在 GPGPU 平行架構下封包內容過濾與分析之研究

洪哲倫
靜宜大學資訊傳播工程學系
clhung@pu.edu.tw

張金源
靜宜大學資訊管理學系
a3014006@gmail.com

王孝熙
靜宜大學資訊管理學系
hhwang@pu.edu.tw

摘要

網際網路技術日以千里的迅速發展下，提供了各式各樣的應用與服務，而這些服務都需要依賴網際網路環境。因此，網路封包過濾的即時入侵偵測對於網路安全是相當重要的議題。封包內容過濾需要一個高效能的過濾機制，針對每一個封包進行過濾，然而在傳統的封包內容過濾是需要耗費大量時間來處理資料，為了提高封包內容比對效率，以實現大量封包數據及內容相依性的即時封包偵測需求，我們針對封包內容比對提出平行封包內容比對演算法應用於 GPGPU 高效能平行架構來提升過濾速度，以加強網際網路資訊的安全性。另外，我們也針對 CUDA 設備上各種記憶體架構進行探討，對於封包內容比對整體效能之影響，實驗結果顯示，GPGPU 顯著地提升了封包分類的整體效能。

關鍵詞：封包內容比對、GPU、CUDA、平行處理、網路入侵即時偵測。

壹、前言

雲端平行化運算架構自 2007 年由 Google 與 IBM 開始在美國大學校園，包括卡內基美隆大學、麻省理工學院、史丹佛大學、加州大學柏克萊分校及馬里蘭大學等，推廣雲端運算的計畫開始，近年來已經成為網路架構的主流趨勢。因此，伴隨而來各式各樣的雲端應用與服務，提供更多使用者使用，而這些服務都深深的依賴著網際網路。然而，為了維護網際網路上資訊的安全，對於日新月異的入侵封包是不夠的，雖然防火牆可阻擋部分的入侵及攻擊，但無法真正的阻擋攻擊行為[1, 2]。雖然在現今複雜的網路環境下，入侵偵測需處理大量的網路封包，執行信息報警(Alert)的工作，通知網管人員處理或可透過防火牆來阻隔，讓網路安全獲得適時的管控。近年來網際網路安全的應用，如入侵檢測，關鍵字阻擋，垃圾郵件過濾和阻擋病毒，都需要更仔細和更複雜的內容過濾。但在內容過濾部分是非常耗時，且字符串匹配通常佔有相當大的工作量[3]。因此，有必要設計一個適當的字符串匹配加速器或利用平行化架構，以減少工作量。

由此可見，比對封包內容對於網路安全是非常重要的[4]。許多相關演算法也紛紛被提出，常見的演算法如 Morris-Pratt Algorithm、Aho-Corasick、Bloom filters 等。在傳統的封包內容過濾演算法需要耗費大量時間來處理封包資訊，因此網路常見的攻擊威脅雲端運算平台上的資訊安全，以致於需要一套高效能即時過濾網路封包內容，來偵測及分析可能的入侵攻擊及非法存取行為。

在現今高速網際網路環境下，如大型資料中心(Data Center)、雲端服務(Cloud Computing Service)，這些環境中的網路封包數量非常龐大，如果透過單一裝置來分析比對網路封包內容，將會造成該裝置的超量負載，這將會導致該裝置當機，造成整個系統無法運作[5]。若需要達到高效能的即時網路封包分析比對，並且能夠符合網路服務品質（QoS），僅透過單一裝置是很難達到的，因此利用平行運算來進行封包比對即為最為理想的方案。現今有許多研究嘗試使用圖形處理器(General-purpose computing on graphics processing units，簡稱為 GPU) 來解決資料密集型的計算問題[6, 7]。所以在許多不同的研究領域都紛紛提出，利用 GPU 進行各領域的運算來提其高效能。利用 GPU 的多核心特性來處理不同領域上的計算問題，已有許多成功的研究，此證明

許多問題可運用 GPU 平行運算來處理。然而，使用 GPU 最大好處在於處理核心的數量眾多(在 CPU 中可使用的核心 2~16, GPU 核心約 128~1024)。在 2006 年，nVidia 提出了統一計算設備架構(Compute Unified Device Architecture, CUDA)。CUDA 架構採用了全新的單指令多執行緒(SIMT)[8]。這種架構允許所有的執行緒(threads)執行獨立且發散的指令流(instruction streams)，這並不是使用常見的單指令流多資料(SIMD)。

在本篇論文中，我們提出一個有效的方法利用 GPU 來進行大量的封包內容比對過濾。利用 nVidia CUDA 協同處理器(GPU)的低成本架構相較 CPU 多核，達成加速封包內容比對的處理量(throughput)，使得封包內容比對可以應用在及時性與高辨識率的即時網絡監控擷取分析。我們實作了封包內容比對的演算法於 GPU，並探討各種記憶體架構的性能差異。實驗結果顯示，相較於單一 CPU，效能可高達 10.x 倍，証明 GPGPU 不僅僅在即時封包分析是可行的，同時也可運用於封包內容比對。在第二章中，簡要說明

