目　录

1 环境变量管理工具 module

1.1 简介

由于不同用户在“天河一号”上可能需要使用不同的软件环境，配置不同的环境变量，故默认配置无法满足所有用户的需要，因而在“天河一号”上安装了“module 软件”来对环境变量进行管理，以方便用户更好地使用机器。

“module”通过配置“modulefile”支持环境变量的动态修改，能够控制不同版本软件对环境变量的依赖关系。用户通过简单的命令即可获得适于自己环境变量的设置，因而提高了用户移植软件的效率。

1.2 基本命令

已经在登录服务节点上配置好“module” 工具，主要用法如下：

module avail：查看可用的模块列表,如图 1-1 所示。

```
[pp_cs@ln4%tianhe ~]$ module avail

------------------------------------------------------------------- /usr/share/Modules/modulefiles -------------------------------------
dot           module-cvs    module-info   modules       mpich2-x86_64 null          use.own

------------------------------------------------------------------------ /etc/modulefiles ------------------------------------------------
CPMD/3-1.5(default)            OPENFOAM_v3.0+/v3.0+          compat-openmpi-x86_64          lammps/10-Aug2015(default)     openmpi-x86_64
Gromacs/5.0.4(default)         Siesta/exit                   cp2k/2.6.0(default)            mvapich-psm-x86_64
MPICH/3.1.4(default)           ansys145/14.5(default)        cp2k_2.61/2.6.1(default)       mvapich2-psm-x86_64
MSC/2013(default)              ansys150/15.0(default)        fftw/3.3.3(default)            mvapich2-x86_64
NAMD/2015-01-21(default)       compat-openmpi-psm-x86_64     intellib_dynamic/11.1(default) openmpi/1.6.5(default)
```

图 1-1 module avail 使用

module load [modulesfile]：能够加载需要使用的 modulefiles，如图 1-2 所示。

```
[pp_cs@ln4%tianhe ~]$ module load ansys150
[pp_cs@ln4%tianhe ~]$ which ansys150
/vol6/appsoftware/ansys_inc/v150/ansys/bin/ansys150
[pp_cs@ln4%tianhe ~]$
```

图 1-2 module load 使用

使用 module 加载软件（lammps/10-Aug2015） 的配置环境，如图 1-3 所示。

```
[pp_cs@ln4%tianhe ~]$ which lmp_g++3
/usr/bin/which: no lmp_g++3 in (/vol6/home/pp_cs/anaconda2/bin:/vol6/home/pp_cs/amber14/bin:/vol6/home/pp_cs/g
r/kerberos/sbin:/usr/kerberos/bin:/opt/intel/Compiler/11.1/059/bin/intel64:/opt/intel/Compiler/11.1/059/bin/in
/bin:/usr/local/sbin:/usr/sbin:/sbin:/vol6/home/pp_cs/gaussian/bin/g09/bsd:/vol6/home/pp_cs/gaussian/bin/g09/l
p_cs/gaussian/bin/g09:/vol6/home/pp_cs/openmpi/bin:/vol6/home/pp_cs/bin)
[pp_cs@ln4%tianhe ~]$ module load lammps
[pp_cs@ln4%tianhe ~]$ which lmp_g++3
/vol6/appsoftware/lammps-10Aug15/bin/lmp_g++3
[pp_cs@ln4%tianhe ~]$
```

图 1-3 module 加载软件的配置环境

module unload [modulesfile]：移除使用 module 加载的软件环境，如图 1-4 所示。

```
[pp_cs@ln4%tianhe ~]$ module unload lammps
[pp_cs@ln4%tianhe ~]$ which lmp_g++3
/usr/bin/which: no lmp_g++3 in (/vol6/home/pp_cs/anaconda2/bin:/vol6/home/pp_cs/amber14/bin:/vol6/hom
r/kerberos/sbin:/usr/kerberos/bin:/opt/intel/Compiler/11.1/059/bin/intel64:/opt/intel/Compiler/11.1/0
/bin:/usr/local/sbin:/usr/sbin:/sbin:/vol6/home/pp_cs/gaussian/bin/g09/bsd:/vol6/home/pp_cs/gaussian/
p_cs/gaussian/bin/g09:/vol6/home/pp_cs/openmpi/bin:/vol6/home/pp_cs/bin)
[pp_cs@ln4%tianhe ~]$
```

