台灣杉一號

Peta 級超級電腦

使用者操作手冊

版次：2.2

更新時間：2021/07/13

台灣杉

TAIWANIA

NARLabs 財團法人國家實驗研究院

國家高速網路與計算中心

National Center for High-performance Computing

聯絡窗口

帳號申請服務：

姓名：陳 小 姐

電話：03 - 5776085 轉 408

Email：iservice@narlabs.org.tw

技術支援服務：

姓名：鄭 小 姐

電話：03 - 5776085 轉 214

Email：isupport@narlabs.org.tw

操作手冊目錄

1. 手冊簡介 1

1.1 修訂紀錄 2

2. 台灣杉一號簡介 5

2.1 系統概觀 5

2.2 可用計算資源 5

2.3 可用儲存資源 6

2.3.1 家目錄區域 /home 6

2.3.2 暫存工作區域 /work1 6

2.3.3 計畫儲存區域 7

2.4 前端伺服器 7

2.4.1 登入節點 7

2.4.2 互動式節點 8

2.4.3 資料傳輸節點 8

3. 進入系統的方式 9

3.1 台灣杉一號帳號註冊 9

3.2 主機帳號、OTP 資訊 12

3.3 由命令列登入主機 15

3.4 由命令列登出主機 17

3.5 檔案傳輸 18

3.5.1 Linux 用戶 18

3.5.2 Windows 用戶 19

4. 編譯與連結 20

4.1 環境模組 20

4.2 Intel 編譯器 21

4.2.1 載入編譯器環境模組 21

4.2.2 序列程式 (serial program) 21

4.2.3 Thread parallel 程式 22

4.2.4 MPI parallel 程式 22
4.3 PGI 編譯器 23
4.3.1 載入編譯器環境 23
4.3.2 序列程式 23
4.3.3 Thread parallel 程式 23
4.3.4 MPI parallel 程式 23
5. 操作 PBS PRO job 24
5.1 Job 佇列 (queue) 24
5.2 Queue 列表 25
5.3 提交 job 26
5.3.1 PBS job script 26
5.3.2 批次提交 job 28
5.3.3 提交 array job (bulk job) 29
5.3.4 Job script 設定 e-mail 通知 30
5.4 刪除 job 31
5.5 顯示 job 狀態 31

1. 手冊簡介

本使用者操作手冊將說明國網中心的 Peta 級超級電腦-台灣杉一號的使用方法。
使用台灣杉一號前請先取得並詳讀最新版的使用者操作手冊。

1.1 修訂紀錄

版次	日期	變更內容	變更者	檢閱者	審核者
0.1	2018/02/08	- 初版	Imura	Ishan	Yamada
1.0	2018/03/28	- 修訂 2.3、4.1、5.2 章節 - 新增第 6 章— 登入節點的資源限制	Imura	Ishan	Yamada
1.1	2018/04/12	- 修訂 4.2.4 章節— MPI parallel 程式 - 修訂 4.3.4 章節— MPI parallel CUDA 程式 - 新增 4.4.2 章節— MPI parallel CUDA 程式 - 新增 5.3.4 章節— 設定 e-mail 通知 - 新增 5.6 章節— 建立和使用預訂	Yoshida	Ishan	Yamada
1.2	2018/07/03	- 修訂 Job Script 範例	Oscar	Oscar	Oscar
	2018/07/25	- 新增 Queue：ct160	Oscar	Oscar	Oscar
1.3	2018/12/28	- 修訂 5.1 章 修訂 Job queue 新增 Queue policy	Oscar 中文版：Viga	Oscar 中文版：Viga	Oscar 中文版：Viga
1.4	2019/03/04	- 修訂 5.1 章	中文版：Viga	中文版：Viga	中文版：Viga

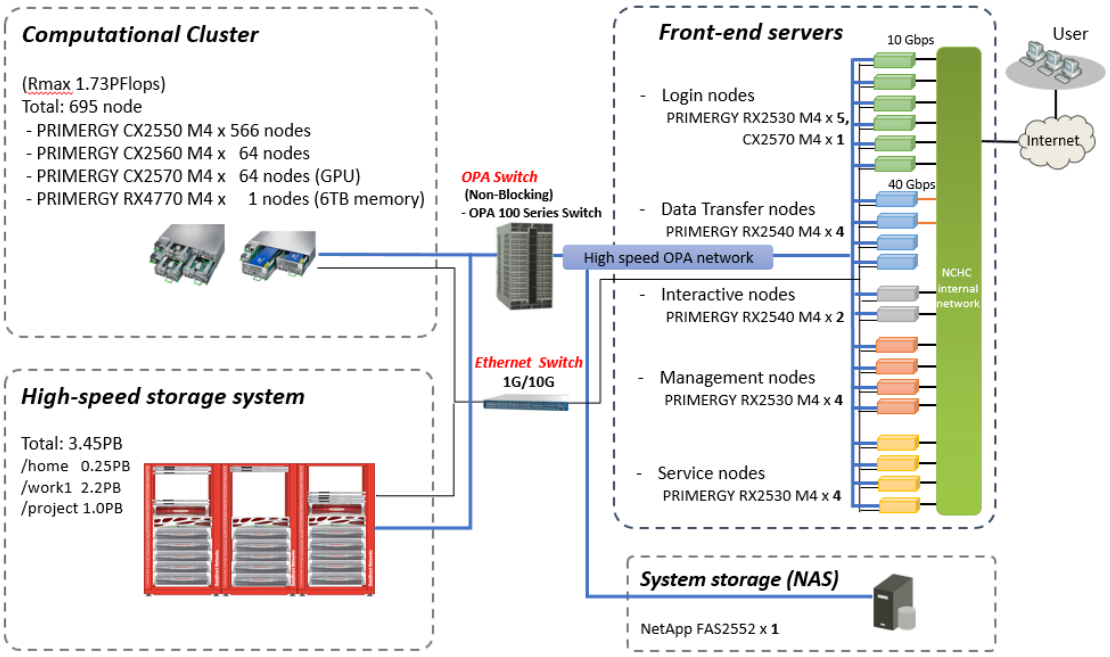
		修訂 Job queue			
1.5	2019/ 03/08	- 修訂 3.2 章 修訂 OTP 內容	中文版：Viga	中文版：Viga	中文版：Viga
1.6	2019/ 07/09	- 修訂 2.3.1 章 檔案刪除說明 - 修訂 3.1 章 圖示更新 - 修訂 3.2 章 增加帳號密碼說明	中文版：Viga	中文版：Viga	中文版：Viga
1.7	2019/ 08/26	- 修訂 5.1 章 - 修訂 Job queue	中文版：Viga	中文版：Viga	中文版：Viga
1.8	2019/ 12/03	- 修訂 2.2 章 - 修訂 2.4.1 章 - 修訂 3.1 章 - 修訂 3.3 章 - 修訂 4.1 章 - 移除 4.4 章 - 修訂 5.1 章 - 修訂 5.2 章 - 修訂 5.3.1 章 - 移除 5.6 章 - 移除第 6 章	中文版：Viga	中文版：Viga	中文版：Viga

