

Windows Azure 기반의 N-스크린 서비스를 위한 실시간 동영상 재생 기법

이 원 주*, 임 현 용**

Real-time Video Playback Method for N-Screen Service Based on Windows Azure

Won-Joo Lee*, Heon-Yong Lim**

요 약

본 논문에서는 Windows Azure 기반의 N-스크린 서비스를 위한 실시간 동영상 재생 기법을 제안한다. 이 기법은 각 노드의 성능에 따라 원본 동영상을 차등 분할하여 재생 블록을 생성한다. 그리고 차등 분할한 재생 블록을 해당 노드에 할당하여 트랜스 코딩함으로써 트랜스 코딩 소요시간을 줄인다. 본 논문에서는 시뮬레이션을 통하여 제안한 동영상 재생 기법이 기존의 기법에 비해 N-스크린 서비스를 위한 실시간 동영상 재생에 효과적임을 보인다. 시뮬레이션에서는 상용화된 클라우드 시스템인 Windows Azure 기반에서 AVI 포맷 300MB 원본 동영상을 각 노드의 성능에 따라 차등 분할한다. 그리고 Windows Azure의 이기종 노드에 분산 할당하여 mp4와 Flv 포맷으로 트랜스 코딩하면서 트랜스 코딩 소요시간을 측정한다. 그 결과 본 논문의 제안 기법이 기존의 균등 분할 기법에 비해 Windows Azure 기반의 N-스크린 서비스의 성능을 향상시킨다.

▶ Keywords : N-스크린 서비스, 클라우드컴퓨팅 Windows Azure, 차등 분할.

Abstract

In this paper, we propose a real-time video playback scheme for the N-Screen service based on Windows Azure. This scheme creates several playback blocks based on the performance of each node by non-uniform splitting of the original video. To reduce transcoding-time, it allocates the

•제1저자 : 이원주 •교신저자 : 임현용

•투고일 : 2014. 3. 1, 심사일 : 2014. 4. 28, 게재확정일 : 2014. 6. 8

* 인하공업전문대학 컴퓨터정보과(Dept. of Computer Science, Inha Technical College)

** 한양대학교 ERICA 캠퍼스 컴퓨터공학과(Dept. of Computer Science & Engineering, Hanyang University ERICA Campus)

※ This work was supported by INHA TECHNICAL COLLEGE Research Grant.

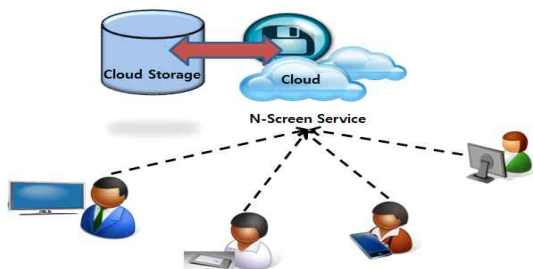
playback blocks to a corresponding node by transcoding the playback blocks. Through the simulation, we show that it is more effective to use real-time video playback for the N-screen service than the previous method. The proposed scheme splits an AVI format 300MB source video with non-uniform playback blocks. It allocates the playback blocks to the heterogeneous node of Windows Azure, the commercial cloud system and measures of transcoding-time by transcoding non-uniform playback blocks to mp4 and Flv format. As a result, the proposed scheme improves the performance of the N-screen service based on Windows Azure compared to the previous uniform split strategy.

▶ Keywords : N-Screen Service, Cloud Computing, Windows Azure, Non-uniform split.

1. 서 론

최근 Amazon, Google, IBM, Microsoft 등의 글로벌 기업과 삼성, NHN, 다음 등의 국내 기업 및 학계에서 클라우드 컴퓨팅 기반의 트랜스 코딩에 관한 많은 연구들을 수행하고 있다. 또한, 기업들은 다양한 IT기술을 활용하여 N-스크린 서비스를 준비하거나 시작하고 있다[1].

N-스크린 서비스란, 하나의 콘텐츠를 다양한 디바이스에서 연속적으로 이용할 수 있는 서비스를 의미한다[2]. N-스크린 서비스는 스마트 디바이스의 다양한 기능적 요소(3D, 고해상도, WiFi/3G, LBS/SNS 등)를 활용하여 그림 1과 같이 다수의 디바이스 간 협업 및 클라우드 컴퓨팅 자원을 활용하는 서비스이다.



