QoS Based Service Selection and Provisioning in Cloud Computing

Mona Eisa

May 2019

School of Engineering, Computing and Mathematics

Oxford Brookes University

A thesis submitted to the Faculty of Technology, Design and Environment in partial fulfillment of the requirements of the award of Doctor of Philosophy

Abstract

Cloud computing has become a disruptive technology which has seen significant growth among consumers of various sizes. Consumers can now have access to seemingly unlimited computing resources over the Internet without making significant investment in computing infrastructure. Consequently, this trend has seen a rise in the number of cloud computing providers. Most of these providers offer various services to consumers, which are commonly classified into three main types of service provisioning models such as, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Cloud computing providers offer multiple services to consumers using one or more of models.

However, the large number of services offered by cloud providers introduces a new set of problems for the consumers. Consumers have to choose from a wide range of services and providers such that they meet consumers' requirements. The problem is further complicated as a large number of consumers do not necessarily have adequate knowledge of cloud services' concepts and terminologies. To add to the complexity, there is no standard benchmark for cloud services. Therefore cloud providers can package the same cloud services in different ways for consumers. Furthermore, there are no standard service level agreements defined for cloud services selection. Each cloud provider has different service level agreements which make consumer selection process more cumbersome. This research aims to bridge this gap by proposing and developing new methods and techniques that take into account different cloud services and providers as well as quality of services attributes that make it easier for consumers to rank and select cloud services which are tailored to their requirements.

This thesis makes various contributions to the current state of knowledge in the cloud service selection and provisioning area. It proposes a new model in order to systematically represent the quality of service (QoS) attributes of cloud services that cover both technical and non-technical aspects of cloud computing. The new model succinctly represents QoS attributes which cloud consumers can easily use and understand when selecting cloud services. The thesis also proposes a new framework for cloud service selection which improves and simplifies the cloud service selection process. It takes into account the level of user's knowledge of cloud computing technologies. The major benefit is to simplify the selection process for 'Beginner' cloud service consumers (who have little knowledge of cloud computing) by presenting the main QoS attributes to them with brief explanations. The other benefit is to give an Intermediate/Expert cloud service consumers an opportunity to go through more details of QoS sub-parameters. Unlike existing approaches, the framework developed in this thesis also ensures the credibility of the service selection by utilizing information from three different sources, including, information from cloud service providers' websites, online monitoring tools and users' reviews of cloud services. Furthermore, the framework integrates Service Level Agreement (SLA) which is an integral part of cloud services as it is important for the consumer to be able to view it as part of their decision making process. The framework is validated by developing a prototype tool using Python, MongoDB and Amazon AWS EC2 server. The tool is then evaluated using various real life scenarios to rank cloud service providers and also by comparing it against existing tools. The results show that the proposed tool outperforms existing tools using a set of criteria such as operability, mode of data selection and number of cloud providers among others for ranking and selecting cloud services.

Publications

Some parts of the work presented in this thesis have previously appeared in the following published papers:

Eisa, M., Younas, M., Basu, K. and Zhu, H., 2016, "Trends and Directions in Cloud Service Selection", Proceedings of the 10th International IEEE Symposium on Service-Oriented System Engineering (SOSE 2016), IEEE CPS, Exeter College, Oxford, UK March 29-April 1, 2016. pp. 423-432.

Eisa, M., Younas, M. and Basu, K., 2018, "Analysis and Representation of QoS Attributes in Cloud Service Selection", Proceedings of the 32nd International IEEE on Advanced Information Networking and Applications (IEEE AINA 2018), Pedagogical University of Cracow, Poland, May 16-18, 2018. pp. 960-967.

Keywords: Cloud Service Selection, Analytical Hierarchical Process, Simple Additive Weight, Multi Criteria Decision Making, Quality of Service, Software as a Service, Platform as a Service, Infrastructure as a Service, Service Level Agreement, Service Measurement Index .

Acknowledgments

Firstly, I would like to say a huge thank you to my supervisors Dr. Muhammad Younas and Dr. Kashinath Basu for the immense amount of guidance and support they have generously provided me. I would also like to thank them for the endless kindness and patience they have given me, particularly during times when I have needed it most and I am very grateful to have had such wonderful supervisors.

I would like to say a big thank you to my family for their emotional support and encouragement whenever I have felt overwhelmed during my PhD. My special thanks go to my parents Mr. Medhat Eisa and Mrs. Amira Marghlani, for their invaluable words of wisdom and reassurance which strengthened me throughout my studies. I will forever be grateful to them for all they have done for me in my formative years until this point.

Dr. Maria Teresa Gonzalez-Aparicio came to Oxford Brookes University from Spain to carry out her research and I had the pleasure of meeting her. During her stay she provided me with her encouragement and emotional support, which has been unforgettable, and I am thankful to have shared many joyous moments with her.

I would like to say a heartfelt thank you to both my children Obai and Sedra for their unbounded patience in living far away from me and for always uplifting me by saying they will be proud of me when I return home with my PhD.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

AHP	Analytic Hierarchy Process
API	Application Programming Interface
AWS	Amazon Web Services
CapEx	Capital Expenditure
CSMIC	Cloud Service Measurement Initiative Consortium
GUI	Graphic User Interface
IaaS	Infrastructure as a Service
ISO	International Standard Organisation
MCDM	Multiple Criteria Decision Making
MODM	Multiple Objective Decision Making
NoSQL	Not Only Standard Query Language
OpEx	Operating Expenditure
OS	Operating System
PaaS	Platform as a Service
QoS	Quality of Service
RPO	Return Point Objective
RTO	Return Time Objective
SaaS	Software as a Service
SAW	Simple Additive Weight
SLA	Service Level Agreement
SMI	Service Measurement Initiative
SOA	Software Oriented Architecture
SST	Sahab Selection Tool
VM	Virtual Machine

CHAPTER 1

Introduction

1.1 Background

In the constantly evolving world of technology cloud computing is one of the technologies that has seen significant growth in popularity and ubiquity in recent years. Cloud computing provides on-demand computing services such as compute power, storage, servers, databases, networking software applications and software development environment over the Internet (Cui *et al.*, 2017). A cloud service can be defined as a service which resides on a (remote) cloud infrastructure that is made available to consumers over the Internet using different provisioning models such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) and so on.

Cloud computing has revolutionized the way small, medium and large scale businesses provide and consume IT services as it provides various benefits such as reduced IT expenditure, on-demand services, large pool of resources, scalability and elasticity of services. The use and growth of cloud computing has exponentially increased (Columbus, 2017). With such a growing demand and popularity of cloud computing, more and more cloud service providers are entering into the cloud service business market and are offering various types of cloud services. Realizing the benefits of cloud computing, an increasing number of businesses and organizations want to outsource their in-house IT services to the cloud infrastructure.

There exist a large number of cloud service providers in the market such as Amazon, IBM, Alibaba, Google Cloud, Microsoft Azure and Oracle. On one hand, such a large number of cloud service providers in the market means that they offer more choices, better services and possible (financial) savings to cloud service consumers. On the other hand, such a large number of cloud service providers make it significantly difficult and complicated for cloud service consumers to select the services they need for fulfilling their IT requirements. This gives rise to various interesting but complicated research problems such as a large number of cloud service providers, varying level of Quality of Service (QoS) attributes among different cloud service providers, absence of standard service structure and measurement for QoS, and different Service Level Agreement (SLA) and business models.

1.2 Research Problem Description

This section describes the main research issues in cloud service selection and provisioning which provide rationale and motivation for the proposed research work.

1.2.1 Service selection and provisioning

Cloud service selection process involves filtering appropriate cloud service providers that can fulfil the requirements of cloud service consumers based on the constraints such as QoS attributes, functional requirements and Service Level Agreement. The selection process matches the input requirements of the cloud service consumers among different cloud service providers. For examples, RankCloudz (RightCloudz, 2014) , a commercial cloud service selection tool allows cloud service consumers to determine their priorities of QoS attributes and recommend the best cloud service provider. On the other hand, provisioning refers to the allocation of cloud providers resources or services based on the requirements or needs of the consumer.

1.2.2 Research problem

Given the large number of cloud service providers and the variety of services they offer, it becomes a challenge for cloud service consumers to select the cloud service provider that meets their specific IT computing requirements in particular cloud infrastructure services. Computing requirements here refers to the Information Technological needs that an organisation may have in order to fulfil a particular business process. In this case, the focus is on IT cloud infrastructure needs. However, the process of cloud service selection is not trivial. There are various factors that contribute to the complexity of service selection process.

First, the large number of cloud service providers means large number of cloud services available in the cloud market. For instance, if there are 100 cloud providers and each provides 20 services (on average) then there will be 2000 services to choose from.

Second, the cloud service providers provide services based on varying levels of QoS (Quality of Service) attributes such as performance, security, reliability, and privacy. For example, each cloud service provider gives different interpretation of QoS attributes, such as security and compliance. Google Cloud and Microsoft Azure have different specification for different attributes. Thus taking into account QoS attributes the process of service selection becomes further complicated.

Third, the service structure and representation are different for each cloud service provider. There is no standardization for the same service as each cloud service provider uses different terminology. Service measurement index (SMI) has been developed to standardize different

QoS. However, more work is needed to cover different subjective attributes such as price, usability, etc. Each cloud service consumer and provider has different expectations for price.

Fourth, the Service Level Agreement (SLA) between the cloud service providers and the cloud service consumers lists all the terms and conditions, contract violation and penalties. Due to lack of standardization, each provider has their own SLA. Such lack of standardization makes it difficult for cloud service consumers to select between different cloud service providers.

Fifth, the cloud service providers have their own proprietary business models and set different price for the same level of service. This makes cloud service selection complicated for the cloud service consumers. Based on the above issues and the literature survey, this research addresses the problems of:

- Taking into account QoS in the cloud service selection:
  Cloud service is associated with different QoS attributes. Depending on the level of knowledge of the different cloud service consumers, QoS attributes should be presented in an easy to understand, clear and structured way.

- Modelling and representing QoS attributes:
  Cloud service QoS consists of many attributes that are related to the characteristics and delivery of the cloud services. As there are enormous QoS attributes, it will be useful to categorize them in a structured way. This research aims to categorize QoS attributes in different categories such as technical, strategic & organizational, economic and political & legislative.

- Gathering information from different sources:
  Information from different sources such as web scraping, monitoring tools and user's review should be collected. This is to determine the score and ranking of cloud selection providers and to provide credible and unbiased results for the cloud service consumers.
- Designing of Analytical Hierarchy Process (AHP) algorithm
  Based on the information collected from different sources this research aims to design an AHP algorithm for ranking cloud service providers.
- Developing the cloud service selection prototype tool based on the proposed framework, that is, taking into account cloud services, QoS attributes, AHP algorithm, and user knowledge of cloud services.

1.3 Aim and Objectives

The aim of this research is to develop a framework for cloud service selection that addresses the aforementioned research problems and shortfalls of existing cloud service selection tools and ultimately this framework can help prospective cloud service consumers selecting the best cloud service providers that match their requirements. This aim is achieved through the objectives defined below:

1. Conduct a thorough and detailed study, review and analysis of academic research and existing commercial tools related to cloud service selection. The goal is to identify the shortfalls of existing academic research and commercial tools related to cloud service selection.
2. Build a new model to represent QoS attributes of cloud services. The objectives of developing this model are twofold. First, it would help prospective cloud service consumers to define their cloud service requirements. Second, this model provides foundations for the development of proposed prototype tool.
3. Develop a new framework for cloud service selection using AHP and SAW (Simple Additive Weight) methodologies. This framework utilizes three different sources of information such as web scrapping, cloudharmony monitoring tool and user's review.
4. Develop a prototype tool for cloud service selection based on the proposed framework. The aim is to improve and simplify the process of selecting the best cloud services by cloud service consumers. It will also take into account SLA and user's knowledge of the cloud technologies in the service selection process.
5. Testing and evaluation. The objective is to test the main functions of the tool. The tool would be compared with related cloud service selection tools based on selected criteria. In addition, the results of the tool would be validated for different consumer's requirements.

1.4 Overview of the Research Methodology

This section provides an overview of the methodology which is used in the design, development and evaluation of the proposed research for the cloud service selection. The methodology is based on a combination of different methods as there is no single methodology that covers all the requirements of the cloud service selection process. Further details and use of the methodology in the proposed research is explained in following chapters of this thesis. But the overall methodology is illustrated as follow.

1. Literature review and analysis

   In this literature review, step study is conducted that covers the area of cloud computing fundamentals, cloud service QoS, and decision making methodologies to be utilized in the cloud service selection process.
2. Design and development methods

Having reviewed the characteristics of cloud services and the process of cloud service selection, a model to represent cloud service QoS attributes is developed. Following the development of QoS attributes model, a complete framework for the whole process of cloud service selection is developed.

One of the underlying principles of cloud service selection is to compare different cloud services and rank them using certain QoS attributes and consumer's requirements. In order to compare and rank different cloud services, the commonly used methods are based on the multi criteria decision making methods(Whaiduzzaman *et al.*, 2014;Sun *et al*., 2014).This research adopts the Analytic Hierarchy Process (AHP) method which is one of the methodologies to solve multi criteria decision making problem. AHP method is a quantitative method and is based on pairwise comparison on each level of the hierarchy (Hwang and Yoon, 1981). In conjunction with the AHP, this research also uses SAW (Simple Additive Weighting) method in which linear additive function represents the preferences of decision makers. The thesis develops a mathematical model using AHP and SAW techniques.

Given the large number of cloud services it becomes unfeasible and inefficient to apply AHP and SAW methods to compare and rank all the cloud services. The proposed approach therefore exploits Skyline operator technique(Ouadah *et al.*, 2015;Borzsony, Kossmann and Stocker, 2001).that helps in reducing the number of cloud services so that the calculation during selection process only includes better and not-dominated cloud providers.

3. Design and development of cloud service selection tool

   In order to test the proposed research, a prototype tool was created and validated against a set of requirements from cloud service consumers. The design and development process involves requirement analysis step, which includes the description of the tool, requirements specification, and functionalities. In addition, it includes, user's interface design. Further, a NoSQL database is designed in order to store various data related to cloud service selection. Finally, evaluation of the tool includes evaluating the tool's functionalities and its results. Given that there is no standard method or tool of a cloud service selection, the proposed tool is compared with existing commercial cloud service selection tools.

1.5 Contributions of the proposed Research

This thesis makes various contributions to the current state of knowledge in the cloud service selection and provisioning area. The main contributions of this research are summarized as follows:

1. It proposes a new model in order to systematically represent the cloud services QoS attributes. The QoS attributes model covers both technical and non-technical aspects of cloud computing. The new model succinctly represents QoS attributes which cloud consumers can easily use and understand when selecting cloud services. These QoS attributes can seamlessly be fed into the multi-criteria decision method of AHP method.
2. Based on the QoS model the thesis proposes a new framework for cloud service selection. It improves and simplifies the cloud service selection process. It takes into account the level of user's knowledge of cloud computing technologies. The major benefit is to simplify the selection process for Beginner cloud service consumers by presenting the main QoS attributes to them with brief explanations. The other benefit is to give an Intermediate/Expert cloud service consumers an opportunity to go through more details of QoS sub-parameters.
3. It ensures the credibility of the service selection by utilizing information from three different sources namely information from cloud service providers, performance data from third-party monitoring tools, and user reviews.
4. The proposed framework integrates Service Level Agreement (SLA). SLA is an integral part of cloud service, and it is important for the consumer to be able to view it as part of their decision making process.
5. Development of a prototype tool that implements the proposed model and framework that outperforms existing commercial cloud services selection tools in terms of service selection, ease of use, and credibility of services ranking.

1.6 Structure of the Thesis

The remainder of the thesis is organized as follows:

Chapter 2	Literature Review – This chapter first introduces the background knowledge of cloud computing and describes the Service Level Agreement and Quality of Service attributes. Then, it presents review and analysis of related work which includes academic research work as well as commercial tools. In addition, it describes the Multi Criteria Decision Making technique and Skyline operator.
Chapter 3	Analysis and representation of QoS attributes- This chapter provides an in-depth analysis of the three commercial cloud service selection tools and presents a new proposed model for QoS.

Chapter 4	Cloud Service Selection Theoretical Framework and Methodology– This chapter discusses the cloud service selection framework components,process flow and describes methodologies used in the framework.
Chapter 5	Analysis and Design of Cloud Service Selection Tool- First, This chapter provides an overview of the service selection tool (SST) and the software/hardware platforms that are used in the implementation of the tool. Second, it defines certain assumptions and constraints about the designing of the tool. Third, it illustrates the designing phases of the tool.
Chapter 6	Cloud Service Selection Tool Implementation- This chapter describes the functional requirement of the SST tool. It also mentions implementation, validation and finally shows an interface design of the tool.
Chapter 7	Cloud Service Selection Tool Testing and Evaluation – This chapter presents the evaluation of the tool in comparison to existing cloud services selection tool.
Chapter 8	Conclusions and Future Work– this chapter concludes the thesis. It provides a summary of the work presented in this thesis. It illustrates the main contributions of the thesis. In addition, it gives a critical analysis of the work and sets directions for future research work.

CHAPTER 2

Background and Literature Review

2.1 Introduction

This chapter first introduces the background knowledge of cloud computing and describes the Service Level Agreement and Quality of Service attributes which are the main building blocks of cloud service selection. It then presents review and analysis of related work which includes academic research work as well as commercial tools. Finally, it describes the evaluation criteria such as Multi Criteria Decision Making and its variation Analytic Hierarchy Process which have been adopted in different existing approaches of cloud service selection.

2.2 Cloud Computing Fundamentals

Cloud computing enables on-demand access to a shared configurable computing resources such as servers, storage, and applications where these resources can be easily provisioned and released (Mell and Grance, 2011). Service oriented architecture (SOA) is a fast growing trend in the Information Technology industry. Cloud computing has therefore become a very attractive option on which consumers can run their business workloads. In addition, cloud computing helps consumer to avoid the huge investment and maintenance cost associated setting up computing infrastructure. Businesses can now convert their capital expenditure (CapEx) into operating expenditure (OpEx) making the cost of running production or service provision to be considerably less (Kushida *et al.*, 2011). Furthermore, small and medium scale businesses can now easily spin up new services over the Internet in a very short time using cloud computing. This allows them to be effectively competitive even with larger organisations. Cloud computing also allows for dynamic allocation of resources where multiple users can share a single hardware (Dawoud *et al*., 2016). This dynamic allocation leads to efficient resource management, low running cost and power savings (Youssef & Alageel, 2012). This large scale sharing of resources introduces new set of challenges such as security, privacy, performance and service quality (Zhang *et al*., 2010; Armbrust *et al*., 2010).

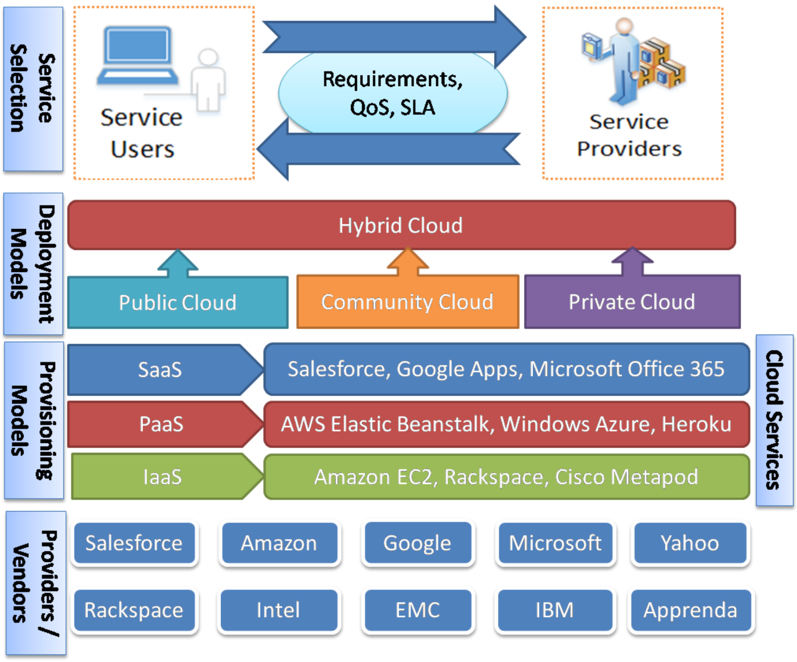


Figure 2.1 Cloud Services Infrastructure

Figure 2.1 describes the cloud service infrastructure which is proposed in (Eisa *et al*., 2016). Nowadays various cloud service providers, such as IBM, Intel, Yahoo, Microsoft, Google, Salesforce, Rackspace, Amazon, Apprenda, and EMC offer cloud service consumers with diverse services, for instance compute, network, storage, software applications, memory/CPU, communication services, etc. This service oriented model is provisioned through the Internet and allows cloud consumers to pay for only what they use (Jadeja and Modi, 2012; Patel, Ranabahu, and Sheth, 2009). Cloud services provided to the cloud service consumers through the service provisioning models. There are three different service models of cloud computing (Mell and Grance, 2011):

1. **Software as a Service (SaaS)**

   A service model where consumer access is on application level and the software runs as a service in a multi-tenant platform (Rimal *et al*., 2009). Applications run on cloud infrastructure and are available to consumer through thin client interface such as internet browser, program interface such as mobile phone application. Cloud service consumers do not have full control. In contrast with traditional software consumers, SaaS consumers pay a subscription from the publisher to use their software as against buying the software outrightly (Choudhary, 2007). They simply use the applications. Examples are Google Docs and Email.

2. **Platform as a Service (PaaS)**

   A service model where consumer is able to deploy their own application onto a cloud infrastructure, while the control of the underlying infrastructure such as network, servers and operating system remains in the hand of cloud providers. PaaS Consumers are able to control the development, deployment and the configuration of the application (Lawton, 2008; Boniface *et al*., 2010). In PaaS cloud service, cloud service consumers receive the required resources from cloud service providers to create their own applications such as different programming languages and tools.

3. **Infrastructure as a Service (IaaS)**

   A service model where consumers have control over the hardware aspects of the cloud infrastructure that includes operating systems, applications, storage, networks and their applications. IaaS consumers lease hardware such as processing power, storage etc. and other basic resources and can deploy their own operating system and application on the computing resources (Peng *et al*., 2009). Examples include Google Compute Engine.

The most common cloud deployment models are as follows (Zissis and Lekkas, 2012):

1. **Public Cloud**

   In this model, cloud infrastructure can be provided to general public. As a result they can deploy or use their services. Such infrastructure may belong to businesses, academic institutions, or government organizations.

2. **Private Cloud**

   In this model, an organization owns the cloud resources, which are used for exclusive use by a single organization. The cloud infrastructure is operated solely by an organization (Sabahi, 2011), it is possibly managed by the organization or a third party and may exist on premise or off premise.

3. **Hybrid Cloud**

   This model is a combination of two or more distinct clouds such as private, community, or public cloud (Goyal, 2014). For example, an organization can use hybrid cloud model where it utilizes public cloud for less critical applications and the private cloud for the critical and sensitive applications.

4. **Community Cloud**

   In this model, the cloud infrastructure is limited for special use by a specific community of cloud service consumers that have shared interests such as, Healthcare community cloud (Garlick, 2011; Li *et al*., 2011; Briscoe and Marinos, 2009).

2.2.1 Service Level Agreement

This section introduces the concept of Service Level Agreement (SLA) and the different types of SLAs. The SLA is described in the context of cloud service selection in Section 2.3.2. Note that the work proposed in this thesis takes into account the main principle or requirements of

SLA in the cloud service selection. However, detailed investigation into SLAs such as designing new SLAs, comparison of different SLAs, are beyond the scope of this thesis.

Service Level Agreement (SLA) is an agreement document made between two different parties namely the cloud service provider and the cloud service consumer (Mirobi and Nadu, 2015). SLA is made and agreed upon for the main purpose of monitoring the performance of the service delivered by the provider, usually based on the service consumer's requirement. SLA is a contract where the agreed upon service and the period of the contracted service is defined. Managing SLAs can sometimes be a cumbersome process (Muthusamy and Jacobsen, 2008).

Specifically in cloud computing, SLA includes the specification of the cloud service provided by the cloud service provider and this represents the necessary commitment and promise between a cloud service consumer and cloud service provider (Taha, Manzoor and Suri, 2017). SLA is important in cloud computing because it is used for the purpose of monitoring and reporting the cloud service performance. Some of the significance of SLA is listed below (Mirobi and Nadu, 2015):

1. SLA provides the cloud service consumer with information about the cloud service providers.
2. SLA provides consumers with detailed information about the cloud service.
3. SLA provides description about the purpose and objectives based on business policies.
4. Through SLA cloud service consumer is able to identify important security and management strategies of the agreement.
5. SLA is used for the purpose of monitoring the quality of service, service performance, and service response time.
6. Through SLA consumer is able to get the notion about what is required of them in managing the service in case the cloud service performed poorly.

Agreement between cloud service consumer and cloud service provider might vary, and therefore there are several classification of SLA as listed below (Mirobi and Nadu, 2015):

1. **Service-Based SLA**

   The service-based SLA is based on the cloud services required to be delivered.

2. **Customer-Based SLA**

   The customer-based SLA includes all the services required by a particular cloud service consumer that need to be delivered by the cloud service provider.

3. **Multi-Level SLA**

a. **Service Level SLA**

SLA that includes service level management issues for a specific cloud service.

b. **Customer Level SLA**

SLA that includes service level management issues for a specific customer group.

c. **Corporate Level SLA**

SLA that includes basic service level management issues for all the customers within a company.

2.2.2 Fundamentals of Quality of Service Attributes

Tied closely to the service selection in cloud computing and its SLA is quality of service which is often abbreviated as QoS. As defined by International Organization for Standardization in ISO 9000, "the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs"(Reeuwijk *et al*., 1998).In ISO's latest update standard, namely ISO 9000:2015, however, quality is defined as a "degree to which a set of inherent characteristics of an object fulfills requirement" (ISO, 2015). The gist of both definitions point to a similar meaning, and applying the definition of quality into service, it can be inferred that quality of service is a degree to which a set of characteristics of a service fulfills requirements.Quality of service attributes refers to these set of characteristics of a particular service that enables said service to fulfill requirements.Quality of Service (QoS) is associated with non-functional aspects of a service.

In the context of cloud computing, QoS symbolizes the level of performance, reliability, security and availability of a cloud service and by the infrastructure upon which the service is hosted (Ardagna *et al.*, 2014). For cloud service consumer, QoS is important because cloud service provider is expected by cloud service consumer to deliver the advertised quality characteristics. In addition, a successful cloud service would provide the necessary functionality within the specified time limit (Wang *et al*., 2014). For cloud service provider QoS helps in finding the best trade-off between QoS levels and operational costs (Ardagna *et al.*, 2014). QoS helps cloud providers find the middle ground where they can deliver service on acceptable level while maintaining reasonable operational costs needed to produce or deliver the service.

2.3 Cloud Service Selection

Cloud service selection is an area that has been the subject for many research works for the past several years. Li and Yang (2010) suggested that cloud service selection is an important task to do in the process of adopting cloud technology. Furthermore, the diversity of services among the numerous cloud providers complicates selection process (Rehman and Hussain, 2011; Achar and Thilagam, 2014). Using the tool called CloudCmp, Li and Yang (2010) compared the performance and cost of four major cloud service providers, namely, Google AppEngine, Microsoft Azure, Amazon AWS, and Rackspace. These cloud providers represent the leaders of the cloud computing service market. The result of this comparison shows that even for the same type of cloud service, cloud providers cost and performance can vary from one to another. This result emphasizes the need for a thorough and thoughtful cloud service provider selection. Zeng *et al*. (2009) suggested two main factors in cloud service selection to include Maximized Gain (in terms of resources and functionality) and minimised cost. The important role that cloud service selection process plays lead many researchers to keep exploring new techniques and methodologies to be applied to service selection. For instance Sundareswaran *et al*. (2012) described a broker based service selection mechanism that can help consumers select cloud services.

The status of researches in cloud service selection field is surveyed in (Sun *et al*., 2014) wherein the authors evaluate around twenty five studies on this topic. All the studies reviewed in this paper are divided based on the different techniques employed in these studies. Accordingly, the following four categories of techniques are defined for cloud service selection.

1. **Cloud service selection with MCDM-based techniques**

   In a decision making process, there are the criteria by which the decision is based on alternatives which are the different options that decision maker can choose as the end result of the decision making process. When there are finite criteria and alternatives, then decision making process can be done with Multi Criteria Decision Making (MCDM) techniques.

2. **Cloud service selection with optimization-based techniques**

   Optimization is the process of finding the most suitable cloud services by optimizing, minimizing or maximizing, the criteria while sticking to the constraints (Dastjerdi and Buyya, 2011). Several optimization techniques have been used in cloud service selection such as dynamic programming, integer programming, and greedy algorithm.

3. **Cloud service selection with logic-based techniques**

   First order logic is a logic-based knowledge representation where each statement is separated into a subject and predicate. A subset of first order logic, description logics structure the knowledge of an application domain based on a knowledge base containing domain-related terminologies. In cloud service selection, logic based algorithms are created for service matching using domain ontologies (concepts, definitions, properties, relations in a particular domain). Match making between service description and user requirements.

4. **Cloud service selection with other (Miscellaneous) Approaches**

   Examples of other approaches used for cloud service selection are matrix factorization, service selection strategies, Fuzzy Inference system (FIS) and computational cost calculation approach.

From twenty five different researchers analysed by Sun *et al*. (2014) it can be concluded that the majority of the studies utilize MCDM techniques for cloud service selection. This research in particular utilises AHP and SAW techniques of MCDM techniques in cloud service selection.

2.3.1 Quality of Service Attributes in Cloud Service Selection

In the context of cloud service selection process, QoS attributes play one of the most important parts because they are used as the criteria for service selection (Armstrong and Djemame, 2009). Previous studies have been done in reviewing and analysing the QoS attributes in cloud service selection. Some studies use single quality attribute while other studies use a multi-dimensional QoS attributes. Salama *et al*.(2012) propose a cloud service selection based on a group of broad, multi-dimensional QoS, and not just a single quality attribute. Where single quality attribute selection is based on performance and cost-benefit analysis, multi-dimensional QoS selection takes into account user's requirements, performance attributes, quality of service, as well as business profitability. An integrated QoS-assured utility model is proposed to address the problem of cloud service selection. There are two phases of selection where consumers need to specify specific scores for the QoS attributes as the requirement to select cloud service providers.

While some researches limit the selection process within a set of pre-defined cloud service criteria, some other researches take advantage of previous effort to make a set of standardized cloud quality of service (QoS) attributes called SMI (Service Measurement Index)proposed by CSMIC (The Cloud Service Measurement Initiative Consortium (CSMIC), 2011). Garg *et al*. (Garg, Versteeg and Buyya, 2011), (Garg, Versteeg and Buyya, 2013) create SMICloud which is a framework for comparing and ranking cloud services. SMICloud uses standardized SMI attributes in order to compare cloud service providers based on cloud service consumer requirements. Similar to aforementioned studies, SMICloud also uses Analytic Hierarchy Process (AHP) as the methodology to rank the cloud services. Baranwal and Vidyarthi (2014) propose a framework for selecting best cloud service providers using SMI framework as the standard QoS attributes. In addition, it also considers more attributes along with the SMI attributes, such as reputation of cloud service provider, location of data centre etc. The difference with SMICloud is that this framework utilizes ranked voting method to find the best cloud service provider. A list of metrics of efficient cloud provider acts as the voter and cloud service providers act as the candidates. In this methodology, voter ranks the candidates in order of preference. First, calculate preference scores for each cloud service providers, then normalized the preference scores for each cloud service providers and the highest normalized preference scores is referred to the best cloud service providers. One of the benefits of this methodology is to quantify each metric separately so that metrics are categorized in two categories as application dependent and cloud service consumer dependent. In addition, it is flexible and dynamic. It is flexible because cloud service consumers can select some attributes according to their needs, and also support if cloud service consumer is interested among different best cloud service providers for each application. It is dynamic due to the ability of cloud service consumer to select cloud service which does not only satisfy its current needs, but also adapt with its future requirement. However, the major issue is that it is complicated for ordinary cloud service consumers to understand this technique and it does not consider criteria interdependency relationship.

2.3.2 Service Level Agreement in Cloud Service Selection

Service Level Agreement (SLA) is an important and integral part of cloud service selection (Son *et al*., 2017). SLA can be included in the selection process to help filter out best providers. A study by Baset (2012) was done to analyse SLAs of several cloud service providers to find the common characteristics as well as compare them with each other. The structure of a typical cloud SLA has the following components (Baset, 2012):

1. **Service guarantee**

   There are measurable metrics in the SLA that cloud service provider tries to meet during a service guarantee time period.

2. **Service guarantee time period**

   It defines the duration or the length of time during which a service guarantee should be delivered by service provider.

3. **Service guarantee granularity**

   It describes the scale that the service provider uses to specify the service guarantee. For example, a service can be guaranteed per instance, per data centre or per transaction basis.

4. **Service guarantee exclusions**

   It describes the instances that are not included in the service guarantee metric calculation.

5. **Service credit**

Service credit is the amount of credit cloud service consumer gets if the service guarantee is not met.

6. **Service violation measurement and reporting**

   It describes the details regarding the procedure to measure and report service guarantee violations.

SLA is always provided by cloud computing providers however, the level of guarantee provided differs from one provider to the other i.e. there is no standard or generalized SLA. SLA can also be utilized for cloud service selection process. Wagle *et al*. (2015) propose an evaluation model to select cloud services from cloud service providers that utilize cloud auditors. Cloud auditors in this model verify cloud services whether or not they are in compliance with the SLA offer from the cloud service providers. The output of the selection process in this model is ranking and recommendation of cloud providers based on providers' SLA and the performance of the service delivered.

2.3.3 Commercial Tools for Cloud Service Selection

In addition to academic studies and research done to analyse and propose different methodologies that are possible to solve cloud service selection problem, there are currently commercial tools available to help cloud service consumers in selecting cloud services. Academic work on cloud service selection has been thoroughly reviewed. However, none of the existing research (academic or commercial) reviews or compares the commercial tools of cloud service selection. Analysis and comparison of the three commercial tools are discussed in the paper (Eisa *et al*., 2016). There are some other commercial tools for cloud service selection available such as CloudScreener, and MagiCloud. But this work only focused on and reviewed the three well known commercial cloud service selection tools, which are Intel Cloud Finder, RankCloudz and Cloudorado. Due to the commercial nature of these tools it is not possible to get hold of the underlying models, algorithms or other technical details related to the design and development of these tools. As a result, this work compared these tools based on the information provided through their web pages which are publicly available on the Internet. This thesis reviews and analyses these three commercial tools as they are related to the proposed approach developed in this research. In relation to the proposed approach, the next chapter provides further analysis of these commercial tools.

2.3.3.1 Intel Cloud Finder

Intel Cloud Finder is a cloud service selection tool built and provided by Intel. There are two types of searches in Intel Cloud Finder, namely Quick Search and Detailed Search where each type of search presents a different sets of cloud QoS attributes (Intel®, 2013). As shown in Figure 2.2, the Quick Search cloud service consumer can quickly search for a cloud services that satisfy their requirements by choosing features that closely match their specification. There are five top-level attributes in Quick Search: Interface Model, Development Support, Subscription Options, Geography and Verticals. Each of these five attributes have several options from which cloud service consumer can choose the ones that are close to their requirements.

- Interface Model reflects the type of interaction available for cloud service consumer for accessing and configuring their cloud infrastructure that cloud service providers are providing.
- Development Support reflects the added support that allows cloud service consumer to customize their platform such as storage services as well as custom OS images.
- Subscription options are the available ways for cloud service consumers to purchase or pay for the services and infrastructure they are leasing.
- Geography is for the region where a cloud service provider is operating.
- Verticals reflect the industry where the cloud service consumer would use the services and infrastructure they are trying to acquire.

Figure 2.2 Intel Cloud Finder Quick Search

The Detailed Search provides cloud service consumers more flexibility and determines the importance of the attributes presented to them. Similar to the Quick Search, the attributes were categorised in a way that contain sub-attributes, and each sub-attribute has several options; all of which contain brief description of what the features are. The parameters are Security, Usability, Quality, Availability, Technology and Business.

- Security reflects how accessible the cloud infrastructure is as well as the security measures put in place. It also implies different regulations and standards.
- Usability reflects the service capabilities and monitoring features that are available to the cloud service consumers when accessing and leasing an infrastructure.

- Quality is composed of how easy it is for the cloud service consumers to use the services as well as level of agreement available for them to choose.
- Availability reflects the way for cloud service consumers to access and control their infrastructure in cases of emergencies. It also implies location availability where a particular cloud service provider operates in a particular region.
- Technology reflects functional requirements of cloud services are presented to the cloud service consumers as different features.
- Business gives information about the cloud service providers such as revenue, data centres locations, etc.

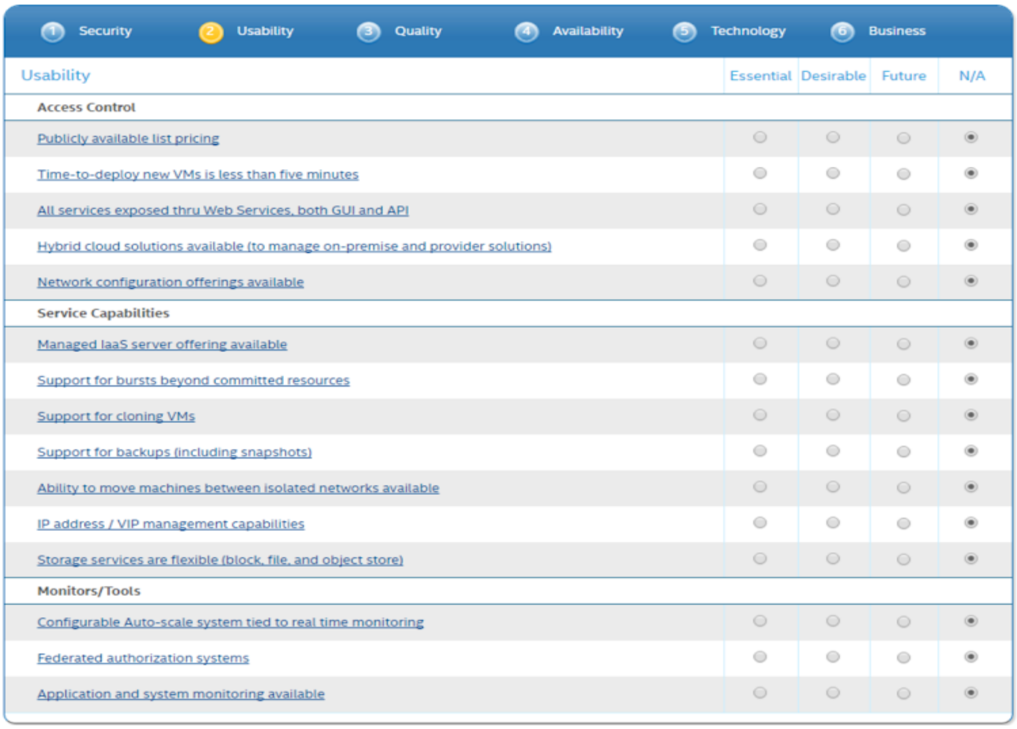


Figure 2.3 Intel Cloud Finder Detailed Search

Cloud service consumers need to determine the level of importance of each option under each sub-parameter. There are four different levels of importance, ranging from the 'most important' to the 'least important': essential, desirable, future, N/A. The fourth option is the default value for all options. Intel provides definitions of all the features in the Detailed Search as shown in Figure 2.3 and it also shows the selection process interface of the Detailed Search.

2.3.3.2 RankCloudz

RankCloudz selection process requires cloud service consumers to determine the level of importance of a set of attributes. There are five different cloud services that cloud service consumer can choose: Dev & Test Infrastructure, Virtual Data Centre, Enterprise Apps &

Hosting, Storage & Backup and Big Data & Analytics. Each of service has its own set of attributes, ranging between 11 to 14 attributes (RightCloudz, 2014). Figure 2.4 below shows the interface for selection process on RankCloudz.

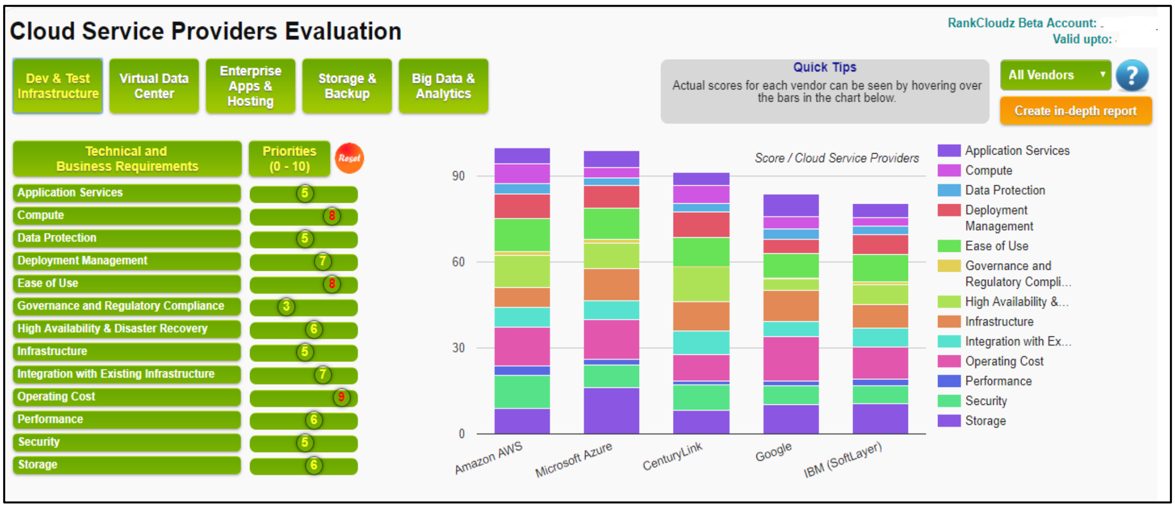


Figure 2.4 RankCloudz Service Selection Interface

The level of importance in RankCloudz is represented by a scale that starts from 0 as the lowest importance and it goes up to 10 as the highest importance. RankCloudz ranks the cloud service providers depending on the attributes and the priority that cloud service consumer has set for each attribute.

2.3.3.3 Cloudorado

Cloudorado selection process works like Intel Cloud Finder Quick Search does where cloud service consumer chooses an option from each attribute that matches their requirements (Cloudorado, 2011). Cloudorado differentiates three different types of cloud services: Cloud Server, Cloud Hosting and Cloud Storage. Each of these cloud services has different set of attributes. Cloud Server and Cloud Hosting categories have the same non-functional attributes presented to the cloud service consumers but with different functional attributes. Cloud service consumers can configure both the functional and non-functional attributes that satisfy their requirements. Figure 2.5 shows an example of service selection on Cloudorado.