1.9	2020/ 01/08	- 修訂 5.1 章 修訂 Job queue	中文版：Viga	中文版：Viga	中文版：Viga
2.0	2020/ 04/24	- 修訂 2.3.3 章 - 修訂 2.4.2 章 - 修訂 3.2 章	中文版：Viga	中文版：Viga	中文版：Viga
2.1	2020/ 04/30	- 修訂 3.3 章 - 修訂 4.3.4 章	中文版：Viga	中文版：Viga	中文版：Viga
2.1	2020/ 07/09	- 修訂 5.1 章 - 修訂 Job queue	中文版：Viga	中文版：Viga	中文版：Viga
2.1	2020/ 11/11	- 修訂 3.3 章 - 修訂 5.1 章	中文版：Oscar	中文版：Oscar	中文版：Oscar
2.1	2020/ 12/18	- 修訂 2.2 章	中文版：Oscar	中文版：Oscar	中文版：Oscar
2.1	2020/ 12/30	- 修訂 5.1 章	中文版：Oscar	中文版：Oscar	中文版：Oscar
2.2	2021/ 04/11	- 修訂封面頁聯絡窗口 - 修訂 3.2 章 - 修訂 3.3 章 - 修訂 3.5.2 章	中文版：Oscar	中文版：Oscar	中文版：Oscar
2.2	2021/ 05/11	- 修訂 5.1 章	中文版：Oscar	中文版：Oscar	中文版：Oscar
	2021/ 06/09	- 修訂 5.1 章	中文版：Oscar	中文版：Oscar	中文版：Oscar
	2021/ 07/13	- 修訂 5.1 章	中文版：Oscar	中文版：Oscar	中文版：Oscar

2. 台灣杉一號簡介

2.1 系統概觀

台灣杉一號的建構概觀呈現如下圖：



此系統主要由以下四項要素組成：

1. 計算叢集 (Computational Cluster)
2. 前端伺服器 (Front-end Servers)
3. 高速儲存系統
4. 系統儲存 (網路附加儲存) (System Storage (NAS))

以上四種要素透過乙太網路與 Intel Omni-Path 高速網路相互串聯。

2.2 可用計算資源

台灣杉一號共有 750 個計算節點 (1500 個處理器與 30000 個核心)，整體效能可達約 1.49 兆次浮點運算 (PFLOPS)。750 個計算節點由雙 CPU 插槽組成，每個插槽具有 Xeon Gold 6148 CPU (20 核心、2.40GHz)。

750 個節點可細分爲以下類別：

- **瘦節點 (Thin Node)**：供大多數高效能運算 (HPC) 應用程式使用
- **胖節點 (Fat Node)**：供需大量記憶體的 HPC 應用程式使用

各類計算節點與其資源總結如下表：

節點種類	節點範圍	總個數 (節點)	單位計算資源 (節點)					
			CPU 插槽數	CPU 核心數	記憶體 (GB)	Tesla P100	10Gbps interface	480 GB SSD
瘦節點	cn0101 – cn0673	438	2	40	192	-	-	-
瘦節點	cn0701 – cn0764	64	2	40	192	-	1	-
瘦節點	cn1301 – cn1360	60	2	40	192	-	-	-
胖節點	cn0801 – cn0864	64	2	40	384	-	-	-
胖節點	cn0901 – cn0964	64	2	40	384	-	-	1
胖節點	cn1201 – cn1260	60	2	40	384	-	-	1

2.3 可用儲存資源

下表爲您在台灣杉一號可用的高速儲存系統資源。掛載爲 lustre 檔案系統，可透過高速 OPA 網路從所有前端伺服器與運算節點進入使用。

	儲存區域	掛載點	容量
1	家目錄區域	/home	0.25 PB
2	暫存工作區域	/work1	2.2 PB
3	計畫儲存區域	/project	1.0 PB

2.3.1 家目錄區域 /home

0.25 PB 的家目錄空間可儲存私人檔案。可於此編輯程式、執行與管理計算工作 (job)。每位使用者皆有預設 100GB 的空間額度可使用。

計畫到期後，儲存於 /home 之檔案將於用戶提出刪除要求後，系統管理員才會手動刪除。

2.3.2 暫存工作區域 /work1

此區域提供 **2.2 PB** 的可儲存空間，主要是做爲儲存用戶計算過程中的暫時資料。每個帳號在 /work1 磁碟下皆預設有 1.5TB 的空間額度。此叢集上的空間係設計給計算工作儲存而非長期儲存用。爲了維持/work1 穩定且高效的狀態，本中心將定期執行自動清除的工作。本系統無對/work1 的資料進行備份，請您自行備份資料。因資料無法復原(含因系統當機或硬體故障而損失的資料)，

請定期備份您重要的資料。在 28 天內，未存取的檔案將被清除，因此強烈建議您定期清除/work1 下的資料以增加使用效能，並定期備份所需保留的資料。

可使用以下指令將資料從 /work1 複製至 /home 或 /project：

```
[user@clogin1]$ cp /work1/<path to target file> /project/<destination path>
```

其他 cp 指令主要搭配使用的選項：

-p 保存修改的時間、讀取的時間、原始檔案的類型

-r 複製整個目錄(含子目錄)