클라우드 스토리지에 있는 동영상 콘텐츠는 플랫폼에 독립적이어야 N-스크린 서비스를 할 수 있다. 디바이스의 운영체

제, 화면 크기, 화면 비율, 동영상 지원 포맷 등 서비스를 요청하는 다양한 디바이스에서 N-스크린 서비스가 동작해야 한다[3]. 단일 동영상에 대하여 여러 해상도와 포맷을 유지하기 위해서는 해상도 및 포맷 변환을 위한 트랜스 코딩 작업이 필요하다. 트랜스 코딩 시간은 화질과 시간에 비례하기 때문에 트랜스 코딩 성능을 개선한다면 N-스크린 서비스 속도를 향상시킬 수 있다.

N-스크린 동영상 서비스는 하나의 동영상 콘텐츠를 여러 디바이스에서 이용할 수 있도록 트랜스 코딩을 수행해야 한다. 여기서 트랜스 코딩은 하나의 압축된 비디오 포맷을 먼저 미처리 상태의 비디오 프레임들로 디코딩한 뒤에 이를 새로운 포맷으로 다시 트랜스 코딩하여 다른 압축 포맷으로 변경하는 것이다[4]. 일반적으로 트랜스 코딩은 높은 성능을 지닌 서버급 컴퓨터에서 수행하거나 적은 수의 클러스터를 구성하여 수행한다. 그러나 인프라 구축 비용이 많이 소요되고, 동영상의 종류와 수가 많아지면 트랜스 코딩 소요시간이 증가하는 문제점이 있다. 이 문제점을 해결할 수 있는 하나의 방법은 클라우드 컴퓨팅 환경에서 트랜스 코딩을 수행하는 것이다. 이 방법은 일반적인 트랜스 코딩 방법에 비해 인프라 소요 비용이 적으며 다수의 노드를 클라우드에 추가함으로써 트랜스 코딩 시간을 단축할 수 있다.

본 논문에서는 클라우드 컴퓨팅 환경에 N-스크린 서비스 속도를 향상시킬 수 있는 실시간 동영상 재생 기법을 제안한다. 이 기법은 노드의 성능에 따라 원본 동영상을 차등 분할하고, 해당 노드에 할당하여 트랜스 코딩함으로써 트랜스 코딩 소요시간을 단축한다. 또한, 본 논문에서는 상용화된 클라우드 시스템인 Windows Azure 기반에서 제안한 기법의 성능을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하고 3장에서는 본 논문에서 제안하는 Windows Azure 기반의 실시간 동영상 재생 기법에 대하여 설명한다. 4장에서는 성능평가 및 분석결과를 제시하고 5장에서 결론을 맺는다.

II. 관련 연구

1. Windows Azure

Windows Azure는 마이크로소프트의 퍼블릭 클라우드 플랫폼이다[6]. Windows Azure는 마이크로소프트 데이터 센터에서 구동되는 다양한 클라우드 서비스들을 제공한다. Windows Azure가 제공하는 서비스들을 각각의 카테고리 별로 분류하면 그림 2와 같다.

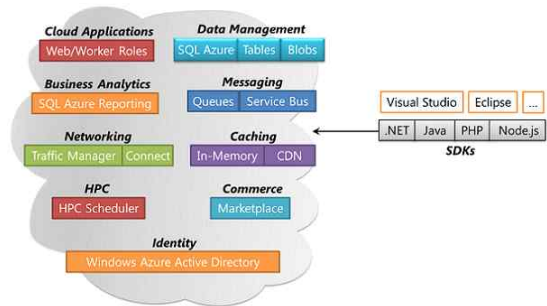


그림 2. Windows Azure 서비스
Fig. 2. Windows Azure Service

1) 클라우드 응용프로그램

Windows Azure 응용 프로그램은 C#, Java, PHP, Node.js 등의 언어로 코드를 작성한다. 윈도우 서버에서 구동되는 가상 머신에서 그 코드를 실행할 수 있다. Windows Azure에서는 모든 응용 프로그램이 하나 이상의 Role로 구현되어야 한다. 각 Role은 응용 프로그램의 기능을 수행하는데 필요한 코드와 구성 정보를 포함한다.

가상 머신에서 실행되는 Role은 그림 3과 같이 Web Role과 Worker Role로 분류 할 수 있다.

Web Role은 웹 브라우저나 기타 HTTP 클라이언트와 직접적으로 상호작용하는 코드를 위해서 설계되었다. Worker Role은 일반적인, 다양한 코드들을 실행하기 위해 설계된 것이다. 각 Role의 코드는 Role 인스턴스에서 실행된다. Role 인스턴스는 윈도우 서버를 구동하고 있는 가상머신(Virtual Machine)이다.

