Web Server Performance Evaluation in a Virtualisation Environment

Performance evaluation of Web Server

Manjur Kolhar
Dept. Computer Science
Prince Sattam Bin Abdulaziz University,
Wadi Ad Dawaser, KSA

***Abstract*—Operational and investment costs are reduced by resource sharing in virtual machine (VM) environments, which also results in an overhead for hosted services. VM machine performance is important because of resource contention. If an application takes a long time to execute because of its CPU or network, it is considered to be a failure because if many VMs are running over a single hardware platform, there will be competition for shared resources, e.g., the CPU, network bandwidth, and memory. Therefore, this study focuses on measuring the performance of a web server under a virtual environment and comparing those results with that from a dedicated machine. We found that the difference between the two sets of results is largely negligible. However, in some areas, one approach performed better than the other.**

Keywords*—Cloud computing; virtual machine; resource sharing; latency sensitive; web server; multi-tier application

I. Introduction

Cloud computing allows the user to store and access data over the Internet using virtualisation technology. Virtualisation is software that serves as an intermediary between the physical network and the cloud. Furthermore, virtualisation allows a cloud service provider (CSP) to run multiple operating systems using a VM over a single hardware system, which reduces operating and investment costs. Virtualisation and cloud computing differ in that virtualisation runs on hardware, whereas cloud computing is a service resulting from the virtualisation [1–2]. CSPs provide these services through a co-tenant scheme. Services or applications running in a virtualised environment demand more processing power from the host hardware system [3-4]. Additionally, virtualisation overhead may occur because of the processing time for various services and tenant schemes. Hence, it is necessary to measure the behavior of an application under various virtual environmental conditions before moving an application permanently to the cloud. Latency-sensitive applications suffer because of resource, network, and CPU sharing, eliminating these virtualisation benefits [5].

In this study, we host a web server on a VM to measure latency-sensitive elements influencing its performance. The performance of a client-side application, i.e., a web browser, is also measured because it is involved in request and response transactions that involves the sharing of major resources, including memory, network I/O virtualisation, and CPU.

In this study, we evaluate a web server's performance in a virtualisation environment and compare it with that of a dedicated web server running on a machine without a VM. We compare with a baseline system for application performance and security resource consumption – that is, a web server and a web server on the client machine – because performance and resource consumption depend on the virtualisation configurations. Additionally, to secure hosted applications, CSPs install security patches on the cloud. This setup induces more latency in the application environment. Therefore, we present the results of our experiments to answer following questions:

- How does the virtualised web server's performance compare to the performance of a dedicated server, including request and response time?
- How is web browser performance affected when multiple tiers are used?
- As the number of multi-tiered applications increases, how is the web browser served to users under the influence of VM (e.g., during content loading)?

This manuscript is organised as follows. First, in Section 2, we summarise past literature. In Section 3, we discuss the methodology for measuring the CPU, network, and other modules that influence the performance of the virtual and dedicated environments. In Section 4, we evaluate the given methodology, and finally, in Section 5, we outline our conclusions.

II. Literature Survey

Real-time data transmission over a network is built on the assumption that the response and request primitives are executed in a specific time, no messages are lost, multiple running applications do not interfere with each other, and transactions are not influenced. However, these transactions are a source of unpredictable patterns of communication over a network. Moreover, these sets of network communication patterns are not communicated in the same fashion on VM operating systems owing to the multi-tenancy concept of cloud computing and a higher consolidation of resource sharing. Furthermore, sharing resources such as CPU, memory, and network adapters between VM tenants and applications makes it more difficult to provide steady service and predictable network performance. However, VMs are capable of executing

concurrently and with the support of underlying hardware and hypervisor.

Running web-based services or a website on these VMs is also a major challenge because they constantly read, write, and update data. These transactions are time-bounded requests and reply primitives and may not be served properly under latency-sensitive environments induced by the resource sharing in cloud computing; it is possible for individual transactions to overlook their own latency requirements [6].

Recently, cloud computing has been a primary focus for numerous computing applications. Using virtualisation, substantial growth has been achieved for many different workloads in both web-based services and cloud clients. Since the establishment of cloud computing for hosting services, researchers have been evaluating cloud performance under virtualisation. Very recently, the usefulness of general-purpose graphical programming units (GPUs) was measured; it was found that the GPU can greatly influence the performance of hosted services by means of a peripheral component interface [7].

