Performance and Scalability Evaluation of Oracle VM Server Software
Virtualization in a 64 bit Linux Environment

Ibidokun Emmanuel Tope, Pavol Zavarsky, Ron Ruhl, Dale Lindskog
Information Systems Security Department
Concordia University College of Alberta
Edmonton, Canada
eibidoku@csa.concordia.ab.ca, {pavol.zavarsky, ron.ruhl, dale.lindskog}@concordia.ab.ca

Abstract — The growing adoption of virtualization in the enterprise environment has resulted in a couple of huge benefits, however, this has not been without its attendant problems and anomalies, such as performance tuning and erratic performance metrics, unresponsive virtualized systems, crashed virtualized servers, misconfigured virtual hosting platforms, amongst others. The focus of this research was the analysis of the performance of the Oracle VM server virtualization platform against that of the bare-metal server environment. The scalability and its support for high volume transactions were also analyzed using 30 and 50 active users for the performance evaluation. Swingbench and LMbench, two open suite benchmark tools were utilized in measuring performance. Scalability was also measured using Swingbench. Evidential results gathered from Swingbench revealed 4% and 8% overhead for 30 and 50 active users respectively in the performance evaluation of Oracle database in a single Oracle VM. Correspondingly, performance metrics of 75% and 87% were obtained with 30 and 50 active users in a dual Oracle VM server, indicating performance scalability improvement with two virtual machines. Our results also revealed significant percentages in latency and bandwidth achievement by Oracle VM server which cannot be overlooked, despite some variances in results obtained from LMbench measurement.

Keywords— Performance and scalability; Virtualization; Oracle VM server; Oracle Database; RedHat; Bare-metal server; Availability

I. INTRODUCTION

Presently, many vendors like Intel, AMD, Sun Microsystems, and IBM are developing virtualization products which target markets with revenues in the billions and growing [3].

Server virtualization efficiently permits software to be separated from the physical hardware. With a secured server virtualized system, there is improvement on the availability of application and the isolation of processes. Server virtualization also has security setbacks, such as exploitable weaknesses in virtualization software, anomaly performance metrics with some benchmark devices, misconfigured virtual hosting platforms and guest operating systems [4]. However, apart from these diverse security threats, once the security controls for the virtual environment are slipshod, there is a greater possibility for infection. Server virtualization is a tool used to reduce the operational time and cost in today’s computing but if done wrong it becomes a security threat to the environment. This saving of time and resources allow businesses to devote more time to new opportunities rather than focusing on maintenance and management [5].

A salient feature of virtualization is scalability, which allows for growth in capacity, performance and availability of the virtualized system.

There are many benchmarking tools designed to measure different performance characteristics. Examples include Swingbench for evaluating transactions per minute [6], LMbench for measuring network bandwidth and latency [7], Httperf for measuring web server performance [8]. This research evaluates the performance and scalability of the Oracle database both in a virtualized and non virtualized environment. We use Swingbench and LMbench as benchmarking tools. Swingbench benchmarking tool has found prominent in performance evaluation. Swingbench is specifically designed to generate load and stress test an Oracle Database. Whereas, LMbench gives an indication of the impact of virtualization on the hardware resources by measuring latency and bandwidth.

While there have been previous studies evaluating various aspects of server virtualization on 32-bit architecture system [2, 9, 10, 11, 12]. The Tolly group conducted a study for Oracle by evaluating the performance of Oracle VM running Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 in a 32-bit Oracle Linux environment [2]. This work aims at updating the performance metrics and extending the experiments with different hardware, software and configuration using 64-bit architecture system [1].

II. RELATED WORK

Significant work has been conducted on performance evaluation of various virtualization environments. One such work was the white paper done by the Tolly group which evaluated the performance of Oracle VM running Oracle 11g Database application in 32-bit environment [2]. Xiang Zhang et al. [11] described solo as a lightweight virtual machine that allows VM to run directly on hardware with the highest precedence and critically alter the overhead done by virtualization. It was further stressed that solo enhances VM performance to the level of traditional OS, and thus meets the requirements of the high performance applications.
without a unique hardware support. Barham et al. [9] gave a comprehensive introduction to the Xen hypervisor and made a thorough performance evaluation of Xen versus VMware. Deshane [12] presented an independent research describing the performance comparison between Xen and KVM, which evaluated the overall performance, security impacts, performance isolation and scalability of Xen and KVM.

Regarding the benchmarks and tools for performance evaluation of virtualized systems, Jianhua Che et al. [10] asserted that virtual machine monitors (VMM) is the main component of virtual machine (VM) system, and its efficiency highly impacts the performance of the entire system. They also measured and analyzed the performance of two open suite virtual machine monitors—Xen and KVM using LINPACK, LMbench and IOzone, and provided a quantitative and qualitative security comparison of both virtual machine monitors. Like any new technology, poor implementation and process may create some vulnerability. In the case of server virtualization, the risk would be loss of security between VMs and overall performance metrics.

This research intends to fill the void in equivalent studies for 64-bit architecture system in comparison to the previous version used by the Tolly group [2].

