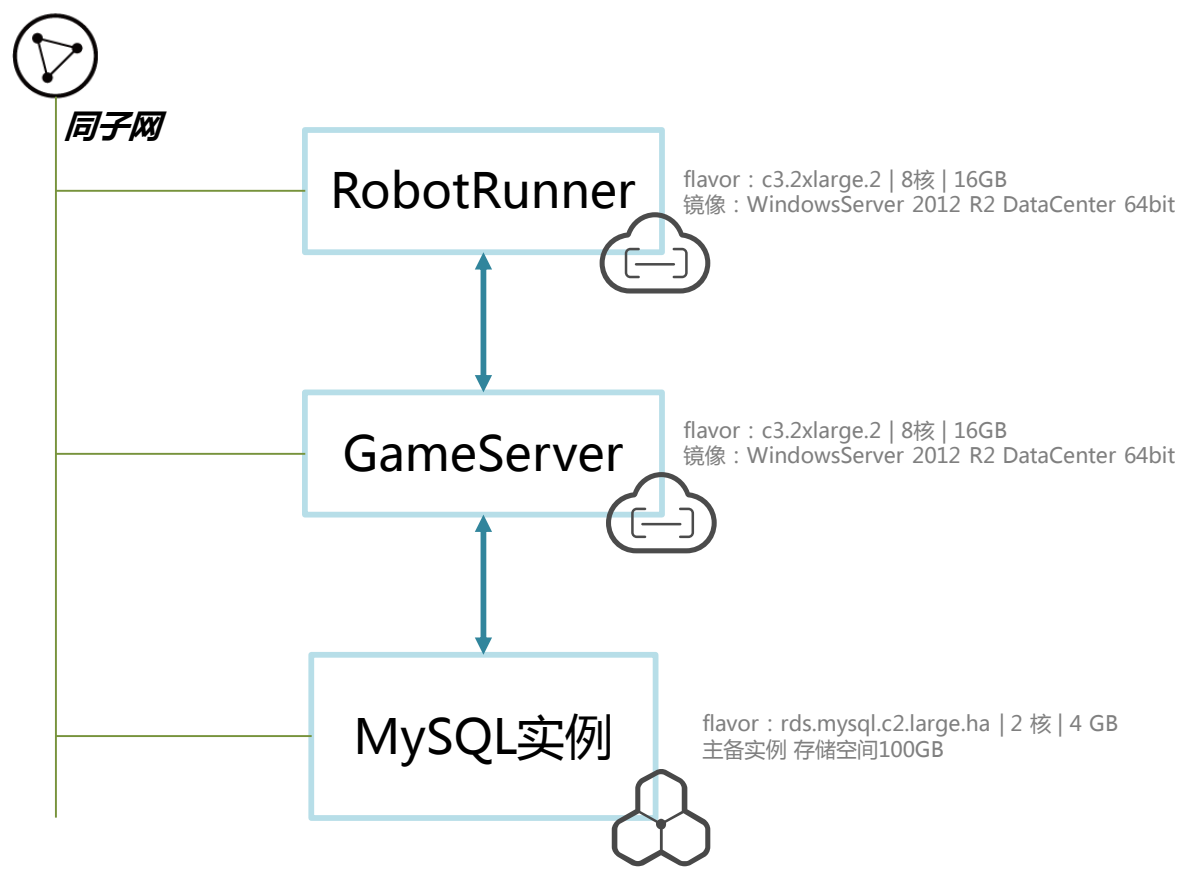


# 游戏业务压力测试案例

WindowsServer2012R2操作系统下压测性能表现差问题定位分析

[www.huawei.com](http://www.huawei.com)

客户该场景下的压力测试构造的业务压力通过内网传输，拓扑图如下（压测场景省略了Web服务等周边系统）



RobotsRunner上可以启动压测机器人脚本  
模拟玩家以下行为场景对服务端构造业务压力

服务器压力测试工具

用户数量 (1-3000) 版本: v3000

输入 help,	显示帮助信息
输入 exit,	然后输入,退出?
输入 baseinfo,	显示基本信息
输入 tologin,	连接登录服务器进行登录
输入 togame,	进入游戏服务器
输入 tomap /id,	进入指定地图
输入 tofuben /id,	进入指定副本
输入 leavefuben,	离开副本
输入 walkon,	开始自动寻路
输入 walkoff,	停止自动寻路
输入 chaton,	开始自动聊天
输入 chatoff,	停止自动聊天
输入 skillon,	开始自动使用技能
输入 skilloff,	停止自动使用技能
输入 taskon /id,	自动主线任务
输入 taskoff,	停止主线任务
输入 taskinfo,	任务信息
输入 logout,	角色下线

