



2006/9/28

NCHC

# Queuing System

吳長興

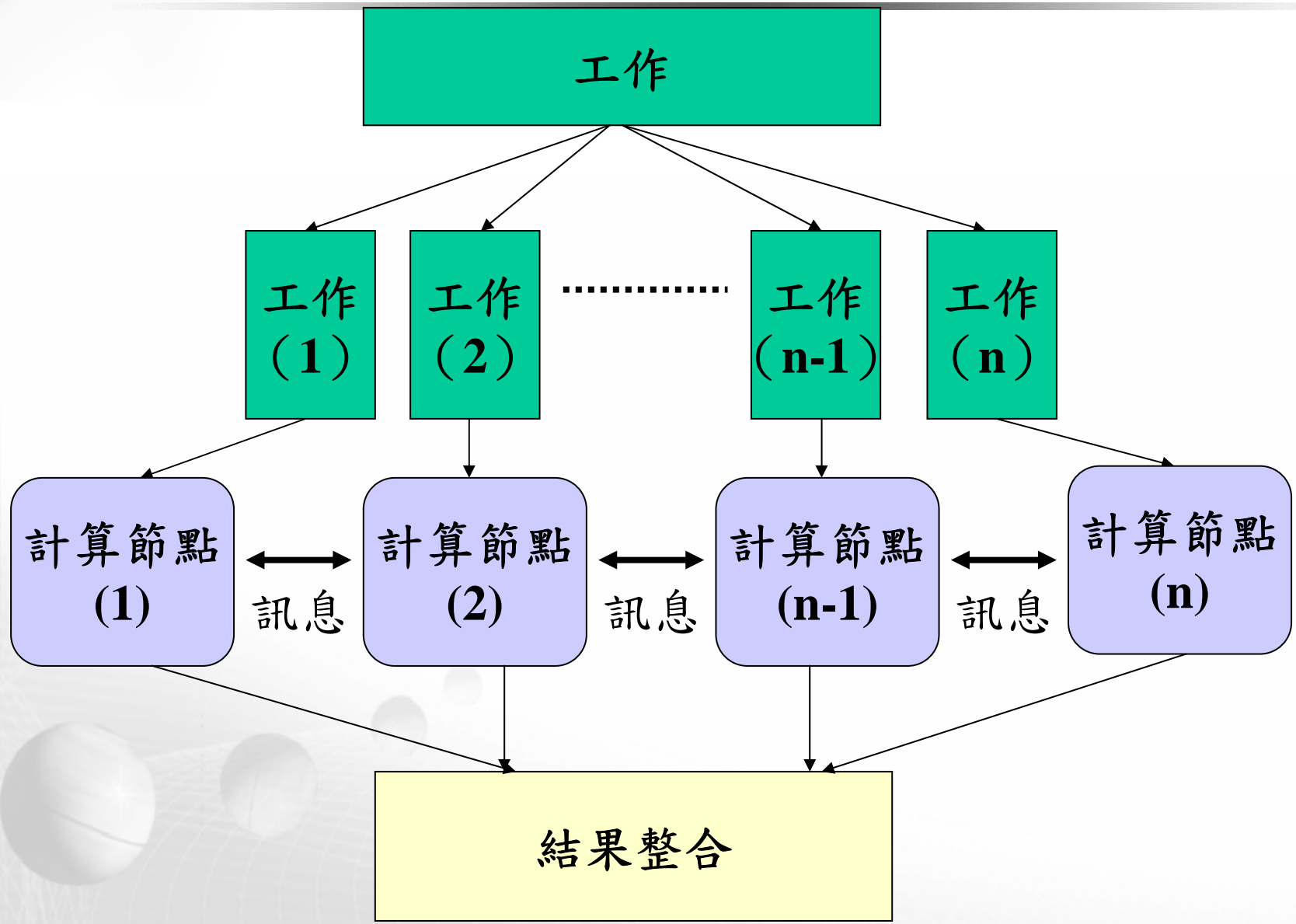
# 大綱

- ◆ Queuing System 應具備的基本功能
- ◆ Queuing System 軟體
- ◆ PBS
- ◆ Cmeta
- ◆ 結論

# 大綱

- ◆ *Queuing System* 應具備的基本功能
- ◆ Queuing System 軟體
- ◆ PBS
- ◆ Cmeta
- ◆ 結論

# 如何使用PC Cluster ?



# 如何使用PC Cluster ?

- ◆ 平行化程式並分散至各計算節點執行
  - 使用mpi的library，來平行工作
  - 透過mpirun來行程式
    - `mpirun -v -machinefile NODEFILE -np 4 /home/hsing/pbs/test/a.out`
  - 依據NODEFILE的內容，挑選第一個計算節點以rsh or ssh方式登入，然後執行mpirun程式，mpirun會依NODEFILE剩下的計算節點，將”其他分割”程式散到其他節點執行。
- ◆ 透過套裝軟體來執行



# 是否需要 Queuing System ?

- ◆ 依照上述觀念，只需要mpich就可以順利使用PC Cluster 為什麼還需要Queuing System 這個東西呢？
  - 規模較大，節點數量多
  - 使用者多
  - 工作內容複雜



# Queuing System 應具備的基本特點

## ◆ 資源：

- 電腦資源能夠被充分使用
- 公平的分配資源

## ◆ 管理：

- 使用者管理
  - 資源(disk、job、time)分配、優先權、權限....
- Cluster管理
  - 切割資源、node 狀態、資源監控、node維護....
- Job的控管
  - 排程變更、平衡負載、錯誤恢復、通知...



