

1. 添加userparameter_io.conf配置文件

在/etc/zabbix/zabbix_agentd.d下添加userparameter_io.conf, 文件内容如下:

```
UserParameter=custom.vfs.dev.read.ms[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$7}'
UserParameter=custom.vfs.dev.write.ops[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$8}'
UserParameter=custom.vfs.dev.write.ms[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$11}'
UserParameter=custom.vfs.dev.io.active[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$12}'
UserParameter=custom.vfs.dev.io.ms[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$13}'
UserParameter=custom.vfs.dev.read.sectors[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$6}'
UserParameter=custom.vfs.dev.write.sectors[*],cat /proc/diskstats | grep $1 | head -1 | awk '{print $$10}'
```

2. 重启zabbix-agent服务

```
service zabbix-agent restart
```

3. 测试zabbix-server获取数据

```
zabbix_get -s 目标服务器IP -p 10050 -k custom.vfs.dev.write.ops[sda]
```

4. 配置zabbix-server模板

□

5. 配置zabbix-server项目(item)

```
第一个指标Name:      Disk:$1:Read:Bytes/sec
Key:                  custom.vfs.dev.read.sectors[sda]
Units:                B/sec
Store value: speed per second    //会进行差值计算
Use custom multiplier    512      //会对值乘以512, 因为这里是一个扇区, 转换为字节为512B
```

□

同理, 其他指标方式, 添加如下:

```
第二个指标Name:      Disk:$1:Write:Bytes/sec
Key:                  custom.vfs.dev.write.sectors[sda]
Units:                B/sec
Store value: speed per second
Use custom multiplier    512
```

```
第三个指标Name:      Disk:$1:Read:ops per second
Key:                  custom.vfs.dev.read.ops[sda]
Units:                ops/second
Store value: speed per second
```

```
第四个指标Name:      Disk:$1:Write:ops per second
Key:                  custom.vfs.dev.write.ops[sda]
Units:                ops/second
Store value: speed per second
```

```
第五个指标Name:      Disk:$1:Read:ms
Key:                  custom.vfs.dev.read.ms[sda]
Units:                ms
Store value: speed per second
```

```
第六个指标Name:      Disk:$1:Write:ms
Key:                  custom.vfs.dev.write.ms[sda]
Units:                ms
Store value: speed per second
```

6. 添加zabbix-server图形(Graphs)

□

7. 效果图展示

□

参考: <http://blog.csdn.net/a294039255/article/details/51546311>

原理

通过分析/proc/diskstats文件，来对I/O的性能进行监控。解释如下：

```
++++++对/proc/diskstats的解释++++++  
[root@localhost bin]# cat /proc/diskstats | grep sda | head -1  
8 0 sda 73840 10263 3178156 91219 1110085 4192562 42423152 1275861 0 447798 1366379
```

第一至第三个域，分别是主设备号，次设备号，设备名称

第4个域：读完成次数 ----- 读磁盘的次数，成功完成读的总次数。

(number of issued reads. This is the total number of reads completed successfully.)

第5个域：合并读完成次数， 第9个域：合并写完成次数。为了效率可能会合并相邻的读和写。从而两次4K的读在它最终被处理到磁盘上之前可能会变成一次8K的读，才被计数（和排队），因此只有一次I/O操作。这个域使你知道这样的操作有多频繁。

(number of reads merged)

第6个域：读扇区的次数，成功读过的扇区总次数。

(number of sectors read. This is the total number of sectors read successfully.)

第7个域：读花费的毫秒数，这是所有读操作所花费的毫秒数（用__make_request()到end_that_request_last()测量）。

(number of milliseconds spent reading. This is the total number of milliseconds spent by all reads (as measured from __make_request() to end_that_request_last()).)

第8个域：写完成次数 ----- 写完成的次数，成功写完成的总次数。

(number of writes completed. This is the total number of writes completed successfully.)

第9个域：合并写完成次数 ----- 合并写次数。

(number of writes merged Reads and writes which are adjacent to each other may be merged for efficiency. Thus two 4K reads may become one 8K read before it is ultimately handed to the disk, and so it will be counted (and queued) as only one I/O. This field lets you know how often this was done.)

第10个域：写扇区次数 ----- 写扇区的次数，成功写扇区总次数。

(number of sectors written. This is the total number of sectors written successfully.)

第11个域：写操作花费的毫秒数 ----- 写花费的毫秒数，这是所有写操作所花费的毫秒数（用__make_request()到end_that_request_last()测量）。

(number of milliseconds spent writing This is the total number of milliseconds spent by all writes (as measured from __make_request() to end_that_request_last()).)

第12个域：正在处理的输入/输出请求数 -- I/O的当前进度，只有这个域应该是0。当请求被交给适当的request_queue_t时增加和请求完成时减小。

(number of I/Os currently in progress. The only field that should go to zero. Incremented as requests are given to appropriate request_queue_t and decremented as they finish.)

第13个域：输入/输出操作花费的毫秒数 ----- 花在I/O操作上的毫秒数，这个域会增长只要field 9不为0。

(number of milliseconds spent doing I/Os. This field is increased so long as field 9 is nonzero.)

第14个域：输入/输出操作花费的加权毫秒数 ----- 加权， 花在I/O操作上的毫秒数，在每次I/O开始，I/O结束，I/O合并时这个域都会增加。这可以给I/O完成时间和存储那些可以累积的提供一个便利的测量标准。

(number of milliseconds spent doing I/Os. This field is incremented at each I/O start, I/O completion, I/O merge, or read of these stats by the number of I/Os in progress (field 9) times the number of milliseconds spent doing I/O since the last update of this field. This can provide an easy measure of both I/O completion time and the backlog that may be accumulating.)

测试命令如下：

```
./zabbix_get -s 10.2.11.11 -p 10050 -k custom.vfs.dev.write.ops[sda]
```