2014 ACM/SPEC International Confernce on Performance Engineering, Dublin, Ireland, Mar 23-26 2014, © ACM

Automated Analysis of Performance and Energy Consumption for Cloud Applications

Feifei Chen, John Grundy, Jean-Guy Schneider, Yun Yang and Qiang He
School of Software and Electrical Engineering
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia 3122
{feifeichen,jgrundy,jschneider,yyang,qhe}@swin.edu.au

ABSTRACT
In cloud environments, IT solutions are delivered to users via shared infrastructure. One consequence of this model is that large cloud data centres consume large amounts of energy and produce significant carbon footprints. A key objective of cloud providers is thus to develop resource provisioning and management solutions at minimum energy consumption while still guaranteeing Service Level Agreements (SLAs). However, a thorough understanding of both system performance and energy consumption patterns in complex cloud systems is imperative to achieve a balance of energy efficiency and acceptable performance. In this paper, we present StressCloud, a performance and energy consumption analysis tool for cloud systems. StressCloud can automatically generate load tests and profile system performance and energy consumption data. Using StressCloud, we have conducted extensive experiments to profile and analyse system performance and energy consumption with different types and mixes of runtime tasks. We collected fine-grained energy consumption and performance data with different resource allocation strategies, system configurations and workloads. The experimental results show the correlation coefficients of energy consumption, system resource allocation strategies and workload, as well as the performance of the cloud applications. Our results can be used to guide the design and deployment of cloud applications to balance energy and performance requirements.

Categories and Subject Descriptors
C.4 [**Computer System Organization**]: Performance of Systems; K.4.1 [**Public Policy Issues**]: Use/abuse of power; [**Software Engineering**] D.2: Tools; B.8.2 [**Performance Analysis and Design Aids**]

General Terms
Measurement, Performance, Experimentation

ICPE'14, March 22 - 26 2014, Dublin, Ireland

http://dx.doi.org/10.1145/2568088.2568093

Keywords
Cloud computing; green cloud; energy consumption; performance analysis; automation.

1. INTRODUCTION
Cloud Computing is a new and promising computing paradigm which delivers computing infrastructure as a utility [1]. It provides rented services for computation, application software, and data storage via the Internet. Key advantages for consumers include flexible scaling on demand to their computing and data storage needs without the traditional large upfront investment and continuing maintenance costs of computing infrastructure. Over the last few years many large-scale data centres have been built to meet the massive growth in demand for high performance cloud data and computational services.

As cloud computing becomes more widespread, increasing data storage and computation needs significantly raise the energy consumption of large cloud infrastructures. Most modern data centres are considered as mega data centres [2, 3] because they house over tens of thousands of servers that consume tens of mega-watts of energy per hour at peak times. High energy consumption directly contributes to data centres' operational costs, especially as the energy unit cost continues to rise significantly. Power consumption currently contributes up to 42% of a data centre's monthly expenses [4]. In addition, the huge amount of power consumption of data centers potentially accelerates global climate change. According to a *New York Times* study, data centres use about 30 billion watts of electricity per hour worldwide, equivalent to the output of about 30 nuclear power plants [5]. Therefore, for both financial and environmental reasons, energy consumption has become a critical concern in designing modern cloud-based systems.

Many efforts have been made to improve energy efficiency in cloud environments. Some simple techniques provide basic energy management for servers in cloud environments, including turning on and off servers, putting them to sleep or using Dynamic Voltage/Frequency Scaling (DVFS) [6] to adjust servers' power states. DVFS adjusts the CPU power, and as a result the performance level, according to the workload. However the scope of DVFS optimisation is limited to CPUs. Another approach for improving energy efficiency is to adopt virtualisation techniques to get better resource isolation and reduce infrastructure energy consumption through resource consolidation and live migration [7]. Using virtualisation techniques, several energy-aware resource allocation policies and scheduling algorithms have been proposed to optimise the total energy consumption in cloud environments [8]. However, the

system performance and energy consumption of cloud systems vary greatly with different system configurations and allocation strategies, as well as the workload and the types of running tasks in cloud environments[9].

One of the important requirements for a cloud system is to provide reliable Quality of Service (QoS). Ideally, the performance of a cloud system must not be jeopardised by the energy consumption minimisation. Therefore, a thorough understanding of the performance and energy consumption patterns in complex cloud systems is imperative. We need to learn how energy consumption and cloud system performance are affected by different workloads and system configurations, including cloud application structuring and deployment. In our earlier work, we proposed an energy consumption model for calculating the energy consumption of specific types of tasks in cloud systems [10]. In our model, runtime cloud tasks are divided into three types: computation-intensive, data-intensive and communication-intensive. We conducted experiments to collect fine-grained system performance and energy consumption data with varying system configurations and workloads based on individual types of tasks [11]. However, profiling and analysing system performance and energy consumption in cloud systems is time consuming. Extensive experiments with different parameters, metrics and workloads need to be conducted. Manual generation of load test plans, change of system configurations and application of load tests are very tedious and error-prone. In addition, most of existing cloud system performance and energy profiling approaches limit the types of tasks running in the profiling process to only discrete individual types [11, 12]. In real cloud environments, users send mixes of computation-intensive, data-intensive and communication-intensive tasks to cloud systems simultaneously. The way different types of runtime tasks are composed and deployed will impact the performance and energy consumption of the cloud application [3]. Therefore, it is essential to investigate how different task and resource allocation strategies impact performance and energy consumption.

In order to address these issues, we have developed StressCloud, a performance and energy consumption profiling and analysis tool for cloud systems. StressCloud can effectively and accurately collect the performance and energy consumption data of cloud systems. We adopt stochastic form charts [13] to model realistic cloud user behaviour load. A stochastic form chart is extended from the basic form chart model which is a technology-independent bipartite state diagram used to simulate user behaviour of submit/response systems. From these stochastic from charts we automatically generate load tests and profile the performance and energy consumption data of a cloud system under test. Using StressCloud, we have conducted extensive experiments to empirically analyse the performance and energy consumption of cloud systems. Our experimental results demonstrate the relationship between the performance and energy consumption of cloud systems with different resource allocation strategies and workloads. Our analytical results can be used as guidelines for resource provisioning and task scheduling in cloud systems to maximise performance and minimize energy usage.

Section 2 briefly summarises the state-of-the-art of energy-saving policies, performance and energy consumption profiling and analysis approaches. Section 3 describes the architecture of StressCloud and the profiling framework of performance and energy consumption. The performance and energy consumption profiling setup and methods are described in Section 4. Section 5 presents a range of profiling results and detailed analysis. The observations derived from the experiments are discussed in Section 6. Finally, we summarise our key findings and discuss directions for future research in Section 7.

2. RELATED WORK

Energy-saving policies of cloud systems have been an active research topic in the past few years. VirtualPower [14] is proposed to exploit power management decisions of guest VMs on virtual power states. The virtual power states of guest VMs are considered as preconditions to run local and global energy management policies across the computation. Verma et al. [15] use the characteristics of VMs, such as cache footprint and the set of applications running on the VMs, to drive power-aware placement of VMs. Liu et al. [16] describe a new cloud infrastructure which can dynamically consolidate Virtual Machines (VMs) based on CPU utilisation of servers to identify idle physical servers. Idle physical servers can be turned off to save energy. However these energy saving policies do not take into consideration the workload in cloud systems and hence are very coarse-grained.

Research efforts have also focused on profiling and analysing the energy consumption of cloud systems. Most existing profiling efforts have been conducted using energy benchmarks or closely monitoring the energy profiles of individual system components at runtime, such as CPU, cache, hard disk and memory. Chen et al. [17] develop a linear power model that presents the behaviour and power consumption of individual hardware components of a single physical server. A framework is proposed by Stoess et al. [18] for energy optimisation and the development of energy-aware operation systems based on the availability of energy models for each hardware component. Joulemeter, a power meter for VMs [19], makes use of software components to monitor the resource usage of VMs and then converts the resource usage into energy consumption based on the power model of each individual hardware component. Although some of the profiling and analysis are conducted based on specific applications in cloud systems, the evaluation only includes an individual type of cloud applications. For instance, Lefèvre and Orgerie [20] evaluate the energy efficiency of cloud systems on a multicore platform. However, they focus only on CPU cores and conduct their evaluation of the energy consumption during migration of VMs only with computation-intensive cloud applications.