# Queuing System 應具備的基本功能

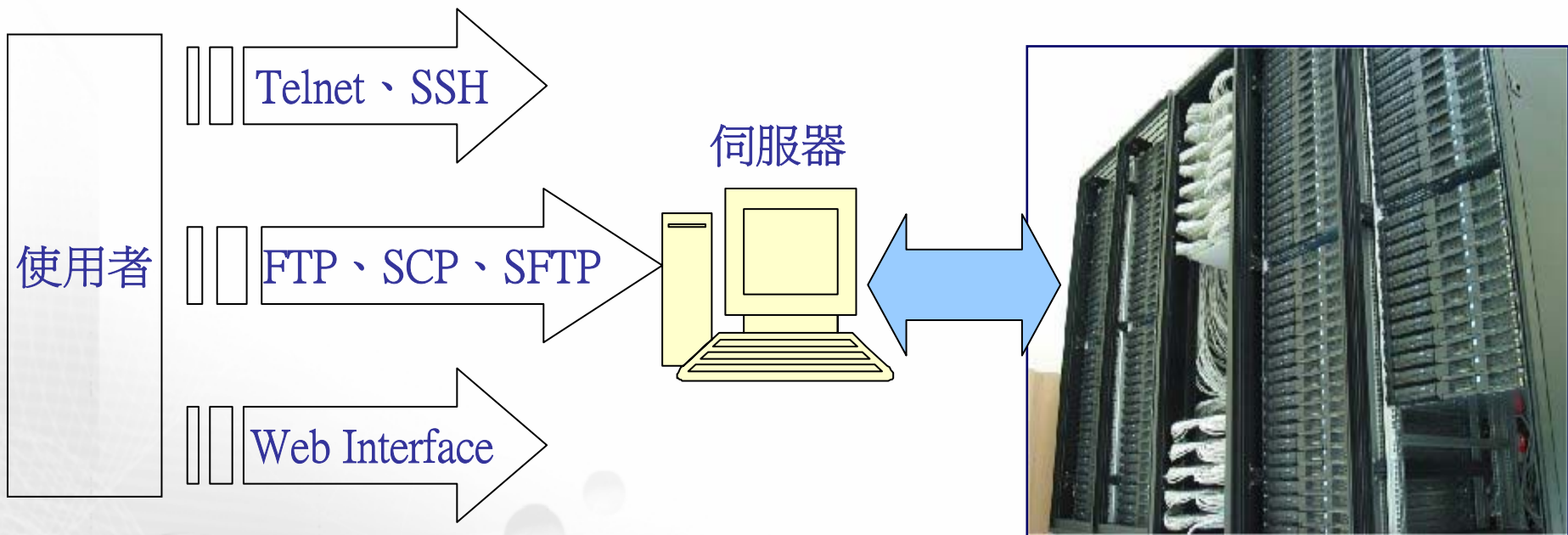
---

## ◆ 使用者：

- Submit job
- Cancel job
- Query job status
- Modify job



# PC Cluster 基本架構



# 基本服務

## ◆ RSH or SSH :

- 登入信任的計算節點不需要密碼 (mpich使用)

## ◆ NFS :

- 共享/home : 讓使用者程式都存在於各計算節點
- 共享/opt : 一些程式共用 (如 pbs、mpich...)

