Virtual Numbers for Virtual Machines?

Alan Y. S. Tan, Ryan K. L. Ko
Cyber Security Lab, Dept. of Computer Science
University of Waikato
Hamilton, New Zealand
Email: {yst1, ryan}@waikato.ac.nz

Veena Mendiratta
Bell Laboratories
Alcatel-Lucent
Naperville, Illinois, USA
Email: veena.mendiratta@alcatel-lucent.com

***Abstract*—Knowing the number of virtual machines (VMs) that a cloud physical hardware can (further) support is critical as it has implications on provisioning and hardware procurement. However, current methods for estimating the maximum number of VMs possible on a given hardware is usually the ratio of the specifications of a VM to the underlying cloud hardware's specifications. Such naive and linear estimation methods mostly yield impractical limits as to how many VMs the hardware can *actually* support. It was found that if we base on the naive division method, user experience on VMs at those limits would be severely degraded. In this paper, we demonstrate through experimental results, the significant gap between the limits derived using the estimation method mentioned above and the actual situation. We believe for a more practicable estimation of the limits of the underlying infrastructure, dominant workload of VMs should also be factored in.**

***Keywords*-cloud computing; virtualization; cloud resource provisioning; load limit prediction**

I. Introduction

Questions like *"How many virtual machines can we support with this physical server?"* and *"How much hardware do we need to buy if we need to support 1000 instances in our cloud?"* are commonly asked questions during the planning and operations phases of cloud computing [1] environments.

The common approach used to answer such questions would be to find the ratio between the hardware specifications of the physical hardware and the requirement of an individual virtual machine (VMs), i.e. taking the total RAM of the physical server and divide it with the RAM requirement of a single instance of a virtual machine that will be hosted on the physical hardware [2]. However, such divisional method of calculating the number of VMs that a physical server can support does not factor in usability (e.g. slow and delayed response from VM terminals, jobs crashing or machine freezes.) and workload of the VMs.

In this paper, we show that there is a need for more realistic methodology for estimating the maximum number of VMs a physical server can support (VM_{max}). We demonstrate how using naive division methods to calculate VM_{max} is impractical by attempting to scale to the calculated VM_{max} with VMs running actual workloads and measure their usability and performance. We then conclude with some insights gained from analysing the results of our experiments.

II. Related Works

Several studies, such as [3–5], have looked into the issue of resource usage by VMs under various workloads and operating environments. These studies mostly focused on understanding the performance of VMs under different conditions. The findings derived from these studies are usually used in determining the optimal placement of VMs in a virtualised environment such as a private or public cloud infrastructure [6].

In this paper, the focus is not on finding the optimal placement or distribution for VMs in a virtual environment. Rather our research attempts to find a method to estimate how many VMs can a given physical server support in a virtualised environment, such that the performance and usability of the provisioned VMs are not affected, while factoring in the workload that will be running on those VMs.

III. Experiments

To show using naive division methods to calculate VM_{max} is impractical, we derive the theoretical limits using divisional methods and scale the number of VMs running on a physical server towards the limit. We then measure the performance and test for the usability (in terms of user experience) of the VMs while we scale the number of VMs running on the physical server.

We used a Dell C6220 PowerEdge server with 4 Quad Core Intel Xeon E5-2670 @ 2.6GHz and 264GB of RAM as the host physical server. At the current stage, we focus our experiments on two commonly seen cloud workloads: CPU and disk I/O workloads. Hence, we only considered the CPU and RAM resources of the server. We used OpenStack (*Grizzly* release [7]) as our cloud management framework and left the resource over provisioning ratio at its default (**CPU ratio**: 16, **RAM ratio**: 1.5). For the VMs, we used VMs with a setting of 1 logical core, 512MB RAM and ran CentOS 6.4 as the operating system.

For our experiment, we measure the performance of VMs and the physical server in terms of CPU and RAM. VMs running CPU or disk I/O-dominant workloads are spawned

Table I
Comparison of maximum VM limit derived using divisional methods and actual experiments

	Over Provisioning Ratio	Physical resource	VM specs	Theoretical VM_{max} limit	Experimental VM_{max} limit
CPU	16	256 logical cores	1 logical core	256 VMs	112 VMs
RAM	1.5	396GB	512MB	792 VMs	24 VMs

