J Internet Serv Appl (2012) 3:173–193
DOI 10.1007/s13174-011-0050-y

ORIGINAL PAPER

On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure

Basem Suleiman · Sherif Sakr · Ross Jeffery · Anna Liu

Received: 26 April 2011 / Accepted: 13 November 2011 / Published online: 1 December 2011

Abstract The exposure of business applications to the web has considerably increased the variability of its workload patterns and volumes as the number of users/customers often grows and shrinks at various rates and times. Such application characteristics have increasingly demanded the need for flexible yet inexpensive computing infrastructure to accommodate variable workloads. The on-demand and per-use cloud computing model, specifically that of public Cloud Infrastructure Service Offerings (CISOs), has quickly evolved and adopted by majority of hardware and software computing companies with the promise of provisioning utility-like computing resources at massive economies of scale. However, deploying business applications on public cloud infrastructure does not lead to achieving desired economics and elasticity gains, and some challenges block the way for realizing its real benefits. These challenges are due to multiple differences between CISOs and application's requirements and characteristics. This article introduces a detailed analysis and discussion of the economics and elasticity challenges of business applications to be deployed and operate on public cloud infrastructure. This includes analysis of various aspects of public CISOs, modeling and measuring CISOs' economics and elasticity, application workload patterns and its impact on achieving elasticity and economics, economics-driven elasticity decisions and policies, and SLA-driven monitoring and elasticity of cloud-based business applications. The analysis and discussion are supported with motivating scenarios for cloud-based business applications. The paper provides a multi-lenses overview that can help cloud consumers and potential business application's owners to understand, analyze, and evaluate important economics and elasticity capabilities of different CISOs and its suitability for meeting their business application's requirements.

Keywords Cloud computing · Cost · Elasticity · Scaling · Economics · Business applications · SLA · Cloud infrastructure service offerings · IaaS

1 Introduction

The significant advancement in several computing technologies including virtualization, grid computing, utility computing, and autonomic computing has led to new ways of offering and consuming hardware and software resources as services under what is now commonly known as the cloud computing paradigm. It has been identified amongst Gartner's top 10 most disruptive technologies for 2008 to

B. Suleiman · S. Sakr (✉) · R. Jeffery · A. Liu
Software Systems Research Group, National ICT Australia
(NICTA), Sydney, NSW, Australia
e-mail: Sherif.Sakr@nicta.com.au

B. Suleiman
e-mail: Basem.Suleiman@nicta.com.au

R. Jeffery
e-mail: Ross.Jeffery@nicta.com.au

A. Liu
e-mail: Anna.Liu@nicta.com.au

B. Suleiman · S. Sakr · R. Jeffery · A. Liu
School of Computer Science and Engineering,
University of New South Wales, Sydney, NSW, Australia

S. Sakr
e-mail: ssakr@cse.unsw.edu.au

B. Suleiman
Business Network Orchestration Research Practice,
SAP Research, Sydney, NSW, Australia

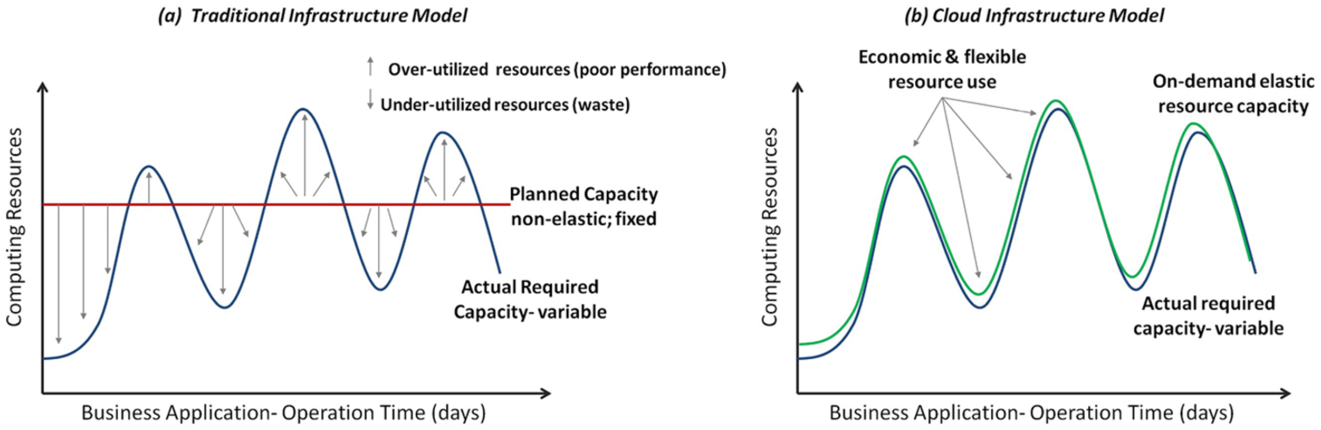

Fig. 1 Cloud infrastructure elasticity and economics illustrated (adapted from [40])

2012 [15]. US Government's National Institute of Standards and Technologies (NIST) defines cloud computing as "*a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort*" [32]. The definition also describes essential characteristics of cloud computing including "*Rapid Elasticity*" of computing resources and "*Measured Service*" of computing resource usage. NIST [32] also distinguishes between three main types of computing services namely; *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)* and *Software as a Service (SaaS)*. It further describes four cloud deployment models; *public*, *private*, *community,* and *hybrid* clouds.

Due to its huge business potential and opportunities, the cloud computing business model, especially that of IaaS, has attracted a large number of organizations including *Amazon*,[1] *Rackspace*,[2] *GoGrid*,[3] *Joyent*,[4] and *ElasticHosts*,[5] to provide computing infrastructure services. We call such organizations *public Cloud Infrastructure Service Providers (CISPs)* and their offerings *public Cloud Infrastructure Service Offerings (CISOs)*. Cloud consumers, e.g., small to medium business organizations, could largely reduce operational costs and increase business flexibility and agility when deploying their applications on (*CISOs*). However, gaining such benefits is restricted by several challenges such as security, interoperability, vendor/technology lock-in, elasticity, and economics [3, 38].

The focus of this article is on investigating the *economics* and *elasticity* challenges of *internet-based business (or e-business) applications* that are deployed on public cloud infrastructure. In this context, the term *economics* refers to the efficient use and management of cloud infrastructure resources required to run *e-business* applications with desired performance levels. The term *elasticity* refers to the ability to dynamically grow and shrink computing infrastructure resources through automatic mechanisms over the internet in order to serve variable workloads of *e-business* applications efficiently [32].

Figure 1 illustrates the *elasticity* and *economics* concepts in terms of computing infrastructure resources. Traditionally, business organizations had planned their computing infrastructure based on maximum expected computing resource capacity (i.e., fixed computing capacity as depicted in Fig. 1(a)). Given today's dynamic and agile changes in business needs and growth, such capacity planning has to be more flexible and economical for two main reasons. First, traditional infrastructure capacity planning involves very large upfront capital investment which could reduce organization's cash flow considerably and it has a very long payback period. Second, such large computing capacity cannot be efficiently utilized. As illustrated in Fig. 1(a), there are time periods where resources are *under-utilized*. In addition to such resource waste, there are additional *on-going costs* which are important to maintain computing infrastructure operational and healthy (e.g., physical space, electricity power, management services, maintenance, etc.). Figure 1(a) also highlights some time periods where required application's computing capacity exceeds its planned capacity. Such scenario could occur because of *unexpected workload spikes* and/or *business growth*. Inability to meet such unexpected and dynamic computing capacity often leads to customer frustration and negative impact on organization's reputation and as a result potential lose of profit and customers [26]. Cloud computing, particularly IaaS model, can highly enable achieving efficient and resilience utilization of computing infrastructure resources (as illustrated in

[1] http://aws.amazon.com/.

[2] http://www.rackspace.com/.

[3] http://www.gogrid.com.

[4] http://www.joyent.com/.

[5] http://www.elastichosts.com/.

Fig. 1(b)). In particular, with IaaS model required computing resource capacity is dynamically launched when and as-much-as needed (*elastic computing infrastructure*) and computing resources are charged based on its usage time only (*economic*; no huge upfront capital and on-going costs). As a result, *under-utilization* and *over-utilization* scenarios in traditional capacity planning can considerably be reduced.

Unlike *PaaS*, deploying and planning capacity of *e-business applications* on *public cloud infrastructure* using *IaaS* model does not require major architecture redesign and coding and, therefore, allowing business organizations to focus on their core business competitive advantages. *Business applications* can derive immense benefits from: (a) massive computing resources at hourly usage costs with no upfront payments or long term commitments [3] and (b) on-demand dynamic computing resources elasticity [32]. We further focus on these internet-based *transactional business applications* such as *Amazon Web store*,[6] *Ticketek,*[7] and Customer Relationship Management (CRM) but not the *analytical* ones. Such class of applications often has fluctuating workload patterns and volumes because it serves a wide range of users/customers that are distributed across several locations where each would have different requirements. Failure to accommodate such variable financial transaction volumes would directly affect critical business metrics of the business such as profit, customer satisfaction, and company reputation. Such factors emphasize the significance of economics and elasticity characteristics of *CISOs* for *transactional business applications*.

The *economics* and *elasticity* benefits are not implicit and require in-depth analysis and investigation from cloud consumer perspectives. This is because there is a large number of *CISOs* offered by different *CISPs* which are differentiated based on computing resources types, capacity and specifications, pricing models, service classes, licensing support, to name but a few. This article analyzes the various aspects of *CISOs* and critically investigates its impact on the economics and elasticity of *transactional business applications* on public cloud infrastructure. Particularly, it identifies *CISOs'* issues that hinder the ability of achieving economics and elasticity requirements of transactional business applications. Based on this analysis, significant research challenges that relate to achieving economics and elasticity requirements are derived and then discussed. The analysis and discussion provide a sound understanding for potential cloud consumers, particularly transactional business application's owners, such that they are better able to identify, evaluate, and decide on relevant *economics* and *elasticity* challenges that may relate to their applications when they are deployed and operated on *public* cloud infrastructure services.

2 Motivating scenario: economics and elasticity of MyShop e-business application

Let us consider an online shopping application, *MyShop*, which sells wide range of products. *MyShop* has three tiers architecture that consists of web server/load balancing, application server and database server. *MyShop's* capacity management team has classified expected workload pattern and required basic computing resources as follows:

- *Normal operation workload*: 1 web server/load balancer, 2 application servers and 1 database server during all business operation times.
- *Mid-Week-Sales workload*: same as normal operation workload plus 2 additional application servers and 1 additional database server works as a slave from 11 am to 9 pm on Wednesday.
- *Weekend-Sales workloads*: same as normal operation workload plus 4 additional application servers and 2 additional database servers work as slaves from 10 am to 10 pm each on Saturday and Sunday.