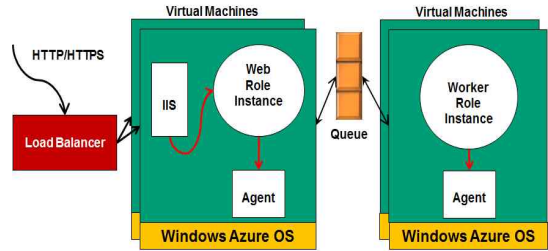


그림 3. 가상머신 내부구조
Fig. 3. Internal Structure of Virtual Machine

그림 4는 두 개의 Web Role 인스턴스를 실행하는 응용 프로그램이 데이터 센터 내에 존재하는 예이다. 각 Web Role 인스턴스는 동일한 윈도우 서버 상에서 동일한 응용 프로그램 코드의 복사본을 실행한다. 이때 Windows Azure는 자동으로 로드 밸런싱을 수행하여 모든 사용자의 요청을 두 개의 인스턴스에 균등하게 배분한다.

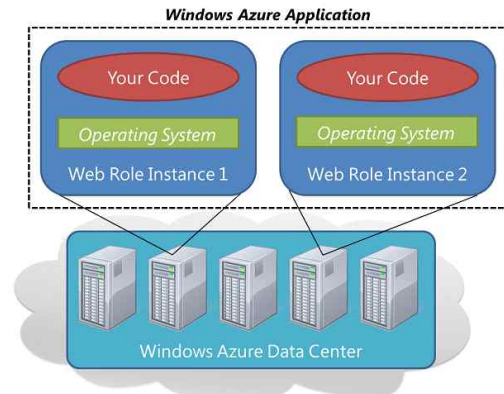


그림 4. Windows Azure 응용프로그램 예
Fig. 4. Example of Windows Azure application

2) 데이터 관리

Windows Azure 응용 프로그램은 하나 이상의 Role 인스턴스 안에서 실행된다. 즉, 하나 이상의 가상 머신 안에서 실행되며, 각각의 가상 머신은 응용 프로그램이 자유롭게 사용할 수 있는 로컬 저장소를 가진다. Windows Azure는 주기적으로 인스턴스들을 종료하고 변경하기 때문에, 로컬 저장소에 저장된 데이터는 아무런 예고도 없이 삭제될 수 있기 때문에 로컬 저장소에 파일이나 데이터를 저장하는 것은 바람직하지 않다. 응용 프로그램이 영구적으로 저장해야 하는 데이터는 가상 머신 외부에 저장한다.

Windows Azure는 그림 5와 같이 3가지 종류의 데이터 관리 옵션을 제공하고 있다.

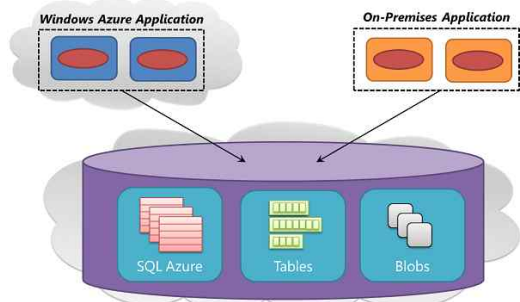


그림 5. Windows Azure의 데이터 관리 유형
Fig. 5. Data management type of Windows Azure

Windows Azure의 데이터 관리 유형은 관계형 스토리지 (SQL Azure), 확장 가능한 테이블(Tables) 그리고 자유 형식의 블랍(Blob)이다. 관계형 스토리지(SQL Azure)는 클라우드 기반의 SQL 서버로 RDBMS의 핵심적인 기능을 모두 제공한다. 테이블(Table)은 응용 프로그램이 string, integer, date 등 다양한 형식의 속성들을 저장할 수 있는 기능을 제공한다. 블랍(Blob)은 자유 형식의 이진 데이터를 저장할 수 있는 기능을 제공한다. 블랍은 동영상 저장, 데이터 백업, 그리고 이진 정보를 저장할 때 사용하면 유리하다.

3) 메세징(Messaging)

Windows Azure는 코드 간의 통신을 위해 큐(queues)와 서비스 버스(service bus)를 이용한다. 큐(queues)는 응용 프로그램이 메시지를 큐에 저장하면, 다른 응용 프로그램이 그 메시지를 읽는다. 그림 3에서 Web Role 인스턴스가 Worker Role 인스턴스와 통신할 때 큐를 사용한다.

