

MPIT00L 使用說明 V1.0

(MPICH2)

by

Po-Jen, Hsu

Last updated:

2007/02/08

mpitool 為 MPICH2 的工具程式, 為了讓使用 MPICH2 及更改設定更方便而發展的, 所使用語言為 Shell Script, 因此不需要 compile, 沒有 32/64bit 限制, 只要是 unix 環境皆可使用, 不需另外拷貝任何檔案, 所有所需檔案均由 mpitool 自行產生, 並且自動判斷檔案是否存在, 而一般使用者僅需注意並修改 mpd_job 檔案即可, 不需要繁鎖的指令步驟, 只要是符合類似以 config.out 作參數修改, 並有一主程式 (a.out) 的任何 mpi 程式均可適用

mpitool 之使用分兩者:

如果您想了解該程式運作原理, 請參考管理者篇, 如您是一般使用者, 請直接跳到使用者篇

管理者篇:

mpitool(附錄會附上原始程式碼以防不時之需)通常是 copy 到 /usr/bin 目錄下, 因此任何使用者均可直接執行, 其中兩個最重要的參數檔為 mpd.hosts 及 mpd.cpu, 兩者均位於 /home/ 目錄下而非一般使用者目錄, 因此請以 root 產生此兩檔案, 並修改其權限(chmod 777 mpd.*)使其能為一般使用者讀取

mpich2 主程式則亦在 /home/ 下, 如執行 mpich2 出現找不到檔案, 請試著將 mpich2 拷貝至一般使用者家目錄下, 並配合修改 .bashrc(bash shell)

mpd.hosts 記載目前所有可以跑的 host 資訊, 請注意如果 host 有增減, 請記得刪除或新增此檔案內容, 此檔案不接受 # 等註解符號, 因為 mpitool 會計算行數來判斷 host 數目, # 註解符號會造成誤判並發生錯誤

mpd.hosts 範例:

```
cp64a:2 ifhn=192.168.1.1
cp64b:2 ifhn=192.168.1.2
cp64c:2 ifhn=192.168.1.3
cp64d:2 ifhn=192.168.1.4
cp64e:2 ifhn=192.168.1.5
cp64f:2 ifhn=192.168.1.6
```

ifhn 表示 interface hostname 必需要加入此段才可使 mpich2 機動性選擇執行程式的 host 分配

mpd.cpu 記載每個 host 的 cpu 數目

mpd.cpu 範例:

2

一般使用者毋需修改上述檔案, 僅在新增或刪除機器時, 管理者必需確保此檔案所有 host 均存在且可運作

使用者篇:

直接在任何目錄下執行 mpitool(不需加./), 此動作之前不需拷貝或新增任何檔案, 也不跟舊有的執行方式有衝突, mpitool 的運作原理是讀取執行目錄下的 mpd_job 檔案, 以 mpd_job 檔取代 mpd.hosts, 因此, 原本需要修改使用者家目錄下的 mpd.hosts 檔, 如今可完全捨棄不用. 每個目錄可以有不同設定的 mpd_job, 因此每個目錄皆可跑出完全不同的 mpi 組態

mpd_job 紀錄在此執行目錄下的 mpi 程式將如何分配資源, 有點跟 mpd.hosts 相像, 但是不同處在 hostname 冒號後的數字代表意義不同, 例如其內容為:

```
cp64a:0 ifhn=192.168.1.1
cp64b:1 ifhn=192.168.1.2
cp64c:2 ifhn=192.168.1.3
cp64d:4 ifhn=192.168.1.4
cp64e:2 ifhn=192.168.1.5
cp64f:5 ifhn=192.168.1.6
```

則 mpi 程式的執行方式為

cp64a 沒有任何程式在上面跑

cp64b: 只跑一個 node

cp64c: 跑兩個 jobs

cp64d: 跑四個 jobs 注意, 雖然可能 cp64d 上只有兩個 cpu, 但是實際上跑四個 jobs 是可以允許的, 只是以效益來說, 並不建議!

