

l0meter 测试指导手册

目录

1 IOmeter 介绍	1
1.1 组成部分	1
1.2 版本下载	2
2 IOmeter 安装	3
2.1 Windows 下的安装	3
2.2 Linux 下的安装	3
3 IOmeter 使用	4
3.1 Windows 下的界面及参数介绍	4
3.1.1 在 Windows 主机打开 IOmeter	5
3.1.2 Disk Target 界面及参数介绍	6
3.1.3 Network Targets 界面及参数介绍	7
3.1.4 Access Specifications 界面及参数介绍	7
3.1.5 Test Setup 界面	9
3.1.6 Result Display 界面	10
3.2 Windows 下单机跑 IOmeter	11
3.3 Windows 下用 IOmeter 跑网络压力	11
3.4 Linux 下的使用	13
4 自动化脚本	14
4.1 根据帮助文档查看 IOmeter 参数	14
4.2 自动化测试脚本	15
5 性能指标	16

1 IOmeter 介绍

IOmeteter 是一个工作在单系统和集群系统上用来衡量和描述 I/O 子系统的工具。

IOmeteter 既是工作负载生成器（也就是说，它可以进行输入输出操作，以便增加系统的负荷），它还是一个测量工具（也就是说，它检查并且记录 I/O 操作的性能和对系统的影响）。它可以被配置为模拟任何程序或者基准测试程序的磁盘和网络 I/O 的负载，或者用来产生整个综合的 I/O 负载。它也可以用来产生并测量单系统或者多系统（网络）的负载。

IOmeteter 可以被用来测量和描述：

- 磁盘和网络控制器的性能
- 总线的带宽和时延容量
- 对于附带驱动器的网络吞吐量
- 共享总线的性能
- 系统级别的硬件驱动的性能
- 系统级别的网络性能

1.1 组成部分

IOmeteter 包含了两个程序，IOmeteter 和 Dynamo.

➤ IOmeteter 是控制程序。使用图形用户接口（GUI），你可以配置负载，设置操作参数，启动和停止测试。IOmeteter 告诉Dynamo 去做什么，搜集分析数据，将分析数据输出到文件中。在某一时刻，只能有一个IOmeteter副本运行；典型的情况是运行在服务器上。

➤ Dynamo 是负载生成器。它没有用户界面。当接收到IOmeteter 发送过来的命令，Dynamo执行相应的I/O 操作并且记录性能信息，然后将数据返回给IOmeteter。它可以有多个副本同时运行；典型的情况是服务器上运行一个副本，每个客户端运行一个副本。Dynamo 是多线程的；每一个副本都可以模拟多客户程序的工作负载。Dynamo中的每一个运行的副本称为一个管理者（Manager）；副本中的一个线程称为工作者（Worker）。

1.2 版本下载

IOmeteter 是一个开源工具，可以到 <http://www.iometer.org> 下载，最新版本是 1.1.0。因更新内容很少，最常使用的版本仍为 2006.07.27。

下面我们就以 iometer-2006.07.27.win32.i386-setup.exe 为例进行介绍。

2 IOmeter 安装

2.1 Windows 下的安装

Windows 下的安装文件为 iometer-2006.07.27.win32.i386-setup.exe，安装过程如下：

- 双击打开安装文件，点击 next 和同意协议->选择加入 MS Access 导入导航->选择安装目录->安装完成。

2.2 Linux 下的安装

Linux 下的安装文件为 iometer-2006_07_27.linux.i386-bin.gz，解压完成后就可以用了：

- 解压该文件：

```
tar xzvf iometer-2006_07_27.linux.i386-bin.gz
```

注：Linux 下只有 dynamo 这个 agent，要想控制还得要在 Windows 的一台主机上安装 IOmeteter 的 console。

3 IOmeter 使用

3.1 Windows 下的界面及参数介绍



打开 IOmeter，上面按钮的作用从左到右依次是：



打开已有的测试配置文件；



保存现在的测试配置文件；



新增一个是负载生成器（Dynamo 支持多线程，一般情况下不需要在同一台主机上运行多个 Dynamo）；



新增一个 Disk Worker（后面会介绍通过修改 OutStanding I/Os 数来增大压力，我在跑 IOmeter 的时候，每台主机只保留一个 Disk Worker）；



在选定的主机下，点击该按钮可以增加一个 Network Worker（一般情况下，我们）；



选中某一个 Disk Worker 或 Network Worker，让后点击该按钮，会复制出一个完全一样的 Worker；



开始测试；



停止当前测试并保存结果；



中断所有测试；



将 Workers 复位 (reset 掉当前的配置, 相当于把 IOmeter 关了再重新打开);



可以方便地删除 Topology 窗体下的 Manager、Worker;




退出软件;

