國立臺灣大學管理學院資訊管理學系

碩士論文

Department of Information Management

College of Management

Master Thesis

在實體與虛擬環境下應用資源使用之比較

On the Comparison of Resource Usage of Applications in Physical and Virtual Environments

林陳駿

Chen-Chun Lin

指導教授：孫雅麗 博士

Advisor: Dr. Yeali S. Sun, Ph.D.

中華民國 104 年 2 月

Feb, 2015

謝詞

首先，我要感謝的是我的指導教授孫雅麗老師，在研究所的幾年內承蒙老師的指導與幫忙，讓我能夠在各個研究領域能有長足的進步；而在我灰心喪志的時候，老師則不停的在後面推我一把，給予我繼續完成論文的動力。此外，我要感謝陳孟彰老師與潘育群博士的指點，讓我在研究的過程中，能夠獲得更寬廣、更全面的視野進而找到不一樣的突破點。

接著要感謝的是研究室中的所有學長姐、學弟妹以及同屆的同學們。每當我在學術研究上遭受到挫折與困難時，能夠靜心地聆聽我的宣洩，使我的壓力能夠適當的找到出口。而在學術研究上碰到困難的時候，學長姐們也能夠適時的給予必要的協助。

最後要感謝的是我的母親。母親總是在我帶著一身疲憊到家的時候，給予我溫暖的擁抱與笑容。缺少母親在我後方的支援，想必我也無法順利的完成此篇論文。

最後，再一次的感謝陪伴我至今的人們，謝謝。

林陳駿 謹識
于臺大資訊管理學研究所
中華民國一零四年二月

論文摘要

論文題目：在實體與虛擬環境下應用資源使用之比較

作者：林陳駿　　　　　　　　　　　　　　　一零四年二月

指導教授：孫雅麗 博士

隨著雲端科技以及虛擬化技術的興起，愈來愈多的企業試圖將內部的應用程式轉移到雲端環境之中。然而，虛擬化技術讓應用程式在執行上會產生額外的資源消耗（overhead），而這些應用程式又擁有各自的效能目標（performance goal）。因此，在這篇論文中，我們希望能夠回答出下列問題：「在給定工作量與效能目標的情況下，讓一個應用程式在虛擬環境上運作需要分配多少資源？」我們側重在找出特定應用程式在實體與虛擬化環境中資源消耗的差異與關係，以利於在移轉應用程式至虛擬環境時能夠得到妥善的配置。

在為了能夠得知應用程式在虛擬環境中所消耗的資源，我們提出了一個模型的雛形來預測。在給定應用程式的效能目標以及工作量下，輸入應用程式在實體環境中資源消耗的參數，進而預測出該應用程式在虛擬環境中資源消耗的量。

此外，我們以串流服務做為主要研究的應用程式。我們嘗試藉著實驗找出客戶端數量、影片位元傳輸速率以及資源消耗間的關係。

關鍵詞：雲端運算、串流服務、應用程式側錄

THESIS ABSTRACT

On the Comparison of Resource Usage of Applications in Physical and Virtual Environments

By Chen-Chun Lin

DEPARTMENT OF IMFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

February 2015

ADVISER : Yeali S. Sun, Ph.D

As the development of cloud computing advances, increasing number of enterprises start to move their existing physical machine hosted applications to virtualized cloud environments. However, virtualization technology will cause extra overhead while running application in virtual environment. Thus, knowing about resource consumption of the application in virtualization environment becomes an important issue. Each application has own performance goal, which will be defined in service level agreement (SLA). According resource consumption and SLA of the application, we'd like to answer the following question: "How much resources are necessary to be allocated to the application in the virtualized environment given workload subject to its target performance goals?" In this thesis, we focus on finding out the relationship of resource consumption of the specific class of application running in physical and virtual environment. Hence, the application can be allocated appropriate resource while moving from physical into cloud.

In order to know the resource consumption (i.e. CPU usage) of application running in virtual environment, we propose a miniature model to predict. By inputting the parameter matrix of workload, resource consumption in physical machine and performance goal, the model can predict the resource requirement of application in virtualization environment subject to performance goal.

In this paper, we choose streaming service as our research application. We try to discover the relationship of number of sessions, bit rate of video and resource consumption. Through the experiment we find out a simple relationship between them.

Keywords： cloud computing, streaming service, application profiling

Table of Contents

List of Figure

1. Introduction

Cloud computing has become a computing paradigm due primarily to its elasticity and cost effectiveness. The computing paradigm organizes a shared pool of servers in datacenters into a cloud infrastructure that can provide on-demand server utilities to users anytime. To enable different applications running on a cloud efficiently, virtualization is often applied, which allows multiple virtual machines (VMs) to run on the same physical machine. As the development of cloud computing advances, increasing number of enterprises start to move their existing physical machine hosted applications to virtualized cloud environments. However, application moving is not just a simple issue. Each application has its own target performance goals. When the application moved to the cloud, these goals are typically specified in the service level agreement (SLA) which is concluded to supply high assurance in terms of Quality of Service (QoS) metrics to confirm the services would run normally and rightly. It is important that cloud service providers (CSP) honor the performance agreement in the SLA.

Low utilization has long been a major issue in running datacenters. The main source of the issue is resource over-provisioning. Datacenters operators often provision resource based on the peak loads, but peak load is not a common case. However, virtualization technology can adjust resource of each VM dynamically. So, in cloud application performance management, a CSP must be capable of knowing when and how much resource are needed by the application and allocating according resources to the application in time. Nevertheless, a CSP faces a problem of how much resources should be provisioned to the application given its performance goals when moving the application to a virtualized cloud environment. In [1], Wood et al. had pointed that virtualization will incur additional overhead due to the existence of hypervisor, in other words, an application will consume more resources in virtualized environment than in physical environment.

According to the above, in this work, we'd like to answer the following question:

➢ How much resources are necessary to be allocated to the application in the virtualized environment given workload subject to its target performance goals?

It's not a simple issue due to the multiple classes of applications and variation of workload on applications; different classes of applications have different runtime behavior and workload characteristics. Ordinarily, for allocating proper resources, a CSP need to distinguish the class of the application and then analyze the characteristic of workload. The work is strenuous and time consuming.

We want to create the "WPR model" of a specific type of applications that maps the native system resource usage file, workload, and performance into the virtualized one. The WPR model helps to predict the resource requirement of an application before moving to the virtual machine.

