

視覺化 TWAREN 高速資料傳輸網路系統之效能優化機制實作

周大源 楊哲男 古立其 劉德隆

財團法人國家實驗研究院國家高速網路與計算中心

E-mail: {1203053, yangen, lku, tlliu}@narlabs.org.tw

摘要

本論文展示視覺化 TWAREN 高速資料傳輸網路 (Data Transfer Network, DTN) 系統之效能優化機制實作。在硬體部份, 利用 TWAREN 的骨幹線路, 我們將六個存取節點, 包含台大、清大、交大、成大、國網中心新竹本部、國網中心台中分部串接起來, 使得連線單位能得到更快的傳輸速率。在軟體部份, 我們採用廣為應用的高速傳輸工具 Fast Data Transfer (FDT) 與 GridFTP, 使得傳輸頻寬運用更加有效。考慮使用上的便利性, 本團隊已開發一套視覺化系統來取代命令列介面 (Command Line Interface, CLI) 的輸入。由於上述的高速傳輸工具有提供多項可調式參數。為了要能夠讓高速傳輸系統優化, 我們提供給使用者能夠手動調控傳輸參數的介面。藉以根據不同網路環境提供最佳的傳輸參數。

關鍵詞: 大資料傳輸, 資料傳輸網路, 台灣高品質學術研究網路

Abstract

This paper demonstrates the construction of a visualized system of Data Transfer Network (DTN) on TWAREN. On the hardware aspect, via the TWAREN backbone, we connect 6 access nodes, including National Taiwan University, National Tsinghua University, National Chiaotung University, National Chengkung University, Hsinchu Headquarters of NCHC, and Taichung Branch of NCHC to yield high-speed data transfer. On the software aspect, we adopt widely-used tools, such as Fast Data Transfer (FDT) and GridFTP so that effective utilization of network bandwidth can be yielded. For the sake of convenience, we develop a visualized system instead of Command Line Interface (CLI) for users. Via the proposed system, which can be easily learned and used, users can transfer big data in high-speed so that the research power of all participants can be integrated.

Keywords: Big Data Transfer, Data Transfer Network, DTN, Grid FTP, Fast Data Transfer, TWAREN

1. 簡介

近年來, 人類的生活與電子產品、網際網路等等高科技技術愈來愈密不可分。依據 Gartner[1] 的分析與預測, 在 2018 年含電腦、平板、手機等等的銷售量都會成長 2.1%。各大產業為了要能夠迎合使用者的喜好, 紛紛推出諸多客製化服務、記錄使用者喜好資訊、識別使用者位置資訊等

等, 藉以抓住更多商機。許多產業紛紛投入大量成本, 希望能夠將人工智慧導入產業中, 希望透過更高階的方式來為產業解決問題。而人工智慧的基礎, 往往是要先針對大量的資料進行分析、訓練, 讓機器能夠進一步進行深度學習。

同時, 隨著全球氣候變遷、天文觀測、國土觀測、農業觀測、... 等等議題愈來愈重要, 目前全球有許多領域都需要大量資料來進行分析。許多重大議題已非單一領域、單一團隊能夠獨立完成。因此, 以資料、資源共享為基礎的跨團隊合作之研究日趨重要。隨著資訊技術的發展, 計算資源也比以往更加進步, 問題分析時往往能夠針對更精細的資料進行分析。資料量的成長從 Giga 為單位, 逐漸躍升為 Tera 與 Peta 等級。

為了要能夠傳輸大量的研究資料, 採用一般的網際網路來進行, 是最簡單的方法。利用傳統的 FTP、HTTP 等方式都可以進行資料的上傳、下載。然而, 在傳輸巨量資料時, 在資料傳輸速率上恐怕會不敷使用。倘若使用者較多, 或者傳輸的資料量過大, 還會發生壅塞的情況。另外, 透過公眾通道有可能會產生資料外洩等等資訊安全疑慮。因此, 建置資料傳輸網路將是相當有效的解決方案。

資料傳輸網路 (Data Transfer Network, DTN) 是以高速的專用線路將多個存取節點 (Access Nodes) 連接在一起。其間所串接的專用線路往往是以專線、VPLS、VPN、tunneling 技術等等進行串接。透過上述的專用線路與一般公眾網路區隔, 可以避免產生與公眾網路共用傳輸線路時的壅塞狀況, 而且能夠排除大部份資料外洩的因素。

在資料傳輸網路中, 需要考慮兩部份:

