Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Modeling cloud business customers’ utility functions

Caesar Wu [*], Rajkumar Buyya, Kotagiri Ramamohanarao

CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Victoria 3010, Australia

ARTICLE INFO

Article history:
Received 20 September 2019
Received in revised form 29 November 2019
Accepted 27 December 2019
Available online 31 December 2019

Keywords:
Utility functions
Creation with cloud customers
Cloud market segment
Business customers
Value co-creation
Business application
Cloud service provider

ABSTRACT

Modeling a utility function for cloud business customers is one of the critical challenges facing many cloud service providers (CSPs) for their pricing strategy. It concerns how to measure various subjective experiences of the business customers and how to translate their cloud service experiences into a quantifiable unit, which can be determined by a utility function that reflects cloud resource consumption. The aim of this modeling process is to set up a pricing foundation so that CSPs can target a broader range of customers from various market segments and identify the optimal price point of their various pricing models. Previous studies have either focused on simple theoretical proof or drifted the meaning of utility between demand and supply or proposed a solution based on a uniform cloud market assumption. This paper proposes a novel and practical solution to define multiple utility functions based on a scenario of six cloud market segments, which are analyzed by three analytic approaches that are known as Markov chains analysis, queueing theory, and risk assessment. The entire pricing strategy emphasizes value co-creation between CSP and cloud business customers. In comparison with other methods, such as calibrated, price-quality, resource-based, simple linear, and capacity-aware, this method provides both internal and external rationalities for CSP to capture the subjective value of cloud business customers. Consequently, our experiment results show that this modeling method can increase a profit margin by 51% and decrease a unit cost by 22% for a CSP.

1. Introduction

The goal of this study is to define multiple utility functions for different cloud business customers’ preferences that are grouped into various cloud market segments so that a Cloud Service Provider (CSP) can create a price strategy based on a broad spectrum of cloud market to maximize its profit. Moreover, the CSP can tailor its limited investment resources and technical expertise to serve its target customers effectively.

In economics, the concept of utility means measuring a choice of individual’s preferences. In other words, it is to evaluate the individual’s subjective satisfaction, happiness and perception of worthiness that “the consumer derives from consumption of goods and services”. [1] The subjective measurement of the utility value reflects on an acceptable price that the individual is willing to pay for [2]. This acceptable price leads to an idea of the utility function definition, in which a subjective value is dependent on the number of goods or services (e.g., Virtual Machine or VM) to be consumed or provisioned. According to Krugman and Wells [1], different individuals would have different utility functions because different people would have different needs and preferences towards a certain amount of goods or services.

However, this economic term of “utility” is often mixed with other connotations of “utility” so that it becomes quite ambiguous and confusing [3] when a utility function is defined. It is necessary to clarify and differentiate meanings of utility at the outset.

The common sense of utility means “the usefulness of something, especially in a practical way”. For example, the utility of database means to implement various processes or functions of the database, such as batch update, rebuild, recovery, backup, etc. Another sense of utility is quite close to the meaning of the usability that often refers to the state of being useful, which is to supply the essential infrastructure services to the general public. These services are offered by incumbent service providers, which are known as “public utilities” or simply, “utilities”. For example, Buyya et al. [4] argued “cloud computing” is the 5th utility. Still, another meaning of utility is the utilization rate, which is to measure the effective usability of something, such as the network’s utility. Its value is between 0 and 1. Although both network and economic utility may adopt a similar function (e.g., isoelastic and alpha-fair function), the contents of two utilities are totally different.

Economically, the utility function is to describe how people consume various amounts of goods and services in terms of their subjective preferences, needs, and experiences in a less or more rational way. “It is simply a convenient device for summarizing the information contained in the consumer’s preference

[*] Corresponding author.
E-mail address: caesar.wu@computer.org (C. Wu).

https://doi.org/10.1016/j.future.2019.12.044
0167-739X/© 2020 Elsevier B.V. All rights reserved.

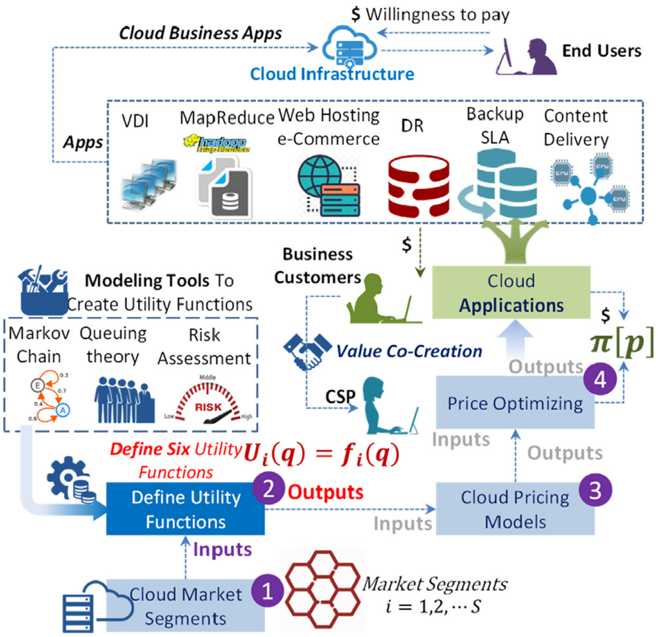

Fig. 1. An overall of cloud pricing strategy.

relation" [5]. This preference is measured by either cardinal or ordinal approaches. "Cardinal" means a marginal value can be quantified by an additional subjective value for one more unit of cloud resources that are acquired. The ordinal approach can only be measured by a ranking method. Our study will adopt a cardinal approach [3] to quantify the cloud utility values because the cloud utility satisfies the criteria of cardinal analysis: (1) the cloud business customers are rational. It means they will systematically and purposefully do the best they can do to achieve their goals, given the available choices, (2) Utility value can be measured numerically in terms of dollar value, and (3) Unit of Infrastructure as a Service (IaaS) is homogeneous. Subsequently, we can define various utility functions in the cloud context.

If a CSP focuses on the business customers, we can consider the customers' utility U_i is equivalent to a cloud customer's business revenue or business income so that a utility function U_i can be defined as a function of the independent variable: "q" (cloud resources or a quantity of VM). Therefore, we can define it as $U_i(q) = f_i(q)$, where i is the number of cloud market segments. This is determined by a market segment assumption and CSP's business and marketing strategy [6].

The focal point of this paper is to create different types of utility functions $U_i(q)$ for various cloud business applications, such as web hosting, content delivery, e-commerce (e.g., an online check out system), database backup, disaster recovery (DR), virtual desktop infrastructure (VDI), and backend processing (e.g., MapReduce, log file analysis). If we assume that the measurement of the business customer's satisfaction (e.g., cloud service metrics) is directly associated with its business revenues, then our modeling process is to estimate how much the customers are willing to pay for a given quantity of the cloud resources that can help them to grow their business revenue or profit. Fig. 1 (π means a profit) provides an overall cloud pricing strategy. There are four necessary steps in the process. The 2nd step highlights the focal point of this paper, which is to define utility functions for CSP to achieve the value co-creation with its business customers or partners.

Fig. 1 provides details on how we create a cloud pricing strategy. The 1st step can be found in our early work of cloud market segmentation [6]. The 3rd step is how to establish multiple cloud pricing models to meet various customers' requirements [7]. The 4th step is to identify the optimal price point of a pricing model for a CSP to achieve profit maximization [7].

According to Nagle et al. [8], the 2nd step is a challenging task because the issue requires multidisciplinary knowledge. Many previous types of research either ignored some parts of the problem or failed to articulate the problem clearly. Consequently, the issue of defining a utility function becomes hard to be implemented in practice. Many previous solutions of utility modeling [9–16] often assume a uniform market or "one size fits all". Moreover, the meaning of utility is often mixed with the demand side of the price and supply side of cost.

To overcome these challenges, this study will clarify the meaning of the cloud customers' or demand side's utility for the cloud services, which is a subjective value measurement for the number of VMs to be provisioned. Often, this value can be represented by cloud customer's experiences (CX) or key performance indicators (KPI) or cloud service metrics (CSM). Both the National Institute of Standards and Technology (NIST) [17] and Oracle [18] have defined CX, KPI, and CSM along with three actual business dimensions that consist of acquisition (increase in sales), retention (monetize relationships), and efficiency (leverage investments). All of the quantitative measurements of business dimensions can be translated into the value of business revenue or profit.

Overall, the core problem of this study is "how to quantify the cloud customer's satisfaction (the business revenue or profit) along with a variation of "q" in each market segment" By defining multiple utility functions related to the segmented cloud market, we provide a novel solution for cloud price modeling that is much more realistic and practicable.

1.1. The advantages of our solution

In comparison with previous solutions, such as empirically calibrated price [19] and capacity [20], resource optimization [12], response time, capacity-aware [21], utility-based-SLA [22], and model-based [9], our solution has a number of advantages:

(1) It is practical and quantifiable for real cloud applications in term of resource needs (internal rationality),
(2) It can be implemented by any CSP for its targeted market,
(3) The utility is derived from the principle of economics
(4) It is agile and flexible to cope with a CSP's business strategy changes,
(5) The utility functions are defined to improve the cloud customer's revenue (or external rationality).
(6) It is a process of value co-creation for both CSP and cloud customers.
(7) It provides a solution for CSPs to achieve more profits by optimizing different cloud price models.

Based on the listed advantages, we have made the following contributions to cloud economics.

1.2. Our contributions and paper organization

(1) To the best of our knowledge, this is the first such study to define multiple utility functions based on the segmented cloud market assumption. It models the utility functions from the value co-creation perspective.

(2) The utility value directly impacts the business revenue or profit of cloud business customers because the utility values are defined by the cloud customers' KPI or CX or CSM (value proposition). All the utility values can be translated into a single and quantifiable unit rather than some indirect or multiple

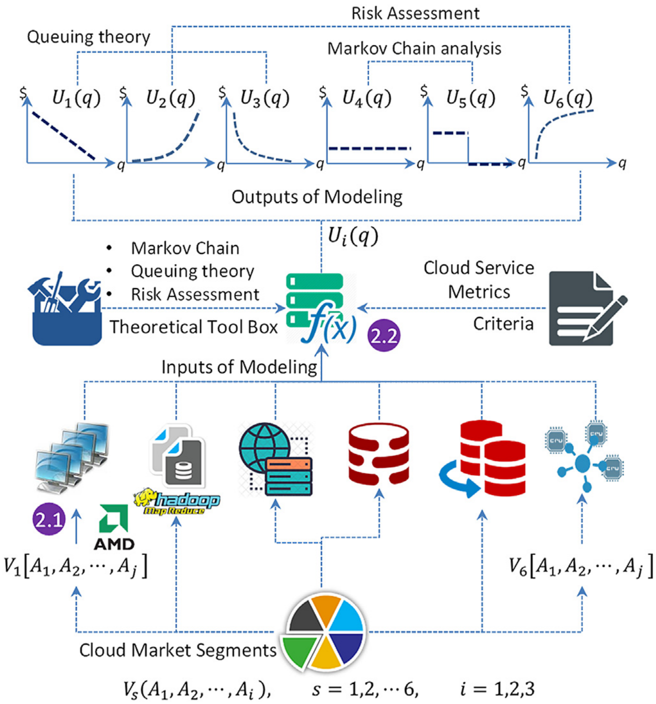

Fig. 2. The approach to modeling cloud customer utility functions.

unquantifiable units. The cloud customer's utility (or subjective) values become the function of the cloud resources or VM.

(3) We use Markov chain analysis to quantify the number of VM needs in terms of the specified SLA for the high availability (HA) applications, such as database backup, CRM, and terminal server. It provides the deliverable SLA and uptime for customers of segments 4 and 5 (Refer to Table 3) to generate their business revenue or profits. It translates SLA into customers' business revenue rather than to use SLA as a direct independent variable for the utility function definition.

(4) By leveraging the queueing theory, we create customers' utility functions that can minimize the response time to deliver the right performance of business applications, such as online checkout system and web-hosting for Segment 1 and 3. This response time length is also translated into customers' revenue contributions.

(5) In addition, we leverage the concept of risk-aversion and risk-taking to model the customers' utility values for content delivery (Segment 2) and log file analysis or MapReduce applications (Segment 6) to maximize the end-users' satisfaction in terms of maximizing application functionality and minimizing cloud running costs.