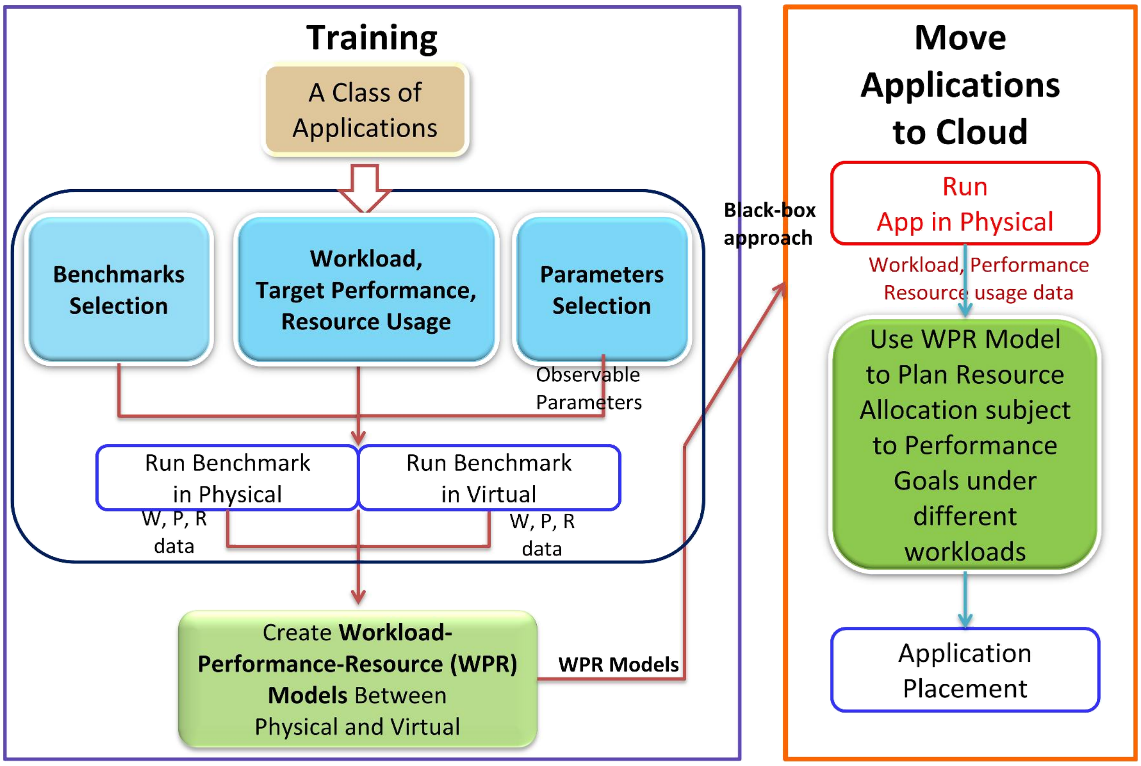

Training
A Class of Applications
Benchmarks Selection
Workload, Target Performance, Resource Usage
Parameters Selection
Observable Parameters
Run Benchmark in Physical
Run Benchmark in Virtual
W, P, R data
W, P, R data
Create Workload-Performance-Resource (WPR) Models Between Physical and Virtual
WPR Models
Black-box approach
Move Applications to Cloud
Run App in Physical
Workload, Performance Resource usage data
Use WPR Model to Plan Resource Allocation subject to Performance Goals under different workloads
Application Placement

Figure 1 Execution flow

Figure 1 is the execution flow of our methodology; there are two phases in our flow – "Training" and "Move application to cloud". In training phase, we'd like to train WPR model to profile a specific class of application. WPR model fingerprints the features of the specific class of application located in physical and virtual machine, such like resource consumption, performance metrics, and workloads. These data are collected to train WPR model which constructs a resource consumption relationship between physical and virtual machine. By using WPR model, we can predict the resource consumption of a specific type of application while moving it to the cloud.

At the first, the flow starts at "A class of applications". We choose a specific class of applications to focus on, and then research the architecture, working flow, resource consumption of the class of application. In this work, we focus on the streaming service for several reasons:

- Streaming service is one of the most popular services currently on the Internet.

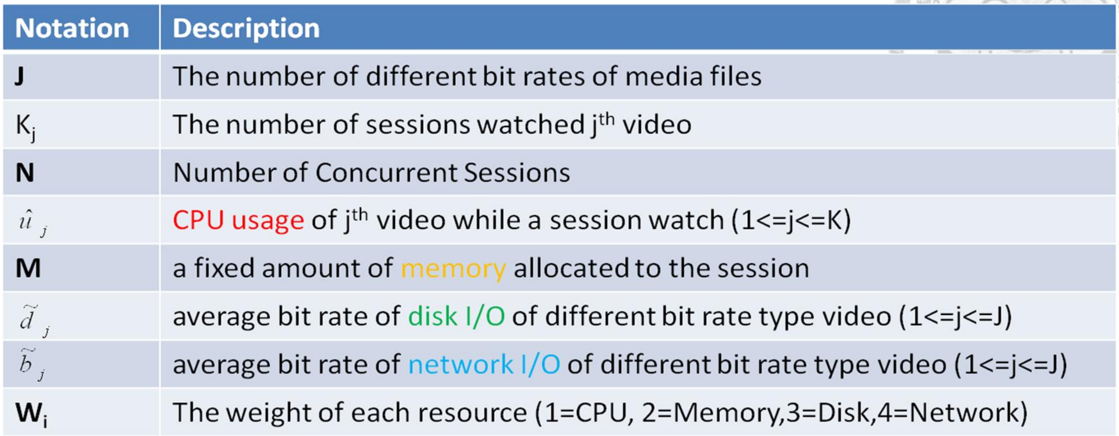
- The live event broadcasting and video on demand are two most common services.
- Specifically, the former has the most stringent real-time requirement.
- The service typically serves a large user population but its workload varies greatly depending on special event like Olympic or Lin Sanity.

We consider streaming service as one of the best candidate applications that can greatly benefit from the resource pooling and rapid, dynamic resource management of the cloud. There are many previous works which focus on the class of CPU-intensive applications such as web applications. However, in [2], Loren Staley had proved that streaming service is different. Depending on various workloads, streaming server is not only bounded by CPU, but also disk I/O.

Next, we introduce the second row which include “Benchmarks Selection”, “Workload, Target performance, Resource usage”, and “Parameters Selection” three steps. In “Benchmarks Selection”, we’ll find out an application of the class of application as the benchmark. We believe that the applications of the same class have similar features; the benchmark application is the standard which represents the class of application.

