Hybrid Clouds for Web Systems: Usability and Performance

Chien-Te Lu[1], C.-S. Eugene Yeh[2], Yung-Chung Wang[1], Chu-Sing Yang[3]

[1] Department of Electrical Engineering National Taipei University of Technology, Taiwan
[2] Department of Information Management Kainan University, Taiwan
[3] Department of Electrical Engineering National Cheng Kung University, Taiwan
ctlu@ntut.edu.tw, eyeh@ieee.org, ycwang@ntut.edu.tw, csyang@mail.ee.ncku.edu.tw

Abstract

The user requests of Web systems are often mutually independent, thus can be processed by processors at separate servers/sites. Irregular and/or burst workloads may occur sometimes. Hybrid Clouds could well be suitable for these kinds of Web systems. This research studies the usability and performance of hybrid Clouds for Web systems. Performance metrics measured are response time (RT), throughput and CPU utilization. Analytical results verify that hybrid Clouds can reach the performance level of private Clouds. They also indicate that both the inverse of RT reduction rate and throughput gain are smaller than the corresponding vCPU gain. The network latency of hybrid Clouds obviously affects CPU utilization and performance in some cases.

Public Cloud providers offer various deployment models with different price policies. The performance-cost relation of hybrid Clouds is derived from experimental data. It offers a guideline for selecting suitable deployment model for performance requirements under a cost constraint. For an arbitrary upper bound of cost, the system using the model of less computing capacities performs better than the one using the model of more computing capacities. This paper presents a novel method to adopt Cloud technology to achieve the balance of security, performance, server sprawl, and CPU utilization.

Keywords: Cloud computing, System performance, Virtualization

1 Introduction

Information technology (IT) has advanced rapidly during the last three decades. IT products and services have become daily necessities, similar to electricity and water. The operations of organizations also rely heavily on IT systems. They are a vital infrastructure element to organizations. They provide organizational management with benefits of convenience, efficiency, flexibility, accuracy and productivity. Providing reliable, stable, fast, secure and easy-to-use IT systems is a major challenge to IT engineers.

Many web-based applications, especially websites, may encounter irregular and/or burst workloads. For instance, a college course election system generally encounters a heavy load in a short time interval (such as 20 minutes around the opening of the course election), but very light workloads at other times. To resolve irregular workloads, burst phenomenon, and degraded performance under cost constraint can be a daunting issue for data centers. Traditionally, a data center buys additional servers and/or upgrades existing servers. Either method increases total costs, and may decrease the utilization of the servers. A new approach is to adopt Cloud computing. Public Clouds are generally acceptable to small or startup companies. However, security, and system performance are important concerns to clients. Hybrid Clouds may balance the concerns among security, performance, server sprawl and the utilization of resources. Adopting a hybrid Cloud, the private Cloud has enough capacity to handle low workloads. Both private and public Clouds can work together to handle high workloads. Confidential data can be stored in the private Cloud. However, minimizing the operating cost of an IT service system adopting a hybrid Cloud is a challenging task. A SaaS provider has to maintain its computing capacities while at the same time limiting the number of running VM instances to reduce the renting cost. This research studies the usability of hybrid Clouds against those practical challenges.

Some public Cloud providers, such as Microsoft Azure [1] and Amazon Elastic Compute Cloud (EC2) [2], offer IaaS. They offer different models of virtual machines (VMs) to be adopted by clients with different price policies. For instance, Azure offers five deployment models for VMs [3]. For each model, the price is calculated according to the usage time along with the number of CPUs and the size of RAMs requested by the client. Clients need to request the appropriate deployment model for their applications,

*Corresponding Author: C.-S. Eugene Yeh; E-mail: eyeh@ieee.org
DOI: 10.3966/160792642018011901018

based on their requirements on, for instance, system availability and response time. Selecting the suitable deployment model for use may require performing some tedious experiments. A practical research on it could provide IT engineers with guidance to select the deployment model suitable for their applications with less efforts for laborious experiments.

In a production data center, engineers typically adopt commercial mature products to deploy a Web system, instead of using "experimental" algorithms or software. Therefore, a method for allocating adequate system resources for irregular or burst workloads could assist IT engineers to deliver satisfactory services effectively. This study designs a complex experimentation to study the usability of a hybrid Cloud to enhance the capacities and performance of IT systems and also minimize the cost, by properly choosing deployment models, the number of VM instances and the correct allocation of VMs among the private and public Clouds. Experiments and load testing are performed to study the operating resources required when adopting a hybrid Cloud. The experimental study is conducted using a production course election system with real-world data from Microsoft Azure.

The rest of this paper is organized as follows. Section 2 briefly covers related work. Section 3 describes the architectural design of a hybrid Cloud. Section 4 describes the test environment, and performs experiments to verify the usability of hybrid Clouds. The performance-cost relation of hybrid Clouds is presented in section 5. It is used to find the best minimum-cost solution. Section 6 covers the conclusions and possible future research.

2 Related Work

Cloud computing is gaining popularity and adoption. Many vendors offer Cloud services of IasS, PaaS and/or SaaS. Some vendors even offer hybrid Cloud services. Examples are Amazon EC2 [2], VMWare vCloud [4], IBM Hybrid Cloud Solution [5], and CloudSwitch [6]. The issues and practical solutions of the migration from private data centers to Clouds are an important research topic.