- 骨幹線路: 連接存取節點互相連接的部份
- 最後一哩 (last mile): 使用者端至存取節點部份。

在硬體的骨幹線路部份, 我們主要將 DTN 建置於台灣先進學術研究網路 (Taiwan Advanced Research and Education Network, TWAREN) [2] 上。透過 2016 年升級完成的 100G 學研網路[3], 可以大幅降低線路建置的成本。而在最後一哩部份, 一般都是校園網路部份, 在傳輸上的頻寬往往有所限制。因此, 現階段而言, 我們優先以連接存取節點的骨幹網路部份為主要考量。

在軟體的傳輸工具部份, 目前有許多商用或非商用的高速傳輸工具, 在[4]中有針對常見工具進行比較。本系統初步採用其中兩項工具, Fast Data Transfer [5]與 GridFTP [6] 作為傳輸工具。Fast Data Transfer 的特性是以 Java 實作, 具有跨平台的特性。不管使用者的電腦是 Linux 或是 Windows 作業系統, 都能夠執行。而 GridFTP 的

特性是提供多種組態選項，能夠針對不同的網路傳輸工作進行效能的優化。然而，以最單純的自由軟體版本而言，目前主要是在 UNIX-like 的環境下執行。

然而，Fast Data Transfer 與 GridFTP 兩項軟體本身的操作方式皆是以命令列介面 (Command Line Interface, CLI) 操作。使用者需要熟悉作業系統中檔案與目錄的操作，並且記憶許多複雜指令。在輸入上傳/下載指令時，使用者必須手動輸入額外檔案與路徑識別碼資料。一旦輸入錯誤，便會造成指令失敗，必須重新輸入。隨著待傳輸之檔案數變多，亦未能使用 tar 工具將大量檔案打包成單一 archive，傳輸效率則會愈降低。這對於不熟悉 CLI 操作的使用者而言，使用這些工具將會成為相當高的門檻。

再者，由於 GridFTP 工具本身有多項認證方式，其中一項就是透過主機的 SSH 方式進行帳號認證。若是 DTN 的使用者增多，在各大存取節點上的帳號管理、存取權限等等皆需要有良好的管理。由於這些工具本身並不會記載用戶相關資訊，故一套有效的帳號管理機制是相當需要的。

本論文擬以 WebUI 方式建置 Fast Data Transfer 與 GridFTP 為基礎的高速傳輸系統。本系統以 LAMP 架構 (包含 Linux 作業系統、Apache 伺服器、MySQL 資料庫、PHP 程式語言) 來為基礎，協助使用者在記憶最少指令的狀況下，更有效地協助使用者進行資料傳輸。結合 MySQL 資料庫，可以記載用戶的帳號資訊、權限資訊等等。

本論文的組織架構如下。第 2 節說明相關的技術。第 3 節說明本系統的架構與實作方式。第 4 節展示系統的畫面。第 5 節展示實驗結果。而第 6 節是我們的結論與未來展望。

2. 相關技術

2.1 高速傳輸技術

針對 MB 等級至 GB 等級的資料，一般常用的資料傳送方法都可以運作正常。然而，當有很大的資料量要傳送來傳送，就得考慮幾個問題，包含封包延遲、TCP 或 UDP 傳輸，或者以平行化 TCP 的方式傳輸。

一般 Ping time 是用以測量到某主機來回一輪的時間 (roundtrip time, RTT)，亦即封包到達目的地，且傳送端收到回應的時間。隨著地理上的距離與網路架構的不同，依據 traceroute 命令，一個 ping 指令需要經過的裝置數並不相同。因此，其延遲就是線路和裝置所造成的封包延遲，會將光傳輸的速率降低。

上述的 RTT 是相當重要的，因為 TCP 協定的運作原理是要確認每一個 packet 都有送達，而每個 packet 都需要一個網路的來回傳輸 (與回應) 時間。如果 ping time 變長，顯然在每個單位時間中可以確認的 packet 數量就變少了。因此，TCP 在長距離網路環境運作比較緩慢。但若能傳輸一串 TCP 封包的串流，含 FTP 等等協定，當 ping time 增加時，則送封包、接收確認或其他資料等等的傳輸速率便會減少。這個限制就是一般認知的頻寬時延乘積 (Bandwidth-delay product)，會讓長距離網路上的效能降低。

有一些方法可以跳過或改善這個問題。如果

在傳輸前針對資料進行壓縮，就可以在相同的封包數量中傳輸更多的資料內容。另外，也可以嘗試增加封包大小，稱之為最大傳輸單元 (Maximum Transmission Unit, MTU)。然而，許多商用的 Internet 產品都將 MTU 設定得相當低 (1500 位元組)，儘管某些高速或學研網路時常可以允許巨大封包 (jumbo frame)，讓 MTUs 的有效負載提升至 9000 位元組。區域網路的 MTUs 則可以高到 64K 位元組，允許更有效的傳輸。