cp64e: 跑兩個 jobs

cp64f: 跑五個

諸如此類, hostname 冒號後的數字可以指定該 job 要如何分配, 其餘皆保留不更動即可

mpd.hosts 與 mpd_job 檔案最大的不同是:

mpd.hosts 是宣告所有可以跑的 hosts 以及 cpu 數目, 及其通訊介面 (ifhn)

mpd_job 是宣告每個 host 要丟幾個 job 上去(0~99 或更多)

mpd.hosts 冒號後的數字是固定的, 也就是 cpu 數目, mpd_job 冒號後的數字則是可自訂, 並沒有限制, 如想要測試一個 cpu 跑 100 個 jobs, 可設定如 cp64a:100 ifhn=192.168.1.1

不過以效益來說, 宣告 mpd_job 大於實際 cpu 數目並沒有好處, 因此建議設定成小於或等於實際 cpu 數(≤ 2)

掌握 mpd_job 的用法即可使用 mpitool 分配 jobs, 實際操作時, mpitool 會先搜尋執行目錄, 若該目錄下沒有 mpd_job 的檔案, 則會詢問您是否要產生 mpd_job, 若回答 y 則程式會自動拷貝 /home/mpd.hosts 的內容到 mpd_job (因為 mpd.hosts 的內容與 mpd_job 格式是一樣的), 但請注意此 mpd_job 檔內容就變成:

```
cp64a:2 ifhn=192.168.1.1
```

```
cp64b:2 ifhn=192.168.1.2
```

```
cp64c:2 ifhn=192.168.1.3
```

```
cp64d:2 ifhn=192.168.1.4
```

```
cp64e:2 ifhn=192.168.1.5
```

```
cp64f:2 ifhn=192.168.1.6
```

也就是每個 hosts 全跑, 因此程式隨後會再詢問是否要修改, 打 y 即可自動進入 vi 修改 mpd_job, 請將其改成符合您需求的組態

mpich2 的所有指令均已經加入 mpitool 裡, 以功能表形式呈現並省略麻煩的步驟, 因此像 mpdallexit, mpdboot 等指令均已包含在內, 毋須在記憶冗長的執行順序

使用 step by step:

請執行並配合對照, 此步驟僅會修改或產生 mpd_job 這個檔, 對於其他主程式的參數檔並不會更動

1. 請先切換到想執行程式之目錄下(如 cul4..), **此步驟一定要執行!**

2. 確定設定檔(config.out)及主程式(a.out 或其他)是否存在, 並記下檔名(mpitool 預設設定檔檔名為 config.out, 執行檔為 a.out, 若搜尋不到則會自動要求輸入正確檔名)

3. 打 mpitool 並<Enter>

=====

MPICH Tool V1.0

This program is written by Po-Jen Hsu

Last updated 2007/02/06

=====

```
#-----  
#Number of available host machine= 6  
#Number of processors per host= 2  
#-----
```

Warning!! mpd_job does not exist !通常第一次執行, 不會有 mpd_job 這個檔案, 因此 mpitool 會提出建立 mpd_job 檔的要求
Please edit mpd_job first

ex:

cp64a:2 ifhn=192.168.1.1

cp64c:1 ifhn=192.168.1.3

cp64f:2 ifhn=192.168.1.6

..etc.

For this case, the job you submitted will be executed on cp64a
with 2 nodes

1 node on cp64c and 2 nodes on cp64f

Do you want to make a default mpd_job file?<y/n> !選 y 會拷貝 mpd.hosts 的內容並產生 mpd_job 檔, 選 n 則不會產生 mpd_job 並跳到主目錄, 注意!!沒有 mpd_job 是無法用 mpitool 執行 mpi 程式的, 只能用傳統方法, 即手動打 mpdrun -n 5 /home/sklai/cul4/cul4.out 之類來執行

: y

Although the default mpd_job file is generated,
it should be modified to fit your purpose