图 1-4 module unload 使用

module其它用法，可在help中查询。

2 编译器

2.1 Intel 编译器

TH-1系统上安装了Intel编译器11.1版本，支持C，C++，Fortran77和Fortran90语言程序的开发。

Intel 11.1编译器的安装路径位于/opt/intel/Compiler/11.1/059/目录中，其中： C和C++编译器，以及Fortran77/90的相应命令程序目录为：/opt/intel/Compiler/11.1/059/bin/intel64，编译命令为icc,icpc,ifort等。

运行所需的lib 库，在/opt/intel/Compiler/11.1/059/lib/intel64下。用户使用时，需导入环境变量LD_LIBRARY_PATH：

export LD_LIBRARY_PATH=/opt/intel/Compiler/11/059/lib//intel64:$LD_LIBRARY_PATH

Intel 11.1对应的mkl的安装路径为/opt/intel/Compiler/11.1/059/mkl，用户可以使用该目录下的lib/em64t 的mkl库。

用户在使用mkl库计算任务时，需要设置相应环境变量LD_LIBRARY_PATH：

export LD_LIBRARY_PATH=/opt/intel/Compiler/11.1/059/mkl:$LD_LIBRARY_PATH

注意：

1. 用户默认环境变量PATH 已经设置包含了/opt/intel/Compiler/11.1/059/bin/intel64，用户可以直接使用icc，icpc，ifort 等进行编译。

2. ln0-4为登录和编译节点，为完整的操作系统，而计算节点为了保证计算效率，安装的为精简操作系统，所以用户运行时需要指定动态链接库为共享文件系统下的目录/vol6/intellib/lib/intel64/，在此目录下指定需要加载的动态链接库。

2.2 GCC 编译器

TH-1 默认安装的 GCC 版本为 4.4.7，相关的编译命令都安装到/usr/bin 目录中。

2.3 MPI 编译环境

由于TH-1采用了自主互连的高速网络，因此底层MPI为自主实现，基于Intel编译器进行编译。

基于Intel编译器的mpi版本安装目录在/usr/loca/mpi3下，为了追求最高效率，该目录下的mpi为自主实现的mpi版本，底层用Intel编译器编译。基本使用时（运行程序没特殊要求时）推荐使用/usr/local/mpi3 版本，有较高的效率。

并行mpi编译环境使用注意事项：

1. TH-1安装了自主实现的mpi，程序如无特殊需要，推荐使用/usr/local/mpi3 目录下的mpi。该mpi调用Intel 11编译器，且该mpi的库均为静态库，用户不用担心动态链接库问题。

2. TH-1具备自主高速互联网络，并提供MPI编程环境，如用户必须使用其他版本mpi，比如openmpi1.4.8，mpich2-1.3.1 等，也可以自己安装并部署。用自行mpi编译的程序，同样可以利用高速互联网络的虚拟以太网运算任务，但性能会较TH-1自主MPI低很多。

MPI编译命令内部会自动包含MPI标准头文件所在的路径，并自动连接所需的MPI通信接口库，所以不需要用户在命令行参数中指定。

如果用户使用makefile或autoconf编译MPI并行程序，还可以将makefile中的CC，CXX，F77，F90等变量设置成mpicc，mpicxx，mpif77，mpif90，或在autoconf 的configure 过程前设置CC，CXX，F77和F90等环境变量为mpicc，mpicxx，mpif77和mpif90等。

2.4 CUDA 编译环境

由于计算节点，每个节点包括一个M2050 GPU，因此LN0-LN3中，有相应的CUDA编译环境。CUDA编译环境包含三个部分，编译器、SDK和设备驱动，目前计算节点CUDA编译环境已经更新至CUDA7.5，用户可以选择相应的编译器。

CUDA编译器及SDK部署在/vol6/cuda7.5/cuda-7.5/目录下，请用户选择cuda-7.5使用。其中cuda-7.5为 CUDA 7.5编译器；目前我们已经有了CUDA7.5的环境。此外，该目录下还有包括4.0和6.0在内的三套早版本的cuda编译器。