另外，可以用 UDP 來取代 TCP。這是相當不可靠的傳輸機制，因為它並沒有確認封包是否送達與送達順序。然而，因為較新的硬體底層愈來愈可靠，因此 UDP 又重新被使用來傳輸資料，並搭配 TCP 進行一致性檢查。因此，巨大封包 (64K 以上) 可以傳送得相當快。

而最後一個解決長距離資料傳輸限制的方法，就是同時傳輸多個 TCP 串流。這種平行傳輸的機制逐漸廣為被使用，或者是將現有的應用包裝而達成這種機制。

2.2 Fast Data Transfer

Fast Data Transfer (FDT) 是開源 (Open Source) 的巨量資料傳輸工具。一般使用者均可自由下載與使用。原始的 FDT 套件就是使用 Java 所開發完成的。

FDT 可以三種模式運行，包含伺服器端 (Server)、客戶端 (Client)，以及 Secure Copy (SCP) 三種模式。使用方式分別為：

- Server 模式：
java -jar fdt.jar [選項]
- Client 模式：
java -jar fdt.jar [選項] -c <host> [file1 ...][[-fl <檔案串列>] -d <目標路徑>
- SCP 模式：
java -jar fdt.jar [選項] [[[使用者@][主機 1:]]檔案 1[[[使用者帳號@][主機 2:]]檔案 2

上述之伺服器端模式中，當 fdt.jar 執行時，就會開始監聽是否有客戶端連線到本機。在最後一個連線結束前，伺服器端不會結束連線。

而在客戶端模式中，當 fdt.jar 執行時，就可以連向指定的伺服器端。該伺服器端必須要有 FDT 的伺服器端正在執行。而客戶端可以針對伺服器端讀取或寫入資料。

另一方面，如果是在 SCP 的模式下，本機的 FDT 會使用 SSH 來啟動 FDT 的客戶端與伺服器端。其安全性是借有 SSH 的認證來達成。而 SCP 的伺服器只會接受來自 SCP 客戶端的連線。若要限制 FDT 伺服器端的連線主機，可以使用 -f 選項。這個選項可以接受一串 IP 位址，以「:」來分隔。

FDT 是利用 Java 程式語言開發，使用者只要擁有 Java 執行環境 (Java Runtime Environment, JRE) 即可執行。雖然 FDT 本身是開源的軟體，針對大檔案傳輸時的效能可以較 FTP 提昇 5 倍至 10 倍。

2.3 GridFTP

GridFTP 由 Globus.org 所提供，包含於 Globus Toolkit 中。Globus Toolkit 以 C 語言實作，目前提供許多作業系統版本的安裝檔。

針對單一檔案，一般使用指令 `globus-url-copy` 的格式如下：

```
globus-url-copy -vb -p 4 <來源url> <目的url>
```

其中，`-vb` 選項用以顯示詳細資訊，包含傳輸的位元組數、自上次更新時的效能（目前是每 5 秒 1 次）、整體傳輸的平均效能。而 `-p` 選項用以指定資料並行連線的數量。這是最常使用的選項。

如果要針對一個目錄來傳輸，則使用以下指令：

```
globus-url-copy -vb -p 4 -r -cd - cc 4  
<來源url><目的url>
```

其中，`-cc` 表示指定在多重傳輸時，同時需要多少 FTP 連線。而 `-cd` 是表示如果需要，則建立目標目錄。針對目錄，則需要 `-r` 選項，用以複製子目錄中的檔案。

而來源端與目的端的 URL 的格式如下：

`file:///path/to/my/file`：從 client 端可以存取到的檔案系統。

`gsiftp://hostname/path/to/remote/file`：從一部 GridFTP 伺服器存取檔案。

如果要在兩部 GridFTP server 之間傳輸檔案，亦即進行第三方傳輸。雙方 URL 可以使 `gsiftp://` 或者 `sshftp://` 作為協定。前者使用 Grid Security Infrastructure (GSI) 認證，而後者使用一般 SSH 認證。

3. 系統架構與實作

3.1 網路環境

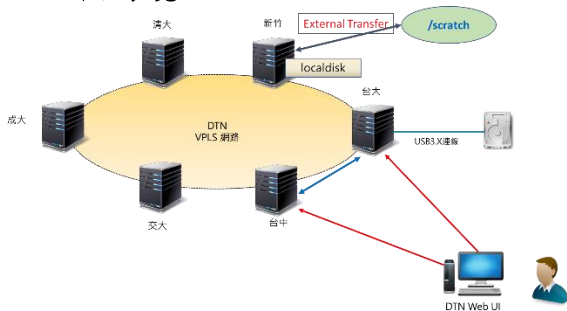


圖 1 DTN 系統環境

如圖 1 所示，我們的環境中有六個存取節點，分別位於台灣大學、清華大學、交通大學、成功大學、國網中心新竹本部，以及國網中心台中分部。在存取節點之間將以 TWAREN 骨幹網路進行連接。