To illustrate the key *economics* and *elasticity* gains that *MyShop* can achieve, we calculate the costs of running *MyShop* application's tier based on three real cloud server offerings which are shown in Table 1. The names of CISPs and their cloud server types have been made *anonymous* as the goal of this example is to illustrate the *economics* and *elasticity* benefits for *MyShop* but not to evaluate and compare CISPs' offerings. We focus our calculations on the application tier because it has *variable workloads*. The *normal operation workload* requires 2 fixed application servers for the whole year. So, *subscription-based* cloud servers option is very *economical* here as most CISPs provide it at *discounted* hourly prices for one year term. *On-demand* cloud server offerings, on the other hand, are *economical* option for *mid-week* and *weekend sales workloads* as it is billed on *hourly-basis* without any long-term commitments. The yearly costs of both workloads are calculated using the following formulas:

$$\begin{aligned}&\textit{Fixed server costs (yearly)}\\&\quad= \textit{no. of app. servers} \times \textit{\$server subscription price/year}\\&\textit{On-demand server costs (yearly)}\\&\quad= \textit{no. of additional servers} \times \textit{no. of usage hours}\\&\qquad \textit{per server/week} \times \textit{\$on-demand server price/hour}\\&\qquad \times 52\ (\textit{weeks})\end{aligned}$$

Table 2 summarizes the *subscription* and *on-demand costs* of *MyShop* application servers' workloads for one year. *CPU-intensive* servers have been selected here as application servers perform most of the business logic process-

[6] http://amazon.com/.

[7] http://premier.ticketek.com.au/.

Table 1 Different cloud server offerings of selected CISPs

CISP	Selected cloud server offerings	On-demand price	Subscription price
CISP1	Server 1- CPU-intensive (virtualized)	$0.17/hour	$455/year
	Server 2- memory-intensive (virtualized)	$0.34/hour	$910/year
	Server 1- CPU-intensive (dedicated)	$0.21/hour	$520/year
	Server 2- memory-intensive (dedicated)	$0.42/hour	$1120/year
CISP2	Server X- CPU-intensive (virtualized)	$0.24/hour	$3300/year
	Server Y- memory-intensive (virtualized)	$0.48/hour	$6450/year
CISP3	Server K- CPU-intensive (virtualized)	$0.18/hour	$1296/year
	Server M- memory-intensive (virtualized)	$0.38/hour	$2718/year

Table 2 Yearly cloud server costs of MyShop application's tier

CISP	Selected cloud servers	On-demand costs (wed.+Sat.-Sun workload)	Subscription costs (normal workload)	Total costs
CISP1	Server 1(virtualized)	$176.8 + $848.64	$910	$1943.44
	Server 1(dedicated)	$218.4 + $1048.32	$1040	$2306.72
	Server 1(virtualized) + Server 1(dedicated)	$176.8 + $848.64	$1040	$2066.44
CISP2	Server X(virtualized)	$294.6 + $1198.08	$6600	$8047.68
CISP3	Server K(virtualized)	$187.2 + $898.56	$2593.66	$3679.42

ing at very large volumes.[8] Based on Table 2, the following observations highlight some of the key economic factors that are important for MyShop's application tier decision-making:

- The costs of CISP1's *virtualized on-demand* and *subscription* cloud servers are *cheaper* than its *dedicated on-demand* and subscription ones, respectively. However, the *dedicated* cloud servers are often more reliable and perform better than virtualized servers. Combining CISP1's *virtualized on-demand* server costs with its dedicated subscription server costs will provide optimal performance-cost benefits.
- The costs of *virtualized on-demand* and subscription cloud servers' of CISP1 are the *cheapest* option (and therefore the total server costs). However, there are some differences between offered cloud servers in terms of included hardware resources in each server (e.g., processing, memory, storage, and network) and its types and capacity as well as included application and system software. Such differences will increase the complexity of evaluating and understanding the economics of different cloud server offerings.
- CISP1 is the *only* provider who offers *on-demand dedicated* cloud servers. Given the performance and reliability of *dedicated* servers, the usage costs of these *on-demand dedicated* servers are very competitive (cheaper than CISP2's *virtualized on-demand* servers).
- While CISP2's *subscription* cloud server costs are the *most expensive*, the server price is packaged with *add-on features* as a competitive advantage; 1 year server and application management services such as monitoring, application/OS support and specialist support. Even when such *add-on services* are added to the other CISP's cloud server costs the total cost differences are likely to remain large.
- CISP2's *on-demand* server costs are competitive compared to the other CISP's *on-demand* server costs. However, the total server costs of CISP2's are significantly influenced by its *subscription* server costs. This indicates that in such scenario major savings can be first made in *subscription-based* servers. In case there is significant variability in the application workload for long periods of times, the *on-demand* server costs could become a *determining cost factor* of the overall costs. Furthermore, when all required cloud infrastructure resources to run *MyShop* application are included there might be one or more other *cost factors* that could influence the total costs.
- Given the cost and performance differences between CISP's offerings, two offerings can be combined to achieve cost-performance balance. For example, CISP1's *subscription-based dedicated* servers (reasonable performance and cost) can be combined with *on-demand* cloud servers of CISP2 (reasonable performance and cost).
- Despite the hourly price differences of CISP's *on-demand* cloud servers are minor (few cents in most cases—see Table 2), the *on-demand* cloud server costs have become

[8] For MyShop's database tier *memory-intensive* cloud servers must be used as all data operations and retrieval are carried here. Note *memory-intensive* cloud servers are more expensive than *CPU-intensive* ones and, therefore, its economics will vary.

more noticeable when the workload and number of *on-demand* usage hours have increased. The issue will even make more *economic-sense* when other tier's servers and infrastructure resources are included. For example, the costs of *memory-intensive* cloud servers and the cost differences between CISPs are higher than those ones used for the application servers.

- Adding and removing *on-demand* cloud servers when is needed at hourly usage prices shows the high degree of computing *infrastructure elasticity* of *MyShop* application. Such infrastructure resilience have significantly reduced the total server costs (instead of paying for 6 cloud servers for the whole year). The cost reduction would also vary from one CISP to another (as illustrated in Table 2). In Sect. 4.3, we further illustrate *economics of different elasticity strategies* for *MyShop* application.
- The costs of *MyShop's* database cloud severs (*on-demand* and *subscription*) and all required cloud computing infrastructure resources (e.g., load balancing, network bandwidth, additional storage) will also have important *economic point of views* as CISPs use different offering packages, specifications and pricing models that have to be understood. In the following section, the key aspects that influence *economics* and *elasticity* of *e-business* applications will be discussed.

3 Cloud infrastructure service offerings (CISOs)

As cloud infrastructure services become increasingly important, a considerable number of Cloud Infrastructure Service Providers (*CISPs*) has emerged with different Cloud Infrastructure Service Offerings (*CISO*) (e.g., *Amazon*, *GoGrid*, *Rackspace*, etc.). We have investigated different *CISOs* of many *CISPs* in terms of: (a) Computing Resources Bundling and Specifications, (b) Pricing Models and Offering Types, (c) Software and System Licensing, (d) Elasticity Support for Infrastructure Resources, (e) locations of *CISOs*. We discuss the various related aspects of each point which focuses on the economics and elasticity aspects which are central for this article.

3.1 Cloud resources bundling and specifications

CISPs offer different cloud infrastructure service bundles such as cloud servers, cloud storage, and internet/network resources. Cloud server bundles are the core service offering as it offers processing capabilities and, therefore, *CISPs* offer them at different fine-grained levels which combine different computing resources such as processing unit, memory, disk and/or network bandwidth. A variation in one or more resources' capacity in a bundle results in what is called server *instance*, *class,* or *size*. Unlike most *CISPs* who have specific number of instance offerings, *ElasticHosts*[9] and *Cloud-Sigma*[10] allow its cloud consumers to customize their cloud servers by varying CPU, RAM, disk and data transfer/bandwidth capacity at very fine-grained levels.

Cloud resources packaging and instance types are important aspects of *CISOs* as it could influence an application's economics and scalability on a public cloud infrastructure. Some cloud server bundles are restricted to a certain computing resource which vary between *CISPs*. For example, *Terremark vCloud*[11] and *FlexiScale Server*[12] are packaged in terms of CPU and RAM resources only. Therefore, if an application workload has variable CPU and I/O requirements, then adding new cloud server instances will only scale the CPU and RAM resources and additional network resources has to be rented and managed separately. *Joyent SmartMachines* and *Rackspace Cloud Servers* instances are more economic in this case as their cloud server bundles include more resources, i.e., CPU, RAM, disk and bandwidth. At the same time, more bundled computing resources could be restrictive for certain application's workloads. For example, *Terremark's vCloud* would be more economical than *Rackspace Cloud Server* bundles for memory-intensive application workload which often requires scaling a RAM resource rather than the whole resources involved in the bundle. Therefore, cloud consumers are restricted with bundled resources in server instances and they are not able to only scale certain resource.

Another related issue to the granularity of the bundles is the ability to choose the most appropriate ration of resource types in a bundle. Computing resources' capacities included in service bundles often have similar proportions. Depending on the workload characteristics, some applications require more resource capacity proportion than other resources. For example, database and memory caching applications, e.g., social networking applications, require more RAM capacity in proportion to CPU capacity. Amazon offers different cloud server instances such as *High-Memory* with proportionally more memory than CPU capacity and *High-CPU* with proportionally more CPU capacity than RAM capacity. However, Amazon offers only three instances at a low level of granularity; the resources' capacity of the minimum instance is too high (17.1GB, 6.5CPU, 420GB storage) and each instance is almost double of the previous instance capacity. Therefore, such instances do not support fine-granular elasticity and cloud consumers are likely to pay for unused resources' capacity. In this regard, *ElasticHosts* and *CloudSigma CISOs* are more flexible in terms of their support for resource sizing as they al-

[9] http://www.elastichosts.com/cloud-hosting/pricing.

[10] http://www.cloudsigma.com/en/pricing/price-schedules.

[11] http://vcloudexpress.terremark.com/pricing.aspx.

[12] http://www.flexiant.com/products/flexiscale/pricing/.

low their cloud consumers to customize any combination of cloud server instances' resources capacity according to their needs.

In the context of this analysis, evaluating the impact of resources bundling and sizing on application's economics and elasticity is a vital challenge for cloud consumers. Particularly, different classes of applications often have different workload patterns and characteristics [4, 6] and, therefore, its resources' elasticity requirements will vary accordingly. Ideally, highest levels of economic elasticity can be achieved by enabling cloud consumers to customize any combination of computing resources' capacity *on-demand* and as much as their application workloads require. However, it is impossible to enforce *CISPs* to change their *CISOs* and therefore the issue has to be addressed from the cloud consumer's side. Cloud consumers need models and metrics to measure the level of customizability of different *CISPs'* computing resources bundles and capacity, and evaluate its impact on scaling flexibility and cost effectiveness of resource utilization based on their workload characteristics.