Some existing research has attempted to leverage the relationship between the performance and energy consumption of cloud systems. Grace et al. [12] investigate the energy efficiency of data centres by running benchmark applications on cloud servers. However, they focus on a black box to benchmark performance and energy consumption of cloud systems without looking into the parameters of the application. Yong and Albert analyse energy efficient utilisation of resources in cloud computing systems [3]. Their results assume that energy consumption scales linearly with the processor without considering the impact of associated RAMs. In fact, cloud resources include not only physical processors but also various RAMs. They conclude that the energy consumption can be reduced when two or more tasks are consolidated rather than solely assigned to one resource. However, they do not consider the performance aspect of such tasks. In our previous research [11], we profile and analyse the performance and energy consumption of cloud systems based on individual types of tasks. The experimental results show that system configurations and task workload highly impact the performance and energy consumption

in cloud systems. However, the types of tasks running in the profiling process are limited to only discrete individual types.

3. STRESSCLOUD

To address the abovementioned issues, we have developed StressCloud, a new tool for profiling the performance and energy consumption of cloud systems. We also conducted extensive and comprehensive experiments to using StressCloud. Our experimental results demonstrate the impact of system resource allocation strategies on system performance and energy consumption; the impact of realistic workloads with mixed types of tasks on system performance and energy consumption; and the relationship between performance and energy consumption.

Based on the high-level workload model and cloud system architecture model specified by the user, StressCloud can automatically deploy load test services to a cloud system and generate load tests. It can also profile the performance and energy consumption of the cloud system automatically. For proposed cloud systems or what-if analysis of proposed re-engineering changes, we allow the user to generate model cloud application services composed of data, compute and communication tasks to load test. In this section, we briefly describe the profiling process, system architecture and user interface of StressCloud.

Figure 1 shows how StressCloud is used to perform load tests to profile the performance and energy consumption of a cloud application. As depicted in Figure 1, the performance engineer first defines the cloud application workload model (1). These are a set of tasks modelling the target cloud application behaviour. Based on the major type of resource consumed by a task, we categorise runtime tasks into three types: computation-intensive, data-intensive and communication-intensive. In real applications, cloud application services are made up of composite tasks that may consume multiple types of cloud resources, including CPU, RAM, data storage and network devices. Thus, we introduce a "composite task" in our workload model to represent such composites. This workload model is then modified by the performance engineer with transition probabilities and properties between different types of tasks to form a workload model. A series of cloud services have been developed in order to model the target cloud application. These services take the user requests to perform tasks defined in the workload model and give corresponding responses. In addition, StressCloud can also stress a real deployed cloud application. Alternatively, instead of specifying a workload model, the performance engineer specifies what deployed cloud services to invoke (1). In this case, the engineer must specify valid requests and data to send to the real deployed application.

For each task, a *stochastic form chart* is created to specify the detailed user requests and required responses from the cloud system. This is a probabilistic model of user and service request behaviour that enables us to model a variety of usage scenarios on cloud application services, whether initiated by users or by other calling services [13]. The performance engineer needs to elaborate the form chart model with suitable probabilities on all transition links between services in the application.

A cloud system architecture model is then defined by the performance engineer to specify the elements in the target cloud system, Figure 1 (2). Our cloud architecture model includes all available resources in the target cloud system and their detailed configurations. After mapping the tasks defined in the user workload model to corresponding resources in the cloud system

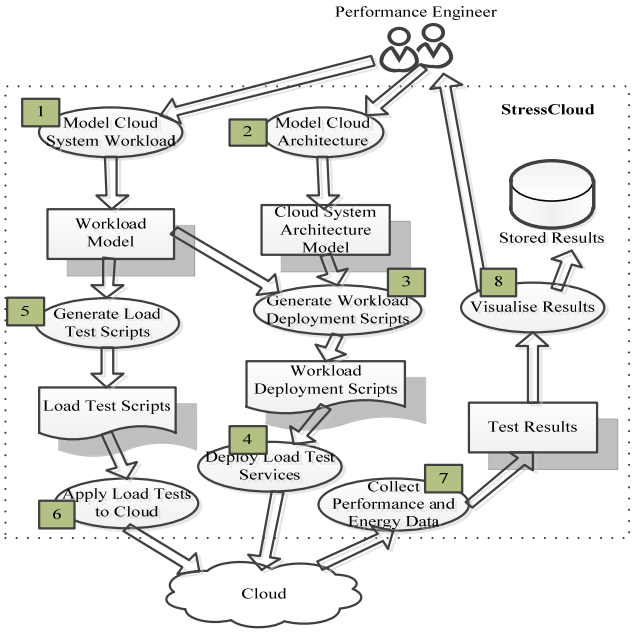


Figure 1. StressCloud Performance and Energy Consumption Data Profiling Process.

architecture model, workload deployment scripts are generated (3). Based on the deployment scripts, load test services are uploaded and deployed to the VMs in the target cloud system (4). These cloud loading services were developed based on our previous research that incorporates CPU, RAM and data-intensive tasks, and support service to service communication-intensive tasks. Load test scripts are then automatically generated based on the workload model (5).

Next, the load tests specified are performed automatically on the target deployed cloud model or application based on the load test scripts (6). The performance and energy consumption information of the target cloud system are collected (7) and visualised (8). The visualised system performance and energy consumption data are updated at a user-specified rate, defaulting to 20 seconds. The test results are stored for future reference and for comparison to new tests run with differing tasks, loads and deployment models.

Figure 2 shows an example of StressCloud in use modelling an exemplar problem - JPetStore[1]. Figure 2 (a) shows a composite model of part of the JPetStore representing data, compute and communication tasks, composed together to form a definition of this cloud application service. For instance, the "*Signin*" task is a composite task of one communication task and one data task. The transition probabilities between different types of tasks have been specified to model the chance of users sending a particular task to the cloud application. A *Client* component represents a client-side start-up component for load test scripts generation, as all testing plans need an entry point. The *Quit* component is also manually added to the generated model to describe the real client behaviour. Figure 2(b) shows an example stochastic form chart model describing one usage scenario of "*GetProductDetail"* task, which is a communication task. The rectangle "*GET*" represents the detailed user requests and the oval "*GETResult*" represents required responses. Figure 2(c) and Figure 2(d) show part of