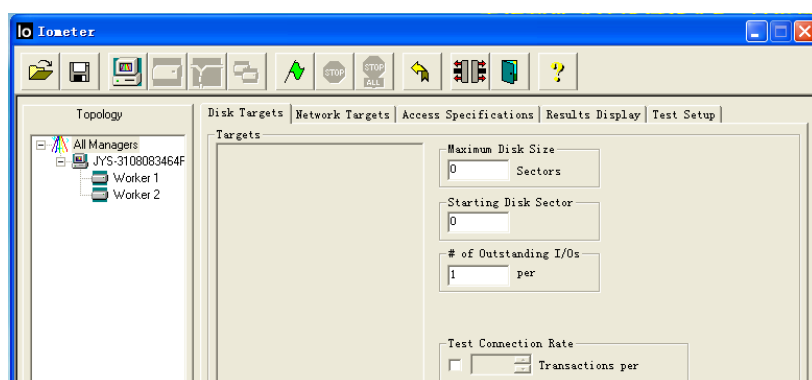


关于 IOmeter, 显示版本和开源许可信息。


下面以在 Windows 下跑 IOmeter 的实力来说明各个参数的作用

3.1.1 在 Windows 主机打开 IOmeter

Console 如下图所示, 默认会在 Topology 里面出现本机以及几个 Disk Worker (软件会根据主机 CPU 个数来决定增加几个 Disk Worker), 在这里, 点选中 Worker2, 然后点击 , 只保留一个 Worker。



大家还会注意到, 运行该工具的同时, 任务栏多了一个 Dos 窗口, 没错, 根据第一章介绍的, 这个 Dos 窗口才是真正的负载生成器, 而上面的图形窗口只是一个控制台。

可以做一个实验, 点击  增加一个 Manager, 我们发现在 Topology 里面多了一台主机, 任务栏也多了一个负载生成器的 Dos 窗口。

```

C:\Program Files\Iometer.org\Iometer 2006.07.27\dynamo /p 1066
Number of processors: 2
Processor speed: 2 GHz.
Number of processors: 2
Processor speed: 2 GHz.

Sending login request...
JYS-3108083464R
10.46.111.119 <port 2810>
Successful PortTCP::Connect
- port name: jys-3108083464r

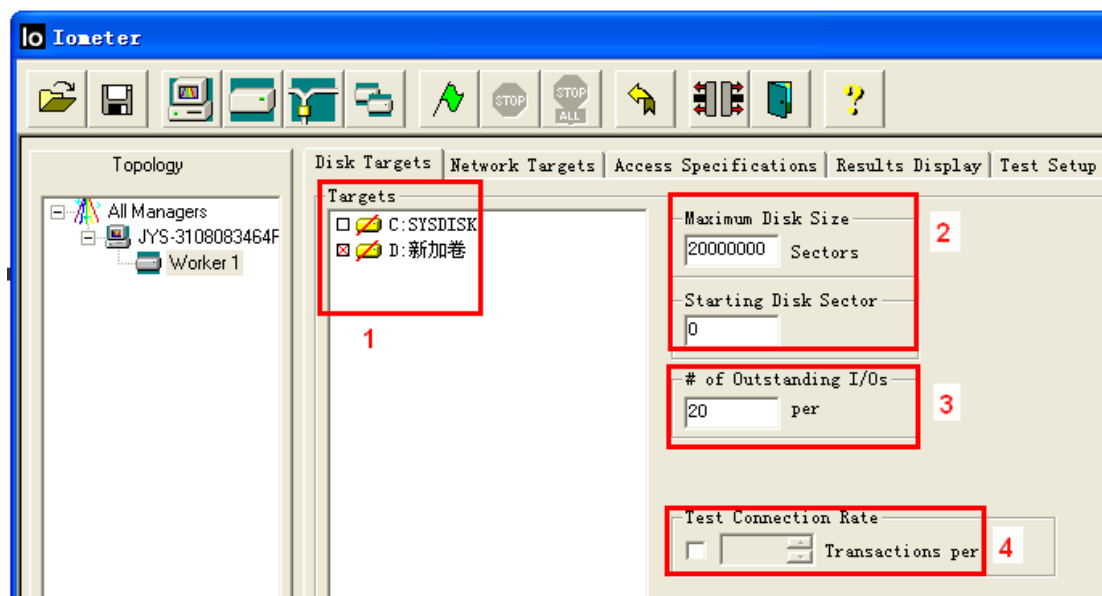
*** If dynamo and iometer hangs here, please make sure
*** you use a correct -n <manager_computer_name> that
*** can ping from iometer machine. use IP if need.
Login accepted.
Reporting drive information...
Physical disk '\\.\PHYSICALDRIVE0' contains partition information.
done.
Reporting TCP network information...
done.

```

3.1.2 Disk Target 界面及参数介绍

如下图，框 1 中指的是 Disk Worker 1 会在用户盘 D 上进行 IO 测试；

框 2 中 “Maximum Disk Size” 单位是 Sectors(扇区)，意思是说 Worker 1 会在 D 盘占用多少个扇区进行压力测试。默认填 0，会将 D 盘的剩余空间全用完。(Windows 系统每个扇区的大小是 512Byte，所以这里填 20000000 就是指 Worker 1 会占用 D 盘 10GB 的空间，另外在做 FCSAN/IPSAN 磁盘测试时设置的大小推荐大于 SAN 设备缓存的 2 倍，太小的话 Iometer 只跑在缓存上，跑出来的性能会偏高不准确。但是从里一个方面来说，如果我们想让最终的结果看起来好看，那么我们可以将该参数设置的小一点，根据测试目的自己把握)；“Starting Disk Sector” 即 Worker1 从哪个扇区开始写它的 iobw.tst 测试文件，这里保持默认的 0 即可。.

