The Potential Use of Service-Oriented Infrastructure Framework to Enable Transparent Vertical Scalability of Cloud Computing Infrastructure

Farkhana Muchtar*1,2, Abdul Hanan Abdullah1, Mohd Helmy Abd Wahab3, Nan Mad Sahar1, Radzi Ambar3, Hafizul Fahri Hanafi4,

1 Faculty of Computing, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia
2 School of Computer Sciences, Universiti Sains Malaysia, 11800, Gelugor, Pulau Pinang, Malaysia
3 Faculty of Electronic and Electrical Engineering, Universiti Tun Hussein Onn, 86400 Batu Pahat, Johor, Malaysia
4 Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, 35900 Tanjung Malim, Perak, Malaysia

E-mail: farkhana@gmail.com, hanan@utm.my, helmy@uthm.edu.my, nan@uthm.edu.my, hafizul@fskik.upsi.edu.my,

Abstract. Cloud computing technology has become familiar to most Internet users. Subsequently, there has been an increased growth in the use of cloud computing, including Infrastructure as a Service (IaaS). To ensure that IaaS can easily meet the growing demand, IaaS providers usually increase the capacity of their facilities in a vertical IaaS increase capability and the capacity for local IaaS amenities such as increasing the number of servers, storage and network bandwidth. However, at the same time, horizontal scalability is sometimes not enough and requires additional strategies to ensure that the large number of IaaS service requests can be met. Therefore, strategies requiring horizontal scalability are more complex than the vertical scalability strategies because they involve the interaction of more than one facility at different service centers. To reduce the complexity of the implementation of the horizontal scalability of the IaaS infrastructures, the use of a technology service oriented infrastructure is recommended to ensure that the interaction between two or more different service centers can be done more simply and easily even though it is likely to involve a wide range of communication technologies and different cloud computing management. This is because the service oriented infrastructure acts as a middle man that translates and processes interactions and protocols of different cloud computing infrastructures without the modification of the complex to ensure horizontal scalability can be run easily and smoothly. This paper presents the potential of using a service-oriented infrastructure framework to enable transparent vertical scalability of cloud computing infrastructures by adapting three projects in this research: SLA@SOI consortium, Open Cloud Computing Interface (OCCI), and OpenStack.

1. Introduction
Cloud computing is not new in the computing world. When the internet became as important as other utilities such as water and electricity, the idea of cloud computing already existed although the term ‘cloud computing’ still did not exist. This was largely ignored until 2007 when big players such as Amazon, Google, Sun, IBM and Microsoft started to gain interest in the idea and the term ‘cloud computing’ entered the mainstream [1].
Cloud computing is an evolution and combination of several ideas – on-demand computing, utility computing, grid computing, software as-a-service, and thin clients – where the locus of data storage and computation shifts to ever-bigger clusters of centrally managed computers [2]. In this paradigm, information is permanently stored in servers on the Internet (“Cloud”) and cached temporarily on clients’ hardware that includes desktops, entertainment centers, table computers, notebooks, wall computers, handhelds, sensors, monitors, etc. [3]. There are more than 20 definitions of cloud computing that seem to only focus on certain aspects of this technology [4].

Cloud computing creates a pool of computing resources (processor, memory, disk and network capacity) to allocate them on demand. In other words, this is a collection or a group of integrated and networked hardware, software and internet working infrastructure using the Internet as their communications and transport links to provide hardware, software and networking services to the end clients [5].

By merging technologies such as networking, virtualization and service-oriented architecture and wrapping it all within an Internet-based delivery model, the said platforms can provide on-demand, always on, anywhere, anytime and anyplace, pay-per-use and as needed, highly elastic (scaled up and down in capacity and functionalities) hardware and software services to the general public, general and vertical corporate and business markets etc. [6].

