**Linux调参**

1. **进程调度**

**调度器：CFS完全公平调度！**

**0（1）调度器！**

**优先级 nice值 0-139 0-99为静态优先级（值越大优先级高）。 100-139动态优先级（值小优先级高）。**

**物理地址分为：**

用户空间

内核空间

虚拟地址与物理地址的映射关系：

虚拟地址

4k页表

页框

物理地址

查看系统调用工具：strace

1. 进程地址空间

栈内存

数据段

堆内存

代码指令段

1. MMU内存管理单元：

线性地址空间分段（类似于1级目录2级目录的结构用于查找加速）

12位

10位

10位

cpu分析工具：htop dstat glances sar sysdig

**cpu内核调整：**

cpu隔离：vim /etc/grub

Kernel 追加 isolcpus=2,3

对3,4颗cpu进行调度隔离。

**cpu中断定义：**

Echo 0,1 >/proc/irq/中断号/sm\_qffinity

此时此中断只在第1和的2颗cpu进行中断。

**进程调度类别：**

实时优先级：1-99

调度器有：

SCHED\_FIFO,

SCHED\_RR 轮训

修改实时优先级：

Chrt -f [1-99] /path/to/program 用于SCHED\_FIFO调度器。

Chrt -r [1-99] /path/to/progran 用于SCHED\_RR调度器。

动态优先级： 100-139

调度器：

SCHED\_NORMAL

SCHED\_OTHER

修改为：nice

**内存调节参数：**

**一个页面4096字节，1M为256个页面，1G为256000个页面。。。。。**

**Overcommit\_memory=#**

**0:自动默认**

**1：不允许过量使用，此时不使用swap！会出现内存耗尽！**

**2：内存拒绝等于或大于总可以swap！**

**此时可用内存为：swap+ram\*overcommit\_ratio**

**Overcommit\_ratio:定义内存可使用比**

**nr\_hugepages:设置可用大内存页的个数：大内存页为2M或1G**

**查看使用的大内存页：cat /proc/memninfo**

**容量相关的内核可调参数：**

**/proc/sys/kernel/**

**msgmax:单个消息的最大字节数。默认为65535**

**msgmnb:单个消息队列的大小。消息需要放在清息队列中。**

**msgmni:消息队列的最大消息数量：64位为1985 32位为1736**

**##########################################################**

**Shmall:定义单共享内存的大小。**

**Shmmax:定义共享内存总大小。**

**Shmmni:定义共享内存总数量： shmall\*shmmni=shmmax**

**与容量相关的文件系统可调参数：/proc/sys/fs/**

**Aio-max-nr:定义异步IO上下文中可允许的最多事件数。**

**File-max:内核最大句柄数**

**Out-of-memory kill:内存用尽kill掉的进程优先级：**

**/proc/pid/**

**Oom\_adj:内存用尽时优先kill掉的优先级.**

**-16到15之间**

**-17是禁止被oom\_killer杀死的进程。**

**虚拟内存调参**

**Swappiness:定义swap交换区的倾性向。**

**0-100,越高越倾向swap的使用。**

**默认为60**

**Min\_free\_kbytes:最小内存空间值。（kb）低过此值则把空闲页面写入sawp交换区。**

**内存脏页（在内存还没写入磁盘的页面）/prco/sys/vm/dirty\_ratio**

**dirty\_ratio:单进程内存脏页的总数上限。默认为20%。过此值则写入盘。**

**drop\_caches: #**

**1:释放页缓存。**

**2：释放slab缓存：**

**3：释放以上两种缓存。**

**此项不会提升性能：因为把缓存都释放了。**

**IO系统：磁盘框架：**

**VFS/file system layer 文件系统层**

**Page cache**

**Page cache**

**Page cache**

**Page cache**

**页面缓存（脏页）**

**bio**

**Block buffer**

**pdflush**

**I/O scheduler**

**IO调度器（写入读出优化）**

**IO Request queue**

**IO调度算法：**

**CFQ：完全公平算法**

**Dadline:最后期调度算法**

**Anticipatory**

**Noop:无调度**

IO压力测试工具：