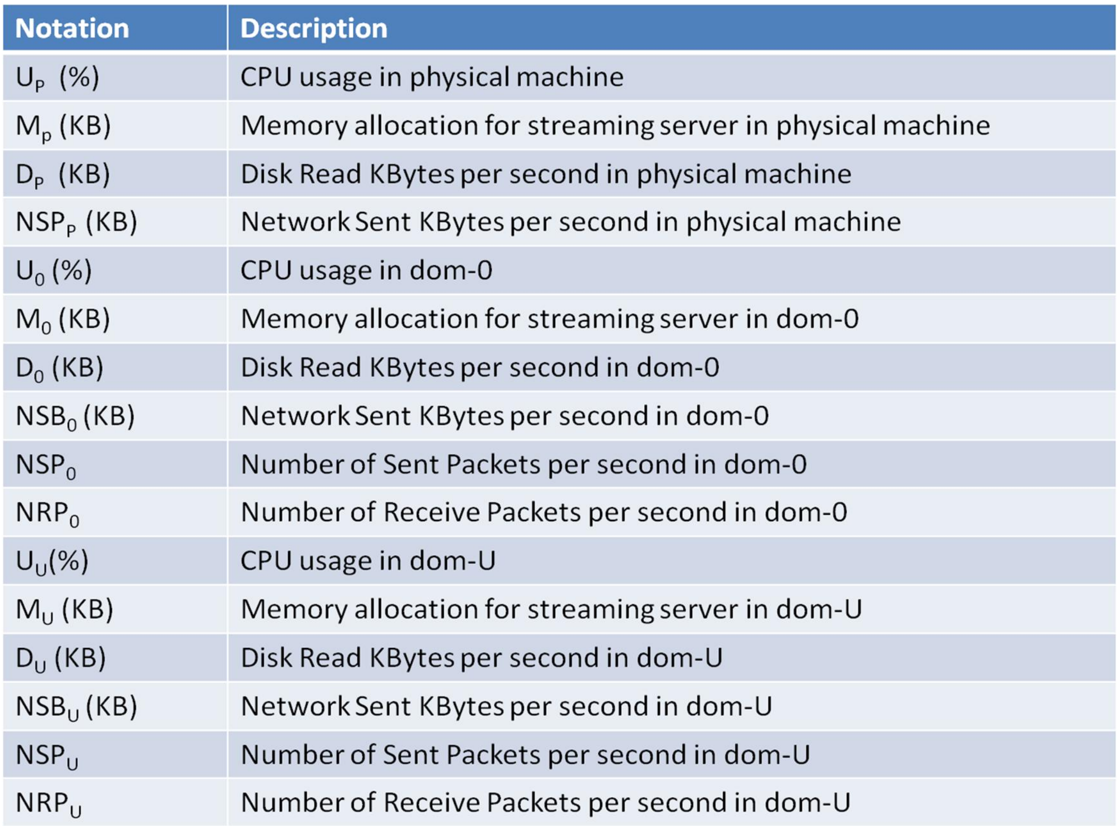
In “Workload, Target performance, Resource usage” step, statistic methods such as linear regression (LR), canonical correlation analysis (CCA) are used to train the WPR model. However, in [3], Sajib et al. tried to model the profile of web application using machine learning such as artificial neural network, but it’s hard to explain the meaning of result. The complex relationship of resource consumption between physical and virtual machine cannot use a simple method to explain.

In the last part of second row, “Parameters Selection”, we’ll define a number of parameters that related to WPR model, these parameters are included in 3 dimensions – workload, performance, and resource. In the dimension of workload, we assume that playing a constant bit rate (CBR) video on server will consume constant resources. The sum of the resource consumptions is system demand. Next, in the dimension of performance, we select the maximum sessions and throughputs that server can handle as our parameters. Last, 4 classes of resource consumption – CPU, memory, disk I/O, and network I/O, will be the resource parameters. All the parameters of 3 dimensions are showed in Figure 2, 3, and 4.



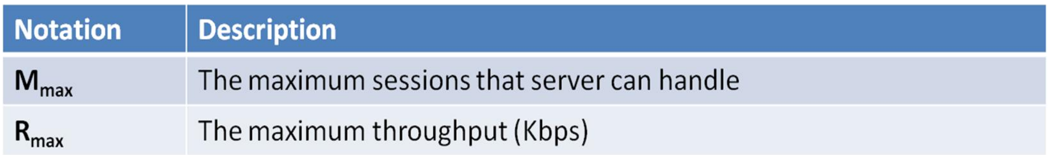
Notation	Description
J	The number of different bit rates of media files
K_j	The number of sessions watched j^{th} video
N	Number of Concurrent Sessions
$\hat{u}_j$	CPU usage of j^{th} video while a session watch (1<=j<=K)
M	a fixed amount of memory allocated to the session
$\tilde{d}_j$	average bit rate of disk I/O of different bit rate type video (1<=j<=J)
$\tilde{b}_j$	average bit rate of network I/O of different bit rate type video (1<=j<=J)
$\mathbf{W_i}$	The weight of each resource (1=CPU, 2=Memory,3=Disk,4=Network)

Figure 2 Parameters of Workload



Notation	Description
U_P (%)	CPU usage in physical machine
M_p (KB)	Memory allocation for streaming server in physical machine
D_P (KB)	Disk Read KBytes per second in physical machine
NSP_P (KB)	Network Sent KBytes per second in physical machine
U_0 (%)	CPU usage in dom-0
M_0 (KB)	Memory allocation for streaming server in dom-0
D_0 (KB)	Disk Read KBytes per second in dom-0
NSB_0 (KB)	Network Sent KBytes per second in dom-0
NSP_0	Number of Sent Packets per second in dom-0
NRP_0	Number of Receive Packets per second in dom-0
U_U(%)	CPU usage in dom-U
M_U (KB)	Memory allocation for streaming server in dom-U
D_U (KB)	Disk Read KBytes per second in dom-U
NSB_U (KB)	Network Sent KBytes per second in dom-U
NSP_U	Number of Sent Packets per second in dom-U
NRP_U	Number of Receive Packets per second in dom-U

Figure 3 Parameters of Resource



Notation	Description
$\mathbf{M}_{max}$	The maximum sessions that server can handle
$\mathbf{R}_{max}$	The maximum throughput (Kbps)

Figure 4 Parameters of Performance

In "Run Benchmark" step, we run the benchmarks both in physical and virtual environment. Based on the same workload, the same performance goal, and the same configuration of machines, we run the benchmarks and take the result such as resource consumption data. The data will put together to input into the method which we chose to train the WPR model and then find out the relationship between physical and virtual machines.

After training phase, we'll start to use WPR model to predict the resource requirement for a certain type of application in the virtual environment. In this work, streaming service is the candidate application. After running the streaming service in a new physical environment, we'll collect some data which WPR model needed. According to the data, we can use WPR model to plan resource allocation subject to performance goals under different workloads in virtual environments. As the result output, finally, we can place the application onto the cloud with the appropriate resource allocation.

2. Related work

2.1 Virtualization Platform and Monitor tool

Virtualization is gaining popularity in cloud environment as a software-based solution for building hardware infrastructures. There are numbers of virtualization software in existence today such as Microsoft Hyper-V [4], VMware [5], and open source Xen [6].

VMware [7] have released numbers of benchmarks for quantifying the performance of virtualized environment. However, the benchmarks do not have ability to characterize virtualization overhead compared to the native system. Xen is an open source platform; there are lots of researches which designed performance monitors for Xen, such as XenMon [8], Oprofile [9]. For Microsoft Hyper-V, the performance monitors are default installed in Windows Server. Users can monitor the performance and resource usage of each virtual machine and get performance metrics data dynamically from these monitors.

2.2 Profiling and Performance Modeling

Wood et al. [1] propose a combination of application modeling and virtualization overhead profiling for estimating the virtual machine CPU utilization of an application when it moved from physical environment to virtualized environment. They create a set of micro-benchmarks to profile the different types of virtualization overhead and use linear regression to train the resource requirement model. The model is similar as our, however it does not consider about the performance goal of each application.

Sajib et al. [3] try to model the application performance in a virtualized environment. They identify a key set of virtualization architecture independent parameters which influence application performance for a diverse and representative set of applications. In addition, they compare the accuracy between different model training techniques which based on regression and artificial neural networks. The work is different from us. They focus on changing resource parameters to modulate application performance in virtualized environment. Given a virtual machine with fixed resource allocation, the model can predict the application performance in the VM with such configuration. The model and our work are just opposing. Given the application performance goal and we'd like to predict the resource requirement for the application in virtualized environment.