Tak *et al.* [7] investigated the key factors affecting the cost of the migration from private data centers to Clouds. Those factors included workload intensity, growth rate, storage capacity and software license costs. Hajjat *et al.* [8] presented a model to explore the benefits of migrating enterprise services to hybrid Clouds. Consideration elements for the model were enterprise-specific constraints, cost-savings, increased transaction delays and wide-area network costs. They evaluated the model by real enterprise applications on Microsoft Azure. They also indicated the importance and feasibility of having a planned approach to making migration decisions. Altmann *et al.* [9] analyzed the cost factors of federated hybrid Clouds, then developed a cost model to estimate the total cost to customers of using Clouds. They developed an algorithm for the placement of services on hybrid Clouds with minimum cost, and indicated that deployment cost could offset data transfer cost.

Shawky [10] presented an approach to locate the optimal set of components of the system to be migrated to Clouds by minimizing a cost function. Their experimental data indicated that less coupled and more generic components were more suitable for migration, and that Web applications were more suitable for Cloud migration than desktop ones. Guo *et al.* [11] developed a system to dynamically determine and move portions of the applications running in a private Cloud to a public Cloud with minimum costs when workload bursts occurred. The system automates the move process at the proper time.

Cloud services use modern virtualization technology, which goes from the operating system (OS) layer to the hardware abstraction layer [12]. A hypervisor is added between the hardware and guest OSs and software above them. An application can be packaged with the required guest OS and other service software into a VM image. Each VM instance is an independent entity. Public Cloud providers offer various deployment models of VMs with different capacities of resources such as the number of CPUs and the size of memory. Understanding the behavior and performance of deployment models with respect to workloads is important for planning and managing data centers or Cloud services. Lu *et al.* [13-14] showed that a system running with n one-core VM instances performs better than a system with one n-core VM instance. Zhang *et al.* [15] adopted LSQ regression to predict the per-URL requirements and CPU utilizations of a system.

Multicore CPUs are a forefront technology to increase computing powers. Researches on multicore chips are vital for improving the performance of Web systems. Veal *et al.* [16] indicated that multicore technology helped improve the performance of Web applications, but found out that the throughput of a system with eight cores was only 4.8 times that of a system with one core. Hashemian *et al.* [17-18] studied the performance of dynamic and static Web applications on a computer system with two quad-core chips, and concluded that configuration adjustment could increase the utilization of multicore. Cui *et al.* [19] indicated that two OLTP applications running on an 8-core platform had processing speeds of 3.68 and 5.26 times, respectively, more than that on a single-core platform. Harji *et al.* [20] examined in-memory and disk I/O static Web systems, and found out that adjusting the Web server was more effective for enhancing system performance.

3 The Architecture of A Hybrid Cloud

Hybrid Clouds may enable data centers to handle irregular workloads and/or burst problems while achieving cost-saving (e.g. without buying new or upgrading servers). This study performs experiments to verify whether or not hybrid Clouds are able to achieve the same performance level as private Clouds. The system under test (SUT) of the experiments is the production course election system currently used by one of our universities.

A VM image is created for the SUT, as shown in Figure 1. The VM contains the Web server and the course election program written in asp.net. Figure 2 shows the flow chart of the program. The Web server is Microsoft Internet Information Services 7.5. Each VM instance has its own software and resources. The test scripts are generated according to actual user behaviors or use cases.

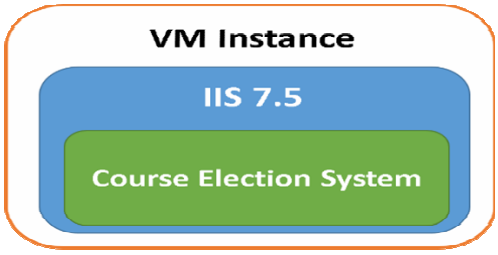

Figure 1. A VM image of the course election system

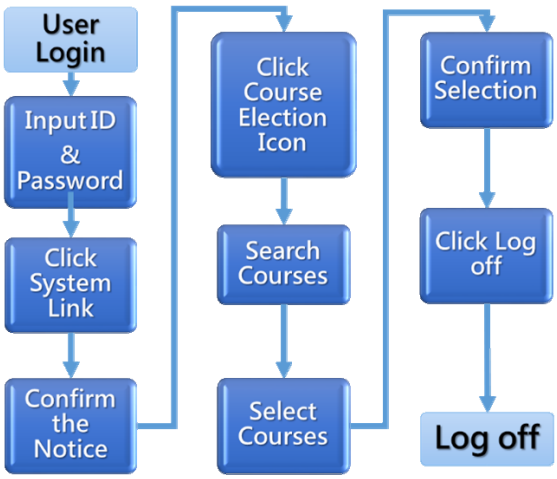

Figure 2. The flow chart of the course election system

Static load testing [21] is adopted to collect test data and calculate the statistics of the response time (RT) and throughput (TP) of the SUT. Microsoft Visual Studio 2010 Load Test is adopted as the load testing tool, as shown in Figure 3. It can perform distributed load testing by simulating multiple users visiting the SUT simultaneously. It needs to work with the Load Test Controller (LTC) and Load Test Agent (LTA). The LTC coordinates LTAs and collects test data. Each LTA simulates multiple users submitting user requests to the SUT.

