應用於即時串流之GPU硬體編碼器分享技術

A GPU Hardware Video Encoder Sharing Method for Live Streaming Applications

羅啓文　　康浩平
Chi-Wen Lo, Hao-Ping Kang

中文摘要

隨著網路與視訊壓縮技術的發展，使用者可以透過各式錄影裝置連上網路，以直播的方式來分享自己的生活。使用者拍攝的視訊，經網路上傳到網路主機分享給許多觀賞者，而在觀看的同時，也可以藉由在視訊內容中加入文字、加入商標/浮水印和解析度轉換等功能，達到視訊內容的加值。但是這樣視訊編碼需要大量的運算資源才能即時處理多路直播串流，這會造成其運算成本大幅增加。因此，顯示卡廠商NVIDIA在顯示卡中放置專門為視訊編碼所使用的硬體編碼器，它可以提供比CPU更強大的編碼能力。但是，低價NVIDIA顯示卡限制只能同時編碼二路視訊，無法有效利用其編碼能力。本篇論文提出一顯示卡硬體編碼器分享方法與一串流排程方法，可有效地利用硬體資源達到多路即時視訊串流編碼。實驗結果顯示我們提出的方法可達到14路HD(High Definition)視訊即時編碼。

Abstract

Nowadays, based on the fast development of network and video codec technology, we can easily share our life through live video streaming. The live video streams generated from cameras are uploaded to remote video servers and shared to many on-line audiences. To increase the value of video content, the video servers can add texts, trademarks, watermarks, enhance video quality, and resize video resolutions. The video content after content processing needs re-encoding and the computation requirement is large. Therefore the cost that the video servers can encode multiple video streams in real-time increases significantly. Hence, NVIDIA, a graphics processing unit (GPU) manufacturer, has designed a dedicated hardware in its GPUs for video encoding. This encoding hardware can provide high video encoding performance compared to the conventional CPU encoding. Nonetheless, the number of concurrent video streams encoded by a low-end GPU is limited to 2, which cannot fully utilize its encoding power. In this paper, we propose a hardware sharing method and a scheduling method to efficiently utilize the encoding resource for encoding multiple real-time streams. The experimental results show that the proposed methods can achieve concurrent 14 HD (High Definition) video encoding in real-time.

關鍵詞(Key Words)

即時視訊編碼 (Real-Time Video Coding)

圖像處理單元 (GPU)

硬體分享 (Hardware Sharing)

1 · 前言

隨著網路與視訊壓縮技術的發展，影音直播串流服務已經可以提供穩定的影音品質。在網路與壓縮技術不成熟的年代，只有電視台或是專業影音製作公司才可能提供串流服務，而現在一般非專業的使用者已經可以用利用各式的錄影裝置，如：IP cam、Web cam、甚至是手機/平板，就可以透過直播分享自己的生活[1][2]。

在直播視訊串流中，可更動視訊畫面來達成視訊內容加值，如：加入文字以說明畫面內容、加入商標/浮水印來達到版權保護、解析度轉換來達到動態串流(adaptive streaming)、視訊強化來達到優化視訊畫質等功能，改變後的畫面需要重新壓縮之後，才能在網路上傳輸給觀賞者。

為了將直播影片分享給多人觀看，攝影機所拍攝的影片以串流的方式上傳至網路上的主機，觀賞者連線到該主機接收直播串流。而進行視訊加值的功能時，該網路主機則需要大量的運算資源，才能即時處理多路直播串流。一般可採用運算能力強大多核心的中央處理器(Central Processing Unit, CPU)或是採用多台中低階主機平行處理，然而視訊處理，如：解析度轉換、視訊強化等功能，也會耗費大量CPU運算資源，所以CPU運算成本大幅增加。

為了解決運算資源的大量需求問題，顯示卡廠商推出GPGPU (General-Purpose computing on Graphics Processing Units，通用圖形處理器)，將顯示卡中GPU強大的運算能力開放給開發人員進行複雜運算，因為GPU的設計是要進行繪圖功能，如擬真遊戲畫面、3D影像處理，其運算能力通常比CPU更強。GPU大廠NVIDIA的提出了CUDA (Compute Unified Device Architecture，統一計算架構)來讓研究者、開發者存取GPGPU資源，許多文獻中提出利用CUDA來加速視訊編碼速度[3]-[6]。

