

台灣杉一號

Peta 級超級電腦

使用者操作手冊

版次：1.7

更新時間：2019/08/26



NAR Labs 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：呂小姐

電話：03 - 5776085 轉 442

Email：iservice@narlabs.org.tw

技術支援服務：

姓名：李先生

電話：04 - 24620202 轉 845

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	4
2.1 系統概觀	4
2.2 可用計算資源.....	4
2.3 可用儲存資源.....	5
2.3.1 家目錄區域 /home	5
2.3.2 暫存工作區域 /work1	5
2.3.3 計畫儲存區域.....	6
2.4 前端伺服器	6
2.4.1 登入節點	6
2.4.2 互動式節點	7
2.4.3 資料傳輸節點.....	7
3. 進入系統的方式	7
3.1 台灣杉一號帳號註冊	7
3.2 主機帳號、OTP 資訊	10
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	17
3.5.1 Linux 用戶	17
3.5.2 Windows 用戶.....	18
4. 編譯與連結.....	19
4.1 環境模組	19
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.....	21
4.2.4 MPI parallel 程式	21

4.3	PGI 編譯器.....	22
4.3.1	載入編譯器環境	22
4.3.2	序列程式	22
4.3.3	Thread parallel 程式.....	23
4.3.4	MPI parallel 程式	23
4.4	使用 CUDA 編譯	24
4.4.1	CUDA 程式.....	24
4.4.2	MPI parallel CUDA 程式.....	24
5.	操作 PBS PRO job	27
5.1	Job 佇列 (queue).....	27
5.2	Queue 列表	29
5.3	提交 job	29
5.3.1	PBS job script	30
5.3.2	批次提交 job	31
5.3.3	提交 array job (bulk job)	32
5.3.4	Job script 設定 e-mail 通知	33
5.4	刪除 job	34
5.5	顯示 job 狀態.....	34
5.6	建立和使用預訂.....	35
5.6.1	提前預訂	35
5.6.2	常設預訂	37
5.6.3	刪除預訂	38
6.	登入節點的資源限制.....	38

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

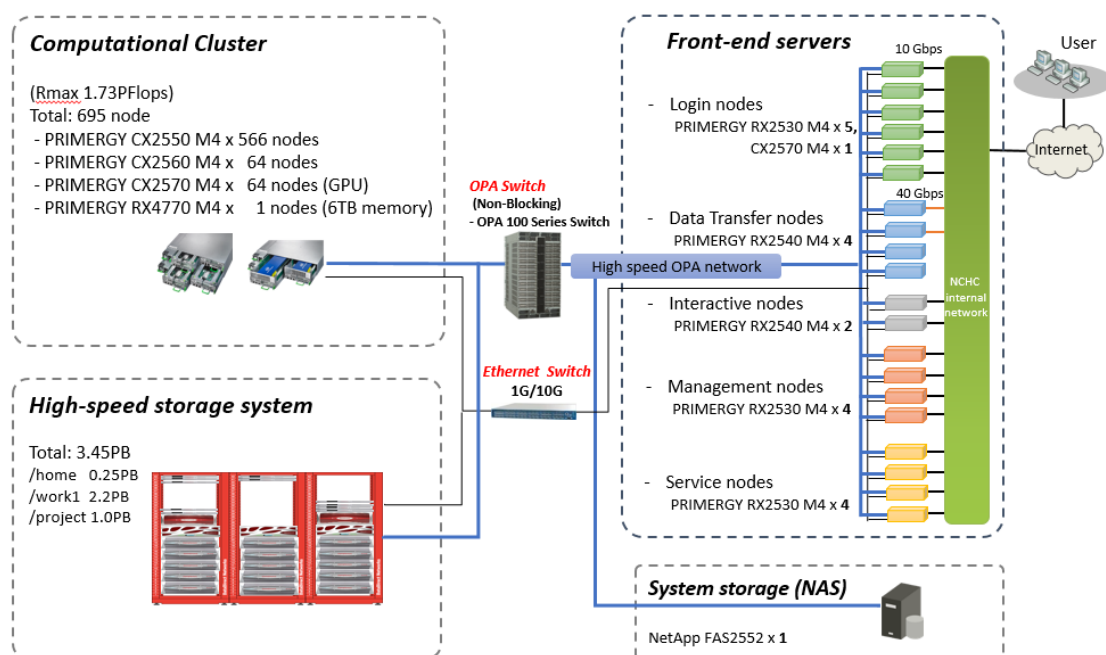
版次	日期	變更內容	變更者	檢閱者	審核者
0.1	2018/02/08	- 初版	Imura	Ishan	Yamada
1.0	2018/03/28	- 修訂 2.3 、4.1 、5.2 章節 - 新增第 6 章— 登入節點的資源限制	Imura	Ishan	Yamada
1.1	2018/04/12	- 修訂 4.2.4 章節— MPI parallel 程式 - 修訂 4.3.4 章節— MPI parallel CUDA 程式 - 新增 4.4.2 章節— MPI parallel CUDA 程式 - 新增 5.3.4 章節— 設定 e-mail 通知 - 新增 5.6 章節— 建立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

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 695 個計算節點 (1392 個處理器與 27856 個核心)，整體效能可達 1.73 兆次浮點運算 (PFLOPS)。694 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)；另 1 計算節點 (大記憶節點) 由四個插槽組成，每一插槽具有 Xeon Platinum 8160M CPU (24 核心、2.1 GHz)。

695 個節點可細分為以下類別：

- 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)：供需大量記憶體之 HPC 應用程式使用
- GPU 節點：供需 GPU 加速器的 HPC 應用程式使用
- 大記憶節點：供需大量記憶體之 HPC 特殊應用程式使用

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	-
胖節點	cn0801 – cn0864	64	2	40	384	-	-	-
胖節點	cn0901 – cn0964	64	2	40	384	-	-	1
GPU 節點	cn1001 – cn1064	64	2	40	192	4	-	-
大記憶節點	cnbm01	1	4	96	6000	-	-	-