Sujesha et al. [10] issue a problem of network affinity-aware resource provisioning for virtual machine. The problem is caused by co-location of communicating virtual machines, which will incur less overhead due to the I/O architecture of hypervisor. The co-location virtual machines can communicate between shared memories instead of transferring packets across network. They build a model based on different resource usage micro-benchmarks to predict the resource usage when moving from non-co-located placements to co-located placement and vice-versa. However, the model does not consider scalable workload, the result of their environment show the small difference in each value and all the values show the low overhead and usage. One feature of cloud service is the large scale of its resources, workload, and architecture, so the hard overhead is needed in evaluation.

Omesh et al. [11] propose three challenges and some approaches about modeling virtual machine performance. The three challenges are (i) The interference caused by the other virtual machines running on the same platform. (ii) Contention in shared resources which are visible (cores, memory, disk I/O …etc) and invisible (cache space, memory bandwidth …etc) between virtual machines. (iii) The difference of specifies of virtualization technology and the scheduling algorithm implemented by virtual machine monitor. However, we believe a good hypervisor will implement complete isolation between virtual machines and thus there will be no contention and interference between them. It’s hard to measure and test about the resource contention between virtual machine due to the immature of virtualization technology and hardware.

3. Background

3.1 Background – Virtualization Platform : Xen 4.0

- ### Basic knowledge

Remarkable advances have been made in technology of virtualization. Xen is one of symbolized virtualization platform. Xen implements para-virtualization which does not emulate I/O device driver in hypervisor but creating an additional administrative virtual machine which help users to control hypervisor. In Xen, the administrative virtual machine is called Domain-0 (Dom-0), and the virtual machines or guest OSs are called Domain-U (Dom-U). Dom-0 is a unique virtual machine running on the Xen hypervisor that has special rights to access physical I/O resources as well as interact with the other virtual machines. According to Dom-0, administrator can monitor resource consumption of each Dom-U at any time and adjust resource allocation of each Dom-U dynamically.

To create a new virtual machine in Xen, it needs to configure the memory, CPU and disk storage for the new VM. The memory can be configured up to 8G bytes; the number of CPU can be set from one to the number of cores which the host machine owned; and for the disk storage, one can assign fixed number of storage on host machine or other existing storage.

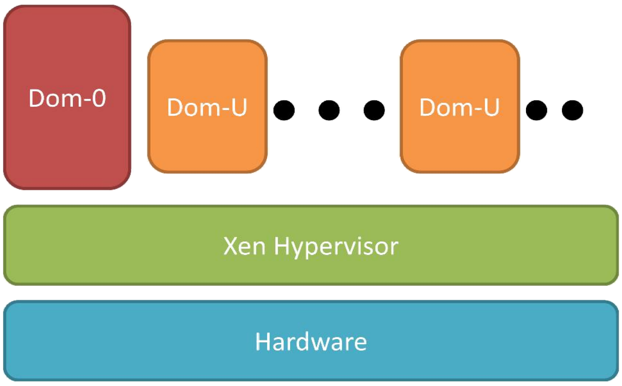
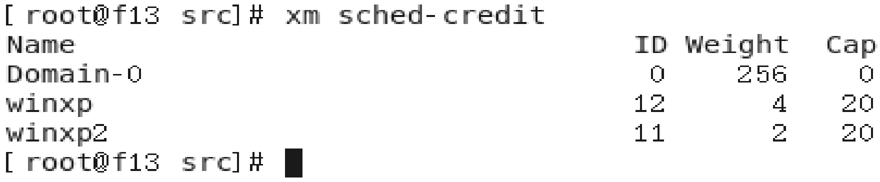


Figure 5 Xen architecture

- ### Credit-based CPU Scheduler

Credit-based CPU scheduler is the default CPU scheduling algorithm in Xen now. For, each domain, Xen assigns two special values – "weight" and "cap". The value "weight" is like the proportion of time that domain can use CPU. A domain with a weight of 512 will get twice as much CPU as a domain with a weight of 256 on a contended host. In the other, the value "cap" is just like the upper bound, the cap optionally sets the maximum amount of CPU a domain will be able to consume. In Xen, the cap is expressed in percentage of one physical CPU. The value of cap with 100 is 1 physical CPU, the value 50 is half of a CPU, and the value 400 is 4 CPUs, etc. In addition, administrator can restrict virtual CPU running on a particular physical CPU by using the generic vcpu-pin interface.

Each physical CPU manages a local run queue of runnable virtual CPUs. Each virtual CPU will be set into two priorities – “Over” and “Under”. The priority “over” means the virtual CPU cannot use physical CPU anymore, and the priority “under” is opposite to “over”. As a virtual CPU runs, it consumes “credits”, which default set to weight. If a virtual CPU’s credit become negative, its priority will change from “under” to “over”. While all the virtual CPUs’ priority change into over, Xen will set their priority back to “under” and give each virtual CPU its credit.



```
[root@f13 src]# xm sched-credit
Name                                      ID Weight  Cap
Domain-0                                   0    256    0
winxp                                     12      4   20
winxp2                                    11      2   20
[root@f13 src]# █
```

Figure 6　Credit-based CPU scheduler in Xen

● Network I/O in Xen

For network I/O in Xen, a VM will be connected to a virtual network interface. Xen will assign a MAC address to this virtual network interface.

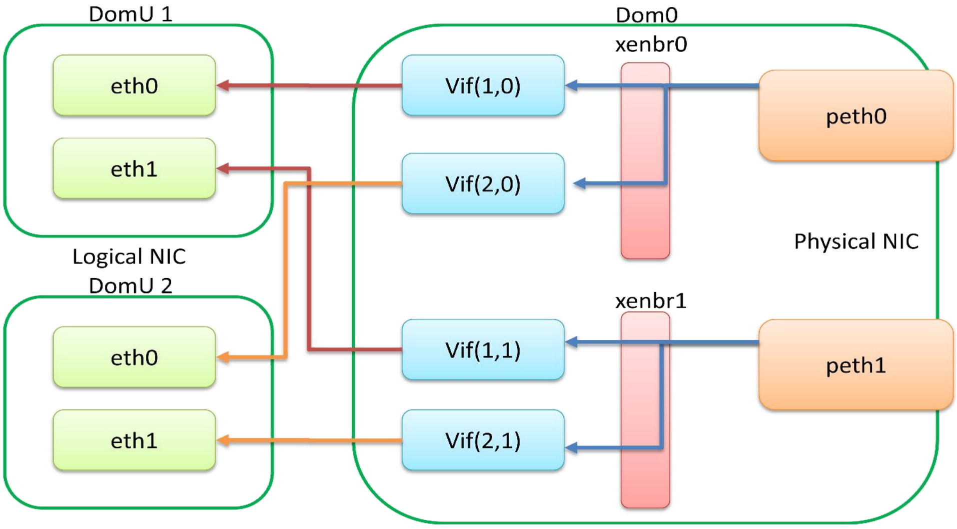


Figure 7　Virtual bridge in Xen 4.0

A packet arrived at physical network interface is handled by Dom-0 Ethernet driver and appears on peth0. The physical NIC is bound to the virtual bridge, and the packet is passed to the bridge. The bridge distributes the packet, just like a switch would

There are numbers of virtual interface (Vif) denoted as vif(X,Y) connected to the virtual bridge. (The X means the number of Dom-U and the Y means the number of logical NIC in Dom-U.) The virtual bridge makes a decision to deliver the packet to which virtual interface according to the receiver's MAC address,. Then the virtual interface put the packet to the hypervisor and then hypervisor transfers the packet into the Dom-U where the virtual inteface leads to.