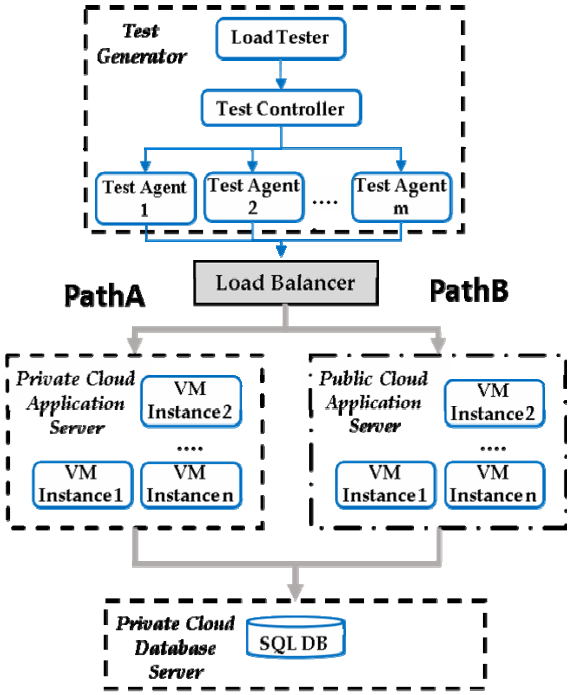

Figure 3. The architecture of a hybrid cloud

Some factors affecting RTs include the application itself, Web server, database server, the number of running VM instances, network latency, load balancing algorithm [21] and others. Hence, when doing load testing, RT, TP, and maximum workload are measured [22]. The analysis of these performance metrics entails the quality improvement of the website.

Load testing can be performed according to users' random behaviors [23]. For a load test, the user's "think time" [24] can be included in the tests. The think time is the time that a user spends to look at a Web page and click on an action button to go to the next Web page. The load tests with think time are similar to actual user behavior. Conversely, no think time needs to be applied to the load tests when measuring the TP of a Web system in order to make it busy always.

The experiments in this study adopt a public Cloud alongside a private Cloud to handle the workloads. Figure 3 shows the architecture of the hybrid Cloud under test. In Figure 3, Path A denotes the tasks initiated by users on the campus, and Path B denotes users requesting the course election service from outside campus. The rapid VM cloning and the elasticity of Cloud computing mean that a new VM instance can be quickly started on the public Cloud when the RT degrades or the workload surges.

The test environment contains several VMs, and allows distributed load tests. Test scripts are sent from the Load Tester to the LTC, then to LTAs, which simulate multiple users to use the service of the SUT.

4 Test Environment and Verification of Cloud Usability

A user session refers to the activities performed within a continuous time interval from login to logoff by a user requesting the service of the SUT. A session is the execution of a test script. For a test run, the Load Tester maintains at any moment a constant number of concurrent sessions (CSs) in the SUT for it to process. The number of CSs (or CS count) is denoted by #CS. A test run lasts for 20 minutes. The RTs and the number of successfully completed sessions are recorded. The same test run is performed three times to form a test case. For a test case, its RT is computed as the average value of the RTs of all successfully completed sessions in it, and its TP is the average number of successfully completed sessions per minute per test run. Each experiment comprises test runs with #CS = 100, 200, …, 1000.

In the experiments, the courses put no limitation on the number of students allowed to take them. Lu *et al.* [13] indicated that a system using multiple one-core VM instances performed better than a system using one multiple-core VM instance. Therefore, this study adopts one virtual core (vCPU) configuration for the private Cloud. Three different Azure deployment models are used for the public Cloud.

4.1 Test Environment

The physical computer used in the private Cloud for the test environment is a Cisco UCS B200 M3 blade server with the specification shown in Table 1. Each VM image in the private Cloud has 1 vCPU and 4 gigabytes of RAMs. The public Cloud is Microsoft Azure, with the specification shown in Table 2. Model A2 (or model A3) has double the resources of model A1 (or model A2). The prices of the deployment models are proportional to the resources allocated to the VM images.

A commercial load balancer is adopted to distribute all independent test scripts to several VM instances. The load balancer is an A10 AX 2500, with the specification shown in Table 3. It provides static LB algorithms such as round-robin, weighted round-robin, least-connection, and weighted least-connection. The experiments are performed using the least-connection algorithm, but other algorithms can also be used.

The hypervisor adopted in the private Cloud is Microsoft Hyper-V 3.0. Table 4 shows the CPU types adopted in the test environment. To isolate the impact caused by database, a physical computer (i.e. a blade) is dedicated to run the VM instance of the Database Management System, which is Microsoft SQL Server 2008 R2. This blade had 12 cores, 40,960 megabytes of memory, and a 10Gbps network card.

Table 1. The specification of the physical server adopted in the private cloud

Feature	Specification
Brand	Cisco UCS blade servers
Process type	Intel Xeon E5-2640
Clock frequency	2.5 GHz
Number of processor chips	2
Cores per chip	6
Total processor cores	12=6*2
Memory size	96 GBytes
Network bandwidth	10 GBytes

Table 2. The specification of Azure deployment models

COMPUTE INSTANCE NAME	CORES	RAM	Cost Ratio
Azure-A1	1	1.75 GB	1
Azure-A2	2	3.5 GB	2
Azure-A3	4	7 GB	4

Table 3. The specification of A10 AX2500 load balancer

Feature	Specification
Process type	Intel Xeon 2.27 GHz
Memory size	6 Gbytes
Network interface	8×1 Gbps Copper
Application throughput	10 Gbps
Maximum Layer 4 connections	300,000 / second
Maximum Layer 7 connections	195,000 / second

Table 4. The CPU types used in the test environment

COMPUTE INSTANCE	CPU
Azure	AMD Opteron 4171 HE 2.10 GHz
Private Cloud	Intel Xeon E5649 2.53 GHz

4.2 Network Latency

The Microsoft Azure data center is located in Singapore. The access to the public Cloud through Internet is unpredictable, unlike the access to the private Cloud through intranet in the campus. The average network latency is measured by Psping tools [25]. Table 5 shows possible paths for accessing the private and public Clouds, and their corresponding average latencies. Table 6 shows the average point-to-point network latencies of three connections.