2.3 可用儲存資源

下表為您在台灣杉一號可用的高速儲存系統資源。掛載為 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，主要是做為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

(待公告)

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

140.110.148.15 glogin1.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术規格，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
GPU 登入節點	glogin1	1	2	40	192	4	-

爲了確保最佳的效能，節點上的所有 GPU 都配置了獨佔模式，以隔離計算並避免在無意中讓同個 GPU 上執行多個計算。**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

(待公告)

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

LOGIN

[還不是會員? **現在就加入會員**] [重發認證信] [忘記密碼]

會員帳號(電子郵件地址)

請輸入您的電子郵件地址

密碼

密碼

☐ 我不是機器人 reCAPTCHA 隱私權-條款

登入

快速登入

f Facebook

g+ Google

3. 按步驟填寫基本資料—設定 iService 會員帳號與密碼

加入會員

閱讀個資及權利義務聲明

填寫會員基本資料

收取認證信

驗證成功

Step 1

Step 2

Step 3

Step 4

填寫會員基本資料

會員資料

主機帳號資料

會員帳號資料

*請輸入您的E-mail，做為登入服務網的會員帳號(或是可以選擇現有Facebook、Google 或EduRoam 帳號快速登入)

123456@gmail.com

因「確認信」為系統自動發出的關係，部分信箱會誤判為垃圾信，煩請您登入您的信箱並到「垃圾信件匣」查看是否有誤判之情形。

*會員密碼

*再次輸入會員密碼

連結 Facebook 帳號登入

連結 Google 帳號登入

連結 EduRoam 帳號

說明:

1. 若已成功連結 Facebook/Google/EduRoam 帳號登入，可不需輸入會員密碼，但仍需定期更改您所選擇綁定的帳號之密碼，以確保您的帳號之安全性。

2. 會員密碼長度至少需12字元，不可過於簡單

3. 會員密碼可為數字、英文字母(大小寫視為2種)、其他特殊字元等4種型式，至少須包含3種

4. 按步驟填寫基本資料—設定台灣杉一號主機帳號與密碼

填寫會員基本資料

會員資料

主機帳號資料

為了讓您體驗及熟悉主機之環境，特別貼心的為初次申請者，自動提供台灣杉一號HPC免費試用額度（不適用於台灣杉二號HPC）。未來如額度不敷使用時，敬請透過此服務網提出計畫申請及購買使用額度。

以下是您未來登入主機之帳號資訊，您可以自行命名或直接採用本中心為您取的主機帳號名稱，此帳號如建立後，不提供更名之服務。

主機帳號資料

*主機帳號

主機密碼

再次輸入密碼

請輸入主機帳號

請輸入主機密碼

請再次輸入主機密碼

產生主機帳號

說明:

1. 主機帳號長度8-12個字元，限定小寫英數混合，首字英文。

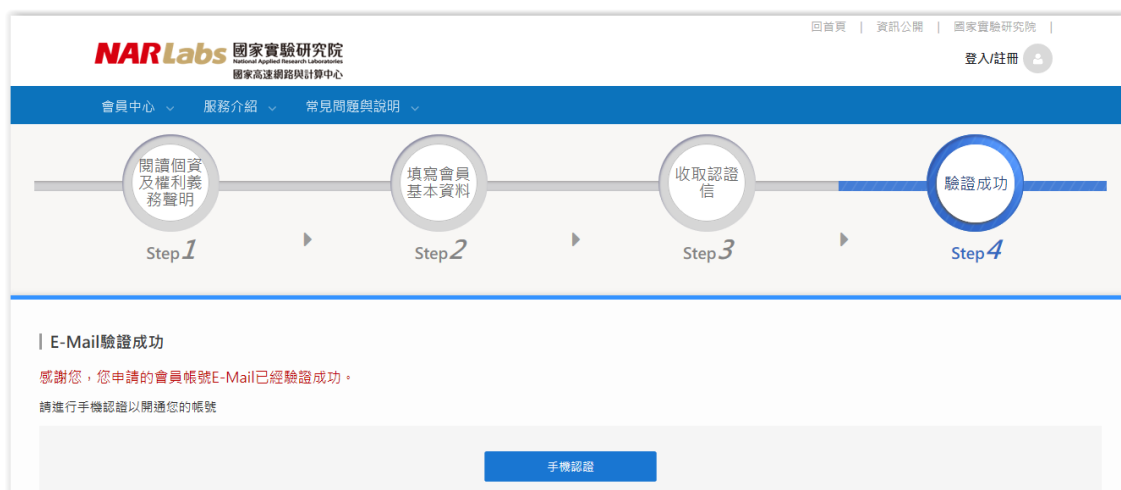
2. 主機密碼長度至少需12字元，不可過於簡單

3. 主機密碼可為數字、英文字母(大小寫視為2種)、其他特殊字元等4種型式，至少須包含3種

填寫會員資料

下一步

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！

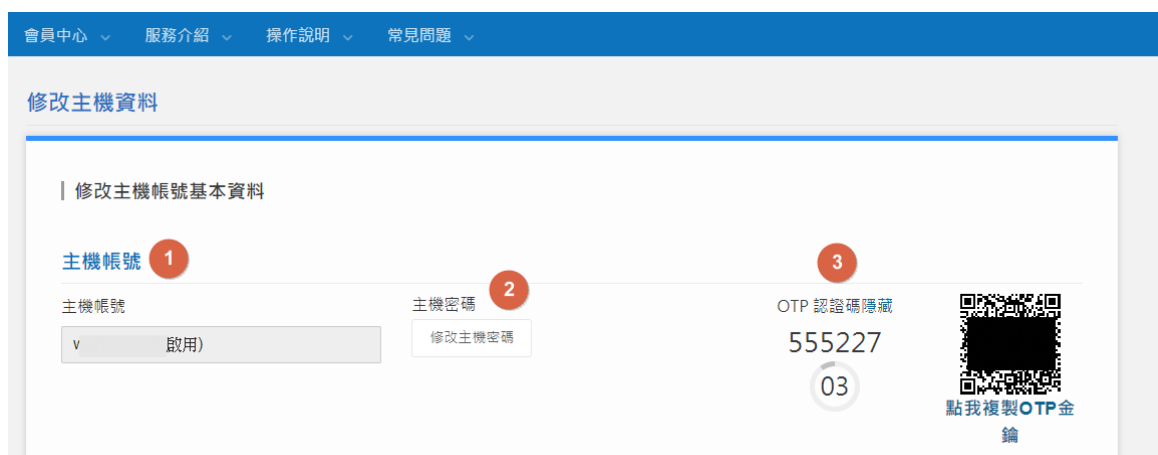


3.2 主機帳號、OTP 資訊

登入台灣杉一號之帳號為註冊 iService 時所設定的「主機帳號」，密碼為「主機密碼 + OTP」。OTP (One Time Password) 認證碼為一次性密碼，又稱為動態密碼。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。



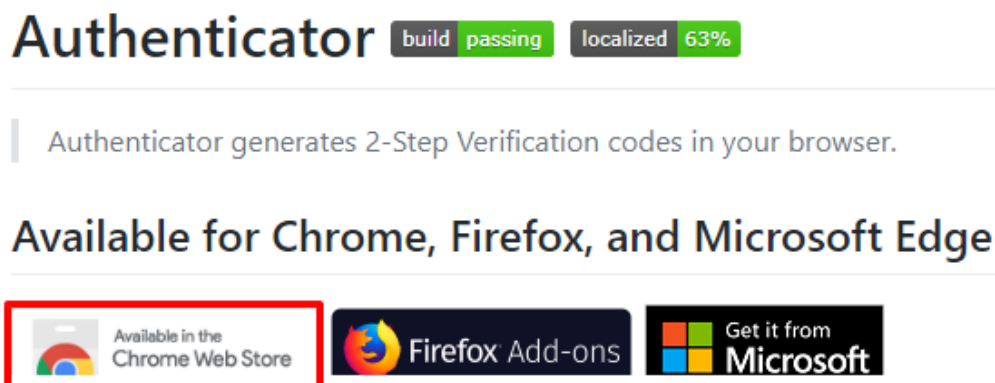
除了 Microsoft Authenticator，其他應用程式可參考：

<https://alternativeto.net/software/microsoft-authenticator/>

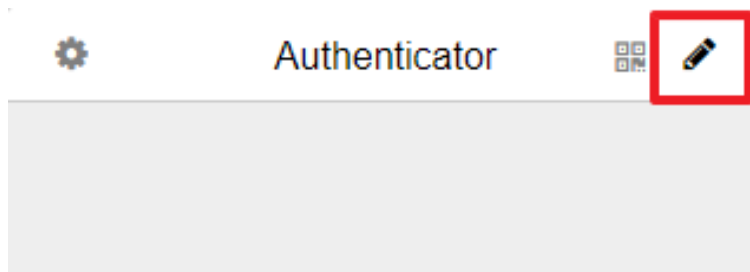
B. 電腦：

- a. 瀏覽器擴充功能，以下以 Google Chrome 為例：

- 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)



- 開啓 Authenticator，點選右上角的編輯



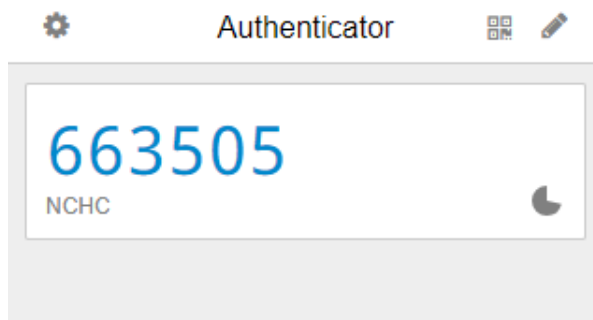
- 點選掃描 QR 碼



- 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



b. WinAuth

(僅限 Windows 用戶，其他系統請參考 <https://alternativeto.net/software/winauth/>)

- 至 <https://github.com/winauth/winauth> 下載 WinAuth

Download Latest Stable Version (WinAuth 3.5)

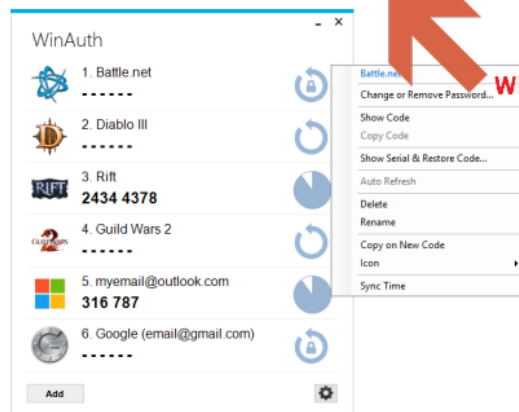
WinAuth provides an alternative solution to combine various two-factor authenticator services in one convenient place.

