# Linux内核参数

# 目录

Linux内核参数 1

1.abi 9

2.crypto 10

3.debug 10

4.dev 10

5. Fs-文件系统参数 11

fs.inode-nr = 157067 55781 11

fs.inode-state = 157067 55781 0 0 0 0 0 11

fs.file-nr = 3584 0 1618273 11

fs.file-max = 1618273 11

fs.nr\_open = 1048576 11

fs.dentry-state = 112240 99795 45 0 0 0 11

fs.overflowuid = 65534 12

fs.overflowgid = 65534 12

fs.dir-notify-enable = 1 12

fs.leases-enable = 1 12

fs.lease-break-time = 45 12

fs.aio-nr = 0 12

fs.aio-max-nr = 65536 12

fs.inotify.max\_user\_instances = 128 13

fs.inotify.max\_user\_watches = 8192 13

fs.inotify.max\_queued\_events = 16384 13

fs.epoll.max\_user\_watches = 3335680 13

fs.binfmt\_misc.status = enabled 13

fs.mqueue.queues\_max = 256 13

fs.mqueue.msg\_max = 10 14

fs.mqueue.msgsize\_max = 8192 14

fs.mqueue.msg\_default = 10 14

fs.mqueue.msgsize\_default = 8192 14

fs.nfs.nlm\_grace\_period = 0 14

fs.nfs.nlm\_timeout = 10 14

fs.nfs.nlm\_udpport = 0 14

fs.nfs.nlm\_tcpport = 0 14

fs.nfs.nsm\_use\_hostnames = 0 15

不知道 15

fs.nfs.nsm\_local\_state = 0 15

fs.nfs.nfs\_callback\_tcpport = 0 15

fs.nfs.idmap\_cache\_timeout = 600 15

fs.nfs.nfs\_mountpoint\_timeout = 500 15

fs.nfs.nfs\_congestion\_kb = 129216 15

6.Kernel-内核参数 15

kernel.sched\_child\_runs\_first = 0 15

kernel.sched\_min\_granularity\_ns = 4000000 15

kernel.sched\_latency\_ns = 20000000 16

kernel.sched\_wakeup\_granularity\_ns = 4000000 16

kernel.sched\_tunable\_scaling = 1 16

kernel.sched\_features = 3183 16

kernel.sched\_migration\_cost = 500000 16

kernel.sched\_nr\_migrate = 32 17

kernel.timer\_migration = 1 17

kernel.sched\_rt\_period\_us = 1000000 17

kernel.sched\_rt\_runtime\_us = 950000 17

kernel.sched\_compat\_yield = 0 17

kernel.sched\_rr\_timeslice\_ms = 100 17

kernel.sched\_autogroup\_enabled = 0 17

kernel.sched\_cfs\_bandwidth\_slice\_us = 5000 17

kernel.panic = 0 18

kernel.panic\_on\_oops = 1 18

kernel.panic\_on\_warn = 0 18

kernel.core\_uses\_pid = 1 18

kernel.core\_pattern=/opt/applog/coredir/core-%e-%p-%h-%t 18

kernel.core\_pipe\_limit = 0 18

kernel.tainted = 0 18

kernel.real-root-dev = 0 19

kernel.print-fatal-signals = 0 19

kernel.ctrl-alt-del = 0 19

kernel.ftrace\_dump\_on\_oops = 0 19

kernel.modprobe = /sbin/modprobe 19

kernel.modules\_disabled = 0 19

kernel.kexec\_load\_disabled = 0 19

kernel.hotplug = 20

kernel.acct = 4 2 30 20

kernel.sysrq = 0 20

kernel.threads-max = 127246 20

kernel.random.poolsize = 4096 20

kernel.random.entropy\_avail = 193 20

kernel.random.read\_wakeup\_threshold = 64 20

kernel.random.write\_wakeup\_threshold = 128 20

kernel.random.boot\_id=ad2889a9-2f17-4eb7-8b25-30e48c8a6451 21

kernel.random.uuid=48a82b76-296a-44a0-8adb-8f54de384356 21

kernel.overflowuid = 65534 21

kernel.overflowgid = 65534 21

kernel.pid\_max = 131072 21

kernel.printk = 4 4 1 7 21

kernel.printk\_ratelimit = 5 21

kernel.printk\_ratelimit\_burst = 10 22

kernel.printk\_delay = 0 22

kernel.dmesg\_restrict = 0 22

kernel.kptr\_restrict = 1 22

kernel.ngroups\_max = 65536 22

kernel.cap\_last\_cap = 33 22

kernel.watchdog = 1 22

kernel.watchdog\_thresh = 60 23

kernel.softlockup\_panic = 0 23

kernel.nmi\_watchdog = 1 23

kernel.randomize\_va\_space = 2 23

kernel.hung\_task\_panic = 0 24

kernel.hung\_task\_check\_count = 4194304 24

kernel.hung\_task\_timeout\_secs = 120 24

kernel.hung\_task\_warnings = 10 24

kernel.compat-log = 1 24

kernel.max\_lock\_depth = 1024 24

kernel.poweroff\_cmd = /sbin/poweroff 24

\*\*kernel.keys.root\_maxkeys = 1000000 25

\*\*kernel.keys.root\_maxbytes = 25000000 25

\*\*kernel.keys.maxkeys = 20 25

\*\*kernel.keys.maxbytes = 20000 25

\*\*kernel.keys.gc\_delay = 300 25

kernel.keys.persistent\_keyring\_expiry 25

kernel.perf\_event\_paranoid = 1 25

kernel.perf\_event\_mlock\_kb = 516 25

kernel.perf\_event\_max\_sample\_rate = 100000 25

kernel.cad\_pid 26

kernel.blk\_iopoll = 1 26

kernel.ostype = Linux 28

kernel.osrelease = 2.6.32-573.35.2.el6.x86\_64 28

kernel.version = #1 SMP Mon Oct 24 14:14:01 EDT 2016 28

kernel.hostname = vm-vmw51102-jbs 28

kernel.domainname = (none) 28

kernel.pty.max = 4096 29

kernel.pty.nr = 3 29

kernel.shmmax = 68719476736 29

kernel.shmall = 4294967296 29

kernel.shmmni = 4096 29

kernel.shm\_rmid\_forced = 0 29

kernel.msgmax = 65536 29

kernel.msgmni = 32768 29

kernel.msgmnb = 65536 29

kernel.sem = 250 32000 32 128 30

kernel.auto\_msgmni = 1 30

以下几个是生产服务器上没有（或当前账号无法查看的） 30

7.net-网络参数 31

net.core.somaxconn = 128 31

net.core.wmem\_max = 124928 31

net.core.rmem\_max = 124928 31

net.core.wmem\_default = 124928 32

net.core.rmem\_default = 124928 32

net.core.dev\_weight = 64 32

net.core.netdev\_max\_backlog = 1000 32

net.core.message\_cost = 5 32

net.core.message\_burst = 10 32

net.core.optmem\_max = 20480 32

net.core.busy\_poll = 0 32

net.core.busy\_read = 0 33

net.core.netdev\_budget = 300 33

net.core.warnings = 1 33

net.ipv4.route.gc\_thresh = 524288 33

net.ipv4.route.max\_size = 8388608 33

net.ipv4.route.gc\_min\_interval = 0 33

net.ipv4.route.gc\_min\_interval\_ms = 500 33

net.ipv4.route.gc\_timeout = 300 33

net.ipv4.route.gc\_interval = 60 33

net.ipv4.route.redirect\_load = 20 34

net.ipv4.route.redirect\_number = 9 34

net.ipv4.route.redirect\_silence = 20480 34

net.ipv4.route.error\_cost = 1000 34

net.ipv4.route.error\_burst = 5000 34

net.ipv4.route.gc\_elasticity = 8 34

net.ipv4.route.mtu\_expires = 600 35

net.ipv4.route.min\_pmtu = 552 35

net.ipv4.route.min\_adv\_mss = 256 35

net.ipv4.neigh.default.mcast\_solicit = 3 35

net.ipv4.neigh.default.ucast\_solicit = 3 35

net.ipv4.neigh.default.app\_solicit = 0 35

net.ipv4.neigh.default.retrans\_time = 99 35

net.ipv4.neigh.default.retrans\_time\_ms = 1000 35

net.ipv4.neigh.default.base\_reachable\_time = 30 36

net.ipv4.neigh.default.base\_reachable\_time\_ms=30000 36

net.ipv4.neigh.default.delay\_first\_probe\_time = 5 36

net.ipv4.neigh.default.gc\_stale\_time = 60 36

net.ipv4.neigh.default.unres\_qlen = 3 36

net.ipv4.neigh.default.proxy\_qlen = 64 36

net.ipv4.neigh.default.anycast\_delay = 99 36

net.ipv4.neigh.default.proxy\_delay = 79 36

net.ipv4.neigh.default.locktime = 99 37

net.ipv4.neigh.default.gc\_interval = 30 37

net.ipv4.neigh.default.gc\_thresh1 = 128 37

net.ipv4.neigh.default.gc\_thresh2 = 512 37

net.ipv4.neigh.default.gc\_thresh3 = 1024 37

net.ipv4.tcp\_timestamps = 1 38

net.ipv4.tcp\_window\_scaling = 1 38

net.ipv4.tcp\_sack = 1 39

net.ipv4.tcp\_retrans\_collapse = 1 39

net.ipv4.ip\_default\_ttl = 64 39

net.ipv4.ip\_nonlocal\_bind = 0 39

net.ipv4.tcp\_syn\_retries = 5 39

net.ipv4.tcp\_synack\_retries = 5 39

net.ipv4.tcp\_max\_orphans = 262144 40

net.ipv4.ip\_dynaddr = 0 40

net.ipv4.tcp\_keepalive\_time = 7200 40

net.ipv4.tcp\_keepalive\_probes = 9 40

net.ipv4.tcp\_keepalive\_intvl = 75 40

net.ipv4.tcp\_retries1 = 3 40

net.ipv4.tcp\_retries2 = 15 40

net.ipv4.tcp\_fin\_timeout = 60 41

net.ipv4.tcp\_syncookies = 1 41

net.ipv4.tcp\_max\_tw\_buckets = 262144 41

net.ipv4.tcp\_tw\_recycle = 0 41

net.ipv4.tcp\_tw\_reuse = 0 41

net.ipv4.tcp\_abort\_on\_overflow = 0 42

net.ipv4.tcp\_stdurg = 0 42

net.ipv4.tcp\_rfc1337 = 0 42

net.ipv4.tcp\_max\_syn\_backlog = 2048 42

net.ipv4.ip\_local\_port\_range = 32768 60999 42

net.ipv4.igmp\_max\_memberships = 20 42

net.ipv4.igmp\_max\_msf = 10 42

net.ipv4.inet\_peer\_threshold = 65664 43

net.ipv4.inet\_peer\_minttl = 120 43

net.ipv4.inet\_peer\_maxttl = 600 43

net.ipv4.tcp\_orphan\_retries = 0 43

net.ipv4.tcp\_fack = 1 43

net.ipv4.tcp\_reordering = 3 43

net.ipv4.tcp\_ecn = 2 44

net.ipv4.tcp\_dsack = 1 44

net.ipv4.tcp\_mem = 1526880 2035840 3053760 44

net.ipv4.tcp\_wmem = 4096 16384 4194304 45

net.ipv4.tcp\_rmem = 4096 87380 4194304 45

net.ipv4.tcp\_app\_win = 31 45

net.ipv4.tcp\_adv\_win\_scale = 2 45

net.ipv4.tcp\_low\_latency = 0 45

net.ipv4.tcp\_no\_metrics\_save = 0 46

net.ipv4.tcp\_moderate\_rcvbuf = 1 46

net.ipv4.tcp\_tso\_win\_divisor = 3 46

net.ipv4.tcp\_congestion\_control = cubic 46

net.ipv4.tcp\_mtu\_probing = 0 46

net.ipv4.tcp\_base\_mss = 512 46

net.ipv4.tcp\_workaround\_signed\_windows = 0 46

net.ipv4.tcp\_slow\_start\_after\_idle = 1 47

net.ipv4.cipso\_cache\_enable = 1 47

net.ipv4.cipso\_cache\_bucket\_size = 10 47

net.ipv4.cipso\_rbm\_optfmt = 0 47

net.ipv4.cipso\_rbm\_strictvalid = 1 47

net.ipv4.tcp\_available\_congestion\_control = cubic reno 47

net.ipv4.tcp\_allowed\_congestion\_control = cubic reno 47

net.ipv4.udp\_mem = 1526880 2035840 3053760 48

net.ipv4.conf.all.forwarding = 0 48

net.ipv4.conf.all.mc\_forwarding = 0 48

net.ipv4.conf.all.accept\_redirects = 0 48

net.ipv4.conf.all.secure\_redirects = 1 48

net.ipv4.conf.all.shared\_media = 1 49

net.ipv4.conf.all.rp\_filter = 0 49

net.ipv4.conf.all.send\_redirects = 1 49

net.ipv4.conf.all.accept\_source\_route = 0 49

net.ipv4.conf.all.proxy\_arp = 0 49

net.ipv4.conf.all.medium\_id = 0 49

net.ipv4.conf.all.bootp\_relay = 0 50

net.ipv4.conf.all.log\_martians = 0 50

net.ipv4.conf.all.arp\_filter = 0 50

net.ipv4.conf.all.arp\_announce = 0 50

net.ipv4.conf.all.arp\_ignore = 0 51

net.ipv4.conf.all.arp\_accept = 0 51

net.ipv4.conf.all.arp\_notify = 0 51

net.ipv4.conf.all.proxy\_arp\_pvlan = 0 51

net.ipv4.conf.all.disable\_xfrm = 0 51

net.ipv4.conf.all.disable\_policy = 0 51

net.ipv4.conf.all.promote\_secondaries = 0 51

net.ipv4.conf.all.accept\_local = 0 52

net.ipv4.conf.default.forwarding = 0 52

net.ipv4.conf.default.mc\_forwarding = 0 52

net.ipv4.conf.default.accept\_redirects = 1 52

net.ipv4.conf.default.secure\_redirects = 1 52

net.ipv4.conf.default.shared\_media = 1 52

net.ipv4.conf.default.rp\_filter = 1 52

net.ipv4.conf.default.send\_redirects = 1 52

net.ipv4.conf.default.accept\_source\_route = 0 52

net.ipv4.conf.default.proxy\_arp = 0 53

net.ipv4.conf.default.medium\_id = 0 53

net.ipv4.conf.default.bootp\_relay = 0 53

net.ipv4.conf.default.log\_martians = 0 53