然而，在網路的建置中，最後一哩部份，亦即是使用者的電腦到骨幹網路存取節點的路徑，往往是傳輸的關鍵。目前假設兩種情境：

- 使用者直接從研究室/實驗室的主機經由校園網路傳輸：透過與單位所屬計算機網路中心設定專用 VPN 連線介接至存取點。這樣的好處是使用方便，不需要每次攜帶資料至計網中心存取點。然而，網路傳輸的瓶頸就會集中最後一哩部份。再者，一般使用者端電腦往往是以 Windows 系統居多，故可選用 FDT 傳輸工具進行傳輸。
- 使用者直接至存取節點傳輸資料：使用者將擬傳輸之資料儲存於外接硬碟中，攜至存取節點之計算機網路中心，將外接硬碟接至存取節點之主機上。透過 USB 3.0 以上的傳輸介面進行傳輸。雖然程序上較為麻煩，但可

以直接使用到骨幹網路的頻寬，較不會受到最後一哩的頻寬所影響。再者，以此情境操作，FDT 或 GridFTP 均可使用。

3.2 軟體架構

本研究所提出的 WebGUI 供裝系統架構如圖 2 所示。使用者透過 DTN Web-UI 系統可以操作兩部存取節點，分別為 Server1 與 Server2 的檔案系統。假設 Server2 為使用者所在的計算機網路中心之伺服器，使用者可以攜帶儲存資料的外接硬碟接到該伺服器上。

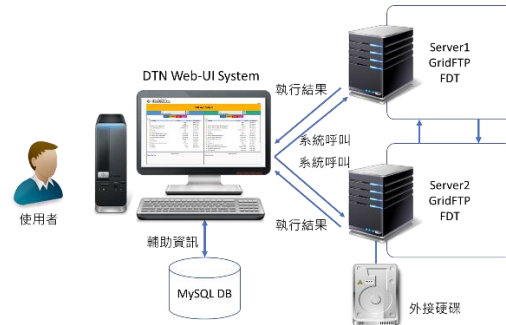


圖 2 Web-UI 系統架構圖

透過 Web-UI 系統，依據使用者的需求而在介面進行的操作都會被收集、並轉成相對應的 CLI 指令。而系統會登入到 Server1 與 Server2 上要求 GridFTP 或 FDT 執行相對應的指令。待本系統收到 GridFTP 或 FDT 傳回的文字結果，本系統便進一步將文字訊息轉成使用者易懂的訊息與圖形化介面，讓使用者能夠快速掌握系統現況。

3.3 HTML5 網頁

現行的網頁語法格式為超文字標記語言 (Hyper Text Markup Language, HTML)，目前的新版本為 HTML5。在 HTML5 中最大的特色就是對於多媒體的支援，包含圖形、聲音、影片等等，都有較前幾個版本有更完整的支援。

另外，HTML 網頁中原本就有的表單 (Web form) 是與使用者互動瀏覽的工具。表單介面中有各種輸入工具，可供使用者透過表單的介面輸入相關資訊，並以 GET 或 POST 方式傳至 server 端。

3.4 JavaScript

JavaScript 程式語言屬於 client 端動態網頁程式語言。在使用者瀏覽網頁時，JavaScript 使用 client 端瀏覽器的解譯功能，得到相對應的結果。

早期 JavaScript 僅僅只是網頁設計者用來增加一些網頁特效的小工具，或者是協助進行資料檢驗用的工具。近年來，在非同步技術 JavaScript (Asynchronous JavaScript and XML, AJAX) 與 Web 技術興起，JavaScript 的地位日趨重要。藉由 JavaScript 的效果，以往在桌上型應用程式的與網頁程式之間的差距變得愈來愈小。許多企業更傾向以 Web 網頁方式來開發各種管理系統。

然而，因為瀏覽器種類繁多，對於 JavaScript 的支援程度並不一致，致使 JavaScript 使用者開發程式時必須留意各種瀏覽器版本的相容性。為了能夠適應各種瀏覽器間的差異，並且降低開