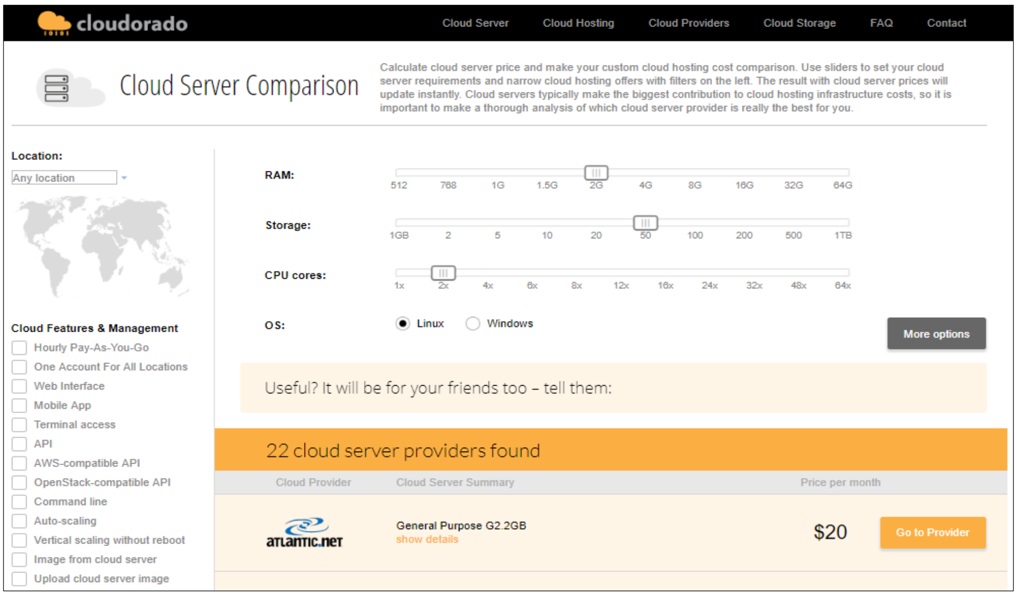


Figure 2.5 Cloudorado Service Selection Interface

2.4 Multi Criteria Decision Making

As previously discussed, MCDM is a popular technique utilized for cloud service selection. MCDM techniques are useful in decision making when there are multiple (and often conflicting) combination of criteria (Pohekar and Ramachandran, 2004). Whaiduzzaman *et al*. (2014) further review the wide application of MCDM in cloud computing. They argue that even though multi criteria decision analysis is a widely applied and well-established technique in the field of operation research, the same multi criteria decision analysis techniques are also proven to be effective to be applied within the field of cloud selection (Whaiduzzaman *et al.*, 2014). For instance, Seo and Lee (2016) apply MCDM methods to develop a model for cloud service selection using Fuzzy Analytical hierarchy process. Rehman *et al*. (2012) use MCDM methods in selection of IaaS services using performance measurements of cloud providers. Gavade (2014) categorizes MCDM into Multiple Attribute Decision Making (MADM) and Multiple Objective Decision Making (MODM). MADM involves selection of best option from a group of specified alternatives or set of attributes and MODM involves the design of alternatives that will optimize the decision making process from among a set of objectives. Around sixteen studies in Whaiduzzaman *et al.*(2014) cloud selection using a wide array of MCDM techniques are discussed in this article with Analytical Hierarchy Process (AHP) being the technique used the most in these studies.

The application of AHP technique in cloud service selection can be found in many studies. Sun *et al*. (2013) introduce AHP into the process of selecting cloud service in a simple way with a defined set of service selection criteria and limited cloud service provider alternatives. A handful of variation to AHP can also be utilized. For example, the combination of fuzzy logic and AHP are combined into a Fuzzy AHP technique in order to select the optimal cloud-path for cloud offloading purpose in mobile cloud computing environment (Singla and Kaushal, 2015). There are also many studies that use AHP in combination with other MCDM techniques. A combination of AHP and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is utilized in a fuzzy environment to find the best and the most optimal cloud service option for offloading purpose in mobile cloud computing systems (Wu, Wang and Wolter, 2012). Ouadah *et al*.(2015) combine MCDM methodologies AHP and Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) with a database operator called Skyline operator to enhance web service selection (Ouadah *et al*., 2015). Even though the selection process is for web services and not cloud services, the main idea is still to pick the best services according to user requirements. Skyline operator is used to reduce the number of services candidates so that the selection process can only focus on services that are not dominated by other web services.

2.4.1 Analytic Hierarchy Process (AHP)

AHP is one of the methodologies to solve multi criteria decision making (MCDM) problem which is first proposed by Saaty in 1977 (Saaty, 1977). AHP models the problem in a hierarchical structure. In Karim (2013), the researchers propose a mechanism to map QoS requirements to QoS specifications making use of AHP to models the QoS specification. The approach in AHP is based on pairwise comparison on each level of the hierarchy. AHP methodology can be divided into several steps (Hwang and Yoon, 1981):

- Determine the hierarchical structure of the problem by decomposing the problem at hand into hierarchy.
- Compare the comparative weight between the attributes of the decision elements to form the reciprocal matrix.
- Synthesize the individual subjective judgement and estimate the relative weight.
- Aggregate the relative weights of the elements to determine the best alternatives.

The hierarchical structure of the cloud service selection problem is shown in Figure 2.6.

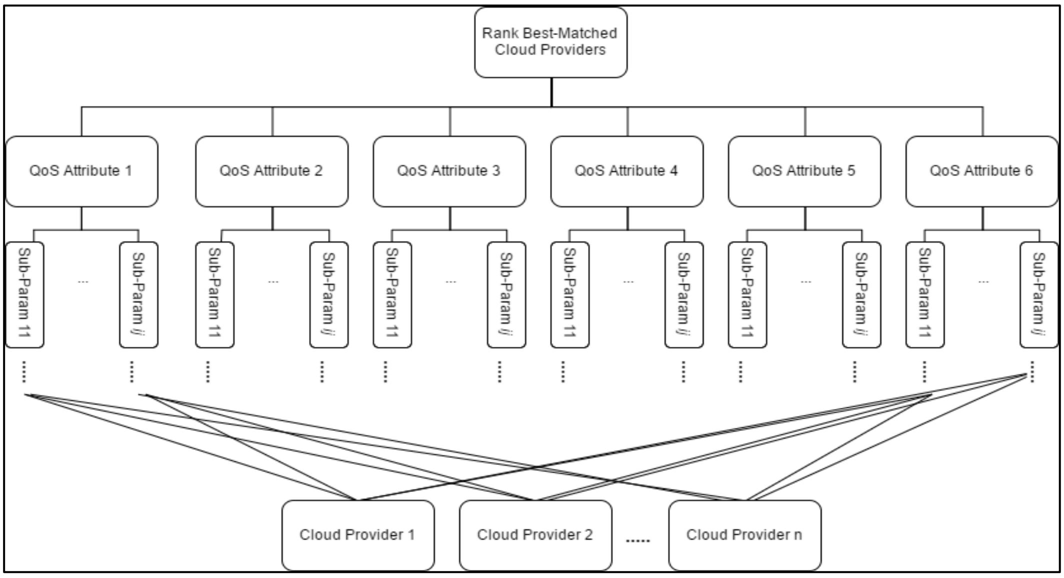


Figure 2.6 AHP Hierarchical Structure

The next step is to process the comparative weight between attributes. Inputs are needed from the cloud service consumer regarding the weight of each attribute relative to another attribute. This comparison is called pairwise comparison because attributes comparison is done in pair. For n attributes involved in the problem, there needs to be $\frac{n^2-n}{2}$ pairwise comparisons. The scale of the weight for each comparison is defined by Saaty below:

Table 2.1 Comparison Weight Importance in AHP

Intensity of Importance	Definition
1	Equal importance
3	Moderate importance of one over another
5	Essential or strong importance
7	Very strong importance
9	Extreme importance

2.4.2 Simple Additive Weight

Simple Additive Weight (SAW) is a method that is often considered as the most intuitive. Firstly utilized as a method to cope with a portfolio selection problem, SAW offers an easy way to deal with multiple criteria decision making problems because the linear additive function can represent the preferences of decision makers (Hwang and Yoon, 1981). Because of its simplicity feature SAW is the most popular technique in multiple attribute decision making.

Simple additive weight is a method that is based on weight of all the attributes involved and how to get the weighted average. The calculation is done by multiplying the value of an attribute with the weight assigned by decision maker and followed by summing of the products for all attributes (Afshari, Mojahed and Yusuff, 2010).Saripalli *et al*. (2011) adds that SAW methodology assumes that each attribute is independent and does not have influence on other attributes.

2.5 Skyline Operator

The term Skyline is used to define a set of points that are not dominated by any other point (Abourezq and Idrissi, 2014). A point is said to dominate another point if it is as good or better in all dimensions and better in at least one dimension (Borzsony, Kossmann and Stocker, 2001). Taking the idea of Skyline operation, Borzsony *et al*. (2001) integrated skyline operation into a database system, and thus born Skyline Operator. In the context of cloud service selection, given S_i and $S_j \in S$, it is defined that S_i dominates S_j, denoted by $S_i \prec S_j$, if S_i is better or equal to S_j on all attributes of QoS and strictly better on at least one attribute of QoS (Ouadah *et al.*, 2015).

Skyline operator essentially filters out only a few select cloud service providers that are not dominated by another cloud service provider on all QoS attributes. Given that the number of cloud service providers available currently is very high, the implementation of Skyline operator can help in lowering the number of cloud service providers so that the calculation during selection process only includes the better, not-dominated cloud service providers.

Two popular techniques to implement Skyline operator are (Borzsony, Kossmann and Stocker, 2001):

1. **Block-nested-loops Algorithm**
   Block nested loops algorithm is similar to naive nested loops algorithm in that this algorithm repeatedly reads set of tuples (Abourezq and Idrissi, 2014). The objective of this technique is to have a window of incomparable tuples in main memory. When a tuple *p* is read from the input, *p* is then compared to all the tuples in the window. If *p* is not dominated by any of the tuples in the windows then *p* is added to the window. Otherwise, *p* is then eliminated. When tuple *p* dominates other tuples in the window, then similarly, these dominated tuples in the window will be eliminated.
2. **Divide and Conquer Algorithm**
   Divide and conquer algorithm breaks down a problem into smaller sub problems and solves the sub problems recursively. It then finally combines the sub problems' solutions to solve the parent or original problem.

In its application to Skyline operator, this algorithm first finds the median *m* of the input for dimension *d*, divides the input into two parts, *P1* and *P2*. *P1* contains all tuples whose value of attribute d is better than m while *P2* contains all the other tuples. It then computes the skylines *S1* of *P1* and *S2* of *P2*. In doing this, the divide and conquer algorithm is recursively applied to *P1* and *P2*. The recursive partitioning stops if a partition contains only one tuple. Finally the overall result of the Skyline is the result of merging S1 and S2. Note that a tuple in S1 is better in dimension d than among all tuple of S2 thus none of the tuples in S1 is dominated by a tuple in S2.

2.6 Summary

Based on reviewing the existing research and development cloud service selection several issues have been identified. This section provides a summary work on of the research issues and challenges which are addressed in the following chapters of this thesis.

One of the major issues is that the current techniques do not provide standard definitions and representation of the cloud services and the QoS attributes. In addition, these techniques are complex for the cloud service consumers with limited knowledge of cloud technologies. Majority of the academic researchers are considering cloud service consumers knowledgeable and expert.

Cloud Service Measurement Initiative Consortium CSMIC introduced the SMI, which is useful step towards standardizing the key performance Indicator KPI for measuring and comparing services. However, some of the quality attributes can be quite subjective and their measurements can be perceived differently by different cloud service consumers or cloud service providers. For instance, the price of the same service under same conditions can be perceived as cheap by some although others consider it as expensive.

Standardising SLAs would be appealing to cloud service consumers, cloud service providers and brokers. However, due to the large number of cloud service providers and cloud service consumers, the different types of cloud services and locations pose a big hurdle to develop standardised SLAs.

Commercial cloud service selection tools facilitate the selection process to some extent. However, there still exist some shortfalls that need further enhancements. For example, improving the usability by providing cloud service consumer knowledge with brief explanations of the QoS attributes, presenting main attributes for Beginner and a more detailed sub-parameter for Intermediate/Expert cloud service consumers respectively. In addition, incorporate a template of SLA to give the cloud service consumers an instance look for the terms and conditions, definitions of availability and migration fees to different cloud service providers before deciding on which cloud service providers to go with and select.

CHAPTER 3

Analysis and Representation of QoS Attributes

3.1 Introduction

In order to design effective cloud service selection system it is crucial to systematically represent various QoS attributes related to cloud services. There exist a large number of cloud services and each service is associated with a number of QoS attributes. This makes the representation of QoS attributes difficult. There is no standard model for representing QoS attributes in cloud service selection. Different tools use proprietary schemes for representing QoS attributes. This chapter first provides an in-depth analysis of the three commercial cloud service selection tools as well as the way they represent QoS attributes. It then presents a new model that succinctly represents QoS attributes of cloud services.

3.2 Analysis of the QoS Attributes in Commercial Tools

The previous chapter described the three commercial service selection tools. This chapter provides an in-depth analysis of these tools with respect to the definition and representation of the QoS attributes. This analysis provides the basis and rationale for the development of the new QoS model which is developed in this thesis.

3.2.1 Intel Cloud Finder

Intel Cloud Finder has two different types of search, Quick Search and Detailed Search. Quick Search is suitable for novice consumers because of the quick and simple process as well as the relatively simple presentation of the QoS attributes. However, explanation is lacking in quick search. Out of five attributes and a total of twenty-five options, Intel only provides explanation for three options which are Pay-as-You-Go (under Subscription Option), Spot Instance Bidding (under Subscription Option), and Reserved Instances (under Subscription Option). The complete QoS attributes for Intel Cloud Finder Quick Search is represented in Figure 3.1.

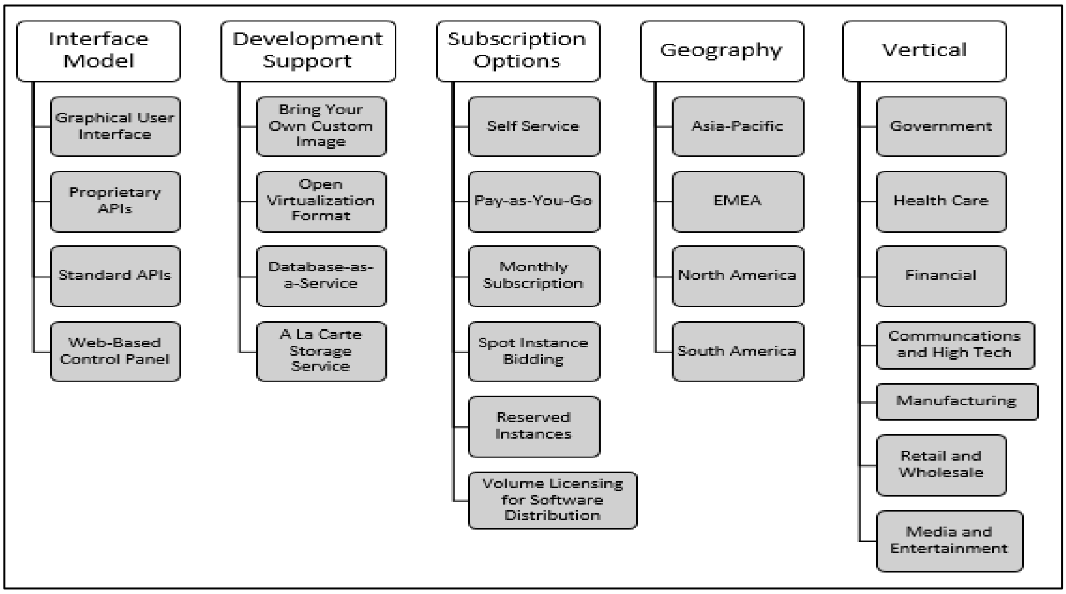


Figure 3.1 Summary of Intel Cloud Finder QoS Attributes

The Interface Model reflects the type of interaction available for cloud service consumers. Below are the four options of interface model that cloud service consumer can select

- Graphical User Interface
- Proprietary APIs
- Standard APIs
- Web-Based Control Panel

Development Support reflects the support added to allow cloud service consumers to customize their platform such as storage services as well as custom OS images. Below are the four options of development support that cloud service consumers can select:

- Bring Your Own Custom Image
- Open Virtualization Format
- Database-as-a-Service
- A La Carte Storage Service

Subscription Options are the different payment methods for cloud service consumers to purchase or pay for the services and infrastructure they are leasing. Below are the six options of subscription options that cloud service consumers can select:

- Self Service
- Pay-as-You-Go
- Monthly Subscription
- Spot Instance Bidding
- Reserved Instances
- Volume Licensing for Software Distributions

Geography is for the region where a cloud service provider is operating. Below are the four options of geography locations that cloud service consumers can select:

- Asia-Pacific
- EMEA
- North America
- South America

Verticals reflect the industry where the cloud service consumer would use the services and infrastructure they are trying to acquire. Below are the seven options of industry that cloud service consumers can select:

- Government
- Health Care
- Financial
- Communications and High Tech
- Manufacturing
- Retail and Wholesale
- Media and Entertainment

For the Detailed Search Intel utilizes 2-level of QoS attributes, represented in Figure 3.2.

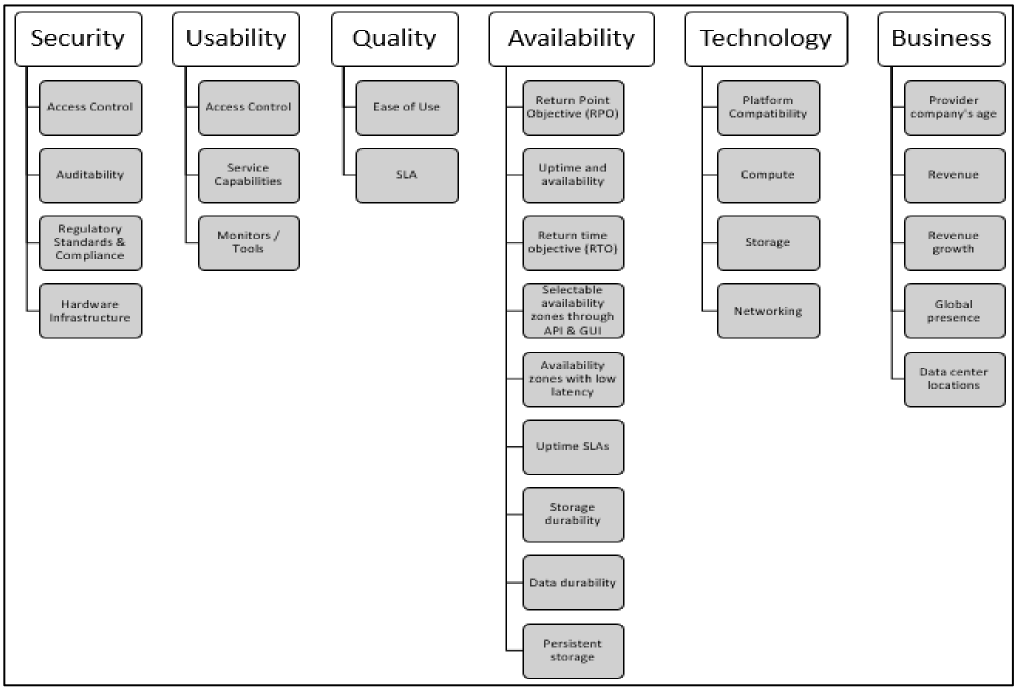


Figure 3.2 Summary of Intel Cloud Finder QoS Attributes – Detailed Search

Security reflects the ways cloud service consumers can access as well as protect the infrastructure they are leasing. It also implies different regulations and standards. Listed below are the sub-parameters for Security:

- **Access Control**
  - Provides clear guidance to allow cloud subscriber to make an informed choice on the required security level.
  - Subscribers can manage access models for internal to subscriber allocation (i.e. not one key for the whole company).
  - Subscribers can choose where their infrastructure and data resides.
  - Offer peer service monitor APIs.
  - Network traffic and threat analysis APIs.
  - Dedicated private cloud with physical isolation offering available.

- **Auditability**
    - Solutions allow assurance levels to be tracked in customer premise SIEM and compliance monitoring tools.
    - Identity management auditability.
    - Patch management auditability.
    - API's for auditing Platform Trust.
    - Import/export of logs from SIEM for auditability.
- **Regulatory Standards & Compliance**
    - ISO 27001 and 27002 certified.
    - SSAE16 compliant.
    - PCI / DSS compliant solutions.
- **Hardware Infrastructure**
    - Dedicated hardware firewalls protects entire environment.
    - VPN-only access to cloud environment with ability to guarantee minimum throughput.
    - Dedicated-circuit-only access to cloud environment with ability to guarantee minimum throughput.
    - Network layer IPS/antivirus options.
    - Layer-2 traffic isolation.

Usability reflects the service capabilities and monitoring features that are available to the cloud service consumers when accessing and leasing an infrastructure. Below are the listed sub-parameters for the Usability:

- **Access Control**
    - Publicly available list pricing.
    - Time-to-deploy new VMs is less than five minutes.
    - All services exposed thru Web Services, both GUI and API.
    - Hybrid cloud solutions available (to manage on premise and provider solutions).
    - Network configuration offerings available.
- **Service Capabilities**
    - Managed IaaS server offering available.
    - Support for bursts beyond committed resources.
    - Support for cloning VMs.
    - Support for backups (including snapshots).
    - Ability to move machines between isolated networks available.
    - IP address / VIP management capabilities.

    - Storage services are flexible (block, file, and object store).

- **Monitors/Tools**
    - Configurable Auto-scale system tied to real time monitoring.
    - Federated authorization systems.
    - Application and system monitoring available.
- **Support**
    - 24 hours support available.
    - Phone support available.
    - Chat support available.
    - Email support available.
    - Forum support available.

Quality is composed of how easy it is for the cloud service consumers to use the services as well as level of agreement available for them to choose. Listed below are the sub-parameters for the Quality:

- **Ease of Use**
    - New deployments in less than one hour.
    - 24-hour support available.
    - Able to chart history of usage, by VM, by tenant, by time, by traffic type.
    - Billing/invoicing reflects actual usage (storage, network, compute, etc.).
- **SLA**
    - Ability to guarantee minimum usage and specify maximum peak usage based on subscription.
    - Data integrity, reliability, and resiliency SLAs.
    - Ability to specify network bandwidth and latency requirements.
    - Ability to specify storage bandwidth and latency requirements.
    - Network Uptime Guarantee on SLA.
    - Latency Guarantee for North America east to west coast on private network.

Availability reflects the way for cloud service consumers to access and control their infrastructure in cases of emergencies. It also implies location availability where a particular cloud provider operates in a particular region. Listed below are the sub-parameters for Availability:

- Return point objective (RPO) and data loss offering available (e.g. back-up) published history of uptime and availability.
- Return time objective (RTO) offering available (e.g. disaster recovery/restore).
- Minimum of two selectable availability zones through an API and GUI.

- Availability zones with low latency less than 1mS or 100km (failover – synchronous mirroring).
- Demonstrable uptime SLAs (greater than 99.999%, 99.99%, etc.).
- Ability to specify storage durability requirements.
- Offers data durability guarantee.
- Persistent storage for virtual servers.

Technology refers to the functional requirements of cloud services and is presented to the cloud service consumers as different features. The sub-parameters of the technology are listed below:

- Platform Compatibility.
- Open industry standards (hardware and software).
- Commitment to backward compatibility support.
- Support for open development standards.
- Compute
    - Subscriber can provide the OS and install using standard tools.
    - Ability to request a specific type of CPU.
- Storage
    - Support for object-based storage service.
    - Support for block-based storage services.
    - Support for files-based storage services.
    - Support for database-as-a-service.
- Networking
    - Ability to provide SLA for interconnectivity, bandwidth and latency (zone, across zones, to internet).
    - Secure, dedicated fibre transit between provider datacentres.
    - Ability to control IO for network traffic, Fibre Channel traffic, and converged fabric traffic.
    - Capable of independent IO control for both input and output traffic.
    - Load balancing available.

Business gives information about the cloud service providers such as revenue, data centres locations, etc. Listed below are the sub-parameters for Business:

- Services business has existed for more than three years.
- Provider has annual company revenue greater than $10M.
- Provider has annual company revenue greater than $100M.

- Cloud services revenue growth more than 10% year-over-year.
- Provider has global presence.
- Provider has a local datacentre in Asia.
- Provider has a local datacentre in Australia / New Zealand.
- Provider has a local datacentre in Canada.
- Provider has a local datacentre in Europe.
- Provider has a local datacentre in Middle East and Africa.
- Provider has a local datacentre in South America.
- Provider has a local datacentre in United States.

Although the explanations for all attributes are fairly covered, consumers are presented with an overwhelmingly large number of features and options. In the selection process consumers need to review all of these options in order to narrow down the cloud provider candidates. In addition, related to explanations to the QoS attributes, some of the explanations offered are not related to the attribute itself and are incorrect.

3.2.2 RankCloudz

RankCloudz provides cloud service consumers with cloud service selection of five different cloud services. Cloud service consumers need to sign up or sign in to be allowed the access to all of the service categories offered by RankCloudz. The diagram below depicts the attributes underneath each of the five RankCloudz cloud services.

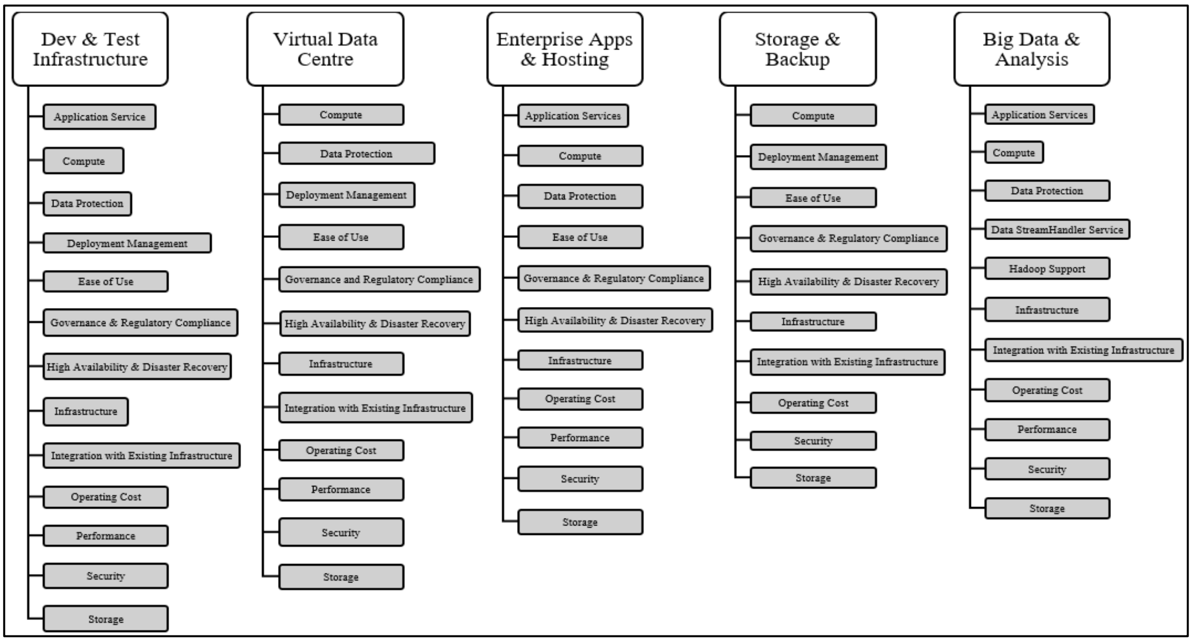


Figure 3.3 RankCloudz QoS Attributes

Dev & Test Infrastructure

- Application Services
- Compute
- Data Protection
- Deployment Management
- Ease of Use
- Governance and Regulatory Compliance
- High Availability & Disaster Recovery
- Infrastructure
- Integration with Existing Infrastructure
- Operating Cost
- Performance
- Security
- Storage

Virtual Data Centre

- Compute
- Data Protection
- Deployment Management
- Ease of Use
- Governance and Regulatory Compliance
- High Availability & Disaster Recovery
- Infrastructure
- Integration with Existing Infrastructure
- Operating Cost
- Performance
- Security
- Storage

Enterprise Apps & Hosting

- Application Services
- Compute
- Data Protection
- Ease of Use
- Governance and Regulatory Compliance
- High Availability & Disaster Recovery
- Infrastructure
- Operating Cost
- Performance
- Security
- Storage

Storage & Backup

- Compute
- Data Protection
- Deployment Management
- Ease of Use
- Governance and Regulatory Compliance
- High Availability & Disaster Recovery
- Infrastructure
- Integration with Existing Infrastructure
- Operating Cost
- Security
- Storage

Big Data & Analytics

- Application Services
- Compute
- Data Protection
- Data Stream Handler Service
- Hadoop Support
- Infrastructure
- Integration with Existing Infrastructure
- Operating Cost
- Performance
- Security
- Storage

Though the process of assigning score that reflects the level of importance is quite straightforward, RankCloudz does not provide explanation for each of these attributes straight away. Novice cloud service consumer might find the process difficult because of the lack of description of the attributes. To gain access to explanation for QoS attributes, cloud service consumer needs to sign up, sign in, and then download the in-depth report in the form of PDF file to access the QoS attributes descriptions. This also means that QoS explanation is only available after the selection process is done, and not before when cloud service consumer needs it the most.

3.2.3 Cloudorado

Cloudorado provides service selection for three types of cloud services: cloud server, cloud hosting and cloud storage. For each of the cloud service Cloudorado presents cloud service consumer with a lot of detailed attributes, but it also provides brief description for all of these attributes. The selection and QoS attributes hierarchy is represented in Figure 3.4.

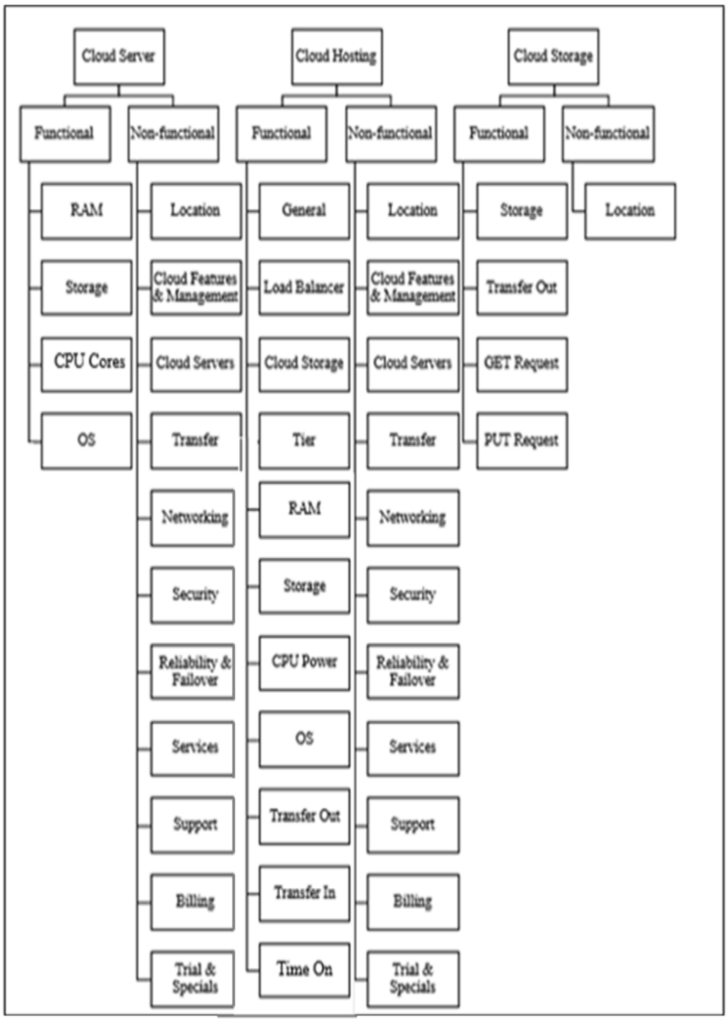


Figure 3.4 Cloudorado QoS Attributes

3.3 Comparison of QoS Attributes in Existing Service Selection Tools

The comparison of QoS attributes in existing tools is done with respect to the cloud service selection. In the process of cloud service selection QoS attributes play an important part as the defining characteristics that help cloud service consumers match their own requirement with that of cloud services. Cloud service consumer is an important entity in this analysis because the end goal is to provide QoS attributes that are structured in such a way that make the cloud service selection process easier for cloud service consumers, both novice consumers and expert consumers. With this in mind, the following criteria are used to analyse and compare QoS attributes from the three existing tools as described above.

- **Structure**: When broken down to the lowest, most-detailed level, the high number of cloud QoS attributes can be overwhelming. Because of the large number of attributes, it is important to organize them in such a way that is easy to use and understand by cloud service consumers. Structure is an important factor because structure provides a frame to organize these many attributes. Structure represents the ability to place and arrange attributes into something that is less complex, or simple enough to understand. In this case, two types are considered such as hierarchical and non-hierarchical structures. All of the previous studies related to QoS attributes attempt to create some sort of hierarchy for QoS attributes.

- **Number of Levels**: Related to the previous criteria, the number of levels indicates how many levels there are in the QoS attributes hierarchy. Number of levels in a QoS attributes hierarchy can indicate how straightforward a cloud service selection process can be. Having more levels may result in complex service selection process which may not be easy to use by novice consumers.
- **Flexibility in Hierarchy**: The flexibility in hierarchy indicates the ability for the hierarchy structure to present the QoS attributes to the cloud service consumer in different depth. This criteria is considered with the idea that cloud service selection consumers have different level of knowledge in cloud computing. With flexibility, the same hierarchy can be presented to the cloud service consumer in different depths. For instance, one consumer might choose to see the QoS attributes presented in a simpler, two-level hierarchy, while another consumer who is more expert might choose to see the QoS attributes presented in a more detailed, three-level hierarchy. With the ability to present the same structure in different depths, a cloud selection tool will be able to cater to cloud service consumer from all levels of cloud computing knowledge.
- **Type of User Input**: Cloud selection process is done by getting input from the consumers where consumers essentially tell the system what kind of requirements they want from the prospective cloud service providers. Different selection tools have different ways to get input from the consumer.
- **Description of QoS Attributes**: Description for each QoS attribute is needed to provide the consumer brief explanation of the attributes. In addition, QoS attribute description can also avoid confusion or misunderstanding over the meaning of certain attributes. The availability of QoS attributes description will help non-technical and novice consumer.

Table 3.1 provides a comparison of the QoS attributes of Intel Cloud Finder (Quick Search and Detailed Search), RankCloudz and Cloudorado.

Table 3.1 Summary of Comparison of Existing Commercial Tools

Criteria	Intel Cloud Finder		RankCloudz	Cloudorado
	Quick	*Detailed*		
Structure	Non-hierarchy	Hierarchy	Non-hierarchy	Hierarchy
No. of Level	1	3	1	2
Flexibility in Hierarchy	None	None	None	None
Type of User Input	Option selection	Level of importance	Level of importance	Option selection
Description of QoS Attributes	Partial	Available	None	Available

Intel Quick Search presents cloud service consumers with five categories of attributes. Each of them has options which consumers can directly choose. There is only one level of QoS attributes categorization. This also means there is no option to go into a more detailed level. The process is very simple and straightforward, but the number of attributes involved is limited. Intel Quick Search provides attributes definition only partially and not all attributes presented have explanation.

Intel Detailed Search provides a more detailed type of cloud selection process. In it, attributes are represented in a 3-level hierarchy. However, there is no flexibility in the hierarchy in that consumer cannot choose to give input for top-level attributes only. Underneath each of level 2 attributes are options for which consumer needs to determine the level of importance: essential, desirable, future, N/A. While there are a large number of options presented to the consumer, Intel Detailed Search provides brief description for all attributes.

RankCloudz lumps technical and business requirements together in one non-hierarchical structure. Because of the flat structure of the attributes, there is no flexibility in hierarchy. The number of attributes involved is fairly manageable where consumer only needs to give input for 11 to 14 attributes. For each of these attributes consumer only needs to determine the importance level which ranges from 0 to 10. Despite the seemingly straightforward process, RankCloudz doesn’t provide description for any of the attributes.

Cloudorado presents attributes in a subtle hierarchy structure. Subtle hierarchy means that the hierarchy is not explicitly written but it is apparent in the user interface where Cloudorado displays functional attributes as one group and the non-functional attributes as another group. For the non-functional attributes group, Cloudorado divides these into a more detailed category resulting in a 2-level hierarchy for the non-functional attributes. Cloud service consumer does not provide any flexibility to choose any input level in the hierarchy. There is quite a large number of attributes which are presented to the consumer. However, consumer does not have to select these options as Cloudorado has set a default value for some of the options. Cloudorado also provides brief description for all QoS attributes.

3.4 The Proposed Model for QoS

The contributions of the proposed model are twofold. First, it thoroughly represents the QoS attributes in a way that help cloud service consumers match their own requirements with that of cloud services. Second, the QoS attributes can be easily fed into the proposed multi-criteria decision making technique (MCDM) which is used to rank different QoS attributes of multiple alternatives in order to decide which services are most suitable for cloud service consumers.

The proposed model follows the phases below in order to structure and represent the QoS attributes of cloud services (Eisa, Younas and Basu, 2018).

3.4.1 Collection Phase

This phase is to collect relevant QoS attributes of cloud services. In the proposed model, QoS attributes are gathered by investigating different type of sources, namely, existing cloud service selection tools, cloud service providers, previous researches related to QoS attributes, and existing standardized service QoS.

Existing cloud service selection tools are included because one of the main objectives of proposed model is to improve existing selection tool by simplifying the selection process. This objective is obtained by referencing several existing selection tools and building upon what currently exists in order to create a better selection framework. For this reason, QoS attributes in existing selection tools are reviewed and included in this QoS attributes classification process.

The second source is cloud service providers. The proposed model looks into several cloud service providers and gathers the QoS attributes from the information published on providers' websites. Therefore, QoS attributes publicly listed on cloud providers' website are included in this process.

The third source is previously published research work. The proposed model takes into account some of the QoS attributes which are commonly used in the research work related to cloud service QoS.

The final source is existing standardized service QoS. The current standardized framework on service attribute is Service Measurement Index (SMI) which is developed by the Cloud Service Measurement Initiative Consortium (The Cloud Service Measurement Initiative Consortium (CSMIC), 2011). Although the proposed model does not fully adopt the SMI framework, QoS attributes are considered and included in the QoS classification process.

3.4.2 Categorization Phase

As described above, cloud service QoS include many attributes that are related to the characteristics and the delivery of cloud service. QoS attributes can be related to the technical aspects of cloud services as well as business aspects of cloud services. Because of the seemingly endless cloud QoS attributes, categorizing them in a structured way is important as it will help in sorting out these attributes. In order to cater for technical as well as business aspects of cloud services, the proposed model represents QoS attributes into four categories: Technical, Strategic & Organizational, Economic, and Political & Legislative. This categorization scheme is adopted from Baldwin *et al*.(Baldwin, Irani and Love, 2001) and also Janssen *et al*. (Janssen and Joha, 2011). However, the focus of the work presented in these papers is different than the proposed model.

For instance, Baldwin *et al*., look at the prospect of outsourcing banking information system while in the case of Janssen *et al*. they look at the prospect of adopting cloud computing technology, specifically SaaS technology. But the similarity of the proposed model to the models presented in (Janssen and Joha, 2011) and (Baldwin, Irani and Love, 2001) is the 'technology adoption by an organization'. The main objective of cloud service selection is to help businesses and organizations in selecting cloud services and thus adopting cloud technology. The proposed model therefore adopted the aforementioned four-category categorization scheme.

Further, it is paramount to include business-related consideration into the service selection process. Cloud computing technology consists of not only technical aspect, but also business aspects. In the proposed model, it is considered as important that QoS attributes should cover both technical and non-technical.

Furthermore, from the review of several sources regarding cloud QoS attributes discussed in the previous Collection Phase above, it is concluded that different providers, different tools, different studies or researches use different ways to structure or categorize QoS attributes. Selecting a categorization scheme, that covers a broader and more general scope, categorizes cloud service QoS attributes into four top categories(Baldwin, Irani and Love, 2001; Janssen and Joha, 2011) : technical, strategic & organizational, economic, and political & legislative. Each of these four categories will be represented by at least one attribute in the proposed model.

This is to make sure the attributes in the proposed model cover all aspects of cloud services. Each of the four categories is briefly described as follows (Polyviou, Pouloudi and Rizou, 2014):

- Technical – factors related to the capabilities and limitations of the technology.
- Strategic & Organizational – factors related to an enterprise's organizational and strategic goal.
- Economic: factors related to the financial aspects of cloud services.
- Political & Legislative: compliance with standards.

In the proposed model, the distribution of the first-level QoS attributes according to the four categories can be seen below in the Figure 3.5.

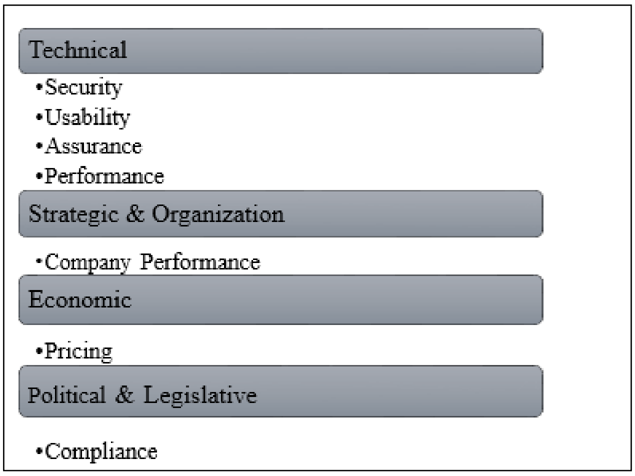


Figure 3.5 Categorization of First Level Attributes

3.4.3 Structuring Phase

This phase represents the QoS attributes in a hierarchical way because of two reasons. The first one is because hierarchy structure adheres to the AHP (Analytic Hierarchy Process) methodology that is commonly used in the cloud service selection framework. AHP methodology breaks down the problem in several levels and models the problem in a hierarchical structure. The second reason is that hierarchical structure provides flexibility to consumers. This is especially tied in with the goal of the proposed model that takes into account user knowledge of cloud computing. By having hierarchical structure, it is easy to place a more general attributes at the top level and the more detailed attributes at the lower levels. And in so doing, the proposed model has the ability to present novice consumers with top-level, general QoS attributes as well as to present expert consumers with the more detailed attributes situated at the lower level. This kind of flexibility is not provided in the three existing selection tools reviewed in the previous section.

In the proposed model there are three level hierarchies with seven top-level attributes: security, usability, assurance, performance, company performance, pricing, compliance. The 2-level structure of the QoS attributes in the proposed model is graphically represented in Figure 3.6, while the complete attributes can be found below in Figures 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, and 3.13.