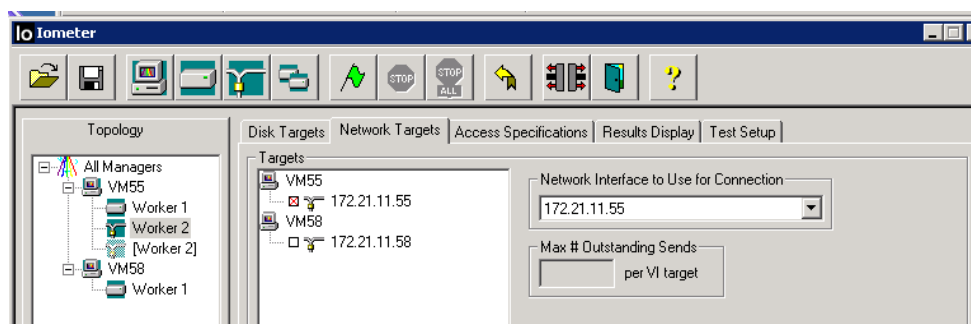


框 3 中 “# of Outstanding I/Os” 指的是 Worker 1 在 D 盘上同时会开多少个异步的 IO 操作，在主机 CPU、内存能力够强时，并发数越多最终跑出来的结果会越准确，默认的是 1 个。具体设置为多少比较好，我们可以实际试一下：先跑 1 个，看看 Result 是多少，5 个时 Result 是多少，10 个、20 个、50 个、100 个、120 个…… 在 CPU 内存承受范围内，找到一个最合适的值。

框 4 中 “Test Connection Rate” 指的是 Worker 1 以什么样的操作频率打开、关闭 D 盘。默认不勾选的意思是，所有的连接都是 open 状态，直到测试停止。这里我们保持默认即可。

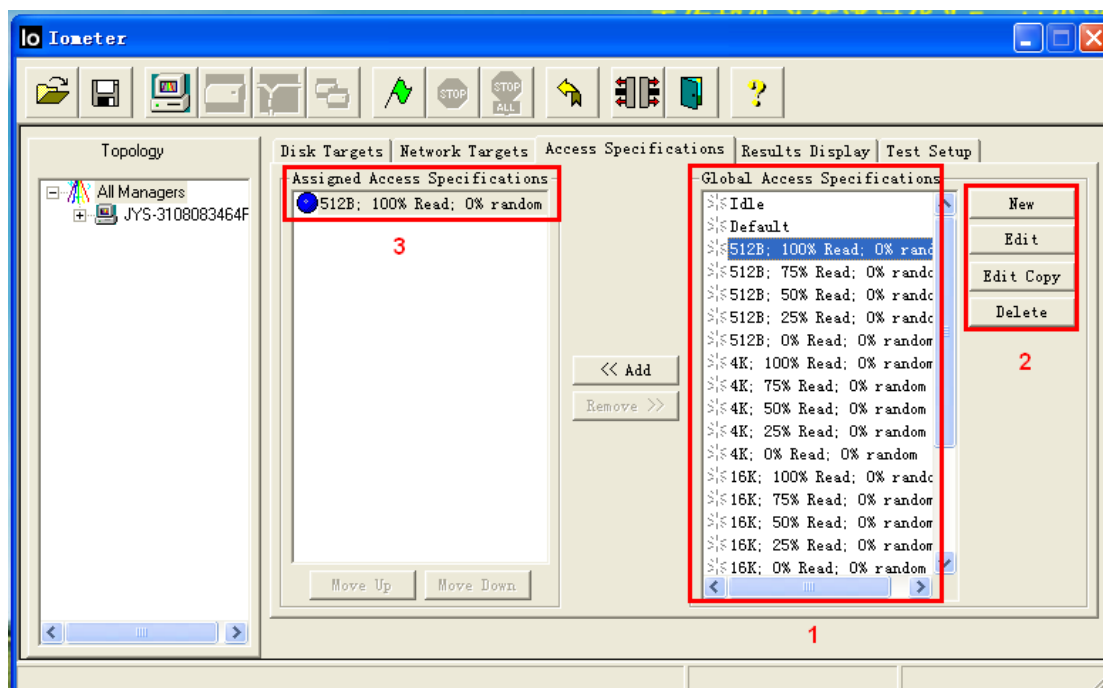
3.1.3 Network Targets 界面及参数介绍

下图是在用 IOMeter 跑网络压力时截的图，这里需要两台主机对跑网络，本例中不涉及，先不去管它，后面章节会详细讲到。



3.1.4 Access Specifications 界面及参数介绍

Access Specification 是 IOMeter 工具根据什么样的规格来跑 IO 测试。这里我们每次只添加一个规格，如下图。（如果添加多个规格的话，IOMeter 会按照从上到下的顺序依次跑每个规格，直到测试停止，用来模拟复杂场景）



如上图，Access Specification，框 1 中是已经定义好的规格。框 2 中可以对 Global Access Specification

做相应的新建、编辑、复制并编辑、删除操作。

框 3 中是已经添加的 Worker1 里面的线程需要在 Target Disk 里面跑的压力的规格。一般我们只跑一个就好了。当然我们也可以 Add 多个规格来模拟复杂的应用，假设我们 Add 了 4 个规格，IOmeter 会依次从上到下的顺序来执行相应的读写操作。

