

数据库性能优化文档

情况描述:

平台信息: ZStack 云平台 4.1.6, 物理机系统 CentOS7.6, 分布式存储计算存储部署。

物理机信息: Intel 6258R, 单颗 CPU 二十八核心, 2 颗 48 个核心, 96 个线程, 该节点只运行一个 96 核 512G 的数据库云主机。

存储信息: 存储万兆网络, 采用纯 SSD 作为存储介质。

网络信息: 业务使用双千兆网络。建议使用双万兆网络

数据库业务: Windows 2016 云主机运行 SQL server。

一: 业务卡顿的具体表现:

数据库业务高峰期 CPU 利用率在 100%运行, 云主机互 ping 网络延迟会高, 会存在丢包, 丢包比例达 0.3%。导致业务卡顿, 影响生产业务。

二: 性能分析:

1.数据库业务高峰期的 CPU 利用率持续在 100%高位运行, 物理机的 CPU 利用率也会同时会提升到 98%。

2.而当时分布式存储 IO 写的延迟最高在 2ms, 存储网络写带宽最高在 116MB/s。云主机业务网络的带宽峰值在 65MB/s。可见存储层面不是整体的瓶颈。网络带宽也不是业务的瓶颈。但 CPU 利用率高后, 导致 CPU 处理网络相关中断出现异常, 存在丢包和延迟问题。瓶颈本身整体还是在 CPU 和内存层面。

3.数据库业务存在全表查询, 会大量占用资源。

三.云平台层面相关的优化配置：

1: Virtio 平台类型：

Windows 云主机系统支持使用 Virtio 来优化磁盘 IO 性能；
在云主机的详情页的云盘界面进行 virtioscsi 的开启和关闭，需要安装性能优化工具，重启虚拟机开启 virtio 的功能，然后更改磁盘的模式。



2: Virtio blk:

Windows 云主机云平台使用 Virtio blk 类型，不使用 Virtio scsi，Virtio blk 类型云盘在效率和性能较 Virtio scsi 略高；



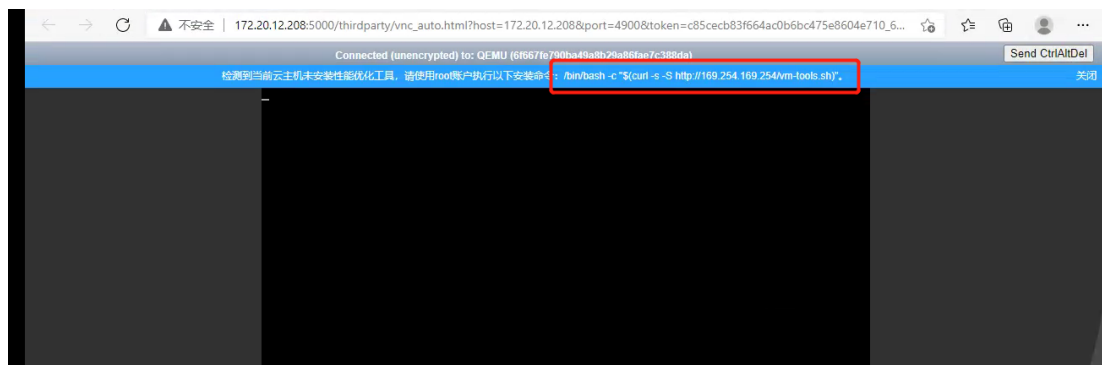
根据需要进行数据盘和系统盘的调整，系统盘将这个 virtioscsi 进行关闭，数据盘进行开启。

3: Virtio 驱动:

