

運用分散式佇列運算系統於建築設計視覺化渲染

李苾文¹、邱建福²、李若榛²、詹嫻柔²、張惠君²、
張允政²、周俊翰²、吳浩璋²、王恩庭²、周玉端³

¹ 實踐大學觀光管理學系講師 cliolil213@hotmail.com

² 嘉南藥理科技大學應用空間資訊系學生 r0926852901@yahoo.com.tw

³ 嘉南藥理科技大學應用空間資訊系副教授 ytchou@mail.chna.edu.tw

摘要

建築設計視覺化的過程中，需要非常大的電腦資源來進行效果渲染，使用單一計算機會耗費極多的時間來完成此工作。本論文以分散式佇列運算系統進行軟硬體之建置，用單位內現有的電腦設備，搭配現有的商用軟體 3ds Max Design 2010 與 Backburner，來進行視覺效果圖與動畫製作效能提升之研究。研究結果顯示，此系統對建築設計視覺化渲染，有相當大的效能提升；對於單張視覺效果圖而言，高解析度效果圖的渲染效能提升較低解析度圖為高；對於動畫影格渲染而言，渲染影格數的多寡所獲得的效能提升趨勢一致，此研究成果將可應用在進行建築設計視覺化渲染，運用分散式佇列運算系統，可有效地提升工作效能。

關鍵詞：設計視覺化、分散式佇列系統、網路渲染、計算機網路

運用分散式佇列運算系統於建築設計視覺化渲染

1. 前言

隨著電腦繪圖效能的增加，人們也逐漸追求視覺感官上的滿足，在多媒體視覺化設計的過程中，為了獲得接近真實的效果圖或動畫，常常需透過執行精確的三維建模、真實色彩與光跡追蹤等執行程序，這些都需要非常多的電腦計算效能來完成；單一台計算機的效能會受限於其作業系統及相關軟硬體之架構，在此視覺化渲染過程中，會耗費相當長的時間；這在多媒體數位內容設計開發過程中，令人相當難以接受。

在以往，為解決計算時間過長的問題，多採用大型電腦主機(Main-Frame)，以強大的計算速度及容量來克服此一缺點，但須耗費極高的軟硬體設置及維護費用，近十年來，隨著區域網路的發展，以內部群組形成一個運算叢集(Cluster)，透過主從端的運算工作分配，一起針對一個專案工作進行渲染，這已有相當多的研究成果出現。

在平行與分散處理的研究中，以美國 Livermore 國家實驗室、史丹福與維吉尼亞大學之 Chromium 計畫最為著名[1]，該計畫使用市售的桌上型電腦及內含的消費用顯示卡組成一個商品化的叢集系統，透過應用程式介面(Application Programming interface, API)將各單一電腦組合成平行繪圖運算系統，可提供達到超級電腦的運算效能。Bach 等人[2]即透過 Chromium 組成的圖形處理系統進行應用的研究，他們針對數種叢集的架構和資料模型進行效能評估，這些研究成果可使用在網路的應用上。而另一種以叢集系統為基礎，融合格點(Grid)架構與網路應用所發展出的 Equalizer 圖形叢集[3]，則是同樣地能使用多重圖形處理單元(GPU)，來進行複雜效果圖形之渲染工作。Gorgan 等人[4]就利用 Equalizer 系統，針對不同的分散渲染機制進行組合，來獲得最佳化的系統架構。

而在個別的研究工作中，為了能建構複雜場景並獲得良好的視覺效果，多個圖形引擎加速裝置也被開發出來，諸如：openVISSAR 工具包[5]，則是利用遠端及協同視覺化的機制，來獲得即時產生視覺化圖形的能力。Ogre 圖形引擎[6]提供相當多選擇組件的自由度，讓你在建立模型及材質時，有更好更快的設計過程；Irrlicht 引擎[7]則是提供以不同檔案格式來定義模型與材質，消除掉即時圖形運算的一些障礙；Mocan 及 Gorgan[8]則是提出物件導向式圖形渲染引擎(object-oriented graphics rendering engine)，以彈性的類別階層來將特定的場景架構成可自由插入的形式，可以同時運用在平行處理的平台上。

