

私立東海大學  
資訊工程學系

碩士論文

指導教授：陳隆彬 博士

以志願運算技術開發具規模適應性的程式語言數位學習系統

A SCALABLE ELEARNING SYSTEM FOR COMPUTER PROGRAM  
LANGUAGES BASED ON VOLUNTEER COMPUTING

研究生：林建安

中華民國九十九年七月

## 摘要

數位學習的風潮隨著電腦的發展與網路的普及越來越盛行，線上評分系統是一個讓學生可以練習程式語言的教學系統，本篇論文提出一個新架構讓線上評分系統不只是使用一般模式運作並可在線上人數過多時可與志願性計算平台結合，以分散處理運算的需求，增加系統效能並且降低運作成本。論文中介紹志願性計算與數位學習系統整合之相關議題與解決方案，系統實作與實踐結果證明本論文所提架構能提升伺服器效能並具有規模適應性。

**關鍵字：**數位學習、e-learning、分散式運算、志願運算

# 目錄

摘要.....	i
目錄.....	ii
圖目錄.....	iii
第一章 前言 .....	1
第二章 研究背景與目的 .....	4
2.1 線上評分系統 .....	5
2.2 BOINC .....	7
2.3 建置程式語言數位學習系統之相關問題 .....	9
第三章 運用 BOINC 平台開發線上評分系統.....	11
3.1 系統架構 .....	12
3.2 系統流程與組件 .....	13
3.3 BOINC Wrapper .....	15
3.4 BOINC Client Manger 加入 Socket 連線功能 .....	16
3.5 安全性問題的考量 .....	17
第四章 系統實驗與分析 .....	18
第五章 未來展望 .....	20
第六章 結論 .....	21
參考文獻.....	22

## 圖目錄

圖 2-1 數位學習架構.....	4
圖 2-2 線上評分系統.....	6
圖 3-1 志願性線上學習管理系統.....	12
圖 3-2 BOINC 平台進行 CPP 與 EXE 工作之流程.....	14
圖 3-3 BOINC Wrapper .....	15
圖 4-1 worker 個數的不同所產生的平均回應時間.....	19

## 第一章 前言

數位學習的風潮隨著電腦的發展與網路的普及越來越盛行，各種線上學習系統也隨之孕育而生。在資訊領域中，程式設計是一門基礎的課程，本論文所提之線上評分系統(Online Judge System)是一個讓學生可以練習程式編碼的系統，學生只需要在能上網的環境下即可練習程式語言。它可以針對學生上傳的程式碼進行編譯和執行，接著使用預先產生的測試數據來檢驗程式碼的正確性，剛開始學程式語言的人通常都會感到很陌生並且在寫程式時也不之如何下手，因此透過實際的題目來練習程式語法將可以讓學生漸漸瞭解程式語言的使用方式。

目前有許多學校或是網路社群自行開發的線上評分系統，主要的功能均相同，只是其他附加功能與開發語言不同而已，但是運作系統的主機均需要強大的處理能力。如果一個班級約 50 人需要在課堂上讓學生透過這樣的系統來學習時，運作主機本身可能必須同時回應十幾位學生的要求，當人數增加時回應時間也隨之下降，若是有多個班級同時使用，通常會限定目前使用者的數量或是增加多台專門負責運算的主機來處理，這些限制不止將造成使用上的困難並且降低使用意願，線上評分系統的運作主機的維護與架設的成本也將隨之增加，並且在

擴充性上，不能快速的反應使用人數的增減，主機效能容易造成浪費或不足。

學生上傳的程式碼在線上評分系統的系統下，會受到比較嚴格的限制，諸如：運行時間、記憶體使用限制、安全限制等等。學生的程式執行結果會被線上評分系統自動保存，然後再將學生執行結果的輸出數據跟標準輸出數據比較差別或是檢驗學生的輸出數據是否滿足一定的邏輯條件，最後系統會再返回一個狀態給學生，讓學生得知程式碼是否正確。

隨著科技進步，個人電腦效能越來越強大，但大多數運算效能並未充分使用，所以志願性計算的概念被提出來—讓志願者可以提供電腦資源以進行分散式運算的分配與安排，因此原本只能用超級電腦來處理的科學問題也希望能讓志願電腦能協助處理這些龐大的運算量，BOINC(Berkeley Open Infrastructure for Network Computing)是一套分散式運算平台 [4][5] 讓我們可以使用志願者的電腦運算能力，讓運算量可以分散到這些電腦上，BOINC 平台只需整合回傳結果。