Currently, cloud computing is considered an important paradigm and is the hottest topic in Information Technology (IT) [7]. Cloud computing is an omnipresent concept that is still about to reach its envisioned potential in the business domain [8, 9]. Everyone sees and uses the cloud differently. Some people offload ordinary desktop software and instead use web-based applications, others want on-demand computing with instant availability of extra processing power and extra storage. Nevertheless, the inescapable fact is that companies of all sizes have been increasingly using the cloud to increase their businesses, perhaps by enabling remote working or simply to cut costs or even to create/implement entirely new business models.

Companies can use clouds for different purposes, such as running batch jobs, hosting web applications or for backup and storage. Three main markets are associated to cloud computing:

- Infrastructure-as-a-Service (IaaS) designates the provision of IT and network resources such as processing, storage and bandwidth as well as management middleware. Examples are Amazon EC2 [10], RackSpace [11] and the new Google Compute Engine [12].
- Platform-as-a-Service (PaaS) designates programming environments and tools supported by cloud providers that can be used by consumers to build and deploy applications onto the cloud infrastructure. Examples of PaaS include Amazon Elastic Beanstalk [13], Heroku [14], Force.com [15], Google App Engine [16], and Microsoft Windows Azure [17].
- Software-as-a-Service (SaaS) designates hosted vendor applications. For example, Google Apps [18] (such as Google Docs, Google Calendar or Google Sites), Microsoft Office 365 [19] and Salesforce.com [20].

The idea of this research focuses on the development of resource management in IaaS Cloud Computing, thus allowing vertically scalable computer resources to cut across different data centers. Furthermore, integration between Data Centers is also considered and the OCCI open standard is used for the purpose of integration. The component has been developed by any open source community, thus the component does not need to be developed from scratch.

This paper is organized as follows. In section II, some related works are presented. In section III, an overview of service oriented infrastructure is described and a proposed solution is presented in section IV. Finally section V includes the conclusions and future studies.

2. Related Work
The key feature of cloud computing that allows these computational resources to be shared or outsourced transparently among a vast number of users over a scalable network of nodes is in the notion of scalability [21, 22]. In cloud computing, scalability means a way to add and remove resources to the services depending on the usage [23].
As for customers who generally do not own the infrastructure, they can merely access or rent and consume resources as a service, forego capital expenditure, and pay for what they use – the same analogy with traditional utilities like consumed electricity, or television bills that are on a subscription basis. And from the viewpoint of cloud providers: when there is a lot of traffic and users, they can scale up the service up so that it will not choke, and when there is little or no volume, they can scale the service down to a bare minimum [24, 25].

For Cloud, there are two dimension of scalability; namely, horizontal cloud scalability and vertical cloud scalability [7, 26, 27]. Horizontal cloud scalability is referred as the scalability of inter scalability. This means the scalability of the infrastructure that is not specific to one place which is related to the integration between different data centres that are located in different geographical locations [26]. The method of vertical scaling uses bigger and higher performing hardware to scale. The method of horizontal scaling relies on using more machines instead of more powerful machines. Horizontal scaling relies on commodity hardware whereas vertical scaling would require specialty hardware. Cloud computing uses horizontal scaling may be easy to implement, but it has more limitation, such as 1) there is no guarantees that sufficiently capable hardware exists or is affordable; 2) the hardware improvement is always expensive; 3) because of hardware changes involved, this approach usually involves downtime.

Contrary to horizontal scalability, vertical cloud scalability is referred as the scalability of intra infrastructure. This means that the scalability of the infrastructures inside a single data centre are related to the resources that any data centre are concerned with, such as the servers, physical storage, bandwidth, IPs, domain name, securities and others [26]. On the other hand, horizontal scaling can take full advantage of cloud computing technologies, which means 1) it can use couple normal computers or devices to implement high-performance computing, 2) it can use the capacity on demand and automatically, and 3) can upgrade without downtime.