Overall, our solution can be easily comprehended for CSP decision-makers when they want to make a critical investment decision for their cloud business. Fig. 2 highlights the details of the process of building utility functions. This process illustrates how to define six utility functions by three modeling tools. In summary, Fig. 1 gives an overall cloud pricing strategy, and Fig. 2 shows a novel approach of defining various customers' utility functions in detail.

The rest of the paper is organized as follows: Section 2 gives a brief literature review regarding previous modeling solutions for utility functions. Section 3 presents how we define multiple utility functions based on the result of six market segments and how we make various assumptions, define the problem, and determine the scaling coefficient and other parameters. Section 4 provides a detailed performance evaluation and validation of our modeling method. Section 5 offers some simple guidelines for CSPs on how to select these utility functions. Section 6 provides both analysis and justification for some assumptions. Section 7 outlines our conclusions and future work. To navigate the discussion easily, Table 1 lists all acronyms used in this paper.

2. Related work

The modeling customer utility functions for various hosting services or applications can be traced back to the beginning of the dotcom-booming era. Doyle et al. [9,23] proposed a model-based approach to optimize the hosting of hardware resources for the specified SLA. The goal of their work is to demonstrate how to provision the server resources for web hosting applications effectively. Although the paper adopted the term "utility" and made good progress in hosting service modeling, the real meaning of utility is the usefulness of server functionality rather than an economic sense of subjective measurement for customer satisfaction. Similarly, Appleby et al. [10] proposed an SLA-based management system, which is named "Oceano" for e-business. It was based on a set of predefined metrics that consists of seven parameters. Their approaches could be considered as a policy-based scheme for computer resource allocation. The policy mainly reflects a supply side's view of value proposition rather than the explicit measurement of the demand side's experiences.

In contrast to Appleby, Walsh et al. [11] gave an explicit measurement for the customer's utility functions in order to automate the computer resource distribution. The utility functions are an autonomic scheme to manage web hosting workloads running on a Linux cluster. They defined the utility *U(S,D)* as a function of two independent variables: service level (S) and current demand (D), which is measured by an average of forecast demand D′. S is a function of the other three independent variables that are control parameters (C), which is responsible for optimizing the utility *U(S,D)*, current resource level (R), and demand (D). Overall, the customer utility value can be estimated by variables of C, R, and D′ and defined as Eq. (1) if the service performance (S) is specified.

$$\widehat{U}\,(R) = \max_{c} U[S\left(C, R, D'\right), D'] \tag{1}$$

This service performance is designed to run the application of IBM WebSphere and DB2. The paper argued that the utility $\widehat{U}\,(R)$ is defined by a sigmoid function in terms of the average response time. Based on the context of the paper, the definition of $\widehat{U}\,(R)$ the utility may mean an estimated utilization rate rather than an economic sense of utility. The value is between 0 and 1. Although they did some pioneering works regarding utility functions. However, the authors left the details on how to model the sigmoid function in a practical way. In comparison, Bennani et al. [12] gave some details for their proposed utility as a sigmoid function (Eq. (2)) regarding online application environments.

$$U_{i,s}\left(R_{i,s}, \beta_{i,s}\right) = \frac{K_{i,s}e^{-R_{i,s}+\beta_{i,s}}}{1+e^{-R_{i,sthe}+\beta_{i,s}}} \tag{2}$$

where, $K_{i,s}$ is a scaling coefficient. $U_{i,s}$ is the function of $\beta_{i,s}$, $R_{i,s}$, $R_{i,s}$ is the response time for "i" type of application environment (equivalent to a type of workload) with "s" type of classes of transactions (equivalent to a type of virtual machines), $\beta_{i,s}$ is desired or targeted SLA (equivalent to the customer performance metrics). The goal of their paper was to come up with a solution (or global with controller) that can automatically assign different types of workloads to the adequate size of the server for the data center infrastructure. The value of their utility function varies between 0 and 1. Its scaling coefficient is corresponding to the upper bound of the throughput of the job or workload

Table 1
Acronym used in this work.

Acronym	Definition	Acronym	Definition
B2B	Business To Business	IaaS	Infrastructure as a Service
B2C	Business To Consumer	IOPS	Input/Output Per Second
Capex	Capital Expenditure	KPI	Key Performance Index
CARA	Constant Absolute Risk Aversion	M/M/1	Markov/Markov/single server
CRRA	Constant Relative Risk Aversion	M/M/S	Markov/Markov/multiple servers
CDN	Content Delivery Network	NFS	Network File Sharing
CPU	Central Process Unit	NIST	National Institute of Standard and Technology
CRM	Customer Relationship Management	AOS	Application Objective Server
CS	Customer Surplus	OLTP	Online Transaction Processing
CSM	Customer Service Metrics	Opex	Operational Expenditure
CSP	Cloud Service Provider	PAYG	Pay as You Go
CX	Customer Experience	PoC	Proof of Concept
DRaaS	Disaster Recovery as a Service	PoS	Point of Sales
DR	Disaster Recovery	QoS	Quality of Service
EDI	Electronic Data Interchange	SLA	Service Level Agreement
FaaS	Function as a Service	SME	Small to Medium Enterprise
GA	Genetic Algorithm	URL	Uniform Resource Locator
GB	Gig Byte	VDI	Virtual Desktop Infrastructure
HA	High Availability	VM	Virtual Machine

completion for the particular application. The authors assume the higher throughput, the higher utility value is. Consequently, the meaning of utility has become "utilization" or "utilization" rate of IT resources.

Following a similar line of reasoning, Kephart et al. [13] proposed a self-management system that is based on the utility framework in order to achieve resource efficiency in a prototype of a data center. The value of the utility function is between -1 and 1. The independent variables of the utility function could be either response time or the number of physical servers. Menache et al. [13] further developed this idea and proposed a long-term solution for cloud computing resources in terms of maximizing the social surplus, which was the aggregated individual user's utility of executed jobs minus workload-dependent operation expenses (Opex). This social surplus is equivalent to Bennani's [12], the global controller. The individual utility function (or local controller) of each user is presented in Eq. (3):

$$U_i\left(z_i\right) = V_i\left(z_i\right) - Pz_iT_i\left(z_i\right) \tag{3}$$

where, $V_i\left(z_i\right)$ is the value that user (i) assigns to executing job required z_{i} amount of resource for P unit price for mean service time, $T_i\left(z_i\right)$. Although it was just a theoretical discussion, their work was the first time to define the utility in an economic sense. However, some assumptions need to be further consolidated. For example, the assumption of M/G/∞ model could mean no resource restriction. If this is a case, the optimal solution could become impracticable. Just as the authors highlighted, their analytic model only provided a convenient starting point for future research topics of cloud computing, such as revenue, profit, and pricing. Nevertheless, the paper indicated there is an optimal point by a linear usage based-tariff (or fixed price/per unit resource/unit time).

Weintraub et al. [24] presented a survey plus ranking (ordinal utility) model that is a conjoint analysis (ranking multiplied by the weighted coefficients) that shows how to maximize the user's total utility from a set of cloud services offered by CSPs. This utility model is the cloud feature or characteristics-based services for selecting a preferred CSP. It is a utility model in terms of customer preference choice.

Regarding the preference choices, Burda et al. [14] examined consumer preferences for the cloud archiving services from a student's perspective. Burda et al. adopt a conjoint analysis (a survey-based statistical technique) to quantify customers' utility levels based on the customers' demographic parameters, such as age and gender. Although the authors' did not explicitly adopt the term of market segmentation, the paper was the first one to introduce a concept of pricing discrimination. However, their study focused on business to consumer (B2C) market rather than business to business (B2B) market.

Minarolli et al. [15] adopted a similar approach as Bennani et al. [12], which was to set up both local (like a transponder) and global controllers (or a central management system) to allocated cloud resources pool. They defined the utility function was a simple linear Eq. (23), which was also the extension work of Walsh [11] and [16]:

$$U_i = \alpha_i \cdot S_i \tag{4}$$

where the amount of dollar α_i is paid per unit of CPU resource, and S_i is the located CPU resource to VM$_{\mathrm{i}}$ or shared physical CPU utilization. However, this resource consumption model is just one of the utility functions if the cloud customers take a risk natural attitude. One of the controversial issues was that the meaning of utility was not the economic sense of utility. The unit of the utility measurement was moving between the quantity of VMs and the length of response time.

Similarly, Garg et al. [25,26] provided an admission control solution for a similar problem, but they described the term of utility as a resource scheduling rather than a subjective measurement. The assumption of their application was the non-interactive or static workload. Their solution was to achieve the optimizing scheduling between the specified quality of service (QoS) requirements and resource provisioning.

In comparison with others, Chen et al. [22] made some contributions to the utility definition in terms of subjective measurement. They presented scheduling solutions from a cloud customer's utility perspective. They showed that cloud customers could effectively bid for different types of VM spot instances under the conditional metrics of profits, customer satisfaction, and cloud resource utilization. The detail of the cloud customer utility function is shown as follows:

$$U\left(p, t\right) = U_0 - \alpha p - \beta t \tag{5}$$

where, U_0 is the maximum utility that the service delivery to the customers. It is proportional to the size of the service request. Both α and β is the coefficient of price "p" and response time "t". Again, the cloud customer utility value is a linear function of price and response time. The application of their utility function is to run the tasks of x264 video scripts' encoding and decoding. Overall, we can highlight the main contributions and gaps of each utility modeling solution of previous works in Table 2.

From Table 2, we can see that the majority of previous works are more like a resource management scheme with the aim

Table 2
Summary of all methods of modeling utility function.

Modeling methods	Modeling idea	Main contributions	Gaps	Application
Model-based [9,23]	Slices of computer resources, & time for resources	Enable a provision of multiple resources in an interactive way	The concept of utility is not the economic one	Web-based service or CDN
SLA metrics [10,22]	Leveraging SLA to allocate resources	Dynamic, flexible, scalable resource allocation	The resources assumption has no limitations.	e-business hosting
Resource-based [11,12,15,22,25,26]	Utility function to allocate resource	Self-optimization of computing capability	A data center management system for resource utilization	Data center, CDN, Video streams
Social surplus-based [13]	Adopting social welfare idea & queueing theory	The social effects of using cloud resources	A theoretical model. Similar to global and local controls	Academic discussion
Empirically calibrated [19]	Empirically calibrated model	Provider an alternative way of utility modeling	Limited applications remain empirical	To explain major cloud leaders' market behaviors
Price-quality [20]	Single and multi-tiered solution	Define the price-quality from nash equilibrium perspective	Only apply it for a particular case	Theoretical interpretation
Capacity-aware [21]	Non-additive utility function	Dynamic	Mixed with users and CSP utilities. Pre-negotiation of deliverable SLA	Negotiable cloud resources or Grid computing
Conjoint analysis [14,24]	Three layers of customer utility	Survey plus ranking	Arbitrary to build the utility function	Cloud customers survey data
Framework-based [27]	Utility function policies	Two types of integrated utility values	Prototype	Data center environment
Simple linear [11,15,16]	The two-tier resource management	The balance between QoS and operation cost	Mixing with CSP and customers utility	VM resource allocation

of managing cloud resources. Strictly speaking, many models did not define cloud customers' utility functions based on the cloud market segmentation. The term "utility" was not defined as a subjective value that can be measured from a cloud customer perspective. The meaning of utility was often swinging between supply and demand. To bridge this gap, our utility modeling process begins with a real-world scenario of cloud business development.

3. Modeling utility functions for cloud applications

3.1. Prior studies and background

To illustrate our modeling process clearly (Shown in Fig. 2), we can consider a case of how a hosting firm to develop its cloud pricing strategy for its new cloud business. Suppose decision-makers of the firm (supply side) decides to expand its traditional hosting business to a cloud market for its business customers (demand side). The goal of the firm is to grow both revenue (market) and profit with a fixed amount of investment budget.

We also assume that the firm understood its own technical capability or expertise and specified its targeted cloud customers. The subsequent issue is how to segment the B2B market, which is to identify potential demand (or the addressable market) for new cloud services. The purpose of segmenting the market is to find an effective solution to serve its targeted customers well and achieve a maximum profit for a given limited resource. Ideally, the CSP should make every customer pays a different price so that it can extract the maximum utility value from each customer [28]. This pricing strategy is known as the perfect or the first order of price discrimination. But it would be too costly to implement because of the higher cost of sales. The alternative way is to group targeted customers who have the same characteristics or similar demands together. This idea leads to the process of "market segmentation".

