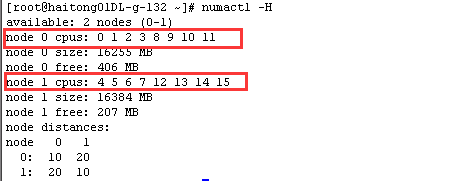
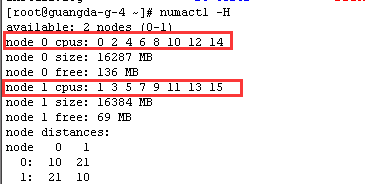
1. **关闭系统超线程。**
2. **分离一颗物理cpu**
3. 查看哪些processor属于同一颗物理cpu： **numactl –H**

**HP服务器：**



**Dell服务器：**



1. 分离cpu

分离规则：4核cpu分离3个核，8核cpu分离6个核

1) find / -name grub.cfg

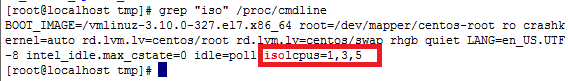
2) 进到指定目录vi grub.cfg

3)第一个rd.lvm.lv=centos/swap rhgb quiet之后增加： **intel\_idle.max\_cstate=0 idle=poll isolcpus=1,3,5……**

3、重启操作系统

**三、验证CPU分离是否正确**

1、grep "iso" /proc/cmdline ，查看分离cpu是否正确



如果输出信息为空，则不正确

**四、通过init函数的后两个参数，绑定input 和output进程(分离出的核中选两个)**

**五、绑定策略主进程（cpu 7）**

方法一：通过命令行绑定（solarflare网卡）

onload --profile=latency numactl --physcpubind=7 ./x1demo

方法二：在启动策略的时候，获取主进程号，然后进行绑定，参考代码如下：

#include <sys/syscall.h>

#define gettidv1() syscall(\_\_NR\_gettid)

#define gettidv2() syscall(SYS\_gettid)

bool BindThreadCpu(unsigned long tid, int cpuNo)

{

#ifdef WIN32

return true;

#else

int core\_num = sysconf(\_SC\_NPROCESSORS\_CONF);

COut::out("system has %d processor(s)\n", core\_num);

if(cpuNo < 0 || cpuNo > core\_num)

{

COut::error("warning: could not set CPU affinity:[out of core range]tid=%lu,cpuNo=%d\n",tid,cpuNo);

return false;

}

cpu\_set\_t mask;

CPU\_ZERO(&mask);

CPU\_SET(cpuNo, &mask);

if (pthread\_setaffinity\_np(tid, sizeof(mask), &mask) < 0) {

perror("pthread\_setaffinity\_np");

COut::error("pthread\_setaffinity\_np");

}

cpu\_set\_t get;

CPU\_ZERO(&get);

if (pthread\_getaffinity\_np(tid, sizeof(cpu\_set\_t), &get) < 0) {

perror("pthread\_getaffinity\_np");

COut::error("pthread\_getaffinity\_np");

}

for (int j = 0; j < core\_num; j++) {

if (CPU\_ISSET(j, &get)) {

COut::info("thread %lu is running in processor %d\n", pthread\_self(), j);

}

}

return true;

#endif

}

**六、观察3个进程是否绑定成功**

1. top –p x1主进程号
2. shift+H 显示所有子进程
3. 单击【f】,调出显示菜单，在【p】的位置单击【空格】选中，并esc退出
4. 此时除主进程外的两个进程应该cpu占用为100%,且CPU逻辑号不变
5. 主进程占用不是100%,但是报单过程中CPU逻辑号不变