## 测试场景及问题现象

### 测试场景：

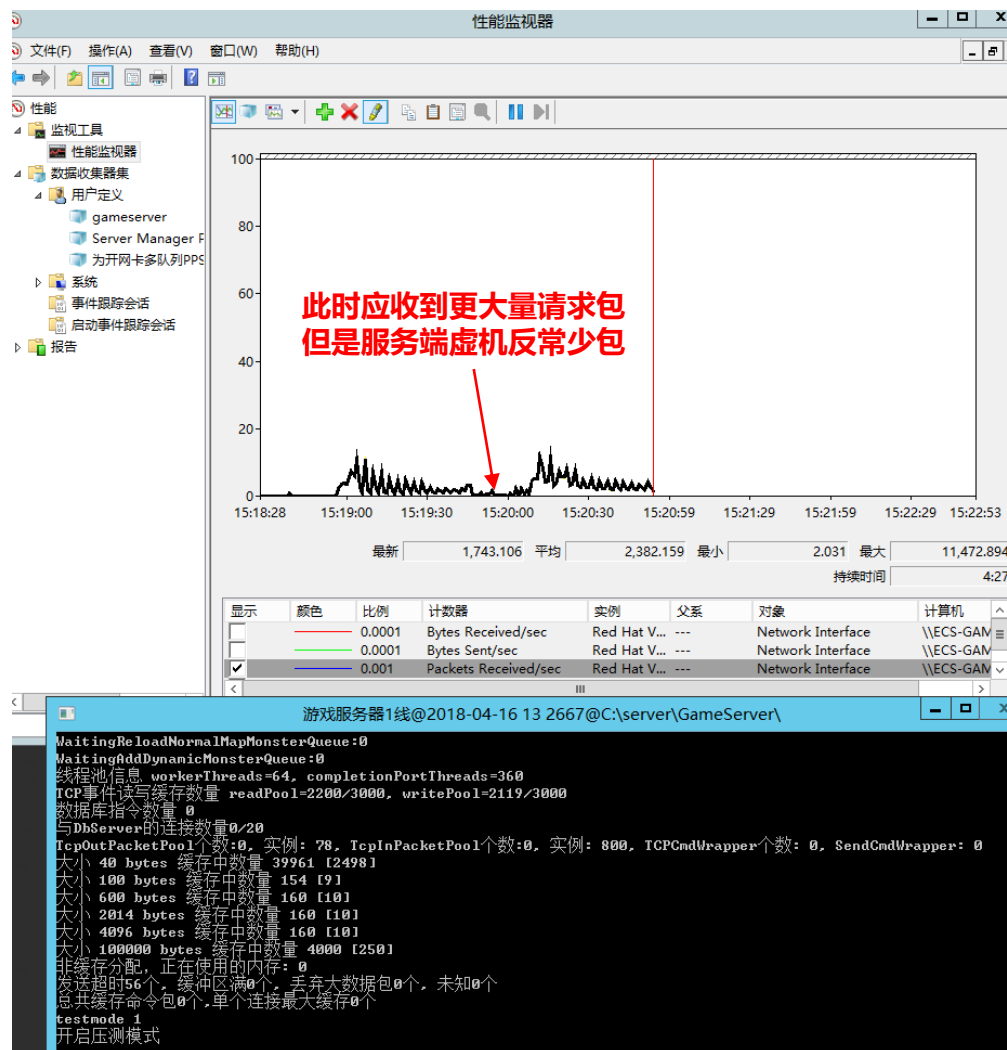
模拟短期内大量玩家登录，按照客户常规测试操作流程，先启动一个压测进程模拟400玩家登入游戏，待第一批400玩家“在游戏中”状态后，再启动一个压测集成模拟下一批400玩家登入游戏，依次叠加。

## 问题现象

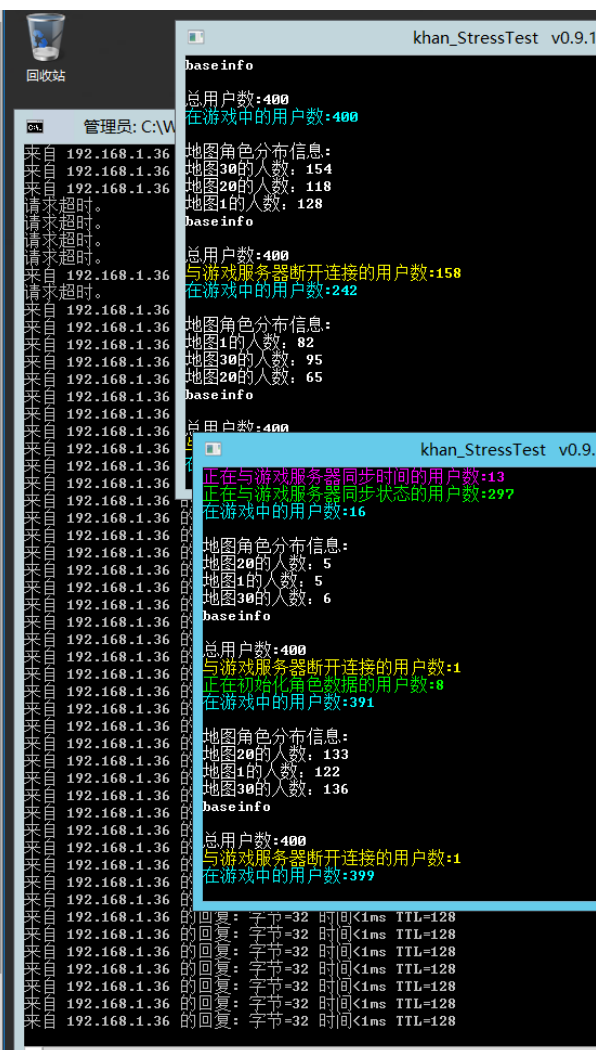
在华为云上部署的服务端，在第2批400玩家尝试登陆的时候，出现第1批400个“在游戏中”的玩家有一定比例掉线的情况，且第2批玩家登入耗时较正常情况明显变长，且不能全部登入游戏。

