

高性能计算实验报告

|  |  |
| --- | --- |
| 实验题目 | OpenMP 、MPI 并行编程 |
| 姓名 | 欧炜标 |
| 学号 | 3121005358 |
| 专业 | 人工智能 |
| 所在学院 | 计算机学院 |
| 指导教师 | 王卓薇 |
| 论文提交日期 | 2023/12/20 |

目录

[一、实验目的 6](#_Toc153823870)

[二、实验内容 6](#_Toc153823871)

[三、实验环境 6](#_Toc153823872)

[四、实验原理 6](#_Toc153823873)

[五、实验步骤及实验结果 7](#_Toc153823874)

[4.1环境配置 7](#_Toc153823875)

[4.2程序编写 8](#_Toc153823876)

[五、实验遇到的问题及其解决方法 10](#_Toc153823877)

[六、实验结论 10](#_Toc153823878)

# 一、实验目的

本实验的目的是通过练习掌握 OpenMP 和MPI 并行编程的知识和技巧。实现多台主机 mpi\_hello\_world 的编译运⾏。

# 二、实验内容

1. 了解并行算法的设计基础；
2. 熟悉OpenMP和MPI的编程环境以及运行环境；
3. 本实验通过在华为鲲鹏上，编译运行简单的 HelloWorld 示例程序。掌握简单的程序编写，集群 MPI 并行计算的配置以及加深对并行计算的了解。

# 三、实验环境

华为鲲鹏云主机、openEuler 20.03 操作系统； 安装 mpich-3.3.2.tar.gz； 每套实验环境可供 1 名学员上机操作

# 四、实验原理

MPI（Message Passing Interface）是一种并行计算的编程模型，它允许多个进程在不同的计算节点上进行通信和协作，从而实现并行计算。MPI 库是实现 MPI 标准的软件库，它提供了一组函数和数据类型，用于在不同的计算节点之间进行通信和同步。

在多台主机上安装 MPI 库后，可以使用相同的编译器和编译选项编译 mpi\_hello\_world 程序。编译后的程序可以在不同的计算节点上运行，每个节点都会启动一个进程，并通过 MPI 库进行通信和同步。

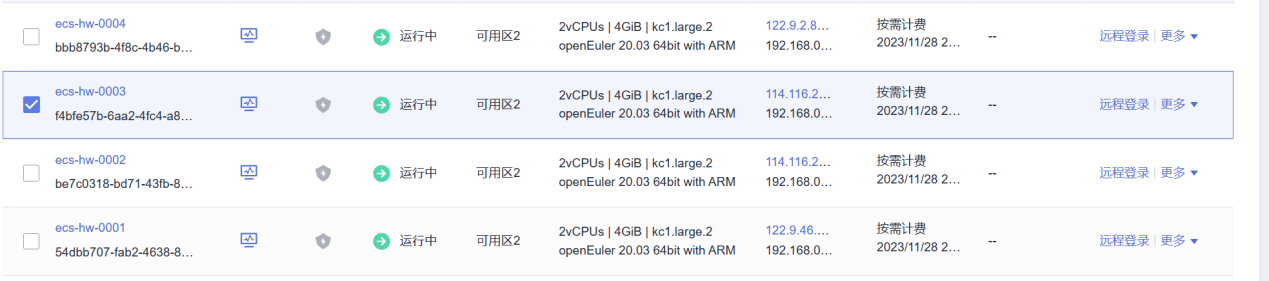
# 五、实验步骤及实验结果

## 4.1环境配置

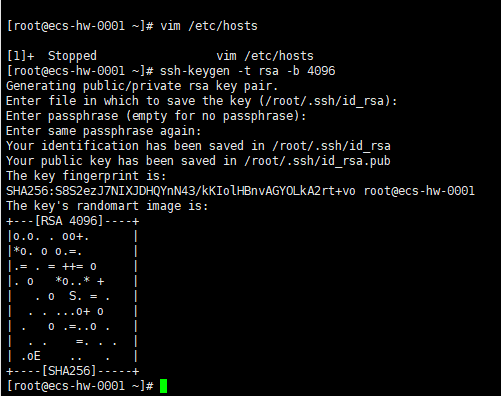
1. 购买华为服务器



1. 创建合适的弹性云服务器



1. 下载xell并再xshell中登录云服务器
2. 在每一台服务器上生成密钥并保存在其他三台服务器中



1. 在每台计算节点上安装相同版本的 MPI 库

## 4.2程序编写

1. 创建示例程序源码，编写 mpi\_hello\_world 程序，并使用相同的编译器和编译选项进行编译。

#include <mpi.h>

#include <stdio.h>

int main(int argc, char\*\* argv) {

// Initialize the MPI environment. The two arguments to MPI Init are not

// currently used by MPI implementations, but are there in case future

// implementations might need the arguments.

MPI\_Init(NULL, NULL);

// Get the number of processes

int world\_size;

MPI\_Comm\_size(MPI\_COMM\_WORLD, &world\_size);

// Get the rank of the process

int world\_rank;

HelloWorld 程序实验手册 第 4 页

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &world\_rank);

// Get the name of the processor

char processor\_name[MPI\_MAX\_PROCESSOR\_NAME];

int name\_len;

MPI\_Get\_processor\_name(processor\_name, &name\_len);

// Print off a hello world message

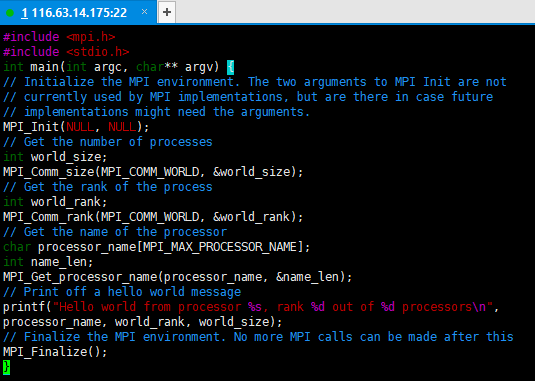
printf("Hello world from processor %s, rank %d out of %d processors\n",

processor\_name, world\_rank, world\_size);