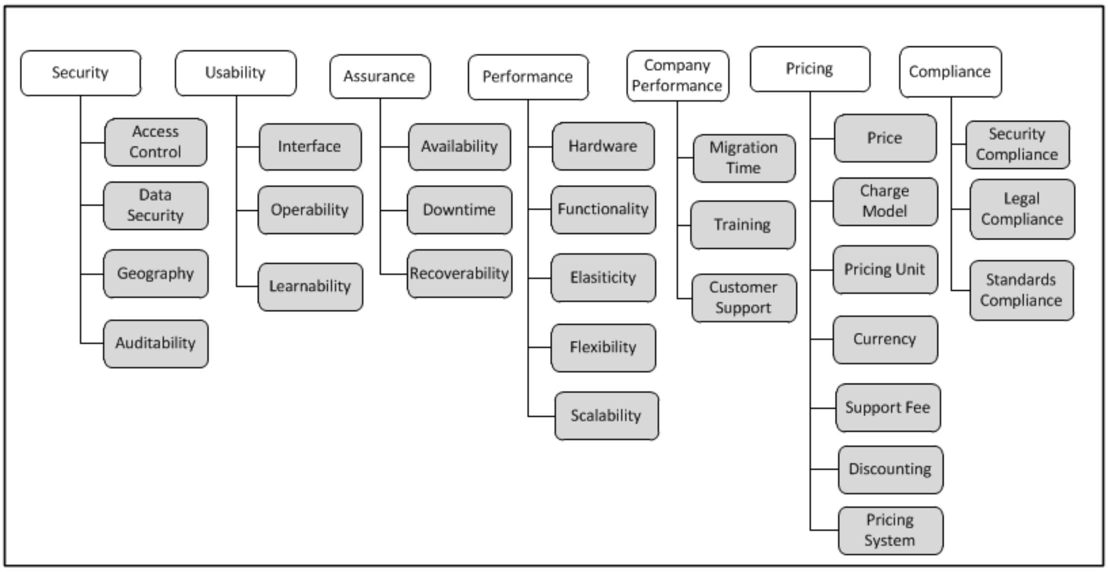


Figure 3.6 QoS Attributes Hierarchical Structure

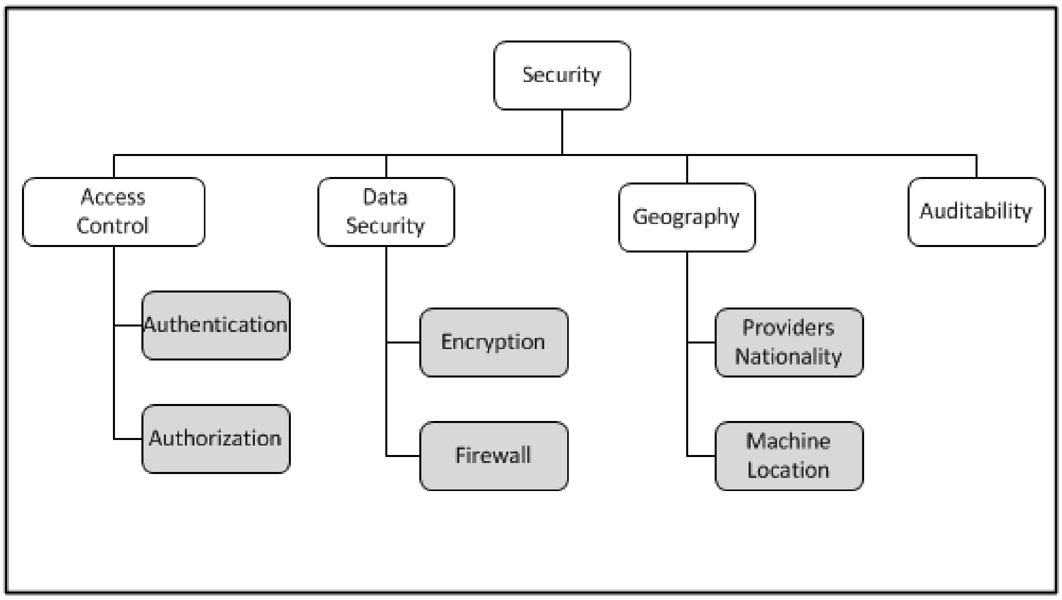


Figure 3.7 QoS Attributes for Security

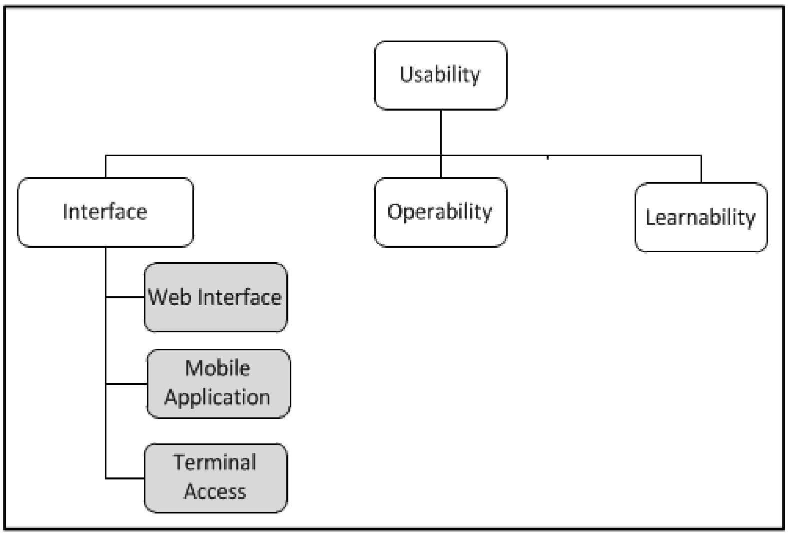


Figure 3.8 QoS Attributes for Usability

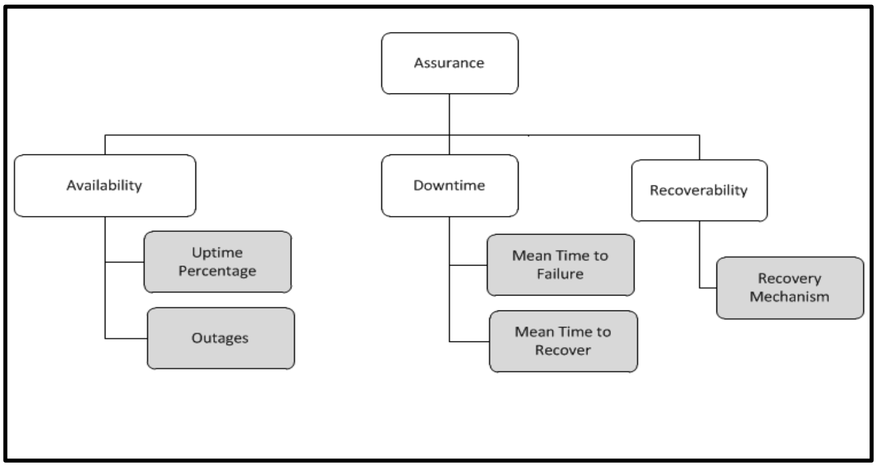


Figure 3.9 QoS Attributes for Assurance

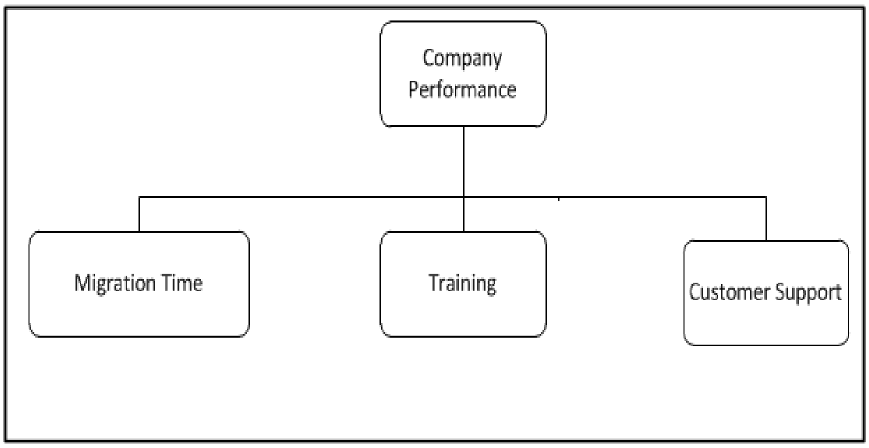


Figure 3.10 QoS Attributes for Company Performance

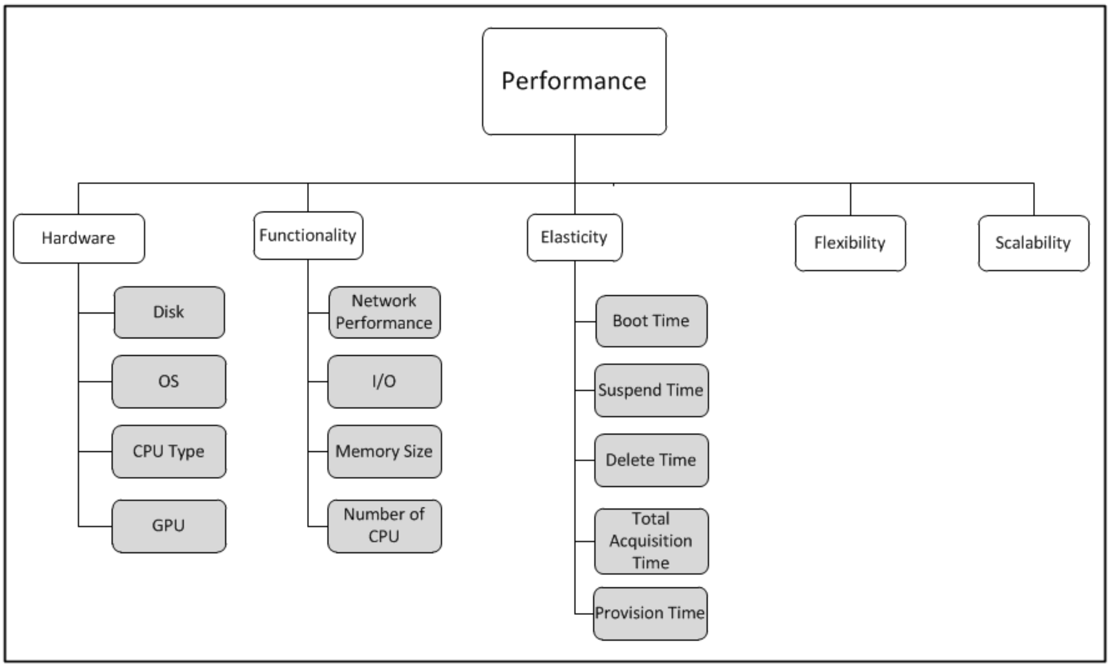


Figure 3.11 QoS Attributes for Performance

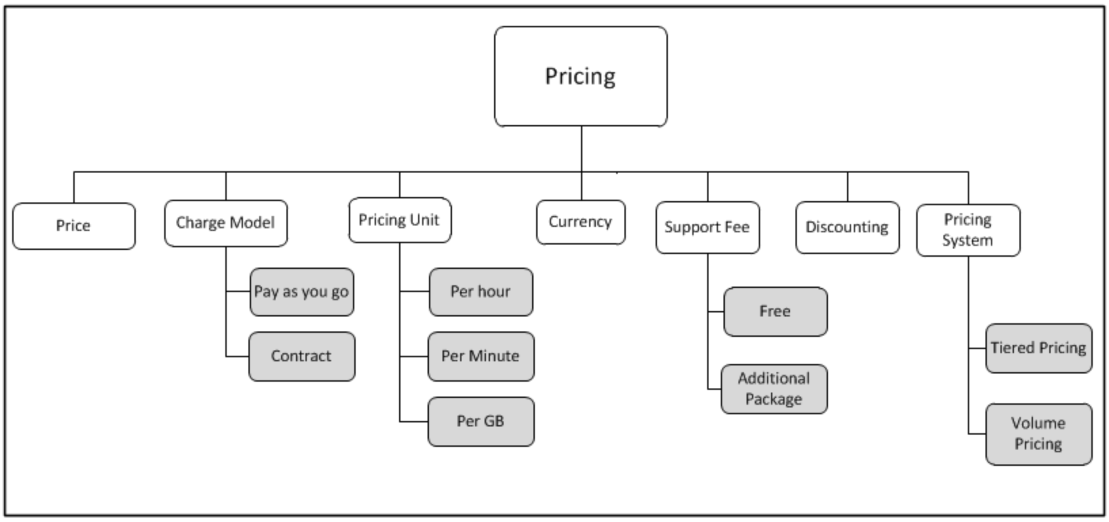


Figure 3.12 QoS Attributes for Pricing

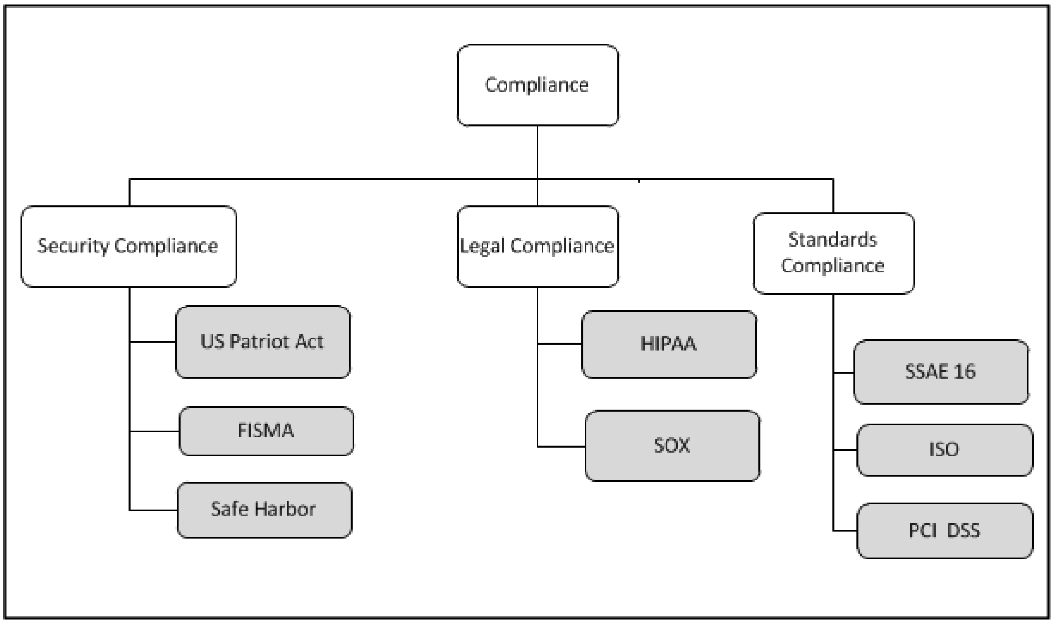


Figure 3.13 QoS Attributes for Compliance

Now that the QoS attributes in the proposed model have been structured, the comparison between the three existing tools and the proposed model is detailed in the Table 3.2.

Table 1.2 The Comparison of the Proposed Model with the Existing Commercial Tools

Criteria	Intel Cloud Finder – Quick Search	Intel Cloud Finder – Detailed Search	RankCloudz	Cloudorado	Proposed Model
Structure	Non-hierarchy	Hierarchy	Non-hierarchy	Hierarchy	Hierarchy
# of Level	1	3	1	2	3
Flexibility in hierarchy	None	None	None	None	Yes
User Input	Option selection	Importance	Importance	Option selection	Importance and option selection
Description	Partial	Available	None	Available	Available

3.5 Summary

The proposed model for representing QoS attributes combines the best value for each criterion. In it, QoS attributes are placed in a hierarchical structure in order to make it easier for cloud service consumer to see a clear categorization of the myriads of attributes. There is flexibility in the hierarchy where in the cloud service selection tool; consumers have the choice to see the QoS attributes in different depth, for instance, the choice to see a 2-level hierarchy or a 3-level hierarchy. The type of input will be different depending on the hierarchy level that consumer chooses. At the top level consumer needs to determine the importance of QoS attributes, while at the lowest level in the hierarchy consumer needs to select from a set of options. Lastly, brief description for all attributes will be provided to help consumers develop further understanding.

Further details on the implementation of the proposed model of QoS attributes are provided in the subsequent chapters of this thesis.

CHAPTER 4

Cloud Service Selection: Theoretical Framework and Methodology

4.1 Introduction

This chapter presents the proposed framework and methodology of cloud service selection. It illustrates the proposed framework at a higher level of abstraction through a generalized architecture and process flow. The chapter also presents an integrated methodology which is used in the proposed service selection process in order to rank cloud services based on the QoS attributes presented in Chapter 3. The methodology combines three different techniques including Analytical Hierarchy Process (AHP), Simple additive weight (SAW), and Skyline Operator. A detailed design and implementation of the proposed framework and methodology are presented in Chapters 5 and 6.

4.2 The Proposed Framework

The proposed framework is illustrated through the architecture shown in Figure 4.1. There are several components that build the overall framework together. The centre component of the framework is a central repository that stores all data related to cloud service providers. Data related to cloud service providers are gathered from three different sources which include, providers' themselves, user reviews, and third party monitoring tools. This is one of the major contributions of the proposed framework as it does not rely on a single data source which is the case in existing approaches. Relying on a single data source may not provide credible results in service selection. For instance, service related QoS data presented on provider website could be biased or it may not reflect the QoS which is experienced by the consumers. However, collecting and analysing data from three different sources complicates the overall process of service selection. But the proposed framework takes into account three different sources in order to provide more credible results in the service selection process. Another input to the framework is the requirements from consumers, presented in the form of cloud service (QoS) attributes, which are presented in Chapter 3.

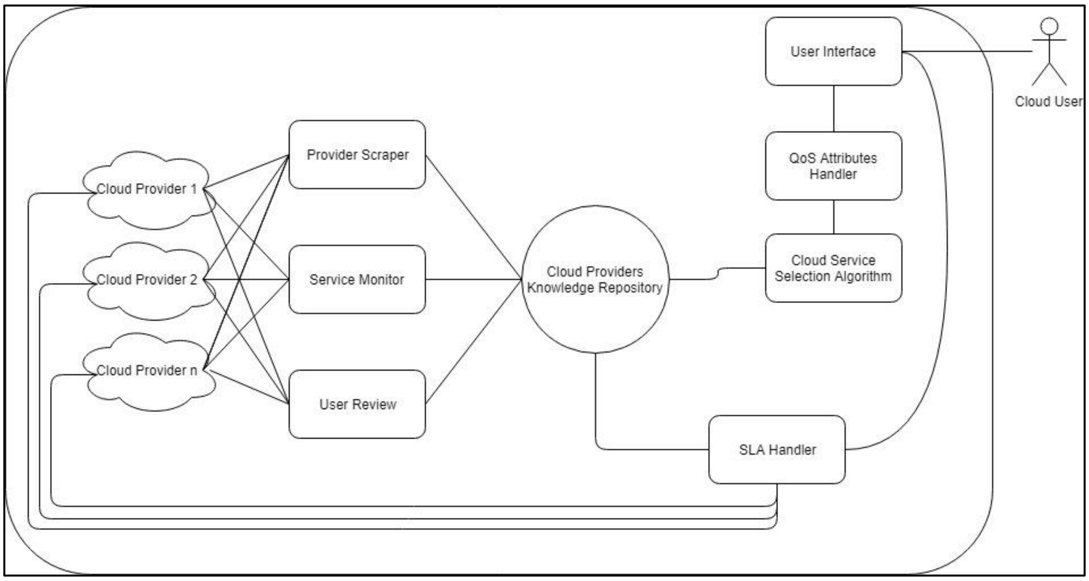


Figure 4.1 Proposed Framework

4.3 Components of the Framework

The main components of the framework are explained as follows.

1. **Cloud Providers Knowledge Repository**

   Cloud Providers Repository is a database that is a central part of the framework. This database stores any data related to cloud service providers that will be used in the selection technique.

```
        "customerSupport" : "yes"
},
"instanceName" : "n1-highmem-4",
"vendorName" : "Google Cloud",
"assurance" : {
        "recoverability" : {
                "recoveryMechanism" : "automatic"
        },
        "availability" : {
                "uptime" : "100",
                "outages" : "0"
        },
        "downtime" : {
                "mttr" : 0,
                "mttf" : 720
        }
},
"usability" : {
        "interface" : {
                "mobile" : "no",
                "web" : "no",
                "terminal" : "yes"
        },
        "learnability" : "yes",
        "operability" : "yes"
}
```


Figure 4.2 Data Record in Repository

   Data stored in this repository includes data gathered from outside source (scraping providers' websites, user reviews, service monitoring tool) as shown in Figure 4.2. It also contains data resulted from internal processing or calculation, namely cloud providers score. These cloud providers score are calculated internally after taking into account all relevant data from the outside sources. In Figure 4.2 above, the data is represented in JSON format as key value pairs. This is the language format of MongoDB. For instance the key "instanceName" attribute shows the name or unique identifier of the particular instance, and holds the value in this case "n1-highmem-4" and the "vendorName" attribute shows the name of the cloud provider i.e. Google Cloud. The other attributes or key-value pairs also show the value for each of the QoS attributes measured.

2. **Provider Scraper**

   This module's main objective is to go through cloud providers' websites continuously on a certain time interval to gather relevant data about certain pre-defined QoS attributes that are included in the framework as mentioned in Section 3.4.3. The needed data are then stored inside the repository which is described above. This module will continuously scrape for data in order to make sure that the provider's repository always has the latest and updated data.

3. **User Review**

   Similar to the previous provider scraper module, this module's objective is to gather user review data as well as it is also responsible to process user review data. User review is an important data source because it generally provides fair assessment from real consumers regarding service from cloud service providers as well as their experience with certain cloud service providers. Score from user review gathered by this module is then stored inside the repository.

4. **Service Monitor**

   Service monitor module is important to provide objectivity in this framework. This module is responsible to select third party service monitoring tools in order to gather cloud service provider's performance data. The gathered data is then stored inside the repository. For example, cloudharmony[1] monitoring tool (third party) monitors service status, service availability percentage and outages.

5. **User Interface**

   User interface module is the interface that deals with the cloud service consumers. It is responsible for all functions that are related to usability, user interface, and most importantly, user experience. One of the main goals of the framework is to create a simple cloud service selection tool and process that cater for different types of consumers, including consumers with minimum knowledge of cloud computing technologies.

---

[1] https://cloudharmony.com/

6. **QoS Attributes Handler**

   QoS Attributes Handler module is enabled cloud service consumers for specifying the required Quality of Service attributes based on their needs. In addition, this module is responsible for concise explanation, categorization and structure of QoS attributes. The more details about this module are illustrated in Chapter 3. Furthermore, QoS Attributes Handler module controls the processing of QoS attributes that are selected, compared and weighted by the consumers.

7. **Cloud Service Selection Algorithm**

   This module is the brain of the framework as it is responsible for the implementation of the integrated methodology which is explained in the subsequent sections of this chapter. This module takes provider's score from the repository and processes these scores numbers together with the QoS attributes requirements from the consumers using AHP and SAW methodologies, and in the end comes up with the final ranking of best providers that closely match consumer's requirements.

8. **SLA Handler**

   SLA Handler handles Service Level Agreement-related functionalities. This module gathers SLAs provided by cloud providers, and reinterprets these SLAs in a simple new SLA universal template, in a way that consumers will get the gist and main important points of the SLAs.

4.4 The Proposed Process Flow of Cloud Service Selection

There are several scenarios of process flow that can happen in the framework. Cloud service consumers with minimum cloud computing knowledge will have different process flow compared to the process flow for more expert consumers with deeper knowledge of cloud computing. The overall process flow diagram is shown in Figure 4.3. The implementation of the different steps of process flow is described in Chapter 6.

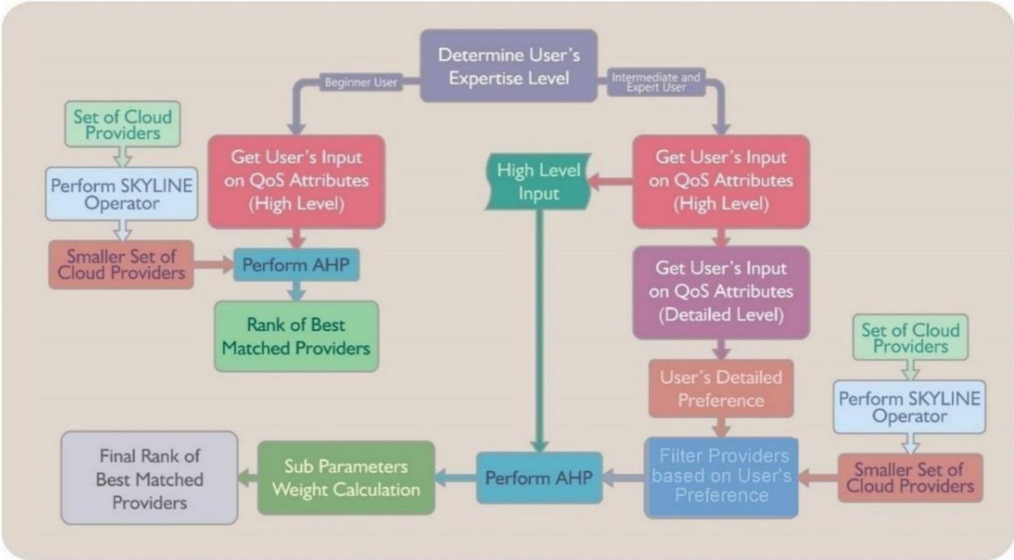


Figure 4.3 Process Flow of Cloud Service Selection

1. **Determine User's Expertise Level**

   Before the calculation starts, it is necessary to determine how familiar a consumer is in cloud computing because there are two different paths of processes that depend on consumer's level of knowledge. If the cloud service consumers consider themselves as Beginners, the process flow of cloud service selection is a little bit shorter for them than the process flow for Intermediate/Expert cloud service consumers. The only difference is that the beginner consumer system needs to perform AHP with one level of attributes to produce final rank of cloud service provider. System will only ask consumer to do pairwise comparison on this most-general level of attribute, given consumer's limited knowledge. On the other hand, for Intermediate and Expert cloud consumer, system will ask consumer to do pairwise comparison in all of higher levels of QoS attributes, and at the lowest, most-detailed level, system will give an option for consumer to specify the details of cloud service's QoS attributes. For example, expert cloud consumer will be able to specify the exact price of cloud service, the location of cloud servers, or the exact type of security certificate that consumer wants.

2. **Get User's Input on QoS Attributes (High Level)**

   System will ask cloud consumer to do pairwise comparison on QoS attributes in the first two levels except the lowest level. As discussed before, the problem of cloud service selection is broken down into several levels in a hierarchical structure. The output of this step is High Level Input, which is basically consumer's preference on each selected QoS attributes.

3. **Get User's Input on QoS Attributes (Detailed Level)**

   System will ask consumer to specify the details of QoS attributes that the consumer wants. The output of this step is User's Detailed Preference. For example, consumer will specify the exact price range they want, or the kind of security certificate they need.

4. **Perform Skyline Operator**

   Skyline operator is applied to the collection of cloud service providers stored in cloud repository. Because the nature of skyline operator is to filter only the better candidates, the result of applying skyline operator is a smaller collection of cloud service providers. Skyline operator will leave out cloud providers that are dominated by any other providers in the database.

5. **Perform AHP**

   After consumer does pairwise comparison on the top first two levels of the QoS attributes, calculation is performed. Referring to the hierarchical structure of AHP problem, in this case AHP is performed on the first two levels .The lowest level of the hierarchy (sub-parameters) will be processed using simple additive weight technique.

6. **Filter Providers Based on User's Preference**

   Referring back to consumer's input on QoS attributes on the detailed level, the output of that step is then used to filter the providers. Given that previously Skyline operator has been performed on the database and the result of this is a smaller set of cloud providers, the filtering process in this step will result in an even smaller collection of cloud providers. This step is straightforward filtering where system will only include providers that match the QoS specification from consumer.

7. **Sub-parameters Weight Calculation**

   Sub-parameters weight calculation is application of simple additive weighting method on the lowest level of the hierarchy. After performing AHP at the higher levels, the system will then use simple additive weighting to calculate the utility of each remaining provider using the weights that have been calculated in the previous AHP calculation. After the utility of each provider is calculated, system will then be able to rank cloud providers based on the utility numbers.

4.5 The Proposed Integrated Methodology

The section describes the proposed methodology which is used in the proposed service selection process. The methodology integrates three different techniques which include, Analytical

Hierarchy Process (AHP), Simple additive weight (SAW), and Skyline Operator technique. The rationale behind using the three different techniques is given below.

4.5.1 Analytical Hierarchy Process (AHP)

AHP is selected to solve the cloud selection problem in this framework because firstly, the problem of cloud selection can be defined as multi criteria decision making problem where there is a decision to make with multiple conflicting criteria (Whaiduzzaman *et al.*, 2014). Second, AHP technique is specifically selected because it is well defined and structured MCDM methods where the problem is defined in a structured way within a hierarchy. The hierarchical structure of AHP helps in defining the best structure to present the QoS attributes, as discussed previously. Hierarchical structure is excellent to represent QoS attributes in a flexible way, especially related to the ability of the framework to adjust to different levels of knowledge possessed by different consumers. When the consumer needs only general attributes, they only need to access the top level attributes in the hierarchy. When the scenario needs to go deeper into the more-detailed attributes, the framework can access the second and third level attributes in the hierarchy.

Third, AHP technique enables the framework to get input from consumer only in the form of consumer's preference of a certain attribute over another attribute (Godse and Mulik, 2009). Without having to delve into a more technical attributes of cloud service, AHP technique allows consumer to just input their preference, whether they prefer attribute A more than attribute B, whether they prefer attribute B more than attribute C, and so on. This simple input is helpful for consumer with limited knowledge of cloud computing. When a more expert consumer is involved and wants to access the more technical attributes, then the hierarchical structure of AHP will also allow this.

The steps involved in the AHP methodology are described below:

Step 1: Set up attributes in pair and ask consumer to compare the two attributes relative to each other. The scale used in pairwise comparison is as follows: 1 (equal importance), 3 (slightly more important), 5 (strongly favour one over the other), 7 (very strongly favour one over the other), and 9 (extremely favour one over the other).

Step 2: Construct pairwise comparison matrix. The resulting score of the pairwise comparison between attributes is then represented through a reciprocal matrix, as given below.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad (1)$$

In this matrix, a_{ii}= 1 and $a_{ij} = \frac{1}{a_{ji}} \mathrm{a_{ij}} = \frac{1}{\mathrm{a_{ij}}}$

Step 3: Construct normalized comparison matrix. Before constructing the normalized matrix, first sum the value for each column in the pairwise matrix. Given that *n* is the column index of the matrix, sum for each column is:

$$S_n = a_{1n} + a_{2n} + \cdots + a_{nn} \quad (2)$$

Normalized matrix is then obtained by dividing each value with the sum of its column.

$$\acute{A} = \begin{bmatrix} \frac{a_{11}}{S_1} & \frac{a_{12}}{S_2} & \cdots & \frac{a_{1n}}{S_n} \\ \frac{a_{21}}{S_1} & \frac{a_{22}}{S_2} & \cdots & \frac{a_{2n}}{S_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{S_1} & \frac{a_{n2}}{S_2} & \cdots & \frac{a_{nn}}{S_n} \end{bmatrix} \quad (3)$$

Step 4: Obtain Eigen vector (priority vector) and Eigen value. Eigen vector *w* can be obtained by averaging across the rows in the matrix.

$$W = \frac{1}{n} \begin{bmatrix} \frac{a_{11}}{S_1} + \frac{a_{12}}{S_2} + \cdots + \frac{a_{1n}}{S_n} \\ \frac{a_{21}}{S_1} + \frac{a_{22}}{S_2} + \cdots + \frac{a_{2n}}{S_n} \\ \vdots \\ \frac{a_{n1}}{S_1} + \frac{a_{n2}}{S_2} + \cdots + \frac{a_{nn}}{S_n} \end{bmatrix} \quad (4)$$

Eigen value *λmax* is obtained from the summation of products between each element of Eigen vector and the sum of columns of the matrix.

$$r_n = \frac{\frac{a_{11}}{S_1} + \frac{a_{12}}{S_2} + \cdots + \frac{a_{1n}}{S_n}}{n} \quad (5)$$

$$\lambda_{max} = \sum_{i=1}^{n} S_i * r_i \quad (6)$$

Step 5: Check comparison consistency through Consistency Index (CI) and Consistency Ratio (CR)

$$CI = \frac{\lambda_{max} - n}{n-1} \quad (7)$$

$$CR = \frac{CI}{RI} \quad (8)$$

For reliable result, CR value should be less than or equal to 0.1

RI is Random Consistency Index where the number is different depending on how many attributes are involved. Values for RI are shown in Table 4.1 below (Hwang and Yoon, 1981).

Table 2.1 Random Consistency Index in AHP

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

```
Input: A - list of parameters
Init: n - number of parameters
//obtain result for pairwise comparison
Init: PCScore - list of pairwise comparison score
For i=0 to n do
    For j=0 to n do
Read user score for i,j comparison
Input score to PCScore
    End for
End for
//create pairwise matrix
Init: M - matrix for pairwise comparison result
Set K to 0
For i=0 to n do
    For j=0 to n do
        If (i==j)do
            M[i][j]=1
Else if (i<j) do
 M[i][j]=PCScore[k]
    k++
Else if (i>j) do
    M[i] [j]=1/m[i] [j]
End if
    End for
End for
//obtain Eigen vector and Eigen value
Init: sum - list of sum of each column in M
//getting sum for each column
For i=0 to n do
    For j=0 to n do
        sum[i] = sum[i]+M[j][i]
 End for
End for
//build normalized matrix
Init: Mnorm - matrix for normalized M
For i=0 to n do
    For j=0 to n do
        Mnorm[i][j] = M[i][j]/sum[j]
    End for
End for

//build priority vector
Init: PVector - list for Eigen vector
Init: sumRow = list of sum value of each row in Mnorm
For i=0 to n do
    For j=0 to n do
        sumRow[i] = sumRow[i] + Mnorm[i][j]
    End for
    PVector[i] = sumRow[i]/n
End for
//calculate Eigen value
Init: LambdaMax - Eigen value
For i=0 to n do
    LambdaMax = LambdaMax + (sum[i]*PVector[i])
End for
//Calculate C.I. and C.R.
```


```
//build priority vector
Init: PVector - list for Eigen vector
Init: sumRow = list of sum value of each row in Mnorm
For i=0 to n do
    For j=0 to n do
        sumRow[i] = sumRow[i] + Mnorm[i][j]
    End for
    PVector[i] = sumRow[i]/n
End for
//calculate Eigen value
Init: LambdaMax - Eigen value
For i=0 to n do
    LambdaMax = LambdaMax + (sum[i]*PVector[i])
End for
//Calculate C.I. and C.R.
```


```
Input: RI - list of values of Random Consistency Index
Init: C.I. - value for Consistency Index
Init: C.R. - value for Consistency Ratio Index
CI = (LambdaMax - n)/(n-1)
CR = CI / RI
```

Figure 4.4 AHP Algorithm Implementation

Figure 4.4 shows an algorithm of the AHP process which is implemented in the proposed framework of cloud service selection.

4.5.2 Simple Additive Weight

In the propose framework, Simple Additive Weight (SAW) method is used in conjunction with the AHP due to the following reasons. First, this methodology involves a very simple calculation process. Simple calculation helps in speeding up the processing time of the tool to be built, especially when it involves a large size of data – i.e., a different QoS attributes and different cloud services from different providers. Second, SAW calculation is centred on the weight of each attribute for each provider (Bao and Dou, 2012). The weight for each attribute is already calculated as part of the AHP calculation process (priority vectors). AHP method calculates in details the weight of each attribute and its sub attributes. This weight calculation will seamlessly flow into the SAW calculation process.

In the proposed framework of cloud service selection, first step of SAW is to calculate the utility of each QoS attribute for each provider. For each provider m, utility of the n^{th} attribute is the total sum of score S of attribute n, sub-parameter i^{th} multiply by the weight w of i^{th} sub-parameter of n^{th} attribute. j is the number of sub-parameters of attribute n.

$$U_{mn} = \sum_{i=1}^{j} = (S_{mn} * w_n)_1 + (S_{mn} * w_n)_2 + .. + (S_{mn} * w_n)_j \qquad (9)$$

$$Where\ U_{mn} = Utility\ calculation\ for\ each\ provider\ sub - attributes$$

Once the utility of each attribute n is calculated, the next step is to calculate the total utility of provider m. Total utility of provider m is the total sum of utility of each attribute.

$$U_m = \sum_{n=1}^{l} U_{mn} * W_n \qquad (10)$$

Where l is the number of attributes, W_n is the weight of main attribute n.

The algorithm implementation of the simple additive weight calculation is shown in Figure 4.5.

```
Input: P - list of providers
Input: A - matrix of level 1 parameters
and level 2 parameters
Output: providers ranked based on score
For each provider in P do
    For each level 1 param in A as i do
        For each level 2 param of Level 1 in A as j do
         utility level 2 = value of j * weight of j
        End for
    utility level 1 = utility level 1 + utility level 2
    weighted score = utility level 1 * weight of i
    End for
    total provider score = total provider score + weighted score
End for
//sort the score in P
Return ranked list of providers
```

Figure 4.5 Algorithm for Simple Additive Weight Method

4.5.3 Skyline Operator

As discussed in previous chapter, there are two different algorithms that can be utilized to implement Skyline operator. Figure 4.6 shows the algorithm of Skyline operator using the block-nested-loops technique. The use of Skyline operator helps in reducing the size of data source (Zhang *et al*., 2016). In this case, the original instances can be reduced. When dealing with much bigger dataset size, Skyline operator can be the solution to reduce the data size into a more manageable size, and therefore can reduce the processing time. Note that in this research skyline operator is not implemented, just the pseudo-code of the block-nested-loops algorithm is described as shown below.

```
Input: L - list of all data points
Output: Skyline S
//if the list L is empty, terminate algorithm
if L is empty then
    Terminates the algorithm
    Return S with no element
Insert the first point in L into S
For each point p in L do
    if dominatesInS(p,S) then
        Add p to S
        Remove all points in S that are dominated by p
    Else i=If !(dominatesInS (p,S) II isDominatedbyS(p,S)) then
        Add p to S
        //meaining that p is incomparable with all the data points in S
        //add p to S
End for
//Return the skyline points
Return S
```

Figure 4.6 Algorithm for Skyline Operator

4.6 Summary

The proposed framework for cloud service selection is described in detail. The proposed structure consists of various modules such as Cloud Providers Repository, Provider Scraper, and User Review. Each of these modules is responsible to perform different tasks. The process flow scenario of the framework is proposed based on the consumer's knowledge level of cloud technologies. According to the proposed scenario, Beginner cloud service consumers perform pairwise comparison solely for the main QoS attributes in the first level. However, Intermediate/Expert cloud service consumers have an option either just to do the pairwise comparison for the first level main attributes or they can go through second level of QoS attributes and perform pairwise comparison. In addition, Intermediate/Expert consumers can delve through the more detailed features of the QoS attributes and specify their specific requirements.

The proposed methodology is the combination of three techniques called AHP, SAW and skyline. All the steps, equations and algorithms for the proposed methodology are listed in detail. The first two levels of the proposed framework is processed using AHP method followed by SAW technique. Then, the best cloud service providers ranked and presented to the cloud service consumers.

CHAPTER 5

Design of the Cloud Service Selection Tool

5.1 Introduction

This chapter describes the design of the proposed cloud service selection tool. First, it provides an overview of the tool and the software/hardware platforms that are used in the implementation of the tool. Second, it defines certain assumptions and constraints about the design and implementation of the tool. Third, it illustrates the design phases of the tool. The design part is divided into different phases as the implementation of the tool is a complex process that involves various tasks such as user interface design, collecting data from cloud providers, mapping of QoS scores and ranking of services using AHP, SAW, and Skyline methods.

5.2 Overview of the Cloud Service Selection Tool

The cloud service selection tool is based on the proposed framework. It is named as Sahab Selection Tool (SST), where Sahab means cloud in Arabic language.

SST is developed as a proof of concept for the proposed cloud service selection framework. Figure 5.1 represents the system architecture within which the SST tool is implemented. It comprises three main building blocks or components: (i) Client side and user interface, (ii) Web server that implements the proposed algorithms/methods, and (iii) Database server for storing data. Further details on the implementation of these components are provided in the following chapter.

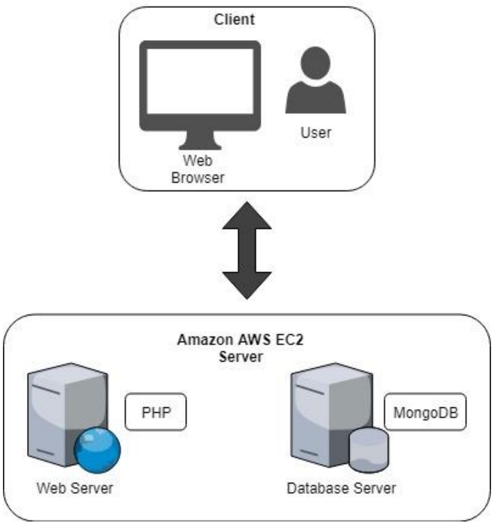


Figure 5.1 SST System Architecture

The SST is to be developed as a web-based cloud application and will be deployed in a cloud server which is hosted by Amazon AWS EC2. The virtual server instance used in the implementation of the STT tool is EC2 c5.xlarge.

The hardware environment specification is listed below:

1. Processor Intel Xeon Platinum 8124M CPU @ 3.00 GHz
2. Memory 8 GB
3. Hard disk 32 GB

The software environment specification is listed below:

1. Operating System Windows Server 2012 R2 Standard 64-bit
2. Web server XAMPP for Windows version 7.0.24. (XAMPP package is a combination of Apache webserver, MySQL and *PhPmyAdmin*)
3. Database MongoDB version 3.4.10

The rationale for the selection of these tools is explained as below.

During the development of SST, availability and reliability were key factors to complete this phase. The server needed to be accessible at any time and from any location. As such, it was necessary to develop all the work using a cloud service. Amazon AWS is the largest cloud providers in terms of Infrastructure and has the cheapest prices when compared to other cloud computing providers (Mohan *et al.*, 2012; Naldi and Mastroeni, 2013). In addition, Amazon boasts of having a very high reliability across all their services and also adds reliability as one of their pillars in their design principle (Amazon, 2018). Also, affordability was an important factor in the choice of service provider. AWS provide on-demand instances where servers can be spun up and shut down when not in use. When the server is not in use, AWS do not charge for compute power. Thus, it was an attractive option that could help save cost. Therefore, due to its affordability and reliability, AWS EC2 became the choice provider on which to do development work.

MongoDB was chosen as the preferred back end database. MongoDB is a NoSQL document database that uses JavaScript Object Notation (JSON) as its implementation language. MongoDB is widely used by a lot of real life application and government services[2]. Also, MongoDB stores data in JSON which provides flexibility needed to store semi-structured data. Most of the data items used in SST are scrapped from different sources and processed using MongoDB. This type of data would have been difficult to process in a traditional relational database due to the extra overhead of table joins and data normalization.

[2] https://www.mongodb.com/industries/government

Finally, XAMPP package was chosen as the web server and is installed on the AWS EC2 as the local web host server on cloud. XAMPP is a combination of Apache webserver, MariaDB (MySQL) database and PHP. Since PHP is a web language, using Apache as the web-server in the XAMPP package provided seamless deployment of these tools. There was no need for heavy configuring. Also, PHP and Python work together, and Python scripts are easily embedded into PHP pages. User interface is a key part of the SST i.e. consumers must find it easy to make use of this tool. Therefore, the choice of using PHP which works easily on web browsers was important.

5.3 Assumptions and Constraints

This section defines the following assumptions and constraints which are related to the design and implementation of the SST. The assumptions made by the SST are limited to the data items collected by the SST in the scrapping phase.