Aio-stoess

Fio

Iozone

调度算法更好：

固态硬盘不需要调度算法：

Echo Noop >/sys/block/盘/queue/scheduler

Ext2

Boot sector

Block GRoup0

Block GRoup1

Block GRoup1

Block GRoupN

Super block

Block group descriptors

Data block bitmaps

I-node bitmaps

I-node table

Data blocks

File frag -v /path/to/file 查看文件是否碎片化

Dumpe2ff /dev/sda1 查看是否离散存放

Atime:文件读时间缀。

挂载时：nodiratime关闭读取时间缀！

网络缓存图

Receive buffer

Send

buffer

Tcp\_mem

socket

socket

socket

r

s

r

s

IPX

Appletalk

Tcp\_rmem

Tcp\_wmem

Send

buffer

Receive buffer

Dom2

网络调参数：

计算网络缓存最多值：公式

1000\*1024\*1024/8\*2.5\*10^-3

1000为千兆网络

处于8表示bit赚Bit

2.5为ping的延时

10的负3次方为转换为妙。

设定缓存：

此设定为所有缓存（tcp/udp/.....）

/etc/sysctl.conf

net.core.rmem\_default 接收缓存默认值

net.core.rmem\_max 接收缓存最大值

net.core.wmem\_default 发送缓存默认值

Net.core.wmem\_max 发送缓存最大值

Tcp缓存设定

net.ipv4.tcp\_mem min defalut max :tcp的总缓存 最小缓存，默认缓存，最大缓存大小。

net.ipv4.tcp\_rmem min defalut max :tcp的接收缓存设置 接收+发送=总的缓存

net.ipv4.tcp\_wmem min defalut max :tcp的发送缓存设置

分片重组缓存设置：

net.ipv4.ipfrag\_high\_thresh :定义tcp分片重组的最大缓存值

net.ipv4.ipfrag\_low\_thresh:定义分片重组的最小缓存值

net.ipv4.ipfrag\_time:分片等待超时时长！

net.ipv4.tcp\_tw\_reuse:是否允许time-watit状态的sockets连接重要，开启提高响应能力。

tcp\_syncookies: sockies backlog队列溢出时是否发送syncookies。并发大量请求应启用, 可抵押小型syn攻击。

tcp\_max\_orphans:没有关联至文件描述符的套接字，最大数。

tcp\_fin\_timout:服务端定义fin\_wait-2的时长

tcp\_max\_tw\_buckets:内核支持有time wait状态的最大数量。应增加。

net.core.somaxconn:listen()状态的队列大小。不用调

httpd

ssh

Mozilla

Library

(libc)

库

用户空间

Kernel

Subsystem

内核系统总线

Slab

Allocator

页面再分割

Zoned

Buddy

Allocator

伙伴内存页面整合工具

MMU

物理内存

Kswapd

Swap交换服务

Pdflush

脏页回写磁盘

Disk

Driver

磁盘

Cat /proc/buddyinfo

进程虚拟地址-内存工作流程

**虚拟地址到线性地址的转换**

进程的虚拟地址

TLB为地址转换寄存器（缓存器）

处理异常

进程进入阻塞状态

处理异常

写时复制

被保护

yes

NO

是否有访问权限

不合法

复制页

进程进入

就绪状态

更新页表

IO完成产生中断

通知cpu

从磁盘读取rep

合法

No

把内存中的页面交换至swap中

yes

内存是否已满？

进程进入阻塞状态

从磁盘加载页面

至内存

缺页异常

引入的页面是否合法

在

不在

页面是否在内存中

每个进程都有自己的页表

未命中

线性地址空间

命中

页表

Updata TLB

TLB