使用 Environment module(环境模块) 设置应用环境

上海交通大学高性能计算中心

http://hpc.sjtu.edu.cn

2013 年 11 月 8 日更新

目录

"Environment module"(环境模块) 是一组环境变量设置的集合。module 可以被加载 (load)、卸载 (unload)、切换 (switch)，这些操作会改变相应的环境变量设置，从而让用户方便地在不同环境间切换。相比与将环境变量设置写入/etc/profile或者~/.bashrc，Environment module 操作只影响当前用户的当前登录环境，不会因错误配置造成全局持续的破坏。普通用户也可以自己编写 module，具有很好的定制性。

1 基本命令

1.1 module命令列表

```
$ module
```

或者，

```
$ module -h
```

1.2 查看可用模块avail

```
$ module avail
```

1.3 查看已加载模块list

```
$ module list
```

1.4 加载模块load

```
$ module load MODULE_NAME
```

1.5 卸载模块unload

```
$ module unload MODULE_NAME
```

1.6 切换模块switch

```
$ module switch OLD_MODULE NEW_MODULE
```

等价于:

```
$ module unload OLD_MODULE; module load NEW_MODULE
```

1.7 卸载所有已加载的模块purge

```
$ module purge
```

1.8 显示模块说明whatis

```
$ module whatis MODULE_NAME
```

1.9 显示该模块内容display

```
$ module display MODULE_NAME
```

2 Pi 集群 module 功能说明

π 集群预设了如下 module:

```
$ module avail
----------------- /lustre/utility/modulefiles -----------------
compiler-default    mpi-default
fftw/impi/3.3.3         fftw/openmpi/gcc/3.3.3 icc/13.1.1
cuda/5.0                fftw/mpich2/gcc/3.3.3  fftw/openmpi/icc/3.3.3 impi/4.1.1.036
cuda-default            fftw/mpich2/icc/3.3.3  gcc/4.8.1              mkl/11.0.3
openmpi/gcc/1.6.4       mpich2/gcc/1.4.1p1     pgi/13.9
openmpi/icc/1.6.4       mpich2/icc/1.4.1p1
```

模块命名规则是:

```
软件名
/库MPI编译器版本//
```

其中“MPI 库”和“编译器”是命名时的可选项。譬如，`fftw/mpich2/gcc/3.3.3`模块表示版本号为 3.3.3 的 FFTW 库，这个库支持在 MPICH2 上并行执行，FFTW 和 MPICH2 库都使用 GCC 生成。又如，`openmpi/gcc/1.6.4`模块表示版本号为 1.6.4 的 OpenMPI 库，这个库使用 GCC 生成。

这些模块按功能大致可分为编译器、MPI 环境、工具库等，下面分别予以说明。

2.1 编译器

π 集群上可以使用的编译器包括：GNU 编译器 (GCC)、Intel 编译器、PGI 编译器。GCC-4.4.6 编译器安装在操作系统目录下，能直接使用，不需要加载模块。GCC-4.8.1、Intel 编译器和 PGI 编译器需要加载相应模块。编译器模块信息如下:

模块	编译器版本	C 编译器	C++ 编译器	F77 编译器	F90 编译器
gcc/4.4.6(默认)	4.4.6	gcc	g++	g77	gfortran
gcc/4.8.1	4.8.1	gcc	g++	g77	gfortran
icc/13.1.1	13.1.1	icc	icpc	ifort	ifort
pgi/13.9	13.9	pgcc	pgc++	无	无

π 集群上的 Nvidia CUDA 开发环境版本为 5.0，使用前请加载模块`cuda-default`或者`cuda/5.0`。

2.2 MPI 环境

π 集群可用的 MPI 库比较丰富，包括 Intel MPI(IMPI)、MPICH2 和 OpenMPI。按照`mpicc`、`mpicxx`、`mpif77`和`mpif90`使用的后端编译器，MPICH2 与 OpenMPI 还可以再细分为不同版本。

模块	版本	mpicc	mpicxx	mpif77	mpif90
impi/4.1.1.036	4.1.1	gcc (mpiicc uses icc)	g++ (mpicpc uses icpc)	g77	gfortran
mpich2/icc/1.4.1p1	1.4.1p1	icc	icpc	ifort	ifort
mpich2/gcc/1.4.1p1	1.4.1p1	gcc	g++	g77	gfortran
openmpi/icc/1.6.4	1.6.4	icc	g++	g77	gfortran
openmpi/gcc/1.6.4	1.6.4	gcc	g++	g77	gfortran

2.3 工具库

π 集群上的工具库模块包括：`mkl`(Intel 数学函数库)、`fftw`（FFTW 快速傅里叶变换库)。用户可根据需要，载入响应模块。

3 在编译和提交作业时使用 module

在集群上使用编译器和特定软件库时，往往需要在启动脚本中设置复杂的环境变量。让用户手工维护这些设置不仅容易出错，而且用户到另一个集群工作时，又需要针对新环境逐一修改变量。使用 Environment module 后，将环境准备的工作交给管理员，用户按需加载模块，用同一组命令就能在不同集群上完成环境设定的工作。

下面以 Intel MPI 程序的编译和提交为例，说明 module 的作用。

3.0.1 编译时使用 module

Intel 编译器/MPI 环境的设定包括一系列复杂的环境变量设定，需要运行脚本完成配置。典型过程如下：

```
$ source /lustre/utility/intel/composer_xe_2013.3.163/bin/compilervars.sh intel64
$ source /lustre/utility/intel/mkl/bin/intel64/mklvars_intel64.sh
$ source /lustre/utility/intel/impi/4.1.1.036/bin64/mpivars.sh
$ mpiicc -o mpihello mpihello.c
```

使用 module 可以使环境设定的过程更清晰：

```
$ module load icc/13.1.1
$ module load mkl/11.0.3
$ module load impi/4.1.1.036
$ mpiicc -o mpihello mpihello.c
```

3.0.2 LSF 提交作业时使用 module

使用 LSF 提交作业时，作业控制脚本中通常也会包含一系列环境设定脚本，譬如：

```
source /lustre/utility/intel/composer_xe_2013.3.163/bin/compilervars.sh intel64
source /lustre/utility/intel/mkl/bin/intel64/mklvars_intel64.sh
source /lustre/utility/intel/impi/4.1.1.036/bin64/mpivars.sh
```

这部分也可以用 module 指令替代。注意，*LSF* 默认使用*/bin/sh*解析作业脚本指令 *(*没有 *module* 功能*)*，且不传递*$HOME*、*$USER*、*$SHELL*、*$LONGNAME*以外的环境变量。若要在 *LSF* 作业脚本中使用 *module*，我们需要指定一个带有 *module* 功能的 *Shell(*如*/bin/bash)*，并正确设定*MODULEPATH*变量。下面这个 LSF 作业片段供参考：

```
#BSUB -L /bin/bash

MODULEPATH=/lustre/utility/modulefiles:$MODULEPATH
```

```
module load icc/13.1.1
module load mkl/11.0.3
module load impi/4.1.1.036
```

4 编写自定义 module

用户可根据modulefile规则，自己编写所需的 module。用户自定义 module 的目录，加入MODULEPATH变量后方能生效。

5 参考资料

- “Environment Module” http://modules.sourceforge.net/
- “Environment Module: Manual Page” http://modules.sourceforge.net/man/module.html
- “Modules Software Environment” https://www.nersc.gov/users/software/nersc-user-environme modules/