



河海大学
HOHAI UNIVERSITY

云计算技术与应用

XEN的配置与性能对比

主讲人：白宏熙
指导老师：孙 宁

0 什么是XEN



介绍:

Xen 是一个开放源代码的 para-virtualizing 虚拟机（VMM），或“管理程序”，是为x86架构的机器而设计的。Xen 可以在一套**物理硬件**上安全的执行多个虚拟机。

应用范围:

- 服务器整合

在虚拟机范围内，**在一台物理主机上安装多个服务器**, 用于演示及故障隔绝。

- 无硬件依赖
- 多操作系统配置

以开发和测试为目的，同时运行多个操作系统

- 内核开发
- 集群运算
- 为客户操作系统提供硬件技术支持

本专题主要内容



1 XEN 的安装配置

2 virt-manager 的安装

3 创建虚拟机

4 使用UnixBench进行性能比较

1 XEN 的安装配置



先决条件:

- 主机操作系统: CentOS-6.5-x86_64-bin-DVD1.iso
- 所有早于 4.4 版的 Xen 都缺省启用 xm 及 xend。xen-4.4.1（及以上）的 rpm 改为支持 xl 及停用 xend
- 我们假设你对虚拟化已经有一定的理解，能分辨半虚拟化（pv）及全虚拟化（hvm）
- 运行 Xen 监督器的主机应该是一台 CentOS-6 / x86_64 的最少安装（**本文只是实践安装过程。注明：安装环境适用采用物理机进行安装，VMware workstation环境安装不成功。**）

1 XEN 的安装配置



基于Xen的操作系统架构:

- 一个Xen系统拥有多个层，最底层和最高特权层是 Xen程序本身。
- 在Xen的术语中，Domain由Xen控制，以高效的利用CPU的物理资源。
- 第一个domain，也就是domain 0（注：其实它就是第一个虚拟的客户系统），是在系统引导时自动创建，它拥有特殊的管理权限。
- 一个被称为xend的服务器进程通过domain 0来管理系统，Xend 负责管理众多的虚拟主机，并且提供进入这些系统的控制台。命令经一个命令行的工具通过一个HTTP的接口被传送到xend。
- 除非特别指定，所有指令均须由 root 用户在 Domain 0 机器上执行。

1 XEN 的安装配置



安装准备工作:

- 编译工具gcc

查看gcc版本: `#gcc -v`

安装gcc: `#yum install gcc gcc-c++ xz -y`
`#yum install -y gcc`

- Python 开发库

- 缺少perl Time HiRes组件(UnixBench运行所需组件)

`#yum -y install perl-Time-HiRes`

1 XEN 的安装配置



安装方式:

- 通过二进制软件包安装（几乎适用所有主要的Linux发行版）
- 从RPM包安装
- 通过源码包安装（CentOS 7.x上只能以源码安装Xen4CentOS）

1 XEN 的安装配置



安装命令:

此堆栈所发行的软件已经放置于 CentOS 镜像网络内一个专用于它的软件库内。要启用这个软件库，请确保你已启用 CentOS-Extras 软件库并输入：

```
#yum install centos-release-xen
```

安装Xen 内核和有关软件：

```
#yum install kernel-xen xen
```

这一命令将下载并安装以下程序包及其依赖包：

kernel-xen：该程序包存放的是主机操作系统和客户操作系统以及系统管理程序所用的Xen内核。

xen：该程序包存放的是跟Xen系统管理程序交互的用户空间工具。

bridge-utils：该程序包存放的是用于配置以太网桥的实用程序，以太网桥的作用是将多个以太网设备连在一起。

libvirt：该程序包存放的是一个C库，它提供了使用Xen框架所需的应用编程接口API。

libvirt-python：该程序包存放的是一个Python 模块，有了它就可以让Python 应用程序使用libvirt提供的Xen框架应用编程接口。

Python-virtinst：该程序包存放的是一个Python 模块，该模块使用libvirt在虚拟机内安装Fedora /Red Hat Enterprise Linux。

Xen-libs：该程序包存放的是Xen的管理程序所需的程序库。

=====REBOOT后=====

检查 xen 是否在运作中：

```
#xl info
```

查看Domain0

```
#xl list
```


1 XEN 的安装配置

GRUB引导配置问题:

修改/boot/grub/grub.conf, 将default=1改为default=0。这样就把系统默认引导的内核从Fedora内核改成了Xen内核（此案例中默认引导就是xen内核，不需要修改，可以在xen引导名称上加上xen以示区分）

2 virt-manager 的安装



为什么是Libvirt:

