

# *Linux 下的 Cluster 实现*



啜立明

---

---



# *Linux 中 实现 HPC 集群技术*



# 实现 *HPC* 集群



# *Linux 下的 HPC-- 高性能计算*

- 传统计算方式 ---- 串行计算

串行计算是指在单个计算机（拥有单个中央处理单元）上执行软件写操作。CPU 逐个使用一系列指令解决问题。

为了加快处理速度，在原有的串行计算的基础上演变出并行计算

# *Linux 下的 HPC-- 高性能计算*

- 并行计算 (Parallel Computing)

指同时使用多种计算机资源解决计算问题的过程。

为执行并行计算，计算资源应包括一台配有多处理器的计算机、一个与网络相连的计算机专有编号，或者两者结合。

---

---

# *Linux 下的 HPC-- 高性能计算*

- 并行计算的主要目的  
用于解决大型且复杂的计算问题
  - 并行计算还可以利用非本地资源，可以使用多台计算机集合在一起共同处理、计算机来取代昂贵的大型计算机或大型服务器
- 
-

# *Linux 下的 HPC-- 高性能计算*

- 并行计算代表— Messages Passing Interface

MPI 本身并非一个软件，它仅仅是一个规格很严密的通讯标准。其主要功能是在处理并行运算之间节点的资料交换。或者说 MPI 属于并行计算语言的函数库

# Linux 下的并行计算 -MPI

- MPI 的主要三个方面

(1)MPI 作为一个库存在，而非一种软件或开发语言

(2)MPI 是一种规范或标准的代表，所有的并行计算机制造商都可提供对 MPI 的支持。

( 3 ) MPI 是一种消息传递编程模型，并成为消息传递编程模型的代表

---

---

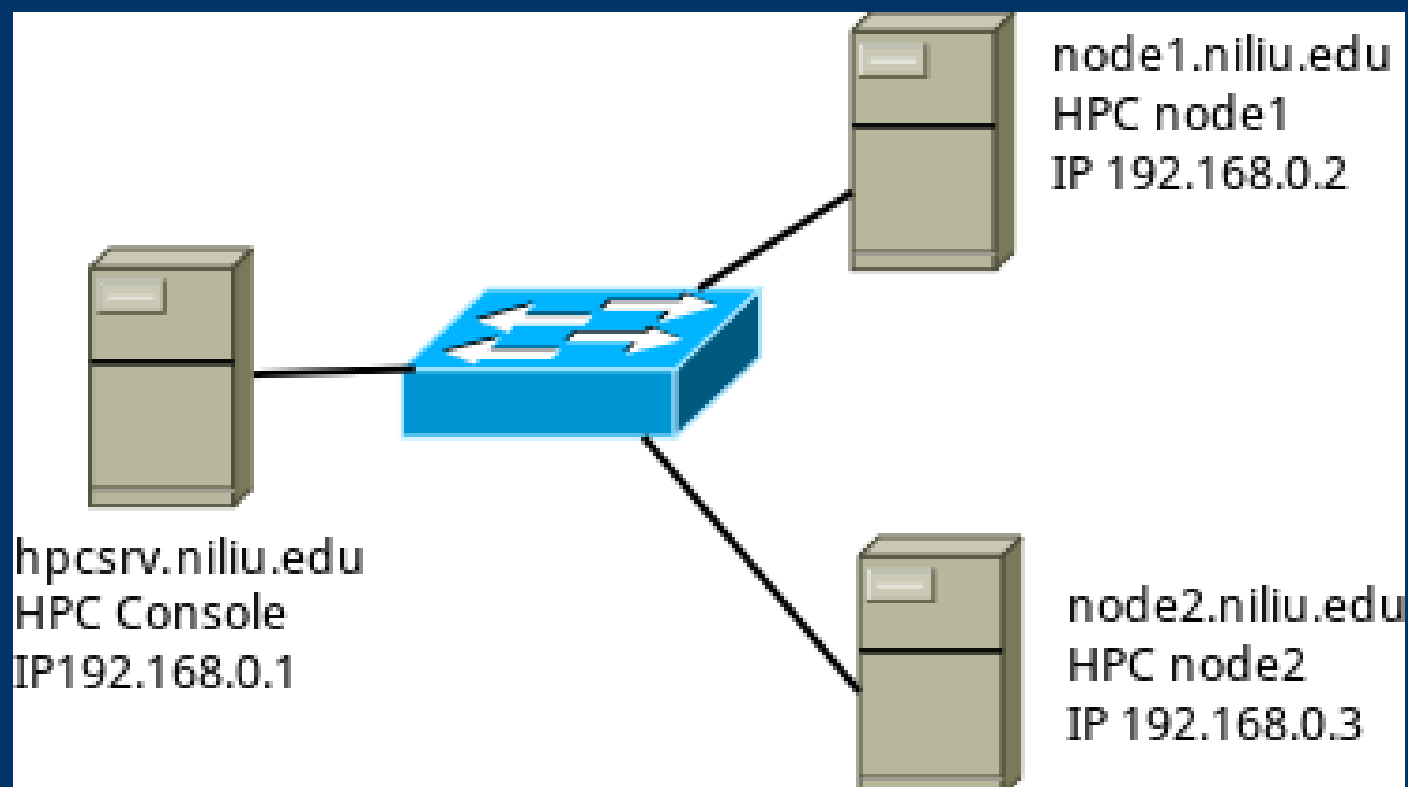


# Linux 下的高性能集群

- 高性能集群
  - 实现高性能集群主要目的是将多台计算机的计算能力合并到一起，使用并行计算软件 (MPICH) 实现并行计算从而实现一个超越单台计算机计算能力的强力系统。
  - 实际上 MPICH 就是符合 MPI 标准通讯协议的一套软件
- 
-

# 实现 *Linux* 下的 HPC

## • 试验拓扑图



# 实现 *Linux* 下的 *HPC*

- ( 1 ) 名称定义

便于 MPICH 配置或调试，对名称节点名称进行定义

```
#vim /etc/hosts
```

```
192.168.0.1 hcpsrv.niliu.edu hpcsrv
```

```
192.168.0.2 node1.niliu.edu node1
```

```
192.168.0.3 node2.niliu.edu node2
```

---

---

# 实现 *Linux* 下的 *HPC*

- ( 2 ) 定义 MPICH 安全通信，用于控制其他节点或与其他节点交互信息时确保安全

hpvsrv:

```
#cd ~
```

```
#ssh-keygen -t rsa // 一路回车，以不使用口令方式连接才能完成 MPICH
```

```
#ssh-copy-id node1
```

```
#ssh-copy-id node2
```

---

---

# 实现 *Linux* 下的 *HPC*

- 测试 ssh , 确认不需要口令就可成功连入

```
#ssh node1
```

```
#ssh node2
```

---

---

# 实现 *Linux* 下的 *HPC*

## (3) 下载 MPICH

//\*CentOS/RHEL 中自带 mpich, 但因为没有测试程序 (涉及到 MPI 开发, 故推荐使用源代码包安装, 其中自带测试程序