視訊直播的趨勢，也開始延燒到遊戲上，即時遊戲畫面直播是近來新興的產業，如：Youtube[7], Twitch[8]等，開始提供遊戲直播串流，由於繪製遊戲畫面與視訊編碼同時競爭CUDA運算資源，於是NVIDIA直接在GPU中放置專門為視訊編碼用的硬體電路，稱為NVENC[9]。NVIDIA從第九代GPU架構Kepler開始置入NVENC技術，到第十代Maxwell架構GPU皆支援NVENC，由於Maxwell架構大幅改進了Kepler架構的效能，NVENC編碼效能上在Maxwell架構也優於Kepler架構。實測結果發現在Maxwell架構使用NVENC編碼解析度1080p視訊速度可達250 fps (frame per second，使用GeForce 970 顯示卡)，而Kepler架構只能達到90 fps (使用Tesla K20顯示卡)。

NVIDIA顯示卡主要可分為四個系列：1) GeForce系列，用途為一般遊戲娛樂；2) Quadro系列，用途為專業繪圖；3) Tesla系列：用途為運算量龐大的科學運算; 4) Grid系列：用途為雲端虛擬化主機用顯示卡。其中GeForce系列是屬於消費性電子產品，價格低廉，新台幣一萬元內即可購得性能不錯的顯示卡，產品週期快，短時間內就有新產品上市，所以已經採用效能較佳的Maxwell架構。其他三個系列屬於專業人士用途，因此提供了大量的GPU核心來達到大量資料運算需求，但是價格高昂，產品週期長，現有產品仍是上一代的Kepler架構。從此可以看出NVENC的效能在GeForce系列顯示卡上會有較佳的表現。

NVIDIA為了讓高價顯示卡與低價GeForce顯卡性能上有所區別，所以限制GeForce顯示卡只能同時編碼二路視訊，而其他系列顯示卡則無此限制。在一般直播串流應用中，編碼速率只需30fps就可達到即時串流的需求，所以限制只進行二路及時串流就沒有利用到GPU強大的視訊編碼能力。所以在本篇論文提出一GPU硬體視訊編碼器分享分法，能夠充分有效率利用硬體編碼器資源，達到多路即時視訊串流編碼，讓GeForce顯示卡成為一高效率低成本的硬體視訊編碼解決方案。

在NVIDIA高價顯卡中，GPU含一個到數個的硬體編碼器NVENC，且同時視訊編碼的路數並無限制，採用了[10]的硬體式分享方法。在[10]中，假設有M個串流要同時編碼，則硬體編碼器中需要為M個串流配置各自的記憶體空間，來存放M個串流各自所需要的中介資料，如：參考畫面(reference frame)等。硬體編碼器採用

分時多工方式，同一時間只有一路串流輸入一張畫面進行編碼，編碼的中介資料則存到緩衝器中，編碼完一個畫面後再切換到另外一路串流進行編碼。然而在本文的串流系統中，我們利用的是市面上現成的顯示卡，無法存取到GPU內部的視訊編碼相依性資料，所以無法進行以畫面為單位的切換。我們提出採用以一個畫面群(GOP)為切換單位，避免畫面切換的相依性問題，並針對多路串流要同時編碼的環境下，提出一排程機制，來對GPU運算資源做最佳化的分配。

在本文中，GPU和顯示卡這兩個名詞會交互使用，但其意義是相同。

後續章節安排如下：在第二章中，我們提出一硬體編碼器多工技術，來達成多路視訊編碼共用一編碼器。在第三章，我們提出一多串流排程方法，來有效的利用硬體編碼器的資源。在第四章，藉由系統實作來驗證效能。最後，第五章提供結論。

2．硬體視訊編碼器多工技術

在即時視訊串流中，每一秒視訊包含R_f張畫面，R_f為畫面更新率(frame rate)，常見的設定為R_f=30，代表每$1/R_f$秒會產生一個畫面。

目前視訊編碼技術(例如H.264和HEVC)會利用畫面和畫面之間相互參考來去除時間上的冗餘資訊(Temporal Redundancy)，以降低資料量。雖然這種技術可以提高壓縮率，但是也會使得畫面之間產生相依性。

