

## 云主机性能测试参考

## 目录

1. 测试环境.....	1
2. 计算能力.....	1
2.1. 测试工具.....	1
2.1.1. UnixBench 跑分基准测试工具.....	1
2.2. 测试方法.....	1
2.3. 测试结果.....	2
3. 存储 IO 能力.....	6
3.1. 测试工具.....	6
3.2. 测试方法.....	6
3.2.1. DD 测试.....	6
3.2.2. FIO 测试.....	6
3.3. 测试结果.....	9
4. 网络 I/O 能力.....	11
4.1. 测试工具.....	11
4.2. 测试方法.....	11
4.2.1. PING 值测试.....	11
4.2.2. Netperf.....	11
4.3. 测试结果.....	14
4.3.1. 访问速度 PING 值测试.....	14
4.3.2. Netperf.....	15

# 1. 测试环境

云主机配置：2 核/4G/ 500G 云硬盘

操作系统：CentOS6.x 64bit

网络接入：被测云主机接入带宽为 5M/bps

测试工具：要求版本一样

# 2. 计算能力

## 2.1. 测试工具

### 2.1.1. UnixBench 跑分基准测试工具

UnixBench 是一个 unix 类（Unix, BSD, Linux）系统下的性能跑分基准测试开源工具，被广泛用于测试 linux 系统主机的性能。Unixbench 的主要测试项目有：系统调用、读写、进程、图形化测试、2D、3D、管道、运算、C 库等系统基准性能提供测试数据。

## 2.2. 测试方法

### 1) 新建 UnixBench.sh 脚本

本测试使用 unixbench.sh 脚本安装 UnixBench5.1.3, UnixBench5.1.3 包含 system 和 graphic 测试，该脚本注释了关于 graphic 的测试项（大多数云主机都是没有显卡或者是集显，所以图像性能无需测试）。

如果你需要测试 graphic，则需要修改 unixbench.sh，注释掉 “sed -i "s/GRAPHIC\_TESTS = defined/#GRAPHIC\_TESTS = defined/g" ./Makefile”，同时需要系统提供 x11perf 命令 gl\_glibs 库。

```
$ vi unixbench.sh

#!/bin/bash

PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:~/bin

export PATH

# Create new soft download dir

mkdir -p /opt/unixbench;
```



```
cd /opt/unixbench;

cur_dir=`pwd`

# Download UnixBench5.1.3
if [ -s UnixBench5.1.3.tgz ]; then
    echo "UnixBench5.1.3.tgz [found]"
else
    echo "UnixBench5.1.3.tgz not found!!!download now....."
    if ! wget -c http://lamp.teddysun.com/files/UnixBench5.1.3.tgz;then
        echo "Failed to download UnixBench5.1.3.tgz,please download it to "${cur_dir}"
        directory manually and try again."
        exit 1
    fi
fi

tar -xzf UnixBench5.1.3.tgz;
cd UnixBench;

yum -y install gcc gcc-c autoconf gcc-c++ time perl-Time-HiRes

#Run unixbench
sed -i "s/GRAPHIC_TESTS = defined/#GRAPHIC_TESTS = defined/g" ./Makefile
make;

./Run -c 1 -c 4;
```

- 2) 执行脚本，会自动安装好 UnixBench5.1.3 并开始运行测试，运行 30 分钟左右得出分数，分数越高，性能越好。

```
$ chmod +x unixbench.sh
$ ./unixbench.sh
```