上述討論的處理方式裡，多屬於有專屬性的作業平台需自行設定系統架構，或是自行開發的繪圖引擎及工具包，與現行商用繪圖軟體不易結合，需再自行開發延伸程式。如何容易裝置協同渲染的運算系統，並與現有商業軟體做無縫(seamless)連結，為各研究者與軟體公司亟需開發的一個項目。Autodesk 公司在設計視覺化的應用領域中，已開發的軟體相當多，諸如：3ds Max, Maya, Softimage, …等等。這些軟體在製作具真實化的圖片或動畫，通常需耗費極多的時間，該公司因應提升渲染效能的需求，特地開發出以分散式佇列運算系統為基礎的 Backburner 軟體，以群組電腦協同算圖的方式，來解決單一機台過長處理時間的問題。本文即以實驗室群組電腦為硬體平台，裝置 3ds Max Design 與 Backburner 軟體，來研究建築設計視覺化渲染過程中的效能問題，並獲得不同數量的群組電腦，所造成的效能增益。

2. 分散式佇列系統(Distributed Queuing System, DQS)

個人電腦有其獨立的處理器與硬體裝置資源，應用網路連接，將散置各處的電腦藉

由分散式作業系統合作執行指定的工作，如此安排是謂“分散式系統(Distributed System)。其優點有：資源共用、加速計算、可信度、具備通訊功能、架構彈性，可採用主從式(Client/Server)架構或是點對點(peer-to-peer)架構。而分散式佇列系統則是一種工作排程的管理裝置，其架構如圖 1 所示。

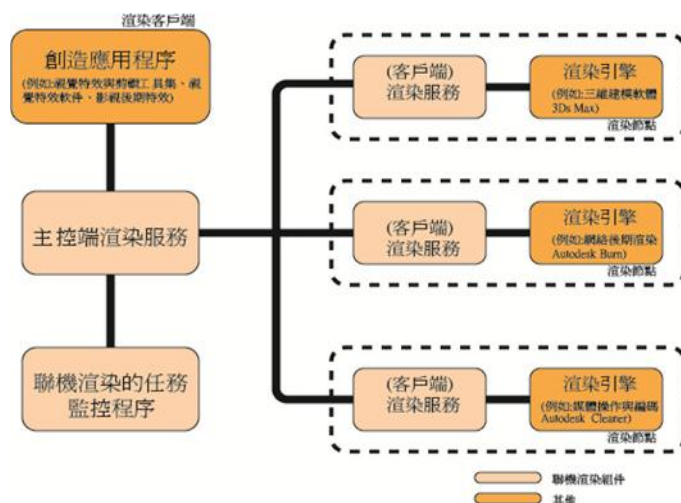


圖 1 分散式佇列系統架構示意圖[9]

其工作的流程如下：

- (1) 應用(Application)程式將要執行的工作(job)分成個各佇列(Queue)，則特定的 Queue 可執行特殊要求的 Jobs。
 - (2) 透過管理(Manager)程式將每一部電腦指派一個(含)以上的佇列(Queue)，其中可針對不同特性的 PC 給特定的 Queue，可以將排隊等候的 Jobs 做刪除或暫停動作。
 - (3) 監控(Monitor)程式可以控管每部 PC 的負載量，並由管理程式自動重新分配佇列。
- 這類的系統優點有：

- (1) 避免 Jobs 集中在固定而且大家熟悉的電腦中執行，分散機器的負載量。
- (2) Jobs 執行時可避免受到其他 Jobs 的干擾而被趕進虛擬記憶體中，降低執行效率。
- (3) 有效利用 PC-cluster 資源。

但仍有其缺點，分別是：

- (1) 多數程式為開放原始碼(Open source)，操作資料取得不易，亦無相關的諮詢服務。
- (2) 操作介面不佳。
- (3) 需要良好的網路環境。

為了因應視覺化設計在渲染過程中所需巨量的計算效能，Autodesk 公司提出了針對其商用應用軟體在進行算圖過程時，使用分散式佇列運算軟體 Backburner，其使用的系統與工作架構如圖 2 所示。

其中渲染佇列管理員(Backburner manager)是一個核心，負責接收客戶端的工作，然後分發到網絡上的渲染節點；渲染佇列管理員能維護並回報網絡服務器的狀態信息，並積極有效率的完成工作。渲染顯示端(Backburner monitor)為終端用戶和管理員相互作用的介面；它是用於監測工作的進展情況，暫停和重新啟動工作，並執行管理任務。渲染節點(Render Nodes)則透過計算機所組成的渲染服務器，包含 plug-ins/adapters 和渲染器。

渲染佇列管理員的任務是經過切割過的工作分配給它的網路渲染服務器，在渲染節點中的 plug-ins/adapters 收到網路渲染服務器的指令來控制渲染器，而渲染請求是由 3ds Max Design 軟體提交給 Backburner 服務器的。Backburner 對每一個渲染節點的可用性