서비스 버스는 클라우드 안에서 서로 다른 응용프로그램을 연결할 때 사용한다. 이때 응용 프로그램은 Windows Azure 응용 프로그램 이거나 다른 클라우드 플랫폼에서 구동되는 응용 프로그램일 수도 있다. 또한, 클라우드 외부에서 구동되는 응용 프로그램일 수도 있다. 서비스 버스는 Windows Azure 큐와 달리 게시(publish)와 구독(subscribe) 메커니즘을 제공한다. 어떤 응용 프로그램은 하나 이상의 토픽으로 메시지를 보낼 수 있고, 다른 응용 프로그램은 특정 토픽으로 전달된 메시지만을 가져올 수 있다.

4) 네트워킹(Networking)

Windows Azure는 미국에 4개, 유럽에 2개, 아시아에 2개로 현재 8개(2012년 6월)의 데이터 센터에서 지원되고 있다. 이러한 데이터 센터들 간에 요청을 전달하기 위해서 트래픽 관리자(Traffic Manager)를 제공한다. 또한, 자체 On-Premiss 서버에서 특정 데이터 센터에 존재하는 응용 프로그램에 연결하기 위해서 Connect 서비스도 제공한다. 세계적으로 많은 사용자들이 이용하는 응용 프로그램은 다양한 데이터 센터에서 Role 인스턴스를 실행하는 것이 효과적이다. 트래픽 관리자는 특정 데이터 센터의 응용 프로그램이 과부하로 수행이 중단 될 경우 자동으로 다른 데이터 센터로 연결하는 기능을 제공한다. Windows Azure의 트래픽 관리자 기능은 그림 6과 같다.

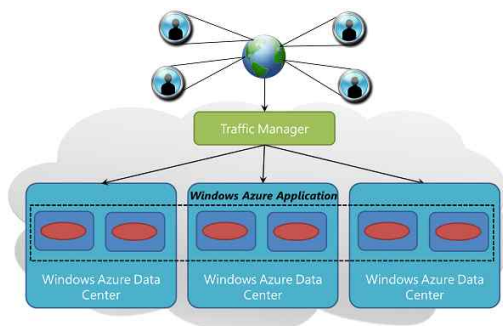


그림 6. Windows Azure의 트래픽 관리자
Fig. 6. Traffic Manager of Windows Azure

사용자는 일반적인 경우 가장 가까운 Windows Azure 데이터 센터로 접속하게 된다. 하지만, 그 데이터 센터의 응답 시간이 허용 범위를 넘어선다면, 다른 데이터 센터로 접속하게 된다. 이러한 문제를 해결해주는 내장 서비스가 트래픽 관리자이다.

Windows Azure의 Connect는 그림 7과 같이 On-Premiss 서버와 Windows Azure 응용 프로그램 간의 안전한 링크를 제공한다. Connect는 Windows Azure 응용

프로그램과 윈도우 서버를 구동하는 컴퓨터 사이에 안전한 IPSec 연결을 제공한다. On-Premiss 서버에 Connect 소프트웨어를 설치하고, Windows Azure 응용 프로그램의 설정을 조정하면 응용 프로그램은 그 컴퓨터와 연결된다. 따라서 로컬 네트워크에 존재하는 것처럼 해당 머신에 있는 데이터베이스에 접근할 수 있다.

5) 캐싱(Caching)

Windows Azure는 2 가지 유형의 캐시 서비스를 제공한다. 하나는 Windows Azure 응용 프로그램에서 사용되는 데이터를 메모리상에 캐시하는 In-Memory 캐싱이고, 다른 하나는 사용자에게 가까운 디스크 상의 블랍 데이터를 캐시하는 것이다. In-Memory 캐싱은 Windows Azure 응용 프로그램이 자주 사용되는 데이터에 빠르게 접근할 수 있는 기능을 제공한다.

Windows Azure CDN(Content Delivery Network)은 전 세계적으로 많은 위치에 블랍의 복사본을 저장할 수 있는 기능을 제공한다.

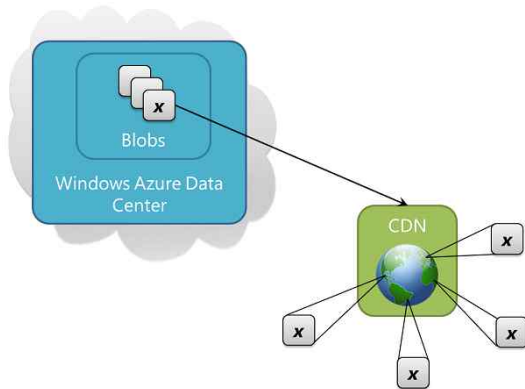


그림 8. Windows Azure의 CDN(Content Delivery Network)
Fig. 8. CDN(Content Delivery Network) of Windows Azure