封包內容比對過濾的相關工作，第三章，描述我們提出的方法。在第四章，展示實驗的結果。最後在第五章，說明本篇論文的結論。

貳、參考文獻與相關工作

隨著科技日以千里的進步，平行化運算架構被提出。利用平行化架構可以提高計算效能，近幾年被各研究領域重視。然而在這樣高科技之下網路安全更是不容忽視。網絡入侵檢測系統（NIDS）檢測到許多惡意活動，如拒絕服務攻擊，端口掃描，甚至試圖打擊到電腦，通過監控網絡流量。因此，字串匹配的入侵檢測系統是一個重要方面[9]。在網際網路封包內容比對過濾方面已有許多研究單位提出演算法及許多相關研究，以常見的演算法如 Morris-Pratt Algorithm、Aho-Corasick、Bloom filters。目前大多字串比對演算法都需要在掃描內容花費很多時間[10]，因此這些演算法是非常緩慢的。傳統字串比對中 Aho-Corasick 為效能及精準度較佳的演算法，此演算法的比對速度較佳[11]，適合用於多樣本比對。而其缺點為占用大量記憶體，如果能有效運用記憶體來提升效能將成為重點。但在過濾演算法中，Aho-Corasick 具有在最差的情況下，最佳效能的多重樣本比對演算法。

近年來平行化運算概念逐漸被提倡，具有多核心的 GPU 裝置也漸漸被重視，由於 GPU 運算能力越來越強大，因此有許多研究單位紛紛進行各領域研究[12]。GPU 不同於傳統的平行運算概念，為 Streaming 的處理架構與平行資料模型，可同時進行眾多資料流中獨立處裡相同指令。而現今 GPU 已包含了數十至數百個 Stream Processor，使得 GPU 平行處理與一般科學平行計算正面臨著展新的挑戰[13]。nVidia 所發布了 Compute Unified Device Architecture (CUDA) SDK[14]，協助各領域研究人員應用 GPU 進行非圖形運算，CUDA 所提供的包含相關函式(Function)可執行於 CPU 或 Host，較小且較為密集的平行計算函式執行於 GPU 中之 Kernel。

然而使用專門的入侵偵測硬體設備執行內容平行比對，能有效的提升比對速度[15]，所以封包內容比對用於平行架構可有效提升比對速度，但這些專業的硬體設備都需要較高的硬體成本[16]。相反，現代 GPU 的設計成本低，而圖形裝置製造商更靈活的的增加了許多相關函式可編程，且提供高層次的 API，提供高的編程能力，並進一步確保版本相容性。其設計成本低，高度並行的計算，通常得到充分利用，適合使用於較費時的計算問題。

目前許多論文利用 GPU 進行字串比對演算法加速，大部分為分工演算法 (demultiplexing algorithms) 採用的皆是決策樹的方式。傳統的決策樹的演算法較不適用於 GPU 平行運算，對於在封包內容複雜度極高的環境下，這些研究論文提出利用 GPU 進行內容過濾相關研究，將 Snort 實作於 GPU 架構運算[17]，此論文方法將封包切割成相同大小長度，交由執行緒進行計算比對，這種方法的優點是，所有執行緒都分配相同的工作量，而缺點是 GPU 區塊重疊，尤其是在小數據包的情況下，資料需要額外處理程序。而在[18]研究論文中將不同長度封包切割交由 Block 的執行緒比對，執行緒依照不同長度封包進行封包內容比對，在封包數量較大時過濾可達數十倍。然而在不同封包

長度時，在 GPU Block 會有許多執行緒閒置。而現今在 GPU 上的封包內容比對研究都利用傳統演算法實作於顯示裝置，將封包切割成小部分交給 GPU 執行緒執行。

現今許多論文提出利用 GPU 加速在字串比對[19]，其顯示裝置 NVIDIA GeForce GTX 460 與 GTX 480 (此裝置分別具有 336 的核心、480 的核心)，內容比對效能提高數十倍，但這些顯示裝置其核心數都具有較多核心。由這些研究顯示 GPU 使用對於字串比對效能高於過去傳統的 CPU 架構[17]，但不是每一種字串比對演算法都適用於 GPU 架構。本研究中使用的顯示裝置具有 Fermi 的架構下可透過 L2 進行快取，因此效能比過去 CUDA 架構優化，使執行緒可以透過此快取來讀取 Global Memory 的資料。而封包內容比對我們將提出一個較適合 GPU 的演算法架構來處理高複雜度的網路封包內容比對。利用 NVIDIA GeForce GTS 450[20]其設備具有 192 的核心實作較適合於 GPU 架構的演算法，應用在高複雜度的封包內容比對其效能也可達到數十倍效能，並且對於 GPU 的各種記憶體進行探討。