## 2.3. 测试结果

采用 unix 系统跑分基准测试工具 Unixbench 做基本的测试。测试云服务器的配置为 2 核



4G, 单进程测试 ( 1 parallel copy of tests ) 的结果是 1338.8, 多进程测试 ( 4parallel copy of tests ) 的结果是 2458.5。

```
=====
BYTE UNIX Benchmarks (Version 5.1.3)

System: host-192-168-1-147: GNU/Linux
OS: GNU/Linux -- 2.6.32-431.el6.x86_64 -- #1 SMP Fri Nov 22 03:15:09 UTC 2013
Machine: x86_64 (x86_64)
Language: en_US.utf8 (charmap="UTF-8", collate="UTF-8")
CPU 0: Intel Xeon E312xx (Sandy Bridge) (4400.0 bogomips)
      x86-64, MMX, Physical Address Ext, SYSENTER/SYSEXIT, SYSCALL/SYSRET
CPU 1: Intel Xeon E312xx (Sandy Bridge) (4400.0 bogomips)
      x86-64, MMX, Physical Address Ext, SYSENTER/SYSEXIT, SYSCALL/SYSRET
00:16:22 up 42 min,  2 users,  load average: 0.47, 0.11, 0.03; runlevel 3

-----

Benchmark Run: Wed Jul 22 2015 00:16:22 - 00:44:33
2 CPUs in system; running 1 parallel copy of tests


Dhrystone 2 using register variables      28300652.2 lps   (10.0 s, 7 samples)
Double-Precision Whetstone                3075.9 MWIPS (9.8 s, 7 samples)
Execl Throughput                          3326.7 lps   (29.9 s, 2 samples)
File Copy 1024 bufsize 2000 maxblocks    832947.6 KBps  (30.0 s, 2 samples)
File Copy 256 bufsize 500 maxblocks      241414.3 KBps  (30.0 s, 2 samples)
File Copy 4096 bufsize 8000 maxblocks    2299407.3 KBps  (30.0 s, 2 samples)
Pipe Throughput                           1618215.7 lps   (10.0 s, 7 samples)
Pipe-based Context Switching              292853.2 lps   (10.0 s, 7 samples)
Process Creation                          9220.4 lps   (30.0 s, 2 samples)
Shell Scripts (1 concurrent)              5673.0 lpm    (60.0 s, 2 samples)
Shell Scripts (8 concurrent)              1259.6 lpm    (60.0 s, 2 samples)
System Call Overhead                      1990195.4 lps   (10.0 s, 7 samples)


System Benchmarks Index Values          BASELINE    RESULT    INDEX
Dhrystone 2 using register variables    116700.0   28300652.2   2425.1
```



Double-Precision Whetstone	55.0	3075.9	559.2
Execl Throughput	43.0	3326.7	773.7
File Copy 1024 bufsize 2000 maxblocks	3960.0	832947.6	2103.4
File Copy 256 bufsize 500 maxblocks	1655.0	241414.3	1458.7
File Copy 4096 bufsize 8000 maxblocks	5800.0	2299407.3	3964.5
Pipe Throughput	12440.0	1618215.7	1300.8
Pipe-based Context Switching	4000.0	292853.2	732.1
Process Creation	126.0	9220.4	731.8
Shell Scripts (1 concurrent)	42.4	5673.0	1338.0
Shell Scripts (8 concurrent)	6.0	1259.6	2099.4
System Call Overhead	15000.0	1990195.4	1326.8
			=====
System Benchmarks Index Score			1338.8

Benchmark Run: Wed Jul 22 2015 00:44:33 - 01:12:46

2 CPUs in system; running 4 parallel copies of tests

Dhrystone 2 using register variables	57916393.7 lps	(10.0 s, 7 samples)
Double-Precision Whetstone	12333.8 MWIPS	(9.9 s, 7 samples)
Execl Throughput	8709.4 lps	(29.7 s, 2 samples)
File Copy 1024 bufsize 2000 maxblocks	1067976.5 KBps	(30.0 s, 2 samples)
File Copy 256 bufsize 500 maxblocks	285098.9 KBps	(30.0 s, 2 samples)
File Copy 4096 bufsize 8000 maxblocks	3225525.6 KBps	(30.0 s, 2 samples)
Pipe Throughput	3327665.2 lps	(10.0 s, 7 samples)
Pipe-based Context Switching	604221.7 lps	(10.0 s, 7 samples)
Process Creation	22253.2 lps	(30.0 s, 2 samples)
Shell Scripts (1 concurrent)	10234.2 lpm	(60.0 s, 2 samples)
Shell Scripts (8 concurrent)	1365.0 lpm	(60.2 s, 2 samples)
System Call Overhead	3204334.1 lps	(10.0 s, 7 samples)

System Benchmarks Index Values	BASLINE	RESULT	INDEX
Dhrystone 2 using register variables	116700.0	57916393.7	4962.8
Double-Precision Whetstone	55.0	12333.8	2242.5
Execl Throughput	43.0	8709.4	2025.4