Most cloud providers offer both scaling up and scaling out solutions, such as Amazon EC2 and Windows Azure [7]. They both support scaling up by using larger (size of) instances and scaling out by adding additional instances. We only focus on scaling out since horizontal scaling is a more cloud-native style. Vertical scaling is limited by the most powerful hardware that is available. Therefore, out of the prior works reviewed, there are only a few studies that focus on the issue of vertical cloud scalability for managing resources in cloud computing have been reported.

For this research, the idea is to look into the potential of using service-oriented infrastructure frameworks to enable transparent vertical scalability of cloud computing infrastructures by adapting three open source projects in this research: SLA@SOI consortium, Open Cloud Computing Interface (OCCI), and OpenStack.

3. Service Oriented Infrastructure
The Service Oriented Infrastructure (SOI) applies Service Oriented Architecture (SOA) to IT Infrastructure [28]. Service Oriented Infrastructure or SOI is the architecture that describes IT infrastructure in terms of service. The architecture encompasses all phases of the SOI life cycle, Design, Provisioning, Operation and Decommissioning and Management of the services. Infrastructure services typically use and/or provide a virtualized pool of shared resources (Servers, Network, Storage, Infrastructure software) that are deployed and managed in a highly automated way. An SOI provides foundational support for a Service-Oriented Application Architecture or other application architectures.

This model shows the conceptual building blocks of SOI. These conceptual building blocks must be instantiated for each type of service or other elements used in an architecture, when that architecture is implemented.

A key element of SOI as opposed to just a service-oriented design is the tight coupling of the infrastructure services to applications and consumers thus capturing the full length and breadth of the IT service chain.

Infrastructure services provide the application designer with a language that enables the designer to work with abstract concepts rather than implementation details. IT infrastructure services provide a standard response to a standard request. This definition could be extended with a classification of the
bandwidth within which this statement is valid. Typically an infrastructure service provides its service based on the actual demand, but is limited to an upper level of the service.

This upper level typically is bound to the maximum capacity the service can use as provided by a resource pool supporting the service. In order to assign the correct amount of resources to deliver the service in the right quality, the service needs to monitor the actual demand as well as the actual supply. This information should be used to adjust the service.

Typically, applications will require a number of infrastructure services in order to support their application interfaces. All the required infrastructure services need to be provided from their respective pools and assigned to the support of the specific application. The application definition should specify a number of aspects:

- The required services or response of the service to a request;
- The initial capacity of the services

The specific quality constraints of the service is defined as the bandwidth in which the service is able to meet its service level requirements. Designing SOI solutions is the process of translating requirements into working solutions. With respect to the type of requirements, functional requirements and non-functional requirements can be distinguished. The functional requirements drive the capabilities within the Block Box: the (static) behavior of the IT Component on a standard request. The non-functional requirements determine the dynamics of the solution and control the provisioning of the resources for the SOI Service.

4. Propose Solution

Based on current trends, there is an increasing number of innovation and new technologies in cloud computing, such as Hybrid Cloud movement, in which many researchers try to create an environment that enables the resources of cloud service providers to be combined.

In addition, open source cloud computing is getting more active, in which there are currently more than one open source project that produces IaaS tools, including Eucalyptus, OpenNebula, and the latest being OpenStack. Moreover, most open source projects, such as KVM, libvirt, glustefs, xen and others have also grown rapidly, and the software developed have become more stable, and has many new features that meet the current demands in IaaS.

There are three open source projects that have an indirect relationship with this research. The first is a research by the SLA@SOI consortium, which focuses on the use of a Service Oriented Infrastructure Framework to attain quality SLA management from cloud service providers to cloud service consumers. The second project is the Open Cloud Computing Interface (OCCI) community. It was created to provide a specification that was agreed upon by the community as a protocol and API for all the needs of management tasks in cloud computing. The third project is OpenStack, which is an open-source software project to develop public and private clouds. OpenStack is a project based on community contribution, aiming at creating a set of open source software to help any organization to have its own cloud computing for virtual computing and storage.

