

Research Article

Secure Hardware Performance Analysis in Virtualized Cloud Environment

Chee-Heng Tan and Ying-Wah Teh

Faculty of Computer Science & Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

Correspondence should be addressed to Chee-Heng Tan; tan.chee.heng@hotmail.com

Received 25 March 2013; Accepted 11 June 2013

Academic Editor: Qingsong Xu

Copyright © 2013 C.-H. Tan and Y.-W. Teh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The main obstacle in mass adoption of cloud computing for database operations is the data security issue. In this paper, it is shown that IT services particularly in hardware performance evaluation in *virtual machine* can be accomplished effectively without IT personnel gaining access to real data for diagnostic and remediation purposes. The proposed mechanisms utilized *TPC-H* benchmark to achieve 2 objectives. First, the underlying hardware performance and consistency is supervised via a control system, which is constructed using a combination of *TPC-H* queries, *linear regression*, and *machine learning* techniques. Second, *linear programming* techniques are employed to provide input to the algorithms that construct stress-testing scenarios in the *virtual machine*, using the combination of *TPC-H* queries. These stress-testing scenarios serve 2 purposes. They provide the boundary resource threshold verification to the first control system, so that periodic training of the synthetic data sets for performance evaluation is not constrained by hardware inadequacy, particularly when the resources in the *virtual machine* are scaled up or down which results in the change of the utilization threshold. Secondly, they provide a platform for response time verification on critical transactions, so that the expected Quality of Service (QoS) from these transactions is assured.

1. Introduction

Cloud computing technology has become a standard model for application and database hosting in today's IT industry, even to serve mission-critical applications. Organizations benefit from this paradigm tremendously due to its scalability, elasticity, and pay-per-use features. With its support for hardware virtualization, almost all applications, serving various functions in an organization, can be hosted on the same platform, with less variety of operational standards. Furthermore, multitenancy hosting solution [1] is made easy with cloud architecture. These significantly reduce the operational efforts needed to support all the IT components in the organization. Realizing these advantages, application and data migration to cloud are extensively studied [2–4]. Optimal performance in the hosting platform ensures that transactions are processed and responded within the boundary of Service Level Agreement (SLA). Such efficiency in the computing hardware also contributes to reduction in capital expenditure as resource wastage is reduced. However, IT services in performance rectification and remediation often

require IT personnel to acquire access to the database to gain visibility of the problem. For instance, in healthcare organization, access by IT personnel into the database host and subsequently into the database which store Protected Health Information (PHI) and Personal Identity Information (PII) information is undesired. Strict Health Insurance Portability and Accountability Act (HIPAA) regulations often required data access only to limited personnel. The breach of this can lead to fines, loss of public confidence in the healthcare services and products, or even jail times in extreme cases of negligence. Besides the healthcare applications, another IT domain that requires tight protection on individual data is the Human Capital Management (HCM) Services. By restricting access to the database, it is difficult for the IT personnel to diagnose and remediate performance issue in the host. In this paper, a mechanism to overcome this problem is envisaged, where a control system utilizing *TPC-H* [5] benchmark data and queries as training data sets instead of real patient data is proposed. *TPC-H* is a decision support benchmark, envisaged and defined by Transaction Processing Performance Council (TPC). The control system

is trained periodically to ensure that the virtual machine's (VM) performance stays consistent and optimal at all time. In this case, as the real data access is avoided for IT management purpose, the data access can be restricted only to relevant medical personnel. *Linear regression* and *machine learning* techniques are employed to construct this system.

Data is often needed to be shared among different locations. Hence, it is imperative for the systems to be highly available and robust as the data is needed by greater community. Some recent interesting studies on utilizing cloud computing for data sharing in the healthcare industry are Utilizing Hybrid cloud to support Electronic Health Records (EHR) [6], palm vein pattern recognition system hosted in the cloud to aggregate patient records across many hospitals [7], and personal health records aggregation and sharing in the Cloud with secure access scheme and encryption [8, 9]. From hosting perspective, this can be best achieved via hardware virtualization, with deployment on either on-premise private cloud or public cloud. The probability of hardware component failure increases as the number of systems and hosts grows. Hardware component failure is the norm rather than an exception in cloud environments [10]. With aggregation of multiple applications and systems in cloud platform, hardware failure in a node can be quickly replaced to preserve the required service level, as these virtualized environments are run on standard and common commodity hardware. The proposed control system is able to detect the degraded hardware performance quickly so that hardware replacement can be accomplished without costly delay.

In cloud environment, frequent provisioning and deprovisioning of hardware is expected. To determine the server load threshold in the host to plan for hardware change, usually extensive load testing that utilizes industrial load testing software is carried out. However, when the VM is placed into production mode, it is difficult to obtain a lengthy downtime to conduct such thorough conventional load testing activities. Here, an effective method is proposed to create high load scenarios in the host, where critical operations bounded by SLA can be tested to ensure satisfactory response time in particular simulated stressed situation in the VM. This load testing method can save time and effort, as the system does not need to give way to the time consuming conventional load testing. In this case *linear programming* technique is employed to develop the proposed load scenario. The same set of *TPC-H* benchmark queries are employed in this case. The hypothetical server load threshold value can be verified in the stressed VM. This value is fed back into the proposed control system, to serve as maximum boundary that the host can take to avoid resource constraint during the periodic data sets training.