net.ipv4.conf.default.arp\_filter = 0 53

net.ipv4.conf.default.arp\_announce = 0 53

net.ipv4.conf.default.arp\_ignore = 0 53

net.ipv4.conf.default.arp\_accept = 0 53

net.ipv4.conf.default.arp\_notify = 0 54

net.ipv4.conf.default.proxy\_arp\_pvlan = 0 54

net.ipv4.conf.default.disable\_xfrm = 0 54

net.ipv4.conf.default.disable\_policy = 0 54

net.ipv4.conf.default.promote\_secondaries = 0 54

net.ipv4.conf.default.accept\_local = 0 54

net.ipv4.ip\_forward = 0 56

net.ipv4.ipfrag\_high\_thresh = 4194304 56

net.ipv4.ipfrag\_low\_thresh = 3145728 56

net.ipv4.ipfrag\_time = 30 56

net.ipv4.icmp\_echo\_ignore\_all = 0 57

net.ipv4.icmp\_echo\_ignore\_broadcasts = 1 57

net.ipv4.icmp\_ignore\_bogus\_error\_responses = 1 57

net.ipv4.icmp\_errors\_use\_inbound\_ifaddr = 0 57

net.ipv4.icmp\_ratelimit = 1000 57

net.ipv4.icmp\_ratemask = 6168 58

net.ipv4.ip\_no\_pmtu\_disc = 0 58

net.ipv4.ipfrag\_secret\_interval = 600 58

net.ipv4.ipfrag\_max\_dist = 64 58

net.ipv6.conf.all.forwarding = 0 60

net.ipv6.conf.all.hop\_limit = 64 60

net.ipv6.conf.all.mtu = 1280 60

net.ipv6.conf.all.accept\_ra = 1 60

net.ipv6.conf.all.accept\_redirects = 1 60

net.ipv6.conf.all.dad\_transmits = 1 60

net.ipv6.conf.all.router\_solicitations = 3 60

net.ipv6.conf.all.router\_solicitation\_interval = 4 60

net.ipv6.conf.all.router\_solicitation\_delay = 1 61

net.ipv6.conf.all.regen\_max\_retry = 5 61

net.ipv6.conf.all.max\_desync\_factor = 600 61

net.ipv6.conf.all.max\_addresses = 16 61

net.ipv6.conf.all.accept\_ra\_defrtr = 1 61

net.ipv6.conf.all.accept\_ra\_pinfo = 1 61

net.ipv6.conf.all.accept\_ra\_rtr\_pref = 1 61

net.ipv6.conf.all.router\_probe\_interval = 60 61

net.ipv6.conf.all.accept\_ra\_rt\_info\_max\_plen = 0 62

net.ipv6.conf.all.proxy\_ndp = 0 62

net.ipv6.conf.all.accept\_source\_route = 0 62

net.ipv6.conf.all.optimistic\_dad = 0 62

net.ipv6.conf.all.mc\_forwarding = 0 62

net.ipv6.conf.all.disable\_ipv6 = 0 62

net.ipv6.conf.all.accept\_dad = 1 62

net.ipv6.conf.default.accept\_source\_route = 0 63

net.ipv6.bindv6only = 0 66

net.unix.max\_dgram\_qlen = 10 66

8.sunrpc 66

sunrpc.tcp\_fin\_timeout = 15 66

9.vm-内存参数 66

vm.overcommit\_memory = 0 67

vm.panic\_on\_oom = 0 67

vm.oom\_kill\_allocating\_task = 0 67

vm.extfrag\_threshold = 500 67

vm.oom\_dump\_tasks = 1 67

vm.overcommit\_ratio = 50 67

vm.overcommit\_kbytes = 0 68

vm.page-cluster = 3 68

vm.dirty\_background\_ratio = 10 68

vm.dirty\_background\_bytes = 0 68

vm.dirty\_ratio = 20 68

vm.dirty\_bytes = 0 68

vm.dirty\_writeback\_centisecs = 500 68

vm.dirty\_expire\_centisecs = 3000 69

vm.nr\_pdflush\_threads = 0 69

vm.swappiness = 1 69

vm.nr\_hugepages = 0 69

vm.nr\_hugepages\_mempolicy = 0 69

vm.hugetlb\_shm\_group = 0 69

vm.hugepages\_treat\_as\_movable = 0 69

vm.nr\_overcommit\_hugepages = 0 69

vm.lowmem\_reserve\_ratio = 256 256 32 70

vm.drop\_caches = 0 70

vm.min\_free\_kbytes = 67584 70

vm.percpu\_pagelist\_fraction = 0 70

vm.max\_map\_count = 65530 70

vm.laptop\_mode = 0 70

vm.block\_dump = 0 70

vm.vfs\_cache\_pressure = 100 71

vm.legacy\_va\_layout = 0 71

vm.zone\_reclaim\_mode = 0 71

vm.min\_unmapped\_ratio = 1 71

vm.min\_slab\_ratio = 5 71

vm.stat\_interval = 1 71

vm.mmap\_min\_addr = 4096 72

vm.numa\_zonelist\_order = default 72

vm.memory\_failure\_early\_kill = 0 72

vm.memory\_failure\_recovery = 1 72

vm.admin\_reserve\_kbytes = 8192 72

# 说明

以下参数取值dotcms 生产环境 5



注意:

1. 已经咨询过运维同事，不同账号下所获得的linux参数一致
2. 由于参数过多，并没有给每一天参数添加序号而是使用超链接方式进行跳转，若有遗漏或错误麻烦更新下
3. 各参数的数值代表的意思，千万不要根据经验来判断
4. 该文档中查找的资料结果，部分可能会因为版本问题有偏差或者遗漏，如有发现，请及时修改

# 正文

## 1.abi

abi.vsyscall32 = 1

## 2.crypto

crypto.fips\_enabled = 0

## 3.debug

debug.exception-trace = 1

debug.kprobes-optimization = 1

## 4.dev

dev.scsi.logging\_level = 0

dev.raid.speed\_limit\_min = 1000

dev.raid.speed\_limit\_max = 200000

dev.hpet.max-user-freq = 64

dev.mac\_hid.mouse\_button\_emulation = 0

dev.mac\_hid.mouse\_button2\_keycode = 97

dev.mac\_hid.mouse\_button3\_keycode = 100

dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17

dev.cdrom.info =

dev.cdrom.info = drive name: sr0

dev.cdrom.info = drive speed: 1

dev.cdrom.info = drive # of slots: 1

dev.cdrom.info = Can close tray: 1

dev.cdrom.info = Can open tray: 1

dev.cdrom.info = Can lock tray: 1

dev.cdrom.info = Can change speed: 1

dev.cdrom.info = Can select disk: 0

dev.cdrom.info = Can read multisession: 1

dev.cdrom.info = Can read MCN: 1

dev.cdrom.info = Reports media changed: 1

dev.cdrom.info = Can play audio: 1

dev.cdrom.info = Can write CD-R: 1

dev.cdrom.info = Can write CD-RW: 1

dev.cdrom.info = Can read DVD: 1

dev.cdrom.info = Can write DVD-R: 1

dev.cdrom.info = Can write DVD-RAM: 1

dev.cdrom.info = Can read MRW: 1

dev.cdrom.info = Can write MRW: 1

dev.cdrom.info = Can write RAM: 1

dev.cdrom.info =

dev.cdrom.info =

dev.cdrom.autoclose = 1

dev.cdrom.autoeject = 0

dev.cdrom.debug = 0

dev.cdrom.lock = 1

dev.cdrom.check\_media = 0

## Fs-文件系统参数

已分配inode数，空闲inode数

### fs.inode-nr = 157067 55781

已分配inode数 空闲inode数 已超出系统最大inode值的数量，此时系统需要清除排查inode列表

### fs.inode-state = 157067 55781 0 0 0 0 0

系统中已分配的文件句柄数量，已分配但没有使用的文件句柄数量，最大的文件句柄号

### fs.file-nr = 3584 0 1618273

该参数决定了系统中所允许的文件句柄最大数目，文件句柄设置代表[linux](http://www.ttlsa.com/linux/" \o "linux" \t "http://www.ttlsa.com/linux/linux-kernel-tuning-section-parameter-description/_blank)系统中可以打开的文件的数量。

### fs.file-max = 1618273

一个进程最多能同时打开的句柄数

### fs.nr\_open = 1048576

保存目录缓存的状态，保存有六个值，只有前三个有效

nr\_dentry：当前已经分配的目录项数量  
nr\_unused：还没有使用的目录项数量  
age\_limit：当内存紧缺时，延迟多少秒后会回收目录项所占内存

### fs.dentry-state = 112240 99795 45 0 0 0

Linux的UID为32位，但有些文件系统只支持16位的UID，此时若进行写操作会出错；当UID超过65535时会自动被转换为一个固定值，这个固定值保存在这个文件中

### fs.overflowuid = 65534

Linux的GID为32位，但有些文件系统只支持16位的GID，此时若进行写操作会出错；当GID超过65535时会自动被转换为一个固定值，这个固定值保存在这个文件中

### fs.overflowgid = 65534

设置是否启用dnotify，已被inotify取代，因为dnotify 需要为每个打算监控是否发生改变的目录打开一个文件描述符。

当同时监控多个目录时，这会消耗大量的资源，因为有可能达到每个进程的文件描述符限制。并且不允许卸载（unmount）支持的设备

0：不使用

1：使用

### fs.dir-notify-enable = 1

是否启用文件的租借锁 1：启用 0：不启用

### fs.leases-enable = 1

当进程尝试打开一个被租借锁保护的文件时，该进程会被阻塞，同时，在一定时间内拥有该文件租借锁的进程会收到一个信号。收到信号之后，拥有该文件租借锁的进程会首先更新文件，从而保证了文件内容的一致性，接着，该进程释放这个租借锁。如果拥有租借锁的进程在一定的时间间隔内没有完成工作，内核就会自动删除这个租借锁或者将该锁进行降级，从而允许被阻塞的进程继续工作。此保存租借锁的超时时间(以秒为单位)

### fs.lease-break-time = 45

当前aio数

如果aio-nr达到aio-max-nr，则io\_setup将失败，并返回EAGAIN。

### fs.aio-nr = 0

可以同时拥有异步aio请求的最大数目

### fs.aio-max-nr = 65536

每个用户可以运行的inotifywait或inotifywatch命令的进程数

### fs.inotify.max\_user\_instances = 128

inotifywait或inotifywatch命令可以监视的文件数量(单进程)

### fs.inotify.max\_user\_watches = 8192

调用inotify\_init函数时分配给inotify队列的事件数目的最大值，

超出这个值得事件被丢弃，但会触发IN\_Q\_OVERFLOW事件

文件系统变化越频繁，这个值就应该越大

### fs.inotify.max\_queued\_events = 16384

IO复用epoll监听文件句柄的数量最大值

### fs.epoll.max\_user\_watches = 3335680

fs.suid\_dumpable = 0

设置binfmt\_misc开启  
0：禁止  
1：开启

### fs.binfmt\_misc.status = enabled

fs.quota.lookups = 16914

fs.quota.drops = 15387

fs.quota.reads = 147

fs.quota.writes = 288

fs.quota.cache\_hits = 16643

fs.quota.allocated\_dquots = 147

fs.quota.free\_dquots = 45

fs.quota.syncs = 17

fs.quota.warnings = 1

系统中允许的消息队列的最大数量

### fs.mqueue.queues\_max = 256

一个消息队列的最大消息数

### fs.mqueue.msg\_max = 10

消息队列中一个消息的最大大小(以字节为单位)

### fs.mqueue.msgsize\_max = 8192

POSIX的消息队列  
此文件保存一个消息队列中消息数量的默认值，如果此值超过msg\_max，则会被设置为msg\_max

### fs.mqueue.msg\_default = 10

消息队列中一个消息的默认大小(以字节为单位)

### fs.mqueue.msgsize\_default = 8192

参数设置服务器重新引导后客户机回收NFSv3锁和NFSv4锁所需的秒数。因此，grace\_period的值可控制NFSv3和NFSv4的锁定恢复的宽延期长度。

### fs.nfs.nlm\_grace\_period = 0

为NFS锁管理器指定默认超时时间，单位是秒。默认值是10秒。取值范围在[3-20]

### fs.nfs.nlm\_timeout = 10

为NFS锁管理器指定UDP端口

### fs.nfs.nlm\_udpport = 0

为NFS锁管理器指定TCP端口

### fs.nfs.nlm\_tcpport = 0

不知道

### fs.nfs.nsm\_use\_hostnames = 0

不知道

### fs.nfs.nsm\_local\_state = 0

设置NFSv4回复通道(callback channel)监听的TCP端口

### fs.nfs.nfs\_callback\_tcpport = 0

设置idmapper缓存项的最大寿命，单位是秒

### fs.nfs.idmap\_cache\_timeout = 600

不知道

### fs.nfs.nfs\_mountpoint\_timeout = 500

不知道

### fs.nfs.nfs\_congestion\_kb = 129216

## 6.Kernel-内核参数

表示在创建子进程的时候是否让子进程抢占父进程，即使父进程的vruntime小于子进程，这个会减少公平性，但是可以降低write\_on\_copy，具体要根据系统的应用情况来考量使用哪种方式

0:先父进程

1:先子进程

### kernel.sched\_child\_runs\_first = 0

表示进程最少运行时间，防止频繁的切换，对于交互系统（如桌面），该值可以设置得较小，这样可以保证交互得到更快的响应（见周期调度器的check\_preempt\_tick过程）

### kernel.sched\_min\_granularity\_ns = 4000000

表示一个运行队列所有进程运行一次的周期

当前这个与运行队列的进程数有关，如果进程数超过sched\_nr\_latency（这个变量不能通过/proc设置，它是由

(sysctl\_sched\_latency+ sysctl\_sched\_min\_granularity-1)/sysctl\_sched\_min\_granularity确定的）,

那么调度周期就是[sched\_min\_granularity\_ns](#_kernel.sched_min_granularity_ns = 4000000)\*运行队列里的进程数，与sysctl\_sched\_latency无关；否则队列进程数小于sched\_nr\_latency，运行周期就是sysctl\_sched\_latency。显然这个数越小，一个运行队列支持的sched\_nr\_latency越少，而且当sysctl\_sched\_min\_granularity越小时能支持的sched\_nr\_latency越多，那么每个进程在这个周期内能执行的时间也就越少，这也与上面sysctl\_sched\_min\_granularity变量的讨论一致。其实sched\_nr\_latency也可以当做我们cpu load的基准值，如果cpu的load大于这个值，那么说明cpu不够使用了

### kernel.sched\_latency\_ns = 20000000

进程被唤醒后至少应该运行的时间的基数，它只是用来判断某个进程是否应该抢占当前进程，并不代表它能够执行的最小时间（sysctl\_sched\_min\_granularity），如果这个数值越小，那么发生抢占的概率也就越高

### kernel.sched\_wakeup\_granularity\_ns = 4000000

当内核试图调整sched\_min\_granularity，sched\_latency和sched\_wakeup\_granularity这三个值的时候所使用的更新方法，0为不调整，1为按照cpu个数以2为底的对数值进行调整，2为按照cpu的个数进行线性比例的调整

### kernel.sched\_tunable\_scaling = 1

表示调度器支持的特性，如:

GENTLE\_FAIR\_SLEEPERS(平滑的补偿睡眠进程)

START\_DEBIT(新进程尽量的早调度)

WAKEUP\_PREEMPT(是否wakeup的进程可以去抢占当前运行的进程)等

所有的features见内核sech\_features.h文件的定义

### kernel.sched\_features = 3183

判断一个进程是否还是hot，如果进程的运行时间（now - p->se.exec\_start）小于它，那么内核认为它的code还在cache里，所以该进程还是hot，那么在迁移的时候就不会考虑它

### kernel.sched\_migration\_cost = 500000

在多CPU情况下进行负载均衡时，一次最多移动多少个进程到另一个CPU上

### kernel.sched\_nr\_migrate = 32

kernel.sched\_time\_avg = 1000

kernel.sched\_shares\_window = 10000000

不知道啊

### kernel.timer\_migration = 1

该参数与sched\_rt\_runtime\_us（下面那个）一起决定了实时进程在以sched\_rt\_period为周期的时间内，实时进程最多能够运行的总的时间不能超过sched\_rt\_runtime\_us

