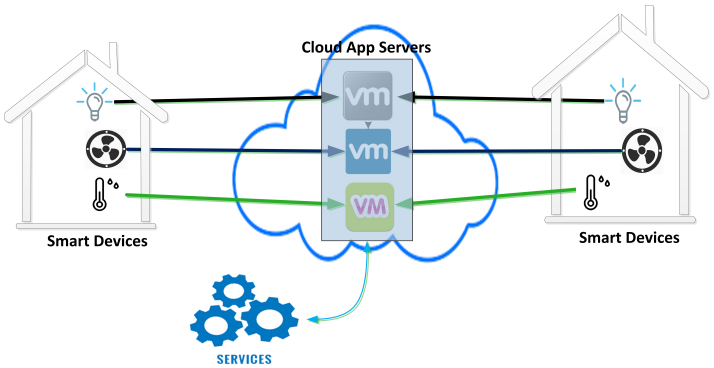

Fig. 1. Cloud Based Smart Home Architecture for controlling Smart Devices

The remainder of this paper is organized as follows. The related work is discussed in Sect.II. In Sect. III, the queuing models, and associated analysis to find the number of VMs in the cloud are described. Next, Sect. IV investigate the analytical model for the application service time. Finally, in

978-3-901882-85-2

Section V, we conclude the paper.

II. Related Work

The vast study has been done on dynamic resource allocation of service deployment in cloud computing systems [21], [**?**], [**?**]. In this Section, some of the relevant research studies are briefly discussed. Similar to our approach, queuing theory is applied to analyze the dynamic resource allocation in cloud computing systems [1]-[16]. For instance, [1] used queuing theory to formulate the revenue maximization of cloud system with QoS constraint over blocking probability of incoming jobs. [2] based on the use of analytic queuing network models combined with combinatorial search techniques defined the cost function to be optimized as a weighted average of the deviations of response time, throughput, and the probability of rejection metrics about their Service Level Agreements(SLAs). [16] also taking queuing theoretical approach proposed an analytical model, based on stochastic reward nets, cloud performance metrics such as utilization, availability, waiting time, and responsiveness is defined and evaluated to analyze the behavior of a cloud data center: A resiliency analysis is also provided to take into account load bursts. Despite all similarities between the literature and this paper, there are main distinctions: First, the objective are different. In this paper, we focused more on the minimization of resources while assuring the application service response time. The objective of this article is more similar to resource allocation strategies proposed in [5] and [6] that aim to minimize the mean response time and the number of servers requested by multiple users. Moreover, due to the heterogeneity and dynamicity of smart home scenarios, an advanced model for workloads is required. MMPP and Batch Markovian Arrival process (BMAP) [15] are the most general models for arrival rate which can cover heterogeneous incoming workloads. Owing to the dynamic variation of the workload over the time, MMPP is selected.
On the other hand, almost all academic discussions related to the smart home scenarios have focused on technological architecture development and [17]. At this stage, researchers address limitations to the functionality of management of applications over home devices connected to the cloud servers. However, the execution time of the application services is one of the main challenges in developing smart home applications [18]. For instance, in [19] and [20], a platform is proposed to schedule smart home applications under execution time and cost constraints costs associated with the cloud service providers. Similar to their approach, an efficient execution time of smart home applications on the cloud platform with different methodology and terminology is addressed in this paper. However, instead of cost, resources in terms of the number of VMs are minimized.

III. Modeling

In this section, first notations and scenario details are introduced, then queuing theoretical approaches will be used to provide the model for the task execution time.

Preliminaries and Notations: We assume that there are R different types of VMs on the cloud serving devices. The data gathered by devices are processed in VMs. Each type of VMs is responsible for one or more homological series of devices which may be located in and belong to different residences. It is also assumed that there are I service applications running on the cloud which takes the results of the processed data of devices over multiple VMs and provides a service for smart home scenarios. Thus, an application service is provided through a collection of tasks running on different VMs. Data events of each type of devices are generated according to a Poisson process with a different parameter. λ_r represents the arrival rate of the type r devices. Each task created by a device requires a VM for its execution. It is assumed that the processing time of the data attributed to each type of device is also different. Service rate of type r VM is represented by μ_r. It is assumed that there are b_r VMs in the cloud to serve the tasks generated by type r devices. SLA outlines all aspects of application service and the obligations of QoS. However, SLA response time of each application service could be considered as the most dominant factor to allocate the cloud resources. T_r^{th} denotes the execution time constraint of type r tasks and D_i^{th} indicates the service execution delay constraint of i^{th} application.