發時之成本，開發者可以套用許多以 JavaScript 程式語言為基礎的公用函式庫來進行延伸應用。

3.5 PHP 動態網頁

在 WebUI 系統中，主要是以 PHP (Hypertext Preprocessor) 程式語言進行開發。其特性如下所述。PHP 程式語言是 server 端的動態網頁，因此會在 server 端先進行轉譯，並收集相關資訊後，可以形成相對應的回應訊息，並傳至 client 端的瀏覽器上。因此，client 端不需要額外安裝特殊的軟體、亦不需要進行額外環境設定。只需要使用網頁瀏覽器，便可以瀏覽與操作這套系統。

PHP 程式語言可以與 MySQL 資料庫進行互動存取。在 GridFTP 中有許多資料都是在 shell 中的 CLI 上顯示。在一般情況下，若是使用者要將 shell 回應內容寫至 MySQL 資料庫中，只能以手動方式輸入、pipeline，或者撰寫 shell script 來達成。而 PHP 程式語言對於與 shell 及 MySQL 的互動兩方面，皆有相當完整的支援。因此，使用 PHP 程式語言作為 shell 與 MySQL 之間溝通的渠道，是相當便利的解決方案。

PHP 程式語言可以進行系統呼叫，亦即可以與本身所在的主機之作業系統直接進行互動，不但可以傳送指令給作業系統，也可以接收作業系統傳回的訊息。因此，若欲整合互動的應用程式是 CLI 版本的應用程式，如本研究中所要整合的 GridFTP，在整合上的困難度將會大大降低。

然而，由於本論文所開發的 Web-UI 系統必須連線至遠端主機，而非單純地呼叫本身所在的系統，故無法使用一般的系統呼叫。因此，我們必須額外安裝 PHP 本身所提供的 SSH 連線函式庫 [7]。透過這套函式庫，系統有兩種模式可以呼叫遠端主機：第一種是一次性函式（即 cmdExec），而第二種是開啟一個 Shell（即 openShell），直到關閉此 Shell 為止。如果要執行單一指令，且每個指令皆為獨立事件，則採用前者。反之，如果要執行多道指令，而且每個指令皆有執行順序的關係，則採用後者。

3.6 MySQL 資料庫

為了日後管理上的便利性，我們利用 MySQL 資料庫將使用者的帳號與密碼記載下來。如表 1 所示，當使用者申請一組 Web-UI 系統的帳號與密碼後，便能登入 Web-UI 系統。其中 WEBID 為主鍵，不能重複。

表 1 Web-UI 系統帳號資料表

欄位名稱	意義
*WEBID	Web-UI 使用者帳號
WEBPASSWD	Web-UI 使用者密碼
USERREALNAME	使用者真實姓名
USEREMAIL	使用者 E-mail 信箱
USERAUXINFO	使用者其他聯絡資訊

表 2 使用者系統帳號資料表

欄位名稱	意義
*SERIALNUM	流水號
WEBID	對應 WEBID，識別使用者
HOSTIP	主機 IP
USERID	使用者在該主機上的帳號
USERPASWD	使用者在該主機上的密碼

USERAUXINFO	其他資訊，如帳號狀態
-------------	------------

而在表 2 中，以一個流水號作為主鍵。WEBID 表示與上述表 1 中的使用者對應。HOSTIP、USERID、USERPASSWD 分別表示對應的主機、帳號與密碼。在建立這個資料表的同時，管理者必須到相對應的主機上建立該使用者的帳號與家目錄，確保使用者登入能夠正常。

3.7 建立信任連線

由於高速傳輸軟體 globus-url-copy 工具有多種認證方式。本系統中採用 SSH 的認證方式。因此，假設使用者在 client 與 server 兩端皆有帳號，若要使用 globus-url-copy 指令時，只要在對端主機顯示要輸入密碼的提示字元時，輸入相對應的密碼即可。

上述步驟在 CLI 介面操作完全沒有問題。然而，若要以 PHP 實作為 Web 介面時，就會發生問題。因為 PHP 程式語言僅會執行一次 SSH 的指令，無法做到互動的機制。因此，我們在 DTN server 間設定 SSH 互相信任的免密碼機制。將雙方存取點伺服器的 RSA public key 複製到對方主機，即可讓兩部主機成為互相信任主機，免去輸入密碼的步驟。

4. 系統畫面

4.1 畫面簡介

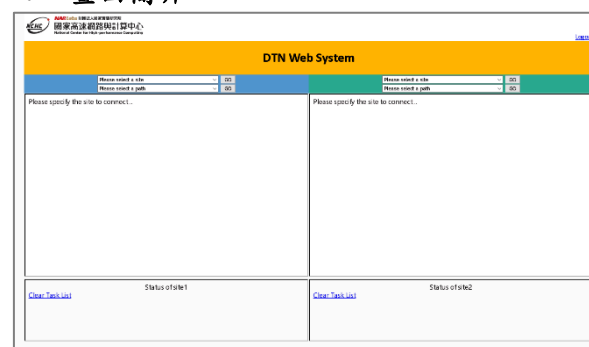


圖 4 登入後的畫面

在使用者登入正確的帳號與密碼後，便會看到如圖 4 的畫面。在畫面上段的部份，左右各有兩組下拉式選單，分別是主機選單與目錄選單。主機選單中只有名稱（或者代號），沒有 IP 位址。

