

Design and Implementation of a Hadoop-based Video Transcoding System

許子衡、吳靖緯
南台科技大學資訊工程學系

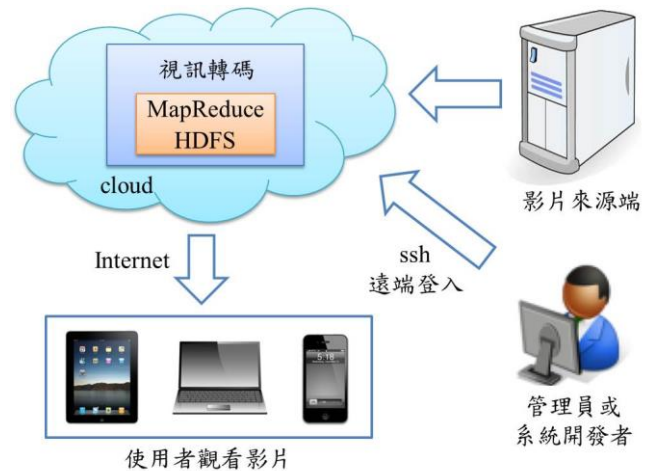
摘要—即時視訊串流系統(Live Streaming)為了滿足即時突發性的視訊傳輸需求,需要大量的頻寬與運算資源。為了降低視訊轉碼延遲(transcoding delay),本文實現一個雲端即時視訊串流轉碼系統,本系統採用 Hadoop 分散式檔案系統(Hadoop Distributed File System)進行視訊儲存,並利用 MapReduce 框架與 FFmpeg 進行分散式轉碼。本論文實現基於 Hadoop 平台的雲端即時視訊串流轉碼系統,並詳細敘述 Mapper 和 Reducer 的運作方式。本論文所實現的雲端即時視訊串流轉碼系統與傳統視訊轉碼系統相比,可減少視訊轉碼時間,降低視訊傳輸起始延遲(Start-up Delay)。

一、簡介

近年來多媒體影音用戶快速增加,如何以最低的成本來滿足所有用戶的視訊需求是現今網路內容服務業者關切的話題。由於網路頻寬的限制以及多樣化的多媒體設備與影音格式,網路內容服務業者將視訊傳遞到用戶必須透過視訊轉碼,將視訊串流轉換成相對應的視訊格式。視訊轉碼的用途主要是將視訊串流進行壓縮以減少資料量的傳遞。視訊是由一連串靜止畫面所組成,在進行視訊壓縮時,通常畫面與畫面之間有極大的相似性,因此會產生時間上累贅(Temporal Redundancy),此時壓縮只參考兩張畫面的關聯性,針對畫面不同的地方做壓縮或直接使用前一張畫面,即可減少資料量達到壓縮的效果。

分散式視訊轉碼是將一視訊影音切割成許多區塊(chunk),再把這些區塊(chunk)分配給多台伺服器主機進行視訊轉碼工作,轉碼完成後再將各個區塊(chunk)合併成一個視訊檔案。此方法可提高視訊編碼的效率,降低用戶觀看視訊的等待時間。然而傳統分散式視訊轉碼系統,軟硬體設備的更新維護及擴充,對於網路內容服務業者成本而言是相當大的負擔。網路內容服務業者為了架設分散式轉碼伺服器,必須聘請額外工作人員進行管理,人力成本也極高。為了克服軟硬體擴充的問題,並改善編碼時間。本論文將分散式轉碼運算應用在雲端服務系統上。雲端服務系統可提供可靠且更安全的網路資料儲存,在軟硬體設備更新與擴充皆由雲端服務業者處理,只需租用雲端即可使用軟硬體設備。在成本方面,相較於自行架設伺服器,租用雲端系統成本較低廉。本系統使用 Hadoop 分散式檔案系統(HDFS)儲存視訊區塊(Chunk),並利用 MapReduce 框架與 FFmpeg 視訊編碼軟體進行分散式轉碼,實現出雲端即時視訊串流轉碼系統,以提高視訊觀看的即時性。系統示意圖如圖一。