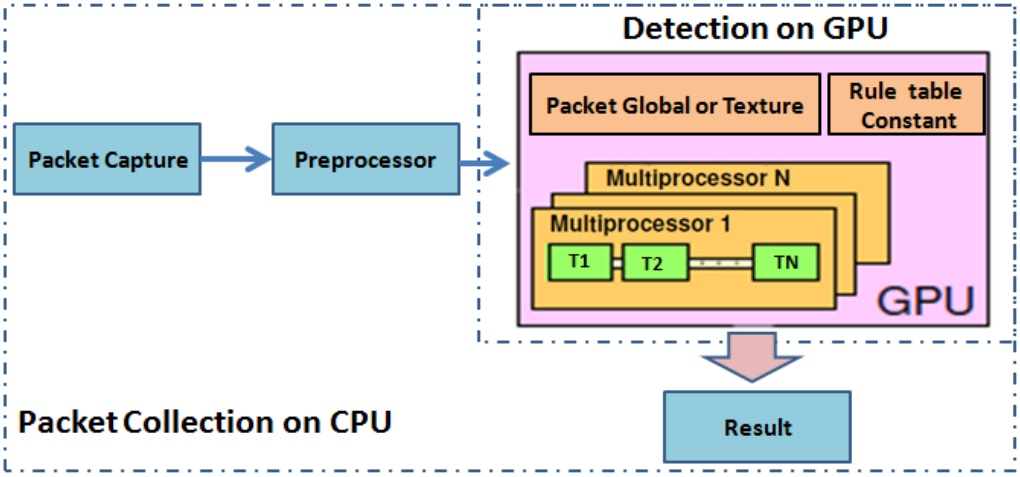

圖 1 整體架構

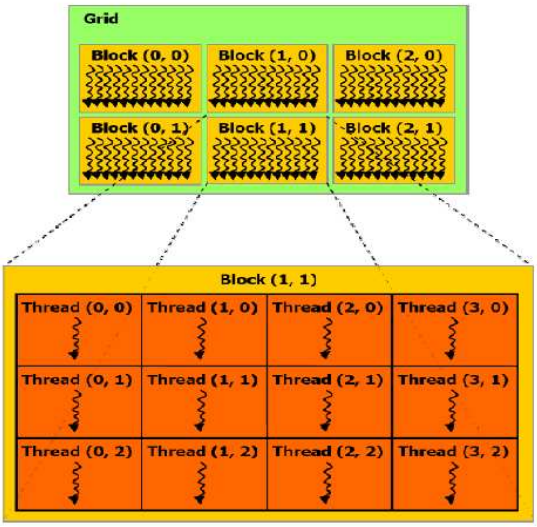

圖 2 GPGPU 平行架構

參、研究方法與執行步驟

在 GPU 架構中 Kernel 將執行緒分配儲存至區塊(Block)，每一個區塊中包含 1024 個執行緒，而這些區塊會定義成一維或二維的網格(Gird)。換句話說，每個執行緒定位於區塊，而每一個區塊則定位於網格(Gird)。硬體架構中每一個區塊由一個多處理器(multiprocessor)所管理與執行，區塊與區塊之間可通過共享記憶體(shared memory)互相傳遞資料。由圖 2 說明了 GPGPU 的平行架構。Kernel 執行時包含許多的執行緒，執行緒執行時獨立計算小部分的資料。如果要利用 GPU 進行運算需先將資料傳入至 GPU，這必須透過主記憶體與 PCI-E bus 之間來傳輸。運算後結果將會儲存在 GPU 的記憶體中，因此所有的記憶體都必須在運算前配置定位。

一、 CUDA 的記憶體架構

在 CUDA 架構中提供了五種記憶體類型，全域記憶體(Global Memory)、常數記憶體(Constant Memroy)、 材質記憶體(Texture Memory)、共享記憶體(Shared Memory)以及暫存器(Register)。GPU 記憶體都會受到各種傳輸的延遲與存取上的限制，因此 GPU 運算效能決定在於各記憶體種類的配置與使用。圖 3 說明了 GPU 顯示裝置的記憶體架構。全域記憶體為 GPU 中最大的記憶體，任意執行緒都能讀寫全域記憶體，適合用於儲存大量資料。最新的 Fermi 的架構下可透過 L1 與 L2 進行快取，因此效能比過去 CUDA 架構優化。常數記憶體位於 GPU 上的 DRAM，屬於唯獨(Read-only)記憶體，可存取全域記憶體上的所有執行緒。此記憶體具有 8K 快取機制，用以節約頻寬加快存取速度。材質記憶體是位於 DRAM 上的快取記憶體，在 GPU 中若需要使用到 2D 或 3D 的存取方式，通常需透過材質記憶體的特殊存取方式來達成。共享記憶體是 on-chip 的快取記憶體，每個區塊都擁有一個共享記憶體，大小僅只有 48KB。GPU 暫存器如同於 CPU 暫存器，此暫存器用來分配與管理 Kernel 中的執行緒，若是超量使用將會大幅將低效能。