规格的具体参数是什么，又有什么意义和影响呢？我们用下面的截图说明，新建一个规格，并分析里面的参数：

框 1 “Name”，需要给新建的规格定义一个易懂的名字，比如用参数，或者 “Web server workload” 等形象的名字；

框 2 “Default Assignment”，我们保持默认的 none 就好了（这一项的意思是，例如你设置某个规格的 “Default Assignment” 为 Disk Workers 就会发现,每当你新增一个 Disk Worker 时，这个规格会被默认添加到该 Disk Worker 的 “Assigned Access Specifications” 里面；同理我们把某个规格的 “Default Assignment” 设置为 All Workers，那么每当我们新增加一个 Worker 时，这个规格都会被默认加到该 Worker 的 “Assigned Access Specifications” 里面。该选项对测试结果没有任何影响，可以不设置，保持默认。）

框 3，可以看到该规格的配置信息，这里我们只跑一个规格，所以不用管它。

框 4，每次 IO 的大小，这个值越小则跑出来的结果里面的 IOPS 就会越大，带宽就会越小。相反，这个值越大，跑出来的结果里面的 IOPS 就会越小，带宽就会越大。只能填写 512Byte 的整数倍，否则跑的时候会报错。

框 5, Percent of Access Specification (该规格在所有规格中占用的连接百分比), 由于这里我们只跑一个规格, 所以保持默认 100%就可以了。


框 6, 读写比例, 这个参数对结果影响也很大, 因为读的性能是比写的性能好很多的。

框 7, 随即/顺序读写的百分比, 这个参数对结果影响也很大, 因为顺序读写, 比随机读写的性能好很多。

框 8, 可以模拟定量的 IOPS, 比如在 Delay 里面填 1000 (单位毫秒, 1000 毫秒即 1 秒钟), 然后在 Burst Length 里面填 10. 意思就是在一秒钟会发生 10 个 IOPS。

框 9、10, 保持默认即可。

在测试中, 我们通过改变三个参数, 即上面黑体部分的块大小、读写百分比、随机顺序百分比来模拟不同场景的应用。 如果我们想得到 IOPS 的极限, 那么我们可以用 “512Byte; 100%Read; 0%random” 规格来跑。 如果我们想得到带宽的极限, 那么我们可以用 “32KB (可以用 64K、128K、1024K 或者更大, 这需要调测几次看哪个值能够得到带宽最大值); 100%Read; 0%random” 的规格来跑。

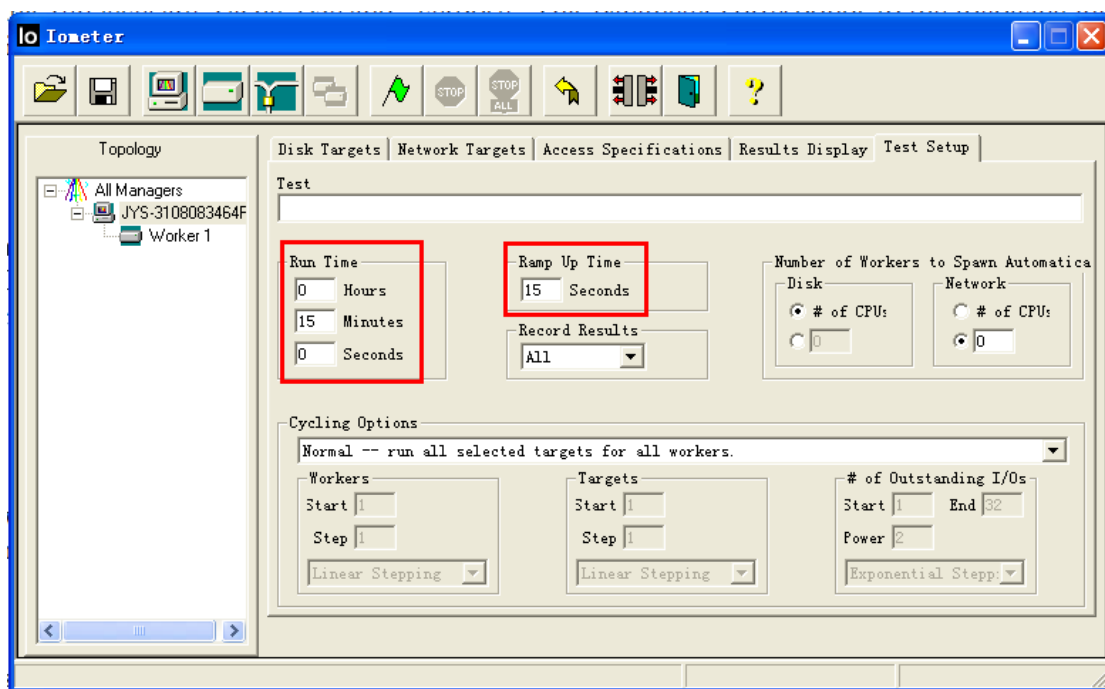
: 传输数据块大小在应用服务器类型测试为 4KB, 数据库服务器类型测试为 8KB; 读写百分比在应用服务器类型测试为读 100%, 数据库服务器类型测试根据实际情况来判断, 如纯查询的数据库读 100%, 一个典型的业务系统的数据库系统, 按照默认的 67%读即可; 随机/连续存取百分比在应用服务器类型测试为 100%, 数据库服务器类型测试为 100%; (测试人员可根据实际情况修改此处数值, 典型的 OLAP 环境: 选择顺序的大 IO, 测试存储所能支持的最大吞吐量以及响应时间; 典型的 OLTP 环境: 选择随机的小 IO, 测试存储所能支持的最大 IOPS 以及响应时间)。