左右兩邊靠中段的部份，則是之後連上 server 後的檔案瀏覽介面。而在左右兩邊靠下段的部份，則是檔案傳輸的狀態顯示。

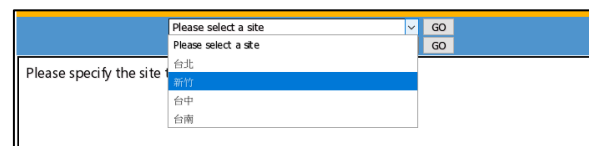


圖 5 Server 選單

如圖 5 所示，在拉下下拉式選單後，就會顯示主機的選項。在左、右兩邊分別選定要傳輸的雙方主機，並按下「GO」按鈕。這樣，系統會以 SSH 方式分別連至雙邊存取點 server 以進行後續主控。

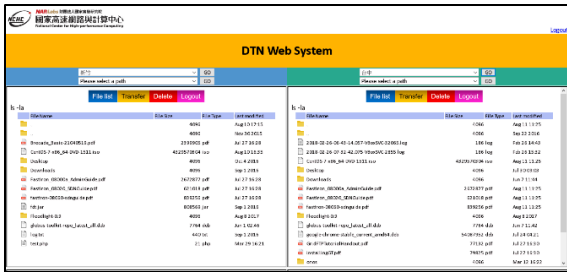


圖 6 兩邊 server 連線後的檔案清單

如圖 6 所示，左半部連線至新竹節點的 server，而右半部連至台中節點的 server。於檔案瀏覽介面中分別顯示兩邊 server 的檔案清單。而在清單上方，有針對檔案進行操作的按鈕，分別為檔案清單 (File List)、檔案傳送 (Transfer)、刪除 (Delete)，以及要登出 (Logout) 此 server 的功能連結。

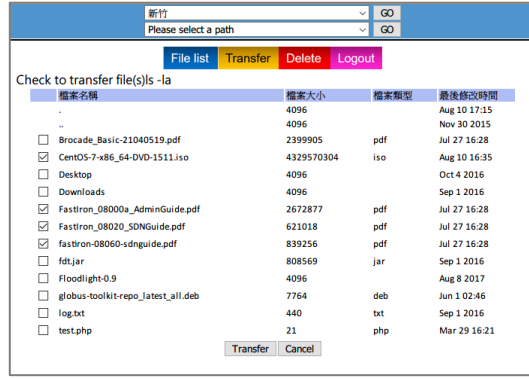


圖 10 按下 Transfer 的畫面

接下來，我們進行多重檔案傳輸。圖 10 顯示按下 Transfer 後的畫面，在檔案左方會顯示 checkbox，讓使用者可以選擇要進行傳送的檔案或目錄。選擇多個要傳輸的檔案，並按下下方的 Transfer 按鈕。