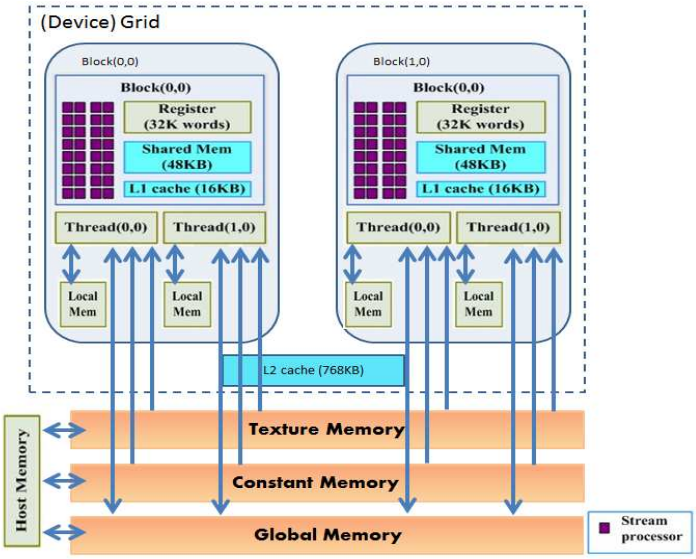

圖 3 NVIDIA GeForce GTS450 記憶體架構

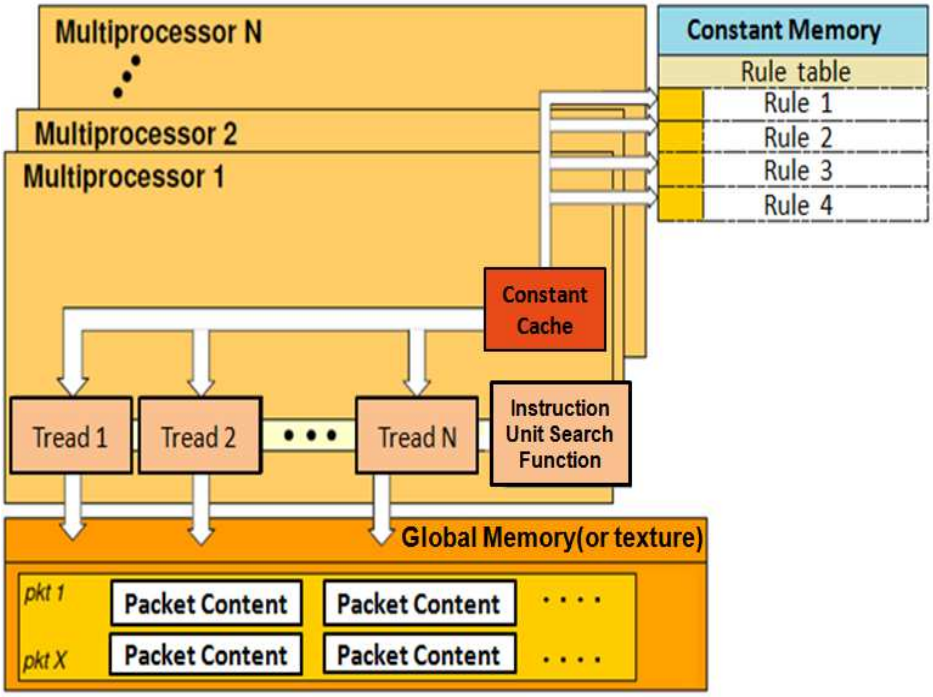

圖 4 GPU 封包內容平行比對架構圖

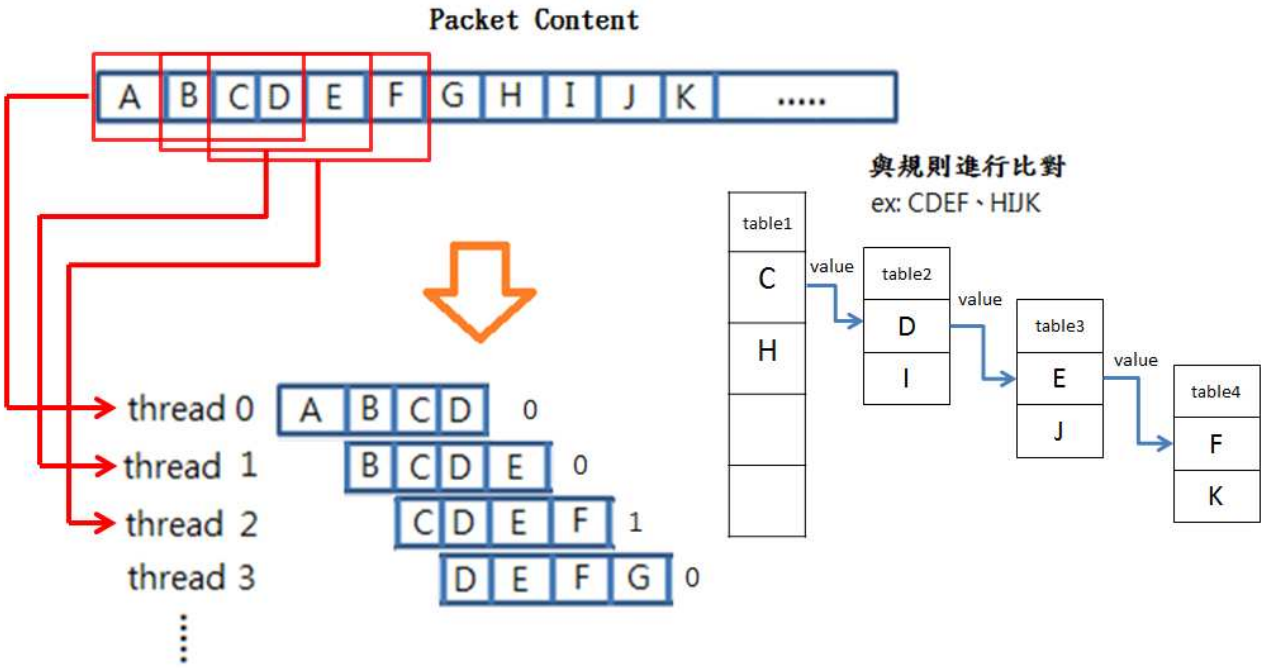

圖 5 執行緒比對模式

二、 GPGPU 封包內容平行比對

GPU 中，每一個實體 Multiprocessor(MP)包含一個暫存器，並同時執行 8 個獨立的處理核心(Processing cores)。Multiprocessor 發送出指令後，暫存器會對所有的執行緒指定行進路線執行，此時無任務之執行緒將進入待命[8]模式。在封包內容平行比對中，我

們將使用率高且更新率低的過濾規則儲存於常數記憶體或暫存器，龐大的封包資料儲存於全域記憶體或材質記憶體，執行時執行緒依照過濾規則，比對每一個獨立的封包內容資料。本研究中提出一個較適合 GPU 的演算法於 GPU 與 Aho-Corasick 演算法於 CPU。我們所設計的演算法是利用 Hash Table 的概念，在常數記憶體中放置我們自行設定之關鍵字其長度為 4 的字元，分別設置四個資料表，每一個資料表中存放一個字元依序存放關鍵字(圖 4)。而封包內容長度一般為 1518，但在 GPU 的架構中資料需符合 32 的倍數所以我們將封包長度設置為 1536。