和效率會進行優先級判別，並相應地分派渲染任務。

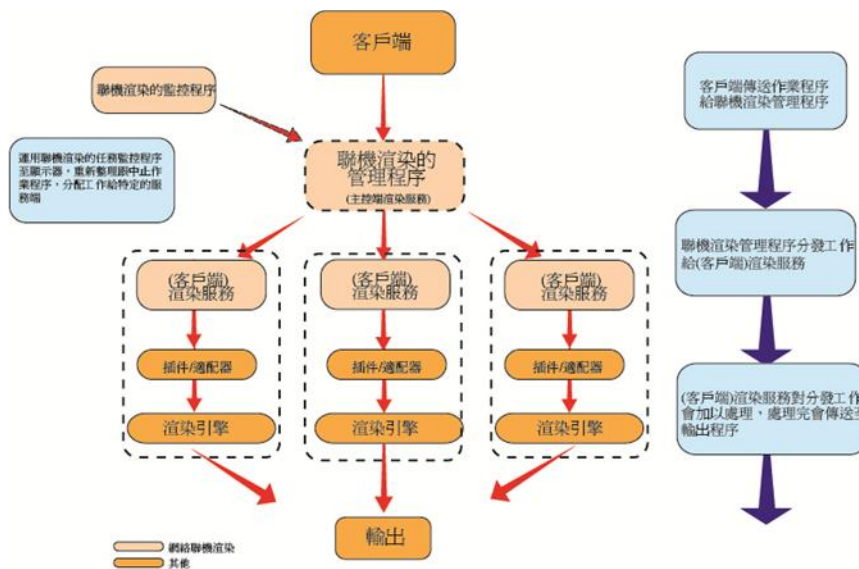


圖 2 商用軟體 Backburner 所使用的系統與工作架構圖[10]

這套工具程式除了提供良好的分散式系統使用穩定性外，也提供了兩種實施模式，一種是單機操作(standalone)模式，適用在具有多核心運算與繪圖處理單元的計算工作站(Workstation)；另一種就是使用在組成俗稱算圖農場(Render Farm)的網路群組電腦，其運算工作與佇列派送之流程如圖 3 所示。

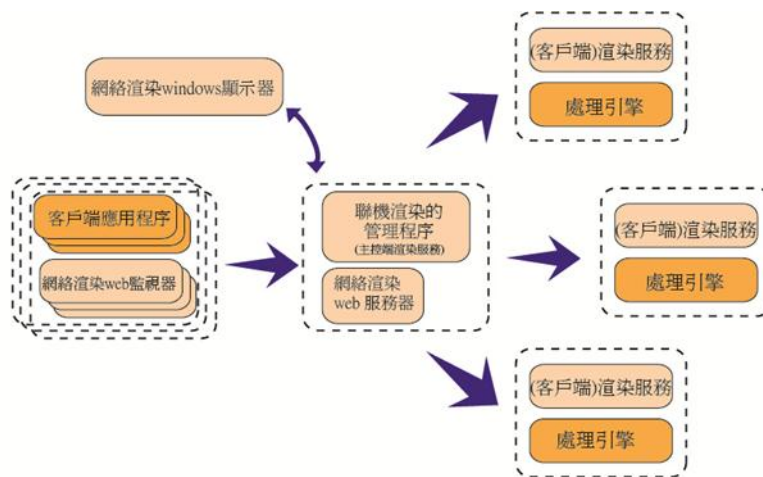


圖 3 網路渲染(net render)群組之工作流程圖[10]