用户使用CUDA进行程序编译时需添加如下环境变量声明：（以cuda7.5为例）

```
export PATH=/vol6/cuda7.5/cuda-7.5/bin:$PATH

export LD_LIBRARY_PATH=/vol6/cuda7.5/cuda-7.5/lib64:$LD_LIBRARY_PATH
```

注意：目前节点的 **CUDA** 版本已经升级至 **7.5**，因此请大家在编译运行 **GPU** 程序时，选择使用 **CUDA7.5** 的编译和运行环境，以及相应的动态链接库。

3 作业提交

注意：“天河一号”系统采用独占式作业提交模式，即作业一旦提交到计算节点，则该节点被您独占使用。“天河一号”每节点核数为 12，因此，使用时请尽量保证使用核数为 12 的整数倍，以节省您的机时。

3.1 节点状态查看 yhinfo 或 yhi

yhi 为 yhinfo 命令的简写，用户用其查看节点状态，如图 3-1 所示。

```
[para08@ln4%tianhe ~]$ yhi
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
u_test*      up       5:00      2  drain cn[1068,1077]
u_test*      up       5:00     16   idle cn[1062-1067,1069-1076,1078-1079]
work         up   infinite    109 drain* cn[108,121,130-131,151,159,251,254,278-279,302,
538,543,549,553,574,578-579,587,604,608,614-615,632,650,654-656,690-691,707,711,715,719-
971,982-983,985,995,1654,1661,1668,1672-1673,1703,1713]
work         up   infinite      4   drng cn[379,606,1721,1727]
work         up   infinite     59  drain cn[264-265,320,430-433,485-486,514-515,539,546,
728-729,732,755,758-759,762-763,765,767,773,776-777,821,984,1604,1635,1639,1653,1656,168
work         up   infinite    335  alloc cn[122-129,132-135,150,152-158,160-175,246-250,
413,426-429,434-436,481,483-484,487-489,491-512,528-530,532,534,540-542,544-545,547,550-
693,708-710,778-784,810,824-831,986-994,1600-1603,1605-1634,1638,1647-1648,1686-1702,170
work         up   infinite    274   idle cn[109-120,145-149,176-191,266-277,280-287,290-
653,658-664,676-689,692,694-696,698,700-706,712-714,716-718,768-772,786-792,794-809,811-
4-1677,1679-1680,1682-1684,1710-1712,1716-1719,1722]
work         up   infinite      1   down cn1678
[para08@ln4%tianhe ~]$
```

图 3-1 查看节点状态

其中 PARTITION 表示分区，NODES 表示节点数，NODELIST 为节点列表，STATE 表示节点运行状态。其中，idle 表示节点处于空闲状态，allocated 表示节点已经被分配了一个或多个作业。

3.2 作业状态信息查看 yhqueue 或 yhq

注意：使用 yhq 命令的时候加上-u 参数，看到的才是自己作业运行的情况。如 yhq -u para08

yhq 为 yhueue 命令的简写，用户用其查看作业运行情况，如图 3-2 所示。

```
[para08@ln4%tianhe ~]$ yhq
             JOBID PARTITION     NAME         USER ST       TIME  NODES NODELIST(REASON)
           1105806      work  high.sh       kd_zyj  R 10-15:46:30     42 cn[305-313,550-552,
           1103834      work juchishu      kd_zyj4  R 16-21:03:47     15 cn[380-394]
           1101175      work 50k-nrbc      kd_zyj3  R 23-19:15:03     10 cn[1705-1709,1723-1
           1101172      work 40k-nrbc      kd_zyj3  R 24-01:23:49     10 cn[150,304,317,379,
           1102909      work   test22      hnu_ckq  R 18-20:43:06      6 cn[778-783]
           1102904      work   test22      hnu_ckq  R 18-20:56:32      6 cn[581-586]
           1105154      work   csz.sh      hnu_ckq  R 13-01:02:10      3 cn[357-359]
           1104834      work frappe8r       pb_wzw  R 14-00:07:58      1 cn534
           1099643      work      gk8      kd_zyj4  R 15-11:40:50      2 cn[252-253]
           1104173      work  frappe9       pb_wzw  R 16-15:58:45      1 cn810
           1104171      work  frappe7       pb_wzw  R 16-15:58:46      1 cn607
           1104170      work  frappe6       pb_wzw  R 16-15:58:47      1 cn580
           1104163      work frappe10       pb_wzw  R 16-16:12:12      1 cn633
```