2.3.3 計畫儲存區域

限付費專案用戶存取。

2.4 前端伺服器

僅限定台灣 IP 進入。

2.4.1 登入節點

以下爲三個主要可由命令列登入台灣杉一號的節點：

140.110.148.11 clogin1.twnia.nchc.org.tw

140.110.148.12 clogin2.twnia.nchc.org.tw

使用者可以由不同的登入節點登入，所有登入節點的配置皆相同。

用戶的資料不是儲存在登入節點的磁碟上，而是統一存放掛載於各節點上的高速儲存系統。

因此，您無需限定登入的節點。

您可從登入節點執行下列工作：

- 提交/管理 HPC job
- 有全權存取高速儲存系統上的檔案
- 編譯 HPC 相關應用程式
- 運行除錯程式以改善程式碼

計算節點與登入節點擁有相似的技術規格，因此這兩種節點擁有相容的環境，提供開發與測試應用程式碼的服務。

計算節點的資源總結如下表：

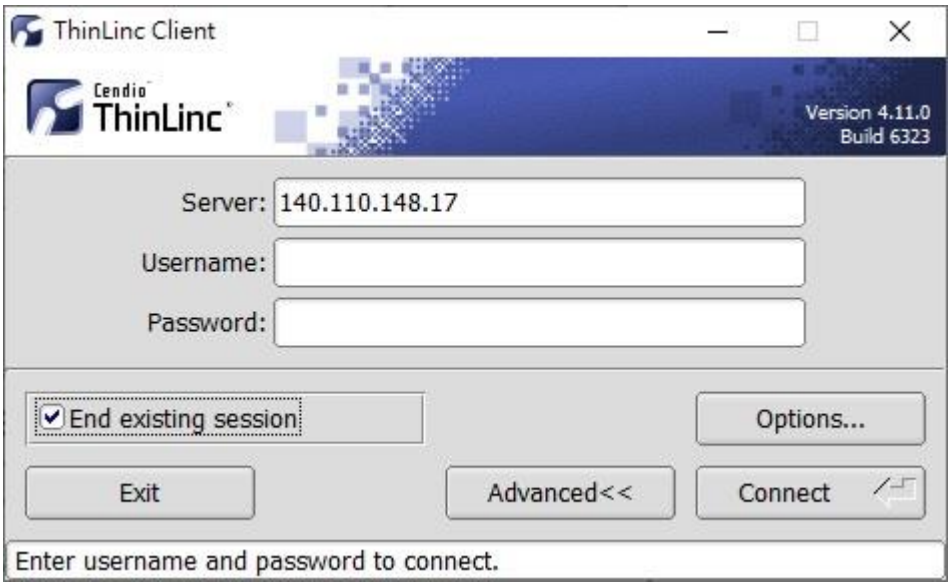
節點種類	節點範圍	總個數 (節點)	每單位的計算資源 (節點)				
			CPU 插槽數	CPU 核心數	記憶體 (GB)	Tesla P100	480 GB SSD
CPU 登入節點	clogin1–clogin2	2	2	40	384	-	1

請勿於登入節點進行計算工作，若多人如此執行，登入節點將會當機，導致其他使用者無法登入此叢集。

2.4.2 互動式節點

140.110.148.17 intact1.nchc.org.tw
140.110.148.18 intact2.nchc.org.tw

互動式節點提供完整的圖形化桌面環境。這些 X Window 系統可讓使用者執行 2D 繪圖與 3D 建模，您的連線電腦請下載與安裝好 Cendio 公司的 ThinLinc Client，依照圖示教學登入到互動式節點。



請勿於互動式節點進行計算工作，若多人如此執行，互動式節點將會當機，導致其他使用者無法登入此叢集。

2.4.3 資料傳輸節點

140.110.148.21 xdata1.twnia.nchc.org.tw
140.110.148.22 xdata2.twnia.nchc.org.tw

以上兩個資料傳輸節點可讓用戶的資料從外部網路傳出/傳入高速運算系統。
每一節點藉由 40Gbps HCA 介面卡連接外部網路，並如同其他節點，由 OPA 介面卡連接高速儲存系統。藉由此配置，資料能在您的來源電腦與高速儲存系統之間傳輸。
也因爲此目的，您只能使用 scp/sftp 的方式透過此類節點進行資料傳輸，
資料傳輸節點不提供 shell，故不能用來登入。

3. 進入系統的方式

3.1 台灣杉一號帳號註冊

1. 進入 iService (帳號與計畫管理平台) 註冊網頁，並點選「立即申請」

2. 點選「現在就加入會員」，加入 iService

3. 按步驟填寫基本資料—設定 iService 會員帳號與密碼

4. 按步驟填寫基本資料—設定台灣杉一號主機帳號與密碼

5. E-mail 帳號認證完成後，點選「手機認證」，再輸入手機簡訊所收到的

SMS 認證碼，便註冊完成！

3.2 主機帳號、OTP 資訊

登入台灣杉一號之帳號爲註冊 iService 時所設定的「主機帳號」，除了輸入「主機密碼」，還要輸入「OTP」認證碼。
OTP (One Time Password) 認證碼爲一次性密碼，又稱爲動態密碼。具高度安全性，30 秒更新一次，能保障用戶權益。

主機帳號、OTP 查看方式如下 (主機密碼僅能修改，無提供查看的功能)：

登入 iService 後可由會員中心→會員資訊–主機帳號資訊

① 查看「主機帳號」、修改主機密碼
② 修改主機密碼
③ 查詢 OTP 認證碼 (網頁將會每 30 秒產生一次)
其他查詢 OTP 認證碼的方式：
A. 手機/平板：
- 至 Google Play 商店/ App Store 搜尋並下載 Microsoft Authenticator 應用程式

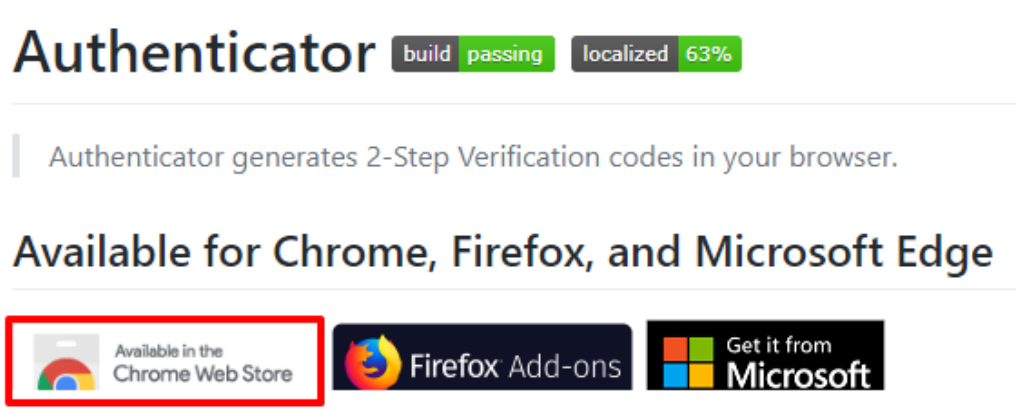
- 安裝完打開應用程式，掃描 iService 網頁「點我複製 OTP 金鑰」上的 QR code，應用程式即能查詢 OTP 認證碼。

除了 Microsoft Authenticator，其他應用程式可參考：
https://alternativeto.net/software/microsoft-authenticator/