[1] http://java.sun.com/developer/releases/petstore/

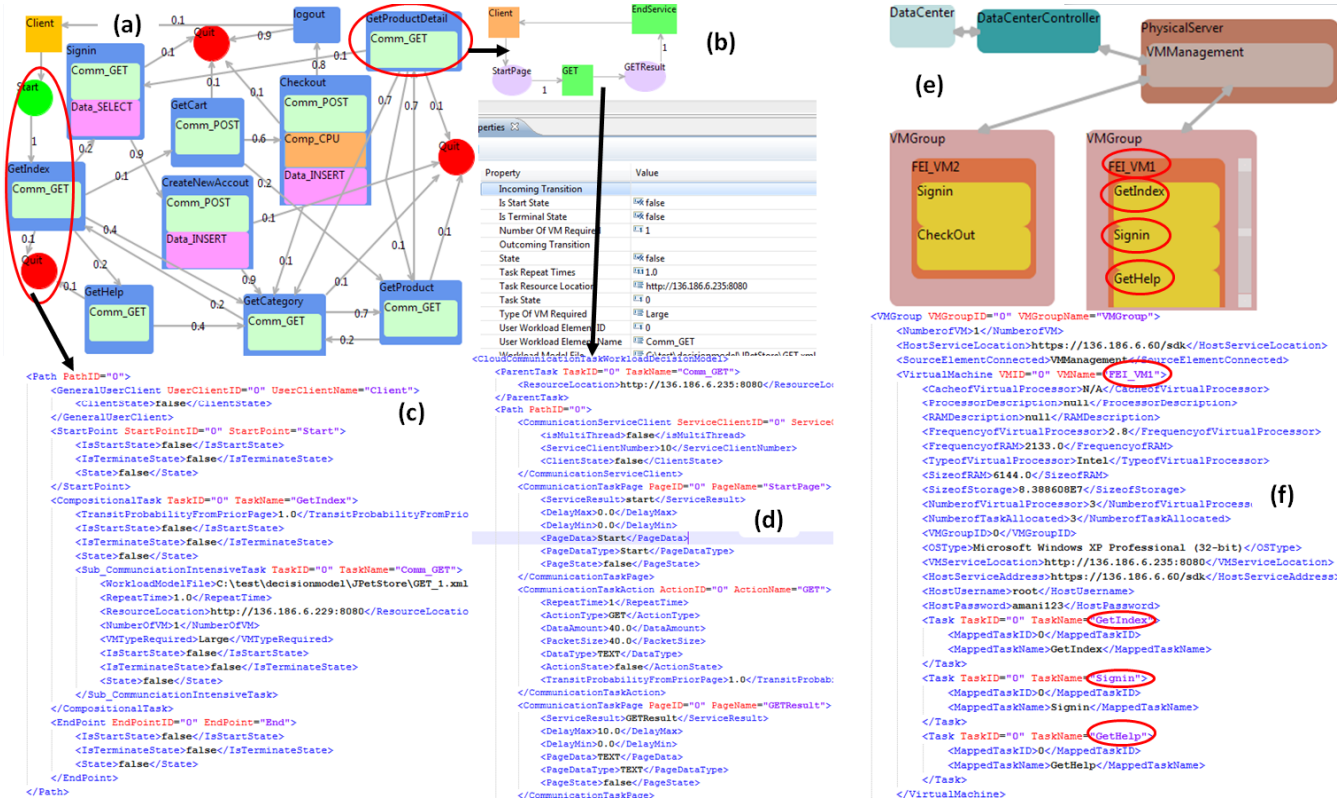


Figure 2. JPetStore Workload Model (a)(b) and Load Test Scripts (c)(d); Cloud Architecture (e) and Deployment Script (f).

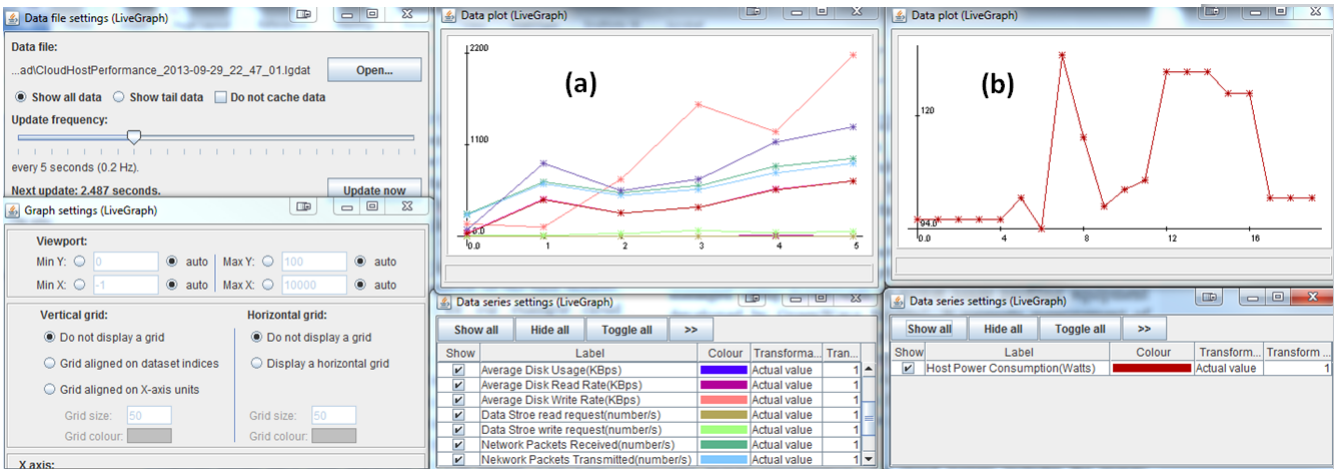


Figure 3. Visualised (a) Performance and (b) Energy Data.

loading scripts generated from a combination of the task models, load models and deployment models for our example cloud application.

Figure 2(e) shows an architecture diagram describing a deployment specification scenario for the cloud application. This shows the cloud environment contains one cloud server. Two VMs have been created on the server and they belong to different VM groups. Tasks "*GetIndex*", "*Signin*" and "*GetHelp*" in the workload model have been deployed on VM named "*FEI_VM1*". Figure 2(f) shows an example script generated from the deployment specification model.

Figure 3(a) shows an example of visualisations of various aspects of cloud system performance, including disk, network and CPU usage, for a running cloud application. The performance engineer can choose and customise the appearance of a range of system KPIs. Figure 3(b) shows the energy consumption of the profiled cloud system.

4. EXPERIMENTAL SETUP

Our new sets of experiments of performance and energy consumption profiling and analysis were performed to replicate and then extend our previous research results [10, 11]. We aimed to collect system performance and energy consumption data for the analysis of the correlation coefficients of system performance,

energy consumption, workloads and resource allocation strategies. The analytical results can be adopted as guidelines for the development of energy efficient cloud resource provisioning and task consolidation strategies.

We profiled the performance and energy consumption of a cloud system by creating heterogeneous VMs in the cloud system and running composite tasks with various workloads and resource allocation strategies. This section describes our experimental setup.

Table 1. Specifications of HP Z400

Basic Specification		Notes
Number of Cores	4	
Number of Threads	2	Intel Hyper-Threading Technology
CPU Frequency	2.8GHz	Fixed CPU Frequency
Memory	8GB	Memory Speed 1333 MHz
Hard Drive	1TB 7200 RTM SATA	
Network Interface	Intel e1000 Gb	

4.1 Test-bed

Our experiments were conducted in SwinCloud, a private cloud that provides a common computational infrastructure to researchers at Swinburne University of Technology. SwinCloud was experimented in the Energy Research Lab (ERL) at Swinburne University of Technology. By using the extensive and sensitive power monitoring facilities provided by the lab, we could precisely monitor the power consumption of the SwinCloud servers. The power consumption measurement was realised and managed using PowerNode, a power usage profiling equipment developed by GreenWave Reality[2]. It supports measurement of both immediate and average power consumption. Collected power data were reported to the GreenWave Gateway, which is used to create a mesh-based Home Area Network (HAN). StressCloud retrieves power consumption from the GreenWave Gateway once every second to guarantee the accuracy of the power consumption data.

Table 2. Type of VM

Virtual Machine	Number of Cores	RAM	Hard Disk
Small	1	2GB	80GB
Medium	2	4GB	80GB
Large	3	6GB	80GB
XLarge	4	8GB	80GB

The energy consumption of a cloud system includes the energy consumed by the constituent servers and the scheduling overhead across the servers. We focused on the energy consumption of individual servers as it is the predominant part [21]. In addition, the cross-server scheduling and communicational overhead of one cloud system can be significantly different from another, depending on the scheduling mechanism adopted by the cloud systems and the distribution of the constituent servers. In this research, we focused on the system performance and energy consumption of tasks running on a single discrete server. The energy consumption incurred by cross-server scheduling and computational overhead is part of our future work.

The server deployed in SwinCloud is a HP Z400. Table 1 lists the specifications of HP Z400. The Virtual Machine Manager (VMM) used is VMware ESX 4.1 and the operating systems running on the VMs are Windows XP Professional. In the experiments, all VMs were assigned with 2GB, 4GB, 6GB or 8GB RAM. The number of virtual CPUs (vCPUs) of each VM varied from 1 to 4 in steps of 1. The number of vCPUs equalled to the number of physical cores assigned to the VM. The configuration scales of the VMs are shown in Table 2.

Figure 4 shows the system performance and energy consumption profiling framework used in our experiments. A PowerNode monitor was connected to the cloud server. StressCloud was installed on a client PC. All workloads were modelled and generated using StressCloud and sent to the cloud server. A series of web services for load tests were deployed on the VMs. These are configured by the generated StressCloud scripts from the cloud application workload models. The system performance and power consumption data were collected by StressCloud for analysis.