本研究由國科會贊助,計畫編號 102-2221-E-218-016-。

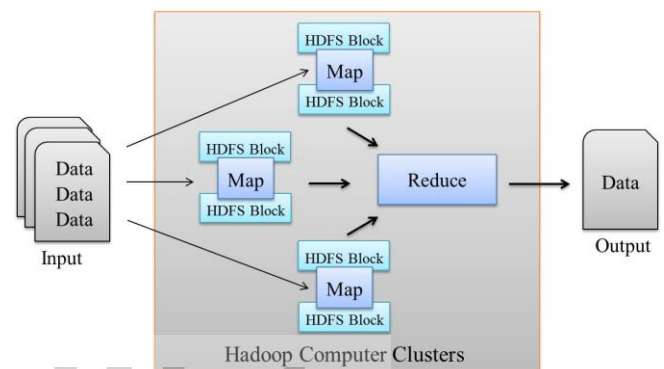


圖一：分散式轉碼系統示意圖

二、相關研究

2.1 雲端運算技術介紹

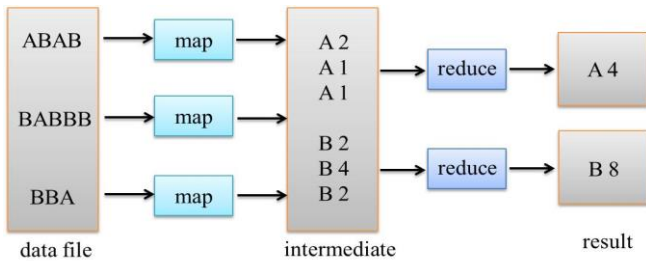
雲端運算架構主要可分為以下三層[1]:基礎設施服務(Infrastructure as a Service, IaaS),像是 Amazon 隸屬的 Amazon EC2 與 S3;平台服務(Platform as a Service, PaaS),如 Amazon Web Service 和 Google APP Engine;軟體服務(Software as a Service, SaaS),例如 Microsoft 的網路更新服務與 Google Maps 等。本系統主要採用 Hadoop 分散式檔案系統(Hadoop Distributed File System, HDFS)進行視訊儲存,並利用 MapReduce 架構與 FFmpeg 進行分散式轉碼。HDFS 與 MapReduce 架構皆屬於雲端運算中的一環,Hadoop 雲端運算架構如下圖二所示。



圖二：Hadoop 雲端運算架構

本論文所採用的 Hadoop 雲端運算架構主要分為 HDFS 與 MapReduce 兩部分。Hadoop 是由 Apache 軟體基金會根據 Google 公司發表的 Google 檔案系統和

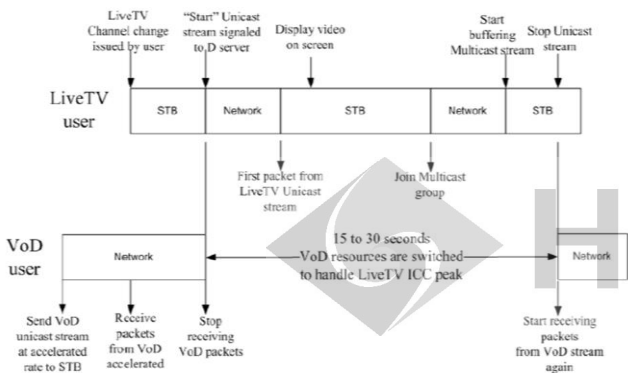
MapReduce的論文自行實作而成。HDFS 採主從式 (Master/Slave)架構[2]，由一個名稱節點(name node)與多個資料節點(data node)這兩種角色組成。以一般佈屬情況下，Master 上只運行一個名稱節點(name node)，負責處理檔案系統的管理及儲存，並記錄檔案的各個資料區塊(HDFS 的基本儲存單位)放置於那些資料節點(data node)上。而在每一個 Slave 上皆有一個資料節點(data node)，負責處理使用者存取資料區塊上的請求，並定時回覆資料區塊的狀態給名稱節點(name node)。當有任何一個資料節點(data node)損壞時，名稱節點(name node)也會自動進行資料的搬遷和複製，使工作能繼續正常運作。Hadoop MapReduce 主要架構是由 Map 與 Reduce 兩個函數所組成。Map 函數的輸入為一組 key/value 序對，輸出則為多組 intermediate key/value 序對組。Reduce 函數則將相同 intermediate key 的 value 做合併動作，最後產生輸出結果 key/value 序對，MapReduce 流程如圖三所示。