本篇論文提出一個線上評分系統，運用BOINC平台來分散處理運算，

增加系統效能並且降低運作成本，本篇論文的架構為：第二節介紹相關技術，第三節討論系統架構與實作方法，第四節展示實驗數據與分析，第五節說明未來展望與第六節結論。

## 第二章 研究背景與目的

本篇論文中提到的線上評分系統與 BOINC 平台本身是兩個獨立的系統，我們透過網路把兩者結合起來，形成一個完整的數位學習架構(圖 2-1)。



圖 2-1 數位學習架構



## 2.1 線上評分系統

「 Association for Computing Machinery (ACM) 」[7]是一個致力於電腦科學教育的協會，出版大量專業期刊、文獻，舉辦重大的計算機科學會議，在資訊界舉足輕重、名聞遐邇。ACM 每年度都會舉辦一次

「 The ACM-ICPC International Collegiate Programming Contest (ACM/ICPC) 」，是一個給全世界大專院校學生參加的演算法程式設計比賽，比賽目的在於考驗選手臨場時的演算法設計能力、程式編寫能力。ACM/ICPC 帶動了演算法程式設計的風氣。世界上許多大專院校的資訊系所，仿照 ACM/ICPC 的比賽模式，紛紛自行開發出即時線上比賽系統，能夠自動批改、評分、計時、統計。學生不必齊聚一堂，就可以相互切磋程式設計技巧。比賽結束之後，便將比賽題目編列題庫，並開放線上批改程式的功能，供學生賽後練習檢討。這套系統大家一般稱之為「 線上評分系統(OnlineJudge)」。

最知名的 線上評分系統 ，是由西班牙知名的瓦雅多利大學

「 Universidad de Valladolid (UVA) 」(圖 2-2)[11]開發的「 UVA 線上評分系統 」。UVA 線上評分系統 亦和 ACM 合作，成為 ACM 推廣的一個 線上評分系統 ，藉此向大眾提倡程式設計。因此 UVA 線上評分系統 除了收集自行舉辦的比賽的題目，也嘗試收錄世界各地