The organization of this paper is as follows. Section 2 examines related works on cloud hosting development particularly in hardware resource management and security related topics. Section 3 discusses problem definition and motivation. Section 4 details the learning phase that introduces the mechanism to examine the performance of the underlying hardware and OS which uses *machine learning* and statistical *linear regression* analysis [11, 12] techniques.

Section 5 explains the creation of the stress-testing scenario in the VM, which utilizes *linear programming* technique on the synthetic *TPC-H* queries to verify the revised server load threshold due to hardware change.

2. Related Works

Khatua et al. [13] proposed automated resource optimization algorithms in virtual environment, using the pool of provisioned resources made available to particular application in a virtual machine. In our works, automation is the main intention, where Perl scripting is envisaged to automate the control system for performance evaluation. Ghoshal et al. [14] indicated that close examination on I/O performance is critical to virtualized cloud environments. They evaluated various cloud offerings for High Performance Computing (HPC) applications. The main reason to use *linear programming* technique in Section 5 in our paper is due to the I/O read variables. These reads must be taken into account in the mechanism to stress the VM. Resource optimization is also studied by Wang et al. [15], by taking the perspective of job scheduling.

In the security arena, Lupse et al. [16] proposed private cloud environment to host centralized patient data, where the data is disseminated to various locations via HL7 CDA messages. Ahuja et al. [17] quoted a list of security measures from Cloud Security Alliance. The suggested implementations are multilayer login authentication, robust administrative capabilities to assign appropriate privileges to users and groups, strong password creation and encryption, encrypted data exchange, and multicopy of data backup. The authors also suggest the creation of a specialized cloud solely to serve the healthcare industry, much like the community cloud but with tighter security architecture. Kumar et al. [18] examined the disadvantages of data hosting in cloud and touched base on the researches currently carried out by some prominent cloud providers. Padhy et al. [19] discussed the security challenges of converting patient data from paper records to online data in cloud. Hamlen et al. [20] scrutinized security in cloud environment, particularly at the storage layer and the data layer. The security concern during the mapping between virtual machines and the physical machines in VM migration is discussed. Zhang and Liu [21] examined Electronic Health Record (EHR) dissemination issues, particularly in developments that involve data propagation to multiple locations. The authors also suggested countermeasures techniques to these issues. Donahue [22] revealed some exciting progress in healthcare, made possible by hosting in cloud computing environment.

Some comparisons on security concern between the cloud and data architecture in our study with the literatures are illustrated in Table 1.

3. Problem Definition and Motivation

Figure 1 denotes a typical implementation of application and *Parallel Database* hosting in Private Cloud, utilizing the VMware Cloud Virtualization Infrastructure [23]. The virtual

TABLE 1: The literature comparison.

Authors	Authors' proposal	Our proposal
Lupse et al. [16]	Data exchange using HL7 CDA standard from various cloud sources.	All data reside in 1 cloud cluster to enable better control of data dissemination.
Ahuja et al. [17]	Tighten login credential and privilege grant.	Totally mask patient data from IT personnel.
Kumar et al. [18]	Examine security concerns in data hosting in public cloud.	On-premise virtualized private cloud hosting to reduce variables on security and privacy.
Padhy et al. [19]	Security issue in converting paper records to online records.	Having on-premise private cloud to host the data almost equivalent to in-house data recording on papers.
Hamlen et al. [20]	Security concern during mapping of VM to physical machines, during VM migration.	The cloud architecture in this paper does not envisage VM migration, where resource scalability can be achieved by expanding existing VM.

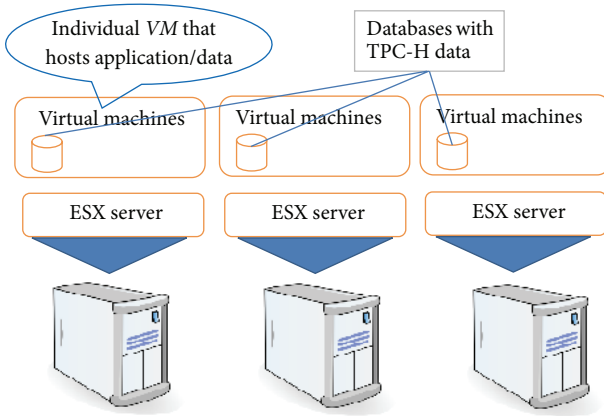


FIGURE 1: Application/data hosting using VMware cloud virtualization infrastructure.

machine (VM) provisioned by this technology offered by VMware Inc. is the platform commonly deployed in the IT industry for virtual hosting purpose. The “ESX server” layer abstracts the hardware resources in the bottom layer and provisions these resources to the upper virtualization layer. Processors, memory, storage, and network resources constitute the lower tier of the hardware resources, while the virtualization tier consists of VMs.

