天河 天河二号超级计算机系统
Introduction of the MilkyWay Ⅱ Supercomputer

吕梁高性能云计算中心

TH-2 用户手册(试用版 V1.0)

国防科学技术大学
吕梁市人民政府
军民融合协同创新研究院
Civil military integration Collaborative Innovation Institute

目录

1. TH-2 作业提交

在 TH-2 中，所有在计算结点中运行的串行或并行应用程序，都必须通过资源管理系统来提交运行。资源管理系统首先将用户提交的应用程序构造成作业进行排队处理，然后根据 TH-2 的实时运行资源状态，决定何时以及在哪些计算结点中加载应用程序的运行，不同的应用程序之间不存在资源的竞争冲突，用户也可以通过作业管理系统来监控应用程序的运行。

TH-2 用户手册(试用版 V 1.0)

1.1 状态查看命令

在用户提交作业前，应查看系统的使用情况，这样利于用户根据系统使用情况，对相应的计算结点进行选择。

1.1.1 结点状态查看 yhinfo 或 yhi

yhi 为 yhinfo 命令的简写，用户可以使用 yhi 或者 yhinfo 命令查看结点的使用情况，从而根据情况做出选择。

其中 PARTITION 表示分区,TIMELIMIT 表示该分区的时间限制,NODES 表示结点数,STATE 表示结点运行状态其中 down 表示未启动，idle 表示启动后出于空闲状态，allocated 表示结点已经分配了一个或多个作业，NODELIST 为结点列表。

1.1.2 作业状态信息查看 yhqueue 或 yhq

yhq 为 yhueue 命令的简写,用户可以使用 yhq 或 yhqueue 命令查看系统中各计算结点的运行情况。

其中 JOBID 表示任务 ID,Name 表示任务名称,USER 为用户,TIME 为已运行时间,NODES 表示占用结点数，NODELIST 为任务运行的结点列表。获取的 jobid，用户在作业取消命令 yhcancel 中会使用到。

用户可以使用 yhq 查看自己提交的作业，为了保证用户的数据安全，普通用户通过 yhq 只能看到自己提交的作业。

1.2 提交作业

目前 TH-2 部署的资源管理系统包括多种作业提交方式，交互作业提交方式 yhrun，批处理作业提交方式 yhbatch 和分配模式 yhalloc。作业终止方式为 yhcancel 命令，需要获取作业的 jobid，如前所述，jobid 可以通过 yhq 命令查看获得。

本手册，为了简化和方便用户，只对相关命令做简单介绍，用户如需更多参数选择，则可以通过相应命令后加入--help 的方式，获取帮助信息，从而满足用户需求。

1.2.1 交互式作业提交 yhrun

系统中作业的运行分成两步：资源分配与任务加载。对于批处理作业，使用 yhbatch 命令提交作业脚本，作业被调度运行后，在所分配的首个结点上执行作业脚本，在作业脚本中使用 yhrun 命令加载作业任务。对于交互式作业，资源分配与任务加载两步均通过 yhrun 命令进行：当在登录 shell 中执行 yhrun 命令时，yhrun 首先向系统提交作业请求并等待资源分配，然后在所分配的结点上加载作业任务。

yhrun 运行的主要格式如下：

yhrun [options] program

yhrun 包括多个选项，用户最常使用的选项如下：

-n, --ntasks=ntasks

指定要运行的进程数。请求 yhrun 分配/加载 ntasks 个进程。省缺的情况是每个 CPU 运行一个进程，但是 -c 参数将改变此省缺值。

-N, --nodes=minnodes[-maxnodes]

请求为此作业至少分配 minnodes 个结点。调度器可能决定在多于 minnodes 个结点上启动作业。可以通过指定 maxnodes 限制最多分配的结点数，如“--nodes=2-4”。最少和最多结点数可以相同以便指定确切的结点数，如“--nodes=2-2”将请求两个并且仅仅两个结点。如果没有指定-N，省缺的行为是分配足够的结点以满足-n 选项的要求。

-p, --partition=partition

从分区 partition 请求资源。如未指定，则省缺为默认分区。您所在的分区为 free，请在提交作业时务必选择 -p free

-t, --time=minutes

设置作业的运行时间限制为 minutes 分钟。省缺值为分区的时间限制值。当到达时间限制时，作业的进程将被发送 SIGTERM 以及 SIGKILL 信号终止执行。