III. THE SOFTWARE AND HARDWARE ENVIRONMENT

The software we used in this research are Oracle VM server 2.2.1, the Red Hat Enterprise Linux (RHEL) 5.5, Oracle Database 11g Enterprise Edition Release 2 application, the benchmark tool Swingbench version 2.3 and LMbench version 3-a9. The hardware: was the machine type-64-bit physical server, 4CPU cores, 8GB RAM, one 64-bit virtual machine with 1CPU core 3.75GB RAM size and two 64-bit virtual machines with 2CPU (1 for each VM), 7.5GB (3.75GB for each VM).

IV. EXPERIMENTAL METHODOLOGY

All experimental evaluations are performed on the AMD PhenonII, with Quad-core processors 64-bit at 3.0GHz, 8GB. The same physical computer is used in both experimental environments to ensure that no machine condition will affect experimental results. In bare-metal environment, Redhat5.5 was the operating system, while Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 was the application used. While in virtualized environment oracle VM was the hypervisor on which two virtual machines were created. On each virtual machines the same software and application was installed in both VMs and bare-metal environment. Swingbench and LMbench were successfully installed on the bare-metal server and on the two virtual machines.

A. Test Description

In the experimental set up, we measured the maximum transactions per minute (tpm) supported by a bare-metal server, versus the transactions served by a single virtual machine and dual virtual machine supporting Oracle VM using Swingbench version 2.3.0.422. The results from the bare-metal server test were compared to the results obtained after running the similar test within a virtual environment for 30 and 50 active users respectively.

We also used LMbench version 3 in the test to measure latency and bandwidths of both bare-metal server and Oracle VM environment so as to compare the results of their process and memory performance test. We ran the test for latency and bandwidth because performance issues are usually caused by latency problems, bandwidth problems, or some combination of the two. Six iterations of test were run and standard deviation of the results was used for analysis in both tests.

V. EXPERIMENT RESULTS AND ANALYSIS

A. Performance Efficiency and Scalability Using Swingbench

In this test, we used swingbench to measure the transaction per minute, with the work load set to 30 and 50 active users. Table I and table II show the database maximum transaction rate results obtained, while Fig. 1 shows for performance comparison graph using the standard deviation value of the results obtained for bare-metal server and the single Oracle VM server.

From Fig. 1, single Oracle VM server has a throughput of 48% and 46% database transaction compared to throughput of 52% and 54% achieved by bare-metal server with 30 and 50 active users correspondingly. The overall analysis showed 4% and 8% overhead for 30 and 50 active users respectively.

| TABLE I. DATABASE MAXIMUM TRANSACTION RATE: 30 AND 50 ACTIVE USERS BARE-METAL SERVER |
|---------------------------------|----------|----------|
| 30 users                        | 50 users |          |
| 6089                            | 9048     |          |
| 6006                            | 8964     |          |
| 5745                            | 8494     |          |
| 6020                            | 8900     |          |
| 5964                            | 9029     |          |
| 5703                            | 8919     |          |
| Standard deviation: 158.49/minute| 203.71/minute |

| TABLE II. DATABASE MAXIMUM TRANSACTION RATE: 30 AND 50 ACTIVE USERS A SINGLE ORACLE VM SERVER |
|---------------------------------|----------|----------|
| 30 users                        | 50 users |          |
| 5644                            | 8484     |          |
| 5660                            | 8518     |          |
| 5570                            | 8474     |          |
| 5343                            | 8105     |          |
| 5384                            | 8449     |          |
| 5369                            | 8597     |          |
| Standard deviation: 145.84/minute| 170.94/minute |
Furthermore, an additional test was run to evaluate the performance of dual virtual Oracle VM server (1 of 2VMs and 2 of 2VM) on the same hardware. Table III and table IV showed test results, while Fig. 2 shows the graph comparison of dual Oracle VM server with the bare-metal server.

In Fig. 2, the two occurrences of Oracle VM server achieved throughput of 76% and 87% with 30 and 50 active users respectively compared to that of the bare-metal server. These two performance test was a confirmation that Oracle VM software can execute comparable performance to a bare-metal server without the cost of installing extra hardware.

We observed the performance scalability feature of Oracle VM server and its efficiency. This was done by scaling the Oracle VM server on the same hardware from one virtual machine (OVM1) to two virtual machines (OVM2). The test was carried out on the first two machines with a load of 30 and 50 active users and measurements were taken in turn. Table V and table VI showed the summary of the results obtained while Fig. 3 shows the comparison of a single Oracle VM server to the two virtual machines.

Furthermore, an additional test was run to evaluate the performance of dual virtual Oracle VM server (1 of 2VMs and 2 of 2VM) on the same hardware. Table III and table IV showed test results, while Fig. 2 shows the graph comparison of dual Oracle VM server with the bare-metal server.
respectively. Whereas, 44.89% and 44.63% maximum transaction load was processed by the first and the second of the two virtual machine with 50 active users correspondingly. Hence the analysis of these two tests showed that from one virtual machine to the second virtual machine, there was a slight flexibility of less than 10% overhead in the performance produced in 30 and 50 active users.