### kernel.sched\_rt\_period\_us = 1000000

该参数与sched\_rt\_period（上面那个）一起决定了实时进程在以sched\_rt\_period为周期的时间内，实时进程最多能够运行的总的时间不能超过sched\_rt\_runtime\_us

### kernel.sched\_rt\_runtime\_us = 950000

### kernel.sched\_compat\_yield = 0

用来指示round robin调度进程的间隔，这个间隔默认是100ms

### kernel.sched\_rr\_timeslice\_ms = 100

启用后，内核会创建任务组来优化桌面程序的调度。它将把占用大量资源的应用程序放在它们自己的任务组，这有助于性能提升  
0：禁止  
1：开启

### kernel.sched\_autogroup\_enabled = 0

不知道

### kernel.sched\_cfs\_bandwidth\_slice\_us = 5000

系统发生panic时内核重新引导之前的等待时间  
0：禁止重新引导  
>0：重新引导前的等待时间(秒)

### kernel.panic = 0

当系统发生oops或BUG时，所采取的措施  
0：继续运行  
1：让klog记录oops的输出，然后panic，若kernel.panic不为0，则等待后重新引导内核

### kernel.panic\_on\_oops = 1

0：只警告，不发生panic  
1：发生panic

### kernel.panic\_on\_warn = 0

kernel.exec-shield = 1

Core文件的文件名是否添加应用程序pid做为扩展  
 0：不添加  
 1：添加

### kernel.core\_uses\_pid = 1

设置core文件保存位置或文件名,只有文件名时，则保存在应用程序运行的目录下

### kernel.core\_pattern=/opt/applog/coredir/core-%e-%p-%h-%t

定义了可以有多少个并发的崩溃程序可以通过管道模式传递给指定的core信息收集程序。如果超过了指定数，则后续的程序将不会处理，只在内核日志中做记录。0是个特殊的值，当设置为0时，不限制并行捕捉崩溃的进程，但不会等待用户程序搜集完毕方才回收/proc/pid目录（就是说，崩溃程序的相关信息可能随时被回收，搜集的信息可能不全）

### kernel.core\_pipe\_limit = 0

1：加载非GPL module  
0：强制加载module

### kernel.tainted = 0

根文件系统所在的设备(写入格式是0x主设备号(16位)次设备号(16位)，例如0x801，主设备号是8，次设备号是1)，只有使用initrd.img此参数才有效

### kernel.real-root-dev = 0

不知道

### kernel.print-fatal-signals = 0

该值控制系统在接收到 ctrl+alt+delete 按键组合时如何反应：  
1：不捕获ctrl-alt-del，将系统类似于直接关闭电源  
0：捕获ctrl-alt-del，并将此信号传至cad\_pid保存的PID号进程进行处理

### kernel.ctrl-alt-del = 0

kernel.ftrace\_enabled = 1

kernel.stack\_tracer\_enabled = 0

确定是否将ftrace的缓冲区的信息打印出来，是通过printk来打印的  
0：不打印  
1：在系统oops时，自动dump堆栈信息到输出终端

### kernel.ftrace\_dump\_on\_oops = 0

该文件给出了当系统支持module时完成modprobe功能的程序的名字（包括路径）。

### kernel.modprobe = /sbin/modprobe

表示是否禁止内核运行时可加载模块  
0：不禁止  
1：禁止

### kernel.modules\_disabled = 0

表示kexec\_load系统调用是否被禁止，此系统调用用于kdump。当发生了一次kexec\_load后，此值会自动设置为1。  
0：开启kexec\_load系统调用  
1：禁止kexec\_load系统调用

### kernel.kexec\_load\_disabled = 0

该文件给出了当前系统支持热插拔(hotplug)时接收热插拔事件的程序的名字（包括路径）。

### kernel.hotplug =

acct功能用于系统记录进程信息，正常结束的进程都会在该文件尾添加对应的信息。异常结束是指重启或其它致命的系统问题，不能够记录永不停止的进程。该设置需要配置三个值，分别是：  
1.如果文件系统可用空间低于这个百分比值，则停止记录进程信息。  
2.如果文件系统可用空间高于这个百分比值，则开始记录进程信息。  
3.检查上面两个值的频率(以秒为单位)。

### kernel.acct = 4 2 30

该文件指定的值为非零，则激活键盘上的sysrq按键。

### kernel.sysrq = 0

系统中进程数量(包括线程)的最大值

### kernel.threads-max = 127246

熵池大小，一般是4096位，可以改成任何大小

### kernel.random.poolsize = 4096

此文件是个只读文件，给出了一个有效的熵(4096位)

### kernel.random.entropy\_avail = 193

该文件保存熵的长度，该长度用于唤醒因读取/dev/random设备而待机的进程(random设备的读缓冲区长度？)

### kernel.random.read\_wakeup\_threshold = 64

该文件保存熵的长度，该长度用于唤醒因写入/dev/random设备而待机的进程(random设备的写缓冲区长度？)

### kernel.random.write\_wakeup\_threshold = 128

此文件是个只读文件，包含了一个随机字符串，在系统启动的时候会自动生成这个uuid

### kernel.random.boot\_id=ad2889a9-2f17-4eb7-8b25-30e48c8a6451

此文件是个只读文件，包含了一个随机字符串，在random设备每次被读的时候生成

### kernel.random.uuid=48a82b76-296a-44a0-8adb-8f54de384356

Linux的GID为32位，但有些文件系统只支持16位的UID，此时若进行写操作会出错；当GID超过65535时会自动被转换为一个固定值，这个固定值保存在这个文件中

### kernel.overflowuid = 65534

目前查到的一个该配置说明，和上面那个作用一样，但是是为gid准备的

### kernel.overflowgid = 65534

进程pid的最大值

### kernel.pid\_max = 131072

该文件有四个数字值，它们根据日志记录消息的重要性，定义将其发送到何处。按顺序是：  
 1.控制台日志级别：优先级高于该值的消息将被打印至控制台  
 2.默认的消息日志级别：将用该优先级来打印没有优先级的消息  
 3.最低的控制台日志级别：控制台日志级别可被设置的最小值(最高优先级)  
 4.默认的控制台日志级别：控制台日志级别的缺省值  
数值越小，优先级越高，级别有(0~7)

### kernel.printk = 4 4 1 7

等待允许再次printk的时间(以秒为单位)，与printk\_ratelimit()函数有关

### kernel.printk\_ratelimit = 5

printk的缓存队列长度

每个printk为一个长度，比如此值为10，而有段代码是连续printk20次，系统的处理是先printk前10次，等待printk\_ratelimit秒后，再打印后面10次

### kernel.printk\_ratelimit\_burst = 10

printk 消息之间的延迟毫秒数

### kernel.printk\_delay = 0

限制哪些用户可以查看syslog日志  
0：不限制  
1：只有特权用户能够查看

### kernel.dmesg\_restrict = 0

是否启用kptr\_restrice，此功能为安全性功能，用于屏蔽内核指针。  
0：该特性被完全禁止;  
1：那些使用“%pk”打印出来的内核指针被隐藏(会以一长串0替换掉)，除非用户有CAP\_SYSLOG权限，并且没有改变他们的UID/GID(防止在撤销权限之前打开的文件泄露指针信息);  
2：所有内核指针使用“%pk”打印的都被隐藏。

### kernel.kptr\_restrict = 1

每个用户最大的组数

### kernel.ngroups\_max = 65536

系统capabilities最高支持的权限等级。

### kernel.cap\_last\_cap = 33

表示是否禁止softlockup模式和nmi\_watchdog(softlockup用于唤醒watchdog)  
0：禁止  
1：开启

### kernel.watchdog = 1

用于设置高精度定时器(hrtimer)、nmi事件、softlockup、hardlockup的阀值(以秒为单位)  
0：不设置阀值

### kernel.watchdog\_thresh = 60

设置产生softlockup时是否抛出一个panic。

Softlockup用于检测CPU可以响应中断，但是在长时间内不能调度（比如禁止抢占时间太长）的死锁情况。这个机制运行在一个hrtimer的中断上下文，每隔一段时间检测一下是否发生了调度，如果过长时间没发生调度，说明系统被死锁。  
0：不产生panic  
1：产生panic

### kernel.softlockup\_panic = 0

kernel.softlockup\_all\_cpu\_backtrace = 0

nmi\_watchdog  
0：禁止  
1：开启

### kernel.nmi\_watchdog = 1

kernel.unknown\_nmi\_panic = 0

kernel.panic\_on\_unrecovered\_nmi = 0

kernel.panic\_on\_io\_nmi = 0

kernel.bootloader\_type = 113

kernel.bootloader\_version = 1

kernel.kstack\_depth\_to\_print = 12

kernel.io\_delay\_type = 0

用于设置进程虚拟地址空间的随机化  
0：关闭进程虚拟地址空间随机化  
1：随机化进程虚拟地址空间中的mmap映射区的初始地址，栈空间的初始地址以及VDSO页的地址  
2：在1的基础上加上堆区的随机化  
(VDSO是用于兼容不同内核与glibc的接口的机制)

### kernel.randomize\_va\_space = 2

kernel.acpi\_video\_flags = 0

设置hung\_task发生后是否引发panic  
1：触发  
0：不触发

### kernel.hung\_task\_panic = 0

hung\_task检查的进程数量最大值  
hung\_task用于检测一个进程是否在TASK\_UNINTERRUPTIBLE状态过长，只有在等待IO的时候进程才会处于TASK\_UNINTERRUPTIBLE状态，这个状态的进程内核不能够通过信号将其唤醒并杀死。

### kernel.hung\_task\_check\_count = 4194304

hung\_task超时时间(以秒为单位)，当一个进程在TASK\_UNINTERRUPTIBLE状态超过这个时间后，会发生一个hung\_task  
linux会设置40%的可用内存用来做系统cache，当flush数据时这40%内存中的数据由于和IO同步问题导致超时。

### kernel.hung\_task\_timeout\_secs = 120

最大产生警告数量，当发生一次hung\_task时会产生一次警告，但警告数量到达此值后之后的hung\_task就不会发生警告

### kernel.hung\_task\_warnings = 10

待补充

### kernel.compat-log = 1

触发死锁检查的嵌套深度值

### kernel.max\_lock\_depth = 1024

执行关机命令的进程路径

### kernel.poweroff\_cmd = /sbin/poweroff

待补充

### \*\*kernel.keys.root\_maxkeys = 1000000

### \*\*kernel.keys.root\_maxbytes = 25000000

### \*\*kernel.keys.maxkeys = 20

### \*\*kernel.keys.maxbytes = 20000

### \*\*kernel.keys.gc\_delay = 300

这一个是生产服务器上没有的，额外补充的

### kernel.keys.persistent\_keyring\_expiry

kernel.slow-work.min-threads = 2

kernel.slow-work.max-threads = 128

kernel.slow-work.vslow-percentage = 50

用于限制访问性能计数器的权限  
0：仅允许访问用户空间的性能计数器  
1：内核与用户空间的性能计数器都可以访问  
2：仅允许访问特殊的CPU数据(不包括跟踪点)  
-1：不限制

### kernel.perf\_event\_paranoid = 1

设置非特权用户能够允许常驻内存的内存大小。默认为516(KB)

### kernel.perf\_event\_mlock\_kb = 516

设置perf\_event的最大取样速率，默认值为100000

### kernel.perf\_event\_max\_sample\_rate = 100000

接收Ctrl-alt-del操作的INT信号的进程的PID，结合[kernel.ctrl-alt-del](#_kernel.ctrl-alt-del = 0)进行参考

### kernel.cad\_pid

待补充

### kernel.blk\_iopoll = 1

kernel.sched\_domain.cpu0.domain0.min\_interval = 1

kernel.sched\_domain.cpu0.domain0.max\_interval = 4

kernel.sched\_domain.cpu0.domain0.busy\_idx = 2

kernel.sched\_domain.cpu0.domain0.idle\_idx = 1

kernel.sched\_domain.cpu0.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu0.domain0.wake\_idx = 0

kernel.sched\_domain.cpu0.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu0.domain0.busy\_factor = 64

kernel.sched\_domain.cpu0.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu0.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu0.domain0.flags = 4143

kernel.sched\_domain.cpu0.domain0.name = CPU

kernel.sched\_domain.cpu1.domain0.min\_interval = 1

kernel.sched\_domain.cpu1.domain0.max\_interval = 4

kernel.sched\_domain.cpu1.domain0.busy\_idx = 2

kernel.sched\_domain.cpu1.domain0.idle\_idx = 1

kernel.sched\_domain.cpu1.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu1.domain0.wake\_idx = 0

kernel.sched\_domain.cpu1.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu1.domain0.busy\_factor = 64

kernel.sched\_domain.cpu1.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu1.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu1.domain0.flags = 4143

kernel.sched\_domain.cpu1.domain0.name = CPU

kernel.sched\_domain.cpu2.domain0.min\_interval = 1

kernel.sched\_domain.cpu2.domain0.max\_interval = 4

kernel.sched\_domain.cpu2.domain0.busy\_idx = 2

kernel.sched\_domain.cpu2.domain0.idle\_idx = 1

kernel.sched\_domain.cpu2.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu2.domain0.wake\_idx = 0

kernel.sched\_domain.cpu2.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu2.domain0.busy\_factor = 64

kernel.sched\_domain.cpu2.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu2.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu2.domain0.flags = 4143

kernel.sched\_domain.cpu2.domain0.name = CPU

kernel.sched\_domain.cpu3.domain0.min\_interval = 1

kernel.sched\_domain.cpu3.domain0.max\_interval = 4

kernel.sched\_domain.cpu3.domain0.busy\_idx = 2

kernel.sched\_domain.cpu3.domain0.idle\_idx = 1

kernel.sched\_domain.cpu3.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu3.domain0.wake\_idx = 0

kernel.sched\_domain.cpu3.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu3.domain0.busy\_factor = 64

kernel.sched\_domain.cpu3.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu3.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu3.domain0.flags = 4143

kernel.sched\_domain.cpu3.domain0.name = CPU

kernel.sched\_domain.cpu4.domain0.min\_interval = 1

kernel.sched\_domain.cpu4.domain0.max\_interval = 4

kernel.sched\_domain.cpu4.domain0.busy\_idx = 2

kernel.sched\_domain.cpu4.domain0.idle\_idx = 1

kernel.sched\_domain.cpu4.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu4.domain0.wake\_idx = 0

kernel.sched\_domain.cpu4.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu4.domain0.busy\_factor = 64

kernel.sched\_domain.cpu4.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu4.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu4.domain0.flags = 4143

kernel.sched\_domain.cpu4.domain0.name = CPU

kernel.sched\_domain.cpu5.domain0.min\_interval = 1

kernel.sched\_domain.cpu5.domain0.max\_interval = 4

kernel.sched\_domain.cpu5.domain0.busy\_idx = 2

kernel.sched\_domain.cpu5.domain0.idle\_idx = 1

kernel.sched\_domain.cpu5.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu5.domain0.wake\_idx = 0

kernel.sched\_domain.cpu5.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu5.domain0.busy\_factor = 64

kernel.sched\_domain.cpu5.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu5.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu5.domain0.flags = 4143

kernel.sched\_domain.cpu5.domain0.name = CPU

kernel.sched\_domain.cpu6.domain0.min\_interval = 1

kernel.sched\_domain.cpu6.domain0.max\_interval = 4

kernel.sched\_domain.cpu6.domain0.busy\_idx = 2

kernel.sched\_domain.cpu6.domain0.idle\_idx = 1

kernel.sched\_domain.cpu6.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu6.domain0.wake\_idx = 0

kernel.sched\_domain.cpu6.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu6.domain0.busy\_factor = 64

kernel.sched\_domain.cpu6.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu6.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu6.domain0.flags = 4143

kernel.sched\_domain.cpu6.domain0.name = CPU

kernel.sched\_domain.cpu7.domain0.min\_interval = 1

kernel.sched\_domain.cpu7.domain0.max\_interval = 4

kernel.sched\_domain.cpu7.domain0.busy\_idx = 2

kernel.sched\_domain.cpu7.domain0.idle\_idx = 1

kernel.sched\_domain.cpu7.domain0.newidle\_idx = 0

kernel.sched\_domain.cpu7.domain0.wake\_idx = 0

kernel.sched\_domain.cpu7.domain0.forkexec\_idx = 0

kernel.sched\_domain.cpu7.domain0.busy\_factor = 64

kernel.sched\_domain.cpu7.domain0.imbalance\_pct = 125

kernel.sched\_domain.cpu7.domain0.cache\_nice\_tries = 1

kernel.sched\_domain.cpu7.domain0.flags = 4143

kernel.sched\_domain.cpu7.domain0.name = CPU

kernel.vsyscall64 = 1

系统类型

### kernel.ostype = Linux

内核版本号

### kernel.osrelease = 2.6.32-573.35.2.el6.x86\_64

版本号

### kernel.version = #1 SMP Mon Oct 24 14:14:01 EDT 2016

主机名(重启失效)