在實驗中我們利用每一條執行緒比對讀取封包資訊與關鍵字規則進行比對，因此 GPU 在比對封包內容時不會錯過關鍵字能快速比對封包。封包內容進行比對，首先每一個執行緒分別讀取封包內容中的 4 個字元(如 ABCD 與規則作 Hash，與關鍵字相符字元回傳 1)與常數記憶體中關鍵字資料表比對，如完全符合關鍵字回傳 1 反之則回傳 0。次之，由執行緒平行計算每一個封包是否需要過濾需要過濾則回傳 1 不需要則回傳 0，將結果回傳給 CPU。例如待比對封包內容為 A~K，將此內容交由我們所提出的 GPU 封包內容比對方法中，執行緒會將封包內容以 4 個字元為單位分割，分割後分別讀取 4 個字元，如執行緒 0 則讀取字串中 ABCD、執行緒 1 讀取 BCDE、執行緒 2 讀取 CDEF，與常數記憶體中的規則表比對，如完全符合則回傳 1 反之回傳 0 如圖 5。與規則表比對後結果，再由執行緒分別計算每一個封包內是否具有關鍵字，把結果回傳給 CPU 是否需要過濾此封包。

本研究提出一個簡單且較適合 GPU 之演算法，並探討 Hash Table 結構是否適用於 GPU 架構下封包內容比對。另外，我們探討資料傳輸時的延遲，利用 ZeroCopy 的技術是否有效提升整體比對較能。由[8]中所提出 ZeroCopy 的方法可用來降低 GPU 上的之間的傳輸時間，以及搭配 GPU 的 Texture memory 與 Global memory 之間效能差異。

