Future Generation Computer Systems 41 (2014) 32–43

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Risk assessment in service provider communities

Ioan Petri [a,c,*], Omer F. Rana [a], Gheorghe Cosmin Silaghi [c], Yacine Rezgui [b]

[a] Cardiff University, School of Computer Science & Informatics, Wales, United Kingdom

[b] School of Engineering, Cardiff University, United Kingdom

[c] Babeş-Bolyai University, Business Information Systems, Cluj-Napoca, Romania

HIGHLIGHTS

- We investigate the notion of risk from the perspective of clients and providers.
- We determine how a service owner can balance the loss and the cost of replication.
- We discover that combining services in types can impact the level of profit/loss.
- We show how a demand for types can impact the overall community.

ARTICLE INFO

Article history:
Received 7 July 2012
Received in revised form
16 August 2014
Accepted 20 August 2014
Available online 27 August 2014

Keywords:
Risk management
Service Level Agreement
Fault tolerance
Multi-tenancy

ABSTRACT

Online service delivery undertaken between clients and service providers often incurs risks for both the client and the provider, especially when such an exchange takes place in the context of an electronic service market. For the client, the risk involves determining whether the requested service will be delivered on time and based on the previously agreed Service Level Agreement (SLA). Often risk to the client can be mitigated through the use of a penalty clause in an SLA. For the provider, the risk revolves around ensuring that the client will pay the advertised price and more importantly whether the provider will be able to deliver the advertised service to not incur the penalty identified in the SLA. This becomes more significant when the service providers outsource the actual enactment/execution to a data centre — a trend that has become dominant in recent years, with the emergence of infrastructure providers such as Amazon. In this work we investigate the notion of "risk" from a variety of different perspectives and demonstrate how risk to a service owner (who uses an external, third party data centre for service hosting) can be managed more effectively. A simulation based approach is used to validate our findings.

1. Introduction

With the emergence of Cloud computing it has become possible to differentiate between a software service owner (responsible for updating and managing a software capability encapsulated as a service) and an infrastructure provider (primarily offering computational, data and network resources that may be used to deploy the software service). A service owner can utilise the capability of one or more such infrastructure providers to offer the capability to clients, whereas an infrastructure provider looks for possible service owners to offer them managed access to resources, often at a pre-advertised price, at multiple capacities (small, medium and large instances in the case of Amazon.com, for instance) and with varying types of Service Level Agreements. Such a differentiation between the service owner and infrastructure provider roles is useful from a market perspective, as it enables different combinations of price–performance tradeoffs to be made available, thereby reducing the barrier to entry within a marketplace (as service owners no longer need to manage complex infrastructure which often incurs significant capital cost) whilst also allowing specialist infrastructure providers to emerge in the market.

Cloud and web applications experience huge and unpredictable variation in the load over time. Defining the required amount of instances to cope with the load experienced in a given moment can incur risks for both clients and providers. In few cases the load demand is known beforehand, thus users could reserve the required amount of instances — a situation which is cheaper than acquiring on-demand instances. However, as loads

[*] Corresponding author.
E-mail addresses: petrii@cardiff.ac.uk (I. Petri), o.f.rana@cs.cardiff.ac.uk (O.F. Rana), gheorghe.silaghi@econ.ubbcluj.ro (G.C. Silaghi), rezguiy@Cardiff.ac.uk (Y. Rezgui).

http://dx.doi.org/10.1016/j.future.2014.08.013
0167-739X/© 2014 Elsevier B.V. All rights reserved.

are unpredictable and variable, users have to combine reserved instances with on-demand instances as well as balance between cost and utilisation of the resources. A variance in the pattern of utilisation by a client gives the provider an opportunity to offer an on-demand option as a strategy to maximise their profit. Providers generally offer guaranteed availability based on a pre-agreed Service Level Agreement (SLA) [1] with a client.

It is therefore important to understand risk from a financial perspective (expressed as cost and profit) in order to enable service owners to successfully utilise the resources of an infrastructure provider. In addition, the problem of risk assessment and cost becomes increasingly important in the context of open markets where various providers can join and contribute computational capacity and where clients can place requests for various services [2].

The focus of this paper is to determine how a service owner can balance: (i) the loss in revenue incurred due to failure, with (ii) the additional cost of replication needed to prevent SLA violation, in a multi-tenancy environment. We investigate the problem of service outsourcing from a financial perspective in a multi-tenancy environment where a number of services can be combined and deployed over server farms. Determining the number of replicas to support service replication needs to be balanced with the revenue achieved through each service instance and the likely penalty that may arise due to unavailability (arising from a failure). Section 2 describes the motivation for this work evaluating risks from different perspectives. Section 3 presents the overall methodology we employ to analyse risk for single service outsourcing, extended in Section 6 to multiple service outsourcing where deployment can be across multiple server farms. Section 4 presents the simulation framework used for conducting the experiments. Sections 5 and 6.1.1 provide the evaluation of the work through a number of experiments carried out with the PeerSim simulator. Section 7 discusses related work in risk management and virtual appliances with a particular emphasis on financial risk. We present our conclusions in Section 8.