The tools and technologies from all three projects will be utilized in this research to facilitate the development of a prototype. Further information on the three projects are delineated below.

4.1. SLA@SOI

SLA@SOI is a European Commission funded research project committed to deliver and showcase an innovative open Service Level Agreement (SLA) Management Framework that provides holistic support for service level objectives - enabling an open, dynamic, SLA-aware market for European service providers.

The rapidly growing service-oriented economy has highlighted key challenges in IT-supported service provisioning. Service consumers are frustrated by the lack of formal, negotiable, readily enforceable SLAs. Service providers find it impractical to create personalized offerings, translate business requirements into technical manifestations, and optimize internal deployments, whilst maintaining all the individual SLAs.
SLA@SOI has addressed these challenges by providing 3 major benefits:

- **Predictability & Dependability**: The quality characteristics of a service could be predicted and enforced at run-time.
- **Transparent SLA Management**: Service level agreements (SLAs) defining the exact conditions under which services were provided/consumed could be transparently managed across the whole business and IT stack.
- **Automation**: The whole process of negotiating SLAs and provisioning, delivery and monitoring of services were automated, thus allowing for a highly dynamic and scalable service consumption.

SLA@SOI defined a holistic view for the management of service level agreements (SLAs), implementing an SLA management framework that could be easily integrated into a service-oriented infrastructure (SOI). The main innovative features of the project were:

- An automated e-contracting framework.
- Systematic grounding of SLAs from the business level down to the infrastructure.
- Exploitation of virtualization technologies at the infrastructure level for SLA enforcement.
- Advanced engineering methodologies for the creation of predictable and manageable services.

### 4.2. OCCI Framework

The Open Cloud Computing Interface (OCCI) is a RESTful Protocol and API for all kinds of management tasks. OCCI was originally initiated to create a remote management API for IaaS1 model based Services, allowing for the development of interoperable tools for common tasks, including deployment, autonomic scaling and monitoring. It has since evolved into an exible API with a strong focus on interoperability, while still offering a high degree of extensibility. The current release of the Open Cloud Computing Interface is suitable to serve many other models, in addition to IaaS, including e.g. PaaS and SaaS.

OCCI began in March 2009 and was initially led by co-chairs from SUN Microsystems, RabbitMQ and Universidad Complutense de Madrid. Today, the working group has a membership of over 250 members and includes numerous individuals, and industry and academic parties.

The reasons driving the development of OCCI were:

- **Interoperability**: Allowing for different Cloud providers to work together without data schema/format translations, facade/proxying between APIs and understanding and/or dependency on multiple APIs.
- **Portability**: There was no technical/vendor lock-in and services were enabled to move between providers, hence allowing clients to easily switch between providers, based on business objectives (e.g. cost) with minimal technical cost, which enabled and fostered competition.
- **Integration**: Implementation of the specifications could be easily integrated with existing middleware, third party software and other applications.
- **Innovation**: Driving modern technologies.

The Open Cloud Computing Interface was a boundary protocol and API that acted as a service front-end to a provider’s internal management framework. Service consumers could be both end-users and other system instances. OCCI was suitable for both cases. The key feature was that OCCI could be used as a management API for all types of resources, while at the same time, maintaining a high level of interoperability.

### 4.3. OpenStack Cloud Computing

OpenStack is a collection of open source technologies delivering a massively scalable cloud operating system. OpenStack is currently developing two interrelated projects: OpenStack Compute and OpenStack Object Storage. OpenStack Compute is a software that is being developed for the provision and management of large groups of virtual private servers, whereas the OpenStack Object
Storage is a software for the creation of redundant, scalable object storage using clusters of commodity servers to store terabytes or even petabytes of data.

