浪潮HPC集群作业调度系统使用培训

inspur 浪潮

集群作业调度系统说明

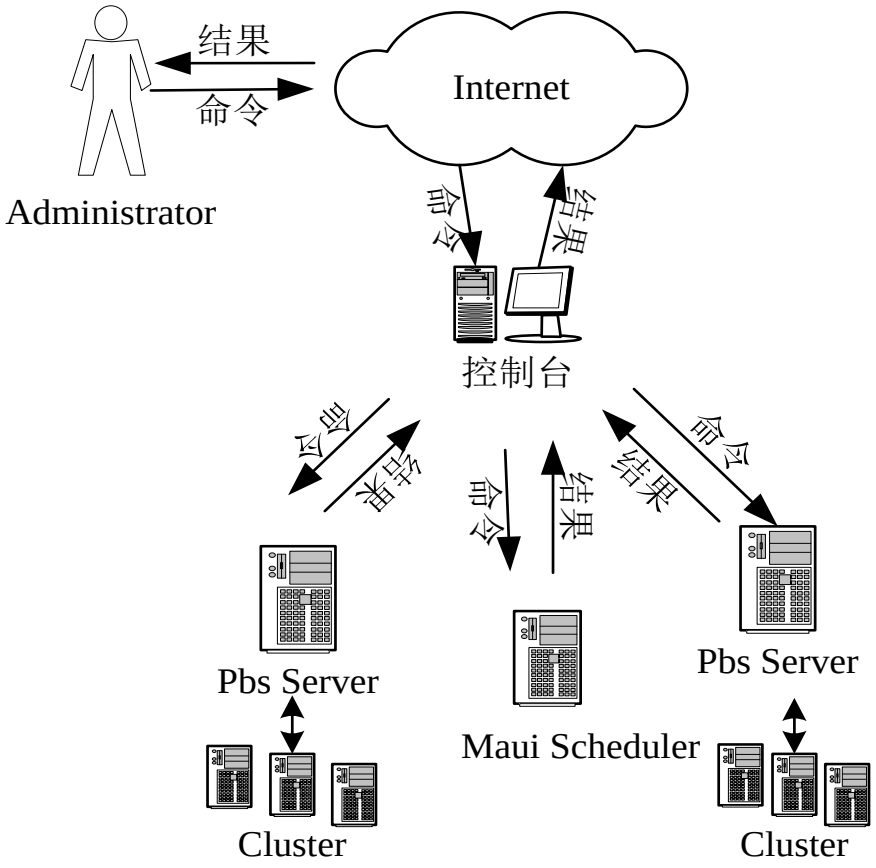
建立一种作业提交的秩序

inspur 浪潮

集群作业调度系统说明

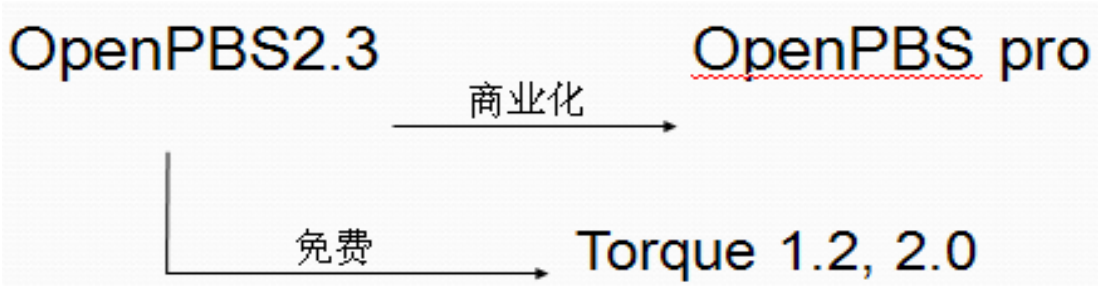
- 软件介绍

浪潮 TSJM 作业调度软件是专为浪潮天梭系列 HPC 产品定制的一款作业调度软件，该软件通过浏览器（IE，firefox等）进行操作，可以管理集群系统中的软硬件资源和用户提交的作业，根据集群中的资源使用情况来合理的调度用户提交的作业，从而达到提高资源的利用率和作业的执行效率的作用。TSJM底层是用openpbs和maui作业调度管理软件。