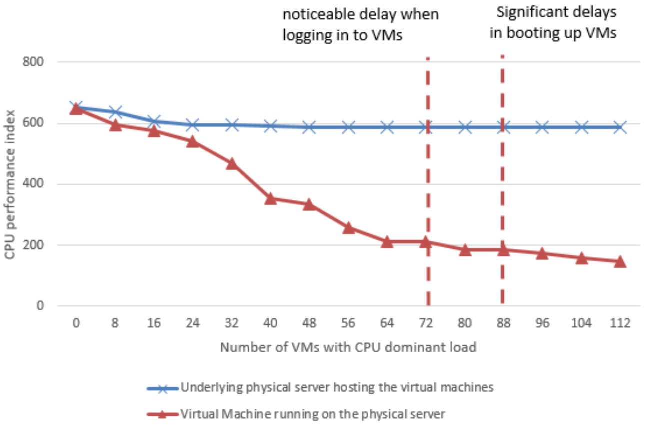

Figure 1. Results of running CPU measurement on both virtual environment and the hosting physical server

increasingly and we ran benchmark tools to derive the performance measurements of the respective resources.

To simulate CPU-dominant workloads on the VMs, we used *stress* [8], v1.04, a tool for simulating CPU workloads on Linux machines. We set *stress* to automatically compute the square of a random number in an infinite loop upon boot. For simulating I/O-dominant workloads, we performed file read/write operations using the *dd* command on the Linux operating system. Table II shows the setting used for simulating I/O-dominant workloads.

To measure the performance of VMs, we used the tool *Unixbench* [9], v4.1.0 and the *dd* command for measuring the performance of CPU and I/O in the VMs respectively.

Unixbench does the measurement by computing a series of integer and floating point math operations such as array computations and monitors the time taken to complete these operations. The results are normalised to a pre-defined baseline result set, averaged and exponentiated. Details of the tool can be found at [9]. The final results are represented as an *index* value. Likewise for measuring I/O performance, we used the *dd* command to execute read/write operations through the RAM and measure the time taken to complete the operations. The setting used for the I/O measurements are shown in Table II.

The results for the CPU and I/O experiments are is shown in Figures 1 and 2 respectively.

IV. Discussions

With reference to Figures 1 and 2, the usability of VMs started to deteriorate as the number of VMs with workload running increases. VMs started to show delayed response (user typing at the command line interface) when 72 VMs with CPU-dominant workload was running concurrently on the physical server. Delays of up to 2-3 seconds can be observed when the number of VMs with CPU-dominant scales up to 96 VMs.

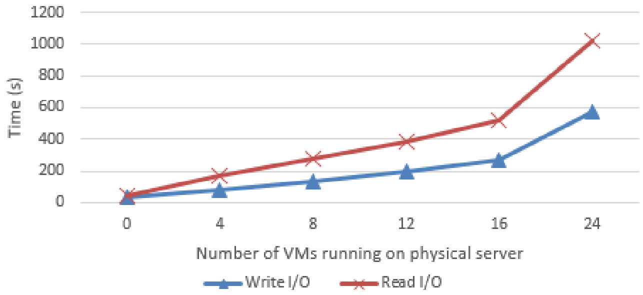

Figure 2. Time taken for I/O test to complete on virtual environment as workload scales

In the I/O-dominant tests, the I/O test took over 20 minutes to complete when the number of I/O-dominant VMs reaches 24. As we considered waiting for 20 minutes for reading and writing of a file of 4GB size, unacceptable in terms of user experience, we terminated our I/O experiment after 24 VMs.

Table I shows the comparison between the theoretically derived VM_{max} and the actual VM_{max} derived through experiments. The theoretical limits were derived by taking the ratio between total values for the respective resource (inclusive of the over provisioning ratio) on the physical server and the corresponding resource requirement of a individual VM (e.g.total RAM available on the server * over provisioning ratio / VM requirement).

From Table I, we can observe that there is a significant difference between the theoretically derived VM_{max} and the VM_{max} obtained from our experiments. This gap between the actual and derived VM_{max} demonstrated the need for a more accurate method for estimating the number of VMs a physical server can support.

Current divisional methods can only provide a disillusion of how much a physical server can support (e.g. a system administrator estimated a new server, based on its specification, can support 100 additional VMs using divisional method. However, he later found out that, due to the heavy workload in the system, the server could only support up to 10 additional VMs in the production environment.).

We argue that a method that factors in workload when estimating VM capacity of a physical server, will not only aid resource provisioning and planning, but also in server maintenance. System administrators will be able to estimate how many VMs can be supported given a physical server's specification and a sample of the current workload. Such knowledge will help system administrators determine, not only what servers to purchase but even the amount of servers required to meet their requirements. Such a method differs from those used in determining VM placement [10], where

Table II
ARGUMENTS USED IN CONJUNCTION WITH THE *dd* COMMAND

	Input file	Output file	Block Size	number of blocks	additional arguments
Write	/dev/zero	writeIO.test	1MB	4000 (4GB)	conv=fdatasync,notrunc oflag=direct
Read	writeIO.test (4GB)	/dev/null	1MB	-	-

the concern is how VMs can be allocated to physical servers without affecting performance.