圖三：MapReduce 流程示意圖

2.2 雲端運算與視訊串流相關文獻

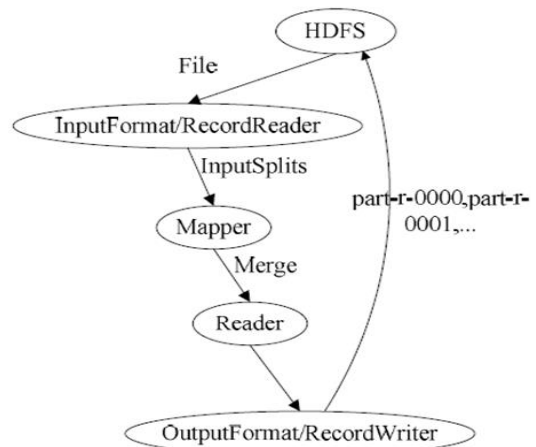
在論文[3]中 Vaneet Aggarwa 等人提到 IPTV 服務需要大量的頻寬與運算資源來處理突發性或即時性的資源需求。因此作者提出一個資源配置的框架，利用虛擬化來使這些服務並存在一個共同的基礎建設上。作者提出最佳化演算法，規定最低限度的伺服器數量來滿足所有服務的請求。作者探討如何配置即時的雲端基礎設施來支援大規模的頻寬與 IPTV 應用程序，如 LiveTV 瞬間通道變化(ICC)和 VoD 請求。論文[3]使用動態分配虛擬機器來處理 LiveTV 與 VoD 之間相關的資源分配，如圖四。由於 LiveTV 比 VoD 需要更快且更加即時的視訊畫面，因此當 LiveTV 用戶需求量增加時，作者把一部分的 VoD 虛擬機資源暫停使用，並將停用的資源即時分配給 LiveTV 以滿足用戶需求，達到流暢的現場直播效果。



圖四：LiveTV ICC 與 VoD 封包緩衝時間安排

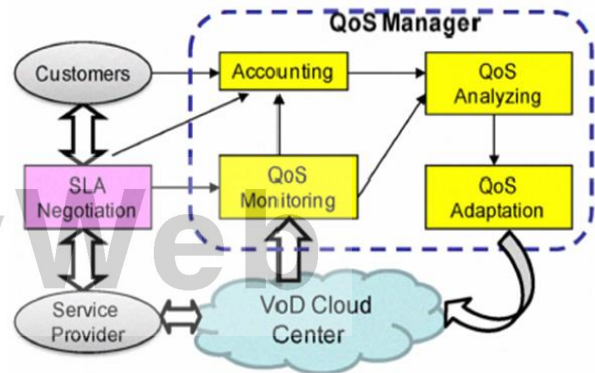
在論文[4]中，SUN Bing-jue 等人根據雲端計算的架

構，設計出基於 Hadoop 系統的跨平台快取伺服器，並將此設計應用於 P2P/VoD。圖五為論文中 Hadoop 資料處理流程圖。在論文[4]的系統架構主要可分為硬體及軟體。硬體方面包括：Web 系統模組、檔案伺服器系統模組、DNS 伺服器、儲存模組和快取伺服器群集。軟體方面伺服器的環境為 Windows 2003 伺服器作業系統與 FreeBSD 作業系統。系統視訊服務執行過程如下：(1)用戶首先訪問視訊網站來查詢所需的視訊並發出視訊請求；(2)Web 伺服器獲得視訊請求後在資料庫裡查詢視訊位址，接著送出視訊相關資料的請求到檔案伺服器；(3)檔案伺服器對用戶發送視訊可透過幾種方式：直接傳輸、透過快取伺服器傳輸和用戶使用 P2P 網路視訊服務傳輸；(4)最後實現平行視訊串流傳送到客戶端，完成高清晰的視訊撥放。而通常情況下 P2P 網路無法滿足所有的用戶需求，所以透過快取伺服器傳遞視訊串流為主要的方法，Hadoop 此時便扮演著重要的角色。



圖五：Hadoop 資料處理流程圖

Xiaoying Wang[5] 等人對於隨選視訊 (Video on Demand)雲端服務中心提出自適應 QoS 管理框架。Qos 管理框架是基於自主性運算的概念。透過 QoS 感知快取替換演算法(QoS-aware Cache Replacement algorithm)可以讓服務供應商在有限的資源裡獲取最大的利潤，自適應 QoS 管理框架如圖六。論文[5]詳細描述了自適應 QoS 管理框架及其演算法，最後評價結果證明所提出的方法，可以大大提高服務供應商的總收益。