本論文的研究內容將針對網路渲染(net render)群組進行相關的運算效能研究。

3.系統配置與架設

為了有效的運用現有實驗室的資源，本運算效能的案例分析直接使用本校的教學用電腦，以下針對計算機內的軟硬體設事進行規格與設定說明。

3.1 硬體建置

現有實驗室中電腦的配置如圖 4 所示。

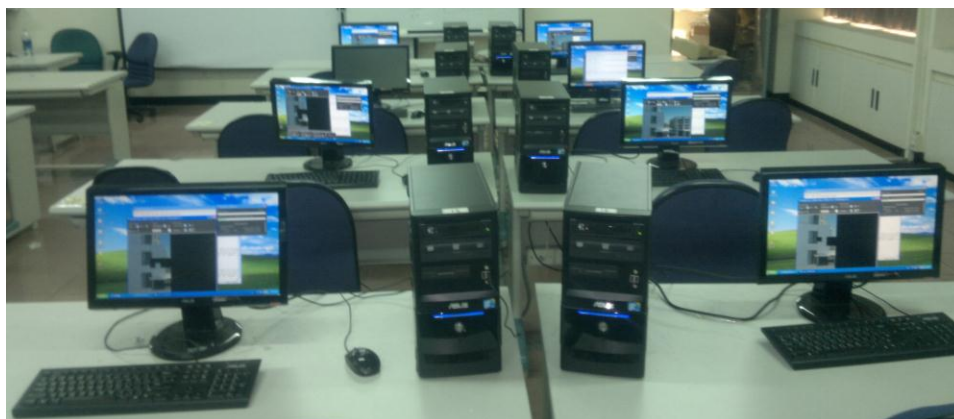


圖 4 實驗室架設運算系統後之設置結果

整間實驗室中取相同的 16 台擔任伺服器運算節點，另外一台擔任管理者及監控的角色，並由此管理者透過資源共享的方式，蒐集由各運算節點所完成的渲染效果圖。各單一電腦之軟硬體規格及網路連結說明如下

- (1) 計算機硬體規格：16 台運算節點的硬體規格如下
 - 中央處理器: Intel(R) Core(TM)2 Quad CPU Q8300 @ 2.50 GHz
 - 記憶體: 4GB DDR3
 - 圖形處理器: NVIDIA GeForce 9500GT
 - 作業系統: Microsoft Windows XP SP3
- (2) 網路架構：由於圖形渲染過程時有相當多的模型與渲染用資料，由管理者端傳送至運算節點，而渲染完成後又會將完成圖由節點傳送至管理端，因此本實驗過程中所採用的乙太網路交換器之規格如下。
 - 24 埠 10/100Mbps 與 2 埠 10/100/1000 Copper-T 自動偵測交換埠
 - 均提供 MDI-II/MID-X 自動辨別功能
 - 8.8Gbps 高速背板頻寬
 - 高速 Store-and-Forward 傳輸模式

整體運算節點與管理者之網路的配置如圖 5 所示。

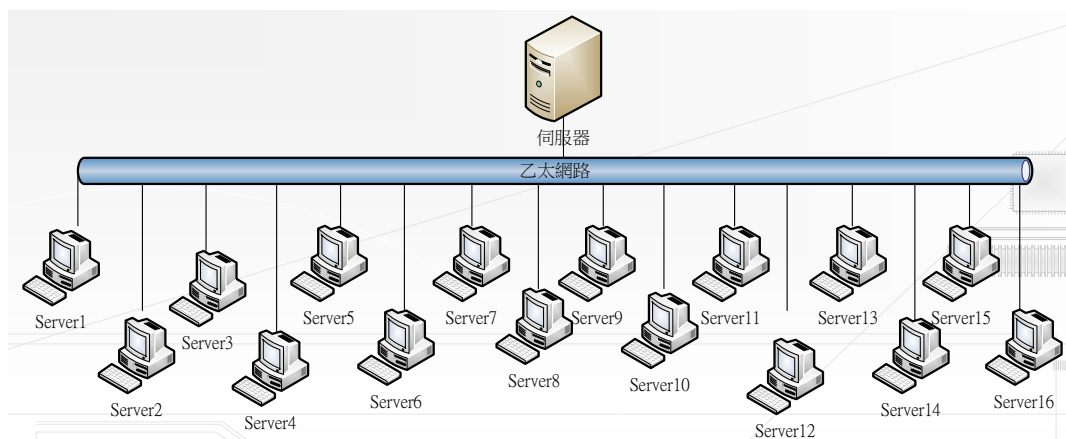


圖 5 分散式佇列系統網路連結示意圖

3.2 軟體配置

在本次的效能運算測試中，主要使用的是 Autodesk 3ds Max Design 2010 及 Backburner 2008.1 兩套軟體，兩者主要的設定介面分別說明如下。

(1) 3ds Max Design 2010: 要達成網路運算的要求，必須在渲染設定(Render Setup)的對話方框中，共用(common)選單下進行下述的設定，包括：在時間輸出(Time Output)欄位中，若是單張渲染圖輸出，則選擇單一(Single)選項；若是動畫渲染輸出，則是勾選範圍(Range)選項。在輸出大小(Output Size)欄位中，若是單張渲染圖輸出，則選擇較大的輸出尺寸；若是動畫渲染輸出，則是選擇較小的輸出尺寸。此外，在渲染輸出(Render Output)欄位中，其檔案(File)位置應選擇管理器所在位置的一個開放共用的資料夾，並且勾選網路渲染(Net Render)選項；相關設定的畫面如圖 6 所示。

