Performance Oriented Genetic Algorithm Framework of Concurrent SLA Negotiations in Cloud

Okta Nurika, Irving V. Paputungan and Mohd Fadzil Hassan

Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar 31750, Malaysia

Submitted: Jan 5, 2014; Accepted: Feb 24, 2014; Published: Mar 12, 2014

Abstract: The maintenance-free and heterogeneous characteristic of cloud computing have attracted personal and corporate consumers to utilize cloud services. Many providers offer various services with varying fees depending on service level agreements. For a new cloud consumer, the phase of choosing a preferred cloud provider can be tricky. Most current works on concurrent service level agreement negotiations focus on provider’s benefits, by bringing the negotiation towards either maximum price or minimum resources utilization. Hence, we propose a consumer oriented framework by developing a performance oriented concurrent service level negotiations. We use genetic algorithm as our methodology for it facilitates concurrency of service level agreements from multiple providers.

Key words: Genetic algorithm · Service level agreement · SLA negotiation · Cloud

INTRODUCTION

The advancement of hardware technologies has prompted computer users both personal and corporate parties to cope up. However, the frequent upgrade of hardware may not always be practical and healthy for the financial aspect of certain parties. When maintenance cost comes into place, the Information Technology (IT) budget rises even higher. A solution nowadays is to outsource the hardware part by utilizing cloud computing infrastructure offered by cloud service providers. The use of cloud services is regulated by Service Level Agreement (SLA) between the consumer and the provider.

For consumers who are relatively new or inexperienced with the cloud-based services, obtaining quality cloud services to meet their business needs is not an easy task. Therefore, a mechanism to automate the negotiation on the cloud quality of service parameters is required, that not only considers the business objectives, but also account for the precise service level values for better service delivery. This enables customers to acquire the best quality of service that meets their needs.

In the proposed work, existing research works on SLA management are reviewed and it is hypothesised that by having a SLA negotiation mechanism that can support inexperienced or new cloud users to negotiate with multiple cloud service providers at the same time will ensure users’ needs are satisfactorily met. To achieve this, qualitative study on the SLA parameters and software agent technology that learns from the SLA measurements, performance and usage during a specific service lifetime will be conducted. Eventually, a concurrent SLA negotiation scheme will be proposed.

Related Works

SLA in Cloud Computing: Service Level Agreement (SLA) is defined as a contract between the service provider and the service consumer in which the expectations of the service provisioning is specified, including penalties that should be applied when a violation occurs (Begnum, Burgess, Jonassen and Fagerness [1], Greenwood, Vitaglione, Keller and Calisti [2], Kandukuri, Paturi and Rakshit [3], Yan et al. [4], Patel, Ranabahu and Sheth [5], Mahbub and Spanoudakis [6]). It contains certain service level objectives (SLOs) that define objectively measurable conditions for the service, e.g. availability, throughput and response time. SLOs can vary depending on the applications or the data that are outsourced (Ahronovitz [7]). SLA is also commonly used to address issues of problem management, legal
compliance, resolution of disputes, customer duties, security and confidential information and termination (Yan et al. [4]).

SLA is an important aspect of cloud computing (Wu and Buyya [8], Jamkhedkar, Lamb and Heileman [9]). SLA is established by a negotiation process between both parties prior to service provisioning (Pichot, Waldrich, Ziegler and Wiedler [10]). Negotiation in this area is commonly defined as the process by which the involved parties come to a mutually acceptable agreement on some matter (Yaqub et al. [11]). During negotiation, the service is designed to the consumer’s needs and provider’s capabilities to resolve conflicts (Chhetri et al. [12]). Consumers usually want to obtain a high-quality service at low costs. Likewise, providers try to achieve the highest possible profit in line with demand, given their currently available Quality of Service (QoS) levels and capacities. Finally, consumers and providers have to decide on a promising negotiation strategy respectively. Since both parties try to achieve the highest possible utility and, due to the business context, do not want to disclose too much private information (e.g., business goals, cost factors), a negotiation on the QoS parameters is necessary.

