Poster: Risk-based Optimization of Resource Provisioning in Mobile Edge Computing

Hossein Badri
Department of Industrial & Systems Eng.
Wayne State University
Detroit, USA
hossein.badri@wayne.edu

Tayebeh Bahreini
Department of Computer Science
Wayne State University
Detroit, USA
tayebeh.bahreini@wayne.edu

Daniel Grosu
Department of Computer Science
Wayne State University
Detroit, USA
dgrosu@wayne.edu

Kai Yang
Department of Industrial & Systems Eng.
Wayne State University
Detroit, USA
kai.yang@wayne.edu

I. Introduction

Resource provisioning is a challenging issue for Mobile Edge Computing (MEC) service providers significantly impacting the efficiency of the system and the Quality of Service (QoS). This is due to the existence of nondeterministic parameters which makes it difficult to manage the resources efficiently. Researches have addressed the resource provisioning problem in Cloud Computing (CC) considering the uncertainty of parameters. Maguluri et al. [3] introduced a stochastic model for load balancing and scheduling in CC clusters, where the arrival time and duration of jobs are stochastic. Chaisiri et al. [2] proposed a stochastic model for the cloud resource provisioning problem under uncertainty of resource prices and demands. Wang et al. [5] developed a model for mapping virtual machines into cloud servers assuming that the completion time of the requests of users is stochastic.

Compared to CC, resource provisioning in MEC is expected to be more challenging. First, resource requirements of mobile applications are unknown prior to running applications on servers, and second, edge servers have more restricted capacity than the cloud servers. In this paper, we propose a risked-based optimization approach to resource provisioning in MEC systems with the aim of taking into account the risk of overloading of edge servers when making allocation decisions. Assuming that resource requirements of mobile applications are stochastic parameters, we formulate the problem as a *chance-constrained stochastic program*. In order to solve the problem in reasonable amount of time, we employ the Sample Average Approximation (SAA) method [1]. We evaluate the efficiency of the proposed approach by conducting an experimental analysis on instances with different problem settings.

Our *contributions* are as follows: (i) We propose a risk-based optimization approach to resource provisioning problem in MEC; (ii) We propose a clustering-based approach to approximate the probability distributions of resource requirements of mobile applications; (iii) We propose the use of the SAA method to solve the chance-constrained stochastic program; and (iv) We provide a comprehensive analysis of the effects of the overloading risk factor on the utilization rates of servers and the QoS.

II. Risk-based Optimization Model

We consider a two-level (i.e., cloud and edge) MEC system, and denote the set of levels by $\mathcal{L}$, where $\ell \in \mathcal{L}$, and $\ell = 1$ represents the edge level, and $\ell = 2$, the cloud level. There are M^ℓ servers at each level. The set of servers at the edge level, the set of servers at the cloud level, and the set of all servers are denoted by $\mathcal{M}^1$, $\mathcal{M}^2$, and $\mathcal{M}$, respectively. These servers provide K types of computing resources to N independent requests from mobile applications. We denote the set of computing resources by $\mathcal{K}$, and the set of requests by $\mathcal{U}$. We assume that application i requires an $\tilde{R}_{ik}$ amount of resource of type k, which is a nondeterministic parameter. We assume that each edge server has an available capacity of C_{jk} for resource of type k. We also assume that cloud servers are uncapacitated. We denote the distance of user i from server j by d_{ij}. We formulate the risked-based resource provisioning in MEC systems as a chance-constrained stochastic program,

$$\text{Minimize} \sum_{i\in\mathcal{U}} \sum_{j\in\mathcal{M}} \sum_{\ell\in\mathcal{L}} \gamma \cdot d_{ij} \cdot x_{ij}^{\ell} \tag{1}$$

Subject to:

$$p\left\{\sum_{i\in\mathcal{U}} \tilde{R}_{ik} x_{ij}^{\ell} \le C_{jk}, \quad \forall j \in \mathcal{M}^1, k \in \mathcal{K}\right\} \ge (1-\alpha) \tag{2}$$

$$\sum_{j\in\mathcal{M}} \sum_{\ell\in\mathcal{L}} x_{ij}^{\ell} = 1 \quad \forall i \in \mathcal{U} \tag{3}$$

$$x_{ij}^{\ell} \in \{0,1\} \quad \forall i \in \mathcal{U}, j \in \mathcal{M}, \ell \in \mathcal{L} \tag{4}$$

(a) (b)