Task Response Time: Using, $M/M/c$ queuing model where $c = b_r$, the average response time of type r tasks related to the type r devices represented by t_r is given by [12],

$$t_r = \frac{1}{1 + (1-\rho)(\frac{b_r!}{(b_r\rho)^{b_r}})(\sum_{k=0}^{b_r-1}\frac{k!}{(b_r\rho)^k})} + \frac{1}{\mu_r} \qquad (1)$$

Note that, b_r should be selected with respect to the QoS response time constraint such that it has to be less than its threshold, $t_r \preceq T_r^{th}$. So, b_r should mapped in to λ_r and by changing the arrival rate b_r should be dynamically modified. Thus, $b_r^* = Min(b_r)|t_r \preceq T_r^{th}$. Despite the fact that the Poisson process can approximate the processing workload of sensors, this assumption can not be held up for all different types of the smart home devices. Therefore, we extend our analysis with Poisson process to MMPP which is a straightforward extension of Poisson process and has already been widely used. It may cover and fit the vast domain of arrival rates of devices.
An MMPP is a Poisson process whose instantaneous rate itself is a stationary random process with varying arrival rate according to an irreducible *M* state Markov chain. It is characterized by the state transition probability Q_r of the underlying Markov process among M states. The detailed description of the *MMPP* with an emphasis on applicability to modeling is given in [9], [10], [11]. $\lambda_{r,m}$ denotes the task arrival rate of the device type r at the *m* state. In an incremental arrival, the underlying Markov chain goes from state i to state j with probability $\sigma_{r,ij}$. k_r also represents the maximum allowable type r tasks of can be existed in the queue so that extra tasks will be dropped. Let us define the diagonal arrival

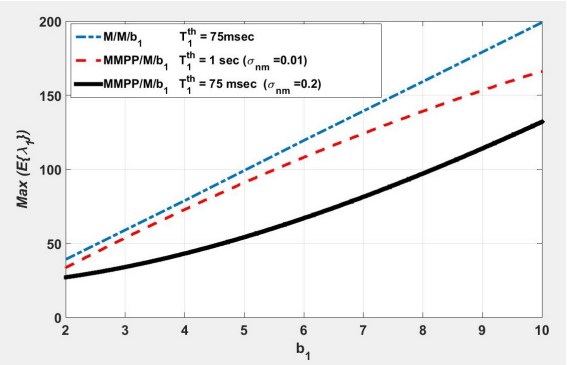

Fig. 2. Numerical results for maximum of achievable average arrival rate of device tasks as a function of number of virtualized application servers and execution time SLA (T_1^{th}) and transition rate σ as the parameter

rate matrix and state transition probability matrix as follows,

$$A_r = \begin{bmatrix} \lambda_{r,1} & 0 & 0 & \dots & 0 \\ 0 & \lambda_{r,2} & 0 & \dots & 0 \\ & \dots\dots & \dots\dots & & \\ 0 & \dots & \lambda_{r,m} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & \lambda_{r,M} \end{bmatrix}$$

$$Q_r = \begin{bmatrix} \sigma_{r,11} & \sigma_{r,12} & \dots & \dots & \sigma_{r,1M} \\ \sigma_{r,21} & \sigma_{r,22} & \dots & \dots & \sigma_{r,2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{r,M1} & \sigma_{r,M2} & \dots & \dots & \sigma_{r,MM} \end{bmatrix}$$

Finding the probability distribution for the number of tasks of the type r device within the system requires the computation of $\pi_r = (\pi_{r,1}, \pi_{r,2}, \dots, \pi_{r,c}, \dots, \pi_{r,c+k_r})$ by solving $\Theta\pi_r = 0$. Even though Θ has a large number of states, π_r may be computed efficiently by using block Gauss-Seidel iteration [10]. Meantime, a normalization matrix e_r also has to be defined in the way of satisfying $\pi_r e_r = 1$. Then, the probability distribution of having i tasks of type r within the system in steady-state equals to,

$$p_{r,i} = \pi_{r,i} e_r \tag{2}$$

From littles theorem, average waiting time of the type r tasks in the system is given by,

$$t_r = \frac{\sum_{i=1}^{b_r+k_r} i p_{r,i} \left(\sum_{m=1}^{M} \sum_{n=1}^{M} \sigma_{r,nm} \right)}{\sum_{m=1}^{M} \lambda_r, m \sum_{n=1}^{M} \sigma_{r,nm}} + 1/\mu_r \tag{3}$$

Result of Eq. 3 is advantageous to calculate the tasks delay. For more details please refer [9], [10], [11]. We provide some numerical results to shed light on required resources for serving the QoS-sensitive applications. Fig. 2 compares the maximum task arrival rate that can be supported in different models for the different number of VMs represented by b_1. The MMPP model has 5 states in which states with arrival rates next to each other has 10% difference. As it is depicted, if the execution time constraint is so tight (75msec) and transitions among the MMPP states are high (highly dynamic demand with $\sigma_{nm} = 0.2$), the number of device tasks that can be supported are so limited. On the other hand, if the system becomes more static(less transition) the maximum average arrival rate will increase. Fig. 2 indicates how the analysis in task response time can be applied to dynamically vary the number of virtual machine application servers according to the characteristics of the incoming workload over the time.