This is the latest stable version and can be downloaded from the [releases](#) page, or directly from:

[Latest Version \(WinAuth-3.5.1\)](#)

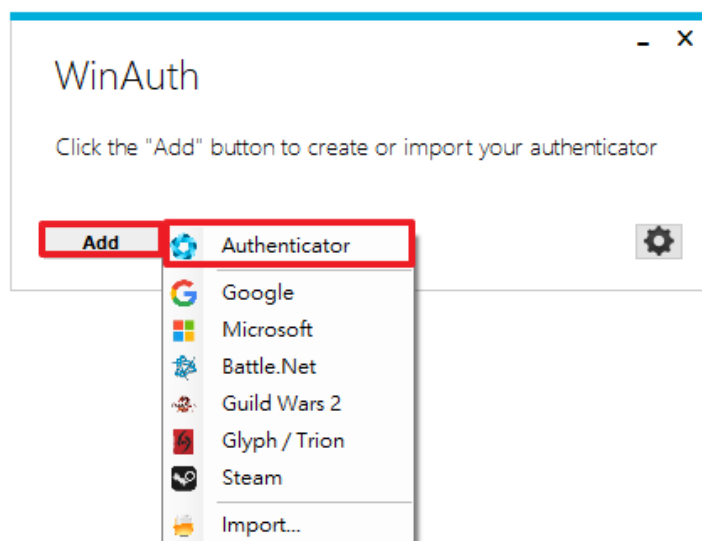
點擊下載最新版本

There is also a [Net 3.5 build of WinAuth](#) that can be run on Windows 7 installations "out of the box".



Windows 7 用戶請點此下載

- 解壓縮後，開啓、點選 Add、點選 Authenticator



- 建立帳戶



Add Authenticator

Name:

建立名稱，可含主機帳號以便辨認

1. Enter the Secret Code for your authenticator. Spaces don't matter. If you have a QR code, you can paste the URL of the image instead.

填入OTP，可於iService點選「點我複製OTP金鑰」並貼上

2. Choose if this is a time-based or a counter-based authenticator. If you don't know, it's likely time-based, so just leave the default choice.

☒ Time-based ☐ Counter-based **選擇Time-based**

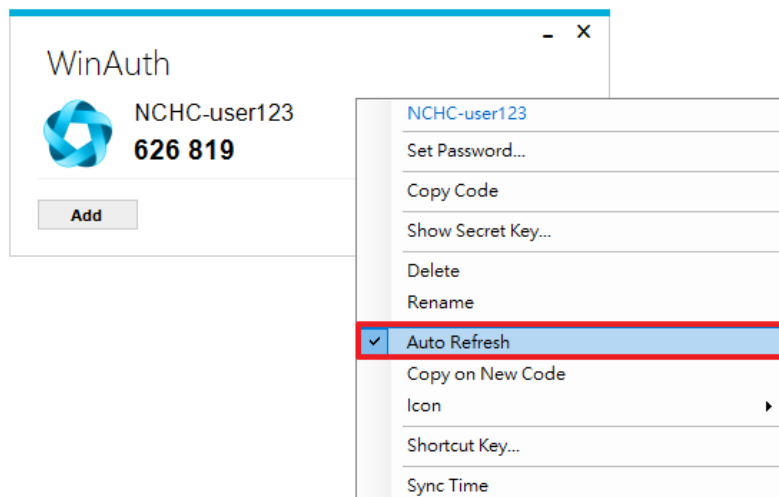
3. Click the Verify button to check the first code.

點擊Verify Authenticator

4. Verify the following code matches your service.

確認顯示的OTP與iService上的一致後，按OK儲存

- 再點右鍵 Auto Refresh，讓 OTP 自動刷新即完成！



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

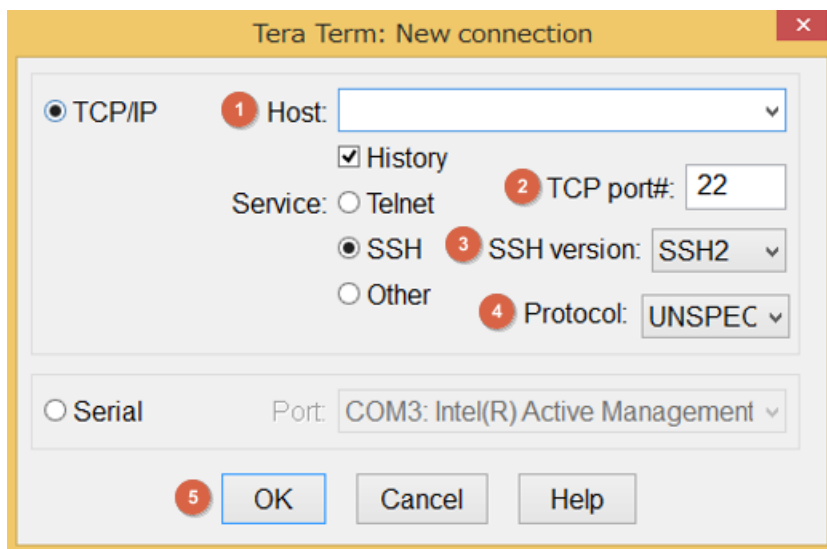
註：此非一次性密碼 (OTP)，為系統主機帳號的密碼。登入高速運算系統需使用此密碼與一次性密碼

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：TeraTerm) 並按照下列步驟，連線至台灣杉一號：