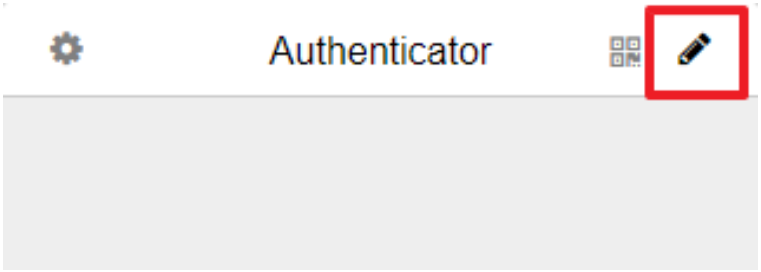
B. 電腦：

a. 瀏覽器擴充功能，以下以 Google Chrome 爲例：

- 至連結下方點入 Authenticator 擴充功能頁面，並新增功能至 Chrome (可選擇 Firefox 或 Microsoft Edge 瀏覽器適用的版本)



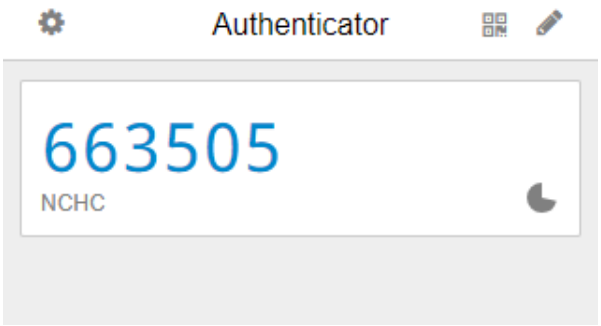
- 開啟 Authenticator，點選右上角的編輯



- 點選掃描 QR 碼

- 使用虛線將 iService 的 OTP QR code 框起來

- 新增成功後，點開擴充程式便可查看 OTP



並可由會員中心→計畫管理–我的計畫，查看「系統計畫代號」：

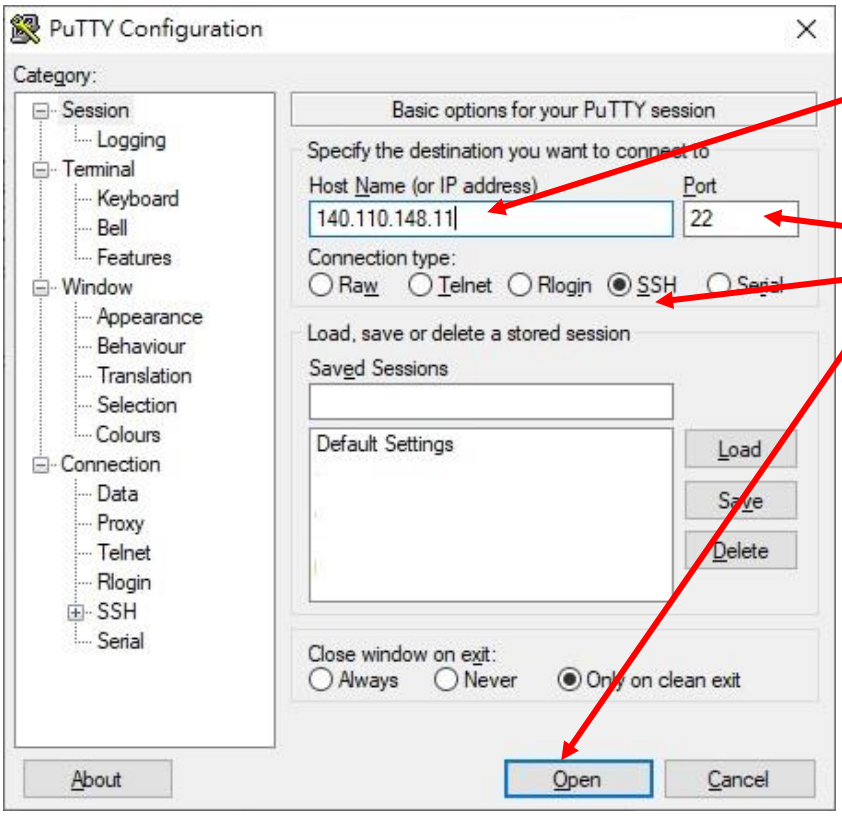
3.3 由命令列登入主機

使用者需使用核可的主機帳號、密碼與一次性密碼 (OTP) 登入系統，請確認您在開始遠端連線至台灣杉一號前，已完成下列設定：

1. 至 iService 會員註冊網站申請登入台灣杉一號的主機帳號密碼
   註：此非一次性密碼 (OTP) ，爲系統主機帳號的密碼。登入高速運算系統需使用此密碼與一次性密碼
2. 登入 iService 後並取得 OTP (取得方法可參考 3.2 章節)

完成後您即可使用電腦，透過 SSH 用戶端軟體 (例：PuTTY, MobaXterm) 並按照下列步驟，連線至台灣杉一號：

1. 在您的電腦開啟 SSH 用戶端軟體
2. 輸入 Host IP 與 port 數
   註：以下的 IP 皆可由台灣各地直接連線進入。若您不在台灣，將無法連線主機



請輸入以下資訊：

①**Host:** 選擇以下任一 IP 輸入

- 140.110.148.11
- 140.110.148.12

②**TCP port:** 22

③**Connection type:** SSH

④點選 **Open**

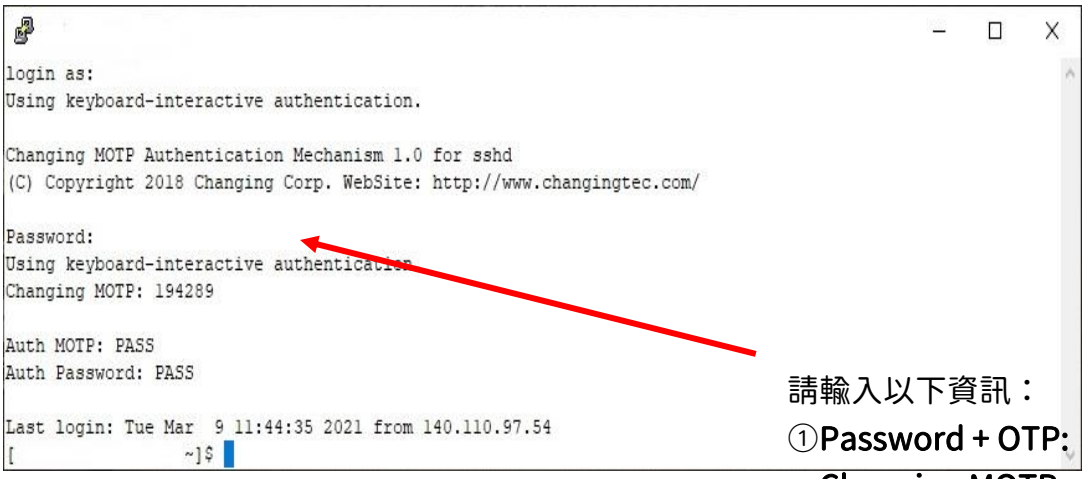
在第一次登入高速運算系統時，會彈出有關 key fingerprint (金鑰指紋)的訊息，請選擇「yes」並繼續