As data security is of interest, private cloud infrastructure is more suitable to be applied as hosting model in the proposals here, as there are security concerns in utilizing public cloud for data hosting which is beyond the scope of this paper. The same sentiment is echoed by Lupse et al. [16]. In Figure 1, the scalability is achieved where hardware resources can be provisioned to the individual VM whenever needed. The ESX server enables flexible hardware provisioning and deprovisioning of resources for particular VM. Each VM is independent, able to host different desired operating systems, for a great variety of functions. For instance, a virtualized environment by 1 ESX server can host database operations, HR applications, and all kinds of other front and back office applications. This diagram is illustrated as it characterizes a common on-premise private cloud configuration. This architecture is typical for Parallel Database hosting, and the virtual machine for experiments in this paper is constituted

of these components. The proof-of-concept environment utilizes Oracle RDBMS hosted on a VM as the database test bed. Oracle is the RDBMS of choice here, as it offers full-fledged database optimizer features. It provides the database transactions with many matured SQL optimization technologies; hence, the need for SQL tuning in all the testing scenarios can be disregarded. The VM is running SUSE Linux operating system, and the captivated server queue length, or server load value obtained from “uptime” command in the OS, is examined in Sections 4 and 5.

To construct the testing environment, the TPC-H benchmark’s data is populated into a database in each VM. This database and the associated TPC-H queries are subsequently employed in Sections 4 and 5 to illustrate the proposed concepts. In Section 4, first the baseline is learned on the queries’ behaviors. Consequently, the training data sets are trained periodically during runtime phase, and comparison on these hypothetical data sets between the baseline and nonbaseline yields understanding on the hardware state for performance evaluation. Runtime phase is termed as the production stage of the application service offering. Section 5 utilized this TPC-H benchmark’s data to compute the stress-testing scenario.

By utilizing the hypothetical workload to achieve the performance evaluation and creation of synthetic stress-testing cases, the objective of not engaging the real data in the actual database for performance evaluation purpose is achieved.

4. Training Phase

4.1. Machine Learning and Training Sets. The baselines in this phase are first obtained when the VM is newly provisioned, using the *machine learning* technique. Subsequently these baselines are reevaluated when the VM’s hardware changes. In this phase, there are 2 objectives to be achieved. First, the server load, or server queue length threshold value using “uptime” command, C_{B-T} for a set of transactions from TPC-H queries is derived. C_{B-T} has dependency on a fixed SQL elapsed time, S_{B-T} value. This value serves as a ballpark figure for subsequent test in the VM to gauge resource adequacy. SQL elapsed time is the total CPU time needed to complete the run of the hypothetical query set. Secondly,

```

Define test time,  $t$  as 2520 s
Define the set of queries to run for the duration of  $t$ 
Define the stabilizing duration as 120 s
 $N = 6$ 
  While  $t > 0$ ,
    Continuously maintain the execution of  $N$  set of queries in the database
    If ( $t == 2000$ )
       $N = 5$ 
    If ( $t == 1600$ )
       $N = 4$ 
    If ( $t == 1200$ )
       $N = 3$ 
    If ( $t == 800$ )
       $N = 2$ 
    If ( $t == 400$ )
       $N = 1$ 
    If ( $t == 0$ )
      Complete test
       $t = t - 1$ 
    If ( $t == 2400$ )
      Start capturing database snapshots to collect values of  $S$  and  $C$  every 30 s
  done

```

ALGORITHM 1

```

Define starting snapshot of the test,  $s$ 
Define ending snapshot of the test,  $e$ 
Define  $t_i$  as begin snapshot for 30 s interval,  $C_i$  as the corresponding server load value
Define  $t_{i+1}$  as end snapshot for 30 s interval,  $C_{i+1}$  as the corresponding server load value
  For  $s \geq t_i$  and  $t_i < e$ 
    If ( $C_i$  does not differ from  $C_{i+1}$  by 10%)
      Record the corresponding SQL Elapsed Time,  $S_i$ 
       $t_i = t_i + 1$ 
  done

```

ALGORITHM 2

the correlation between SQL elapsed time, S , in the database and server load, C , or the gradient, M_B , of the linear lines is obtained via linear regression plot. M_B is asserted as baseline for expected performance on allocated resource in the VM. Iterative learning process is conducted to arrive at the most accurate values of C_{B-T} and M_B . Subsequently in the runtime phase, testing data sets are collected and compared with C_{B-T} and M_B values, to gauge the performance level.

The state of the SQL elapsed time, S , and corresponding server load, C , is captured every 30 s by the database workload repository utility. The database is loaded with continuous execution of the query sets, and they are varied in concurrent number of executions in order to simulate the high to low resource consumption scenarios. In parallel to these load tests, database snapshots are captured every 30 s to provide information at each instance of interest. The algorithm for the test is shown in Algorithm 1. Each set of test in the experiment

in Section 4.2 is automated using Shell script and scheduled to complete in 42 minutes.

It is noteworthy that prolonged test time will produce better accuracy. The server load is taken as in 1-minute average, and it is noticed that these values fluctuate quite substantially during particular instance of time. Hence, some technologists term this parameter as simplistic and poorly defined in Unix environments. However, as demonstrated by experiments, if longer test duration is allocated, this parameter can be useful in measuring the queuing processes information. Algorithm 2 shows how these values are captured. In order to ensure accurate S and C data points, the start and end values of C in the 30 s interval are assessed so that they are less than 10% different from each other to ensure that consistent state is achieved before recording is performed.