图 3-2 查看作业运行情况

其中 JOBID 表示任务 ID，Name 表示任务名称，USER 为用户，ST 为任务状态，TIME 为已运行时间，NODES 表示占用节点数，NODELIST 为任务运行的节点列表。

3.3 交互式作业提交 yhrun

3.3.1 简介

交互式提交作业：在 shell 窗口中执行 yhrun 命令，主要命令格式如下：

yhrun [options] program

3.3.2 yhrun 常用选项

yhrun包括多个选项，其中最常用的选项主要有以下几个：

• -n, --ntasks=number

指定要运行的任务数。请求为number 个任务分配资源，默认为每个任务一个处理器核。

• -c, --cpus-per-task=ncpus

告知资源管理系统控制进程，作业步的每个任务需要ncpus 个处理器核。若未指定此选项，则控制进程默认为每个任务分配一个处理器核。

• -N, --nodes=minnodes[-maxnodes]

请求为作业分配至少minnodes个节点。调度器允许在多于minnodes个节点上运行作业。可以通过maxnodes限制最多分配的节点数目（例如“-N 2-4”或“--nodes=2-4”）。最少和最多节点数目可以相同，以指定特定的节点数目（例如，“-N 2”或“--nodes=2-2”将请求两个节点）。分区的节点数目限制将覆盖作业的请求。如果作业的节点限制超出了分区中配置的节点数目，作业将被拒绝。如果没有指定-N，缺省行为是分配足够多的节点以满足-n和-c参数的需求。在限制范围内以及不延迟作业运行的前提下，作业将被分配尽可能多的节点。

• -p, --partition=partition name

在指定分区中分配资源。如未指定，则由控制进程在系统默认分区中分配资源。

• -w, --nodelist=node name list

请求指定的节点名字列表。作业分配资源中将至少包含这些节点。列表可以用逗号分隔的节点名或节点范围（如cn[1-5,7,...]）指定，或者用文件名指定。如果参数中包含“/”字符，则会被当作文件名。如果指定了最大节点数如-N 1-2，但是文件中有多余2个节点，则请求列表中只使用前2个节点。

• -x, --exclude=node name list

不要将指定的节点分配给作业。如果包含“/”字符，参数将被当作文件名。yhrun将把作业请求提交到控制进程，然后在远程节点上启动所有进程。如果资源请求不能立即被满足，yhrun将阻塞等待，直到资源可用于运行作业。如果指定了--immediate选项，则yhrun将在资源不是立即可用时终止。

• -h, --help

若需使用 yhrun 更多选项，可通过“yhrun -h”或“yhrun --help”查看。

3.4 批处理作业 yhbatch

3.4.1 简介

批处理作业是指用户编写作业脚本，指定资源需求约束，提交后台执行作业。提交批处理作业的命令为yhbatch，用户提交命令即返回命令行窗口，但此时作业在进入调度状态，在资源满足要求时，分配完计算节点之后，系统将在所分配的第一个计算节点（而不是登录节点）上加载执行用户的作业脚本。

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。由于计算节点为精简环境，只提供 sh 和 bash 的默认支持。

3.4.2 使用示例

例如用户的脚本名为 test.sh，内容如图 3-3 所示：

```
#!/bin/bash
yhrun -p work -N 1  hostname > mf1
fluent 3d -g -t24  -cnf=mf1  < inputfile
```

图 3-3 使用示例

根据该脚本用户提交批处理作业，需要明确申请的资源为分区的1个节点。

注意：需给该文本文件设置 test.sh 可执行权限，利用命令：chmod +x test.sh，如图 3-4 所示。

```
[para08@ln4%tianhe 3d]$ chmod +x test.sh
[para08@ln4%tianhe 3d]$ ll -ltr
total 26580
-rw-rw-r-- 1 para08 para08 11089721 Dec 18  2003 turbo1.cas
-rw-rw-r-- 1 para08 para08 16116392 Dec 18  2003 turbo1.dat
-rw-rw-r-- 1 para08 para08      178 Aug  1 16:01 inputfile
-rwxrwxr-x 1 para08 para08       89 Aug  1 16:05 test.sh
[para08@ln4%tianhe 3d]$
```

