UCL DEPARTMENT OF
COMPUTER SCIENCE

Research Note
RN/13/16

Customer-centric Optimal Software Release Problem in Cloud

September 13, 2013

Mustafa Bozkurt

Yuanyuan Zhang

Abstract

As the number of cloud providers increases, the competition forces providers to reduce the prices to stay competitive. A recent testimony to this claim is the “price wars” between the cloud storage service providers [1]. In such competitive business environments, it is safe to assume that the providers will want to know more about the needs of their customers in order to shape their services based on customer demand. At the same time, try to find ways reducing the cost of their services in order to maintain the profitability. This problem is an instance of optimal software release problem which is NP-Hard [2]. In this paper, we propose an approach aimed at helping service providers to optimise their service configurations based on user demand. Unlike the existing work, our approach does not just provide n number of optimal configurations but rather allows the decision maker to discover trade-offs between service cost and customer satisfaction by comparing a set of Pareto optimal solutions. Our approach can help providers to discover the configurations that maximises profit while providing a high level of customer satisfaction.

Customer-centric Optimal Software Release Problem in Cloud

Mustafa Bozkurt
University College London
m.bozkurt@cs.ucl.ac.uk

Yuanyuan Zhang
University College London
yuanyuan.zhang@cs.ucl.ac.uk

***Abstract*—As the number of cloud providers increases, the competition forces providers to reduce the prices to stay competitive. A recent testimony to this claim is the "price wars" between the cloud storage service providers [1]. In such competitive business environments, it is safe to assume that the providers will want to know more about the needs of their customers in order to shape their services based on customer demand. At the same time, try to find ways reducing the cost of their services in order to maintain the profitability. This problem is an instance of optimal software release problem which is NP-Hard [2]. In this paper, we propose an approach aimed at helping service providers to optimise their service configurations based on user demand. Unlike the existing work, our approach does not just provide n number of optimal configurations but rather allows the decision maker to discover trade-offs between service cost and customer satisfaction by comparing a set of pareto optimal solutions. Our approach can help providers to discover the configurations that maximises profit while providing a high level of customer satisfaction.**

***Keywords*-Optimal software release problem, Customer satisfaction, Service pricing, Cloud services**

I. Introduction

According to Heskett et al. [3], one of the common characteristics of long lasting successful businesses is that they can achieve outstanding results for their customers with the lowest costs. Based on this classical business analysis the two main business concerns of a cloud provider should be:

Customer Satisfaction which requires providing the service your customer want/need with a competitive price.

Reduced cost which requires finding the optimal service to provide. This optimal service provision includes all aspects of that service such as hardware configuration, level of Quality of Service (QoS) and customer service.

The problem of service cost/profit and customer satisfaction in cloud services has been addressed at two different dimensions. The first group of existing work are aimed at maximising profit using resource management strategies [4], [5]. The advantage of these approaches is that they might help reducing the service cost as well as increasing the customer satisfaction by minimising the Service Level Agreement violations. The rest of the existing work aimed at discovering the optimal service configurations and price based on user demand such as Knapper et al. [6].

Even though, these approaches can help with customer satisfaction and profitability, we believe that they do not provide a complete solution to the problem at hand. The existing work in the first category cannot be used in discovering service configurations that will increase customer satisfaction. The existing work in the second category do not provide any insight to the decision maker due to only providing a small number of "optimal" service configurations (to the best of authors' knowledge). There are also other issues such as the cost difference between versions of a service which Knapper et al. [6] considers negligible. However, we believe cost difference is an important business aspect that needs to be included in the problem formulation because it directly affects competitiveness.

In this paper, we present an approach aimed at providing a more comprehensive solution to optimal service release problem. The proposed approach differentiate itself from the existing work in two ways. First, the proposed approach incorporates all of the mentioned business concerns of a service provider. Second, the approach enables the decision maker to discover trade-offs between these concerns in order to determine optimal service releases. This is facilitated by providing the decision maker with a set of pareto optimal solutions discovered using user feedback and NSGA-II algorithm [7]. We propose the use of NSGA-II as it is expected outperform other multi-objective algorithms in similar problems according to the literature [2].

II. Pareto Optimal Software Release Problem

Since we consider service release problem as an instance of optimal software release problem, we propose adapting one of the existing approaches, which was proposed by Zhang et al. [2], to cloud domain.

A. Optimal Software Release Problem

We adopt the definitions used by Zhang et al. for cloud services. In this approach, there is a set of customers,

$$C = \{c_1, c_2, ..., c_n\}$$

whose preference is considered as the requirements

$$R = \{r_1, r_2, ..., r_m\}$$

for the services to be provided.

In cloud services, the customer requirements are often of two types: **System configurations** (such as operating system and printing functions or the amount of system resources such as memory, storage and bandwidth) and **QoS configurations** (such as availability, security, latency). Each of these requirements have a certain cost for the service provider. The cost of a service is the total cost of each included configuration and it is denoted as:

$$Cost = \{cost_{r_1}, cost_{r_2}, ..., cost_{r_m}\}$$

where $cost_{r_j}$ represent the cost of requirement r_j.