肆、實驗結果

我們所設計實作的演算法，使用 NVIDIA GeForceGTS graphics card (Fermi 架構)以及 Intel CPU i3 540 GHz 與 4G DDRIII-1333 記憶體，環境為 Linux 作業系統(如表 1)。在實驗中我們模擬亂數封包，以封包長度 1536 長度為標準，最多可容納 4 萬條封包內容。實驗封包過濾關鍵字規則共有 1200 個規則，每個關鍵字 4 個字元，透過執行緒進行比對。

表 1 實驗環境

CPU	Intel CPU i3 540 GHz
記憶體	4G DDRIII-1333
GPU	NVIDIA GeForceGTS 450 (Fermi 架構)
作業系統	Linux

表 2 各種記憶體搭配

	Packet Location	Rule Location	Data Transfer
1	Global	Constant	NA
2	Global	Constant	Zero Copy
3	Texture	Constant	NA
4	Texture	Constant	Zero Copy

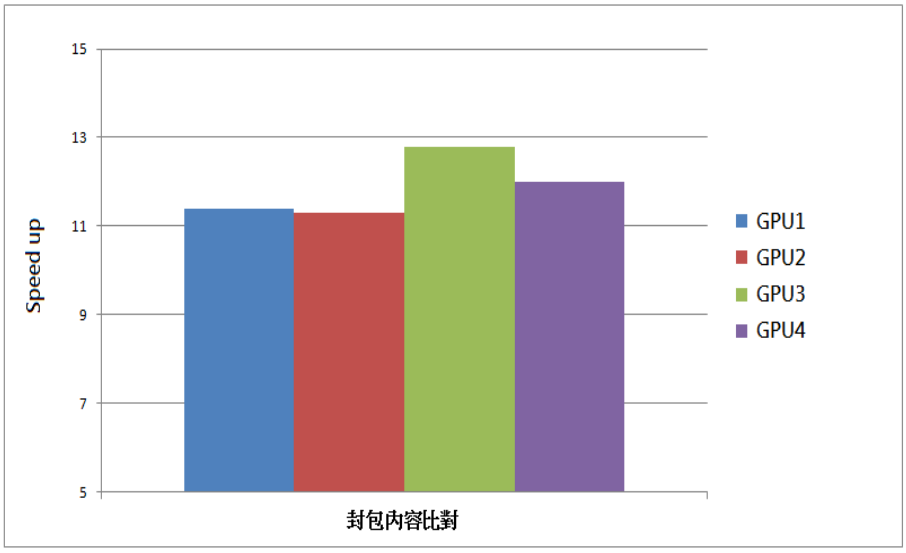

圖 6 封包內容比對方法之效能比較圖(註 1)

一、 記憶體之間的效能差異

實驗中我們利用 GPU 各種記憶體進行 4 種搭配，因 GPU 掩飾裝置暫存器容量較小，故實驗中不採用，表 2 為我們實驗 GPU 封包內容比對記憶體搭配表。我們將存取率較高的關鍵字規則配置於常數記憶體中，將低更新率的資料結構配置於全域記憶體與材質記憶體。在我們的實驗設計演算法中應用於比對封包內容之 GPU 效能，如圖 6 所示，結果顯示出 GPU 分類比 CPU 分類速度高於 11.x，利用 ZeroCopy 資料傳輸可以大大減少資料上傳延遲的缺點，整體效能可提高將近 13.x。

由實驗可得知 GPU 實作封包內容比對能有效提高效能，而實驗結果顯示利用 ZeroCopy 的資料傳輸將封包上傳至 GPU 裝置，可以降低上傳資料所花費的時間效能可提升將至 13.x，明顯比 CPU 封包內容比對效能要優化。在實驗結果顯示，GPU 封包內容比對演算法很明顯比 CPU 封包內容比對演算法效能高出許多。而在比對時間上，全域記憶體搭配 ZeroCopy 資料傳輸可使比對時間降低很多，材質記憶體因為有 8k 的快取限制，以至於比對速度無法明顯加速如圖 7。

註 1: AC 於 CPU、GPU1 於 Global Memory、GPU2 於 Texture Memory、GPU3 於 Global 搭配 ZeroCopy、GPU4 於 Texture 搭配 ZeroCopy