File Copy 1024 bufsize 2000 maxblocks	3960.0	1067976.5	2696.9
File Copy 256 bufsize 500 maxblocks	1655.0	285098.9	1722.7
File Copy 4096 bufsize 8000 maxblocks	5800.0	3225525.6	5561.3
Pipe Throughput	12440.0	3327665.2	2675.0
Pipe-based Context Switching	4000.0	604221.7	1510.6
Process Creation	126.0	22253.2	1766.1
Shell Scripts (1 concurrent)	42.4	10234.2	2413.7
Shell Scripts (8 concurrent)	6.0	1365.0	2275.0
System Call Overhead	15000.0	3204334.1	2136.2
			=====
System Benchmarks Index Score			2458.5

## 3. 存储 IO 能力

### 3.1. 测试工具

DD 命令行工具

FIO 是一个用来对硬件进行压力测试和验证 I/O 的工具，支持 13 种不同的 I/O 引擎，包括：sync, mmap, libaio, posixaio, SG v3, splice, null, network, syslet, guasi, solarisaio 等等。

### 3.2. 测试方法

#### 3.2.1. DD 测试

DD 顺序写

```
dd if=/dev/zero of=1.img bs=4096 count=2M conv=fdatasync
```

DD 顺序读

```
dd of=/dev/null if=1.img bs=4096
```

#### 3.2.2. FIO 测试

宝德云主机系统镜像已安装 FIO 工具，可以直接使用。

##### 3.2.2.1. FIO 测试系统盘

- 1) 在 /root 目录下新建测试目录 datadisk,
- 2) 创建 fio\_sys.conf 配置文件。(该配置文件的测试路径已指定为步骤 1 所新建的目录: /root/datadisk, 若测试其他路径请修改 directory 的值)

```
$ cat >> fio_sys.conf <<EOF
[global]
ioengine=libaio
direct=1
thread=1
norandommap=1
randrepeat=0
runtime=60
```



```
ramp_time=6  
size=512m  
directory=/root/datadisk
```

```
[read4k-rand]  
stonewall  
group_reporting  
bs=4k  
rw=randread  
numjobs=8  
iodepth=32
```

```
[read64k-seq]  
stonewall  
group_reporting  
bs=64k  
rw=read  
numjobs=4  
iodepth=8
```

```
[write4k-rand]  
stonewall  
group_reporting  
bs=4k  
rw=randwrite  
numjobs=2  
iodepth=4
```

```
[write64k-seq]  
stonewall  
group_reporting  
bs=64k  
rw=write  
numjobs=2  
iodepth=4
```



```
EOF
```

### 3) 执行测试

```
$ fio fio_sys.conf
```

#### 3.2.2.2.FIO 测试云硬盘

- 1) 挂载云硬盘到/home 目录后，在/home 下新建测试目录 datadisk，
- 2) 创建 fio\_disk.conf 配置文件。(该配置文件的测试路径已指定为步骤 1 所新建的目录：  
/home/datadisk，若测试其他路径请修改 directory 的值)

```
$ cat >> fio_disk.conf <<EOF
```

```
[global]
```

```
ioengine=libaio
```

```
direct=1
```

```
thread=1
```

```
norandommap=1
```

```
randrepeat=0
```

```
runtime=60
```

```
ramp_time=6
```

```
size=512m
```

```
directory=/home/datadisk
```

```
[read4k-rand]
```

```
stonewall
```

```
group_reporting
```

```
bs=4k
```

```
rw=randread
```

```
numjobs=8
```

```
iodepth=32
```

```
[read64k-seq]
```

```
stonewall
```

```
group_reporting
```

```
bs=64k
```

```
rw=read
```



```

numjobs=4
iodepth=8

[write4k-rand]
stonewall
group_reporting
bs=4k
rw=randwrite
numjobs=2
iodepth=4

[write64k-seq]
stonewall
group_reporting
bs=64k
rw=write
numjobs=2
iodepth=4
EOF
    
```

### 3) 执行测试

```
$ fio fio_disk.conf
```

## 3.3. 测试结果

云主机配置	DD		IOPS			
	顺序读 (MB/s)	顺序写 (MB/s)	4K 随机读	4K 随机写	64K 顺序读	64K 顺序写
2 核/4G/ 系统盘	261MB/s	307MB/s	1152	1543	2573	3663
2 核 /4G/500G 云硬盘	76.1MB/s	97.3MB/s	16712	1093	4141	895