截至此点，你已经可以创建首台虚拟机器，而达致此目的的方法很多。对于新用户来说，如果要寻找最简单的安装途径，我们推荐采用 **Libvirt**。高级用户普遍会采用 **xen** 的命令行工具，以手动方式设置网络、存储方案及 **xen** 的执行环境。

- 采用 **Xen** 命令行工具，包括 **xl**
- 采用 **LibVirt/ Virt-install** 及 **Virt-Manager** 来管理 **xen4centos**

2 virt-manager 的安装



安装命令:

安装 libvirt 所需的基本组件:

```
#yum install libvirt python-virtinst libvirt-daemon-xen
```

安装 virt-manager:

找到virt-manager-0.9.0-31的CentOS版本, 进行本地安装(直接yum安装的virt-manager-0.9.0-34版本, 其对中文字符集支持不好):

```
#yum localinstall virt-maclearnager-0.9.0-31.el6.x86_64.rpm
```

启动virt-manager:

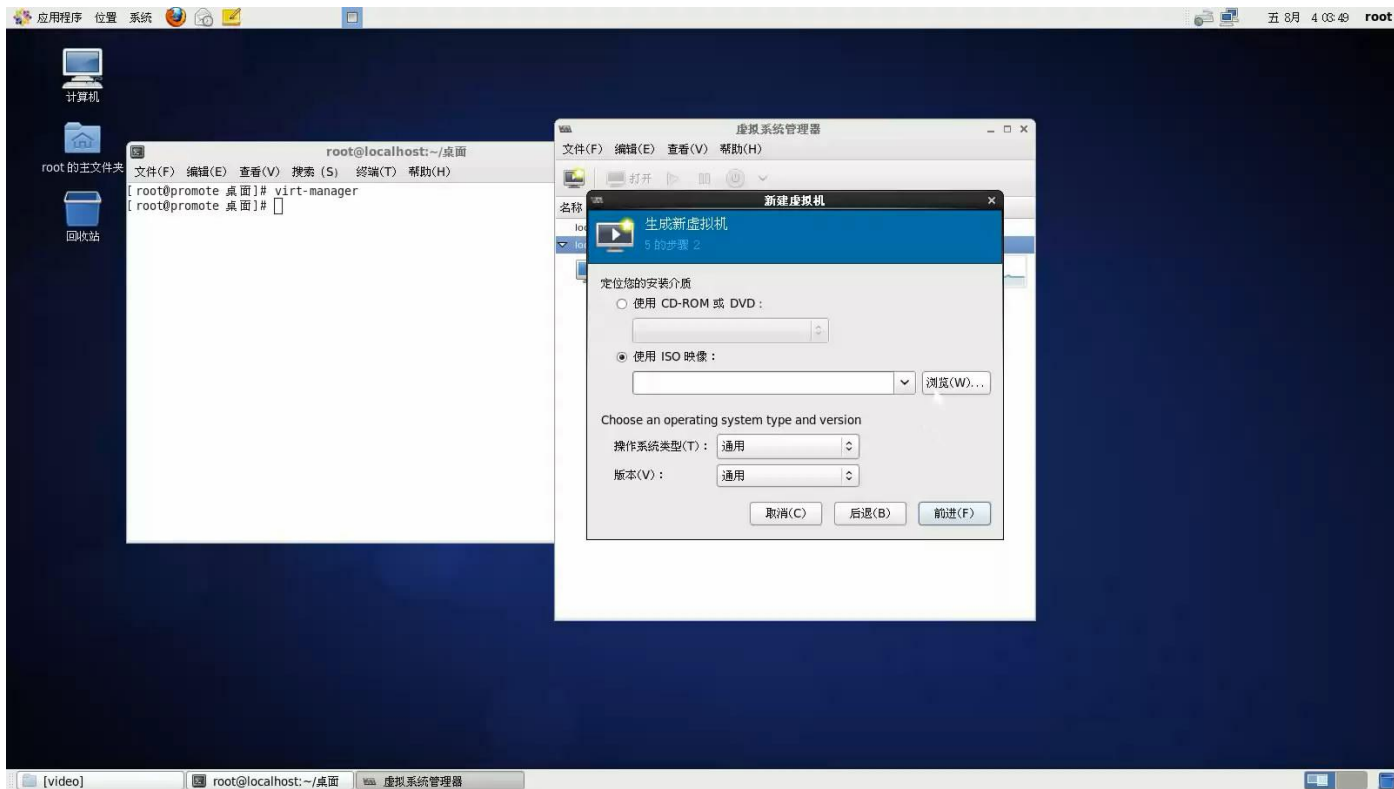
```
#virt-manager
```

3 创建虚拟机



图形化界面:

虚拟机操作系统: CentOS-6.9-x86_64-LiveDVD.iso



4 使用UnixBench进行性能比较



介绍:

unixbench是一个用于测试unix系统性能的工具，也是一个比较通用的benchmark，此测试的目的是对类Unix系统提供一个基本的性能指示，很多测试用于系统性能的不同方面，这些测试的结果是一个指数值（index value，如520），这个值是测试系统的测试结果与一个基线系统测试结果比较得到的指数值，这样比原始值更容易得到参考价值，测试集合里面所有的测试得到的指数值结合起来得到整个系统的指数值。

各项的测试有得分，然后有一个综合的得分。通常情况下1000分以上的VPS是性能较好的。

4 使用UnixBench进行性能比较



安装:

- 解压下载好的的unixbench-5.1.2.tar.gz:
 #tar -xzf unixbench-5.1.2.tar.gz
 #cd unixbench-5.1.2
- 我们不需要进行图形测试或者不在图形化界面下测试, 则将Makefile文件中GRAPHICS_TEST = defined注释掉
- 编译:
 #make
- 运行程序:
 #./Run

4 使用UnixBench进行性能比较



河海大学
HOHAI UNIVERSITY

结果分析:

```
root@localhost:~# cd /home/bhc; unixbench-5.1.2

File Copy 1024 bufsize 2000 maxblocks 3860.0 274079.1 692.1
File Copy 256 bufsize 500 maxblocks 1655.0 73092.2 441.6
File Copy 4096 bufsize 8000 maxblocks 3200.0 878305.3 1605.6
Pipe Throughput 12460.0 411896.4 351.2
Pipe-based Context Switching 4000.0 60858.3 152.1
Process Creation 126.0 2546.3 252.1
Shell Scripts (1 concurrent) 42.4 3561.3 844.7
Shell Scripts (8 concurrent) 6.0 318.3 1268.0
System Call Overhead 15060.0 412485.6 275.0

System Benchmarks Index Score *****
555.3

-----
Benchmarks Run 2017 04 20 07:01:03.28 - 08:31:33
4 CPUs in system; running 4 parallel copies of tests

Dhrystone 2 using register variables 127116042.3 lps (10.0 s, 7 samples)
Double-Precision Whetstone 15464.9 WMIPS (9.9 s, 7 samples)
Excel Throughput 3708.4 lps (30.0 s, 2 samples)
File Copy 1024 bufsize 2000 maxblocks 613225.5 KBps (30.0 s, 2 samples)
File Copy 256 bufsize 500 maxblocks 157316.7 KBps (30.0 s, 2 samples)
File Copy 4096 bufsize 8000 maxblocks 2086911.0 KBps (30.0 s, 2 samples)
Pipe Throughput 1576476.8 lps (10.0 s, 7 samples)
Pipe-based Context Switching 297071.1 lps (10.0 s, 7 samples)
Process Creation 9476.8 lps (30.0 s, 2 samples)
Shell Scripts (1 concurrent) 9605.1 lps (60.0 s, 2 samples)
Shell Scripts (8 concurrent) 1281.3 lps (60.1 s, 2 samples)
System Call Overhead 1506088.0 lps (10.0 s, 7 samples)

System Benchmarks Index Values
-----
BASELINE RESULT INDEX
Dhrystone 2 using register variables 118700.0 127116042.3 10692.6
Double-Precision Whetstone 65.0 15464.9 2393.6
Excel Throughput 62.0 3708.4 1141.7
File Copy 1024 bufsize 2000 maxblocks 3860.0 613225.5 1559.1
File Copy 256 bufsize 500 maxblocks 1655.0 157316.7 950.1
File Copy 4096 bufsize 8000 maxblocks 3200.0 2086911.0 6529.3
Pipe Throughput 12460.0 1576476.8 1267.3
Pipe-based Context Switching 4000.0 297071.1 742.7
Process Creation 126.0 9476.8 752.1
Shell Scripts (1 concurrent) 42.4 9605.1 2234.2
Shell Scripts (8 concurrent) 6.0 1281.3 2152.2
System Call Overhead 15060.0 1506088.0 1004.1

System Benchmarks Index Score *****
1745.3

You have new mail in /var/spool/mail/root
[root@promote unixbench-5.1.2]#
```

```
root@promote:~# cd /home/unixbench-5.1.2

File Edit View Search Terminal Help
CPU 0: Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz (6385.7 bogomips)
Hyper-Threading: 480-64, MMX, Physical Address Ext., SYSCALL/SXSEXT,
SYSCALL/SXSEXT
16:38:03 up 1 min, 2 users, load average: 0.60, 0.25, 0.09; runlevel 5

Benchmarks Run: Thu Aug 03 2017 16:38:03 - 17:06:00
1 CPU in system; running 1 parallel copy of tests

Dhrystone 2 using register variables 38455726.2 lps (10.0 s, 7 samples)
Double-Precision Whetstone 4167.8 WMIPS (9.9 s, 7 samples)
Excel Throughput 4941.2 lps (30.0 s, 2 samples)
File Copy 1024 bufsize 2000 maxblocks 1043452.8 KBps (30.0 s, 2 samples)
File Copy 256 bufsize 500 maxblocks 274114.3 KBps (30.0 s, 2 samples)
File Copy 4096 bufsize 8000 maxblocks 2826693.2 KBps (30.0 s, 2 samples)
Pipe Throughput 1781922.8 lps (10.0 s, 7 samples)
Pipe-based Context Switching 388667.0 lps (10.0 s, 7 samples)
Process Creation 19272.4 lps (30.0 s, 2 samples)
Shell Scripts (1 concurrent) 5881.4 lps (60.0 s, 2 samples)
Shell Scripts (8 concurrent) 704.1 lps (60.1 s, 2 samples)
System Call Overhead 2746468.8 lps (10.0 s, 7 samples)

System Benchmarks Index Values
-----
BASELINE RESULT INDEX
Dhrystone 2 using register variables 118700.0 38455726.2 3239.3
Double-Precision Whetstone 65.0 4167.8 757.8
Excel Throughput 62.0 4941.2 1149.1
File Copy 1024 bufsize 2000 maxblocks 3860.0 1043452.8 2635.6
File Copy 256 bufsize 500 maxblocks 1655.0 274114.3 1657.5
File Copy 4096 bufsize 8000 maxblocks 3200.0 2826693.2 4063.3
Pipe Throughput 12460.0 1781922.8 1432.4
Pipe-based Context Switching 4000.0 388667.0 951.2
Process Creation 126.0 19272.4 1529.6
Shell Scripts (1 concurrent) 42.4 5881.4 1368.3
Shell Scripts (8 concurrent) 6.0 704.1 1368.3
System Call Overhead 15060.0 2746468.8 1831.0

System Benchmarks Index Score *****
1538.1

[root@promote unixbench-5.1.2]#
```