1. 在您的電腦開啓 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



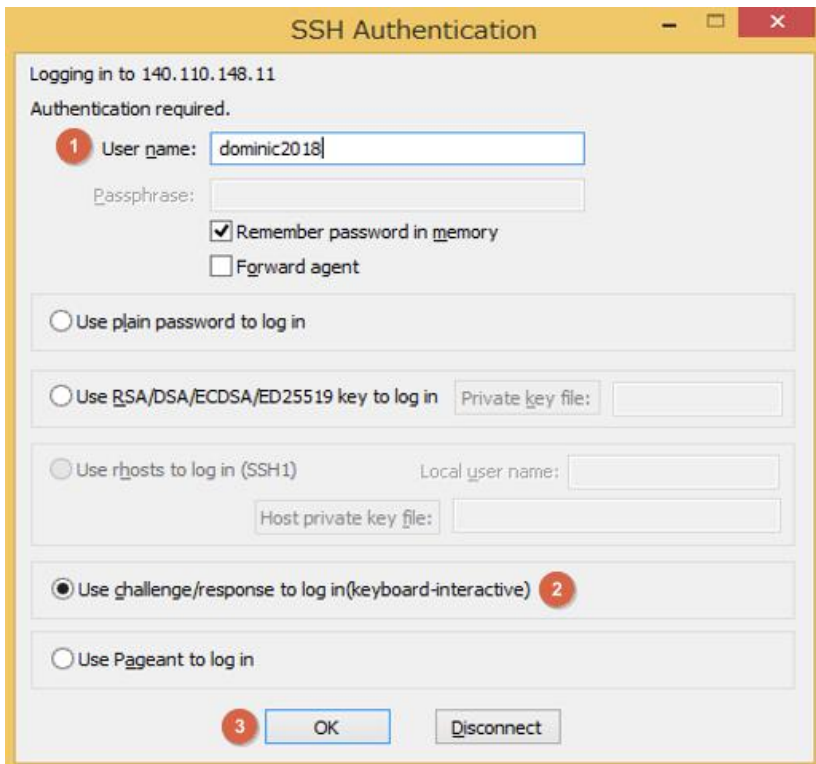
請輸入以下資訊：

- ① Host: 選擇以下任一 IP 輸入
 - 140.110.148.11
 - 140.110.148.12
 - 140.110.148.15
- ② TCP port: 22
- ③ SSH version: SSH2
- ④ Protocol: UNSPEC
- ⑤ 點選 OK

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「User name」欄位輸入您的主機帳號，點選「keyboard-interactive」，並按下「OK」



The image shows an 'SSH Authentication' dialog box. At the top, it says 'Logging in to 140.110.148.11' and 'Authentication required.' Below this, there are several input fields and options. A red circle with the number '1' points to the 'User name:' field, which contains the text 'dominic2018'. Below it is a 'Passphrase:' field. There are two checkboxes: 'Remember password in memory' (checked) and 'Forward agent' (unchecked). Below these are five radio button options for authentication methods. A red circle with the number '2' points to the 'Use challenge/response to log in(keyboard-interactive)' option, which is selected. At the bottom, there are two buttons: 'OK' and 'Disconnect'. A red circle with the number '3' points to the 'OK' button.

請輸入以下資訊：

- ① User name: 主機帳號

- 140.110.148.11

- 140.110.148.12

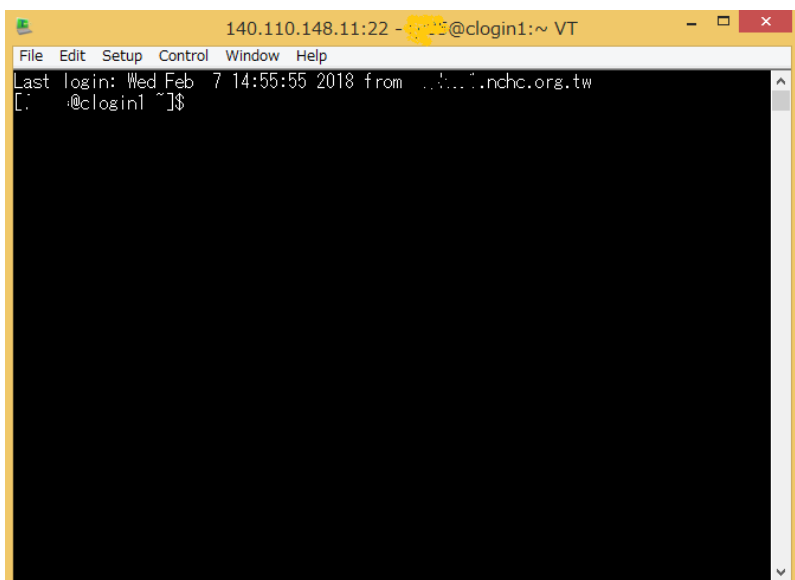
- 140.110.148.15

- ② 點選 keyboard-interactive

- ③ 點選 OK 以連線系統

4. 此時您需輸入密碼，請在輸入您的主機密碼之後，輸入一次性密碼 (OTP)，並按下「Enter」
密碼 = 帳號密碼 + 一次性密碼 (OTP)

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：

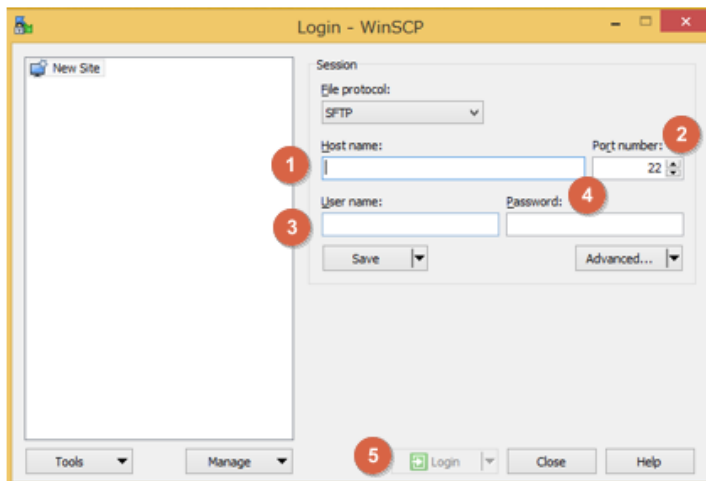
- p 保存修改的時間、讀取的時間、原始檔案的類型
- r 複製個目錄 (含子目錄)

