AQUAMan: An Adaptive QoE-Aware Negotiation Mechanism for SaaS Elasticity Management

(Extended Abstract)

Amro Najjar
MINES Saint-Etienne, CNRS
Lab Hubert Curien UMR 5516
amro.najjar@emse.fr

Olivier Boissier
MINES Saint-Etienne, CNRS
Lab Hubert Curien UMR 5516
olivier.boissier@emse.fr

Gauthier Picard
MINES Saint-Etienne, CNRS
Lab Hubert Curien UMR 5516
gauthier.picard@emse.fr

ABSTRACT

Client churn is a key challenge confronted by SaaS providers. Recent research in QoE suggested providers should rely on quantiles & percentiles to assess the service acceptability rate. In this article we introduce AQUAMan, an Adaptive QoE-Aware multi-agent negotiation mechanism for SaaS elasticity Management. Based on its estimation of the percentage of users finding the service acceptable and a learned model of the user negotiation strategy, AQUAMan adjusts the provider negotiation process in order to restore the desired service acceptability rate while meeting the budget limits (*i.e.* the cost paid to rent cloud resources) of the provider. The proposed mechanism is implemented and its results are examined and analyzed in light of comparable results.

Keywords: One-to-many negotiations, Service Acceptability, QoE, SaaS, Cloud Computing.

1. INTRODUCTION

In today's cloud computing market, a Software as a Service (SaaS) provider has to balance two concerns: minimizing client churn while meeting its budget constraints. In the context of cloud computing, this issue is known as elasticity management or auto-scaling [3, 10], and it has received considerable attention in the recent years. However, in most of the works tackling this issue the provider takes the resource allocation decision unilaterally [10, 2]. Consequently, the end-user preferences are mostly overlooked and it is often presumed that their acceptability threshold tolerates the best-effort service proposed by the provider.

Quality of Experience (QoE) is the quality of service as perceived subjectively by the end-user [7]. QoE-management emerged as process whereby the provider seeks to maximize user satisfaction while at the same time maximizing resource efficiency and economy [15]. Despite the promises of being *personal* and *subjective*, most of existing works in QoE rely on the Mean Opinion Score (MOS). Since it is an average of user opinion, MOS has been criticized for hiding important information about user diversity [4]. Consequently, other metrics have been proposed. In particular, recent empirical studies on QoE [5, 6] recommend providers to rely on quantiles and percentiles to gauge the users' acceptability of service more precisely. Thus, the provider can ascertain that, say, at least 95% of its users found the service to be acceptable.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

One-to-many multi-agent negotiation provides an interesting platform to involve the end-user into the elasticity management process. EMan [8, 11, 9] is a multi-agent architecture for QoE-aware elasticity management. In EMan each end-user is represented by a personal agent that seeks to satisfy her preferences and maximize her QoE. The elasticity management decision results from mutually acceptable agreement reached via a negotiation process between provider and user agents. This allows to realize the theoretical vision of QoE where QoE is considered to be personal and influenced by *Human Influence Factors* and the context [14]. While EMan manages to involve the end-user into the elasticity management decision process, it does not allow the provider to adjust its negotiation behavior in order to achieve a precise predefined acceptance rate. In this article we develop AQUAMan to address this limitation. AQUAMan is an Adaptive & QoE-Aware SaaS elasticity Management mechanism implemented in the EMan architecture to allow the provider to achieve a target service acceptability rate while satisfying its budget constraints. User agents can decide whether to accept or reject the proposed service depending on their subjective estimation of the service quality. Based on its estimation of the portion of users finding the service unacceptable, the provider adjusts its negotiation strategy in order to restore the acceptability rate to its predefined goals.

2. THE ADAPTIVE NEGOTIATION MECHANISM

The EMan architecture depicted in [9] is a one-to-many negotiation architecture where one SaaS can negotiate simultaneously with multiple users [9, 8]. The provider is represented by two types of agents: delegates (a delegate is denoted as da_i) and a single coordinator (ca). Each da_i is responsible of a bilateral negotiation session with one user while ca oversees the negotiation process. User agents (denoted as sa_i) derive their utility functions from user preferences, rely on evidence from Psychophysics (the Weber-Fechner Law [16]) and results from empirical QoE studies [13] in order to maximize the QoE of end-users. Provider agents seek to minimize the cost paid to rent resources from the cloud. For further information about the negotiation strategies and protocols in EMan please refer to [9].

2.1 Triggering the Adaptation Mechanism

Whenever a bilateral negotiation session i is terminated, the coordinator is notified by da_i about the outcome of this session. Using these data, the coordinator runs a quantile estimation algorithm to detect the current service acceptability rate.

