实验 1 高性能计算初步实践

一、实验目的

1. 了解高性能计算的原理
2. 掌握高性能计算集群配置的要点和关键
3. 掌握分布式内存并行计算的两大框架 MPI 和 CUDA 编程的基本要求

二、实验原理

常用的高性能计算框架主要有并行计算和分布式计算，其主要的原理主要分为 SIMD 和 MIMD 两大类。

MPI 是 Message Passing Interface 的缩写，是分布式内存之间实现消息通信最为流行的编程模型或规范/标准/协议。在集群系统中，集群的各节点之间可以采用 MPI 编程模型进行程序设计，每个节点都有自己的内存，可以对本地的指令和数据直接进行访问，各节点之间通过互联网络进行消息传递，这样设计具有很好的可移植性，完备的异步通信功能，较强的可扩展性等优点。MPICH 和 OpenMPI 是最常用的两种采用 MPI 标准的并行计算开源软件库。

近年来，随着 GPU 的快速发展，CUDA 编程得到了广泛的应用。本课程我们要学习 CUDA 编程的基础，有兴趣的同学可在课外学习更多这块的知识。

三、实验步骤

1. 如何用 PC 配置高性能计算集群

- 集群的整体设计

  集群主要由一个管理节点和若干个计算节点组成：

  1) 管理节点的 IP 为 192.168.5.x0；计算节点为 192.168.5.x1-192.168.5.x4；网关为 192.168.5.1，子网掩码为 255.255.255.0；DNS 设置为 202.120.2.101。
  2) 这里的 x 可以为 1-9 之间的数值，保证组与组之间互不冲突。
  3) 管理节点上启动 NFS 服务，共享目录为/opt 和/share/home，其他计算节点作为客户端。
  4) 管理节点作为 NIS 的服务端，其他计算节点作为客户端。
  5) 计算任务只能从管理节点提交，其他计算节点执行计算任务。

- 配置网络

  1) 修改所有节点上的文件/etc/syconfig/network-script/ifcfg-eth0

```
IPADDR=”192.168.5.11”
NETMASK=”255.255.255.0”
BROADCAST=”192.168.5.255”
GATEWAY=”192.168.5.1”
DNS=”201.120.2.101”
```

同时注意关闭网络防火墙和 selinux：

- ✓ 对于 CentOS6：采用 iptables -F 命令；
- ✓ 对于 CentOS7：采用 systemctl stop firewalld.service 关闭防火墙；

2) 修改所有节点的文件/etc/hosts，包含集群中的所有节点的主机名与 IP 的对应关系，例如

```
192.168.5.10    bio10
192.168.5.11    bio11
192.168.5.12    bio12
192.168.5.13    bio13
192.168.5.14    bio14
```

◆ 配置 NFS

NFS 是网络文件系统（Network File System）的缩写，其目的是实现集群内部的共享文件系统。配置的关键是主节点（master node）上的/etc/exports 文件和从节点（slave nodes）上的/etc/fstab 文件。注意的是，需要启动主节点上的 nfs 服务。

1) 在管理节点和计算节点上安装 nfs-utils 和 rpcbind

2) 管理节点上启动 nfs 服务，先修改/etc/exports：

```
/home   192.168.5.0/24(rw,no_root_squash,sync)
```

然后启动 rpcbind 和 nfs 服务：

```
shell# systemctl enable rpcbind.service
shell# systemctl enable nfs-server.service
shell# systemctl start rcpbind.service
shell# systemctl start nfs-server.service
```

3) 管理节点上运行 exportfs -a

4) 计算节点上修改配置/etc/fstab 文件，然后执行 mount -a：

```
# <file system> <mount point> <type> <options> <dump> <pass>
mu01:/home      /home   nfs        defaults            0          0
```

◆ 配置 NIS

NIS 是网络信息服务（Network Information Service）的缩写，也称为黄页（Yellow Pages, YP），是一种分布式文件系统的客户端-服务器端目录服务协议，实现集群内部用户和主机名的映射配置。也就是说，一旦在服务器端配置了用户，就相当于在客户端配置了该用户。

1) **NIS 服务器端**用 yum 或 apt-get 安装 NIS 相关软件包 ypserv、yp-tools、ypbind；而 **NIS 客户端**安装 yp-tools 和 ypbind。

2) 修改/etc/sysconfig/network 设置 NIS 的域名：

```
# Created by anaconda
NISDOMAIN=bioinfo
```

3) **服务器端**修改文件/etc/rc.d/rc.local，实现开机自动加入 NIS 域：

```
/bin/nisdomainname bioinfo
```