-D, --chdir=path

加载的作业进程在执行前将工作目录改变到 path 。省缺情况下作业 yhrun 进程的当前工作目录。

-l, --label

在标准输出/标准错误的每行之前添加任务号。通常，远程任务的标准输出和标准错误通过行缓冲直接传递到 yhrun 的标准输出和标准错误。--label 选项将在每行输出前面添加远程任务的 ID。

-J, --job-name=jobname

指定作业的名字。省缺值是可执行程序的名字 program 。

-W, --wait=seconds

指定在第一个任务退出后，到终止所有剩余任务之前的等待时间。0 表示无限等待（60 秒后将发出一个警告）。省缺值可由系统配置文件中的参数设置。此选项用于确保作业在一个或多个任务提前退出时能够及时终止。

-w, --nodelist=nodelist|filename

请求指定列表中的结点。分配给作业的将至少包含这些结点。nodelist 可以是逗号分割的结点列表或范围表达式（如 cn[1-5,7,12]）。如果包含“/”字符，则 nodelist 将会被当作是一个文件名，其中包含了所请求的结点列表。

-x, --execlude=nodelist|filename

排除指定列表中的结点。分配给作业的将不会包含这些结点。

--checkpoint-path=path

指定任务检查点映像文件的保存目录。省缺为任务的当前工作目录。

--checkpoint-period=number[h|m]

指定对作业进行自动周期性检查点操作。如果 number 后没有跟时间单位，则默认为 h（小时）。

--restart-path=path

指定本次任务加载为从以前的检查点映像恢复执行。path 为检查点映像文件所在的路径。

--exclusive

此作业不能与其它运行的作业共享结点，加入此选项，则表示用户需要针对此作业使用独占的处理器，如果没有足够的处理器，则作业的启动将会被推迟。

以上选项中，由以 -N ，-n，-p，-w，-x 等选项最常用，-N 指定结点数，-n 指定进程数，-p 指定分区名，-w 指定结点列表，-x 指定不参加分配的结点列表（用于排除自己认为有问题的结点）。

TH-2 上的资源使用非抢占式调度方式，即作业如果没有占满结点，则如有别的作业提出需求，若剩余资源合适，也会将资源分配给新的作业。例如一个作业占用了一个结点的 4 核，另外有新的作业也需要 4 核，则该作业也会分配在该结点上。

示例：

1）在分区 corpor，结点 cn[256-268] 上运行 hostname；

```
$ yhrun -w cn[256-268] -p corpor hostname
yhrun: XXXXX: use ‘-t’ option to set time limit of job. defaults to 5 (minutes)
yhrun: job 9637 queued and waiting for resources
yhrun: job 9637 has been allocated resources
cn256
cn259
...
cn267
```

2）运行在 tryout 分区，运行 4 任务的 MPI 程序 cg.C.4，每个结点一个任务，分配的结点中至少包含结点 cn[4-5]；作业运行时间不超过 20 分钟；

```
$ yhrun -w cn[4-5] -n 4 -N 4 -t 20 ./cg.C.4
NAS Parallel Benchmarks 3.2 --CG Benchmark
Size: 150000
Iterations: 75
Number of active processes: 4
```

```
Number of nonzeroes per row: 15
Eigenvalue shift: .110E+03
iteration ||r|| zeta
1 0.15244429457374E-12 109.9994423237398
2 0.45529118072694E-15 27.3920437146522
3 0.45039339889198E-15 28.0339761840269
4 0.44936453849220E-15 28.4191507551292
yhrun: interrupt (one more within 1 sec to abort)
yhrun: task[0-3]: running
5 0.44884028024712E-15 28.6471670038895
6 0.44551302644602E-15 28.7812969418413
```

特别注意：

1. yhrun 基本可以替代 mpirun，特别是使用/usr/local/mpi3 目录下 mpi 编译的程序，完全可以使用 yhrun 提交任务，而不需使用 mpirun。

2. yhrun 为交互式作业提交方式，用户如需要和程序进行交互，则选择直接使用 yhrun 提交任务，如果不需要交互，则需使用批处理作业提交方式。

3. yhrun 提交的任务，如果没有进行输入输出的重定向，在关闭登陆客户端软件时，会导致任务中断，因此如无特殊需要，请直接使用 yhrun 提交任务时，重定向输入输出，并保留相应的 log 文件，方便遇到问题时，技术人员及时解决。