硬體編碼器可利用分時多工技術，讓一個編碼器同時編碼多個串流，如圖1所示。若以一個畫面當作切換單位，硬體編碼器需要依照畫面相依性存取不同串流的緩衝器空間[10]，否則會產生錯誤的編碼結果。然而這需要顯示卡內部硬體與韌體配合，一般軟體服務業者難以變動顯示卡的運作方式。因此，在本系統中以一個GOP為分時多工切換單位，因GOP間不會有參考相依性，所以無需更動顯卡內部功能，只要以軟體方式就可以達到分時多工的功能。

圖2為GOP為切換單位的多工編碼示意圖，系統中有二個串流A_1和A_2，每個串流都有各自的畫面緩衝器負責儲存畫面。經過前一級的影像處理後，串流A_1和A_2會先將畫面儲存到畫面緩衝器裡，等到緩衝器裡已經累計到一個GOP的畫面後，便會將GOP移至佇列。硬體編碼器會從佇列開頭的GOP以先來先服務(First Come, First Served)的方式來進行編碼。

上述以GOP為編碼單位的多工技術雖然比較簡單且容易實作，但是在即時視訊串流中，因為它必須等待串流累計到一個完整的GOP才開始編碼，這會造成硬體編碼器閒置時間和編碼延遲增加，如圖3所示。編碼延遲定義為前一級產生GOP最後一個畫面到編碼完成的時間。gop^{A_i}為串流A_i的GOP大小，t_e為編碼一個畫面的時間。

圖4顯示為尚未累計到一個GOP就提前開始編碼的範例。N_b^{Ai}為串流緩衝器中的畫面數目。畫面累積到6張就進行編碼，利用等待剩下2張畫面時，編碼器可完成數個畫面的編碼，在等待剩下的畫面接收到並編碼後，才切換至其他串流。與圖3的範例比較，編碼器閒置時間和編碼延遲皆大幅降低，顯示提前進行編碼更有效的利用硬體編碼器的運算資源。