### kernel.hostname = vm-vmw51102-jbs

网络域名(重启失效)

### kernel.domainname = (none)

所能分配的PTY的最多个数(pty为虚拟终端，用于远程连接时)

### kernel.pty.max = 4096

当前分配的pty的个数

### kernel.pty.nr = 3

系统所允许的最大共享内存段的大小（byte）。

### kernel.shmmax = 68719476736

系统上可以使用的共享内存的总量（byte）

### kernel.shmall = 4294967296

整个系统共享内存段的最大数量。

### kernel.shmmni = 4096

强制SHM空间和一个进程联系在一起，所以可以通过杀死进程来释放内存  
0：不设置  
1：设置

### kernel.shm\_rmid\_forced = 0

消息队列中单个消息的最大字节数

### kernel.msgmax = 65536

系统中同时运行的消息队列的个数

### kernel.msgmni = 32768

单个消息队列中允许的最大字节长度(限制单个消息队列中所有消息包含的字节数之和)

### kernel.msgmnb = 65536

第一列，表示每个信号集中的最大信号量数目。

第二列，表示系统范围内的最大信号量总数目。

第三列，表示每个信号发生时的最大系统操作数目。

第四列，表示系统范围内的最大信号集总数目。

（第一列）\*（第四列）=（第二列）

### kernel.sem = 250 32000 32 128

系统自动设置同时运行的消息队列个数。  
0：不自动  
1：自动

### kernel.auto\_msgmni = 1

### 以下几个是生产服务器上没有（或当前账号无法查看的）

是否开启numa\_balancing？这块具体看代码

kernel.numa\_balancing

单个进程每次进行numa\_balancing扫描的间隔时间

kernel.numa\_balancing\_scan\_delay\_ms

每次扫描最多花费的时间

kernel.numa\_balancing\_scan\_period\_max\_ms

每次扫描最少花费的时间

kernel.numa\_balancing\_scan\_period\_min\_ms

一次扫描进程多少MB的虚拟地址空间内存

kernel.numa\_balancing\_scan\_size\_mb

kernel.numa\_balancing\_settle\_count

=====================================================================

perf分析工具最大能够占用CPU性能的百分比  
0：不限制  
1~100：百分比值

kernel.perf\_cpu\_time\_max\_percent

PPC专用，如果开启，则使用nap节能模式，关闭则使用doze节能模式  
0：关闭  
1：开启

kernel.powersave-nap

kernel.pty.reserve

====================================================================

判断该调度域是否已经均衡的一个基准值

kernel.sched\_domain.{CPUID}.{域ID}.imbalance\_pct

设置此CPU进行负载均衡的最长间隔时间，上一次做了负载均衡经过了这个时间一定要再进行一次

kernel.sched\_domain.{CPUID}.{域ID}.max\_interval

设置此CPU进行负载均衡的最小间隔时间，在上一次负载均衡到这个时间内都不能再进行负载均衡

kernel.sched\_domain.{CPUID}.{域ID}.min\_interva

kernel.sched\_shares\_window\_ns

kernel.sched\_time\_avg\_ms

## 7.net-网络参数

net.netfilter.nf\_log.0 = NONE

net.netfilter.nf\_log.1 = NONE

net.netfilter.nf\_log.2 = NONE

net.netfilter.nf\_log.3 = NONE

net.netfilter.nf\_log.4 = NONE

net.netfilter.nf\_log.5 = NONE

net.netfilter.nf\_log.6 = NONE

net.netfilter.nf\_log.7 = NONE

net.netfilter.nf\_log.8 = NONE

net.netfilter.nf\_log.9 = NONE

net.netfilter.nf\_log.10 = NONE

net.netfilter.nf\_log.11 = NONE

net.netfilter.nf\_log.12 = NONE

用来限制监听(LISTEN)队列最大数据包的数量，超过这个数量就会导致链接超时或者触发重传机制

全局参数

### net.core.somaxconn = 128

net.core.xfrm\_aevent\_etime = 10

net.core.xfrm\_aevent\_rseqth = 2

net.core.xfrm\_larval\_drop = 1

net.core.xfrm\_acq\_expires = 30

发送套接字缓存区大小的最大值

### net.core.wmem\_max = 124928

接收套接字缓存区大小的最大值

### net.core.rmem\_max = 124928

发送套接字缓存区大小的默认值

### net.core.wmem\_default = 124928

接收套接字缓存区大小的默认值（Byte）

### net.core.rmem\_default = 124928

每个CPU一次NAPI中断能够处理网络包数量的最大值

### net.core.dev\_weight = 64

当网卡接收数据包的速度大于内核处理的速度时，会有一个队列保存这些数据包。这个参数表示该队列的最大值

### net.core.netdev\_max\_backlog = 1000

net.core.netdev\_rss\_key = 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

设置每一个警告的度量值，缺省为5，**用来防止DOS攻击时设置为0**

### net.core.message\_cost = 5

设置每十秒写入多少次请求警告；**可以用来防止DOS攻击**

### net.core.message\_burst = 10

表示每个socket所允许的最大缓冲区的大小(byte)

### net.core.optmem\_max = 20480

net.core.rps\_sock\_flow\_entries = 0

默认对网络设备进行poll和select操作的超时时间(us)，具体数值最好以sockets数量而定

### net.core.busy\_poll = 0

默认读取在设备帧队列上数据帧的超时时间(us)，推荐值：50

### net.core.busy\_read = 0

每次软中断处理的网络包个数

### net.core.netdev\_budget = 300

目前查到的资料，该参数已经不再使用

### net.core.warnings = 1

路由hash table的大小，当cache中的路由条数超过此值时，开始垃圾回收.

### net.ipv4.route.gc\_thresh = 524288

路由高速缓存的最大项数，超过会进行清除旧项操作.

### net.ipv4.route.max\_size = 8388608

**被net.ipv4.route.gc\_min\_interval\_ms取代**

### net.ipv4.route.gc\_min\_interval = 0

路由表垃圾回收的最小间隔(ms)

### net.ipv4.route.gc\_min\_interval\_ms = 500

设置一个路由表项的过期时长(秒).

### net.ipv4.route.gc\_timeout = 300

路由表垃圾回收的间隔(秒)

### net.ipv4.route.gc\_interval = 60

决定是否要向特定主机发送更多的ICMP重定向的时间因子.一旦达到load时间或number个数就不再发送.

### net.ipv4.route.redirect\_load = 20

决定是否要向特定主机发送更多的ICMP重定向的数量因子.一旦达到load时间或number个数就不再发送.

（和上面的文件一起决定）

### net.ipv4.route.redirect\_number = 9

重定向的超时.经过这么长时间后,重定向会重发,而不管是否已经因为超过load或者number限制而停止.

### net.ipv4.route.redirect\_silence = 20480

这个参数和error\_burst（下面）一起用于限制有多少个icmp不可达消息被发送.当数据包不能到达下一跳时会发送icmp不可达数据包.  
当一些主机忽略我们的icmp重定向消息时也会打印一些错误信息到dmesg.这个选项也控制打印的次数.  
error\_cost值越大，那么icmp不可达和写错误信息的频率就越低.(单位 秒)

### net.ipv4.route.error\_cost = 1000

这个参数和error\_cast（上面）一起用于限制有多少个icmp不可达消息被发送.当数据包不能到达下一跳时会发送icmp不可达数据包.  
当一些主机忽略我们的icmp重定向消息时也会打印一些错误信息到dmesg.这个选项也控制打印的次数.(单位 秒)

### net.ipv4.route.error\_burst = 5000

用来控制路由缓存垃圾回收机制的频率和行为

当路由表一个hash项的长度超过此值时，会进行缓存缩减,当路由缓存项长度超过  
ip\_rt\_gc\_elasticity << rt\_hash\_log(表示路由高速缓存hash table的容量以2为对数所得的值) 时会进行强烈的回收.

### net.ipv4.route.gc\_elasticity = 8

该文件表示PMTU信息缓存多长时间（秒）。

### net.ipv4.route.mtu\_expires = 600

该文件表示最小路径MTU的大小。

### net.ipv4.route.min\_pmtu = 552

该文件表示最小的MSS（Maximum Segment Size）大小，取决于第一跳的路由器MTU。(以字节为单位)

### net.ipv4.route.min\_adv\_mss = 256

net.ipv4.route.secret\_interval = 600

在把记录标记为不可达之前， 用多播/广播方式解析地址的最大次数.

### net.ipv4.neigh.default.mcast\_solicit = 3

arp请求最多发送次数

### net.ipv4.neigh.default.ucast\_solicit = 3

在使用多播探测前，通过netlink发送到用户空间arp守护程序的最大探测数

### net.ipv4.neigh.default.app\_solicit = 0

重发一个arp请求前的等待的秒数

### net.ipv4.neigh.default.retrans\_time = 99

重发一个arp请求前的等待的毫秒数

### net.ipv4.neigh.default.retrans\_time\_ms = 1000

一旦发现相邻记录，至少在一段介于 base\_reachable\_time/2和3\*base\_reachable\_time/2之间的随机时间内，该记录是有效的.  
如果收到上层协议的肯定反馈， 那么记录的有效期将延长.(单位 秒)

### net.ipv4.neigh.default.base\_reachable\_time = 30

一旦发现相邻记录，至少在一段介于 base\_reachable\_time/2和3\*base\_reachable\_time/2之间的随机时间内，该记录是有效的.  
如果收到上层协议的肯定反馈， 那么记录的有效期将延长.(单位 毫秒)

### net.ipv4.neigh.default.base\_reachable\_time\_ms=30000

发现某个相邻层记录无效后，发出第一个探测要等待的时间。(单位 秒)

### net.ipv4.neigh.default.delay\_first\_probe\_time = 5

决定检查一次相邻层记录的有效性的周期.

当相邻层记录失效时，将在给它发送数据前，再解析一次.(单位 秒)

### net.ipv4.neigh.default.gc\_stale\_time = 60

最大挂起arp请求的数量，这些请求都正在被解析中

### net.ipv4.neigh.default.unres\_qlen = 3

能放入代理 ARP 地址队列的数据包最大数目.

### net.ipv4.neigh.default.proxy\_qlen = 64

对相邻请求信息的回复的最大延迟时间(单位 秒)

### net.ipv4.neigh.default.anycast\_delay = 99

当接收到有一个arp请求时，在回应前可以延迟的时间，这个请求是要得到一个已知代理arp项的地址.(单位 百毫秒)

### net.ipv4.neigh.default.proxy\_delay = 79

防止相邻记录被过度频繁刷新，引起抖动，只有距邻居上次刷新时间超过这时才允许被再次刷新.(单位 秒)

### net.ipv4.neigh.default.locktime = 99

垃圾收集器收集相邻层记录和无用记录的运行周期(单位 秒)

### net.ipv4.neigh.default.gc\_interval = 30

存在于ARP高速缓存中的最少个数，如果少于这个数，垃圾收集器将不会运行

### net.ipv4.neigh.default.gc\_thresh1 = 128

保存在 ARP 高速缓存中的最多的记录软限制. 垃圾收集器在开始收集前，允许记录数超过这个数字，在创建新表项时如果发现5秒没有刷新过，那么进行强制回收

### net.ipv4.neigh.default.gc\_thresh2 = 512

保存在 ARP 高速缓存中的最多记录的硬限制， 一旦高速缓存中的数目高于此， 垃圾收集器将马上运行

### net.ipv4.neigh.default.gc\_thresh3 = 1024

net.ipv4.neigh.lo.mcast\_solicit = 3

net.ipv4.neigh.lo.ucast\_solicit = 3

net.ipv4.neigh.lo.app\_solicit = 0

net.ipv4.neigh.lo.retrans\_time = 99

net.ipv4.neigh.lo.base\_reachable\_time = 30

net.ipv4.neigh.lo.delay\_first\_probe\_time = 5

net.ipv4.neigh.lo.gc\_stale\_time = 60

net.ipv4.neigh.lo.unres\_qlen = 3

net.ipv4.neigh.lo.proxy\_qlen = 64

net.ipv4.neigh.lo.anycast\_delay = 99

net.ipv4.neigh.lo.proxy\_delay = 79

net.ipv4.neigh.lo.locktime = 99

net.ipv4.neigh.lo.retrans\_time\_ms = 1000

net.ipv4.neigh.lo.base\_reachable\_time\_ms = 30000

net.ipv4.neigh.eth0.mcast\_solicit = 3

net.ipv4.neigh.eth0.ucast\_solicit = 3

net.ipv4.neigh.eth0.app\_solicit = 0

net.ipv4.neigh.eth0.retrans\_time = 99

net.ipv4.neigh.eth0.base\_reachable\_time = 30

net.ipv4.neigh.eth0.delay\_first\_probe\_time = 5

net.ipv4.neigh.eth0.gc\_stale\_time = 60

net.ipv4.neigh.eth0.unres\_qlen = 3

net.ipv4.neigh.eth0.proxy\_qlen = 64

net.ipv4.neigh.eth0.anycast\_delay = 99

net.ipv4.neigh.eth0.proxy\_delay = 79

net.ipv4.neigh.eth0.locktime = 99

net.ipv4.neigh.eth0.retrans\_time\_ms = 1000

net.ipv4.neigh.eth0.base\_reachable\_time\_ms = 30000

net.ipv4.neigh.eth1.mcast\_solicit = 3

net.ipv4.neigh.eth1.ucast\_solicit = 3

net.ipv4.neigh.eth1.app\_solicit = 0

net.ipv4.neigh.eth1.retrans\_time = 99

net.ipv4.neigh.eth1.base\_reachable\_time = 30

net.ipv4.neigh.eth1.delay\_first\_probe\_time = 5

net.ipv4.neigh.eth1.gc\_stale\_time = 60

net.ipv4.neigh.eth1.unres\_qlen = 3

net.ipv4.neigh.eth1.proxy\_qlen = 64

net.ipv4.neigh.eth1.anycast\_delay = 99

net.ipv4.neigh.eth1.proxy\_delay = 79

net.ipv4.neigh.eth1.locktime = 99

net.ipv4.neigh.eth1.retrans\_time\_ms = 1000

net.ipv4.neigh.eth1.base\_reachable\_time\_ms = 30000

Timestamps 用在其它一些东西中﹐可以防范那些伪造的 sequence 号码。

一条1G的宽带线路或许会重遇到带 out-of-line数值的旧sequence 号码(假如它是由于上次产生的)。Timestamp 会让它知道这是个 '旧封包'。(该文件表示是否启用以一种比超时重发更精确的方法（RFC 1323）来启用对 RTT 的计算；为了实现更好的性能应该启用这个选项。)