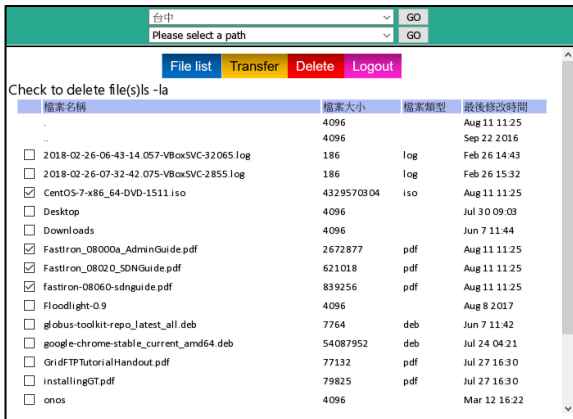


圖 7 按下 Delete 的畫面

圖 7 顯示按下 Delete 後的畫面，在檔案左方會顯示 checkbox，讓使用者可以選擇要進行刪除的檔案或目錄。按下下方的 Delete 即可進行刪除。

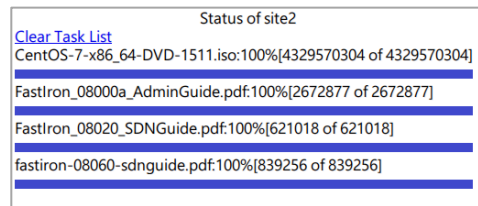


圖 11 傳送檔案的狀態欄

如圖 11 所示，在狀態欄會顯示所有檔案傳輸的進度，依據檔案容量百分比顯示。

4.2 效能優化功能實作

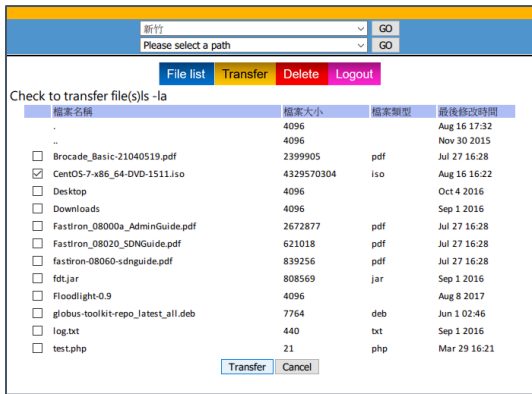


圖 8 按下 Transfer 的畫面

如圖 8 所示，按下上方的 Transfer 後，檔案清單左方會有 Checkbox。針對其中一個檔案核取後，並按下下方 Transfer 按鈕，就會進行傳輸。

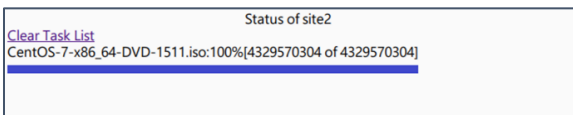


圖 9 傳輸狀態顯示

如圖 9 所示，在傳輸單一檔案時，會依據檔案容量顯示相對應的百分比。

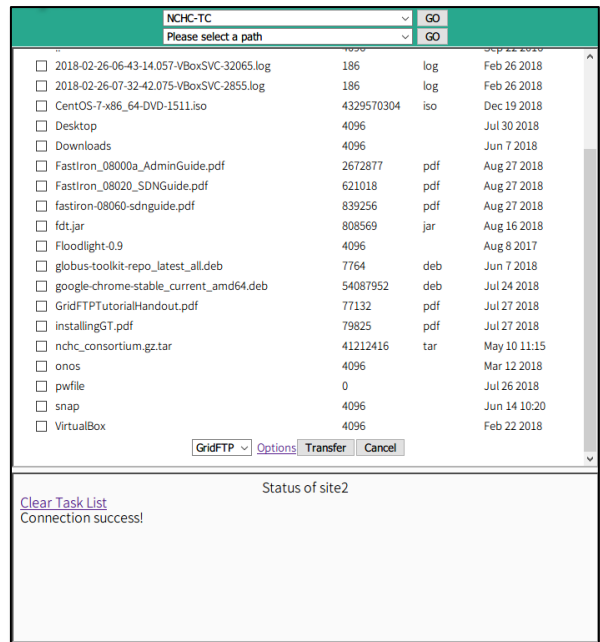


圖 12 傳送功能中的多重功能選單與選項連結

如圖 12 所示，為了要能夠讓使用者能夠進行高速傳輸測試，並提供進一步效能优化的可能性，我們在傳輸功能中實作多重工具選單，與工具選項連結供使用者選擇。如圖 13 所示，使用者可以透過下拉選單選擇要使用 GridFTP、FDT、Nuttcp、scp 等等工具。

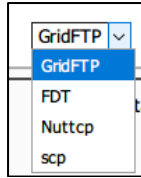


圖 13 傳輸工具下拉式選單

另外，如圖 12 所示，在 Transfer 按鈕旁還有另外一個標籤 Options。點選 Options 標籤後會出現如圖 14 與圖 15 的參數設定畫面，分別為 GridFTP 的參數設定與 Fast Data Transfer 的設定畫面。