During these tests, the resource in the VM must not be constrained. Hence, any data points beyond the server load

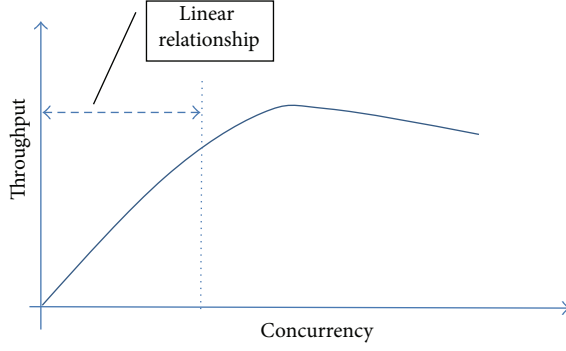


FIGURE 2: Linear correlation between throughput and concurrency processing.

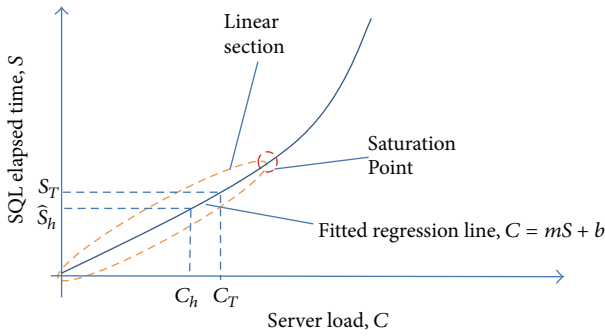


FIGURE 3: Linear correlation between SQL elapsed time and server load.

threshold, C_T , which is a value obtained from Section 5, are discarded.

4.2. Proof of Concept. This section provides an explanation on why linear regression analysis can be applied for the correlation between SQL elapsed time, S , and server load, C .

As experimented by Banga and Druschel [24] and Mosberger and Jin [25], the correlation between throughput and concurrency of processes in a server is linear before a breakpoint, as illustrated in Figure 2. Subsequently according to Little's Law of queuing theory [26, 27], a server's mean queue length, Q , is the product of its response time per visit, R , and throughput, X , which is $Q = R \times X$. Utilizing these concepts, it is derived that ideally the same relationship will apply to database SQL elapsed time, S , and server load, C , as depicted in Figure 3. In this paper, the interest is to ensure that the database transactions are processed within this linear correlation to ensure consistency in hardware performance. If this linear relationship is not conformed to, resource contention, hardware performance degradation, total or partial hardware failure, and undesired OS processes might have occurred in the host.

4.3. Experiments. Figures 4, 5, 6, and 7 show the experimental results. These experiments show the linear relationship between SQL elapsed time, S , in the database and the server load, C , beyond saturation point of the database DB CPU

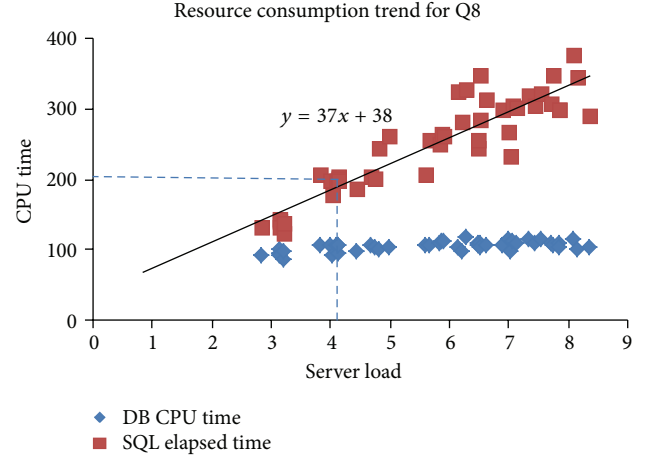


FIGURE 4: Testing result from continuous run of Query 8.

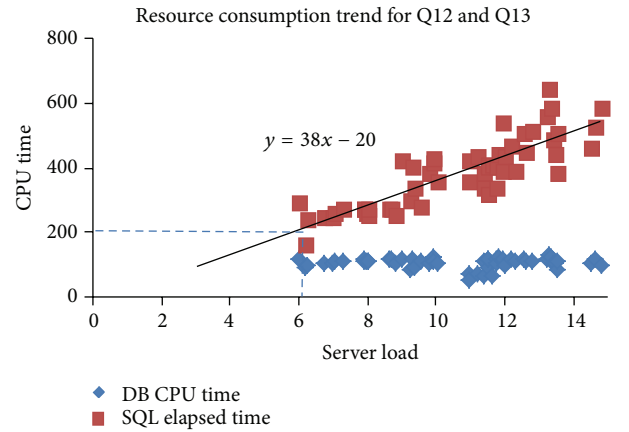


FIGURE 5: Testing result from continuous run of Queries 12 and 13 combination.

time, which is the CPU time needed by the database itself to process the queries. As the interest is to discover the hypothetical server load threshold, C_{B-T} , and gradients, M_B , of the scattered plots, these regression lines computed during the initialization phase of VM provide the baselines for subsequent hardware performance analysis.

Each set of test which consists of different combination of TPC-H queries has different values of C_{B-T} and M_B . This is because the server load, C , has dependency not only on the number of CPU, but also on the logical and physical IO reads. Section 5 details how IO reads are taken into consideration in determining the server load threshold.

These regression lines are formulated using the following equations.