重大程式設計比賽的題目，以臻豐富完整。

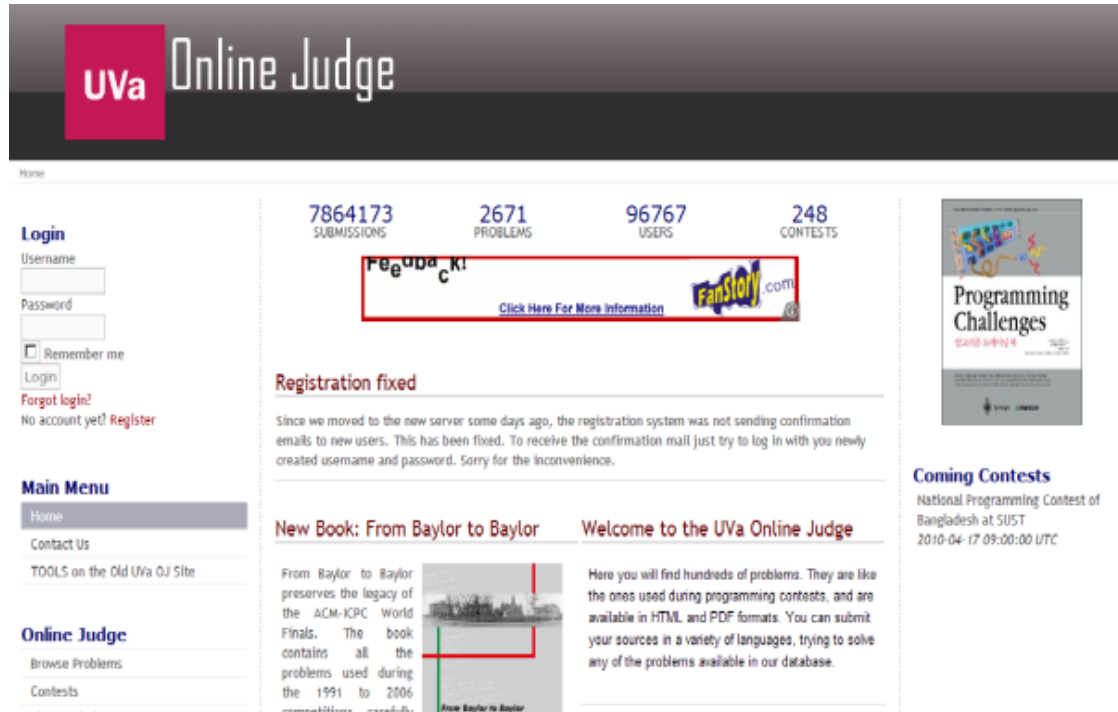


圖 2-2 線上評分系統

## 2.2 BOINC

柏克萊開放式網絡計算平台(Berkeley Open Infrastructure for Network Computing, 簡稱 BOINC) [4][8]是目前主流的分佈式計算平台之一, 由加州大學柏克萊分校的電腦學系發展出來的分佈式計算系統, BOINC 平台最初設計用於 SETI@home 項目[5], 目前也應用於包括數學、醫學、天文學和氣象等其他領域[6] [3] [10]。

安裝 BOINC 軟件的個人電腦在閒置時會使用個人電腦的 CPU 進行運算。即使個人電腦正被使用, 假如仍有空閒的 CPU 週期, BOINC 也會用作計算。當參與者使用個人電腦參與 BOINC 項目時, BOINC 會與項目的伺服器通信, 伺服器會向個人電腦提供工作包(Workunit, 簡稱為 WU), 然後個人電腦會對工作包作出運算。完成後 BOINC 會把結果上傳至項目的伺服器。每個項目有它的伺服器, 它用作協調各參與電腦的工作, 包括發送工作包、接收已處理的結果、核對大量的結果再作處理, 成為研究人員需要的數據。由於個別的個人電腦可能會在運算過程出現錯誤, 所以伺服器一般會把同一工作包傳送至多個參與者, 並比較各個結果。

BOINC 設有積分系統, 積分間接反映參與者的貢獻。因為在 BOINC

上可以運行的項目千差萬別，比如項目 A 的工作包在某台電腦上需要 3 個小時完成，而項目 B 的工作包在該電腦上需要 30 個小時才能完成，顯然用完成的工作包的數目來衡量工作量是不可行的；類似的，電腦性能也有差別，用 CPU 時間來衡量工作量更是不行的。積分系統只能通過一定的算法得到參與者實際完成的計算量。

目前有許多分散式系統平台，如 XtremWeb [1] [2] 可以使用，但其中最普遍也最熱門的分散式系統平台為 BOINC，所以我們利用這套平台管理自運電腦的運算效能，使用 BOINC 可以快速的布建一個分散式平台作為我們送出工作與蒐集資料使用，官方網站提供全部的程式碼，讓各領域的人可以修改程式架構配合計畫的執行。

## 2.3 建置程式語言數位學習系統之相關問題

線上評分系統 在編譯執行學生的程式碼的時候，往往會占用 5MByte ~ 20MByte 的記憶體空間，而在檢驗的時候又需要開檔讀檔寫入等要花費許多時間的動作，這樣的系統若是在只有數人至十幾人的環境下，使用一般主機尚可以負荷，但若是學生有數十人甚至數百人，所佔用的記憶體空間就高達數十 GB，每次瞬間的檔案讀出寫入動作就有數萬次了，這麼高的需求量，只能使用專業伺服器或是多台負責處理的伺服器才能負荷了。

線上評分系統通常會遇到這些困難：

- 規模適應性(scalability)是網路計算系統的一個重要因素

線上評分系統通常開發給多人同時練習程式編寫—它代表使用者人數少至個位數，多至數百人時，系統是否能維持可接受的效能，為應付可預期的大量使用者人數，必須添購新的電腦或是擴充硬體設備，這將大幅提高學習網站的運作成本。因此如何建置一個具規模適應性的系統就成了重要的課題。