3.2 Background – Darwin Streaming Server

In this work, we choose Darwin Streaming Server (DSS) [12] as the benchmark streaming application. DSS is an open source version of Apple's QuickTime Streaming Server technology. Users can stream QuickTime and MPEG-4 media by using live streaming or video on demand (VOD) service and they can modify the existing streaming server code to fit their needs. DSS provides streaming media to clients across the Internet using the standard RTP and RTSP protocols and supports a variety of platforms like Linux, Windows, and Solaris. Also, DSS supports a number of media formats including QuickTime File Format (.mov), MP3(.mp3), MPEG-4 (.mp4), and 3GPP(.3gp).

In the following section, we'll introduce the streaming service workflow in Darwin streaming server.

- **Workflow in Physical Machine**

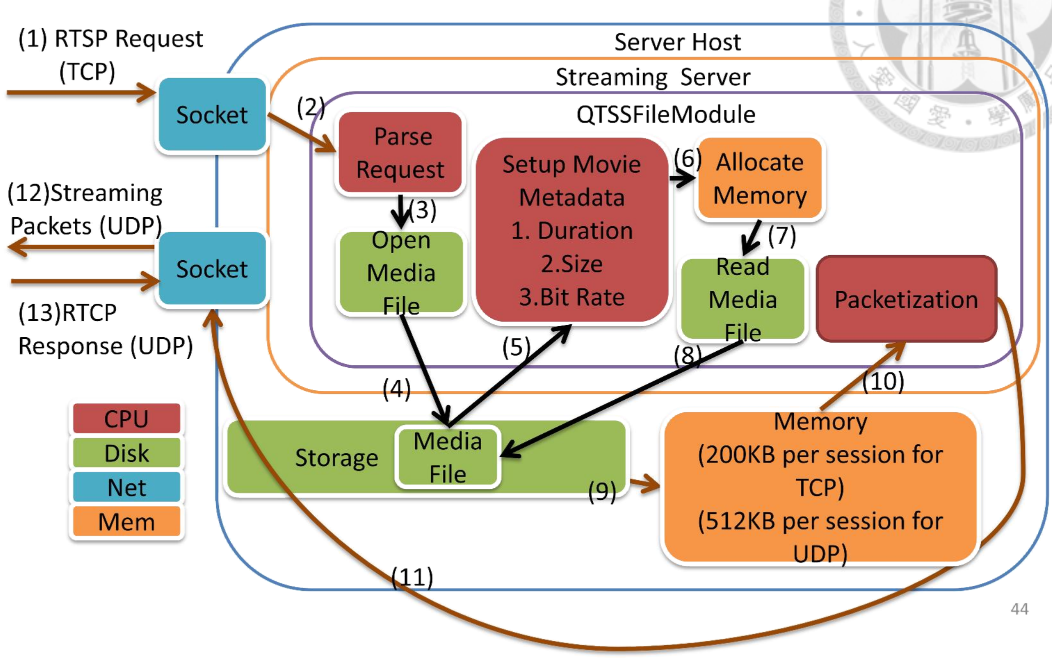


Figure 8 Workflow in physical machine

We describe the workflow of streaming service in physical machine in figure 8. At first, client sends RTSP request to server host trying to construct the connection. While server host receives the request, it sends the request to QTSSFileModule, which executes main service in Darwin streaming server. The module starts to parse the request and analyzes which media file are requested by client. Then server tries to open the media file in storage and setup the metadata about the media file, the metadata includes the duration of media, the file size and the bit rate of the file. After opening the media file, server host allocates memory for the connection and begin to read media file into allocated memory. Thus, the media file could be bundled into packets and then server can send streaming packets to client. Client will send RTCP response beck to server during streaming.