1. Attribute preference input from the consumer is assumed to be valid and consistent. Checking the consistency of consumer input is beyond the scope of the SST implementation. The proposed tool user interface will be designed in a way in order to minimise inconsistency when attributes are entered. But the problem of automatic checking and validity of inputting data is more related to the website design and HCI area.
2. The process of scraping cloud service data from cloud provider's websites is carried out by taking into account a limited number of eight cloud providers. The remaining data is gathered manually. The SST tool can be enhanced to scrape data from other websites. However, scrapping data from distinct websites is a complicated process as different cloud service providers represent data in different structure, format and different description or interpretation — a problem that falls in the realm of Information Searching and Retrieval research area.
3. Due to the large number of cloud services it is possible that some QoS attributes will have missing data (values). The common approaches to dealing with missing data are: (a) eliminate the missing data (b) mark the missing data values as unknown, or (c) use imputation or interpolation techniques which replace missing data with mean or median values of similar data. However, such replacement may not create accurate results. Thus, based on the QoS attributes model presented in Chapter 3, the SST tool takes into account those QoS attributes whose data (values) are determined (or available). Attributes with missing values are not considered in the implementation stage.

5.4 Design Phases of the SST Tool

The design and implementation of the SST tool is not trivial given that cloud service selection process involves various functions such as user interface design, collecting (scraping) data from different sources, QoS-based service ranking using the proposed methodology (based on AHP, SAW, Skyline) and so on. In order to systematically design and develop the STT tool this chapter defines the following design phases:

- Architecture design
- Collecting (scraping) data from cloud providers
- Levels of QoS attributes
- Mapping of QoS scores using
  - Boolean values
  - Numerical values
  - Uniform scores
- Users' reviews
- Service Level Agreement

The following sections illustrate each of the above phases.

5.4.1 Architectural Design

Based on the STT architecture presented in Figure 5.1, the following diagram in Figure 5.2 shows the main components of SST and how they interact with each other in order to implement the main functions required by the cloud service selection process.

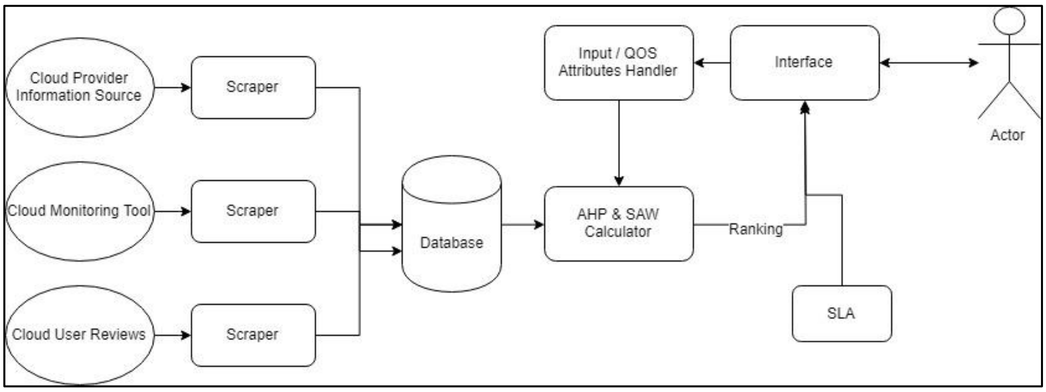


Figure 5.2 SST Tool Parts

The SST needs to be able to read input from the consumer, process consumer's input using the proposed methodologies and then present the consumer with ranking of cloud services that best match consumer's preference in the input. SST gathers data about cloud service providers along with cloud services that these providers provide. All the data gathered from different sources is stored in the main database. The consumer preference cloud service matching process will compare consumer's input with the data available in the database.

SST asks consumer for their attribute preference in the form of pairwise comparison which adheres to the process of proposed methodology that selects the best-matched cloud services for consumer. SST then processes consumer input and factors to produce the end ranking of cloud services that match the best with consumer's preference.

5.4.2 Collecting (scraping) Data from Cloud Services Providers

Cloud service data is gathered (or scraped) from eight different cloud service providers using the Scraper component of the SST tool. These eight providers were chosen based on the leading cloud providers in the industry according to Gartner Magic Quadrant for cloud computing released in 2017 (Leong *et al.*, 2017). This section gives a brief overview of the services provided by the cloud vendors which are used in the SST.

Amazon AWS

Amazon Web Services is the largest cloud computing provider in the world with respect to infrastructure. Amazons main compute service is known as EC2 i.e. Elastic Cloud Compute. Amazon also provides storage services and is popularly known for its S3 storage services. Among the companies competing in the infrastructure space, AWS has the cheapest prices (Amazon, 2018). Amazon also has a wide variety of products ranging from analytics to storage, which makes them attractive to small, medium and large scale sized organizations.

Rackspace[3]

Rackspace is cloud computing company that provide mainly infrastructure as a service. Rackspace solution allows consumers to adopt the cloud model that fits them. Consumers can run their workloads entirely on the public cloud, or in a private cloud reserved for the consumer .Also, consumers can run a hybrid cloud which is a combination of an on premise data centre and a cloud solution. However, Rackspace solutions are limited to infrastructure and platform services.

Google Cloud[4]

Google is a major cloud computing provider. Google mainly provide platform services, analytics and machine learning services. Google compute is known as Google Compute Engine. Google also provides a fully managed server-less platform known as App Engine which enables developers to deploy and run their code without explicitly spinning up or managing any server.

---

[3] https://www.rackspace.com/en-gb/cloud/private
[4] https://cloud.google.com/

Consumers are charged for compute power used in running their codes. The App Engine automatically scales depending on traffic on the application. Google is also known for providing software services such as Gmail.

Oracle Cloud[5]

Oracle is a household name in the relational database market. However, of recent Oracle are investing more in cloud services. Oracle has a range of services running on the cloud. Oracle provides SaaS, PaaS and IaaS services. Oracle provides compute services running on Virtual Machines. They also provide Oracle Enterprise manager via cloud.

Alibaba Cloud[6]

Alibaba is an Internet based service provider that provides a range of services outside of cloud computing. Alibaba also provide cloud services known as Ali Cloud. They provide virtual compute services known as Elastic Compute Service. Ali Cloud operates mainly in the IaaS sector.

Microsoft Azure[7]

Microsoft is one of the largest computer providers in the world. Their cloud service is known as Azure. Microsoft provide SaaS, PaaS and IaaS services. They have one of the widest ranges of services among all service providers as most of the Microsoft products also have their cloud versions. For instance, the Microsoft Office suite is now provided via cloud and known as Office 365. Consumers can access their office documents from anywhere via a web interface. Microsoft Azure has a wider range of products including SQL Server, HD Insight, and Virtual Machines etc.

Fujitsu Cloud[8]

Fujitsu is a cloud service provider that provide platform and infrastructure services on the cloud. Their products include virtual servers, operating systems and storage services. Fujitsu also provides Hybrid cloud services and managed infrastructure services. Furthermore, they sell enterprise and cyber security services.

[5] https://cloud.oracle.com/home
[6] https://www.alibabacloud.com/
[7] https://azure.microsoft.com/en-gb/
[8] https://www.fujitsu.com/emeia/products/

IBM Cloud[9]

IBM is a household name in computer services. IBM cloud provide mainly infrastructure as a service. Products include Virtual server, containers, storage servers etc. It also provide automated services on their servers such as Auto-scaling, Network services and load balancing services among others.

5.4.3 Quality of Service Attributes

In the implementation stage, QoS attributes of the cloud services are gathered from a combination of cloud providers' websites, users' reviews scores, and cloud monitoring tool. The data is then collected and stored in the database. The QoS attributes specified here are slightly different from the QoS attributes specified in the model proposed in Chapter 3. During the process of data gathering, not all QoS attributes data is readily available. Therefore some QoS attributes are eliminated for the implementation stage (as discussed above). Below is the complete structure of QoS attributes which are considered in the implementation of the SST tool.

Table 5.1 QoS Attributes for Implementation

Level 1	Level 2	Level 3	Description
Security	Access Control	Authentication	Verifying the identity of the consumer
		Authorization	Checking permission of the consumer
	Data Security	Encryption	Encoding data to prevent unauthorized consumers to interpret the data
		Firewall	Protection of access to network from unauthorized consumers
	Geography	Provider's Nationality	Geographic location of service provider
		Machine Location	Geographic location of service or data centres
	Auditability	Auditability	Ease in measuring cloud services and indicates whether or not cloud provider allows auditors to do a thorough assessment
Usability	Interface	Web Interface	Interface to cloud service through web browser
		Mobile App	Interface to cloud service through mobile phone application
		Terminal Access	Interface to cloud service through shell terminal or command prompt
	Operability	Operability	How easy to operate and the ability to keep cloud service in a safe and reliable functioning condition

[9] https://www.ibm.com/cloud/

Level 1	Level 2	Level 3	Description
	Learnability	Learnability	How easy to learn and the capability of the cloud service to enable consumer to learn how to use it
Assurance	Availability	Uptime Percentage	How long the service has been running without outage
		Outages	Number of times cloud service is down or not available
	Downtime	Mean Time to Failure	The average time that an item will function before it fails
		Mean Time to Recover	Average time taken to recover from failure
	Recoverability	Recovery Mechanism	Mechanisms put in place to recover from failures
Performance	Hardware	Disk	Hard disk drive size of the cloud machine
		OS	Operating Systems of the cloud machine
		CPU Type	The brand and the type of the CPU(Central Process Unit)
		GPU	Graphic Processing Units-processor specialized for display functions
	Functionality	Network Performance	Measures of service quality of a network as seen by consumer
		Memory Amount	Size of the memory(Random access memory
		Number of CPU	Number of CPU cores
	Flexibility	Flexibility	The ability to customize or change a service easily
	Scalability	Scalability	The ability for cloud consumer to scale up and down the size of the workload in the existing infrastructure
Company Performance	Training	Training	Support provided by cloud providers to assist cloud consumers in how to use the cloud service
	Customer Support	Customer Support	Support provided by cloud provider to assist cloud consumers in any aspect of the cloud service
Pricing	Price	Windows	Cost of Windows Operating System
		Linux	Cost of Linux Operating System
	Charge Model	Pay As You Go	Cost of Pay As You Go instance
		Contract	Cost of an instance in contract
	Pricing Unit	Per hour	Service price that is calculated based on the time duration the service is running, price listed is price for each hour
		Per minute	Service price that is calculated based on the time duration the service is running, price listed is price for each minute
		Per GB	Service price that is calculated based on the storage used by consumer, price listed is price per Giga Byte of storage used

Level 1	Level 2	Level 3	Description
	Currency	Currency	Currencies that are accepted by cloud service provider as payment for service
	Support Fee	Free	Support available to cloud consumer at no additional price
		Additional Package	Support available to cloud consumer at additional price
	Discounting	Discounting	Price deduction for qualified consumer when cloud provider fails to deliver service according to service level agreement
	Pricing System	Tiered Pricing	Pricing model where cloud providers provide the service at different pricing package
		Volume Pricing	Pricing model that gives cloud consumers discounts for bulk purchase of service
Compliance	Security Compliance	US Patriot Act	US government Access to Information-compliance to provide appropriate tools required to intercept and obstruct terrorism act
		FISMA	How compliant the computing system is to Federal Information Security Management Act, which is a compliance to protect government information, operations and assets against threat
		Safe Harbor	Does storing information in another country comply with Safe Harbor laws, which is a compliance for US organizations to protect personal data of EU parties
	Legal Compliance	HIPAA	Compliance for Health Companies - Health Insurance Portability and Accountability Act
		Sarbanes-Oxley Act (SOX)	Act to protect investors by improving accuracy and reliability of corporate disclosures, which is a compliance related to financial and accounting to prevent errors and frauds
	Standard Compliance	SSAE 16	Statement on standards for attestation engagements, an auditing standard for reporting on control at a service organization
		ISO 27001	International Standard Organization related to information security management system
		ISO 9001	International Standard Organization related to requirements for quality management system
		ISO 27017	International Standard Organization related to information security controls aspect of cloud computing

Level 1	Level 2	Level 3	Description
		PCI DSS	Payment Card Industry Data Security Standard- information security standard related to the handling of credit card payment
		SOC 1	Service Organisational Control (Controlled by American Institute of Certified Public Accountants). Report regarding internal controls over financial reporting measured on a particular date
		SOC 2	Additional to SOC 1, it is a report related to security, availability, processing integrity, confidentiality, and privacy of cloud service
		SOC 3	Service Organisational Control (Controlled by American Institute of Certified Public Accountants). Additional to SOC 1, reporting on security, availability, integrity, confidentiality and privacy. Report Intended for general audience

5.4.4 Mapping of QoS Attributes Scores

The SST tool gathers data from three different sources which include, cloud provider websites, cloud monitoring tools, and user reviews. Though gathering data from different sources provide a more credible and un-biased service selection process, it complicates the process of data representation and mapping. In gathering QoS attributes data from the three sources values for each attribute can be varied. Some attributes have Boolean (yes/no) values. Some attributes have numerical values such as price and uptime percentage. Some attributes have string values, such as provider nationality and location. Because the values of QoS attributes vary, a mechanism is needed to map these different values into a numerical score. These values can then be quantified numerically and used in the proposed methodology based on AHP and SAW.

There are three different methods which are used to map attribute value into numeric score:

1. Assign score for Boolean value
2. Calculate score for numerical value
3. Assign uniform score for attributes that are hard to judge

It is important to note that there is no general standard followed when assigning weighting to a particular attribute. For instance, for some attributes, if a cloud provider fails to offer that service, the cloud provider is scored at 50 points out of 100 points for that attribute. On the other hand, on some attributes, a failure to provide that service is scored at 1 point. This weight assigned is based on how important each attribute is perceived to be.

5.4.4.1 Assigning Score to Boolean Values

The score range for cloud service is from 1 to 100 with 1 being the worst score and 100 being the best score. For attributes with Boolean value, if a 'yes' means a positive trait, then 'yes' will be assigned with 100 and 'no' will be assigned with 1. On the other hand, if a 'no' means a positive characteristic, 'no' will be assigned with 100 and 'yes' will be assigned with 1. For example, in the case of the Training attribute, if the value for training is 'yes', then it means that cloud provider provides training for customers. Thus the cloud service will get score of 100. Another example is Health Insurance Portability and Accountability Act (HIPAA) that is a security standard compliance which determines specific standards and regulations in order to protect the health information and patient critical data (Calloway *et al*., 2002). If the value of HIPAA attribute is 'no', then it means that the cloud provider does not comply with HIPAA regulation. Thus the cloud service will get a score of 1.

5.4.4.2 Calculate Score for Numerical Value

Several attributes have numerical values that could be easily used in AHP and SAW calculation. However, these numerical values cannot be directly used in calculation because they need to be converted into the score range of 1 to 100 which is considered in the implementation of SST tool.

The procedure to map the numerical value is as follows:

- **Sort**

  For one particular attribute, cloud services need to be sorted in ascending or descending order depending on the characteristic of an attribute. For example lower price is considered as a better option, cloud services are sorted in ascending order, with lowest price in the first place and highest price in the last place. Another example is uptime percentage attribute. The bigger the uptime percentage the better the cloud services are — so cloud services are sorted in descending order, with the highest uptime percentage in the first place and lowest uptime percentage in the last place.

- **Distribution of score range**

  After sorting the cloud services, score is then distributed for each of the service. Cloud service in the first place will automatically get the highest score of 100 while cloud service in the last place will get the lowest score of 1.

For the cloud services that fall in the mid-range, the score is assigned according to the following formula.

For attributes that are sorted in ascending order:

$$ScoreforcloudserviceA = Highestscoreintherange - \left(\frac{AttributevalueofserviceA - Bestvalue}{AttributeValueDelta}\right) * ScoreDelta$$

Where

$$AttributeValueDelta = Highestattributevalue - lowestattributevalue$$

$$ScoreDelta = Highestscore - lowestscore$$

While for attributes that are sorted in descending order:

$$ScoreforcloudserviceA = Highestscoreintherange - \left(\frac{BestValue - AttributevalueofserviceA}{AttributeValueDelta}\right) * ScoreDelta$$

Where

$$AttributeValueDelta = Highestattributevalue - lowestattributevalue$$

$$ScoreDelta = Highestscore - lowestscore$$

5.4.4.3 Assign Uniform Score for Non-Judgeable Attribute

For attributes that are hard to judge whether the value of a certain attribute means a positive characteristic or a negative characteristic, a uniform score of 100 is assigned to all cloud services.

An example of a non-judge-able attribute is provider's nationality and machine location. In this research it is considered that different nationalities and locations/regions do not have priorities over each other. Therefore all cloud services will receive 100 for these attributes. A full list of the score mapping for attributes is in Appendix I.

5.4.5 User's Reviews

User review is the score given by the cloud service consumers for a certain cloud service provider to indicate consumer's overall satisfaction with the service provided by the provider. For the implementation stage, user review score is gathered per cloud provider basis, and not per cloud service basis. User review for cloud providers is available and easy to gather while user review for particular cloud services from cloud provider is not completely available.

According to the literature survey of this research there is no standard sources that provide users reviews and their scores about cloud services. Therefore this thesis uses the following two sources for user review in the implementation:

Trust Radius[10]

Trust Radius provides reviews for business technology in order to help buyers and vendors in making better decisions about products and services. It covers wide range of reviews and some of them are related to cloud providers.

Gartner[11]

Gartner is a well-known global research and advisory organisation based in the US. They provide insights and advice in Information Technologies tools. They also provide benchmark analytics for Information Technology companies. Gartner Peer Insights is a platform provided by Gartner that offers authentic reviews and ratings for enterprise IT solutions and cloud services. These reviews are verified by Garter before they are published.

Note that a user's review score is given by consumer for a particular cloud provider, meaning that it covers user's whole experience with that provider. Thus it is difficult to dissect the user score based on QoS attributes. Therefore, in the SST implementation, user review score for a particular cloud service provider is evenly distributed to all attributes of a cloud service (from the given cloud provider).

In terms of score distribution, SST assigns 50% of total score that comes from attribute score (resulted from score mapping) and the other 50% comes from user's review. Both the value of mapping score and user's review score make up the final score of a cloud service.

An example of final score calculation is shown below in Table 5.2. In this example, security QoS main attribute in SST consists of four sub-attributes (level two) which are Access Control, Data Security, Geography and Auditability. The level three also contains seven sub-attributes as shown in the previous Table 5.1. These attributes are Authentication, Authorization, Encryption, Firewall, Provider's Nationality, Machine Location and Auditability. Accordingly, the total maximum perfect score for security QoS attribute is equal to 700 because each sub-attribute score is considered as 100. The score mapping of these seven sub-attributes is assumed to be equal to 500 and the score of the user review is 8 out of 10, which is 0.8 or 80%. As a result, the total score for the security attribute is the sum of user reviews score (weight is considered as 50% or 0.5) and the attribute score (weight is considered as 50% or 0.5).

Total Score = Attribute Score + User Review Score

((0.5*500) + (0.5*(0.8*700)) note that the user review score is the score for the provider in general, not for each specific sub-attributes. Therefore, multiplying 700 with 0.8 (or 80%), the

[10] https://www.trustradius.com/

[11] https://www.gartner.com/reviews/market/public-cloud-iaas

total score of security QoS attributes including the user review is equal to 530 as shown in Table 5.2.

Table 5.2 Example of User Review Score Mapping Calculation

Total Score		Example: Security	
Attribute Score	50%	Security Max Score (Perfect Score for Security)	700
User Review Score	50%	Attribute Value Mapping Score	500
TOTAL	100%	User Review Score	8 out of 10 (80%)
Total Score			
Attribute Score	50% * 500	250	
User Review Score	50% * (80% * 700)	280	
TOTAL		**530**	

5.4.6 Service Level Agreement (SLA)

Contents of typical SLA have been discussed in Chapter 2. In the implementation stage, SLAs from several cloud service providers are analysed in order to define a concise and simple SLA template that contains only the important information that consumer might be interested in.

1. Amazon AWS SLA

   The general structure of the Amazon AWS SLA involves the following factors:

   - Included Product
   - Service Commitment
   - Definitions
   - Service Commitments and Service Credits
   - Credit Request and Payment Procedures
   - SLA Exclusions

2. Google Cloud

   Structure:

   - Definitions
   - Financial Credits
   - SLA Exclusions

3. Microsoft Azure

Structure:

- Introduction
- General Terms
- SLA Details
    - Additional Definitions
    - Monthly Uptime Calculation and Service Levels & Service Credit (in Availability Zones)
    - Monthly Uptime Calculation and Service Levels & Service Credit (in Availability Set)
    - Monthly Uptime Calculation and Service Levels & Service Credit (Single Instance Virtual Machine)

From the above three examples, it can be concluded that the two common contents for all SLAs are term definitions and the amount of service credit for certain monthly uptime percentage. These two types of contents will be included in the SLA template. In addition, the template will include egress price, or the price consumers have to pay if they want to move their data out of the cloud infrastructure of a certain cloud provider. If data transfer price is available, this means moving data out of one cloud provider to another place is possible for consumer but at a certain price.

The following template for the SLA information is included in the SST:

```
Service Level Agreement - SLA

Definitions
    Definitions of Downtime
    Definitions of Monthly Uptime Percentage
    Definitions of Service Credits

Service Credits
    Monthly uptime percentage & corresponding service credit

Ability to Migrate Data (Yes / No)

Fee to Migrate Data Out (Yes / No)

Data Transfer Fee
    Price list for transferring data out of cloud infrastructure
```

Figure 5.3 SLA Template

5.5 Summary

This chapter provided an overview of the proposed tool known as SST. The SST comprises three main components which include the web server, database layer and the client side / user interface. These three components have different features. The database layer (MongoDB) acts as the storage layer for data collected from the different sources. The Client interface provides a user interface that makes it easy for consumers to make use of the tool while the web server implements the algorithm proposed in this work. The web server also interacts with the back-end database using embedded queries to submit requests and receives responses. The system runs on an AWS EC2 server.

Assumptions and constraints regarding the design and implementation of SST tool are also discussed. Assumptions include: consumer preference inputs of QoS attributes are accurate; scrapping of QoS attribute data from eight cloud service providers; neglecting all the other QoS attributes with missing values in the implementation stage.

The design phase of the SST tool is thoroughly explained. The architectural design of the SST tool shows the main building blocks of this tool and how each of these blocks is integrated to rank the best cloud service providers. This ranking is based on the consumer's requirements. Cloud service data are gathered from eight distinct cloud service providers, such as, Oracle Cloud, Fujitsu Cloud and IBM cloud. The complete structure of QoS attributes used in the implementation stage and the different techniques for mapping QoS value into numerical scores are also discussed. User's review is one of the sources used to gather information to provide credible and unbiased results. These reviews basically contain texts and rate numbers. However, SST tool just considered the rate number and taken this into account for the evaluation of the cloud service providers.

SST provides SLA templets with brief information such as downtime, monthly uptime percentage, and corresponding service credit, etc. This information helps the consumers to have an instance look for the service guarantee levels that are offered by each cloud service providers. The next chapter will explain the implementation of the SST.

CHAPTER 6

Implementation of the Sahab Service Selection Tool

6.1 Introduction

This chapter describes the implementation of the Sahab Service Selection Tool. Firstly, the chapter describes the functional requirements that drive the design of the tool. The non-functional requirements are also highlighted briefly in this chapter. The chapter then explains the codes used for calculating scores and ranking cloud providers based on the Quality of Service attributes listed in section 5.4.3. Furthermore, the chapter explains the consumer various consumer journeys that are possible on the SST tool. Finally, the various interfaces that the consumer is presented with are explained with snippets and diagrams.

6.2 Sahab Service Selection Tool: Requirements

The SST needs to be able to read input from the consumer, process consumer's input using the AHP and SAW methodologies and present the cloud consumer with ranking of cloud services that best match consumer's preference. Because SST's main purpose is to help consumer in cloud selection process, SST needs to able to gather data about cloud service providers along with cloud services that the providers provide. SST gathers cloud service data from three different sources: cloud provider-provided information, users' reviews, and third party monitoring tool. All the data gathered from these sources will then be stored in a main database. The consumer preference – cloud service matching process will compare consumer's input with the data available in the database.

SST also provides additional information to the consumers that can help them in their decision making process. SST provides complete description of all QoS attributes used in the system. SST also provides brief information on Service Level Agreement (SLA) for each cloud service provider to help consumer view the service guarantee level offered by each cloud provider. Each of the requirements is presented, followed by a description of the requirement for clarity.

6.2.1 Functional Requirement Specification for SST

The Functional requirements for the SST are listed and explained below. Though the tool can be extended to implement additional functional requirements, the following requirements are considered to be sufficient to validate the tool.

REQ-F-01

SST should give the option for consumers to select their level of knowledge in cloud computing.

This requirement puts into consideration the fact that some consumers may not be as knowledgeable as others. Therefore, depending on the consumers understanding of cloud services and quality of service attributes, they can decide on how much detail they want the SST to present to them. They can also determine how much of their preferences they want to input into the SST for calculation.

REQ-F-02

SST gives option to consumers to select the number of main-level QoS attributes which are to be included in the selection process.

The main level (level one) QoS attributes, listed in section 5.4.3, are the main service attributes that the SST tool focuses on. The tool must allow the consumer to be able to compare and rank the service providers based on these attributes. In addition, the tool should allow the consumer to select the required number of attributes from the available list. Therefore, the tool should be flexible enough to allow the consumers to choose the attributes they want to compare.

REQ-F-03

SST is able to process cloud service selection for Beginner-level consumer.

The SST should take into consideration the fact that some consumers may not be as knowledgeable on cloud services. The tool should therefore understand that a Beginner level consumer may not need extra details when trying to make a choice on cloud providers. In addition, to help a novice consumer, the tool must provide, where possible, tips that can help the consumer understand the attributes that they select.

REQ-F-04

SST is able to process cloud service selection for Intermediate/Expert-level consumer.

The SST tool should also allow a more experienced consumer to effectively use the tool. The tool must be able to provide various levels of details for various consumers. That is, the tool should provide as much details as possible which will allow a more experienced cloud consumer to make more informed decisions when they want to choose a provider.

REQ-F-05

SST gathers cloud service data from cloud provider websites, third party monitoring tool, and user review websites.

The SST must be able to collect data from a variety of sources. The tool cannot rely on data from an individual source to make the decision as this may not be sufficient to rank all the given attributes (see section 5.4.3). The SST must therefore have the ability to collect data from a variety of sources. Also, the SST must not rely solely on the information provided by the cloud providers as this may be biased. The tool must have the ability to act as a mesh i.e. to collect data from different sources including third party monitoring tools and cloud service review websites in order to guarantee the credibility of the tool and provide unbiased results.

REQ-F-06

SST is able to map QoS attributes value into a score that can be processed using AHP and SAW methodologies.

The tool must have the ability to score, aggregate and calculate various data from different sources. The tool must be able to convert various data types such as Boolean into numerical values that can be used in calculation and ranking. Also, the tool must have the ability to rank cloud providers based on the listed QoS attributes using the AHP and SAW methodologies. A combination of SAW and AHP methodologies are used for calculating the sub-attributes while AHP alone is used for calculating the first level attributes.

REQ-F-07

SST is able to utilize AHP and SAW methodologies to read consumer input and then to produce a ranking of best-matched cloud services.

The tool must have the ability to apply AHP and SAW methodologies to the converted scores. This will allow the tool to provide a score for all the attributes and use these scores to provide final ranking. This is an important requirement for this tool. The tool makes use of these two methodologies in the calculation of service’s score and ranking.

REQ-F-08

SST provides SLA information.

The proposed tool must be able to provide the service level agreement information for each of the service providers which are being ranked. The SLAs of service providers incorporated in this tool are listed in Section 5.4.6.

6.2.2 Non-Functional Requirement Specification for SST

The non-functional requirements of the SST are listed below.

REQ-NF-01

SST assists consumer in understanding QoS attributes by providing brief description for each QoS attribute which consumer will use.

This requires that the SST provides a description for each of the QoS attributes to be considered. This will ensure that all the consumers have the right understanding of the attributes. This requirement is especially useful for novice consumers who may not have a good understanding of cloud services.

REQ-NF-02

SST provides a simple interface for consumer.

This requires that the interface for the SST must be user-friendly. All levels of consumers must find it easy to use and interact with the tool. The consumer journeys must be easy to navigate in order to retain consumer engagement. It must also be easy for the consumers to find their way around the tool.

The next section explains the various use case scenarios that the SST presents.

6.2.3 Use Case Specification for SST

After specifying the functional and non-functional requirements of the tool, a set of use case is then created to represent all the actions that happen in the tool. Use cases are created based on the functional requirements. The use cases are also used to “determine the functional scope of the objects in a system” (Anda and Sjoberg, 2005). The complete list of use cases for SST is explained below. For each of the use cases explained, the requirements that the use case covers are identified. For all the use cases, the actors are the consumer and the SST.

Choose Level of Knowledge (UC-01)

This use case is the first activity that the consumer can perform on the SST tool. As such, there is no pre-conditioned activity for this use case. The Consumer can choose the level of their knowledge of cloud services. Consumer can choose either Beginner level or Intermediate/Expert level. This use case covers requirement REQ-F-01.

Show Attribute Description (UC-02)

This use case is a follow on activity from UC-01. The Consumer is presented with a set of attributes. Consumer can decide to see a brief description of each of the attributes. The pre-condition for this activity is that the Consumer must have selected a knowledge level from the previous page. If the consumer does not select a level of knowledge, he/she will not be allowed to progress to this page. To show the description for each attribute, the consumer must hover his/her mouse around the question mark icon beside each of the attributes. The description of the attribute is then presented to the consumer.

Choose Main-level Attributes (UC-03)

Use-Case UC-03 states that consumer can choose as many attributes as they want from among the presented attributes. The pre-condition for this use case is that consumers must have selected a level of knowledge. Also, there is a limitation to this use case that the consumer must select a minimum of two attributes. If the consumer does not select up to two attributes, the system pops up a dialog box asking the consumer to select a minimum of two attributes. The consumer can then click on the "OK" button in the dialog box. The consumer is not allowed to progress on his/her journey until this condition has been satisfied. This use case covers requirement REQ-F-02.

Beginner-level Pairwise (UC-04)

This use case allows a Beginner consumer to be able to carry out a pairwise comparison based on attributes chosen in UC-03. The pre-condition for this use case is that consumers must have selected the Beginner level of knowledge and also selected a minimum of two attributes. After selecting this attribute, the consumer is presented with the attributes and must click on the 'calculate' button to carry out a Beginner level pairwise calculation. This use case covers requirement REQ-F-03.

Int/Exp-level Pairwise (UC-05)

Use case UC-05 allows a consumer to perform an Intermediate/Expert level pairwise comparison on the SST tool based on the attributes chosen in UC-03. The pre-condition for this is that consumers must have chosen an Intermediate/Expert level in UC-01. Also, the consumer must have chosen at least two attributes from the UC-03. The consumer must then click the 'calculate' button and choose to view result. This use case covers requirement REQ-F-04.

Int/Exp-level Sub Param Pairwise (UC-06)

Use case UC-06 allows a consumer to perform an Intermediate/Expert level pairwise comparison on the SST tool based on level two QoS attributes of the level one attributes chosen in UC-03. The pre-condition is that the consumer must be on the Intermediate/Expert consumer journey on the tool and must select the 'next level 2' button from the UC-05 instead of choosing to view the result. This use case covers requirement REQ-F-04.

Int/Exp-level Level 3 Filtering (UC-07)

Use case UC-07 allows a consumer to perform an Intermediate/Expert level filtering on the SST tool based on level three QoS attributes of the level one attributes chosen in UC-03. The pre-condition is that the consumer must be on the Intermediate/Expert user journey on the tool. In addition, the consumer must select the 'next level' button in UC-05 and must also select the 'next level 3' button from UC-06 instead of choosing to view the result. After this, the consumers can then view the result. This use case covers requirement REQ-F-04.

Beginner Level rank calculation (UC-08)

This use case helps the consumer to provide a ranking of cloud services from different providers. Consumers must have chosen the Beginner level in UC-01 to reach this point. Consumer would also have chosen a minimum of two level one attributes and then select the calculate button. The SST then provides the ranking. This use case covers requirements REQ-F-03, REQ-F-06 and REQ-F-07.

Int/Exp Level rank calculation (UC-09)

This use case provides an Intermediate or Expert level consumer with a ranking of cloud services from different providers. Consumers must have chosen the Intermediate/Expert level in UC-01 to reach this point. Consumer would also have chosen a minimum of two level-one attributes and then select the next level 2 button in UC-05. The SST then provides the ranking after the consumer has selected on View result. This use case covers requirements REQ-F-04, REQ-F-06 and REQ-F-07.

Data Scrapping (UC-10)

As mentioned in REQ-F-05, the SST must be able to gather data from multiple sources including cloud service providers websites. This use case is used to collect data from various sources and store the data in the back end database. This process can be run periodically to update the data in the repository. The data is then available for the SST to calculate the cloud provider ranking. This use case covers requirements REQ-F-05.

Show SLA (UC-11)

This use case allows the consumer to view the Service Level Agreement of cloud providers. The pre-condition for this use case is that the consumer must have completed either the Beginner or Intermediate/Expert level journeys on the SST tool. The consumer is then presented with the ranking. The consumer can select any of the providers to view their SLAs. This use case covers REQ-F-08.

6.3 Implementation of the SST

The SST is implemented as a set of a number PHP scripts, JavaScript scripts, HTML5, and MongoDB documents. The different components of the SST tool interact with each other in order to process consumer's requests of selecting required cloud services.

The first step of the SST is to scrap data into the back end MongoDB. There are multiple files for scrapping data from different sources. Recall from section 5.4.2, data is scrapped from the websites for each of the (eight) cloud providers used in the SST. In addition, data is also scrapped from user review websites as well as the cloud monitoring website. The data scraping component is written using python. For each of the data sources, there is a file for scrapping data. The Figure 6.1 below is a snippet from the <cloudharmony.py> file. This file scraps data from the cloud harmony website, parses the data and writes it into the MongoDB for the SST to access.

```
  "outputs": [],
  "source": [
   "# url\n",
   "url = 'https://cloudharmony.com/status-of-compute'\n",
   "page = requests.get(url)\n",
   "page_string = page.text\n",
   "soup = BeautifulSoup(page_string, 'html.parser')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {
   "collapsed": false,
   "scrolled": false
  },
  "outputs": [],
  "source": [
   "# Get First Table\n",
   "table = soup.find_all('table')[0]\n",
   "tbody = table.find_all('tbody')[0]\n",
   "\n",
   "hold_all = []\n",
   "hold_data = []\n",
   "\n",
```

Figure 6.1 Code Snippet from cloudharmony.py File

The three diagrams depicted below show the flow for SST actions. Fig. 6.2 provides an option to consumer for choosing the level of knowledge, Fig. 6.3 and Fig. 6.4 respectively show the flow of actions for Beginner and Intermediate/Expert consumers.

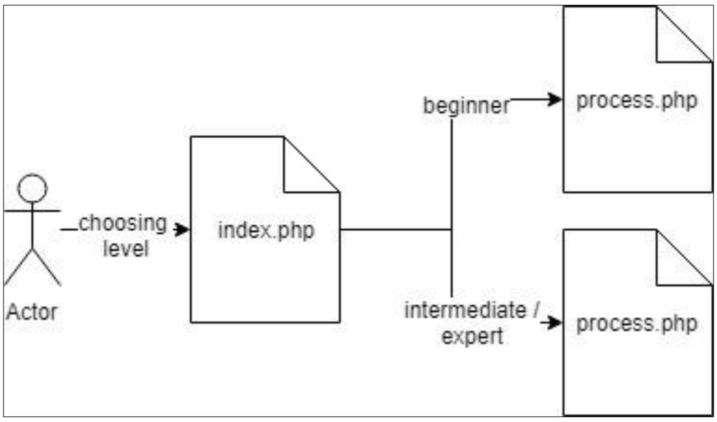


Figure 6.2 Code Flow for Choosing Knowledge Level

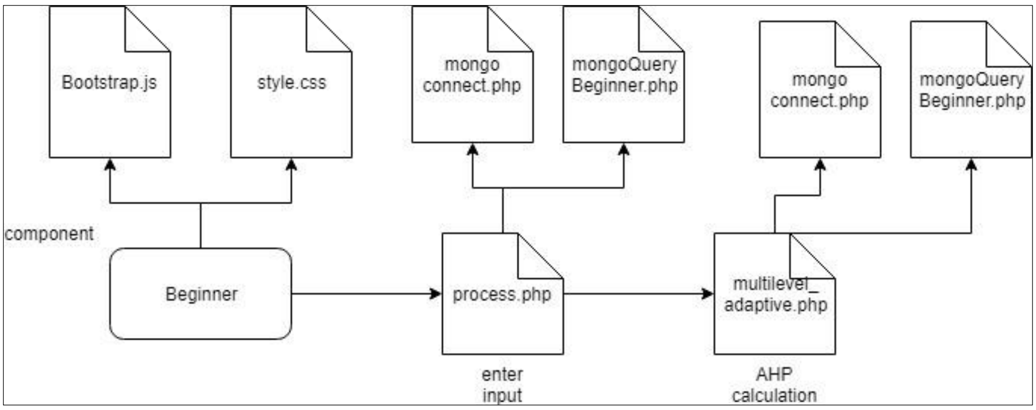


Figure 6.3 Code Flow for Beginner Consumer

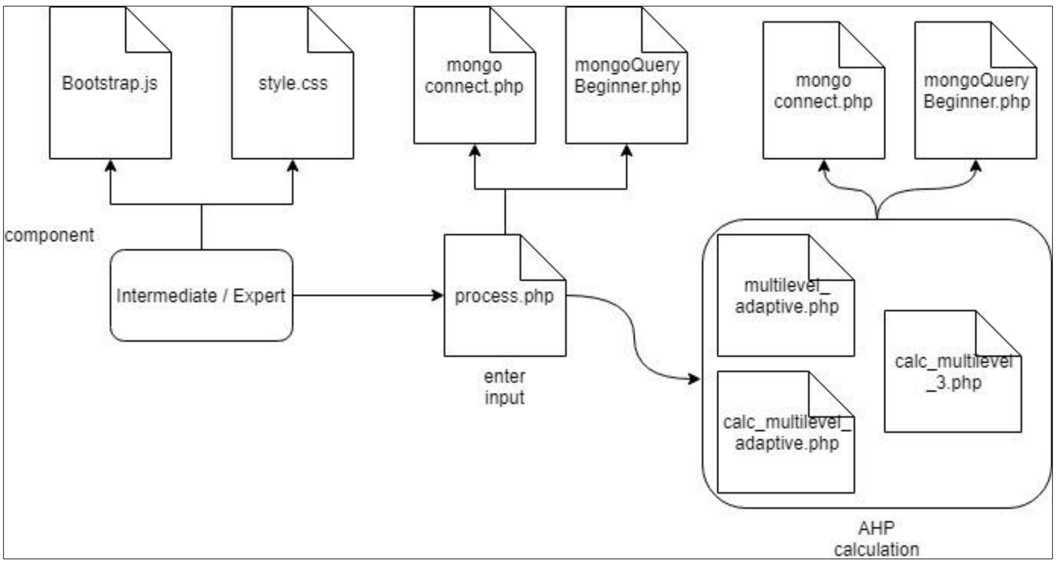


Figure 6.4 Code Flow for Intermediate / Expert Consumer

1. Interface

   <index.php> is the main interface where the main page of SST is located. When consumer opens the tool for the first time, consumer will see this page.

   Through this page, consumer will select the level of knowledge in cloud computing. <index.php> will also process this input from consumer and forward it to the input processing part.

```
      <a href="http://localhost/ahp/ui/index.php">HOME</a>
      <small>Dashboard</small>
    </h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
      <li class="active">Main Page</li>
    </ol>
  </section>

  <!-- Main content -->
  <section class="content">
    <div class="row">

      <!-- col right -->
      <?php
      //include "db.php";

        // (notice undefined index)

      if (!isset($_POST['next'])) {
        include "main_page.php";
      }
      elseif (isset($_POST['next'])) {
        if ($_POST['level']=='beginner') {
          include "beginner/level_1.php";
        }
        elseif ($_POST['level']=='intermediate') {
          include "level_1.php";
        }
        elseif ($_POST['level']=='intermediate-lvl2') {
          //if (isset($_POST['level_1'])) {
            include "level_2.php";
          //}
        }
        elseif ($_POST['level']=='intermediate-lvl3') {
          include "level_3.php";
        }
        else {
          include "main_page.php";
        }
      }

      ?>
      <!-- /.col -->
    </div>
```

Figure 6.5 The SST Home Page (Code Snippets)

2. Input Processing

   <process.php> is responsible for pairwise comparison interface as well as responsible for processing all the inputs from consumer. This file has script component as part of the interface.

   <process.php> processes the input from the main interface, as well as input from pairwise comparison step.

```
            ";
            ?>
<script>
$(function() {
    var handle = $( "#custom-handle_<?php echo 't-'.$i.'-'.$j;?>" );
    $( "#slider_<?php echo 't-'.$i.'-'.$j;?>" ).slider({
      value : 5,
      min : 1,
      max : 9,
      create: function() {
        handle.text( vals[$( this ).slider( "value" )-1] );
      },
      slide: function( event, ui ) {
        var vslider = vals[ui.value-1];
        var v = Math.abs(eval(vslider));
        handle.text( v );
        $('#txt_<?php echo 't-'.$i.'-'.$j;?>').val(vslider);
      }
    });
  } );
</script>
```

Figure 6.6 The SST process.php Page (Code Snippets)

3. AHP Calculation

<multilevel_adaptive.php> is responsible for the AHP calculation, which include processing consumer's input and producing the final ranking of providers. This component is specifically responsible for calculation of level one AHP (i.e., for Beginner consumer).

<calc_multilevel_adaptive.php> is responsible for the AHP and SAW calculation in the second level. This component deals with sub-parameters, unlike <multilevel_adaptive.php> that deals with top-level parameters only. This component is responsible from processing consumer's input in level-2 attributes to producing the final ranking of providers. AHP and SAW methodologies are implemented as part of this component.

```
 echo json_encode($score); */
//exit();
//####################################
//        1. BEGIN PAIRWISE MATRIX
//####################################
echo "<h4>Step 1 : PAIRWISE MATRIX</h4>";
// Create diagonal 1 value
$data = array();
for ($i = 1; $i < count($criteria); $i++) {
    for ($j = 1; $j < count($criteria); $j++) {
        $data[$i][$j] = 1;
    }
}

// Fill other value except the diagonal
for ($i = 1; $i < count($criteria); $i++) {
    for ($j = 1; $j < count($criteria); $j++) {
        if (!empty($_POST['t-' . $i . '-' . $j])) {
            $v = $_POST['t-' . $i . '-' . $j];
            if ($v < 0) {
                $data[$i][$j] = abs($v);
                $data[$j][$i] = 1 / abs($v);
            } else {
                $data[$i][$j] = 1 / abs($v);
                $data[$j][$i] = abs($v);
            }
        }
    }
}

echo '<div class="table-responsive"><table class="table table-bordered">';
echo '<tr>';
foreach ($criteria as $c) {
    echo '<th>' . $c . '</th>';
}
echo '</tr>';
```

Figure 6.7 Code Snippet from multilevel_adaptive.php

<calc_multilevel_3.php> is responsible for activities in the level-3 of the attribute hierarchy. In level 3, consumer can filter cloud services based on specific value on certain attributes. This component will handle this filtering process.