- 硬體成本過於昂貴

線上評分系統的瓶頸在於編譯與執行程式都是須佔用大量 CPU 時間。為了提供高穩定度的系統，必須使用昂貴的高階伺服器。

- 安全性

在處理學生所上傳之程式碼時，無法判斷程式碼的內容是否會造成伺服器的主機發生錯誤，或系統的安全性是否會受到影響。

- 時效性

使用線上評分系統主要就是讓學生可以在上傳程式碼後，可以在一定時間內得到回傳結果

，在線上學生不多時，反應時間通常都在幾秒之間，但是當數十人同時練習時，反應時間將會大幅下降，甚至系統無法回應，時效性問題會降低學生的意願。

### 第三章 運用 BOINC 平台開發線上評分系統

考量前面所述之問題，我們運用 BOINC 平台把編譯與執行的動作分散給志工電腦，讓線上評分系統本身只需負責學習進度的管理，進而提昇線上評分系統的穩定度。

### 3.1 系統架構

我們將志願性線上評分系統分割成兩大部分，線上學習管理系統與BOINC平台。概念如下：

架構圖提示在(圖 3-1)，學生使用者透過線上學習管理系統進行教材學習，依進度上傳作業程式。線上學習管理系統會將資料分割傳送給BOINC派送給志工電腦進行運算，運算結果也將儲存到線上學習管理系統的資料庫進行分析。使用者能從網頁介面得知處理進度與結果。

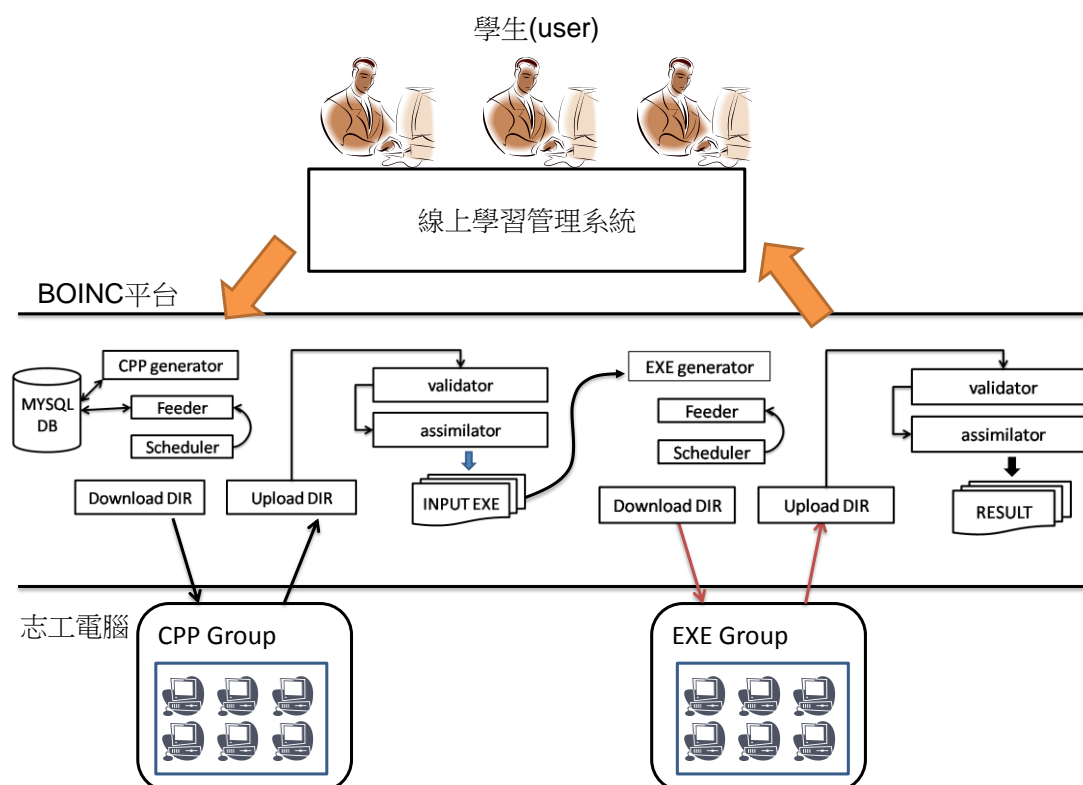


圖 3-1 志願性線上學習管理系統



## 3.2 系統流程與組件

系統組件與相關名詞的說明：

- WU：WorkUnit，在 BOINC 中代表一個工作。
- Feeder：負責存取資料庫目前尚未送出的 WU。
- Scheduler：負責存取與管理尚未送出的 WU。
- Generator：BOINC 工作產生程式。
- CPP group：執行 compile 工作的電腦群組。
- EXE group：執行 execute 工作的電腦群組。
- Validator：工作驗證程式。
- Assimilator：工作合併程式。