IV. Application Service time

As it mentioned earlier, different connected home devices or sensors can execute a group of tasks collectively or collaboratively for providing a particular home application service. In real scenarios, some executing tasks have a sequential transitive dependency. Otherwise, executing tasks can be executed concurrently depending on the type of the services which diminishes the application service time. So, in the service deployment for the smart home scenarios, the parallelism ought to be maximized. Two scenarios namely Fully dependent and Fully independent are considered in this paper. It is convenient that the lower bound for the execution time belongs to the Fully independent scenario while the upper one belongs to fully dependent one. The service time of other application services with the same number of tasks will be between of these two scenarios. Thus, we calculate the service time for both scenarios and generalize it to all application services with similar tasks accordingly using the cross-correlation process.

A. Fully Dependent Scenario

In this scenario, the average service time of the application will be equal to sum of the average service time of tasks and is given by,

$$D_E = \sum_{i=1}^{I} \frac{1}{\mu_i} = \sum_{r=1}^{R} \frac{a_r}{\mu_r} \tag{4}$$

where the a_r denote the number of type r tasks in the application.

B. Fully independent

In Fully independent case, the application execution is not finished unless all tasks running in parallel get terminated. Under this circumstance, the application service time will be equal to the maximum service time of the parallel threads and can be written by,

$$D_F = max(t_1, t_2, ..., t_i, ..., t_I) \tag{5}$$

Assuming service times of I parallel tasks are exponentially distributed with different average values. The corresponding CDF of application service time is given by [14],

$$F(t) = \prod_{i=1}^{I} \left(1 - e^{-\mu_i t}\right) = \prod_{r=1}^{R} \left(1 - e^{-\mu_r t}\right)^{\alpha_r} \tag{6}$$

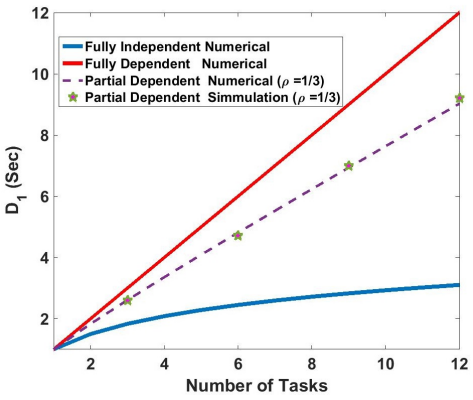

Fig. 3. Numerical and simulation results for partial dependent application service time as a function of number of tasks and ρ as the parameter

Then, the expected value of the service time is obtained by solving,

$$D_F = \int_0^\infty t f(t) dt = \int_0^\infty t \left[\frac{\partial}{\partial t} \prod_{i=1}^{I} \left(1 - e^{-\mu_i t}\right) \right] dt \quad (7)$$

Using integration by parts technique, the problem can be solved. However, to represent the solution, some definitions have to be made. Let us define the $\mathbb{N}_i$ as the collection of all tuples order j of the task service times of the application. For instance, $\mathbb{N}_1 = \{\mu_1, \mu_2, ..., \mu_i, \mu_j ..., \mu_{I-1}, \mu_I\}$ $\mathbb{N}_2 = \{(\mu_1, \mu_2), ..., (\mu_i, \mu_j), ..., (\mu_{I-1}, \mu_I)\}$ then solution can be represented by,

$$D_F = \frac{(-1)^{I-1}}{\sum_{i=1}^{I} \mu_i} + \sum_{j=1}^{I} \left(\sum_{\forall K_j \in \mathbb{N}_j} \frac{(-1)^{I-j}}{\sum_{\forall k \in \mathbb{N}_j, k \notin K_j} \mu_k} \right)$$

In this subsection, ρ_j is defined as the concurrency coefficient among the tasks of j^{th} application and can be approximated by the number of correlation links among tasks divided into $I-1$. Then, the average service time of the application can be approximated by,

$$D_j = \rho_j D_F + (1 - \rho_j) D_E \quad (8)$$

Eq.8 can be notably used to approximate the various application service times. The numerical results of the application service time related to fully independent, fully dependent and partial dependent scenarios are presented in Fig. 3. Moreover, a discrete event simulation method (similar to simulation in [14]) is applied to find the average service time for the partial dependent scenario with $\rho = 0.333$. As it is depicted, there is a close agreement between the simulation and numerical results obtained by Eq.8 which validates the analysis of this Section.

