DLECP: A Dynamic Learning-based Edge Cloud Placement Framework for Mobile Cloud Computing

Xiaoqun Yuan, Mengting Sun and Qing Fang
School of Information Management, Wuhan University
Wuhan, China 430072
Email: ida_sun@foxmail.com

Changlai Du
School of Computing and Informatics,
University of Louisiana at Lafayette
Lafayette, LA, USA

***Abstract*—In our poster, we deal with the edge cloud placement of Mobile Cloud Computing(MCC) by proposing a dynamic edge cloud placement framework, which integrates Machine Learning methods and Spot Instance pricing model of cloud computing. The former is implemented to predict the information of mobile users' requests and resource price, and the latter is used to reduce the edge cloud placement cost. Simulation results demonstrate that our proposed framework achieves higher performance with lower time complexity.**

***Keywords*-Edge Cloud Placement; Mobile Cloud Computing; Machine Learning; Spot Instance.**

I. Introduction

Mobile Cloud Computing(MCC) leverages unified elastic resources of varied clouds and network technologies to serve mobile devices anywhere and anytime based on the pay-as-you-go model [1]. It extends the central clouds to Internet edge by using the nearby edge clouds to provide mobile users with service of quality assurance. Therefore, edge cloud placement problem is an important issue and affects the performance of MCC significantly and directly. But existing solutions, whose placement strategies are achieved based on the existing information, do not yield their excepted results due to the features of mobility and randomness of mobile users [2]. Thus, a dynamic edge cloud placement strategy is necessary for MCC to deploy the edge clouds according to the change of mobile users' requests.

In this paper, we try to deal with this problem by proposing a machine learning-based edge cloud placement framework, which takes both the service delay and placement cost into account and is suitable for the scenario of MCC. In our framework, the pay-as-you-go model and Spot Instance pricing model of cloud computing are implemented into edge cloud placement to deploy the service resources at arbitrary location cost-efficiently. What's more, LSTM, a deep learning method, is applied to predict the end user request and resource price information with trace data, which acts as the real-time input for the generation of placement strategy. The fast edge cloud placement strategy proposed based on the prediction is to achieve the edge cloud placement strategy, including the location of edge clouds and the allocation of their corresponding resources. The experiment results with trace data show that our placement framework can achieve higher placement performance with lower time complexity.

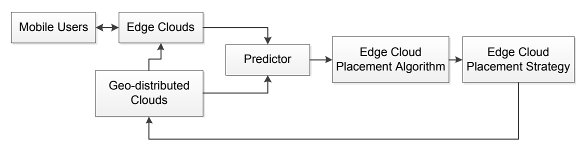


Figure 1. The edge cloud placement framework for MCC

II. Framework of Edge Cloud Placement for MCC

Edge cloud placement strategy affects MCC's service performance significantly and directly, but the mobile users' mobility requires that the edge clouds should be deployed anywhere they are required, and the randomness of end users also means that the edge cloud placement strategy should be achieved with real-time prediction information. Unfortunately, there is no prediction-based solution for this problem. Therefore, we design a dynamic placement framework which can adjust the edge cloud placement strategy based on the prediction information, as shown in Fig. 1. The workflow of our framework is as follows: Edge Clouds collect the mobile users' information such as distribution, request demand etc., and send the information to Predictor; Predictor then collects the information of geo-distributed clouds such as resource price, conducts the prediction to the mobile users' requests and resource price, and sends the results to Edge Cloud Placement Algorithm model; Edge Cloud Placement Algorithm model then generates the placement strategy for next time slot with predicted parameters according to the metric of the tradeoff between the service delay and placement cost. Based on the generated placement strategy, the corresponding clouds are chosen and the resources are allocated for each chosen cloud.

III. The Edge Cloud Placement Strategy

In this section, we introduce our optimized virtual edge cloud placement strategy, including the parameter prediction and the edge cloud placement strategy generation.