- 在客户端 ( RobotRunner ) ping服务端 ( GameServer ) , 发现问题出现时有丢包情况 ;
- 在服务端使用Windows计数器检测收包情况 , 按道理此时刻客户端发出来的包数量更大 , 但是问题出现时服务端收到的包反常的明显变少 ;

## 服务端

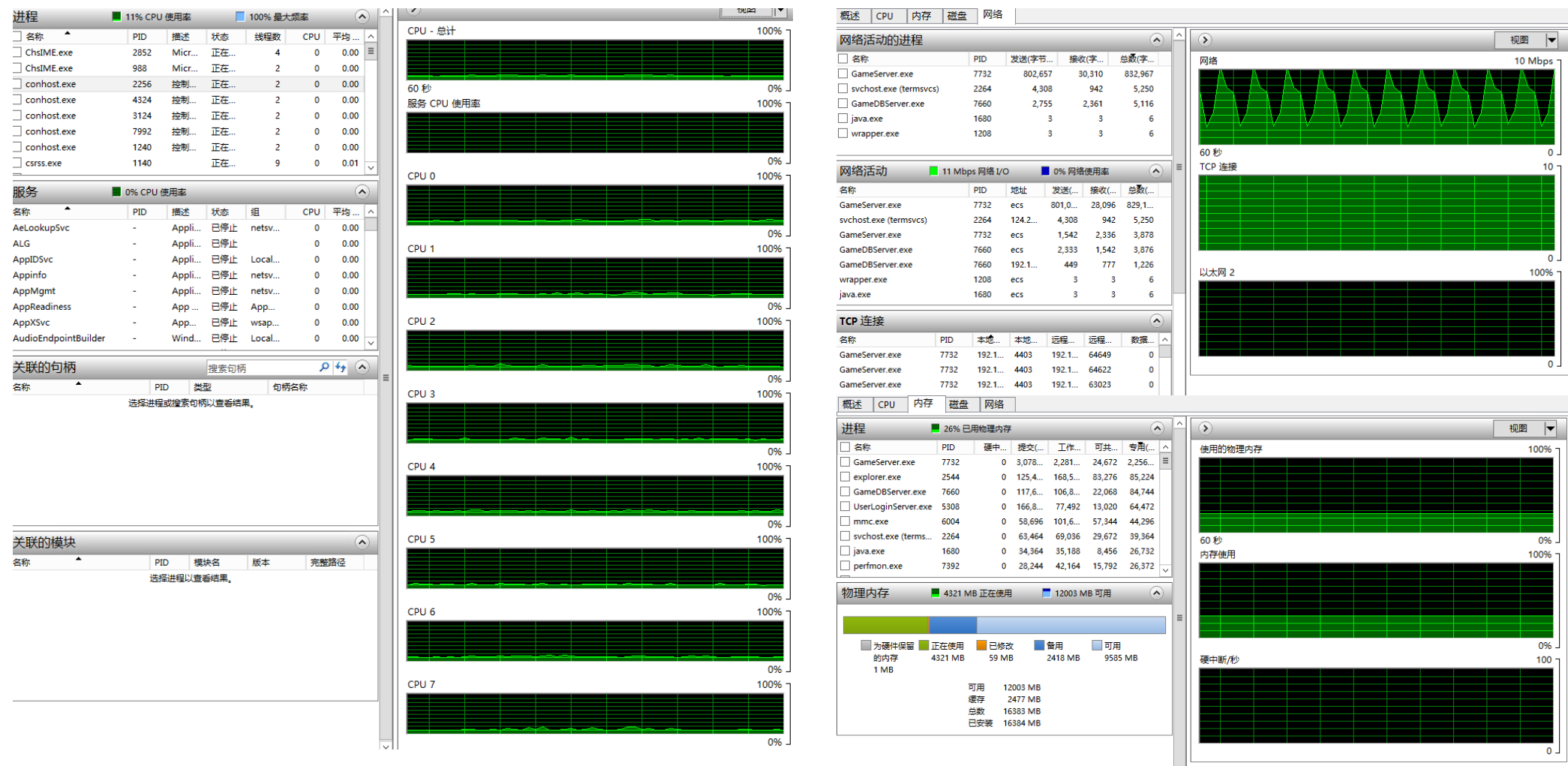


## 客户端



# Step1：排查虚拟机CPU、内存、带宽是否达到瓶颈

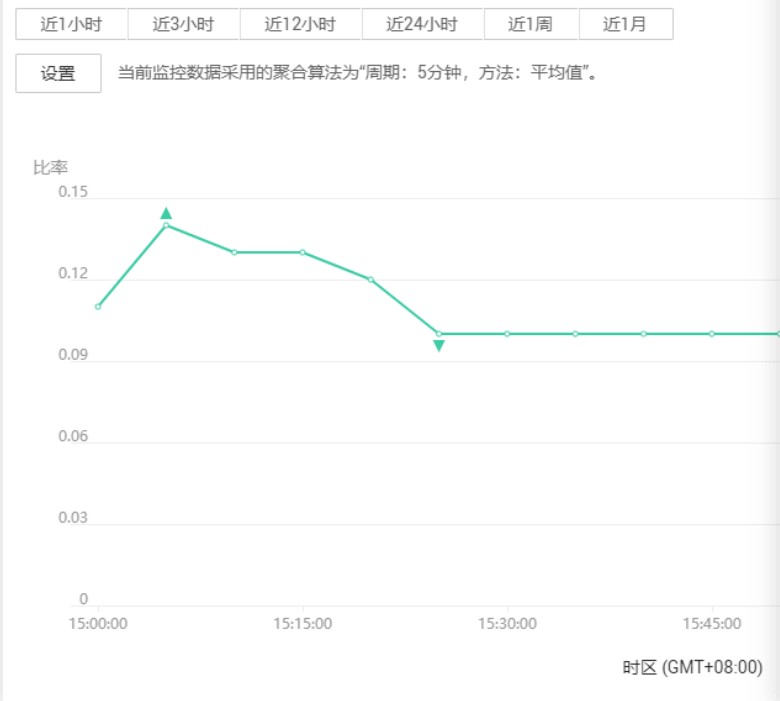
问题出现期间，服务端CPU占用<13%,内存占用<20%，带宽<30Mbps；  
通过Windows性能监视器观察，8个核负载基本均衡，没有出现个别CPU负荷过高的现象；



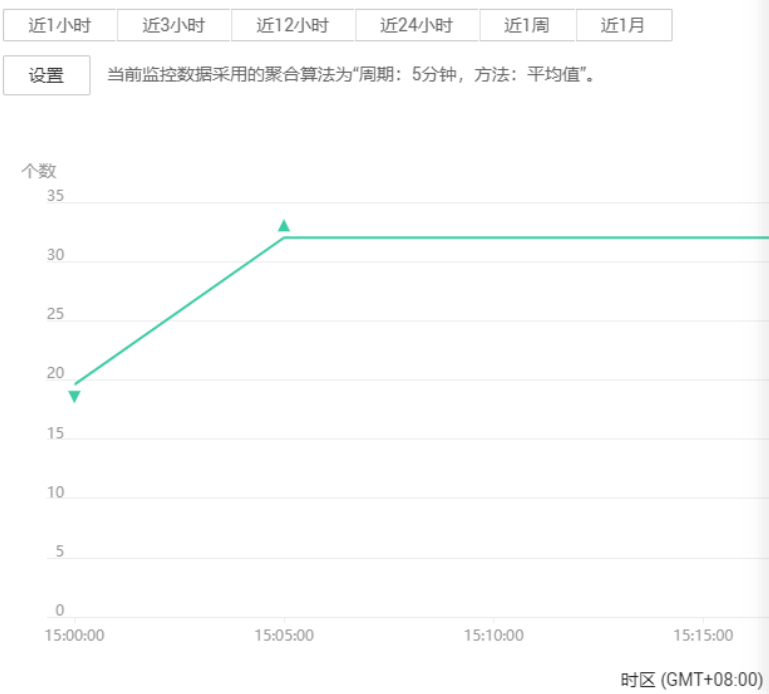
# Step2：排查数据库是否达到瓶颈

问题出现期间，数据库的各项监控指标正常，离数据库能力瓶颈非常远  
且通过后台协助查询未发现慢日志等明显问题现象，基本可以排除是由于数据库瓶颈造成出现问题现象

CPU使用率



数据库总连接数



IOPS



# Step3：排查并发连接数是否超过虚拟机的上限值

实际查看连接数只在1千多，通过查找资料发现Windows Server 2012 R2M默认TCP最大连接数并没有明确的查询未知，因此不能完全排查是否连接数限制  
不过接下来我们通过另一个场景，确认了不是连接数限制导致的该问题

```
管理员: C:\Windows\system32\cmd.exe
TCP 192.168.1.213:4403 192.168.1.20:64774 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64775 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64776 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64777 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64778 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64779 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64780 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64781 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64782 ESTABLISHED
TCP 192.168.1.213:4403 192.168.1.20:64783 ESTABLISHED
TCP 192.168.1.213:49745 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49746 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49747 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49748 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49749 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49750 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49751 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49752 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49753 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49754 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49755 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49756 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49757 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49758 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49759 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49760 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49761 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49762 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49763 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49764 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49765 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49766 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49767 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49768 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49769 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49770 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49771 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49772 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49773 192.168.1.92:3306 ESTABLISHED
TCP 192.168.1.213:49774 192.168.1.92:3306 ESTABLISHED
C:\Users\Administrator>netstat -an |find /c "ESTABLISHED"
1293
```

问题  
请教一下如何查询WindowsServer2012R2 TCP最大连接数！

尊敬的微软工程师，  
依据微软这份链接可以修改TCP最大连接数  
<https://technet.microsoft.com/zh-cn/library/cc938216.aspx?f=255&MSPPError=-2147217396>  
  
我查看了一下默认系统注册表是无该项数值的，因此我想知道系统在没有设置TcpNumConnections这个值的时候最大连接数是多少？  
有没有命令方式去查询？谢谢！

回复 我有相同的问题(13) ...