3.2. Assumptions

3.2.1. Market segment assumptions

From our previous work [6], we can find that there are six possible cloud market segments based on the various parameters or characteristics of cloud customers' usage patterns. Fig. 3 shows the result of the cloud market segmentation (a clustering dendrogram). The decision to adopt the scenario of six market segments can be justified by the following criteria:

(1) The optimal number of market segments is between 4 and 8, which is identified by a hierarchical clustering method [6].
(2) McDonald [29] suggested that the optimal number of the market segment should be between 5 and 10.
(3) We assume the firm is a traditional hosting company that wants to explore a limited number of market segments with limited investment capital.
(4) The cloud business strategy intends to avoid some high risks in some new or niche market segments.

If the firm's cloud business strategy wants to meet all the above criteria, the number of market segments should be around six (Shown in Fig. 3) because criterion 1 suggests the number of market segments between 4 and 8 and criterion 2 suggests between 5 and 10, while the criteria 3 and 4 indicate the segments at the lower end. The details of how to use a hierarchical clustering method to segment the cloud market and how to define each cloud market segment can be found in our previous work [6]. Table 3 shows the result of each market segment based on various cloud usage parameters.

The six market segments are just one of the business scenarios. If a firm is one of the existing CSPs that has more investment budget and attempts to take the risk of a niche market (See Fig. 14), it can decide to have more than six market segments. The bottom line is that the CSP should clarify its cloud business strategy and targeted customers or market, competitive service, projected business revenue, and anticipated profit margin.

3.2.2. Mapping cloud application to a segment

From the result of the cloud market segment, we can also map onto cloud customers' resource usage patterns with a particular business application (e.g., web hosting, e-commerce, database backup, log file processing, content delivery, etc.). Consequently,

Table 3
Cloud utility functions and market segments.

Utility functions	$U_1(q)$	$U_2(q)$	$U_3(q)$	$U_4(q)$	$U_5(q)$	$U_6(q)$	Total[a]
Market segments	S_1	S_2	S_3	S_4	S_5	S_6	
Example of market segments or applications	Virtualized desktop infrastructure, email server	MapReduce, log analysis file & print	Web hosting server & Online checkout	Disaster recovery	Database backup & Terminal server, SLA	Dynamic content delivery, terminal workload	
Ave. Job priority	1	0	2	0	3	3	
Ave number of cores	2	23	1	1	3	13	
Ave size of memory (GB)	7	6	6	3	86	102	
Percentage	30.1%	23.0%	10.0%	26.3%	9.1%	1.4%	100%
Demand	269	205	90	235	81	13	893

[a] "Total" is the sum of all addressable sales volume for a particular type of virtual machine. It also represents the total percentage of all market segments. The percentage of each market segment is extracted from Google's dataset. If a local CSP has its own forecast sales volume, the CSP can work out addressable sales volume for each market segment.

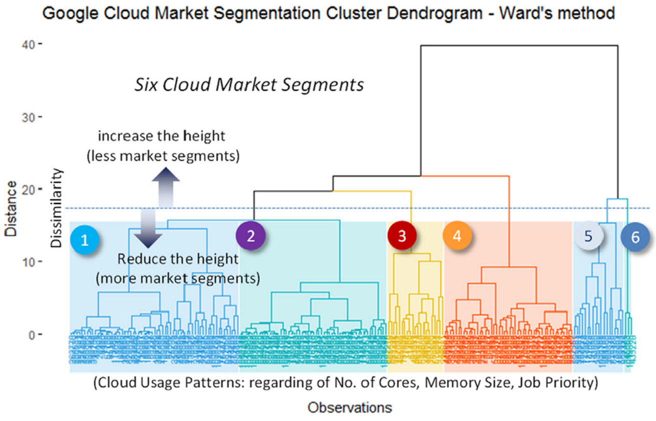

Fig. 3. Proposed six cloud market segments.

we approximately have a linkage between each cloud market segment and a particular cloud application [6]. The mapping process is mainly dependent on the parameters of market segments such as job priority, an average number of cores, and memory size, as shown in Table 3.

The "q" of Table 3 is an independent variable to represent the number of VMs. We also assume the maximum number of VMs that the cloud customers will purchase for a particular type of VM is less than "q_m". According to [30], the definition of a job priority means the critical level for workload scheduling. The larger number is, the high priority should be. For example, if a higher priority job requests for a computational resource, a lower priority job will be killed or stopped. The lower priority VM is equivalent to Google Cloud Platform (GCP) preemptible virtual machine (VM) or Amazon Web Services or AWS' spot instance, in which a cloud customer is willing to take the risk for the job to be interrupted rather than to pay a higher price VM.

AMD [31], Young [32], Michalski [33] Feitelson [34] and Calzarossa [35] provided some guidelines to identify some common cloud application workload patterns. Notice that Google's public cloud trace or dataset [36] only released the limited number of parameters due to some commercial reasons. Therefore, the mapping process can only depend on some available parameters. If a CSP has its own operational dataset, which may include more parameters (e.g., network bandwidth, storage size, and cache memory), the mapping process will become much accurate. Practically, the more parameters of a market segment have, the accurate mapping process becomes. The basic idea of this process is to estimate a possible type of workload based on a profile of the cloud market segment.

3.3. Problem definitions

From Fig. 1, we can have an overall picture of how to establish a cloud pricing strategy for cloud business. The 1st step is how to segment a cloud market based on an available dataset. Its inputs are the parameters of each segment. Its output is the number of segments and a linkage between market segments and possible cloud applications. The 2nd step is to leverage the information from the 1st step to create utility functions for various market segments. This is the research problem for this paper.

With some prior knowledge of some cloud applications [6], we can simplify the modeling process into three categories based on the specified cloud customer's metrics (CSM), parameters of cloud market segments, and cloud workload patterns [32,37]. The detail of the modeling process is shown as follows:

(1) Modeling the utility functions of backup or terminal server for segment 4 and Disaster Recovery (DR) for segment 5. This category focuses on the specified service level agreement (SLA) and business continuity metrics.
(2) Creating the utility functions for the data processing of Online Checkout or web hosting application for segment 3, Virtual Desktop Infrastructure (VDI) for segment 1. This category is dependent on response time.
(3) Constructing the utility functions of backup — dynamic content delivery for segment 6 and MapReduce workloads for segment 2. This category is to concern the customer's attitude towards risk.

The reason to group some segments together for the modeling process is that each pair has some common characteristics of workload patterns so that we can simplify the modeling process. For example, database backup and disaster recovery (DR) can be considered as the same category because both backup and DR may need high availability (HA) cloud infrastructure. We can use the detail modeling process to clarify our argument.

3.4. Utility functions for HA workloads

HA workloads may also be considered as a mission-critical business application. These workloads often require redundant cloud infrastructure, e.g., backup servers (or VMs). If we assume the downtime should be less than 5 min/per annum, then SLA must be higher than five-9s (or 99.999%). It means any

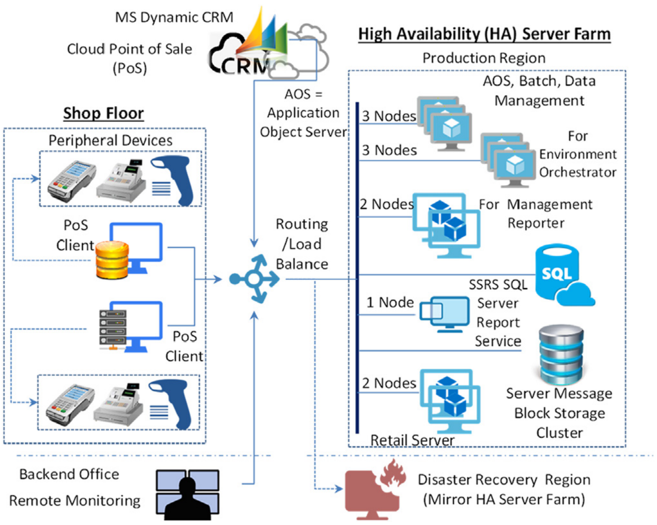
MS Dynamic CRM
Cloud Point of Sale (PoS)
CRM
High Availability (HA) Server Farm
Production Region
AOS = Application Object Server
AOS, Batch, Data Management
3 Nodes
3 Nodes
For Environment Orchestrator
2 Nodes
For Management Reporter
Shop Floor
Peripheral Devices
PoS Client
Routing /Load Balance
SQL
SSRS SQL Server Report Service
1 Node
PoS Client
2 Nodes
Server Message Block Storage Cluster
Retail Server
Backend Office
Remote Monitoring
Disaster Recovery Region (Mirror HA Server Farm)

Fig. 4. A typical architecture of CRM.

failure of cloud infrastructure would lead to a catastrophic consequence [38] for a running business. One of the examples (Fig. 4) is a Customer Relationship Management (CRM) system (e.g., Seibel, SAP, Microsoft Dynamic CRM), financial portal (e.g., e-Trade), online banking platform, fast delivery ordering application, and etc.

Suppose a cloud business customer hosts a CRM application on a cloud platform that is offered by a CSP. It means the cloud supports this mission-critical application shown in the Fig. 4 [39,40] (Here, nodes are equivalent to a server or a VM. We use these terms interchangeably). The CRM system consists of the front interface (or shop floor) and backend cloud infrastructure (or an HA server farm). The server farm or a VM cluster is the critical cloud infrastructure to support CRM application.

If one of the VMs (or nodes) fails, its workload will automatically failover to another node. If one VM cluster fails, its workloads can be automatically failover to another VM cluster shown as in Fig. 5. Here, we assume the faulty VM is equivalent to a hosting server or infrastructure failure so that each fault is an independent event. In other words, a cluster of VMs will be deployed on various servers. On the other hand, if a cluster of VMs is built up one large server hardware, one hardware failure will propagate into the entire cluster. We exclude this scenario from our model assumptions. Furthermore, to simplify our modeling process, we also exclude the failure of cloud software such as hypervisor and application software. We will discuss these cases separately.

There are two types of high availability (HA) applications. One is Disaster Recovery (DR) that two possible VM (or node) clusters are physically located in different data centers. The other application is that a cluster of VMs is built in one data center, but allocated in multiple nodes. If one node is down, its workload can be automatically failover to other node shown in Fig. 4 within one data center. We can consider this case as an SLA-driven application. Looking from a quantitative perspective, the difference between these two HA applications is that DR needs more nodes because it mirrors the entire operational environment, and SLA-driven HA only needs a specified number of nodes.

If we assume a VM failure rate is μ (assume each VM is allocated in the different physical machines), its restoration rate is λ; the question is how many VMs are needed to support the customers' mission-critical application.

We can use the Markov chain analysis tool for this problem. Assume we need k VMs to support the requirement of five 9s SLA.

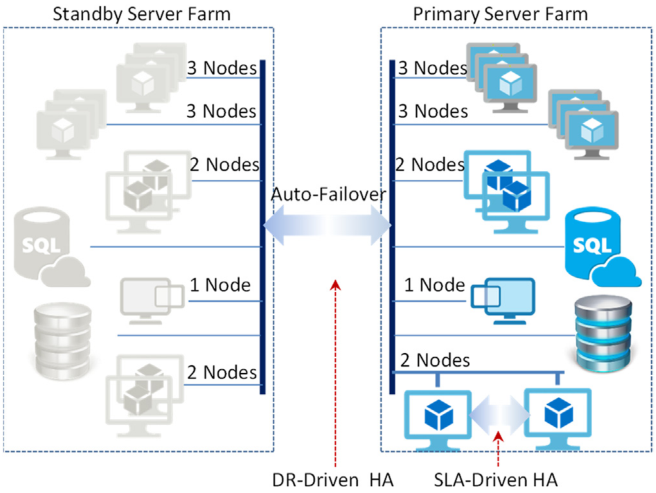

Fig. 5. Disaster recovery for VM cluster failover.