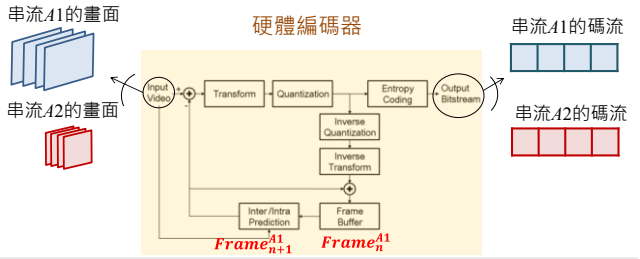

圖 1　硬體編碼器分時多工示意圖

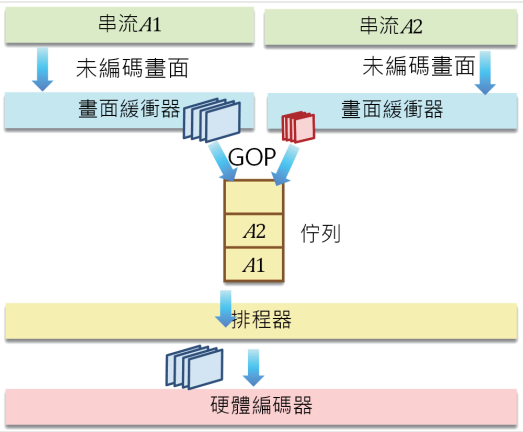

圖 2　GOP為切換單位的多工示意圖

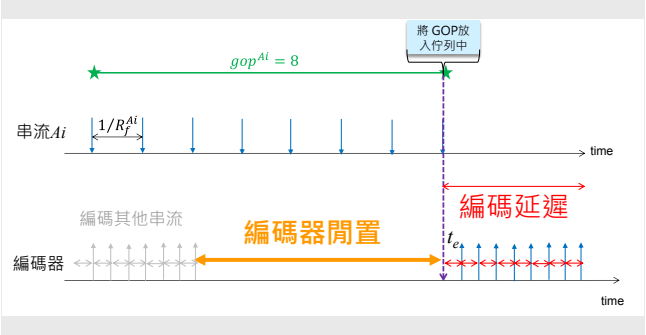

圖 3　累積一個畫面群後才進行編碼

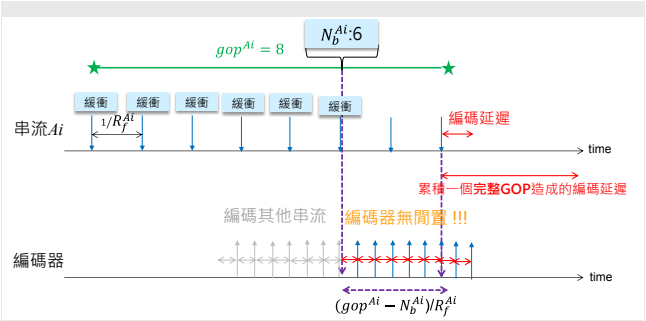

圖 4　尚未累積到一個GOP就進行編碼

3．即時視訊串流編碼排程方法

根據第2章分析顯示，在串流畫面未累積至一個GOP即進行編碼，可增加硬體編碼器多工效率。當系統中有多個串流時需要編碼時，每個串流所需的編碼時間不同、等待累積至完整GOP的時間也不同，需要一個多串流排程方法，來選擇對硬體編碼器利用率最佳的串流進行編碼。

多工排程方法流程如圖5所示，當硬體編碼器閒置時，查看佇列中是否有累積一個GOP的串流畫面，若有則取出佇列中第一個GOP進行編碼，以降低佇列中GOP的編碼延遲；若無，則計算各個串流的接收比例，選擇接收比例最高的串流進行編碼，以增加硬體編碼器的效率。接收比例為編碼器在編碼各個串流緩衝時的資源利用率，此比例與畫面解析度、畫面更新率、畫面複雜度、累計畫面數量有關。假設目前系統中有M個串流$A_1, A_2, \ldots, A_M$，A_i代表第i個串流，則各串流的接收比例R_{recv}與視訊串流GOP大小gop^{Ai}、畫面解析度與編碼時間的比例R_s^{Ai}、畫面更新率R_f^{Ai}、累計畫面數量N_b^{Ai}有關，其公式如(1)。

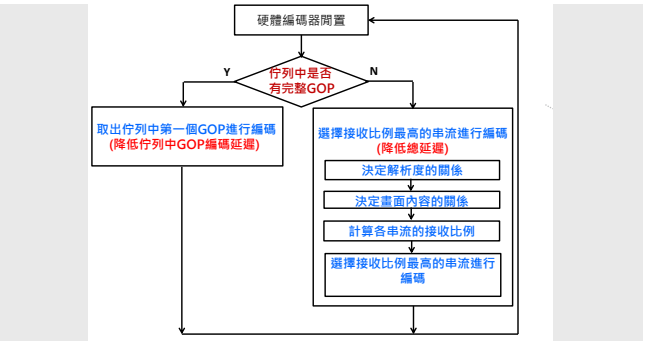

圖 5　多串流排程方法流程圖

$$R_{recv} = \frac{gop^{Ai} \cdot R_s^{Ai}}{(gop^{Ai} - N_b^{Ai})/R_f^{Ai}} \quad (1)$$

其中分母項代表串流提前編碼的時間長度，如圖4所示。分子項代表編碼此串流所需要的運算量。R_{recv}越高代表在此時間內硬體編碼器有較多時間在進行編碼，較少時間在閒置。

一串流Ai的解析度$W^{Ai} \times H^{Ai}$與編碼時間的關係定義為R_s^{Ai}。R_s^{Ai}可以由多種方式來決定，列舉如下：

1) 面積比例法：

以1080p的解析度面積(1920×1080)為參考基準，則$R_s^{Ai} = (W^{Ai} \times H^{Ai})/(1920 \times 1080)$

2) 窮舉法：

硬體編碼器事先編碼各種解析度的影片並量測編碼時間，R_s^{Ai}則為$(W^{Ai} \times H^{Ai})$編碼時間與1080p編碼時間的比例。

3) 分類法：

硬體編碼器事先量測數個解析度的編碼時間(例如1080P、720P和480P)，將A_i分類為與自身解析度最接近之解析度，$R_s^{A_i}$則為此被分類的解析度的編碼時間與1080P的編碼時間的比例。