The authors of [8] have evaluated network virtualisation overheads in the Xen environment using different workloads and under different configurations. A micro-level web server stressed the overall networking system. The stress test involved data transmission and connection establishment and closing [8]. The Xen VM was used for monitoring CPU usage of different VM overheads in the device driver domain due to I/O of VM, which was intended to quantify and measure the overhead caused due to I/O-intense jobs [8].

VM clusters based on I/O communication have improved and optimised network usage in data centres [10]. This study used a greedy algorithm to guarantee that the migration of lower-priority placement decisions was swift, thus making it suitable for large data centres. To maintain a service level agreement (SLA), an algorithm is proposed that is based on adaptive utilisation thresholds [11]. To reduce memory footprints, page-sharing models were introduced for VM co-hosting [12].

An online self-reconfiguration-based reallocating framework for VMs is proposed in [13]. The framework accurately forecasts the workloads of VM requests with Brown's quadratic exponential smoothing. Linear programming and heuristics are used for VM migration, which helps in prioritising VMs with fixed capacity [14]. In [15], an energy-aware heuristic framework is proposed for VMs to maintain SLAs and to use minimum power for maximum utilisation.

In [16], a VM resource demand predictor is proposed for allocating cloud applications. Researchers proposed a heuristic scheduling VM with adaptive resource allocation for reducing the number of physical machines.

Researchers also performed live migration of multiple VMs to reduce the traffic load on network links. Migration is carried out using distributed reduplication of VMs' memory images [17]. In [18], authors studied virtual switching overhead on a server and proposed virtual switching-aware algorithms.

In [19], a novel analytical model is proposed that is built on a queuing network to measure the performance of virtualised multi-tier applications. The effectiveness of the proposed model is assessed by a series of comprehensive trials of different configurations of multi-tier applications.

However, none of the above literature considers security features, the migration of server security along with the web server, or of application to the VM. Hence, our work is focused on these issues. A single powerful hardware system may host multiple VMs; these VMs compete for network adapters. Hence, these virtualisations environments induce overhead because of network I/O virtualisation. As with network interface cards and memory, the CPU must be shared among the hosted VMs. Therefore, the CPU also induces latency for the hosted applications.

III. Experimental Setup

Figure 1 shows the virtualisation and dedicated environment of our testbed, hereafter referred to as our testbed. We run experiments in two systems with an identical setup. We compare the performance of virtual box with that of a dedicated system. For a virtualised environment configuration, our physical machine may host one or more virtual boxes because we are interested in multi-tiered applications. Similarly, the dedicated server is also hosting multitier applications.

Default server installations on the testbed have default OS configurations, system services, and network services that are not secure. Unnecessary services and their ports are open and not used for the testbed environment. Hence, these services and ports are closed to avoid malicious intrusion. Our testbed environment needs remote access; it has secure remote access using tunnelled and encrypted protocols. We have enabled and allowed file and network service permissions and privileges on our testbed. To secure our web-based application, we have updated security patches to the latest versions.

Web-based applications are being developed for various scenarios ranging from small- to large-scale business environments. We are running time-sensitive web applications developed with the help of the Apache web-server, Java, and MySQL databases. A major threat to the availability of web applications comes from distributed denial of service attacks, which is the overloading of fake requests to a web server so as to deny a legitimate user access. Such threats work at both the lower (TCP/IP) and upper (application) layers of a network. To run a web server on the Internet, the network administrator should be well-versed in these threats. Hence, our testbed is also armed with security solutions to avoid such a threat.

To begin our experiments a dedicated testbed server was used. On this server, we have utilised a time-sensitive web application that uses a Tomcat server and the Java programming language. Our web application is a voice-over-Internet protocol (VoIP), which is a real-time media transmission protocol. This server requires end-users to register before using the web server for making audio and video calls over the Internet. We measured response times by registering more than 100 users at a time on the web server. During registration, the web server must perform a number of tasks;

first, it takes the user name and password from the user, cross-verifies to authenticate them, and replies to the user with a "200" message code. Once the user sends the registration requests to the server, the client request crosses a network connection from the client to the server. We implemented the above experimental testbed in our university lab; thus, decreasing the time required for a packet to travel from source to destination. Furthermore, we are very much concerned with the response messages originating from the server and not on the network path.