Press <Enter> to edit mpd_job by vi !選 y 後 mpitool 會再要求修改剛剛產生的 mpd_job, 請依據您想分配 node 的方式修改

hostname 冒號後的數字,再退出 vi(使用方法跟 vi 一模一樣)

Your mpd_job is successfully modified: !退出後,秀出 mpd_job 已經修改成功,下面則是目錄所在,以及 mpd.hosts 的內容(不需動任何 mpd.hosts 的內容)

```
#Current directory is /home/xanadu
#-----
#The available host machine are:
cp64a:2 ifhn=192.168.1.1
cp64b:2 ifhn=192.168.1.2
cp64c:2 ifhn=192.168.1.3
cp64d:2 ifhn=192.168.1.4
cp64e:2 ifhn=192.168.1.5
cp64f:2 ifhn=192.168.1.6
#-----
```

Press <Enter> to continue

#This shows how your job be submitted to each node !接下來秀出 mpd_job 的內容,請再確認一下

```
cp64a:1 ifhn=192.168.1.1
cp64b:2 ifhn=192.168.1.2
cp64c:0 ifhn=192.168.1.3
cp64d:2 ifhn=192.168.1.4
cp64e:0 ifhn=192.168.1.5
cp64f:0 ifhn=192.168.1.6
#-----
```

<<MPICH2 TOOL MENU>>: !mpitool 的主目錄

0. Run config.out !直接執行 config.out,若找不到則會要求輸入正確檔名

1. Start a mpd job !跑 mpi 程式,mpd_job 檔必需存在才可使用此選項

2. Clear all mpd jobs and restart a mpd environment !重設 mpich 環境,也就是 mpdallexit 以及 mpdboot 的指令,注意,此指令只有在沒有任何 mpd 常駐的時候(如重開機或 mpdallexit 後)才需執行,執行過

後 mpd 就會一直長駐, 不需再點選這個項目, 此選項亦會刪除所有正在執行的 mpd jobs, 因此請小心使用

3. Edit mpd_job file !mpitool 在這裡提供編輯 mpd_job 的選項

4. List all mpd jobs !秀出所有在跑的程式清單以及 cpu 使用情況, 可利用此訊息單獨刪除某 job

5. Quit

:

子選項之解說:

0. Run config.out

執行 config.out 若找不到, 則會秀出:

config.out does not exist, please indicate your config file name

:請輸入正確檔名

1. Start a mpd job

最常用的選項之一, 執行 mpd 的程式

Please make sure the mpd_job file in you executable directory is correct

cp64a:1 ifhn=192.168.1.1 !再秀一次 mpd_job 的設定提醒您

cp64b:0 ifhn=192.168.1.2

cp64c:2 ifhn=192.168.1.3

cp64d:2 ifhn=192.168.1.4

cp64e:0 ifhn=192.168.1.5

cp64f:0 ifhn=192.168.1.6

Please input the number of processes.

This value should not be greater than the total number of nodes in mpd_job

:5

!如果 mpd_job 的設定如上, 則此處應為 $1+0+2+2+0+0=5$, 因此所輸入的數字必需小於或等於 5(如輸入 4, 則 cp64d 會有一個 node 沒跑到, 這是 mpich2 可以允許的, 但是若輸入超過總合, 如 6, 則會出現錯誤, 因為多出一個不知道要丟那個 node)

Please input the full name of the executable file, ex:a.out

:a.out

!輸入主程式檔名, 如 a.out 或 Ag13Cu25 等...