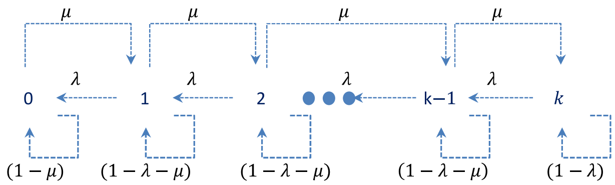

Fig. 6. SLA driven high availability system for "k" number of VMs.

$$
\begin{array}{c|c}
 & \begin{array}{ccccccc} \mathbf{0} & \mathbf{1} & \mathbf{2} & \cdots & \mathbf{k\text{-}2} & \mathbf{k\text{-}1} & \mathbf{k} \end{array} \\
\hline
\begin{array}{c} \mathbf{0} \\ \mathbf{1} \\ \mathbf{2} \\ \vdots \\ \\ \mathbf{K\text{-}2} \\ \mathbf{K\text{-}1} \\ \mathbf{k} \end{array} &
P = \begin{bmatrix}
1-u & \mu & 0 & 0 & 0 & 0 & 0 \\
\lambda & 1-\lambda-\mu & \mu & 0 & 0 & \vdots & \\
0 & \lambda & 1-\lambda-\mu & \cdots & \vdots & & \\
0 & 0 & \lambda & \ddots & 0 & & \vdots \\
\vdots & \vdots & 0 & 0 & \mu & 0 & \\
 & 0 & \vdots & \cdots & 1-\lambda-\mu & \mu & 0 \\
0 & 0 & 0 & 0 & \lambda & 1-\lambda-\mu & \mu \\
0 & 0 & 0 & 0 & 0 & \lambda & 1-\lambda
\end{bmatrix}
\end{array}
$$

Fig. 7. Markov chain probability matrix with "k" of nodes or VMs.

We can have a probability matrix, as shown in Fig. 7 based on the above assumptions. The k number of VMs can form a Markov chain system. This system is ergodic because we can verify the number of steps of the system would be exact "$k+1$" transitional states from any state to any other state. It means that the process can be characterized as a steady-state vector for a long run [41].

According to the Markov chain matrix (shown in Fig. 7), we should have a $(k+1)\times(k+1)$ Markov chain probability transitional matrix. From this matrix, we can derive a steady-state vector illustrated in Eq. (6):

$$V = [V_0, V_2, V_3, \dots, V_{k-1}, V_k] \tag{6}$$

If we assume a failure probability μ of a VM or physical server [41] is 0.004, and a faulty restoration rate λ is 0.2, we can calculate the result of the steady-state vector from a transition probability matrix (number of VM failed) (shown in Eq. (7))

$$V = [0.98,\ 0.0196,\ 0.000392,\ 7.84E{-}06\ ] \tag{7}$$

The calculation result illustrates the number of hot-standby VMs is at least three nodes if we want the specified SLA is higher than five 9s and the failure rate μ is 0.04, the number of VMs should be six.

To generalize this transitional matrix, we can define a function V_k as a probability of k VMs is down. We want to find the minimum number of k such that the probability of this downtime is less than a specified time ϵ (e.g., five minutes/per annum) shown in Eq. (8).

$$V_{(k-1)} * \mu \leq \epsilon \tag{8}$$

Based on Figs. 6 and 7, we can step-by-step derive Eq. (13) from Eqs. (8), (9) (10), (11), and (12). We can use Eq. (13) to calculate the k value, which is to decide the minimum number of VMs.

$$V_i = V_0 \left(\frac{\mu}{\lambda}\right)^i, \qquad \alpha = \frac{\mu}{\lambda} < 1 \tag{9}$$

$$V_k = V_0 \alpha^{k-1} \mu \leq \epsilon \tag{10}$$

$$V_0 = \frac{1-\alpha}{1-\alpha^{k+1}}, \qquad \frac{1-\alpha}{1-\alpha^{k+1}} \alpha^{k-1} \mu \leq \epsilon, \tag{11}$$

$$\alpha^{k-1} \leq \frac{\epsilon}{(1-\alpha)\,\mu + \epsilon\alpha^2} \tag{12}$$

$$k \geq \left\lceil 1 + \frac{ln\epsilon - ln\left[(1-\alpha)\,\mu + \epsilon\alpha^2\right]}{ln\alpha} \right\rceil \tag{13}$$

where any V_i is a probability distribution vector in the ergodic system, V_0 is the initial state of the probability distribution vector. V_i also indicates the probability that the system had i failures and now using the resource (i + 1). $\epsilon = 1 - 0.99999$ (Five-9s: a specified SLA).

Note that Eq. (6) defines the probability transition from the $k-1$ state (the last VM) to the k state (or all VMs failure). The k value of Eq. (13) should be round up to the up ceiling that is not less than k. Once we have the result of k, we can define two customers' utility functions for market segment 4 or DR-driven HA and segment 5 or SLA-driven HA shown in Fig. 4.

For the SLA-driven HA system, we can consider all VMs have an equal utility value to contribute to cloud customer's business revenue or profit equally. For example, if six nodes can guarantee five 9s SLA delivery for a CRM application that generates a profit of $9/per hour, then each node should contribute $1.5/per hour. As a result, we can use a discrete function to represent the customer's utility for segment 5 shown as follows (Eq. (14)):

$$U_5(q) = \begin{cases} K_5, & 1 \leq q \leq k \\ 0, & k < q \leq q_m \end{cases} \tag{14}$$

where "K_5" is a revenue coefficient value, "q" is the variable of number of VMs, q_m is the maximum number of VMs (Refer to Section 3.2). The interpretation of Eq. (15) is that the cloud customers will only purchase the k number of VMs to meet their SLA requirements. If the number of VM is higher than k, the value of a VM for its revenue contribution will be diminished to zero. If a CSP's market strategy is to target Small Medium Enterprise (SME), then we can define the scaling coefficient K_i values by Eq. (15).

$$K_i = B_i / \left(\sum_{q=1}^{q_m} u_i[q]\right), \quad i = 1 \cdots S \tag{15}$$

where B_i is the annual revenue in each market segment of different categories of Small Medium Enterprise (SME) [42]. "S" the maximum number of market segments, and "i" is a variable of the market segment.

For segment 4, it can also be considered as another type of mission-critical workload for business continuity because we have concluded this segment is to run DR or DR as a Service (DRaaS) applications. According to Luetkehoelter [43], the definition of DR is "*the process of mitigating the likelihood of a disaster and the process of returning the system to a normal state in the event of a disaster*". It can be considered as one type of the HA workload, which is similar to a database backup (See Fig. 4). In comparison with SLA-driven, DR-driven often requires more VMs than normal SLA-driven, but this requirement is dependent on a "likelihood" of the disaster event.

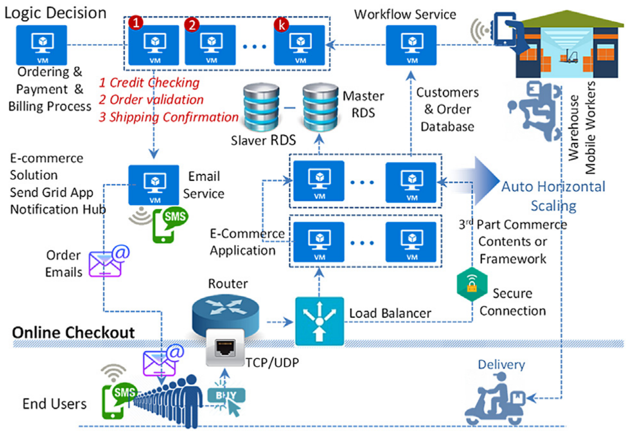

Fig. 8. Typical architecture of online checkout or payment processing.

We can use a likelihood coefficient θ to describe this riskiness [44] in term of business impacts for the maximum quantity q_m of VMs. This risk assessment should be determined by a business continuity plan. We can formulate Eq. (16) for the cloud customer's utility value.

$$U_4(q) = \theta K_4, \quad \theta \in (0,1), \quad 1 \leq q \leq q_m \tag{16}$$

where θ is a potential risk rate (a percentage) to impact the cloud customers' revenue when a disaster occurs. The value is between 0 and 1. Practically, this equation means that the cloud customers will only purchase the q_m number of VMs when the price of VM (p) is below the specific threshold level of their utility value (e.g., $\theta K_4 > p$). If the VM price is higher than their utility value for a likelihood disaster, they will stop to purchase cloud resources from CSPs and build the on-premises infrastructure. In addition to the mission critical applications, the utility function of e-commerce application can also be modeled by a Markov chain process.

3.5. Utility functions for online checkout and web hosting workload

E-commerce applications, such as shopping cart, electronic data interchange (EDI), online catalogs, consist of a business processing module [45], which can be characterized as queueing workload patterns [46]. One of the typical examples is the online checkout (payment) processing system shown in Fig. 8 for a web hosting service. It merely means that the end-users are lining up a queue for checking out due to online purchasing or ordering. The cloud platform that supports the online checkout application consists of the back-end of a VM cluster (or a server farm) and some auto-horizontal scaling VMs and the workflow service.

Assume there is only one virtual machine (VM) that has been allocated to handle the end-users' checkout requests (λ_1) with a specified process capacity (μ_1), we would like to know how long (w_1) the end-users have to wait to complete the checkout process. Here, the checkout request or arrived rate λ_1 is determined by a Poisson distribution, and the expected service time T is assumed to follow an exponential distribution. We can use the simple M/M/1 [41] to model [47] this process, which we can calculate the expected waiting time for the end-users (online purchasers) from Eq. (17):

$$w_1 = E[T] = \frac{\lambda_1}{\mu_1(\mu_1 - \lambda_1)} + \frac{1}{\mu_1} = \frac{1}{\mu_1 - \lambda_1} \tag{17}$$

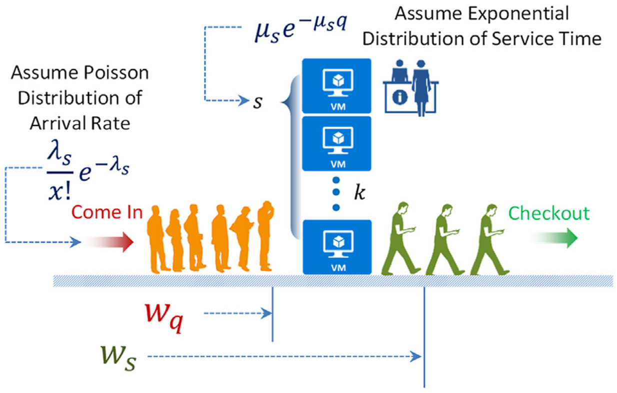
Assume Exponential Distribution of Service Time
$\mu_s e^{-\mu_s q}$
Assume Poisson Distribution of Arrival Rate
$\frac{\lambda_s}{x!} e^{-\lambda_s}$
s
VM
VM
k
VM
Come In
Checkout
w_q
w_s

Fig. 9. M/M/S queueing model.

where $E[T]$ is the total expected time for an end-user within the checkout system, which includes the waiting time to be processed for checkout, this expected waiting time is critical for the cloud business customer who runs an e-commerce business (e.g., an online shop). If the time is too long, the end-user will start to lose patience and switch to another portal (online) shop at the click of a finger. In other words, the expected waiting time will impact the cloud customer's e-commerce business revenue. On the other hand, if the business allocates too many VMs resources to the checkout system, many VMs will stay idle. It will increase cloud business' operation expenditure (Opex). The issue is how to model an adequate utility function to describe the hosting business value.

If we assume the average arrived rate of the end-user as $\lambda_1 = 8$/per hour, and servicing rate $\mu_1 = 10$/per hour [48], the expected average waiting time will be 24 min in the queue. If we include the average 6 min of the processing time for checkout (payment), a random end user will spend the total of average 30 min in the system shown in Eq. (18):

$$w_1 = \frac{1}{\mu_1 - \lambda_1} = \frac{1}{10-8} \times 60 = 30 \text{ min} \tag{18}$$

Based on our online shopping experiences, 24 min of queueing time would be unacceptable. To reduce this expected waiting time, we have two possible solutions: one is the vertical scaling, which is to increase the VM's capacity μ_1 by selecting large capacity VM so that the time of the checkout process can be reduced. For example, if we double the VM capacity $\mu_1 = 20$/per hour, the waiting time w_1 can be reduced to about 5 min. The other solution is the horizontal scaling that is to add more VMs with the same capacity of VM into the checkout system, which can also decrease the queueing time w_q. If this is a case, the problem of M/M/1 becomes an M/M/S [47] model, which can be described in Fig. 9.

If the workload of e-commerce applications is highly fluctuant, then horizontal scaling is a preferred solution. It also adds a bonus of the high availability (HA) into the system, which we have illustrated this point in the previous Section 3.4. Moreover, the different end-user might have different lengths of responding time to the checkout system. For example, a new end-user may take more time to respond to the checkout system than a frequent user.

If we select a horizontal scaling solution, then Erlang's delay formula [47] (Eqs. (19) to (21)) can calculate both queueing and the total processing time (w_q and w_s) for the number of VMs required.

$$w_q = \frac{\alpha^s p_0}{s!\, s\mu_s (1-\rho)^2} \tag{19}$$

Table 4
Calculation results for M/M/S model.