V. ON-GOING WORK

Our experiments in Section III opened paths for research towards a methodology for predicting virtual resource count on physical machines. We hypothesize that there is a formula which enables administrators to accurately calculate VM_{max} based on workload related parameters and the specifications of a physical server.

The formula must be able to (1) give a realistic estimate of the number of VMs which a hardware configuration can support even before the VMs are spawned, and (2) estimate the number of VMs an administrator can further trigger on top of the current load. With this hypothesis in mind, we propose the notion of a "usability utility meter" to help administrators gauge the maximum number of VMs the hardware can support, before performance and user experience of VMs are affected.

Our next step is to investigate assignment of weightages to different resource types used in virtual resource provisioning for different types of expected workloads (e.g. if VMs are expected to run CPU intensive workloads, how much weightage should be assigned to CPU factor in the formula such that an accurate estimate can be given.)

As part of formulating the formula for estimating VM_{max}, we plan to also look into what other components in a computer system, can affect the performance and usability of a VMs, for different workload types. For example, studies [6, 11] have shown how network factors can affect the performance of VMs. Having said that, we see the need to also consider more fine-grain resources such as the hard disk speed and bus size, within a physical server.

VI. FUTURE RESEARCH AREAS

Even though the benchmarking indices we used, normalised and simplified the understanding of the experimental results, we believe that a new research area in cloud benchmarking with respect to limit calculations has been opened. Such benchmarks should establish a fair and representative index which are highly sensitive - revealing trends even towards the upper limits (e.g. when user experience degrades drastically).

VII. CONCLUDING REMARKS

Through the results of our experiments, it can be observed that using naive divisional methods to estimate how many VMs can a physical server support is impractical. Scaling the amount of VMs with actual workload running towards that limit will usually result in poor performance and user experience on those VMs. There needs to be methodologies that factor in workload and other related factors in order to produce a more accurate estimation. Such methodologies will benefit cloud administrators or private cloud owners when designing and managing their cloud infrastructures.

The gap between the maximum number of VMs supported between the CPU and the I/O tests showed that not all resources should be evaluated equally when factoring in workload running on VMs. As such, some form of weightage should be introduced for different resources when estimating the maximum number of VMs that can be supported.

REFERENCES

[1] R. K. L. Ko, "Cloud Computing in Plain English," *ACM Crossroads - Plugging into the Cloud*, vol. 16, pp. 5–6, 2010.

[2] TechNet Archive, "Virtual Server Performance Tips," Available: http://blogs.technet.com/b/megand/archive/2005/06/09/406145.aspx (Accessed: 27/11/2013).

[3] S. Smith, "Measuring Performance of Applications on Virtualized Systems Under Test (SUTs)," Available:http://software.intel.com/en-us/articles/measuring-performance-of-applications-on-virtualized-systems-under-test-suts (Accessed: 27/11/2013).

[4] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, "Modeling Virtual Machine Performance: Challenges and Approaches," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, pp. 55–60, 2009.

[5] J. Che, Q. He, Q. Gao, and D. Huang, "Performance Measuring and Comparing of Virtual Machine Monitors," in *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, vol. 2, 2008, pp. 381–386.

[6] A. Gupta, D. Milojicic, and L. V. Kale, "Optimizing VM Placement for HPC in the Cloud," in *Proceedings of the 2012 workshop on Cloud services, federation and the 8th Open Cirrus Summit*, 2012, pp. 1–6.

[7] "OpenStack Grizzly," Available: http://www.openstack.org/software/grizzly/ (Accessed: 29/11/2013).

[8] A. Waterland, "Stress," Available: http://people.seas.harvard.edu/~apw/stress/(Accessed: 11/11/2013).

[9] "Unixbench," Available: https://code.google.com/p/byte-unixbench/ (Accessed: 11/11/2013).

[10] C. Isci, J. E. Hanson, I. Whalley, M. Steinder, and J. O. Kephart, "Runtime Demand Estimation for Effective Dynamic Resource Management," in *Proceedings of 12th IEEE/IFIP Network Operations and Management Symposium (NOMS'10)*, 2010, pp. 381–388.

[11] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proceedings of the 29th conference on Information Communications (INFOCOM'10)*, San Diego, CA, USA, 2010, pp. 1154–1162.