4. NoSQL Database

The component <mongoconnect.php> is responsible for establishing connection with mongoDB, which is a NoSQL database.

<mongoQueryBeginner.php> is responsible for identifying each attribute name in the database and assign variable name for each of these attributes. In addition, this component is also responsible for attribute's brief description. This component communicates with the component responsible for AHP calculation.

The snippet below (Fig. 6.8) shows the QoS value's for g1-small compute engine by Google cloud. The snippet shows the various level-2 attributes (recoverability, availability and downtime) of the level 1 assurance QoS attribute.

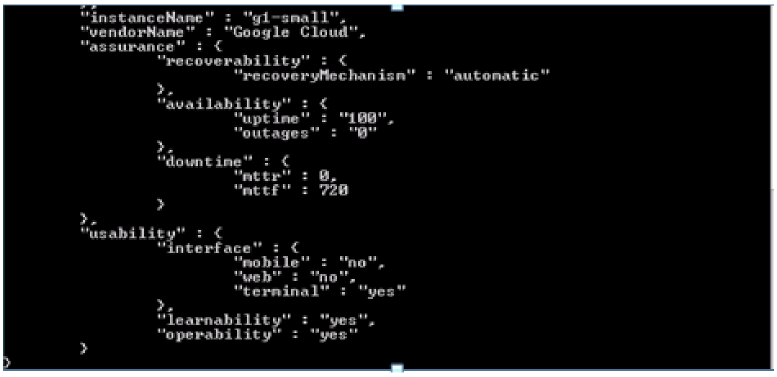


Figure 6.8 QoS Data in MongoDB (Snippet)

6.4 Validation of the Tool

All the main functional requirements (defined above) have been implemented. The following section shows the screen shots of the implementation in order to demonstrate how the SST tool carries out the cloud service selection process.

6.4.1 Interface Design

HOME Page of the SST tool (UIF-01)

Figure 6.9 shows the main (home) page of the SST. The user interface presents the consumers with an option to choose their level of expertise – based on how much knowledge on cloud computing services they have. The consumer is presented with a drop down box to choose from. The available choices are limited to Beginner or Intermediate/Expert. However, the SST tool can be extended to include other levels, if needed. The interface diagram below (Home Page, UIF-01) covers the use case UC-01 presented in section 6.2.3.

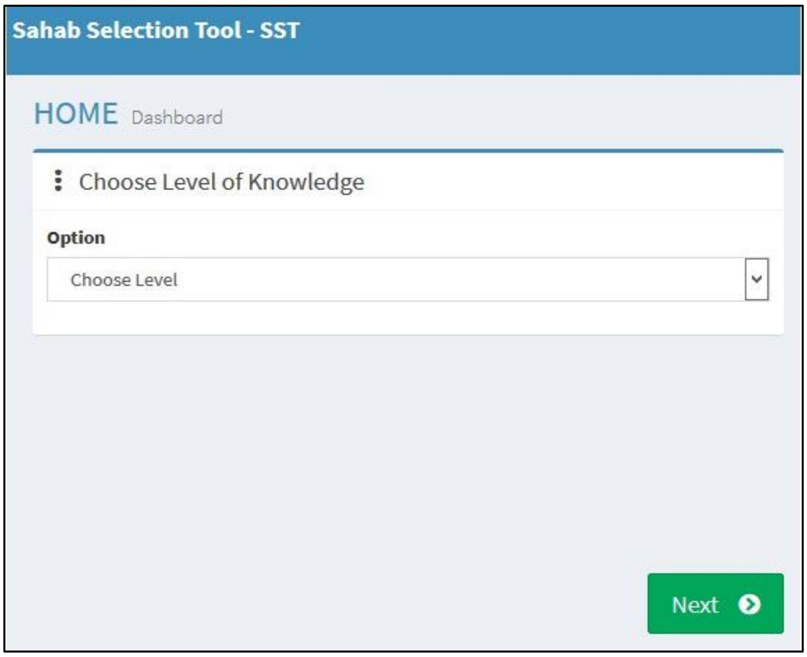


Figure 6.9 Interface UIF-01 SST Main Page

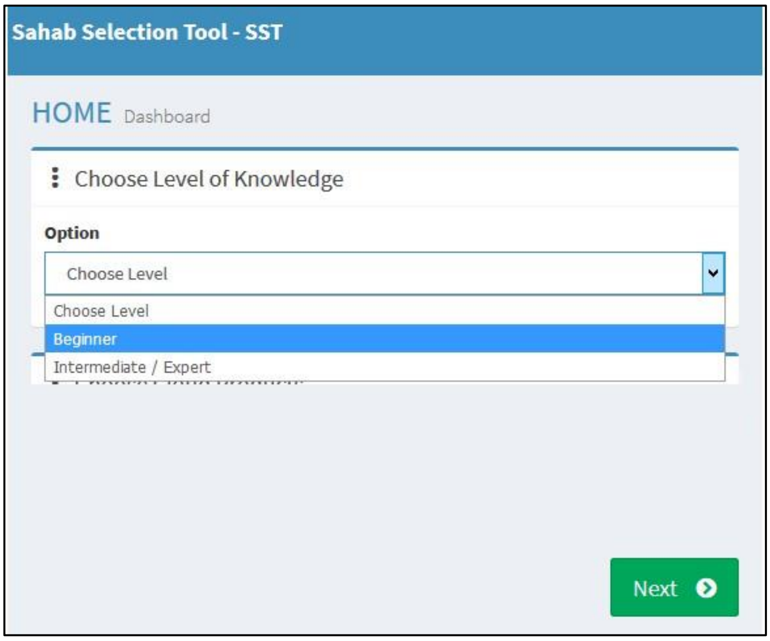


Figure 6.10 Interface UIF-01 – Choose Knowledge

<u>Parameter Selection of the SST tool (UIF-02)</u>

After selecting their level of knowledge (or expertise level), the consumers are presented with the parameter page that contains the level 1 attributes discussed in section 5.4.3. The consumer is expected to choose from the attributes listed. Each attribute has a check box beside it that allows the consumer to select more than one attribute. In addition, each of the attributes listed is designed with a "*tooltiptext*" HTML function. This function allows SST to provide more information about each of the attributes. Therefore, when the consumer hover the mouse on the button beside each of these attributes, the definition of each of the attributes are presented to the consumer. This feature is important for the consumer to have a clear understanding of each of the level one attributes as defined by the SST. Any attribute selected by the consumer will be considered as part of the attributes on which the cloud providers will be scored. In addition, the SST tool puts a restriction on this page such that a consumer must select at least two attributes. Once the attributes are selected, the customer can proceed to process their choice. This page covers Use cases UC-02 and UC-03 as explained in section 6.2.3.

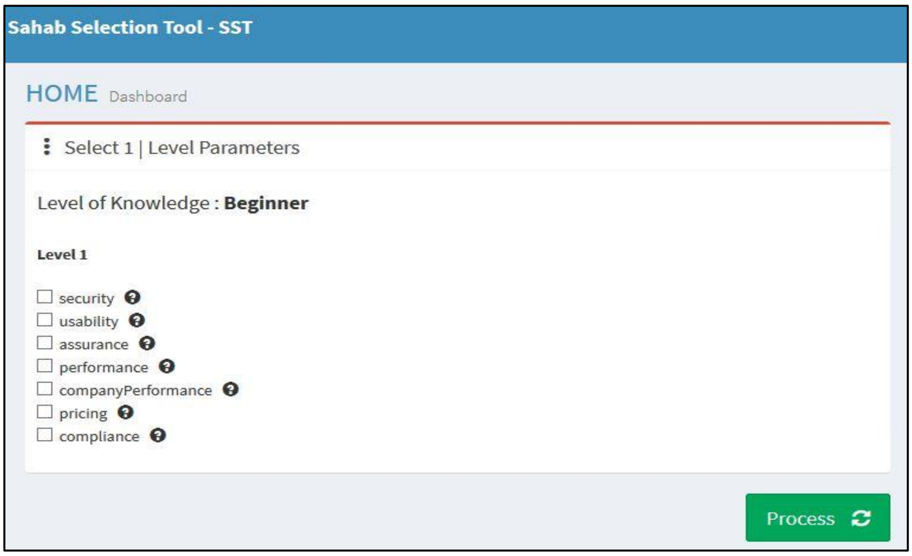


Figure 6.11 Interface UIF-02- Choose Attributes

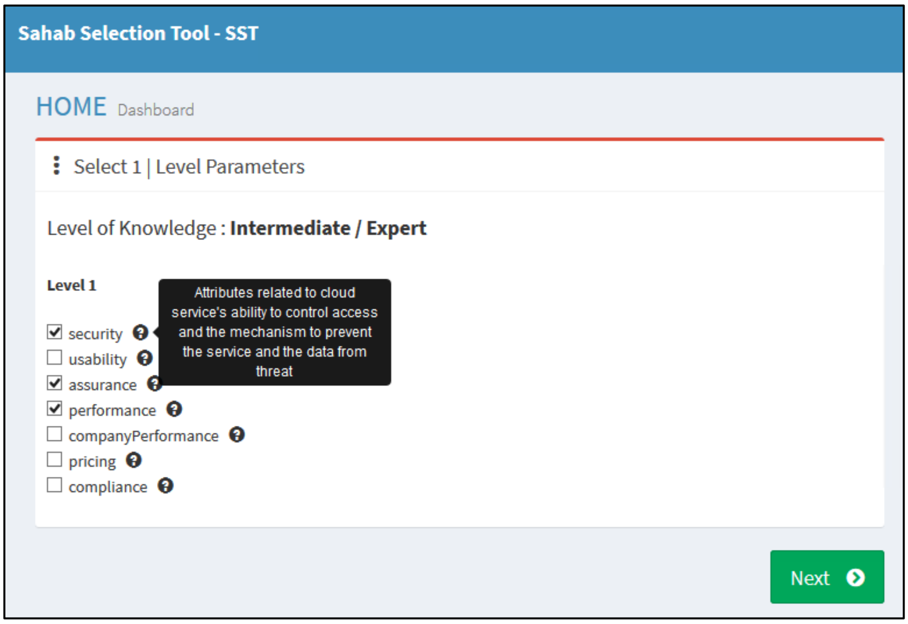


Figure 6.12 Interface UIF-02 Attribute Descriptions

Level One Pairwise Comparison (UIF-03)

The SST implements a pairwise comparison as explained in section 4.5. One novel and key feature which the SST boasts of is that it is flexible enough to allow consumers to determine which attribute is more important to them. This page implements the pairwise comparison in two steps. Firstly, the page presents each pair of all the attributes selected by the consumer. For instance, if the consumer selects three attributes in the previous page, then each of the three attribute is compared against another attribute and presented on this page. Secondly, the consumer is allowed decide which attributes are of more priority to him/her. This is achieved by displaying a slider between each pair of attributes. The consumer can then assign an 'importance measure' to an attribute which they feel is more important. This feature makes the SST more user-friendly and helps the consumer feel that they are able to customize the tool to their preference. The consumer can then click on the "calculate" button. The button then posts the users input to the next page (multilevel_adaptive.php) that implements the algorithm for ranking cloud provider scores. This page implements the use cases UC-04 and UC-05 defined in section 6.2.3

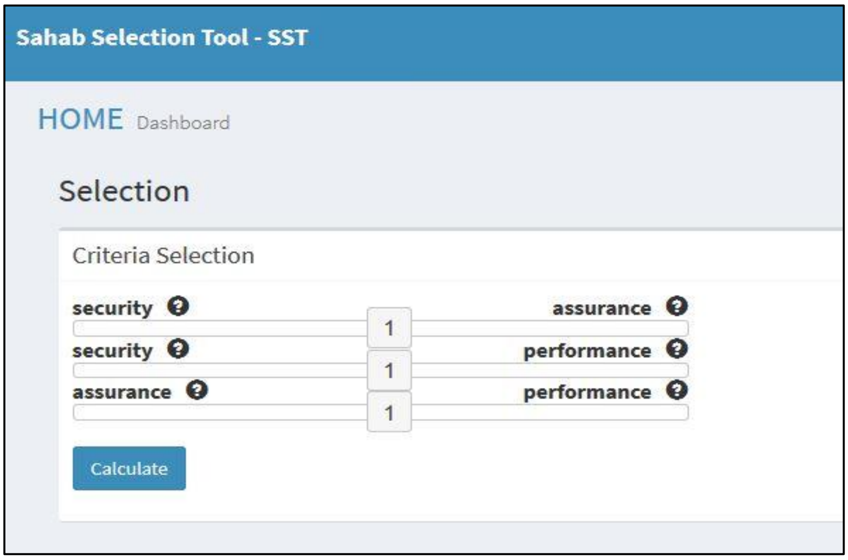


Figure 6.13 Interface UIF-03 - Pairwise Comparison

Sub Parameter Pairwise Comparison (UIF-04)

This page is restricted to Intermediate or Expert level consumers, wherein a consumer has chosen the Intermediate or Expert level from the drop down option box in the SST home page (UIF-01). Therefore, if the consumer chooses a Beginners option, this page will not appear to the consumer. This page allows the consumer to further input their preference in the SST calculation. It also makes use of pairwise comparison as explained in UIF-03. However, this page allows the consumer to make a pairwise comparison on level two QoS attributes as explained in section 5.4.3. The level two attributes of the level one attributes selected by the consumer in page UIF-02 are pulled and presented to the consumer. The consumer can then grade each of the attributes with a score based on his or her preferences. This consumer preference is then calculated as part of the SST's internal algorithm. To progress, the consumer is asked to click on the calculate button. This page implements the use case UC-06 defined in section 6.2.3.

Comparing SST to other tools such as Intel Cloud Finder (Intel®, 2013), Cloudorado (Cloudorado, 2011) and Rankcloudz (RightCloudz, 2014), SST is a more flexible tool that allows consumers to go in any depth and select their priority attributes or preferred attributes. However, other existing tools do not have this capability.

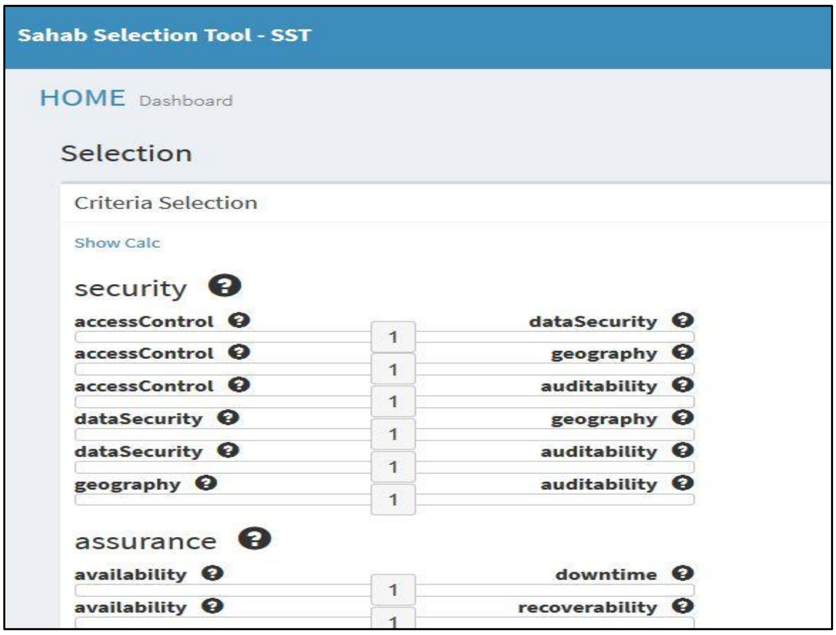


Figure 6.14 Interface UIF-04 - Sub-parameter Pairwise Comparison

<u>Level Three Filtering Attributes (UIF-05)</u>

This Page (UIF-05) performs similar function as pages UIF-04 which is to allow consumers to input their preferences to the SST calculation. However the function performed here is to allow consumers to filter certain attributes that they think are important. This filter function applied is limited to level three QoS attributes. Before completing the previous page (level two attribute preference selection page), the consumer is asked if he/she wants to view the ranking or proceed to input his/her preferences on level 3 attributes. If the consumer chooses to view his/her results, then the SST implements the AHP and SAW methodologies without proceeding to this page. However, if the consumer is an expert level consumer and has some level three attributes which he/she believes must carry some level of importance; he/she can choose to proceed to level three attributes which this page presents. This is an important function because a main design consideration for the SST is that consumers must be able to input their preferences at all levels of attributes. The consumers can then select certain criteria which they believe needs to be filtered in the SST calculation. The filtering here makes use of binary type of selection and not a comparison i.e. consumers can either select or deselect attributes. Again, this page does not appear on the consumer journey if the consumer chose Beginner level from the home page. This page implements the use case UC-07 defined in section 6.2.3.

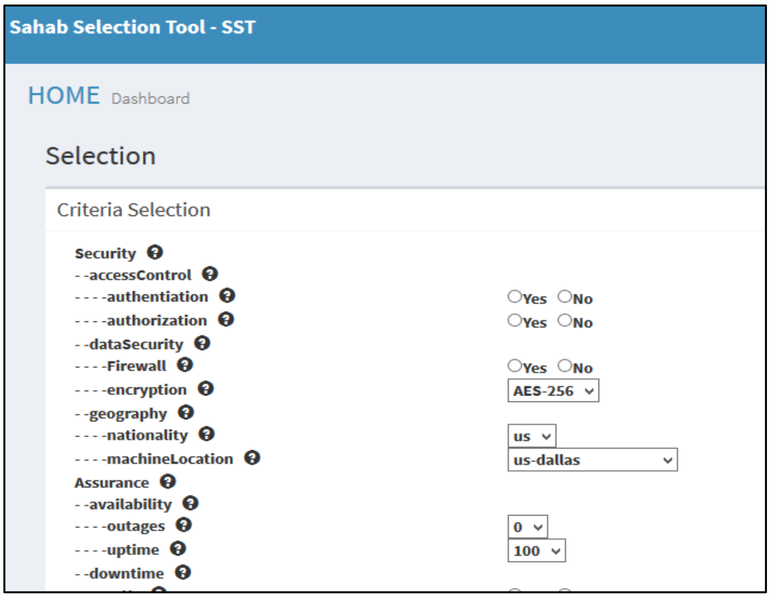
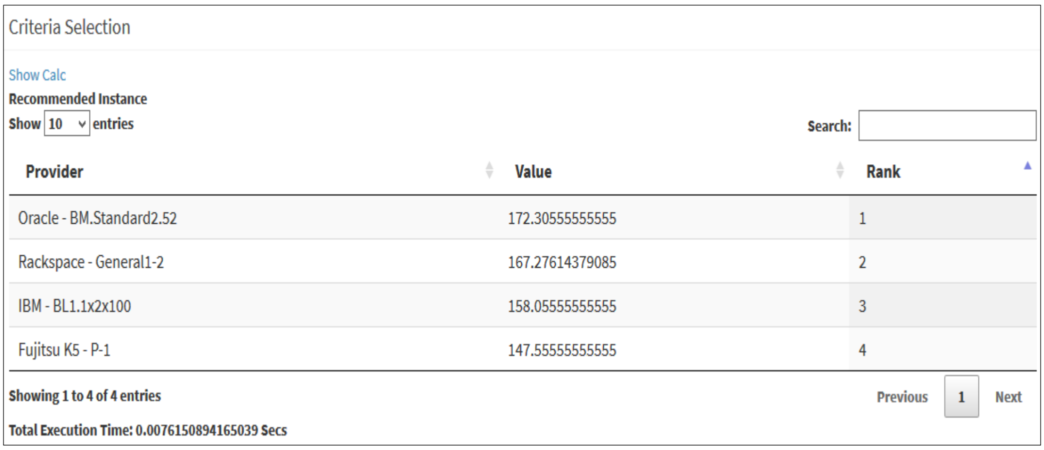


Figure 6.15 Interface UIF-05 - Level 3 Attributes Filtering

Cloud Service Ranking (UIF-06)

This page is the last page of the SST which presents the result to the consumer based on all the criteria that the consumer has selected throughout their journey on the SST tool. The page shows the ranking of all the cloud providers as well as their scores. The page also has a search box that can allow consumers to search for a particular cloud provider if they do not want to scroll through the entire list. Consumers can also determine the number of entries they want to show up in a page. If the result entries cannot fit an entire page, the consumer can click on the 'next' page to continue viewing the results. In addition, there is a 'preview' button beside each result entry. If the consumer clicks on the button, it pops up with a PDF file which gives the consumer more information on the cloud provider that they selected. The PDF file contains the SLAs guaranteed by the particular provider that they selected. This ranking page is very important, as consumers can then see the actual result per provider on some of the attributes they selected. Furthermore, the PDF file contains definition of some of the SLA parameters provided by the cloud provider such as the uptime percentage SLA guaranteed by the provider. Consumers are allowed to download the PDF file. This page implements the use cases UC-08, UC-09, UC-10 and UC-11 defined in section 6.2.3.



Criteria Selection

Show Calc

Recommended Instance

Show 10 **entries** **Search:**

Provider	Value	Rank
Oracle - BM.Standard2.52	172.30555555555	1
Rackspace - General1-2	167.27614379085	2
IBM - BL1.1x2x100	158.05555555555	3
Fujitsu K5 - P-1	147.55555555555	4

Showing 1 to 4 of 4 entries Previous 1 Next

Total Execution Time: 0.0076150894165039 Secs

Figure 6.16 Interface UIF-06 - Cloud Service Ranking

The SST is deliberately designed in a concise manner such that it contains all the necessary pages that help consumers in selecting the cloud services. A design decision was taken to make sure that while the consumers are making use of the tool, they can reach a conclusion as quickly as possible. Therefore, the number of steps/stages (from the beginning to the end of the consumer journey) is kept to a minimum. In addition, the design journey for a Beginner consumer is different from the design journey of an Intermediate/Expert consumer. For instance, a Beginner consumer can complete a ranking in just 4 steps (or clicks) while an Intermediate consumer journey would take more steps. The diagram below shows the full consumer journey that can happen on the SST tool. However, Beginner consumers may not specify detailed criteria as that of Intermediate/Expert consumers when selecting the services.

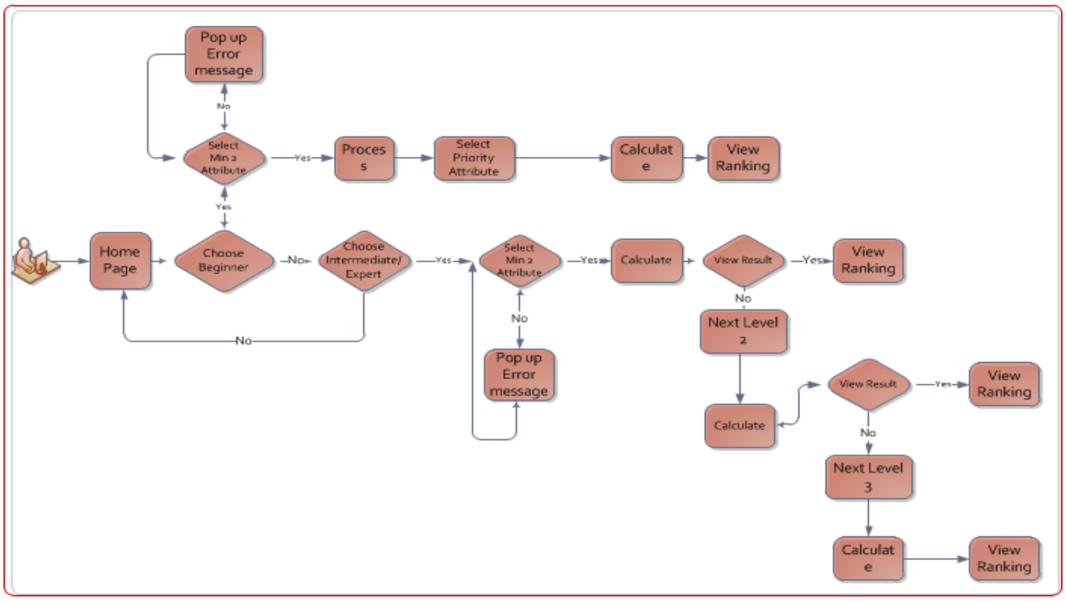


Figure 6.17 Possible Consumer Journeys on SST

6.5 Summary

This chapter has explained the implementation of the SST tool. The chapter highlighted the functional requirements of the SST tool. In order to simply the understanding of the process, various diagrams and flow charts have been designed in order to explain consumer journeys/steps involved in using the SST tool. The SST is flexible to accommodate various types of consumers based on their expertise and understanding of Quality of Service attributes. The consumer can either be a Beginner, Intermediate or an Expert consumer. The SST considers different levels of calculation for the different consumers. Depending on how experienced the consumer is, the SST can present consumers with various levels of details. This is an important feature of the SST which is not available in existing cloud service selection tools.

Further, screenshots of various SST interfaces (presented to consumers) are explained in this chapter. Furthermore, the tool provides facilities for different consumer journeys for different classes of consumers. The consumer can decide on how much detail they are willing to provide while using the tool. Overall, the SST tool was designed in such a way so that it is easier to use and understand by the consumers, whether they are Beginner, Intermediate or Expert consumers. In addition, the tool enables consumers to efficiently carry out the service ranking and selection. The next chapter will explain the various testing that was carried out on the SST.

CHAPTER 7

Testing and Evaluation of the SST

7.1 Introduction

This chapter describes the testing and evaluation of the SST. Black box testing method was used for testing the SST tool. The chapter explains the rationale behind the choice of black box testing. The various test cases used in testing the SST are analysed and the results derived from the test cases are explained. The chapter also presents different scenarios and shows the results based on various combination of attributed selected by a consumer.

There exist various research-based approaches and different commercial tools for cloud service selection, as described in chapter 2. But there is no standard evaluation benchmark for cloud services selection. Though the adoption of the AHP method in the proposed framework is in common with some of the existing approaches e.g. (Sun *et al*., 2013; Sun *et al.,* 2014), they consider fewer number of cloud providers and limited number of QoS attributes and do not consider the number of levels (or hierarchy) of QoS attributes that best represent cloud services. The proposed framework is therefore not compared with existing research-based approaches. Instead, the evaluation of the selection process of the proposed framework is carried as follows. First, a baseline method was used to analytically evaluate the proposed method using various numerical tests. Second, it was evaluated using the SST tool which is more efficient and which caters for simple as well as complex scenarios with different number of QoS attributes.

Finally, this chapter evaluates the SST by comparing it with related cloud service selection tools. The comparison reveals that SST has some novel features which are missing from the existing tools. The chapter concludes that the SST tool provides appropriate facilities for cloud service selection and it meets the main objectives set for this research.

7.2 Functionality Testing of the SST Tool

In general, software testing is used to test and demonstrate that the software performs its intended function. Testing is mainly executed with the intent of finding errors or faults in the software (Myers *et al*., 2004). Black box testing is one of the existing testing methodologies used for testing software. Black box testing is also known as functional testing. As the name denotes, black box or functional testing method is based mainly on testing that the software performs its expected function. Testing techniques and test cases are based on the information provided in the specification of the software (Nidhra and Dondeti, 2012). Black box testing is usually done for finished application. In the case of SST, black box testing was the most appropriate testing method as the aim was to ensure that the system behaves the way it is expected to and provides the consumer with the right results. Black box testing is not mainly concerned with the contents of the codes but rather the input and output of the program (Jorgensen, 2002). Since the SST will expect inputs from the consumer, it is important to apply black box testing techniques to prove that the system behaves according to specification, by providing the right results based on consumers input.

The next section looks at various test cases on the SST. The test cases are based on the use cases described in section 6.2.3.

7.2.1 Black Box Testing Use Case UC-01: Choose Level of Knowledge

<u>Test: Consumer Can Choose Level of Knowledge (Test Case ID TC-01-001)</u>

This test case (TC-01-001) is used to test use case UC-01. Use Case UC-01 allows the consumer to choose his/her level of knowledge. The software has two main levels which include Beginner and Intermediate/Expert level. It expected that consumers must choose only one level of knowledge and be able to proceed to the next page.

The expected result of this is that when the consumer chooses a level of knowledge, they are able to progress to the next page without any error. The next page is supposed to present the consumer with the seven attributes where they can choose the attributes they want to check.

Result

This test was carried out twice on both options, i.e., the consumer chose Beginner level first and then chose Intermediate/Expert level of knowledge. For both options, the consumer was able to be directed to the next page without any errors. The SST performed as was expected and when the ‘next’ button was clicked, the SST presented the consumer with the attributes page.

Remark

The expected result was achieved and the system functioned according to the specification.

<u>Test: Consumer MUST Choose Level of Knowledge (Test Case ID TC-01-002)</u>

This test case (TC-01-002) is used as a follow up to test case TC-01-001. It is expected that consumers who use the tool must have to choose a level of knowledge. This is not an optional function as the SST needs to use this input from the consumers to provide ranking. The tool must have a way to check that consumers have chosen a level of knowledge. To test this, no option was selected and the tester clicked on the ‘next’ button.

The expected result is that if a consumer does not choose either of the available options, the system is should prevent the consumer from progressing to the next page on the tool.

Result

When the consumer did not choose any option and clicked on ‘Next’ on the home page (UIF-01), the SST did not progress to the next page. It kept redirecting the consumer back to the home page.

Remark

The expected result was achieved. However, the SST did not display any error message to the consumer which may make it difficult for them to know what the error was.

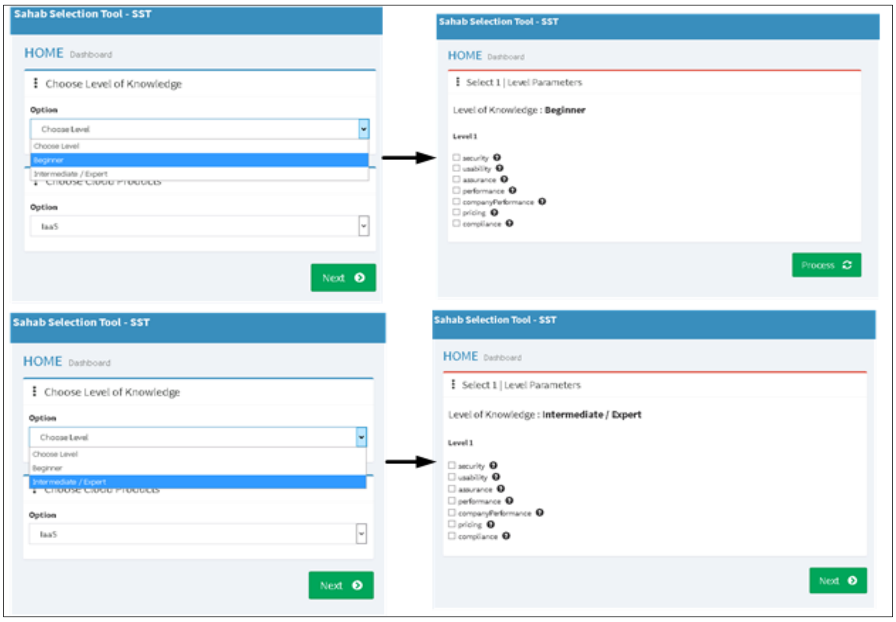


Figure 7.1 TC-01-001 Result Screenshot

7.2.2 Black Box Testing Use Case UC-02: Show Attribute Description

<u>Test: Consumer Can View Description of Attributes (Test Case ID TC-02-001)</u>

This test case is developed from use case UC-02. The use case states that consumer can choose to see a brief description of each of the attributes presented to the consumer on user interface UIF-02. Therefore, the page must have the capability to give consumers more information on the attributes when consumers need assistance with attribute’s definition. This is especially important as SST is designed to support consumers who varying knowledge on cloud services.

The expectation is that when the consumer hovers around each of the attribute, they should be presented with the description of the attributes. The attributes invloved here are the level one attributes (main). The tool must be able to show the description for all the attributes in a way that will be very clear for the consumer to see. The tool must also ensure that only one attribute definition is shown when the consumers mouse is over the attribute.

Result

When the mouse pointer is placed on each of the attribute, a brief description of each attribute is shown. The description is brief and easy for the consumer to read. The Figure 7.2 below shows the result of the test.

Remark

The expected result was the same as the achieved result. The SST is able to perform this function for all the attributes being displayed. Use case UC-02 has been properly implemented.

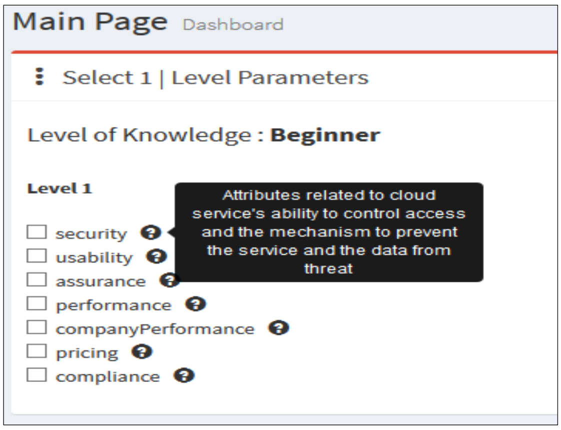


Figure 7.2 TC-02-001 Result Screenshot

7.2.3 Black Box Testing Use Case UC-03: Choose Main Level Attribute

<u>**Test: Consumer Can Choose Main Attributes (Test Case ID TC-03-001)**</u>

Test case TC-03-001 is developed to test use case UC-03. In use case UC-03, the consumer must be able to choose as many level one attributes as possible. These are the level one attributes that the consumer would like to use the SST to rank the cloud service providers on. These attributes must have been displayed to the consumer from the previous use case.

It is expected that consumers can choose multiple attributes. The system uses a check box to give consumers the flexibility of choosing as many options as they want. Consumers are expected to choose a minimum of two attributes. Once the attributes have been chosen, the consumer can proceed to click on the 'next' button. It is expected that once conditions have been satisfied, the consumers should progress seamlessly to the next page which should present them with a pairwise comparison of the attributes.

Result

The SST allowed the consumer to choose as many attributes as they wanted to. The checkbox ensured that consumers could both select and de-select attributes as they deemed fit. Also, after selecting attributes, the consumer was able to progress to the next page without any issue.

Remark

This test case demonstrated that the SST achieved the expected function. Therefore, the expected result was achieved.

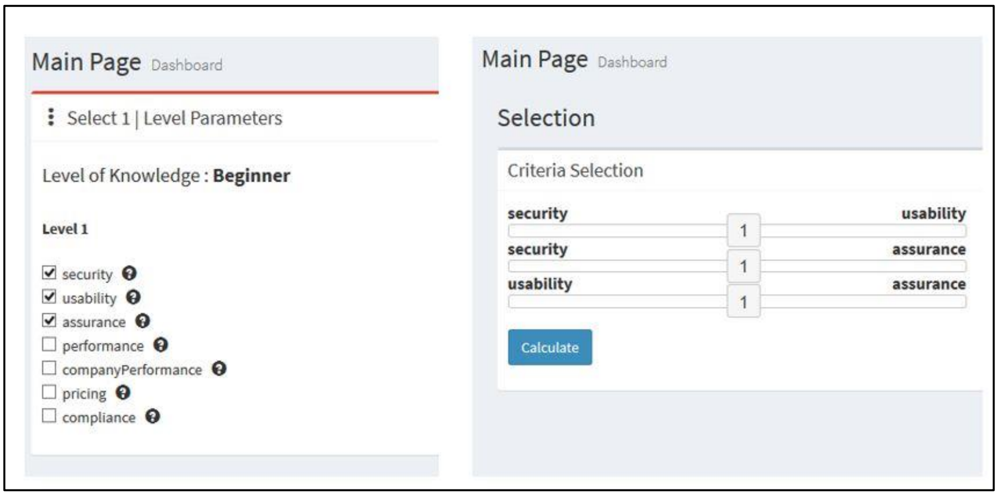


Figure 7.3 TC-03-001 Result Screenshot

<u>Test: Where Consumer Choose less than Two Attributes (Test Case ID TC-03-002)</u>

Test case TC-03-002 is a continuation from test case TC-03-001. In Test case TC-03-001, the consumer is invited to select multiple attributes. However, there is a condition that the consumer must select a minimum of two attributes. Therefore, this test is to check that before the consumer progresses to the next page, he/she must have selected a minimum of two attributes. To run this test, the consumer selects less than two attributes and clicks on ‘next’ to progress to the next stage.

The expected behavior is that the system should not progress to the next page when they click on next. In addition, there should be an error message displayed to the consumer notifying them the reason why the SST is unable to progress to the next page.

Result

When the consumer selected less than two attributes, and clicks on next, the system pops up an error (alert) box with a message. The message instructs the consumer to select a minimum of two attributes to be ranked by the SST. The alert box does not allow the consumer to progress on the SST journey until they click on 'OK' in the alert box. This is to ensure that the consumer has read the error message before giving them the opportunity to retry.

Remark

The expected result was achieved. The SST behaved as was expected. The error message is very clear to the consumer.

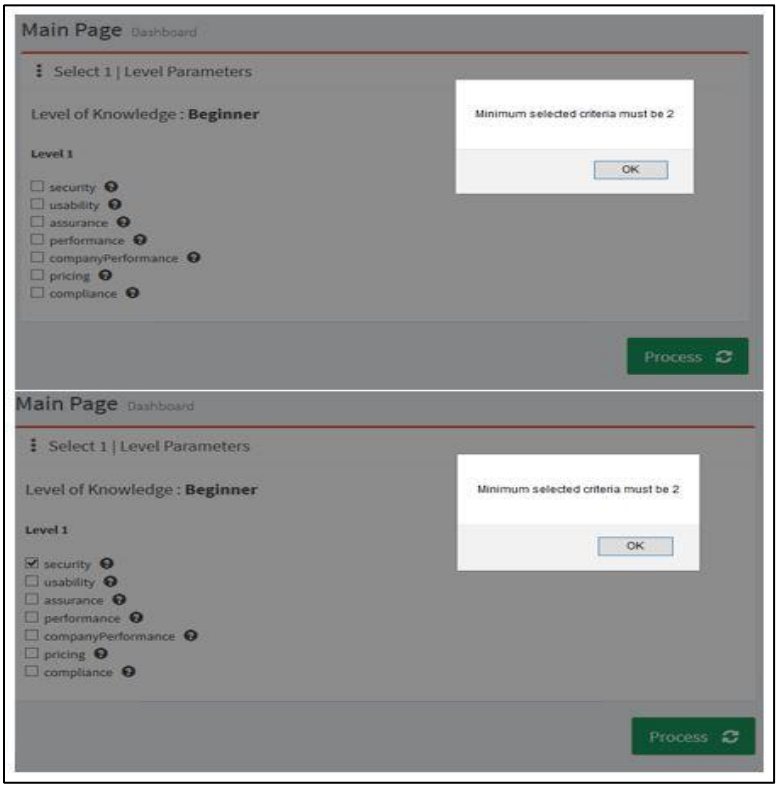


Figure 7.4 TC-03-002 Result Screenshot

7.2.4 Black Box Testing Use Case UC-04: Beginner-level Pairwise

<u>Test: Beginner Can Do Pairwise Comparison (TC-04-001)</u>

This test case is used to confirm that use case UC-04 works as expected. The use case UC-04 expects that consumers must be able to do a pairwise comparison based on the attributes they have chosen in the previous page UIF-02. In addition, the tool should allow the consumers to determine which attributes are more of a priority to them. Their preference must be recorded and used by the SST calculation to provide ranking.

The expected result is that the SST should be able to provide an interface where consumers can see all the attributes they have chosen from the previous page. In addition, they should be able to determine which attributes are more important to them.

Result

The results chosen by the consumer in the previous page were presented in pairs to the consumer. Between each pair of result, the SST presented the consumer with a slider function. The consumer is able to slide the button either left or right according to their preference on the scale. The SST stores the preference of each consumer as can be seen in the Figure 7.5. Also, only the attributes that was chosen by the consumers in the previous page appears on this page.

Remark

The expected result was achieved. The SST allows consumers to input and store their priorities.

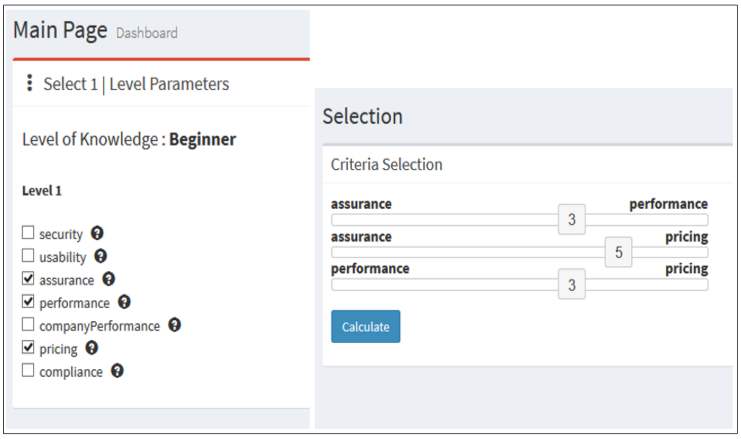


Figure 7.5 TC-04-001 Result Screenshot

7.2.5 Black Box Testing Use Case UC-05: Int/Exp-level Pairwise

Test: Intermediate/Expert Consumer Can Do Pairwise Comparison (TC-05-001)

This test case is similar to test case TC-04-001. The test case is based on use case UC-05 and allows Intermediate/Expert level consumers to perform a pairwise comparison. The previous page will allow the consumers to choose the attributes they want to measure and this page should display all the attributes selected by the consumers.

The expected result is that the SST should present all the attributes selected by the (Intermediate/Expert level) consumers and allow them to decide which attributes are more important to them. This preference should be stored and used by the SST for calculation.

Result

The SST presented the exact attributes selected by the consumer. The consumer is also able to see the slider between each pair of attributes and can adjust the slider towards the attributes they think is more important.

Remark

The SST behaved as expected when this test was run. The expected result was achieved as shown in Figure 7.6.

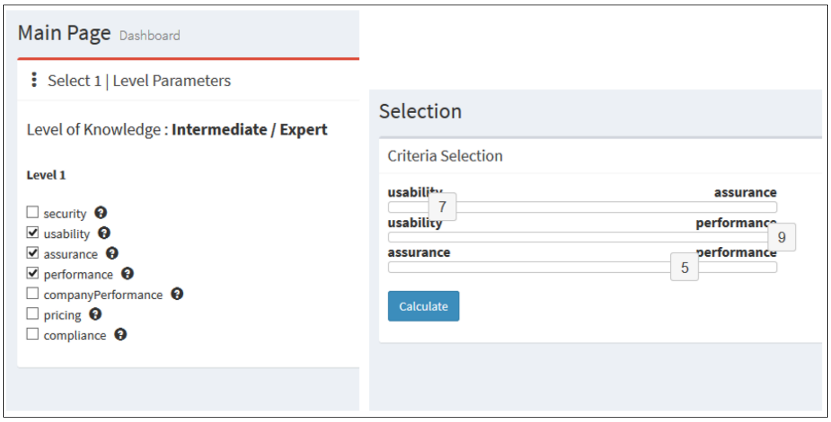


Figure 7.6 TC-05-001 Result Screenshot

7.2.6 Black Box Testing Use Case UC-06: Int/Exp-level Subparam Pairwise

<u>Test: Intermediate/Expert Consumer Can Do Level Two Pairwise Comparison (TC-06-001)</u>

This test is to check that Intermediate/Expert level consumers are able to perform pairwise comparison based on level two attributes. This function is performed in use case UC-06 and is restricted to Intermediate/Expert level consumers.