使用最新的 Windows Virtio 驱动，以提升 IO 性能，Windows virtio 底层对应的位置。



在 ui 界面上，打开 linux 的云主机的 vnc 界面会有提示安装性能优化工具，linux 需要输入命令行进行性能优化工具的安装

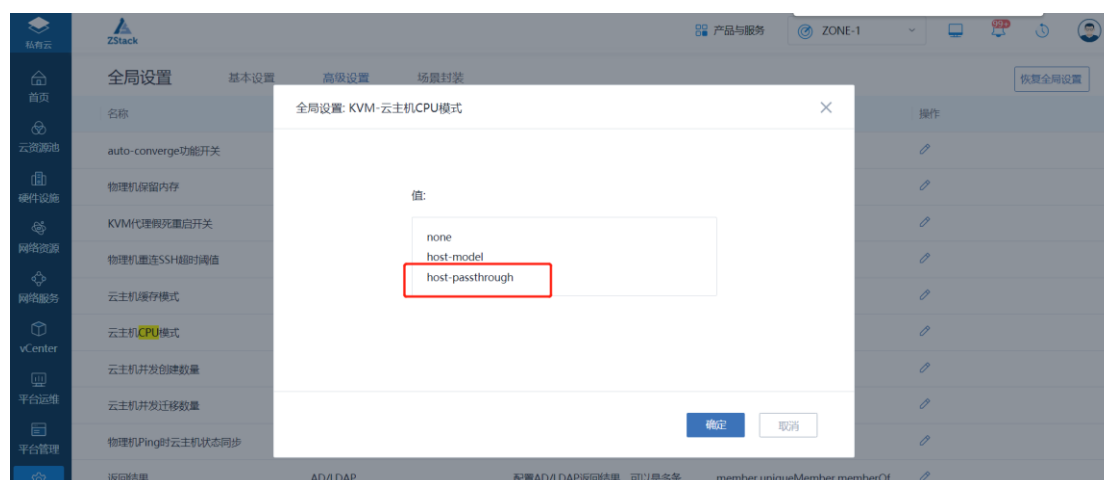


Windows 虚拟机点击安装性能优化工具，然后在虚机的此电脑里面进行驱动的安装，安装完以后重启物理机。

4: CPU 模式:

云主机开启 CPU 模式设置为 host-passthrough，以支持数据库业务对特殊指令集的要求；

全局设置里面根据需要选择 cpu 的模式，一共支持三种模式，host-passthrough host-model none 模式，对于性能优化采用 host-passthrough 模式，或者 host-model 模式



5:CPU 数量:

云主机的 CPU 数量并非越高越好，数量不合理会导致 CPU 异常使用率高。最佳实践：关闭物理机超线程，单个云主机的 CPU 上限不超过单个 NUMA 节点 CPU 的数量；Cpu 数量建议是根据单个 numa 节点的数量进行对虚机的优化和更改。

```
[root@172-20-12-208 x86_64]# cd c76/
[root@172-20-12-208 c76]# ls
docs          isalinux      product_name  script_bins   vmware-vix-disklib-distrib-101230.tar.gz
EFI           ks.cfg        qemu-system-x86_64-20200813.tar.gz  zs_drivers
elasticsearch-rhel6-rhel7-12.0.193.22-1.tgz  liveOS        repodata      tools          zstack-image-1.4.qcow2
Extra         MicroCore-Linux.ova  repos         upgrade_repo.sh  zstack-installer.bin
GPU           nbdkit-1.14.2.tar.gz  RPM-GPG-KEY-CentOS-7  VERSION         zstack-windows-virtio-driver.iso
Images        Packages      RPM-GPG-KEY-CentOS-Testing-7  VMware-vix-disklib-5.5.0-1284542.x86_64.tar.gz

[root@172-20-12-208 c76]# pwd
/opt/zstack-dvd/x86_64/c76
[root@172-20-12-208 c76]# lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 2
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 79
Model name: Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
Stepping: 1
CPU MHz: 2200.194
BogoMIPS: 4400.38
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 4995K
cflags: -march=x86-64 -mtune=generic -O2 -fcommon -fPIE -pie -fstack-protector-strong -fstack-clash-protection -fcf-protection
numa node0 CPU(s): 0-7
flags: fpu_vmmde_pae tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpebgh rdtscp lm constant tsc arch.perfmon rep_good nopl xtopology eap
erfpu pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch tpr_shadow vmmi flexpriority ept vpid fsgbase ts
c_adjust bmi1 bti avx2 smap bmi2 erms invpcid rtm rdtseed adx smap xsaveopt arat
[root@172-20-12-208 c76]#
```