"s" No of VMs	ρ	p_0	w_q (min)	$1/\mu_s$ (s)	w_s (s)
1	0.800	1	1440	360	1800
2	0.400	0.4285714	68.57143	360	428.6
3	0.267	0.4471545	8.514412	360	368.5
4	0.200	0.5020080	1.204819	360	361.2
5	0.160	0.5392432	0.150254	360	360.2

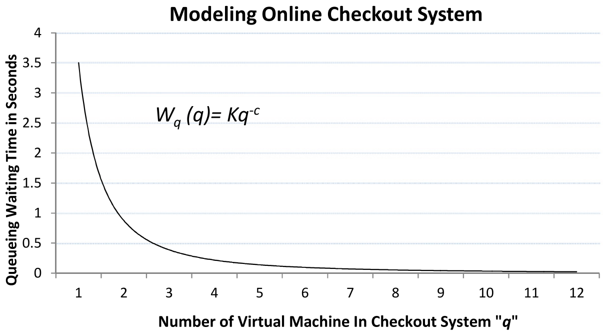

Fig. 10. Modeling utility function for market segment 3.

$$p_0 = \left[\sum_{k=0}^{s-1} \frac{\alpha^k}{k!} + \frac{(\alpha)^s}{s!} \left(1 - \frac{\alpha}{s}\right)^{-1} \right]^{-1} \tag{20}$$

$$\alpha = \frac{\lambda_s}{\mu_s} < 1, \qquad \rho = \frac{\alpha}{s} = \frac{\lambda_s}{s\mu_s}, \qquad w_s = w_q + \frac{1}{\mu_s} \tag{21}$$

where w_q is the queuing time for the end-users in the queue to be served. w_s is the processing time in the checkout system. "s" is the number of VMs required to reduce the queuing time for the end-users. "k" is a variable of VMs.

Using the same assumption of μ_1 and λ_1 in the M/M/1 model, we should have the following calculation results for M/M/S model in Table 4.

Note that we adopt an analytic approach for M/M/S queueing network based on some simple assumptions, such as the Poisson distribution of arrived rate and the exponential of distribution service time. If some of our assumptions are not held, and the queueing network system becomes complex, we should adopt a simulation method to analyze the queueing network behavior. Bose [49] highlighted some advantages and disadvantages of a simulation method for queueing network.

Now, if we plot out the result of queueing time against the incremental number of VMs shown in Fig. 10, we can have an approximate trend line in a power function. According to both Table 4 and Fig. 10, we see that queueing time decreases sharply after the 2nd VM or 3rd VM. Therefore, we can use Eq. (22) to approximate a utility function for market segment 3.

$$U_3(q) = K_3 q^{-c}, \qquad 1 < q < q_m \tag{22}$$

where K_3 is a scaling coefficient. "c" is a constant that is to determine the gradient of the power equation. q_m is the maximum quantity of VM that the customers of segment 3 may purchase.

On the other hand, if the CSM wants to reduce the overall processing time (both queueing and processing time), the cloud business customer can have a solution of combining both vertical and horizontal scaling.

If the λ_s value is relatively small in comparison with μ_s, the power function is sufficient to model the customer utility value. If the λ_s the value becomes larger, then adopting a discrete function (e.g., Eq. (14)) is a good idea to describe the cloud customer's utility value because a guaranty to deliver SLA becomes a significant issue when the average number of end-users increases.

Alternatively, we can also use a linear function [33] to provide a solution when there is a constant rate of changing in terms of VM demand and utility value. This idea leads to the next issue of how to model customer utility for segment 1. The workload characteristics of this segment have been classified as "Virtual Desktop Infrastructure (VDI)". There are many VDI performance metrics of a hosting environment regarding users' experiences, such as the peak of Input/Output Per Second (IOPS), storage capacity, response time, Read/Write ratio, future growth, etc. If we assume these metrics have been prefixed during the Proof of Concept (PoC) stage before VDI rollout, the additional VM may add Opex to the cloud business, although it delivers a certain amount of utility value. In other words, the marginal utility has a negative value. So, we can use a linear function (Eq. (23)) to represent the cloud customer's utility value because an end user's response time is calculated as a linear model based on the cloud resource request [50,51].

$$U_1(q) = K_1(rq + q_m), \qquad r < 0 \tag{23}$$

where r is a constant, but the value is negative to reflect the economic principle of the diminishing return. It means that every additional VM has less utility value than the existing VM. The utility function is linear because the VDI hosting environment is mainly driven by storage resources that have an additive impact on customers' utility values.

3.6. Utility functions for backend and content delivery workloads

When we encounter backend and dynamic data processing types of workload, such as dynamic content (optimized dynamic content) delivery, clone server, and Network File Sharing (NFS), we can use different mathematical models to measure the cloud customer's utility values in term of the end-users' experiences. According to [52,53], we can use isoelastic utility function (Eq. (24)) to model the customers' utility value for the dynamic content workload for market segment 6 because it is network-oriented service delivery. If α is greater than zero, it means the constant relative risk aversion (CRRA) when the cloud customer is facing some uncertainties of cloud resources q.

$$U_2(q) = K_2 \begin{cases} \dfrac{q^{1-\alpha}}{1-\alpha}, & \alpha \neq 1 \\ \ln(q), & \alpha = 1 \end{cases} \tag{24}$$

where "α" is to measure the degree of relative risk aversion. Based on the Pratt–Arrow absolute risk aversion function (Eq. (25) R_r), we can measure the absolute value of risk aversion, which is to define the coefficient value at "q". R_r is a negative exponential (or inverse) function at "q" when α is greater than zero.

$$R_r = -\frac{U_2''(q)}{U_2'(q)} = \frac{dU_2'(q)}{dq}\,\frac{q}{U_2'(q)} = \frac{\alpha q^{-\alpha-1}}{q^{-\alpha}} = \frac{\alpha}{q} \tag{25}$$

Practically, it means that if R_r is decreasing with respect to VM quantity "q", the cloud customer will be less sensitive towards risk aversion when the number of VMs is increasing.

We can also use the exponential utility function to model the backend type of workload for market segment 2. The exponential function gives us the value of constant absolute risk aversion (CARA) (Refer to Eq. (26)):

$$\begin{aligned} U_6(q) &= K_6 \begin{cases} \dfrac{\left(1-e^{-\alpha q}\right)}{\alpha}, & \alpha \neq 0 \\ q, & \alpha = 0 \end{cases} \\ R_a &= -\frac{U_6''(q)}{U_6'(q)} = \alpha \end{aligned} \tag{26}$$

where α represents the constant absolute risk aversion [54]. When $\alpha = 0$, it means risk neutral, and when $\alpha < 0$, it is risk-seeking. In this paper, we set the value of $\alpha < 0$ because MapReduce or log file analysis type of workload is interruptible in terms of operational cost saving.

The MapReduce workload may require a large amount of VM resources and the processing environment is complicated because it involves different issues of cloud architecture, planning, and resources scaling, e.g., database replication (1:1 replication of both master and slave for zero-downtime), read replica, in-memory caches (Key-Value Store for the session and state data, across cloned instances), etc. As a result, we can set the α value, either less than zero, to estimate the customers' utility values. In other words, we use the exponential function with $\alpha < 0$ to describe a customer's utility values in terms of acquiring VM resources.

3.7. Defining coefficient values

The final issue is how to determine the value of K_i and α. The scaling coefficient of K_i is dependent on the business revenue or profit that a particular type of VM instance (such as AWS's extra-large instance) can help cloud customers to generate their business revenue or profit. For example, if we target the average profit of SME is around $41K-$90K/per annum [55], we can approximately estimate the profit for each VM to generate is between $0.95 and $1.9/per hour [56] for various cloud applications across six market segments.

It is challenging to determine the value of risk aversion α because it measures cloud customers' subjective feelings when they are facing uncertain outcomes [57]. Based on [57] and [58] recommendations, we set the value of risk aversion is equal to 0.3 in this paper.

Once the targeted SME customers are specified, we can normalize the average utility value up to $1.50 (per/hour), and set the minimum utility value is equal to $0.00 across all six market segments. The maximum number of VM is an arbitrary number. Here, we set to $q_m = 12$. It is reasonable to assume a typical CRM architecture needs about 11 VMs (Refer to Fig. 11). It is just a matter of scale. Fig. 11 shows the approximate quantity of VMs may require run on a single cloud platform. This quantity of q_m could be various from one case to another. In other words, if the solution architecture is changed, the value of q_m will also be changed. For example, if the business requires running different types of database, the solution architecture should be altered. However, the standard architecture of web hosting should work for the majority of SME customers.

Fig. 11 also shows an example of an architecture solution for cloud resource scaling, which can be either horizontal or vertical. The decision of cloud resources, whether it should be vertical or horizontal scaling, depends on the definition of customers' business requirements, such as CSM or KPI.

Once all the above assumptions are put in place, the numerical values of six utility functions can be created in Table 5. It provides a solution to the problem that has been raised in Section 3.3. Table 5 is a part of a comprehensive framework of cloud price strategy for a CSP to achieve the maximized profits by capturing the full spectrum market share.

3.8. Summary of modeling multiple utility method

Overall, we have defined all six utility functions based on the market segmentation assumptions. Table 6 covers multiple utility functions with different cloud customers' preferences for various business applications. The pre-condition of the utility function definition is dependent on the result of cloud market segments. The number of market segments is derived from a CSP's cloud business strategy and targeted customers. We classify these market segments into three categories.

Table 5
Cloud customers' utility functions values.

VM No. q	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6
1	$1.50	$0.01	$1.50	$0.75	$1.50	$0.29
2	$1.36	$0.02	$0.75	$0.75	$1.50	$0.45
3	$1.23	$0.03	$0.50	$0.75	$1.50	$0.60
4	$1.09	$0.05	$0.38	$0.75	$1.50	$0.72
5	$0.95	$0.08	$0.30	$0.75	$1.50	$0.84
6	$0.82	$0.13	$0.25	$0.75	$1.50	$0.95
7	$0.68	$0.19	$0.21	$0.75	$0.00	$1.05
8	$0.55	$0.29	$0.19	$0.75	$0.00	$1.14
9	$0.41	$0.44	$0.17	$0.75	$0.00	$1.24
10	$0.27	$0.67	$0.15	$0.75	$0.00	$1.33
11	$0.14	$1.00	$0.14	$0.75	$0.00	$1.42
12	$0.00	$1.50	$0.13	$0.75	$0.00	$1.50

Table 6
Cloud customers' six utility functions.

Business application workload	Market segment	Analytic approach	Addressable market demand	Cloud customers utility function
High Availability (HA)	5	Markov Chain analysis	81	$U_5(q) = \begin{cases} K_5, & 1 \le q \le k \\ 0, & k < q \le q_m \end{cases}$
Disaster Recovery (DR)	4		235	$U_4(q) = \begin{cases} \theta K_5 & 1 \le q \le \mathrm{k} \\ 0 & k \le q \le q_{\mathrm{m}} \end{cases}$
Hosting	3	Queueing theory	90	$U_3(q) = K_3 q^{-c}$
VDI	1		269	$U_1(q) = K_1(q_m + rq), \quad r < 0$
Content delivery, terminal servers	6	Risk assessment	13	$U_2(q) = K_6 \begin{cases} \frac{q^{1-\alpha}}{1-\alpha}, & \alpha \neq 1 \\ \ln(q), & \alpha = 1 \end{cases}$
Big data	2		205	$U_2(q) = K_2 \begin{cases} \frac{\left(1 - e^{-\alpha q}\right)}{\alpha}, & \alpha \neq 0, \alpha < 0 \\ q, & \alpha = 0 \end{cases}$

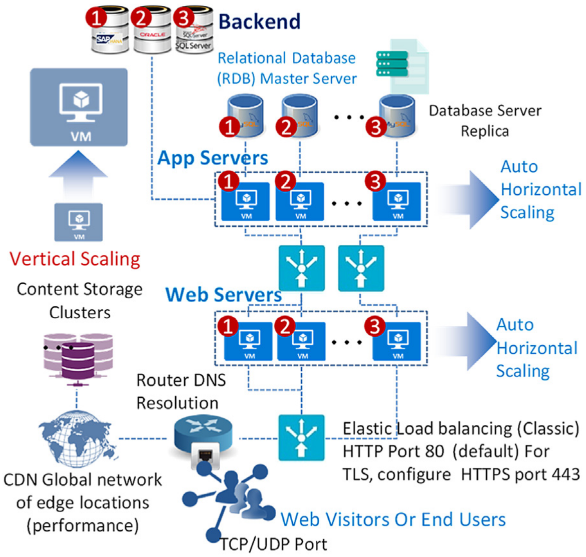

Fig. 11. Typical architecture of web application hosting.