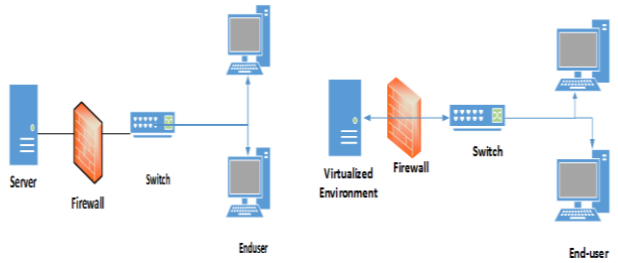

Fig. 1. Dedicated server (Left) Virtualized server environment (Right)

IV. Performance Study

We performed testbed server experiments that exercise our network and database traffic in order to estimate the CPU, network, and memory usage caused by the registration process of a web server application. All experiments were performed on an OptiPlex 3020 Micro PC. For these measurements, we used Linux 2.6.8.1, as mentioned in Table 1.

TABLE I. Configuration for the Experimental Setup

Node	Hardware configuration	
	Type	***CPU***
PC1@ university campus	OptiPlex 3020 Micro	Intel ® Pentium G3250T Processor (Dual Core, 3 MB, 2.8 GHz w/HD Graphics)
PC2@ university campus	OptiPlex 3020 Micro	Intel ® Pentium G3250T Processor (Dual Core, 3 MB, 2.8 GHz w/HD Graphics)

We began the first group of web server performance evaluations, measuring the number of requests served by the web server, its response time in milliseconds, and its throughput measured in bytes per seconds. The measurements were performed under a variable number of registration requests from clients. Web server performance under high workloads is network-bounded and under low workloads, CPU-bounded. Hence, we measured both conditions to evaluate the CPU and network interfaces.

To evaluate the CPU overhead of a dedicated server of varying web traffic, we used an Apache Tomcat HTTP server running on the testbed and a PC for sending VoIP registration requests. We used the session initiation protocol (SIP) tester to generate VoIP traffic registration requests. This tool issues a variable number of registration requests and is specifically designed for evaluating VoIP servers. We can increase the registration request rate until we receive a low reply rate from the server; that is, until the server becomes saturated. We formed a group of SIP server workloads, each generating registration requests from a variable number of clients: 10 to 100 clients per second, in steps of 10 clients.

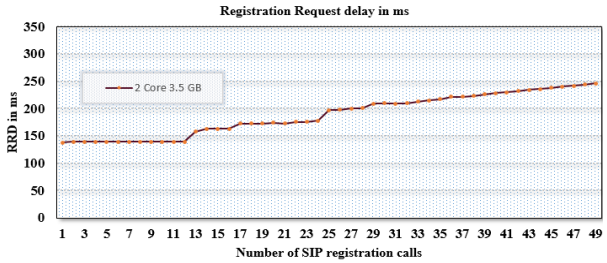

Fig. 2. Dedicated server performance

The maximum load applied to the web server is 100 registrations per second. Figure 2 shows that this is equivalent to a CPU load of 50 requests per second. Similarly, minimum throughput was achieved under a workload with 2 requests per second, a value related to the applied load. Figure 2 shows the performance of the web server on a dedicated machine. Figure 3 shows the performance of the web server on a VM.

According to the International Telecommunication Union (ITU), end-to-end, one-way delay in media transmission is 400 ms. However, there exist different delays for different codec algorithms. Media transmission protocols should abide by this law to successfully provide VoIP service. However, our testbed performance in both experiments found the end-to-end, one-way delay in media transmission to be much less than 400 ms, as shown in Figures 2 and 3.

Hence, we conclude that registration of VoIP under VMs is considered acceptable. Furthermore, we have also measured the CPU load, requests (using the GET method) made on the web server by a virtual user (VU), and bandwidth between the web server and client. We found that the TCP connections made per second are proportional to the number of VUs. We noticed that both bandwidth and CPU are directly proportional to the requests made to the web server in an attempt to obtain service from it. Figures 4 and 5 illustrate trace tests on dedicated and VM web servers, respectively. In the testbed, the CPU performance of a VM is higher than that of the dedicated machine, as expected. However, the difference is negligible and depends on bandwidth, because requests and responses are I/O-bound and hence, the CPU is involved in I/O requests. For 8000 requests, only 4% of the CPU is consumed under a VM, whereas this percentage is 3.26% in the case of a dedicated machine. The difference is further reduced with proper usage of para-virtualised devices to services in the VM.