Another important issue that is related to *CISOs* is the different metrics used by *CISPs* to express the capacity/capability of their offered computing resources. Furthermore, the specification, type and virtualization/allocation policy of computing resources often vary from one *CISP* to another. For example, while Amazon expresses processing power in terms of the number of *Elastic Compute Cloud (EC2) Compute Units (ECU)*, GoGrid and Rackspace use the number of *Virtual Cores*. An Amazon *ECU* provides the equivalent CPU capacity of a 1.0–1.2 GHz 2007 Opteron or 2007 Xeon processor.[13] GoGrid does not provide capacity details of their cores. It specifies that Xen-based hypervisor technology is used to virtually allocate RAM and CPU cores on Intel-based computers. The maximum number of CPU cores that can be allocated to each virtual server is equivalent to the server size expressed in RAM size, e.g., 1GB, 2GB, 4GB, etc. Unlike GoGrid, Rackspace's cloud servers utilizes the Quad-Core AMD Opteron processor type. The number of assigned virtual cores is determined by the server size; e.g., 1GB cloud server gets 1 virtual core, 2GB and 4GB cloud servers get 2 virtual cores. The amount of CPU cycles allocated to each core is weighted for Linux distribution whereas CPU cycles allocation for a Windows distribution has equal weight. Similarly, other computing resource offerings, e.g., disk storage and storage services, network bandwidth and IP addresses, etc., have differences in allocation way, hardware types/structure and/or capacity metrics.

The above analysis and examples highlight the following research challenges to cloud consumers:

- *How to benchmark performance of CISP's cloud servers?* This includes investigating types and specifications of CPU used in cloud servers and its performance (see [34] for comprehensive list of CPU benchmarks), virtualization technology employed, resource allocation policies, and conversions between different capacity metrics of key *CISP's* cloud servers. Investigating such aspect is crucial to perform an "*apples-to-apples*" comparison in terms of performance and costs. This would also vary from one application workload to another (e.g., CPU-intensive, memory-intensive, etc.). Given the vagueness of computing resources' specifications and measurement units, performance evaluation is crucial for cloud consumers as it supports their decision making on selecting the best performing *CISP* for their application workloads.
- *How to evaluate elasticity of CISP's cloud servers?* Again evaluating cost/performance is also crucial for cloud consumers given the differences in capacity metrics, hardware specifications, and resource allocation policies. This also would depend on application workload characteristics. Like performance benchmarking, this requires investigating standardizing capacity measurement units of used hardware resources in order to decide on the most economic elasticity among *CISPs* for particular application workloads.

3.2 Pricing models and offering types

During our investigation of various *CISOs,* we observed four types of pricing models which are correlated with certain offering types. The following analysis discusses the key aspects of each pricing model.

1. *Per-use model*: It is also known as *pay-as-you-go* where computing resources are bundled and billed per unit of time usage. This model is most commonly used by *CISPs* for pricing cloud server instances in which prices are often varied based on CPU, RAM, disk storage and/or bandwidth capacity (it varies between *CISPs*). Other computing resources are similarly billed at usage-based but per quantity/amount per unit of time. Examples of this pricing type include $ per GB input/output data transfer, $ per GB data storage per period of time and per IP address usage per unit of time. The *per-use* model is simple and does not require any upfront payment and/or long-term commitments and computing resources can be requested anytime and therefore it is called, *on-demand* or *cloud servers*.
2. *Subscription model*: In this model, cloud consumers subscribe in advance for computing resources usage for a specific period of time by signing a contract/agreement. Computing resources are grouped into different packages often called *Dedicated Servers* or *Reserved Instances* in which prices vary according to included resources' capacity. Unlike *pay-as-you-go*, the *subscription* model requires upfront payment and long-term/short-term com-

[13] http://aws.amazon.com/ec2/instance-types/.

Table 3 Classification of pricing models and cloud server offering types

Pricing model	Offering type	Commitment
Per-use	On-demand servers ($ per hour use)	Nil
	Examples: Amazon on-demand and spot instances, Rackspace cloud servers, Terremark vCloud (per-hour)	
Subscription	Dedicated servers (upfront $ per time period)	Short-term (less than 6 months) and Long-term (1–3 years)
	Examples: GoGrid dedicated servers (monthly), Joyent Smart-Machines (monthly), Rackspace servers (monthly)	
Prepaid per-use	On-demand servers ($ per hour use deducted from prepaid credit)	Nil
	Examples: ElasticHosts hourly-burst cloud servers, GoGrid cloud servers (hourly), Joyent SmartMachines (daily)	
Subscription+per-use	Dedicated servers (upfront $ per month/year) + on-demand instances $ per hour use)	Short-term (less than 6 months) and Long-term (1–3 years)
	Examples: ElasticHosts monthly cloud servers+ hourly usage, Joyent monthly SmartMachines + daily usage, Amazon reserved instances (1 or 3 years)	

mitment, ranging from monthly to yearly. *Dedicated server* packages are often offered at discounted rates.

3. *Prepaid per-use model*: it is a variation of the *per-use* pricing model. In this model, *on-demand* servers are billed hourly but from a prepaid credit.
4. *Subscription + per-use model*: It is an intermediate model between *per-use* and *subscription* models. In this model, *Dedicated Servers* must be rented in advance for a period of time and additional *cloud servers* can be requested *on-demand* and billed at *per-use* charges.

Table 3 summarizes a classification of the four pricing models. A *subscription* model is usually cheaper than a per-use model as long as the application workload is constant. Joyent's *daily-usage SmartMachines* are an intermediate solution, it is also cheaper than *on-demand hourly-usage* servers. *Dedicated Server* offerings are often physical machines and *on-demand* servers are often virtualized. Physical machines are often more reliable as it does not rely on dynamic resource scheduling and sharing like *on-demand* ones. In the *prepaid per-use*, prepaid credit must not go below a certain limit and some *CISPs* such as *ElasticHosts* may not refund unused credit but they still charge their consumers on a *per-use* basis. The *subscription + per-use* combines the advantage of discounted *Dedicated Servers* which is fit for continuous and stable fixed workload and the availability of *on-demand* instances for variable application workloads. Some pricing parameters are also used to differentiate the offerings. Most of cloud server offerings of all pricing models differentiate between Windows and Linux/Unix servers; Windows servers are often more expensive as Linux/Unix systems often have open source licenses which does not incur any upfront costs to purchase and install. Most *CISPs* allow cloud consumers to customize their cloud servers with different software applications and server instance prices are adjusted accordingly. Amazon offers most of its cloud servers at three main regional areas and varies the prices accordingly.

Given the characteristics of the pricing models and parameters discussed above, one key challenge that will often confront cloud consumers is *which offering could be the most economical and elastic?* The answer will be highly dependent on the application workload patterns and characteristics, in addition to other factors. Table 4 represents a general classification of four main types of application workloads and refer to the most suitable pricing models and offering types for each pattern in terms of economics and elasticity. The table highlights an important point; there is no *one-size-fits-all* pricing model or offering type that would suit various application workload patterns. The *variable* and *unpredictable* workload can highly benefit from the cost effectiveness of *per-use* pricing and *on-demand* offerings which allows the addition/removal of cloud servers on an hourly and/or daily basis. The *fixed* workload can always benefit from discounted *dedicated server* rates. The *fixed workload with predictable spikes* benefits from both reduced *dedicated servers* and hourly *on-demand* servers and the high elasticity of the latter.

Making appropriate decisions about which offering, or combination of offerings, would achieve the most economical and elastic solution is a more challenging task than it is generally highlighted and the following issues remain unanswered:

- How to determine fixed workload periods and variable ones in the context of *CISO* types and pricing models?
- Which offerings/combination of offerings, dedicated and on-demand, would best suit each workload period in terms of economics and elasticity?
- Is there a point where the cost of the overall on-demand elasticity would be equivalent to the cost of a dedicated

Table 4 Workload patterns and economics and elasticity of pricing models and offerings

Workload pattern	Economics of pricing models	Elasticity of offerings
Continuously fixed workload	Subscription – discounted monthly/yearly rates	Nil. fixed cloud resources' capacity monthly/yearly dedicated servers
Variable workload with variable volumes	Per-use/prepaid per-use – almost as much as needed computing resources Capacity to avoid over or under provisioning scenario	Very high elasticity – as much as needed resources on-demand (scale up/down or out/in) hourly/daily
Fixed workload with predictable spikes	Subscription + per-use – avoid over/under-provisioning for the predictable spikes	High elasticity for predictable spikes using hourly on-demand servers to meet predictable spikes on-demand
Unpredictable workload	Prepaid per-use + per-use	Very high elasticity – (daily+hourly on-demand servers)

server during a certain period of time? If there is one, how to determine this point?

3.3 Software and system licensing

CISPs customize cloud server instances with different software and system configurations. There are two main types of software; Operating Systems, e.g., range of Windows and Linux/Unix, and Application Software, e.g., Oracle Web Logic, MySQL Enterprise and Apache HTTP. *CISPs* differ in the type and number of software and system they offer with their cloud server instances. *Amazon EC2 instances* and *Joyent SmartMachines* are offered with many pre-configured and charged software and system configurations. Some *CISPs* such as *Terremark*, *GoGrid* and *Amazon* also allows customizing their blank cloud server instances with a wide range of applications software and system applications. Some CISOs such as *Joyent's SmartMachines* have an added-value configuration feature called *SmartOS*. Unlike other virtual OS configurations, SmartOS enables as-needed access to a large pool of available resources while still providing each SmartMachine with minimum guaranteed access to resources based on a pre-established fair share schedule. Such configuration enhances CPU, memory and I/O optimization and therefore SmartMachine's performance.[14] One important issue with software and systems is the *licensing model* which is not compatible with cloud computing business model, i.e., *per-use* pricing [3]. Recently, many software vendors including *Oracle*, *IBM,* and *Microsoft* have provided software licensing that support cloud resources usage. Some CISPs offer different licensing options for certain software/system applications. For example, Amazon provides three different licensing options for running *MySql*, *Oracle* database, and *IBM DB2* on Amazon Relational Database Service (RDS)[15]:

1. *Bring Your Own Software License (BYOSL)*: Customers can bring their own license to Amazon RDS with no additional software licensing or support charges.
2. *On-demand DB instances*: Customers are charged per-hour license use per RDS DB instance running Oracle DB.
3. *Reserved DB instances*: Customers pay one-time prepaid charge per RDS DB instance to get reduced hourly-usage rate.

Unlike *on-demand* and *reserved licensing*, *BYOL* puts additional management overhead on cloud consumers. This is because adding/removing cloud server instances is easy to automate and launch dynamically, software applications could be launched on instances without having proper license or license thresholds are reached; e.g., maximum number of simultaneous/concurrent users or CPU is exceeded. Application workload pattern is one factor that could influence licensing management. Software licensing issues could have a direct impact on the application's economic value, i.e., penalties/additional licensing fees, and elasticity, restricting the number of servers to be launched or concurrent/simultaneous. Some CISPs offers licensing as added-value services included with their cloud servers. For example, GoGrid includes Windows Server 2003 and 2008, and Red Hat Enterprise Linux licensing fees for free with each account.[16] Compared to Amazon licensing charges, GoGrid's free licensing add-on feature reduces the licensing and management costs and complexity of running instances. Regarding software and system licensing, we raise the following issues that face cloud consumers:

- Given the different licensing options offered by CISPs, which licensing model would best suit certain application workload?
- How licensing models on cloud server instances would impact the application's economics and scalability?

[14] http://www.joyent.com/products/smartmachines/.

[15] http://aws.amazon.com/rds/.

[16] http://www.gogrid.com/cloud-hosting/cloud-hosting-pricing.php.

- How to monitor and control different types of software and system licenses on all running server instances?