詳細的系統實做說明(圖 3-2)：

- (1)學生進入到線上評分系統，並上傳程式碼，系統存入資料庫中。
- (2) CPP generator 會持續的掃描資料庫，確認是否有尚未編譯的程式碼，如果有尚未編譯的程式碼，就會包裝成一個 WU，會在資料庫中產生一筆資料說明 WU 的資訊，並把相關的檔案複製到 download 資料夾之下。
- (3)Feeder 會存取資料庫中尚未送出的 WU，scheduler 用來回應 CPP 志工電腦群目前是否有多少尚未被執行的 WU 數量，如果有尚未被執

行的 WU 則會把資訊告知 CPP 志工電腦群。

(4) CPP 志工電腦群依據相關資訊到 BOINC 的 download 資料夾取得相關檔案後，執行動作，並回傳結果到 upload 的資料夾。

(5) Validator 在 upload 資料夾中若發現兩個以上的執行結果存在時，就會作驗證是否結果一致，假如結果一致代表正確，便讓 assimilator 把這兩個結果合併起來，並送到我們指定的資料夾 inputEXE。

(6)(7)(8)為 EXE 模組的工作流程與 CPP 的動作流程相似。

(9) 最後 assimilator 會通知一即將執行結果與線上學習管理系預存之答案進行比對，線上學習管理系統進行驗證與評分，並把評分結果展示在網頁上讓學生知道目前的評分結果。

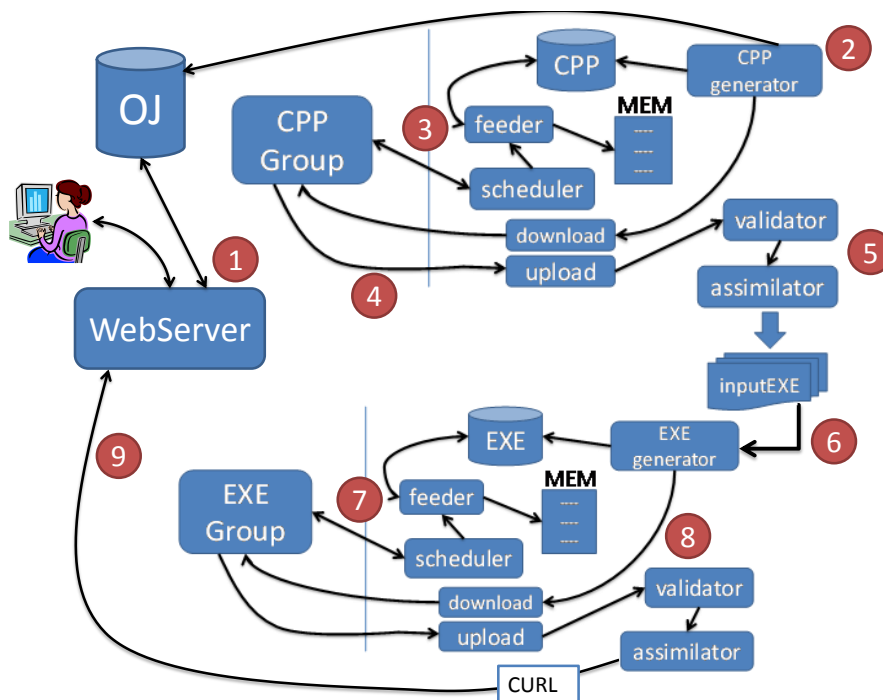


圖 3-2 BOINC 平台進行 CPP 與 EXE 工作之流程

### 3.3 BOINC Wrapper

志工電腦會有各種不同作業系統，在不同的作業系統中執行外來執行檔時，需要以 shell 或 system 等指令來處理，為了解決這個問題，我們因此使用了 BOINC 另外一項功能 BOINC Wrapper [9] 的功能來實現編譯與執行的功能其內容(圖 3-3)。