The OpenStack Community Project is backed by Rackspace, NASA, Dell, Citrix, Cisco, Canonical and over 50 other organizations. OpenStack has grown to become a global software community of developers, technologists, researchers and corporations; collaborating on a standard and massively scalable open source cloud operating system. Their mission was to enable any organization to create and offer cloud computing services running on standard hardware.

OpenStack offers open source software to build public and private clouds. OpenStack is a community and a project, as well as a stack of open source software to help organizations run clouds for virtual computing or storage. OpenStack contains a collection of open source projects that are community-maintained, including OpenStack Compute (code-named Nova), OpenStack Object Storage (code-named Swift) and OpenStack Imaging Service (code-named Glance).

OpenStack provides an operating platform or toolkit for orchestrating clouds. OpenStack is more easily defined once the concepts of cloud computing become apparent, to provide scalable, elastic cloud computing for both public and private clouds, large and small. Their mission is that clouds must be simple to be implemented and massively scalable.

5. Prototype Design

As stated previously, the main component that will be developed is resource management. However, other components, such as the scheduler and management policy must be developed to support the resource management component.

At the same time, those components will not be developed from scratch, but instead will be developed through the process of additions and modifications to the source code for those components that are already in existence in the OpenStack Cloud Computing project.

OpenSTack, SLA@SOI and OCCI projects are open source projects. Hence, those components could be modified, as they also provide the source code of all the available components and are readily available. The components that are proposed in this study are as follows:

i. **Additional scheduler/resource management component**

This component enables the resources of different data centers to be combined into a single resource management. At the same time, this component is created to be integrated with management policy. The resource management component serves to monitor the resources that exist locally or at other locations that have been integrated with each other. At the same time, this component also manages the requests from cloud users, i.e. to create, use, alter and destroy virtual machines in a cloud environment. On the other hand, a scheduler is an implementer of policy management and receives information from the resource management component to determine the quantity of resources to be used in the local cloud. The scope of a scheduler is only based on local resources and resources other than the local cloud are only known by the resource management component, as integration between clouds occurs only in resource management.

ii. **Trusted connections**

This component is created to ensure that the connection between two cloud computing data centres at different locations is secured. This is because there is the possibility of two virtual machines (usually owned by the same user) interacting with each other but are in different data centre locations.

Therefore, a secured connection is an additional requirement to ensure that the data between two (or more) virtual machines cannot be seen by unauthorized parties. In this requirement, it is recommended that OpenCA be used to generate a digital certification for the interconnection between different cloud computing data centres.
iii. **Policy Management**
The main role of management policy is to establish the behavior of the scheduler in determining the utilization of local resources that exist in a local cloud. In this research, management policy also serves as the determinant of resource management component behavior in implementing vertical scalability between cloud resources in different locations. In other words, besides interacting with the scheduler for the local management of cloud resources, the management policy component also interacts with the resource management component in the management of inter-cloud resources.

iv. **Data Centre Integration**
This is the integration that will be used between the data centres. It is proposed that the OCCI standard be used to connect different data centres.

The OCCI standard is chosen for a variety of reasons, including:
- It is an open standard. Thus, the integration between cloud computing across different cloud computing products can be implemented.
- The APIs are straightforward and simple.
- The use of RESTful Web Services instead of other complex web services approaches.
- Almost all cloud computing implementers have already started developing the interface for OCCI open standard

![Figure 1. OCCI in the Cloud Service Provider](image)
6. Conclusion

This paper discusses the potential of using a service-oriented infrastructure framework to enable the transparent vertical scalability of cloud computing infrastructures.

The contribution of the paper can be categorized into the following: reviewing the three projects that have been proposed to be adapted in this research: SLA@SOI consortium, Open Cloud Computing Interface (OCCI), and OpenStack, and suggesting four components to be created in this research: additional scheduler/resource management component, policy management, data centre integration and trusted connections. For future works, a tool will be developed and tested in suitable case studies.

