基因组所高性能计算指南

高性能集群 192.168.118.11 包括 61 个计算节点。每个计算节点 32 个计算核心，128GB 内存共划分为 4 个队列。用户登录 IP 是 192.168.118.11

高性能集群 192.168.118.1 包括 266 个计算节点。其中计算节点 12 个计算核心，48GB 内存共 180 个，8 核 16GB 内存节点共 80 个，32 核 1T 内存节点 1 台。32 核 512GB 内存节点共 5 台。用户登录 IP 是 192.168.118.1

118.11 集群队列名：

lowq 包含 10 个计算节点。每个节点上单个任务最大可用内存 30GB。每个任务占用核数不超过一个。

middleq 包括 40 个计算节点。每个节点上单个任务最大可以占用内存 60GB.每个任务占用核数小于等于 15 个。

largeq 包括 10 个计算节点.每个节点上单个任务最大可以占用 120GB 内存。每个任务占用核数小于等于 30 个。

bigmem 包括 1 个大内存服务器。每个计算节点上单个任务占用最大内存 500GB。每个任务占用核数小于等于 46 个。

118.1 集群队列名：

bioque 包含 176 个计算节点。每个节点上单个任务最大可用内存 20GB。每个任务占用最大核数不超过 5 个。

genomics 包括 80 个计算节点。每个节点上单个任务最大可以占用内存 12GB.每个任务占用核数小于等于 6 个。

Fat02que 包括 1 个大内存节点.每个节点上单个任务最大可以占用 1000GB 内存。每个任务占用核数小于等于 30 个。

asmque 包括 5 个大内存服务器。每个计算节点上单个任务占用最大内存 500GB。每个任务占用核数小于等于 30 个。

注意请大家认真申请任务占用资源量。尽量不要浪费集群资源。

作业提交：

该集群上的任何用户都有权使用上面的队列。

dsub 不支持多节点并行只支持单节点内并行计算但并不代表本系统不支持多节点并行只是为了防止资源浪费做出的提交策略。

```
[root@manager bin]# ./dsub
dsub script.sh
Format: cat script.sh
 #PBS -q batch
 #PBS -l mem=1gb|mb|kb,walltime=01:00:00
 #HSCHED -s hschedd
```

用户在提交作业的时候兼容 pbs qsub 的参数。但不支持命令行参数。所有的参数只需要编写到自己的脚本里面。其中上面的参数用户是必须指定的否则作业无法提交。#PBS -l mem=1gb|mb|kb,walltime=01:00:00 这个参数内存设置可以是 gb,mb 或 kb。核数如果不指定默认是 1. 所有需要的参数可以在这一行用逗号分隔。

例如我需要投递一个作业占 30GB 内存 需要 2 个核参与计算。计算时间是 5 小时。跑到 middleq 队列上。

vi script.sh

#PBS -q batch

#PBS -l mem=30gb,walltime=05:00:00,nodes=1:ppn=2

#HSCHED -s hschedd

…….

mpidsub 主要用于提交多核多节点并行的作业.如果用户的程序本身支持多核多节点并行可以用这种方式提交。如果用 mpidsub 指定的节点数必须大于 1.

vi script.sh

#!/bin/sh

#PBS -q quename

#PBS –l walltime=5:00:00,cput=40:00:00,nodes=4:ppn=2,mem=1gb|mb|kb //nodes>1

cd $PBS_O_WORKDIR

/opt/MeSC/bin/mpiexec -comm mpich-gm mpi_program arg1 arg2

提交方式

dsub script.sh 或 mpidsub script.sh

交互式提交作业 qlogin

```
[root@manager bin]# ./qlogin
Format: qlogin -I -q batch -l mem=1gb|mb|kb,walltime=01:00:00
[root@manager bin]#
```

csub 使用说明(注意该方式只适合单节点并行或串行)

csub -q <queue> -N <jobname> -p <np> -m <mem> -w <Walltime> -c <Cmd>

Options:

<queue> Select jobs queue name. //选择要运行作业的队列

<jobname> Job name . //作业名

<np> Computing cores .//提交作业占用核数

<mem> Application of memory. //提交作业申请内存数

<Walltime> Task start and end time. //任务结束时间

<Cmd> command executed by hschedd ,use quotation mark. //要跑的命令引号括起来

Format: csub -q batch -N firejob -p 2 -m 2gb -w 01:00:00 -c "exec cmd "

例如要跑某个软件可以如下参考

要运行命令为 mpirun -np 8 /software/biosoft/software/mrbayes_3.2.2/src/mb CDS.fasta.nex

提交方式可以如下

csub –q batch -N firejob -p 8 -m 20gb -w 01:00:00 -c "mpirun -np 8 /software/biosoft/software/mrbayes_3.2.2/src/mb CDS.fasta.nex "

任务的查看

mstat 参数与 qstat 相通

Hcluster_usage 对高性能集群监控

```
[root@manager bin]# ./Hcluster_usage
                                                    Page 1
                                  CLUSTER FREE
 QUEUE          NODE      MEMFREE     PPNFREE
 ------------------------------------------------------
|middleq     |node49     |120        |30
|largeq      |node60     |120        |30
|middleq     |node47     |120        |30
|largeq      |node56     |120        |30
|lowq        |node27     |120        |30
|lowq        |node28     |120        |30
|largeq      |node59     |120        |30
|lowq        |node30     |120        |30
|largeq      |node58     |120        |30
|middleq     |node46     |120        |30
|middleq     |node50     |120        |30
|middleq     |node48     |120        |30
|lowq        |node26     |120        |30
|bigmem      |node61     |120        |46
|largeq      |node57     |120        |30
|lowq        |node29     |120        |30
============================================================

-------------------CLUSTER TOTAL PPN MEM USAGE------------
ppn_usage=0.00---mem_usage=0.00
```