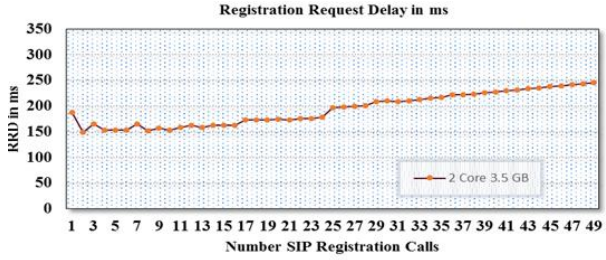

Fig. 3. Virtualized web-server performance

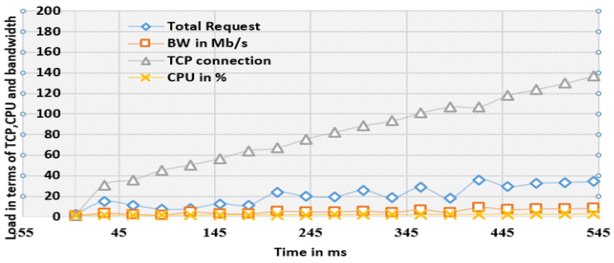

Fig. 4. Dedicated web browser performance

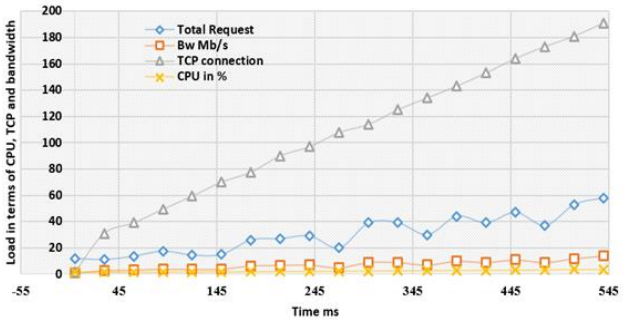

Fig. 5. Dedicated web browser performance

The performance evaluation of our testbed setup has so far been conducted in situations in which there is barely any CPU conflict from multiple VMs. However, in real deployment setups, the shared environment of many VMs in a single host efficiently utilises existing resources. Special physical machine access, enabled by the latency feature, allows the VM to achieve better results because VMs use virtual network interface cards, virtual kernels, and I/O for network-based operations. These physical elements are accessed by the VM through software. If the VM does not obtain the physical machines' power, then VM performance may degrade. In real deployment, multi-tier applications are deployed on different machines. Similarly, it is better to host multi-tier applications on different VMs rather than on a single VM because physically separated applications, such as an application server, database, or other business logic modules, are not directly reachable by hackers at a single machine. Apart from this security benefit, I/O and CPU load are also equally distributed on each of the modules of multi-tier applications. Web browsers load page elements sequentially. These elements include scripts (in HTML, PHP, or other scripting languages), style sheets, and images. However, all these elements are not accessed or downloaded to the web browser at once. Browsers open a limited number of HTTP and TCP connections based on the referenced web page on the server, because of their capacity to load only a limited amount of data per second. Furthermore, the GET and POST methods, respectively, are used to fetch and send data from the server. Therefore, these methods are expensive and are economical models for the CSP. At the same time, these methods are also critical in the performance of any website.

We have measured these methods in our testbed; our website is made from HTML, JavaScript, CSS style sheets, images, and Flash. As our testbed does not have a domain name service (DNS) server, we do not have DNS or an Internet service provider. In table 2, there are two critical methods, namely DNS and Secured Socket Layer (SSL) negotiation. These have consumed much less CPU and bandwidth in both testbed environments. Acquiring a DNS is time consuming in the first instance only. Loading HTML and the corresponding referenced pages are completely network-based operations and incur time costs of more than 2.2 ms and 2.4 ms in dedicated and VM environments, respectively. I/O operations, such as Java scripts, need the CPU and network; therefore, during execution of Java scripts, we noticed bandwidth and CPU consumption of 2.5 ms and 0.5 ms on the VM and dedicated machine, respectively. However, loading images cost more on a dedicated machine than on a VM. Congestion avoidance algorithms are used to control exponential reduction when congestion occurs because of the behaviour of the TCP protocol.