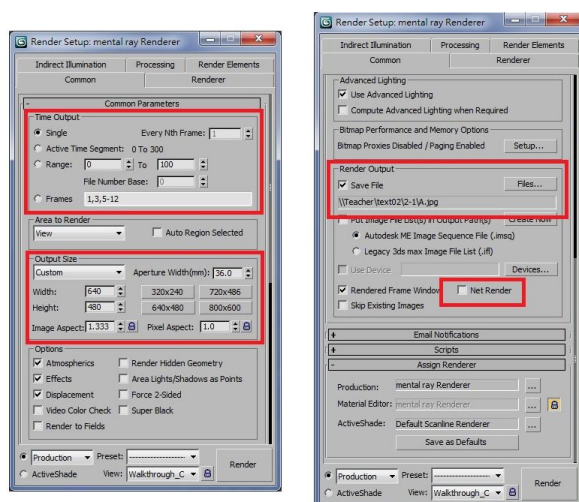


圖 6 3ds Max Design 2010 的渲染設定選項

(2) Backburner 2008.1: 在這套軟體中有三個部分必須分開設定，首先，必須在管理者上執行 Manager 程式，設定執行的工作(Job)名稱，以及連結的運算節點 IP 或網路芳鄰的位置；其次，在各運算節點上執行 Server 程式，此時會出現相關的等待訊息；之後在管理端按下執行工作，這時所有的運算節點都開始執行運算工作，我們可透過 Monitor 程式來監控每一台運算節點的工作情形，相關的操控介面如圖 7 所示。

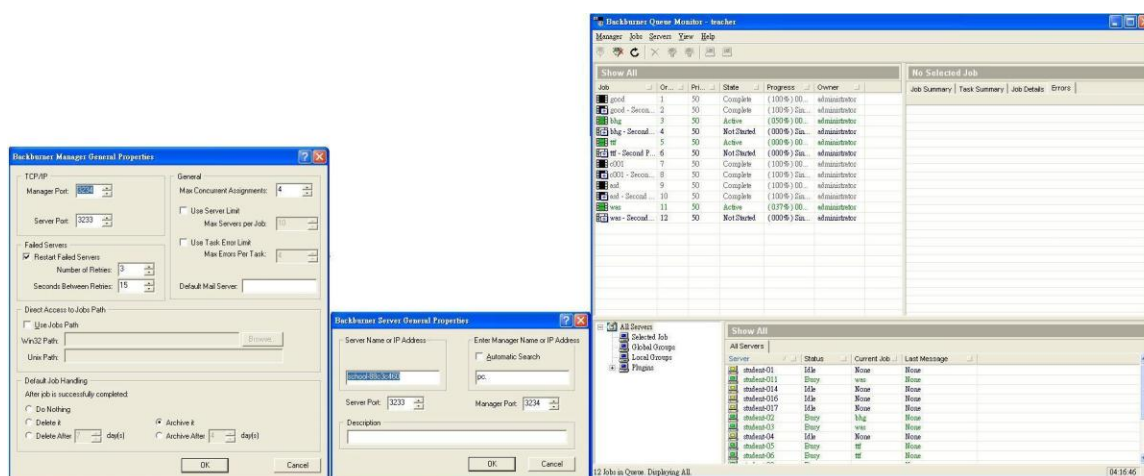


圖 7 Backburner 2008.1 的操控介面 (a)Manager (b) Server (c) Monitor

4. 案例分析

為了驗證整體分散式佇列系統的效能，我們選擇了兩個不同的案例來探討分析，一個是單一大型建築室內設計的視覺化效果圖，場景中有相當多家具模型、材質、貼圖與燈光，可以獲得單張圖形分區渲染的運算效能；另一個則是小型辦公室內的近真實動畫，透過多個影格的單影格分散渲染的方式，可以了解影格分散渲染的運算效能。