### net.ipv4.tcp\_timestamps = 1

该文件表示设置tcp/ip会话的滑动窗口大小是否可变。参数值为布尔值，为1时表示可变，为0时表示不可变。tcp/ip通常使用的窗口最大可达到 65535 字节，对于高速网络，该值可能太小，这时候如果启用了该功能，可以使tcp/ip滑动窗口大小增大数个数量级，从而提高数据传输的能力(RFC 1323)。（对普通地百M网络而言，关闭会降低开销，所以如果不是高速网络，可以考虑设置为0）

### net.ipv4.tcp\_window\_scaling = 1

使用 Selective ACK﹐它可以用来查找特定的遗失的数据报，因此有助于快速恢复状态。

该文件表示是否启用有选择的应答（Selective Acknowledgment），这可以通过有选择地应答乱序接收到的报文来提高性能（这样可以让发送者只发送丢失的报文段）。(对于广域网通信来说这个选项应该启用，但是这会增加对 CPU 的占用。)

### net.ipv4.tcp\_sack = 1

对于某些有bug的打印机提供针对其bug的兼容性。(一般不需要这个支持,可以关闭它)

### net.ipv4.tcp\_retrans\_collapse = 1

该文件表示一个数据报的生存周期（Time To Live），即最多经过多少路由器

### net.ipv4.ip\_default\_ttl = 64

允许进程绑定到非本地地址  
0：禁止  
1：允许

### net.ipv4.ip\_nonlocal\_bind = 0

表示在内核放弃建立连接之前发送SYN包的数量

不应该大于255，默认值是5，对应于180秒左右时间。(对于大负载而物理通信良好的网络而言,这个值偏高,可修改为2.这个值仅仅是针对对外的连接,对进来的连接,是由[tcp\_retries1](#_net.ipv4.tcp_retries1 = 3) 决定的)

### net.ipv4.tcp\_syn\_retries = 5

表示系统允许SYN连接的重试次数。为了打开对端的连接，内核需要发送一个SYN并附带一个回应前面一个SYN的ACK包。也就是所谓三次握手中的第二次握手。这个设置决定了内核放弃连接之前发送SYN+ACK包的数量。

对于远端的连接请求SYN，内核会发送SYN ＋ ACK数据报，以确认收到上一个 SYN连接请求包。这是所谓的三次握手( threeway handshake)机制的第二个步骤。这里决定内核在放弃连接之前所送出的 SYN+ACK 数目。不应该大于255，默认值是5，对应于180秒左右时间。(可以根据上面的tcp\_syn\_retries来决定这个值)

（两种解释表达意思一样，不明白的可以结合起来看）

### net.ipv4.tcp\_synack\_retries = 5

缺省值是8192  
系统所能处理不属于任何进程的TCP sockets最大数量。假如超过这个数量﹐那么不属于任何进程的连接会被立即reset，并同时显示警告信息。之所以要设定这个限制﹐纯粹为了抵御那些简单的 DoS 攻击﹐千万不要依赖这个或是人为的降低这个限制(这个值Redhat AS版本中设置为32768,但是很多防火墙修改的时候,建议该值修改为2000)

### net.ipv4.tcp\_max\_orphans = 262144

拨号上网大部分都是使用动态IP地址，我们不知道远程拨号服务器的IP地址是多少，也不可能知道它会给电脑分配什么IP地址  
0：使用静态IP  
1：使用动态IP地址

### net.ipv4.ip\_dynaddr = 0

表示当keepalive启用时，TCP发送keepalive消息的频度。默认是2小时，若将其设置得小一些，可以更快地清理无效的连接。

### net.ipv4.tcp\_keepalive\_time = 7200

在认定连接失效之前，发送多少个TCP的keepalive探测包。系统默认值是9。这个值乘以tcp\_keepalive\_intvl之后决定了，一个连接发送了keepalive探测包之后可以有多少时间没有回应

### net.ipv4.tcp\_keepalive\_probes = 9

与net.ipv4.tcp\_keepalive\_probes一同使用

探测消息未获得响应时，重发该消息的间隔时间（秒）。系统默认75秒

### net.ipv4.tcp\_keepalive\_intvl = 75

放弃回应一个TCP连接请求前﹐需要进行多少次重试。RFC 规定最低的数值是3﹐这也是默认值﹐根据RTO的值大约在3秒 - 8分钟之间。(注意:这个值同时还决定进入的syn连接)

### net.ipv4.tcp\_retries1 = 3

在丢弃激活(已建立通讯状况)的TCP连接之前﹐需要进行多少次重试。默认值为15，根据RTO的值来决定，相当于13-30分钟(RFC1122规定，必须大于100秒).、

### net.ipv4.tcp\_retries2 = 15

减少处于FIN-WAIT-2连接状态的时间，使系统可以处理更多的连接

### net.ipv4.tcp\_fin\_timeout = 60

当出现SYN等待队列溢出时，启用cookies来处理，可防范SYN flood攻击，默认为0，表示关闭

只有在内核编译时选择了CONFIG\_SYNCOOKIES时才会发生作用

**注意：该选项千万不能用于那些没有收到攻击的高负载服务器，如果在日志中出现synflood消息，但是调查发现没有收到synflood攻击，而是合法用户的连接负载过高的原因，你应该调整其它参数来提高服务器性能，参考:**

**tcp\_max\_syn\_backlog  
tcp\_synack\_retries  
tcp\_abort\_on\_overflow**

**syncookie严重的违背TCP协议，不允许使用TCP扩展，可能对某些服务导致严重的性能影响(如SMTP转发)**

**(注意,该实现与BSD上面使用的tcp proxy一样,是违反了RFC中关于tcp连接的三次握手实现的,但是对于防御syn-flood的确很有用.)**

### net.ipv4.tcp\_syncookies = 1

1. 表示系统同时保持TIME\_WAIT套接字的最大数量。如果超过此数，TIME\_WAIT套接字会被立刻清除并且打印警告信息。之所以要设定这个限制，纯粹为了抵御那些简单的DoS攻击，不过，过多的TIME\_WAIT套接字也会消耗服务器资源，甚至死机。
2. 默认值是180000

### net.ipv4.tcp\_max\_tw\_buckets = 262144

以下两参数可解决生产场景中大量连接的服务器中TIME\_WAIT过多问题。

表示开启TCP连接中TIME\_WAIT套接字的快速回收，默认为0，表示关闭

**除非得到技术专家的建议或要求﹐请不要随意修改这个值**

### net.ipv4.tcp\_tw\_recycle = 0

表示允许重用TIME\_WAIT状态的套接字用于新的TCP连接,默认为0，表示关闭。

(这个对快速重启动某些服务,而启动后提示端口已经被使用的情形非常有帮助，**但是，别乱动**)

### net.ipv4.tcp\_tw\_reuse = 0

缺省值是0  
当守护进程太忙而不能接受新的连接，就象对方发送reset消息，默认值是false。

这意味着当溢出的原因是因为一个偶然的猝发，那么连接将恢复状态。

**只有在你确信守护进程真的不能完成连接请求时才打开该选项，该选项会影响客户的使用。**(对待已经满载的sendmail,apache这类服务的时候,这个可以很快让客户端终止连接,可以给予服务程序处理已有连接的缓冲机会,所以很多防火墙上推荐打开它)

0：关闭  
1：开启

### net.ipv4.tcp\_abort\_on\_overflow = 0

默认值为0  
使用 TCP urg pointer 字段中的主机请求解释功能。

### net.ipv4.tcp\_stdurg = 0

缺省值为0  
这个开关可以启动对于在RFC1337中描述的"tcp 的time-wait暗杀危机"问题的修复。启用后，内核将丢弃那些发往time-wait状态TCP套接字的RST 包.

### net.ipv4.tcp\_rfc1337 = 0

表示那些尚未收到客户端确认信息的连接（SYN消息）队列的长度，默认为1024，加大队列长度为262144，可以容纳更多等待连接的网络连接数。

### net.ipv4.tcp\_max\_syn\_backlog = 2048

表示允许系统打开的端口范围

tcp初始化时会修改此值

### net.ipv4.ip\_local\_port\_range = 32768 60999

net.ipv4.ip\_local\_reserved\_ports =

限制加入一个多播组的最大成员数.

### net.ipv4.igmp\_max\_memberships = 20

限制多播源地址过滤数量.

### net.ipv4.igmp\_max\_msf = 10

INET对端存储器某个合适值，当超过该阀值条目将被丢弃。

该阀值同样决定生存时间以及废物收集通过的时间间隔。条目越多，存活期越低，GC 间隔越短

### net.ipv4.inet\_peer\_threshold = 65664

条目的最低存活期。

在重组端必须要有足够的碎片(fragment)存活期。这个最低存活期必须保证缓冲池容积是否少于 inet\_peer\_threshold。该值以 jiffies为单位测量。

### net.ipv4.inet\_peer\_minttl = 120

条目的最大存活期。

在此期限到达之后，如果缓冲池没有耗尽压力的话(例如：缓冲池中的条目数目非常少)，不使用的条目将会超时。该值以 jiffies为单位测量

### net.ipv4.inet\_peer\_maxttl = 600

net.ipv4.inet\_peer\_gc\_mintime = 10

net.ipv4.inet\_peer\_gc\_maxtime = 120

在近端丢弃TCP连接之前﹐要进行多少次重试。默认值是7﹐相当于 50秒 - 16分钟﹐视 RTO 而定。

如果您的系统是负载很大的web服务器﹐那么也许需要降低该值﹐这类 sockets 可能会耗费大量的资源。

### net.ipv4.tcp\_orphan\_retries = 0

缺省值为1  
打开FACK拥塞避免和快速重传功能（当[tcp\_sack](#_net.ipv4.tcp_sack = 1)设置为0的时候，这个值即使设置为1也无效）

### net.ipv4.tcp\_fack = 1

默认值是3  
TCP流中重排序的数据报最大数量 。 (一般有看到推荐把这个数值略微调整大一些,比如5)

### net.ipv4.tcp\_reordering = 3

缺省值为0  
打开TCP的直接拥塞通告功能。

0：禁止  
1：启用

### net.ipv4.tcp\_ecn = 2

缺省值为1  
允许TCP发送"两个完全相同"的SACK。

0：禁止  
1：启用

### net.ipv4.tcp\_dsack = 1

确定TCP栈应该如何反映内存使用，每个值的单位都是内存页（通常是4KB）。

第一个值是内存使用的下限；low模式

当TCP使用了低于该值的内存页面数时，TCP不会考虑释放内存

(理想情况下，这个值应与指定给 tcp\_wmem 的第 2 个值相匹配 - 这第 2 个值表明，最大页面大小乘以最大并发请求数除以页大小 (131072 \* 300 / 4096)。 )

第二个值是内存压力模式开始对缓冲区使用应用压力的上限；pressure模式

当TCP使用了超过该值的内存页面数量时，TCP试图稳定其内存使用，进入pressure模式，当内存消耗低于low值时则退出pressure状态

(理想情况下这个值应该是 TCP 可以使用的总缓冲区大小的最大值 (204800 \* 300 / 4096)。 )

第三个值是内存使用的上限。High模式

允许所有tcp sockets用于排队缓冲数据报的页面量

如果超过这个值，TCP 连接将被拒绝，这就是为什么不要令其过于保守 (512000 \* 300 / 4096) 的原因了。 在这种情况下，提供的价值很大，它能处理很多连接，是所预期的 2.5 倍；或者使现有连接能够传输 2.5 倍的数据。

第一个值为1526880\*4/1024/1024=5.8G，第二个值为2035840\*4/1024/1024=7.76G，第三个值为3053760\*4/1024/1024=11.6G

### net.ipv4.tcp\_mem = 1526880 2035840 3053760

min：为TCP socket预留用于**发送**缓冲的内存最小值。每个tcp socket都可以在建议以后都可以使用它。默认值为4096(4K)。  
default：为TCP socket预留用于**发送**缓冲的内存数量，默认情况下该值会影响其它协议使用的net.core.wmem\_default 值，一般要低于net.core.wmem\_default的值。默认值为16384(16K)。  
max: 用于TCP socket**发送**缓冲的内存最大值。该值不会影响net.core.wmem\_max，"静态"选择参数SO\_SNDBUF则不受该值影响。默认值为131072(128K)。（**对于服务器而言，增加这个参数的值对于发送数据很有帮助**）

### net.ipv4.tcp\_wmem = 4096 16384 4194304

min：为TCP socket预留用于**接收**缓冲的内存数量，即使在内存出现紧张情况下tcp socket都至少会有这么多数量的内存用于接收缓冲，默认值为8K。  
default：为TCP socket预留用于**接收**缓冲的内存数量，默认情况下该值影响其它协议使用的net.core.wmem\_default 值。该值决定了在tcp\_adv\_win\_scale、tcp\_app\_win和tcp\_app\_win=0默认值情况下，TCP窗口大小为65535。默认值为87380  
max：用于TCP socket**接收**缓冲的内存最大值。该值不会影响 net.core.wmem\_max，"静态"选择参数 SO\_SNDBUF则不受该值影响。默认值为 128K。默认值为87380\*2 bytes。（可以看出，.max的设置最好是default的两倍,对于NAT来说主要该增加它）

### net.ipv4.tcp\_rmem = 4096 87380 4194304

默认值是31  
保留max(window/2^tcp\_app\_win, mss)数量的窗口由于应用缓冲。当为0时表示不需要缓冲。

### net.ipv4.tcp\_app\_win = 31

默认值为2  
计算缓冲开销

bytes/2^tcp\_adv\_win\_scale:(如果tcp\_adv\_win\_scale > 0) OR

bytes-bytes/2^(-tcp\_adv\_win\_scale)(如果tcp\_adv\_win\_scale <= 0）。

### net.ipv4.tcp\_adv\_win\_scale = 2

net.ipv4.tcp\_frto = 2

net.ipv4.tcp\_frto\_response = 0

缺省值为0  
允许 TCP/IP 栈适应在高吞吐量情况下低延时的情况；这个选项一般情形是的禁用。

### net.ipv4.tcp\_low\_latency = 0

开启的话，tcp会在连接关闭时也就是LAST\_ACK状态保存各种连接信息到路由缓存中，新建立的连接可以使用这些条件来初始化.。

通常这会增加总体的系统性能，但是有些时候也会引起性能下降.  
0：关闭  
1：开启

### net.ipv4.tcp\_no\_metrics\_save = 0

接收数据时是否调整接收缓存  
0：不调整  
1：调整

### net.ipv4.tcp\_moderate\_rcvbuf = 1

控制根据拥塞窗口的百分比，是否来发送相应的延迟tso frame  
0：关闭  
>0：值越大表示tso frame延迟发送可能越小.

### net.ipv4.tcp\_tso\_win\_divisor = 3

当前正在使用的拥塞控制算法.

### net.ipv4.tcp\_congestion\_control = cubic

net.ipv4.tcp\_abc = 0

是否开启tcp层路径mtu发现，自动调整tcp窗口等信  
0：关闭  
1：开启

### net.ipv4.tcp\_mtu\_probing = 0

tcp探察路径上mtu的最低边界限制, mss+TCP头部+TCP选项+IP头＋IP选项.