In the first category, we model the HA types of cloud workload for market segments 5 and 4, respectively. We show how to use Markov chain analysis to decide the minimum number of VMs in order to meet the specified SLA. There are few critical assumptions of modeling for this category of utility functions. First, if the specified SLA is changed, the number of VMs will be altered. Intuitively, the number of provisioned VMs will be either increased or reduced. Second, the difference between segments 5 and 4 is how to estimate a customer surplus (profit contribution) in comparison to the offering price with the decision criteria. If the quantity of VMs is more than a threshold level of customers' business requirements, the utility value of segment 5 will be diminished to zero. For segment 4 (DR-driven HA workloads), the customers' utility function also shows as a constant value for each VM, but the utility value is justified by CSP's offering price in comparison with its on-premises infrastructure costs if CSP's offering price is higher than on-premises cost, the cloud customer will switch to an option of building its own cloud infrastructure.

The second category of utility functions focuses on response time. Segment 3 is based on the queueing theory, which can also be derived from the Markov chain analysis for an e-Commerce business, such as an online checkout system. Segment 1 is to model virtual desktop infrastructure (VDI) mainly. The linear utility model can represent the customer utility value, which shows an additive relationship for each incremental VM. In other words, if the business application has a workload pattern that is similar to Online Transaction Processing (OLTP), we can approximately use the power function to model the utility value because the value of additional VM is declining sharply for an additional cloud resource. If the workload is storage-related applications, we can use a linear function for each additional VM because the additional utility will decrease linearly. These two functions are closely related.

The third category of functions is dependent on the economic idea of risk assessment towards computational resources. If the cloud workload of the segment 6 is a type of dynamic data processing, such as clone server, Network File Sharing (NFS), State sharing, URL rewriting, and dynamic content delivery (such as online booking), the customer's utility can be described as an isoelastic or power function. Furthermore, if the workload would allow a relative risk aversion when the number of VM is increasing, CRRA is the adequate model for the utility value. In contrast, if the application workload is a backend type of data processing that allows a certain degree of risk for the computational interruption, the risk-seeking utility function can be applied. In other words, a customer may be willing to take the risk of workload interruption

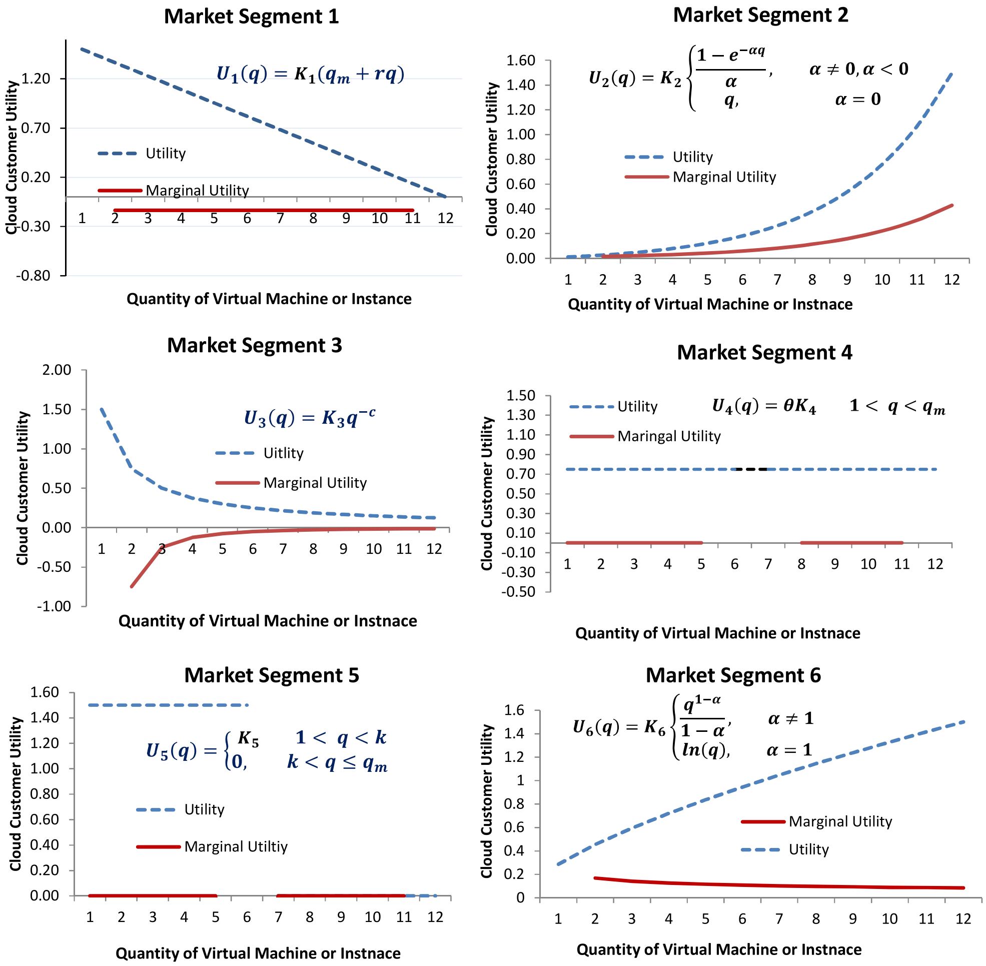

Fig. 12. Six cloud utility functions for six cloud market segments.

rather than pay a high cost for more VM resources. Big Data Analytics is one of the applications because many MapReduce workloads can be interrupted.

A CSP can have more or less than 6 market segments. According to [29], the suggested number of the market segments is between 5 and 10. Overall, the number is dependent on a market portfolio analysis to meet the CSP's business objectives by balancing sales growth, capital investment budget, cash flow, cloud technology expertise and business strategy. For example, if the CSP would like to explore other niche market (e.g., cage-level physical security), a customer's utility function will be defined differently. Ultimately, a CSP should focus on utility functions that are capable of generating value co-creation with its business customers. It means how to estimate the value of K_i coefficients and how to balance the coefficients across all market segments.

When we estimate K_i coefficients, we balance the values of all the scaling coefficients to be equivalent by grouping SMEs that have similar revenue amounts together. If a gap of the coefficient value is too large, then the higher value of the coefficients would have more influence on the optimal price of a VM. To visualize all utility functions of Table 5, we plot out all six cloud customer utility functions along with the number of VM variation in Fig. 12.

We have now demonstrated that our method of defining cloud customer utility functions that depend on multiple criteria decision making. These decision criteria consist of both internal rationality (strategic objectives, cost, expertise, cash flow, targeted customers, etc.) and external rationality (CSM, KPI, cloud customers' revenue or profits, and market segments) for a CSP to achieve the maximum profit by identifying the optimal price. Our key idea of modeling the cloud customers' utility functions is to assign SME customer's revenue to each VM that can help cloud customers to generate business revenue, which is the concept of value co-creation [59,60] across segmented cloud market. To compare with other solutions of modeling, we can evaluate the performance of different models in terms of market share and possible profit margin.

4. Performance evaluation

The performance evaluation is divided into two parts. The first part is to compare the market share between our solution of six market segments and other solutions with the single market assumption. The second part is to compare all economic values, which include business revenue, profit, the optimal price, and a unit cost based on the popular price model, namely "on-demand".

4.1. Comparison of cloud market share

In comparison with some previous modeling methods of the utility functions (Refer to Table 7 for details comparison), our modeling method has the following advantages: First, the unit of all six utility functions is measured by the customer's revenue or profit contribution in terms of the dollar. Second, this unit is tangible and can be compared across all market segments. Third, each utility function is associated with one type of cloud business application or market segment. Fourth, we have identified a total of six market segments. It avoids "one size fits all". Fifth, we focus on the cloud customers or demand side's utility functions. Sixth,

Table 7
Characteristics of different methods of utility modeling and addressable market share.

Methods of utility modeling	Utility functions	Potential cloud market share	Measurement of the utility unit	Ind. variables of the utility function
Multi-utilities method	$U_i\,(q)\,,\quad i = 1 \cdots S\ ,\quad q = 1 \cdots q_m$	**100%**	Customer's revenue & profit	VMs
Model-based	$U(H) = \frac{1-M^{\alpha}}{1-T^{\alpha}}$	Less 16%∼17%	Hit rate	Memory "M" & workload (object "T")
SLA metrics	$\widehat{U}\,(R) = \max_c U[S\left(C, R,\ D'\right),\ D']$	Less 16%∼17%	Service level	control "C" Resource "R", Demand "D"
Resource-based	$U_i = \alpha_i S_i$	Less 16%∼17%	Performance metrics (e.g., response time)	A throughput of response time $R_{i,s}$, specified time $\beta_{i,s}$
Social surplus-based	$U_i\,(z_i) = V_i\,(z_i) - Pz_iT_i\,(z_i)$	Less 16%∼17%	Expected resource within time and price limit	Price P, resource z_i Execution time T_i Expected value V_i
Empirically calibrate	$U_{ijk} = w\left(v_i - \frac{c_i+2^{k-1}p_{j1}}{2^{k-1}\alpha_j^{k-1}q_{j1}}\right)$	Less 16%∼17%	Expected value v_i minus combination of three variables	Price p_{j1}, delay time sensitivity c_i, quality level q_{j1}, workload w
Price-quality	$U\,(pr,\ s) = P_i\,(pr, s) = \lambda_i\,(pr_1 - c_i - \rho_i) - \frac{\rho_i}{\overline{rt} - s_i}$	Less 16%∼17%	Payoff (resource request capacity) vs. Price	CSP price pr, response time s, unit Opex c_i and unit Capex ρ_i
Capacity aware	$U\,(P_{N+1}, T_{N+1}) = \sum_{i=1}^{N+1} U_i^Q - \sum_{i=1}^{N} U_i^{R=\{SLA\}}$	Less 16%∼17%	CSP's profit $U\,(P_{N+1}, T_{N+1})$	Service price P_{N+1} and response time T_{N+1}
Conjoint analysis	$U\,(R_i, p_i\ ) = \sum_{i=1}^{n} R_i p_i$	Less 16%∼17%	Preference ranking	Attribute ranking R_i & weight p_i
Framework based	$U\,(f) = -e^{-5e^{-0.5*f(A,E,R)}} + 1$	Less 16%∼17%	Sigmoid function value	Specified scenario parameters A, E, R
Simple linear	$U\,(p, t) = U_0 - \alpha p - \beta t$	Less 16%∼17%	Service request satisfaction level	VM price "p" & response time "t"

In contrast to previous SLA research works for cloud contents, we explicitly specified the number of VMs to be provisioned for a guaranty of cloud customer's revenue delivery. Seventh, we lay out a clear definition of utility in upfront to avoid any possible misinterpretation of utility.

Table 7 of the market share calculation is dependent on the assumptions of the number of market segments. If the model assumes the market is a unified or single market, the pricing model can only address small proportional customers. For example, if a CSP only offers only one price for all its cloud customers, such as an auction-based spot instance price, the majority of business customers will be left out because spot instance cannot guarantee to deliver some mission-critical business applications. Therefore, the single price model can only target 16% ∼17% (1/6) market shares if the cloud market has actually six market segments. This is self-explanatory.

The ultimate goal of market segmentation is to set up a pricing foundation for CSP to achieve its maximum profit. Therefore, it is vital to validate our solution through the experiment based on a particular price model, which can be demonstrated in the second part of the performance evaluation.

4.2. Economic values comparison

The details of pricing comparison can be found in our other work [61], which has been highlighted in Fig. 1 for steps 3 and 4. In this work, we only give brief information on how we implemented our experiment through a particular price model, and then we show the experimental results with different solutions. Table 8 provides the justification for our solution.

