

An Efficient Method to Outsource Applications among Multiple Clouds

Feng Huang

College of Communication & Information Engineering

Xi'an University of Science and Technology

Xi'an, China

huangf@xust.edu.cn

Keywords: Cloud computing, Outsource, Efficient, Scheduler, Task schedule

Abstract. Cloud computing, as a novel scalable and on-demand computing service, is emerging as a leading solution for deploying on-demand application or data storage services. This paper studies the problem how to outsource applications to multiple clouds by a cost-effective way. We first formulate the process of user's cloud requests, and introduce an optimal pricing method based on linear optimization strategy. Finally, the extensive experiments demonstrate the effectiveness and practicality of our proposed strategy.

Introduction

Nowadays, the emergence of cloud computing has caught much more attentions in industry and academia areas. Cloud computing is a computing model enabling ubiquitous access to a shared and virtualized pool of configurable computing resources (e.g., CPU, storage, and network) that can be rapidly provisioned with minimal management effort [1, 2].

Comparable to traditional parallel or grid computing, cloud computing relies on sharing resources rather than having local servers handle applications. However, considering many different cloud service platforms, choosing the best or suitable cloud platform is not a simple issue. In general, different public cloud may offer different options in pricing, performance, and feature set [1]. On the other hand, the attribute of application itself is another major factor to choose the suitable cloud platform. That is to say, we should consider many correlated factors to find the best clouds [3].

Problem Statement

Cloud computing architecture consists of front-end platforms called clients or cloud clients. The Fig1 shows the application scenarios of our scheduler in multiple cloud environments. The scheduler is designed to deploy the proposed optimal strategy. The cloud user's requests, can be analyzed by scheduler before sending to public clouds. The scheduler splits the service requests by optimal strategy, taking consideration of service price, total computation, and cloud performance.

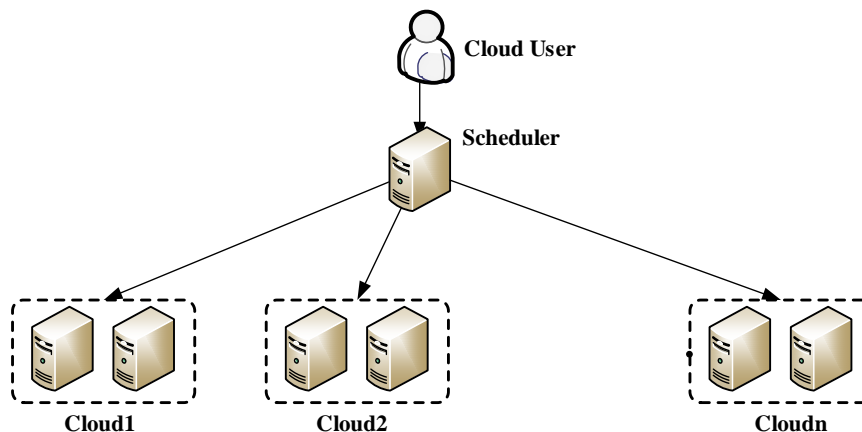


Fig.1 System architecture

The scheduler can record the price rules of different public cloud. The user formulates and sends a cloud service request to the scheduler before outsourcing to clouds. The scheduler takes consideration of three main aspects: (1) the amount computation of user's application, (2) the price of cloud resource per time unit for computation and storage, and (3) the amount of application's data storage on cloud. Hence the scheduler can deliver the service request to corresponding public cloud by minimal price strategy based on linear optimization techniques.

The Proposed Strategy

In this section, we will give our strategy in detail. Cloud computing refers to the logical computational resources (application, data storage and network) accessible via internet rather than from a local computer [4, 5]. Cost-aware application scheduling remained an active area of research since emergence of cloud and grid computing for massive applications scenario.