Let Q be the quantile/percantile estimation function. Let R be the dataset containing the outcomes of the terminated sessions. R can

contain either 0's, for failed sessions, or 1's for successful sessions. If the coordinator seeks to ensure that β percent of users who requested the service so far have had a successful negotiation session (hereby accepting the proposed service quality), the coordinator needs to verify that the $(100-\beta+1)^{th}$ percentile equals 1:

$$Q(R, 100-\beta+1) = 1 \quad (1)$$

As long as this condition holds, the coordinator has no need to intervene into the negotiation process.

Once the condition in (1) is violated, the coordinator triggers the adaptation mechanism by commanding all working delegates to activate their adaptive mode.

2.2 Opponent Learning and Modeling Algorithm

When a delegate da_i receives an offer $o^t_{sa_i}$ at cycle t from the corresponding sa_i it estimates the concession made by sa_i by comparing $o^t_{sa_i}$ with $o^{t-1}_{sa_i}$, the previous offer made by sa_i. da_i relies on its own utility function to estimate the concession made by sa_i by assuming that a concession made by sa_i is synonymous with a utility gain for da_i. Thus, $c^t_{sa_i}$, da_i's estimation of the concession made by sa_i at the negotiation cycle t is defined as:

$$c^t_{sa_i} = M_{da_i}(o^t_{sa_i}) - M_{da_i}(o^{t-1}_{sa_i}) \quad (2)$$

Based on sa_i's concession behavior da_i infers T_{sa_i}, sa_i's negotiation time deadline. This can be achieved, as has been shown in the literature [1] using non-linear regression assuming that all users follow time-based concession strategies. Unlike the literature, a delegate can run the non-linear regression algorithm only once. To decide when to launch it, first the adaptation mechanism should be active (c.f. Section 2.1). Second, da_i examines the rate of change of sa_i concessions. When the rate of change decays into significant negative values, this means that sa_i has made most of its significant concessions. da_i runs the non-linear regression algorithm at this stage. The output is an estimated value of sa_i deadline (denoted as $\bar{T_{sa_i}}$). Based on $\bar{T_{sa_i}}$, da_i computes $\bar{r_i} = \bar{T_{sa_i}} - t$ the estimated number of cycles remaining in session i.

2.3 Negotiation Adaptation

The coordinator (ca) receives $\bar{r_i}$ from all negotiation sessions i whose delegate da_i considers that sa_i is approaching its time deadline. ca stores these estimations in its *priority list* which is continuously sorted in ascending manner. The sessions are not synchronized. Therefore, ca repeats the sorting continuously. Whenever a session is terminated either successfully or not, ca removes its record from the priority list. To determine how many users will be selected from the list, ca calculates p, the number of successful sessions needed to restore the desired acceptability rate. Then ca chooses the first p sessions from the priority list and adjust their negotiation strategies. The strategy of da_i is adjusted as follows.

First, the negotiation time deadline of da_i becomes $\bar{r_i}$ instead of T_{da_i}. Second, the reservation cost of da_i,a variable denoted as RC which is the maximum cost the provider spends on a user, is increased by a value denoted as $RcPrio$. $RcPrio$ is computed by dividing the surplus available in *Surplus* among all the prioritized sessions. *Surplus* is a variable that contains the surplus accumulated from successful negotiation sessions. $surplus_i$, the surplus obtained from the session i, is computed as follows:

$$surplus_i = RC - Cost(\hat{o_i}) \quad (3)$$

where $\hat{o_i}$ is the offer accepted by both parities in session i and $Cost(\hat{o_i})$ is the cost paid to the cloud provider to realize this offer.

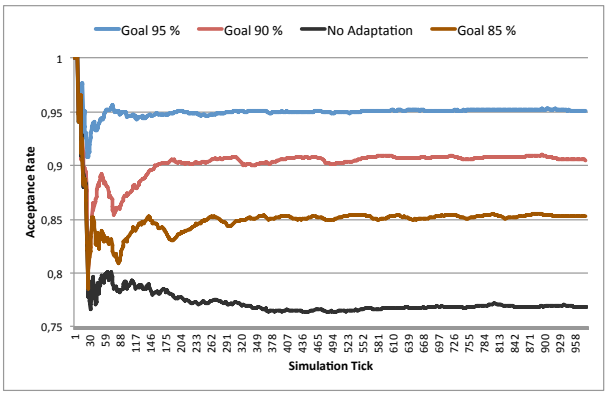

Figure 1: Acceptance rate adjustment.

Thus, by using this surplus redistribution mechanism, the provider ensures the satisfaction of its budget constraints.