重定向举例如下：

```
yhrun -p test -N 16 -n 128 ./a.out >log 2>&1 &
```

>为重定向符号，2>&1 表示标准错误输出重定向至标准输出，最后的&表示后台提交方式，这样保证了该任务在登陆客户端关闭时依然保持不中断。

4. 再次提示，为了保证任务的稳定性，如无特殊需要请使用批处理作业提交方式。

1.2.2 批处理作业 yhbatch

由于交互需求,才考虑直接使用 yhrun 提交任务。如无交互需求,或不能直接使用 yhrun

提交任务，请使用批处理作业提交任务。

批处理作业是指用户编写作业脚本，指定资源需求约束，然后作为作业提交。提交批处理作业的命令为 yhbatch，用户提交命令后即执行结束，返回命令行窗口，但此时作业在进行排队调度，在资源需求被满足是，分配完计算结点之后，系统将在所分配的第一个计算结点上加载执行用户的作业脚本。

批处理作业使用 yhbatch 命令提交，用户在 yhbatch 的参数中指定资源分配的需求约束，编写的作业脚本中，也可以使用 yhrun 命令加载计算作业，此时 yhrun 通过环境变量感知已经分配了资源，从而直接创建作业而不再次提交作业。

批处理作业的脚本为一个文本文件，脚本第一行以“#!”字符开头，并制定脚本文件的解释程序，如 sh，bash，rsh ，csh 等。

这种作业提交方式，适合那些需要指定资源，且带有自己执行命令的计算作业，或者需要连续执行多个任务的作业，用户可以在脚本中提交多个任务，逐个计算。

如前所述，系统中作业的运行分成两步：资源分配与任务加载。批处理作业使用 yhbatch 提交脚本的方式运行，yhbatch 负责资源分配，yhbatch 获取资源后，会在获取资源的第一个结点运行提交的脚本。

举例一如下：

用户的脚本为 mybash.sh 如下：

```
#! /bin/bash

yhrun  -n 16  -p  tryput  -w cn[0-1] hostname
```

根据该脚本用户提交批处理作业，需要明确申请的资源为 tryout 分区的结点 cn[0-1]，因此用户提交如下的批处理命令即可：

```
yhbatch  -w cn[0-1]  -p tryout ./mybash.sh
```

此时注意，给文本文件可执行权限，利用 chmod 命令，使用为：chmod +x filename（其中 filename 替换为你需要修改的文件名。）

计算完成后，工作目录中会生成以 slurm 开头的.out 文件为输出文件。

yhbatch 包含多个选项，基本和 yhrun 类似，用户可以通过 yhbatch --help 命令查看相应所需参数。

举例二 ：

yhbatch 提交的脚本中即可以包含 yhrun，也可以支持 mpirun 等提交作业方式。例如

使用了/usr/lib64/openmpi/1.4-gcc 目录下的 openmpi 编译生成可执行程序 a.out，需要运行在结点 cn12-cn27，共计 16 个结点 128 个进程。则安装 mpirun 提交任务的规则，需要撰写 hostlist 文件包含 cn12-cn27，如下所示：

```
cn12:8
cn13:8
cn14:8
cn15:8
cn16:8
cn17:8
cn18:8
cn19:8
cn20:8
cn21:8
cn22:8
cn23:8
cn24:8
cn25:8
cn26:8
cn27:8
```

之后撰写脚本 sub.sh 如下：

```
#!/bin/bash
/usr/lib64/openmpi/1.4-gcc/bin/mpirun -hostfile hostlist -np 128 ./a.out
```

用户根据该脚本（chmod 修改该脚本可执行权限 chmod +x sub.sh），提交批处理命令如下：

```
yhbatch -N 16 -p test -w cn[12-27] ./sub.sh
```

特别提示：

批处理作业提交模式，试用范围很广，由于手册篇幅限制，不能详述，如果您在提交批处理作业的过程中遇到了任何问题，请联系中心技术人员。

1.2.3 分配模式作业 yhalloc

分配作业模式类似于，交互式作业模式和批处理作业模式的融合。用户需要指定资源分配的需求条件，向资源管理器提出作业的资源分配请求。作业排队，当用户请求资源被满足时，将在用户提交作业的结点上，执行用户所指定的命令，指定的命令执行结束后，也运行结束，用户申请的资源被释放。

yhalloc 后面如果没有跟定相应的脚本或可执行文件，则默认选择了/bin/sh，用户获得了一个合适环境变量的 shell 环境。

yhalloc 和 yhbatch 最主要的区别是，yhalloc 命令资源请求被满足时，直接在提交作业的结点执行相应任务。而 yhbatch 则当资源请求被满足时，在分配的第一个结点上执行相应任务。

yhalloc 在分配资源后，再执行相应的任务，很适合需要指定运行结点，和其它资源限制，并有特定命令的作业。例如 ansys 或其他工程仿真软件的模块，以 ansys 的 lsdyna 模块为例，在并行计算机系统中，lsdyna12.1 版本，需要指定相应的 memory，相应的执行结点列表。由于用户需要在命令中指定相应计算结点，则适合用 yhalloc。