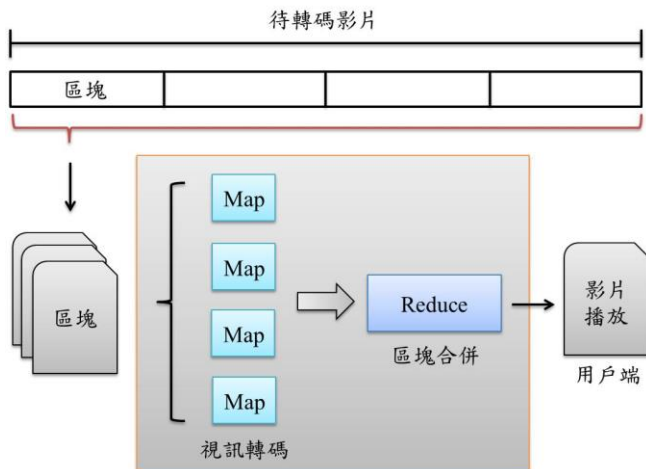


圖六：結合雲端的 QoS 管理框架

三、基於 Hadoop 平台的雲端即時視訊串流轉碼系統

本系統使用分散式的方法進行視訊轉碼，影片切割方面是有幾個工作節點就把一部影片切割成幾個區塊(chunks)，而每一個區塊(chunk)裡都有固定數量的幀(frames)。最後將這些區塊(chunks)平均分配給各個工作節點去做視訊轉碼的工作，等到整部影片轉碼完成後再將影片傳遞給用戶觀看。舉個簡單的例子，假設有四個工作節點，此時將一部完整且未轉碼的影片先切割成四個區塊(chunks)，接著將這四個區塊(chunks)分配給四個工作節點執行視訊轉碼任務。當這四個工作節點轉碼任務完成時，再將已轉碼的四個區塊(chunks)作合併，形成一部已轉碼後的影片，最後輸出至用戶端播放影片。

本論文將分散式轉碼的概念結合 Hadoop 雲端運算架構的優點，實現出基於 Hadoop 平台的雲端即時視訊串流轉碼系統，系統架構圖如圖七。

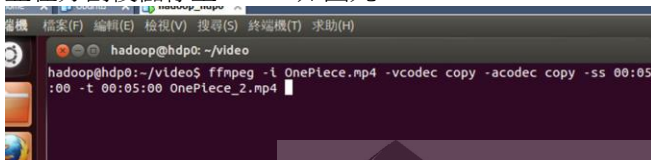


圖七：基於雲端的分散式轉碼系統架構圖

四、系統實現

4.1 實現方法

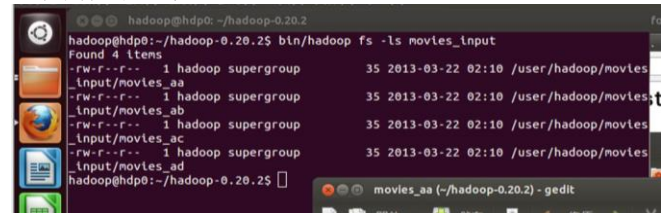
本系統建立在虛擬機(virtual machines)上，虛擬機作業系統為 ubuntu 12.04。而系統將影片所做的切割、轉檔與合併動作採用 FFmpeg。FFmpeg 是一套開放原始碼的影音轉檔軟體，可以執行音訊和視訊多種格式的轉檔功能。視訊儲存部分使用 Hadoop 分散式檔案系統，並搭配 MapReduce 框架來完成雲端即時視訊串流轉碼系統。系統實作首先利用 FFmpeg 將影片作分割如圖八，並在分割後儲存至 HDFS 如圖九。



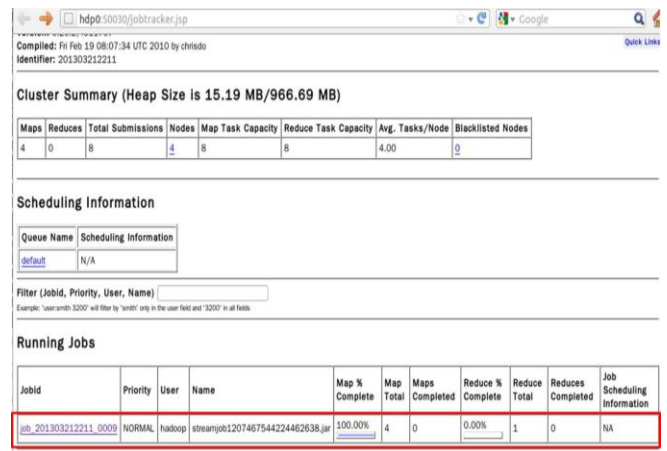
圖八：FFmpeg分割影片

要將儲存在 HDFS 上的影片提交給 MapReduce 作影片轉碼工作(Job)，基本上會經過四個步驟：(1)使用者提交 MapReduce Job；(2)由一個 JobTracker 控制 Job 的進行；(3)TaskTracker 執行 Job 所分割出來的任務(Task)；(4)HDFS 負責儲存 Job 所需的輸入檔案與運算結果。接下來將針對本系統 MapReducer 的流程做詳細介紹。