Please input a short alias of this job<ex:Ag13Cu25>
Or enter none for no alias<Please don't skip this procedure>
:

:Ag13Cu25 !可以區分 job, 同樣都是 a. out, 但是 job
alias 卻可不同, 請勿使用空白, 以底線表示; 若輸入 none 則 Job
alias 為空的

Job alias=Ag13Cu25 !顯示 jobalias 內容

1. Submit job in foreground

2. Submit job in background

:2

!可選擇跑前景或跑背景, 執行完畢自動跳出 mpitool

2. Clear all mpd jobs and restart a mpd environment

**If there is no mpd daemon running, it will show some error message. This is normal

This will stop all running jobs

Are you sure?<y/n>

:

!Error message 只有在已經執行過 mpdallexit 才會出現, 這是正常的可不予理會, 如下:

mpdallexit: cannot connect to local mpd

(/tmp/mpd2.console_xanadu); possible causes:

1. no mpd is running on this host

2. an mpd is running but was started without a "console" (-n option)

In case 1, you can start an mpd on this host with:

mpd &

and you will be able to run jobs just on this host.

For more details on starting mpds on a set of hosts, see the MPICH2 Installation Guide.

如無錯誤訊息, 則會跳至以下:

Available hosts:

cp64a

cp64b

cp64c

cp64e

cp64d
cp64f

Hostname_PID# IP
cp64a_29481 (192.168.1.1)
cp64b_36188 (192.168.1.2)
cp64c_49601 (192.168.1.3)
cp64e_37895 (192.168.1.5)
cp64d_37742 (192.168.1.4)
cp64f_39030 (192.168.1.6)

If you see all nodes are successful appeared, you could start
mpd jobs now

Press <Enter> to continue

!不管有沒有錯誤訊息, 只要出現 available hosts 且內容正確無誤即可, 注意, available host 是 mpd.hosts 的內容是否成功確認, 和 mpd_job 內容無關, available 並不代表跑的時候一定要用到, 而是確保這個 hosts 俱有 mpi 程式的執行功能

3. Edit mpd_job file

You may setup the configuration of MPI machines by adding something like

cp64a:2 ifhn=192.168.1.1
cp64c:1 ifhn=192.168.1.3
cp64f:2 ifhn=192.168.1.6
...etc.

For this case, the job you submitted will be executed on cp64a with 2 nodes

1 node on cp64c and 2 nodes on cp64f

Would you like to generate a default mpd_job?<y/n>

:

!同樣, 一開始會詢問是否產生 mpd_job 範例檔, 如果已經產生過就不需要(打 n), 打 y 會覆蓋現有的 mpd_job

Now please modify mpd_job to fit your purpose

Press <Enter> to edit mpd_job by vi

!接下來程式會要求修改 mpd_job, 所有操作與 vi 同, 請記得儲存並退

出(:wq)

4. List all mpd jobs

顯示所有執行中的 mpi 程式, 以及其他使用者的程式, 此方式比使用 top 更方便, 並且不需要修改 a.out 的檔名, 而可以把詳細資訊顯示在 jobalias 中

若沒有 mpi 程式正在執行, 則僅秀出其他使用者正在執行的程式(不限 mpi 程式)

mpi 程式執行資訊的解說如下:

```
jobid      = 1@cp64a_9176      !mpich 自訂 jobid, 相同 id 為同一組
之 mpi 程式
jobalias =                      !可設定更容易了解的代號
username = xanadu              !哪個使用者執行此程式
host       = cp64a             !在哪個 host 上執行
pid        = 25255             !top 資訊的 pid number, 如想取消,
打 kill 25255
sid        = 25252
rank       = 0                 !node 的順位, 由 0 起跳
pgm        = a.out             !主程式名稱
```

```
jobid      = 1@cp64a_9176      !剩下四個 nodes 的資訊, 由 jobid 判斷
屬於同一組
jobalias =
username = xanadu
host      = cp64a
pid       = 25254
sid       = 25253
rank      = 1
pgm       = a.out
```

```
jobid      = 1@cp64a_9176
jobalias =
username = xanadu
host      = cp64b
pid       = 29621
```

sid = 29619
rank = 2
pgm = a. out

jobid = 1@cp64a_9176
jobalias =
username = xanadu
host = cp64b
pid = 29622
sid = 29620
rank = 3
pgm = a. out

jobid = 1@cp64a_9176
jobalias =
username = xanadu
host = cp64c
pid = 18926
sid = 18925
rank = 4
pgm = a. out

jobid = 2@cp64a_9176
jobalias = Ag13Cu25
username = xanadu
host = cp64a
pid = 25420
sid = 25417
rank = 0
pgm = a. out