3. EVALUATION

AQUAMan is implemented in the EMan architecture. The latter is a one-to-many multi-agent negotiation architecture implemented using Repast Simphony [12]. In the experiments, 10000 users enter the system. The user profiles are generated randomly. The cost of services acceptable by users ranges from 0.1 $ to 0.9$. RC, delegate reservation cost (the maximum cost allocated to a non-prioritized user) is set to 0.60$. This parameter represents the provider's budget constraints. $Goal$ is the percentage of users that the provider seeks to satisfy.

Figure 1 shows the results of the experiment. The blue, the red and the brown curves plot the acceptance rate when $Goal = 95\%$, $Goal = 90\%$ and $Goal = 85\%$ respectively. The figures show that the adaptation mechanism achieved the predefined acceptance rate. The black curve plots the acceptance rate when the mechanism is deactivated. Intuitively, the enhancement of the acceptability rate comes with more cost invested per user. However, this increase does not violate the provider budget constraint (*i.e.* $RC = 0.6$\$ per user): the average costs per user were 0.55 $, 0.52$, 0.5$ and 0.44$ for $Goal = 95\%$, $Goal = 90\%$, $Goal = 85\%$ and non-adaptive respectively.

Furthermore, the experimental evaluation proved that AQUAMan manages to cope with load spikes where thousands of users may enter the system simultaneously. These results are not included due to space limitations.

4. CONCLUSION AND PERSPECTIVES

The proposed mechanism (AQUAMan) integrates the users' QoE as well as their subjective evaluation of service acceptability and endows the provider with a fine-grained control of the desired acceptability rate while meeting its budget constraints.

Our future research work will be directed towards giving the provider a finer-grained control over the level of user satisfaction it seeks to attain. In particular, beyond acceptability adjustment, the provider should be able to ensure that a predefined percentage of users consider the service to be *Good* or *Better* [6].

REFERENCES

[1] T. Baarslag, M. J. Hendrikx, K. V. Hindriks, and C. M. Jonker. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, pages 1–50, 2015.

[2] E. F. Coutinho, F. R. de Carvalho Sousa, P. A. L. Rego, D. G. Gomes, and J. N. de Souza. Elasticity in cloud computing: a survey. *annals of telecommunications-annales des télécommunications*, 70(7-8):289–309, 2015.

[3] G. Galante and L. Bona. A survey on cloud computing elasticity. In *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*, pages 263–270. IEEE, 2012.

[4] T. Hobfeld, R. Schatz, and S. Egger. Sos: The mos is not enough! In *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, pages 131–136. IEEE, 2011.

[5] T. Hoßfeld, P. E. Heegaard, and M. Varela. Qoe beyond the mos: Added value using quantiles and distributions. In *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*, pages 1–6. IEEE, 2015.

[6] T. Hoßfeld, P. E. Heegaard, M. Varela, and S. Möller. Qoe beyond the mos: an in-depth look at qoe via better metrics and their relation to mos. *Quality and User Experience*, 1(1):2, 2016.

[7] S. Möller and A. Raake. *Quality of Experience*. Springer, 2014.

[8] A. Najjar. *Multi-Agent Negotiation for QoE-Aware Cloud Elasticity Management*. PhD thesis, École nationale supérieure des mines de Saint-Étienne, 2015.

[9] A. Najjar, C. Gravier, X. Serpaggi, and O. Boissier. Modeling user expectations satisfaction for saas applications using multi-agent negotiation. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 399–406, Oct 2016.

[10] A. Najjar, X. Serpaggi, C. Gravier, and O. Boissier. Survey of elasticity management solutions in cloud computing. In *Continued Rise of the Cloud*, pages 235–263. Springer, 2014.

[11] A. Najjar, X. Serpaggi, C. Gravier, and O. Boissier. Multi-agent systems for personalized qoe-management. In *Teletraffic Congress (ITC 28), 2016 28th International*, volume 3, pages 1–6. IEEE, 2016.

[12] M. J. North, T. R. Howe, N. T. Collier, and J. Vos. The repast simphony runtime system. In *Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms. Argonne, Illinois, USA: Argonne National Laboratory*. Citeseer, 2005.

[13] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo. The logarithmic nature of qoe and the role of the weber-fechner law in qoe assessment. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[14] U. Reiter, K. Brunnström, K. De Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank. Factors influencing quality of experience. In *Quality of Experience*, pages 55–72. Springer, 2014.

[15] R. Schatz, M. Fiedler, and L. Skorin-Kapov. Qoe-based network and application management. In *Quality of Experience*, pages 411–426. Springer, 2014.

[16] L. L. Thurstone. Psychophysical analysis. *The American journal of psychology*, 38(3):368–389, 1927.