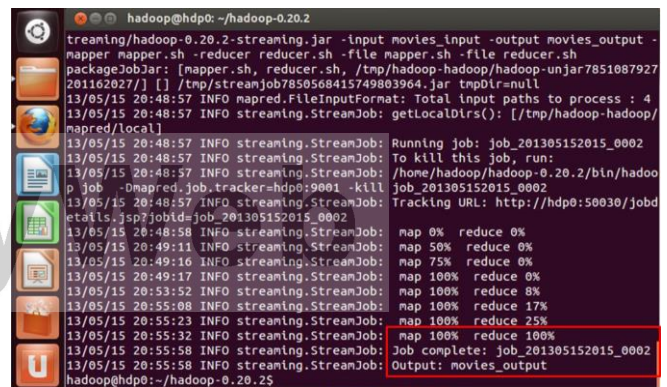
本系統採用 Hadoop Streaming 工具執行 MapReduce 任務，Hadoop Streaming 是 Hadoop 所提供的一個編譯工具，允許用戶使用任何可執行文件或腳本(Script)充當 mapper 或者 reducer。Hadoop 預設情況下會把輸入的數據資料分割為固定大小的 64MB，如檔案大小不足 64MB 則只儲存原檔案大小。Hadoop 對於每一個分割產生一個相對應的 Map Task，Map Task 執行使用者定義的 Map 函式。本系統所定義的 Map 函式主要是紀錄分割後儲存在 HDFS 上的影片路徑，並透過該路徑抓取影片至 TaskTracker 執行轉碼任務。TaskTracker 轉碼後的影片片段會先存放在 TaskTracker 本地硬碟，之後再上傳至 HDFS 上。藉由 Hadoop 網頁管理介面可得知 MapReduce 各階段任務執行結果與資訊，由圖十得知此時 Map 成功轉碼完成的影片。Map Task 完成後接著執行 Reduce Task。本系統定義 Reduce 函式內容主要是從 HDFS 上抓取以轉碼過的影片片段至 TaskTracker，再由 TaskTracker 將各個影片片段合併為一部完整的影片，合併後將這部影片上傳至 HDFS 上。由圖十一可了解 MapReduce 任務是否執行成功。最後由圖十二將轉檔後的影片成功上傳至 HDFS。



圖九：影片儲存至 HDFS



圖十：Hadoop 網頁管理介面



圖十一：MapReduce 任務執行成功

```

hadoop@hdp0: ~/hadoop-0.20.2
hadoop@hdp0:~/hadoop-0.20.2$ bin/hadoop fs -ls movies_put
Found 5 items
-rw-r--r-- 1 hadoop supergroup 82889409 2013-03-29 03:27 /user/hadoop/mov
ies_put/movies_output.flv
-rw-r--r-- 1 met_video supergroup 21551708 2013-03-29 03:27 /user/hadoop/mov
ies_put/output-OnePlece_1.mp4.flv
-rw-r--r-- 1 met_video supergroup 22009098 2013-03-29 03:24 /user/hadoop/mov
ies_put/output-OnePlece_2.mp4.flv
-rw-r--r-- 1 met_video supergroup 18350222 2013-03-29 03:27 /user/hadoop/mov
ies_put/output-OnePlece_3.mp4.flv
-rw-r--r-- 1 met_video supergroup 20979960 2013-03-29 03:27 /user/hadoop/mov
ies_put/output-OnePlece_4.mp4.flv
hadoop@hdp0:~/hadoop-0.20.2$

```

圖十二：轉檔後的影片成功上傳至 HDFS

4.2 效能分析

本文將比較單機轉碼與使用 Hadoop 平台的雲端即時視訊串流轉碼系統之間效能上的差異。本次實驗採用的影片片長為 20 分 42 秒的海賊王動畫，影片大小為 76.9MB，影片轉碼將原始影片格式 mp4 採用 H.264 重新編碼一次。實驗結果如圖十三。

轉碼方法	mapper 時間	reducer 時間	總轉碼時間	虛擬機 memory
單機轉碼	非分散式轉碼		8分27秒	1G
雲端轉碼系統	7分31秒	1分30秒	8分2秒	1G
單機轉碼	非分散式轉碼		7分56秒	2G
雲端轉碼系統	3分30秒	48秒	4分11秒	2G
單機轉碼	非分散式轉碼		7分51秒	3G
雲端轉碼系統	3分27秒	45秒	4分2秒	3G