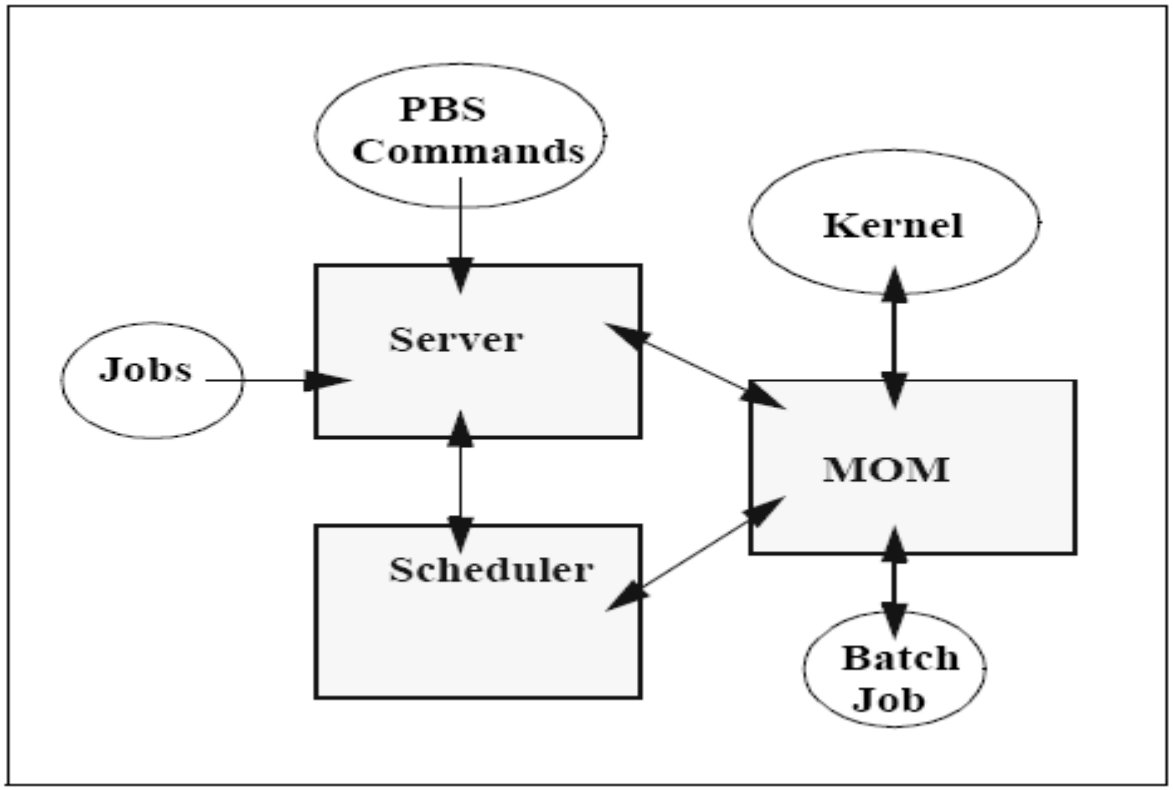
集群作业调度系统说明

- OpenPBS 介绍
  - PBS: Portable Batch System
  - 做为集群作业调度系统。作业管理又称为工作负载管理，负载共享或负载管理。它有效地管理系统中的各种资源，以及用户提交的作业。目的是为了充分利用集群的软硬件资源及宝贵的CPU时间，有效地管理集群，合理地调度作业，使系统具有高的吞吐率和利用率。
  - 目前天梭10000中使用的作业调度软件为:torque 2.3.0
- PBS历史:

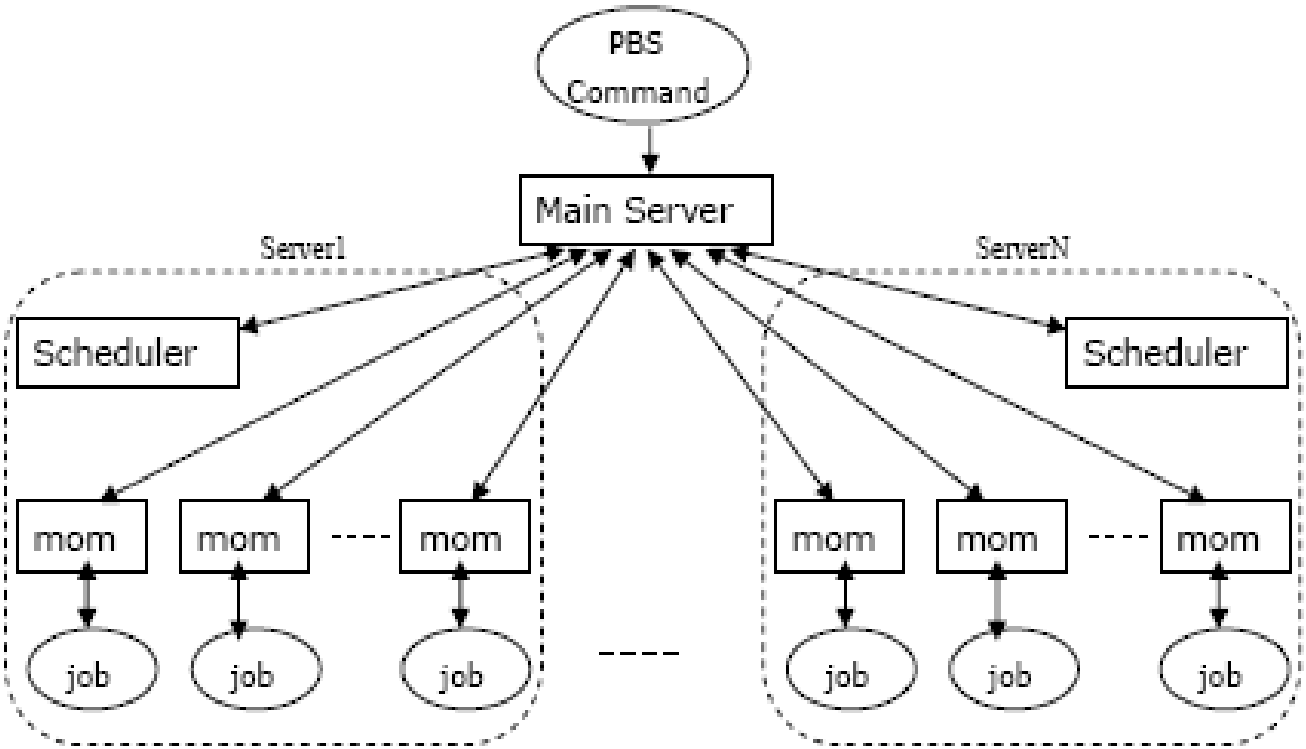


集群作业调度系统说明

- PBS基本组件



集群作业调度系统说明



集群作业调度系统说明

- PBS基本组件
  - Pbs command:用于提交、监视、修改和删除作业。
  - Pbs_server: 提供基本的批处理服务，例如接收/创建一个批处理作业，管理维护作业队列，管理输出结果等。
  - Pbs_mom:是一个守护进程，从pbs server处接收作业后放入其执行队列中等待执行。
  - Scheduler: 对用户提交的作业进行调度
    - 当前集群上用的调度器是maui

集群作业调度系统说明

- Maui是Cluestering公司为了弥补torque自带的调度器pbs_shced的调度策略而开发了一款调度器软件 。
- Maui优先级系统

```
[root@ln1 ~]# qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
1292200.cm1                CNPA_wt          ljz01            2658:44: R Infini-l
1292203.cm1                CNPA_mut         ljz01            2622:39: R Infini-l
1292225.cm1                KIF_long         ljz01            2164:53: R Infini-l
1292255.cm1                STDIN            yangfang               0 R General
1292511.cm1                KIF_inter        ljz01            1650:03: R Infini-l
1293205.cm1                KIF_short        ljz01            1451:17: R Infini-l
1295292.cm1                c12d_run_001     zhuhongtao       354:19:2 R Para-s
```

集群作业调度系统使用方法

- Torque应该如何使用?
  - 熟悉Torque提供的几个命令
  - 编写作业提交脚本
  - 了解使用注意事项
- PBS命令
  - qsub 作业提交脚本
  - qstat [参数]
  - qdel 作业号