It is expected that when an Intermediate/Expert level consumers have selected their main level attribute, they must have the option to drill down into the level two attributes of their chosen level one attributes. The tool should display all the sub-attributes but only of the chosen main level attributes. In addition, consumers must be able to set their preferences among the level two attributes.

Result

The SST is able to perform a pairwise comparison of level two attributes. The user interface pops up with a message asking the consumers to proceed to the level two attributes after they have chosen the level one attributes. The advantage of SST is that the Intermediate/Expert level consumers are not forced to choose their preference for level two attributes. The consumer can choose to proceed to ranking without viewing the level two attributes. If the consumer chooses to proceed to level two attributes, the consumers are presented with a list of all paired level two attributes, which are grouped according to their level one attributes. This is displayed in user interface UIF-04. The consumers can also choose the attributes that are of a priority to them using the slider. As in the previous case, the slider is placed between the pair of attributes.

Remark

The SST has performed this function as expected and consumers can progress to next page.

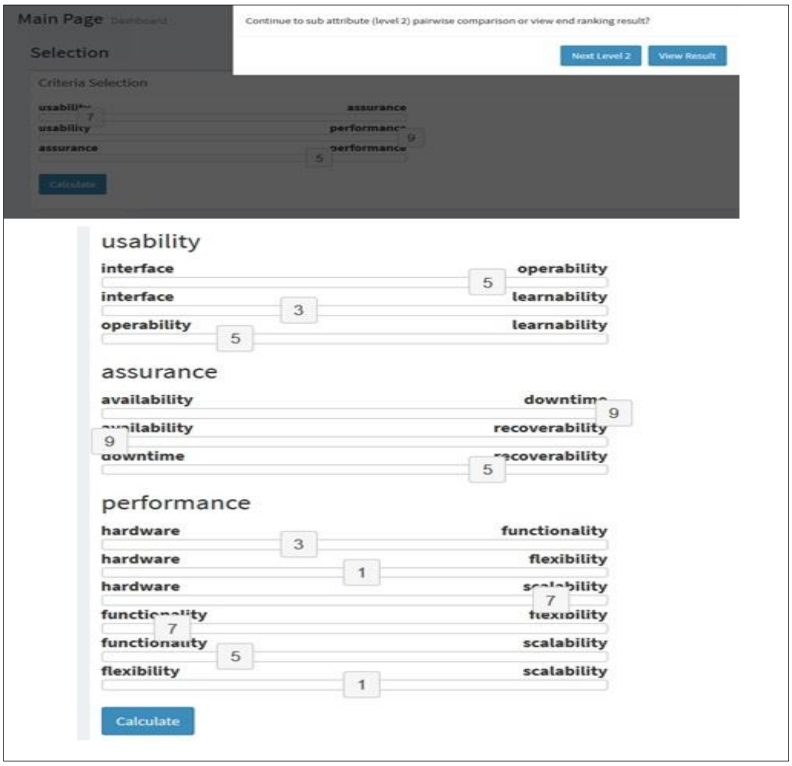


Figure 7.7 TC-06-001 Result Screenshot

7.2.7 Black Box Testing Use Case UC-07: Int/Exp-level Level 3 Filtering

Test: Intermediate/Expert Consumer Can Do Level Three Parameter Filtering (TC-07-001)

This test case is focused on use case UC-07 and is restricted to Intermediate/Expert level consumers. The use case allows the consumer to choose what level three attributes should be considered in the SST calculations and what attributes need to be filtered out.

It is expected that consumers will be presented with user interface UIF-05 containing level three attributes as soon they have done the pairwise comparison of level two attributes. The consumer can then choose the attributes that are to be filtered. Again, the level three attributes shown should be level three attributes that belong to the level one attributes which are chosen by the consumer.

Result

The SST is able to perform a filtering of the level three attributes. The user interface pops up with a message asking the consumer if they want to proceed to the level three attributes filtering. Again, the SST does not make it compulsory for consumers to use the level three filtering function (binary selection). Consumers can choose to view results based on their input on level two attributes (UIF-04). If the consumer chooses to proceed to level three filtering, the level three attributes are presented to the consumer with a set of check boxes (and radio buttons in some attributes) beside each attributes. Consumers can then click on any of the options on which they want to filter and can select values for level three attributes. The SST also displayed definition for each of the level three attributes presented to the consumer.

Remark

The expected result has been achieved. Figure 7.8 shows filtering of the attributes.

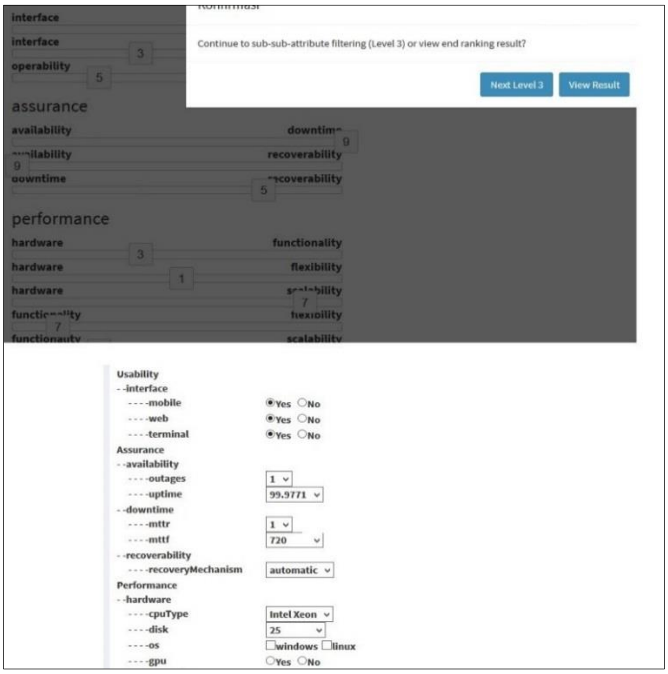


Figure 7.8 TC-07-001 Result Screenshot

7.2.8 Black Box Testing Use Case UC-08: Beginner-level Rank Calculation

Test: Present Beginner Consumer with Ranking (TC-08-001)

This test case checks that the SST is able to provide ranking to Beginner consumers. This function represents the final step of the ranking provided by the SST.

The expected behavior is that the consumers should be able to view the ranking of cloud services as provided by the SST.

Result

The SST provided ranking based on the input that the consumer provided. The SST is able to display the name of the provider, the value based on the AHP method and the ranking. In addition, consumers can search for a particular provider in the list, i.e., they do not need to scroll all the way to the page where the provider is listed. This function makes it easier for the consumer to find the ranking of a specific provider.

Remark

The SST achieved the expected output as shown in Figure 7.9.

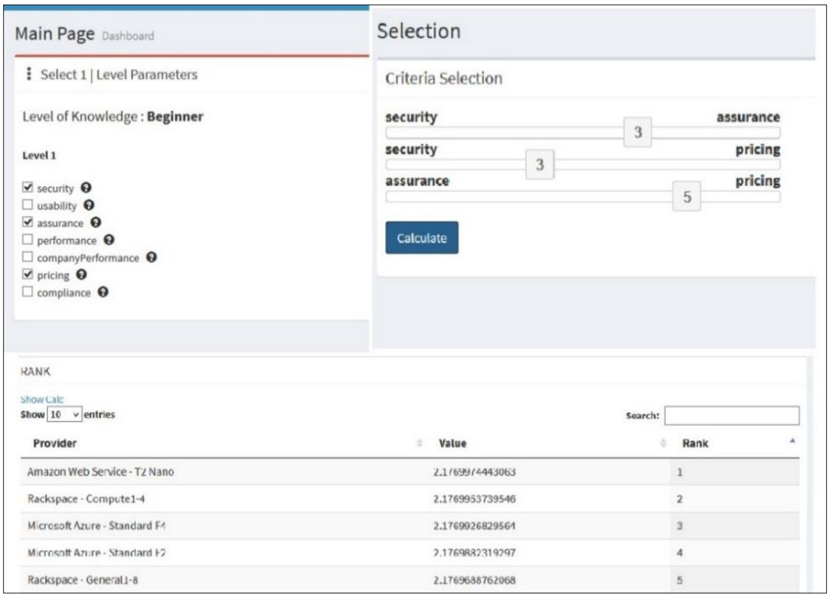


Figure 7.9 TC-08-001 Result Screenshot

7.2.9 Black Box Testing Use Case UC-09: Int/Exp-level Rank Calculation

Test: Present Intermediate/Expert level Consumer with Ranking (TC-09-001)

This test case checks that the SST is able to provide ranking to Intermediate/Expert level consumers. This function represents the final step of the ranking provided by the SST for Intermediate/Expert level consumers.

The expected behavior is that the consumers should be able to view the ranking of cloud services as provided by the SST.

Result

The SST provided ranking based on the input that the consumer provided. The SST is able to display the name of the provider, the value based on the AHP & SAW methods and the ranking. As with the Beginner level consumers, Intermediate/Expert level consumers can also search for a particular provider in the list.

Remark

The SST achieved the expected output. See Figure 7.10.

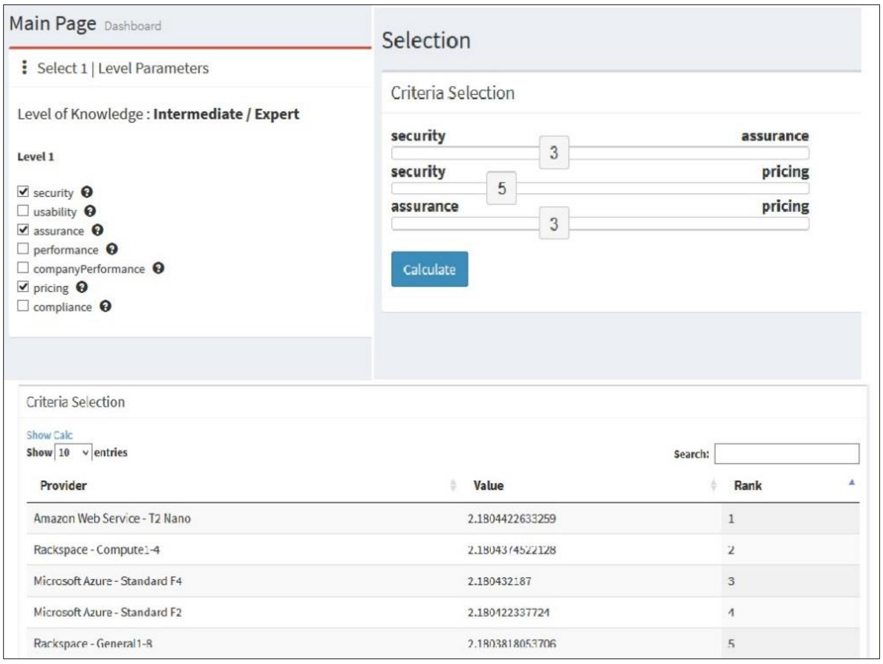


Figure 7.10 TC-09-001 Result Screenshot

7.2.10 Black Box Testing Use Case UC-10: Data Scraping

<u>Test: Testing the Data Scrapping Function (TC-10-001)</u>

This test case is based on use case UC-10. UC-10 expects that the SST must be able to gather data from multiple sources including cloud service providers websites. The tool must have access to the Internet to pull data from all these sources and write the data into the MongoDB database.

It is expected that when the tool scrapping script is run, it should perform this task seamlessly.

Result

The tool was able to pull data from all the variety of sources. The data was then parsed using python (Beautiful Soup and Pandas) libraries. The data is also converted into JSON format and was stored in MongoDB. The data can be viewed in the Figure 7.11.

Remark

The SST is able to scrap data and load into the database.

Figure 7.11 TC-10-001 Result Screenshot

7.2.11 Black Box Testing Use Case UC-11: Show SLA

<u>Test: Testing that SST Can Display SLA to Consumer (TC-11-001)</u>

In addition to the service ranking, the SST should be able to show the consumer a brief Service Level Agreement (SLA) document that is linked to consumer-selected cloud service provider. This is a requirement which is fulfilled by use case UC-11.

It is expected that the SLA document provided by each of the cloud providers should be visible to consumers who choose to view them.

Result

The consumer is able to view the SLA for any of the cloud providers which fall in the scope of the SST tool. When the SST displays the ranking to the consumer, there is a ‘preview’ button besides each cloud provider ranking under the SLA column. When the consumer clicked on the button, the SST displayed PDF file that contains the SLA guarantee provided by that particular cloud service provider. As mentioned in section 1.5, this feature is a novel feature of this tool. It is important for cloud service consumers to be able to view the SLA guarantees provided by the cloud providers, as part of the decision making process.

Remark

The SST achieved the expected result for this test as shown in Figure 7.12.

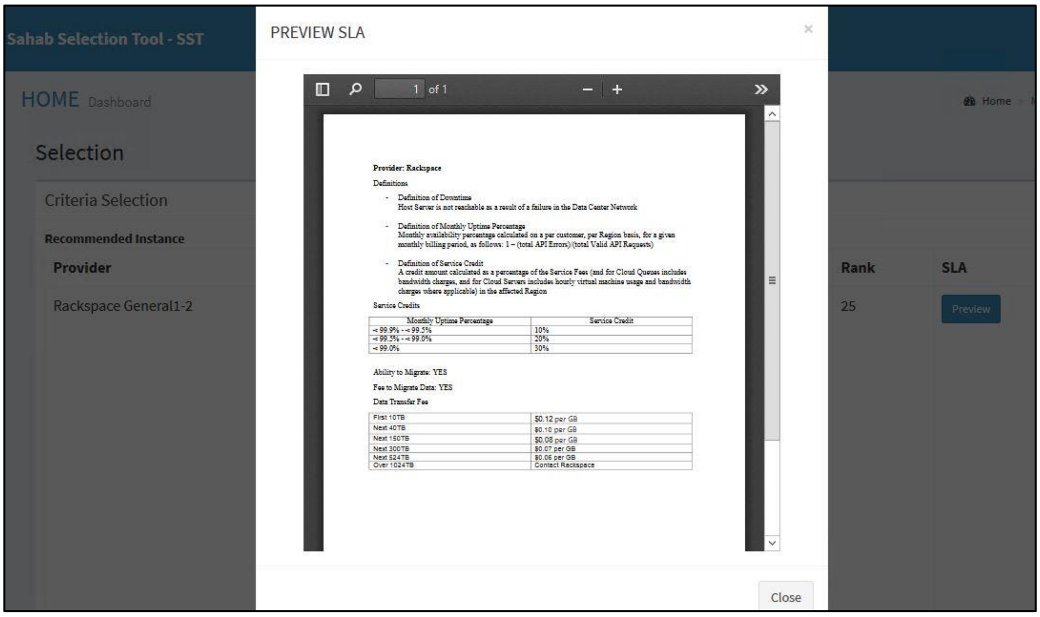


Figure 7.12 TC-11-001 Result Screenshot

7.3 Baseline Evaluation

Before implementing and evaluating the proposed methodology in the SST tool, it was first evaluated using analytical method with numerical tests. This was used as a baseline method in order to test the relative accuracy of proposed methodology (of AHP and SAW). It considers the scenario with the main attributes (security, usability and performance).The scenario used for this test will carry out calculation based on main level QoS attributes and sub-attributes. The QoS attributes chosen for this evaluation test include security, usability and performance. The scenario assumes both Beginner and Intermediate/Expert level consumer. The providers and instances been considered for this test is shown in Table 7.1 below.

Table 7.1 List of Providers for Numerical Calculation

No	Provider	Instance
1	IBM	BL1.1x2x100
2	Fujitsu	P-1
3	Oracle	BM.Standard2.52
4	Rackspace	General1-2

The numerical calculation is carried out from bottom (level three attributes) and aggregated them to the top (level one attributes). The first step is to apply the score mapping for each of the providers to their level three attributes values which are resulted from the scrapping. The next three tables (Tables 7.2, 7.3 and 7.4) show the scores of each of the provider based on the score mapping (see Appendix I).

Table 7.2 Cloud Providers' Score for Level Three Attributes of Security

	Authentication	Authorization	Encryption	Firewall	Provider's Nationality	Machine Location	Auditability
IBM	100	100	100	100	100	100	100
Fujitsu	100	100	100	100	100	100	100
Oracle	100	100	100	100	100	100	100
Rackspace	100	100	100	100	100	100	100

Table 7.3 Cloud Providers' Score for Level Three Attributes of Usability

	Web Interface	Mobile App	Terminal Access	Operability	Learnability
IBM	50	50	100	100	100
Fujitsu	50	50	100	100	100
Oracle	50	50	100	100	100
Rackspace	100	50	100	100	100

Table 7.4 Cloud Provider Score for Level Three Attributes of Performance

	Disk	OS	CPU Type	GPU	Network Performance	Memory Size	Number of CPU	Flexibility	Scalability
IBM	100	100	100	50	1	28	1	100	100
Fujitsu	1	100	100	50	1	1	1	100	100
Oracle	100	100	100	50	1	100	100	100	100
Rackspace	12.64706	100	100	50	100	28	100	100	100

The tables above (Tables 7.2, 7.3 and 7.4) show the providers' score for the level three attributes. The scores are then aggregated (through summation) and grouped according to their level two attributes as shown in the following tables (Tables 7.5 and 7.6)

Table 7.5 Score for Security Attributes Level 2

	Access Control	Data Security	Geography	Auditability
IBM	200	200	200	100
Fujitsu	200	200	200	100
Oracle	200	200	200	100
Rackspace	200	200	200	100

Table 7.6 Score for Usability and Performance Attributes Level 2

	Interface	Operability	Learnability	Hardware	Functionality	Flexibility	Scalability
IBM	200	100	100	350	30	100	100
Fujitsu	200	100	100	251	3	100	100
Oracle	200	100	100	350	201	100	100
Rackspace	250	100	100	262.65	228	100	100

Furthermore, the level two scores are also aggregated and grouped according to their top level attributes shown in Table 7.7

Table 7.7 Score for Security, Usability and Performance Attributes Level 1

	Security	Usability	Performance
IBM	700	400	580
Fujitsu	700	400	454
Oracle	700	400	751
Rackspace	700	450	690.65

For the AHP method, it is also required to calculate scoring for level 2 and level 1 attributes. The next step shows the pairwise comparison of these attributes. The assumption made in this test scenario is shown in Table 7.8. The attributes in the 'Preference / priority of consumer' column are the attributes that the consumer assigns a high priority. For instance, in the first row, the consumer made security of more priority than usability with a score of nine (9) on Security. Recall from the black box testing (Section 7.2) that the user interface of the SST presents users with the option of setting priorities when performing pairwise comparison. The second row in Table 7.8 shows that the consumer set a priority score of three (3) on Performance over Security. The rest of the table explains the priority choices made by the consumer in this test scenario.

Table 7.8 Pairwise Comparisons for Calculation Testing

Attribute 1	Attribute 2	Preference / Priority of Consumer
Security	Usability	Priority score of 9 on Security
Security	Performance	Priority score of 3 on Performance
Usability	Performance	Priority score of 5 on Performance
Access Control	Data Security	Priority score of 3 on Data Security
Access Control	Geography	Priority score of 3 on Access Control
Access Control	Auditability	Priority score of 5 on Access Control
Data Security	Geography	Priority score of 5 on Data Security
Data Security	Auditability	Priority score of 7 on Data Security
Geography	Auditability	Priority score of 3 on Geography
Interface	Operability	Priority score of 5 on Interface
Interface	Learnability	Priority score of 5 on Interface
Operability	Learnability	Priority score of 3 on Learnability
Hardware	Functionality	Priority score of 7 on Functionality
Hardware	Flexibility	Priority score of 5 on Flexibility
Hardware	Scalability	Priority score of 3 on Scalability
Functionality	Flexibility	Priority score of 3 on Functionality
Functionality	Scalability	Priority score of 5 on Functionality
Flexibility	Scalability	Priority score of 3 on Flexibility

The score is then calculated using the AHP formulas described in section 4.5.1. The table (Table 7.9) shows the outcome of the calculation and the ranking provided based on level 1 attributes. Recall that from the SST interface design (in section 6.4.1) and testing (section 7.2) that a Beginner level consumer will calculate the ranking based on level one attributes while Intermediate/Expert level consumers can choose to either calculate their ranking based on level one attributes, level two attributes or level three filtering.

Table 7.9 AHP 1-level calculation result

	Security	Usability	Performance	TOTAL	RANK	FINAL RANK
IBM	0.09	0.02	0.13	0.24	3	Oracle
Fujitsu	0.09	0.02	0.10	0.21	4	Rackspace
Oracle	0.09	0.02	0.17	0.28	1	IBM
Rackspace	0.09	0.02	0.16	0.27	2	Fujitsu

The table (Table 7.10) shows the result based on level two attributes. The result shows the ranking based on the analytical method and numerical tests.

Table 7.10 AHP, SAW 2-level calculation result

	TOTAL	RANK	FINAL RANK
IBM	124.28	3	Rackspace
Fujitsu	112.44	4	Oracle
Oracle	178.82	2	IBM
Rackspace	187.15	1	Fujitsu

It was observed that the baseline method could produce basic results using simple scenario of cloud services. However, using this method in complex scenario (of the AHP and SAW methodologies) will result in a cumbersome process that could take significant amount of time even for a combination over a few set of records. This also makes it incredibly difficult if not impossible for any consumer to perform this (ranking) task on the number of resources (virtual machines) in the SST repository. The SST repository contains attributes about one hundred and seventeen virtual machines (See Appendix II for the various configurations). Each of these (117) virtual machines has a value for each of the QoS attributes described in section 5.4.3 thus totalling to three thousand seven hundred and forty one (3,741) records in the repository. See Figure 7.13 below

```
> db.iaas.count()
3741
> _
```

Figure 7.13 Total Number of Records in the database

Furthermore, it was not possible to carry out the level three filtering using the baseline method.

7.4 Evaluation using SST

This section presents the evaluation of the proposed method using the SST tool which is more efficient and which caters for simple as well as complex scenarios with any number of QoS attributes. The aim is to evaluate the proposed method using the SST tool in order to verify that cloud services are appropriately ranked and selected.

In order to carry out the evaluation, various tests are carried out using different scenarios. The results for each of the scenarios are then explained and analysed.

In the first test case, a consumer chooses three attributes including security, usability and performance. In the second case, a consumer chooses three attributes including pricing, compliance and performance. Each of these scenarios is evaluated by taking into consideration different levels of consumer’s knowledge and different priorities for each of attributes. This is necessary as the SST tool calculates the score for the Beginner consumer differently than the Intermediate/Expert level consumer. In the case of the Beginner, score is calculated using only the AHP methodology while the Intermediate/Expert level consumer makes use of both AHP and SAW. Therefore, it is expected that both methods will produce different results for different scenarios. For clarity, Table 7.11 shows the five attributes and their sub-attributes that would be used in the evaluation.

Table 7.11 Sample QoS Data Attributes

Level 1	Level 2	Level 3
Security	Access Control	Authentication
		Authorization
	Data Security	Encryption
		Firewall
	Geography	Provider's Nationality
		Machine Location
	Auditability	Auditability
Usability	Interface	Web Interface
		Mobile App
		Terminal Access
	Operability	Operability
	Learnability	Learnability
Performance	Hardware	Disk
		OS
		CPU Type
		GPU
	Functionality	Network Performance
		Memory Amount
		Number of CPU
	Flexibility	Flexibility
	Scalability	Scalability
Pricing	Price	Windows
		Linux
	Charge Model	Pay as you go
		Contract
	Pricing Unit	Per hour
		Per minute
		Per GB
	Currency	Currency
	Support Fee	Free
		Additional Package
	Discounting	Discounting
	Pricing System	Tiered Pricing
		Volume Pricing
Compliance	Security Compliance	US Patriot Act
		FISMA
		Safe Harbor
	Legal Compliance	HIPAA

Level 1	Level 2	Level 3
		Sarbanes-Oxley Act (SOX)
	Standard Compliance	SSAE 16
		ISO 27001
		ISO 9001
		ISO 27017
		PCI DSS
		SOC 1
		SOC 2
		SOC 3

7.4.1 Evaluation Scenario – Beginner Consumers

In this scenario, a Beginner consumer decides to use the SST to rank cloud providers providing different attributes as priority for test case one (security, usability and performance) and test case two (pricing, compliance and performance). As this is a Beginner level consumer, the program performs a pairwise comparison of level one attributes only using the AHP methodology.

7.4.1.1 Test Scenario One Evaluation with No Priority

In the first evaluation, the consumer uses the SST to rank the cloud providers based on security, usability and performance. The consumer has no preference or priority among the three attributes. See Figure 7.14 below.

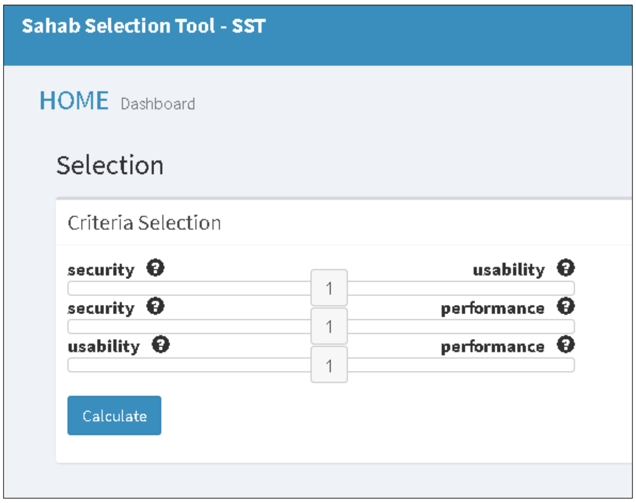
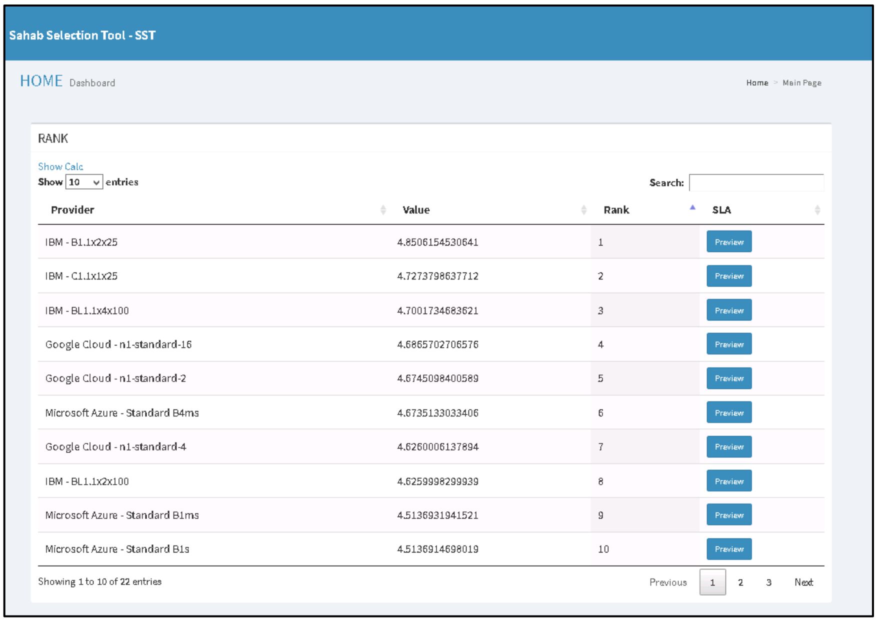


Figure 7.14 Consumer Level One Attributes – No Priorities

As can be seen in the Figure 7.14, the consumer has no priorities, i.e., the consumer did not select a priority attribute. The Figure 7.15 below shows the result calculated by the SST. For ease of readability, only the top ten (10) results are shown.



Provider	Value	Rank	SLA
IBM - B1.1x2x25	4.8506154530641	1	Preview
IBM - C1.1x1x25	4.7273798637712	2	Preview
IBM - BL1.1x4x100	4.7001734683621	3	Preview
Google Cloud - n1-standard-16	4.6865702706576	4	Preview
Google Cloud - n1-standard-2	4.6745098400589	5	Preview
Microsoft Azure - Standard B4ms	4.6735133033406	6	Preview
Google Cloud - n1-standard-4	4.6260006137894	7	Preview
IBM - BL1.1x2x100	4.6259998299939	8	Preview
Microsoft Azure - Standard B1ms	4.5136931941521	9	Preview
Microsoft Azure - Standard B1s	4.5136914698019	10	Preview

Figure 7.15 SST Ranking – No Priority Attribute

The result above shows that the IBM (B1.1x2x25) virtual machine has the highest ranking based on security, usability and performance (where the consumer did not input any preferences). This is followed by the IBM (C1.1x1x25) and the IBM (BL1.1x4x100). In the next test, the consumer uses the SST to calculate the ranking with the same attributes but changes the priority scores in the pairwise comparison.

7.4.1.2 Test Scenario One Evaluation with Priority

In this test, the consumer has priority attributes of usability and performance and is not too concerned with security. However, the consumer also prioritises usability against performance. For instance, a consumer may have staff members in his/her organisation who are managing an online application that does not contain sensitive information and are not necessarily very experienced in cloud infrastructure management. Therefore, usability and performance are priority for such consumer over security. The consumer therefore awards a priority score of 9 to both usability and performance against security and a priority score of 5 to usability against performance. See Figure 7.16 below.

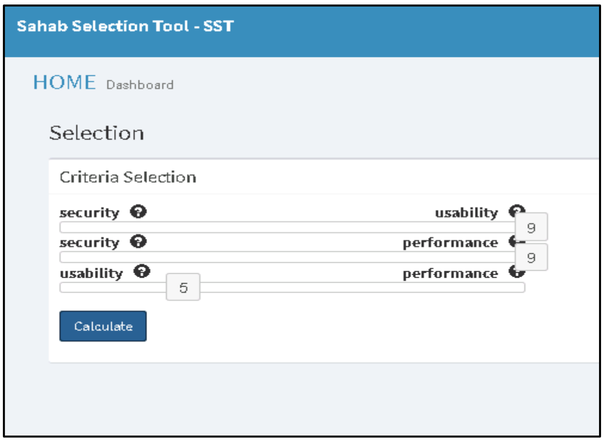


Figure 7.16 Consumer Level One Attributes – with Priorities

The consumer then uses the SST to calculate the ranking based on these priorities as shown above in Figure 7.16. A subset of the results of the ranking is shown below in Figure 7.17.

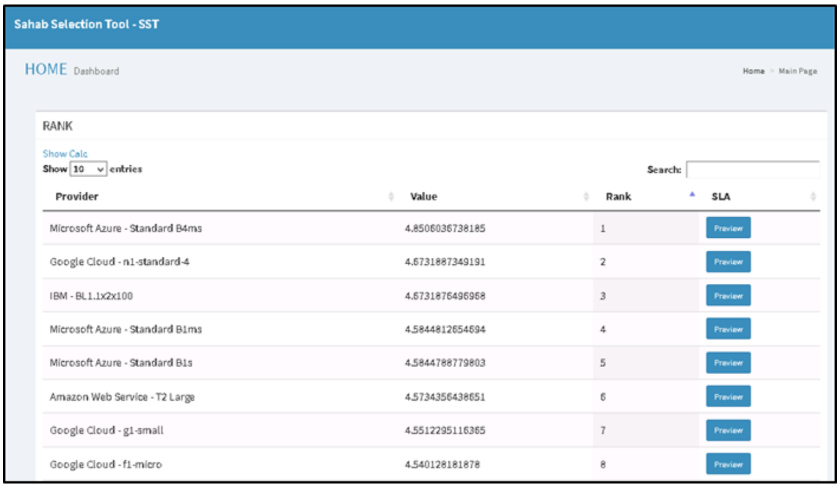


Figure 7.17 SST Ranking – (With Priority Attributes)

The result above (ranking) shows that based on the priorities described in Figure 7.16, the Microsoft Azure (Standard B4ms) has the highest ranking followed by the Google Cloud (-n1-standard-4).The interpretation of this ranking is that when it comes to usability and performance being two priority attributes, Microsoft Azure server is more reliable.

7.4.1.3 Test Scenario Two Evaluation with No Priority

This test scenario will evaluate the Beginner level ranking of the SST based on pricing, compliance and performance. The first test assumes that the consumer has no priorities.

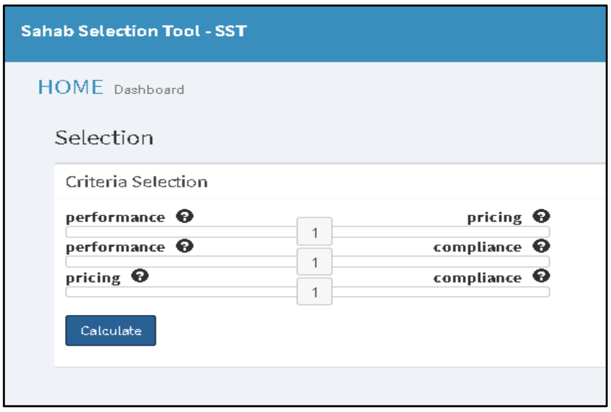


Figure 7.18 Consumer Level One Attributes 2 – No Priorities

The pairwise comparison, in Figure 7.18 above shows that the consumer has no preferred attributes. The result in Figure 7.19 shows that the IBM (C1.1x1x25) virtual machine has the highest ranking based on performance, pricing and compliance. This is followed by the IBM (BL1.1x4x100) and the Google Cloud (n1-standard-16).

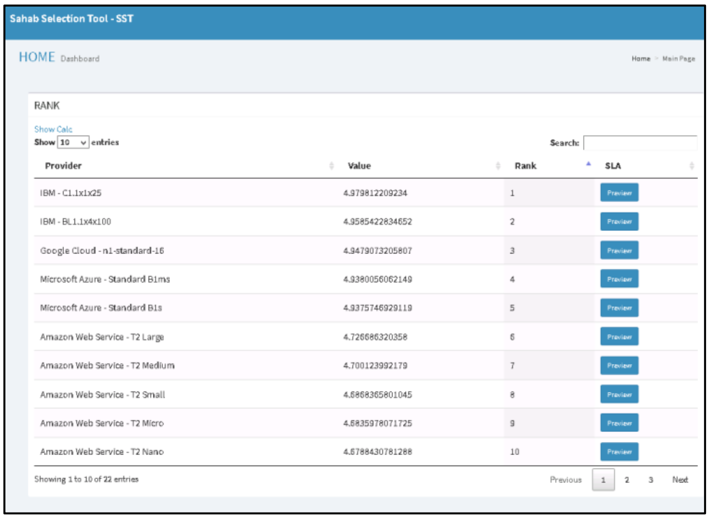


Figure 7.19 SST Ranking 2 – No Priorities

The next test assumes that the consumer has specified priority attributes.

7.4.1.4 Test Scenario Two Evaluation with Priority

In this case, the consumer prioritises pricing and compliance over performance. For instance, a consumer may have sensitive data that may not be accessed frequently and therefore performance may not be an issue, however, compliance and cost (pricing) could be of more importance to the consumer. Therefore in the pairwise comparison, the consumer assigns a priority of 9 to both pricing and compliance against performance. Furthermore, the consumer assigns a priority of 5 to pricing against compliance. See Figure 7.20 below.

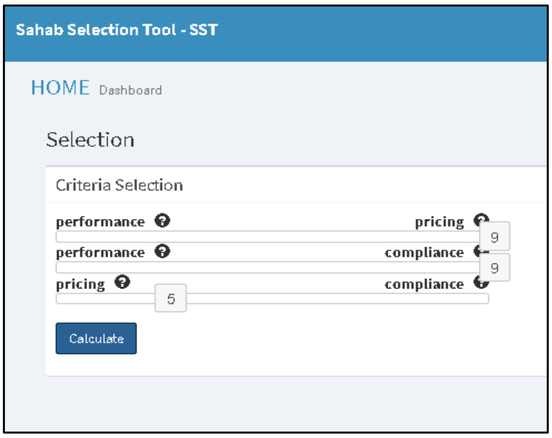
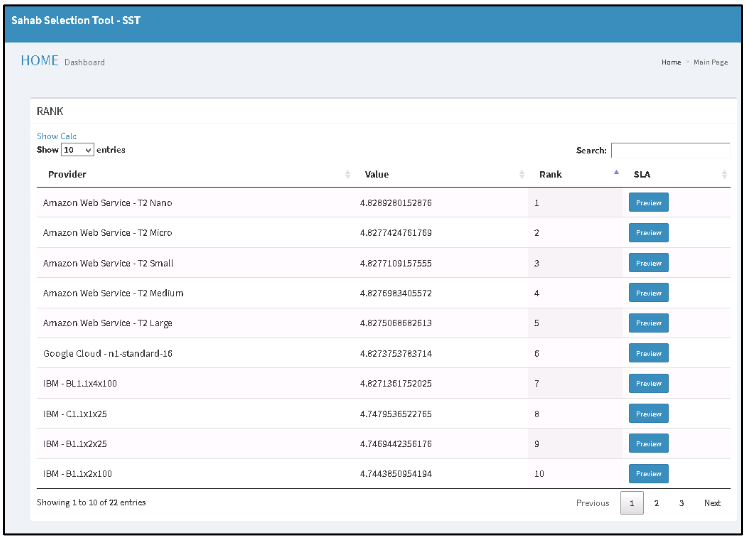


Figure 7.20 Consumer Level One Attributes 2– With Priorities

The Figure 7.20 above shows the consumers preference. The SST result ranking is shown below (Figure 7.21). The result shows that Amazon Web Service virtual machines are highly ranked when pricing and compliance are the priority. The Amazon Web Service (T2 Nano) tops the ranking. Google Cloud (n1-standard-16) and IBM virtual machines follow on the ranking behind the Amazon Web Service virtual machines. See ranking below (Fig. 7.21).



Provider	Value	Rank	SLA
Amazon Web Service - T2 Nano	4.8289280152876	1	Preview
Amazon Web Service - T2 Micro	4.8277424761769	2	Preview
Amazon Web Service - T2 Small	4.8277109157555	3	Preview
Amazon Web Service - T2 Medium	4.8276983405572	4	Preview
Amazon Web Service - T2 Large	4.8275068682613	5	Preview
Google Cloud - n1-standard-16	4.8273753783714	6	Preview
IBM - BL1.1x4x100	4.8271361752025	7	Preview
IBM - C1.1x1x25	4.7479536522765	8	Preview
IBM - B1.1x2x25	4.7469442356176	9	Preview
IBM - B1.1x2x100	4.7443850954194	10	Preview

Figure 7.21 SST Ranking 2 – (With Attributes)

Note that in the ranking above, the Amazon Web Service T2 virtual machines are top in the rankings. This is because the consumer prioritised pricing over compliance and performance. The Amazon Web Service T2 virtual machines are known to be the cheapest virtual machines. The T2-Nano virtual machines fall under free tier category.

7.4.2 Evaluation Scenario – Intermediate / Expert Consumers

This set of tests evaluates the scenarios described in section 7.4. In the first scenarios, a consumer chooses three attributes including security, usability and performance. In the second scenarios, the consumer chooses pricing, compliance and performance. However, the consumer in both tests is assumed to be an Intermediate/Expert level consumer. Recall that an Intermediate/Expert level consumer can choose to perform pairwise comparison on level one and level two attributes and can perform filtering on level three attributes. In addition, for these scenarios, the SST tool calculates the results based on the AHP and SAW methodologies, i.e., cumulative addition of sub attributes in an attribute.

7.4.2.1 Test Scenario One Evaluation with No Priority

This test evaluates a scenario where the consumer chooses security, usability and performance but has no priorities on both level one and level two. In addition, there is no level three filtering. See Figure 7.22.

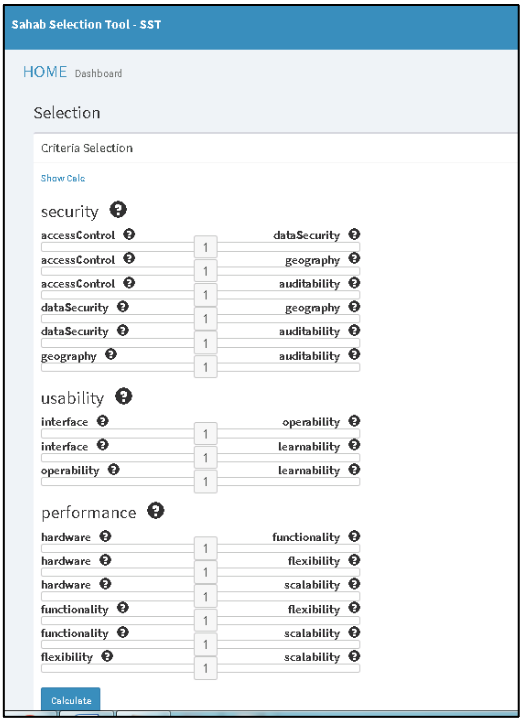


Figure 7.22 Consumer Level Two Attributes – No Priorities

The SST ranking is shown below in (Figure 7.23). The result shows that where the consumer did not input any priorities, the Google Cloud (n1-standard-96) virtual machine ranked highest based on security, usability and performance. In addition, the Microsoft Azure machine ranks immediately after the first three Google Cloud VMs.

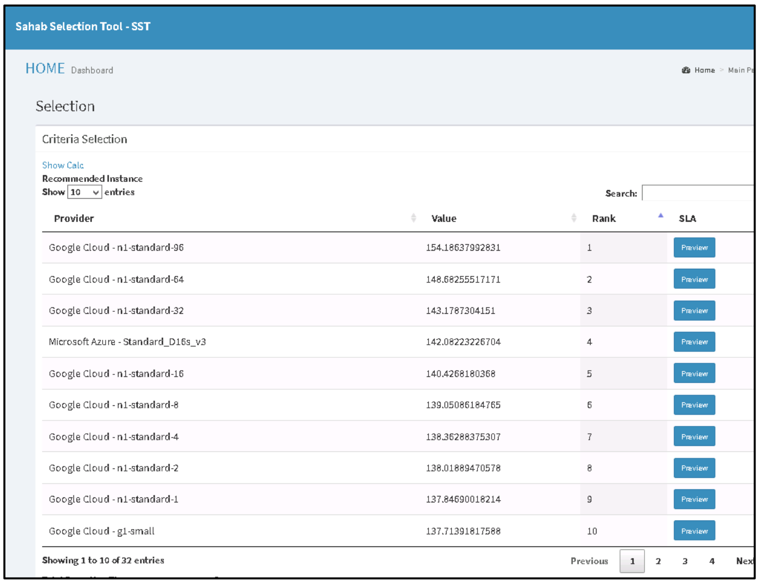


Figure 7.23 SST Ranking Level Two Attributes (No Priority)

The ranking shown above in Figure 7.23, reveals the Intermediate/Expert ranking is of the same level one attributes as in section 7.4.1.1 (Beginner Test Scenario – with no priority attribute). This Intermediate/Expert level calculation makes use of the SAW methodology. The result derived is consistent with the expected behaviour, i.e., when the Intermediate/Expert level consumer does not select any priority attribute (or filtering). The SAW methodology performs an addition of the sub-attributes of a level one attribute. Therefore, the level one attributes that have higher numbers of sub-attributes tend to have more priorities since their sum adds up to be greater than attributes with fewer sub-attributes. Generally, it is expected that Intermediate/Expert level consumers should have level two attributes that are of priority to them in order to get accurate results that meet their requirements.