The gradient of the regression line:

$$m = \frac{N \sum_{i=1}^N C_i S_i - \left(\sum_{i=1}^N S_i \right) \left(\sum_{i=1}^N C_i \right)}{N \sum_{i=1}^N S_i^2 - \left(\sum_{i=1}^N S_i \right)^2}. \quad (1)$$

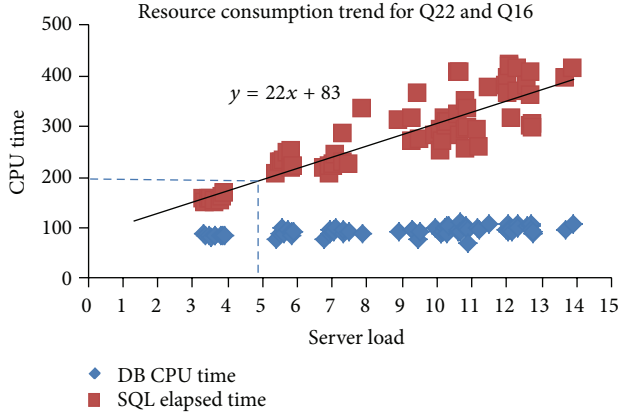


FIGURE 6: Testing result from continuous run of Queries 22 and 16 combination.

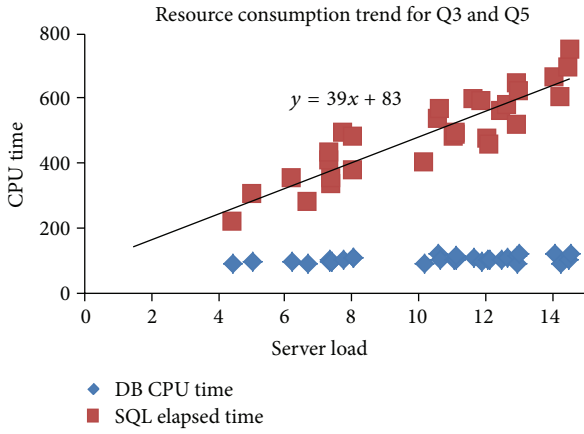


FIGURE 7: Testing result from continuous run of Queries 3 and 5 combination.

The y -intercept:

$$b = \frac{(\sum_{i=1}^N C_i)(\sum_{i=1}^N S_i^2) - (\sum_{i=1}^N S_i)(\sum_{i=1}^N S_i C_i)}{N \sum_{i=1}^N S_i^2 - (\sum_{i=1}^N S_i)^2}. \quad (2)$$

The correlation coefficient:

$$r = \frac{\sum_{i=1}^N S_i C_i - (1/N)(\sum_{i=1}^N S_i)(\sum_{i=1}^N C_i)}{\sqrt{[\sum_{i=1}^N S_i^2 - (1/N)(\sum_{i=1}^N S_i)^2][\sum_{i=1}^N C_i^2 - (1/N)(\sum_{i=1}^N C_i)^2]}}. \quad (3)$$

The correlation coefficient, r , is used in Section 4.4 for performance evaluation purpose.

4.4. Performance Analysis. Figure 8 shows linear regression lines of the resources state in 2 situations in the VM. Correlation A is the Fitted Regression Line obtained from the initial load testing of particular set of TPC-H queries, from the control system in Section 4.1. If the VM shows condition A' from the same test using the same queries after running

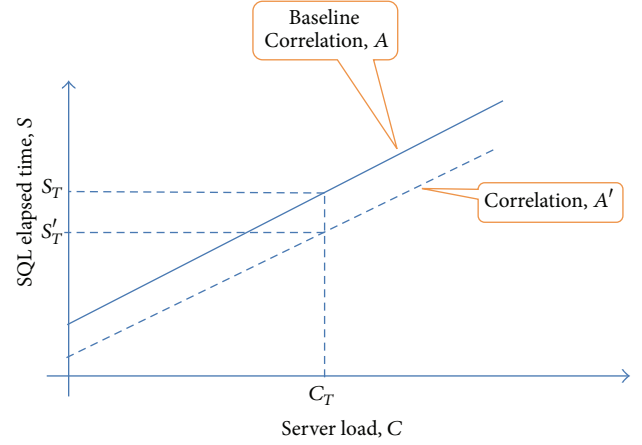


FIGURE 8: Potential change of linear correlation between S and C .

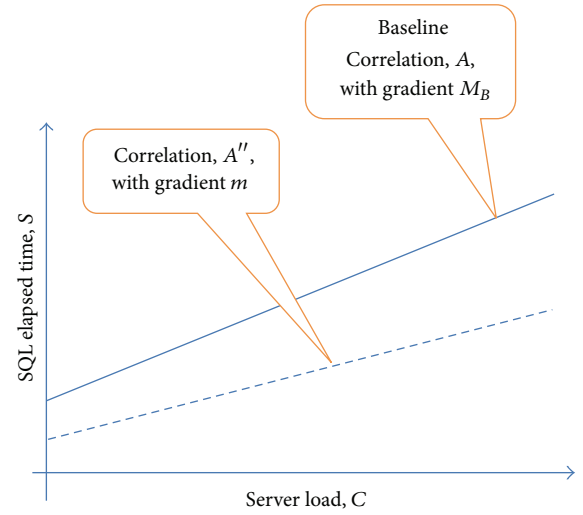


FIGURE 9: Potential change of linear correlation between S and C .

operations for a while, it signals that the capability of the VM has deteriorated. This could be due to persistent noises in the OS or partial hardware malfunction. For this, $\Delta s = ((S_T - S'_T)/S_T) \times 100\%$ is defined.