그림 8의 CDN은 세계적으로 대단히 많은 사이트를 가지고 있다. 각 CDN 사이트는 Windows Azure 블랍(blob)의 복사본을 저장하는 기능을 한다. 어떤 지역의 사용자가 처음으로 특정 블랍에 접근하면, 그 정보와 데이터가 Windows Azure 데이터 센터에서 해당 사용자와 지리적으로 가까운 위치의 CDN 지역 저장소에 복사된다. 이후에 동일 지역의 사용자가 접근하게 되면 그때부터는 해당 지역 CDN에 캐시된 복사본이 사용되기 때문에 더욱 빠르게 접근할 수 있다.

6) 고성능 컴퓨팅(High-Performance Computing)

클라우드 플랫폼이 제공하는 가장 큰 장점 중 하나는 병렬 처리이다. Windows Azure에서 동시에 수 많은 Role 인스턴스를 실행시키고, 모든 인스턴스가 병렬로 작업을 수행한다. 이때 여러 인스턴스에 작업을 할당하는 스케줄러가 필요하다. Windows Azure는 그림 9와 같이 HPC 스케줄러를 제공한다.

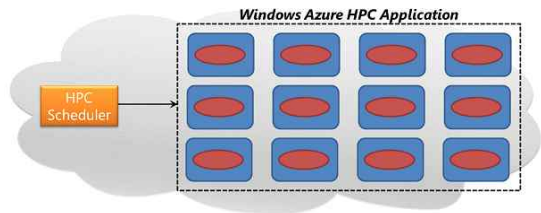


그림 9. HPC 스케줄러
Fig. 9. HPC Scheduler

HPC 스케줄러는 많은 Role 인스턴스에서 동시에 실행되는 병렬 응용 프로그램의 일정을 조정하는 기능을 제공한다.

7) Windows Azure SDK

Windows Azure는 윈도우 서버 기반의 환경으로 .NET 뿐만 아니라 다른 언어로도 Windows Azure 응용 프로그램을 개발할 수 있다. 마이크로소프트는 .NET, Java, PHP, Node.js 등 각각의 언어에 특화된 SDK를 제공하고 있다. 또한, C++나 Python과 같은 언어에도 Windows Azure SDK를 제공한다. 이러한 SDK는 비주얼 스튜디오와 이클립스(Eclipse)와 함께 사용할 수 있다.

2. 개인 기반 N-스크린 서비스

개인 기반 N-스크린 서비스는 스마트 콘텐츠 프레임 워크 적용된 서비스를 개인의 홈 네트워크 공간에 설치하여 개인 기반 스마트 콘텐츠를 이용할 수 있도록 함과 동시에 공공 스마트 콘텐츠 서비스와 연동하여 콘텐츠를 이용할 수 있는 기술을 의미한다[7].

그림 10의 개인 기반 N-스크린 서비스는 기존의 단말별로 상이했던 플랫폼 구조를 벗어난 단일 플랫폼을 사용함으로써 OS 및 단말의 제한을 받지 않는 서비스의 제공이 가능하여 추후 나오게 될 신규 단말에 대한 대응 및 확장이 용이하다는 특징을 가진다. 또한 단말 독립적인 플랫폼 구조를 통해 기존에 관계없이 HTML5 규격을 지원하는 웹 브라우저를 이용하여 어떤 단말에서도 동일한 콘텐츠의 재생 및 실행이 가능하여 지원 규격의 제한으로부터도 자유롭다는 특징을 가진다.

N-스크린 서비스 기술은 대용량 대용량 스마트 콘텐츠 품질 보증 기술을 지원한다. 웹 표준 기술을 기반으로 개발된 스마트 콘텐츠를 N-스크린 디바이스에서 안정적으로 서비스 하기 위해 N-스크린 디바이스의 성능을 확인하고 그에 따라 최적의 방식으로 운용되어야 한다. N-스크린 디바이스의 성능과 네트워크 속도 그리고 스마트 콘텐츠의 구성 요소 유형을 고려하여 최종 서비스 품질을 실시간으로 확인하고, 콘텐츠 서버, 인증 및 관리서버, 스트리밍서버, 동영상 포맷 변환 서버 및 데이터 연동서버의 기능을 가지는 설치형 서버 기술을 적용하여 지원 규격 및 플랫폼 독립적인 서비스가 가능하고, 동시에 연속적인(seamless) 동영상 재생 서비스가 필요하다.