B. Performance Efficiency Using LMbench

The performance tests were run with the use of LMbench to verify that latencies and bandwidth of bare-metal server and Oracle VM environment were comparable. Fig. 4 and Fig. 5 below show the result graphs obtained while executing the simple function latency and context switching for the comparison of a bare-metal server vs. Oracle VM server.

Fig. 4 shows the measurement of the context switching latency. This is the time needed to save the state of one process and restore the state of other. The performance of Oracle VM server was 18% more than the bare-metal server in the 64 process, while in 96 process, the Oracle VM server was 44% more. Overall Oracle VM server shows a rapid context switch.

We also obtained bandwidth measurement for communication bandwidth facilities: Socket bandwidth using localhost at different speed, AF_UNIX sock stream and pipe bandwidth, read bandwidth, memory read and memory write, all bandwidth data collected are in megabytes per seconds. Hence the bare-metal server’s results were compared to that of Oracle virtual environment. See Fig. 6, 7, 8, 9, and 10 for the graph comparison results of bandwidth measurement of a bare-metal server vs. Oracle VM server.

The Fig. 4 depicts the simple function latency measurement, the two functions were observed: process fork+execve and process fork+/bin/sh-c. Process fork +execve is the way of creating a process and executing the process whereas process fork+/bin/sh-c is the standard command to start a process already executed and a complicated new process. Considering process fork+execve, the results showed that OracleVM server was 6% better in performance compared to the bare-metal server. While in fork+/bin/sh-c the Oracle VM server’s performance was less than 50% compared to that of the bare-metal server. These are the anomaly results that occurred due to the time drift while using benchmarking tools [13].
The Fig. 6 shows the socket bandwidth using localhost with different speed. From the results the performance of bare-metal server with 0.000512MB/s was more by 6% than that of Oracle VM server by 6%. The performance of bare-metal server with 0.001024MB/s was more by 62%. The performance of bare-metal server with 0.001437MB/s was more by 64%. The performance of bare-metal server with 10MB/s was more by 34%.

Fig. 7 shows the interprocess communication bandwidth. Two features were observed: AF_UNIX socket stream bandwidth and Pipe bandwidth. AF_UNIX socket stream bandwidth is the TCP connection socket while Pipe bandwidth is measured by creating a writer and a reader. The results showed that Oracle VM server’s performance was 92% more than the bare-metal server’s for AF_UNIX socket stream bandwidth. On the contrary, the bare-metal server’s performance was 86% more than the Oracle VM server’s for the Pipe bandwidth. These results were unpredictable due to the fact that there was some flawed in the use of benchmarking tool.

From Fig. 8, the Oracle VM server has a significant performance of 62% more in reading a file compared to the bare-metal server. This also showed that the higher the block size, the better the reading of the files in Oracle VM server.

Fig. 9 shows the memory read bandwidth. This is the ability to allocate the specified amount of memory, zero it and then multiply it with the reading of the memory. The results showed that the Oracle VM server performed 12% more than the bare-metal server. Oracle VM showed a better performance except at the block size of 4.19Mbytes which could be as a result of the time drift.

Fig. 10 shows the memory write bandwidth. This is the ability to allocate the specified amount of memory, zero it and then multiply with the writing of that memory. The results showed that the bare-metal server performed more than the Oracle VM server by 14%.

From all the LMbench analysis above, the results showed that the performance of Oracle VM server could not be discarded. Because we discovered on the overall that LMbench measurement has given a great insight which will be useful to the small business enterprise. On the other hand, one cannot discount the merits of published articles and the suggestions claiming that LMbench may be flawed [10]. Variances in LMbench results can be critical in decision by small business enterprise to determine the effectiveness of Oracle VM server compared to bare-metal server.

VI. CONCLUSION

Experimental results and observations from Swingbench lead us to the conclusion that Oracle VM server can efficiently support a variety of workloads in a virtualized environment. This is justified by the 4% and 8% overhead obtained using 30 and 50 active users respectively. Within dual Oracle VM server, the performance metrics of 75% and 87% with 30 and 50 active users respectively, indicated a performance scalability improvement with two virtual machines compared to a single virtual machine. This indicates that Oracle VM server effectively utilized its hardware resources, and there was improvement in the performance metrics obtained in dual VM as a result of the availability of a running Oracle database application. This research also revealed that having sufficient Swingbench data, the core virtualization technology has a marked positive effect on availability, however, the effect on confidentiality and integrity is less positive.

LMbench results further proved that Oracle VM server achieved significant percentages in latency and bandwidth which cannot be ignored, despite some erratic results obtained using LMbench benchmark tool. These results provide a premonition knowledge that will help the small scale enterprise to evaluate the Oracle VM server’s performance compared to the bare-metal server.

Further researches can be built on this research by scaling the number of virtual machines and testing for scalability. An alternative extension of this research is a performance analysis of the Oracle database in virtual environments.
ACKNOWLEDGMENT

The research was supported by the Concordia Faculty of Professional Education and Faculty of Graduate Studies. Also, thanks go to Dominic Giles at Oracle UK (Swingbench) and Dr. Carl Staelin at HP Labs Israel (LMbench) for the tools they developed which made this research doable.

REFERENCES