Using *Fuzzy Computing with Words* [28] concept.

(i) If Δs is large, the OS and hardware condition need to be examined.

The explained theory in Figure 8 assumes strong linear correlation between the SQL processing time and server load. However, this might not be the case in actual production system. In this case, the correlation coefficient, r , [29] is employed. It is a barometric measurement of the linear association between the data points and the subsequent test's Fitted Regression Line. In this case, r will vary between 0 and 1, with value nearer to 1 denoting stronger linear correlation. Correlation A'' in Figure 9 shows such condition.

The fluctuation in the gradient can signify hardware or OS issue. For instance, there are significant persistent noises

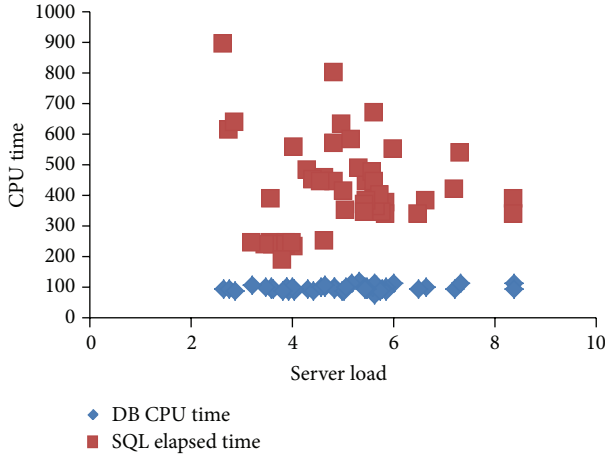


FIGURE 10: Erratic behavior of hardware performance during backup process.

in the OS, and the CPU is not able to access the second core in a dual-core machine, memory shortage due to failure in the memory modules, or the increase in physical I/O operations as a result of partial failure in the SAN storage. m_j is derived from same *TPC-H* query sets, and it has less positive value than M_B , which is the benchmarked gradient obtained from training phase. Again, using *Fuzzy Computing with Words* concept.

(ii) If correlation coefficient, r_j , is less positive, the OS and hardware need to be examined.

Figure 10 shows a condition where the underlying storage is going through a backup process. In this timeframe, the host's environment is not conducive for any transactions, as uncharacteristic performance results are expected due to inconsistent IO subsystem performance during the backup snapshotting. This behavior is shown in this graph, where erratic data points are collected from the load test. Hence, the control system can be used to examine the consistency of the hardware performance.

5. Changed Hardware Problem in Runtime Phase

The most significant parameter to analyze the hardware adequacy in the VM is the server load threshold, C_T , in the VM. This value is obtained from initial load testing of the workloads using a predefined set of *TPC-H* queries. Unless there is significant change in the queries that comprise the workloads, reexecution of load testing is deemed unnecessary, as value of C_T should remain for the tenure of the hosting. If the VM is scaled up in the number of CPU which increases the level of concurrency in the VM, the equitable linear increment of C_T value can be assumed, provided that another hardware component does not post as the constraining factor. In this paper, C_h , which is the 70th percentile value of C_T [30], is taken as the threshold value.

In this following section, a mechanism is illustrated to provide the stress testing platform. At high level, the steps are the following.

- (1) Execute a synthetic mixed workload, to maintain the server load in the VM at renewed C_h level. For analogous comparison of workloads between the initial load testing using real transactions and synthetic mixed workloads, this set of synthetic mixed workloads need to have equivalent number of consistent gets, CG (memory reads), and physical gets, PG (physical reads in IO subsystem), as the real transactions during the before-changed C_h in the initial workload.
- (2) Trigger the set of transactions that are SLA-bound. From here determine if the response time is within required boundary.

Here, the platform for step (1) to be deployed is illustrated in details. The real workload and data is not used here as the objective of this paper is to steer away from using real data for any purpose in the hosting environment.

In this case, *Linear Programming* [31] method is employed. Some deployments using *Linear Programming* are discussed in [32, 33]. This method is suitable in this case, as the objective is to find the most optimized combined run frequency of SQLs that constitute the mixed workload, which has the same amount of CG and PG . This is because the aim is not only to load the VM to the desired server load level, but also to ensure that the memory reads and physical reads are consistent with the baseline load from the initial load testing. This is imperative to ensure that relevant comparative variables are in place for before and after hardware change.

As mentioned, the goal here is to maximize the linear objective function of the dot product of vectors, for response time of SQL and the corresponding run frequency. The general form is quoted as the following:

to maximize

$$\sum_{i,j \in [n]} x_{i,j} (f_i \cdot f_j)$$

(4)

subject to

$$\sum_{i,j \in [n]} a_{i,j,k} (f_i \cdot f_j) \leq b_k \quad \text{for } \forall k.$$

Here, n is the variety of SQLs used to construct the synthetic workload. The SQLs are retrieved from the standard set in *TPC-H* benchmark. f_i denotes the frequency of each SQL, and x_i represents the individual run time of each SQL. To simplify the explanation here, only 2 SQLs are chosen. So now, the objective is

to maximize

$$Q = x_1 f_1 + x_2 f_2$$

subject to

$$0 \leq P_1 f_1 + P_2 f_2 \leq P,$$

$$0 \leq C_1 f_1 + C_2 f_2 \leq C,$$

$$f_1 \geq 0, \quad f_2 \geq 0,$$

because run frequency cannot be negative, (5)

where P_i is the individual PG of SQL_i , P is the total PG , which matches the PG during initial load testing. C_i is the individual CG of SQL_i , C is the total CG , which matches the CG during initial load testing.