答案  
Tommy Xing 回复时间 2017/07/21  
↳ 2017/07/21 中对 谷青松 文章的回复

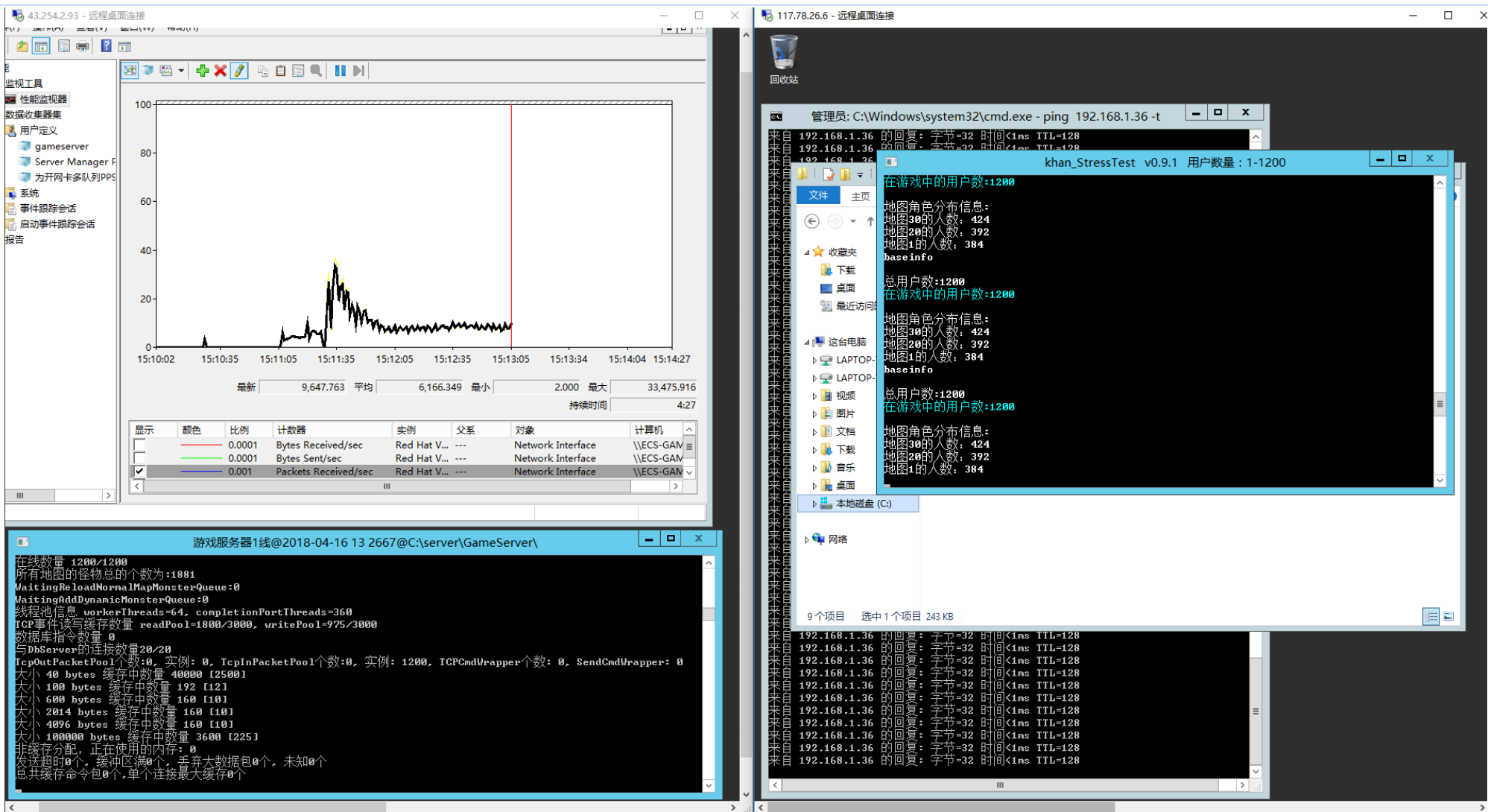
尊敬的合作伙伴，您好！  
  
感谢您的回帖。  
  
这个官方链接是公开的，请您随意。  
  
根据我的理解，默认情况是没有TcpNumConnections的，也就是它的值是16,777,214。当然，这只是理论上的TCP最大连接数。实际情况还是会受到CPU、内存等因素的影响。因为每建立一个TCP连接，都会占用一部分的资源。当资源不足时，就无法建立TCP连接。  
  
希望以上信息有所帮助。如果您有任何疑问请及时联系我。  
  
此致，  
  
Tommy Xing  
微软合作伙伴社区技术支持工程师  
微软全球合作伙伴服务

提示：  
Windows Server自带的powershell不使用find会报参数格式不正确，换成cmd才能正常使用

```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) 2014 Microsoft Corporation. 保留所有权利。
PS C:\Users\Administrator> netstat -an | find /c "ESTABLISHED"
FIND: 参数格式不正确
PS C:\Users\Administrator>
```

# Step4：确认是否虚拟存在某种约束导致不支持400+玩家登入游戏服

通过单开一个压测进程模拟1200玩家直接登录，竟然全部顺利登入游戏！  
因为带宽离内网瓶颈差别很远，我们使用Windows计数器重点观察包数量（PPS）指标，曲线图如下图所示，推测可能是业务流量叠加导致包处理能力不足；