따라서 본 논문에서는 N-스크린 서비스의 OSMU 특징을 반영하고 여러 디바이스에서 하나의 콘텐츠를 연속적으로 서비스 할 수 있는 실시간 동영상 재생 기법을 제안한다.

III. Windows Azure 기반의 실시간 동영상 재생 기법

본 논문에서 제안하는 Windows Azure 기반의 동영상 재생 기법은 사용자가 요청한 디바이스와 플랫폼에 적합한 동영상 콘텐츠를 제공하기 위해 트랜스 코딩한다.

1. 동영상 분할-병합

제안하는 실시간 동영상 재생 기법은 실시간으로 트랜스 코딩을 수행하기 위해 그림 11과 같이 원본 동영상을 분할하여 재생 블록을 생성한다.

그림 11의 파일 분할과정에서는 원본 동영상을 분할하여 인트로 블록과 재생 블록을 생성한다. 인트로 블록은 서비스 할 동영상의 시작점부터 일정 구간을 트랜스 코딩한 블록이다. 인트로 블록을 생성하는 목적은 사용자로부터 최초 재생 요청이 오면 요청 디바이스에 적합한 인트로 블록을 전송하여 즉시 동영상 재생 서비스를 제공하는 것이다. 또한, 인트로

블록을 제외한 나머지 부분을 트랜스 코딩 할 때 각 노드의 성능을 측정하는 것이다. 재생 블록은 원본 동영상에서 인트로 블록을 제외한 나머지 부분을 분할하여 생성한 블록이다. 실시간 재생 서비스를 제공하기 위해서는 제한시간 내에 재생 블록을 트랜스 코딩 하는 것이 필수적이다.

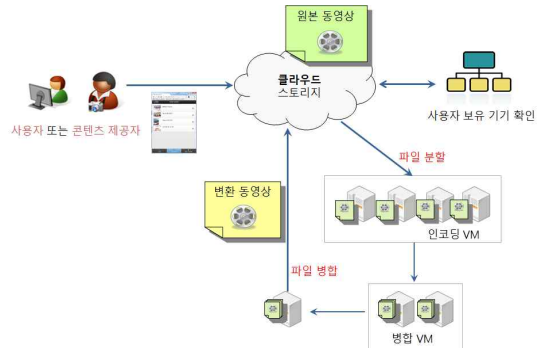


그림 11. 동영상 분할-병합 과정
Fig. 11. Video block split and merge process

원본 동영상은 사용자가 보유한 디바이스와 플랫폼에 적합하도록 트랜스 코딩함으로써 N-스크린 서비스를 위한 실시간 동영상 재생 서비스가 가능하다. 분할된 동영상 파일의 트랜스 코딩은 Windows Azure의 여러 가상 머신 인스턴스에서 수행된다. 이 트랜스 코딩 과정에서는 대용량 데이터 처리에 적합한 Map/Reduce 기법을 사용한다. Windows Azure에서 N-스크린 서비스를 위해서는 사용자가 요청한 작업을 수신하는 Web Role, Web Role에서 요청한 작업을 처리하는 Worker Role, 트랜스 코딩 파일을 저장하는 Blob이 필요하다. Worker Role에서 Map/Reduce 작업을 수행한다.

2. 동영상 재생기법

동영상 재생 서비스를 제공하기 위해서는 먼저 인트로 블록과 재생 블록을 생성한 후 사용자로부터 서비스 요청이 오면 인트로 블록을 전송하고 재생 블록을 트랜스 코딩하여 서비스를 제공한다.

재생 블록 생성과정에서 동영상 분할은 균등 분할과 차등 분할로 분류한다. 균등 분할은 각 머신의 성능에 관계없이 균등하게 분할하여 트랜스 코딩하는 방법으로 그림 12와 같다.

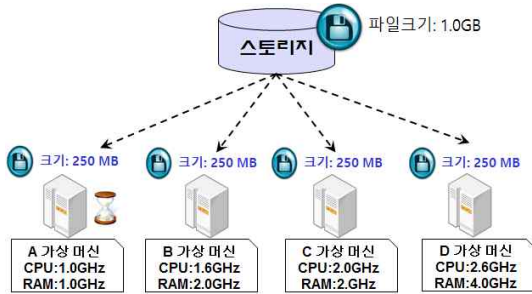


그림 12. 동영상 균등 분할 과정
Fig. 12. Video block uniform split process

그림 12에서는 1GB 동영상을 4개의 가상 머신 A~D에 250MB 단위로 균등 분할하여 할당한다. 각 가상 머신에 할당된 동영상의 트랜스 코딩 시간은 그림 13과 같이 해당 가상 머신의 성능에 따라 달라진다.