Table 5. Average network latencies of 2 access paths

Path Name	Path	Average Latency
Path A	User ⇒ Load Balance ⇒ Private VM ⇒ DB ⇒ Private VM ⇒ Load Balance ⇒ Uese	29.46 ms
Path B	User ⇒ Load Balance ⇒ Azure VM ⇒ DB ⇒ Azure VM ⇒ Load Blance ⇒ User	134.58 ms

Table 6. Average point-to-point network latencies

Source	Destination	Average Latency
User	Azure	36.07 ms
Load Balance	Azure	27.04 ms
User	Load Balance	13.21 ms

4.3 Verification of Cloud Usability

This experiment is performed in four test groups, as shown in Table 7. Each test group runs one VM instance. For the public cloud, three deployment models are tested. In the following, E*x*TG*y* denotes test group *y* in experiment *x*.

Table 7. Test groups with one VM instance

Test Group	Deployment Model	VM Instances	Cores per VM	Total Virtual Cores
E4TG1	Azure-A1	1	1 cores	1 cores
E4TG2	Azure-A2	1	2 cores	2 cores
E4TG3	Azure-A3	1	4 cores	4 cores
E4TG4	Private Cloud	1	1 cores	1 cores

E1TG1, E1TG2, and E1TG3 adopt deployment models A1, A2, and A3 provided by Azure, respectively. Figure 4 shows the RTs of the four test groups in this experiment. The RTs of E1TG3 are smaller than the RTs of E1TG4, implying that adding vCPUs in VM instances reduces the queue times of the tasks, and compensates for network latencies. Table 8 shows the trend curves of the RTs. R^2 is the coefficient of determination [26]. The inverse of RT reduction rate is smaller than the corresponding vCPU gain, e.g. (RT of E1TG1) / (RT of E1TG3) = 2.22 < (4 vCPU) / (1 vCPUs) = 4.

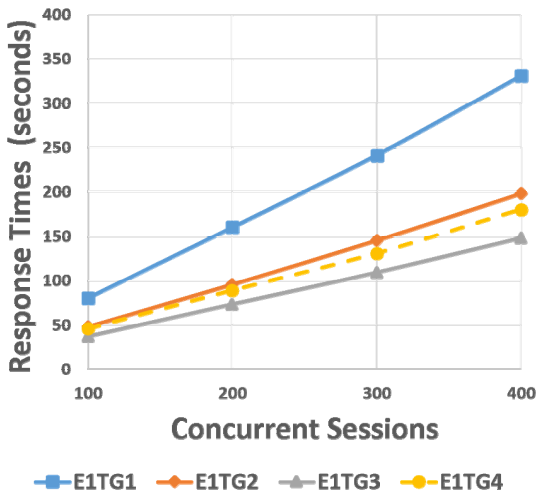

Figure 4. The response times of the experiment

Figure 5 shows the TPs of the experiment. It indicates that the TP gains are lower than the corresponding vCPU gains, e.g. (TPs of E2TG2) / (TPs of E2TG1) < (#vCPU of E2TG2) / (#vCPU of E2TG1) = 2. E1TG3 processes more CSs than E1TG4, and performs best with respect to both RT and TP. These findings indicate that adding more vCPUs into VM instances compensates the delay by the Internet.

Table 8. Trend lines of the response times

Test Group	Equation	R^2	Ratio
E4TG1	$y = 0.8317x - 4.7426$	0.9992	1.86
E4TG2	$y = 0.5005x - 3.333$	0.9995	1.12
E4TG3	$y = 0.3689x - 0.3856$	0.9997	0.83
E4TG4	$y = 0.4461x + 0.0529$	0.9987	1

x: the number of concurrent sessions, y: response time.

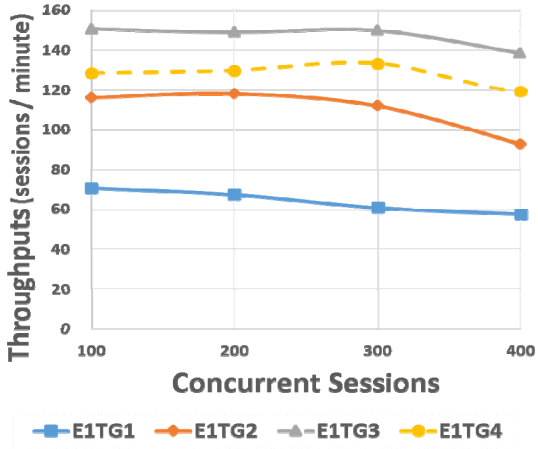

Figure 5. The throughputs of the experiment

4.4 Effect of Multiple VM Instances

This experiment is performed with four test groups, as shown in Table 9. The experiment is similar to the previous one, except that each test group runs 12 VM instances.