## ◆ NIS or 自行統一帳號 :

- 統一使用者帳號

## ◆ NTP-Server 以及 NTP- Client

- 統一系統時間

## ◆ Firewall :

- 防禦入侵

# PC Cluster 基本架構

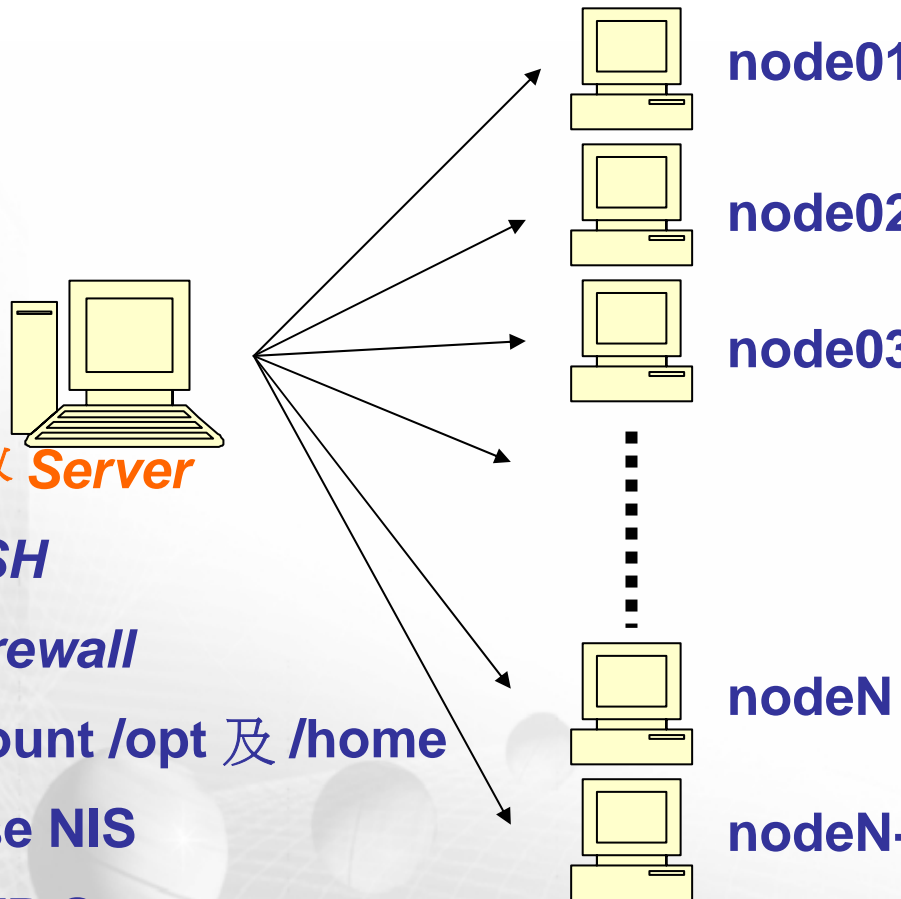
## Node

- rsh, rlogin, rexec  
or ssh, scp, sftp
- Mount /opt 及 /home
- Use NIS
- NTP-Client

## File Server

### NFS Service

- Export /opt 及 /home
- NIS Service



## 對外 Server

- SSH
- Firewall
- Mount /opt 及 /home
- Use NIS
- NTP-Server

# 大綱

- ◆ Queuing System 應具備的基本功能
- ◆ *Queuing System* 軟體
- ◆ PBS
- ◆ Cmeta
- ◆ 結論

# Queuing System 軟體

◆ PBS ( Portable Batch System )



◆ DQS ( Distributed Queuing System )

◆ Cmeta ( Cluster management enabling toolbox for administrators)



◆ Platform LSF **Platform**

◆ .....

# 大綱

- ◆ 為什麼要使用 Queuing System?
- ◆ Queuing System 應具備的基本功能
- ◆ Queuing System 軟體
- ◆ *PBS*
- ◆ Cmeta
- ◆ 結論

