

# 并行政程示例

1	OpenMP示例 .....	1
1.1	使用GCC编译器 .....	2

本文档介绍如何在 $\pi$ 超级计算机上编译和提交并行作业任务。 $\pi$ 支持OpenMP、MPI、CUDA等并行编程模型。再继续阅读本文档之前，您应该知道如何登录 $\pi$ 、使用LSF提交作业、Module的基本概念。下面几个文档可以帮助您完成准备工作：

- 使用SSH登录高性能计算节点
- LSF作业管理系统使用方法
- 使用Environment Module设置运行环境

## 1 OpenMP示例

Pi集群上GCC和Intel编译器都支持OpenMP扩展。示例代码omp\_hello.c内容如下：

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
    int nthreads, tid;

    /* Fork a team of threads giving them their own copies of variables
*/
    #pragma omp parallel private(nthreads, tid)
    {

        /* Obtain thread number */
        tid = omp_get_thread_num();
        printf("Hello World from thread = %d\n", tid);

        /* Only master thread does this */
        if (tid == 0)
        {
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);
        }
    }
}
```

```

    } /* All threads join master thread and disband */
}

```

## 1.1 使用GCC编译器

GCC编译OpenMP代码时加上``-fopenmp``:

```
$ gcc -fopenmp omp_hello.c -o ompgcc
```

在本地使用4线程运行程序:

```
$ export OMP_NUM_THREADS=4 && ./ompgcc
```

正式运行时需要提交到LSF作业管理系统, 提交脚本``ompgcc.lsf``如下, 仍使用4线程运行(增加约束条件让所有线程分配到一台物理机上):

```

#BSUB -L /bin/bash
#BSUB -J HELLO_OpenMP
#BSUB -n 4
#BSUB -e %J.err
#BSUB -o %J.out
#BSUB -R "span[hosts=1]"
#BSUB -q cpu

export OMP_NUM_THREADS=4
./ompgcc

```