Our approach also considers the fact that each customer might have different preference over different requirements. For example, for some customers the amount of storage might be very important while the amount of memory might

be a secondary concern. Whereas, for other customers the amount of memory might be the main concern. In order to accommodate customer-centricity, the customer preference on requirements is represented as a value function. The value function formulated as:

$$value(c_i, r_j) = \begin{cases} value > 0 & \text{if } c_i \text{ has } r_j \\ value = 0 & \text{otherwise} \end{cases}$$

where each customer $c_i (1 \leq i \leq n)$ assigns a value to each requirement $r_j (1 \leq j \leq m)$.

We assume that a provider might value their customers in different degrees. In this case, the provider might want to assign varying degrees of importance to their satisfaction. This is also another important aspect that needs to be reflected in the formulation. As a result, the approach includes a weight factor assigned to each customer c_i as

$$Weight = \{w_{c_1}, w_{c_2}, ..., w_{c_n}\}$$

where w_{c_y} is the weight of customer $c_y, w_y \in [0, 1]$ and $\sum_{y=1}^{z} w_y = 1$.

The importance (or score) of a requirement combining the formulation above can be calculated as:

$$Score_{r_j} = \sum_{y=1}^{z} w_y * value(c_y, r_j)$$

where $score_{r_j}$ is the importance of requirement r_j for customer c_y which represents the requirement's overall value for the provider.

B. Multi-objective Formulation

In our multi-objective formulation we do not consider service cost and customer satisfaction just as constraints but rather set them as objectives. This will allow the decision maker to explore all the solution set discovered on the pareto optimal front. As a result, the decision maker will be able to discover and understand trade-offs between these two objectives to help him/her making a more informed decision.

In the case of the customer satisfaction, it is safe to assume that the provider will want to maximise it. As a result, the objective function for the customer satisfaction is formulated as:

$$Maximise \sum_{j=1}^{m} score_{r_j} * r_j$$

On the other hand, the provider will want to reduce the cost of the provided services. As a result, the objective function for the service cost is formulated as:

$$Minimise \sum_{j=1}^{m} cost_{r_j} * r_j$$

III. Case Study

We will demonstrate the feasibility of the proposed approach on the Virtual Desktop Service (VRS) domain. In VRS domain, customers are provided with a cloud service that can be used as a computer replacement. VRS providers offer configuration options similar to a traditional computer such as CPU, storage and memory.

According to the literature, one of the most common ways of determining customer demand and satisfaction is conducting surveys [8]. As a result, we will try to characterise the customer demand, the importance of each configuration option for different customers and the price that they are willing to pay by conducting two public surveys.

In the first survey, we will ask the participants about the system configuration that they prefer and the amount of money they are willing to pay for this configuration. This study will allow us to discover the configurations that customers might prefer. The data from the survey will be used discover a pareto front of possible configurations.

In the second survey, we will ask the participants about their QoS expectations. The data from this survey will be used to determine the possible QoS configurations that users might prefer. The second survey will be focussing on customers with knowledge of QoS concepts in order to collect realistic data. Each configuration discovered on the first front (combined with data from the second survey) can be used to generate a new pareto front which enables exploring trade-offs between profit and customer satisfaction. The solutions on the new front will be optimal QoS versions for the selected configuration(s) from the first front.

IV. Conclusion

In this paper, we presented a pareto efficient solution to optimal service release problem. The proposed approach differentiate itself from the existing approaches by enabling the providers to discover trade-offs between service cost and customer satisfaction rather than just providing a number of optimal solutions. The approach helps service providers to discover optimal service configurations that provide high level of customer satisfaction with low cost, thus increasing customer retention and profitability.

References

[1] D. Meyer. (2012, December) Microsoft joins cloud storage price war with 28 percent azure cut. http://www.zdnet.com/microsoft-joins-cloud-storage-price-war-with-28-percent-azure-cut-7000008412/.

[2] Y. Zhang, M. Harman, and S. A. Mansouri, "The multi-objective next release problem," in *GECCO '07*, London, UK, 7-11 July 2007, pp. 1129–1137.

[3] J. L. Heskett, W. E. Sasser, and L. A. Schlesinger, *Service Profit Chain*. Simon and Schuster, 1997.

[4] H. J. Moon, Y. Chi, and H. Hacigumus, "SLA-aware profit optimization in cloud services via resource scheduling," in *SERVICES-1*. IEEE, 2010, pp. 152–153.

[5] J. Chen, C. Wang, B. B. Zhou, L. Sun, Y. C. Lee, and A. Y. Zomaya, "Tradeoffs between profit and customer satisfaction for service provisioning in the cloud," in *HPDC '11*. ACM, 2011, pp. 229–238.

[6] R. Knapper, C. M. Flath, B. Blau, A. Sailer, and C. Weinhardt, "A multi-attribute service portfolio design problem," in *SOCA 2011*. IEEE, 2011, pp. 1–7.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[8] B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," in *FOSE '07*. Minneapolis, MN, USA: IEEE, May 2007, pp. 285–303.