集群作业调度系统使用方法

- PBS命令详解
  - 提交作业的命令

    qsub 作业提交脚本

    此命令执行后,会给出个作业号
  - 查询作业命令

    qstat［参数］

qstat 命令详解

命令格式：qstat [-f][-a][-i] [-n][-s] [-R] [-Q][-q][-B][-u]

参数说明：

-f jobid 列出指定作业的信息

-a 列出系统所有作业

-i 列出不在运行的作业

-n 列出分配给此作业的结点

-s 列出队列管理员与scheduler所提供的建议

-R 列出磁盘预留信息

-Q 操作符是destination id，指明请求的是队列状态

-q 列出队列状态，并以alternative形式显示

-au userid 列出指定用户的所有作业

-B 列出PBS Server信息

-r 列出所有正在运行的作业

-Qf queue 列出指定队列的信息

-u 若操作符为作业号，则列出其状态。

若操作符为destination id，则列出运行在其上的属于user_list中用户的作业状态。

集群作业调度系统使用方法

- pbsnodes查看节点状态

- pbsnodes -l all
- cu01 free （代表空闲状态，可接受作业）
- cu02 job-exclusive （代表正在运行作业，不可接受作业）
- cu03 offline （代表掉线状态，不可接受作业）
- cu01 down （代表关机或者故障，作业不可接受作业）
- cu02 down,job-exclusive （代表关机或者故障，且关闭前有作业在进行）

集群作业调度系统使用方法

- PBS命令详解
  - 作业删除命令

    qdel 作业号

    其中作业号为qsub提交后系统所给出的一个号码
- 注意事项

  1、非管理员只能删除自己提交的作业

  2、在提交作业时估计自己需要运行的时间把其写进作业提交脚本里。

  3、Maui里的策略一但制定了，对于作业的优先级，普通用户是不可见且不可调的。

PBS脚本写作

脚本包含三部分：
资源声明：即规定所需要的节点数，核数，作业名，所要递交的队列
环境变量：即运行作业时，需要的各个节点的基本属性，比如某些软件的路径等
可执行程序：即需要通过MPI来运行的并行程序
如下例子说明
脚本声明部分：

```
#PBS -N vasp                \\设定应用程序名字
#PBS -l nodes=2:ppn=12       \\启动2个节点每个节点12个核心
#PBS -l walltime=999:00:00   \\申请999小时的工作，不满足将无法继续进行计算
#PBS -q batch                 \\指明作业队列
#PBS -V
#PBS -S /bin/bash            \\让pbs脚本识别bash命令
```

环境变量部分：

```
### intel                     \\intel包环境变量生效
source /opt/intel/composer_xe_2015/bin/compilervars.sh intel64
source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh
source /opt/intel/impi/5.0.2.044/bin64/mpivars.sh
```

可执行程序部分：

```
cd $PBS_O_WORKDIR
nprocs=`wc -l < $PBS_NODEFILE`
exec=/opt/soft/vasp/vasp
mpirun -genv I_MPI_DEVICE rdma -machinefile $PBS_NODEFILE -np $nprocs  $exec\\执行并行程序
date
```

资源声明部分写作

脚本声明部分：

```
#!/bin/bash
#PBS -N vasp                    \\设定应用程序名字
#PBS -l nodes=2:ppn=12          \\启动2个节点每个节点12个核心
#PBS -l walltime=999:00:00      \\申请999小时的工作，不满足将无法继续进行计算
#PBS -q batch                   \\指定作业队列（即节点属性）
#PBS -a date_time               \\格式为[[[[CC]YY]MM]DD]hhmm[.SS]表示经过date_time时间后作业才可以运行
#PBS -e path                    \\将标准错误信息重定向到path
#PBS -o path                    \\将标准输出信息重定向到path
#PBS -l resource_list           \\定义资源列表。以下为几个常用的资源种类
```