Table 9. Test groups with multiple VM instances

Test Group	Deployment Model	VM Instances	Cores per VM	Total Virtual Cores
E4TG1	Azure-A1	12	1 cores	12 cores
E4TG2	Azure-A2	12	2 cores	24 cores
E4TG3	Azure-A3	12	4 cores	48 cores
E4TG4	Private Cloud	12	1 cores	12 cores

From Figure 6, when #CS<300, E2TG1 to E2TG3 have close RTs, which are larger than the RT of E2TG4. E2TG1 to E2TG3 are performed in the hybrid Cloud. Since E2TG4 is performed in an intranet environment, its network latency is negligible. This behavior can be explained as that any one of the four systems is capable of handling the workloads when #CS<300. Hence, the network latency is the key factor for the differences among RTs. The queue times of the workloads have a higher impact on RTs when #CS>400. Therefore the differences of RTs among E2TG1 to E2TG3 are wider and obvious.

The RTs of E2TG2 and E2TG4 are close, possibly indicating that the system using deployment model A2 performs as well as the system using one-core VM in the private Cloud. The regression technique is adopted to calculate the trend equations of the RTs for 400<#CS<1000, which are shown in Table 10.

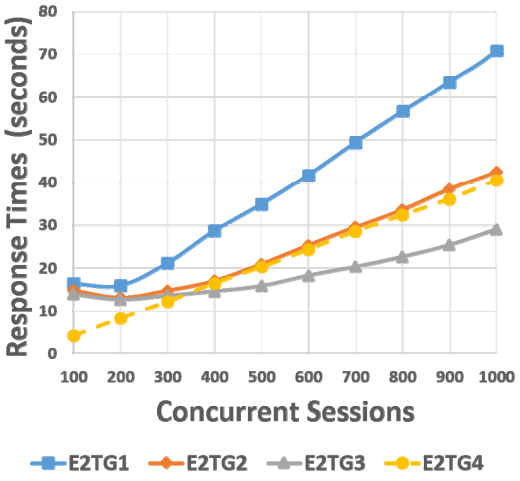

Figure 6. The response times of the experiment

Table 10. Trend lines of the response times

Test Group	Equation	R^2	Ratio
E4TG1	$y = 0.0708x + 0.1003$	0.9945	1.73
E4TG2	$y = 0.0442x - 1.37$	0.9992	1.08
E4TG3	$y = 0.03x - 0.0047$	0.9817	0.73
E4TG4	$y = 0.0409x - 0.2074$	0.9996	1

Note. x: the number of concurrent sessions, y: response time.

Figure 7 shows the TPs in this experiment. The TPs of E2TG4 are close to a constant line. E2TG4 processes 28,493 to 29,508 sessions, while E2TG1 to E2TG3 process 6,876, 7,908 and 8,477 sessions at #CS=100, and 16,497, 27,141 and 35,509 sessions at #CS=500, respectively. Figure 6 and Figure 7 derive the same inequalities for RT reduction rate and TP gain as obtained in sub-section 4.3.

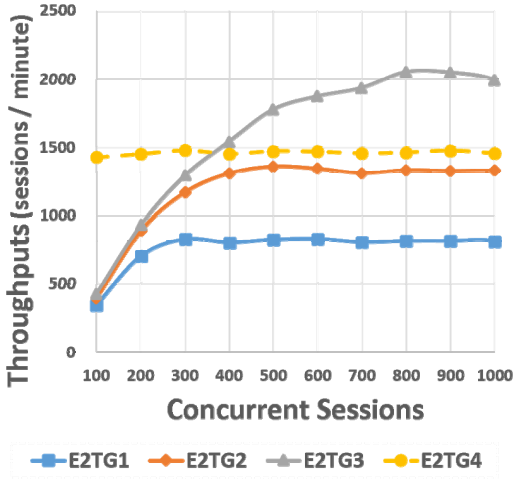

Figure 7. The throughputs of the experiment

The processor utilization of E2TG4 is close to 100%, as indicated in Figure 8, because the network latency of the private Cloud is negligible, thus keeping processors constantly busy. Conversely, due to the network latency factor of the public Cloud, the maximum processor utilizations of E2TG1 to E2TG3 are 62.2%, 88.2% and 92.6%, respectively. In the hybrid Cloud, messages are exchanged between the public Cloud and the private Cloud. E2TG1 to E2TG3 sometimes need to wait for message transmissions through the Internet, reducing the processor utilization. This situation is especially obvious for E2TG1, which has the least computing capacity. The network latency of hybrid Clouds affects obviously their CPU utilization as well as performance in some cases.

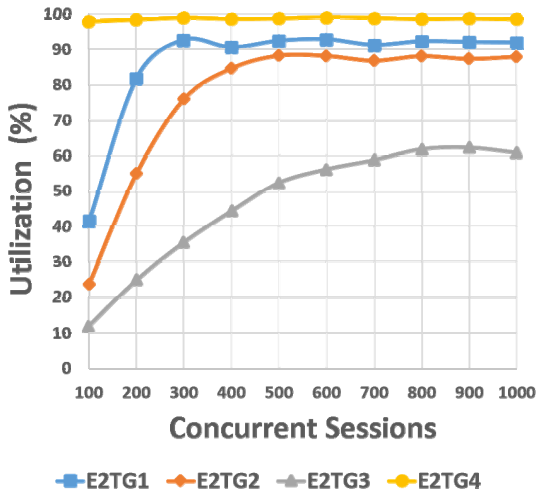

Figure 8. VM vCPU utilization of the experiment

5 The Performance-Cost Relation of a Hybrid Cloud

5.1 Public Cloud Using the Same Deployment Model

This experiment was performed with five test groups, all using Azure-A1, as shown in Table 11. Load tests are performed to study the performance using various numbers of VM instances. The cost is proportional to the number of running VM instances.