3.1.5 Test Setup 界面

可以设置测试跑多长时间, 即 “Run Time”。

设置 “Ramp Time” 的目的是保证结果更准确, 因为 IO 测试开始的前几秒钟得出的数据与实际数据误差较大, 例如我们这里设置 “15” 的意思是, 真正的 IO 测试开始 15 秒钟后, 才会在 “Result Display” 界面中记录测试结果。

其它设置保持默认。

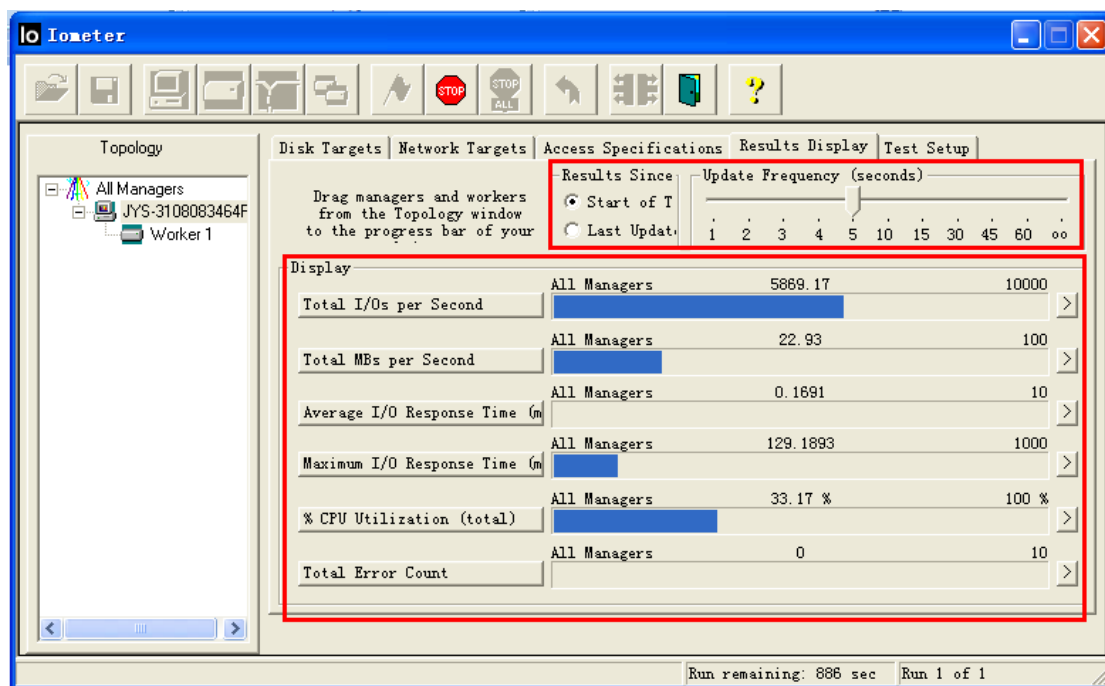


3.1.6 Result Display 界面

Update Frequency, 指下面的蓝色条的结果更新的频率, 默认是无穷大, 如果不点击鼠标修改的话会一直看不到更新的结果。当进行一项高负载测试时, 不建议设置更新频率过小, 否则会影响系统性能。


Results Since, 选择上面的 Start of Test 是说从记录结果开始一直到现在的平均结果。选择下面的 Last Update 意思是, 只显示最近 5 秒钟的性能结果。

这个界面可以通过点击鼠标, 来选择显示不同的参数, 动手试试吧。



3.2 Windows 下单机跑 IOmeter

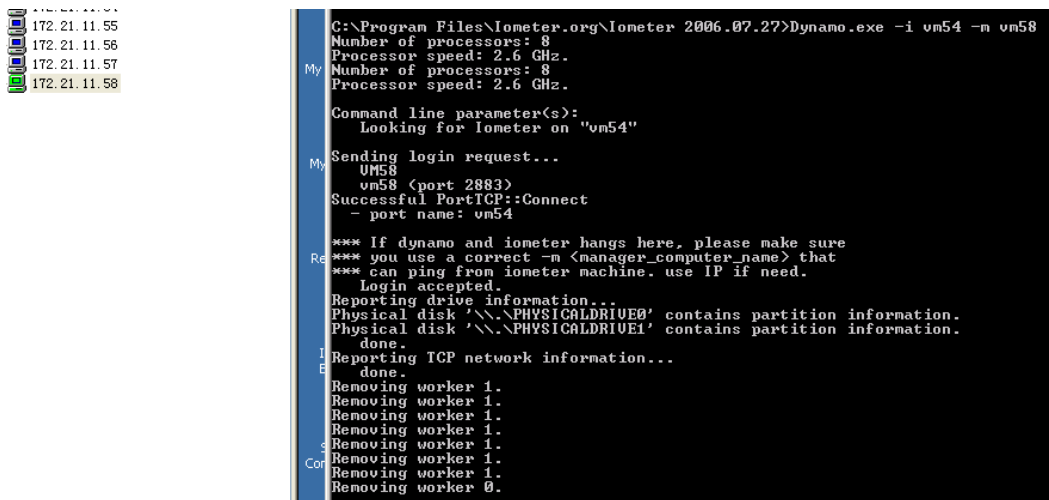
熟悉了上面讲的参数的意思后，Windows 上单机跑 IOmeter 就很简单了。