Dom0

DomU

4 使用UnixBench进行性能比较



结果分析:

项目	Dom0 INDEX	DomU INDEX	性能损耗
System Benchmarks Index Score	1748.8	1638.1	6.3%

4 使用UnixBench进行性能比较



测试项目	项目说明	基准线	Dom0 INDEX	DomU INDEX	性能损耗(正为损耗百分比)
Dhrystone 2 using register variables	测试 string handling	116700.0lps	10892. 5	3295. 3	0. 697471
Double-Precision Whetstone	测试浮点数操作的速度和效率	55.0MWIPS	1341. 7	1149. 1	0. 143549
Execl Throughput	此测试考察每秒钟可以执行的 execl 系统调用的次数	43.0lps	1555. 1	2635	-0. 69442
File Copy 1024 bufsize 2000 maxblocks	测试从一个文件向另外一个文件传输数据的速率	3960.0KBps	950. 5	1657. 5	-0. 74382
File Copy 256 bufsize 500 maxblocks	测试从一个文件向另外一个文件传输数据的速率。	1655.0KBps	3620. 5	4863. 3	-0. 34327
File Read 4096 bufsize 8000 maxblocks	测试从一个文件向另外一个文件传输数据的速率。	5800.0KBps	1267. 3	1432. 4	-0. 13028
Pipe-based Context Switching	测试两个进程（每秒钟）通过一个管道交换一个不断增长的整数的次数	12440.0lps	742. 7	951. 2	-0. 28073
Pipe Throughput	一秒钟内一个进程可以向一个管道写 512 字节数据然后再读回的次数	4000.0lps	752. 1	1529. 6	-1. 03377
Process Creation	测试每秒钟一个进程可以创建子进程然后收回子进程的次数（子进程一定立即退出）。	126.0lps	2284. 2	1368. 3	0. 400972
Shell Scripts (8 concurrent)	测试一秒钟内一个进程可以并发地开始一个shell 脚本的 n 个拷贝的次数，n 一般取值1, 2, 4, 8.	42.4lpm	2152. 2	1173. 6	0. 454698
System Call Overhead	测试进入和离开操作系统内核的代价，即一次系统调用的代价。	6.0lpm	1004. 1	1831	-0. 82352



河海大学
HOHAI UNIVERSITY

云计算技术与应用

谢谢大家！