圖十三：實驗數據

由圖十三可明顯看出使用基於 Hadoop 平台的雲端即時視訊串流轉碼系統與單機轉碼相比，雲端即時視訊串流轉碼系統可大幅節省轉碼所需的時間。由於影像處理需要大量的運算資源(包含 CPU 與記憶體)，因此提高虛擬機的記憶體(memory)可縮短視訊轉碼的時間。透過網頁管理介面，可得知 MapReduce 任務各階段執行所需的時間，如圖十四。如將此系統移植到大型的叢集運算(Cluster)上，相信對於高畫質影音的轉碼時間能有顯著的改善。

Hadoop map task list for [job_201306271450_0002](#) on [hdp0](#)

All Tasks

Task	Complete	Status	Start Time	Finish Time	Errors	Counters
task_201306271450_0002_m_000000	100.00%	Records R/W=1/401	27-六月-2013 15:16:17	27-六月-2013 15:19:32 (3mins, 15sec)	0	0
task_201306271450_0002_m_000001	100.00%	Records R/W=1/405	27-六月-2013 15:16:18	27-六月-2013 15:19:33 (3mins, 15sec)	0	0
task_201306271450_0002_m_000002	100.00%	Records R/W=1/382	27-六月-2013 15:16:18	27-六月-2013 15:19:27 (3mins, 9sec)	0	0
task_201306271450_0002_m_000003	100.00%	Records R/W=1/424	27-六月-2013 15:16:18	27-六月-2013 15:19:46 (3mins, 27sec)	0	0

Hadoop reduce task list for [job_201306271450_0002](#) on [hdp0](#)

All Tasks

Task	Complete	Status	Start Time	Finish Time	Errors	Counters
task_201306271450_0002_r_000000	100.00%	Records R/W=1539/1 > reduce	27-六月-2013 15:19:2	27-六月-2013 15:20:12 (45sec)	0	10

圖十四：MapReduce 任務執行資訊

結論

隨著科技的發展與網路的日益普及，越來越多的用戶喜歡在網路上觀看影片。高解析度、高頻寬和大規模的數位視訊串流傳輸需求快速增加，網路多媒體應用服務，如重大即時視訊新聞，為了提供即時性的(Time-to-Market)視訊串流給不同的行動裝置，對視訊編碼處理速度有顯著的要求。當大量的資料需要編碼時，會佔用大量的伺服器運算資源，並需要相當長的時間進行視訊編碼，導致系統負載增加，用戶觀看影片等待時間變長。傳統視訊服務業者需自行架設伺服器，所需的軟硬體成本與人員管理，成本上是個很大的負擔。因此本論文針對傳統轉碼系統的缺點提出基於 Hadoop 平台的雲端即時視訊串流轉碼系統，減少視訊轉碼時間，降低視訊傳輸起始延遲(Start-up Delay)。

參考文獻

- [1] Jiann-Liang Chenz, Szu-Lin Wuy, Yang-Fang Li, Pei-Jia Yang, Yanuaris Teofilus Larosa, "IMS Cloud Computing Architecture for High-Quality Multimedia Applications," *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1463-1468, July. 2011
- [2] Rafael Pereira, Marcello Azambuja, Karin Breitman, Markus Endler, "An Architecture for Distributed High Performance Video Processing in the Cloud," *IEEE 3rd International Conference on Cloud Computing (CLOUD)*, pp. 482-489, July. 2010
- [3] Vaneet Aggarwal, Vijay Gopalakrishnan, Rittwik Jana, K. K. Ramakrishnan, Vinay A. Vaishampayan, Xu Chen, "Exploiting Virtualization for Delivering Cloud based IPTV Services," *IEEE Conference on Computer Communications Workshops*, pp. 637-641, April. 2011
- [4] SUN Bing-jue, WU Kai-jun, "Research on Cloud Computing Application in the Peer-to-Peer Based Video-on-Demand Systems," *3rd International Workshop on Intelligent Systems and Applications*, pp. 1-4, May. 2011
- [5] Xiaoying Wang, Zhihui Du, Xiaojing Liu, Hui Xie, Xuhan Jia, "An Adaptive QoS Management Framework for VoD Cloud Service Centers," *IEEE 11th International Conference on Computer and Information Technology*, pp. 379-384, Sep. 2011