4) 內差法：

硬體編碼器事先量測數個解析度的編碼時間(例如1080P、720P和480P)，並以面積$(W^{A_i} \times H^{A_i})$內插方式推算一編碼時間，$R_s^{A_i}$則為推得的時間與1080P的編碼時間的比例。

4．實驗分析

在這個章節中，我們實作出硬體編碼器多

工技術與多串流排程方法，並整合至FFmpeg[11]中。在實驗中，串流資料經由FFmpeg利用CPU解碼之後，再經由硬體編碼器編碼，在這裡比較了數個不同的方法：1) NVIDIA所提供的NVENC，標註為"NVENC"，2) 累積一個GOP才進行多工切換方法，標註為"GOP-based"，3)本論文提出的提前編碼與排程機制，標註為"Proposed"，4)使用Intel i5-4690K@3.50GH CPU執行FFmpeg進行編碼，標註為"CPU"，5)使用高價顯卡Tesla進行編碼，Tesla無同時編碼路數限制，但架構較舊，編碼效能較差，標註為Telsa。我們以編碼速度與平均編碼延遲時間當作效能指標。當編碼速度小於輸出串流的畫面更新率，如30 fps，則代表此系統無法提供即時串流。我們將比較NVIDIA不同顯示卡的硬體效能，分別是NVIDIA Tesla K20、NVIDIA GeForce GTX750和GTX970，這三個GPU的硬體架構不一樣，Tesla K20是較舊的Kepler，而GTX750和GTX970分別為目前最新的第一代Maxwell和第二代Maxwell。接收比例的算法，則採用公式(1)，解析度的計算則採用窮舉法。

首先，我們比較不同硬體編碼器在編碼多路1080p@30fps, $gop^{Ai} = 15$串流的編碼效能。圖6是使用GTX 970的編碼速度，當編碼速度小於30fps時，將省略資料。在接收比例計算時，因各個串流的解析度與畫面更新率相同，則串流的選擇將依據尚未抵達的畫面數目$(gop^{Ai} - N_b^{Ai})$，如果此值越小，代表編碼此串流的等待時間越少，硬體編碼器較少閒置。本文提出的方法可以突破NVENC只能同時編碼兩路的限制，將可即時編碼的路數擴展到14路。而且Proposed採用提前編碼機制與多串流排程方法，可以更有效率的使用硬體編碼器資源，所以比GOP-based多工方法多了3路即時編碼。使用Telsa顯示卡只可達2路，顯示昂貴顯卡不一定有較佳效能。而CPU能達到1路串流編碼，若要達到多路及時編碼勢必大幅增加成本。

圖7比較了兩個多工方法的平均延遲時間，直接使用NVENC的方式雖然可以達到最低的延遲時間(約2.1ms)，但是卻只能處理2路。使用多工方法因為多了多路串流間切換的時間，造成延遲增加。GOP-based方法在累積完一個GOP才以先到先服務的方式編碼，所以有較高的編碼延遲。當輸入串流超過12路時延遲大於1秒，代表一個GOP編碼完又多累積了2個GOP等待編碼，那麼輸入大於輸出的情況下，就無法達到即時編碼。Proposed方法使用了提前編碼，並考慮提前編碼時間內何路串流可以有最佳的編碼效率，可大幅降低平均延遲時間，所以在輸入少於13路串流時，延遲皆低於50ms。