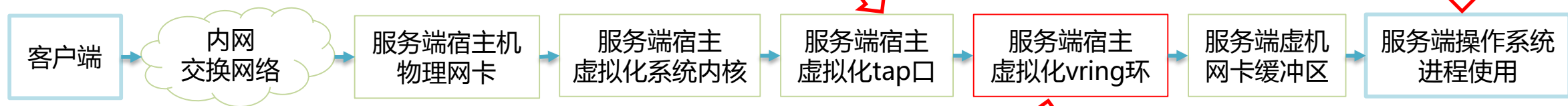
# Step5：抓包确认问题出现时丢失的数据包具体丢在哪个环节

Seq3928-3930发出去了但是无响应

Source	Destination	Protocol	Length	Info
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3926/22031, ttl=128 (reply in 132125)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3926/22031, ttl=128 (request in 132124)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3927/22287, ttl=128 (reply in 143386)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3927/22287, ttl=128 (request in 143383)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3928/22543, ttl=128 (no response found!)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3929/22799, ttl=128 (no response found!)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3930/23055, ttl=128 (no response found!)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3931/23311, ttl=128 (reply in 182271)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3931/23311, ttl=128 (request in 182270)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3932/23567, ttl=128 (reply in 188603)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3932/23567, ttl=128 (request in 188532)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3933/23823, ttl=128 (reply in 192784)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3933/23823, ttl=128 (request in 192780)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3934/24079, ttl=128 (reply in 198096)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3934/24079, ttl=128 (request in 198092)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3935/24335, ttl=128 (reply in 203233)
192.168.1.36	192.168.1.20	ICMP	74	Echo (ping) reply id=0x0001, seq=3935/24335, ttl=128 (request in 203230)

Seq3928-3930没有收到

Source	Destination	Protocol	Length	Info
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3915/19215, ttl=128 (reply in 7128)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3916/19471, ttl=128 (reply in 23980)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3917/19727, ttl=128 (reply in 35342)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3918/19983, ttl=128 (reply in 46708)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3919/20239, ttl=128 (reply in 64445)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3920/20495, ttl=128 (reply in 76104)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3921/20751, ttl=128 (reply in 90509)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3922/21007, ttl=128 (reply in 107205)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3923/21263, ttl=128 (reply in 120548)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3924/21519, ttl=128 (reply in 133996)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3925/21775, ttl=128 (reply in 152181)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3926/22031, ttl=128 (reply in 164076)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3927/22287, ttl=128 (reply in 175234)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3931/23311, ttl=128 (reply in 196168)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3932/23567, ttl=128 (reply in 202500)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3933/23823, ttl=128 (reply in 206681)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3934/24079, ttl=128 (reply in 211993)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3935/24335, ttl=128 (reply in 217130)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3936/24591, ttl=128 (reply in 232698)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3937/24847, ttl=128 (reply in 243673)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3938/25103, ttl=128 (reply in 246703)
192.168.1.20	192.168.1.36	ICMP	74	Echo (ping) request id=0x0001, seq=3939/25359, ttl=128 (reply in 249216)



通过后台发现vring环提供的缓冲区大小2048个包，在问题现象出现时缓冲区被写满溢出，造成丢包  
问题发生的点已定位明确，需要制定优化方案



# 优化操作1：开启网卡多队列支持，提升包处理速度

通过确认Windows Server 2012 R2默认开启网卡多队列支持，因此通过复制公共镜像做一个私有镜像，并将“网卡多队列”设置为“支持”  
由此镜像创建的虚拟机，在虚拟化层将支持操作系统进行网卡多队列读写（提供8个vring环），虚拟机的网络处理能力将提升6倍左右！  
*提醒：如果操作系统自身不支持网卡多队列，请勿在镜像中开启此功能*