其他主要 sftp 可用的內建指令：

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

3.5.2 Windows 用戶

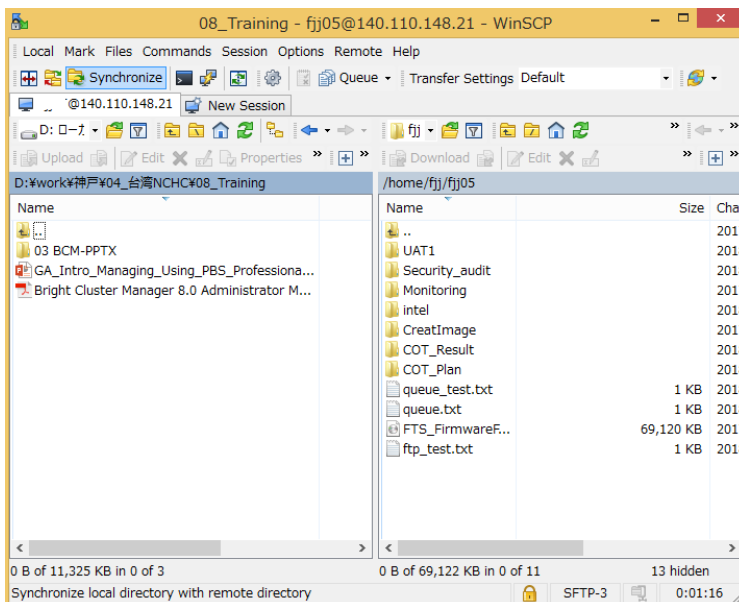
開啓 WinSCP 並連接系統任一資料傳輸節點。連接進入後，您即可用拖曳方式傳輸檔案
 以下是 WinSCP 的登入視窗：



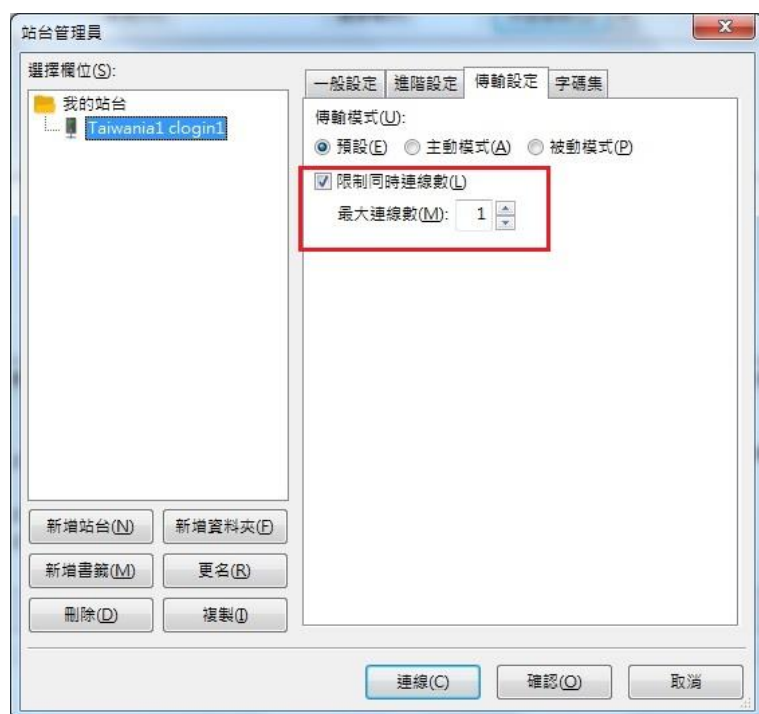
請輸入以下資訊：

- ①Host name：選擇以下任一 IP 輸入
 - 140.110.148.21
 - 140.110.148.22
- ②Port number：22
- ③User name：主機帳號
- ④Password：主機密碼 + 一次性密碼 (OTP)
- ⑤按下 login 按鈕

連線成功後，WinSCP 視窗會呈現如下：



若使用 FileZilla FTP 用戶端傳輸檔案，需先至「站台管理員」—「傳輸設定」— 設定「限制同時連線數」為 1，如下所示：



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. 以下為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
cuda/8.0.61	CUDA library for GPU
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：

```
$ vim example01.sh
#!/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
mpirun ./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 Fortran 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 Fortran 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 openmpi/pgi/2.1.2/2017
```

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 Fortran 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

```
$ vim example01.sh
#!/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 pgi/17.10
module load openmpi/pgi/2.1.2/2017
cd ${PBS_O_WORKDIR:-"."}
mpirun ./sample_mpi.exe
```

4.4 使用 CUDA 編譯

4.4.1 CUDA 程式

1. 載入 CUDA 編譯器環境：

```
# 清空環境變數模組，確保能順利載入 cuda 程式模組
[user@glogin1]$ module purge
# 載入 cuda 程式模組
[user@glogin1]$ module load cuda/8.0.61
```

選擇可搭配該版本使用的模組

2. 編譯/連結 CUDA C 程式

```
[user@glogin1]$ nvcc -o sample.exe sample.cu
```

原始檔的副檔名必須是「.cu」

3. 編譯/連結 CUDA C++ 程式

```
[user@glogin1]$ nvcc -o sample.exe sample.cu
```

原始檔的副檔名必須是「.cu」

4. 編譯/連結 CUDA Fortran 程式

```
[user@glogin1]$ module add pgi
[user@glogin1]$ pgfortran -o sample.exe sample.cuf
```

原始檔的副檔名必須是「.cuf」

4.4.2 MPI parallel CUDA 程式

以下是利用 OpenMPI 編譯和運行 CUDA MPI 程式的範例。例如要編譯兩個程式 simpleMPI.cu (CUDA 原始檔)與 simpleMPI.cpp (C++ 原始檔)

編譯：

1. 載入 OpenMPI 與 CUDA 環境模組：

```
$ module load openmpi/gcc/64/1.10.4
$ module load cuda/8.0.61
```

2. 確認模組有載入成功：

```
$ module list
Currently Loaded Modulefiles:
  1) openmpi/gcc/64/1.10.4  2) cuda/8.0.61
```

3. 確認您有載入正確的 nvcc 與 mpic++ 指令：

```
$ which nvcc
/pkg/cuda/8.0.61/bin/nvcc
$ which mpic++
/usr/mpi/gcc/openmpi-1.10.4-hfi/bin/mpic++
```

4. 使用 nvcc 編譯 simpleMPI.cu：

```
$ nvcc -c simpleMPI.cu -o simpleMPI_cuda.o
```

5. 使用 mpic++ 編譯 simpleMPI.cu：

```
mpic++ -c simpleMPI.cpp -o simpleMPI_mpi.o
```