Fig. 1: (a) Average utilization ratio of edge servers, and (b) Quality of service vs. overloading risk factor

where, the objective function is to minimize the total communication cost between mobile applications and servers. We assume that the communication cost is proportional to the distance between users and the server that performs the request, considering a coefficient of proportionality, γ. In this formulation, x_{ij}^{ℓ} is a binary variable, that is 1 if the request of user i is allocated to server j at level ℓ, and, 0 otherwise. Constraint (2) ensures that overloading of an edge server does not occur more than as specified by the risk factor α. Constraint (3) ensures that each request is satisfied and is not allocated to more than one server. Finally, Constraint (4) guarantees the integrality of the decision variables. Chance-constrained stochastic programs are difficult to solve, due to the nonconvexity and feasibility checking issues. Here, we employ the SAA method to solve the chance-constrained program. Let us define, $G(x,\xi^s) = \sum_{i\in\mathcal{U}} \tilde{R}_{ik}^s x_{ij} - C_{jk}$ where, $\tilde{R}_{ik}^s$ is the realization of parameter $\tilde{R}_{ik}$ based on scenario s. Then, we can formulate the SAA problem as a mixed-integer program (MIP),

$$\text{Minimize} \sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{M}}\sum_{\ell\in\mathcal{L}} \gamma\cdot d_{ij}\cdot x_{ij}^{\ell} \tag{5}$$

Subject to:

$$G(x,\xi^s) \le W\cdot z_s \quad \forall j\in\mathcal{M}^1, k\in\mathcal{K}, s\in\Theta \tag{6}$$

$$\sum_{s\in\Theta} z_s \le \alpha\cdot S \tag{7}$$

$$\sum_{j\in\mathcal{M}}\sum_{\ell\in\mathcal{L}} x_{ij}^{\ell} = 1 \quad \forall i\in\mathcal{U} \tag{8}$$

$$x_{ij}^{\ell}\in\{0,1\} \quad \forall i\in\mathcal{U}, j\in\mathcal{M}, \ell\in\mathcal{L} \tag{9}$$

$$z_s\in\{0,1\} \quad \forall s\in\Theta \tag{10}$$

where the objective function is the same as is in the previous chance-constrained model. In Constraint (6), W is a very large positive number, and z_s is a binary variable. If z_s is 1, the capacity constraint can be violated under realization of scenario s, and if it is 0, otherwise. Also, Θ is the independent identically distributed (iid) sample of S realizations of $\tilde{R}_{ik}^s$. Constraint (7) guarantees that number of violated capacity constraints is not more than $\alpha\cdot S$. Constraints (8) and (9) were described in the chance-constrained model. Constraint (10) guarantees the integrality of z_s.

III. Experimental Analysis

In this section, we present our experimental analysis on the effects of the overloading risk factor on the performance of the MEC system when employing our risk-based optimization approach. We use the dataset provided by [4] on smartphones. The dataset used in our analysis consists of 156,017 records. We use two thirds of the records (randomly selected) for clustering and fitting probability distributions, and the remaining one third of the records for the analysis of our optimization approach. We apply the K-means clustering method to cluster applications based on the following features: (i) amount of data transmitted; (ii) number of packets received; (iii) total CPU utilization in percentage; and (iv) total memory used in the Android heap. In each cluster, we approximate the probability distribution parameters of the usage of two important resources, CPU and memory. In our analysis, we cluster applications into ten clusters. We consider a MEC system with five edge servers, and two servers at the cloud level. We perform our analysis by taking a sample of 100 applications in each run of our algorithm. We solve the MIP model using the CPLEX solver provided by IBM ILOG CPLEX optimization studio for academics initiative. We run each experiment five times and perform our analysis based on the average value of the metrics.

Figures 1a and 1b show the effects of the overloading risk factor, α, on the system performance when employing our risk-based optimization approach. We use two measures for the performance of the MEC system, the utilization ratio of edge servers, and the QoS. We use the actual resource usage of the applications to compute the utilization rate of each edge server. We define the QoS as, $QoS = \frac{\sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{M}^1}\bar{x}_{ij}^1}{N}$, where $\bar{x}_{ij}^1$ is the value of variable x_{ij}^1 taken from the MIP solution.

In Figure 1a, we observe that the average utilization ratio of both resources (i.e., CPU and memory) increases with the increase in the overloading risk factor. We also observe that edge servers have higher CPU utilization ratios than those of the memory. When a low risk factor is used, our approach makes the system operate at low resource utilization levels to avoid overloading. With higher levels of the risk factor, our approach considers a higher allowance to allocate more requests to edge servers. Figure 1b shows the effect of the overloading risk factor on the QoS. Similar to the utilization ratio, higher levels of the risk factor results in a higher QoS through allocating more requests at the edge level.

REFERENCES

[1] S. Ahmed and A. Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 261–269. INFORMS, 2008.

[2] S. Chaisiri, B.-S. Lee, and D. Niyato. Optimization of resource provisioning cost in cloud computing. *IEEE Trans. Services Comp.*, 5(2):164–177, 2012.

[3] S. T. Maguluri, R. Srikant, and L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *Proc. IEEE INFOCOM*, pages 702–710, 2012.

[4] Y. Mirsky, A. Shabtai, L. Rokach, B. Shapira, and Y. Elovici. Sherlock vs moriarty: A smartphone dataset for cybersecurity research. In *Proc. ACM Workshop on Artificial Intelligence and Security*, pages 1–12, 2016.

[5] Z. Wang, M. M. Hayat, N. Ghani, and K. B. Shaban. A probabilistic multi-tenant model for virtual machine mapping in cloud systems. In *Proc. 3rd IEEE Int. Conf. on Cloud Networking*, pages 339–343, 2014.