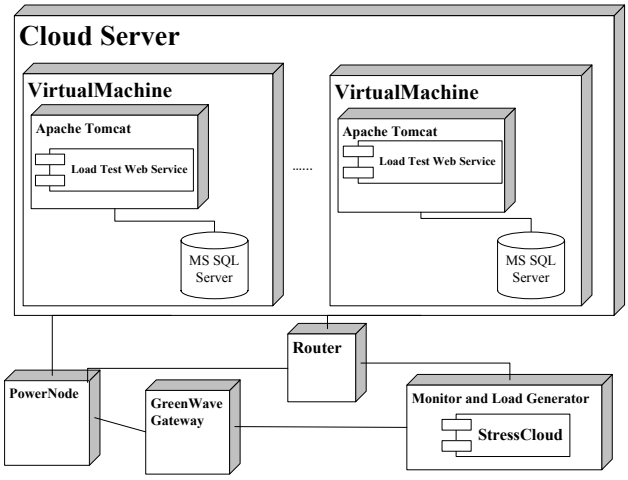


Figure 4. Performance and Energy Data Profiling Framework

4.2 Profiling Method

The define energy consumption for a task as the difference of average power consumption between the server with and without workload multiplied by the execution time of the task. We firstly retrieved the average power consumption measured by PowerNode with no workload in the cloud system as our idle state benchmark. Then, we used StressCloud to retrieve the real-time power consumption measured by PowerNode during the load tests every second. After that, we calculated the average power consumption and then multiplied the average power consumption by the average execution time of a single task to obtain the total energy consumption of the task.

Based on our previous research results [11], system performance and energy consumption are highly influenced by the workload and system configuration. As such, we took the cloud system workloads and system configurations as inputs of our experiments, and set energy consumption and system performance as the outputs. We selected the throughput of the system as one of the key performance indicators (KPI). This is because throughput is

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

often the key performance parameter monitored in cloud systems and it has the advantage of reflecting resource usage accurately [22]. The other KPI selected is the response time as it is a major performance QoS requirement in cloud environments [23]. For computation-intensive tasks, the throughput is defined as the total number of user interactions requested and completed successfully per hour. For data-intensive tasks and communication-intensive tasks, the throughput is defined as the total number of user interactions requested and completed successfully per second. The response time is defined as the interval from the initiation of a request to the receipt of the corresponding response. We also selected and profiled other KPIs, such as CPU usage, memory usage etc.

4.3 Test Case Design

The basic types of cloud workload tasks modelled in our experiments were computation-intensive, data-intensive and communication-intensive, depending on the major resource consumed by the task. We designed and conducted five series of experiments. The first three series of experiments focused on individual types of tasks. We aimed to further investigate the impact of workload and resource allocation strategies on system performance and energy consumption of single type of tasks, as well as validate the correctness and effectiveness of StressCloud. We were able to compare these results to our previous results obtained by hand-developed workload models and loading scripts.

The last two series of experiments focused on the mixed type of tasks, examining energy and performance for tasks with e.g. a 75% compute and 25% data intensive mix of workload. The objective of the last two series of tests was to model the workloads of real cloud applications and investigate the system performance and energy consumption with different resource allocation strategies e.g. what happens when split data and compute load over different VMs vs same VM? Only one aspect was changed in each test set to try and isolate the impact factors of system performance and energy consumption. The detailed five experimental designs are described as follows.

1. Computation-intensive tasks: The major cloud resources consumed by computation-intensive tasks are CPU cores and RAM. We can further divide the computation-intensive tasks into CPU-intensive tasks and memory-intensive tasks. We deployed a web service in StressCloud which calculates a Fibonacci sequence as a representative CPU-intensive task. Each invocation of this web service was a CPU-intensive task. As the largest number of the Fibonacci sequence determined the duration of each calculation, we mapped this number to the workload of each CPU-intensive task – defined as *LN*. We deployed another web service in StressCloud to process big file using memory. The web service consumes as much memory as possible based on the size of memory allocated to it. Each invocation of this web service was a memory-intensive task. The size of processed file determined the workload of the memory-intensive task. We first ran CPU-intensive tasks to calculate Fibonacci sequence and increased the *LN* of the tasks gradually with fixed resources allocated in test suite *I* described in Section 5.1.1. Then, we keep resources allocated to the tasks and *LN* constant. In test suite *II* described in Section 5.1.1, we keep the number the tasks and total resource allocated constant while changing the resource allocation strategy. Another major resource consumed by a computation-intensive task is the RAM. We also measured the energy consumption and system performance by running memory-intensive tasks in test set *I* described in Section 5.1.2. We firstly fixed the resources allocated to the memory-intensive tasks and increased the file size of each task. Then we increased the resources allocated to the tasks while keeping the workload of each task constant.

2. Data-intensive tasks: A data-intensive task in a cloud environment is usually I/O bound and needs to process large volumes of data. It devotes most of its processing time to the movement and manipulation of data in databases or files. In order to investigate the system performance and energy consumption of this type of task, we deployed a web service in StressCloud which could query and manipulate data records on a rational database as representative of data-intensive tasks. In test suite *I* described in Section 5.2, we profiled the system performance and energy consumption of different operation types (query, add, update, delete and combinations) and data size to investigate the impact of different types of database operations on the energy consumption and system performance. The database operations included “insert”, “delete”, “select” and “update”, i.e., the most common ones. In test suite *II* presented in Section 5.2, we mixed all four types of database operations and kept the ratio of each type of operation constant while changing the data size of each operation, the total number of the requests and the resource allocation strategies.

3. Communication-intensive tasks: A communication-intensive task in a cloud application usually generates a huge amount of network transactions between cloud user devices and cloud systems. We have identified that the number of user requests and the data size of each request can highly impact the system performance and energy consumption [11]. In addition, the resource allocation strategies also impact the energy consumption of communication-intensive tasks [24]. Therefore, we profiled and analysed the system performance and energy consumption of communication-intensive tasks with different task workloads and resource allocation strategies. We deployed a web service in StressCloud that took client requests of varying frequency and with varying payload size and generated responses of varying size. In test suite *I* described in Section 5.3, we firstly fixed the resource allocation while increasing the number of user requests and the packet size of each requests. We then fixed the packet size of each request while changing the number of user request per second and resource allocation in test suite *II* described in Section 5.3.

4. **Mixed Computation-intensive and Data-intensive tasks:** Increasing computation and data processing power allow more and more scientific and business applications to be deployed in cloud environments. A scientific task is usually a mix of both computation-intensive and data-intensive tasks [25]. As a rapidly increasing number of scientific tasks have been moved to the cloud, it is important to investigate the system performance and energy consumption of cloud systems with such types of mixed task types. We modelled the client load of some representative scientific applications with mixed computation-intensive and data-intensive tasks using StressCloud. We then analysed the system performance and energy consumption of the target cloud system with different workload models and resource allocation strategies in test suites *I* and *II* described in Section 5.4.

5. Mixed Computation-intensive, Data-intensive and Communication-intensive tasks: As most cloud applications require a Web server to handle user requests and a database server to process the database queries in response to the user requests. Similarly, service-oriented architectures have multiple distributed

compute and data services with significant inter-service communication. Therefore, a cloud application typically has workload tasks composed of a mixture of communication-intensive tasks, data-intensive tasks and computation-intensive tasks. Different services have different mixes of these workload task types. We aimed to investigate the impact of the application workloads and the allocated resources on the system performance and energy consumption of the cloud system. We selected JPetStore as the cloud application to test in our experiment as it has been widely used as a representative Web application that produces a transactional workload. We modelled the workload of JPetStore using StressCloud based on the client load model introduced by Cai [26]. We profiled and analysed the system performance and energy consumption with different client load models and different resource allocation and deployment strategies in test suites *I* and *II* presented in Section 5.5.

5. EXPERIMENTAL RESULTS

We conducted five major sets of tests to analyse the system performance and energy consumption incurred by different types of cloud tasks in order to analyse the impact of workload and resource allocation strategy on system performance and energy consumption. We took system workloads and system configurations as inputs. The energy consumption of each task, the system throughput and the response time were the outputs of our experiments. In order to reduce measurement error, each set of tests was repeated ten times. We evaluated the correctness of StressCloud by comparing the test results of computation-intensive tasks and communication-intensive tasks conducted by StressCloud to our manually obtained previous test results presented in [11]. We analysed the correlation coefficients of energy consumption, system resource allocation strategies and workload, as well as system performance in cloud systems. The results can be used as guidelines to improve overall energy efficiency of cloud systems.