!另一組 mpi 程式
!這次有指定代號為 Ag13Cu25

jobid = 2@cp64a_9176
jobalias = Ag13Cu25
username = xanadu
host = cp64a
pid = 25419
sid = 25418
rank = 1

!同上屬同一組

pgm = a.out

jobid = 2@cp64a_9176 !同上

jobalias = Ag13Cu25

username = xanadu

host = cp64b

pid = 29662

sid = 29661

rank = 2

pgm = a.out

And the status of current users: !其他使用者資訊

15:57:15 up 7 days, 6:47, 4 users, load average: 0.48, 0.10, 0.03

USER	TTY	LOGIN@	IDLE	JCPU	PCPU	WHAT
xanadu	:0	Wed11	?xdm?	2:21	0.06s	/bin/sh
/opt/kde3/bin/startkde						
xanadu	pts/0	Wed11	5days	0.00s	0.76s	kded
[kdeinit]						
xanadu	pts/1	10:48	5:02m	0.03s	0.03s	/bin/bash
xanadu	pts/2	15:42	0.00s	0.11s	0.00s	- bash

!由上可知有一個mpi程式, jobid為1@cp64a_9176, 共使用5個nodes, 分別是cp64a*2, cp64b*2, cp64c*1;此外, 程式執行檔為a.out, 使用者為xanadu, 無job暱稱;而另一組id為2@cp64a_9176, 共使用3個nodes, 分別是cp64a*2, cp64b*1;暱稱為Ag13Cu25;要刪除某一組job, 請kill掉其中一個pid number, 並使用本選項確認是否清除乾淨!

Q&A:

與傳統方式不同點:

傳統:使用 mpdallexit, mpdboot 並修改 mpd.hosts 初始化 mpi 環境

mpitool:不需理會 mpd.hosts 檔, 使用選項 2 啟用 mpi 環境

傳統:無法機動性分配 job 在哪個 node 上

mpitool:利用每個目錄下不同的 mpd_job 組態達成以上目的

傳統:使用 top 判斷程式的執行狀態, 需適時改變 a.out 名稱以區分 job

mpitool:使用選項 4 可準確區分每個 job 的執行狀況, 並從 job alias 了解此 job 內容為何, 可不用修改 a.out 檔名

傳統:需要分開執行數種程式才可跑主程式

mpitool:全部整合, 已包含 config.out

注意:mpitool 完全不跟舊方法衝突, 亦即也可使用舊方式執行 mpi 程式

附錄一

進階選項, 管理者專用

輸入 mpitool su 則會進入進階選單! 多出 6, 7, 8 選項

<<MPICH2 TOOL MENU>>:

0. Run config.out

1. Start a mpd job

2. Clear all mpd jobs and restart a mpd environment

3. Edit mpd_job file

4. List all mpd jobs

5. Quit

Advance options:

6. Edit /home/mpd.hosts file !由管理者建立/home/mpd.hosts,
建議使用 root 執行

7. Test MPICH2 Performance !測試 mpich2 的效能,

8. Reset MPICH2 environment !重新執行 mpich 的環境變數, 其會要
求輸入 mpich2 的所在目錄

:

子選項說明:

6. Edit /home/mpd.hosts file

Edit /home/mpd.hosts

直接進入 vi 編輯/home/mpd.hosts, 若出現錯誤請修改/home/mpd.hosts 權限或
以 root 執行 mpitool

7. Test MPICH2 Performance

Please input the number of test loop:<ex:100>

:100

送一訊息至所有 hosts, 測試 n 個 loops 所花的時間, 如輸入 100 則測試 100 個
loops

time for 100 loops = 0.105799913406 seconds

測試結果

8. Reset MPICH2 environment

Please input your MPICH2 location

You could try /home or /home/your_account

Note: do not use /home/ or /home/your_account/

The additional character '/' will cause error

:

輸入 mpich2 的所在目錄, 如果一般使用者家目錄下沒有 mpich2 的目錄, 請試著輸入 /home ; 注意, 請勿在目錄後多加一個 " / " 以免造成程式誤判

Configurating bash environment for MPICH2

Configurating complete!