To construct the optimal goal of our strategy, we should formulate the service request process firstly. Let cp_i denote the computation capacity of public $cloud_c$, and $task_i$ denote the amount of computations of application i . So, the computation time $t_{i,c}$ of application i on cloud c can be expressed by Eq. 1. It should be noted that $t_{c,i}$ is the theoretical computation time.

$$t_{i,c} = \frac{task_i}{cp_c} \quad (1)$$

Suppose $p(cloud_i)$ is pricing of cloud i per unit time for computation process. Then, the total computation cost p_c of application i can be expressed by Eq. 2, which illustrates that the p_c is only related to the length of computation and the cloud pricing.

$$p_c = t_{i,c} \times p(cloud_c) \quad (2)$$

Besides computation service, user's application usually needs a certain amount of storage space during the running time on cloud. Hence, we should to quantify the storage cost of application on different clouds. Let sp_c denote the storage price on $cloud_c$, and the $p(cloud_c)$ denote the total storage of application i . And, apparently, the length of data storage time of application on cloud is determined by the length of application computation time on the cloud. Hence the storage price during computing on cloud can be expressed by Eq. 3.

$$p_s = sp_c \times t_{i,c} \times p(cloud_c) \quad (3)$$

The total cost of application is the sum of computation cost and storage cost. Obviously, the $total_i$ can be expressed by following Eq. 4.

$$total_i = p_c + p_s \quad (4)$$

As previously mentioned, the goal of our strategy tries to find out the optimal cloud delivery among multiple cloud providers while minimizing the price which users will have to pay for the total service. Hence, the optimal goal of the application delivery can be depicted by Eq. 5.

$$Min(total_i) = Min(p_c + p_s) \quad (5)$$

In theory, the linear optimization solution can be used to construct the optimal delivery model. Then, we can formulate each application on different cloud platforms by Eq. 5. In theory, linear programming is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships [6].

The scheduler can delivery the user's server requestes to corresponding cloud by the optimal results. This is the core design of our proposed strategy. Based on the above-mentioned theoretical discussion, the scheduler could analysys each application before outsourcing to public cloud.

Evaluation

To evaluate the validity of our strategy, we run real computation-intensive applications HPL (a measure of a system's floating point computing power.) in different simulated cloud platforms, and we compare the price taken by those platforms between random task schedule and our proposed strategy [7]. In addition, the price function and other parameters are randomly generated. We deploy HPL application to multiple clouds by random task schedule strategy and our proposed strategy on different dataset, and the total price is recorded for each application with different dataset. The experimental results are depicted in Fig. 2.

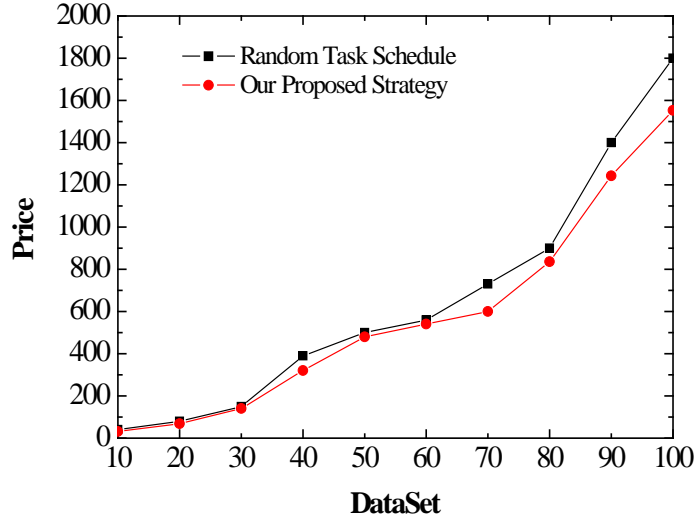


Fig 2. The effectiveness of our proposed strategy in multiple cloud environments

From Fig.2, we can see that the effectiveness of our proposed strategy in multiple cloud environments. The price taken by random task schedule strategy is always higher than that by our proposed strategy. To sum up, the experimental results approve our proposed strategy is effective.