In addition, there are multiple competing providers in the market, which offer the same type of service, but with different properties. This situation is complicated for new or inexperienced users who just started to try exploiting this emerging technology. Therefore, this type of consumers wishes to have a helpful mechanism in negotiating SLA terms at ease. Likewise, it is also required to conduct concurrent negotiations with multiple providers from a customer’s point of view in order to determine the highest possible profit. Further challenges arise in relation to the self-understanding of how good QoS values the customer should need. While most customers are willing to have the maximum service level, they in turn must acquire the necessary and acceptable QoS level or resources needed. Hence, besides the issue of negotiation, the baseline measurement on performance and resource usage prior to negotiation also need to be considered.

Due to the large number of cloud providers and potential cloud consumers, the information exchange between the parties involved is very complex. Thus, a dynamic, scalable and automated approach is required for negotiating SLAs with multiple providers.

**Introduction to Genetic Algorithm (GA):** Whitley [13] defines genetic algorithm as an optimization model that uses population on which selection and recombination operator will work to produce new individuals in a broad search space. Each of these individual contains a combination of near optimum solution.

The elements of genetic algorithm as informed by Mitchell [14] are briefly described below:

**Population of Chromosomes:** Every chromosome holds combination of solution to a specific problem. The fitness value of each chromosome will be evaluated.

**Selection:** Process of selecting chromosomes with better fitness value. The selected chromosomes will go into reproduction phase.

**Crossover:** The exchange of chromosome genes (subsequences) between two chromosomes to produce two new child chromosomes.

**Mutation:** Process of changing a gene’s value in a chromosome in order to create diversity of solution. It happens according to the specified probability rate.

We choose genetic algorithm as our methodology to optimize performance oriented concurrent SLA negotiations in cloud, because it accommodates concurrent combination of SLA solutions for different service providers. Furthermore, the fitness objective can be adjusted to suit consumer’s preference such as application processing time or network speed.

**Previous Concurrent SLA Negotiations:** Shi and Sim [15] have developed concurrent agent-based negotiation and commitment management for multiple web services procurement. Multiplication in this case refers to SLA negotiation between a consumer and multiple web services of different functionalities and QoS (Quality of Service), which are offered by various providers. Concurrent multiple SLA negotiation between a consumer and multi web services are made possible by Shi and Sim [15], or in other words they provide multiple one-to-many SLA negotiations by proposing a regression-based coordination strategy.

A breakthrough made by Shi and Sim [15] is that they offer cancellation of contract while SLA negotiation process is taking place, therefore the cancelled services can be given to other consumer and to let the agent keep searching for the best contract even before the current negotiation terminates. The cancellation can be initiated by either the consumer or provider.
The SLA negotiation strategies proposed by Shi and Sim [15] consist of commitment management strategy (CMS) for single service and adaptive CMS profiles for multiple services. In CMS for single service, a consumer and providers of the same service each prepare a proposal and provider’s agent will negotiate these proposals. The consumer will accept the offer with the highest expected utility after taking the rounds of SLA negotiation with multiple providers. While in adaptive CMS profiles for multiple services, each market type for each service is identified based on 3 criteria; favorable market, balanced market and unfavorable market. Favorable market is identified when consumer finds many providers that offer the requested service and only few consumers bidding for that service. Similar number of providers and consumers will make a balanced market, while unfavorable market occurs when there are many consumers with only few providers available. For each market type, Shi and Sim inform the appropriate CMS strategy for a consumer to get higher utility.

The SLA negotiation approach by Shi and Sim [15] looks more into the economical aspect of the consumer’s bid instead of the performances of the services. Furthermore, as the number of providers increases, the consumer may involve in a long search for the highest utility offered and might well lose the best provider if they move to the next round and instead of finding a better provider, they find a provider offering less utilities.

Aknine, Pinson and Shakun [16] presented an extension of the contract net protocol (CNP) in order to support concurrent many-to-many negotiations. Basically, the contract net protocol is a simple one round-based protocol for task distribution in IT systems. The authors introduce a two-phase negotiation process which enables prospective contractors to overbid other offers. However, strategies and the assignment of multiple tasks are not considered in their approach. This approach also looks more into economical bids than the wanted performances by the consumer.