重設環境成功

!這裡的選項即附錄二的環境設定, 一般使用者如要長期使用 mpich2, 建議將附錄二環境變數加至家目錄下的 ".bashrc" 檔

若是 C Shell 請參考 MPICH2 手冊

附錄二:

.bashrc 內容加入:

```
export F90FLAGS="-i4"
export PATH=/home/mpich2/bin:$PATH
export LD_LIBRARY_PATH=/home/mpich2/lib:$LD_LIBRARY_PATH
```

附錄三

Source code: (請存成 mpitool 檔, 權限改成可執行(chmod 755), 並放到 /usr/bin 下)

```
#Directory setting
# copy this file to /usr/bin
mpd_dir=/home # the location of mpd.hosts and mpd.cpu
mpd_secret_word=su # Password for Advance options. To enter advance mode , type mpitool
su
mpd_config=config.out #Default config file name"
echo -e " "
echo -e "=====
MPICH Tool V1.0"
echo -e " This program is written by Po-Jen Hsu"
echo -e " Last updated 2007/02/06"
echo -e "=====
echo -e " "
test -e $mpd_dir/mpd.hosts && file_exist=1 || file_exist=0 #=1 exist; =0 not exist
if [ "$file_exist" == 0 ]; then
```

```

echo -e $mpd_dir"/mpd.hosts does not exist"
echo -e "Please put your mpd.hosts to "$mpd_dir"/ directory"
echo -e ", not your home directory"
echo -e "This file should contain all available host machine"
echo -e "ex:"
echo -e "cp64a:2 ifhn=192.168.1.1"
echo -e "cp64b:2 ifhn=192.168.1.2"
echo -e "..etc."
read -p "Press <Enter> to edit mpd.hosts by vi"
vi $mpd_dir/mpd.hosts
else
    declare -i num_hosts='wc -l $mpd_dir/mpd.hosts | cut -c1-2'
    echo -e " "
    echo -e "#-----"
    echo -e "#Number of available host machine= "$num_hosts
# store number of hosts in $num_hosts
fi
test -e $mpd_dir/mpd.cpu && file_exist=1 || file_exist=0    #=1 exist ; =0 not exist
if [ "$file_exist" == 0 ]; then
    echo -e $mpd_dir"/mpd.cpu does not exist"
    echo -e "Please put your mpd.cpu to "$mpd_dir"/ directory"
    echo -e ", not your home directory"
    echo -e "This file should contain number of processors per host"
    echo -e "ex:"
    echo -e "2 "
    echo -e "When you finish, please press Ctrl-D"
    cat > $mpd_dir/mpd.cpu
else
    declare -i cpu_per_host='cut -c1-2 $mpd_dir/mpd.cpu'
    echo -e "#Number of processors per host= "$cpu_per_host
    echo -e "#-----"
fi

test -e mpd_job && file_exist=1 || file_exist=0    #=1 exist ; =0 not exist
if [ "$file_exist" == 0 ]; then
    echo -e " "
    echo -e "Warning!! mpd_job does not exist"
    echo -e "Please edit mpd_job first"