这个例子中通过 lscpu 的数量看到单个 numa 的 cpu 数量是 8，因此建议虚机的时候可以根据 numa 的数量进行合理的创建虚机。将虚机使用的 cpu 数量和底层的 numa 的 cpu 可以进行一对一绑定，不出现跨 numa 的现象。

6: CPU Pin 绑定:

云主机的 VCPU 绑定到物理的 CPU，减少中断在 CPU 之间的切换调度，注意：物理机的 CPU 0 一般处理中断、虚拟化等任务，不建议将其 Pin 到云主机上；

Cpu 的绑定, 根据底层物理机 numa 结构进行绑定。比如我底层一个 numa 有 8 个 cpu，可以将其中 7 个 cpu 进行和虚机的 cpu 绑定，物理机 cpu 从 1 开始进行绑定，虚机 cpu 从 0 开始进行绑定，直到整个 numa 用完。

```
[root@172-20-12-208 x86_64]# cd c76/
[root@172-20-12-208 c76]# ls
docs          isalinux      product_name  script_bins   vmware-vix-disklib-distrib-101230.tar.gz
EFI           ks.cfg        qemu-system-x86_64-20200813.tar.gz  zs_drivers
elasticsearch-rhel6-rhel7-12.0.193.22-1.tgz  liveOS        repodata      tools          zstack-image-1.4.qcow2
Extra         MicroCore-Linux.ova  repos         upgrade_repo.sh  zstack-installer.bin
GPU           nbdkit-1.14.2.tar.gz  RPM-GPG-KEY-CentOS-7  VERSION         zstack-windows-virtio-driver.iso
Images        Packages      RPM-GPG-KEY-CentOS-Testing-7  VMware-vix-disklib-5.5.0-1284542.x86_64.tar.gz

[root@172-20-12-208 c76]# pwd
/opt/zstack-dvd/x86_64/c76
[root@172-20-12-208 c76]# lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 2
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 79
Model name: Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
Stepping: 1
CPU MHz: 2200.194
BogoMIPS: 4400.38
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 4995K
cflags: -march=x86-64 -mtune=generic -O2 -fcommon -fPIE -pie -fstack-protector-strong -fstack-clash-protection -fcf-protection
numa node0 CPU(s): 0-7
flags: fpu_vmmde_pae tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpebgh rdtscp lm constant tsc arch.perfmon rep_good nopl xtopology eap
erfpu pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch tpr_shadow vmmi flexpriority ept vpid fsgbase ts
c_adjust bmi1 bti avx2 smap bmi2 erms invpcid rtm rdtseed adx smap xsaveopt arat
[root@172-20-12-208 c76]#
```



7: VNUMA 配置:

NUMA 架构下，多路 CPU 的跨 CPU 访问内存效率会降低，配置 VNUMA 结构后，使云主机在 CPU 访问内存时不跨物理 CPU；

1，获取 xml 文件

使用的虚机是一个 8C16G 的虚拟机，底层的物理机的 numa 结构是 4 个 numa 结构。在 sqlserver 所在的物理机上执行（需要在云主机启动状态下才能 dumpxml）

```
virsh dumpxml 9a1cb866f3e4401e8acc93e9922a333e >> 9a1cb866f3e4401e8acc93e9922a333e.xml
```

```
[root@duruimn-3 ~]# virsh list
-----
 Id         Name                                     State
-----
 4          9a1cb866f3e4401e8acc93e9922a333e      running

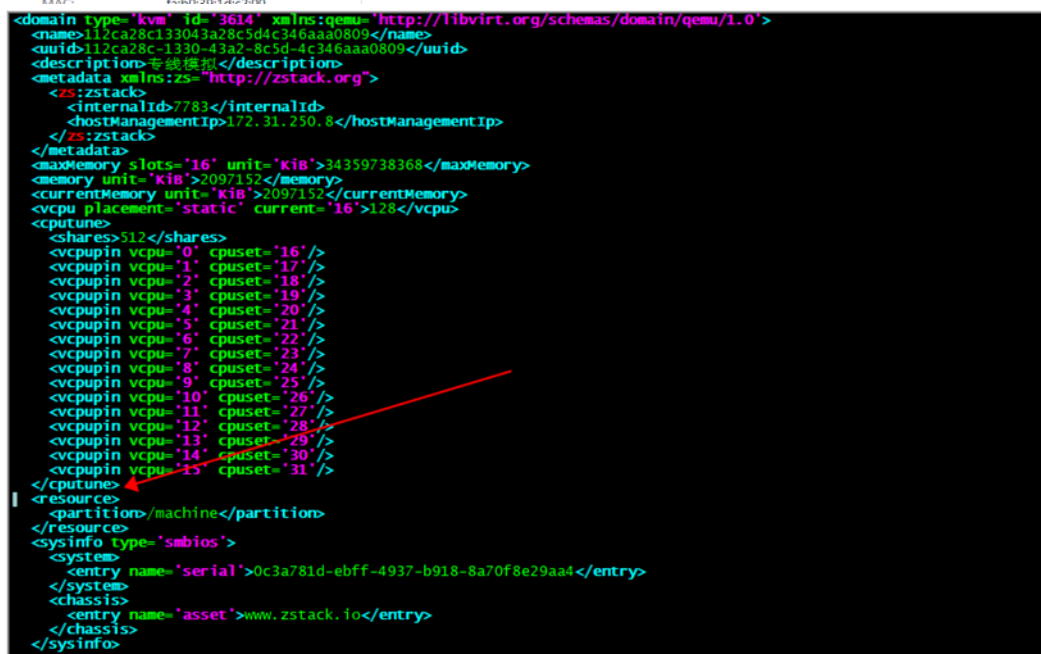
[root@duruimn-3 ~]#
[root@duruimn-3 ~]# virsh dumpxml 9a1cb866f3e4401e8acc93e9922a333e >> 9a1cb866f3e4401e8acc93e9922a333e.xml
[root@duruimn-3 ~]#
```

2，编辑 xml 文件

```
[root@duruimn-3 ~]# vim 9a1cb866f3e4401e8acc93e9922a333e.xml
```