将参数设置好，点击 ，根据提示选择日志保存的位置和文件名 xxx.csv 就 OK 了，现在就在你的 PC 机上练练手吧。(PC 机性能有限，建议将# of Outstandings 设置在 10 以下，以免 PC 机卡死)

3.3 Windows 下用 IOmeter 跑网络压力

有两台虚拟机，VM54 和 VM58，以这两台 VM 为例介绍如何用 IOmeter 工具跑网络压力。

在 VM54 打开 IOmeter 界面作为 Console 端，如下图在 VM58 打开 CMD 进入 IOmeter 的安装目录路径下，然后输入命令 “Dynamo -i vm54(或者 VM54 的 IP 地址，-i 是指定 console 端) -m VM58 (或 VM58 的 IP，-m 是指定 Worker)”。(注意，要先打开 Console 端再在 client 端运行该命令，否则 Console 找不到 client)



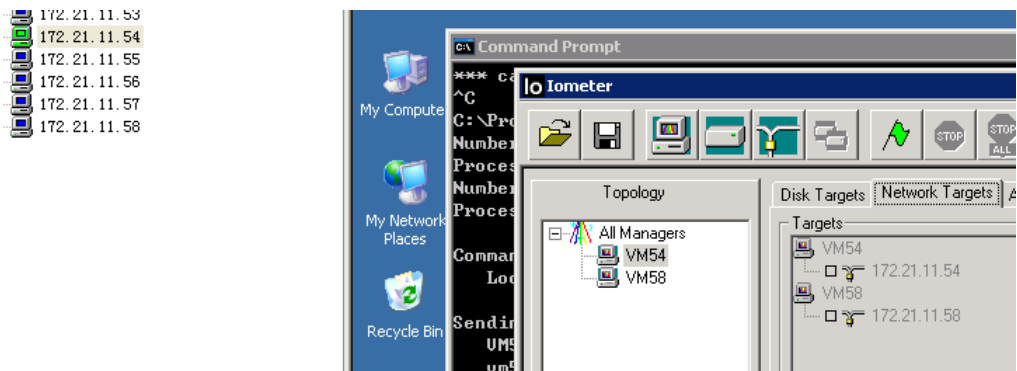
```
C:\Program Files\Iometer.org\Iometer 2006.07.27>Dynamo.exe -i vm54 -m vm58
Number of processors: 8
Processor speed: 2.6 GHz.
Number of processors: 8
Processor speed: 2.6 GHz.

Command line parameter(s):
Looking for iometer on "vm54"

Sending login request...
VM58
vm58 (port 2883)
Successful PortTCP::Connect
- port name: vm54

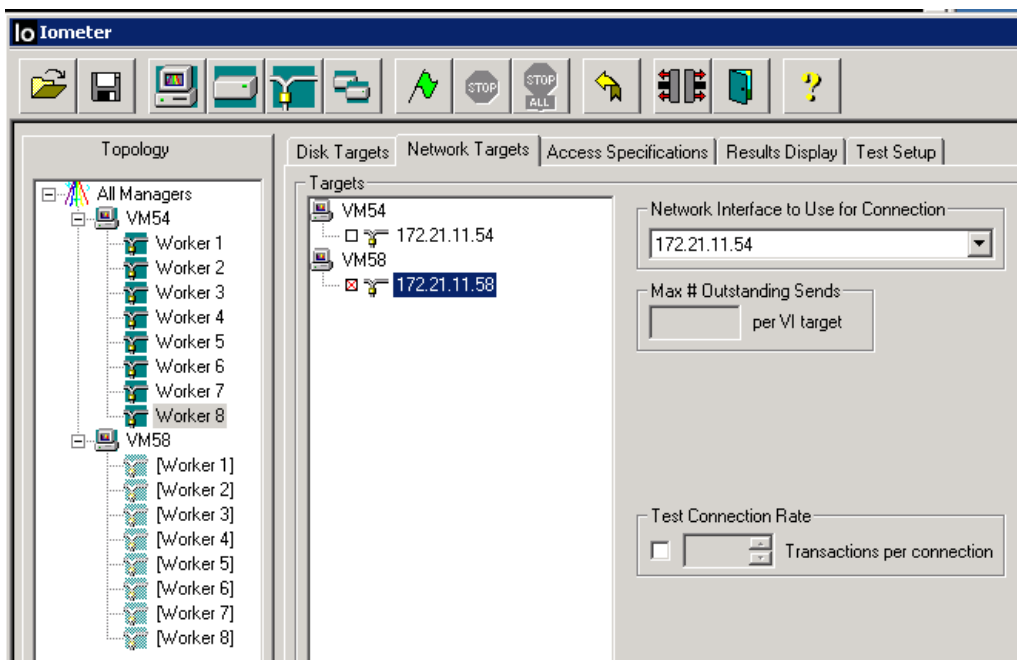
*** If dynamo and iometer hangs here, please make sure
*** you use a correct -m <manager.computer_name> that
*** can ping from iometer machine. use IP if need.
Login accepted.
Reporting drive information...
Physical disk '\\.\PHYSICALDRIVE0' contains partition information.
Physical disk '\\.\PHYSICALDRIVE1' contains partition information.
done.
Reporting TCP network information...
done.
Removing worker 1.
Removing worker 1.
Removing worker 1.
Removing worker 1.
Removing worker 1.
Removing worker 1.
Removing worker 1.
Removing worker 0.
```