### net.ipv4.tcp\_base\_mss = 512

0：假定远程连接端正常发送了窗口收缩选项，即使对端没有发送.  
1：假定远程连接端有错误,没有发送相关的窗口缩放选项

### net.ipv4.tcp\_workaround\_signed\_windows = 0

net.ipv4.tcp\_challenge\_ack\_limit = 1000

net.ipv4.tcp\_limit\_output\_bytes = 262144

net.ipv4.tcp\_dma\_copybreak = 4096

如果设置满足RFC2861定义的行为，在从新开始计算拥塞窗口前延迟一些时间，这延迟的时间长度由当前rto决定.  
0：关闭  
1：开启

### net.ipv4.tcp\_slow\_start\_after\_idle = 1

是否启用cipso缓存。  
0：不启用  
1：启用

### net.ipv4.cipso\_cache\_enable = 1

限制cipso缓存项的数量，如果在缓存中新添加一行超出了这个限制，那么最旧的缓存项会被丢弃以释放出空间。

### net.ipv4.cipso\_cache\_bucket\_size = 10

是否开启cipso标志优化选项，如果开启，数据包中标志将会在32bit对齐  
0：关闭  
1：开启

### net.ipv4.cipso\_rbm\_optfmt = 0

是否开启cipso选项的严格检测  
0：不开启  
1：开启

### net.ipv4.cipso\_rbm\_strictvalid = 1

列出了tcp目前可以使用的拥塞控制算法

### net.ipv4.tcp\_available\_congestion\_control = cubic reno

列出了tcp目前允许使用的拥塞控制算法，只能在下面可用的算法中选择

### net.ipv4.tcp\_allowed\_congestion\_control = cubic reno

net.ipv4.tcp\_max\_ssthresh = 0

net.ipv4.tcp\_thin\_linear\_timeouts = 0

net.ipv4.tcp\_thin\_dupack = 0

net.ipv4.tcp\_min\_tso\_segs = 2

low：当UDP使用了低于该值的内存页面数时，UDP不会考虑释放内存。  
presure：当UDP使用了超过该值的内存页面数量时，UDP试图稳定其内存使用，进入pressure模式，当内存消耗低于low值时则退出pressure状态。  
high：允许所有UDP sockets用于排队缓冲数据报的页面量。

### net.ipv4.udp\_mem = 1526880 2035840 3053760

net.ipv4.udp\_rmem\_min = 4096

net.ipv4.udp\_wmem\_min = 4096

在该接口打开转发功能  
0：禁止  
1：允许

### net.ipv4.conf.all.forwarding = 0

是否进行多播路由。**只有内核编译有CONFIG\_MROUTE并且有路由服务程序在运行**该参数才有效。  
0：禁止  
1：允许

### net.ipv4.conf.all.mc\_forwarding = 0

收发接收ICMP重定向消息。对于主机来说默认为True，对于用作路由器时默认值为False  
0：禁止  
1：允许

### net.ipv4.conf.all.accept\_redirects = 0

仅仅接收发给默认网关列表中网关的ICMP重定向消息  
0：禁止  
1：允许

### net.ipv4.conf.all.secure\_redirects = 1

发送或接收RFC1620 共享媒体重定向。会覆盖ip\_secure\_redirects的值。  
0：禁止  
1：允许

### net.ipv4.conf.all.shared\_media = 1

1：通过反向路径回溯进行源地址验证(在RFC1812中定义)。对于单穴主机和stub网络路由器推荐使用该选项。  
0：不通过反向路径回溯进行源地址验证。

### net.ipv4.conf.all.rp\_filter = 0

允许发送重定向消息。**(路由使用)**  
0：禁止  
1：允许

### net.ipv4.conf.all.send\_redirects = 1

接收带有SRR选项的数据报。主机设为0，路由设为1

### net.ipv4.conf.all.accept\_source\_route = 0

net.ipv4.conf.all.src\_valid\_mark = 0

打开arp代理功能。  
0：禁止  
1：允许

### net.ipv4.conf.all.proxy\_arp = 0

通常,这个参数用来区分不同媒介.两个网络设备可以使用不同的值,使他们只有其中之一接收到广播包.通常,这个参数被用来配合proxy\_arp（上面）实现roxy\_arp的特性即是允许arp报文在两个不同的网络介质中转发.  
0：表示各个网络介质接受他们自己介质上的媒介  
-1：表示该媒介未知。

### net.ipv4.conf.all.medium\_id = 0

接收源地址为0.a.b.c，目的地址不是本机的数据包，是为了支持bootp服务  
0：关闭  
1：开启

### net.ipv4.conf.all.bootp\_relay = 0

记录带有不允许的地址的数据报到内核日志中。  
0：禁止  
1：允许

### net.ipv4.conf.all.log\_martians = 0

net.ipv4.conf.all.tag = 0

0：内核设置每个网络接口各自应答其地址上的arp询问。

这项看似会错误的设置却经常能非常有效，因为它增加了成功通讯的机会。在Linux主机上，每个IP地址是网络接口独立的，而非一个复合的接口。只有**在一些特殊的设置的时候，比如负载均衡的时候会带来麻烦。**1：允许多个网络介质位于同一子网段内，每个网络界面依据是否内核指派路由该数据包经过此接口来确认是否回答ARP查询(这个实现是由来源地址确定路由的时候决定的),换句话说，允许控制使用某一块网卡（通常是第一块）回应arp询问

### net.ipv4.conf.all.arp\_filter = 0

对网络接口上，本地IP地址的发出的，ARP回应，作出相应级别的限制:

确定不同程度的限制,宣布对来自本地源IP地址发出Arp请求的接口  
 0： 在任意网络接口（eth0,eth1，lo）上的任何本地地址  
 1：尽量避免不在该网络接口子网段的本地地址做出arp回应. 当发起ARP请求的源IP地址是被设置应该经由路由达到此网络接口的时候很有用.此时会检查来访IP是否为所有接口上的子网段内ip之一.如果改来访IP不属于各个网络接口上的子网段内,那么将采用级别2的方式来进行处理.   
 2：对查询目标使用最适当的本地地址.在此模式下将忽略这个IP数据包的源地址并尝试选择与能与该地址通信的本地地址.首要是选择所有的网络接口的子网中外出访问子网中包含该目标IP地址的本地地址. 如果没有合适的地址被发现,将选择当前的发送网络接口或其他的有可能接受到该ARP回应的网络接口来进行发送.

### net.ipv4.conf.all.arp\_announce = 0

定义对目标地址为本地IP的ARP询问不同的应答模式  
0：回应任何网络接口上对任何本地IP地址的arp查询请求  
1：只回答目标IP地址是来访网络接口本地地址的ARP查询请求  
2：只回答目标IP地址是来访网络接口本地地址的ARP查询请求,且来访IP必须在该网络接口的子网段内   
3：不回应该网络界面的arp请求，而只对设置的唯一和连接地址做出回应  
8：不回应所有（本地地址）的arp查询

### net.ipv4.conf.all.arp\_ignore = 0

默认对不在ARP表中的IP地址发出的APR包的处理方式  
0：不在ARP表中创建对应IP地址的表项  
1：在ARP表中创建对应IP地址的表项

### net.ipv4.conf.all.arp\_accept = 0

arp通知链操作  
0：不做任何操作  
1：当设备或硬件地址改变时自动产生一个arp请求

### net.ipv4.conf.all.arp\_notify = 0

回应代理ARP的数据包从接收到此代理ARP请求的网络接口出去

### net.ipv4.conf.all.proxy\_arp\_pvlan = 0

禁止internet协议安全性加密  
**0：禁止禁止  
1：开启禁止**

### net.ipv4.conf.all.disable\_xfrm = 0

禁止internet协议安全性验证  
**0：禁止禁止——？？为什么开启要这样说？？  
1：开启禁止——？？为什么禁止要这样说？？**

### net.ipv4.conf.all.disable\_policy = 0

net.ipv4.conf.all.force\_igmp\_version = 0

0：当接口的主IP地址被移除时，删除所有次IP地址  
1：当接口的主IP地址被移除时，将次IP地址提升为主IP地址

### net.ipv4.conf.all.promote\_secondaries = 0

设置是否允许接收从本机IP地址上发送给本机的数据包  
0：不允许  
1：允许

### net.ipv4.conf.all.accept\_local = 0

net.ipv4.conf.all.route\_localnet = 0

[net.ipv4.conf.all.forwarding](#_net.ipv4.conf.all.forwarding = 0)

### net.ipv4.conf.default.forwarding = 0

[net.ipv4.conf.all.mc\_forwarding](#_net.ipv4.conf.all.mc_forwarding = 0)

### net.ipv4.conf.default.mc\_forwarding = 0

收发接收ICMP重定向消息。对于主机来说默认为True，对于用作路由器时默认值为False  
0：禁止  
1：允许

### net.ipv4.conf.default.accept\_redirects = 1

[net.ipv4.conf.all.secure\_redirects](#_net.ipv4.conf.all.secure_redirects = 1)

### net.ipv4.conf.default.secure\_redirects = 1

[net.ipv4.conf.all.shared\_media](#_net.ipv4.conf.all.shared_media = 1)

### net.ipv4.conf.default.shared\_media = 1

[net.ipv4.conf.all.rp\_filter](#_net.ipv4.conf.all.rp_filter = 0)

### net.ipv4.conf.default.rp\_filter = 1

[net.ipv4.conf.all.send\_redirects](#_net.ipv4.conf.all.send_redirects = 1)

### net.ipv4.conf.default.send\_redirects = 1

接收带有SRR选项的数据报。主机设为0，路由设为1

### net.ipv4.conf.default.accept\_source\_route = 0

net.ipv4.conf.default.src\_valid\_mark = 0

[net.ipv4.conf.all.proxy\_arp](#_net.ipv4.conf.all.proxy_arp = 0)

### net.ipv4.conf.default.proxy\_arp = 0

[net.ipv4.conf.all.medium\_id](#_net.ipv4.conf.all.medium_id = 0)

### net.ipv4.conf.default.medium\_id = 0

见[net.ipv4.conf.all.bootp\_relay](#_net.ipv4.conf.all.bootp_relay = 0)

### net.ipv4.conf.default.bootp\_relay = 0

[net.ipv4.conf.all.log\_martians](#_net.ipv4.conf.all.log_martians = 0)

### net.ipv4.conf.default.log\_martians = 0

net.ipv4.conf.default.tag = 0

见[net.ipv4.conf.all.arp\_filter](#_net.ipv4.conf.all.arp_filter = 0)

### net.ipv4.conf.default.arp\_filter = 0

作用见[net.ipv4.conf.all.arp\_announce](#_net.ipv4.conf.all.arp_announce = 0)

### net.ipv4.conf.default.arp\_announce = 0

见[net.ipv4.conf.all.arp\_ignore](#_net.ipv4.conf.all.arp_ignore = 0)

### net.ipv4.conf.default.arp\_ignore = 0

默认对不在ARP表中的IP地址发出的APR包的处理方式  
0：不在ARP表中创建对应IP地址的表项  
1：在ARP表中创建对应IP地址的表项

### net.ipv4.conf.default.arp\_accept = 0

见[net.ipv4.conf.all.arp\_notify](#_net.ipv4.conf.all.arp_notify = 0" \o "net.ipv4.conf.all.arp_notify)

### net.ipv4.conf.default.arp\_notify = 0

[net.ipv4.conf.all.proxy\_arp\_pvlan](#_net.ipv4.conf.all.proxy_arp_pvlan = 0)

### net.ipv4.conf.default.proxy\_arp\_pvlan = 0

[net.ipv4.conf.all.disable\_xfrm](#_net.ipv4.conf.all.disable_xfrm = 0)

### net.ipv4.conf.default.disable\_xfrm = 0

[net.ipv4.conf.all.disable\_policy](#_net.ipv4.conf.all.disable_policy = 0)

### net.ipv4.conf.default.disable\_policy = 0

net.ipv4.conf.default.force\_igmp\_version = 0

[net.ipv4.conf.all.promote\_secondaries](#_net.ipv4.conf.all.promote_secondaries = 0)