5.1 Computation-Intensive Workloads

A computation-intensive task can be further categorised into CPU-intensive and memory-intensive based on the major resources it consumes. A CPU-intensive task in cloud systems requires a number of isolated processes to perform the computation. A memory-intensive task consumes large amount of memory to store and manipulate data during task execution.

We deployed a web service in StressCloud that calculates Fibonacci sequences as CPU-intensive task. We mapped the largest number of the Fibonacci sequence to the workload of each CPU-intensive task – defined as LN (See Section 4.3). We deployed another web service in StressCloud to process big file using memory. The web service consumes as much memory as possible based on the size of memory allocated to it. Each invocation of this web service was a memory-intensive task. The size of processed file determined the workload of the memory-intensive task.

5.1.1 CPU-Intensive Workload

Test Suite *I*: Keeping the number of tasks constant, while gradually increasing the CPU cores allocated to the task, and the workload of the task.

The total number of tasks was set to 10. This set of tests was initially run on a Small VM (see Table 2 for specification details). We then gradually increased the number of CPU cores configured on the VM in the test. The results of performance and energy consumption are presented in Figure 5 (a) and Figure 5(b). In order to validate the correctness of StressCloud, we compared the results obtained manually, shown in Figure 5(c) and Figure 5(d). We draw the same conclusions from both sets of experimental results. We observed increasing energy consumption per task caused by increasing the LN of the Fibonacci sequence as showed in Figure 5 (a). Moreover, the energy consumption of each task decreased dramatically as the number of CPU cores allocated to the task increased. This is because the execution time of a task will decrease as more CPU cores are allocated to the task. However, the increase in average energy usage rate caused by an extra core is not as much as the execution time of the task. Therefore, the energy consumption decreases accordingly. In addition, we observed a slight turning point of the energy consumption when the number of CPU cores allocated to the task reaches three. For instance, when we set the largest number of the Fibonacci sequence *LN* to 54, the energy consumption with a Large VM increased 3.6% compared to energy consumption with an XLarge VM. This shows that the overhead of scheduling an extra core can cancel out the task running time saved and will also cause more energy consumption. The system throughput is shown in Figure 5(b). As expected, the more resources allocated to the task the better the system throughput obtained. This result shows that, for CPU-intensive tasks, the system throughput rises with the number of allocated cores and the increase of system throughput is nonlinear.

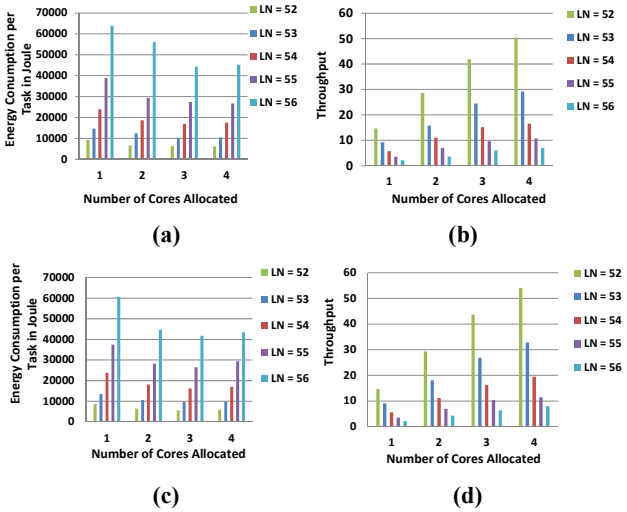
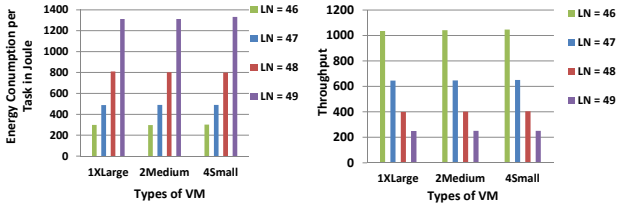


Figure 5. Energy Consumption (a) and Throughput (b) obtained by StressCloud; Energy Consumption (c) and Throughput (d) obtained manually.

Test Suite *II*: Keeping the number of tasks and resource allocated to the tasks constant, while changing the resource allocation strategy. The total number of tasks was set to 16. This set of tests was run on one XLarge VM, two Large VMs and four Small VMs respectively. All workloads were evenly distributed on all the VMs. The results of performance and energy consumption are presented in Figure 6(a) and Figure 6(b). The energy consumption per task and throughput were at the same level under different resource allocation strategies. However, the energy consumption and throughput increased slightly when we changed the resource allocation from one XLarge VM to four Small VMs. For instance, when we set the largest number of the Fibonacci sequence *LN* to 46, the energy consumption increased 1.1% and throughput increased 1.1% with four Small VMs

compared to energy consumption and throughput with an XLarge VM. As the more VMs configured, the more scheduling overhead was introduced. Therefore, energy consumption was higher. On the other hand, more VMs configured make all the running tasks take full advantage of other resources such as memory. Thus, throughput was improved.



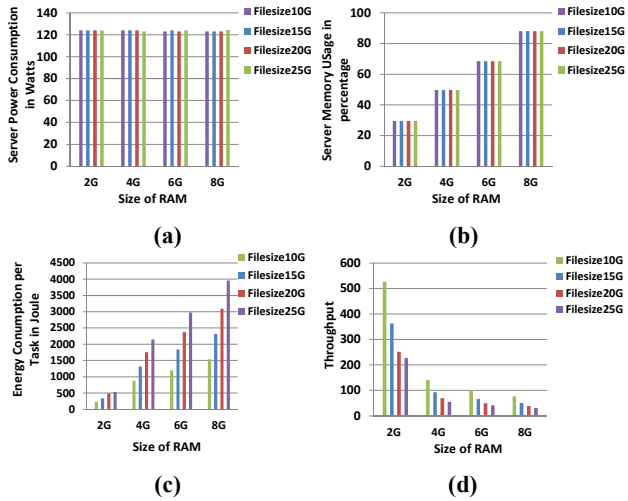
(a) (b)

Figure 6. Energy Consumption (a) and Throughput (b).

5.1.2 Memory-Intensive Workload

Test suite *I*: Keeping the number of tasks constant, while gradually increasing the size of RAM allocated to the task, and the workload of the task.

The total number of tasks was set to 10. This set of tests was initially run on a Small VM. We then gradually increased the size of RAM configured on the VM in the test. With each RAM configuration, the size of file processed was set to 10G, 15G, 20G and 25G respectively. The server power consumption and average memory usage are presented in Figure 7(a) and Figure 7(b). When we increased total memory allocated to the tasks from 2GB to 8GB, the average memory usage of the server increased accordingly as showed in Figure 7 (b). However, the power consumption of the server remained at the same level as displayed in Figure 7(a). Task memory usage has only slight impacted on total power consumption**.** Other research on the power consumption of memory reports that the power consumption of memory remains constant regardless of the workloads. However, power consumption of memory is proportional to the number of memory chips [27]. In addition, the execution time of a task will increase when we increase the size of file processed by the task. Therefore, the energy consumption of each task will increase and the system throughput will decrease, as presented in Figure 7(c) and Figure 7(d).



(a) (b)

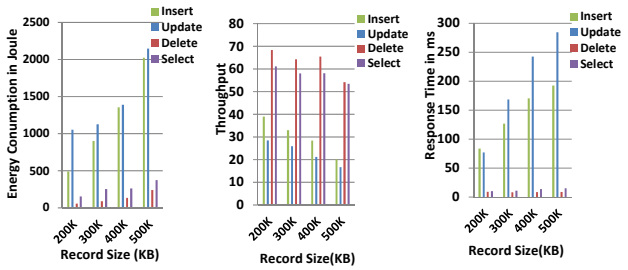
(c) (d)

Figure 7. Server Power Consumption (a) and Memory Usage (b); Task Energy Consumption (c) and Throughput (d).