```



```

echo -e "ex:"
echo -e "cp64a:2 ifhn=192.168.1.1"
echo -e "cp64c:1 ifhn=192.168.1.3"
echo -e "cp64f:2 ifhn=192.168.1.6"
echo -e "..etc."
echo -e "For this case, the job you submitted will be executed on cp64a with 2 nodes"
echo -e "1 node on cp64c and 2 nodes on cp64f"
echo -e " "
echo -e "Do you want to make a default mpd_job file?<y/n>"
read -p ": " yn
if [ "$yn" == "y" ]; then
    cat $mpd_dir/mpd.hosts > mpd_job
    echo -e "Although the default mpd_job file is generated,"
    echo -e "it should be modified to fit your purpose"
    read -p "Press <Enter> to edit mpd_job by vi"
    vi mpd_job
    echo -e "Your mpd_job is successfully modified:"
    echo -e " "
fi
fi

echo -e " "
echo -n "#Current directory is " && pwd
echo -e "#-----"
echo -e "#The available host machine are:"
cat $mpd_dir/mpd.hosts
echo -e " "
echo -e "#-----"
until [ "$sel" == 5 ]
do
#####Loop Menu#####
    echo -e " "
    read -p "Press <Enter> to continue"
    echo -e "#This shows how your job be submitted to each node"
    cat mpd_job
    echo -e "#-----"
    echo -e "<<MPICH2 UTILITY MENU>>:"
    echo -e "0.Run config.out"

```

```

echo -e "1.Start a mpd job"
echo -e "2.Clear all mpd jobs and restart a mpd environment"
echo -e "3.Edit mpd_job file"
echo -e "4.List all mpd jobs"
echo -e "5.Quit"
if [ "$1" == "$mpd_secret_word" ]; then
    echo -e "Advance options:"
    echo -e "6.Edit "$mpd_dir"/mpd.hosts file"
    echo -e "7.Test MPICH2 Performance"
    echo -e "8.Reset MPICH2 environment"
fi
read -p ": " sel

if [ "$sel" == 0 ]; then
    test -e $mpd_config && file_exist=1 || file_exist=0    #1 exist ; =0 not exist
    if [ "$file_exist" == 1 ]; then
        ./$mpd_config
    else
        echo -e "config.out does not exist, please indicate your config file name"
        read -p ": " filename
        ./$filename
    fi
fi

#####Start mpd job#####
if [ "$sel" == 1 ]; then
    echo -e "Please make sure the mpd_job file in you executable directory is correct"
    cat mpd_job
    declare -i total_nodes=$num_hosts*$cpu_per_host
    echo -e " "
#    echo -e "The max number of nodes= "$total_nodes
    echo -e "Please input the number of processes."
    echo -e "This value should not be greater than the total number of nodes in mpd_job"
    read -p ": " nodes
    echo -e "Please input the full name of the executable file,ex:a.out"
    read -p ": " filename
    test -e $filename && file_exist=1 || file_exist=0    #1=exist 0=not exist
    if [ "$file_exist" == 1 ]; then
        echo -e "Please input a short alias of this job<ex:Ag13Cu25>"
    fi
fi

```

```

echo -e "Or enter none for no alias<Please don't skip this procedure>"
read -p ": " jobalias
if [ "$jobalias" == "none" ]; then
    echo -e "No job alias"
else
    echo -e "Job alias=" $jobalias
fi
echo -e "1.Submit job in foreground"
echo -e "2.Submit job in background"
read -p ": " background
echo -e "Are you sure?<y/n>"
read -p ": " yn
if [ "$yn" == "y" ]; then
    if [ "$background" == 1 ]; then
        if [ "$jobalias" == "none" ]; then
            mpiexec -machinefile mpd_job -n $nodes -path . $filename
        else
            mpiexec -a $jobalias -machinefile mpd_job -n $nodes -path . $filename
        fi
        exit 0
    fi
    if [ "$background" == 2 ]; then
        if [ "$jobalias" == "none" ]; then
            mpiexec -machinefile mpd_job -n $nodes -path . $filename > /dev/null &
        else
            mpiexec -a $jobalias -machinefile mpd_job -n $nodes -path . $filename >
/dev/null &
        fi
        exit 0
    fi
else
    echo -e "Abort and return to menu.."
fi
fi
if [ "$file_exist" == 0 ]; then
    echo -e "Error! file does not exist, please try again"
fi
fi