Table 11. Various numbers of VM instances of the same deployment model

Test Group	Deployment Model	VM Instances	Vitual Cores per VM	Total Virtual Cores	Cost Ratio
E4TG1	Azure-A1	1	1 cores	1 cores	1
E4TG2	Azure-A1	2	1 cores	2 cores	2
E4TG3	Azure-A1	4	1 cores	4 cores	4
E4TG4	Azure-A1	6	1 cores	6 cores	6
E4TG5	Azure-A1	12	1 cores	12 cores	12

Figure 9 shows the RTs of the test groups in this experiment for #CS=300, where cost ratios are shown in Table 11. The power regression method gives RT = $257.33 \times C^{-1.01}$, where C denotes the cost ratio. The coefficient of determination R^2=0.9933. From the above equation, the more virtual cores in a VM instance, the shorter the RT. As the cost ratio increases, the RT approaches a constant, which is 16 seconds, the shortest in this experiment. RTs include the network transmission times and processing times of the user requests.

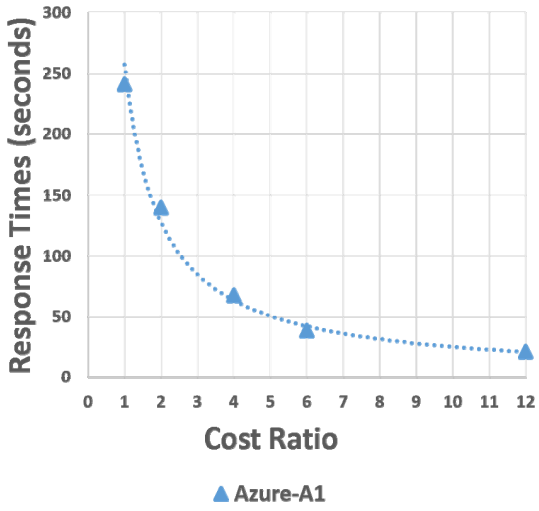

Figure 9. The response times of the experiment at 300 concurrent sessions

5.2 Public Cloud Using Different Deployment Models

This experiment is performed on Azure A2 and A3 models, with three test groups per model, as shown in Table 12.

Table 12. Various combinations of deployment models and the numbers of VM instances

Test Group	Deployment Model	VM Instances	Vitual Cores per VM	Total Virtual Cores	Cost Ratio
E4TG1	Azure-A2	1	2 cores	2 cores	2
E4TG2	Azure-A2	6	2 cores	12 cores	12
E4TG3	Azure-A2	12	2 cores	24 cores	24
E4TG4	Azure-A3	1	4 cores	4 cores	4
E4TG5	Azure-A3	6	4 cores	24 cores	24
E4TG6	Azure-A3	12	4 cores	48 cores	48

Figure 10 shows the RTs of the test groups in this experiment for #CS=300. The regression equations are calculated by the same methods as those in sub-section 5.1. For Azure-A1 model, RT=$257.33 \times C^{-1.01}$ and R^2=0.9933. For Azure-A2 model, RT=$268.98 \times C^{-0.951}$ and R^2=0.9866. For Azure-A3 model, RT=$345.45 \times C^{-0.881}$ and R^2=0.9609. Obviously, Azure-A1 has the smallest RT for #CS=300 under the same cost condition.

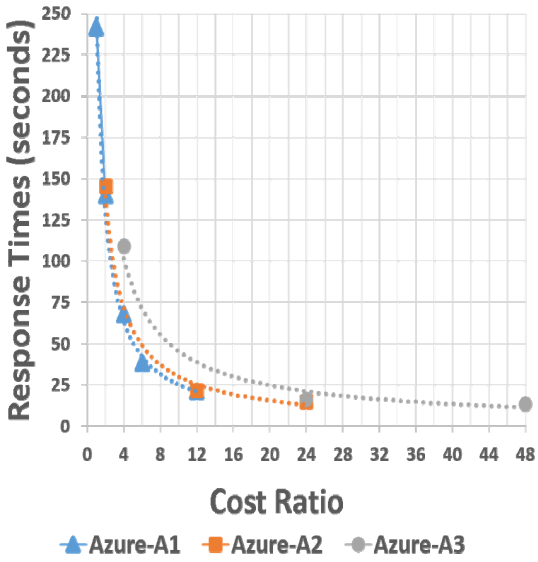

Figure 10. The response times for three Azure deployment models at 300 concurrent sessions

Figure 11 shows the CPU utilization for different deployment models. Azure-A1 had the highest CPU utilization, and Azure-A3 the lowest. As the number of vCPUs increases, the corresponding utilization decreases.

Table 13 shows the minimum numbers of VM instances required under the condition RT < 40 seconds. From the trend equations derived above. Azure A1, A2, and A3 models need at least 7, 4, and 3 VM instances, with corresponding RTs of 34.67, 37.23, and 38.69 seconds, respectively. Thus the Azure-A1 model has the lowest cost to satisfy the RT restriction.