3.4 Elasticity support for cloud infrastructure resources

CISPs often relate elasticity with different types of computing resources. Common elasticity examples include:

- Adding/removing server instances or resizing server capacity by adding/removing additional CPUs and/or RAMs.
- Increasing/decreasing storage capacity by adding/removing additional disks or virtual storage.
- Increasing/decreasing network speed and number of IP addresses.
- Increasing/decreasing amount of data transfer and number of data operations/requests.

CISPs offer several granular classes/instances, in terms of capacity, for their computing resources offerings to allow cloud consumers to not only be able to select computing resources that meet their application needs but also to scale the computing resources at any time. Therefore, elasticity is not implicitly offered with *CISOs* and it becomes the responsibility of cloud consumers to scale their computing resources. *CISPs* often have large numbers of customers of different application workloads and requirements and it is impossible for *CISPs* to cater for all customers' workloads. Alternatively, *CISPs* provide additional services, tools and Application Programming Interfaces (*APIs*) to support scaling their computing resources and to help cloud consumers to manage rented computing resources. Examples of such services include:

- *Amazon's Elastic Load Balancing* [41]: a service to distribute application's requests across different server instances.
- *GoGrid's server configuration tool* [16]: to increase/decrease RAM capacity of a cloud server.
- *Rackspace CloudKick Monitoring and Dashboards* [22]: tools to monitor, control, and visualize different cloud infrastructure resources metrics that support automated scaling techniques.
- *Amazon Auto Scaling* [1] and *RightScale Autoscaling* [37]: tools to automate scaling cloud infrastructure resources based on cloud consumers' configurations.

RightScale is an example of a *third-party Cloud Infrastructure Management Provider (CIMP)*. It provides tools for automated deployment, monitoring and auto-scaling for Amazon, GoGrid and Rackspace cloud infrastructure resources. Although such tools can improve elasticity automation and management, it adds considerable costs to the resource billing as it incur additional service charges.

Due to the significance of elasticity it is important to understand key factors that could smooth or constrain the ability to achieve required levels of elasticity. Tools, services, and APIs to support scaling are prime examples. Clearly, tools and mechanisms can enable the ability of specifying elasticity policies and automating its enactment to achieve near on-time elasticity. One important point that is related to the *Amazon Auto-Scaling* tool, and accordingly *RightScale Autoscaling*, is that it does not allow the launching of more than 20 server instances unless an *increase limit* form is submitted. *Terremark* allows running up to 60 cloud servers under one account and it requires contacting its sales people if more than 60 cloud servers are needed. Such condition will restrict cloud consumers from achieving flexible automated cloud infrastructure elasticity especially for unexpected very large workload increases. Therefore, it is important for cloud consumer to know the maximum number of servers they can run with their CISP and its impact on achieving automated elasticity and potential scaling delays.

Scaling speed is also an important elasticity factor that could smooth/restrain the ability for achieving the elasticity requirements. Ability to respond quickly to application workloads is significant as it is likely to have a direct impact on customer satisfaction and potential profits [3]. Amazon claims less than 10 minutes from the initiation of the execution "*RunInstances*" command until all instances begin their boot sequences. However, the exact initiation time is still undetermined as Amazon further states that it is dependent on a number of factors including the instance size, number of instances to be launched and how recently those instances have been launched (first time instances take longer to boot). Li et al. [29] measured *allocation latency* and *booting latency* of the smallest instances, both Windows and Linux OS, for three providers including Amazon server instances. They found that all providers have average allocation latency below the 10 minutes and Windows instances take longer than Linux instances to be created and/or booted. Collecting metrics such as average booting times and its variability over time/locations would be useful for performance modeling of cloud server instances.

Another important factor that could ease/restrain the ability for achieving the elasticity requirements is the bundling of resources capacity and granularity of its instances capacity. Unlike most other providers, *Terremark* and *FlexiScale* have the least number of bundled resources, i.e., CPU and RAM only, in their cloud servers. Furthermore, they allow bundling number of CPU sizes with each RAM capacity; *FlexiScale* 8 CPU sizes × 6 RAM sizes. *Amazon* and *Joyent*'s bundled resources, CPU, RAM, storage and network bandwidth cannot be easily changed independently of each other. Amazon's on-demand servers have a factor of four CPU capacity differences between standard instances when compared to *GoGrid* and *Joyent* on-demand servers which have almost a factor of two CPU and RAM differences. Amazon's RAM capacity differences are even higher; i.e., $\frac{1}{2}$GB, 1.7GB, 7.5GB. If it is needed to add 4GB RAM capacity then either three instances of 1.7GB or one instance of 7.5GB should be added. With *GoGrid* and *Joyent,* this can be achieved by adding 1 instance with 4GB RAM. *ElasticHosts* and *CloudSigma* allow cloud consumers to customize the capacity of one or more resource with very fine-granular increments.

Such situation leads cloud consumers to some important questions including:

- What are the key factors that could influence the ability to achieve the elasticity requirements of certain application workload patterns? Although we have discussed initial factors, there still other factors to be identified along with certain metrics from the cloud consumers' perspectives.
- How to evaluate the capabilities of elasticity tools and services, and accordingly the level of elasticity support of different *CISPs* and *CIMPs* from the cloud consumer's perspectives?
- What are the key service level guarantees and quality of service properties that are crucial for elasticity from the cloud consumer perspectives? We discuss the details of this challenge in Sect. 5.

3.5 Locations of CISOs

Most *CISPs*' datacenters such as Amazon, *Terremark,* and *Rackspace* are geographically distributed across several locations which cover key regions/zones worldwide. Such geographical distribution has two important benefits for *CISPs*. First, it increase service availability and reduces chances for single point of service failure. Second, *CISPs* ensure that their *CISOs* reach the widest range of consumers around the globe. Similarly, cloud consumers can benefit from *CISOs* at multiple locations. Specifically, it allows cloud consumers to distribute their application services at different locations/regions that are closer to their users. Data replication and recovery is also another key advantage in case of the occurrence of catastrophes or natural disasters. One limitation of *CISOs* distributions is the data protection regulatory compliance and standards in some countries [36] that require certain data, e.g., banks and citizens' data, not to be stored or transferred to servers outside that country.

In contrast to most *CISPs*, Amazon allows its cloud consumers to choose the locations of their cloud servers, and other *CISOs*, in 5 locations within 3 regions and it differentiates its pricing according to the locations. Table 5 quantifies the cost of 720 hours usage (30 days) of three on-demand server instances at Amazon's 5 locations.[17] As shown in the table, servers' costs of the same instance type in North Virginia are the lowest. On the other hand, the server costs of the same instance type in Tokyo are the most expensive. This could be due to the factor of expenses differences in energy, hardware property rental, and human labor. Server costs of the same instance type in other locations are identical. The server prices also vary between locations for Windows instances due to the high licensing fees of Windows compared with Linux.In addition to servers' price differences, pricing of some other CISOs such as storage and internet data transfer vary at different locations. Other *CISPs* often do not give such server/resources location options, especially for on-demand servers, and it is not clear where *CISOs* are located although they have datacenters at different locations or regions. In such offerings, one probable way that cloud providers use to efficiently allocate cloud servers/resources could be based on the provider's resource utilization and reduction of carbon footprint policy. For example, it is more efficient for a cloud provider to serve various computing resource demands of multiple consumer's by allocating computing resources from one cluster or location than allocating resources from multiple clusters within or at different locations/regions. Such policy will help cloud providers to save on operation costs (e.g., energy costs) as well as to reduce its carbon emission. A recent industry research results [23] have shown significant decrease of CO2 emission and energy costs when hosting their applications on Microsoft's cloud infrastructure compared to in-house application hosting.

As an example, *OrionVM*[18] cloud infrastructure offerings are located in Australia only. This is beneficial for Australian banks and government organizations, for example, as *CISOs* comply with data protection regulations. However, *OrionVM* might not be beneficial for organizations in other areas, e.g., Europe and America. Another way to overcome data protection obstacles is to use hybrid cloud deployment in which critical parts of the applications remain within country/organization walls and other parts are deployed on public cloud infrastructure.

Geographical distribution of *CISOs* challenges cloud consumers with important issues that relate to application economics and elasticity. Among these key challenges are the following:

- *Economics of Locations of CISOs*: location of *CISOs* could influence application economics. For example, Amazon cloud servers, network, data storage, and operations vary based on 5 locations. Deploying an application at different location servers will be charged different

[17] The costs are calculated based on server instances' prices as for September 15, 2011. The prices of server instances may be change by the provider over the time.

[18] http://orionvm.com.au/.

Table 5 Example of different Amazon cloud servers' costs at different locations

Serve instance type (on-demand)	Server instance costs for 720 hours (30 days) based on server location/region				
	US N. Virginia	US N. California	EU Ireland	APAC Singapore	APAC Tokyo
One "Extra Large" instance (Linux)	$489.60	$547.20	$547.20	$547.20	$576.00
One "High-CPU Medium" instance (Linux)	$122.40	$135.80	$135.80	$135.80	$144.00
One "High-Memory Double Extra Large" instance (Linux)	$720.00	$820.80	$820.80	$820.80	$864.00

rates of CPU-hour and storage and operation volumes. More importantly, it will increase data transfer between different locations which incurs considerable costs that can add-up quickly. Even in the case of hybrid cloud deployment, to comply with data protection regulations, the economics issue of cloud servers location, storage, and network will still hold. One of Jim Gray's key conclusion about distributed computing economics is data should be as close as possible to servers to avoid considerable network costs of data transfer [19]. Armbrust et al. [3] illustrated that physical shipping of 10 TB of data would cost considerably less and require less time than transferring it over the internet from one datacenter location to another using Amazon storage and network services. This is only one example where cloud data storage and network transfer impacts application economics and there are many variations of cases that need to be investigated during application deployment at geographically distributed locations. The application workload pattern would also have an influence on its economics measure, e.g., data-intensive workloads require intensive data storage and transfer. Based on this analysis, we raise the following important questions about economics of application distribution and its economics:

- How various pricing of *CISOs* at different locations could influence application economics if it is deployed at multiple locations or at one location?
- What is the impact of geographic distribution of application's users and workload patterns on its economics?
- How to evaluate economics of geographically distributed deployment architectures of applications on public or hybrid cloud infrastructure?
- What is the cost effective way to deploy application with geographically-distributed users on multi-location cloud infrastructure?

- *Cloud Infrastructure Performance Variability*: performance of cloud infrastructure resources, e.g., servers, network bandwidth and disk I/O, have been proven to be variable over different periods of times [12, 24] and over different locations [39]. Such performance variability could be logically expected as CISPs cannot completely determine/predict how much computing resources of their cloud infrastructure will have been utilized, at what growth rate and for how long. The cloud resource utilization also depends on the amount of resources that cloud consumers would dynamically request or release from different regions/locations which also hard to predict. Such performance variability raises important question marks about its potential impact on application performance and economics as well as its potential impact on achieving desired elasticity levels. including the following issues:
  - Which *CISPs*' offering location would best suit certain application workload pattern given performance variability of different locations at different times?
  - How to measure cloud infrastructure performance variability indicator? How much stable/variable is the performance of CISPs' offerings at different locations?
  - How to determine the causality of scaling cloud infrastructures? Which scaling needs are due to cloud infrastructure performance variability which ones are due to my application's workloads?