# PBS – 組成元件

## ◆ commands

- 以命令列或圖形介面方式，讓透過socket來讓使用者進行提交(submit)、監督 (monitor)、修改 (modify) 和刪除(delete) 工作

## ◆ pbs\_server

- 接收、產生、管理及保護使用者的批次工作

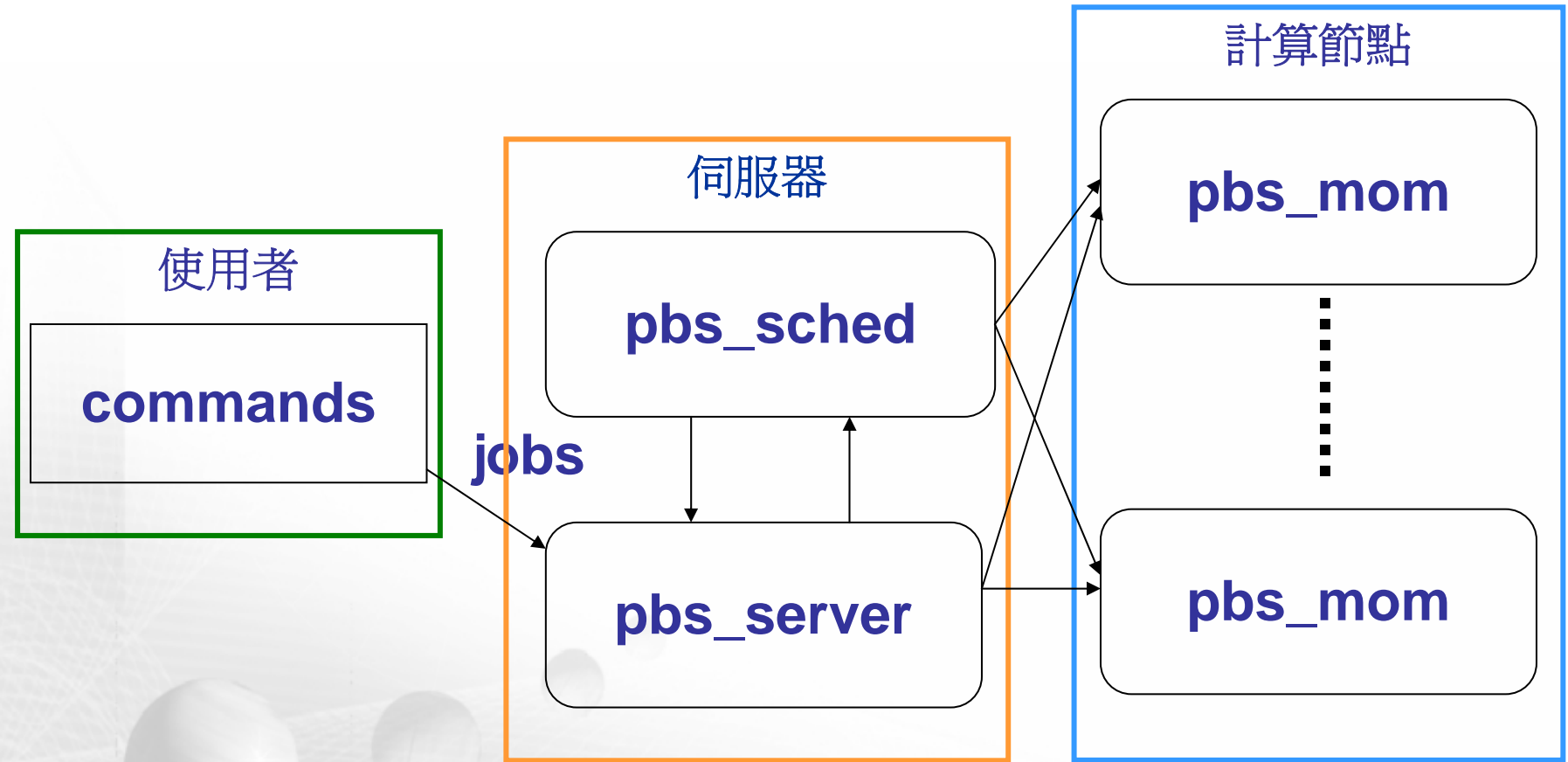
## ◆ pbs\_mom

- 接收pbs\_server 給予的批次工作，並呼叫對應的程式來執行，完成後將結果回報給pbs\_server

## ◆ pbs\_sched

- 負責排程工作、資源分配及節點管理

# PBS – 組成元件







# PBS 安裝 - 下載

## ◆ Open PBS (Open Portable Batch System)

- 網址 <http://www.openpbs.org>

## ◆ Torque (建議使用)

- 網址: <http://www.clusterresources.com/downloads/torque/>

## ◆ PBSWeb (Web介面)

- 網址 : <http://www.cs.ualberta.ca/%7Epinchak/PBSWeb/>
- Lite: <http://seventeen.mit.edu/blog/platin/pbsweb/>

## ◆ PBS PRO (商業化軟體)

- 網址 : <http://www.pbspro.com>



# PBS 安裝 – 環境需求

## ◆ 環境需求

- 確定已啟動 NIS 以及 NFS 服務
- 假如有需要使用圖形介面，則需要安裝 Tcl 8.0 和 Tk 8.0
- 確定使用者可以使用 ssh 到各個計算節點，而不需使用密碼



# PBS 安裝 - 步驟一：編譯設定

## ◆ 下載

- <http://www.clusterresources.com/downloads/torque/torque-2.1.2.tar.gz>

## ◆ 解壓縮

- `tar zxvf torque-2.1.2.tar.gz -C /opt/src/`

## ◆ 設定(執行configuration shell script)

- `./configure --prefix=/opt/torque-2.1.2` (安裝目錄) .....

## ◆ 編譯 (GNU 相容的make 程式)

- `make`

## ◆ 安裝

- `make install`



# PBS 安裝 - 步驟一 : 編譯設定

```
[root@pika150 src]# wget
  http://www.clusterresources.com/downloads/torque/torque-
  2.1.2.tar.gz
[root@pika150 src]# tar zxvf torque-2.1.2.tar.gz
[root@pika150 src]# cd torque-2.1.2
[root@pika150 torque-2.1.2]# ./configure --prefix=/opt/torque-2.1.2
[root@pika150 torque-2.1.2]# make
[root@pika150 torque-2.1.2]# make install
[root@pika150 torque-2.1.2]# ln -s /opt/torque-2.1.2 /opt/torque
```



# PBS 安裝 - 步驟二：設定Node

- ◆ 設定檔預設路徑/var/spool/torque/
- ◆ 增加計算節點(有兩種方式):
  - 直接修改設定檔
    - /var/spool/torque/server\_priv/ 建立一個檔名為nodes的檔案

```
[root@pika150 ~]# vi /var/spool/torque/server_priv/nodes
client01 np=2
client02 np=2
.....
```



# PBS 安裝 - 步驟二：設定Node

- 方法二：利用 qmgr 將 node 加入
  - Qmgr:create node hpcs001
  - Qmgr:create node hpcs002
  - .....
  - Qmgr:active node hpcs001
  - Qmgr:active node hpcs002
  - .....
- 使用方法二時，pbs\_server 及 pbs\_sched 要先常駐



# PBS 安裝 - 步驟三：設定Server名稱

## ◆ 設定Server描述檔

- /var/spool/torque/server\_name

```
[root@pika150 src]# vi /var/spool/torque/server_name  
pika150.nhc.org.tw
```



# PBS 安裝 - 步驟四：設定pbs\_mom

## ◆ 設定pbs\_mom

- /var/spool/torque/mom\_priv/config

```
[root@pika150 src]# vi /var/spool/torque/mom_priv/config
$clienthost pika150
$logevent 255
$restricted *
$usecp pika150.nchc.org.tw:/home /home
```





# PBS 安裝 - 步驟五：啟動服務

- ◆ 初始化並啟動pbs\_server及pbs\_sched
  - /opt/torque/sbin/pbs\_server **-t create**  
(第一次執行，需要建立資料庫)
  - /opt/torque/sbin/pbs\_sched



# PBS 安裝 - 步驟六：新增佇列規則

- ◆ 設定並啟動排程佇列 (自行參考手冊)
  - `/opt/torque/bin/qmgr < /tmp/setup.conf`
- ◆ 啟動設定的兩個佇列 default 及 long(選擇)
  - 假如沒在 Setup.conf 啟用時
    - `/opt/torque/bin/qstart default long`
    - `/opt/torque/bin/qenable default long`



