

Network Bandwidth Configuration Tool for Xen Virtual Machines

Kuriakose Mathew, Purushottam Kulkarni and Varsha Apte
Indian Institute of Technology Bombay
Powai, Mumbai, India
Email: {kmathew, puru, varsha}@cse.iitb.ac.in

Abstract—Performance differentiation has always been a requirement in virtualized environments, especially in case where virtualization is used in high-end data center applications. In this paper, we present a command-line tool for Network bandwidth differentiation in Xen, which is an open source solution for virtualization. Though the CPU credit scheduler in Xen can be configured using command-line tools to specify weights and caps for a virtual machine, no such tools exist to specify the network bandwidth limits. Through our tool, *xmsetbw*, network bandwidth limits for a virtual machine in Xen can be specified and dynamically reconfigured (without kernel recompilation). Experimental evaluations show that bandwidth utilization is limited within specified values.

I. INTRODUCTION

Virtualization, the technology by which a host computer's physical resources can be shared by multiple “virtual machines”, is being adopted rapidly by data center operators, as a means to achieve server consolidation and increased resource utilization. Such consolidation can be successful only if the different types of physical resources of a computer (e.g. CPU, disk, memory, network bandwidth, etc.) can be allocated to virtual machines (VM) in a *quantitative* and *guaranteed* manner.

The Xen Hypervisor [1], [4] is an open-source virtualization software which can guarantee a virtual machine a fixed share of the CPU, memory, and disk space of the host computer. However, currently there is no method to specify a fixed share of the *network bandwidth* of the computer.

It is clear though, that such a capability is required. E.g., a VM with a streaming video server deployed on it would have significantly different network I/O requirement than that of a VM running a chat server. However, Xen only offers a fair scheduler which aims to divide the available I/O bandwidth *equally* among all the VMs. Nonetheless, studies have shown that the actual amount of I/O bandwidth utilized by any particular VM is an unpredictable quantity [6].

While various improvements for better I/O performance have been proposed [5], [6], to the best of our knowledge, a tool to specify *different* bandwidth requirements for Xen VMs, and a scheduler to guarantee these, does not exist. Therefore, in this work, we develop a command-line tool for dynamic configuration of network I/O bandwidth allocated to a VM, and enhance the existing scheduler to offer *weighted* fair scheduling of I/O requests across VMs.

In the next section, we briefly describe the details of the Xen networking architecture which are relevant to our tool. We then discuss the implementation details of our tool and conclude with validation experiments.

II. XEN BACKGROUND

Xen consists of three core components—the virtual machine monitor (also known as the hypervisor), virtual machines (also known as domains) and applications executing within the virtual machines. The hypervisor enables execution of domains on virtualized environments. There are two types of domains in Xen—privileged (Dom-0) and unprivileged (Dom-U). Dom-0 is booted along with the Xen hypervisor.

Xen implements a split device driver model—a front-end driver in the guest domain and a back-end driver in Dom-0, to provide an interface to the underlying hardware. The I/O ring, a shared memory data structure, is used for communication between the front-end and back-end drivers. Communication between domains is facilitated by an out-of-band exchange of information (reference of shared memory page, advertisement on the event channel etc.). Xen provides a mechanism called *XenStore* for this purpose and an interface called *XenBus* to interact with *XenStore*.

XenStore is a storage abstraction provided by Xen for inter-domain interaction [2]. It has a namespace similar to the directory tree of the Unix file system. Data in *XenStore* is stored in the form of a key-value pair. *XenStore* runs as a daemon (*xenstored*) in Dom-0 and supports transactions and atomic operations. User-space tools can store and retrieve data from the *XenStore*.

XenBus [3] provides an API to read, write and monitor key-value pairs in the *XenStore*. A *XenBus watch* is a monitoring mechanism to signal modifications to key-value pairs. A *watch*, set on a key, triggers a call back function when the value corresponding to the key is updated. As part of our tool, a kernel process sets a *watch* on a key, and the key is updated by a user process. This feature can be used for user-kernel interaction and forms a core part of our tool.

III. *xmsetbw*: THE NETWORK BANDWIDTH ALLOCATION TOOL

A system administrator can specify the network I/O bandwidth limit for a domain using the tool as follows,

```
xmsetbw -d <domain_name> -l <limit>
```

Specified Limit(MBps)	1	2	3	4
Bandwidth(MBps)	0.96	1.92	2.91	3.87

TABLE I
COMPARISON OF SPECIFIED AND MEASURED BANDWIDTH UTILIZATION.

domain_name is the name of the domain to be configured and *limit* is the maximum network I/O bandwidth to be allocated. The tool itself is a user-level process in Dom-0.