4 Economics of elastic cloud-based applications

In this section, we analyze and discuss the main challenges that could face cloud consumers to understand and achieve economic elasticity for their e-business applications when deployed on public cloud infrastructure.

4.1 Economics and elasticity modeling of cloud-based applications

Economics and elasticity of public cloud infrastructure have been heavily reported and demonstrated in research and industry communities as fundamental drivers for different application domains [3, 21, 28, 36]. Some research work [2, 28] investigated the migration costs of software applications to the public cloud infrastructure. However, such work does not consider the elasticity dimension of operational costs. Moreover, it is based on simple calculation models which are tailored for specific application use cases with fixed workload patterns. Li et al. [5] proposed comprehensive financial models for calculating total cost of ownership and utilization costs of elastic cloud infrastructure but from the cloud provider's perspective. However, similar costs and elasticity models are still required from the cloud consumer's perspective.

We believe that cloud infrastructure consumers are still challenged with the need for generic models for capturing the economics of their applications on any public cloud infrastructure. Specific challenges include:

- How to model and calculate operational costs of an application of certain workload on different public cloud infrastructure?
- How to model and calculate the *Cost of Elasticity (CoE)* and the *Return on Elasticity (RoE)* of an application of certain workload on different public cloud infrastructure?

Such models are non-trivial as it depends on various factors including the application's workload volumes and patterns [30], pricing structures and specifications of different computing resources (software and hardware) [21], and cloud infrastructure management services [3] (e.g., monitoring and control). Such models have to also consider different types of cloud service offerings which are packaged with different hardware and software configurations that can be charged in hourly, daily, monthly, or yearly basis. The workload variability, for example, requires dynamic allocation of different types of computing resources for different periods of times and at different pricing tiers. Such parameters become increasingly complex and highly variable in economics equations. Furthermore, in the business world other financial parameters such as *time value of money* are often important for consideration in such calculations [36].

Similarly, cloud consumers need generic models for modeling and measuring their application elasticity on any public cloud infrastructure. Particularly, the key challenges in this part include:

- How to visually illustrate application's workload variability and amount of resources to be allocated or removed to meet the application's workload volumes?
- How to measure efficiency of scaling mechanisms of different *CISPs* and resources?
- How to model and measure business and technical elasticity metrics for e-business applications?

Modeling costs and elasticity of cloud-based applications are crucial for business organizations. In principle, cloud consumers often need to conduct cost/performance analysis and reason about the effectiveness of different scaling strategies for their applications. In practice, business organizations have a defined budget for their IT resources that need to be met. Dynamic elasticity policies can play effective roles to achieve such business goals. In practice, *elasticity is not an autonomic feature and cloud consumers have to configure appropriate scaling policies to enable it*. Without advanced modeling and analysis tools this is almost impossible. The flexibility of on-demand computing resources makes an application's operational costs variable and more finely granular as it is tightly coupled with defragmented fine-granular pricing schemes that are metered differently such as: CPU usage-per-hour, number of I/O operation call, size of storage volumes. Therefore, having appropriate elasticity and economics models and metrics is essential to enable cloud consumers to analyze, plan, and control costs and scaling policies of their e-business applications on any public cloud infrastructure at fine-granular levels.

4.2 Economics-driven decision-making for elasticity of cloud-based applications

The on-demand provisioning of fine-grained computing infrastructure services has introduced new ways for cloud consumers to achieve business resilience and agility at reduced costs. However, the *economics and elasticity benefits* of CISOs are not an automatic gain for cloud consumers because public cloud infrastructure does not automatically scale computing resources based on application's workload requirements. It is almost impossible for *CISPs* to enable automatic elasticity for all their cloud consumer's applications. This is because such applications have diverse business and technical metrics that need to be monitored to enable economic scaling decisions. Instead, cloud consumers are challenged with the need for automated mechanisms (e.g., intelligent elasticity engine) that make and execute economic scaling decisions on the right time, to the right cloud resources, and with the right amount of cloud resources. Such engine should be configured with appropriate elasticity policies based on the application's business and technical metrics (SLAs) and application's workload changes. The elasticity decisions can then be triggered automatically in proactive or reactive ways as it will be discussed later in this section.

Some research work [31, 44] proposed automated scaling mechanisms based on predefined scaling policies/rules and configurations. Such mechanisms trigger scaling actions based on user-defined cloud resource utilization rules and therefore it is reactive. In contrast to [7], these approaches [31, 44] also did not investigate the economics of scaling actions. Bonvin et al. [7] proposed *economic viability* model which is used by each *server agent* to balance usage of server resources. However, they assumed availability of dedicated servers and did not consider usage-based on-demand cloud server offerings.

Some *CISPs'* scaling tools such as Amazon's *Autoscaling* [1] and *RightScale's Autoscaling* [37] enable automated infrastructure elasticity based on predefined scaling policies correlated with cloud infrastructure utilization metrics. Both tools are specific to Amazon infrastructure services and require knowing when and how much cloud servers to scale in advance to be reactively triggered.

Based on our analysis of elasticity of cloud infrastructure offerings and tools and analysis of existing research work we perceive the key research challenges that relate to economics-oriented elasticity decision-making for cloud-based applications explained in the following subsections.

4.2.1 *What to scale?*

We also believe deciding on which cloud resources to scale is another important pillar for economics-driven decision-making elasticity. This specifically requires identifying performance bottlenecks in cloud-based application architecture and collecting monitoring data about it and characteristics of its workload. Based on our best investigations, we noticed that most scaling approaches and tools [1, 7, 31, 37, 44] assume that performance bottlenecks are related to degrading servers. Reese [36] identified reasonable list of potential capacity bottleneck points in a typical web application architecture deployed into the cloud. These bottleneck points include:

- *Network bandwidth*: between the load balancer and the application servers as well as between the application servers and database servers.
- *Load balancing*: ability of a load balancer to properly distribute load across the application servers.
- *Computing capacity*: the CPU, RAM and internal storage utilization of application and database servers.
- *Computing resource performance*: number of I/O or read/write operations per unit of time for application and database servers.

It is crucial to notice that such bottlenecks cannot be necessarily addressed by adding more servers or replacing existing servers with more powerful ones. There are different possible scaling points have to be investigated to decide what to scale before deciding on how to scale. *GoGrid* [17] identified a list of scaling points that could cause performance bottlenecks. These points include:

- *Processing power*: CPU speed measured in GHz.
- *Memory*: RAM capacity measured in GB.
- *Network bandwidth*: network speed in Gbps.
- *Database performance*: number of transactions/second.
- *Disk storage*: system storage capacity in GB or TB.

As an example, in memory intensive applications such as *Twitter* performance bottlenecks often occur in memory and database I/O [42] which requires retrieving data very frequently. In contrast, transactional e-business applications tend to have CPU performance bottlenecks and network bandwidth as transactions require frequent processing and frequent data transformation between different application layers. We believe determining the potential roots of different resources bottlenecks in cloud-based application architecture and measuring its impact on application performance is an important issue to investigate. Based on determined bottlenecks possible scaling decisions can then be explored given the cloud infrastructure provider's resource capabilities and bundles; different providers enable different scaling points. This will contribute to the ability of being able to make effective economic decisions on what infrastructure resources need to scale.

4.2.2 *How much to scale?*

The key challenges of this dimension of the decision-making process are to decide on how much computing resource to acquire or release? And what are its types and capacity specifications? Most current approaches and tools [1, 31, 37, 44] require specifying the type and capacity of cloud servers to be added/released in predefined auto-scaling policies. For example, RightScale's auto-scaling [37] offers different types of pre-configured server templates of specific instance size/capacity. Similarly, Amazon's auto-scaling [1] requires cloud consumers to configure an Amazon Machine Instance (AMI) of a specific size, e.g., standard small, standard large, high-memory, etc. Both tools allow specifying the number of server instances in increments or decrements. Any cloud servers to be acquired or released by the auto scaling tool must be of the same size and configurations of the pre-configured ones. We can see two key problems with this auto scaling feature. First, there is no flexibility in the size and type of resources that can be scaled. Such flexibility is important as application's workloads are often variable [6] and the type and specifications of cloud servers are mainly dependent on the application's workloads. Furthermore, it might be more efficient to add one large instance instead of adding two small instances given an increasing workload load. For example, a Large instance would quickly and more efficiently accommodate sharp workload rise than two instances as the later needs more resources (e.g., load balancer, monitoring services) which adds more cost and performance overhead (as workload has to be redistributed among two instances.) Second, there is no clue about the relationship between the number of servers, and its specification, and the scaling effect in terms of performance gains. There is no clear evidence about scaling linearity that could result from the addition of more computing servers. According to Amdahl's law [20], the overall speedup that can be achieved by a program running on a parallel platform is limited by the program's sequential portion. In particular, the higher the parallelized program's portion the higher speedup that can be obtained when the number of processors increases. In practice, there are other factors besides the percentage of program parallelism that could also impact the linearity of scaling. For example, Bodik et al. [6] demonstrated different workload characteristics that seem to have distributed effects on different portions of the applications. Therefore, adding more computing resources does not necessarily result in proportional performance improvements.

4.2.3 *How to scale? Which scaling strategy?*

Horizontal scaling is widely discussed and supported type in research [7, 31, 44] and industry [1, 37]. In principle, horizontal scaling, also known as *scaling out* is easier to automate without service interruption when compared to vertical scaling. In addition, horizontal scaling tends to be a cheaper option than vertical scaling. However, its management becomes increasingly complex over time number of running servers and other resources increase and require careful monitoring and control. It is important to notice that not every application component can benefit from being scaled out as this depends on the component's code structure. It is important to notice that horizontal scaling does not necessarily result in the desired proportional increase of performance especially for those classes of applications that were not originally designed to operate on distributed environments. Amdahl's law [6] illustrates that proportional performance gains of an application cannot be achieved by increasing number of CPUs but it depends on the percentage of parallelism in the application. In such cases, vertical scaling could achieve considerable performance benefits and reduce the manageability of different servers. Despite the advantages of horizontal scaling over vertical scaling, we believe it is important to conduct a trade-off analysis study of both scaling types. Such research should provide benchmarks that include the price/performance metric. It would be also interesting to identify circumstances under which one scaling type is more suitable than other based on some key criteria from the cloud consumer's perspective. Michael et al. [33] carried out such an experimental case study to evaluate performance/price scale out and scale up but on dedicated servers, IBM's specific clusters, and specific to Nutch/Lucene framework for implementing search applications. However, it would be of key interest to conduct a similar study using on-demand cloud infrastructure offerings and on other application domains, e.g., e-business applications, and development framework, e.g., LAMP (Linux, Apache, MySQL, and Perl/PHP/Python). It is also of key interest to investigate the characteristics of application workloads that affect its scalability and which scaling type is more suitable for certain workloads of specific characteristics.

Most research work such as [7, 31, 35, 44] have not examined the economics of elasticity, i.e., scaling down and scaling in, and economics of different scaling types, e.g., horizontal vs. vertical. Scaling in/down is economically crucial as elasticity is related with application workload variability and to avoid the resource over-provisioning scenario where unused resources should be released as soon as workload volumes go down.