3. 在「Connection type」欄位點選「SSH」，並按下「Open」
4. 此時您需輸入帳號與密碼，請在輸入您的主機帳號之後，按下「Enter」，再分開輸入密碼與一次性密碼 (OTP) ，並按下「Enter」

請輸入以下資訊：

①**User name:** 你的主機帳號

②點選 **Enter** 以連線系統



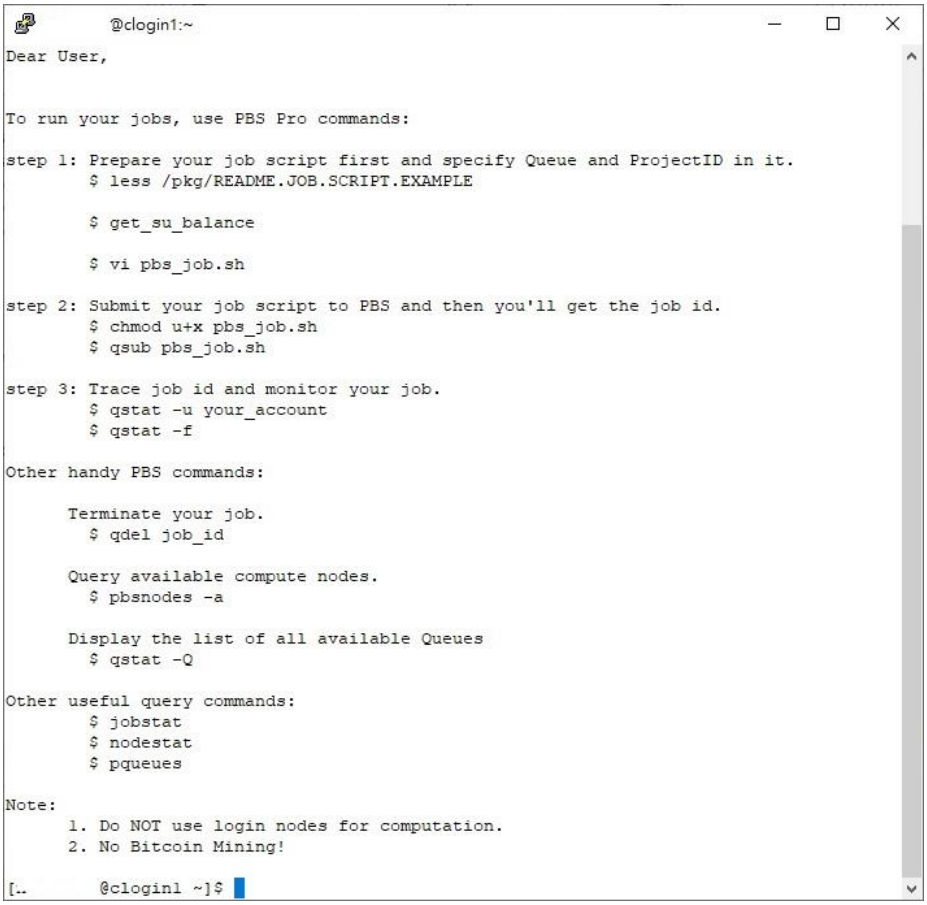
請輸入以下資訊：

①**Password + OTP:** 你的密碼

Changing MOTP: 你的 OTP 碼

②點選 **Enter** 以連線系統

在正確認證後，便可成功登入節點，如下圖所示：



3.4 由命令列登出主機

執行以下的指令便可登出主機：

```
[user@clogin1~]$ exit
```

3.5 檔案傳輸

請使用 scp/sftp 來將您的電腦/工作站的檔案傳出/傳入台灣杉一號：
Linux/UNIX 用戶請使用 scp 或 sftp 指令來傳輸；Windows 用戶請使用用戶端軟體來傳輸 (例：WinSCP)

3.5.1 Linux 用戶

使用 scp 指令並連接任一個資料傳輸節點：

```
$ scp [option] <source host>:<local path of directory or file>
<destination host>:<remote path of directory or file>
```

scp 指令主要可搭配使用的選項：

-p 保存修改的時間、讀取的時間、原始檔案的類型
-r 複製個目錄 (含子目錄)

使用 sftp 指令並連接任一資料傳輸節點：

```
$ sftp [option] [username@]<destination host>
Connected to <destination host>.
# 將檔案下載至本機的當前目錄
sftp> get <remote path of directory or file>
# 將檔案上傳至伺服器的當前目錄
sftp> put <local path of directory or file>
# 離開 sftp
sftp> bye
```

其他主要 sftp 可搭配使用的選項：

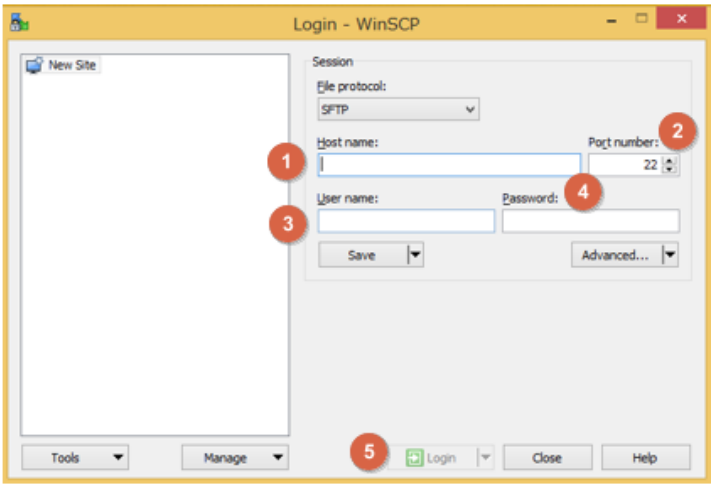
-p 保存修改的時間、讀取的時間、原始檔案的類型
-r 複製個目錄 (含子目錄)

其他主要 sftp 可用的內建指令：

cd *<path>* 更改遠端目錄至*<path>.*
pwd 顯示遠端當前工作目錄
lcd *<path>* 更改本機目錄至*<path>.*
lpwd 顯示主機當前工作目錄.