# PBS 安裝 - 步驟七：安裝 Client

- ◆ 從步驟1~6已經完成Server及Client的設定
- ◆ 複製設定檔到各個Client的  
/var/spool/torque/裡面

```
[root@pika150 opt]# tar zcvf torque_config.tar.gz /var/spool/torque/  
[root@pika150 opt]# ssh client01  
[root@client01 opt]# cd /opt  
[root@client01 opt]# tar zxvf torque_config.tar.gz
```

- ◆ 啟動pbs\_mom
  - /opt/torque/sbin/pbs\_mom



# PBS 安裝 - 步驟八：自動啟動服務設定

## ◆ Server端

- 設定pbs\_server開機執行 /etc/rc.d/rc.local
  - 增加 /opt/torque/sbin/pbs\_server
- 設定pbs\_sched開機執行 /etc/rc.d/rc.local
  - 增加 /opt/torque/sbin/pbs\_sched

## ◆ Client端

- 設定pbs\_mom開機執行 /etc/rc.d/rc.local
  - 增加 /opt/torque/sbin/pbs\_mom



# PBS 安裝 - 步驟八：自動啟動服務設定

## ◆ 設定 pbs\_server 開機執行

- 建立 /etc/init.d/pbs\_server (改成可執行)
- cd /etc/rc.d/rc3.d
- ln -s /etc/init.d/pbs\_server S98pbs\_server

## ◆ 設定 pbs\_sched 開機執行

- 建立 /etc/init.d/pbs\_sched (改成可執行)
- cd /etc/rc.d/rc3.d
- ln -s /etc/init.d/pbs\_server S98 pbs\_sched

## ◆ 設定 pbs\_mom 開機執行(Client)

- 建立 /etc/init.d/pbs\_mom (改成可執行)
- cd /etc/rc.d/rc3.d
- ln -s /etc/init.d/pbs\_server S98 pbs\_mom



# PBS 安裝 - 步驟九：檢視是否安裝完成

## ◆ 觀察nodes的狀態

- /opt/torque/bin/pbsnodes -a

## ◆ 結果

client01

state = free

np = 2

ntype = cluster

status = opsys=linux,uname=Linux client01 2.6.11-6mdksmp #1 SMP Tue Mar 22 15:40:42 CET 2005  
i686,sessions=2647,nsessions=1,nusers=1,idletime=1474747,totmem=3969112kb,availmem=3923080kb,physmem=515180kb,ncpus=2,loadave=0.00,netload=2482164563,state=free,rectime=1129879449



# PBS Script 的撰寫

## ◆ PBS Script 規則

- 一個#為PBS指令
- 三個#為註解

## ◆ 循序程式範例

## ◆ 平行程式範例



# PBS-Control Job

- ◆ 遞交工作(submit job)
  - /opt/torque/bin/qsub script
  - 成功後系統會回覆Job ID 的訊息如下：  
336.pika150.nchc.org.tw
- ◆ 刪除工作(delete job)
  - qdel 336
- ◆ 詢問執行狀況(Query job status)
  - qstat



# 大綱

- ◆ 為什麼要使用 Queuing System?
- ◆ Queuing System 應具備的基本功能
- ◆ Queuing System 軟體
- ◆ PBS
- ◆ *Cmeta*
- ◆ 結論



# Cmeta-下載及環境準備

## ◆ 註冊下載

- <http://pccluster.nchc.org.tw/FileDownload/>

## ◆ 環境準備

- Perl 完整安裝
- Perl 的模組 Crypt-PasswordMD5-1.2
- perl-suidperl 的檔案
- 確定已啟動 NIS 以及 NFS 服務
- Client 端開啟 RSH daemon
- 允許 root 權限 rsh 到後端 Client



# Cmeta-安裝設定

## ◆ 解壓縮

- `tar zxvf Cmeta-1.0-release.tar.gz`

## ◆ 修改 [install.in](#)

## ◆ 安裝

- `./install -install`

## ◆ 反安裝

- `./install -uninstall`

## ◆ 把index.htm移到Web的根目錄，並修改index.htm連結到cgi的路徑



# Cmeta-啟動

## ◆ 啟動

- `cd /opt/cmeta/bin/`
- `./pccmopt.sh start`

## ◆ 停止

- `cd /opt/cmeta/bin/`
- `./pccmopt.sh stop`

## ◆ 設定開機執行 `/etc/rc.d/rc.local`

- `/opt/cmeta/bin/pccmopt.sh start`



# Cmeta-使用說明

PC Cluster System - Microsoft Internet Explorer

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

← 上一頁 · → 下一頁 · 搜尋 · 我的最愛 · 保羅 · 網路 · 列印 · 移至

網址(A) http://hpcserv2.nchc.org.tw/cmeta/

Google 搜尋網頁 查詢網址 PageRank 網頁詳情 高階 說明

## Cmeta:

### Cluster management enabling toolbox for administrators

Version 1.0.May 2003

### NCHC PC Cluster Project

National Center for High-Performance Computing, Taiwan

<http://hpcserv2.nchc.gov.tw/project>  
<http://pccluster.nchc.gov.tw>

username:

password:

[user's guide](#)

網路網路



# Cmeta-Menu

## Operation

[Submit Job](#)

[Monitor Jobs](#)

[Tune Job Queue](#)

## System State

[Cluster State](#)

[Node State](#)

[Job State](#)

## Cluster Configuration

[Node](#)

[User&Policy](#)