4.2.4 *When to scale?*

Event-driven (or reactive) scaling approaches include delays until scaling actions take full effect. Li et al. [29] demonstrated *scaling latency* of different cloud infrastructure services and argued about its impact on application's performance and costs. Making timely decisions on when to scale also depends on *how legitimate is a workload increase/decrease or spike? how long it would last? what are its value and impact on application's metrics*? [6, 36]. Accordingly, we deem a *proactive scaling* approach that consider such data that can be obtained from log files and real-time measurements to decide on triggering scaling actions at the right time. Predicting genuine and high-value high-impact workloads can contribute to the ability of being able to make effective and economic elasticity decisions. Another important factor for deciding when to scale is SLA satisfaction/violations and related financial and non-financial penalties which are discussed in the next section.

4.3 Elasticity example: MyShop scaling strategies

In our scenario, let us consider possible *elasticity strategies* that could be planned for *MyShop's* application tier. There could be three scaling strategies:

- *Scaling out-in (Horizontal Scaling)*: by adding four additional application servers to the existing two main application servers every Saturday and Sunday from 10 a.m. to 10 p.m. and removing it at all other times. All servers should have the same processing capacity (we use *CPU-intensive* server of small computing capacity)
- *Scaling up-down (Vertical Scaling)*: by replacing the two main application servers with one more very powerful application server (i.e., we use one *CPU-intensive* server with computing capacity equivalent to six small servers) and then switching to the two main application servers at all other times
- *Hybrid Scaling*: a combination of *horizontal* and *vertical* scaling strategies with variation of number of cloud servers.

Each scaling strategy has different costs and impact on *MyShop's* performance attributes. Table 6 summarizes the key aspects that are important for deciding on which strategy would better suit *MyShop's* application tier based on CISOs of CISP1. The cost calculations are based on *Weekend-Sales workload*.[19] The *monitoring costs* shown in the table are important for scaling; to ensure appropriate performance levels of all servers and in case desired performance is not achieved or a server failure occurs appropriate scaling actions are taken.

[19] Cost calculations can be similarly applied to the *Mid-Week-Sales workload* and the database tier workloads.

Table 6 Potential elasticity strategies for MyShop application's tier (weekend workload)

Scaling strategy	Server scaling costs	Monitoring costs	Application's availability and reliability
Horizontal scaling	24 hrs/w × \$0.085/hr × 6 servers × 52 weeks = \$636.48/yr	Defining and configuring 7 metrics for 6 servers Costs: \$3.5 per server/mo × 6 servers × 12 months = \$252	Highly available—no single point of failure Highly likely reliable—quick recovery time
Vertical scaling	24 hrs/w × \$0.68 × 1 server × 52 weeks = \$848.64	Defining and configuring 7 metrics for 1 server Costs: \$3.5 per server/mo × 1 server × 12 months = \$42	Low availability—single point of failure Highly likely unreliable—long recovery time
Hybrid scaling	(24 hrs/w × \$0.085/hr × 3 servers × 52 weeks) + (24 hrs/w × \$0.34 × 1 server × 52 weeks) = \$742.48	Defining and configuring 7 metrics for 4 servers Costs: \$3.5 per server/mo × 4 servers × 12 months = \$168	Improved availability—no single point of failure Improved reliability—medium recovery time

Based on the calculations summarized in Table 6, the following observations are important to consider in MyShop's *elasticity strategies* decision-making:

- While the server costs of *horizontal scaling* strategy is *cheaper* than *vertical scaling*, *monitoring costs* of horizontal scaling are *significantly higher* than monitoring costs of vertical scaling; 6 times more expensive as there are 6 servers need monitoring compared to 1 server in *vertical scaling*. Generally speaking, the more servers are used to scale the more complex and expensive to monitor it.
- *Horizontal scaling* strategy also incurs additional charges (not included in the example) such as load balancing server costs (including data processing costs by each load balancer) and internet data transfer. Like *monitoring costs*, such additional charges will increase the total costs of *horizontal scaling* especially as the number of servers to be added are always higher than the ones in *vertical scaling* (in analogy to the monitoring costs.)
- *Vertical scaling* has one major drawback, particularly its influence on application's *availability* and *reliability*. Having one application server will lead to a *single point of failure* scenario which will influence application's *availability* considerably. In *horizontal scaling*, if one server fails all its incoming workload traffic can be dynamically and transparently distributed to the other five servers.[20] This ensures *high application availability*. Furthermore, CISP1 provides a *health check service* to continuously check the health status of pre-configured scaling servers and if any server has poor performance or is down then that server will be automatically restarted to reduce its *recovery time*. In a *vertical scaling* case, if a server underperforms or fails then the whole application server will become *unavailable* and it will need a considerable time to be brought back to normal operation (long recovery time.)
- The *hybrid scaling* strategy provides a *cost-performance balance*. Servers' costs and *monitoring costs* are intermediate. At the same time, *single point of failure* can be avoided and, therefore, application's *availability* becomes better than vertical scaling. Similarly, application's *reliability* becomes better than vertical scaling as availability has been improved and *recovery time* from any server failure becomes less with 4 servers instead of 1 server.
- The *economics* of illustrated *elasticity strategies* above will become more challenging when different CISOs and *elasticity costs* and support of other CISPs are considered. As we illustrated in the economics example of *MyShop* application's tier, there will be several differences between providers' offerings, pricing, tools, and support for enabling elasticity strategies. The *scaling costs* will also make more *economic-sense* when other tier's servers and other required cloud infrastructure resources (scaling the network bandwidth between servers, storage, etc.) are included in the total scaling costs of each strategy.

5 Service level agreement of elastic cloud-based applications

While deployed applications on a public cloud infrastructure can highly benefit from *on-demand* access to various computing resources at low prices, application's service levels are highly likely to become uncertain. We distinguish between two types of service level agreements (SLAs):

1. *Cloud Infrastructure SLA (CI-SLA)*: These guarantees are offered by a public CISPs to its cloud consumers[21] to as-

[20] This functionality can be configured through load balancer configurations and a *health check service* provided by CISP1.

[21] Application owners or companies who their applications or part of it are deployed and operated on public cloud infrastructure.

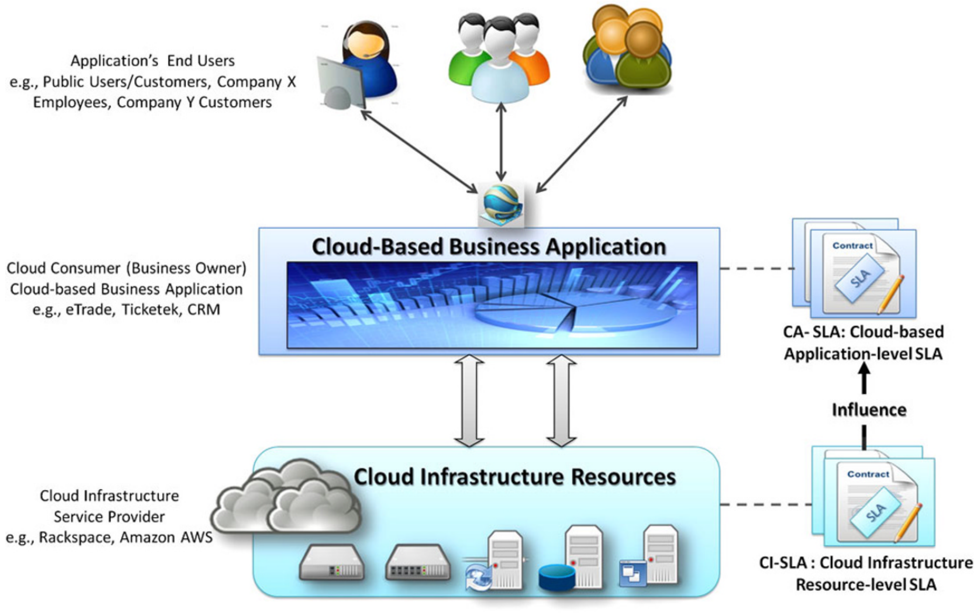
Application's End Users
e.g., Public Users/Customers, Company X Employees, Company Y Customers
Cloud Consumer (Business Owner)
Cloud-based Business Application
e.g., eTrade, Ticketek, CRM
Cloud-Based Business Application
Contract
SLA
CA- SLA: Cloud-based Application-level SLA
Influence
Cloud Infrastructure
Service Provider
e.g., Rackspace, Amazon AWS
Cloud Infrastructure Resources
Contract
SLA
CI-SLA : Cloud Infrastructure Resource-level SLA

Fig. 2 Dependency between cloud infrastructure SLA and cloud-based application SLA

sure certain quality levels of their cloud computing resource capabilities and specifications (e.g., server performance, network speed, resources availability, storage capacity). In other words, *CI-SLAs* reflect the perspective of public *CISPs*.

2. *Cloud-based Application SLA (CA-SLA)*: These guarantees relate to the levels of quality of an application which is running on a public cloud infrastructure. In particular, cloud consumers often offer such guarantees to their application's customers/end users to assure quality of services they offer to them such as application's response time, availability and security. Hence, *CA-SLAs* reflect the perspective of *cloud consumers*. For example, cloud consumers who deploy their Customer Relationships Management (CRM) on public cloud infrastructure would be interested in monitoring the *CA-SLA of their application*. In such case, average waiting time, average service time, and queue length are good examples of essential properties of *CA-SLA* which need to be monitored and maintained at the consumer side.

We illustrate the relationship between *CI-SLA* and *CA-SLA* in Fig. 2. Unlike traditional application's *SLAs*, *CA-SLA* becomes highly dependent on *CI-SLA* as the application, or part of it, is running on the public cloud infrastructure as illustrated in Fig. 2. Given the reported cloud infrastructure performance variability [12, 24, 39] and failure [45], whenever cloud infrastructure quality levels degrade or fail application's service levels (or at least part of it) are highly likely to be influenced. Therefore, such SLAs dependency puts management burden on cloud consumers as it becomes their responsibility to assure satisfaction of their application SLAs to their application customers or end users. To reduce the influence of CI-SLA on CA-SLA, cloud consumers need to understand the relationship and requirements as well as related challenges of both CI-SLA and CA-SLA which are discussed in the next sections.

5.1 Cloud infrastructure SLA (CI-SLA)

Currently, *CISPs* do not provide adequate *SLAs* for their cloud infrastructure service offerings [3]. Particularly, most providers guarantee service availability only [3, 13]. Providing appropriate *CI-SLAs* is crucial for cloud consumers as their application's quality levels become significantly dependent on cloud infrastructure resources service levels. The costs of service down-time are always very expensive. Several costs have been reported in a Gartner research [43] including revenue, financial performance, employee productivity, and damaged reputation. Durkee [13] argues that whatever compensation a cloud infrastructure provider may offer for cloud service degradation or failure often it will not compensate for the cost of lost revenue, breach of customer *SLA* or loss of market share credibility.