这样，如图在 VM54 的控制台上出现 VM58

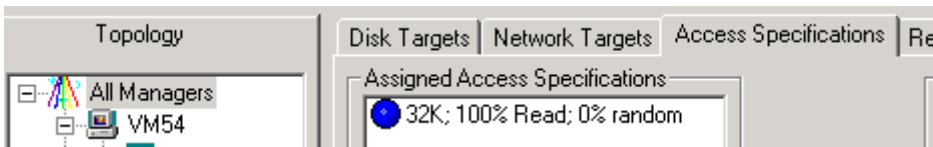


由于这里只演示跑网络压力，所以将 Disk Worker move 掉。

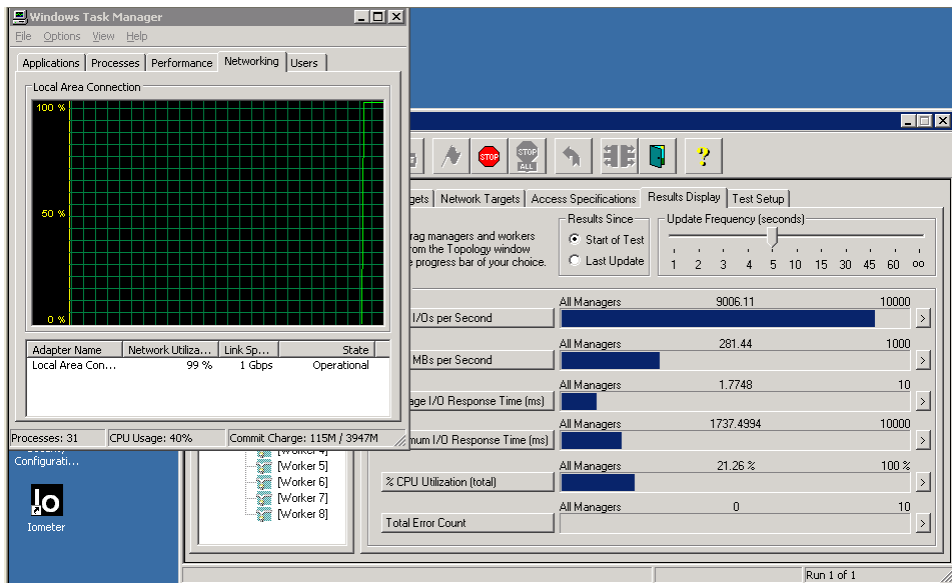
在 VM54 增加 8 个 Network 的 Worker，并添加 VM58 为 Targets，如下图