[Time&Job](#)

[Priority&Policy](#)

[Tune Node Equivalence](#)

[Tune User Priority](#)

GLOBAL ▾

del Cluster

new Cluster

[Submit Job](#)

[Monitor Jobs](#)



# Cmeta-Submit Job

user <i>hsing</i> login	<a href="#">See Node State</a> <a href="#">See Job State</a>
Description file	<input type="button" value="Load"/> <input type="button" value="Delete"/>
Save file	<input type="text"/> <input type="button" value="Save"/>
Job name*	<input type="text"/>
Resubmit if job or node crash happens (System will resubmit the job automatically)	<input type="button" value="OFF"/>
Target cluster	<input type="button" value="Test"/> Upper Bound Nodes: <input type="text" value="0"/> Available Nodes Now: <input type="text" value="0"/>
Node number required by the job	<input type="button"/>
Job type	(SEQ:Sequential program) <input type="button" value="SEQ"/>
SuperJob enabled	<input type="button" value="OFF"/>
Specify <b>executable</b> files <input type="text"/> <input type="button" value="select"/>	<input type="button" value="submit"/>
Specify <b>input</b> files <input type="text"/>	
Dependency list <input type="text"/>	<input type="button" value="set dependency"/>
Working directory	<input type="text"/>
STDOUT directory	<input type="text"/>
STDERR directory	<input type="text"/>
E-mail address	<input type="text"/>
When to send E-mail	begin <input type="checkbox"/> end <input type="checkbox"/> suspend <input type="checkbox"/> restart <input type="checkbox"/>



# Cmeta-Monitor Jobs

job alias	job id	program name	state	none	delete	suspend	wakeup	Processor number	Submission time	Query SuperJob
seqjob1	2	javac	W	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1	2003/9/16 Tue 16:49:06	
seqjob2	3	aaa	W	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1	2003/9/16 Tue 16:49:24	
mpijob1	4	mpicc	W	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1	2003/9/16 Tue 16:49:43	
mpijob2	5	mpicc	W	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1	2003/9/16 Tue 16:49:56	<input type="button" value="subjob"/>
mpijob3	6	mpicc	W	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1	2003/9/16 Tue 16:50:11	







# Cmeta-Cluster State

cluster1	
Policy : <b>PRIORTIY</b>	
Max Time/Job : <b>2880</b>	
Max Job/User : <b>4</b>	
Nodes	
Node Name	Max Job Number
<b>hpcs004</b>	<b>1</b>
<b>hpcs008</b>	<b>4</b>
<b>hpcs009</b>	<b>1</b>
<b>hpcs011</b>	<b>1</b>
<b>hpcs012</b>	<b>2</b>
<b>hpcs013</b>	<b>2</b>



# Cmeta-Node state

Cluster Name: <b>cluster1</b>	
Host Name	State
hpcs004	Alive
hpcs008	Alive
hpcs009	Alive
hpcs011	Alive
hpcs012	Alive
hpcs013	Alive



# Cmeta-Job State

Cluster Name: GLOBAL			
Host Name	User	Job Name	Start Time

Cluster Name: cluster1			
Host Name	User	Job Name	Start Time
hpcs008	Iven	mpitest1	2003/3/30 Sun 08:03:06
hpcs008	Iven	mpitest2	2003/3/30 Sun 08:03:09
hpcs008	Iven	mpitest2	2003/3/30 Sun 08:03:20
hpcs008	Iven	mpitest2	2003/3/30 Sun 08:03:20



# Cmeta-增加Node

node list  
name:#processor

- hpcs001 : 2 ( )
- hpcs002 : 2 ( )
- hpcs003 : 2 ( )
- hpcs004 : 2 (L)
- hpcs005 : 2 ( )
- hpcs006 : 2 ( )
- hpcs007 : 2 (L)
- hpcs008 : 2 ( )
- hpcs009 : 2 ( )
- hpcs010 : 2 ( )

node name:

#job:

# 大綱

- ◆ 為什麼要使用 Queuing System?
- ◆ Queuing System 應具備的基本功能
- ◆ Queuing System 軟體
- ◆ PBS
- ◆ Cmeta
- ◆ 結論

# 參考資料

## ◆ NCHC PC Cluster 論壇

- <http://pccluster.nchc.org.tw/>

## ◆ MPICH

- <http://www-unix.mcs.anl.gov/mpi/mpich/>

## ◆ OpenPBS

- <http://www.openpbs.org/>

## ◆ PBSWeb

- <http://www.cs.ualberta.ca/~pinchak/PBSWeb/>

## ◆ PBS Quick Reference Guide

- <http://www.pcf.sinica.edu.tw/sw/pbs/>

## ◆ .....

# 結論

- ◆ Open PBS
  - 安裝使用
- ◆ Cmeta
  - 安裝使用



---

**Thank you !!**

A decorative graphic in the bottom-left corner consists of four white, glossy spheres of varying sizes arranged in a receding line. The background of this area is a light gray grid pattern that recedes into the distance, creating a sense of depth.