4.2.1. Process of evaluation

To implement the 2nd part of the evaluation, we adopt the "on-demand" price model for cloud pricing, in which every leading CSP offers this price model to its cloud customers to reflect one of the cloud characteristics: pay as you go (PAYG). This "on-demand" price model can be defined as Eq. (27)

$$CS_i\,[p] = \left(\left(\sum_{j=1}^{q_m} U_i\,[j]\right) - pq\right) \geq 0 \qquad (27)$$

$$q_i\,[p] = arg\,\max_q CS_i\,[p]\ \ i = 1, \ldots, S$$

where S is the number of market segments. The q_i is a number of VMs to be provisioned by the cloud customers in the market segment "i". The VM quantity is decided by a maximum customer's surplus-value $CS_i\,[p]$ that is greater than zero for the given price p which is offered by a CSP. It also depends on the defined utility function $U_i\,[j]$ which represents the external rationality (Refer to both Table 6.) for the "i" market segment while j is a variable of VM between i and q. q_i is a dependent variable of a price p. It means if the cloud price is changed, the sales quantity of each market segment will also be changed.

If the cloud customer's surplus $CS_i\,[p]$ has been quantified, the maximum profit $\pi\,[p]$ of a CSP can also be achieved by identifying the optimal price p^*. Based on microeconomics, the profit equation can be easily defined as Eq. (28). Further details can be found in [61]

$$p^* = arg\,\max_p \pi\,[p] \qquad (28)$$

4.2.2. Dataset

To optimize Eq. (28), we adopt the genetic algorithm to run our experiment. There are a number of software applications that can be applied to implement the genetic algorithm, such as Matlab, R and even Microsoft Excel Solver. The R package has two convenient packages: GA and Genalg that can deliver quick results.

4.2.3. Experiment results

If a CSP assumes the cloud market is a single market with only one defined utility function (e.g., either resource-based or

Table 8
Experiment results of comparison between resource-based single market and multi-utilities based six market segment.

Comparison for “on-demand” price model	Resource-based With single market $U_i = \alpha_i S_i, i = time$	Simple linear with single market $U(p,t) = U_0 - \alpha p - \beta t$	Multi-utilities with 6 market segments $U_i(q), i = 1 \cdots 6$	Multiple utilities compare with resource-based	Multiple utilities compare with simple linear
Optimal price	$0.750	0.955	**$0.7499**	−0.013%	−27.35%
Unit cost	$0.3455	0.3761	**$0.2814**	−22.78%	−33.65%
Total sales Vol.	2581	2077	**5256**	50.89%	60.48%
Total revenue	$1936	$1983	**$3942**	50.89%	49.70%
Total cost	$892	$781	**$1479**	39.69%	47.19%
Total profit	$1044	$1202	**$2463**	57.61%	51.20%

simple linear utility function), they can only achieve either $1044 or $1202 profit respectively (Refer to Table 8). In comparison with six market segments with multiple utility functions, the profit margin can reach $2463. In other words, our method of defining utility function can achieve more than 57.6% profit than the resource-based method and 51.2% more profit than a simple linear while the unit cost drops over 22% (resource-based) and 33% (simple linear) respectively.

This is because not all customers' utility functions are continuous. Some utility functions are discrete. If the evolution of different charging prices is plotted out (as shown in Fig. 13) for the on-demand price model, we can see there is a sharp drop in revenue and profit while the unit cost increases dramatically beyond the optimal price for the multiple-utility functions. The principle is similar to many retailers that adopt a psychological price, such as $0.99 instead of $1 to boost sales volume or to increase their revenue.

4.2.4. Evaluation of the experiment results

Notice that the above experiment result is dependent on some key assumptions (See Section 3.2). If these critical assumptions are changed, the experiment results could be different. As the paper [6] indicated, a CSP's pricing strategy is driven by the overall CSP's business strategy, long and short term goals of a firm (e.g., growing market size or growing profit margin), an investment budget, technology expertise, return on investment, and etc. This leads to a decision of the number of market segments and the targeted market or cloud customers.

The above experiment results primarily demonstrate that in comparison with a single market segment assumption, a CSP can achieve a higher profit margin if it can divide the cloud market with multiple segments and target different cloud business applications. If a CSP assumes there are only a few cloud market segments, the profit margin could be lower.

Once we understand the principle of identifying multiple utility functions, we can put this principle into a cloud business practice. Here are some simple guidelines to apply to this state of the art.

5. Discussion of model selection (simple guidelines)

Based on various parameters of six cloud market segments, as shown in Table 3, the type of business application can be estimated, which is mapping to each corresponding cloud market segment (See Fig. 3). If an analyst has the real cloud operational dataset, this step will become much more manageable. The crucial issue is how to define the utility function for different customers' business applications. The basic guidelines can be summarized as follows:

1. If the business customers host a web site or run e-commerce applications, such as online checkout, one of the significant value propositions for a customer to purchase more VMs is to reduce the queuing time and create the good customers' experience of online shopping. The process of reducing queueing time has been demonstrated. The useful model to describe the cloud customers' utility value is the power function for SME. However, the parameter of the exponent has to be negative to reflect the diminishing of return for the marginal utility. Fig. 10 illustrated the value proposition for an e-commerce type of business application
2. When the exponent component of a power function is equal to one, the power function becomes linear. To reflect the diminishing of return for the marginal utility, the coefficient value of the linear variable is negative. The primary driver behind adopting a linear function for the VDI application is to increase the storage performance while the customers can reduce operational costs. According to [62, 63], there are 19 performance metrics, such as “Copy Read Hits”, “Disk Time”, “Pool Paged Bytes”, “Network Interface Bandwidth”, etc. Different performance metrics might change the utility function parameters. It is dependent on CSP's targeted customers. The best practice is to set up an initial model and then have a fine-tuning with a simulation model based on the real operation dataset.
3. If the exponent of the power function is set to zero, the function becomes a constant within a certain quantity of VM. This function can present a cluster of VMs to support the specified SLA (e.g., 5 nines) for mission-critical business applications, such as CRM database backup. The utility function becomes a discrete function because the utility will diminish to zero after a threshold level of VM quantity.
4. In comparison with the database backup, the DR application requires more VMs to mirror the entire operational environment. From a utility function perspective, it means the number of VMs is more than the backup application. However, the coefficient “θ” of the function is less than one to reflect the possibility of a disaster event that may occur.
5. Regarding a risk assessment of a customer's operational cost (e.g., CSP's offering price for cloud resources) and a possibility of workload interruption (e.g., performance), we can use the isoelastic (power) utility function to model the business customers' decision in term of acquiring the number of VMs [64].
6. In contrast, if the customers prefer to take more risks for multiple interruptions of their computational process (such as MapReduce application) rather than pay a high price of cloud resources, the exponential utility function can be applied and the value of α is less zero. Usually, the type of cloud application often requires massive computing power, and job priority is quite lower and workload can be interrupted.

6. Assumptions analysis for defining utility functions

6.1. Assumptions analysis

Throughout this paper, we adopt an analytic approach to define multiple utility functions based on various assumptions of

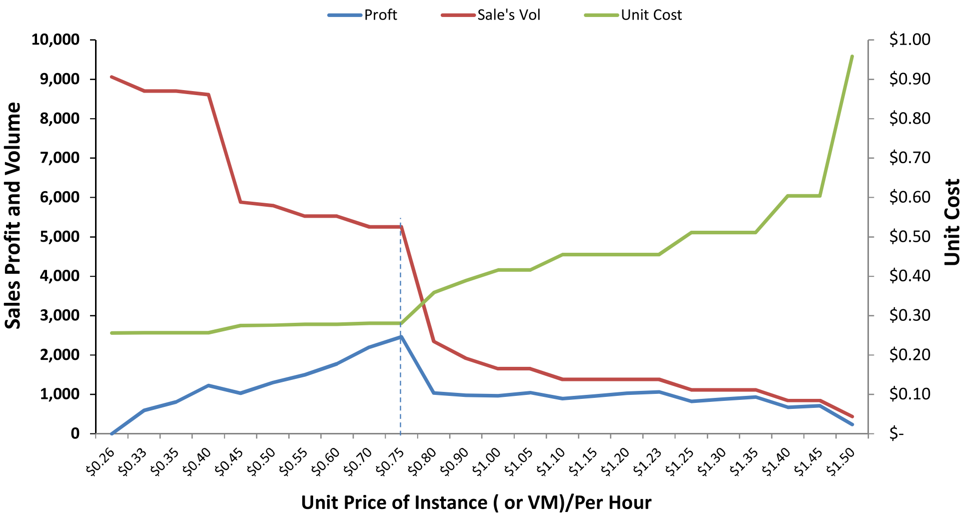

Fig. 13. Price evolution of "On-Demand" price model for six market segments.

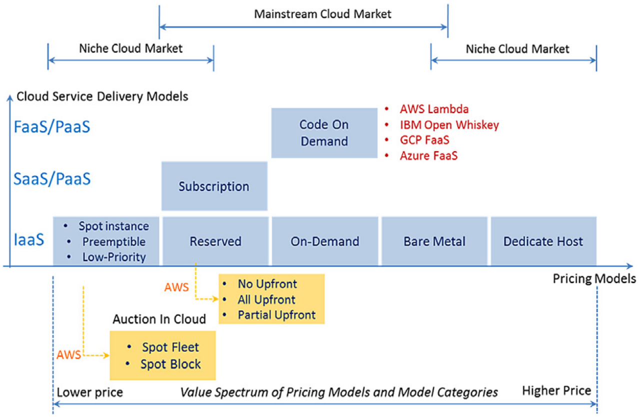

Fig. 14. Summary of cloud pricing spectrum in the current cloud industry.

business strategy, technology expertise, investment capital, and segmented cloud market so that CSP can explore a broader addressable cloud market to define business customers' utility functions. If some assumptions are not held or become uncertainty, we can combine analytic, simulation, and statistical approaches. The pre-condition of simulation and statistical method is that an operational dataset should become accessible.

In comparison with the analytic approach, statistic or simulation modeling methods can further consolidate many assumptions. The advantages of combining various methods would provide a much balance view of cloud customers' preference in terms of marginal VM demand. Therefore, a combination of approach enables a CSP to know more about how much the customers are willing to pay for what type of cloud resource (e.g., VM instance).

The idea of the defining cloud utility function based on the segmented market is to measure cloud business customers' preferences and tastes in terms of less or more VM resources to be purchased. In this study, the unit of subjective metrics can be interpreted as the cloud customer's revenue or profit contribution. Practically, many factors may impact the business customers' revenue and profits, such as end-user' experiences, response time, latency, throughput, availability, market environment, etc. Various CSM measurements may result in different shapes of utility functions. However, the above six utility functions cover some basic cloud business applications.

6.2. *Current practice of cloud pricing*

Based on the current cloud business practice, we can see that some leading global CSPs, such as AWS, MS Azure, Google Cloud Platform (GCP), IBM Cloud, Right Scale, Oracle cloud, DELL/EMC/VMware, Salesforce.com, and Dropbox, divide cloud market into five to seven market segments and offer different cloud services with different pricing models [61]. Fig. 14 illustrates the overall cloud market spectrum. Most CSPs provide essential cloud services (e.g., on-demand and reserved instances) to cover the mainstream cloud market segments. Only a few leading CSPs attack niche market segments. For example, only AWS provides an auction-based spot instance in the cloud market.

Fig. 14 demonstrates that different CSPs have different pricing strategies based on their company goal, targeted market segments, and technology expertise, investment budget and etc. If a CSP's pricing strategy and the number of market segments is determined, the utility function can be defined. The main aim of this paper is to demystify the process of how to define utility functions for cloud business customers so that any CSP can adopt this modeling process to develop its own cloud pricing strategy in detail.

7. Conclusions and future work

The issue of how to define the cloud customers' utility functions from the cloud customer's perspective is vital to any CSP because it would help the CSP to generate the optimal cloud price to maximize the profits for its cloud business. Based on our intensive literature review on this topic, we show that one way to improve CSP's profit is to determine the cloud market segments and then define multiple utility functions from a value co-creation perspective.

Our solution provides external rationality, which is closely tied to the business applications that can help cloud business customers to generate revenue or profit. In comparison with previous modeling methods, our solution is based on both market segments and value co-creation. It is tangible and direct for many cloud practitioners because all utility values are measured by the cloud business customer's profit for its provisioned cloud resources.

Overall, the modeling process is just one of four processing steps for CSPs to create a value-based pricing strategy. There are two more steps, which they are to build various value-based pricing models (step 3) and identify the optimal price point for each price model so that both CSPs and cloud business customers can achieve the goal of value co-creation (step 4) in [7]. In future work, we will consolidate the step 1 (cloud market segmentation) and 2 (defining multiple utility functions) when we can access some live datasets from CSPs so that we can improve this state of the art for many cloud practitioners to generate some practical value-based pricing models. Moreover, we will develop other types of utility functions to cover more niche market segments, such as Function as a Service (FaaS) market.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P.R. Krugman, R. Wells, Economics Fourth Edition, Worth Publishers, New York, 2015, pp. 282–283.