3.5.2 Windows 用戶

開啟 WinSCP 並連接系統任一資料傳輸節點。連接進入後，您即可用拖曳方式傳輸檔案
以下是 WinSCP 的登入視窗：



請輸入以下資訊：

①**Host name**：選擇以下任一 IP 輸入

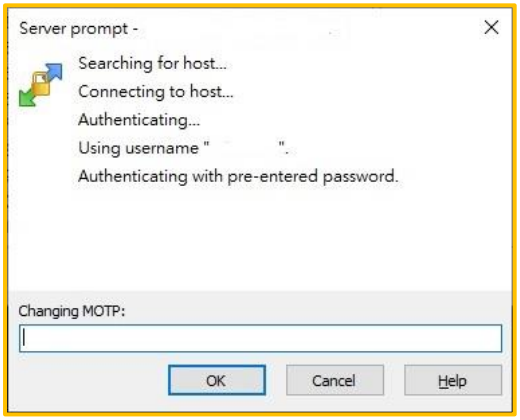
- 140.110.148.21
- 140.110.148.22

②**Port number**：22

③**User name**：你的主機帳號

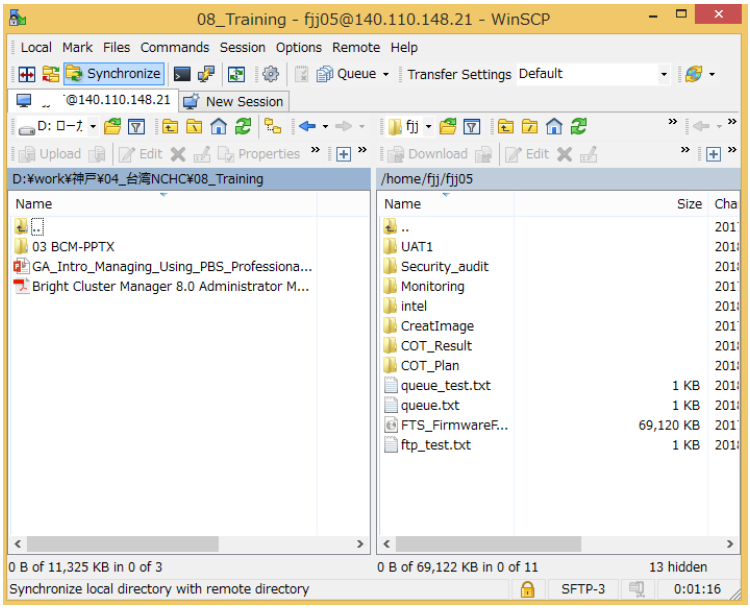
④**Password**：你的密碼

Changing MOTP：你的 OTP 碼



⑤按下 **login** 按鈕

連線成功後，WinSCP 視窗會呈現如下：



4. 編譯與連結

4.1 環境模組

使用編譯器、函式庫、應用程式必須先設定環境，而環境可由 module 指令更換

1. 在登入節點列出所有可用的模組：

```
[user@clogin1]$ module avail
```

2. 載入編譯器 (compiler) 、函式庫 (library) 、應用程式 (application) 所須使用的模組：

```
[user@clogin1]$ module load <module name>
```

3. 增加模組：

```
[user@clogin1]$ module add <module name>
```

4. 以下爲台灣杉一號主要模組的名稱與描述：

模組名稱	描述
blacs/openmpi/gcc/64/1.1patch03	Blacs library
blas/gcc/64/3.7.0	Basic Linear Algebra Subprograms for GNU
bonnie++/1.97.1	Bonnie++ library
fftw2/openmpi/gcc/64/double/2.1.5	FFTW library
fftw2/openmpi/gcc/64/float/2.1.5	FFTW library
fftw3/openmpi/gcc/64/3.3.6	FFTW library
gdb/7.12.1	GNU Cross Compilers
hdf5/1.10.1	Hierarchical Data Format
hwloc/1.11.6	Hardware Locality
intel/2017_u4	Intel Parallel Studio XE 2017 update 4
intel/2018_init	Intel Parallel Studio XE 2018 Initial
intel/2018_u1	Intel Parallel Studio XE 2018 update 1
iozone/3_465	File system benchmark tool
lapack/gcc/64/3.7.0	Linear Algebra package
mvapich2/gcc/64/2.2rc1	MVAPICH MPI library
netcdf/gcc/64/4.6.0	Network Common Data Form library
netperf/2.7.0	Network benchmark
openmpi/gcc/64/1.10.3	GCC compiled OpenMPI
openmpi/pgi/2.1.2/2017	PGI compiled OpenMPI

petsc/openmpi/gcc/3.8.0	PETSc data structure library
pgi/17.10	PGI compilers and development tools
scalapack/openmpi/gcc/64/2.0.2	Scalable Linear Algebra Library

註：openmpi/gcc 與 openmpi/pgi 的設定是互相衝突的，所以您無法同時載入兩種模組。intel/2017_u4、intel/2018_init 與 intel/2018_u1 之間也有相同的限制，您只能一次載入一種模組

5. 列出所有已載入的模組：

```
[user@clogin1]$ module list
```

6. 卸載模組：

```
[user@clogin1]$ module unload <module name>
```

7. 卸載所有的模組：

```
[user@clogin1]$ module purge
```

4.2 Intel 編譯器

4.2.1 載入編譯器環境模組

1. 載入 Intel 編譯器環境模組：

```
[user@clogin1]$ module load intel/2018_u1
# 選擇可搭配該版本使用的模組
```

4.2.2 序列程式 (serial program)

1. 編譯/連結 C 程式

```
[user@clogin1]$ icc –o sample.exe  sample.c
```

2. 編譯/連結 C++ 程式

```
[user@clogin1]$ icpc –o sample.exe sample.c
```

3. 編譯/連結 Fortran 程式

```
[user@clogin1]$ ifort –o sample.exe sample.f
```

4.2.3 Thread parallel 程式

1. 編譯/連結 C 程式

```
[user@clogin1]$ icc -qopenmp –o sample_omp.exe sample_omp.c
```

2. 編譯/連結 C++ 程式

```
[user@clogin1]$ icpc -qopenmp –o sample_omp.exe sample_omp.c
```

3. 編譯/連結 Fortran 程式

```
[user@clogin1]$ ifort -qopenmp –o sample_omp.exe sample_omp.f
```

4.2.4 MPI parallel 程式

1. 編譯連結 MPI 函式庫的 C 原始碼：

```
[user@clogin1]$ mpiicc –o sample_mpi.exe sample_mpi.c
```

2. 編譯連結 MPI 函式庫的 C++ 原始碼：

```
[user@clogin1]$ mpiicpc –o sample_mpi.exe sample_mpi.c
```

3. 編譯連結 MPI 函式庫的 Fortan 原始碼：

```
[user@clogin1]$ mpiifort –o sample_mpi.exe sample_mpi.f
```

4. 運行 Intel 函式庫編譯的 parallel 程式任務腳本 (Job script) 範例：

```
#!/bin/bash
#PBS -P TRI654321
#PBS -N sample_job
#PBS -l select=2:ncpus=40:mpiprocs=4
#PBS -l walltime=00:30:00
#PBS -q ctest
#PBS -j oe

module load intel/2018_u1
cd ${PBS_O_WORKDIR:-"."}
export I_MPI_HYDRA_BRANCH_COUNT=-1
mpiexec.hydra -PSM2 ./sample_mpi.exe
```