Di Nitto, Di Penta, Gambi, Ripa and Villani [17] suggested a search-based solution for SLA negotiation. Similar to the work by Chhetri et al. [12], each negotiation participant is represented by a coordinator and several negotiators. In contrast to Ahronovitz et al. [7], Di Nitto, Di Penta, Gambi, Ripa and Villani [17] make use of an intermediate mediator in the form of a marketplace. The marketplace issues proposals to the participants based on an optimization algorithm in order to improve the convergence of the offers. The authors do not explicitly state a protocol for message exchange and private information is disclosed to the marketplace [28-29].

Dang and Huhns [18] adapted the alternate offers protocol in order to support concurrent negotiations with multiple issues in case of many-to-many negotiations. Their approach is based on the extended CNP [6] and also introduces negotiation phases. Since counter-proposals can overbid formal proposals, it is obvious, that the negotiation may result in an infinite loop. The authors argue that this situation can be prevented by enforcing time constraints. However, time-dependent strategies are not considered in their approach. Dang and Huhns’ [18] approach may be affected by issues concerned in Shi and Sim [15] (lengthy negotiation and risk of skipping the better service provider).

Sim [19] focused on market-based SLA negotiations in cloud computing. The work considers a three-tier model where negotiation takes place between consumers, brokers and resource providers. A market-driven strategy is applied, where the concession amount depends on time, trading alternatives and competition. In his model, an agent can also breach an intermediate contract by paying a penalty fee. The main goal for consumers is price minimization, or in other words, Sim [19] offers another economy oriented approach rather than performance of the required service. Moreover, for general SLA negotiation support, other QoS parameters must also be considered to achieve cloud service’s performance.

Zulkernine, Martin, Craddock and Wilson [20] then presented a negotiation broker middleware framework to support automatic SLA negotiation in SOA (Service Orientation Architecture). A policy specification expressing the business goals, contexts, preferences, constraint and values is negotiated using time-dependent cost-benefit model. There is a need to include negotiation history in the framework to apply learning process hence can fasten the negotiation duration. Zulkernine and Martin [21] then presented a novel trusted SLA negotiation broker to help bilateral bargaining in adaptive and intelligent ways between service provider and service consumer based on certain level of business requirements. The broker allows the negotiating parties to communicate their context, goals, business rules and negotiation preferences. The adaptive algorithm is able to accommodate such parameters updates to generate more effective counteroffers. Both works did not consider the customer offers as the main parameter in the first place.

Hasselmeyer, Koller, Kotsiopoulos, Kuo and Parkin [22] argued that dynamic pricing components can be the point of interest during negotiation in a SOA. The dynamic pricing is based on the resource capabilities and resource availability on provider side. Provider adjusts the price in response to the changing supply and demands
and historical data of the marketplace. This SLA negotiation research by Hasselmeyer, Koller, Kotsiopoulos, Kuo and Parkin [22] brings up another economy perspective of the cloud’s SLA.

Kwon, Shin and Kim [23] applied context-aware computing to support pervasive negotiation support system by acquiring and managing the user's current context. User agents actively announce what their users want to purchase, which will entice many more promotions from a broader range of suppliers and hence increase user satisfaction.

Vahidov [24] proposes a distinct SLA negotiation management where it handles one service provider with multiple consumers (one-to-many). His approach took on the price of service and CPU utilization (hours) as the parameters to consider, while aiming to create a win-win agreement for both consumer and service provider. The length of negotiation itself is based on assumption that it will last less than a day.

The SLA negotiation framework by Vahidov [24] consists of agents and manager. The agents on behalf of consumers will commit and inform the manager on behalf of provider, about the utilized CPU hours. If it is more than targeted then the manager will reduce the discount values for the additional CPU hours. Furthermore, if there are not enough successful negotiations as planned, then the service price will be decreased. However, both the discount and price adjustment are limited by the setting created from human’s consideration. Moreover, the negotiation focuses on price adjustment towards provider’s profit. Thus it is a provider oriented SLA negotiation mechanism. The service performance parameters are also not considered.