// Finalize the MPI environment. No more MPI calls can be made after this

MPI\_Finalize();

}



1. 创建 makefile

EXECS=mpi\_hello\_world

MPICC?=mpicc

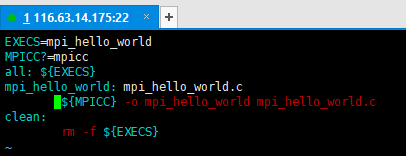
all: ${EXECS}

mpi\_hello\_world: mpi\_hello\_world.c

${MPICC} -o mpi\_hello\_world mpi\_hello\_world.c

clean:

rm -f ${EXECS}



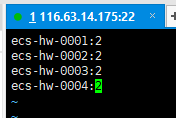
1. 建立主机配置文件，并查看是否配置成功

ecs-hw-0001:2

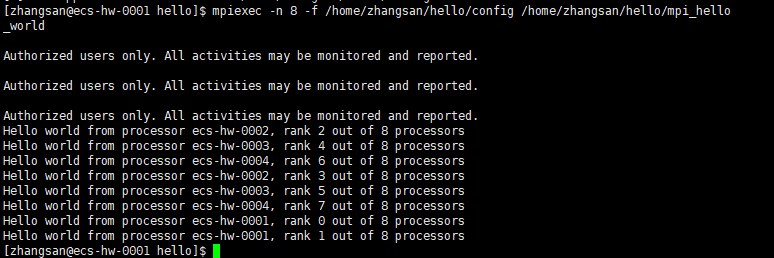
ecs-hw-0002:2

ecs-hw-0003:2

ecs-hw-0004:2



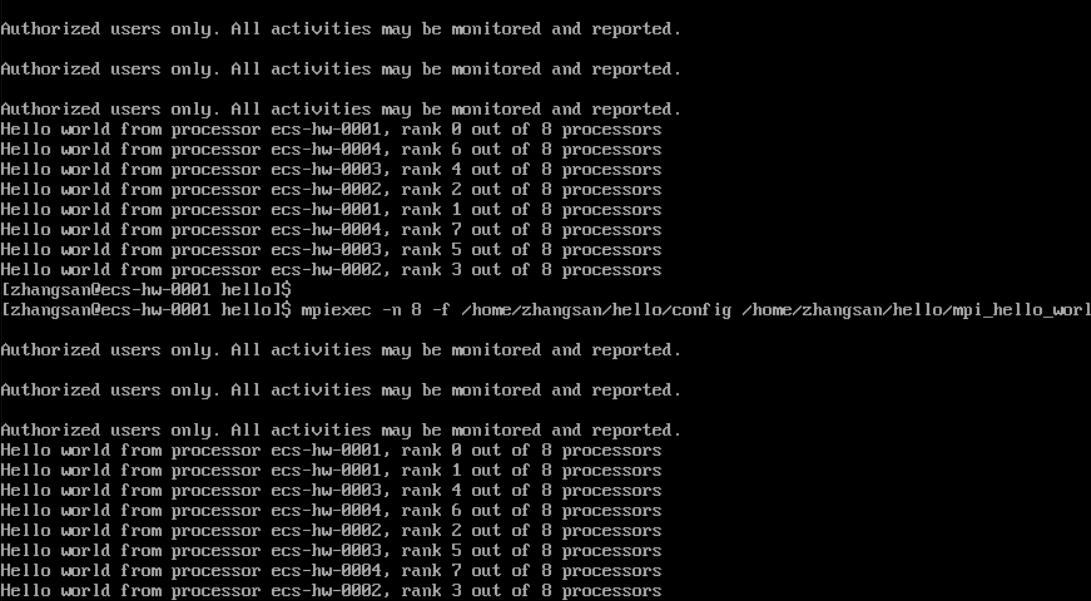
1. 进行make操作



1. 进行运行监测

在ecs-hw-0001 上执行

|  |
| --- |
| mpiexec -n 8 -f /home/zhangsan/hello/config /home/zhangsan/hello/mpi\_hello\_world |



# 五、实验遇到的问题及其解决方法

1. 若按照实验书步骤则会遇到问题，在配置环境章节中执行**./configure**: 这个命令运行名为**configure**的脚本，通常用于配置软件包以进行编译。这一步会根据系统和环境设置配置MPICH。会出问题，因为缺少了g++的安装包，可通过提前额外安装。或使用**./configure ---disable-cxx**
2. 需要区分公有跟私有ip。

# 六、实验结论

本实验的主要任务为在多台主机上编译和运行mpi\_hello\_world，在多台主机上编译和运行 mpi\_hello\_world，需要在每台主机上安装 MPI 库，并使用相同的编译器和编译选项。

在编译时，使用 MPI 的编译器和编译选项，例如 mpicc。在运行时，使用 MPI 的运行命令，例如 mpirun。

在 mpirun 命令中，需要指定使用的主机数量和主机名。例如：

mpirun -np 4 -hostfile hostfile.txt ./mpi\_hello\_world

其中，-np 指定使用的进程数量，-hostfile 指定使用的主机名列表文件，./mpi\_hello\_world 指定要运行的可执行文件。

多台主机之间需要能够互相通信。可以通过设置防火墙规则或者使用 VPN 等方式实现。

通过本实验，我了解了在多台主机上编译和运行mpi\_hello\_world 的基本步骤。这些步骤对于编译和运行其他 MPI 程序也适用，我对并行计算有了更深的认知。