5.2 Data-Intensive Workloads

A data-intensive task in cloud environment usually involves processing and manipulating large amounts of data to and from storage. We deployed a web service in StressCloud that can query and manipulate data records in a rational database to process data-intensive tasks. We selected mixes of database operations, "insert", "update", "delete" and "select". Each invocation of this web service was a data-intensive task. The size of data processed by the database operations determined the workload of the data-intensive task.

Test suite *I*: Keeping the number of tasks constant, while gradually increasing the workload of each task. The total number of tasks was set to 1000 and user request rate was set to 10 per second. This set of test was run on an XLarge VM. SQL server 2005 was installed to process all the database requests. The result of energy consumption is presented in Figure 8(a). System throughput and response time are presented in Figure 8(b) and Figure 8(c) respectively. As illustrated in Figure 8(a), energy consumption of "insert" and "update" operations increased dramatically when we increased the record size of each request. However, the energy consumption of "delete" and "select" operations only had a slight increase. The "insert" and "update" operations both require reading and writing large amount of data on the disk compared to "select" and "delete" operations, which results in more power consumption of the server. In addition, the response time of "insert" and "update" operations were much longer than "delete" and "select" operations. Therefore, task execution time of "insert" and "update" operations increased, which caused high energy consumption and low throughput as displayed in Figure 8(b).



(a) (b) (c)

Figure 8. Energy Consumption (a), Throughput (b) and Response Time (c).

Test suite *II*: Keeping the ratio of each type of operation and total number of tasks constant while changing record size of database requests and user request number per second. A research on relational database workload characterisation reports that the ratio of "select", "delete", "update" and "insert" in database workload are 75.86%, 4.69%, 7.75% and 11.69% respectively [28]. We adopted the abovementioned ratio of each database operation in our tests. The total number of tasks was set to 1000. We gradually increased the user request rate from 10 to 40. In this set of tests, the record size of each database request was set to 400KB and 500KB respectively. This set of tests was run on an XLarge VM. SQL server 2005 was installed with default configurations to process all the database requests. The system performance and energy consumption are presented in Figure 9. As displayed in Figure 9(b), the throughput decreased slightly while increasing the record size from 400KB to 500KB. This is because response time increased as the record size increased as displayed in Figure 9(c). Therefore, the task

execution time increased and throughput decreased accordingly. As presented in Figure 9(a), the energy consumption of all the tasks increased dramatically as record size increased. Total amount of data read and write on hard disk increased as the record size increased, which resulted in longer task execution time. In addition, the bigger record size introduced more data reading/writing scheduling overhead and the power consumption of the server increased. Therefore, the energy consumption increased. As presented in Figure 9(a) and Figure 9(b), the energy consumption decreased and throughput increased when we increased the number of user requests per second. However, there was a turning point when the number of user requests per second reached 30. For instance, when we set the record size to 500KB and user requests per second to 40, the energy consumption increased 7.8% and the throughput decreased 3.2%. This is because task consolidation will increase the resource utilisation which will reduce the total execution time. However, when the user requests reach 40 per second, the overhead of scheduling and synchronising user requests can cancel out the task running time saved and will cause more energy consumption.

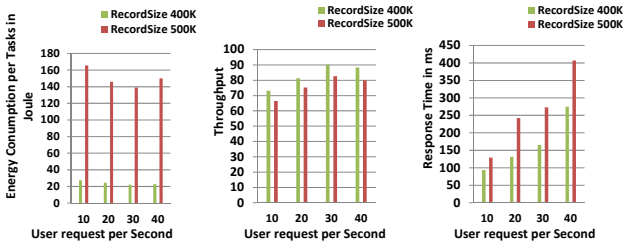


Figure 9. Energy Consumption (a), Throughput (b) and Response Time (c).

5.3 Communication-Intensive Workloads

A communication-intensive task in cloud environments usually generates a huge amount of network transactions between cloud user devices and cloud systems. Therefore, we deployed a web service in StressCloud that handled user requests and generated responses upon the receipt of user requests.

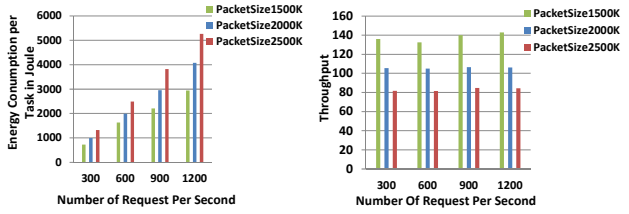


Figure 10. Energy Consumption (a) and Throughput (b).

Test suite *I*: Keep the resource allocation strategy constant while increasing the number of user requests and the packet size of each request. This set of test was running on a Small VM. We increased the user requests per second from 300 to 1200 in steps of 300. The packet size of each user request was increased from 1500KB to 2500KB in steps of 500.The results of energy consumption and throughput are presented in Figure 10. As presented in Figure 10(a), there was an increase in the energy consumption of the task when we increased the packet size. For instance, when we set the user requests per second to 300, the energy increased 36.7% when we increase the packet size from 1500KB to 2500KB. Furthermore, the throughput decreased as the packet size increased. Bigger packet size usually leads to more transmission time over the network and more processing time in the cloud environment. Accordingly, throughput decreases and energy consumption increases for the communication-intensive task.

Test suite *II*: Keep the packet size of each request constant while changing the number of user request per second and resource allocation strategy. The packet size in this test set was set to 2500KB. The results of energy consumption and throughput are presented in Figure 11. When we increased the VM allocated to the task from Small to XLarge, the energy consumption decreased while system throughput increased in general. Intuitively, the more resources used the greater the energy consumption. However in this case, the smaller the instance the higher the disk accesses due to the thrashing of the cache, which leads to increase in energy consumption. Noticeably, when the size of the VM changed from Large to XLarge, the system throughput decreased and the system energy consumption increased in general. When we set the type of VM to XLarge, the total capacity of the VM reached the full capacity of the physical server. The resources left for VM management were less, which led to longer processing time of each user request. Therefore, deploying the task on a Large VM is the most energy efficient.

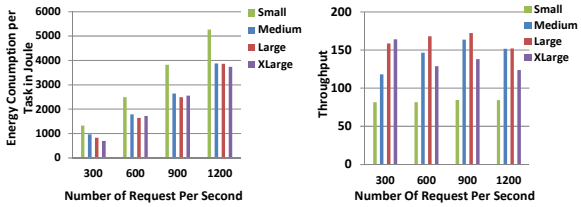
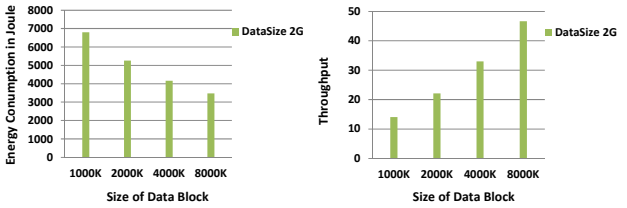


Figure 11. Energy Consumption (a) and Throughput (b).

5.4 Mixed Computation- and Data-Intensive Workloads

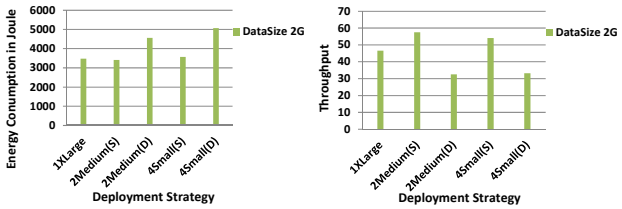
Increasing computation and data processing power allow more and more scientific and business applications to be deployed in cloud environments. A scientific application is both computation-intensive and data-intensive, where computed and retrieved data sets from the cloud data centre are often gigabytes or even terabytes. We modelled the client load of a small scale scientific application with 50% computation-intensive tasks and 50% data-intensive tasks. Firstly, a computation-intensive task and a data-intensive task are executed sequentially. Then the process is repeated until all data have been processed.