By now, all of the above SLA negotiation methods are economy oriented, in the sense that they strive to achieve an expected pricing level. None of them are considering cloud service performance aspects such as network speed and application processing time. Considering cloud service performance parameters is a good way to approach new or inexperienced cloud users, who are concerned about the service quality more than the pricing aspect.

**Genetic Algorithm (GA) in the Internet or Cloud Services:** Genetic algorithm has been utilized by Campegiani [25] to optimize the allocation of virtual machines to the physical hosts. The more virtual machines and the less physical nodes occupied, the more profitable it is for the service provider. This optimization framework works with concurrent SLA negotiations, where a solution involving every SLA for every virtual machine. Each virtual machine has more than one SLA attached to it. The SLA takes into consideration the tier type of the physical node (e.g. web, database, or application), the number of available physical nodes, the number of CPU core, the memory (RAM) size and the planned profit. The best combination of SLA (highest fitness value) is the one with the highest number of free resources (host by host aggregate), which means that some other physical hosts are optimally occupied while the others are well preserved (efficiency). This SLA negotiation does not involve consumer’s concerns.

Wada, Suzuki, Yamano and Oba [26] develop a multi-objective genetic algorithm optimized service composition of SLA. This framework aims to find the cost of each service level (platinum, gold, silver) according to their generated throughput and latency. The optimized cost will assist in deciding how many hosts to procure or how many services to utilize certain hosts. However, this framework is provider’s economy oriented since it does not involve consumer’s perspective or request.

Hashmi, Alhosban, Malik and Medjahed [27] came up with an improved genetic algorithm for simultaneous web services negotiations. They introduced a new genetic algorithm operator called ‘norm’ that represents common value that specific people follow when doing negotiation. This improved method has outperformed the plain genetic algorithm and Hill-Climber based genetic algorithm. However, the fitness value of their method only mentions about the disagreement level between the client and providers; the lower the better it is. Their approach can be further more comprehensive if the fitness value involves specific performance parameters. Additionally, the genes included in their chromosome can also be applied to SLA negotiation if the SLA parameters are brought in.

**Proposed GA Optimized Performance Oriented Concurrent SLA Negotiations in Cloud Framework Proposed Framework and Methodology:** Our proposed framework will fill a research gap within concurrent SLA negotiation in cloud, where the optimized SLA is consumer oriented, in the case that it will consider service’s performance parameters as the fitness objective in our genetic algorithm methodology.

Our chromosome will consist of multiple service providers with their physical hosts’ specification (number of CPU cores, number of virtual CPU and RAM size). The fitness value is the recorded sample application processing speed. The recorded processing speed will be matched against the consumer’s requirement. A broker agent will send this chromosome of solution to the respective service providers, then each provider will set
its virtual machine (VM) specification according to their setting part in the chromosome and then the broker will send a sample application for every provider to process it on their set-up VM. The sample application processing time from each provider’s VM will be aggregated by the broker. This process will repeat according to the specified genetic algorithm properties (number of populations and number of generations). At the end of each generation, the best chromosome with the best fitness value (lowest processing time of sample application) is selected. Finally, the overall best solution is the chromosome with the best fitness value among all generations. Figure 1 depicts our proposed framework.

After the best chromosome (solution) has been selected, the consumer will look into the price for each provider’s generated SLA. Based on it, the consumer will choose their preferable provider, most probably the one with the lowest subscription fee.

The fitness \( f \) function of our application processing speed minimization is shown below in equation (1):

\[
f = \frac{1}{S}
\]

where \( S \) represents processing speed. Thus the lower the processing speed, the higher the fitness value.

**SLA Parameters to be Optimized by GA:** The SLA parameters to be optimized are based on the VM service parameters provided by the service providers, which are listed below:

- Number of CPU cores (cpu_core): the range of number of CPU core starts from 1 until 8, with the increment every 1 core.
- Number of virtual CPU (vcpu): it reflects the number of physical CPU in a real physical machine. The value is either 1 or 2.
- RAM size (ram): The allocated RAM size for the VM. It is ranging from 512 until 4096. The increment for this parameter is every 512.