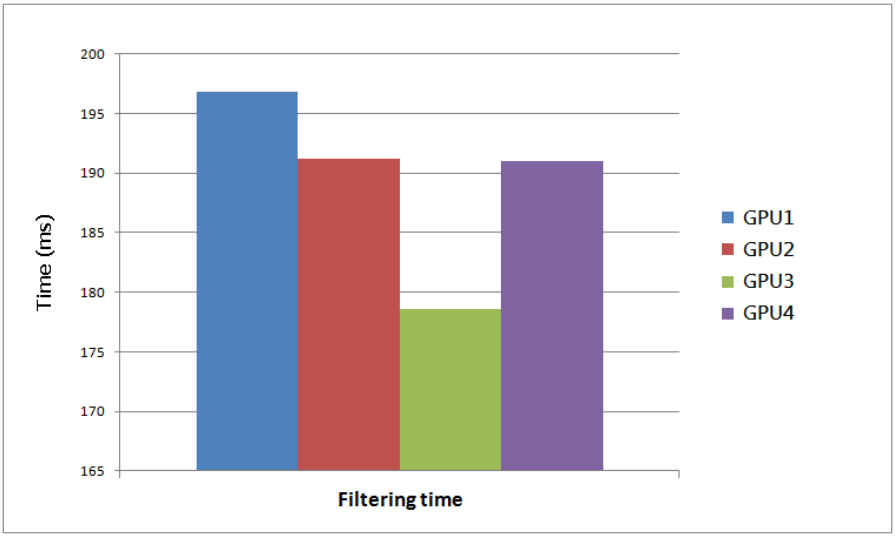

圖 7 內容比對時間

二、 GPGPU 整體效能評估

在記憶體上全域記憶體與材質記憶體，全域記憶體效能較佳。這是因為早期 CUDA 架構全域記憶體是沒有快取，因而需要利用材質記憶體的快取來提升運算效能。而最新的 CUDA Fermi 架構全域記憶體提供 768KB 的 L2 快取記憶體，而材質記憶體有 8K 快取，以至於全域記憶體效能表現較佳於材質記憶體。在比對演算法部份，RCF 架構利用 Hash table 的方式進行比對，而每一個執行緒獨立比對將結果存在全域記憶體，比對與關鍵字相符則回傳 1 反之回傳 0，執行緒再判斷該封包是否需要過濾，將結果回傳給 CPU，與我們熟知的 CPU 分支預測不同。

CPU 在不同規則數量比對，我們測試 100 至 1200 的規則，規則越多比對所花費時間越多。而 GPU 在封包內容比對不會因為規則數量的多寡而影響封包比對速度，其比對吞吐量可趨近 2.5 Gbit/s 。本實驗也依據不同封包長度進行測試，由 128 至 1536 可知在不同封包長度 CPU 比對時間沒有明顯的變化，而 GPU 因為不同封包長度資料傳輸耗費時間以及比對時間不同受到影響，但其比對速度依然明顯優於 CPU； GPU 利用 ZeroCopy 進行資料傳輸，封包長度大小影響 Throughput 連續傳輸之時間，其比對速度依然有明顯優於 CPU 比對，但比對長度達 1536 全域記憶體受到 L2 快取的影響所以使用 ZeroCopy 傳輸時效能沒有明顯提升。

伍、結論與未來計畫

在本研究中，我們設計並建立了封包內容並行比對，作為處理網際網路上的入侵封包內容偵測的解決方案。在論文中，我們證明了使用 CUDA 所開發的 GPGPU 實作封包內容比對是可行的。並且探討了我們所提出的演算法其資料結構下封包內容比對演算法。實驗中我們的架構有效提升封包內容比對的效能高達 13.x，明顯比 CPU 效能要優化。不同規則數量不影響我們提出的比對方法，GPU 可以維持一定的比對速度。在於不

同封包大小因為資料傳輸所花費時間及比對時間不同，資料傳輸上我們利用 ZeroCopy 試圖降低資料傳輸時間提升效能，但全域記憶體受到 L1 快取的影響，所以全域記憶體沒有明顯提升效能,而材質記憶體受到 8K 的快取影響及利用 ZeroCopy 連續傳輸資料效能不佳。而在比對的 Through put 上實現了高性能，處理吞吐量超過 2 Gbit/s，遠比 CPU 處理量高出許多。

此外也探討了 GPU 各記憶體架構之間的差異，雖然不同記憶體對於整體效能有著不一樣的影響，但是這些影響卻也有關鍵所在，只要能善加利用這些記憶體特性，便可得到更好的效能提升，來達到最終需求。不同規則數量不影響本研究所提出的比對方法，而不同封包長度則影響比對及上傳至 GPU 之時間，但封包內容比對的速度依然優於 CPU 比對速度。在未來我們目標是建立一種過濾演算法，更有效利用 GPU 架構特性使封包內容比對效能更加優越及在 GPU 減少使用分支指令。因此我們注重在改善目前的方法，並提供 Giga-bit 即時過濾與快速封包分析應用。