网络和共享中心

以太网 2 属性

以太网 2 状态

Red Hat VirtIO Ethernet Adapter 属性

修改镜像

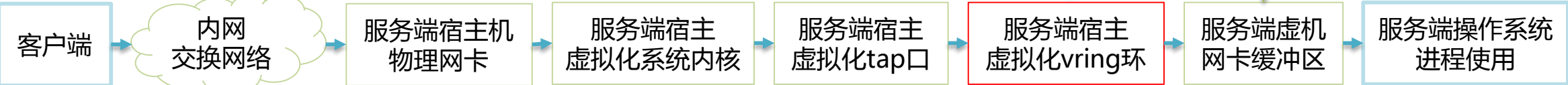
名称: test-windows-net

描述: 测试Windows网卡多队列

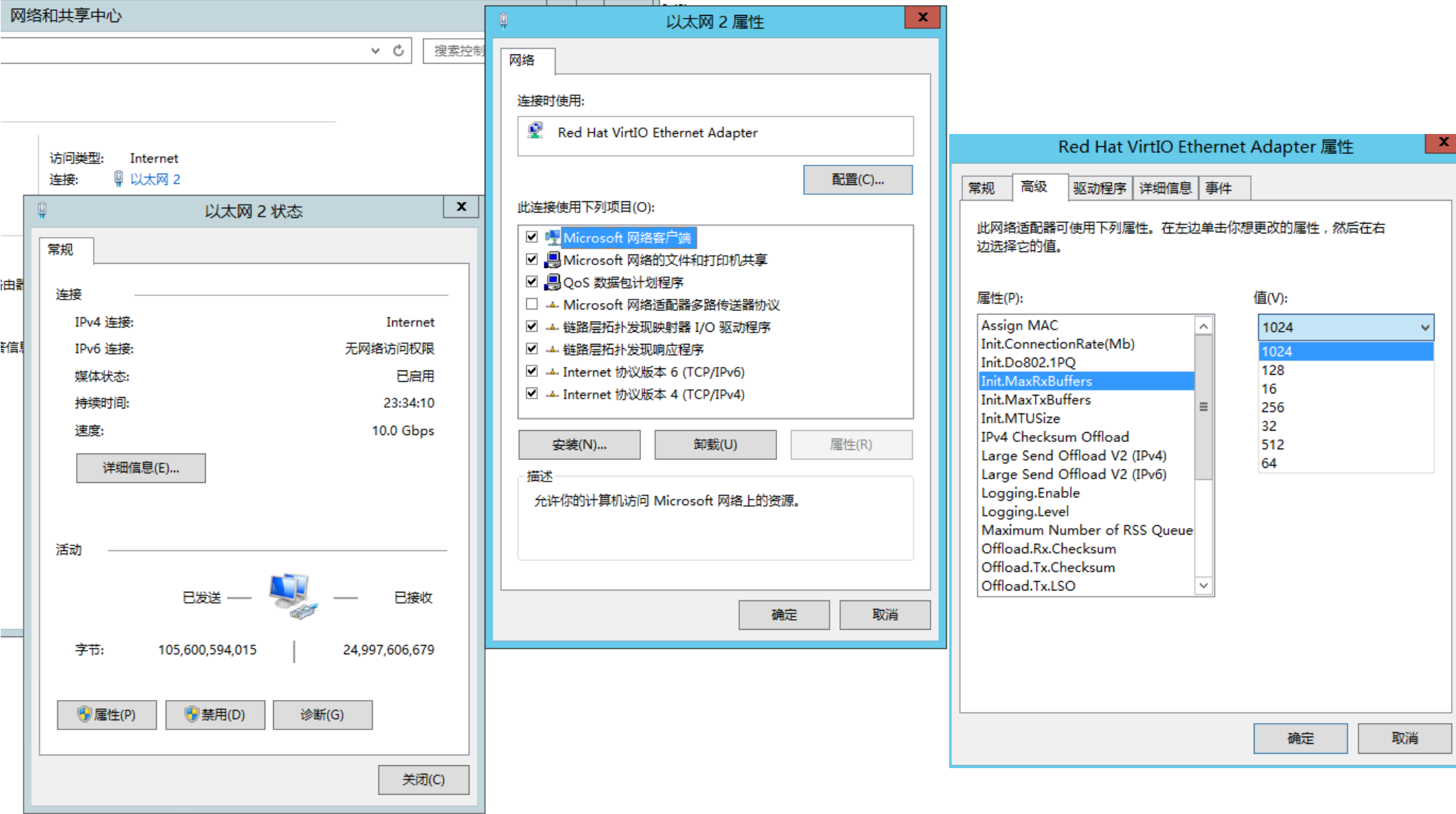
最小内存: 0MB, 1GB, 2GB, 4GB, 8GB, 16GB, 32GB, 64GB, 128GB