Test suite *I*: Keep the resource allocation strategy and total amount of data processed constant, while changing the size of each data set. The scientific application was deployed on an XLarge VM in this set of test. We set the total amount of data processed to 2GB. We increased the data set size of the data-intensive task from 1000KB to 8000KB. The computation-intensive task scale *LN* was increased from 36 to 39. The results of energy consumption and system throughput are presented in Figure 12. From Figure 12(a), we can see that the application energy consumption decreased when we increased the size of the data set in a linear manner. As the size of the data set increased, less overhead information needed to be processed and stored, which led to shorter execution time of the same workload. Therefore, system throughput increased, shown in Figure 12(b).



(a) **(b)**

Figure 12. Energy Consumption (a) and Throughput (b).



(a) **(b)**

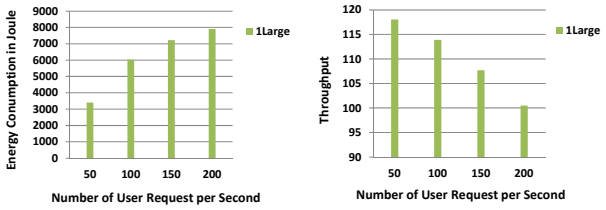
Figure 13. Energy Consumption (a) and Throughput (b).

Test suite *II*: Keep the total amount of data processed and size of each data set constant, while changing the resource allocation strategy. In this set of test, the data set size of the data-intensive task was set to 8000KB and the task scale *LN* of computation-intensive task was set to 39. Firstly, we deployed the scientific application on one XLarge. Then, we deployed the scientific application on two Medium VMs and four Small VMs respectively with computation-intensive task and data-intensive task deployed on the same VM. We named the deployment strategies "2Medium(S)" and "4Small(S)" respectively. The workload was evenly distributed on all the VMs. Finally, we deployed the scientific application to two Medium VMs and four Small VMs respectively with computation-intensive task and data-intensive task on different VMs. We named the deployment strategies "2Medium(D)" and "4Small(D)" respectively. The workload was also evenly distributed on all the VMs. The results of application energy consumption and system throughput are presented in Figure 13. As displayed in Figure 13(a), the energy consumption of the application varied with different deployment strategies. When we deployed the application on the VMs with the same scale, the energy consumption increased when we changed the deployment strategy of the computation-intensive task and data-intensive task from the same VM to different VMs. In contrary, the system throughput increased as displayed in Figure 13(b). For instance, when we change the deployment strategy from "2Medium(S)" to "2Medium(D)", the energy consumption increased 33.5% while throughput decreased 40.3%. This is because deploying the computation-intensive task and data-intensive task on different VMs will introduce more communication overhead between VMs, which will result in more processing time. In addition, when we increase the number of VMs, the energy consumption increased and system throughput decreased no matter how the two kinds of tasks were deployed (on the same VM or different VMs). For instance, when we change the deployment strategy from "2Medium(S)" to "4Small(S)", the energy consumption increased 4.5% and throughput decreased 5.8%. This is because more VMs will introduce extra operation system scheduling overhead, which will cause longer service requests processing time of the cloud application. The more VMs are allocated to the cloud application, the more concurrent processes are created to process the service requests of the cloud application. However, the extra service requests processing time introduced by the extra VM operation system scheduling overhead cannot be cancelled out by the new created concurrent processes. The overall application execution time will be longer. Therefore, energy consumption will increase and throughput will decrease accordingly. In summary, deploying the scientific application on two Medium VMs with all kinds of cloud services on the same VM is the most energy efficient while achieving the best system performance.

5.5 Mixed Computation-, Data- and Communication-Intensive Workloads

Most cloud applications have workload tasks composed of a mix of communication-intensive tasks, data-intensive tasks and computation-intensive tasks. Different services have different mixes of these workload task types. JPetStore was selected as the cloud application to test in our experiment as it has been widely used as a representative Web application that produces a transactional workload. We modelled the workload of JPetStore using StressCloud based on the client load model introduced by Cai [26] and shown in Section 3.

Test suite *I*: Keep the resource allocation strategy constant while changing workload. This cloud application was deployed on a Large VM in this set of test. The initial number of users was set to 10. We increased the concurrent requests number of each user from 50 to 200 in steps of 50. The results of energy consumption and system throughput are presented in Figure 14. As the number of concurrent requests increased, the energy consumption increased as displayed in Figure 14(a). The throughput decreased accordingly as shown in Figure 14(b). Intuitively, more user requests will introduce more scheduling and synchronising overhead in the cloud application, which will result in increase of the processing time of each user request.



(a) **(b)**

Figure 14. Energy Consumption (a) and Throughput (b).

Test suite *II*: Keep the workload constant while changing the resource allocation strategy. The initial number of users was set to 10 and the concurrent user requests of each user were set to 100 in this set of tests. We firstly deployed the cloud application on one Large VM. Then we deployed the cloud application on three Small VMs with computation-intensive tasks, data-intensive tasks and communication-intensive tasks on different VMs respectively, named "3Small(D)". Finally, we deployed the cloud application on three Small VMs with workload evenly distributed on all three VMs, named "3Small(S)". The energy consumption and system throughput are presented in Figure 15. Although the total resources such as CPU and RAM allocated were the same, the energy consumption decreased when deploying the cloud application on multiple VMs compared to deploying the cloud application on single VM as shown in Figure 15(a). The system throughput increased accordingly as displayed in Figure 15(b). When deploying the cloud application on multiple VMs, the

service requests of the cloud application were processed in more concurrent processes, which reduced the execution time of the cloud application. In addition, we observed that the energy consumption with deployment strategy "3Small(S)" increased 0.8% compared to "3Small(D)". The system throughput of "3Small(S)" decreased 2.1% compared to "3Small(D)". This is because in the client workload we have modelled in this test, the majority of all the tasks are communication-intensive. Deploying all communication-intensive tasks on one single VM greatly reduces the overhead of concurrent processes between different VMs. In summary, deploying this cloud application on three Small VMs and separating different types of cloud services on different VM is most energy efficient while achieving the best system performance.

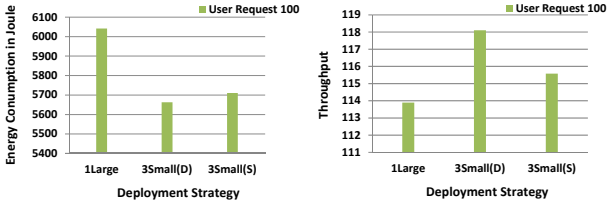


Figure 15. Energy Consumption (a) and Throughput (b).

6. DISCUSSION

Based on our experimental results to date, we have derived a set of guidelines which can be adopted to achieve energy efficient cloud application deployment. Note that performance engineers can use StressCloud to model an application workload and its cloud platform deployment model in a wide variety of ways. They can then generate and run extensive tests and obtain energy and performance data for these specific application models. They can thus make specific judgements for each application and deployment about their best configuration for energy efficiency and performance.

From our results above we see that the organisation of cloud application workload does indeed highly impact energy consumption and system performance. As seen in Section 5.4, when we scaled up each data set processed by the scientific application and kept the total amount of data processed constant, the system performance increased while energy consumption decreased. For some cloud applications, their workload is either known or can be empirically determined and is relatively constant. However, due to the dynamic nature of many cloud applications and the demand of different hosting platforms, the workload of different cloud applications can drastically change over time. The need to find out the workload patterns for different cloud applications, in order to schedule them for optimal performance and energy consumption, has emerged.

The type of cloud application workload impacts energy consumption and system performance. For instance, as discussed in Section 5.1.1, the energy consumption of CPU-intensive task increased dramatically when CPU usage increases. However, as presented in Section 5.1.2, the energy consumption of memory-intensive remained at the same level regardless of the memory usage. The elasticity in the pay-as-you-go cloud business model requires allocating cloud resources to different cloud applications adaptively according to on-demand user requirements. However, it is very challenging as resource under-provisioning will unavoidably jeopardise system performance and cause SLA violations, while resource over-provisioning will result in resource idleness and energy waste. Thus, it is important to accurately predict the resources needed by the cloud application.