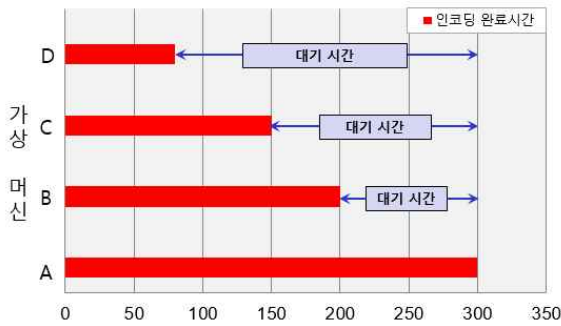


그림 13. 균등 분할 동영상 트랜스 코딩 시간
Fig. 13. Uniform split video block encoding time

그림 13에서 가상 머신 A의 성능이 가장 낮기 때문에 트랜스 코딩 시간이 300초로 가장 길다. 가상 머신 A보다 성능이 우수한 가상 머신 B, C, D는 할당된 동영상 트랜스 코딩을 먼저 완료하지만 A 머신의 트랜스 코딩 완료시간 까지 대기해야한다. 따라서 1GB 동영상의 트랜스 코딩 완료시간은 가상 머신 A의 트랜스 코딩 완료시간인 300초로 결정된다. 균등 분할 트랜스 코딩 방법의 문제점은 전체 트랜스 코딩 완료시간이 가장 성능이 낮은 가상 머신의 트랜스 코딩 시간에 따라 결정된다는 것이다. 이것은 이기종 클라우드 환경의 전체 성능을 저하시키는 요인이 된다.

따라서 본 논문에서는 각 가상머신의 성능 차이로 발생하는 대기시간을 줄일 수 있는 차등 분할(Non-uniform split) 방법을 제안한다.

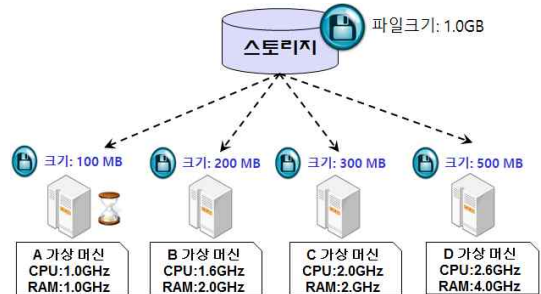
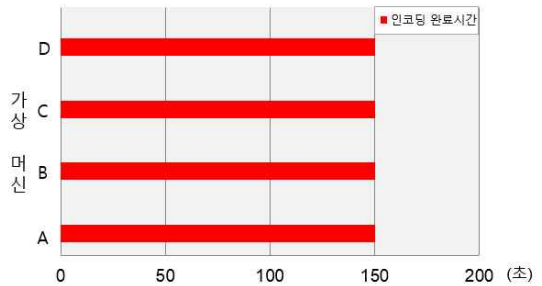


그림 14. 동영상 차등 분할 과정
Fig. 14. Video block non-uniform split process

차등 분할 방법은 각 가상머신의 성능에 따라 동영상을 차등 분할하여 할당한다. 그림 14와 같이 각 가상머신의 성능에 따라 가상머신 A, B, C, D에 각각 100MB, 200MB, 300MB, 500MB 크기로 동영상을 분할하여 할당함으로써 각 가상머신의 대기시간을 단축한다. 차등 분할 방법은 각 가상머신의 성능에 따라 동영상을 차등 분할하여 할당함으로써 각 가상머신의 트랜스 코딩 완료시간이 거의 동일하다. 또한 각 가상머신의 트랜스 코딩 완료시간이 전체 트랜스 코딩 완료시간과 동일하다. 그림 15의 차등 분할 방법의 전체 트랜스 코딩 완료시간은 그림 13의 균등 분할 방법에 비해 단축됨을 알 수 있다.



본 논문에서는 이기종 노드로 구성된 클라우드 컴퓨팅 환경을 전제로 하고 있기 때문에 각 노드들의 성능이 동일하지 않다. 따라서 각 머신의 성능에 따라 차등 분할하여 트랜스 코딩 한다. 이러한 트랜스 코딩 과정은 그림 16과 같다.