4) **服务器端**修改文件/etc/ypserv.conf，设置特定的 NIS 服务器访问权限：

```
# Host  :       Domain          :       Map     :       Security
127.0.0.0:*:*:none
192.168.5.0/24:*:*:none
*:*:*:deny
# *     :       *       :       passwd.byname   :       port
# *     :       :       :       passwd.byuid    :       port
```

这里 Host/Domain/Map/Security 的意义分别是：

- ✓ Host：指定客户端，可以是 IP，也可以是网段；
- ✓ Domain：制定 NIS 域名；
- ✓ Map：设置可用数据库，这里用“*”表示任意数据库；
- ✓ Security：安全设置，可以有 none/port/deny 三种
  - ◆ none：没有任何安全限制；
  - ◆ port：只允许 1024 以下的端口访问连接 NIS 服务器
  - ◆ deny：拒绝访问

5) **服务器端**启动相关服务 rpcbind 和 ypserv

```
shell# systemctl start rpcbind
shell# systemctl start ypserv
shell# rpcinfo -p localhost
shell# rpcinfo -u localhost ypserv
```

6) 在**管理节点**添加本地用户

```
shell# useradd bio01
shell# useradd bio02
shell# passwd bio01
shell# passwd bio02
```

7) 在**管理节点**建立 NIS 用户的账户数据库（以后但凡有账户信息改变都必须重新运行）：

```
shell# /usr/lib64/yp/ypinit -m
```

8) 修改**客户端**用户信息、密码、组、主机的认证顺序文件/etc/nsswitch.conf:

```
passwd: files nis
shadow: files nis
group: files nis

hosts: files dns nis
```

表示在搜索账户、密码、组和主机名的信息时，搜索的顺序：

- ✓ files：搜索本地文件；
- ✓ dns：查询 DNS
- ✓ nis：搜索 NIS 数据库

9) 修改**客户端**配置文件/etc/yp.conf：

```
domain bioinfo server mu01
```

10) 修改**客户端**系统认证文件/etc/sysconfig/authconfig，加入：

```
USENIS=yes
```

11) 修改客户端系统认证文件/etc/pam.d/system-auth：

```
password    sufficient    pam_unix.so md5 shadow nis nullok try_first_pass
use_authtok
```

12) 在**客户端**启动服务：

```
shell# systemctl start rpcbind
Shell# systemctl start ypbind
```

13) **客户端**检测是否配置成功：

```
shell# yptest
```

14) 最后要记住用 chkconfig **或** systemctl 将需要启动执行的服务加入启动程序中。

- 配置 SSH，实现集群内部相互访问不需要密码
    1) 用 su 登录到某个用户
    2) 保证安装了 openssh，否则用 yum 或者 apt 安装 openssh；
    3) 运行 ssh-keygen -t rsa 在用户目录的.ssh 目录下生成密钥对文件 id_rsa 以及 id_rsa.pub
    4) 将文件 id_rsa.pub 复制为文件 authorized_keys
    5) 将“StrictHostKeyChecking no” 写入文件.ssh/config
    6) 将目录.ssh 的权限修改为 700
    7) 将文件.ssh/authorized_keys 和 config 的权限修改为 600
- 安装和配置 MPICH
    1) 从 http://www.mpich.org/downloads/下载 mpich-3.2
    2) 解压并安装。

请参考 https://www.asmodeus.cn/archives/393 进行安装和配置。

- 安装和配置 Torque 作业调度和管理系统
    1) 确保管理节点和所有计算节点上的/etc/hosts 已经包含所有的节点主机名以及对应 IP 的信息；
    2) 确保所有的节点上的防火墙和 SELINUX 已经关闭；
    3) 在管理节点上下载和安装 Torque，安装结束后运行 make packages 产生的 torque-package-*.sh 文件复制到所有计算节点，并在这些节点上安装运行；
    4) 计算节点上配置 config 文件，并启动运行 pbs_mom 服务；
    5) 启动管理节点上的 pbs_server、pbs_sched 服务；
    6) 提交任务脚本，查看是否能正常执行。

请参考 https://my.oschina.net/zctzl/blog/1580071 进行安装和配置。

2. MPICH **编程实践**

完成 lab1b-mpich.pdf **中的上机实验内容。**

3. CUDA **编程实验**

完成 lab1c-cuda.pdf **中的上机实验内容。**

四、实验结果

1. 记录安装配置过程中出现的一些问题，并说明你是如何解决的;
2. 将 MPI 程序和对应的 PBS 脚本保存为附录 1 和附录 2，记录计算过程的结果，并用图形分析节点数与计算效率之间的关系，并进行分析。

3. 将 CUDA 程序和 PBS 脚本保存为附录 2，对计算结果进行分析。

五、总结

针对本实验课程进行总结。