打开的 xml 文件如图所示，

再云平台进行 cpu 的绑定，之后底层的 xml 文件会有 cpu 绑定的内容。



3.在箭头处编辑加入如下参数将 numa 结构透传给虚拟机

加入参数后,如下图所示

底层物理机是有 4 个 numa, 因此 vnuma 建议和物理机底层 numa 结构保持一致。
如果只有一个 numa 结构, 不用设置 vnuma 结构

```
<numatune>
  <memory mode='strict' nodeset='4-7'/>
  <memnode cellid='0' mode='strict' nodeset='4'/>
  <memnode cellid='1' mode='strict' nodeset='5'/>
  <memnode cellid='2' mode='strict' nodeset='6'/>
  <memnode cellid='3' mode='strict' nodeset='7'/>
</numatune>
```

```

<domain type='kvm' id='3614' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <name>112ca28c133043a28c5d4c346aaa0809</name>
  <uuid>112ca28c-1330-43a2-8c5d-4c346aaa0809</uuid>
  <description>专线模拟</description>
  <metadata xmlns:zs='http://zstack.org'>
    <zs:zstack>
      <internalId>7783</internalId>
      <hostManagementIp>172.31.250.8</hostManagementIp>
    </zs:zstack>
  </metadata>
  <maxMemory slots='16' unit='KiB'>34359738368</maxMemory>
  <memory unit='KiB'>2097152</memory>
  <currentMemory unit='KiB'>2097152</currentMemory>
  <vcpu placement='static' current='16'>128</vcpu>
  <cpupin>
    <shares>512</shares>
    <vcupin vcpu='0' cpuset='16' />
    <vcupin vcpu='1' cpuset='17' />
    <vcupin vcpu='2' cpuset='18' />
    <vcupin vcpu='3' cpuset='19' />
    <vcupin vcpu='4' cpuset='20' />
    <vcupin vcpu='5' cpuset='21' />
    <vcupin vcpu='6' cpuset='22' />
    <vcupin vcpu='7' cpuset='23' />
    <vcupin vcpu='8' cpuset='24' />
    <vcupin vcpu='9' cpuset='25' />
    <vcupin vcpu='10' cpuset='26' />
    <vcupin vcpu='11' cpuset='27' />
    <vcupin vcpu='12' cpuset='28' />
    <vcupin vcpu='13' cpuset='29' />
    <vcupin vcpu='14' cpuset='30' />
    <vcupin vcpu='15' cpuset='31' />
  </cpupin>
  <numatune>
    <memory mode='strict' nodeset='4-7' />
    <memnode cellid='0' mode='strict' nodeset='4' />
    <memnode cellid='1' mode='strict' nodeset='5' />
    <memnode cellid='2' mode='strict' nodeset='6' />
    <memnode cellid='3' mode='strict' nodeset='7' />
  </numatune>
  <resource>
    <partition>/machine</partition>
  </resource>
  <osinfo type='rhel' />

```

4,在红圈处将 mode=""host-passthrough"改为 mode=""host-model",
箭头处加入 numa 内存绑定

这个地方根据上面绑定的设置的 vnuma 的个数进行配置, 16C, 120G 的内存, 每个 vnuma4 个 cpu, 内存是 30G 的容量。

```

<numa>
<cell id='0' cpus='0-3' memory='30' unit='GiB' />
<cell id='1' cpus='4-7' memory='30' unit='GiB' />
<cell id='2' cpus='8-11' memory='30' unit='GiB' />
<cell id='3' cpus='12-15' memory='30' unit='GiB' />
</numa>

```

修改 cpu mode 并加入参数后,如下图所示

```

<cpu mode='host-passthrough' check='none'>
  <topology sockets='32' cores='4' threads='1' />
</cpu>
<clock offset='utc' />
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>

```

```
<cpu mode='host-model' check='none'>
  <topology sockets='2' cores='8' threads='1' />
  <numa>
    <cell id='0' cpus='0-3' memory='30' unit='GiB' />
    <cell id='1' cpus='4-7' memory='30' unit='GiB' />
    <cell id='2' cpus='8-11' memory='30' unit='GiB' />
    <cell id='3' cpus='12-15' memory='30' unit='GiB' />
  </numa>
```

5: ssh 登陆到 sqlsever 云主机所在的物理机上，使用 virsh create 9a1cb866f3e4401e8acc93e9922a333e.xml 即可完成调优。

8: 物理机设置 isolcpus:

将运行业务云主机的相关物理 CPU 预留，将其隔离出来，底层物理机操作系统不可使用，单独配置给业务云主机使用；

在需要优化的云主机所在物理机 shell 中操作，修改 grub 隔离 cpu，（需重启物理机生效）

```
[root@build ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=zstack/root rd.lvm.lv=zstack/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

vim /etc/default/grub，比如在 quiet 后面添加 `isolcpus=16-31`

修改后如图所示

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=zstack/root rhgb quiet isolcpus=16-31"
GRUB_DISABLE_RECOVERY="true"
```