By analyzing the price consumed by those two strategies, it is evident that our scheduler make some contributions to user's application process while reducing execution cost and time. Next, we will validate the running time of application scheduled by our proposed strategy. The experimental results is shown as in Fig.3.

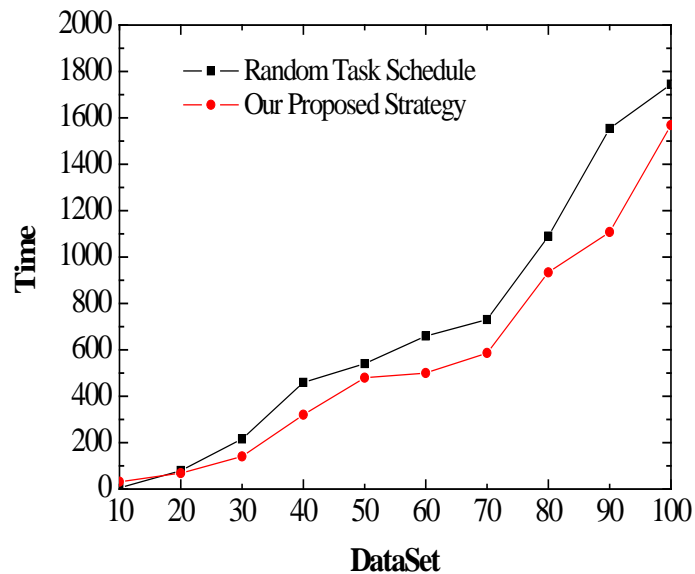


Fig 3. The running time of application by our proposed strategy in multiple cloud environments

We can conduct that the running time of user's application outsourced by random task schedule is higher than that by our proposed strategy in most case from Fig.3. That is to say, our proposed strategy is very effective in task scheduling process.

Taking consideration into experimental results shown as in Fig.2 and Fig.3, the results show that, dealing with multiples cloud providers, the scheduler has increasing flexibility and opportunities in offering methods cloud user at competitive prices situation. In all, the effectiveness and practicality of our proposed strategy is validated.

Conclusion

In this paper, we present how to schedule application among multiple public clouds taking consideration into both price and computer capacity. Our proposed strategy supports different cloud service delivery model, such as IaaS (Infrastructure as a Service), PaaS (Platform as a Service) or SaaS (Software as a Service). And the experiment demonstrates that our proposed strategy is efficient.

References

- [1] I. Houidi, M. Mechtri, W. Louati, D. Zeglache: *Cloud Service Delivery across Multiple Cloud Platforms*, 2011 IEEE International Conference on Services Computing, p 741-742. (2011)
- [2] H. M. Fard, R. Prodan, T. Fahringer, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 6. (2013)
- [3] D. Petcu: *Consuming Resources and Services from Multiple Clouds*, *Journal of Grid Computing*, Vol. 12, No. 2, p 321-345. (2014)
- [4] M. E. Frincu, S. Genaud, J. Gossa: *Comparing Provisioning and Scheduling Strategies for Workflows on Clouds*, 2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum, p 2101-2110. (2013)
- [5] T. A. L. Genez, L. F. Bittencourt, E. R. M. Madeira: *Using Time Discretization to Schedule Scientific Workflows in Multiple Cloud Providers*, 2013 IEEE 6th International Conference on Cloud Computing (CLOUD 2013), p 123-130. (2013)
- [6] E. N. Alkhanak, S. P. Lee, S. U. R. Khan, *Future Generation Computer Systems*, Vol. 50, p 3-21. (2015)
- [7] N. Ferry, H. Song, A. Rossini, F. Chauvel, A. Solberg: *CloudMF: Applying MDE to Tame the Complexity of Managing Multi-Cloud Applications*, 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. p 269-277. (2014)