cput=N 请求N秒的CPU时间; N也可以是hh:mm:ss的形式。 -l cput=1:00:00

mem=N[K|M|G][B|W] 请求N {kilo|mega|giga}{bytes|words} 大小的内存。 -l mem=100mb

nodes=N:ppn=M 请求N个结点，每个结点M个处理器。 -l nodes=2:ppn=10

walltime表示任务最大时限。 -l walltime=23:00:00

nodes=X:host 分配X个主机名称中含有host的执行节点 -l nodes=12:cu01+12:cu12

ncpus=5 请求的cpu数 -l ncpus=5

pcput 任务的任何一个进程拥有的最大cpu执行时间 -l pcput=1:00:00

pmem 任务的任何一个进程能够分配到的最大物理内存数 -l pmem=45mb

pvmem 任务的任何一个进程能够使用的虚拟内存的最大数 -l pvmem=100mb

vmem 任务的所有并发进程能够使用的最大虚存数 -l vmem=100mb

qsub -l select=2:ncpus=3:mem=4gb:arch=linux ，select=2表示需要2个这样的资源块，一个资源块包括3个cpu，4gb的内存，系统结构要求是linux，即总共需要6个cpu，8gb的内存。再如：

-l select=2:ncpus=1:mem=10GB+3:ncpus=2:mem=8GB:arch=solaris注意中间的+号，是两个资源块的分隔符

请求全任务(job-wide)资源格式为-l keyword=value[,keyword=value...]，如：qsub -l ncpus=4,mem=123mb,arch=linux

#PBS -p priority ： 任务优先级，整数[-1024,1024]若无定义则为0

PBS脚本实例（lammps应用为例）

```
#PBS -N lammps                  \\设定应用程序名字
#PBS -l nodes=2:ppn=12          \\启动2个节点每个节点12个核心
#PBS -l walltime=999:00:00      \\申请999小时的工作，不满足将无法继续进行计算
#PBS -q batch
#PBS -V
#PBS -S /bin/bash               \\让pbs脚本识别bash命令，#!/bin/bash

### intel                       \\intel包环境变量生效
source /opt/intel/composer_xe_2015/bin/compilervars.sh intel64
source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh
source /opt/intel/impi/5.0.2.044/bin64/mpivars.sh

cd $PBS_O_WORKDIR
EXEC=/opt/soft/lammps/lmp_mkl   \\指定lammps程序绝对路径
NP=`cat $PBS_NODEFILE | wc -l`
NN=`cat $PBS_NODEFILE | sort | uniq | tee /tmp/nodes.$$ | wc -l`
cat $PBS_NODEFILE > /tmp/nodefile.$$
export MPD_CON_EXT=${PBS_JOBID}
export I_MPI_JOB_CONTEXT=$PBS_JOBID
mpdboot -f /tmp/nodefile.$$  -n $NN        \\启动集群
mpiexec -genv I_MPI_DEVICE rdma -machinefile /tmp/nodefile.$$ -n $NP  $EXEC
< in\ relax.relax       \\执行并行程序
mpdallexit
rm -f /tmp/nodefile.$$
```

PBS脚本实例（程序lammps应用为例）

```
[inspur@cu01 lammps_test]$ cat lammps.pbs
#PBS -N lammps
#PBS -l nodes=1:ppn=12
#PBS -l walltime=12:00:00
#PBS -q batch
#PBS -V
#PBS -S /bin/bash

### intel
source /opt/intel/composer_xe_2011_sp1/bin/compilervars.sh intel64
source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh
source /opt/intel/impi/4.0.3/bin64/mpivars.sh

cd $PBS_O_WORKDIR
EXEC=/opt/inspur-soft/lammps-12Aug10/src/lmp_mkl
NP=`cat $PBS_NODEFILE | wc -l`
NN=`cat $PBS_NODEFILE | sort | uniq | tee /tmp/nodes.$$ | wc -l`
cat $PBS_NODEFILE > /tmp/nodefile.$$
export MPD_CON_EXT=${PBS_JOBID}
export I_MPI_JOB_CONTEXT=$PBS_JOBID
mpdboot -f /tmp/nodefile.$$  -n $NN
mpiexec -genv I_MPI_DEVICE rdma -machinefile /tmp/nodefile.$$ -n $NP  $EXEC < in\ relax.relax
mpdallexit
rm -f /tmp/nodefile.$$
```