图 3-4 设置 test.sh 可执行权限

用户yhbatch批处理命令如下：

yhbatch -p work -N 1 test.sh

计算开始后，工作目录中会生成以slurm开头的.out 文件为输出文件。

```
[para08@ln4%tianhe 3d]$ yhbatch -p work -N 1 -n 1 test.sh
Submitted batch job 1108979
[para08@ln4%tianhe 3d]$ yhq -u para08
             JOBID PARTITION     NAME         USER ST       TIME  NODES NODELIST(REASON)
           1108979      work  test.sh       para08  R    INVALID      1 cn576
[para08@ln4%tianhe 3d]$ ll -ltr
total 51552
-rw-rw-r-- 1 para08 para08 11089721 Dec 18  2003 turbo1.cas
-rw-rw-r-- 1 para08 para08 16116392 Dec 18  2003 turbo1.dat
-rw-rw-r-- 1 para08 para08      178 Aug  1 16:01 inputfile
-rwxrwxr-x 1 para08 para08       89 Aug  1 16:05 test.sh
-rw-rw-r-- 1 para08 para08 25522095 Aug  1  2016 turbo1-end.dat
-rw-rw-r-- 1 para08 para08    22301 Aug  1  2016 slurm-1108975.out
-rw-rw-r-- 1 para08 para08        6 Aug  1  2016 mf1
-rw-rw-r-- 1 para08 para08    12394 Aug  1  2016 slurm-1108979.out
[para08@ln4%tianhe 3d]$
```

更多选项，用户可以通过 yhbatch --help 命令查看。

3.5 节点资源抢占命令 yhalloc

3.5.1 简介

该命令支持用户在提交作业前，抢占所需计算资源（注意：抢占成功之后就开始计算所用机时）。

3.5.2 使用示例

yhalloc提交方式如下：

首先申请资源，执行如图 3-5 所示命令：

```
[para08@ln4%tianhe fluent]$ yhalloc -p work -N 1
yhalloc: Granted job allocation 1108981
```

图 3-5 执行命令

通过 yhq（-u 指定超算帐号名）查看相应的 jobID 为 1108981，节点为 cn576，如图 3-6 所示。

```
[para08@ln4 fluent]$ yhq -u para08
             JOBID PARTITION     NAME         USER ST       TIME  NODES NODELIST(REASON)
           1108981      work     bash       para08  R    INVALID      1 cn576
[para08@ln4 fluent]$
```

图 3-6 查看相应的 jobID

用户可以选择如图 3-7 所示方式，切换到 cn576 节点，之后执行程序。

```
[para08@ln4 fluent]$ ssh cn576
Last login: Mon Aug  1 16:54:30 2016 from ln4-gn0
[para08@cn576%tianhe ~]$
```

图 3-7 ssh 到计算节点

5.6 任务取消 yhcancel

用户使用yhcancel命令取消自己的作业。命令格式如下：

yhcancel jobid ， jobid可通过yhq获得。

```
[para08@cn576%tianhe 3d]$ yhq -u para08
             JOBID PARTITION     NAME         USER ST       TIME  NODES NODELIST(REASON)
           1108981      work     bash       para08  R       9:07      1 cn576
[para08@cn576%tianhe 3d]$ yhcancel 1108981
yhalloc: Job allocation 1108981 has been revoked.
                                                Killed by signal 1.
[para08@ln4 fluent]$
```

5.7 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，而且具体使用参数也会因执行程序以及计算作业的不同略有差异，如您有其他需求也可以联系超算中心技术人员。