陸、致謝

本研究受行政院國家科學委員會補助(NSC 100-2221-E-126-009 及 NSC 99-2632-E-126-001-MY3)，特此致謝。

參考文獻

1. 林仁傑,"自動排序入侵偵測," 逢 甲 大 學 資訊工程學系碩士班 碩 士 論 文, 2004,1.
2. 陳永烈,"以入侵偵測系統為基礎之主動式網頁過濾及阻擋機制," 逢甲 大 學 資 訊 工 程 學 系 碩 士 班 碩 士 論 文, 2004,6.
3. Y.-D. L. a. T.-H. L. Y.-C. L. Kuo-Kun Tseng, "A Parallel Automaton String Matching with Pre-Hashing and Root-Indexing Techniques for Content Filtering Coprocessor," *Proceedings of the16th International Conference on Application-Specific Systems, Architecture and Processors* 2005.
4. L. B. JIANG Bo, "High-Speed Discrete Content Sensitive Pattern Match Algorithm for Deep Packet Filtering," *Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing* 2003.
5. A. N. a. B. Irwin, "Parallel packet classification using GPU co-processors," *SAICSIT Conf.ACM,* pp. pp. 231-241, 2010.
6. A. N. a. B. Irwin, "GPU packet classification using OpenCL: a consideration of viable classification methods," *SAICSIT Conf.ACM,* pp. pp. 160-169, 2009.
7. P. T. M. Charalambous, and A. Stamatakis, "Initial experiences porting a bioinformatics application to a graphics processor," *Advances in Informatics,* pp. pp. 415-425, 2005.
8. C. NVIDIA, *NVIDIA cuda c best practices guide, version 4*, March, 2011.
9. S. R. Subramanian N, "Content-Split based Effective String-Matching for Multi-Core based Intrusion Detection Systems," *2009 First International Conference on Computational Intelligence, Communication Systems and Networks,* 2009.
10. K. Z. Ming Gao, Jiahua Lu, "Efficient Packet Matching for Gigabit Network Intrusion Detection using TCAMs," *Proceedings of the 20th International Conference on Advanced Information Networking and Applications,* 2006.
11. J. L. SarangDharmapurikar, "Fast and Scalable Pattern Matchingfor Content Filtering," *2005, Princeton, New Jersey, USA, ANCS'05,* October 26-28,2005.
12. J. D. Owens, "A Survey of General Purpose Computation on Graphics Hardware," *Computer Graphics Forum,* vol. pp. 80-113, 2007.
13. S. L. N. K. Govindaraju, J. Gray, and D. Manocha, "A memory model for scientific algorithms on graphics processors," *SC'06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing,* 2006.
14. CUDA. (2007, http://developer.nvidia.com/object/cuda.html). *Home page maintained by nvidia*.
15. S. Y. a. W. Luk, "Bitwise optimised CAM for network intrusion detection systems," *In Proceedings of International Conference on Field Programmable Logic and Applications,* pp. pages 444–449, 2005.

16. R. H. K. F. Yu, and T. V. Lakshman, "Gigabit Rate Packet Pattern-Matching Using TCAM," *In Proceedings of the 12th IEEE International Conference on Net-work Protocols (ICNP '04),* pp. pages 174–183, October 2004.
17. S. A. Giorgos Vasiliadis, Michalis Polychronakis, Evangelos P. Markatos and Sotiris Ioannidis, "Gnort: High Performance Network Intrusion Detection Using Graphics Processors," *Recent Advances in Intrusion Detection 11th International Symposium , RAID 2008,* 2008.
18. G. C. Alessandro Margara, "High Performance Content-Based Matching Using GPUs," *Copyright 2011 ACM,* July 11–15, 2011.
19. M. P. Giorgos Vasiliadis, Sotiris Ioannidis, "MIDeA: A Multi-Parallel Intrusion Detection Architecture," *Chicago, Illinois, USA. Copyright 2011 ACM,* 2011.
20. Nvidia. (2012, http://www.geforce.com/Hardware/GPUs/geforce-gts-450). *GeForce GTS 450*.

A Study on Packet Content Filtering with GPGPU-based Parallel Architecture

Che-Lun Hung
Department of Computer Science and Communication Engineering Providence University
Taichung, Taiwan
clhung@pu.edu.tw

Chin-Yuan Chang
Department of Computer Science and Information Management Providence University
Taichung, Taiwan
a3014006@gmail.com

Hsiao-Hsi Wang
Department of Computer Science and Information Management Providence University
Taichung, Taiwan
hhwang@pu.edu.tw

Abstract

The rapid development of Internet technology provides a wide variety of applications and services, and these services need to rely on the Internet environment. Packet content filtering is an important role of network security that typically relies on a flexible packet filtering system to extrapolate important packet information from each processed packet. However, such content-based filtering is computation-intensive. In order to enhance the performance of searching pattern of multi-dimensional network packet contents, we propose a parallel packet content-based filter method based on GPGPU platform. The experimental results show that GPGPU significantly enhance the overall performance of packet classification.

Keywords:Packet content matching, GPGPU, CUDA, Parallel process, Network Intrusion Detection