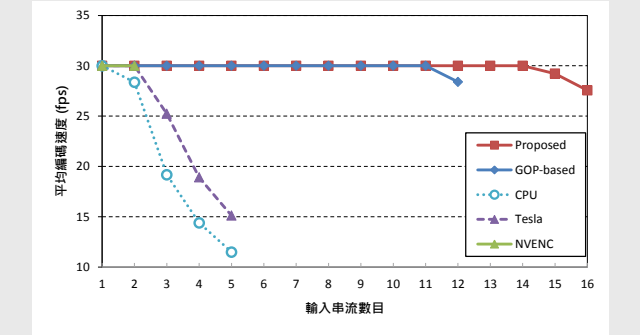

圖 6　GTX 970編碼速度實驗

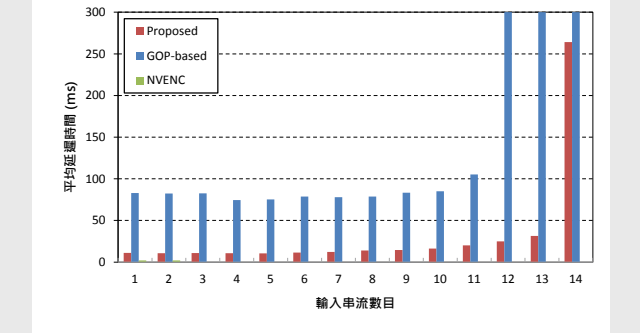

圖 7　GTX 970平均延遲實驗

圖8與圖9顯示了使用GTX 750的編碼速度與延遲時間。GTX 750的架構較GTX 970效能較低，但仍可以到達7路的即時編碼，且Proposed方法的編碼延遲可小於40ms，可以運用在低延遲的應用。

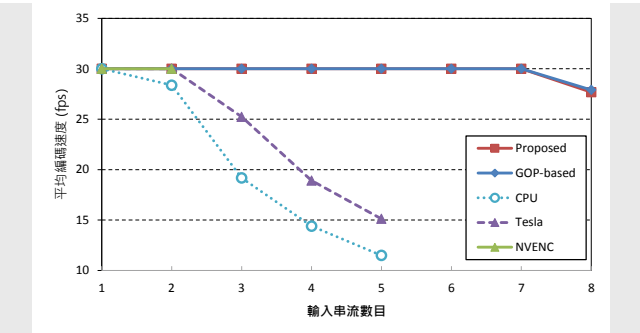

圖 8　GTX 750編碼速度實驗

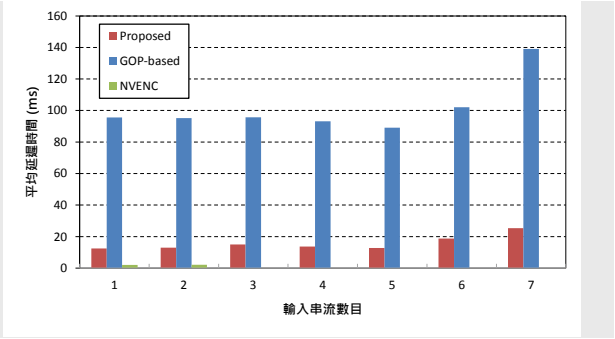

圖 9 GTX 750平均延遲實驗

接著輸入4種不同視訊規格不同的串流，分別為1080P@30fps、1080P@60fps 720P@30fps和720P@60fps，我們將此4個串流稱為1組串流。表1為GTX970編碼多組串流結果。使用多工技術可最多編碼2組共8路即時串流，Proposed所計算的接收比例內考慮了各個串流的畫面更新率與解析度(在此使用窮舉法)，所以可以有效的安排各串流編碼的順序，進而降低編碼延遲。

硬體編碼器在GOP-based方法與Proposed方法都是編碼一個串流的一個完整GOP畫面後，才會切換至另一個串流，所以並不會犧牲運動估測的編碼效率，所以兩個方法會有相同的視訊畫質。然而，GOP-based方法中，需等待全部串流中至少有一串流累積到一個GOP畫面，硬體編碼器才會選擇該串流進行編碼；在Proposed方法中，當硬體編碼器閒置時，排程器會計算各個串流的接收比例，選擇高接收比例串流中已存在佇列GOP畫面編碼，並等待整個GOP接收完整並編碼後，再轉為閒置狀態，以減少硬體編碼器閒置時間，並增加編碼效率。

表 1 GTX970編碼不同規格串流的平均延遲

輸入組數	Proposed	GOP-based
1	8.31 ms	64.01 ms
2	17. 87 ms	83.67 ms

表2顯示了實驗中使用到硬體設備的價格。Tesla的價格最高，但也擁有優越的規格，足以應付資料量龐大的科學複雜運算，其內建的NVENC無編碼路數限制，但顯示卡架構較舊，無法達成多路HD視訊編碼，所以不適合作為視訊編碼伺服器。相反地，GeForce價格低廉，且採用較新的顯示卡架構，NVENC擁有不錯的效能，搭配本論文提出的多工技術，可達成多路(7~14路)HD即時編碼，達成一低成本高效能的直播串流編碼伺服器解決方案。而CPU要達到相同效能需要多顆多核心CPU才能達成，其成本會大幅增加。