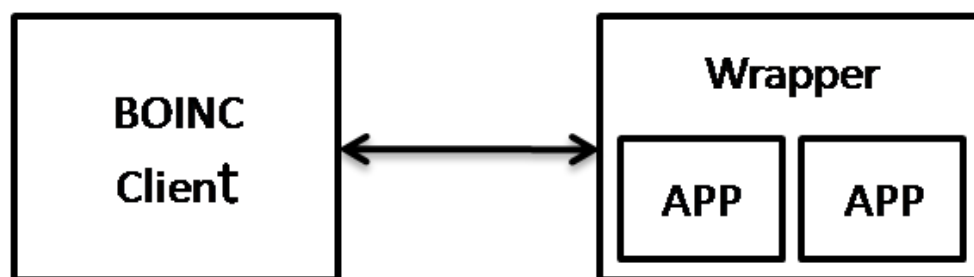


圖 3-3 BOINC Wrapper

### 3.4 BOINC Client Manger 加入 Socket 連線功能

BOINC 伺服器本身與志工電腦的動作都屬於被動型態的，因此伺服器需要等到客戶端主動要求下達指令，因此我們修改了客戶端的 BOINC client manager 管理程式，增加一個 Socket 連線到 BOINC 伺服器，讓伺服器可以在工作產生後，隨即送出一個 Socket 指令到客戶端，要求客戶端來取得工作。

### 3.5 安全性問題的考量

為了防止評分時答案可能被志工修改，如志願電腦為學生本身的電腦時，每一份工作將同時被複製(Redundant)為兩份，並且送到不同電腦去運作，最後由主機驗證工作結果是否一致，因此可以避免誤判，而在降低惡意程式碼風險的部分也將在以後透過虛擬機器的方式，可以派送已經安裝好相關程式的映像檔到志願電腦，讓志願電腦可以執行此虛擬機器，而不透過本身電腦來執行運算工作，將可以確保本機的使用安全性無虞。

## 第四章 系統實驗與分析

為了評估我們所運用 BOINC 平台建立的線上學習管理系統的效率，我們選擇了線上學習管理系統裡的一個題目，這個題目本身會有 6 個驗證資料，經過我們的流程會產生 2 個 CPP 的工作與 12 個 EXE 的工作，每個工作在 client 的平均完成時間約 10 秒，這個平均完成時間會受到目前系統的覆載狀況而影響，因此我們實驗的電腦都不做其他動作，而專門執行 worker 工作，實驗電腦均具備雙核心處理器—Intel Core 2 Duo E6550。

我們主要評估使用 BOINC 平台的模式下，從程式碼上傳到線上學習管理系統到結果產生在學生面前所需的時間，而 worker 個數的不同所產生的平均回應時間(圖 4-1)。