# /etc/hosts 檔案內容

```
192.168.1.1    pika150.nchc.org.tw    pika150
```

```
192.168.1.11  client01
```

```
192.168.1.12  client02
```

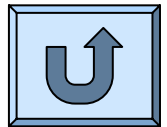
```
.....
```



# Setup.conf 檔案內容

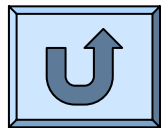
```
# Create and define queue long
create queue long
set queue long queue_type = Execution
set queue long Priority = 60
set queue long max_running = 10
set queue long resources_max.cput = 12:00:00
set queue long resources_min.cput = 02:00:01
set queue long resources_default.cput = 12:00:00
set queue long enabled = True
set queue long started = True
```

```
# Create and define queue default
create queue default
set queue default queue_type = Route
set queue default max_running = 10
set queue default route_destinations = long
set queue default enabled = True
set queue default started = True
```



# Setup.conf 檔案內容(Cont.)

```
# Set server attributes.  
set server scheduling = True  
set server max_user_run = 6  
set server acl_host_enable = True  
set server acl_hosts = *  
set server default_queue = default  
set server log_events = 63  
set server mail_from = hsing@nchc.org.tw  
set server query_other_jobs = True  
set server resources_default.cput = 01:00:00  
set server resources_default.neednodes = 1  
set server resources_default.nodect = 1  
set server resources_default.nodes = 1  
set server scheduler_iteration = 60  
set server default_node = 1 #shared
```



### Job 名稱

#PBS -N mytest

### 輸出檔案

#PBS -e mytest.err

#PBS -o mytest.log

### 放到哪一個佇列裡 (default, long)

#PBS -q default

###=====

# 顯示目錄及時間資訊

echo Working directory is \$PBS\_O\_WORKDIR

cd \$PBS\_O\_WORKDIR

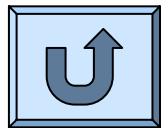
echo Running on host `hostname`

echo Time is `date`

echo Directory is `pwd`

# 執行檔案

/home/hsing/pbs/test/a.out





```
###....(略)
```

```
### 使用node數
```

```
#PBS -l nodes=4
```

```
#顯示目錄及時間資訊
```

```
echo Working directory is $PBS_O_WORKDIR
```

```
cd $PBS_O_WORKDIR
```

```
echo Running on host `hostname`
```

```
echo Time is `date`
```

```
echo Directory is `pwd`
```

```
echo This jobs runs on the following processors:
```

```
echo `cat $PBS_NODEFILE`
```

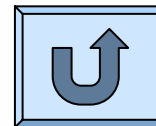
```
# 計算使用的node數
```

```
NPROCS=`wc -l < $PBS_NODEFILE`
```

```
echo This job has allocated $NPROCS nodes
```

```
# 執行 MPI 平行程式 "a.out"
```

```
mpirun -v -machinefile $PBS_NODEFILE -np $NPROCS \  
/home/hsing/pbs/test/a.out
```



# Install.in

#設定cmeta軟體安裝的位置

**PACKAGEPATH=/opt/cmeta**

#設定Server 端的IP

**PCCMIP=192.168.1.1**

#設定介面存放的目錄(必須確定可以執行cgi的程式)

**INTERFACEPATH=/var/www/cgi-bin/Cmeta**

#設定Cmeta的管理者

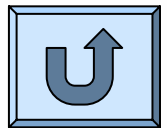
**PCCMACCOUNT=pccmAdm**

#設定rsh程式的位置

**RSHPATH=/usr/bin/rsh**

#設定MPI程式路徑

**MPIRUNPATH=/opt/mpich/bin/mpirun**





# NIS Service (Server)

◆ 相關套件：yp-tools ypbind ypserv portmap

1. 啟動 time 與 time-udp 這兩個預先要啟動的 daemon

```
[root @server root]# chkconfig --level 35 time on  
[root @server root]# chkconfig --level 35 time-udp on  
[root @server root]# /etc/rc.d/init.d/xinetd restart
```

2. 建立 NIS 的領域名稱 (我這裡是設定為 cluster)：

```
[root @server root]# nisdomainname cluster  
[root @server root]# echo "/bin/nisdomainname cluster" >> /etc/rc.d/rc.local  
[root @server root]# echo "NISDOMAIN=cluster" >> /etc/sysconfig/network
```

3. 建立 NIS 設定檔：

```
[root @server root]# vi /etc/ypserv.conf (在這個檔案內增加三行即可)  
127.0.0.0/255.255.255.0 : * : * : none  
192.168.10.0/255.255.255.0: * : * : none  
* : * : * : deny  
[root @server root]# touch /etc/netgroup
```

4. 啟動 NIS：

```
[root @server root]# /etc/rc.d/init.d/ypserv start  
[root @server root]# /etc/rc.d/init.d/yppasswdd start
```

5. 製作資料庫：(每次有更動使用者資訊時，就必須要進行這個步驟！)

```
[root @server root]# /usr/lib/yp/ypinit -m
```

6. 開啟 ypserv 為系統服務

```
[root @server root]# chkconfig --level 35 ypserv on  
[root @server root]# chkconfig --level 35 yppasswdd on
```



# NIS Service (Client)

- ◆ 準備好以下兩個套件：ypbind 及 yp-tools  
按下列步驟設定：

## 1. 建立 NIS 的領域名稱 (我這裡是設定為 cluster)：

```
[root @node1 root]# nisdomainname cluster  
[root @node1 root]# echo "/bin/nisdomainname cluster" >> /etc/rc.d/rc.local  
[root @node1 root]# echo "NISDOMAIN=cluster" >> /etc/sysconfig/network
```