## 4. 网络 I/O 能力

### 4.1. 测试工具

#### PING 值测试

看服务器的基本返回数据和响应时间。响应速率越小，速度越好。

#### Netperf

Netperf 工具以 client/server 方式工作。server 端是 netserver，用来侦听来自 client 端的连接，client 端是 netperf，用来向 server 发起网络测试。在 client 与 server 之间，首先建立一个控制连接，传递有关测试配置的信息，以及测试的结果；在控制连接建立并传递了测试配置信息以后，client 与 server 之间会再建立一个测试连接，用来来回传递着特殊的流量模式，以测试网络的性能。

### 4.2. 测试方法

#### 4.2.1. PING 值测试

在线 ping 小工具 <http://ping.chinaz.com/>，通过该工具可以多个地点 Ping 服务器以检测服务器响应速度,同时也可以测试网站的响应速度,解析时间,

#### 4.2.2. Netperf

Netperf 测试必须关闭所有的防火墙：

- 关闭系统防火墙
- 在云平台中的防火墙规则中添加 ALL TCP、ALL UDP 进口和出口规则

客户端和服务端都必须安装 Netperf

```
$ wget -c ftp://ftp.netperf.org/netperf/netperf-2.7.0.tar.gz
$ tar -xvf netperf-2.7.0.tar.gz
$ cd netperf-2.7.0
$ ./configure
$ make
$ make install
```



netperf 常用的全局参数有：

- -h help
- -H 指定 server 端 IP
- -t 指定测试类型，包括 TCP\_STREAM、TCP\_RR、UDP\_STREAM、UDP\_RR4 种。
- -l 指定测试时间，单位是秒。

服务器端执行：（ip: 58.67.219.89）

```
[root@netserver netperf-2.7.0]# netserver
Starting netserver with host 'IN(6)ADDR_ANY' port '12865' and family AF_UNSPEC
[root@test-1 netperf-2.7.0]#
```

客户端运行

TCP\_STREAM

```
[root@netclient netperf-2.7.0]# netperf -H 58.67.219.89 -l 20 -t TCP_STREAM
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 58.67.219.89
() port 0 AF_INET
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time  Throughput
bytes bytes bytes secs.  10^6bits/sec

87380 16384 16384 20.71 4.95
```

测试结果表明，在路由器总带宽为 5Mbps 的环境下，云主机的 TCP 带宽为 4.95Mbps。

TCP\_RR

```
[root@netclient netperf-2.7.0]# netperf -H 58.67.219.89 -l 20 -t TCP_RR
MIGRATED TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to
58.67.219.89 () port 0 AF_INET : first burst 0
Local /Remote
Socket Size Request Resp. Elapsed Trans.
Send Recv Size Size Time Rate
```



```
bytes  Bytes  bytes  bytes  secs.  per sec
16384  87380  1      1      20.00  1534.33
16384  87380
```

测试结果中，第一行是本地统计结果，第二行是远端统计结果，Request 和 Response 报文大小都是 1 个字节，C/S 两端每秒的 transaction 次数是 1534.33。

#### UDP\_STREAM

```
[root@netclient netperf-2.7.0]# netperf -t UDP_STREAM -H 58.67.219.89 -- -m 1024 -R 1
MIGRATED UDP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 58.67.219.89
() port 0 AF_INET

Socket  Message  Elapsed      Messages
Size    Size      Time          Okay Errors  Throughput
bytes   bytes    secs          #      #    10^6bits/sec

124928  1024    10.00      719117      0    589.07
124928             10.00      6070             4.97
```

第一行结果是本地的发送统计，也就是本地发送 UDP 的吞吐量为 589.07Mbps，第二行是 server 端的接收统计，吞吐量为 4.97Mbps。

#### UDP\_RR

```
[root@netclient netperf-2.7.0]# netperf -H 58.67.219.89 -l 20 -t UDP_RR -l 20 -- -R 1
MIGRATED UDP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to
58.67.219.89 () port 0 AF_INET : first burst 0

Local /Remote

Socket Size  Request  Resp.  Elapsed  Trans.
Send  Recv  Size    Size    Time    Rate
bytes Bytes bytes   bytes  secs.  per sec

124928 124928 1      1      20.00  1610.79
124928 124928
```



测试结果中，第一二行分别是本地和远端统计结果，Request 和 Response 报文大小都是 1 个字节，C/S 两端每秒的 transaction 次数是 1610.79。