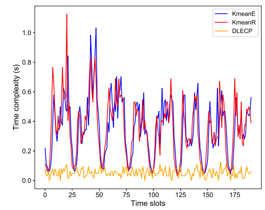
A. The Prediction Method Determination

As mentioned above, edge cloud placement strategy should be determined based on the prediction of end users' requests and resource price. However, due to the mobility, randomness and huge population of mobile users, there are obviously complex nonlinear correlations among mobile users' distribution of different time slots. What's more, to reduce the placement cost, the Spot Instance, where cloud providers offer their idle resources to end users with lower price based on the on-demand usage scheme, is used as the resource pricing model. Although this pricing strategy may provide resources with lower financial cost, the availability of resources is not guaranteed because they are accessed by bidding. Besides, the resources constructing the edge clouds in our placement strategy come from different geo-distributed clouds and each of them has its own Spot Instance pricing strategy. Thus, there are also obviously complex nonlinear correlations among the resource prices of different clouds.

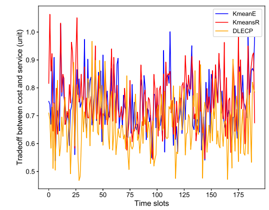
Considering that deep learning methods like LSTM can learn and capture the complicated nonlinear correlations based on the history time-series data. We apply LSTM to predict the service requests and resource prices of different clouds for our placement strategy.

B. The Edge Cloud Placement Strategy

Based on the predicted parameters, we design a fast edge cloud placement solution consisting of clustering tree construction and edge cloud placement strategy generation. Clustering tree construction aims to transfer the predicted mobile user information into a binary clustering tree, where edge clouds can be located and resources for each located edge cloud can be allocated. This clustering tree is constructed through subclusters merging with hierarchical clustering. To this end, we first divide the whole clustering space V into M subclusters with the constraint of $N >> M >> K$ (where N is the number of mobile users and K is the number of edge clouds; $>>$ means far larger, for example $c >> g$ means c is more than 10 times larger than g). Then LSTM is used to predict the number n_i and the centroid x_i of the mobile users in subcluster i. Based on these parameters, the Wards hierarchical clustering algorithm is implemented to construct our hierarchical clustering tree. The distance of each subcluster pair of this tree can be calculated based on the equation $d(i,j) = \sqrt{\frac{2n_in_j}{n_i+n_j}||\bar{x_i} - \bar{x_j}||}$, in which $||||$ is Euclidean distance, and the centroid of the new subcluster x_{new} can be updated by the equation $x_{new} = \frac{n_ix_i+n_jx_j}{n_i+n_j}$. Then, the location of edge clouds are determined based on the clustering tree and the metric of $min(Co+Ser)$, where Co means the average placement cost and Ser means the response delay. At last, all mobile users are reassigned to these located edge clouds to access the service resources.



(a) The time complexity



(b) The tradeoff between placement cost and service delay

Figure 2. The placement performances of different solutions

IV. Performance Evaluation

To verify the effectiveness of our placement framework, the placement performances of K-Means-based placement strategy and our placement strategy are compared with the New York City taxi trip datasets [3] and the Amazon EC2 price datasets [4], as shown in Fig. 2. In Fig. 2, the blue, red and orange curve denotes K-Means-based strategy with existing information(KmeanE), K-Means-based strategy with real data(KmeanR) and our strategy(DLECP), respectively. Fig. 2(a) plots the time complexity of different solutions including prediction and placement time of our strategy as well as the placement time of K-Means-based placement strategies, while Fig. 2(b) presents the placement performances of these three strategies in terms of the tradeoff value between the placement cost and service delay. From Fig. 2, we can find out that compared with KmeanE, our placement framework can achieve better placement performance shown in Fig. 2(b) with significant lower and stabler time complexity shown in Fig. 2(a). What's more, all curves in Fig. 2(a) are below 0.5 seconds while the Spot Instance service time is 5 minutes. Obviously, our placement framework is more suitable for MCC.

V. Conclusion

In this poster, we propose a machine learning-based edge cloud placement framework. It is based on the prediction information of the end users' requests and resource price in future time slots by implementing LSTM. The experiment results show that our placement framework is suitable for the scenario of MCC for that it can achieve higher placement performance with lower time complexity.

References

[1] Z. Sanaei, S. Abolfazli, A. Gani, R. Burra, *Heterogenity in mobile cloud computing: taxonomy and open challenges*, IEEE Communacion Suvery Tutorials Vol. 10, No. 1, PP: 14-22, 2013.

[2] H. Yin, X. Zhang, H. H. Liu, Y. Luo, C. Tian, S. Zhao, F. Li, *Edge Provisioning with Flexible Server Placement*, IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 4, PP: 1031-1045, 2017.

[3] *http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml*

[4] *http spot.scem.uws.edu.auec2siHome.jsp*