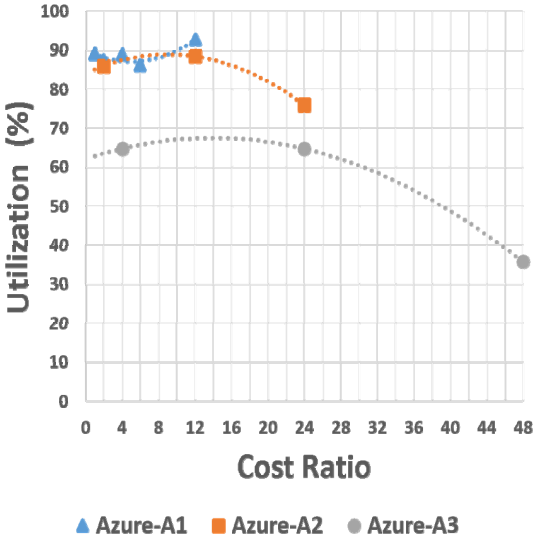

Figure 11. Average CPU utilizations for three deployment models

Table 13. Minimum VM instances required with response times under 40 seconds

Deployment Model	Minimum VM Instance Count	Response Time	Total Virtual Cores	Cost Ratio
Azure-A1	7	34.67	7 cores	1
Azure-A2	4	37.23	8 cores	1.14
Azure-A3	3	38.69	12 cores	1.71

Table 14 shows the computational results for the constraint of RT < 20 seconds. Azure A1, A2, and A3 models need at least 13, 8, and 7 VM instances, with corresponding RTs of 18.8, 19.26, and 18.34 seconds, respectively. Again, the Azure-A1 model had the lowest cost to satisfy the RT restriction.

Table 14. Minimum VM instances required for response times under 20 seconds

Deployment Model	Minimum VM Instance Count	Response Time	Total Virtual Cores	Cost Ratio
Azure-A1	13	18.80	13 cores	1
Azure-A2	8	19.26	16 cores	1.23
Azure-A3	7	18.34	28 cores	2.15

In general, the system using the model of less computing capacities performs better or costs less than the one using the model of more computing capacities. This is because the price of the former model is lower than that of the latter one, so that a system using the former model can run more VM instances.

For the case of the Azure-A1 model, the cost needed for RT < 20 seconds is 13/7=1.85 (< 2) times the cost needed for RT < 40 seconds. Based on the regression equations and the corresponding tables (e.g. Table 13), IT engineers are able to determine the suitable deployment model and the number of VM instances to run their systems in order to minimize the cost and satisfy the response time constraint.

6 Conclusions and Future Work

Organizations are increasingly adopting Cloud computing. Some are building their own private Clouds, some use only public Clouds, and some adopt hybrid Clouds to balance security, performance, server sprawl, and the utilization of resources.

This study designs a complex experimentation to study the usability and performance of hybrid Clouds for some types of Web applications. The performance metrics measured are response time (RT), throughput and CPU utilization. The trend lines of RTs are derived from experimental data in terms of the workloads. Some valuable outcomes are the following. The queue times of tasks are the key factor for the performance of a hybrid Cloud with large workloads, due to the network latency of the public Cloud becomes negligible. Running sufficient VM instances in the public Cloud can increase the performance to the same level as that of the private Cloud. This verifies the usability of hybrid Clouds for Web applications. The experimental data indicate that both the inverse of RT reduction rate and throughput gain are smaller than the corresponding vCPU gain.

Public Cloud providers offer different deployment models with associated price policies. A deployment model with more CPUs and memory has a higher price. Experiments are performed to calculate the regression equations of RTs as functions of cost. The Azure-A1 model has better RTs and throughputs than Azure A2 and A3 models under the same cost condition. Conversely, if the RT is required to be within a specific value, again Azure-A1 model provides the lowest-cost solution than the other two models. This paper presents a novel method to adopt Cloud technology to achieve the balance of performance and cost. A guideline is presented for selecting the suitable deployment model for the performance requirements under a particular cost constraint.

This study can be extended to use other public Clouds (e.g. Amazon EC2) and/or other virtualization platforms (e.g. VMWare). Different combinations of virtualization platforms used in private and public Clouds can also be studied. A more general guideline could be derived to help select an appropriate public Cloud and deployment model for needed services.

Acknowledgment

This work was supported by the Ministry of Science and Technology, Taiwan under Grant MOST 103-2221- E-027-002.

References

[1] Microsoft Azure, *Virtual Machines Documentation*, http://docs.microsoft.com/en-us/documentation/services/virtual-machines/

[2] Amazon, *Amazon EC2*, http://aws.amazon.com/ec2/.

[3] Microsoft Azure, *Linux Virtual Machines Pricing*, http://azure.microsoft.com/en-us/pricing/details/virtual-machines/.

[4] Vmware, *Consistent Infrastructure Consistent Operations*, http://cloud.vmware.com.

[5] IBM, *IBM PureApplication Hybrid Cloud Demo*, https://www.ibm.com/ibm/puresystems/us/en/hybrid-cloud/see-it.html.

[6] Verizon, *Solutions & Products*, http://www.verizonenterprise.com/peoducts

[7] B. C. Tak, B. Urgaonkar, A. Sivasubramaniam, To Move or Not to Move: The Economics of Cloud Computing, *Proc. of the 3rd USENIX conference on Hot Topics in Cloud Computing*, Portland, OR, 2011, pp. 5-10.