- **Workflow in Virtual Machine**

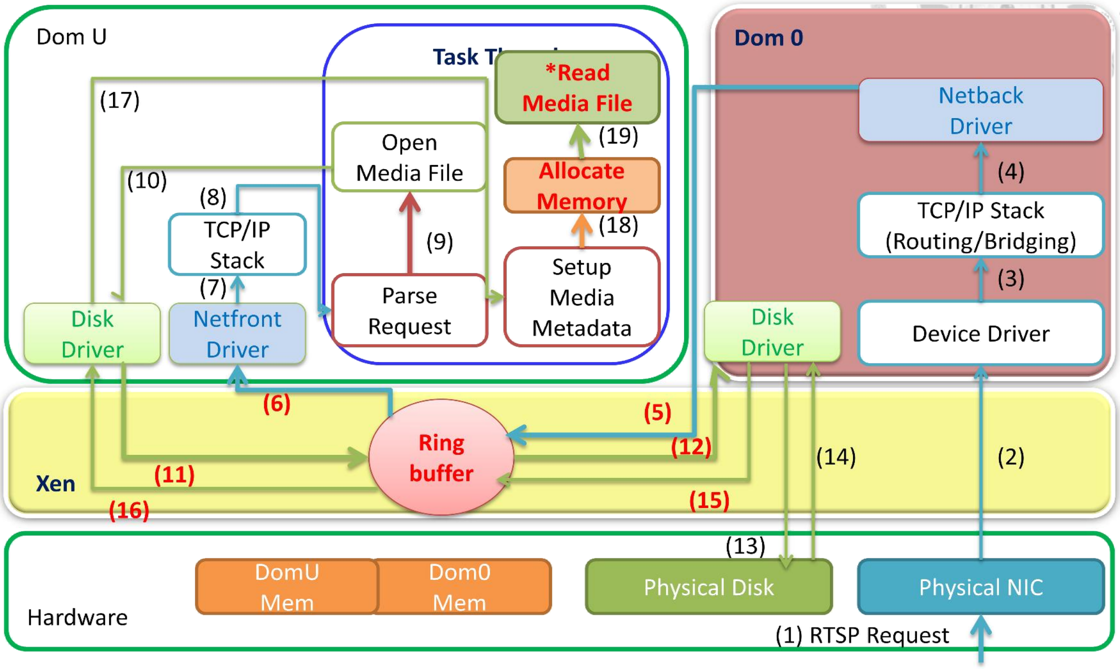


Figure 9 Establish session in virtual machine

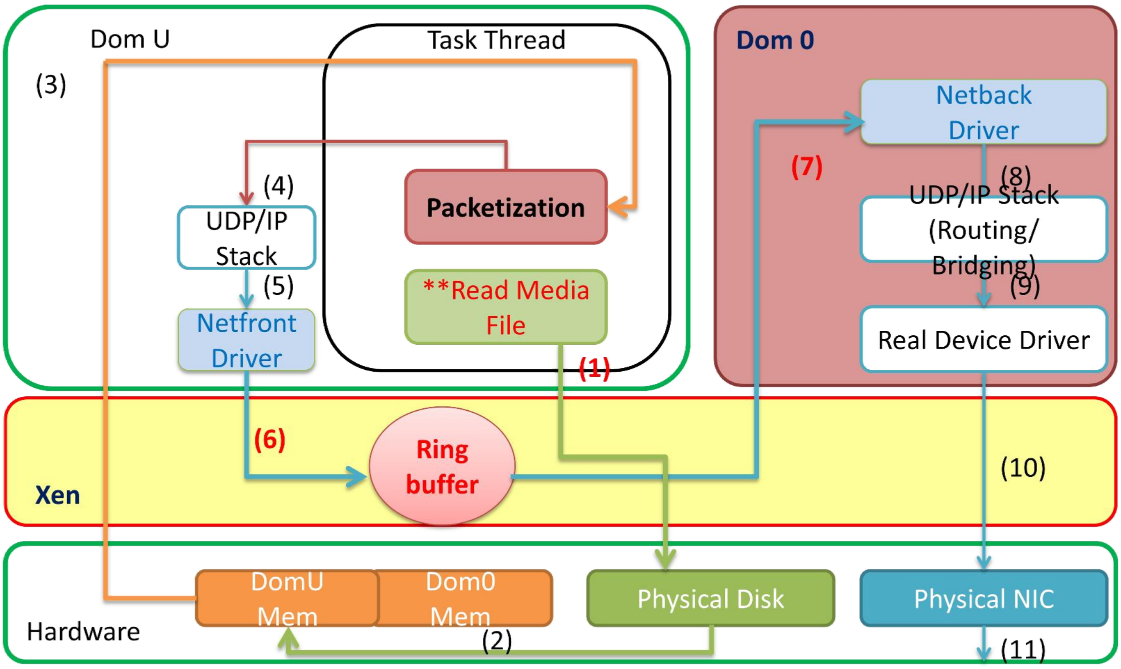
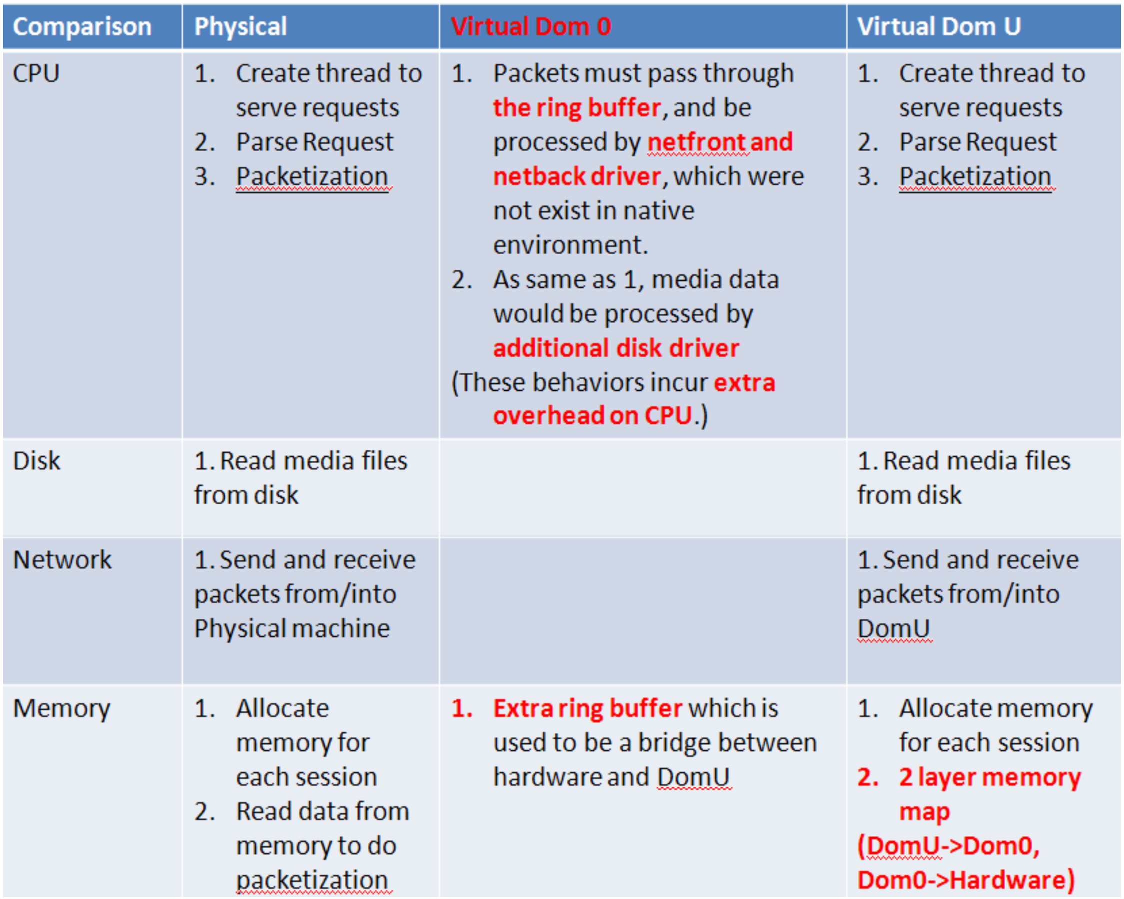


Figure 10 Send content packets in virtual machine

In figure 9 and figure 10, we'd like to describe the workflow of streaming server in Xen. Most parts are similar as the workflow in physical environment. The different points are the attendance of Xen hypervisor and Domain 0.

At first, client sends the RTSP request trying to connect the host server. The request will reach physical NIC first, and then is delivered to Dom-0's device driver. After Dom-0 receive the request, Dom-0 sends the request to netback driver and copy the request into the ring buffer in Xen hypervisor. Xen distinguishes the request and delivers it into corresponding Dom-U. In Dom-U, the request is proceeded like in physical environment. However, opening and reading the media file, allocating memory and sending packet, these commands of the same type which must communicate to the hardware will pass through Xen hypervisor and Dom-0. These additional works will incur the extra overhead due to the virtualization.

In figure 11, we classify the workflow which works in physical machine, Dom-0 and Dom-U into 4 catalogs – CPU, Disk, Network and Memory.



Comparison	Physical	Virtual Dom 0	Virtual Dom U
CPU	1. Create thread to serve requests 2. Parse Request 3. Packetization	1. Packets must pass through **the ring buffer**, and be processed by **netfront and netback driver**, which were not exist in native environment. 2. As same as 1, media data would be processed by **additional disk driver** (These behaviors incur **extra overhead on CPU**.)	1. Create thread to serve requests 2. Parse Request 3. Packetization
Disk	1. Read media files from disk		1. Read media files from disk
Network	1. Send and receive packets from/into Physical machine		1. Send and receive packets from/into DomU
Memory	1. Allocate memory for each session 2. Read data from memory to do packetization	**1. Extra ring buffer** which is used to be a bridge between hardware and DomU	1. Allocate memory for each session **2. 2 layer memory map (DomU->Dom0, Dom0->Hardware)**

Figure 11 Comparison of physical and virtual machine by resource consumption

4. Evaluation

In this section, we perform some experiments to evaluate the relationship between workload and resource consumption.