Existing research work [8, 25, 27] focus on enabling SLA-based provisioning of cloud infrastructure services. Bouchenak [8] introduced *SLAaaS* (SLA aware Service) as a new cloud model that consider integrating cloud QoS and SLA requirements with the cloud. Based on *SLAaaS*, she also identified and discussed research challenges to support dynamic control of cloud elasticity to meet cloud QoS and SLAs. Kertesz et al. [27] proposed a resource virtualization architecture that incorporates automated SLA-based meta-negotiation, meta-brokering, brokering and deployment of cloud resources on-demand. Reese [36] discussed important quality attributes of current cloud service offerings namely availability, reliability, performance, security, disaster recovery, and legal and regulatory issues. He explained how some existing service level guarantees offered by cloud providers, e.g., Amazon's EC2 and S3, RightScale's cloud servers, cannot be fully relied on as it is vague and non-trustable. Furthermore, Reese discussed with some examples how cloud hardware resources and technologies used to provide CISOs such as virtualization technologies could impact performance, availability, and reliability of web applications when it is deployed on public cloud infrastructure resources.

In this research dimension, the following research questions are important to be raised and investigated:

- What are the essential Quality of Service(QoS) properties, from the cloud consumer's perspective, of offered cloud infrastructure services?
- At what level of granularity these QoS properties should be defined and monitored?

This includes identifying quality attributes, specifications and capabilities of all types of cloud infrastructure services, defining its semantic meaning and metrics and, where appropriate, the ways to measure them, and finally the roles and responsibilities of involved parties in the *SLA*. It is crucial for cloud consumers to have adequate details about the infrastructure resources and its performance levels especially with the heavy use of virtualization technology and multi-tenancy model by cloud providers. This will be a significant challenge for cloud providers as some cloud infrastructure services have been recently proved to have considerable performance variability [12, 24, 39]. This is because cloud providers often have a large number of cloud consumers with variable usage patterns and requirements that increase the complexity of predicting cloud resources performance levels. Such cloud infrastructure performance variability will likely influence *CA-SLAs*.

In the context of our research discussion, we believe that *Quality of Elasticity (QoE)* should be one of the most essential quality properties for cloud consumers. Here, we define *QoE* as all attributes that define how well elasticity is achieved, its resources specifications and capabilities, and also the assurance of elasticity efficiency. *QoE* properties may include:

- Elasticity time: is the total time (resource allocation and booting and shutdown times) which will be taken to scale out/in and up/down for each computing resources/instance type.
- Minimum and maximum number and/or capacity of allowed server instances to be added.
- Specifications and types of allowed computing resources to be added.
- Availability of server instances and related resources to enable elasticity.

QoE properties are important because elasticity is one of the most significant incentives for cloud consumers to utilize cloud infrastructure services due to its economics. Therefore, having adequate elasticity guarantees will likely increase cloud consumer's trust in relying on such dynamic cloud features. Currently all *CISPs*, according to our best knowledge, do not offer any guarantee to their elasticity capability. For example, *Amazon Auto Scaling* [1] terminates any server instance in an *auto scaling group* when its performance degrades and launches another one. So, such issues would be critical for cloud consumers as it will cause interruption to their application's services.

5.2 Cloud-based application SLA (CA-SLA)

Unlike *CI-SLAs*, this type of *SLAs* is more concerned with the quality metrics of applications running on cloud infrastructure resources. Achieving *CA-SLA* at the provider side could be impractical and very challenging because of the heterogeneity workload characteristics of various cloud consumers and, therefore, their *SLA* requirements. Therefore, we think it should fall under the responsibility of cloud consumers, i.e., application owners, not cloud providers. Application's owners often set application-specific IT and business metrics to ensure certain service levels for their application's customers. *CISPs* are not supposed to support such metrics of each organization. Accordingly, we perceive the following key challenges that face cloud consumers in this context:

- *Mapping Application-specific Metrics*: particularly how to transparently model and map application-specific metrics to corresponding cloud infrastructure resources and reason about its dynamic impact? As *CA-SLA* becomes highly dependent on *CI-SLA*, we believe ensuring satisfaction of an application's Service Level Objectives (*SLOs*) will require:
  - Identifying cloud infrastructure resources that could impact application's *SLOs*.
  - Modeling relationships between cloud infrastructure resources and application *SLOs*.
  - Modeling impact of performance variability of cloud infrastructure resources on application's *SLOs*, e.g., *SLA* violation and economical aspects.

  It should be indicated that achieving such needs is not an easy or straightforward task at all. Due to its significance, some researchers tried to address similar issues [9, 11]. In their *SLA Decomposition* approach, Chen et al. [11] proposed analytical models for capturing the relationships between high level system *SLOs* and low-level system component goals. The low-level goals are then used to efficiently allocate and monitor corresponding computing resources in a private virtualized datacentre. There is good potential in this work if the mapping can be achieved from the cloud consumer perspective and in public cloud environments. It is almost impossible to address this problem from the *CISPs'* perspective due to the significant variability of cloud consumer's application workloads and proprietary application metrics and requirement [12, 24, 39]. We see this mapping as a crucial input for enabling *SLOs* compliance mechanisms as will be explained next.

- *SLA Monitoring of Cloud-based Applications*: the key challenge in this dimension is how to continually and transparently assure compliance of different application-specific *SLOs* of an application running fully or partially on cloud infrastructure? Based on our vision of the key requirements of the mapping application's specific-metrics we perceive the following essential issues that are related to the monitoring challenges of *CA-SLAs*:
  - Performing periodic measurements of all cloud infrastructure resources that relate to application's *SLOs*.
  - Correlating and aggregating measurements based on *SLOs* using relationship models.
  - Evaluating and reasoning about the impact of cloud infra-structure resources measurements on potential *SLOs* violation and subsequent costs using impact models.

  Some key research work [10, 14, 18] has been conducted as an attempt to address the above listed issues. In his conceptual architecture, Goyal [18] focuses on managing the application's services that are deployed on multiple cloud infrastructures based on the enterprise's governance, risk, and service management policies. Ferretti et al. [14] proposed a middleware that can be integrated with cloud infrastructure platform to dynamically configure, manage, and optimize cloud resources based on the *SLA* of the cloud-based application. Both approaches [14, 18] do not consider modeling the relationship between application's *SLOs* and cloud infrastructure resources. Likewise, Chazalet's monitoring and service level checking architecture [10] does lack application-specific SLOs mappings. Ferretti et al. [14] and Chazalet [10] investigate the relationship between resource utilization and *SLA* violations but specifically for private cloud infrastructure. The monitoring issues we raised become more challenging in public cloud environments where the massive scale of economies is appealing but at the expense of its performance variability.
- *SLA-driven Elasticity of Cloud-based Applications*: specifically, the main challenge here is concerned with *how to make automated and economic SLA-based elasticity decisions on behalf of cloud consumers?* In this regard, we see the mapping of application-specific metrics and SLA monitoring of cloud-based applications aspects as a fundamental to achieve this *SLA-driven elasticity*. This is because such metrics mapping and monitoring provide comprehensive information that is crucial for deciding when to trigger appropriate scaling policies to avoid SLA violations and its financial impact on e-business applications. As we previously discussed, economic elasticity decisions are about triggering appropriate scaling policies at the right time to the right cloud resources with the right amount of cloud resources. We see this as being impossible without the mapping and monitoring of application-specific *SLOs* to cloud infrastructure resources. Current scaling approaches that we have previously discussed have a same view, and thus same approach. Particularly, approaches such as [7, 31, 44] are designed to perform *application/business SLA-neutral elasticity*. Its cloud infrastructure scaling decisions and policies are based on cloud resource-specific metrics (e.g., resource utilization) and, therefore, it does not consider application-level metrics, e.g., application SLOs, which are important to e-business applications. Such cloud resource-metrics cannot be meaningful or useful in many cases depending on the workload characteristics and *application SLA* requirements. In addition, the economics of scaling decisions achieved in [7] is based on resource utilization and is independent from application SLA and financial impact of application SLA violation. Similarly, *Amazon Auto Scaling* [1] and *RightScale's Autoscaling* [37] scaling policies are based on cloud infrastructure monitoring metrics which means that they do not map and correlate these metrics to the application's *SLOs*. *SLAaaS* [8] is a work in progress which represents a research roadmap for SLA-driven elasticity approaches. However, it focuses on the cloud provider's perspective, i.e., satisfying cloud provider's SLAs. As we have previously argued application SLA monitoring and elasticity should be managed individually by each cloud consumer and it is almost impossible to be comprehensively solved solely at the cloud infrastructure provider side. This is because e-business applications have different SLAs/SLOs that require specific monitoring and elasticity approaches when compared to cloud resource SLA monitoring and elasticity.

6 Conclusions

The provision of transactional business applications through the Internet has dramatically increased its dynamism in terms of variability of workloads volumes and patterns demanding highly flexible yet cost effective IT infrastructure and resources. The on-demand provisioning of various computing resources as services at massive economies of scale has perfectly matched this need. Deploying such e-business applications on public cloud infrastructure can highly contribute to achieving its elasticity requirements at economic costs but this is not a self-inherent capability of cloud infrastructure resources. The fast-growing number of diverse

Table 7 Summary of economics and eleasticity factors and related challenges of cloud-based e-business applications

Research aspect	Related factors	Open research challenges
Cloud resource bundling and specifications	– Fine-grained cloud resource packaging and instance types or classes – Customizability levels of CISOs' capacity – Different metrics and semantics for CISOs' capacity	– How to model and measure level of CISOs' customizability? – How to benchmark performance of CISP's cloud servers? – How to evaluate elasticity of CISP's servers?
Pricing models and offering types	– Different pricing models correlated with certain CISOs – Economics and elasticity of CISOs of different pricing models	– How to determine fixed and variable workload periods in the context of CISO types and pricing models? – Which CISO or combination of offerings would best suit each workload period in terms of economics and elasticity? – How to determine the point where the cost of overall on-demand elasticity is equivalent to the cost of dedicated server instances?
Software and system licensing	– Different licensing models for software and systems – Different ways of bundling software and systems with CISOs	– Which licensing model would best suit workloads of e-business applications? – How software and system licensing models provided with cloud server instances could impact application's economics and elasticity? – How to monitor and control different software and system licenses on multiple cloud server instances?
Elasticity support for cloud infrastructure resources	– Scaling limits (number of cloud server can be added automatically) – Scaling speed (time to add and run a cloud server) – Limited capabilities of elasticity tools (resource-specific elasticity)	– What are the key factors that could restrain achieving elasticity requirements of certain application workload? – How to evaluate capabilities of elasticity tools and level of elasticity it supports?
Locations of CISOs	– Performance variability of CISOs – Location of data and government regulations – Different pricing at different locations and cost of data transfer between different locations	– How different pricing of CISOs at different locations could influence economics and elasticity of applications deployed at different locations? – What is the most cost-effective way to deploy e-business applications that have geographically-distributed users at different locations cloud resources? – How to evaluate performance variability of cloud servers at different locations and its impact on application's economics and elasticity?
Economics and elasticity modeling of cloud-based applications	– Different variables influence application costs and flexibility on public cloud infrastructure – Understanding cloud infrastructure offering and pricing models and its impact on application costs and flexibility	– How to model and calculate operational costs of e-business application workloads on different public cloud infrastructures? – How to model and calculate cost of elasticity and return of elasticity of e-business application workloads on different cloud infrastructures? – How to visually illustrate e-business application's workload variability and its elasticity changes? – How measure efficiency of scaling mechanisms of different CISPs and resources? – How to model and measure elasticity metrics from business and technical perspectives?
Economics-driven decision support for elasticity of cloud-based applications	– Different performance bottleneck possibilities in application architecture – Multiple cloud resource capacity choices with different effects on scaling costs and performance – Different scaling mechanisms with different cost and performance effects – Time for scaling to take effect and its impact on application operational costs and performance	– How to determine genuine performance bottlenecks in cloud-based applications and which related cloud infrastructure resources to scale? – How to decide on how much cloud computing resources (its type and specifications) to add or remove to meet application's workload changes without affecting applications performance? – How to decide on most efficient scaling mechanism (vertical or horizontal) and its impact on application's operational costs and performance? – How to determine legitimate, period, value and performance/costs impact of application workloads? – How to decide between proactive and reactive scaling approaches and its impact on application costs and performance?