网卡多队列: 支持

网络收发能力  
提升约6倍

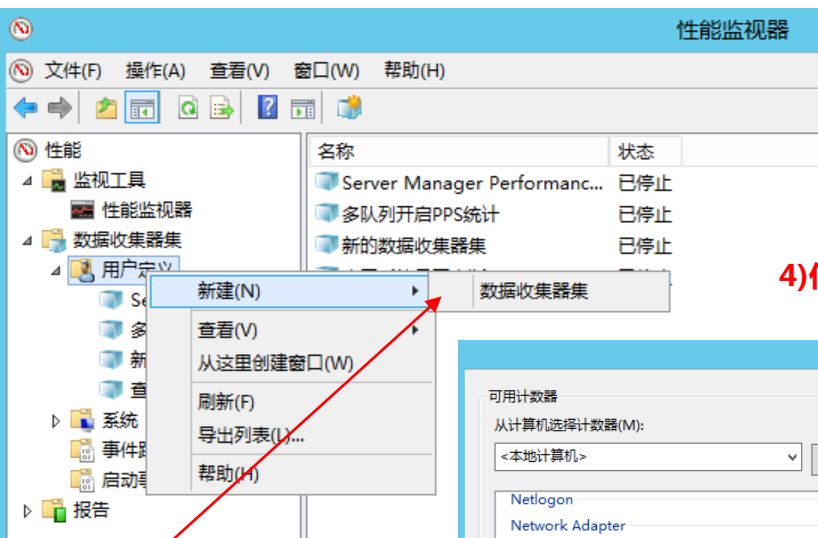


# 优化操作2：调高网卡驱动的收包缓冲区大小，提升高并发应对能力

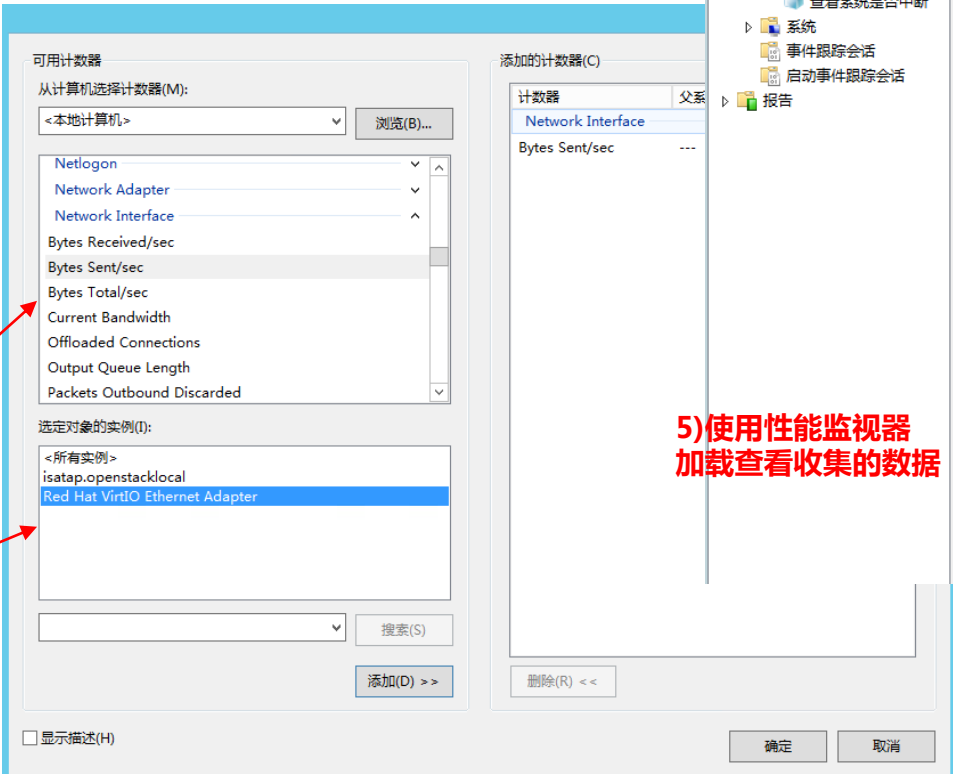


# 工具使用：Windows性能监视器（计数器）

Windows Server可以通过“运行>输入perfmon”调出性能监视器，以下介绍基本使用方法，如果还需要更多信息，可以google搜索进一步了解

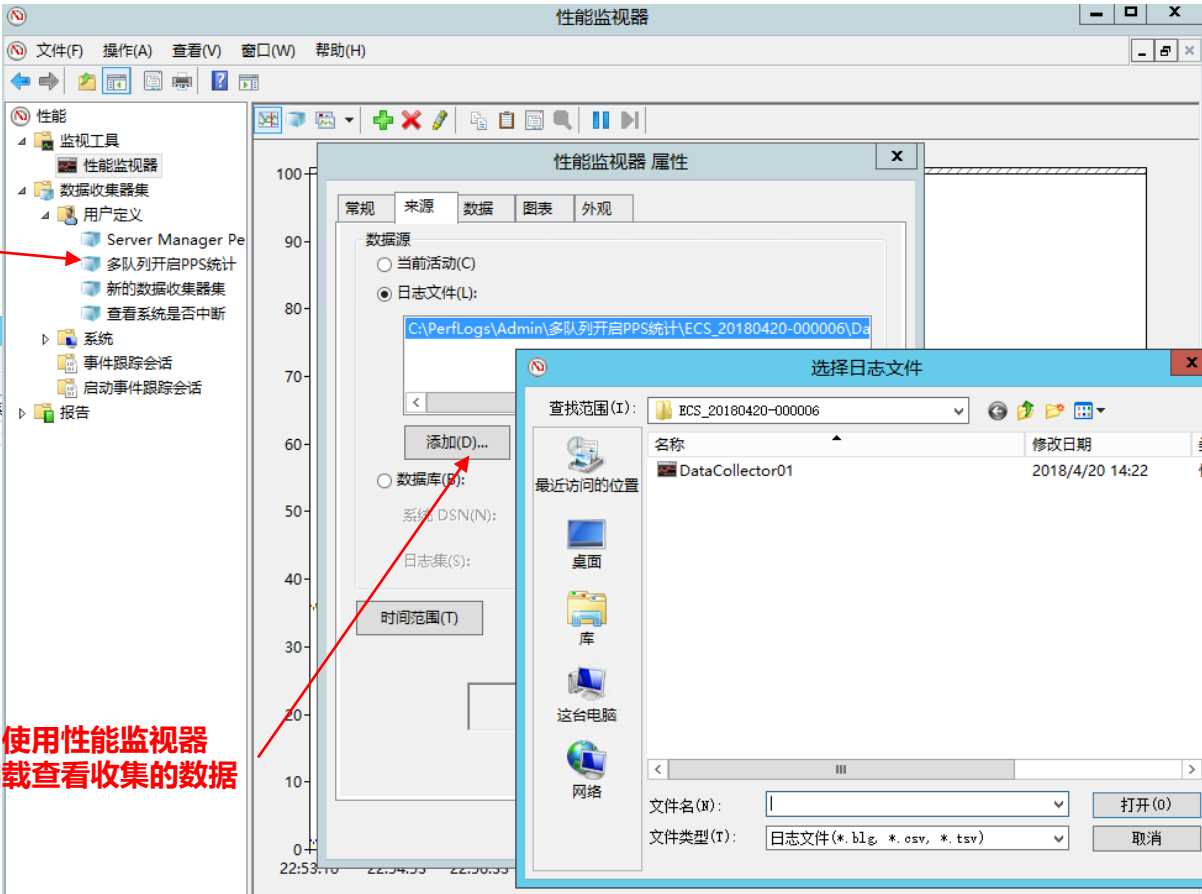


1) 用户定义新建数据收集器



2) 选择需要观察的指标项

3) 选择对应实例并添加



4) 任务上右键启动数据收集

5) 使用性能监视器加载查看收集的数据