

服务器性能瓶颈分析方法

1. 内存分析方法

内存分析用于判断系统有无内存瓶颈，是否需要通过增加内存等手段提高系统性能表现。

内存分析需要使用的计数器：**Memory** 类别和 **Physical Disk** 类别的计数器。内存分析的主要方法和步骤：

(1) 首先查看 **Memory\Available Mbytes** 指标

如果该指标的数据比较小，系统可能出现了内存方面的问题，需要继续下面步骤进一步分析。

注：在 **UNIX/LINUX** 中，对应指标是 **FREE(KB)**

(2) 注意 **Pages/sec**、**Pages Read/sec** 和 **Page Faults/sec** 的值

操作系统会利用磁盘较好的方式提高系统可用内存量或者提高内存的使用效率。这三个指标直接反应了操作系统进行磁盘交换的频度。

如果 **Pages/sec** 的计数持续高于几百，可能有内存问题。但 **Pages/sec** 值不一定就表明有内存问题，可能是运行使用内存映射文件的程序所致。**Page Faults/sec** 说明每秒发生页面失效次数，页面失效次数越多，说明操作系统向内存读取的次数越多。此事需要查看 **Pages Read/sec** 的计数值，该计数器的阈值为 **5**，如果计数值超过 **5**，则可以判断存在内存方面的问题。

注：在 **UNIX/LINUX** 系统中，对于指标是 **(page)si** 和 **(page)so**。

(3) 根据 **Physical Disk** 计数器的值分析性能瓶颈

对 **Physical Disk** 计数器的分析包括对 **Page Reads/sec** 和 **%Disk Time** 及 **Average Disk Queue Length** 的分析。如果 **Pages Read/sec** 很低，同时 **%Disk Time** 和 **Average Disk Queue Length** 的值很高，则可能有磁盘瓶颈。但是，如果队列长度增加的同时 **Pages Read/sec** 并未降低，则是内存不足。

注：在 **UNIX/LINUX** 系统中，对应的指标是 **Reads(Writes)per sec**、**Percent of time the disk is busy** 和 **Average number of transactions waiting for service**。

2. 处理器分析法

(1) 首先看 **System\%Total Processor Time** 性能计数器的计数值

该计数器的值体现服务器整体处理器利用率，对多处理器的系统而言，该计数器提醒所有 **CPU** 的平均利用率。如果该值持续超过 **90%**，则说明整个系统面临着处理器方面的瓶颈，需要通过增加处理器来提高性能。

注：多处理器系统中，该数据本身不大，但 **PUT** 直接负载状况极不均衡，也应该视作系统产生处理器方面瓶颈。

(2) 其次查看每个 **CPU** 的 **Processor\%Processor Time** 和 **Processor\%User Time** 和 **Processor\%Privileged Time**

Processor\%User Time 是系统非核心操作消耗的 CPU 时间，如果该值较大，可以考虑是否能通过友好算法等方法降低这个值。如果该服务器是数据库服务器，Processor\%User Time 值大的原因很可能是数据库的排序或是函数操作消耗了过多的 CPU 时间，此时可以考虑对数据库系统进行优化。

(3) 研究系统处理器瓶颈

查看 System\Processor Queue Length 计数器的值，当该计数器的值大于 CPU 数量的总数+1 时，说明产生了处理器阻塞。在处理器的%Process Time 很高时，一般都随处理器阻塞，但产生处理器阻塞时，Processor\%Process Time 计数器的值并不一定很大，此时就必须查找处理器阻塞的原因。

%DOC Time 是另一个需要关注的内容，该计数器越低越好。在多处理器系统中，如果这个值大于 50%，并且 Processor\%Precessor Time 非常高，加入一个网卡可能回提高性能。

3.磁盘 I/O 分析方法

(1) 计算每磁盘的 I/O 数

每磁盘的 I/O 数可用来与磁盘的 I/O 能力进行对比，如果经过计算得到的每磁盘 I/O 数超过了磁盘标称的 I/O 能力，则说明确实存在磁盘的性能瓶颈。

每磁盘 I/O 计算方法

RAID0 计算方法: $(\text{Reads} + \text{Writes}) / \text{Number of Disks}$

RAID0 计算方法: $(\text{Reads} + 2 * \text{Writes}) / 2$

RAID0 计算方法: $[\text{Reads} + (4 * \text{Writes})] / \text{Number of Disks}$

RAID0 计算方法: $[\text{Reads} + (2 * \text{Writes})] / \text{Number of Disks}$

(2) 与 Processor\Privileged Time 合并进行分析

如果在 Physical Disk 计数器中，只有%Disk Time 比较大，其他值都比较适中，硬盘可能会是瓶颈。若几个值都比较大，且数值持续超过 80%，则可能是内存泄漏。