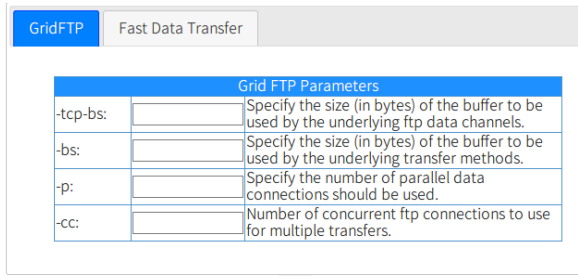


圖 14 GridFTP 的參數設定畫面

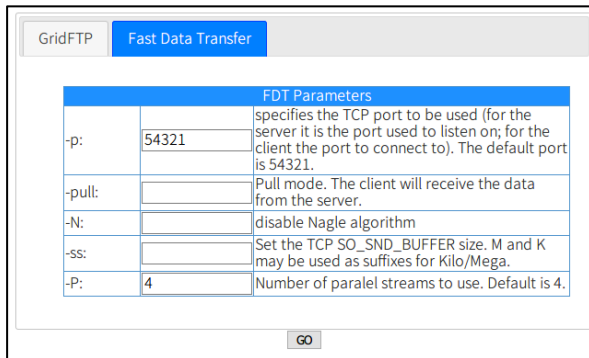


圖 15 Fast Data Transfer 的參數設定畫面

為了進行傳輸優化，可以針對傳輸工具進行進一步的參數調整，例如：

同一時間的資料傳輸連線數：Grid FTP 的 `--p` 與 Fast Data Transfer 的 `--P` 選項可以指定（預設值為 4）。

緩衝區大小：Grid FTP 的 `--tcp-bs`、`-bs`，以及 FastData Transfer 的 `--ss` 可以指定。

5. 效能測試

在本節中，我們主要針對台中與新竹兩節點進行測試。目前兩個節點僅為實驗環境，透過 1G 的線路進行通訊。

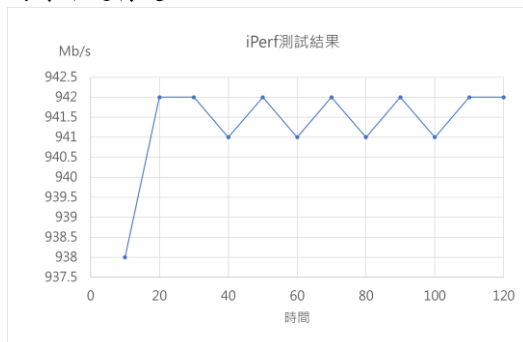


圖 17 兩端頻寬測試

如圖 12 所示，我們使用 iPerf 工具測試兩端的頻寬，得到頻寬均在 938Mbps/sec 以上，接近 line rate。

接下來，我們針對 CentOS 的原始 ISO 檔案，（檔案大小為 4329570304 位元組）進行傳輸。

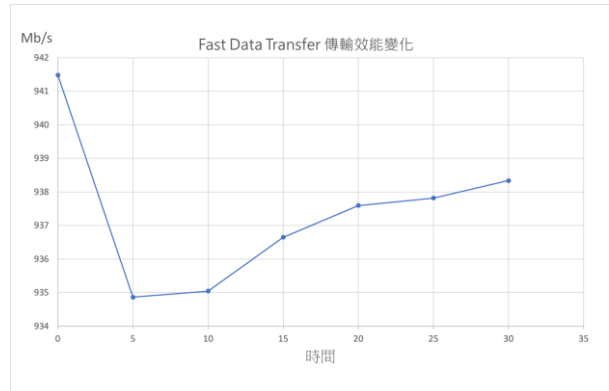


圖 18 Fast Data Transfer 傳輸結果

圖 18 顯示利用 Fast Data Transfer 的傳輸效能變化。待 DTN 線路與主機實際建置完成後，將會有更佳的性能表現。

6. 結論與未來展望

本論文展示一套視覺化 TWAREN 高速資料傳輸網路（Data Transfer Network, DTN）系統之建置。在硬體部份，利用 TWAREN 的骨幹線路，我們將六個存取節點，包含台大、清大、交大、成大、國網中心新竹本部、國網中心台中分部串接起來，使得連線單位能得到更快的傳輸速率。在軟體部份，我們採用廣為應用的高速傳輸工具 Fast Data Transfer (FDT) 與 GridFTP，使得傳輸頻寬運用更加有效。我們也展示我們所開發的視覺化系統來取代命令列介面的輸入。實驗結果顯示，兩種傳輸工具均有相當不錯的表現。未來，我們將繼續進行系統建置，並持續測試以求最佳參數，讓 DTN 系統能發揮最大的效益。

參考文獻

- [1] Gartner：全球各種裝置（個人電腦、平板、手機）出貨量估 2018 年增 2.1%，達 23.2 億台
<http://iknow.stpi.narl.org.tw/Post/Read.aspx?PostID=14166>
- [2] 台灣先進學術研究網路 TWAREN, <http://www.twaren.net/>
- [3] 10G 變 100G 台灣新學研網路啟用
<http://www.chinatimes.com/realtimenews/20161006004628-260405>
- [4] How to transfer large amounts of data via network
http://mooc.nac.uci.edu/~hjm/HOWTO_move_data.html
- [5] Fast Data Transfer – FDT,
<http://monalisa.cern.ch/FDT/>
- [6] GT 6.0 GridFTP,
<http://toolkit.globus.org/toolkit/docs/latest-stable/gridftp/>
- [7] Secure Shell2 of PHP,
<http://php.net/manual/en/book.ssh2.php>