[8] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, M. Tawarmalani, Cloudward Bound:

Planning for Beneficial Migration of Enterprise Applications to the Cloud, *ACM SIGCOMM Computer Communication Review*, Vol. 40, No. 4, pp. 243-254, September, 2010.

[9] J. Altmann, M. M. Kashef, Cost Model Based Service Placement in Federated Hybrid Clouds, *Future Generation Computing Systems*, Vol. 41, pp. 79-90, December, 2014.

[10] D. M. Shawky, A Cost-effective Approach for Hybrid Migration to the Cloud, *International Journal of Computer and Information Technology*, Vol. 2, No. 1, pp. 57-63, January, 2013.

[11] T. Guo, U. Sharma, T. Wood, S. Sahu, P. Shenoy, Seagull: Intelligent Cloud Bursting for Enterprise Applications, *USENIX Annual Technical Conference*, Boston, MA, 2012, pp. 361-366.

[12] K. Hwang, G. Fox, J. Dongarra, *Distributed and Cloud Computing: From Parallel Processing to Internet of Things*, Morgan Kaufman, 2013.

[13] C. T. Lu, C. S. Yeh, Y. C. Wang, S. A. Lee, Enhance the Performance of the Campus IT System by Virtualization Technology (in Chinese), *TANET*, Taichung, Taiwan, 2013, #B32-728-1, Session B3.

[14] C. T. Lu, C. S. Yeh, Y. C. Wang, F. T. Tsai, Research on Load Testing of Web Applications with Virtualization (in Chinese), *National Computing Symposium*, Taichung, Taiwan, 2013, pp. 64-69.

[15] Q. Zhang, L. Cherkasova, N. Mi, E. Smirni, A Regression-based Analytic Model for Capacity Planning of Multi-tier Applications, *Cluster Computing*, Vol. 11, No. 3, pp. 197-211, September, 2008.

[16] B. Veal, A. Foong, Performance Scalability of a Multi-core Web Server, *Proc. of the 3rd ACM/IEEE Symposium on Architecture for Networking and Communications Systems*, New York, NY, 2007, pp. 57-66.

[17] R. Hashemian, D. Krishnamurthy, M. Arlitt, N. Carlsson, Characterizing the Scalability of a Web Application on a Multi-core Server, *Concurrency and Computation: Practice and Experience*, Vol. 26, No. 12, pp. 2027-2052, August, 2014.

[18] R. Hashemian, D. Krishnamurthy, M. Arlitt, N. Carlsson, Improving the Scalability of a Multi-core Web Server, *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, New York, NY, 2013, pp. 161-172.

[19] Y. Cui, Y. Chen, Y. Shi, Scaling OLTP Applications on Commodity Multi-core Platforms, *IEEE International Symposium on Performance Analysis of Systems and Software*, White Plains, NY, 2010, pp. 134-143.

[20] A. S. Harji, P. A. Buhr, T. Brecht, Comparing High-performance Multi-core Web-server Architectures, *Proc. of the 5th Annual International Systems and Storage Conference*, New York, NY, 2012, pp. 1-12.

[21] Wikipedia, *Load Testing*, https://en.wikipedia.org/wiki/Load_testing

[22] Y. Pu, M. Xu, Load Testing for Web Applications, *Proceedings of the 1st International Conference on Information Science and Engineering*, Nanjing, China, 2009, pp. 2954-2957.

[23] D. Draheim, J. Grundy, J. Hosking, C. Lutteroth, G. Weber, Realistic Load Testing of Web Applications, *Proc. of the Conference on Software Maintenance and Reengineering*, Bari, Italy, 2006, pp. 57-70.

[24] Microsoft, *Editing Think Times to Simulate Web Site Human Interaction Delays in Load Tests Scenarios*, http://msdn.microsoft.com/ en-us/library/dd997697.aspx

[25] Microsoft, *PsPing v2.0*, http://technet.microsoft.com/zh-tw/sysinternals/ jj729731.aspx.

[26] Wikipedia, *Coefficient of Determination*, https://en.wikipedia.org/wiki/Coefficient_of_determination.

Biographies

Chien-Te Lu received his M.S. degree in E.E. from National Taipei University of Technology (NTUT), Taiwan in 2009. He is currently a Ph.D. student in the E.E. Department, NTUT. His research interests include cloud computing, web and mobile application architecture.

Chiunn-Shyong Yeh received his Ph.D. degree in E.E. from University of Southern California, USA in 1983. He was the Director of National Center for High-performance Computing (NCHC) in Taiwan. Currently he is a professor at Kainan University. His interests include Cloud Computing, information security, and software methodology. He is a member of IEEE (Life Senior) and ACM.

Yung-Chung Wang received the Ph.D. degree in E.E. from National Tsing Hua University, Taiwan in 2000. He has been with E.E. Department, NTUT since 2001, where he is a Full Professor and the Director of Computer Center now. His research interests include cloud computing, wireless networks, software defined network, and performance evaluation of communication networks.

Chu-Sing Yang received the Ph.D. degree in E.E. from National Cheng Kung University (NCKU), Taiwan in 1987. He was a Professor at National Sun Yat-sen University and the Deputy Director at NCHC. He currently is a Professor of Electrical Engineering at the Institute of Computer and Communication Engineering, NCKU. His interests include security, network virtualization, cloud computing, multi-core embedded system, smart grid, and intelligent computing.