請在運行 mpirun 之前，匯出環境值「export I_MPI_HYDRA_BRANCH_COUNT= - 1」

4.3 PGI 編譯器

4.3.1 載入編譯器環境

1. 載入 PGI 編譯器環境

```
[user@clogin1]$ module load pgi/17.10
```

4.3.2 序列程式

1. 編譯/連結 C 程式

```
[user@clogin1]$ pgcc –o sample.exe sample.c
```

2. 編譯/連結 C++ 程式

```
[user@clogin1]$ pgc++ –o sample.exe sample.c
```

3. 編譯/連結 Fortan 程式

```
[user@clogin1]$ pgfortran –o sample.exe sample.f
```

4.3.3 Thread parallel 程式

1. 編譯/連結 C 程式

```
[user@clogin1]$ pgcc –mp –o sample_omp.exe sample_omp.c
```

2. 編譯/連結 C++ 程式

```
[user@clogin1]$ pgc++ -mp –o sample_omp.exe sample_omp.c
```

3. 編譯/連結 Fortan 程式

```
[user@clogin1]$ pgfortran –mp –o sample_omp.exe sample_omp.f
```

4.3.4 MPI parallel 程式

1. 載入編譯程式的環境模組

```
[user@clogin1]$ module load pgi/19.4
```

```
[user@clogin1]$ module load mpi/openmpi-2.1.3/pgi194
```

2. 編譯/連結 C 程式

```
[user@clogin1]$ mpicc –o sample_mpi.exe sample_mpi.c
```

3. 編譯/連結 C++ 程式

```
[user@clogin1]$ mpic++ –o sample_mpi.exe sample_mpi.c
```

4. 編譯/連結 Fortan 程式

```
[user@clogin1]$ mpifort –o sample_mpi.exe sample_mpi.f
```

5. 運行 PGI 函式庫編譯的 parallel 程式的 job script 範例：
   可參考第五章更詳細的 job script

```
#!/bin/bash
#PBS -P TRI654321
#PBS -N sample_job
#PBS -l select=2:ncpus=40:mpiprocs=40
#PBS -l walltime=00:30:00
#PBS -q ctest
#PBS -j oe
module load pgi/19.4
module load mpi/openmpi-2.1.3/pgi194
cd $PBS_O_WORKDIR
mpiexec –mca pml cm –mca mtl psm2 ./sample_mpi.exe
```

5. 操作 PBS PRO job

5.1 Job 佇列 (queue)

Queue 名稱	CPU 核心數範圍	節點總記憶體	SSD 資源	最長執行時間 (小時)	高優先權	每位用戶最多同時可執行 job 數量	最多可執行 job 數量
serial	1 (1 node)	384GB	1	96:00:00		10	120
cf40	2-40 (1 node)	384GB	1	96:00:00		10	200
cf160	2-160	384GB		96:00:00		4	160

	(1-4 nodes)						
cf1200	161-1200 (5-30 nodes)	384GB		48:00:00	V	2	5
ct160	2-160 (1-4 nodes)	192GB		96:00:00		4	200
ct400	161-400 (5-10 nodes)	192GB		96:00:00		3	22
ct800	401-800 (11-20 nodes)	192GB		72:00:00		2	10
ct2k	801-2000 (21-50 nodes)	192GB		24:00:00	V	2	4
ct6k	2001-6000 (51-150 nodes)	192GB		12:00:00	V	1	2
ctest	1-800 (1-20 nodes)	192GB		00:30:00		2	60
ct_ind	2-400	192GB		168:00:00	V		
cf_ind	2-160	384GB		72:00:00	V		

1.ct 開頭 Queue 是使用一般 192GB 記憶體 CPU 節點，cf 開頭 Queue 則是使用 384GB 大記憶體 CPU 節點，叢集內部 192GB 記憶體 CPU 節點有 562 台，384GB 大記憶體 CPU 節點僅有 188 台，若無高記憶體運算需求，選擇使用 ct 開頭 Queue 來派送計算工作，將可以減少排隊等待時間。

2.使用者於台灣杉一號上最多能提交 50 個計算工作、使用 6000 個 CPU 核心；然而各 queue 亦有限制用戶同時執行計算工作數目的上限，逾上限之計算工作需排隊等候。

3. Queue *cf40* 與 *serial* 所分配的計算節點，本機端具有一顆 SSD 硬碟 (容量大小爲 400GB)，適合單節點且需高速資料讀寫的計算工作使用，如 Gaussian 應用程式。使用者可在 select statement 中添加 `lscratch = ? gb` 來要求儲存空間，並透過 job script 中的環境變數 `$ PBS_JOBDIR` 存取此暫存目錄。

4.ct_ind 與 cf_ind 這二種 Queue 提供企業與個人計畫使用，需要額外付費提出申請，使用這二種 Queue 可減少排隊等候時間與即時執行計算。

5.計算工作的排隊演算法是採 FairShare 演算法，對於每一位使用者並無區分高低優先權 (priority)。有關 FairShare 演算法，請參考 https://en.wikipedia.org/wiki/Fair-share_scheduling。

5.2 Queue 列表

```
$ qstat -Q
Queue              Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
```

```
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
serial              0       0 yes yes     0     0     0     0     0     0    Exec
cf40                0       0 yes yes     0     0     0     0     0     0    Exec
cf160               0       0 yes yes     0     0     0     0     0     0    Exec
cf1200              0       0 yes yes     0     0     0     0     0     0    Exec
ct160               0       0 yes yes     0     0     0     0     0     0    Exec
ct400               0       0 yes yes     0     0     0     0     0     0    Exec
ct800               0       0 yes yes     0     0     0     0     0     0    Exec
ct2k                0       0 yes yes     0     0     0     0     0     0    Exec
ct6k                0       0 yes yes     0     0     0     0     0     0    Exec
ctest               0       0 yes yes     0     0     0     0     0     0    Exec
```

5.3 提交 job

在提交 job 之前請先確認您的計畫(計畫名稱)內有足夠的餘額：

```
$ get_su_balance
499023,TRI107693,試用計畫(ISSUE)
$ get_su_balance TRI107688
-150
$ qsub testjob.sh
qsub: No balance available for the User
```

5.3.1 PBS job script

PBS job 由以下三種要素構成：

1. Shell 的說明
2. PBS 的指令
3. 程式或指令

範例：

```
# Shell 說明
#!/bin/bash
# PBS 指令
#PBS -l walltime=00:30:00
#PBS -l select=2:ncpus=16:mpiprocs=16
#PBS -N sample_job
#PBS –q ctest
```

```
#PBS –P TRI654321
#PBS –j oe
# 程式與指令
cd $PBS_O_WORKDIR

module load intel/2018_u1
NODE=`cat $PBS_NODEFILE | wc`

mpiexec.hydra -PSM2 ./myprogram
```