例如：ansys 用户需要 8 个结点，32 个进程，每个结点 4 核的计算资源，利用

yhalloc，有两种提交方式。

第一种首先申请资源，执行如下命令：

```
yhalloc  -N 8  -n 32
```

通过 yhq 查看相应的 jobID 为 163，结点为 cn[50-57],则用户可以选择如下方式：

```
ssh cn50
```

切换到 cn50 结点，之后执行如下命令：

```
lsdyna121  pr=dyna  -dis  memory=250m  i=test.k  o=test.out \
 -machines  cn60:4:cn61:4:62:4:63:4:64:4:65:4:66:4:67:4
```

则可以正常执行 lsdyna 程序。

第二种作业提交方式：

首先通过 yhi 命令，查看哪些结点空闲，确定 8 个空闲的结点，如确定的 8 个空闲结点为 cn[54-61]，则写如下脚本 lsdyna.sh：

```
#!/bin/bash
lsdyna121  pr=dyna  -dis  memory=250m  i=test.k  o=test.out \
```

-machines cn64:4:cn65:4:66:4:67:4:68:4:69:4:70:4:71:4

然后执行如下命令：

yhalloc -N 8 -n 32 -w cn[54-61] ./lsdyna.sh

使用如上方式，请注意，通过 chmod +x lsdyna.sh 给脚本加可执行权限。

yhalloc 包含多个选项，基本和 yhrun 类似，用户可以通过 yhalloc --help 命令查看相应所需参数。

特别提示：

1. yhalloc 和 yhbatch 的使用方法类似，主要区别为任务加载点不同，yhalloc 命令资源请求被满足时，直接在提交作业的结点执行相应任务。而 yhbatch 则当资源请求被满足时，在分配的第一个结点上执行相应任务。

2. yhalloc 提交的作业，如果需要关闭客户端，请重定向输入输出，并后台提交，可参考 3.3.1 小节的特别提示第三条。

1.3 任务取消 yhcancel

用户可以使用 yhcancel 命令取消自己的作业或作业步。命令格式如下：

yhcancel jobid

对于排队作业，取消作业将简单地把作业标记为 CANCELLED 状态而结束作业。对于运行中或挂起的作业，取消作业将终止作业的所有作业步，包括批处理作业脚本，将作业标记为 CANCELLED 状态，并回收分配给作业的结点。一般地，批处理作业将会马上终止；交互作业的 yhrun 进程将会感知到任务的退出而终止；分配模式作业的 yhalloc 进程不会自动退出，除非作业所执行的用户命令因作业或任务的结束而终止。但是在作业被取消时，控制进程都会发送通知消息给分配资源的 yhrun 或 yhalloc 进程。用户可以选择通过 yhalloc 的 --kill-command 选项设置在收到通知时向所执行的命令发送信号将其终止。

1.4 备注

由于手册篇幅限制，只列出了对于绝大多是用户比较重要的相关内容，如您有其他需求也可以联系中心技术人员。

附录（简明常用命令）

一、查看节点及分区状态

1、yhi 查看所有节点及分区状态

2、yhi －n cn[xx-yy]查看指定节点及这些节点所在分区状态

3、yhi －p partitionName 查看指定分区及该分区里节点的状态

二、查看作业

1、yhq 查看所有节点

2、yhq －n xhpl 查看所有作业名为 xhpl 的作业

3、yhq －t R 查看状态为 R（其它状态有 PD，S，CG 等等）的所有作业

4、yhq －w cn[xx-yy]查看分配给 cn[xx-yy]节点的作业

5、以上几条可以一起使用，比如 yhq －u test －n xhpl －p work

6、yhacct 查看历史作业信息

7、yhattch 8999.0 查看作业步任务输出，8999.0 为 yhq 查到的作业 id 加上 .0

三、作业控制

1、yhcancel 取消所有作业

2、yhcancel －u test 取消用户 test 的所有作业

3、yhcancel －n xhpl 取消所有作业名为 xhpl 的作业

4、yhcancel －t R 取消状态为 R（其它状态有 PD，S，CG 等等）的所有作业

5、yhcancel －w cn[xx-yy]取消分配给 cn[xx-yy]节点的作业

6、以上 5 条可以一起使用，比如 yhcancel －u test －n xhpl －p work

7、yhcontrol suspend job_id 挂起指定作业

8、yhcontrol resume job_id 恢复指定作业