V. Conclusion

In this paper, the execution time of the application tasks of smart home scenarios is modeled. Next, the minimum number of VMs required to deploy for home application service ensuring QoS constraints over execution time is calculated. Finally, Considering the impact of concurrency and dependency among tasks of smart home devices, the execution time of the smart home application services is approximated.

References

[1] Feng, Guofu, and Rajkumar Buyya. "Maximum revenue-oriented resource allocation in cloud." International Journal of Grid and Utility Computing, vol 7, no.1, pp. 12-21,2016.

[2] Bennani, Mohamed N., and Daniel A. Menasce. "Resource allocation for autonomic data centers using analytic performance models." Proceedings Second International Conference in Autonomic Computing,IEEE, pp.229-240,2005.

[3] Chandra, Abhishek, Weibo Gong, and Prashant Shenoy."Dynamic resource allocation for shared data centers using online measurements." Quality of ServiceIWQoS 2003. Springer Berlin Heidelberg, pp.381-398,2003.

[4] Levy, Ron, Jay Nagarajarao, Giovanni Pacifici, Mike Spreitzer, Asser Tantawi, and Alaa Youssef. "Performance management for cluster based web services." Integrated Network Management VIII. Springer US, pp. 247-261,2003.

[5] Zhu, Huican, Hong Tang, and Tao Yang. "Demand-driven service differentiation in cluster-based network servers.", In proceeding of INFOCOM 2001. 20th Annual Joint Conference of the IEEE Computer and Communications Societies. , vol 2, pp. 679–688,2001.

[6] Li, Jiayin and Qiu, Meikang and Niu, Jian-Wei and Chen, Yu and Ming, Zhong. "Adaptive resource allocation for preemptable jobs in cloud systems." 10th International Conference on Intelligent Systems Design and Applications (ISDA), IEEE, pp. 31-36, 2010.

[7] Paridel, Koosha, Engineer Bainomugisha, Yves Vanrompay, Yolande Berbers, and Wolfgang De Meuter. "Middleware for the internet of things, design goals and challenges." Electronic Communications of the EASST, vol 28, 2010.

[8] Atzori, Luigi and Iera, Antonio and Morabito, Giacomo. "The internet of things: A survey." Computer networks,vol 54, no. 15, pp.2787-2805,2010.

[9] Meier-Hellstern, Kathleen S."The analysis of a queue arising in overflow models.", IEEE Transactions on Communications, vol 37, no. 4, pp.367-372,1989.

[10] Meier-Hellstern, Kathleen S. "Parcel overflows in queues with multiple inputs.",Proceedings of the 12th International Teletraffic Congress (ITC)(M. Bonatti, ed.), pp. 1359-1366. 1988.

[11] Hayes, Jeremiah. Modeling and analysis of computer communications networks,Springer Science & Business Media, 2013.

[12] Leonard Kleinrock. Queuing Systems, vol. I. John Wiley and Sons, 1976.

[13] S. K. Bose, "An Introduction to Queueing Systems", Kluwer Academic,The Rosen Publishing Group, Springer, 2002.

[14] S.Vakilinia, MM. Ali, and D. Qiu. "Modeling of the Resource Allocation in Cloud Computing Centers", Computer Networks, vol. 9, no. 1, pp. 453-470, 2015.

[15] Khazaei, Hamzeh, Jelena Misic, and Vojislav Misic."Performance of cloud centers with high degree of virtualization under batch task arrivals.",IEEE Transactions on Parallel and Distributed Systems, vol.24, no.12, pp.24292438, 2013.

[16] Bruneo, Dario."A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems",IEEE Transactions on Parallel and Distributed Systems, vol 25, no.3, pp.560-569,2014.

[17] R. Harper,*Inside the smart home*, 3^{rd} edition, Springer Science & Business Media; London, UK, 2006.

[18] Aazam, Mohammad, Imran Khan, Aymen Abdullah Alsaffar, and Eui-Nam Huh."Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved."In Applied Sciences and Technology (IBCAST), 11th International Bhurban Conference,pp. 414-419, 2014.

[19] Igarashi, Yuichi, Kaustubh Joshi, Matti Hiltunen, and Richard Schlichting."Vision: towards an extensible app ecosystem for home automation through cloud-offload.",Proceedings of the fifth international workshop on Mobile cloud computing & services,pp. 35-39, 2014.

[20] Igarashi, Yuichi, Matti Hiltunen, Kaustubh Joshi, and Richard Schlichting."An Extensible Home Automation Architecture based on Cloud Offloading",18th International Conference on Network-Based Information Systems (NBiS),(pp. 187-194, 2015.

[21] S.Vakilinia, D. Qiu, and MM. Ali. "Optimal multi-dimensional dynamic resource allocation in mobile cloud computing." EURASIP Journal on Wireless Communications and Networking,no. 1, pp.1-14, 2014.