4.1 單張視覺化效果圖

在這個案例中，我們使用的場景如圖 8 所示。

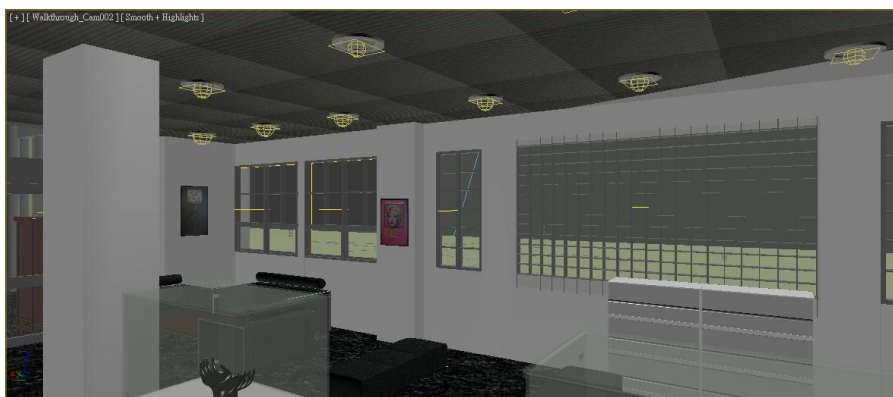


圖 8 單張視覺化圖形渲染所使用的場景

場景中三維模型的個數有 19 個，二維模型有 159 個，總計渲染面數(Faces)個數達 890947 個；另外材質(Material)與貼圖(Map)數量為 16 個；外加光源(Light)數 21 個。在網路群組渲染過程中，我們使用不同數量的運算節點進行單一圖形分區運算，之後再傳回管理器中進行圖片合成，圖 9 中顯示的是四台運算節點進行 1024X768 圖形大小尺次算圖完畢後的情形，以及在管理器中合成圖片的結果。

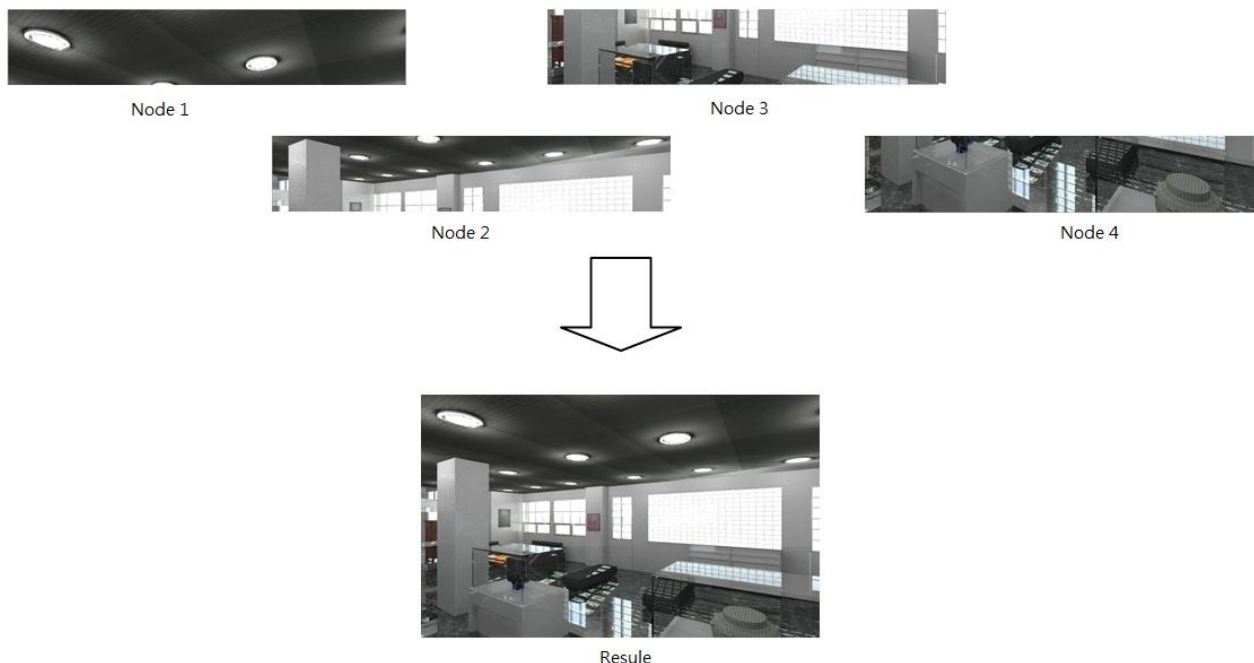


圖 9 四台運算節點渲染完成及管理器合成結果

從合成圖中的結果可看出，其接合的邊界並沒有出現色差或圖形扭曲的現象，看不

出有任何分區渲染的情形，這表示以分散式佇列系統進行分開運算的方式，所得的結果與單一計算機所獲得結果是一致的。接下來我們要了解不同運算節點數對渲染效能的提升，我們分別使用一台、二台、四台、八台與十六台運算節點，進行高(1920X1024)、中(1024X768)與低解析度(320 X240)的比較運算，以運算時間作為效能評斷的依據，表 1 所呈現的結果就是不同解析度圖形渲染所需的時間。

表 1 不同運算節點數對高低解析度圖形渲染所需的時間

	低解析度 (320X240)	中解析度 (1024X768)	高解析度 (1920X1024)
1	49 分鐘	348 分鐘	923 分鐘
2	41 分鐘	332 分鐘	705 分鐘
4	29 分鐘	200 分鐘	462 分鐘
8	20 分鐘	107 分鐘	224 分鐘
16	9 分鐘	64 分鐘	148 分鐘