重要提示：

登录节点指ln开头的节点，如ln0,ln1等；计算节点指cn开头的节点，如cn1021等。

1）请不要在登录节点直接运行可执行程序（极大的影响其他用户的登录和使用效率）。

2）如无特殊需要，请使用批处理方式（yhbatch）提交任务，如果有任何问题请联系超算中心技术人员。

3）请保存好运行程序的log文件，从而方便超算中心技术人员在作业出问题后，协助解决问题。

4）若需登录计算节点运行程序，需要先分配计算节点，方可登录。

5）请在提交命令中加入参数选项“-p 分区名”，即提交命令应为“yhrun -p 分区名 ...”或者“yhbatch -p 分区名 ...”。

6 常见上机问题（FAQ）

1. 提交作业后，提示“Invalid partition name specified”。

报该错时，建议用户先用“yhi”查看是否可以看见自己所在的分区。若无法看见分区，则是您的机时已到限制。

2. 提交作业后，提示“Failed to allocate resources: User's group not permitted to use this partition”。

用户提交作业时通常需要加“-p 分区名”这一参数，同时该参数应写在程序名前。分区可用“yhi”来查看所在分区。

3. 如果遇到一些作业运行时报库无法找到，如何处理？

用户可通过locate命令查找相应的库，并将对应的库路径加入环境变量LD_LIBRARY_PATH中。如果还是不行，可将缺少的库拷贝到自己的文件夹如~/lib 中，并设置环境变量 ：“export LD_LIBRARY_PATH=~/lib:$LD_LIBRARY_PATH”。

4. 采用yhrun提交作业，关闭界面后，再次登录时发现作业被killed。

yhrun是交互式提交作业模式，一旦作业提交的界面关闭作业就会被killed。若需要较长时间运行的作业，建议用户采用yhbatch批处理提交方式。yhbatch负责资源分配，yhbatch获取资源后会在获取资源的第一个节点运行提交的脚本，当前登录shell断开后，加载作业仍可正常运行。

5. “ls”等访问文件夹操作很慢。

出现“ls”等访问文件夹操作慢的原因主要有3个：一是网络慢，网络时延大；二是有大量的I/O操作正在进行，造成I/O阻塞；三是该文件夹下的文件过多（有成千上万个文件）。若是原因一和二，通常等一段时间后即可恢复正常；若是原因三，则建议用户整理、清理一下自己的文件夹。

6. 采用yhbatch提交多节点作业失败的原因。

采用yhbatch提交作业首先进行的是分配资源，因此对于多节点作业，采用yhbatch提交时应在提交命令中指定-N参数，即提交命令是“yhbatch -N nodenum -n pronum -p pardition job.sh”。

7. 计算节点无法登录。

目前我们对计算节点做了限制，除非用户分配了计算节点，否则无法登录。用户若想登录计算节点再算题，首先需要yhalloc分配节点，方可登录节点算题。

8. yhalloc分配资源，退出yhalloc后发现作业断掉。

yhalloc与yhbatch最主要的区别是，yhalloc命令资源请求被满足时，直接在提交作业的节点执行相应任务，适合需要指定运行节点和其他资源限制，并有特定命令的作业。当当前登录shell断开后，申请获得的资源以及加载作业任务会退出。

9. 如果遇到一些作业报错，应该如何处理？

较为常见的报错如：“No enough endpoint resources”，“Job credential expired”，“bus error”，用户可以通过日志找到相关的报错节点，在提交作业命令中使用参数“-x 节点名称”剔除掉问题节点重新进行作业提交，如“-x cn1”表示在我申请的资源中不使用cn1这个节点。如遇到相关报错问题也希望您能及时与我们进行联系，并提供您的报错日志信息(并加上错误发生的时间,提交命令等信息)，以便我们进行有效的分析和处理。

10. “天河一号”作业提交模式。

目前“天河一号”系统采用独占式作业提交模式，即作业一旦提交到计算节点，则该节点被您独占使用。也就是说，一旦作业提交到计算节点，即使该节点的CPU核没有用满，其他人的作业也无法提交上去。

11. 作业退出后仍显示CG状态，是否影响作业退出？

CG状态是作业退出时，部分节点上的进程没有完全停止导致，并不影响作业的正常退出。

12. 作业完成退出时显示部分进程被killed，然后退出。

这种情况下，建议用户首先检查所需的输出是否已正常输出完成。导致这种情况出现的原因是有部分进程先完成了计算而提前结束，而当一个作业的部分进程结束，系统默认为作业已完成，在一定时间内其他进程若不结束，则会被强制结束。