TABLE II. BROWSERS ENVIRONMENT

Server	Dedicated and Virtualisation	
	Dedicated	*Virtualisation*
DNS	0.2	0.25
Connect	0.4	0.42
SSL	0.7	0.7
HTML	2.5	2.6
JS	0.75	0.67
CSS	0.35	0.2
image	3.2	2.6
flash	-	--
font	0.2	0.6

V. CONCLUSION

The performance evaluation of web servers and browsers shows that latency-sensitive applications have successfully run without major delay. However, on some occasions, congestion avoidance has caused some issues to both environments because of the built-in features of the TCP protocol. Some latency sensitivity features provided by major VM vendors can be used to improve performance of hosted services on the Internet cloud.

REFERENCES

[1] M. Armbrust et al., "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2] T. Ma, Y. Chu, L. Zhao, and O. Ankhbayar, "Resource allocation and scheduling in cloud computing: Policy and algorithm," *IETE Techn. Rev.*, vol. 2;31, no.1, pp. 4–16, Jan. 2014.

[3] W. D. Mulia, N. Sehgal, S. Sohoni, J. M. Acken, C.L. Stanberry, and D. J. Fritz, "Cloud workload characterization," *IETE Techn. Rev.*, vol. 1;30, no. 5, pp. 382–397, Sep. 2013.

[4] M. Kolhar, M. Abu-Alhaj, S. M. Abd El-atty, "Cloud Data Auditing Techniques with a Focus on Privacy and Security," IEEE Security & Privacy, vol. 15, no. 1, pp. 42-51, Jan.-Feb. 2017.

[5] M. Kolhar, S. Abd El-atty, Mohammed Rahmath, "Storage allocation scheme for virtual instances of cloud computing". Neural Computing and Applications, pp.1-8, Dec-Jan 2016. 1-8.

[6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[7] A. J. Younge, et al., "Evaluating GPU passthrough in Xen for high performance cloud computing," in Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014.

[8] A. Menon et al., "Diagnosing performance overheads in the Xen virtual machine environment," in *Proc. 1st ACM/USENIX Int. Conf. Virtual Execution Environments*, 2005.

[9] L. Cherkasova and R. Gardner. "Measuring CPU overhead for I/O processing in the Xen virtual machine monitor," in *USENIX Annual Tech Conf.*, vol. 50, 2005.

[10] D. Kakadia, N. Kopri, and V. Varma, "Network-aware virtual machine consolidation for large data centers," in *Proc. 3rd Int. Workshop on Network-Aware Data Management*, 2013, pp. 6.

[11] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proc. 8th Int. Workshop on Middleware for Grids, Clouds and e-Science*, vol. 4, 2010.

[12] M. Sindelar, R. K. Sitaraman, and P. Shenoy, "Sharing-aware algorithms for virtual machine colocation," in *Proc. 23rd Annu. ACM Symp. Parallelism in Algorithms and Architectures*, pp. 367–378, 2011.

[13] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, and L. Yuan, "Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers," in *IEEE Int. Conf. Services Computing*, pp. 514–521, 2010.

[14] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros, and C. A. F. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1027–1034, 2011.

[15] Z. Cao and S. Dong, "An energy-aware heuristic framework for virtual machine consolidation in cloud computing," *J. Supercomputing*, vol. 69, no. 1, pp. 429-451, 2010.

[16] Q. Huang, S. Su, S. Xu, J. Li, P. Xu, and K. Shuang, "Migration-based elastic consolidation scheduling in cloud data center," in *33rd IEEE Int. Conf. Distributed Computing Systems Workshops*, 2013, pp. 93–97.

[17] U. Deshpande, U. Kulkarni, and K. Gopalan, "Inter-rack live migration of multiple virtual machines," in *Proc. 6th Int. Workshop Virtualization Technologies in Distributed Computing Date*, pp. 19–26, 2012.

[18] M. Li, J. Bi, and Z. Li, "Improving consolidation of virtual machine based on virtual switching overhead estimation," *J. Network and Computer Appl.*, 2015.

[19] K. R. Zadeh, A.L. Morteza, P. Kabiri, and B. Javadi, "Performance modeling and analysis of virtualized multi-tier applications under dynamic workloads," *J. Network and Computer Appl.*, vol. 56, pp. 166–187, 2015.