4.1 Environment

The experiments are constructed on 2 physical machines and 1 virtual machine. All physical machines are connected through a 10/100/1000 Mbps Ethernet switch. Each physical machine is equipped with Intel i3-550 CPU with 2 cores and 4 threads. The RAM is 8GB and the storage is 1TB. The virtual machine host on one of the physical machines and the other one works as workload generator. We use Xen 4.0.1 as our virtualization hypervisor and Fedora 14 as OS. The VM is equipped with 2 cores CPU, 4GB of RAM and 40GB storage.

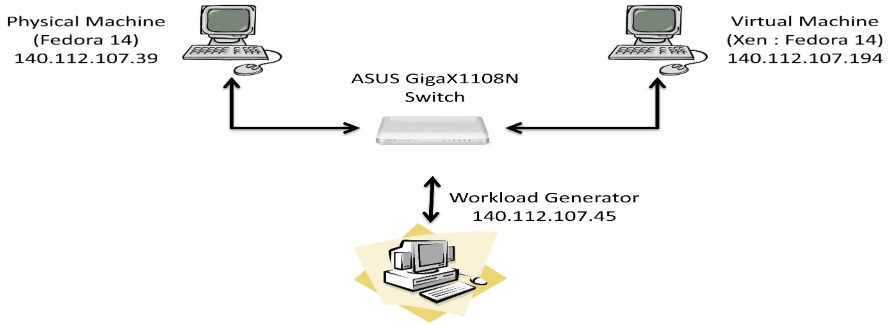


Figure 12 Experiment environment

We use Darwin Streaming Server as our benchmark application and choose 2 constant bit rate (CBR) videos (100kbps and 300kbps) as our test data. The reason that we choose CBR video is that CBR videos are simpler to observe and control than variable bit rate (VBR) videos. In addition, a famous live streaming server – hicahnnel, is also using CBR video. The workload is generated by "streaming load tool", an open source application which can generate numbers of clients connect to the streaming server at same time and record log.

4.2 Resource consumption under different workloads

In this section, we are trying to evaluate the relationship between workloads and resource consumption of the benchmark application both in physical and virtual environments. At first, we run streaming server on physical machine and add the number of sessions to show the simplest case.

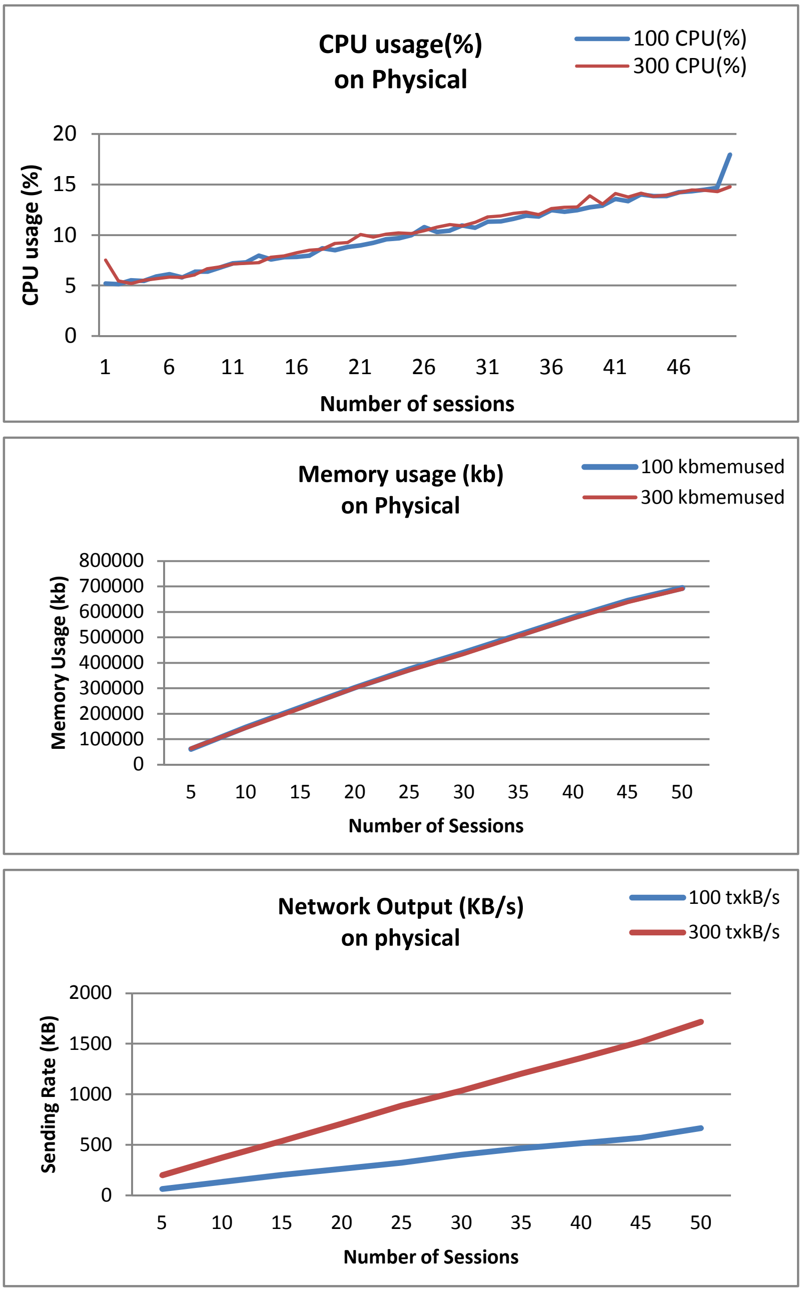


Figure 13 streaming server on physical machine

It's clear to see while the bit rate of video is fixed, there is a linear relationship between each type of resources (CPU, memory and network I/O) and the number of sessions. In Darwin streaming server, each session will be allocated a fixed number of memory. It’s predictable that memory consumption is closely linked to the number of sessions. We can see that the result of memory consumption have no different between play 300kbit video and 100kbit video. However, the result of CPU usage is not obvious.

Next, we run the streaming server on the virtual machine to exam whether the result is similar to run on physical machine.