以下詳細說明上述三個要素：

1. **Shell 的說明**—job script 首行爲 shell 的說明：

```
#!/bin/bash
```

2. **PBS 的指令**—使用者可設定 job 屬性，格式如下：

```
# 指定資源種類與數量
#PBS -l <resource name>=<value>
# 指定 job 名稱 (選擇性)
#PBS -N <job name>
# 指定 queue 名稱
#PBS –q <destination queue>
# 指定計畫名稱
#PBS –P <project name>
# 合併 std-err 與 std-out (選擇性)
#PBS -j eo
```

範例：

```
# 序列 job (1 核心)
#PBS -l select=1:ncpus=1
# MPI job (2 節點、每節點 8 個處理器)
#PBS -l select=2:ncpus=8:mpiprocs=8
# 結合 MPI 和 OpenMP 的 job (2 MPI 與 16 threads)
#PBS -l select=2:ncpus=8:mpiprocs=1:ompthreads=8
# 計算時間爲 1 小時
#PBS -l walltime=1:00:00
```

註：請務必設定 job 正確的資源上限數 (每節點 ncpus <= 40)

3. 程式與指令—Job script 的語法大致上皆與 shell 的說明語法相同，如下所示：

```
cd $PBS_O_WORKDIR

module load intel/2018_u1
NODE=`cat $PBS_NODEFILE | wc`

mpiexec.hydra -PSM2 ./myprogram
```

5.3.2 批次提交 job

PBS 的 qsub 指令可提交 job
批次 job 可藉由 (a) job script 提交，或 (b) 直接使用命令列提交，格式如下：

```
$ qsub <name of job script>
```

(a) 由 script 提交批次 job

1. 建立 job script 檔案：

```
$ vim example01.sh
#!/bin/bash
#PBS -P TRI107693
#PBS -N sample_job
#PBS -l select=2:ncpus=40:mpiprocs=40
#PBS -l walltime=00:30:00
#PBS -q ctest
#PBS -o jobresult.out
#PBS -e jobresult.err

module load intel/2018_u1
cd ${PBS_O_WORKDIR:-"."}

mpiexec.hydra -PSM2 ./myprogram
```

2. 提交 job：

```
$ qsub example01.sh
```

(b) 直接使用命令列設定 PBS 指令，並提交 job：

```
$ qsub -l select=1:ncpus=1 –q ctest –P TRI654321
-j oe ./example01.sh
```

5.3.3 提交 array job (bulk job)

藉由 PBS 提供的 Array 功能，您可藉由 script 一次提交一系列大量帶有類似的輸入值與輸出值的 job。提交時請使用「-J」選項與 qsub 指令。任何 job 都可以使用於 array job，以下是可用於 array job 的範例，且 script 內無指定 array 變數：

```
$ vim hello_mpi_1.sh
mpirun -np 80 /home/user/array/hello_mpi.exe
$ vim hello_mpi_2.sh
mpirun -np 80 /home/user/array/hello_mpi.exe
$ vim hello_mpi_3.sh
mpirun -np 80 /home/user/array/hello_mpi.exe

$ vim array.sh
#!/bin/bash
#PBS -l walltime=00:01:00
#PBS -l select=2:ncpus=4:mpiprocs=4
#PBS -N hello-mpi-array-job
#PBS -q ctest
#PBS -P TRI654321
#PBS -J 1-3
#PBS -j oe

echo "Main script: index " $PBS_ARRAY_INDEX
/home/user/array/hello_mpi_$PBS_ARRAY_INDEX.sh

$ qsub array.sh
```

以下提交一 array job (標記從 1 至 100)，作用等同於不使用「-J」選項，執行 100 次「qsub」：

```
$ qsub –J 1-100 example.sh
```

以下提交一 array job (標記從 200 至 400，間距爲 2，例：102、104、106…)

```
$ qsub -J 100-200:2 example.sh
```

5.3.4 Job script 設定 e-mail 通知

PBS 能夠寄送 e-mail 給指定的收件人，通知 job 已運行到特定的程度
設定此功能的兩個步驟：

1. 使用 PBS 指令 -M (大寫 M) 設定 e-mail 收件人：

```
#PBS –M user@example.com
```

2. 使用 PBS 指令 -m (小寫 m) 設定 job 運行的程度，到達便會寄送 e-mail 通知：

```
#PBS –m be
```

以下列出部分 e-mail 寄送的參數 (argument)：

E-mai 寄送參數	敘述
a	當 job 或 subjob 被批次處理系統中止時寄送 e-mail
b	當 job 或 subjob 開始執行計算時寄送 e-mail
e	當 job 或 subjob 結束計算時寄送 e-mail
n	不寄送 e-mail

以下是在 job script 內設定 e-mail 通知的範例：

```
#!/bin/bash
#PBS -P TRI654321
#PBS -N sample_job
#PBS -l select=2:ncpus=40:mpiprocs=40
#PBS -l walltime=00:30:00
#PBS -q ctest
#PBS -j oe
#PBS –M user@example.com
#PBS –m be

module load intel/2018_u1
cd ${PBS_O_WORKDIR:-"."}

mpiexec.hydra -PSM2 ./myprogram
```

5.4 刪除 job

PBS qdel 的指令可刪除 job。使用者只能刪除自己建立的 job。

格式：

```
$ qdel <job ID>
```

範例：

```
$ qdel 51
$ qdel 1234[].server
# 指令「qstat」可確認 job ID
```

可使用 qdel 指令與 -W force 選項，強迫刪除未完成的 job：

```
$ qdel –W force <job ID>
```

5.5 顯示 job 狀態

qstat 指令可監看 job 的狀態。S 欄內共有三種狀態：

(a) Job 狀態爲「Q」- Queueing: 正在「ctest」的 queue 中排隊計算

```
$ qstat –u user01
Job id                    Name             User            Time Use S Queue
------------------------------------------------------------------------------------------------------------------
12.localhost              example01        user01                 0 Q ctest
```

(b) Job 狀態爲「R」- Running：正在計算中

```
$ qstat –u user01
Job id                    Name             User            Time Use S Queue
------------------------------------------------------------------------------------------------------------------
12.localhost              example01        user01                 0 R ctest
```

(c) Job 狀態爲「C」- Completed：已計算完成

```
$ qstat –u user01
Job id                    Name             User            Time Use S Queue
------------------------------------------------------------------------------------------------------------------
12.localhost              example01        user01          00:00:55 C ctest
```