上面显示的内容分两部分：第一部分是每个节点所在队列及每个节点可用的内存和核数。对应普通用户执行该命令可以发现自己要跑的任务是不是有计算节点满足条件。第二部分是整个高性能集群计算核心的使用比例和内存的使用比例。可以很好的评估当前我们计算集群的使用率。

用户查看自己的作业状态（mustat 和 xtop）

Mustat 说明

```
[root@manager bin]# ./mustat
User jobs monitor!
 Format: hmon -u username or hmon -j jobid
[root@manager bin]# ./mustat  -u zhonghua

          ==================@@@@@@@@@@@@@@@@@|||||||||Mon May 12 10:37:54 HKT 2014|||||||||@@@@@@@@@@@@@@@@@==================

JOBID----JobName----used.cput----used.mem----used.vmem----used.walltime----state----queue----List.mem----List.nodes----List.walltime
6568.manager   work.sh   zhonghua@login1   619:33:00   70641796kb   7098071 6kb   620:01:24   R   largeq   40gb   node60:ppn=20   1000:00:00
8114.manager   scriptAll-UN-4.sh   zhonghua@login1   332:42:59   2789248kb   6263620kb   97:18:04   R   largeq   10gb   node53:ppn=20   10000:00:00
8481.manager   scriptAll-UN-5.sh   zhonghua@login2   63:53:42   2725356kb   6258292kb   19:43:43   R   largeq   10gb   node55:ppn=6   10000:00:00
```

用户可以方便查看自己作业的所占用的系统资源及资源申请信息。如果退出按<ctrl-c>每隔 2 分钟自动刷新输出当前任务监控状态。

名词解释：

JOBID----作业 id 号

JobName----作业名

Job_Owner----作业提交用户

used.cput----到当前时间消耗的 cpu 计时

used.mem----当前时间消耗内存

used.vmem----当前时间消耗虚拟内存

used.walltime----到当前时间任务运行时间

state----运行作业 R 状态

queue----队列名

List.mem----作业申请内存

List.nodes----作业申请节点

List.walltime---作业申请最大运行时间

Xtop 说明

实际上 xtop 是一种用户远程监控用户任务的工具。具体使用如图 如果要退出可以按<ctrl-c>

```
[root@login1 ~]# xtop
<---- CTRL-C ---exit----->
Format: xtop nodex
[root@login1 ~]# xtop node3
<---- CTRL-C ---exit----->
Warning: Permanently added 'node3,11.11.10.3' (RSA) to the list of known hosts.
1
top - 10:13:53 up 96 days, 23:29,  1 user,  load average: 3.78, 4.97, 5.19
Tasks: 1165 total,   7 running, 1158 sleeping,   0 stopped,   0 zombie
Cpu(s): 17.4%us,  0.4%sy,  0.0%ni, 82.1%id,  0.1%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  132136220k total, 96874556k used, 35261664k free,   243584k buffers
Swap: 33554424k total,        0k used, 33554424k free, 93734380k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
16693 wangxuan  20   0  4300  676  308 R 100.0  0.0  12:27.36 gzip
15987 wangxuan  20   0  4300  676  308 R 99.9  0.0  39:03.05 gzip
16606 wangxuan  20   0  4300  680  308 R 99.9  0.0  13:47.00 gzip
15986 wangxuan  20   0  159m 6848 2324 S 80.8  0.0  32:03.79 /software/biosoft/s
16605 wangxuan  20   0  159m 6888 2324 R 80.1  0.0  11:35.43 /software/biosoft/s
16692 wangxuan  20   0  159m 6884 2324 R 80.1  0.0  10:26.59 /software/biosoft/s
15988 wangxuan  20   0  4432  580  404 S  9.9  0.0   4:05.45 gzip -cd /share_bio
16694 wangxuan  20   0  4432  584  404 S  8.9  0.0   1:09.83 gzip -cd /share_bio
16607 wangxuan  20   0  4432  576  404 R  7.9  0.0   1:18.56 gzip -cd /share_bio
16939 root      20   0 15828 2108  944 R  1.3  0.0   0:00.32 top -c
 3191 root      20   0 78792 6220 1292 S  1.0  0.0 696:53.91 /opt/lampp/bin/pyth
    4 root      20   0     0    0    0 S  0.7  0.0   8:31.24 [ksoftirqd/0]
  101 root      20   0     0    0    0 S  0.3  0.0   2:11.97 [ksoftirqd/24]
  156 root      20   0     0    0    0 S  0.3  0.0   4:17.05 [events/25]
 2202 root      20   0     0    0    0 S  0.3  0.0   8:16.03 [kondemand/4]
 2206 root      20   0     0    0    0 S  0.3  0.0 109:49.72 [kondemand/8]
^C9  2222 root      20   0     0    0    0 S  0.3  0.0  16:11.18 [kondemand/24]
```