### net.ipv4.conf.default.promote\_secondaries = 0

设置是否允许接收从本机IP地址上发送给本机的数据包  
0：不允许  
1：允许

### net.ipv4.conf.default.accept\_local = 0

net.ipv4.conf.default.route\_localnet = 0

net.ipv4.conf.lo.forwarding = 0

net.ipv4.conf.lo.mc\_forwarding = 0

net.ipv4.conf.lo.accept\_redirects = 1

net.ipv4.conf.lo.secure\_redirects = 1

net.ipv4.conf.lo.shared\_media = 1

net.ipv4.conf.lo.rp\_filter = 1

net.ipv4.conf.lo.send\_redirects = 1

net.ipv4.conf.lo.accept\_source\_route = 0

net.ipv4.conf.lo.src\_valid\_mark = 0

net.ipv4.conf.lo.proxy\_arp = 0

net.ipv4.conf.lo.medium\_id = 0

net.ipv4.conf.lo.bootp\_relay = 0

net.ipv4.conf.lo.log\_martians = 0

net.ipv4.conf.lo.tag = 0

net.ipv4.conf.lo.arp\_filter = 0

net.ipv4.conf.lo.arp\_announce = 0

net.ipv4.conf.lo.arp\_ignore = 0

net.ipv4.conf.lo.arp\_accept = 0

net.ipv4.conf.lo.arp\_notify = 0

net.ipv4.conf.lo.proxy\_arp\_pvlan = 0

net.ipv4.conf.lo.disable\_xfrm = 1

net.ipv4.conf.lo.disable\_policy = 1

net.ipv4.conf.lo.force\_igmp\_version = 0

net.ipv4.conf.lo.promote\_secondaries = 0

net.ipv4.conf.lo.accept\_local = 0

net.ipv4.conf.lo.route\_localnet = 0

net.ipv4.conf.eth0.forwarding = 0

net.ipv4.conf.eth0.mc\_forwarding = 0

net.ipv4.conf.eth0.accept\_redirects = 1

net.ipv4.conf.eth0.secure\_redirects = 1

net.ipv4.conf.eth0.shared\_media = 1

net.ipv4.conf.eth0.rp\_filter = 1

net.ipv4.conf.eth0.send\_redirects = 1

net.ipv4.conf.eth0.accept\_source\_route = 0

net.ipv4.conf.eth0.src\_valid\_mark = 0

net.ipv4.conf.eth0.proxy\_arp = 0

net.ipv4.conf.eth0.medium\_id = 0

net.ipv4.conf.eth0.bootp\_relay = 0

net.ipv4.conf.eth0.log\_martians = 0

net.ipv4.conf.eth0.tag = 0

net.ipv4.conf.eth0.arp\_filter = 0

net.ipv4.conf.eth0.arp\_announce = 0

net.ipv4.conf.eth0.arp\_ignore = 0

net.ipv4.conf.eth0.arp\_accept = 0

net.ipv4.conf.eth0.arp\_notify = 0

net.ipv4.conf.eth0.proxy\_arp\_pvlan = 0

net.ipv4.conf.eth0.disable\_xfrm = 0

net.ipv4.conf.eth0.disable\_policy = 0

net.ipv4.conf.eth0.force\_igmp\_version = 0

net.ipv4.conf.eth0.promote\_secondaries = 0

net.ipv4.conf.eth0.accept\_local = 0

net.ipv4.conf.eth0.route\_localnet = 0

net.ipv4.conf.eth1.forwarding = 0

net.ipv4.conf.eth1.mc\_forwarding = 0

net.ipv4.conf.eth1.accept\_redirects = 1

net.ipv4.conf.eth1.secure\_redirects = 1

net.ipv4.conf.eth1.shared\_media = 1

net.ipv4.conf.eth1.rp\_filter = 1

net.ipv4.conf.eth1.send\_redirects = 1

net.ipv4.conf.eth1.accept\_source\_route = 0

net.ipv4.conf.eth1.src\_valid\_mark = 0

net.ipv4.conf.eth1.proxy\_arp = 0

net.ipv4.conf.eth1.medium\_id = 0

net.ipv4.conf.eth1.bootp\_relay = 0

net.ipv4.conf.eth1.log\_martians = 0

net.ipv4.conf.eth1.tag = 0

net.ipv4.conf.eth1.arp\_filter = 0

net.ipv4.conf.eth1.arp\_announce = 0

net.ipv4.conf.eth1.arp\_ignore = 0

net.ipv4.conf.eth1.arp\_accept = 0

net.ipv4.conf.eth1.arp\_notify = 0

net.ipv4.conf.eth1.proxy\_arp\_pvlan = 0

net.ipv4.conf.eth1.disable\_xfrm = 0

net.ipv4.conf.eth1.disable\_policy = 0

net.ipv4.conf.eth1.force\_igmp\_version = 0

net.ipv4.conf.eth1.promote\_secondaries = 0

net.ipv4.conf.eth1.accept\_local = 0

net.ipv4.conf.eth1.route\_localnet = 0

是否打开ipv4的IP转发。  
0：禁止  
1：打开

### net.ipv4.ip\_forward = 0

net.ipv4.xfrm4\_gc\_thresh = 4194304

表示用于重组IP分段的内存分配最高值

一旦达到最高内存分配值，其它分段将被丢弃，直到达到最低内存分配值。

### net.ipv4.ipfrag\_high\_thresh = 4194304

表示用于重组IP分段的内存分配最低值

### net.ipv4.ipfrag\_low\_thresh = 3145728

表示一个IP分段在内存中保留多少秒

### net.ipv4.ipfrag\_time = 30

忽略所有接收到的icmp echo请求的包(会导致机器无法ping通)  
0：不忽略  
1：忽略

### net.ipv4.icmp\_echo\_ignore\_all = 0

忽略所有接收到的icmp echo请求的广播  
0：不忽略  
1：忽略

### net.ipv4.icmp\_echo\_ignore\_broadcasts = 1

某些路由器违背RFC1122标准，其对广播帧发送伪造的响应来应答。这种违背行为通常会被以警告的方式记录在系统日志中。  
0：记录到系统日志中  
1：忽略

### net.ipv4.icmp\_ignore\_bogus\_error\_responses = 1

当前为 ICMP 错误消息选择源地址的方式

0:则使用退出接口的主地址发送ICMP错误消息

1:消息将与接收导致ICMP错误的数据包的接口的主地址一起发送

请注意，如果所选接口不存在主地址，则无论此设置如何，都将使用具有主地址的第一个非环回接口的主地址

### net.ipv4.icmp\_errors\_use\_inbound\_ifaddr = 0

限制发向特定目标的匹配icmp\_ratemask的ICMP数据报的最大速率。配合icmp\_ratemask使用。  
0：没有任何限制  
>0：表示指定时间内中允许发送的个数。(以jiffies为单位)

### net.ipv4.icmp\_ratelimit = 1000

在这里匹配的ICMP被icmp\_ratelimit参数限制速率.  
匹配的标志位: ＩＨＧＦＥＤＣＢＡ９８７６５４３２１０  
默认的掩码值: ００００００１１００００００１１０００ (6168)  
0 Echo Reply  
3 Destination Unreachable \*  
4 Source Quench \*  
5 Redirect  
8 Echo Request  
B Time Exceeded \*  
C Parameter Problem \*  
D Timestamp Request  
E Timestamp Reply  
F Info Request  
G Info Reply  
H Address Mask Request  
I Address Mask Reply  
\* 号的被默认限速

### net.ipv4.icmp\_ratemask = 6168

net.ipv4.rt\_cache\_rebuild\_count = 4

net.ipv4.ping\_group\_range = 1 0

该文件表示在全局范围内关闭路径MTU探测功能

### net.ipv4.ip\_no\_pmtu\_disc = 0

net.ipv4.ip\_forward\_use\_pmtu = 0

hash表中ip碎片队列的重建延迟.(单位 秒)

### net.ipv4.ipfrag\_secret\_interval = 600

相同的源地址ip碎片数据报的最大数量. 这个变量表示在ip碎片被添加到队列前要作额外的检查.如果超过定义的数量的ip碎片从一个相同源地址到达，那么假定这个队列的ip碎片有丢失，已经存在的ip碎片队列会被丢弃

如果为0关闭检查

### net.ipv4.ipfrag\_max\_dist = 64

net.ipv6.neigh.default.mcast\_solicit = 3

net.ipv6.neigh.default.ucast\_solicit = 3

net.ipv6.neigh.default.app\_solicit = 0

net.ipv6.neigh.default.delay\_first\_probe\_time = 5

net.ipv6.neigh.default.gc\_stale\_time = 60

net.ipv6.neigh.default.unres\_qlen = 3

net.ipv6.neigh.default.proxy\_qlen = 64

net.ipv6.neigh.default.anycast\_delay = 99

net.ipv6.neigh.default.proxy\_delay = 79

net.ipv6.neigh.default.locktime = 0

net.ipv6.neigh.default.retrans\_time\_ms = 1000

net.ipv6.neigh.default.base\_reachable\_time\_ms = 30000

net.ipv6.neigh.default.gc\_interval = 30

net.ipv6.neigh.default.gc\_thresh1 = 128

net.ipv6.neigh.default.gc\_thresh2 = 512

net.ipv6.neigh.default.gc\_thresh3 = 1024

net.ipv6.neigh.lo.mcast\_solicit = 3

net.ipv6.neigh.lo.ucast\_solicit = 3

net.ipv6.neigh.lo.app\_solicit = 0

net.ipv6.neigh.lo.delay\_first\_probe\_time = 5

net.ipv6.neigh.lo.gc\_stale\_time = 60

net.ipv6.neigh.lo.unres\_qlen = 3

net.ipv6.neigh.lo.proxy\_qlen = 64

net.ipv6.neigh.lo.anycast\_delay = 99

net.ipv6.neigh.lo.proxy\_delay = 79

net.ipv6.neigh.lo.locktime = 0

net.ipv6.neigh.lo.retrans\_time\_ms = 1000

net.ipv6.neigh.lo.base\_reachable\_time\_ms = 30000

net.ipv6.neigh.eth0.mcast\_solicit = 3

net.ipv6.neigh.eth0.ucast\_solicit = 3

net.ipv6.neigh.eth0.app\_solicit = 0

net.ipv6.neigh.eth0.delay\_first\_probe\_time = 5

net.ipv6.neigh.eth0.gc\_stale\_time = 60

net.ipv6.neigh.eth0.unres\_qlen = 3

net.ipv6.neigh.eth0.proxy\_qlen = 64

net.ipv6.neigh.eth0.anycast\_delay = 99

net.ipv6.neigh.eth0.proxy\_delay = 79

net.ipv6.neigh.eth0.locktime = 0

net.ipv6.neigh.eth0.retrans\_time\_ms = 1000

net.ipv6.neigh.eth0.base\_reachable\_time\_ms = 30000

net.ipv6.neigh.eth1.mcast\_solicit = 3

net.ipv6.neigh.eth1.ucast\_solicit = 3

net.ipv6.neigh.eth1.app\_solicit = 0

net.ipv6.neigh.eth1.delay\_first\_probe\_time = 5

net.ipv6.neigh.eth1.gc\_stale\_time = 60

net.ipv6.neigh.eth1.unres\_qlen = 3

net.ipv6.neigh.eth1.proxy\_qlen = 64

net.ipv6.neigh.eth1.anycast\_delay = 99

net.ipv6.neigh.eth1.proxy\_delay = 79

net.ipv6.neigh.eth1.locktime = 0

net.ipv6.neigh.eth1.retrans\_time\_ms = 1000

net.ipv6.neigh.eth1.base\_reachable\_time\_ms = 30000

net.ipv6.xfrm6\_gc\_thresh = 2048

所有网络接口开启ipv6转发  
0：关闭  
1：开启

### net.ipv6.conf.all.forwarding = 0

缺省hop限制

### net.ipv6.conf.all.hop\_limit = 64

ipv6的最大传输单元

### net.ipv6.conf.all.mtu = 1280

接受IPv6路由通告.并且根据得到的信息自动设定.  
0：不接受路由通告  
1：当forwarding禁止时接受路由通告  
2：任何情况下都接受路由通告

### net.ipv6.conf.all.accept\_ra = 1

是否接受ICMPv6重定向包  
0：拒绝接受ICMPv6，当forwarding=1时，此值会自动设置为0  
1：启动接受ICMPv6，当forwarding=0时，此值会自动设置为1

### net.ipv6.conf.all.accept\_redirects = 1

net.ipv6.conf.all.autoconf = 1

接口增加ipv6地址时，发送几次DAD包

### net.ipv6.conf.all.dad\_transmits = 1

假定没有路由的情况下发送的请求个数

### net.ipv6.conf.all.router\_solicitations = 3

在每个路由请求之间的等待时间(秒).

### net.ipv6.conf.all.router\_solicitation\_interval = 4

在发送路由请求之前的等待时间(秒).

### net.ipv6.conf.all.router\_solicitation\_delay = 1

net.ipv6.conf.all.force\_mld\_version = 0

net.ipv6.conf.all.use\_tempaddr = 0

net.ipv6.conf.all.temp\_valid\_lft = 604800

net.ipv6.conf.all.temp\_prefered\_lft = 86400

尝试生成临时地址的次数

### net.ipv6.conf.all.regen\_max\_retry = 5

DESYNC\_FACTOR的最大值，DESYNC\_FACTOR是一个随机数，用于防止客户机在同一时间生成新的地址

### net.ipv6.conf.all.max\_desync\_factor = 600

所有网络接口自动配置IP地址的数量最大值  
0：不限制  
>0：最大值

### net.ipv6.conf.all.max\_addresses = 16

是否接受ipv6路由器发出的默认路由设置  
0：不接受  
1：接受

### net.ipv6.conf.all.accept\_ra\_defrtr = 1

当accept\_ra开启时此选项会自动开启，关闭时则会关闭

### net.ipv6.conf.all.accept\_ra\_pinfo = 1

### net.ipv6.conf.all.accept\_ra\_rtr\_pref = 1

路由器探测间隔(秒)

### net.ipv6.conf.all.router\_probe\_interval = 60

路由通告中路由信息前缀的最大长度

### net.ipv6.conf.all.accept\_ra\_rt\_info\_max\_plen = 0

此功能类似于ipv4的nat，可将内网的包转发到外网，外网不能主动发给内网。  
0：关闭  
1：开启

### net.ipv6.conf.all.proxy\_ndp = 0

接收带有SRR选项的数据报。主机设为0，路由设为1

### net.ipv6.conf.all.accept\_source\_route = 0

是否启用optimistic DAD(乐观地进行重复地址检查)  
0：关闭  
1：开启

### net.ipv6.conf.all.optimistic\_dad = 0

是否使用多路广播进行路由选择，需要内核编译时开启了CONFIG\_MROUTE选项并且开启了多路广播路由选择的后台daemon  
0：关闭  
1：开启

### net.ipv6.conf.all.mc\_forwarding = 0

是否禁用ipv6  
0：不禁用  
1：禁用

### net.ipv6.conf.all.disable\_ipv6 = 0

0：取消DAD功能  
1：启用DAD功能，但link-local地址冲突时，不关闭ipv6功能  
2：启用DAD功能，但link-local地址冲突时，关闭ipv6功能

### net.ipv6.conf.all.accept\_dad = 1

net.ipv6.conf.default.forwarding = 0

net.ipv6.conf.default.hop\_limit = 64

net.ipv6.conf.default.mtu = 1280

net.ipv6.conf.default.accept\_ra = 1

net.ipv6.conf.default.accept\_redirects = 1

net.ipv6.conf.default.autoconf = 1

net.ipv6.conf.default.dad\_transmits = 1

net.ipv6.conf.default.router\_solicitations = 3

net.ipv6.conf.default.router\_solicitation\_interval = 4

net.ipv6.conf.default.router\_solicitation\_delay = 1

net.ipv6.conf.default.force\_mld\_version = 0

net.ipv6.conf.default.use\_tempaddr = 0

net.ipv6.conf.default.temp\_valid\_lft = 604800

net.ipv6.conf.default.temp\_prefered\_lft = 86400

net.ipv6.conf.default.regen\_max\_retry = 5

net.ipv6.conf.default.max\_desync\_factor = 600

net.ipv6.conf.default.max\_addresses = 16

net.ipv6.conf.default.accept\_ra\_defrtr = 1

net.ipv6.conf.default.accept\_ra\_pinfo = 1

net.ipv6.conf.default.accept\_ra\_rtr\_pref = 1

net.ipv6.conf.default.router\_probe\_interval = 60

net.ipv6.conf.default.accept\_ra\_rt\_info\_max\_plen = 0

net.ipv6.conf.default.proxy\_ndp = 0

[net.ipv6.conf.all.accept\_source\_route](#_net.ipv6.conf.all.accept_source_route = 0)