References
[1] Carolan J Gaede S Baty J Brunette G Licht A Remmell J et al. 2009 Introduction to Cloud Computing architecture White Paper (1st Edition)
[2] Baker S 2007, Dec 24. Business Week
[3] Cassarino I Geuna A Martiis 2008. Background paper on ICT and Creative Industries CReATE – Creating a Joint Research Agenda for Promoting ICT-Innovations in Creative Industries across, Europe EU-Project, FP7-Regions-2007-1
[4] Vaquero L M Rodero-Merino L Caceres J and Lindner M 2008 SIGCOMM Computer Communication Review, 39(1) pp 50-55
[5] Galen G (2008, April 7 InfoWorld
[6] GaneK A (2009, Apr 30). The Cloud: a leap created from combining existing technologies
[7] Seyyed Mohsen Hashemi and Amid Khatibi Bardsiri 2012 ARPN Journal of Systems and Software 2(5)
[8] Leavitt N 2009 Is cloud computing really ready for prime time? Computer 42(1) pp 15–20
[9] Raza M H Adenola AF Nafarieh A Robertson W 2015 Procedia Comput. Sci. 52 pp 1114–1119
[10] Amazon Elastic Compute Cloud (Amazon EC2). http://http://aws.amazon.com/ ec2/ [Online; accessed 13-September-2017].
[11] Rackspace. The open cloud company. http://www.rackspace.com/ [Online; accessed 13-September-2017].
[12] Google Compute Engine. http://cloud.google.com/products/compute-engine. html/ [Online; accessed 13-September-2017].
[13] AWS Elastic Beanstalk (beta). Easy to begin, Impossible to outgrow. http://aws.amazon.com/elasticbeanstalk/. [Online; accessed 13-September-2017].
[14] Heroku. Cloud application platform. http://www.heroku.com/ [Online; accessed 13-September-2017].
[15] force.com (salesforce). http://www.force.com/ [Online; accessed 13-September-2017].
[16] Google App Engine. http://cloud.google.com/products/ [Online; accessed 13-September-2017].
[17] Microsoft Windows Azure. https://www.windowsazure.com/en-us/ [Online; accessed 13-September-2017].
[18] Google Apps for Business. http://www.google.com/intl/es/enterprise/apps/business/products.html [Online; accessed 13-September-2017].
[19] Microsoft Office 365. http://www.microsoft.com/en-us/office365/ online-software.aspx. [Online; accessed 13-September-2017].
[20] Salesforce.com. http://www.salesforce.com/ [Online; accessed 13-September-2012].
[21] Anderson J E Wiles F A and Young K P 2008. Issues in Information System 9 (No 1-2) pp 203-206
[22] Mikkilineni R and Sarathy V 2009. 1st Workshop on Collaboration and Cloud Computing (WETICE ’09), pp 57-62
[23] Korri T 2009. Technical Reports in Computer Science and Engineering (TKK-CSE-B5), Department of Computer Science and Engineering, Helsinki University of Technology, Finland
[24] Aymerich F M Fenu G and Surcis S 2008. In 1st International Conference on the Applications of Digital Information and Web Technologies pp 113–118
[25] Mc Evoy G V and Schulze B 2008. Proc. of the 6th International workshop on Middleware for grid computing (MGC ’08) (New York, NY, USA: ACM) pp 1–6
[26] Mei L Chan L and Tse T H 2008. Proc. of 2008 IEEE Asia-Pacific Services Computing Conference, APSCC’08 (Los Alamitos, CA, USA: IEEE Computer Society Press) pp 464–469
[27] Mei L Zhang Z and Chan W K 2009. 33rd Annual IEEE International Computer Software and Applications Conference, COMPSAC’09 (Los Alamitos, CA: IEEE Computer Society Press) pp 464-469.
[28] Yan Z 2008 Proceedings of the 2008 IEEE Congress on Services - Part I. SERVICES ’08, (Washington, DC, USA: IEEE Computer Society) pp 99-100