PBS脚本实例（程序ansys多节点并行脚本例）

```
[inspur@mu01 test]$ cat ansys.pbs
#!/bin/bash
#PBS -N ansys
#PBS -l nodes=3:ppn=12
#PBS -q batch
#PBS -j oe

cd $PBS_O_WORKDIR
source /opt/intel/composer_xe_2011_sp1/bin/compilervars.sh intel64
source /opt/intel/mkl/bin/intel64/mklvars_intel64.sh
source /opt/intel/impi/4.0.3/bin64/mpivars.sh
export MPIRUN_OPTIONS="-prot"
export MPI_REMSH=/usr/bin/ssh
export MPI_IC_ORDER=IBV:TCP
machines=`uniq -c $PBS_NODEFILE | awk '{printf ":"$2":"$1}'`
echo $machines > inspur-duyk-ansys.host
sed 's/cu/ibcu/g' inspur-duyk-ansys.host >inspur-duyk-ansys.host1
sed 's/^.//g' inspur-duyk-ansys.host1 > inspur-duyk-ansys.host2
/opt/ansys_inc/v140/ansys/bin/ansys140 -b -dis -machines `cat inspur-duyk-ansys.host2` -i d92.inp -o d92.out
rm -rf inspur-duyk-ansys.host*
```

提交作业、查询作业

```
[root@ln1 ~]# qsub sleep.pbs
1295316.cm1
[root@ln1 ~]# qstat -an 1295316

cm1:
                                                                  Req'd  Req'd   Elap
Job ID               Username Queue    Jobname          SessID NDS   TSK Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- --- ------ ----- - -----
1295316.cm1          root     Infini-l sleep.test         --      3  36    --  12:00 Q   --
   --
```

删除作业

```
[root@ln1 ~]# qdel 1295316
[root@ln1 ~]# qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
1292200.cm1                CNPA_wt          ljz01           2662:28: R Infini-l
1292203.cm1                CNPA_mut         ljz01           2626:24: R Infini-l
1292225.cm1                KIF_long         ljz01           2168:36: R Infini-l
1292255.cm1                STDIN            yangfang              0 R General
1292511.cm1                KIF_inter        ljz01           1653:47: R Infini-l
1293205.cm1                KIF_short        ljz01           1455:01: R Infini-l
1295292.cm1                cl2d_run_001     zhuhongtao      354:19:2 R Para-s
1295314.cm1                initmodel1       sfeng           00:00:16 R fat
1295316.cm1                sleep.test       root            00:00:00 C Infini-l
```

作业状态解析

- 队列中的S代表含义
- R代表运行
- Q代表排队
- C代表运算完毕，或者在退出
- E代表运算有问题

```
[inspur@su01 vasptest]$ qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
143.su01                   VASP             inspur          00:00:00 R anda
144.su01                   VASP             inspur          00:00:00 C anda
145.su01                   VASP             inspur                 0 Q anda
```

作业状态解析

- 纠错举例：

```
[root@ln1 ~]# qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
1292200.cm1                CNPA_wt          ljz01           2667:43: R Infini-1
1292203.cm1                CNPA_mut         ljz01           2631:38: R Infini-1
1292225.cm1                KIF_long         ljz01           2173:42: R Infini-1
1292255.cm1                STDIN            yangfang               0 R General
1292511.cm1                KIF_inter        ljz01           1659:01: R Infini-1
1293205.cm1                KIF_short        ljz01           1460:06: R Infini-1
1295292.cm1                c12d_run_001     zhuhongtao      354:19:2 R Para-s
[root@ln1 ~]# qstat -an 1292255

cm1:
                                                                  Req'd  Req'd   Elap
Job ID               Username Queue    Jobname          SessID NDS   TSK Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- --- ------ ----- - -----
1292255.cm1          yangfang General  STDIN             25146     1  12    --  240:0 R   --
   c02b20/11+c02b20/10+c02b20/9+c02b20/8+c02b20/7+c02b20/6+c02b20/5+c02b20/4
   +c02b20/3+c02b20/2+c02b20/1+c02b20/0
[root@ln1 ~]# pbsnodes -l c02b20
c02b20               down,job-exclusive
[root@ln1 ~]# ping c02b20
PING c02b20 (10.0.2.20) 56(84) bytes of data.
From ln1 (10.0.0.2) icmp_seq=2 Destination Host Unreachable
From ln1 (10.0.0.2) icmp_seq=3 Destination Host Unreachable
From ln1 (10.0.0.2) icmp_seq=4 Destination Host Unreachable

--- c02b20 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3999ms
, pipe 3
```

谢谢大家！

inspur 浪潮