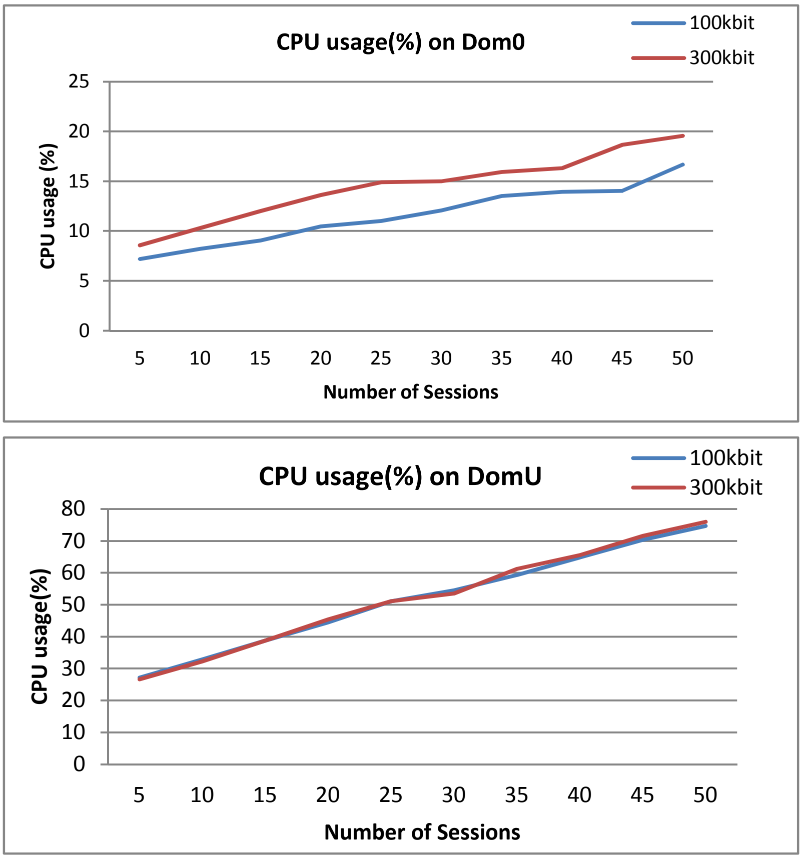


Figure 14 CPU usages in virtual machine

In figure 14, we can see that the application which played 300kbit video consumed more CPU than 100kbit video in dom-0 but is not obvious in dom-U. We believed the result is caused by the extra packets which are produced by 300kbit video and transmitted by dom-0.

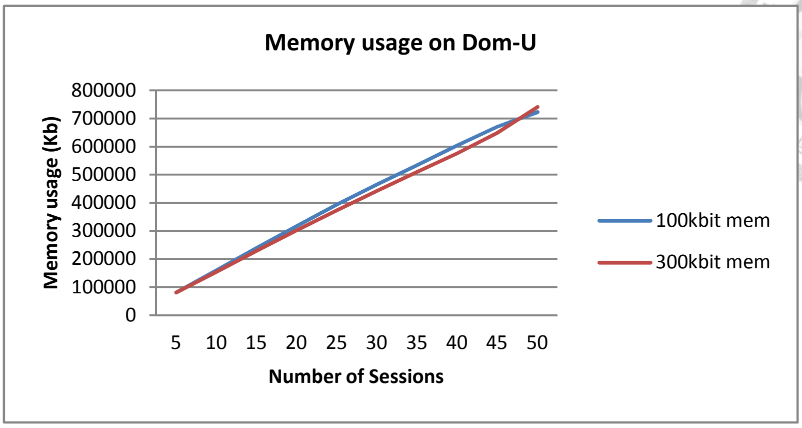


Figure 15 Memory usages on Dom-U

In figure 15, we can find the result is similar to the memory usage on physical machine. The result proves that memory usage is only associated to the number of sessions.

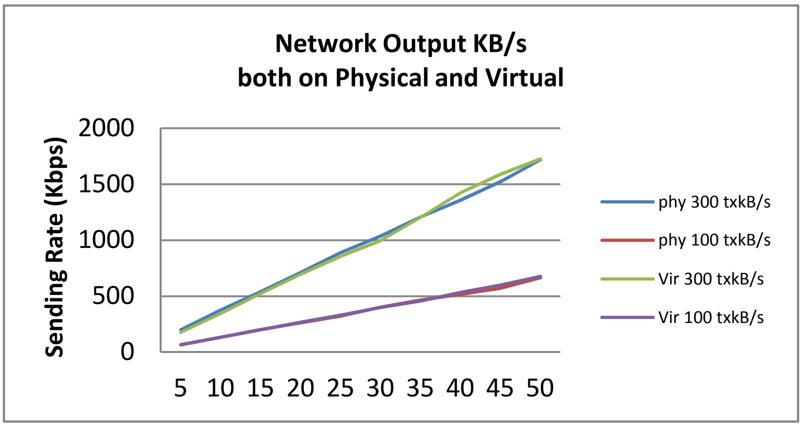


Figure 16 Comparison of Network output

In figure 16, we can see that the output rate of network have no changes between run on physical and virtual machine. However, the sending rate of playing 300kbit video is 3 times larger than playing 100kbit video.

5. Conclusion and Future work

In this work, we describe the architecture and schedule algorithm detail of Xen clearly. Meantime, we research a specific type of application – streaming server. By studying Darwin streaming server, we understand the architecture of streaming server and its workflow both in physical and virtual environment. In addition, we analyze the resource consumption of streaming server step by step.

Based on the background, we introduce the miniature and execution flow of our WPR model which can help to predict the resource requirement of an application before moving to the virtual machine subject to its target performance goals. In the same time, we exam the relationships of resource consumption between different bit rate videos and the number of sessions both in physical and virtual environments.

However, the WPR model is not complete in this work. In the future, we are focus on establishing a better evaluation environment and create a WPR model to predict the resource requirement of streaming server.

6. Reference

[1] T. Wood, L. Cherkasova, K. Ozonat, and P. Shenoy, "Profiling and Modeling Resource Usage of Virtualized Applications," Proc. ACM/IFIP/USENIX 9th Int'l Conf. Middleware (Middleware), pp. 366-387, 2008.

[2] L. Staley, L. Cherkasova, L. Cherkasova, "Building a Performance Model of Streaming Media Applications in Utility Data Center Environment," Proc. of ACM/IEEE Conference on Cluster Computing and the Grid (CCGrid), May, 2003.

[3] S. Kundu, R. Rangaswami, K. Dutta, and M. Zhao, "Application Performance Modeling in a Virtualized Environment," Proc. IEEE High Performance Computer Architecture, 2010.

[4] Microsoft Hyper-V. "Windows server 2012 R2 white paper," Microsoft, 2012. [Online]. Available: http://download.microsoft.com/download/5/B/A/5BA1BA97-280F-4DB3-9775-3E47372A059C/Windows_Server_2012_R2_Cloud_Optimize_Your_Business_White_Paper.pdf

[5] VMware. "VMware distributed power management concepts and use," VMware, 2009. [Online]. Available: http://www.vmware.com/

[6] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," Proc. ACM Symposium on Operating Systems Principles (SOSP 03), Oct. 2003, pp. 164-177.

[7] VMmark: A Scalable Benchmark for Virtualized Systems. [Online]. Available: http://www.vmware.com/pdf/vmmark intro.pdf

[8] D. Gupta, R. Gardner, and L. Cherkasova, "XenMon: QoS Monitoring and Performance Profiling Tool," In Technical report, HPL-2005-187, 2005.

[9] A. Menon, J. Santos, and Y. Turner, "Diagnosing Performance Overheads in the Xen Virtual Machine Environment," Proc. ACM/USENIX international conference on Virtual Execution Environments (VEE 05), June 2005.

[10] S. Sudevalayam and P. Kulkarni, "Affinity-aware Modeling of CPU Usage for Provisioning Virtualized Applications," Proc. IEEE 4th International Conference on Cloud Computing , 2011

[11] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, "Modeling Virtual Machine Performance: Challenges and Approaches," Proc. ACM SIGMETRICS Performance Evaluation Review Volume 37 Issue 3, December 2009.

[12] Darwin streaming server. [Online]. Available: http://dss.macosforge.org/