然后执行：grub2-mkconfig -o /boot/grub2/grub.cfg

重启物理机，等待物理机启动完成，查看 cpu 隔离生效，cat /proc/cmdline

9: 大页内存

使用 2M 的大页内存，替换默认的 4K 页面，减少云主机页面缓存和缺页中断的切换，提高运行性能。注意：大页内存需配合物理机的保留内存使用，为操作系统相关的基础业务保留必要的内存；

本地存储和 san 存储保留 16g 的内存给物理机使用。保证系统运行服务正常。

超融合环境，内存的分配根据 top 展示的内存的分配进行容量保留，一个 osd 要保留 5G 的内存，因此保留内存是 16G+5G*osd 的数量。



10: 网卡多队列

支持 Virtio 类型的网卡流量分配给多个 CPU, 并行处理中断, 虚拟机网卡多队列, 减轻虚拟机 cpu0 的压力, 减少网络的延迟。可以再集群的高级设置里面进行多队列数目的调整。调整 4C 8C 16C 即可。



11：关闭内存 KSM：关闭物理机内存同页合并技术，避免相同内存页面被共用

```
ksm.service                                ksmtuned.service
[root@172-20-12-208 ~]# systemctl status ksm.service
● ksm.service - Kernel Samepage Merging
   Loaded: loaded (/usr/lib/systemd/system/ksm.service; enabled; vendor preset: enabled)
   Active: active (exited) since Fri 2021-08-20 15:13:37 CST; 1h 10min ago
     Main PID: 1953 (code=exited, status=0/SUCCESS)
       Tasks: 0
      CGroup: /system.slice/ksm.service

Aug 20 15:13:36 172-20-12-208 systemd[1]: Starting Kernel Samepage Merging...
Aug 20 15:13:37 172-20-12-208 systemd[1]: Started Kernel Samepage Merging.
[root@172-20-12-208 ~]#
```

使用 systemctl stop ksm.service

Systemctl disable ksm.service

进行 ksm 的关闭

12：关闭 CPU 节能模式：

物理机在 BIOS 层面关闭 CPU 节能模式，以性能模式运行，这个再物理机的 bios 界面找到 advance 高级设置里面 cpu 的选项将 cpu c state 进行关闭。

13：升级物理机内核：

更高的内核在系统层面性能会提升，对云主机相关性能有一定提升，内核版本使用的是以下内核，有需要可以联系我们进行获取。新的内核版本优化了网络相关的一些内容。

https://kernel-uek-5.4.17-2011.4.6.el7uek.x86_64.rpm

四：针对此数据库场景做的优化：

针对以上可配置的优化, 基于便捷及高效的原则, 逐步优化, 针对上述业务进行了以下变更：

1.Virtio 平台类型、Virtio blk 云盘、Virtio 驱动配置；

2.CPU 模式配置为 host-passthrough；

3.整体 CPU 数量配置为 84 个，CPU 配置了 VNUMA 模式，并配置了 CPU Pin 绑定；

4.物理机配置了大页内存，云主机使用大页内存；

4.云主机网卡开启多队列模式；

6.物理机在 BIOS 关闭了节能模式，更新了内核版本；

7.物理机 grub 设置了 isolcpus 的 CPU 隔离；

8.物理机关闭了 KSM，避免后续云主机的内存同页合并。

配置完毕，云主机的 CPU 在业务高峰期使用平均下降到 40%左右，云主机 ping 延迟已恢复正常，网络不再丢包。

上述的云平台在数据库层面的优化实践，可以在相关 CPU 压力较大的业务场景可借鉴参考优化。针对优化的效率，一般而言，

Virtio 对性能的优化最高，其次是 VNUMA 的正确调度，接着是大页内存，再次是关闭 KSM、物理机更新内核、CPU Pin 影响效果差不多，最后是 isolcpus 的 CPU 隔离及其他相关优化。