### net.ipv6.conf.default.accept\_source\_route = 0

net.ipv6.conf.default.optimistic\_dad = 0

net.ipv6.conf.default.mc\_forwarding = 0

net.ipv6.conf.default.disable\_ipv6 = 0

net.ipv6.conf.default.accept\_dad = 1

net.ipv6.conf.lo.forwarding = 0

net.ipv6.conf.lo.hop\_limit = 64

net.ipv6.conf.lo.mtu = 65536

net.ipv6.conf.lo.accept\_ra = 1

net.ipv6.conf.lo.accept\_redirects = 1

net.ipv6.conf.lo.autoconf = 1

net.ipv6.conf.lo.dad\_transmits = 1

net.ipv6.conf.lo.router\_solicitations = 3

net.ipv6.conf.lo.router\_solicitation\_interval = 4

net.ipv6.conf.lo.router\_solicitation\_delay = 1

net.ipv6.conf.lo.force\_mld\_version = 0

net.ipv6.conf.lo.use\_tempaddr = -1

net.ipv6.conf.lo.temp\_valid\_lft = 604800

net.ipv6.conf.lo.temp\_prefered\_lft = 86400

net.ipv6.conf.lo.regen\_max\_retry = 5

net.ipv6.conf.lo.max\_desync\_factor = 600

net.ipv6.conf.lo.max\_addresses = 16

net.ipv6.conf.lo.accept\_ra\_defrtr = 1

net.ipv6.conf.lo.accept\_ra\_pinfo = 1

net.ipv6.conf.lo.accept\_ra\_rtr\_pref = 1

net.ipv6.conf.lo.router\_probe\_interval = 60

net.ipv6.conf.lo.accept\_ra\_rt\_info\_max\_plen = 0

net.ipv6.conf.lo.proxy\_ndp = 0

net.ipv6.conf.lo.accept\_source\_route = 0

net.ipv6.conf.lo.optimistic\_dad = 0

net.ipv6.conf.lo.mc\_forwarding = 0

net.ipv6.conf.lo.disable\_ipv6 = 0

net.ipv6.conf.lo.accept\_dad = -1

net.ipv6.conf.eth0.forwarding = 0

net.ipv6.conf.eth0.hop\_limit = 64

net.ipv6.conf.eth0.mtu = 1500

net.ipv6.conf.eth0.accept\_ra = 1

net.ipv6.conf.eth0.accept\_redirects = 1

net.ipv6.conf.eth0.autoconf = 1

net.ipv6.conf.eth0.dad\_transmits = 1

net.ipv6.conf.eth0.router\_solicitations = 3

net.ipv6.conf.eth0.router\_solicitation\_interval = 4

net.ipv6.conf.eth0.router\_solicitation\_delay = 1

net.ipv6.conf.eth0.force\_mld\_version = 0

net.ipv6.conf.eth0.use\_tempaddr = 0

net.ipv6.conf.eth0.temp\_valid\_lft = 604800

net.ipv6.conf.eth0.temp\_prefered\_lft = 86400

net.ipv6.conf.eth0.regen\_max\_retry = 5

net.ipv6.conf.eth0.max\_desync\_factor = 600

net.ipv6.conf.eth0.max\_addresses = 16

net.ipv6.conf.eth0.accept\_ra\_defrtr = 1

net.ipv6.conf.eth0.accept\_ra\_pinfo = 1

net.ipv6.conf.eth0.accept\_ra\_rtr\_pref = 1

net.ipv6.conf.eth0.router\_probe\_interval = 60

net.ipv6.conf.eth0.accept\_ra\_rt\_info\_max\_plen = 0

net.ipv6.conf.eth0.proxy\_ndp = 0

net.ipv6.conf.eth0.accept\_source\_route = 0

net.ipv6.conf.eth0.optimistic\_dad = 0

net.ipv6.conf.eth0.mc\_forwarding = 0

net.ipv6.conf.eth0.disable\_ipv6 = 0

net.ipv6.conf.eth0.accept\_dad = 1

net.ipv6.conf.eth1.forwarding = 0

net.ipv6.conf.eth1.hop\_limit = 64

net.ipv6.conf.eth1.mtu = 1500

net.ipv6.conf.eth1.accept\_ra = 1

net.ipv6.conf.eth1.accept\_redirects = 1

net.ipv6.conf.eth1.autoconf = 1

net.ipv6.conf.eth1.dad\_transmits = 1

net.ipv6.conf.eth1.router\_solicitations = 3

net.ipv6.conf.eth1.router\_solicitation\_interval = 4

net.ipv6.conf.eth1.router\_solicitation\_delay = 1

net.ipv6.conf.eth1.force\_mld\_version = 0

net.ipv6.conf.eth1.use\_tempaddr = 0

net.ipv6.conf.eth1.temp\_valid\_lft = 604800

net.ipv6.conf.eth1.temp\_prefered\_lft = 86400

net.ipv6.conf.eth1.regen\_max\_retry = 5

net.ipv6.conf.eth1.max\_desync\_factor = 600

net.ipv6.conf.eth1.max\_addresses = 16

net.ipv6.conf.eth1.accept\_ra\_defrtr = 1

net.ipv6.conf.eth1.accept\_ra\_pinfo = 1

net.ipv6.conf.eth1.accept\_ra\_rtr\_pref = 1

net.ipv6.conf.eth1.router\_probe\_interval = 60

net.ipv6.conf.eth1.accept\_ra\_rt\_info\_max\_plen = 0

net.ipv6.conf.eth1.proxy\_ndp = 0

net.ipv6.conf.eth1.accept\_source\_route = 0

net.ipv6.conf.eth1.optimistic\_dad = 0

net.ipv6.conf.eth1.mc\_forwarding = 0

net.ipv6.conf.eth1.disable\_ipv6 = 0

net.ipv6.conf.eth1.accept\_dad = 1

net.ipv6.ip6frag\_high\_thresh = 4194304

net.ipv6.ip6frag\_low\_thresh = 3145728

net.ipv6.ip6frag\_time = 60

net.ipv6.route.gc\_thresh = 1024

net.ipv6.route.max\_size = 16384

net.ipv6.route.gc\_min\_interval = 0

net.ipv6.route.gc\_timeout = 60

net.ipv6.route.gc\_interval = 30

net.ipv6.route.gc\_elasticity = 0

net.ipv6.route.mtu\_expires = 600

net.ipv6.route.min\_adv\_mss = 1

net.ipv6.route.gc\_min\_interval\_ms = 500

net.ipv6.icmp.ratelimit = 1000

默认监听ipv6端口(不管监听与否，都与是否关闭ipv4监听无关)  
0：不监听  
1：监听

### net.ipv6.bindv6only = 0

net.ipv6.ip6frag\_secret\_interval = 600

net.ipv6.mld\_max\_msf = 64

允许域套接字中数据包的最大个数，在初始化unix域套接字时的默认值.  
在调用listen函数时第二个参数会复盖这个值.

### net.unix.max\_dgram\_qlen = 10

## 8.sunrpc

sunrpc.rpc\_debug = 0

sunrpc.nfs\_debug = 0

sunrpc.nfsd\_debug = 0

sunrpc.nlm\_debug = 0

sunrpc.transports = tcp 1048576

sunrpc.transports = udp 32768

sunrpc.transports = tcp-bc 1048576

sunrpc.udp\_slot\_table\_entries = 16

sunrpc.tcp\_slot\_table\_entries = 2

sunrpc.tcp\_max\_slot\_table\_entries = 65536

sunrpc.min\_resvport = 665

sunrpc.max\_resvport = 1023

默认值是 60  
对于本端断开的socket连接，TCP保持在FIN-WAIT-2状态的时间。对方可能会断开连接或一直不结束连接或不可预料的进程死亡。默认值为 60 秒。过去在2.2版本的内核中是 180 秒。您可以设置该值﹐但需要注意﹐如果您的机器为负载很重的web服务器﹐您可能要冒内存被大量无效数据报填满的风险﹐FIN-WAIT-2 sockets 的危险性低于 FIN-WAIT-1 ﹐因为它们最多只吃 1.5K 的内存﹐但是它们存在时间更长。另外参考 [tcp\_max\_orphans](#_net.ipv4.tcp_max_orphans = 262144)。

### sunrpc.tcp\_fin\_timeout = 15

## 9.vm-内存参数

是否允许内存的过量分配，允许进程分配比它实际使用的更多的内存。  
0：当用户申请内存的时候，内核会去检查是否有这么大的内存空间，当超过地址空间会被拒绝  
1：内核始终认为，有足够大的内存空间，直到它用完了为止  
2：内核禁止任何形式的过量分配内存

### vm.overcommit\_memory = 0

用于控制如何处理out-of-memory，可选值包括0/1/2  
 0：当内存不足时内核调用OOM killer杀死一些rogue进程，每个进程描述符都有一个oom\_score标示，oom killer会选择oom\_score较大的进程  
 1：发生了OOM以后，如果有mempolicy/cpusets的进程限制，而这些nodes导致了内存问题的时候，OOM Killer会干掉这些中的一个，系统也会恢复  
 2：OOM后必然panic

### vm.panic\_on\_oom = 0

决定在oom的时候，oom killer杀哪些进程  
 非0：它会扫描进程队列，然后将可能导致内存溢出的进程杀掉，也就是占用内存最大的进程  
 0：它只杀掉导致oom的那个进程，避免了进程队列的扫描，但是释放的内存大小有限

### vm.oom\_kill\_allocating\_task = 0

待补充

### vm.extfrag\_threshold = 500

如果启用，在内核执行OOM-killing时会打印系统内进程的信息（不包括内核线程），信息包括pid、uid、tgid、vm size、rss、nr\_ptes，swapents，oom\_score\_adj和进程名称。这些信息可以帮助找出为什么OOM killer被执行，找到导致OOM的进程，以及了解为什么进程会被选中。  
0：不打印系统内进程信息  
1：打印系统内进程信息

### vm.oom\_dump\_tasks = 1

vm.would\_have\_oomkilled = 0

内存可过量分配的百分比。

### vm.overcommit\_ratio = 50

内存可过量分配的数量(KB)

### vm.overcommit\_kbytes = 0

参数控制一次写入或读出swap分区的页面数量。

它是一个对数值，如果设置为0，表示1页；如果设置为1，表示2页；如果设置为2，则表示4页。

### vm.page-cluster = 3

当脏页所占的百分比（相对于所有可用内存，即空闲内存页+可回收内存页）达到dirty\_background\_ratio时，write调用会唤醒内核的flusher线程开始回写脏页数据，直到脏页比例低于此值，与dirty\_ratio不同，write调用此时并不会阻塞。

### vm.dirty\_background\_ratio = 10

当脏页所占的内存数量超过dirty\_background\_bytes时，内核的flusher线程开始回写脏页

### vm.dirty\_background\_bytes = 0

脏页所占的百分比（相对于所有可用内存，即空闲内存页+可回收内存页）达到dirty\_ratio时，write调用会唤醒内核的flusher线程开始回写脏页数据，直到脏页比例低于此值，注意write调用此时会阻塞

### vm.dirty\_ratio = 20

当脏页所占的内存数量达到dirty\_bytes时，执行磁盘写操作的进程自己开始回写脏数据。  
注意： dirty\_bytes参数和 dirty\_ratio参数是相对的，只能指定其中一个。当其中一个参数文件被写入时，会立即开始计算脏页限制，并且会将另一个参数的值清零

### vm.dirty\_bytes = 0

设置flusher内核线程唤醒的间隔，此线程用于将脏页回写到磁盘，单位是0.01秒

### vm.dirty\_writeback\_centisecs = 500

脏数据的过期时间，超过该时间后内核的flusher线程被唤醒时会将脏数据回写到磁盘上，单位是0.01秒。

### vm.dirty\_expire\_centisecs = 3000

只读文件，保存了当前正在运行的pdflush线程的数量

### vm.nr\_pdflush\_threads = 0

表示尽量使用内存，减少使用磁盘swap交换分区，内存速度明显高于磁盘一个数量级

该值越高则linux越倾向于将部分长期没有用到的页swap，即便有足够空余物理内存(1~100)

### vm.swappiness = 1

大页的最小数目，需要连续的物理内存；**oracle使用大页可以降低TLB的开销，节约内存和CPU资源，但要同时设置memlock且保证其大于大页**；其与11gAMM不兼容

### vm.nr\_hugepages = 0

与nr\_hugepages（上面那个）类似，但只用于numa架构，配合numactl调整每个node的大页数量

### vm.nr\_hugepages\_mempolicy = 0

指定组ID，拥有该gid的用户可以使用大页创建SysV共享内存段

### vm.hugetlb\_shm\_group = 0

用来控制是否可以从ZONE\_MOVABLE内存域中分配大页面。如果设置为非零，大页面可以从ZONE\_MOVABLE内存域分配。ZONE\_MOVABLE内存域只有在指定了kernelcore启动参数的情况下才会创建，如果没有指定kernelcore启动参数， hugepages\_treat\_as\_movable参数则没有效果。

### vm.hugepages\_treat\_as\_movable = 0

保留于紧急使用的大页数

系统可分配最大大页数= [nr\_hugepages](#_vm.nr_hugepages = 0) + nr\_overcommit\_hugepages

### vm.nr\_overcommit\_hugepages = 0

决定了内核保护这些低端内存域的强度。预留的内存值和lowmem\_reserve\_ratio数组中的值是倒数关系，如果值是256，则代表1/256，即为0.39%的zone内存大小。如果想要预留更多页，应该设更小一点的值。

### vm.lowmem\_reserve\_ratio = 256 256 32

写入数值可以使内核释放page\_cache，dentries和inodes缓存所占的内存。  
1：只释放page\_cache  
2：只释放dentries和inodes缓存  
3：释放page\_cache、dentries和inodes缓存

### vm.drop\_caches = 0

每个内存区保留的内存大小(KB)——66MB

### vm.min\_free\_kbytes = 67584

vm.extra\_free\_kbytes = 0

vm.unmap\_area\_factor = 0

vm.meminfo\_legacy\_layout = 1

每个CPU能从每个zone所能分配到的pages的最大值(单位每个zone的1/X)，0为不限制

### vm.percpu\_pagelist\_fraction = 0

定义了一个进程能拥有的最多的内存区域

### vm.max\_map\_count = 65530

设置开启laptop mode，此模式主要是通过降低硬盘的转速来延长电池的续航时间。  
0：关闭  
1：启动

### vm.laptop\_mode = 0

如果设置的是非零值，则会启用块I/O调试

### vm.block\_dump = 0

表示内核回收用于directory和inode cache内存的倾向；

缺省值100表示内核将根据pagecache和swapcache，把directory和inode cache保持在一个合理的百分比；

该值低于100，将导致内核倾向于保留directory和inode cache；

该值超过100，将导致内核倾向于回收directory和inode cache

### vm.vfs\_cache\_pressure = 100

进程地址空间内存布局模式  
0：经典布局  
1：新布局  
对于64位系统，默认采用经典布局

### vm.legacy\_va\_layout = 0

参数只有在启用CONFIG\_NUMA选项时才有效,zone\_reclaim\_mode用来控制在内存域OOM时，如何来回收内存。  
0：禁止内存域回收，从其他zone分配内存  
1：启用内存域回收  
2：通过回写脏页回收内存  
4：通过swap回收内存

### vm.zone\_reclaim\_mode = 0

只有在当前内存域中处于[zone\_reclaim\_mode](#_vm.zone_reclaim_mode = 0)（上面那个）允许回收状态的内存页所占的百分比超过min\_unmapped\_ratio时，内存域才会执行回收操作。

### vm.min\_unmapped\_ratio = 1

只在numa架构上使用

如果一个内存域中可以回收的slab页面所占的百分比（应该是相对于当前内存域的所有页面）超过min\_slab\_ratio，在回收区的slabs会被回收。

这样可以确保即使在很少执行全局回收的NUMA系统中，slab的增长也是可控的。

### vm.min\_slab\_ratio = 5

VM信息更新频率(以秒为单位)

### vm.stat\_interval = 1

指定用户进程通过mmap可使用的最小虚拟内存地址，以避免其在低地址空间产生映射导致安全问题；如果非0，则不允许mmap到NULL页，而此功能可在出现NULL指针时调试Kernel；mmap用于将文件映射至内存；  
该设置意味着禁止用户进程访问low 4k地址空间

### vm.mmap\_min\_addr = 4096

设置内核选择zonelist的模式：  
0：让内核智能选择使用Node或Zone方式的zonelist  
1：选择Node方式的zonelist，Node(0) ZONE\_NORMAL -> Node(0) ZONE\_DMA -> Node(1) ZONE\_NORMAL  
2：选择Zone方式的，Node(0) ZONE\_NORMAL -> Node(1) ZONE\_NORMAL -> Node(0) ZONE\_DMA

### vm.numa\_zonelist\_order = default

vm.scan\_unevictable\_pages = 0

控制发生某个内核无法处理的内存错误发生的时候，如何去杀掉这个进程。

1：在发现内存错误的时候，就会把所有拥有此内存页的进程都杀掉  
 0：只是对这部分页进行unmap，然后把第一个试图进入这个页的进程杀掉

### vm.memory\_failure\_early\_kill = 0

是否开启内存错误恢复机制  
 1：开启  
 0：一旦出现内存错误，就panic

### vm.memory\_failure\_recovery = 1

给有cap\_sys\_admin权限的用户保留的内存数量(默认值是 min(free\_page \* 0.03, 8MB))

### vm.admin\_reserve\_kbytes = 8192