從表中可以看出，運算節點越多，所需的算圖時也相對的減少，但其呈現的並非是線性的趨勢，以高解析度圖形來說，從一個運算節點到兩個運算節點時，後者所耗費的時間約為前者的 5/6；但是到了 16 個運算節點時，其所耗費時間則為一個節點的 1/5；由於透過網路進行資料數據的傳輸，在節點數多的時候，網路傳輸的時間反而會成為耗費時間的主因，由此也可以推斷，增加再多的運算節點，其所能減少的運算時間也相當的有限。此外，在表中可看出低解析度的渲染圖，採用多運算節點所獲得的效能比起高解析度圖形來的低，其原因也是在低解析度時，單一節點的運算效能即足以快速的渲染出圖，用多節點反而增加的分割圖形與網路傳輸時間。

4.2 動畫渲染

在這個實作過程中，我們使用的場景如圖 10 所示。



圖 10 動畫渲染所使用的場景

場景中攝影機所行進的路徑固定，各影格均使用 640X480 尺寸進行渲染，但使用不

同數量的影格數來進行分散影格運算，之後再傳回管理器中進行影格圖片排序，動畫的製作因採用另外的製作程序，則不列入渲染耗費時間中。圖 11 中顯示的是 100 個動畫影格完成後所獲得 20, 40, 60, 80, 100 影格的渲染圖。

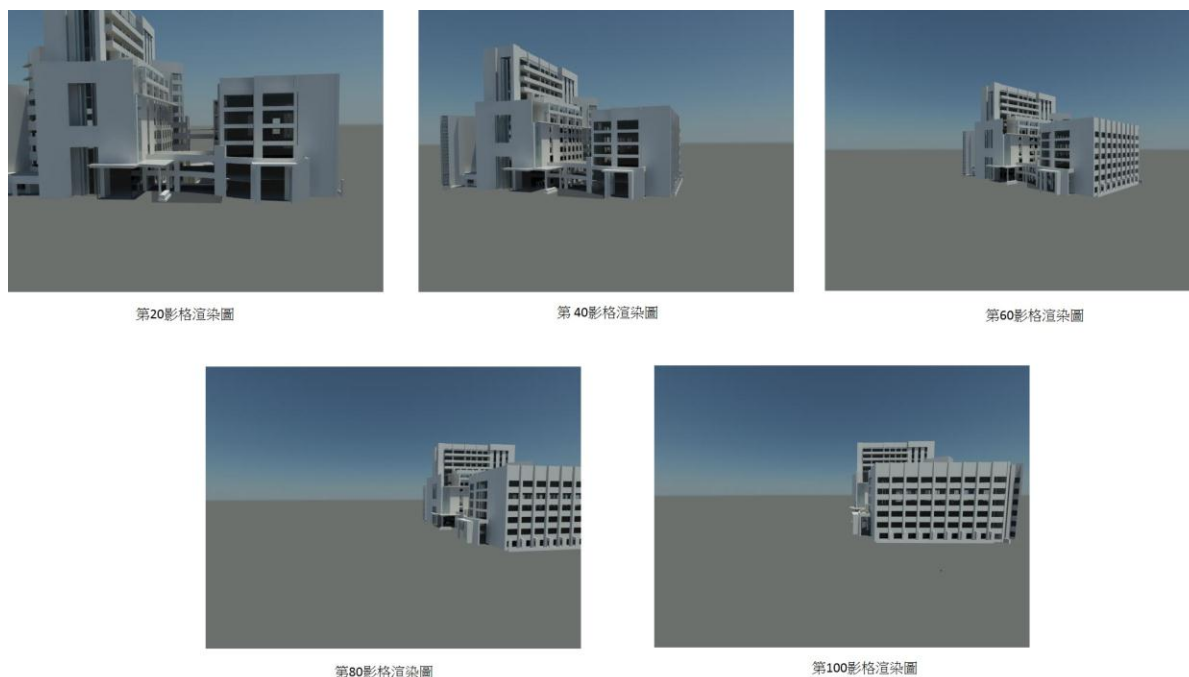


圖 11 100 個動畫影格之第 20, 40, 60, 80, 100 影格的渲染圖

接下來我們要了解不同運算節點數對動畫渲染效能的提升，同樣地，我們分別使用一台、二台、四台、八台與十六台運算節點，進行 100, 200, 300 個動畫影格渲染的比較運算，以運算時間作為效能評斷的依據，表 2 所呈現的結果就是不同影格數對動畫圖形渲染所需的時間。