## 4.3. 测试结果

### 4.3.1. 访问速度 PING 值测试

监测点	响应IP	响应时间	TTL
江西[电信]	<a href="#">58.67.202.194</a>	37毫秒	116
贵州[电信]	<a href="#">58.67.202.194</a>	48毫秒	114
湖北[电信]	<a href="#">58.67.202.194</a>	35毫秒	117
陕西西安[电信]	<a href="#">58.67.202.194</a>	36毫秒	116
云南昆明[电信]	<a href="#">58.67.202.194</a>	42毫秒	116
浙江[电信]	<a href="#">58.67.202.194</a>	31毫秒	114
香港[电信]	<a href="#">58.67.202.194</a>	10毫秒	117
四川成都[电信]	<a href="#">58.67.202.194</a>	44毫秒	114
广东东莞[电信]	<a href="#">58.67.202.194</a>	4毫秒	116
安徽淮北[电信]	<a href="#">58.67.202.194</a>	42毫秒	116
秦皇岛[电信]	<a href="#">58.67.202.194</a>	50毫秒	115
深圳[电信]	<a href="#">58.67.202.194</a>	6毫秒	115
河南[电信]	<a href="#">58.67.202.194</a>	39毫秒	114
江苏扬州[电信]	<a href="#">58.67.202.194</a>	28毫秒	114
湖南[电信]	<a href="#">58.67.202.194</a>	21毫秒	116
福州[电信]	<a href="#">58.67.202.194</a>	23毫秒	116
上海[电信]	<a href="#">58.67.202.194</a>	28毫秒	116
郑州[多线]	<a href="#">58.67.202.194</a>	45毫秒	114
上海[多线]	<a href="#">58.67.202.194</a>	30毫秒	113
安徽[多线]	<a href="#">58.67.202.194</a>	33毫秒	115
四川[多线]	<a href="#">58.67.202.194</a>	47毫秒	116



监测点	响应IP	响应时间	TTL
香港[多线]	<a href="#">58.67.202.194</a>	10毫秒	111
江西[多线]	<a href="#">58.67.202.194</a>	18毫秒	114
浙江[多线]	<a href="#">58.67.202.194</a>	26毫秒	115
北京[多线]	<a href="#">58.67.202.194</a>	45毫秒	109
洛阳[多线]	<a href="#">58.67.202.194</a>	43毫秒	114
广东[多线]	<a href="#">58.67.202.194</a>	10毫秒	117
江苏[多线]	<a href="#">58.67.202.194</a>	38毫秒	113
安徽滁州[联通]	<a href="#">58.67.202.194</a>	53毫秒	115
河北[联通]	<a href="#">58.67.202.194</a>	53毫秒	112
泉州[联通]	<a href="#">58.67.202.194</a>	49毫秒	116
山西[联通]	<a href="#">58.67.202.194</a>	50毫秒	116
北京[联通]	<a href="#">58.67.202.194</a>	43毫秒	112
天津[联通]	<a href="#">58.67.202.194</a>	44毫秒	109
江苏[联通]	<a href="#">58.67.202.194</a>	46毫秒	116
辽宁[联通]	<a href="#">58.67.202.194</a>	59毫秒	117
河南[联通]	<a href="#">58.67.202.194</a>	40毫秒	115
徐州[联通]	<a href="#">58.67.202.194</a>	42毫秒	114
郑州[联通]	<a href="#">58.67.202.194</a>	42毫秒	115
山东[移动]	<a href="#">58.67.202.194</a>	47毫秒	114
深圳[移动]	<a href="#">58.67.202.194</a>	8毫秒	113

线路	最快节点	最慢节点	平均响应
所有线路	广东东莞[电信] 4毫秒	辽宁[联通] 117毫秒	35毫秒
电信	广东东莞[电信] 4毫秒	秦皇岛[电信] 50毫秒	30毫秒
多线	广东[多线] 10毫秒	四川[多线] 116毫秒	31毫秒
联通	河南[联通] 40毫秒	辽宁[联通] 117毫秒	47毫秒
移动	深圳[移动] 8毫秒	山东[移动] 47毫秒	27毫秒

在全国范围内的电信、移动、联通线路的PING值都比较优秀，且比较均匀。

### 4.3.2. Netperf

在路由器总带宽为 5Mbps 的环境下：

云主机的 TCP 带宽为 4.95Mbps。C/S 两端每秒的 transaction 次数是 1534.33。

本地发送 UDP 的吞吐量为 589.07Mbps，server 端的接收统计的吞吐量为

4.97Mbps，C/S 两端每秒的 transaction 次数是 1610.79。