In programming perspective, the formulas to generate the values of every gene above are stated below, with gene(x) is the variable of float random number generator used in our genetic algorithm methodology.

\[
\text{cpu_core} = (\text{gene(0)} \times 7) \times 1 + 1 \quad (2)
\]

\[
\text{vcpu} = (\text{gene(1)} > 0.5) \ ? 1 : 2 \quad (3)
\]

\[
\text{ram} = (\text{gene(2)} \times 7) \times 512 + 512 \quad (4)
\]

**GA Application’s Representation of Chromosome:**

Every parameter mentioned previously is a gene that makes a chromosome. There are 3 genes as described in the previous section B. Every of these 3 genes is the SLA for a service provider. The gene index itself starts from 0. Thus, for service provider A, the genes start from index 0 until 2. The gene indexes for service provider B start from 3 to 5 and so on for the next service providers.

**Discussion of the Proposed Framework:** Our proposed framework to establish a performance oriented concurrent SLA negotiations will fulfill consumer’s demand of quality of service (QoS), by selecting the provider that has the wanted quality, whether it is processing speed, network speed, or even the one with the safest temperature level.

A consumer oriented SLA negotiation will earn consumer’s trust and confidence to use the cloud service, because they know the selected provider will be based on the desired priority of qualities, for example if processing speed is more important than the network speed.
It will also assist the consumer to select the service provider in a more flexible manner, because the selected SLAs from all providers are presented, even though the one offering the lowest fee has higher chance to win the consumer’s subscription. It is mainly useful and efficient when there are numerous service providers to consider.

Although our proposed framework is performance based, but it does not completely exclude the economy aspect. The service fee of each provider is still presented after the selection process ends. And if the customer prefers the cheaper one, then the economical preference applies.

Furthermore, the fitness objective is not limited to application processing time as we initially proposed in section III.A, but another aspect can also be the fitness value, for example the network speed.

When it comes to network speed preference, the geographic locations of the service providers matter, since they affect the link vulnerability and the data transfer duration as well; the further the distance, the more vulnerable the link and the longer the transmission.

Network speed may also be preferred over computational speed, since there might be a provider with lower computational speed, but can transfer data faster than others. Therefore, the final selection is flexible.

There are some constraints in our methodology. The first one is when one or more providers go offline for some reasons (data center down, network disconnection, etc), then the genetic algorithm will not accomplish.

The second constraint is related to the local law where the provider’s data center resides. Some governments oblige the cloud providers to disclose their clients’ credentials, whenever they are asked to do so. This is required especially when a malicious hacking activity is suspected to be done through a cloud service. Such policy might concern cloud customers, who demand privacy reservation. For this issue, the policy complied in every data center can also be presented to the customers.

The third constraint is still related to the above issue, it concerns the current world rumor that certain countries are monitoring their Internet traffic. Such rumor might influence customer’s decision to select providers.

The fourth constraint is the web portal authentication security. The cloud service is accessible via a web portal, where user must login before entering the service. But it also means that the portal is open to potentially harmful hacking trials. The authentication might be vulnerable to brute force attack, SQL Injection, or any other hacking techniques.

**CONCLUSIONS**

The proposed framework in this paper is assumed to be suitable in a concurrent SLA negotiations situation, where the consumer is inexperienced or has never used a cloud service before. Additionally, it is more appropriate to be applied in a private cloud environment, where there are some internal cloud providers that can be chosen from.

The performance oriented feature of our framework is directed to calculate the specific quality of service of the available providers. Hence, it is suitable for users who are more concerned about quality of service rather than affordability.

Overall, our framework is aimed to simplify the multiple SLA negotiations and at the same time it matches which VM specification suits which application. Specifically it gives broader perspective (performance oriented) to the most of current concurrent SLA negotiation schemes, where they focus more on business profit. Moreover, the comfort of consumer’s decision and their trust towards the cloud are indirectly taken into place.

**Future Works:** Our future works will be to simulate or implement our framework. In a private cloud perspective, our framework can also be used to measure or to compare the power utilization of each cloud provider, by setting the power utilization as the fitness objective. Temperature measurement or comparison is also possible by querying the temperature from the VM and making it as the fitness objective.

**REFERENCES**