7.4.2.2 Test Scenario One Evaluation with Level Two Priority

In this test, the consumer uses the same attributes (security, usability and performance) and sets priorities on level one attributes similar to the one in section 7.4.1.2, i.e., awards a priority score of 9 to both usability and performance against security and a priority score of 5 to usability against performance. In addition to this, the consumer sets level two attribute priorities based on a set of conditions.

Security:

The consumer has certain level two security attributes priorities. The consumer prioritises access control over geographical location, i.e., the consumer believes access control features are more important than the geographic location of the server. The consumer prioritises data security over geographic location, access control and auditability. Finally the consumer prioritises auditability over geographic location.

Usability:

The consumer prioritises the operability of the cloud infrastructure over learnability. The consumer is more interested in how easy it is to learn to use the cloud services (learnability) as against the user interface. The consumer prioritises interface over operability.

Performance:

The consumer prioritises functionality, scalability and flexibility over the underlying hardware. The consumer prioritises flexibility over functionality. The consumer prioritises functionality and flexibility over scalability.

The Figure 7.24 below shows the consumer's priorities.

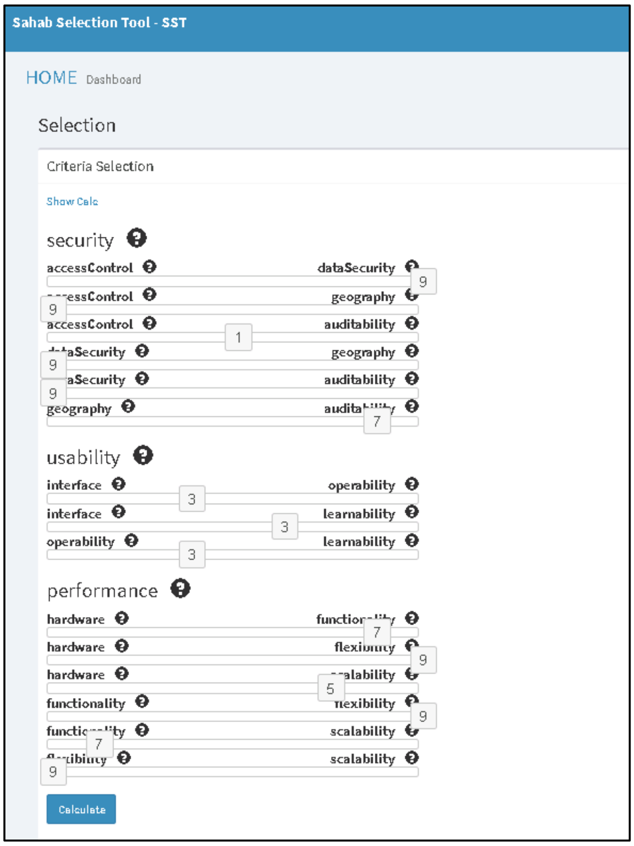


Figure 7.24 Consumer Level Two Attributes – With Priorities

The result from the consumer's priorities is shown below (See Figure 7.25).

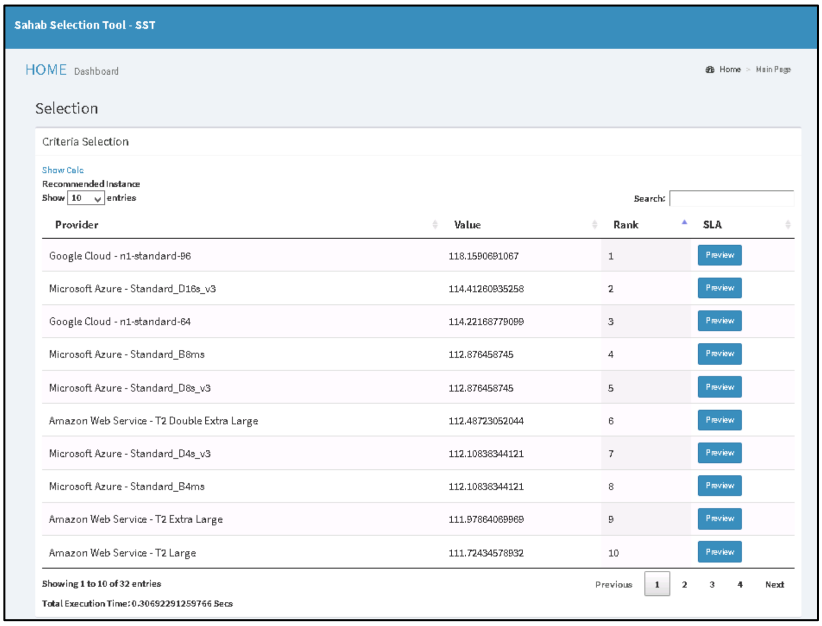


Figure 7.25 SST Ranking Level Two Attributes – With Priority

The result above in Figure 7.25 shows that the Google Cloud (n1-standard-96) topped the ranking followed by the Microsoft Azure (Standard_D16s_v3) and the Google Cloud (n1-standard-64). Recall that the test scenario here has the same level one attributes and priorities as in section 7.4.1.2, however, the rankings provided by the SST show different results. In section 7.4.1.2, the Microsoft Azure (Standard B4ms) topped the ranking. This is because the test in this scenario is an Intermediate/Expert level calculation and the consumer adjusted the priorities on the level two attributes. Therefore, the result is different from the Beginner level. Also, note that the smaller VMs do not rate among the top ranking VMs in this test. This is because the smaller machines do not necessarily meet with the criteria specified by the consumer in the level two attributes of security, usability and performance.

7.4.2.3 Test Scenario One Evaluation with Level Three Filtering

This test will use the same scenario as above. The consumer selected the attributes security, usability and performance. The level one priorities set are the same as the preferences selected in section 7.4.1.2 (i.e. a priority score of 9 is assigned to both usability and performance against security and a priority score of 5 to usability against performance). The consumer level two priorities are also the same as in section 7.4.2.2. The consumer then further performs level three as shown in Figure 7.26 below.

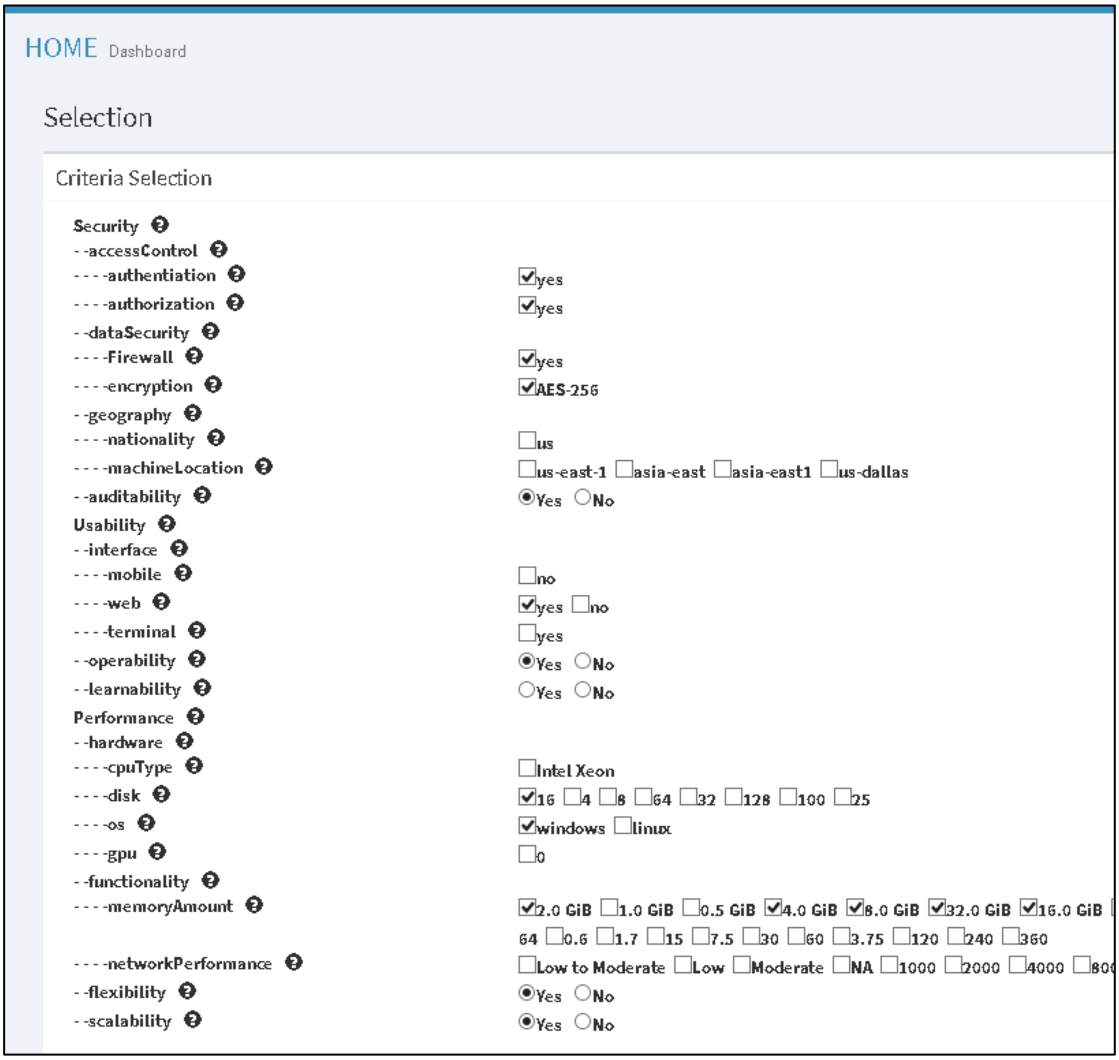


Figure 7.26 Consumer Level Three Filtering

In Figure 7.26, the consumer has made selection to some attributes. The consumer's preference is for virtual machines that support authentication, authorization and encryption (specifically AES-256). In addition, the consumer has only selected virtual machines that support windows operating system with memory of 2GB to 32GB. Most importantly, the consumer has filtered only disk size 16 GB. The ranking by the SST is shown below (Figure 7.27).

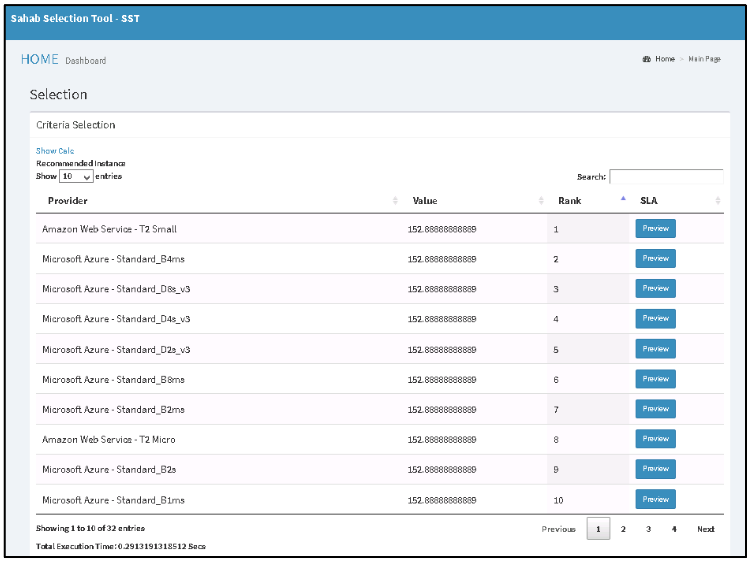


Figure 7.27 SST Ranking (Level Three Filtering)

The ranking above shows that the SST has filtered out some of the results derived in section 7.4.2.2. The ranking in section 7.4.2.2 shows that Google Cloud (n1-standard-96) topped the ranking followed by the Microsoft Azure (Standard_D16s_v3). However, even though the consumer selected the same level one and two priority combinations, some of the results have been filtered out. Therefore the Amazon Web Service T2 Small topped the ranking in this experiment. The comparison in the Figure 7.28 below from the data in the repository shows that the Google cloud (n1-standard-96) has a disk size of 64 GB and has therefore been filtered out of the ranking.

Figure 7.28 Comparison Google n1-standard-96 vs Amazon T2 Small

7.4.2.4 Test Scenario Two Evaluation with No Priorities

This test evaluates a scenario where the consumer chooses pricing, compliance and performance (from test case two in sections 7.4.1) but has no priorities on both level one and level two. In addition, there is no level three filtering. See below (Fig. 7.29)

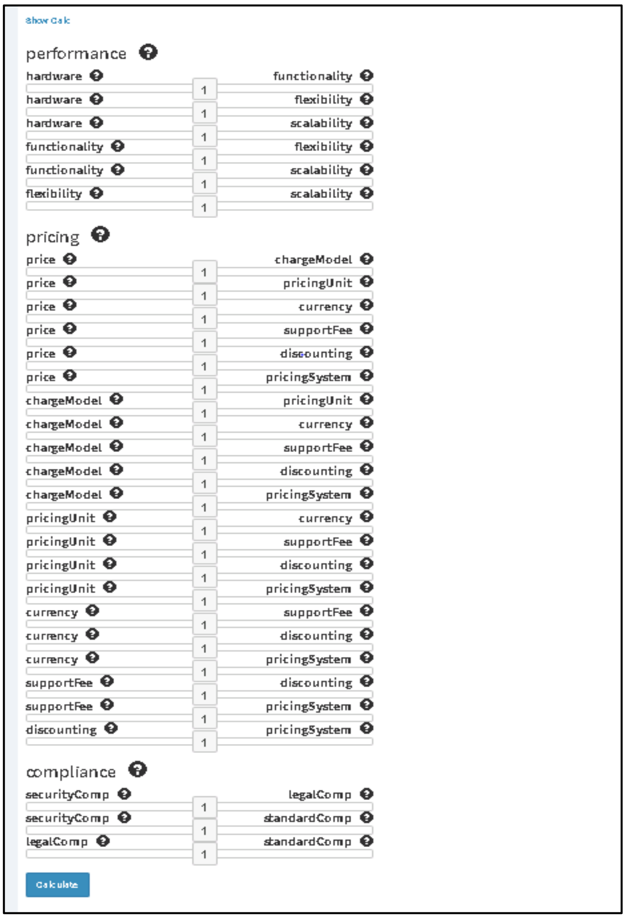


Figure 7.29 Consumer Level Two Attributes 2 – No Priorities

The Figure 7.29 shows that the consumer does not have any level two priorities. Again, as with section 7.4.2.1, it is expected that if the Intermediate/Expert level consumer does not make any level two priorities (with no filtering), the result or ranking would not necessarily be the same as a Beginner level consumer. This is because if the consumer added no priority weighting to the level two priorities, the level one attributes with more sub-attributes are expected to carry more weight. It is expected that Intermediate/Expert consumers should be able to identify attributes that are of importance to them.

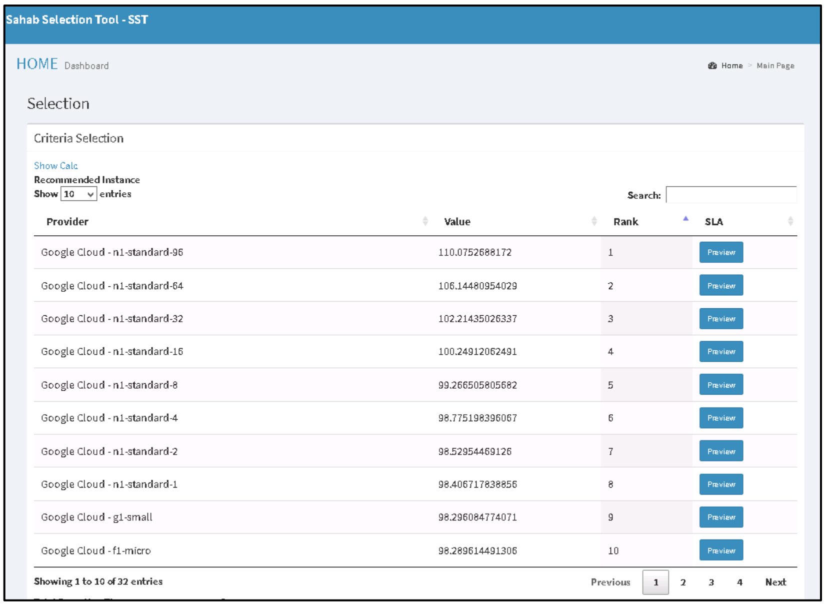


Figure 7.30 SST Ranking Level Two Attributes 2 (No Priority)

In section 7.4.1.3, the consumer also chose the same set of attributes (compliance, performance and pricing) with no priorities. The result showed that IBM (C1.1x1x25) topped the rank; however, with an Intermediate/Expert level calculation, the result is different. The Google Cloud (n1-standard-96) topped the ranking in this experiment (Figure 7.30). Also, comparing the result with section 7.4.2.1, where the consumer selected security, usability and performance, the results are somewhat similar with the Google Cloud VMs topping the rankings.

7.4.2.5 Test Scenario Two Evaluation with Level Two Priorities

In this test scenario the consumer uses the same priorities of performance, pricing and compliance. The consumer also sets their level one priority the same as the Beginner's level test in section 7.4.1.4, i.e., a priority of 9 on both pricing and compliance against performance. Furthermore, the consumer assigns a priority of 5 to pricing against compliance. The consumer, however, in this case (Intermediate/Expert level), sets their level two priorities based on the following conditions.

Performance:

The consumer prioritises functionality, scalability and flexibility over the underlying hardware. The consumer prioritises flexibility over functionality. The consumer prioritises functionality and flexibility over scalability.

Pricing:

The consumer sets price to be more important than the charge model, pricing unit, currency and support fee. The consumer sets discounting to be more important than price, pricing unit, currency, pricing system, charge model and support fee. The consumer sets charge model to be more important than currency, support fee and pricing unit. The consumer sets support fee to be more important than currency and pricing unit. The consumer sets pricing system to be more important than currency. In addition, consumer gives pricing system as the same level of importance as the charge model, pricing unit and support fee.

Compliance:

The consumer sets security compliance to be more important than legal and standard compliance. The consumer sets legal compliance to be more important than standard compliance. See Figure 7.31 below.

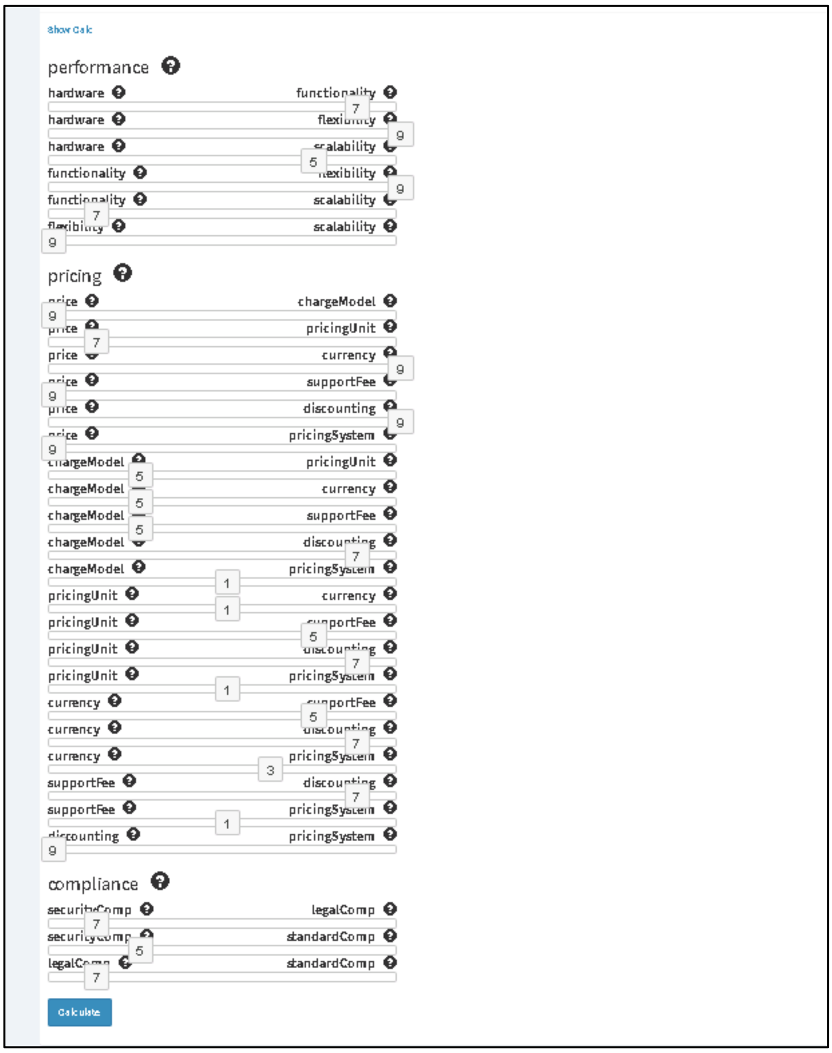


Figure 7.31 Consumer Level Two Attributes 2 – With Priorities

The ranking for the above conditions are shown in Figure 7.32.

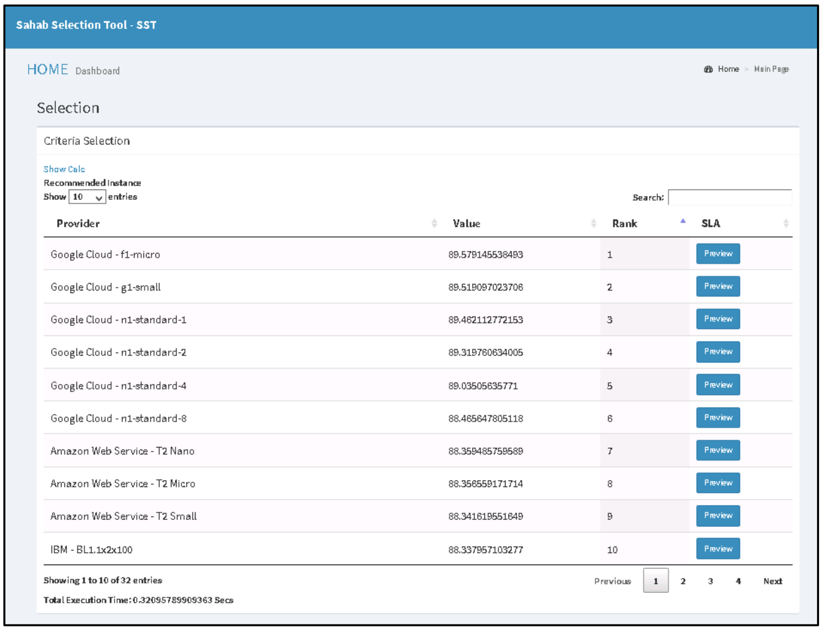


Figure 7.32 SST Ranking Level Two Attributes 2 (With Priority)

The result shows that the Google Cloud (f1-micro) topped the ranking. The Amazon Web Service T2 Virtual machines also followed in the ranking after the Google Cloud VMs. Note that the consumer set pricing and discounting as priority level two attributes. Therefore, it is expected that the smaller VMs will be the cheapest. Comparing with the ranking in section 7.4.2.2, it can be seen that the consumer ranked cloud providers based on security, usability and performance. Therefore, the virtual machines with higher ranking in section 7.4.2.2 are relatively bigger machines than the machines in this ranking – which is ranked based on pricing, performance and compliance. Furthermore, comparing the ranking of this experiment with the ranking in section 7.4.1.4 (which also has priority attributes as pricing, performance and compliance); the Amazon Web Service virtual machines are ranked higher than the Google Cloud VMs. The rankings are different because the consumer in this test case has performed a deeper (Intermediate/Expert) level of analysis.

7.4.2.6 Test Scenario Two Evaluation with Level Three Filtering

This test also uses the same scenario in section 7.4.2.5 above. The consumer selected the attributes performance, pricing and compliance. The level one priority is the same as the preferences selected in section 7.4.1.4. That is, a priority of 9 was assigned to both pricing and compliance against performance. Furthermore the consumer assigns a priority of 5 to pricing against compliance. The consumer level two priorities are also the same as the selection shown in section 7.4.2.5. The consumer then further performs level three as shown in Figure 7.33 below.

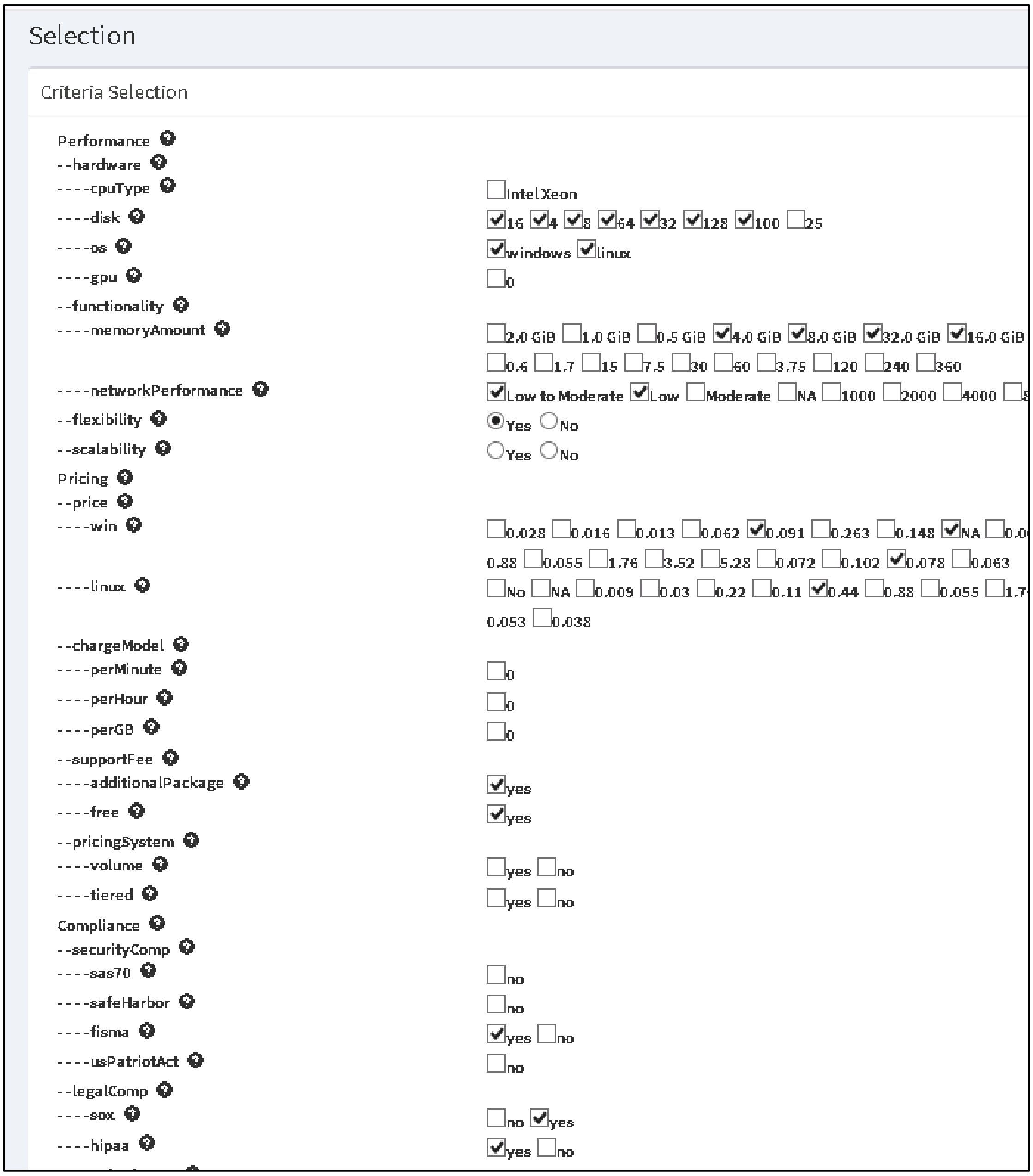


Figure 7.33 Consumer Level Three Filtering – 2

The result from the level three filtering above (Fig. 7.33) is shown in the figure below (Fig. 7.34).

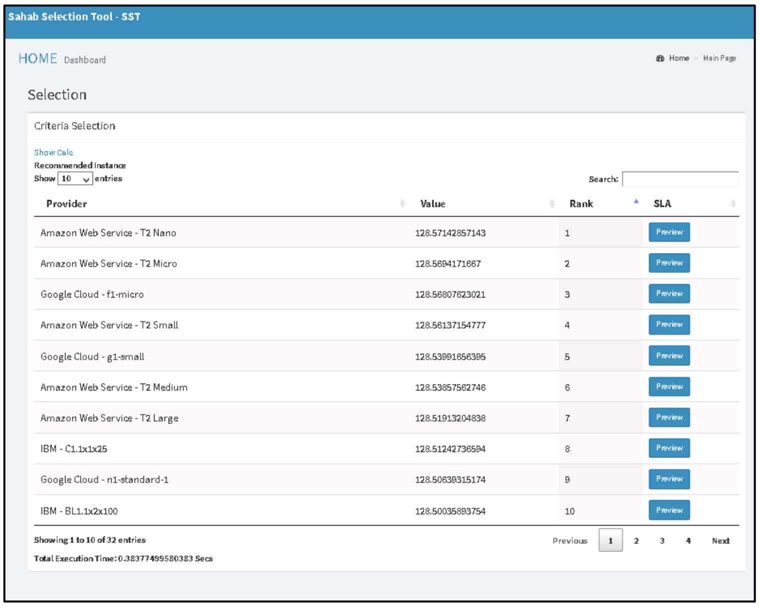


Figure 7.34 SST Ranking (Level Three Filtering) – 2

From the result shown above, it can be seen that the level three filtering is applied and then ranking is provided using SST. This test scenario has the same attributes and priorities (level one and two) as the test in section 7.4.2.5. The result in Figure 7.32 showed that Google Cloud (f1-micro) topped the ranking. In addition, all the top 5 systems are Google Cloud virtual machines. However, when the consumer applied the level three filtering, most of the Google Cloud machines have been filtered out of the top results. The consumer has chosen to filter only machines that are FISMA compliant. The comparison in the Figure 7.35 below from the data in the repository shows that the Google cloud (f1-micro) has been filtered out of the ranking because it is not FISMA Compliant.

```
{
  "_id" : ObjectId("5b3ec735e02f711d3feab7c4"),
  "instanceName" : "f1-micro",
  "vendorName" : "Google Cloud",
  "assurance" : {
    "recoverability" : {
      "recoveryMechanism" : "automatic"
    },
    "availability" : {
      "uptime" : "100",
      "outages" : "0"
    },
    "downtime" : {
      "mttr" : 0,
      "mttf" : 720
    }
    "compliances" : {
    "standardComp" : {
      "iso9001" : "no",
      "soc3" : "yes",
      "soc1" : "yes",
      "iso27017" : "yes",
      "soc2" : "yes",
      "pcidss" : "yes",
      "ssae16" : "no",
      "iso27001" : "yes"
    },
    "securityComp" : {
      "sas70" : "no",
      "safeHarbor" : "no",
      "fisma" : "no",
      "usPatriotAct" : "no"
```

```
{
  "_id" : ObjectId("5b3ec8d3e02f711d6eef4206"),
  "instanceName" : "T2 Nano",
  "vendorName" : "Amazon Web Service",
  "assurance" : {
    "recoverability" : {
      "recoveryMechanism" : "automatic"
    },
    "availability" : {
      "uptime" : "100",
      "outages" : "0"
    },
    "downtime" : {
      "mttr" : 0,
      "mttf" : 720
    }
  },
  "compliances" : {
    "standardComp" : {
      "iso9001" : "yes",
      "soc3" : "yes",
      "soc1" : "yes",
      "iso27017" : "yes",
      "soc2" : "yes",
      "pcidss" : "yes",
      "ssae16" : "no",
      "iso27001" : "yes"
    },
    "securityComp" : {
      "sas70" : "no",
      "safeHarbor" : "no",
      "fisma" : "yes",
      "usPatriotAct" : "no"
    },
```


Figure 7.35 Comparison Google Cloud f-1 micro vs Amazon T2 Nano

Each of these tests was carried out multiple times to verify that the results are consistent and accurate. Further, these results are also in line with those of the baseline method which were evaluated using analytical and numerical tests (as in Section 7.3).

7.5 Comparison with Existing Cloud Service Selection Tools

This section evaluates the SST by comparing the functionalities with existing (commercial) cloud comparison tools. The main tools to be compared include Intel Cloud Finder[12], Cloudorado[13] and RankCloudz[14]. All of these tools are well known cloud selection tools and perform functions which are closely related to the SST functionality. But these tools have

[12] https://www.intelcloudfinder.com/cloud-provider-search
[13] https://www.cloudorado.com/cloud_server_comparison.jsp
[14] http://rightcloudz.com/RankCloudzOnline

different interfaces and characteristics as shown in the figures below (Figures 7.36, 7.37 and 7.38).

Figure 7.36 Intel Cloud Finder Interface

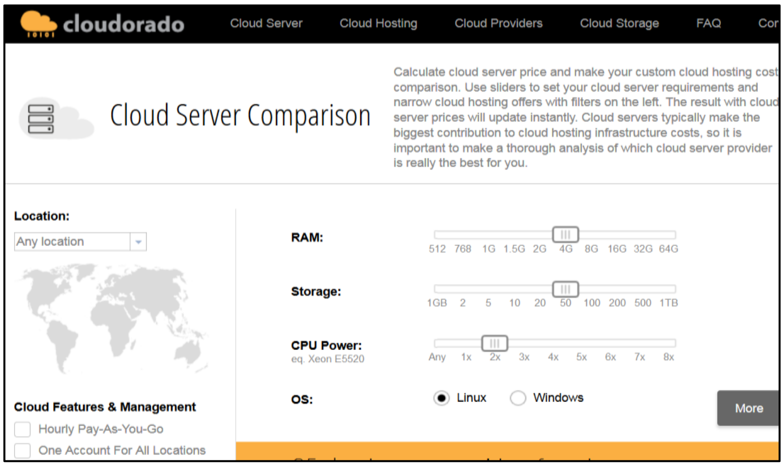


Figure 7.37 Cloudorado Interface

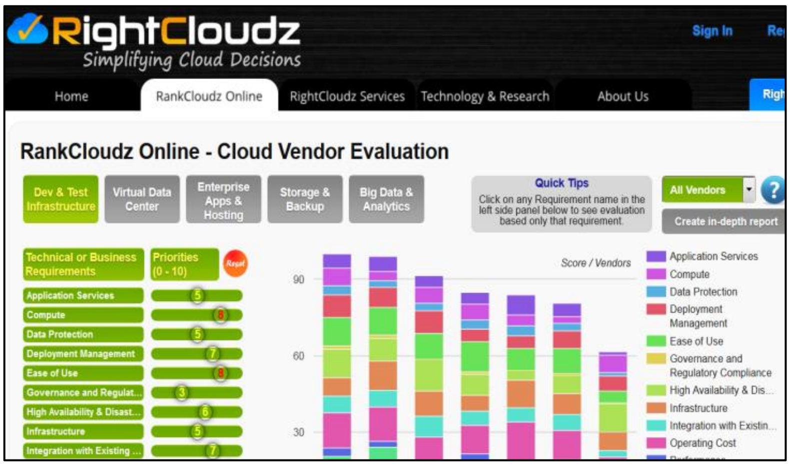


Figure 7.38 RankCloudz Interface

Note that the underlying algorithms and methods used in ranking the cloud services by these tools are not publically available (nor published). Therefore this evaluation is particularly focussed on the web interface functionality which is available through their webs links. At of the time of these writing, each of these tools is still available on the websites links provided with the exception of the Intel Cloud Finder. This particular tool is not anymore available online. However, this evaluation was performed before it was shut down. Each of these tools is evaluated against the SST based on particular functionality. As part of the evaluation, the results (or ranking) of each of the existing cloud selection tools cannot be compared directly with the result of the SST for multiple reasons. Firstly, the attributes measured by each of the tools are different from each other as well as different from the SST. Therefore, the results will be based on different attributes. Secondly, while the (commercial) tools only measure cloud providers, the SST measures both the providers and the different instances (or services) supported by each provider. For instance, the SST repository contains a minimum of five different types of instances from each provider. This makes the SST more robust than all the other tools in the ranking and selection process. However, this also implies that the ranking provided by the SST will be different from the other tools. Thirdly, the cloud providers that each of these tools rank are different. While the SST collects data for eight different cloud providers, RankCloudz only ranks five different cloud providers. Cloudorado on the other hand only shows the results for the cloud providers that meet the criteria selected by the consumer. For instance, when a consumer selects attributes in Cloudorado that both the Cloudorado and SST have in common, the Cloudorado ranking shows a few providers in its ranking that are not among the providers that the SST have in its repository. However, extra providers can be easily added to SST repository.

Given the above reasons and differences, it is impractical to compare the (cloud service/provider) ranking of the commercial tools with that of SST tool. Therefore, the comparison is carried out on a set of identified criteria explained in (Eisa *et al*., 2016). These criteria have been extended to include ease of access, mode of data collection and flexibility and customization.

7.5.1 Operability

Operability refers to how easy it is for consumers to operate or use the functions of the tool. With respect to Operability, the SST is very easy to operate than the other existing tools. Cloudorado and Intel Cloud Finder are more difficult to operate than RankCloudz. RankCloudz is easier to operate because it allows the consumer to adjust their priorities and consumers can see the ranking of the cloud providers as they adjust their priorities on the slider. Cloudorado on the other hand is more difficult to operate because the interface presents consumers with a lot of check boxes, sliders etc., which can be cumbersome for consumers. The SST on the other hand considers that consumers have different degree of knowledge and understanding. Therefore, from the initial page, the SST asks the consumer about their level of knowledge on cloud services. This feature is very important because it allows Beginner consumers to interact with the tool without going into the complex technical details which they would not understand. Therefore, the SST is considered to be more appropriate with respect to operability.

7.5.2 Number of Cloud Providers

This feature compares the number of cloud providers that each of these tools considers in their ranking. When compared with the SST, both Cloudorado and Intel Cloud Finder have higher number of cloud providers than the SST and RankCloudz. The SST, however, has a higher number of cloud providers in its repository than the RankCloudz. The system is, however, capable of handling more information and more providers will be added in a fully working system.

7.5.3 Understanding of QoS Attributes

This section evaluates how easy it is to understand the attributes that are used in the ranking. This feature is only provided by Cloudorado and the SST. Other tools do not have this feature. Intel Cloud Finder, in particular, provides attributes description that appear to be less clear to consumers. The SST implements a function that enables consumers to hover mouse on each of the attributes that are used to assess the provider. With ‘mouse hover’ the description of the attribute is then displayed to the consumer. This feature allows the consumer to understand each of the attributes and is especially useful for consumers who are not very familiar with cloud services. This functionality is also used by Cloudorado. Therefore, the SST and Cloudorado provide attribute description which make it easier for consumers to understand.

7.5.4 Support for Service Provisioning Models

This section refers to the type of cloud service that the tool ranks. As mentioned earlier, cloud services are grouped into three main types which include IaaS, PaaS and SaaS. Most of the tools, including the SST assess cloud providers based on Infrastructure as a Service (IaaS). The reason for this is that it is slightly more complicated to assess the performance of Platforms and Software given the underlying infrastructure and the lack of access to related data. Cloudorado has some limited support for assessing platform as a services attributes of cloud providers. The model can, however, be extended in developing attributes that can be used to measure PaaS and SaaS using the SST.

7.5.5 SLA Integration

Integration with SLA means that the tool is able to provide the cloud providers SLA along with the ranking. All the available tools do not integrate SLA in their tools. This is one novel feature of the SST. The SST integrates the Service Level Agreements of each cloud provider in its search and ranking process. This was demonstrated in section 7.2.11. Also this particular feature was part of the functional requirement (REQ-F-08**)** in the design of the SST (See section 6.2.1).

7.5.6 Mode of Data Collection

The section evaluates how the tools collect and store data about cloud providers. It is not possible to know how the existing commercial tools collect data. All of the three tools keep only static data about the providers and their services. The SST also makes use of static data; the difference, however, is that the SST collects data from a variety of sources as explained earlier. The richness in the variety of data sources makes the SST very robust for the consumers. The data is collected through scrapping. At the moment, data is collected at random intervals when the script is run and is not automated.

7.5.7 Ease of Access

Ease of access evaluates how easy it is to access this tool. All the tools are accessed through their web pages. RankCloudz provides consumers with limited access when they have not registered an account. This may discourage some consumers from wanting to use the tool for concern that their information could be used for other marketing purposes. The SST on the other hand does not need the consumer to register to be able to have access to the full features. This is also the same for Cloudorado and Intel Cloud Finder.

7.5.8 Visualizing Result

This section explains how results are presented to the consumers. Intel Cloud Finder presents results to consumers in a tabular list. Cloudorado presents results to consumers in a sorted list, while RankCloudz presents bar charts to consumers. The SST presents the results to consumers in a sorted list. In addition, the SST allows consumers to use a search function that allows the consumer to search for a particular cloud provider. This feature is useful because when the SST presents consumers with a list that spans multiple pages, the consumers can then search for a particular cloud provider if needed. In addition, the SST allows the consumer to specify the number of results that they want the tool to display on each page. Consumers are also allowed to determine the order in which results should be presented either by their ranking or alphabetically. None of the other tools have these functions.

7.5.9 Flexibility and Customization

Flexibility refers to how much the ranking process in the tools can be customized based on consumers preference. RankCloudz has just one level (attributes), i.e. non-hierarchical attributes structure, however, consumer are also allowed to adjust the priority of each attribute. Cloudorado on the other hand has hierarchal structure but is not very flexible. Consumers are presented directly with level two attributes where they can select their options, however, they are unable to explicitly choose their level one QoS attributes. Also, with Cloudorado, consumers are not allowed to set their priority attributes. In addition, the consumers can only make binary attribute selection. The Intel Cloud Finder is also not flexible. Consumers are presented with level three attributes and can assign their priority on level three attributes. However, they cannot determine the level one or level two attributes they want. This may not be ideal for consumers who are Beginners as they may not understand such level of details. However, the SST has hierarchal structure with flexibility, and consumer can select the attributes they want on all levels. In addition, the consumer can assign priority to both level one and level two attributes. Furthermore, consumers also have the options to apply filtering to level three attributes. All these make the SST the most flexible among all the tools.

7.6 Overall Comparison

Based on the criteria (and factors) described above, this section devises a scoring strategy for the aforementioned cloud service selection tools. The table below (Table 7.12) shows how each of the tools scores on each criterion.

Table 7.12 Measurement Criteria and Scoring for the Tools

Criteria	Description	Intel Cloud Finder	Cloudorado	RankCloudz	SST
Operability		Poor	Poor	Good	Very Good
Number of Cloud Providers	High = Good Medium = Fair Low = Poor	Good (98)	Fair (21)	Poor (5)	Poor (8)
Understanding of QoS Attributes		Very Poor	Very Good	Very Poor	Very Good
Support for Service Provisioning Models	Support for All three = Very Good Support for 2 = Good Support for 1 = Fair Support for None = NA	Fair = IaaS	Good = IaaS (Partly PaaS)	Fair = IaaS	Fair = IaaS

SLA Integration	Yes = Very Good No = Very Poor	Very Poor	Very Poor	Very Poor	Very Good
Mode of Data Collection	Scrapping = Fair Real-time streaming = Very Good	NA	NA	NA	Fair = Scrapping
Ease of Access	Full Access No Login = Very Good Full Access Login = Fair Full Access (Paid) = Very Poor	Very Good	Very Good	Fair	Very Good
Visualizing Result		Fair	Fair	Fair	Very Good. Integrated with search function and adjustable ordering patterns
Flexibility & Customization		Fair	Fair	Fair	Very Good

From the table above (Table 7.12), the following scores are assigned to the fuzzy or linguistic terms as shown in the table below (Table 7.13).