Table 7 (*Continued*)

Research aspect	Related factors	Open research challenges
Cloud infrastructure SLAs	– Performance variability of public cloud infrastructure services – Lack of adequate SLA for CISOs that suit cloud consumers	– What are QoS properties of cloud infrastructure services that are most important to the cloud consumers? – What are essential quality of elasticity properties that CISPs should guarantee to cloud consumers? – At what level of granularity these QoS guarantees should be defined and monitored?
Cloud-based application SLAs	– Relationship between application-level metrics and cloud infrastructure resources – Managing application-level SLOs on public cloud infrastructure – SLA-based elasticity support for cloud consumers	– How to transparently model and map application-level metrics to corresponding cloud infrastructure resources and reason about its impact on application economics and elasticity? – How to continually and transparently monitor various application-specific SLOs of an application running on public cloud infrastructure? – How such monitoring could provide support for real-time decision engine that makes economics and SLA-based elasticity decisions on behalf of cloud consumers?

cloud infrastructure services has attracted considerable attention from potential cloud consumers but at the same time it has introduced new challenges for them. Realizing cost-effective and resilience use of on-demand usage-based CISOs require extensive understanding and analysis of various aspects related to the CISOs and business requirements of cloud consumers.

In this paper, we have introduced a comprehensive analysis and discussion of the economics and elasticity challenges that cloud consumers, i.e., e-business application's owners, should consider for their transactional e-business applications when deployed on a public cloud infrastructure. Table 7 summarizes these key aspects along with related factors influencing each aspect and research challenges. Collectively, the analysis and discussion provide a *multi-lenses* insights that can help cloud consumers to understand, analyze and evaluate economic factors as well as elasticity capabilities of different CISPs' cloud service offerings, especially with regards to its suitability for their applications requirements. The open research challenges summarized in Table 7 provide a future research agenda for research and industry communities to help cloud consumers to investigate and tackle defined research challenges.

References

1. Amazon auto scaling documentation (2011) http://aws.amazon.com/documentation/autoscaling/, March 2011
2. Amazon web services case studies (2011) http://aws.amazon.com/solutions/case-studies/, March 2011
3. Armbrust M, Fox A, Rean G, Joseph A, Katz R, Konwinski A, Gunho L, David P, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a Berkeley view of cloud computing. Technical report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009
4. Beitch A, Liu B, Yung T, Griffith R, Fox A, Patterson D (2010) Rain: a workload generation toolkit for cloud computing applications. Technical report UCB/EECS-2010-14, EECS Department, University of California, Berkeley, Feb 2010
5. Bibi S, Katsaros D, Bozanis P (2010) Cloud computing economics. Advanced design approaches to emerging software systems: principles, methodology and tools. IGI Global Publishing, Hershey
6. Bodík P, Fox A, Franklin M, Jordan M, Patterson D (2010) Characterizing, modeling, and generating workload spikes for stateful services. In: Proceedings of the 1st ACM symposium on cloud computing (SoCC), pp 241–252. doi:10.1145/1807128.1807166
7. Bonvin N, Papaioannou T, Aberer K (2010) An economic approach for scalable and highly-available distributed applications. In: Proceedings of the IEEE international conference on cloud computing (IEEE CLOUD), pp 498–505. doi:10.1109/CLOUD.2010.45
8. Bouchenak S (2010) Automated control for SLA-aware elastic clouds. In: Proceedings of the 5th international workshop on feedback control implementation and design in computing systems and networks, pp 27–28. doi:10.1145/1791204.1791210
9. Breitgand D, Henis E, Shehory O, Lake J (2007) Derivation of response time service level objectives for business services. In: Proceedings of the 2nd IEEE/IFIP international workshop on business-driven IT management (BDIM), pp 29–38. doi:10.1109/BDIM.2007.375009
10. Chazalet A (2010) Service level checking in the cloud computing context. In: Proceedings of the IEEE international conference on cloud computing (IEEE CLOUD), pp 297–304. doi:10.1109/CLOUD.2010.15
11. Chen Y, Iyer S, Liu X, Milojicic D, Sahai A (2008) Translating service level objectives to lower level policies for multi-tier services. Clust Comput 11(3): 299–311. doi:10.1007/s10586-008-0059-6
12. Cooper B, Silberstein A, Tam E, Ramakrishnan R, Sears R (2010) Benchmarking cloud serving systems with YCSB. In: Proceedings of the 1st ACM symposium on cloud computing (SoCC), pp 143–154. doi:10.1145/1807128.1807152
13. Durkee D (2010) Why cloud computing will never be free. Commun ACM 53(5): 62–69. doi:10.1145/1735223.1735242
14. Ferretti S, Ghini V, Panzieri F, Pellegrini M, Turrini E (2010) QoS-aware clouds. In: Proceedings of the IEEE international conference on cloud computing (EuroSys), pp 237–250. doi:10.1145/1755913.1755938

15. Gartner (2008) Gartner top 10 disruptive technologies for 2008 to 2012. Technical report, Emerging trends and technologies roadshow, Gartner
16. GoGrid (2011) Gogrid cloud server user manual. https://wiki.gogrid.com/wiki/index.php/Cloud_Server_User_Manual
17. GoGrid (2011) Scale your internet business white paper. http://www.gogrid.com/downloads/GoGrid-scaling-your-internet-business.pdf/, March 2011
18. Goyal P, Mikkilineni R (2009) Policy-based event-driven services-oriented architecture for cloud services operation & management. In: Proceedings of the IEEE international conference on cloud computing (IEEE CLOUD), pp 135–138. doi:10.1109/CLOUD.2009.76
19. Gray J (2008) Distributed computing economics. ACM Queue 6(3): 63–68. doi:10.1145/1394127.1394131
20. Gregory A (2000) Foundations of multithreaded, parallel, and distributed programming. Addison-Wesley, Reading
21. Hilley D (2009) Cloud computing: a taxonomy of platform and infrastructure-level offerings. Technical report, College of Computing, Center for Experimental Research in Computer Systems, Georgia Institute of Technology, April 2009
22. Rackspace Hosting (2011) Cloudkick cloud management. https://www.cloudkick.com/
23. Accenture in collaboration with WSP (2010) Cloud computing and sustainability: the environmental benefits of moving to the cloud. Technical report, Accenture, November 2010
24. Iosup A, Yigitbasi N, Epema D (2010) On the performance variability of production cloud services. Technical report 1387-2109, Parallel and Distributed Systems Report Series, Delft University of Technology, Jan 2010 (CCGRID), pp 104–113. doi:10.1109/CCGrid.2011.22
25. Guitart J, Oriol Fitó J, Goiri Í (2010) SLA-driven elastic cloud hosting provider. In: Proceedings of the 18th Euromicro conference on parallel, distributed and network-based processing (PDP), pp 111–118. doi:10.1109/PDP.2010.16
26. JupiterResearch (2006) Retail web site performance: consumer reaction to a poor online shopping experience. Technical report, Akamai and JupiterKagan, June 2006
27. Kertesz A, Kecskemeti G, Brandic I (2009) An SLA-based resource virtualization approach for on-demand service provision. In: Proceedings of the 3rd international workshop on virtualization technologies in distributed computing, VTDC '09, pp 27–34. doi:10.1145/1555336.1555341
28. Khajeh-Hosseini A, Greenwood D, Sommerville I (2010) Cloud migration: a case study of migrating an enterprise IT system to IaaS. In: Proceedings of the IEEE international conference on cloud computing (IEEE CLOUD), pp 450–457. doi:10.1109/CLOUD.2010.37
29. Li A, Yang X, Kandula S, Zhang M (2010) CloudCMP: comparing public cloud providers. In: Proceedings of the 10th annual conference on internet measurement, pp 1–14. doi:10.1145/1879141.1879143
30. Li X, Li Y, Liu T, Qiu J, Wang F (2009) The method and tool of cost analysis for cloud computing. In: Proceedings of the IEEE international conference on cloud computing (IEEE CLOUD), pp 93–100. doi:10.1109/CLOUD.2009.84
31. Marshall P, Keahey K, Freeman T (2010) Elastic site: using clouds to elastically extend site resources. In: Proceedings of the IEEE international symposium on cluster computing and the grid (CCGRID), pp 43–52. doi:10.1109/CCGRID.2010.80
32. Mell P, Grance T (2009) Definition of cloud computing. Technical report, National Institute of Standards and Technologies (NIST), July 2009
33. Michael M, Moreira J, Shiloach D, Wisniewski R (2007) Scale-up x scale-out: a case study using Nutch/Lucene. In: International symposium on parallel and distributed processing, pp 1–8. doi:10.1109/IPDPS.2007.370631
34. PassMark (2010) CPU Benchmarks. http://www.cpubenchmark.net/cpu_list.php/
35. Pujol J, Siganos G, Erramilli V, Rodriguez P (2009) Scaling online social networks without pains. In: Proceedings of the 5th international workshop on networking meets databases (NetDB), co-located with SOSP
36. Reese G (2009) Cloud application architectures: building applications and infrastructure on the cloud. O'Reilly, Sebastopol
37. RightScale (2011) Adaptable automation engine. http://www.rightscale.com/products/features/adaptable-automation-engine.php/, March 2011
38. Sakr S, Liu A, Batista D, Alomari M (2011) A survey of large scale data management approaches in cloud environments. In: IEEE communications surveys and tutorials (IEEE COMST), pp 311–336
39. Schad J, Dittrich J, Quiané-Ruiz J (2010) Runtime measurements in the cloud: observing, analyzing, and reducing variance. Proc VLDB Endow, 3(1): 460–471
40. Amazon Web Services (2011) AWS economic center. http://aws.amazon.com/economics/
41. Amazon Web Services (2011) Elastic load balancing. http://aws.amazon.com/elasticloadbalancing/
42. Weaver E (2009) Improving running components. http://qconlondon.com/dl/qcon-london-2009/slides/EvanWeaver_ImprovingRunningComponentsAtTwitter.pdf/
43. Witty R (2011) Best practice in business continuity planning, Gartner, 2001. http://www.gartner.com/5_about/news/bcpbestpractices.ppt/, March 2011
44. Yang J, Qiu J, Li Y (2009) A profile-based approach to just-in-time scalability for cloud applications. In: Proceedings of the IEEE international conference on cloud computing (IEEE CLOUD), pp 9–16. doi:10.1109/CLOUD.2009.87
45. ZDNet (2011) AWS disrupted by us east coast failure. http://www.zdnet.co.uk/blogs/mapping-babel-10017967/aws-disrupted-by-us-east-coast-failure-10022283/