Xen's backend driver (termed as the *netback* driver) uses a credit scheduler to schedule the network I/O traffic for each Dom-U. The default credit distribution for each domain is equal, resulting in fair share of network I/O bandwidth for each. *xmsetbw* is a user-level tool, to change the credit allocation associated with each domain. The command line argument *limit* specifies the credits to be allocated to a domain per 100 ms. The credits are set to be refreshed every 100 ms for each domain. For e.g., if two domains are supposed to get a maximum bandwidth limit of 1MBps and 2MBps respectively, the command usage would be,

```
xmsetbw -d domain_1 -l 100000
xmsetbw -d domain_2 -l 200000
```

xmsetbw allows system administrators to set absolute value limits on network I/O usage.

xmsetbw uses the *XenBus* API along with *XenStore* to achieve dynamic network I/O bandwidth allocation. As part of *XenStore* initialization, a call back function registers a watch for a *XenStore* key defined to store the network I/O bandwidth setting for a domain. On execution of the *xmsetbw* command, the value corresponding to the key in *XenStore* is changed. This change event triggers a call back function which reads the changed value and updates the state corresponding to the domain under consideration. The *netback* driver, when transmitting a packet of the corresponding domain, updates the new credit values for the domain (which maps to its bandwidth limit) before transmission. All transmissions are limited by the updated credit value till further changes. The above approach used by *xmsetbw* provides an easy, user-level implementation to change network bandwidth at runtime, thus avoiding the need for manual changes in kernel code and kernel recompilation.

IV. EXPERIMENTAL EVALUATION

The experimental evaluation to determine the correctness of *xmsetbw* used the following setup, a 2.832 GHz Intel Core 2 Quad Processor with 3 GB RAM and a 10 MBps Ethernet card (at saturation, it was found that the card delivered close to 11 MBps). *SCP* transfer of a large file, from Dom-U to a machine outside the virtual environment was used as the load. Xen version 3.1.3 along with Ubuntu release 6.06 was used for the virtualization setup.

Table I reports the comparison between specified and measured network I/O bandwidth utilization values. As seen from the table, for all specified limits 1–4 MBps, the measured network I/O rate is similar. The result shows that *xmsetbw*

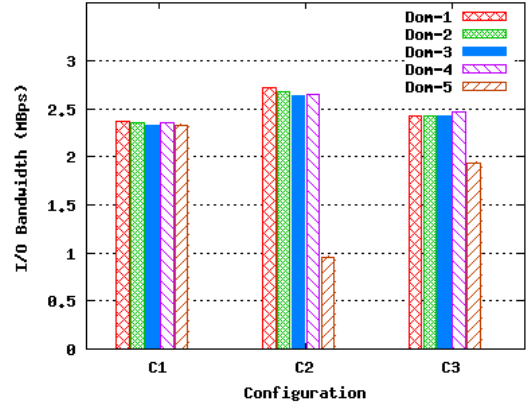


Fig. 1. Sharing of network bandwidth across domains.

correctly updates credit limits to restrict network I/O bandwidth utilization to the specified value.

Figure 1 depicts the effect of network bandwidth allocation across five domains (Dom-1 to Dom-5) in three different configurations. In the first configuration *C1*, no credit limit is set, and each domain shares the bandwidth in a fair manner (with each receiving around 2.3 MBps). The second configuration *C2*, has the bandwidth limit of Dom-5 changed to 1 MBps. As seen, Dom-5 receives a bandwidth of 0.96 MBps and the residual bandwidth is shared fairly among the other domains. *C3*, the third configuration has Dom-5's bandwidth limit set to 2 MBps. In this case as well, Dom-5 receives 1.93 MBps of bandwidth and the others share the residual bandwidth in a fair manner. The experiment demonstrates the correctness in network bandwidth utilization per domain, when the limits of allocation are updated dynamically.

V. CONCLUSIONS

In this paper, we describe the implementation of *xmsetbw*, a tool for dynamic reconfiguration of network I/O bandwidth limits, for virtual machines in Xen. Experimental evaluation verified the correctness of the tool when network I/O usage limits were changed dynamically. As part of future work, we aim to port *xmsetbw* to newer versions of Xen and explore the interaction and co-existence of limits on CPU and I/O utilization of Xen domains.

Acknowledgement: This research was funded by the IBM Faculty Award 2008.

REFERENCES

- [1] David Chisnall, *The Definitive Guide to the Xen Hypervisor*, Technical report, Prentice Hall, 2007.
- [2] *XenStore Reference*, <http://wiki.xensource.com/xenwiki/XenStoreReference>
- [3] *XenBus*, <http://wiki.xensource.com/xenwiki/XenBus>
- [4] Paul Barham et. al., *Xen and the Art of Virtualization*, Symposium on Operating Systems Principles, 2008. Neugebauer, Ian Pratt, and Andrew Warfield, *Xen and the Art of Virtualization* SOSP, October 2003.
- [5] Sriram Govindan et. al., *Xen and co.: Communication-aware CPU Scheduling for Consolidated Xen-based Hosting Platforms*, International Conference On Virtual Execution Environments, 2007.
- [6] Diego Ongaro, Alan L. Cox, and Scott Rixner, *Scheduling I/O in Virtual Machine Monitors*, International Conference on Virtual Execution Environments, 2008.