## 2. 建立 NIS 查尋的主機名稱：

```
[root @node1 root]# vi /etc/yp.conf  
domain cluster  
ypserver server.cluster
```

## 3. 修改密碼驗證方式：

```
[root @node1 root]# vi /etc/passwd (在這個檔案的最底下新增如下一行)  
+::::::  
[root @node1 root]# vi /etc/nsswitch.conf  
passwd: files nis nisplus  
shadow: files nis nisplus  
group: files nis nisplus  
hosts: files nis dns
```

## 4. 啟動 NIS：

```
[root @server root]# /etc/rc.d/init.d/ypbind start  
[root @server root]# chkconfig --level 35 ypbind on
```





# RSH Service

## ◆ 1. 要使用 rsh 功能先要安裝：

rsh-server-0.17-12mdk.i586.rpm (Server 端)

rsh-0.17-12mdk.i586.rpm (Client 端)

## 2. 啟動 RSH 囉！

```
[root @node1 root]# chkconfig --level 35 rsh on
```

```
[root @node1 root]# chkconfig --level 35 rlogin on
```

```
[root @node1 root]# chkconfig --level 35 rexec on
```

```
[root @node1 root]# /etc/rc.d/init.d/xinetd restart
```

## 3. 編輯可使用 R command 的主機設定檔：

```
[root @node1 root]# vi /etc/hosts.equiv
```

```
LinuxServer
```

```
LinuxClient01
```

```
LinuxClient02
```

## 4. 編輯家目錄下的.rhosts (內容同 hosts.equiv)

```
vi /home/test/.rhosts
```



# NFS Service

◆ 1. 啟動 portmap 並且設定開機啟動：

```
[root @server root]# /etc/rc.d/init.d/portmap start  
[root @server root]# chkconfig --level 35 portmap on
```

2. 設定 NFS 分享出去：

```
[root @server root]# vi /etc/exports  
/home 192.168.10.0/24(rw,async,no_root_squash)  
/opt 192.168.10.0/24(rw,async,no_root_squash)  
[root @server root]# exportfs -rv  
[root @server root]# /etc/rc.d/init.d/nfs start  
[root @server root]# chkconfig --level 35 nfs on
```

3. 掛載並將指令寫入 /etc/rc.d/rc.local 當中

```
[root @server root]# echo "mount -t nfs Server.cluster:/home /home"  
>> /etc/rc.d/rc.local  
[root @server root]# echo "mount -t nfs Server.cluster:/opt /opt" >>  
/etc/rc.d/rc.local
```



# 讓root 可以RSH 到某台機器

◆ 修改/etc/securetty 將

```
echo "rlogin" >> /etc/securetty
```

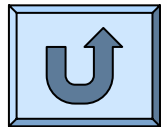
```
echo "rsh" >> /etc/securetty
```

```
echo "rexec" >> /etc/securetty
```



# /etc/init.d/pbs\_server

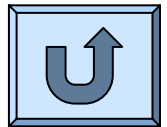
```
#!/bin/sh
FLAGS="defaults 90"
test -f /opt/torque/sbin/pbs_server || exit 0
case $1 in
'start')
    /opt/torque/sbin/pbs_server -a true
;;
'stop')
    pid=`/bin/ps -e | grep 'pbs_server' | sed -e 's/^ *//' -e 's/ .*//'^
    if [ "$pid" != "" ]; then kill -9 $pid; fi
;;
'restart')
    $0 stop
    $0 start
;;
*)
    echo "Usage: /etc/init.d/pbs_server { start|stop|restart}"
    exit 1
;;
esac
exit 0
```





# /etc/init.d/pbs\_sched

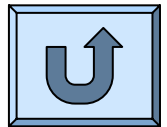
```
#!/bin/sh
FLAGS="defaults 90"
test -f /opt/torque/sbin/pbs_sched || exit 0
case $1 in
'start')
    /opt/torque/sbin/pbs_sched
;;
'stop')
    pid=`/bin/ps -e | grep 'pbs_sched' | sed -e 's/^ *//' -e 's/ .*//'^
    if [ "$pid" != "" ]; then kill -9 $pid; fi
;;
'restart')
    $0 stop
    $0 start
;;
*)
    echo "Usage: /etc/init.d/pbs_sched {start|stop|restart}"
    exit 1
;;
esac
exit 0
```





# /etc/init.d/pbs\_mom

```
#!/bin/sh
FLAGS="defaults 90"
test -f /opt/torque/sbin/pbs_mom || exit 0
case $1 in
'start')
/opt/torque/sbin/pbs_mom
;;
'stop')
pid=`/bin/ps -e | grep 'pbs_mom' | sed -e 's/^ *//' -e 's/ .*//'^
if [ "$pid" != "" ]; then kill -9 $pid; fi
;;
'restart')
$0 stop
$0 start
;;
*)
echo "Usage: /etc/init.d/pbs_mom {start|stop|restart}"
exit 1
;;
esac
exit 0
```





# SSH 免密碼登入設定

## ◆ Server端

- vi /etc/ssh/sshd\_config

.....

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

## ◆ Client端

- ssh-keygen -t rsa

然後按三下Enter

即可在~/.ssh/ 下找到

~/.ssh/ id\_rsa (Private Key)

~/.ssh/id\_rsa.pub (Public Key)

- 將id\_rsa.pub (Public Key)複製到Server端家目錄裡的.ssh裡面，並將名稱改成authorized\_keys