6. 確認編譯程式都有建立以下檔案：

```
$ ls *.o
simpleMPI_cuda.o
simpleMPI_mpi.o
```

7. 連結以上兩個檔案：

```
$ mpic++ -o simpleMPI simpleMPI_cuda.o simpleMPI_mpi.o -L/pkg/cuda/8.0.61/lib64 -lcudart
```

8. 確認 simpleMPI 已建立並可被執行：

```
$ ls simpleMPI
simpleMPI
```

建立 job script：

1. 台灣杉一號的每個 GPU 節點皆共安裝了 4 個 GPU。請在執行 PBS job 前，先執行以下將計算工作分配給不同 GPU 處理器的 script，每一節點最多運行四個 GPU 執行計算。

註：您也可在 CUDA 程式中指定使用 CUDA 設備。若您如此設定，便不用建立以下的 script。

將以上已可執行的 simpleMPI 設定於以下的 script 中：

```
$ vim run_4gpu.sh
#!/bin/bash

#load cuda library
module load cuda/8.0.61
#location of Binary
EXEC_DIR=`pwd`
APP=$EXEC_DIR/simpleMPI
lrank=$OMPI_COMM_WORLD_LOCAL_RANK
case ${lrank} in
[0])
    export CUDA_VISIBLE_DEVICES=0
    $APP
    ;;
[1])
    export CUDA_VISIBLE_DEVICES=1
    $APP
    ;;
[2])
    export CUDA_VISIBLE_DEVICES=2
    $APP
    ;;
[3])
    export CUDA_VISIBLE_DEVICES=3
    $APP
    ;;
esac
```

2. 接著在您的 PBS job 中呼叫以上的 script run_4gpu.sh，如下所示。

PBS job 的細節將會在第五章詳細說明。

請記得將 MPI 的處理器數量設定為 4 (mpiprocs = 4)，每個節點至多分配到四個運算。若您將數量設定大於 4，job 將不能執行計算。

```
$ vim go.sh
```

```
#!/bin/bash
#PBS -P TRI654321
#PBS -N sample_cuda_mpi_job
#PBS -l select=2:ncpus=40:ngpus=4:mpiprocs=4
#PBS -l walltime=00:10:00
#PBS -q gp16
#PBS -j oe

module load openmpi/gcc/64/1.10.4
module load cuda/8.0.61
cd ${PBS_O_WORKDIR:-"."}

mpirun -np 8 -hostfile $PBS_NODEFILE
    --mca pml cm --mca mtl psm2
    ./run_4gpu.sh
```

5. 操作 PBS PRO job

5.1 Job 佇列 (queue)

Queue 名稱	CPU 核心數 範圍	GPU 個 數範圍	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	節點類型	適用計畫
ct6k	2001-6000		24	V	1	2	一般 CPU 節點	各式計畫
ct2k	801-2000		48	V	2	4	一般 CPU 節點	各式計畫
ct800	401-800		72		2	10	一般 CPU 節點	各式計畫
ct400	161-400		96		3	23	一般 CPU 節點	各式計畫
ct160	2-160		96		4	62	一般 CPU 節點	各式計畫
serial	1		96		10	120	一般 CPU 節點	各式計畫
ctest	1-80		0.5		10	60	一般	各式計畫

							CPU 節點	
cf1200	161-1200		48	V	2	5	大記憶體 CPU 節點	各式計畫
cf160	2-160		96		4	140	大記憶體 CPU 節點	各式計畫
cf40	2-40		96		10	120	大記憶體 CPU 節點	各式計畫
gp32		17-32	48		1	1	GPU 計算節點	各式計畫
gp16		5-16	96		1	3	GPU 計算節點	各式計畫
gp4		1-4	96		2	10	GPU 計算節點	各式計畫
gtest		1-8	0.5		2	10	GPU 計算節點	各式計畫
ct_ind	2-400		72	V			一般 CPU 節點	企業與 個人計畫
gp_ind		1-16	72	V			GPU 計算節點	企業與 個人計畫

1. 一般 CPU 節點具 192GB 記憶體，大記憶體 CPU 節點具 384GB 記憶體；每一節點各有 Intel Xeon Gold 6148 CPU 2 顆，共計 40 核心。
2. GPU 計算節點具 4 張 NVIDIA P100，其餘硬體規格皆與一般計算節點相同。
3. 使用者於台灣衫一號上最多能提交 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。
4. Queue *cf40* 與 *serial* 所分配的計算節點，本機端具有一顆 SSD 硬碟（最大大小為 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。
5. 若計算工作無須用到 GPU 時，請勿使用 *gtest*、*gp4*、*gp16* 的 queue。若違反，系統管理員將會停止該筆計算工作，避免不合適的計算工作占用有限的 GPU 計算資源。
6. 欲使用 GPU 計算工作，請於 job script 加入以下資源的組合：

```
ncpus=10:ngpus=1, ncpus=20:ngpus=2, ncpus=30:ngpus=3, or ncpus=40:ngpus=4
```

原則上 1 個 GPU 最多搭配 10 個 CPU，若 CPU 用量逾此比例，請適度增加 GPU 數量，否則系統不予接受。

7. 收費費率請參考[台灣杉一號\(Taiwania 1\)的費率](#)。

8. *ct_ind*與 *gp_ind* 兩種 Queue 提供企業與個人計畫使用，可減少排隊等候時間與即時執行計算工作，如須使用者，可透過電子郵件方式通知客服(iservice@narlabs.org.tw)，經設定權限後即可使用。