表 2 不同影格數對動畫圖形渲染所需的時間

	100 個影格	200 個影格	300 個影格
1	123 分鐘	179 分鐘	329 分鐘
2	54 分鐘	88 分鐘	176 分鐘
4	25 分鐘	44 分鐘	87 分鐘
8	17 分鐘	25 分鐘	43 分鐘
16	9 分鐘	17 分鐘	32 分鐘

從表中可以看出，運算節點越多，所需的算圖時也相對的減少，但其呈現的非常接近線性的趨勢，以 100 個影格圖形來說，從一個運算節點到兩個運算節點時，後者所耗費的時間約為前者的 1/2；到了 16 個運算節點時，其所耗費時間則為一個節點的 1/13；由於影格進行分散運算，在節點數多的時候，有更多的運算節點將可以讓單一節點所需渲染的影格數減少，由此也可以推斷，增加更多的運算節點，其所能減少的運算時間也應該相當的顯著，當然考量網路傳輸的需求，並無法達成絕對的線性衰減趨勢。此外，在表中可看出少影格(100)的動畫渲染，與多影格(300)的動畫渲染，所獲得的運算所需時間之衰減趨勢一致，其原因也是跟上述說明相同。

5. 結論與建議

本文以分散式佇列運算系統進行視覺效果圖渲染，透過軟硬體的架設及兩個不同場景案例的探討與分析，我們可獲得以下數點結論：

(1) 分散式佇列運算系統對建築設計視覺化渲染，不論對單張視覺效果圖或是動畫製作，均有有相當大的效能提升。

(2) 對於單張視覺效果圖而言，以分散式佇列運算系統進行高解析度圖渲染，比低解析度圖渲染所獲的效能提升更為顯著；同時，運算節點數達到一定數量時，再增加數量對效能的提升也有限。

(3) 對於動畫影格渲染而言，分散式佇列運算系統對少影格數與多影格數所獲得的效能提升趨勢一致，表示此系統用在動畫製作上，有非常顯著的效果。

參考文獻

1. Chromium Homepage, <http://chromium.sourceforge.net/doc/LLNLcopy.html>
2. V. Bacu, L. Muresan, D. Gorgan, "Cluster Based Modeling and Remote Visualization of Virtual Geographical Space", Workshop on Grid Computing Applications Development (GridCAD) at the SYNASC Symposium, Timisoara, September 2008.
3. Equalizer: Parallel Rendering, <http://www.equalizergraphics.com/>
4. D. Gorgan, C. M. Mocan, V. Bacu, "Remote Graphical Visualization of Interactive Virtual Geographical Space", 32rd International Convention on Information and Communication Technology. Electronics and Microelectronics, MIPRO, Opatija, May 2009.
5. F. Goetz and G. Domik, "Remote and Collaborative Visualization with openVISSAR", Proc. of the Third IASTED International Conference on Visualization, Imaging, and Image Processing, Benalmádena, Spain, September 2003.
6. Ogre Graphics Engine, <http://www.ogre3d.org>
7. Irrlicht Engine, http://en.wikipedia.org/wiki/Irrlicht_Engine
8. C. M. Mocan, D. Gorgan, "Cluster Based Modeling and Graphical Visualization of Interactive Large Spatial Data", MIPRO 2010 Proceedings of the 33rd International Convention, p 258-263, 2010.
9. Autodesk Backburner 2008.1 Installation Guide, Autodesk document.
10. Autodesk Backburner 2008.1 User Guide, Autodesk document.

Rendering on Architecture Design Visualization by Distributed Queuing Processing System

Bi-Wen Lee¹, Chien-Fu Chui², Ruo-Chen Li², Yen-Jou, Chan², Hwei-Jyun, Jhang², Yun-Cheng, Chang², Jyun-Han, Jhou², Hao-Wei Wu², Er-Ting Wang², Yu-Tuan Chou³

¹ Shih-Chien University, clioli1213@hotmail.com

² Chia-Nan University of Pharmacy and Science, r0926852901@yahoo.com.tw

³ Chia-Nan University of Pharmacy and Science, ytchou@mail.chna.edu.tw

Abstract

For obtaining the realistic image of design visualization, a lot of computer effort is needed in the rendering process. It will be spent massive time by one computer operation. In this paper, the distributed queuing operation system is setup by the existed laboratory computer, and the commercial software, 3ds Max Design 2010, Backburner 2008.1 are implemented for studying the rendering performance in architectural visualization.

The result is shown that the rendering performance is increased by using distributed queuing system. At single picture, the increasing effort in high-resolution rendering is better than low-resolution; At multiple frames of animation, the increasing effort in lots of frames are similar as little of frames. It can be easily implemented in the visualization rendering process by this studying approach.

Keywords: Design Visualization, Distributed Queuing System, Net Render, Computer Networking