```

```
#####Start MPD ring#####
if [ "$sel" == 2 ]; then
    echo -e "**If there is no mpd daemon running, it will show some error"
    echo -e "message. This is normal"
    echo -e " "
    echo -e "This will stop all running jobs"
    echo -e "Are you sure?<y/n>"
    read -p ": " yn
    if [ "$yn" == "y" ]; then
#   read -p "Please input number of hosts: " hosts
#   read -p "Please input number of processor per host: " cpu
        mpdallexit
#   mpdboot --file=/home/mpd.hosts --ncpus=$cpu
#       mpdboot --file=$mpd_dir/mpd.hosts -n $num_hosts --ncpus=$cpu_per_host
        mpdboot --file=$mpd_dir/mpd.hosts -n $num_hosts
        echo -e "Available hosts:"
        mpdtrace
        echo -e " "
        echo -e "Hostname_PID# IP"
        mpdtrace -l
        echo -e " "
        echo -e "If you see all nodes are successful appeared, you could start mpd jobs
now"
    else
        echo -e " Nothing changed"
    fi
fi

#####Edit mpd_job ilfe#####
if [ "$sel" == 3 ]; then
    echo -e "You may setup the configuration of MPI machines by adding something like"
    echo -e "cp64a:2 ifhn=192.168.1.1"
    echo -e "cp64c:1 ifhn=192.168.1.3"
    echo -e "cp64f:2 ifhn=192.168.1.6"
    echo -e "..etc."
    echo -e "For this case, the job you submitted will be executed on cp64a with 2 nodes"
    echo -e "1 node on cp64c and 2 nodes on cp64f"
    echo -e " "
    echo -e "Would you like to generate a default mpd_job?<y/n>"

```

```

read -p ": " yn
if [ "$yn" == "y" ]; then
    cat $mpd_dir/mpd.hosts > mpd_job
fi
echo -e "Now please modify mpd_job to fit your purpose"
read -p "Press <Enter> to edit mpd_job by vi"
vi mpd_job
fi
#####List MPD jobs#####
if [ "$sel" == 4 ]; then
    mpdlistjobs
    echo -e "And the status of current users:"
    w
fi
#####Exit#####
if [ "$sel" == 5 ]; then
    time='date'
    echo -e " "
    echo -e "Now it is "$time
    echo -e "Have a nice day"
    echo -e " "
    exit 0
fi
#####Edit mpd.hosts file in /home/mpd.hosts#####
if [ "$sel" == 6 ]; then
    echo -e "Edit "$mpd_dir"/mpd.hosts"
    read -p "Press <Enter> to continue"
    vi $mpd_dir/mpd.hosts
fi
#####Test MPD loop performance#####
if [ "$sel" == 7 ]; then
    echo -e "Please input the number of test loop:<ex:100>"
    read -p ": " num_loop
    echo -e " "
    mpdringtest $num_loop
    echo -e " "
fi
if [ "$sel" == 8 ]; then

```

```

echo -e "Please input your MPICH2 location"
echo -e "You could try /home or /home/your_account"
echo -e "Note: do not use /home/ or /home/your_account/"
echo -e "The additional character '/' will cause error"
read -p ": " mpich2_dir
echo -e "Configurating bash environment for MPICH2"
export F90FLAGS="-i4"
#export PGI=/home/pgi          #alternative pgi default path in /home
export PATH=$mpich2_dir/mpich2/bin:$PATH
export LD_LIBRARY_PATH=$mpich2_dir/mpich2/lib:$LD_LIBRARY_PATH
#export PATH=/home/mpich2/bin:$PATH          #alternative MPICH2 path in
/home
#export LD_LIBRARY_PATH=/home/mpich2/lib:$LD_LIBRARY_PATH    #alternative MPICH2
path in /home
#Please use either notation one or non-notation one
echo -e "Configurating complete!"
fi
done
#####End of MPICH Tool#####

```