(3) 根据 Disk sec/Transfer 进行分析

一般来说，定义该数值小于 15ms 为 Excellent，介于 15~30ms 之间为良好，30~60ms 之间为可以接受，超过 60ms 则需要考虑更换硬盘或是硬盘的 RAID 方式了。

4.进程分析方法

(1) 查看进程的%Processor Time 值

每个进程的%Processor Time 反映进程所消耗的处理器时间。用不同进程所消耗的处理器时间进行对比，可以看出具体哪个进程在性能测试过程中消耗了最多的处理器时间，从而可以据此针对应用进行优化。

(2) 查看每个进程产生的页面失效

可以用每个进程产生的页面失效(通过 `PROCESS\PAGE FAILURES/SEC` 计数器获得)和系统页面失效(可以通过 `MEMORY\PAGE FAILURES/SEC` 计数器获得)的比值,来判断哪个进程产生了最多的页面失效,这个进程要么是需要大量内存的进程,要么是非常活跃的进程,可以对其进行重点分析。

(3) 了解进程的 Process/Private Bytes

Process/Private Bytes 是指进程所分配的无法与其他进程共享的当前字节数量。该计数器主要用来判断进程在性能测试过程中有无内存泄漏。例如:对于一个 IIS 之上的 WEB 应用,我们可以重点监控 `inetinfo` 进程的 **Private Bytes**,如果在性能测试过程中,该进程的 **Private Bytes** 计数器值不断增加,或是性能测试停止后一段时间,该进程的 **Private Bytes** 仍然持续在高水平,则说明应用存在内存泄漏。

注:在 UNIX/LINUX 系统中,对应的指标是 **Resident Size**

5.网络分析方法

(1) `Network Interface\Bytes Total/sec` 为发送和接收字节的速率,可以通过该计数器值来判断网络链接速度是否是瓶颈,具体操作方法是使用该计数器的值和目前网络的带宽进行相除,结果小于 50%。

$1\text{Mdit/sec(兆比特/秒)}=131072\text{bytes/sec(字节/秒)}$

$1\text{byte}=8\text{bit}$

IOMeter 安装、配置

IOMeter 是业界广泛使用的 IO 性能测试软件。通过使用不同的 IOMeter 设定值来呈现文件服务器（选择范围为 512Bytes 到 64KB）和网站服务器（选择 512Bytes 到 512KB 作为 I/O 范围）的相应负载。

在实际测试过程中可以使用 IOMeter 进行如下测量：

Max IO:

为了得到完整的磁盘性能数据，可以测试 512Bytes 到 512MBytes 等多种数据块大小，并分别测试了 100%读取操作或 100%写入操作下的表现，测试随机率可以从 0%-100%，用于检测磁盘子系统的最大输入输出能力。从所有的数据中，可以分别得到的数据指 IOps（每秒操作数）以及 MBps（传输速率）。

File Server:

文件服务器应用在一个多用户或网络化的环境中专门用于文件的储存，恢复和管理。不同用户的访问形成了多线程，因此，其工作负载是随机性的。输入输出大小是可变的，取决于所操作的文件大小。读/写分布也是变化的。总的来说，一个频繁使用的多用户服务器应该采用较高的队列深度来模拟。

Web Server:

网站服务器应用于处理多用户通过 Internet 或 Intranet 进行的访问。这些请求具有典型的随机性，并且在用户访问不同页面或视频应用时大量偏向于读请求。同样，一个访问频繁网站服务器也应该采用较高的队列深度来模拟。最后，测试读取尺寸从 512Bytes 到 512KB 不等，100%读取操作，随机率为 100%，用于模拟 Web 服务器的性能。在实际的桌面环境中，小于 4KB 的块访问数量不少，但是由于操作系统的缓存的存在而具有一定的降低，主要的块大小基于 4KB。

IOMeter 硬盘性能测试对 CPU、内存等参数并不是很敏感——当然 CPU 要保证在测试中不能出现 100%的占用率。

第 1 页

接下来就让我们一起来看看 IOMETER 的安装与配置的全过程。

我现在所使用的 IOMETER 版本是 iometer-2004.07.30.win32.i386 版。

其安装过程如下：

第一步双击打开安装文件：



点击 NEXT:



选择加入 MS Access 导入导航:



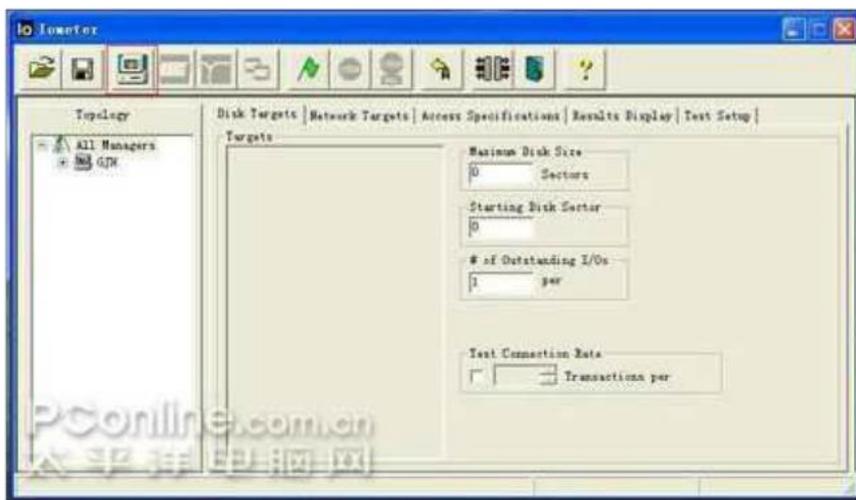
点击 finish:



从所有程序列表中打开 IOMETER。



选中下图中的标记的图标可以产生一个新的测试机。



增加一个新的测试机之后的情形。



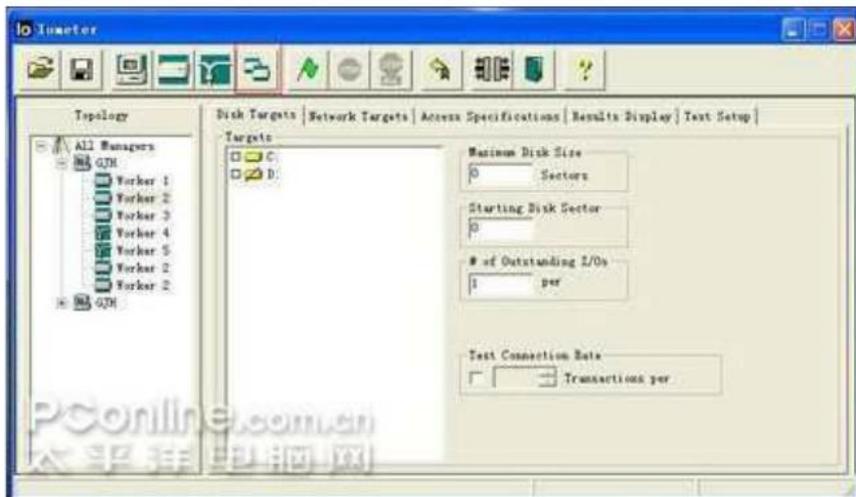
点击以下被标记图标可以新增一个任务组：



点击一下被标记的图标可以增加一个网络连接。



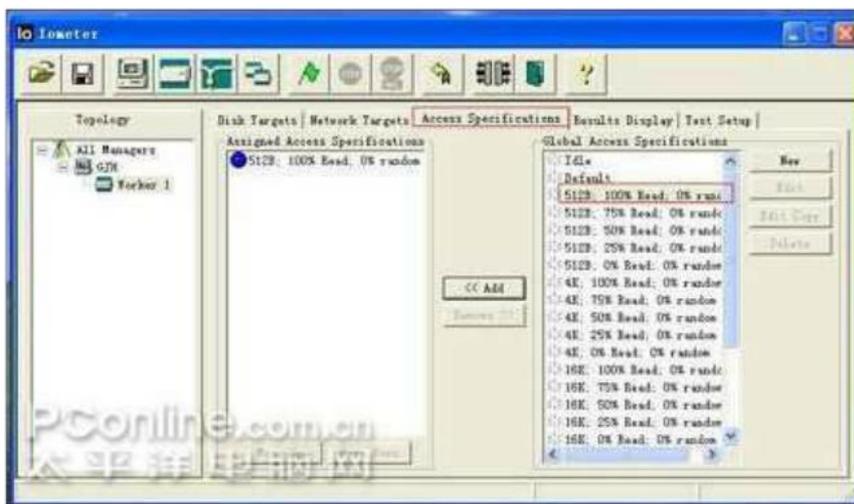
选中一下被标记的图标可以复制所选中的任务组。



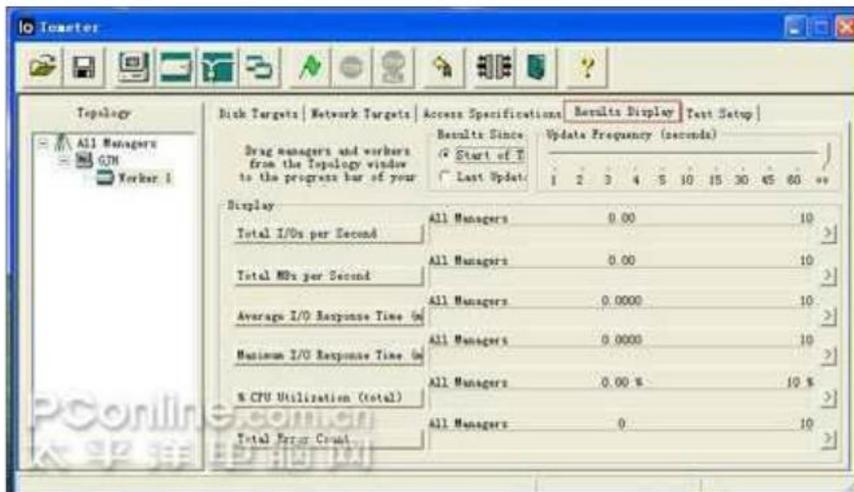
选中需要被测试的硬盘。



添加需要被测试的数据库模式。



察看测试结果。



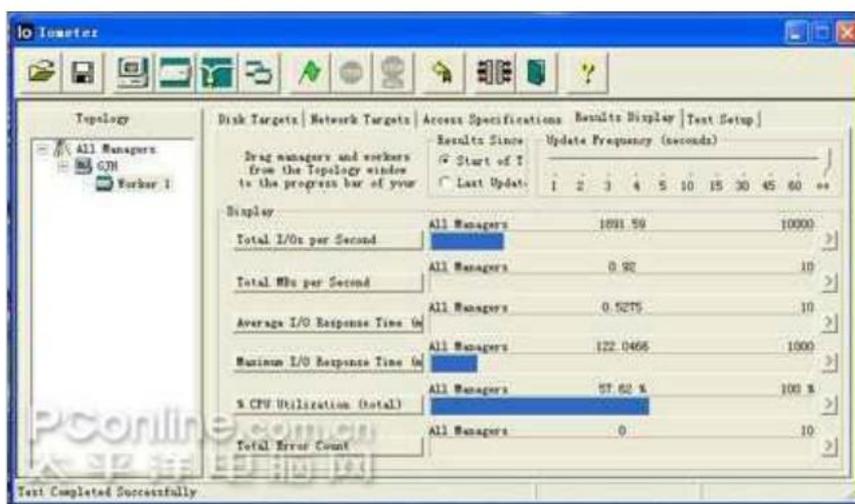
选中以下被标记的图标可以设置测试的时间间隔。



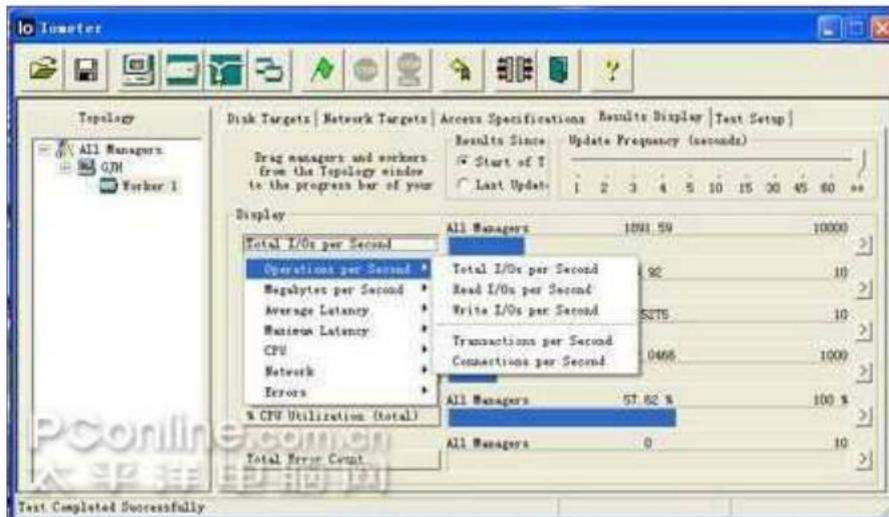
选中以下被标记的图标开始测试，选择结果存放地址。



开始测试后，察看测试结果。



测试结果中包含的结果项。



总的 I/O 速度。



读数据的 I/O 速度。



写数据的 I/O 速度。



总的数据传输速度。



总的每秒联接数。



总的每秒读写数据量。



每秒读取的数据量。



每秒写入的数据量。



总的平均响应时间。



平均的读取响应时间。



平均的写入响应时间。



平均的传输响应时间。



平均的连接响应时间。



最大的 I/O 响应时间。



最大的读取响应时间。



最大的写入响应时间。



最大的传输响应时间。



最大的连接响应时间。



总的 CPU 使用率。



CPU 使用时间的占用率。



优先使用时间占用率。



缩节时间占用率。



中断时间占用率。



CPU 的有效使用率。



网络数据包传输速率。



总的传输错误数。



TCP 中转错误数。



总的出错数。



读取出错数。



写入出错数。