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
gp4	0	0	yes	yes	0	0	0	0	0	0	Exec
gp16	0	0	yes	yes	0	0	0	0	0	0	Exec
gp32	0	0	yes	yes	0	0	0	0	0	0	Exec
gtest	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=1:00:00
#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`

mpirun ./myprogram
```

以下詳細說明上述三個要素：

1. Shell 的說明—job script 首行為 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
# 結合 MPI 和 CUDA 的 job (每節點 4 GPUs)
#PBS -l select=2:ncpus=40:ngpus=4:mpiprocs=4
# 計算時間為 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 $\text{ncpus} \leq 40$, $\text{ngpus} \leq 4$)

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`

mpirun ./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:-"."}

mpirun ./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，間距為 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：

```
$ vim example01.sh
```

```
#!/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:-"."}

mpirun ./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 狀態為「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 狀態為「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 狀態為「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
```

5.6 建立和使用預訂

此章節僅說明一些 PBS 預訂的基本功能，詳細說明請參考以下線上 PBS 文件：

- [PBS PRO 使用者手冊](#)
- [PBS PRO 參考手冊](#)

PBS 能輸入兩種預訂：提前預訂與常設預訂。`pbs_rsub` 指令可建立以上兩種預訂。當預訂建立完成，相關特定的佇列種類也已建立，使用者可以此佇列類別提交 job。

5.6.1 提前預訂

提前預訂可保留特定時間區段的資源。`pbs_rsub` 指令可回傳預訂的 ID 名稱，提前預訂的 ID 有以下格式：

R<unique integer>.<server name>

提前預訂可使用以下三種 `pbs_rsub` 指令的選項，建立預訂的資訊：

- D：時長 (Duration)
- E：結束時間 (End time)
- R：開始時間 (Start time)

建立提前預訂的範例：

- 建立提前預訂的開始時間、時長：

```
pbs_rsub -R 1130 -D 00:30:00
```

- 建立提前預訂的開始時間、結束時間：

```
pbs_rsub -R 2000 -E 2200
```

以下說明提前預訂的作業方式：

- 以下是建立提前預訂的指令，每項皆搭配使用 `ncpus=40` 的資源，時間設定 16:34 開始執行計算，17:00 結束。當執行 `pbs_rsub` 指令時，不會顯示出預訂的名稱，但狀態會顯示「未確認」。不會立即顯示該預訂是已確認建立或是拒絕建立的狀態。

```
$ pbs_rsub -R 1634 -E 1700 -l select=2:ncpus=40
R8985.srvc1 UNCONFIRMED
```

- 預訂的狀態可用 `pbs_rstat` 指令確認，以下列有建立預訂的資訊：

```
$ pbs_rstat
Resv ID   Queue   User      State      Start / Duration / End
-----
R8969.srvc R8969   user1234 RN         Today 16:23 / 1620 / Today 16:50
R8985.srvc R8985   user1234 CO         Today 16:34 / 1560 / Today 17:00
```

以下列有幾項重要的預訂狀態：

代碼	狀態	描述
UN	RESV_UNCONFIRMED	預訂尚未確認
CO	RESV_CONFIRMED	預訂已確認
RN	RESV_RUNNING	預訂的時間已被占用

有關預訂特定 queue 的詳細訊息，可使用 `qstat` 指令確認，如下所示：

請注意，此 queue 有定義 `acl_users`，並且只有指定的用戶可以使用此 queue 提交 job。

```
$ qstat -Qf R8985
Queue: R8985
  queue_type = Execution
  total_jobs = 0
  state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0 Exiting:0 Begun
               :0
  acl_user_enable = True
```

```

acl_users = user1234@clogin4
resources_max.ncpus = 80
resources_max.walltime = 00:26:00
resources_available.ncpus = 80
resources_available.walltime = 00:26:00
resources_assigned.mpi_procs = 0
resources_assigned.ncpus = 0
resources_assigned.nodect = 0
enabled = True
started = True

```

包含在 `acl_users` 定義內的用戶，可以使用以下的 queue 建立 job：

```

#PBS -P sample-project
#PBS -N jobname
#PBS -l select=2:ncpus=40:mpi_procs=40
#PBS -l walltime=00:10:00
#PBS -q R8985
#PBS -j oe

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

mpirun ./myprogram

```

5.6.2 常設預訂

常設預訂是某個時間區段常態保留特定的資源。常設預訂的 ID 名稱代表一整個系列的預訂，ID 名稱有以下格式：

`S<unique integer>.<server name>`

每天早上 8 點 ~ 10 點的常設預訂，共執行 10 次，設定如下：

```
pbs_rsub -R 0800 -E 1000 -r "FREQ=DAILY;COUNT=10"
```

每個週間日從早上 6 點 ~ 晚上 6 點執行常設預訂，至 2008 年 12 月 10 日，設定如下：

```
pbs_rsub -R 0600 -E 1800 -r "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR;UNTIL=20081210"
```


每週一、三、五的下午 3 點 ~ 5 點執行計算，為期 3 週，共 9 次，設定如下：

```
pbs_rsub -R 1500 -E 1700 -r "FREQ=WEEKLY;BYDAY=MO,WE,FR;COUNT=9"
```

「確認預訂」和「使用特定 queue」的方式與提前預訂的方式相同。

5.6.3 刪除預訂

可以使用 `pbs_rdel` 刪除提前預訂或常設預訂。刪除常設預訂只能刪除一系列的預訂，不能刪除單一預訂。刪除時，已提交的 job 也會一併被刪除。除了預訂的建立者本身，PBS 的維運者或管理者也都都有權刪除預訂。

刪除預訂的範例：

```
$ pbs_rdel R8969.srv
```

更簡單的方式：

```
$ pbs_rdel R8969
```

6. 登入節點的資源限制

基本上使用者都應透過使用 PBS 將計算工作派送到計算節點上執行。但為了測試之目的，使用者也能在節點上編寫程式與執行一些測試。為了防止使用者執行超過一小時的計算，在登入節點上設有使用限制，超過時限的計算將強行刪除。

	資源	限制	描述
1	CPU 使用時間上限	1 小時	此計算已強行刪除