[2] S. Landsburg, Price Theory And Applications, West Pub. Co, Minneapolis/St. Paul, 1995.

[3] F.S. Roberts, Encyclopedia of mathematics and its applications. Volume 7. Measurement theory with applications to decision-making, utility and the social sciences Fred S. Roberts, Current Sci. (13) (2009) 6–8.

[4] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT Platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Gener. Comput. Syst. 25 (2009) 599–616.

[5] G.A. Jehle, P.J, Reny Advanced Microeconomic Theory, Pearson Education, England, 2011, p. 13.

[6] C. Wu, R. Buyya, K. Ramamohanarao, Cloud computing market segmentation, in: 13th International Conference On Software Technologies, Proceedings of the 13th International Conference on Software Technologies, ICSOFT, 2018, pp. 888–897.

[7] C. Wu, R. Buyya, K. Ramamohanarao, Value-based cloud price modeling for segment business to business market, Future Gener. Comput. Syst. (2019).

[8] T.T. Nagle, J. Hogan, The strategy and tactics of pricing, in: A Guide to Growing More Profitably, Pearson/Prentice Hall, Upper Saddle River, N.J., 2006.

[9] R.P. Doyle, J.S. Chase, O.M. Asad, W. Jin, A. Vahdat, Model-based resource provisioning in a web service utility, in: Processing of the Usenix Symposium on Internet Technologies and Systems, 2003, p. 57.

[10] K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D.P. Pazel, J. Pershing, B. Rochwerger, Oceano-SLA based management of a computing utility, in: 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium, 2001, p. 855.

[11] W.E. Walsh, G. Tesauro, J.O. Kephart, R. Das, Utility functions in autonomic systems, in: 2004, International Conference on Autonomic Computing, Proceedings., Autonomic Computing, 2004, p. 70.

[12] M.N. Bennani, D.A. Menasce, Resource allocation for autonomic data centers using analytic performance models, in: 2005, Second International Conference on Autonomic Computing, 2005, p. 229.

[13] J.O. Kephart, R. Das, Achieving self-management via utility functions, IEEE Internet Comput. 2 (2007) 40–48.

[14] D. Burda, F. Teuteberg, Exploring consumer preferences in cloud archiving – a student's perspective, Behav. Inform. Technol. 35 (2) (2016) 89–105.

[15] D. Minarolli, B. Freisleben, Utility-based resource allocation for virtual machines in cloud computing, in: 2011 IEEE Symposium on Computers and Communications (ISCC), 2011, p. 410.

[16] M. Cardosa, M.R. Korupolu, A. Singh, Shares and utilities based power consolidation in virtualized server environments, in: 2009 IFIP/IEEE International Symposium on Integrated Network Management, 2009, pp. 327–334.

[17] NIST Cloud Computing Service Metrics Description, NIST, https://www.nist.gov/sites/default/files/documents/itl/cloud/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf.

[18] Viewing Instance Metrics, Using Oracle Cloud Infrastructure Compute Classic https://docs.oracle.com/en/cloud/iaas/compute-iaas-cloud/stcsg/viewing-instance-metrics.html.

[19] C. Kilcioglu, M.R. Justin, Competition on price and quality in cloud computing, in: Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2016, pp. 1123-1132.

[20] R. Pal, P. Hui, Economic models for cloud service markets: Pricing and capacity planning, in: Theoretical Computer Science, (Distributed Computing and Networking (ICDCN 2012), 2012, pp. 113–124.

[21] N. Ranaldo, E. Zimeo, Capacity-aware utility function for SLA negotiation of cloud services, in: 2013 IEEE/ACM 6Th International Conference On Utility And Cloud Computing, Utility And Cloud Computing (UCC), 2013, p. 292.

[22] J. Chen, C. Wang, B.B. Zhou, L. Sun, Y.C. Lee, A.Y. Zomaya, Tradeoffs between profit and customer satisfaction for service provisioning in the cloud, in: Proceedings of the 20th International Symposium: High-Performance Distributed Computing, 2011, p. 229.

[23] G. Baltas, P. Doyle, Random utility models in marketing research: a survey, J. Bus. Res. (2001) 115.

[24] E. Weintraub, Y. Cohen, Optimizing user's utility from cloud computing services in a networked environment, Int. J. Adv. Comput. Sci. Appl. 6 (10) (2015) 153–163.

[25] S.K. Garg, A.N. Toosi, S.K. Gopalaiyengar, R. Buyya, SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter, J. Netw. Comput. Appl. 45 (2014) 108–120.

[26] J. Yu, R. Buyya, K. Ramamohanarao, Workflow scheduling algorithms for grid computing, in: Metaheuristics for Scheduling in Distributed Computing Environments, 2008, p. 173.

[27] M. Koehler, S. Benkner, Design of an adaptive framework for utility-based optimization of scientific applications in the cloud, in: 2012 IEEE Fifth International Conference on Utility and Cloud Computing, Utility and Cloud Computing (UCC), 2012, p. 303.

[28] J. Claycamp, M. William, A theory of market segmentation, J. Mark. Res. 5 (4) (1968) 388–394.

[29] M. McDonald, I. Dunbar, Market Segmentation, How to Do it and How to Profit from it, John Wiley & Sons, 2012.

[30] C.C. Reiss, J. Wilkes, J.L. Hellerstein, Google Cluster-Usage Traces Format+ Schema, White Paper, Google Inc., 2011, pp. 1–14.

[31] https://www.amd.com/Documents/AMD_WP_Virtualizing_Server_Workloads-PID.pdf Accessed in 20/Aug/2018.

[32] M. Young, Implementing cloud design patterns for aws, create highly efficient design patterns for scalability, redundancy, and high availability, in: The AWS Cloud, Packt Publishing, Birmingham, 2015.

[33] O. Michalski, S. Demiliani, Implementing Azure Cloud Design Patterns, Packt Publishing, Birmingham, 2018, pp. 109–119.

[34] G.D. Feitelson, Workload modeling for performance evaluation, in: IFIP International Symposium on Computer Performance Modeling, Measurement, and Evaluation, Springer, Berlin, Heidelberg, 2002.

[35] M.C. Calzarossa, M.L. Della Vedova, L. Massari, D. Petcu, M.I. Tabash, D. Tessera, Workloads in the clouds, in: Principles of Performance and Reliability Modeling and Evaluation, 2016, p. 525.

[36] https://github.com/google/cluster-data/blob/master/TraceVersion1.md.

[37] J. Schaffner, Multi-tenancy for cloud-based in-memory column databases workload management and data placement, 2014.

[38] Downtime cost calculator, Storagepipe http://downtimecost.com/.

[39] Rahul Mohta, Yogesh Kasat, J.J. Yadav, Implementing Microsoft Dynamics 365 for Finance and Operations, Packt Publishing, 2017, p. 84.

[40] https://docs.microsoft.com/en-us/dynamics365/unified-operations/retail/retail-components.

[41] W.J. Stewart, Probability, Markov Chains, Queues, and Simulation, the Mathematical Basis of Performance Modeling, Oxford Princeton University Press, Princeton, N.J., 2011, p. 193.

[42] Small business statistic report, 2016, Australia Statistic Bureau, https://www.asbfeo.gov.au/sites/default/files/Small_Business_Statistical_Report-Final.pdfAccessedin20/Aug/2018.

[43] J. Luetkehoelter, What is disaster recovery? in: Pro SQL Server Disaster Recovery, 2008, pp. 1–12.

[44] 2017 Top 10 European Cloud Providers, Cloud Spectator, https://cloudspectator.com/top-10-european-cloud-service-providers/.

[45] H.A. Schmid, G. Rossi, Modeling, and designing processes in E-commerce applications, IEEE Internet Comput. Internet Comput. (1) (2004) 19.

[46] V. Datla, K. Goseva-Popstojanova, Measurement-based performance analysis of e-commerce applications with web services components, in: IEEE International Conference on E-Business Engineering (ICEBE'05), e-Business Engineering, 2005, p. 305.

[47] U. Bhat, An Introduction to Queueing Theory: Modeling and Analysis in Applications, Birkhäuser, Boston, 2015, pp. 34–40.

[48] M. Khan, X. Xu, W. Dou, S. Yu, OSaaS, Online shopping as a service to escalate e-commerce in developing countries, in: 2016 IEEE 18th International Conference on High-Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems, 2016, p. 1402.

[49] S.K. Bose, An Introduction to Queueing Systems, Springer Science & Business Media, 2002, pp. 257–262.

[50] J.D. Strunk, E. Thereska, C. Faloutsos, G.R. Ganger, Using Utility to Provision Storage Systems, in: Proceedings of the Fast Conference on File and Storage Technologies, 2008, pp. 1-16.

[51] J.D. Strunk, Using Utility Functions to Control a Distributed Storage System, No. CMU-PDL-08-102, Carnegie-Mellon University Pittsburgh PA Department of Electrical and Computer Engineering, 2008.
[52] Carlee Joe-Wong, Soumya Sen, Mathematical frameworks for pricing in the cloud: Revenue, fairness, and resource allocations, 2012, CoRR abs/1212.0022.
[53] Hong Xu, Baochun Li, A study of pricing for cloud resources, ACM SIGMETRICS Perform. Eval. Rev. 40 (4) (2013) 3–12.
[54] M. Ikefuji, R.J. Laeven, J.R. Magnus, C. Muris, Pareto utility, Theory and Decision 75 (2013) 43–57.
[55] Australian Bureau of Statistics small enterprise and medium Enterprise, business growth and performance survey http://www.abs.gov.au/.
[56] J. Chen, et al., Tradeoffs Between Profit and Customer Satisfaction for Service Provisioning in the Cloud, in: Proceedings of the 20th International Symposium: High-Performance Distributed Computing, 2011, p. 229.
[57] P.J. Thomas, Measuring risk-aversion: The challenge, Measurement 79 (2016) 285–301.
[58] A. Kim, I.S. Moskowitz, Incentivized cloud computing, a principal-agent solution to the cloud computing dilemma, 2010, http://oai.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=ADA530441.
[59] J. Marcos-Cuevas, S. Nätti, T. Palo, J. Baumann, Value co-creation practices and capabilities: Sustained purposeful engagement across B2B systems, Ind. Mark. Manage. 56 (2016) 97–107.
[60] H. Alves, C. Fernandes, M. Raposo, Value co-creation, concept and contexts of application and study, J. Bus. Res. 69 (2016) 1626–1633.
[61] C. Wu, R. Buyya, K. Ramamohanarao, Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges, ACM Comput. Surv. 52 (6) (2019) 108.
[62] Citrix Virtual Apps and Desktops on AWS, Citrix, 2014, https://www.citrix.com.au/global-partners/amazon-web-services/xendesktop-on-aws.html.
[63] A. Paul, Citrix xenapp 7.5 desktop virtualization solutions, plan, design, optimize, and implement, in: XenApp Solution to Mobilize Your Business, Packt Publishing, Birmingham, 2014.
[64] http://buyya.com/papers/ValueCloudPriceModeling.pdf.

Caesar Wu is a senior IEEE member. He is one of the authors of Cloud Data Center and Cost Modeling. He was a senior domain specialist in Telstra. He managed and operated many of Telstra's enterprises and IT data centers. He has over 30 years working, researching, and academic experiences across various industries. He was the program chair of Computer Information System (CIS) Doctorial Colloquium of The University of Melbourne.

Dr. Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the CLOUDS Laboratory at the University of Melbourne. He has authored more than 600 publications. He is recognized as a "Web of Science Highly Cited Researcher" both in 2016 and 2017 by Thomson Reuters, a Fellow of IEEE, and Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud computing.

Dr. Kotagiri Ramamohanarao (Rao) is received the Ph.D. degree from Monash University. He was awarded the Alexander von Humboldt Fellowship in 1983. He was Research Director for the Cooperative Research Center for Intelligent Decision Systems, the program co-chair for VLDB, PAKDD, DASFAA, and DOOD conferences. He was awarded Distinguished Contribution Award in 2009 by the Computing Research and Education Association of Australasia. He is a Fellow of Australian Academy of Science, a Fellow of Australian Academy of Technology and Engineering and a Fellow of Institution of Engineers Australia.