For a specific cloud application, the resource allocation strategy can greatly affect the energy consumption and system performance. For instance, deploying the scientific application on two Medium VMs with computation-intensive tasks and data-intensive tasks isolated on two different VMs is the most energy efficient. On the contrary, when we deploy the JPetStore to three Small VMs with all tasks evenly distributed on all the VMs is the most energy efficient. As discussed, both of the abovementioned resource allocation strategies result in the reducing communication and scheduling overhead inside the deployed cloud application. Therefore, it is important to avoid communication overhead within the cloud application when deciding deployment options.

7. SUMMARY AND FUTURE WORK

Understanding performance and energy consumption dynamics is important for the design and deployment of cloud applications to balance energy and performance requirements. In this paper, we presented StressCloud, a new tool for profiling the performance and energy consumption of cloud systems. Using StressCloud, we conducted extensive experiments to profile and analyse system performance and energy consumption with different types and mixes of runtime tasks in a controlled, representative cloud system. We profiled the performance and energy consumption of cloud application models under various task workloads and resource allocation strategies. The correlation of system performance and energy consumption was analysed based on our experimental results. These results provide guidelines for developing resource provisioning and management solutions at minimum energy consumption while still guaranteeing Service Level Agreements (SLAs).

Currently, we are running further experiments including large scale composite workloads on heterogeneous cloud servers. We compare the results of performance and energy consumption with different resource allocation strategies. We analyse overhead of cross-server scheduling and communication overhead of a cloud system. In addition, an energy cost rate and an "energy dirtiness rate" will be adopted to factor in the costs – monetary and environmental - of cloud energy generated by different resources. The energy cost will be investigated in order to achieve the best energy, cost and performance balance in cloud systems.

ACKNOWLEDGMENTS

We thank Professor Ryszard Kowalczyk for providing the facilities of Swinburne Energy Research Lab. This research is partly supported by the Australian Research Council under Discovery Project DP110101340.

REFERENCES

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., et al. *Above the clouds: a Berkeley view of cloud computing*. Technical Report UCB/EECS-2009-28, UC Berkeley Reliable Adaptive Distributed Systems Laboratory, USA, 2009.

[2] Greenberg, A., Hamilton, J., Maltz, D. A. and Patel, P. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM Computer Communication Review*, 39(1):68-73, 2009.

[3] Lee, Y. C. and Zomaya, A. Y. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 60(2):268-280, 2012.

[4] Hamilto, J. Cooperative expendable micro-slice servers (CEMS): low cost, low power servers for internet-scale services. In *Proceedings of the 4th Biennial Conference on Innovative Data Systems Research(CIDR2009)*, pages 1-8, Asilomar, California, USA, 2009.

[5] Babcock, C. *NY Times data center indictment misses the big picture*. InformationWeek Cloud, New York, USA, 2012.

[6] Shang, L., Peh, L.-S. and Jha, N. K. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proceedings of the 9th International Symposium on High-Performance Computer Architecture(HPCA2003)*, pages 91-102, Anaheim, California, USA, 2003.

[7] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., et al. Live migration of virtual machines. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation(NSDI2005)*, pages 273-286, Boston, Massachusetts, USA, 2005.

[8] Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z. and Zhu, X. No "power" struggles: coordinated multi-level power management for the data center. In *Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems(ASPLOS2008)*, pages 48-59, Seattle, WA, USA, 2008.

[9] Zhang, Z. and Fu, S. Characterizing power and energy usage in cloud computing systems. In *Proceedings of the 3rd IEEE International Conference on Cloud Computing Technology and Science(CloudCom2011)*, pages 146-153, Athens, Greece, 2011.

[10] Chen, F., Schneider, J.-G., Yang, Y., Grundy, J. and He, Q. An energy consumption model and analysis tool for Cloud computing environments. In *Proceedings of the 1st International Workshop on Green and Sustainable Software(GREENS2012)*, pages 45-50, Zurich, Switzerland, 2012.

[11] Chen, F., Grundy, J., Yang, Y., Schneider, J.-G. and He, Q. Experimental Analysis of Task-based Energy Consumption in Cloud Computing Systems. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering(ICPE2013)*, pages 295-306, Prague, Czech Republic, 2013.

[12] Metri, G., Srinivasaraghavan, S., Shi, W. and Brockmeyer, M. Experimental Analysis of Application Specific Energy Efficiency of Data Centers with Heterogeneous Servers. In *Proceedings of the IEEE 5th International Conference on Cloud Computing*, pages 786-793, Honolulu, Hawaii, USA, 2012.

[13] Draheim, D., Grundy, J., Hosking, J., Lutteroth, C. and Weber, G. Realistic Load Testing of Web Applications. In *Proceedings of the 10th European Conference on Software Maintenance and Reengineering (CSMR'06)*, pages 70-81, Bari, Italy, 2006.

[14] Nathuji, R. and Schwan, K. VirtualPower: coordinated power management in virtualized enterprise systems. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles(SOSP2007)*, pages 265-278, Stevenson, Washington, USA, 2007.

[15] Verma, A., Ahuja, P. and Neogi, A. Power-aware dynamic placement of hpc applications. In *Proceedings of the 22nd Annual International Conference on Supercomputing(ICS2008)*, pages 175-184, Island of Kos, Greece, 2008.

[16] Liu, L., Wang, H., Liu, X., Jin, X., He, W., et al. GreenCloud: A New Architecture for Green Data Center. In *Proceedings of the 6th International Conference Industry Session on Autonomic Computing and Communications Iindustry Session (ICAC-INDST '09)*, pages 29-38, Barcelona, Spain, 2009.

[17] Chen, Q., Grosso, P., van der Veldt, K., de Laat, C., Hofman, R., et al. Profiling energy consumption of VMs for green cloud computing. In *Proceedings of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing(DASC2011)*, pages 768-775, Sydney, Australia, 2011.

[18] Stoess, J., Lang, C. and Bellosa, F. Energy management for hypervisor-based virtual machines. In *Proceedings of the 2007 USENIX Annual Technical Conference(USENIX2007)*, pages 1-14, Santa Clara, CA, USA, 2007.

[19] Kansal, A., Zhao, F., Kothari, N. and Bhattacharya, A. A. Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM Symposium on Cloud Computing(SoCC2010)*, pages 39-50, Indianapolis, Indiana, USA, 2010.

[20] Lefèvre, L. and Orgerie, A.-C. Designing and evaluating an energy efficient Cloud. *Journal of Supercomputing*, 51(3):352-373, 2010.

[21] Zhang, Q., Cheng, L. and Boutaba, R. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7-18, 2010.

[22] Koller, R., Verma, A. and Neogi, A. WattApp: An Application Aware Power Meter for Shared Data Centers. In *Proceedings of the 7th International Conference on Autonomic Computing(ICAC2010)*, pages 31-40, Reston, VA, USA, 2010.

[23] Wang, Q., Kanemasa, Y., Li, J., Jayasinghe, D., Kawaba, M., et al. Response Time Reliability in Cloud Environments: An Empirical Study of n-Tier Applications at High Resource Utilization. In *Proceedings of the 31st Symposium on Reliable Distributed Systems*, pages 378-383, Irvine, CA, USA, 2012.

[24] Dargie, W., Strunk, A. and Schill, A. Energy-aware service execution. In *Proceedings of the IEEE 36th Conference on Local Computer Networks(LCN)*, pages 1064-1071, Bonn, Germany, 2011.

[25] Yuan, D., Yang, Y., Liu, X., Li, W., Cui, L., et al. A Highly Practical Approach toward Achieving Minimum Data Sets Storage Cost in the Cloud. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 24(6):1234-1244, 2013.

[26] Cai, Y., Grundy, J. and Hosking, J. Synthesizing Client Load Models for Performance Engineering via Web Crawling. In *Proceedings of the 22rd IEEE/ACM International Conference on Automated Sofeware Engineering(ASE)*, pages 353-362, Atlanta, Georgia, USA, 2007.

[27] Peter, K., Bergman, K., Borkar, S., Campbell, D., Carlson, W., et al. *ExaScale Computing Study:Technology Challenges in Achieving Exascale Systems*. FA8650-07-C-7724, 2008.

[28] Yu, P. S., Chen, M.-S., Heiss, H.-U. and Lee, S. On Workload Characterization of Relational Database Environments. *IEEE Transactions on Software Engineering*, 18(4):347-355, 1992.