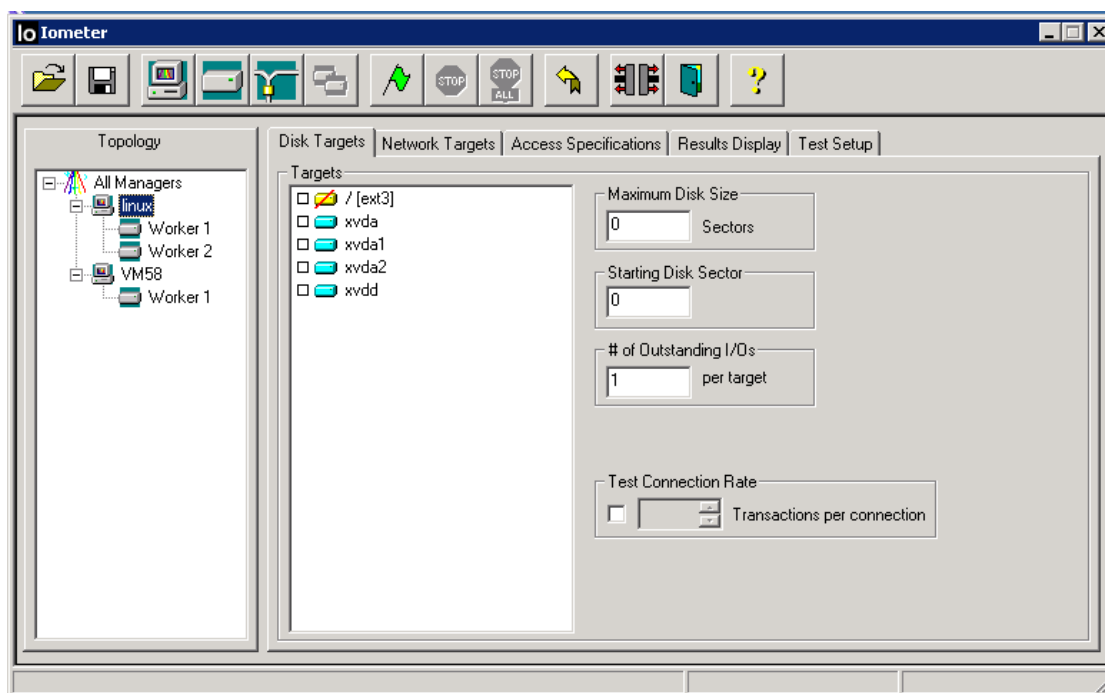
增加如下规格的连接



开始 run，如下图，网络利用达到了 100%



测试结果



在 Console 端出现 Linux 的 client 后，其他操作同在 Windows 单机跑 IOmeter。

4 自动化脚本

4.1 根据帮助文档查看 IOmeter 参数

打开 cmd，在 IOmeter 目录下，输入命令 `iometer ?` 得到以下帮助文档


```
Version 2006.07.27

SYNTAX
iometer
iometer /?
iometer config_file [result_file [timeout_value]]
iometer [/c config_file] [/r results_file] [/t timeout_value] [/p port] [/m 1]

? - show Iometer version number and command line syntax

config_file - the name of the Iometer configuration file to restore
              when Iometer starts (must be a valid .icf file)

results_file - the name of a writable text file to which the test
               results should be appended. If the file doesn't exist, it will be
               created.

timeout_value - the number of seconds to wait for managers to log in
                when restoring a config file.

port_number - the port number iometer listens on.

specify /m option will have BigMeter shown automatically.

If both config_file and results_file are specified, Iometer will
attempt to run the test in batch mode, requiring no user intervention
to load the config file, start the tests, stop the tests (if a run time
was specified in the Test Setup tab), and close Iometer. Batch mode
can be canceled by aborting the manager waiting list dialog or by
manually stopping a test while it is running.
```

4.2 自动化测试脚本

如果有多台机器要跑相同的 IOmeter 测试，就可以先在一台机器上将参数设置好，并保存为配置文件 xxx.icf。我们把这个配置文件拷贝到每台机器的 IOmeter 安装目录下，同时日志也放到安装目录下

创建如下脚本，并保存为 .bat 文件，双击就可以运行了

```
cd "%Program Files\IOmeter.org\IOmeter 2006.07.27"
```

```
iometer iometer_config.icf iometer_result.csv
```

5 性能指标

本章主要讲述在结果显示页签中各个性能指标的含义。

- Total I/Os per Second (IOPS): 每秒 I/O 次数, 包含读 I/O 和写 I/O。对于磁盘来说, 一次磁头的连续读或者连续写就是一次 I/O。
- Read I/Os per Second: 每秒读 I/O 次数。
- Write I/Os per Second: 每秒写 I/O 次数。
- Transaction per Second: 每秒事务处理数。当在存取规则中设置 Reply Size 为”no reply”时, Transaction per Second=IOPS, 即事务只包含发送数据块; 当在存取规则中设置了 Reply Size 不为 0 时, Transaction per Second 将发送和接收数据块作为一个事务。
- Connections per Second: 每秒连接数。
- Total MBs per Second: 每秒数据传输量, 也就是常说的吞吐量, 包含读取和写入。Total MBs per Second=IOPS*传输数据块大小= Transaction per Second* (传输数据块+接收数据块)
- Read MBs per Second: 每秒读取数据量。
- Write MBs per Second: 每秒写入数据量。
- Average I/O Response Time(ms): 平均 I/O 响应时间。
- Avg.Read Response Time(ms): 平均读 I/O 响应时间。
- Avg.Write Response Time(ms): 平均写 I/O 响应时间。
- Avg. Transaction Time(ms): 平均事务处理时间。
- Avg. Connections Time(ms): 平均连接时间。
- Maximum I/O Response Time(ms): 最大 I/O 响应时间。
- Max.Read Response Time(ms): 最大读 I/O 响应时间。
- Max.Write Response Time(ms): 最大写 I/O 响应时间。
- Max. Transaction Time(ms): 最大事务处理时间。
- Max. Connections Time(ms): 最大连接时间。
- %CPU Utilization (total): CPU 占用率。
- %User Time: 非内核级应用程序占用时间。
- %Privileged Time: CPU 内核程序占用时间, 是在特权模式下处理线程执行代码所花时间的百分比
- %DPC Time: 处理器在网络处理上消耗的时间。%DPC Time 是%Privileged Time 的一部分。
- %Interrupt Time: 中断时间是在采样间隔期间接收和处理硬件中断处理器花费的时间百分比, 如系统时钟, 鼠标, 磁盘驱动器, 数据通信线路, 网络接口卡和其它外围设备。
- Interrupts per Second: 每秒中断数。
- CPU Effectiveness: CPU 效率的一个表征: 将 IOPS 除以%CPU Utilization 即可得到。
- Network Packets per Second: 每秒网络数据包发送/接收数。

-
- Packets Errors: 错误包个数。
 - TCP Segments Retrans.per Sec: TCP 数据段每秒返回数。
 - Total Error Count: 总错误数。
 - Read Error Count: 读取错误数。
 - Write Error Count: 写入错误数。