表 2 顯示卡和CPU參考價格(2015年10月)[12]

型號	價格 (美金)	記憶體	GPU核心數
GeForce GTX 750	125	1GB	512
GeForce GTX 970	325	4GB	1664
Telsa K20	1,875	5GB	2496
i5-4690	225	-	-

5 · 結論

在此論文中，我們針對低成本的顯示卡提出一個硬體分享方法與多串流排程方法，以突破低成本顯示卡只能同時進行2路視訊編碼。在硬體分享方法中，一個串流的畫面尚未累積到一個 GOP就提前編碼，以降低硬體編碼器閒置時間與編碼延遲。多串流排程方法會利用每個串流的畫面更新率、解析度、緩衝器內畫面數目計算接收比例，此比例越高代表該串流有較佳的資源利用率，依照接收比例來選擇要編碼的串流，提高硬體編碼器利用率。實驗顯示在不同顯示卡的環境下，可進行7~14路的HD即時視訊編碼。

低成本顯示卡搭配本文提出的技術可成為一個低成本高效率的直播串流編碼伺服器解決方案。讓直播串流伺服器可以對多路直播串流進行視訊加值的處理，而不影響串流的即時性。

在未來工作方面，將考慮一個系統中有多個GPU，要如何對多個GPU做分配與串流排程是個重要的課題；此外，GOP大小也是影響效能因素之一，GOP越小，硬體編碼器可以更快速的切換，但是編碼壓縮率也較差，研究如何兼顧多工效率與編碼壓縮率的多工架構，也是未來的研究目標。

參考文獻

[1] Livehouse.in. [Online]. Available:// livehouse.in/

[2] Facebook Mentions [Online]. Available:// www.facebook.com/about/mentions/

[3] W.-N. Chen, and H.-M. Hang, “H.264/AVC motion estimation implmentation on Compute Unified Device Architecture (CUDA),”pp. 697-700, Proc. IEEE International Conference on Mutimedia and Expo 2008, pp. 697-700, 2008

[4] E. Monteiro, B. Vizzotto, C. Diniz, and B. Zatt, “Applying CUDA Architecture to Accelerate Full Search Block Matching Algorithm for High Performance Motion Estimation in Video Encoding,”Proc. 23rd International Symposium on Computer Architecture and High Performance Computing 2011, pp. 128-135, 2011.

[5] R. Rodriguez, J.L. Martinez, GFernandez-Escribano, J.M. Claver, “Accelerating H.264 inter prediction in a GPU by using CUDA,”Proc. IEEE 2010 Digest of Technical Papers International Conference on Consumer Electronics, pp. 463-464, 2010.

[6] S. Momcilovic, A. Ilic , N. Roma, L. Sousa, “Dynamic Load Balancing for Real-Time Video Encoding on Heterogeneous CPU+GPU Systems,” IEEE Transactions on Multimedia, Vol 16, No. 1, pp. 108-121, 2014.

[7] Youtube gaming [Online]. Available:// youtube-global.blogspot.tw/search/label/YouTube%20Gaming/

[8] Twitch [Online]. Available://www.twitch.tv/

[9] NVIDIANVENC [Online]. Available:// developer.nvidia.com/nvidia-video-codec-sdk/

[10]O.-T. MAN, and H.-G. Mahn, “Multi-channel image encoding method and system High-speed digital-to-RF converter,” U.S. Patent 7050496 , May 23, 2006.

[11]FFmpeg [Online]. Available://ffmpeg.org/

[12]Amazon [Online]. Available:// www.amazon.com/

作者簡介

羅啓文

現任職於工研院資通所網路服務技術部影音匯流技術部工程師，專長為視訊串流技術、P2P網路技術等。

康浩平

現任職於工研院資通所網路服務技術部影音匯流系統技術部工程師，專長為高效能計算、CUDA平行處理等。