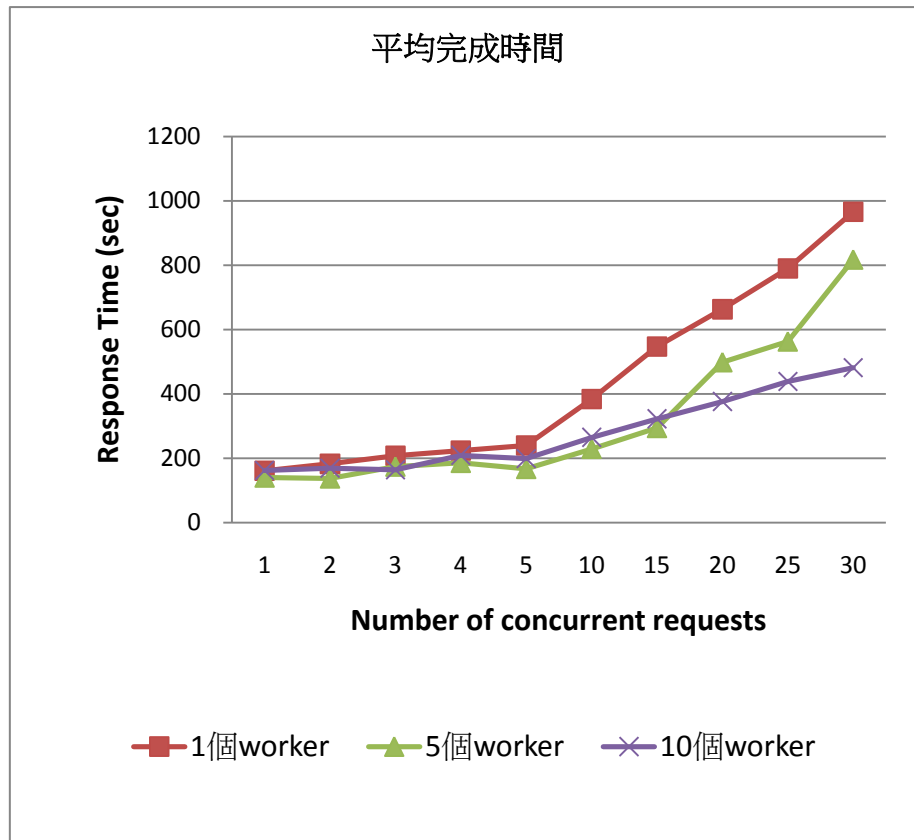


圖 4-1 worker 個數的不同所產生的平均回應時間

這個數據(圖 4-1)顯示當同時要求數量少時，平均反映時間的差異不大，但是當同時要求數量增加時，反應時間將受到 worker 數量的不同，而有明顯的差異，因此，當 worker 數量越多時，反應時間也會越趨平緩，達到反應時間不會隨著同時要求數的增加而上揚，這個實驗結果也與我們預期的結果相符合。

## 第五章 未來展望

未來的研究中，我們將就以下幾點作為之後開發的目標：

- 系統流程追蹤

當線上學習管理系統與 BOINC 平台結合完成時能整合如 Workflow 的模式進行細部的工作流程管理。

- 程式碼安全性

線上評分系統提供學生上傳程式碼，由伺服器端包裝成工作後給 CPP 志工電腦群組與 EXE 志工電腦群組來完成編譯與執行，取得最後的結果，但是因為無法控制學生的程式碼編寫方式，因此如果學生的程式碼中有危險程式碼時，必須以事先偵測的機制或是虛擬機器來避免志工電腦受到危害。



## 第六章 結論

數位學習系統的穩定性與擴充性也隨著不同的學生規模而受到考驗，本篇論文中，提出線上評分系統與 BOINC 分散式運算平台的結合應用。藉由把工作派送給學生們的志工電腦或是由電腦教室中的閒置電腦做為志願電腦，降低伺服器的建置成本，並在學生使用者人數大量變動的情況下，仍能提供穩定的運算能力。

- 透過 BOINC 平台，我們的架構能快速佈建許多的志工電腦，來分擔運算量，並且在設置上僅需要幾個簡單的步驟即可完成。

未來隨著分散式運算概念的普及與相關平台和軟體越來越成熟，這類的結合應用將會越來越多。

## 參考文獻

- [1] A. Chien, B. Calder, S. Elbert, K. Bhatia. "Entropia:architecture and performance of an enterprise desktopgrid system," Journal of Parallel and Distributed Com-puting 63, 5. pp. 597-610, May 2003
- [2] F. Cappello, S.Djilali, G.Fedak, T.Herault. F.Magniette, V.Neri, and O.Lodygensky, "Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with Grid," FGCS Future Generation Computer Science, 2004. Nov. 2002, Vol. 45 No. 11, pp. 56-61.
- [3] V. Pande et al., "Atomistic Protein Folding Simula-tions on the Submillisecond Time Scale Using World-wide Distributed Computing," Biopolymers, Vol. 68.2003. pp. 91-109.
- [4] BOINC "A System for Public-Resource Computing and Storage. David P. Anderson," 5th IEEE/ACM International Workshop on Grid Computing, Nov. 8, 2004, Pittsburgh, USA.
- [5] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D.Werthimer. "SETI@home: An experiment in public-resource

computing,” Communications of the ACM, Vol. 45 No. 11, Nov. 2002, pp. 56-61.

- [6] D. Stainforth et al., ”Climateprediction.net: Design Principles for Public-Resource Modeling Research,” Proceedings of the 14th IASTED International Conference on Parallel and Distributed Computing Systems, Nov. 4-6, 2002, Cambridge, USA.
- [7] Association for Computing Machinery (ACM)  
Apr. 15, 2010, <http://www.acm.org/>
- [8] BOINC  
Dec. 20, 2009, <http://boinc.berkeley.edu/>
- [9] BOINC Wrapper  
Feb. 25, 2010, <http://boinc.berkeley.edu/trac/wiki/WrapperApp>
- [10] Predictor@home  
Nov. 15, 2009, <http://predictor.scripps.edu>
- [11] UVa Online Judge  
Apr. 14, 2010, <http://uva.onlinejudge.org/>