The constraints need to be converted to *slack form*, in order to be solved. So,

$$\begin{aligned} P_1 f_1 + P_2 f_2 + s &= P, \\ C_1 f_1 + C_2 f_2 + t &= C, \end{aligned} \quad (6)$$

where s and t are slack variables.

With the aforementioned, the problem can now be solved by *Simplex method* [34]. To illustrate this, the variables' are assumed to have the following values:

$$\begin{aligned} x_1 &= 10s, & x_2 &= 15s, & P_1 &= 400, & P_2 &= 300, \\ P &= 5000, & C_1 &= 400, & C_2 &= 600, & C &= 12500. \end{aligned} \quad (7)$$

In real practice, values of x_1 , x_2 , P_1 , P_2 , C_1 , C_2 can be obtained easily by running the individual SQL in the database. P and C are the values that match the total PG and CG during initial load testing, which corresponds to initial C_h . Now,

$$\begin{aligned} Q - 10f_1 - 15f_2 &= 0, \\ 400f_1 + 300f_2 + s &= 5000, \\ 400f_1 + 600f_2 + t &= 12500. \end{aligned} \quad (8)$$

These data are then put into tableau format, as in Table 2. The italic section shows the data processed by *Simplex* algorithm.

With this result, the *objective equation* becomes

$$Q + 10f_1 + \frac{s}{20} = 250. \quad (9)$$

Hence, the optimized solution is $Q = 250$ as the rule requires the variables in the objective function to be 0. With this value, the optimized values for f_1 and f_2 are obtained. So now,

$$10f_1 + 15f_2 = 250. \quad (10)$$

If $f_1 = 10$, f_2 is then 10. The frequency ratio to run the combination of mixed workload S_1 and S_2 in the new hardware configuration is 1:1. With this ratio, the VM is loaded with the 2 $SQLs$ to reach the new induced C_h . When the VM is stabilized at this level, the SLA-bound transactions

TABLE 2: Tableau depicts the Simplex algorithm. Bold value shows the pivot.

Q	f_1	f_2	s	t	Values
1	-10	-15	0	0	0
0	400	300	1	0	5000
0	400	600	0	1	12500
<i>1</i>	<i>10</i>	<i>0</i>	<i>1/20</i>	<i>0</i>	<i>250</i>
<i>0</i>	<i>4/3</i>	<i>1</i>	<i>1/300</i>	<i>0</i>	<i>50/3</i>
<i>0</i>	<i>-400</i>	<i>0</i>	<i>-2</i>	<i>1</i>	<i>2500</i>

are executed for validation purpose to serve the objective of meeting the SLA requirement.

The previous calculation assumes no change in the database which hosts the *TPC-H* benchmark data and queries. If there are changes in database parameters, particularly those that involve memory alteration, the 2 $SQLs$ need to be rerun and reevaluated, as values for x_1 , x_2 , P_1 , P_2 , C_1 , and C_2 will be different. Nevertheless, P and C are maintained. In practice, when more synthetic $SQLs$ are involved in this synthesized load testing, the mixed workload scenario can be simulated more accurately.

6. Conclusion

In this paper, algorithms that employ *machine learning* and *linear regression* analysis on *TPC-H* benchmark data to support resource performance evaluation in *VM* are presented. Subsequently *linear programming* technique is utilized to construct the stress-testing scenarios in the *VM* for resource threshold and transactions' response time verification when the *VM* undergoes hardware change. In both cases, real data is not involved; hence, the objective to avoid access to real data in order to diagnose and resolve performance issue in *VM* is achieved. These 2 proposals are beneficial to organizations that have stringent requirement on data access. With these proposals, normal IT provider's services are resumed, and at the same time data is secured as required.

7. Future Works

The heuristic method of finding the adequate load needed during the learning phase will need further enhancement and automation, in order to avoid underloading or overloading scenarios, for the linear correlation to sustain. It is also interesting to learn and automate the fitting run duration in this phase for accurate data capture. At this juncture, the test bed environment is provisioned on a *VM* running an Oracle database, utilizing SQL as querying and loading method. It will be interesting to efficiently detail the same concept on nodes in Hadoop MapReduce cluster.

Acknowledgment

This research has been funded by the University of Malaya, under the Grant no. RG097/12ICT.