Table 7.13 Score Mapping for Criteria Measurement

Score Mapping		
Linguistic Terms	Value	Code
Very Poor	0	
Poor	1	
Fair	2	
Good	3	
Very Good	4	
No Value	NA	

Based on the Score mapping the following heat map has been generated. The heat map provides a graphical view of how each of the tools measured against the criteria discussed.

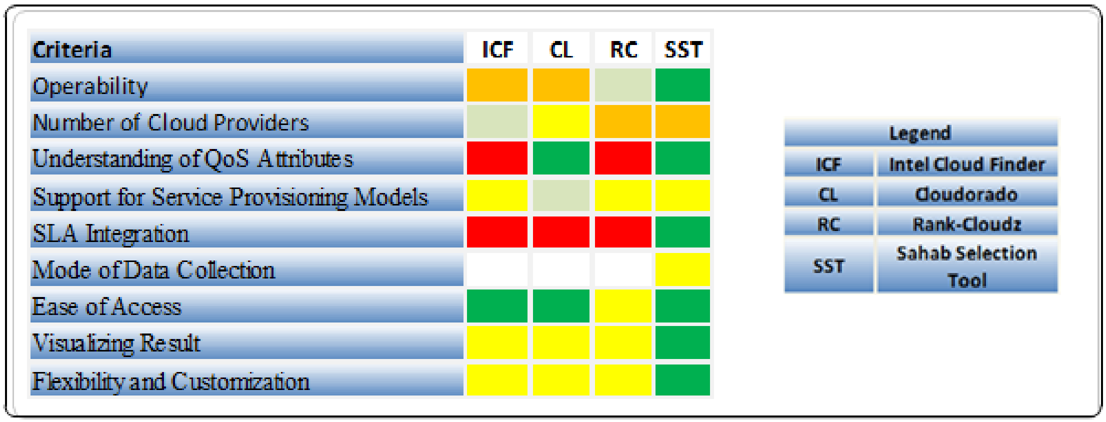


Figure 7.39 Overall Heat map

The Figure 7.39 above shows how each of the tools performed on the criteria based on the critical analysis performed on the existing tools. The figure shows that overall; the SST has more green sections than each of the other tools. The SST has fewer cloud providers in the repository; however, it covers the main cloud service providers.

7.7 Summary

This chapter has presented the important aspects of this research. First, a detailed testing of the functionality of the SST tool was carried out. Each of the functions was rigorously tested using the widely used black box testing technique. Second, an extensive evaluation of the proposed method for cloud service selection was conducted. Various experiments were completed and results produced. The experiments took into consideration all the aspects which were defined and proposed as part of this research. These included, for example, knowledge level of consumers, different cloud providers, different number of QoS attributes and their level of hierarchy, and so on. The results showed that the SST tool appropriately implemented the AHP and SAW methods and produced useful and credible results when ranking and selecting cloud services provided by a large number of cloud providers. Third, a comparative evaluation of the SST tool and three popular commercial cloud service selection tools was carried out using various factors such as operability, SLA integration, flexibility and customization, and so on. The comparison showed that each of the tools have some strong as well as weak points. However, compared to existing tool, the SST generally possesses useful and novel features.

CHAPTER 8

Conclusions and Future Work

8.1 Summary of the Work

This thesis researched into the problem of cloud service selection and provisioning wherein consumers can select cloud services that meet their requirements based on QoS attributes. The thesis addressed a timely and complex research issue given the exponential increase in the number of cloud providers and the number of services they offer to consumers. Each of these cloud service providers often promise better cloud services to their potential consumers, who may not be knowledgeable enough about the appropriate services that better meet their requirements. In addition, consumers may not know where and how to get the right information about the multiple cloud service providers and the services they offer. This can lead to consumers paying more when they select the wrong service provider or in some cases, consumers selecting services that do not meet their needs or requirements. Therefore cloud service selection has become a huge challenge to the cloud consumers. The rising number of services available among the cloud providers and the different classification of QoS attributes further complicate this problem. This research therefore has addressed this problem by developing new methods and tool that enable the consumers to assess, rank and select appropriate cloud service providers. The solution presented in this research follows solid theoretical approach which is based on the AHP and SAW methodologies. The main research work completed in this thesis is summarised as follow.

Firstly, this research analysed and defined the existing problems (explained in section 1.2.2) such as the lack of standardized metrics for QoS attributes. This problem is addressed by identifying and defining the various QoS attributes of cloud services. Secondly, these attributes are then grouped into categories including technical, strategic & organisational, economic and political & legislative (see section 3.4.2). The above categorisation made it possible to design a hierarchical grouping of attributes into three levels using the Analytical Hierarchy Process (AHP) algorithm (section 5.4.3). Data is then collected from various sources including cloud providers websites, user review websites and cloud monitoring website. The data collected from the various sources were of different types and was normalised and converted into uniform numerical values so that they can be used for ranking and selecting the cloud services. (See section 5.4.4). Cloud based NoSQL, MongoDB was used as a central repository (backend database) since MongoDB is ideal and well used in the industry for storing and querying large scale and semi-structured data. The functional and non-functional requirements are then defined and clearly specified (section 6.2). Based on these requirements, the use case scenarios are identified and the tool (SST) is developed. The SST tool is designed and implemented such as that it performs all the necessary tasks that are needed in cloud services ranking and selection. Consumers select the QoS attributes which they want to measure and to perform a pairwise comparison (to set their priority). Consumers can either be Beginner or Intermediate/Expert level depending on how much knowledge they have on cloud services. The SST applies AHP to provide ranking for Beginner level consumers while it applies AHP and SAW methodologies (on sub attributes) for Intermediate/Expert level consumers.

This research performs a black box testing to rigorously test the SST. The results of the testing shows that the SST behaved as expected for all the test cases. The validation and evaluation of the proposed framework were carried out using a baseline evaluation of the methodologies with numerical test as well as through the implementation of the SST tool. The ranking provided by the SST is further evaluated using different scenarios for both Beginner and Intermediate/Expert level consumers. Finally, the SST is evaluated against related (commercial) tools based on identified functionalities (see section 7.5). The results of the comparison showed that the SST outperforms all the existing tools on the identified functionalities.

The remainder of this chapter is organized as follows.

First, it explains the summary of the main contributions of this research. It then provides a critical analysis of the solution developed in this thesis. The critical analysis appraises and explains the strengths and the weaknesses of the solution. The critical analysis also provides a roadmap for future research opportunities. The chapter concludes with identifying directions for future research work.

8.2 Contributions

The contributions to knowledge made by this research are highlighted and explained below.

1. Development of a new model to represent the Quality of Service attributes (QoS). This research provided a novel set of attributes which can be used as standard criteria to measure cloud services. Cloud service providers offer numerous services to consumers and services are associated with various QoS attributes. These QoS are used as the criteria for the service selection process. The QoS attributes proposed in this work covers technical and non-technical aspects of cloud services. This thesis defined a set of QoS attributes which capture the essential characteristic of cloud services and developed a hierarchical structure for

representing these characteristics. This contribution is an important step towards standardizing the definition and representation of QoS attributes of cloud services.

2. Development of a new framework for cloud service selection. The framework developed and described in this thesis presents consumers with a simplified way for selecting cloud services. The framework is implemented using the AHP and SAW methodologies to provide ranking for cloud services based on convoluted weights computed from pairwise comparison of a set of attributes selected by the consumers. The framework designed also takes into account that potential / intending cloud service consumers have different levels of knowledge with respect to cloud services. Therefore, the framework is designed to support consumers who have little knowledge in cloud services as well expert level consumers. Furthermore, the framework allows consumers to input their priorities and to determine the required level of details.
3. Guarantee of the credibility and validity of the service selection by gathering data from multiple sources. The service selection framework gathers data from cloud service providers, user review websites and third party monitoring tool. The decision of gathering data from multiple sources (aside from the cloud providers), produces un-biased results during service selection process. This is because cloud providers generally do not provide negative information about their services. Therefore it was important to collect data from neutral sources such as third party monitoring tool. User reviews also provided an important source of information as consumers can provide first-hand experience of cloud services.
4. The integration of cloud provider Service Level Agreements (SLAs) into cloud selection process. This feature is a novel feature and none of the existing tools have taken this into account. The SLA integration allows consumers to review the service level guarantees that each of the cloud service providers provide as part of their service provisioning process. This can provide insights to the consumers so that they can understand the service level agreements of the different cloud service providers. Consumers can have a central or one stop location where they can view the various SLAs without the burden of going to each cloud provider's websites to look for the SLAs.
5. The development of a prototype service selection tool that implements the model and the framework presented in this thesis. Based on various tests and validation criteria the SST has proved to perform better than the existing commercial tools. The evaluation performed demonstrated that the tool is more efficient, flexible and easier to use than the available tools. The SST has been designed to assist consumers in making decisions on which cloud service provider will meet their requirements. In addition, this research provides a platform for cloud service providers to assess their services against competitors. Cloud service

providers can use the SST to know the areas where they are lacking in terms of service provisioning and use this information to improve the services they offer.

8.3 Critical Analysis

The SST is a prototype system which is built as a proof of concept to implement and demonstrate the methodologies proposed in this research. The black box testing carried out in this research shows that the SST tool performs as expected. In addition, the evaluation of the work confirms that the SST outperforms the existing commercial tools. However, the SST (being a prototype system) also has some limitations which are analysed in this section. But these limitations also provide an opportunity for future research work in the area of cloud services selection and provisioning.

The SST is presently installed on a local server on the Amazon AWS cloud platform. The SST tool can only be accessed on the server and cannot be accessed via the Internet. This means that at the moment, general consumers cannot access SST tool unless they are granted access to the AWS server on which the SST is hosted. As a result of this, the SST has not been evaluated under stress to see how it will perform when there is a huge load or high number of requests to the SST tool. But the performance evaluation is beyond the scope of this research.

Further, the data gathering in the SST is not carried out in real time. Implementing real-time data scrapping would ensure that the data in the repository is always up to date. However, since it is not expected that the data will change frequently, it could be more useful to develop an application that will notify the SST when any data items changes on the source system. This can then trigger the scrapping file to collect the new data. Future work may explore this issue to run this script dynamically in order to provide real-time access to results.

Moreover, the SST does not provide sophisticated facility to collect user reviews. In other words, unlike other online systems, SST does not provide facility where users can directly enter user reviews about the cloud services they have been used. Instead it relies on third party user's reviews. Also, systems for third parties user reviews on cloud services have not been properly developed as other users reviews systems (e.g., Amazon user reviews about products). Being a cloud selection tool that helps consumers to decide which service is ideal for them, it could be useful for the SST to implement functionality for consumers to directly upload their reviews. These reviews would, however, need to be validated in order to prevent consumers from entering random and inaccurate reviews. Also, some of the web pages of the SST may need a few enhancements in order to improve consumer experience on the SST tool. For instance, there could be pop up messages that can give consumers more information about error or constraints of the system. In particular, user interface UIF-01 holds a constraint that a consumer must choose a knowledge level before progressing. This page can be enhanced to display the error message to the consumer.

8.4 Future Work

This section describes some of the open research problems that provide directions for future research work in the area of cloud services selection and provisioning.

Performance evaluation of the cloud service selection tools: In addition, to SST there exist commercial cloud services selection tools as well as some research-based system. However, current research work does not provide performance evaluation of cloud service selection tools. For instance, performance of the selection tools should be evaluated by taking into account various factors such as the number of cloud consumers, the number of QoS attributes, and the number of cloud services and providers.

Increasing the number of cloud service providers: Due to the time limitation the SST tool collects QoS attributes from eight cloud providers in its repository. Future, research work can collect QoS attributes from more cloud providers and store them in the repository. This will enable cloud consumers to select cloud services from a wide range of cloud providers.

Providing Service Selection ranking for PaaS and SaaS: This research only supports ranking for IaaS services. Future work can provide support for PaaS and SaaS service selection. To achieve this, future work can adopt the approach followed in this research by identifying the QoS attributes that can be used to measure PaaS and SaaS services. The future research can also adopt the methodologies developed in this thesis to provide ranking and selection of other types of cloud services.

Development of a user review system: Existing cloud service selection tools do not allow consumers to directly enter review on cloud services. It would be important to allow consumers to directly enter user reviews and ratings into the cloud service selection tool. Future work can consider a different set of pages and functionalities for consumers to rate cloud provider services based on their experiences. Consumers can be invited to rate cloud provider services based on their experience. The ratings provided by the consumers can then be collected and stored in the tool repository, thereby enriching the data stored in the repository.

Development of a cloud provider competitor service search: Future research can allow cloud providers to compare their services directly against a competitor. This will give the cloud provider more information about the service attributes that they need to improve. This can also be operated in a pay-as-you-go model where cloud providers can pay for access to QoS information about other competitors and rank their services against the competitor.

Extending SST to implement other methodologies: Currently, the SST is implemented using AHP and SAW methodologies. Future research can consider extending the calculations using other multi-criteria decision methodologies (MCDM) apart from AHP and SAW. Some of these methods are described in section 2.4. This can make the SST ranking process more robust.

REFERENCES

[1] Abourezq, M. and Idrissi, A. 2014. A Cloud Services Research and Selection System. International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, 2014, pp. 1195-1199. doi: 10.1109/ICMCS.2014.6911370.

[2] Abourezq, M. and Idrissi, A. 2014.Skyline In Cloud Computing. Journal of Theoretical and Applied Information Technology. Vol. 60 No.3.

[3] Achar, R. and Thilagam, P. S. 2014. A broker based approach for cloud provider selection. International Conference on Advances in Computing, Communications and Informatics (ICACCI) pp 1252-1257.

[4] Afshari, A., Mojahed, M. and Yusuff, R.M., 2010. Simple additive weighting approach to personnel selection problem. *International Journal of Innovation, Management and Technology*, *1*(5), p.511.

[5] Amazon. 2018. AWS Well-Architected Framework. [Online] Available at: https://d1.awsstatic.com/whitepapers/architecture/AWS_Well Architected_Framework.pdf [Accessed 12th January 2019].

[6] Anda, B. and Sjoberg, D. I. K. 2005. Investigating the Role of Use Cases in the Construction of Class Diagrams. In *Empirical Software Engineering*. 10(3), pp 285-309.

[7] Ardagna, D., Casale, G., Ciavotta, M., Pérez, J.F. and Wang, W., 2014. Quality-of-service in cloud computing: modelling techniques and their applications. *Journal of Internet Services and Applications*, *5*(1), p.11.

[8] Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, A., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D. and Rabkin, A. 2010. A view of cloud computing. Commun. ACM, vol. 53, no. 4, p. 50.

[9] Armstrong, D. and Djemame, K. 2009. Towards Quality of Service in the Cloud in Proceedings of 25th UK Performance Engineering Workshop, (Leeds, 2009).

[10] Baldwin, L.P., Irani, Z. and Love, P.E., 2001. Outsourcing information systems: drawing lessons from a banking case study. *European Journal of Information Systems*, *10*(1), pp.15-24.

[11] Bao, H. and Dou, W. 2012. A QoS-Aware Service Selection Method for Cloud Service Composition. IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, Shanghai, 2012, pp. 2254-2261.

[12] Baranwal, G. and Vidyarthi, D.P., 2014, February. A framework for selection of best cloud service provider using ranked voting method. In *Advance Computing Conference (IACC), 2014 IEEE International* (pp. 831-837). IEEE.

[13] Baset, S. (2012) 'Cloud SLAs: present and future', Amazon Slas, pp. 57–66. Available at: http://dl.acm.org/citation.cfm?id=2331586.

[14] Boniface, M., Nasserm B., Papay, J., Phillips, S. C., Servin, A., Yang, X., et. al. 2010. Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds. Fifth International Conference on Internet and Web Applications and Services, Barcelona, pp. 155-160.

[15] Borzsony, S., Kossmann, D. and Stocker, K., 2001. The skyline operator. In Data Engineering, 2001. Proceedings. 17th International Conference on (pp. 421-430). IEEE.

[16] Briscoe, G. and Marinos, A. 2009. Digital Ecosystems in the Clouds: Towards Community Cloud Computing. 3rd IEEE International Conference on Digital Ecosystems and Technologies Istanbul, 2009, pp. 103-108. doi: 10.1109/DEST.2009.5276725.

[17] Calloway, S.D. and Venegas, L.M., 2002. The new HIPAA law on privacy and confidentiality. Nursing administration quarterly, 26(4), pp.40-54.

[18] Choudhary, V. 2007. "Software as a Service: Implications for Investment in Software Development," 40th Annual Hawaii International Conference on System Sciences (HICSS'07), Waikoloa, HI, pp. 209a-209a.

[19] Cloud Services Measurement Initiative Consortium (CSMIC). 2014. Service Measurement Index Framework Version 2.1.

[20] Cloudorado (2011) Cloud Computing Price Comparison | Cloudorado - Find Best Cloud Server from Top Cloud Computing Companies. Available at: https://www.cloudorado.com/ (Accessed: 20 November 2017).

[21] Columbus, L. (2017) Roundup of Cloud Computing Forecasts Update, 2013, Forbes. Available at: https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts-2017/#7eac163831e8.

[22] Cui, H., Liu, X., Yu, T., Zhang, H., Fang, Y. and Xia, Z., 2017. Cloud service scheduling algorithm research and optimization. Security and Communication Networks, 2017.

[23] Dastjerdi, A.V. and Buyya, R., 2011. A taxonomy of qos management and service selection methodologies for cloud computing. Cloud computing: methodology, systems, and applications, 15, pp.16-76.

[24] Dawoud, M. M., Ebrahim, G. A. and Youssef, S. A., 2016. A Cloud Computing Security Framework Based on Cloud Security Trusted Authority. Proceedings of the 10th International Conference on Informatics and Systems. pp. 133-138.

[25] Eisa, M., Younas, M. and Basu, K., 2018, May. Analysis and representation of QoS attributes in cloud service selection. In 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA) (pp. 960-967). IEEE.

[26] Eisa, M., Younas, M., Basu, K. and Zhu, H., 2016, March. Trends and directions in cloud service selection. In Service-Oriented System Engineering (SOSE), 2016. IEEE Symposium on (pp. 423-432). IEEE.

[27] G. J. Myers, T. Badgett, T. M. Thomas, and C. Sandler. 2004. The Art of Software Testing. John Wiley & Sons.

[28] Garg, S.K., Versteeg, S. and Buyya, R., 2011, December. SMICloud: A framework for comparing and ranking cloud services. In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on (pp. 210-218). IEEE.

[29] Garg, S.K., Versteeg, S. and Buyya, R., 2013. A framework for ranking of cloud computing services. Future Generation Computer Systems, 29(4), pp.1012-1023.

[30] Garlick, G. 2011. Improving Resilience with Community Cloud Computing. Sixth International Conference on Availability, Reliability and Security. Vienna, 2011, pp. 650-655.

[31] Gavade, R. K. 2014.Multi-Criteria Decision Making: An overview of different selection problems and methods.(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , pp. 5643-5646.

[32] Godse, M. and Mulik, S. 2009. An Approach for Selecting Software-as-a-Service (SaaS) Product, IEEE International Conference on Cloud Computing, Bangalore, pp. 155-158.doi: 10.1109/CLOUD.2009.74.

[33] Goyal, S. 2014. Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review. I.J. Computer Network and Information Security, Vol 3, 20-29.

[34] Hwang, C.-L. and Yoon, K. (1981) Multiple Attribute Decision Methods and Applications. Boca Raton: CRC Press.

[35] Intel® (2013) Intel® Cloud Finder - Best Cloud Service and Solution providers. Available at: https://www.intelcloudfinder.com/. (Accessed: 20 November 2017).

[36] ISO (2015) ISO 9000:2015(en), Quality management systems — Fundamentals and vocabulary, Geneva, Switzerland: International Organization for Standardization. Available at: https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en (Accessed: 16 December 2017).

[37] Janssen, M. and Joha, A., 2011, June. Challenges for adopting cloud-based software as a service (saas) in the public sector. In ECIS (p. 80).

[38] Karim, R, Ding, C. and Miri, A. 2013. An End-to-End QoS Mapping Approach for Cloud Service Selection. IEEE Ninth World Congress on Services, Santa Clara, CA, 2013, pp. 341-348.

[39] Kushida, K.E., Murray, J. & Zysman, J. 2011. Journal of Industry, Competition and Trade. Volume 11, Issue 3, pp 209–237.

[40] Lawton, G. 2008. Developing Software Online With Platform-as-a-Service Technology," in Computer, vol. 41, no. 6, pp. 13-15, doi: 10.1109/MC.2008.185.

[41] Lee, S. and Seo, K. 2016. A Hybrid Multi-Criteria Decision-Making Model for a Cloud Service Selection Problem Using BSC, Fuzzy Delphi Method and Fuzzy AHP. Wireless Pers Commun Vol 86: Issue 1. Pp 57-75.

[42] Leong, L., Toombs, D., Gill, B., Petri, G. and Haynes, T, 2017. 'Magic Quadrant for Cloud Infrastructure as a Service, Worldwide', Gartner, (August), pp. 1–36. Avalable at http://www.gartner.com/technology/reprints.do?id=11UM941C&ct=140529&st=sb

[43] Li, A., Yang, X., Kandula, S. and Zhang, M., 2010, November. CloudCmp: comparing public cloud providers. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (pp. 1-14). ACM.

[44] Li, A., Yang, X., Zhang, M. and Kandula, S., 2010. CloudCmp: Shopping for a Cloud Made Easy. HotCloud, 10, pp.1-7.

[45] Li, J., Li, Q., Khan, S. U. and Ghani, N. 2011. Community-Based Cloud for Emergency Management. 6th International Conference on System of Systems Engineering, Albuquerque, NM. Pp. 55-60.

[46] Mell, P. and Grance, T. 2011. 'The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology', NIST Special Publication, 145, p. 7. doi: 10.1136/emj.2010.096966.

[47] Mirobi, G. J. and Nadu, T. (2015) 'Service Level Agreement in Cloud Computing. An Overview', pp. 1–6.

[48] Modi, K. and Jadeja, Y. 2012. Cloud Computing - Concepts, Architecture and Challenges. 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET) pp. 877-880.

[49] Mohan, M. M. K., Sanjay, H. A., and Ashwini, J. P. 2012. Pricing Models and Pricing Schemes of IaaS Providers: A Comparison Study. International Conference on Advances in Computing, Communications and Informatics. (pp. 143 – 147).

[50] Muthusamy, V. and Jacobsen, H. 2008. SLA-Driven Distributed Application Development. Proceedings of the 3rd Workshop on Middleware for Service Oriented Computing. Pages 31-36.

[51] Naldi, M. and Mastroeni, L. 2013. Cloud Storage Pricing: A Comparison of Current Practices. Proceedings of the 2013 International Workshop on Hot Topics in Cloud Services. Pp. 27-34.

[52] Nidhra, S. and Dondeti, J. 2012. Black Box and White Box Testing Techniques: A Literature Review. International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012.

[53] Ouadah, A., Benouaret, K., Hadjali, A. and Nader, F., 2015, April. Combining skyline and multi-criteria decision methods to enhance Web services selection. In Programming and Systems (ISPS), 2015 12th International Symposium on (pp. 1-8). IEEE.

[54] P. Jorgensen. 2002. Software testing: A Craftsman's Approach, CRC Press, p. 359.

[55] Patel, P., Ranabahu, A. H. and Sheth, A. P. (2009) Service Level Agreement in Cloud Computing. CORE Scholar Library.

[56] Peng, J., Zhang, X., Lei, Z., Zhang, B., Zhang, W. and Li, Q. 2009. Comparison of Several Cloud Computing Platforms. Second International Symposium on Information Science and Engineering. pp. 23-27.

[57] Pohekar, S. D. and Ramachandran, M. 2004. Application of multi-criteria decision making to sustainable energy planning—A review. Renewable and Sustainable Energy Reviews. Volume 8, Issue 4.Pages 365-381.

[58] Polyviou, A., Pouloudi, N. and Rizou, S., 2014, January. Which Factors Affect Software-as-a-Service Selection the Most? A Study from the Customer's and the Vendor's Perspective. In System Sciences (HICSS), 2014 47th Hawaii International Conference on (pp. 5059-5068). IEEE.

[59] Rehman, Z. U., Hussain, F. K. and Hussain, O. K.2011. Towards Multi-criteria Cloud Service Selection. Proceedings of the 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing series. Pp 44-48.

[60] Rehman, Z. U., Hussain, F. K. and Hussain, O. K.2012. IaaS Cloud Selection using MCDM Methods. Ninth International Conference on e-Business Engineering, Hangzhou, 2012, pp. 246-251.

[61] RightCloudz (2014) Cloud Comparison | Cloud Evaluation | Best IaaS | RightCloudz. Available at: https://rightcloudz.com/RankCloudzOnline (Accessed: 20 November 2017).

[62] Rimal, B. P., Choi, E and Lumb, I. 2009. A Taxonomy and Survey of Cloud Computing Systems. Fifth International Joint Conference on INC, IMS and IDC.

[63] Saaty, T. L. 1977. "Modelling unstructured decision-making - AHP", International Conference on Mathematical Modelling.

[64] Sabahi, F. 2011. Cloud computing security threats and responses. 3rd International Conference on Communication Software and Networks.

[65] Salama, M., Shawish, A., Zeid, A. and Kouta, M., 2012, July. Integrated QoS utility-based model for cloud computing service provider selection. In Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual (pp. 45-50). IEEE.

[66] Saripalli, P. and Pingali, G. 2011. MADMAC: Multiple Attribute Decision Methodology for Adoption of Clouds. 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, pp. 316-323.

[67] Singla, C. and Kaushal, S., 2015, December. Cloud path selection using fuzzy analytic hierarchy process for offloading in mobile cloud computing. In Recent Advances in Engineering & Computational Sciences (RAECS), 2015 2nd International Conference on (pp. 1-5). IEEE.

[68] Son, S., Choi, H., Oh, B. T., Kim, S. W. and Kim, B. S. 2017. Cloud SLA relationships in multi-cloud environment: Models and practices. Int. Conference on Communications (ICCMS) Canberra, Australia.

[69] Sun, L., Dong, H., Hussain, F.K., Hussain, O.K. and Chang, E., 2014. Cloud service selection: State-of-the-art and future research directions. Journal of Network and Computer Applications, 45, pp.134-150.

[70] Sun, M., Zang, T., Xu, X. and Wang, R., 2013, April. Consumer-centered cloud services selection using AHP. In 2013 International Conference on Service Sciences (ICSS) (pp. 1-6). IEEE.

[71] Sundareswaran, S., Squicciarini, A and Lin, D. 2012. A Brokerage-Based Approach for Cloud Service Selection. IEEE Fifth International Conference on Cloud Computing, Honolulu, pp. 558-565.

[72] Taha, A., Manzoor, S. and Suri, N., 2017, June. SLA-based service selection for multi-cloud environments. In Edge Computing (EDGE), 2017 IEEE International Conference on (pp. 65-72). IEEE.

[73] The Cloud Service Measurement Initiative Consortium (CSMIC) (2011) 'Service Measurement Index Introducing the Service Measurement Index (SMI)', Http://Www.Cloudcommons.Com/About-Smi; Accessed 2013-04-12, (September), pp. 1–8. Available at: http://www.cloudcommons.com/about-smi.

[74] Van Reeuwijk, L.P. and Houba, V.J.G., 1998. Guidelines for quality management in soil and plant laboratories (FAO Soils Bulletin 74). Safety, 4, p.1.

[75] Wagle, S.S., Guzek, M., Bouvry, P. and Bisdorff, R., 2015, November. An evaluation model for selecting cloud services from commercially available cloud providers. In Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on (pp. 107-114). IEEE.

[76] Wang, S., Liu, Z., Sun, Q., Zou, H. and Yang, F. 2014. Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing. Journal of Intelligent Manufacturing. Volume 25, Issue 2, pp 283–291.

[77] Whaiduzzaman, M., Gani, A., Anuar, N.B., Shiraz, M., Haque, M.N. and Haque, I.T., 2014. Cloud service selection using multicriteria decision analysis. The Scientific World Journal, 2014.

[78] Wu, H., Wang, Q. and Wolter, K., 2012, December. Methods of cloud-path selection for offloading in mobile cloud computing systems. In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on (pp. 443-448). IEEE.

[79] Youssef, A. E. and Alageel, M. 2012. A framework for secure cloud computing. International Journal of Computer Science Issues (IJCSI) 9, 4 (Jul. 2012), pp 487-500.

[80] Zeng, W., Zhao, Y. and Zeng, J. 2009. Cloud Service and Service Selection Algorithm Research. GEC'09, June 12–14, 2009, Shanghai, China.

[81] Zhang, F., Hwang, K., Khan, S. U. and Malluhi, Q. M. 2016. Skyline Discovery and Composition of Multi-Cloud Mashup Services in IEEE Transactions on Services Computing, vol. 9, no. 1, pp. 72-83.

[82] Zhang, Q., Cheng, L. and Boutaba, R. 2010. Cloud computing: State-of-the-art and research challenges. J. Internet Serv. Appl., vol. 1, no. 1, pp. 7–18, 2010.

[83] Zissis, D. and Lekkas, D. 2012. Addressing cloud computing security issues. Future Generation Computer Systems, Volume 28, Issue 3, Pages 583-592, https://doi.org/10.1016/j.future.2010.12.006.

Appendix

I. Score Mapping

Level 1	Level 2	Level 3	Value	Point
Security	Access Control	Authentication	yes / no	yes = 100, no = 1
		Authorization	yes / no	yes = 100, no = 1
	Data Security	Encryption	type of encryption	all 100
		Firewall	yes / no	yes = 100, no = 1
	Geography	Provider's Nationality	country name	all 100
		Machine Location	location	all 100
	Auditability	Auditability	yes / no	yes = 100, no = 1
Usability	Interface	Web Interface	yes / no	yes = 100, no = 50
		Mobile App	yes / no	yes = 100, no = 50
		Terminal Access	yes / no	yes = 100, no = 1
	Operability	Operability	yes / no	yes = 100, no = 1
	Learnability	Learnability	yes / no	yes = 100, no = 1
Assurance	Availability	Uptime Percentage	percentage	SORT, highest percentage get 100 and score lowers as it goes down the rank
		Outages	number of outage	SORT, lowest outage number get 100, score lowers as it goes down the rank
	Downtime	Mean Time to Failure	hours	SORT, highest mean time hour get 100, score lowers as it goes down the rank
		Mean Time to Recover	hours	SORT, lowest mean time hour get 100, score lowers as it goes down the rank
	Recoverability	Recovery Mechanism	auto / no	If Automatic , then it gets 100.else if not , then it gets 1
Performance	Hardware	Disk	disk size	SORT, The highest disk capacity gets higher score
		OS	Operating Systems	all 100
		CPU Type	brand and the type	all 100
		GPU	yes / no	If more than zero , then it gets 100.else if less , then it gets 50

Level 1	Level 2	Level 3	Value	Point
	Functionality	Network Performance	Measures of service quality	SORT
		Memory Amount	Size of the memory	SORT
		Number of CPU	CPU cores	SORT
	Flexibility	Flexibility	yes / no	yes = 100, no = 1
	Scalability	Scalability	yes / no	yes = 100, no = 1
Company Performance	Training	Training	yes / no	yes = 100, no = 1
	Customer Support	Customer Support	yes / no	yes = 100, no = 1
Pricing	Price	Windows	price in USD	SORT
		Linux	price in USD	SORT
	Charge Model	Pay as you go	yes / no	all 100
		Contract	yes / no	all 100
	Pricing Unit	Per hour	yes / no	all 100
		Per minute	yes / no	all 100
		Per GB	yes / no	all 100
	Currency	Currency	accepted payment currency	If more than 1 currency , then it gets 100,else if no , then it gets 90
	Support Fee	Free	yes / no	yes = 100, no = 50
		Additional Package	yes / no	all 100
	Discounting	Discounting	yes / no	yes = 100, no = 50
	Pricing System	Tiered Pricing	yes / no	all 100
		Volume Pricing	yes / no	all 100
Compliance	Security Compliance	US Patriot Act	yes / no	yes = 100, no = 1
		FISMA	yes / no	yes = 100, no = 1
		Safe Harbor	yes / no	yes = 100, no = 1
	Legal Compliance	HIPAA	yes / no	yes = 100, no = 1
		Sarbanes-Oxley Act (SOX)	yes / no	yes = 100, no = 1
	Standard Compliance	SSAE 16	yes / no	yes = 100, no = 1
		ISO 27001	yes / no	yes = 100, no = 1
		ISO 9001	yes / no	yes = 100, no = 1
		ISO 27017	yes / no	yes = 100, no = 1

Level 1	Level 2	Level 3	Value	Point
		PCI DSS	yes / no	yes = 100, no = 1
		SOC 1	yes / no	yes = 100, no = 1
		SOC 2	yes / no	yes = 100, no = 1
		SOC 3	yes / no	yes = 100, no = 1

II. Data Set Sample Stored in SST Repository

No	Provider	Instance	Windows ($/hr)	Linux ($/hr)	Avg. Price ($/hr)	Uptime (%)	Memory (GB)
1	IBM	B1.1x2x25	0.078	0.053	0.0655	100	2
2	IBM	B1.1x2x100	0.078	0.053	0.0655	100	2
3	IBM	BL1.1x2x100	0.072	0.047	0.0595	100	2
4	IBM	BL1.1x4x100	0.102	0.077	0.0895	100	4
5	IBM	C1.1x1x25	0.063	0.038	0.0505	100	1
6	Fujitsu	P-1	0.006	0.006	0.006	100	0.5
7	Fujitsu	T-1	0.012	0.012	0.012	100	1
8	Fujitsu	C-1	0.025	0.025	0.025	100	2
9	Fujitsu	P2-1	0.007	0.007	0.007	100	0.5
10	Fujitsu	T2-1	0.015	0.015	0.015	100	1
11	Oracle	BM.Standard1.36	0.0638	0.0638	0.0638	100	256
12	Oracle	BM.Standard1.36	0.0638	0.0638	0.0638	100	256
13	Oracle	BM.Standard2.52	0.0638	0.0638	0.0638	100	768
14	Oracle	BM.Standard2.52	0.0638	0.0638	0.0638	100	768
15	Oracle	BM.DenseIO1.36	0.1275	0.1275	0.1275	100	512
16	Oracle	BM.DenseIO1.36	0.1275	0.1275	0.1275	100	512
17	Oracle	BM.DenseIO2.52	0.1275	0.1275	0.1275	100	768
18	Oracle	BM.DenseIO2.52	0.1275	0.1275	0.1275	100	768
19	Oracle	VM.Standard1.1	0.0638	0.0638	0.0638	100	7
20	Oracle	VM.Standard1.16	0.0638	0.0638	0.0638	100	112
21	Oracle	VM.Standard2.1	0.0638	0.0638	0.0638	100	15
22	Oracle	VM.Standard2.24	0.0638	0.0638	0.0638	100	320
23	Oracle	VM.GPU2.1	1.275	1.275	1.275	100	104
24	Oracle	VM.GPU2.1	1.275	1.275	1.275	100	104
25	Oracle	BM.GPU2.2	1.275	1.275	1.275	100	192
26	Oracle	BM.GPU2.2	1.275	1.275	1.275	100	192
27	Oracle	VM.GPU3.1	2.25	2.25	2.25	100	90
28	Oracle	VM.GPU3.4	2.25	2.25	2.25	100	360
29	Oracle	BM.GPU3.8	2.25	2.25	2.25	100	768
30	Oracle	BM.GPU3.8	2.25	2.25	2.25	100	768
31	Rackspace	General1-2	0.084	0.064	0.074	100	2

No	Provider	Instance	Windows ($/hr)	Linux ($/hr)	Avg. Price ($/hr)	Uptime (%)	Memory (GB)
32	Rackspace	General1-2	0.084	0.064	0.074	100	2
33	Rackspace	General1-4	0.168	0.128	0.148	100	4
34	Rackspace	General1-8	0.336	0.256	0.296	100	8
35	Rackspace	General1-8	0.336	0.256	0.296	100	8
36	Rackspace	Compute1-4	0.11	0.08	0.095	100	4
37	Rackspace	Compute1-8	0.22	0.16	0.19	100	8
38	Rackspace	Compute1-15	0.44	0.32	0.38	100	15
39	Rackspace	Compute1-30	0.88	0.64	0.76	100	30
40	Rackspace	Compute1-60	1.76	1.28	1.52	100	60
41	Rackspace	I/O1-15	0.6	0.48	0.54	100	15
42	Rackspace	I/O1-30	1.2	0.96	1.08	100	30
43	Rackspace	I/O1-60	2.4	1.92	2.16	100	60
44	Rackspace	I/O1-90	3.6	2.88	3.24	100	90
45	Rackspace	I/O1-120	4.8	3.84	4.32	100	120
46	Rackspace	Memory1-15	0.24	0.18	0.21	100	15
47	Rackspace	Memory1-30	0.48	0.36	0.42	100	30
48	Rackspace	Memory1-60	0.96	0.72	0.84	100	60
49	Rackspace	Memory1-120	1.92	1.44	1.68	100	120
50	Rackspace	Memory1-240	3.84	2.88	3.36	100	240
51	Amazon	T2 Nano	0.013	0.013	0.013	100	0.5
52	Amazon	T2 Micro	0.016	0.016	0.016	100	1
53	Amazon	T2 Small	0.028	0.028	0.028	100	2
54	Amazon	T2 Medium	0.062	0.062	0.062	100	4
55	Amazon	T2 Large	0.091	0.091	0.091	100	8
56	Amazon	T2 Extra Large	0.148	0.148	0.148	100	16
57	Amazon	T2 Double Extra Large	0.263	0.263	0.263	100	32
58	Alibaba	ecs.sn2ne.large	0.228	0.133	0.1805	99.9771	8
59	Alibaba	ecs.sn1ne.xlarge	0.388	0.224	0.306	99.9771	8
60	Alibaba	ecs.sn1ne.large	0.194	0.112	0.153	99.9771	4
61	Alibaba	ecs.se1.large	0.249	0.158	0.2035	99.9771	16
62	Alibaba	ecs.se1.xlarge	0.498	0.316	0.407	99.9771	32
63	Alibaba	ecs.d1ne. 8xlarge	7.359	4.672	6.0155	100	480
64	Alibaba	ecs.i2.xlarge	0.53	0.354	0.442	100	32
65	Alibaba	ecs.i1-c10d1.8xlarge	3.585	2.306	2.9455	100	128
66	Alibaba	ecs.hfg5.14xlarge	5.607	2.916	4.2615	100	160
67	Alibaba	ecs.se1ne.large	0.261	0.165	0.213	100	16
68	Alibaba	ecs.hfg5.large	0.217	0.121	0.169	100	8
69	Alibaba	ecs.cm4.6xlarge	3.09	1.77	2.43	100	96
70	Alibaba	ecs.hfc5.6xlarge	2.366	1.317	1.8415	100	48
71	Alibaba	ecs.c4.xlarge	0.44	0.26	0.35	100	8

No	Provider	Instance	Windows ($/hr)	Linux ($/hr)	Avg. Price ($/hr)	Uptime (%)	Memory (GB)
72	Alibaba	ecs.gn5-c4g1.xlarge	2.232	1.716	1.974	100	30
73	Alibaba	ecs.cm4.6xlarge	3.01	1.71	2.36	100	96
74	Alibaba	ecs.i1.14xlarge	6.038	3.758	4.898	100	224
75	Alibaba	ecs.hfg5.8xlarge	3.47	1.932	2.701	100	128
76	Alibaba	ecs.c4.xlarge	0.38	0.23	0.305	100	8
77	Alibaba	ecs.hfg5.6xlarge	2.603	1.449	2.026	100	96
78	Microsoft	Standard_F1	0.063	0.063	0.063	100	2
79	Microsoft	Standard_F2	0.126	0.126	0.126	100	2
80	Microsoft	Standard_F4	0.252	0.252	0.252	100	8
81	Microsoft	Standard_DS11_v2	0.2	0.2	0.2	100	14
82	Microsoft	Standard_B8ms	0.512	0.512	0.512	100	32
83	Microsoft	Standard_D2s_v3	0.125	0.125	0.125	100	8
84	Microsoft	Standard_F16	1.042	1.042	1.042	100	32
85	Microsoft	Standard_E2s_v3	0.16	0.16	0.16	100	16
86	Microsoft	Standard_NC12	1.8	1.8	1.8	100	112
87	Microsoft	Standard_NC6	0.9	0.9	0.9	100	56
88	Microsoft	Standard_E64i_v3	4.469	4.469	4.469	100	432
89	Microsoft	M8ms	2.2279	2.2279	2.2279	100	218.75
90	Microsoft	Standard_NC24	3.6	3.6	3.6	100	224
91	Microsoft	Standard_NC6	0.9	0.9	0.9	100	56
92	Microsoft	Standard_D5_v2	1.345	1.345	1.345	100	56
93	Microsoft	Standard_A1_v2	0.05	0.05	0.05	100	2
94	Microsoft	Standard_NV24	6.36	6.36	6.36	100	224
95	Microsoft	Standard_H8	1.185	1.185	1.185	100	56
96	Microsoft	Standard_E64i_v3	4.814	4.814	4.814	100	432
97	Microsoft	M8ms	2.2279	2.2279	2.2279	100	218.75
98	Google	f1-micro	0.009	0.009	0.009	100	0.6
99	Google	g1-small	0.03	0.03	0.03	100	1.7
100	Google	n1-megamem-96	12.362	12.362	12.362	100	1433.6
101	Google	n1-highmem-4	0.2696	0.2696	0.2696	100	26
102	Google	n1-standard-96	5.48	5.48	5.48	100	360
103	Google	n1-highmem-2	0.1421	0.1421	0.1421	100	13
104	Google	n1-highcpu-96	4.08	4.08	4.08	100	86.4
105	Google	n1-megamem-96	10.674	10.674	10.674	100	1433.6
106	Google	n1-highcpu-96	3.4032	3.4032	3.4032	100	86.4
107	Google	n1-highmem-16	1.0786	1.0786	1.0786	100	104
108	Google	n1-standard-32	1.52	1.52	1.52	100	120
109	Google	n1-highcpu-64	2.6022	2.6022	2.6022	100	57.6
110	Google	n1-highcpu-16	0.6506	0.6506	0.6506	100	14.4
111	Google	n1-standard-1	0.0535	0.0535	0.0535	100	3.75
112	Google	n1-highmem-8	0.7514	0.7514	0.7514	100	52

No	Provider	Instance	Windows ($/hr)	Linux ($/hr)	Avg. Price ($/hr)	Uptime (%)	Memory (GB)
113	Google	n1-highcpu-64	2.4992	2.4992	2.4992	100	57.6
114	Google	n1-highmem-64	4.8736	4.8736	4.8736	100	416
115	Google	n1-standard-16	0.9792	0.9792	0.9792	100	60
116	Google	f1-micro	0.0086	0.0086	0.0086	100	0.6
117	Google	n1-highmem-32	2.336	2.336	2.336	100	208