```
#wget www.mpich.org/static/downloads/3.1.3/  
mpich-3.1.3.tar.gz
```

---

---

# 实现 *Linux* 下的 *HPC*

(4) 在每个节点上安装编译 MPICH 所需要的工具

```
#yum install gcc gcc-gfortran gcc-c++
```

(5) 在每个节点上解压软件包

```
#tar xvfz mpich-3.1.3.tar.gz
```

(6) 在每个节点上进入 MPICH 目录

```
#cd mpich
```

---

---

# 实现 *Linux* 下的 *HPC*

(7) 在每个节点上编译并安装 mipch

```
#./configure
```

```
#make
```

```
#make install
```

---

---



# 实现 *Linux* 下的 *HPC*

(8) 在每个节点上创建 HPC 所需的主机

```
hpcsrv:  
#cat ~/hpcnode  
hpcsrv  
node1  
node2
```

---

---

# 实现 *Linux* 下的 *HPC*

(8) 在每个节点上创建 HPC 所需的主机

Node1:

```
#cat ~/hpcnode
```

```
hpcsrv
```

```
node1
```

```
node2
```

---

---

# 实现 *Linux* 下的 *HPC*

(8) 在每个节点上创建 HPC 所需的主机

Node2:

```
#cat ~/hpcnode
```

```
hpcsrv
```

```
node1
```

```
node2
```

---

---

# 实现 *Linux* 下的 *HPC*

(9) 在 hpcsrv 端进行测试 MPICH

```
#mpiexec -n 6 -machinefile ~/hpcnode ~/mpic  
h-3.0.4/examples/cpi
```

-n: 处理 6 次

-machinefile: 指定节点文件

---

---

# 实现 *Linux* 下的 *HPC*

- 以下是三台计算机建立 HPC 测试成功截图

```
[root@hpcsrv ~]# mpiexec -n 6 -machinefile ~/hpcnode mpich-3.1.3/examples/cpi
Process 0 of 6 is on node1.niliu.edu
Process 3 of 6 is on node2.niliu.edu
Process 2 of 6 is on node1.niliu.edu
Process 5 of 6 is on node2.niliu.edu
Process 4 of 6 is on node1.niliu.edu
Process 1 of 6 is on node2.niliu.edu
pi is approximately 3.1415926544231243, Error is 0.0000000008333312
wall clock time = 0.112033
```

# HPC 集群实现

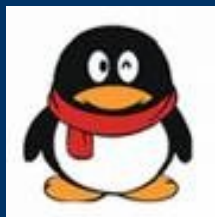
- 使用 MPICH 实现 Linux 下的 HPC 集群
  - 试验目的：掌握 HPC 集群的作用与配置
  - 试验人员：个人
  - 所需要计算机设备：推荐 3 台或更多
  - 试验时间：60 分钟
- 
-

# *Linux 下的 Cluster 实现*

## 结 束



master.chuai@gmail.com



304630723



152990419