References

- [1] L. Z. Cui, T. T. Zhang, G. Q. Xu, and D. Yuan, "A scheduling algorithm for multi-tenants instance-intensive workflows," *Applied Mathematics & Information Sciences*, vol. 7, no. 1, pp. 99–105, 2013.
- [2] W. Chen, K. C. Yin, D. L. Yang, and M. C. Hung, "Data migration from grid to cloud computing," *Applied Mathematics & Information Sciences*, vol. 7, no. 1, pp. 399–406, 2013.
- [3] M. A. Babar and M. A. Chauhan, "A tale of migration to cloud computing for sharing experiences and observations," in *Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing (SELOUD '11)*, pp. 50–56, May 2011.
- [4] V. Kumar and K. K. Garg, "Migration of services to the cloud environment: challenges and best practices," *International Journal of Computer Applications*, vol. 1, no. 6, pp. 1–6, 2012.
- [5] TPC, 2012.
- [6] G. Fernández-Cardenosa, I. de la Torre-Díez, M. López-Coronado, and J. J. P. C. Rodrigues, "Analysis of cloud-based solutions on EHRs systems in different scenarios," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3777–3782, 2012.
- [7] N. Karthikeyan and R. Sukanesh, "Cloud based emergency health care information service in India," *Journal of Medical Systems*, vol. 32, no. 6, pp. 4031–4036, 2012.
- [8] T. S. Chen, C. H. Liu, T. L. Chen, C. S. Chen, J. G. Bau, and T. C. Lin, "Secure dynamic access control scheme of PHR in cloud computing," *Journal of Medical Systems*, vol. 36, no. 6, pp. 4005–4020, 2012.
- [9] M. Li, S. C. Yu, Y. Zheng, K. Ren, and W. J. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.
- [10] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10)*, pp. 193–204, June 2010.
- [11] M. Bichler and T. Setzer, "Admission control for media on demand services," *Service Oriented Computing and Applications*, vol. 1, no. 1, pp. 65–73, 2007.
- [12] J. Neter, M. Kutner, C. Nachtsheim, and W. Wasserman, *Applied Linear Regression Models*, The McGraw-Hill Companies, New York, NY, USA, 1996.
- [13] S. Khatua, A. Ghosh, and N. Mukherjee, "Optimizing the utilization of virtual resources in cloud environment," in *Proceedings of the 8th IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS '10)*, pp. 82–87, Taranto, Italy, September 2010.
- [14] D. Ghoshal, R. S. Canon, and L. Ramakrishnan, "I/O performance of virtualized cloud environments," in *Proceedings of the 2nd International Workshop on Data Intensive Computing in the Clouds*, pp. 71–80, November 2011.
- [15] X. W. Wang, J. J. Sun, H. X. Li, C. Wu, and M. Huang, "A reverse auction based allocation mechanism in the cloud computing environment," *Applied Mathematics & Information Sciences*, vol. 7, no. 1, pp. 75–84, 2013.
- [16] O. Lupse, M. Vida, and L. Stoicu-Tivadar, "Cloud computing and interoperability in healthcare information systems," in *The 1st International Conference on Intelligent Systems and Applications*, 2012.
- [17] S. Ahuja, S. Mani, and J. Zambrano, "A survey of the state of cloud computing in healthcare," *Network & Communication Technologies*, vol. 1, no. 2, 2012.
- [18] V. Kumar, M. Swetha, and M. S. P. S. Muneshwara, "Cloud computing: towards case study of data security mechanism," *International Journal of Advanced Technology and Engineering Research*, vol. 2, no. 4, 2012.
- [19] R. Padhy, M. Patra, and S. Satapathy, "Design and implementation of a cloud based rural healthcare information system model," *Universal Journal of Applied Computer Science and Technology*, vol. 2, no. 1, pp. 149–157, 2012.
- [20] K. Hamlen, M. Kantarcioglu, L. Khan, and B. Thuraisingham, "Security issues for cloud computing," *International Journal of Information Security and Privacy*, vol. 4, no. 2, pp. 39–51, 2010.
- [21] R. Zhang and L. Liu, "Security models and requirements for healthcare application clouds," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 268–275, Miami, FL, USA, July 2010.
- [22] S. Donahue, "Can cloud computing help fix health care," *Cloudbook Journal*, vol. 1, no. 6, 2010.
- [23] Anonymous, *VM Ware Infrastructure Architecture Overview*, 2006.
- [24] G. Banga and P. Druschel, "Measuring the capacity of a web server," in *Proceeding of the USENIX Symposium on Internet Technologies & Systems*, 1997.
- [25] D. Mosberger and T. Jin, *HttpperfMA Tool For Measuring Web Server Performance*, 1998.
- [26] A. O. Allen, *Probability, Statistics, and Queueing Theory*, Academic Press, Boston, Mass, USA, 2nd edition, 1990.
- [27] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Operations Research*, vol. 9, pp. 383–387, 1961.
- [28] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, 1996.
- [29] D. G. Kleinbaum, L. L. Kupper, K. E. Muller, and A. Nizam, *Applied Regression Analysis and Other Multivariable Methods*, Duxbury Press, Pacific Grove, CA, USA, 1998.
- [30] C. H. Tan and Y. W. Teh, "Harnessing cloud computing for dynamic resource requirement by database workloads," *Journal of Information Science and Engineering*, vol. 29, no. 5, 2013.
- [31] L. Vandenberghe, S. Boyd, and M. Nouralishahi, "Robust linear programming and optimal control," in *Proceedings of the 15th Ifac World Congress on the International Federation of Automatic Control*, pp. 293–293, Barcelona, Spain, 2002.
- [32] H. Wang, W. Xu, F. Wang, Jia, and C. ., "A cloud-computing-based data placement strategy in high speed railway," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 396387, 15 pages, 2012.
- [33] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebretd, "Applications of second-order cone programming," in *Proceedings of the International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing*, pp. 193–228, 1998.
- [34] S. M. Sinha, *Mathematical Programming—Theory and Methods*, Elsevier Science, New Delhi, India, 2006.