2. Motivation and approach

Utilising external infrastructure to deploy services incurs risks for both the service owner and the infrastructure provider. Our focus is primarily on financial risk, invoking the notion of uncertainty and randomness within an exchange between a client and a provider. Significant literature exists about the notion of risk in financial markets, with this term being used synonymously with the "probability of a loss or gain arising from unexpected changes in market conditions" [3]. Although in a financial market, risk is often associated with a change in the market price of a product or derivative, in the context of this work, we associate risk with the likely financial loss that a service owner or infrastructure provider will incur due to their inability to deliver an advertised capability. It is therefore necessary for the service owner to consider one of the following three options: (i) *trust* the infrastructure provider and assume a certain degree of fault tolerance and resilience; (ii) *establish a Service Level Agreement (SLA)* to ensure that if a provider is unable to deliver the advertised capability, the infrastructure provider incurs a financial penalty that must be paid to the service owner; (iii) *utilise resilience mechanisms* directly to ensure that any possible faults that may arise can be overcome through a pre-identified strategy, thereby ensuring continued, fault free operation for clients. In (i), when dealing with trusted participants the process is simplified as there are already a number of approaches to ensure correct service provisioning. Trust may be established based on prior interaction with an infrastructure provider or based on the general reputation of the provider within the marketplace. This aspect has been investigated previously by a number of researchers [4,5]. On the other hand, in the context of untrusted environments ensuring fault free operation can be difficult due to a variety of possible outcomes that may arise during operation. This scenario is particularly prevalent when these parties are unknown to each other and therefore the level of risk associated with the transaction is considerably increased. Expanding on the three considerations identified above, we have the following:

1. *Using trust mechanisms* — this is applicable when the environment is trusted and either: (i) clients and service providers have already interacted with each and have a history of prior (un)successful interactions; or (ii) clients and service providers have access to feedback from other entities they trust — or through an aggregated reputation service they can access. Reputation can either be based solely on prior transactions, or be considered as a multi-dimensional characteristic involving technology, business preferences and usage/business policy — and their combinations [6]. With (ii), the feedback data provided by others to calculate the reputation may be misleading and/or sparse — thereby limiting its benefit.

Hence, entities providing feedback can have different types of behaviours (both truth telling and deception), whereby feedback about a particular provider may be influenced by particular incentives that a client may have. By using existing trust mechanisms such a malicious intent (based on incorrect feedback) can bias the overall trust establishment within a community of clients and service providers and trust values may change with the number of clients involved in the community and with those providing feedback [7].

2. *Using Service Level Agreements* — this is applicable when the participants are unknown to each other – and therefore untrusted – with the behaviour of the participants being regulated through a previously agreed SLA. Such agreements can be particularly efficacious for mediating business transactions providing a useful reference point for monitoring the capability exchanged between a client and a provider (given that monitoring is carried out by either a trusted third party or through a pre-trusted component known to the client and the provider). An SLA may be used to specify Quality of Service (QoS) terms, the measurement criteria, reporting criteria and penalty/reward clauses between participants. Within an electronic market, an SLA may be used for: (i) an economic expression/proof of debts as well as credits — debts to the client and credits to the service provider; (ii) as a token of exchange between participants; (iii) as an identification of responsibilities of participants involved (such as the client and service provider). Establishing an SLA between two parties (client & service provider) implies that the service provider has agreed to provide a particular capability to the client subject to some QoS constraints. In return, the client must provide a monetary payment (most often) or credit (Bitcoins or other alternative currency) to the provider once the service has been delivered (subject to a penalty, often also monetary, in case the quality of service terms have not been adhered to) [8].

3. *Using fault tolerance techniques* — this is applicable when dealing with unknown participants whose behaviour cannot be predetermined. Although a client (the service owner) may have an SLA with the provider, the client may still wish to minimise risk by ensuring that suitable fault tolerance strategies are available. For instance, establishing SLAs with entities that may exhibit faulty behaviours may represent a high risk. In order to mitigate these risks we propose a fault tolerance mechanism where various services are replicated among a number of peer nodes.

In the context of service provision, fault-tolerance has moved from hardware to software, making failure a "normal" event that has to be managed efficiently. Referring to hardware failures within a cluster of 1800 servers that Google uses as the building