그림 16의 트랜스 코딩 과정을 수행하기 위해서 그림 3의 Web Role에서는 사용자가 요청한 트랜스 코딩 작업을 수신하고 Worker Role에서는 요청받은 트랜스 코딩 작업을 처리한다. 또한 Worker Role은 다수의 인스턴스로 구성되어 있고 각각 분할, 트랜스 코딩, 병합 작업을 포함한다. 이러한 과

정에서는 Map/Reduce[8][9] 기법을 활용하여 대용량의 동영상 콘텐츠를 처리한다.

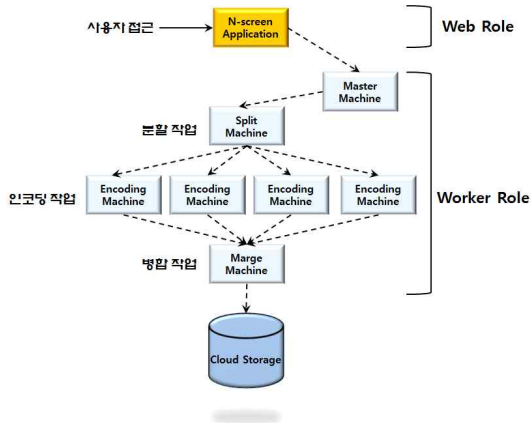


그림 16. Windows Azure에서 트랜스 코딩 과정
Fig 16. Transcoding Process on Windows Azure.

IV. 성능평가

본 논문에서 제안한 Windows Azure 기반의 동영상 재생 기법은 연속적인 동영상 재생 서비스를 제공한다. 따라서 성능평가에서는 Windows Azure 기반의 N-스크린 서비스에서 연속적인 동영상 재생 서비스가 가능한지를 검증한다.

1. 성능평가 환경

성능평가 환경은 Windows Azure 에뮬레이터를 기반으로 다수의 가상 인스턴스(Web Role, Worker Role)를 생성하여 수행한다. Windows Azure 실행 환경은 표 1과 같다.

표 2는 에뮬레이터에서 적용한 이기종의 성능을 지닌 가상 인스턴스의 성능과 역할을 나타낸다. 마스터노드는 동영상을 블록으로 분할하고 블록을 노드에 할당한다는 기능을 한다.

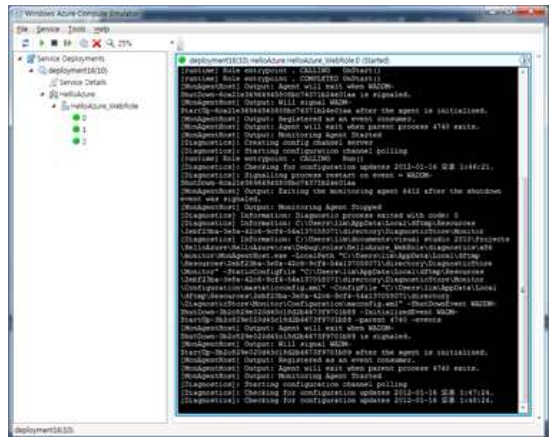
클라이언트 노드는 동영상 서비스를 요청하는 기능을 한다. 트랜스 코딩 노드는 마스터로부터 블록을 할당받아 트랜스 코딩을 수행한다. 트랜스 코딩에는 FFmpeg라이브러리를 사용한다[4].

표 2. Windows Azure 에뮬레이터의 가상 인스턴스 성능
Table 2. Virtual Instance specification of Windows Azure emulator

인스턴스	역할
Web Role Instance	클라이언트
Worker Role Instance	마스터
	트랜스 코딩

클라이언트는 Web Role Instance이며 마스터 및 트랜스 코딩은 Worker Role Instance를 나타내며 각 인스턴스는 큐를 통gkdu 작업을 송수신한다. 원본 동영상과 트랜스 코딩된 동영상은 스토리지(Bolb) 노드에 저장된다.

Windows Azure 에뮬레이터 구현 결과는 그림 17과 같다.



2. 성능평가 결과분석

성능평가에서는 AVI 포맷의 300MB 원본 동영상을 균등 분할 및 차등 분할하여 mp4와 Flv 포맷으로 트랜스 코딩한다. 성능평가 요소로는 트랜스 코딩 소요시간을 사용한다. 트랜스 코딩 소요시간은 원본 동영상을 트랜스 코딩 하는데 소

요되는 시간을 의미한다.

300MB 원본 동영상을 균등 분할하여 Windows Azure 애플레이터의 이기종 노드에 분산 트랜스 코딩하면서 소요시간을 측정 한 결과는 표 3과 같다.

표 3. 트랜스 코딩 소요시간(균등 분할)
Table 3. Transcoding time(Uniform split)

노드	할당량	소요시간	
		AVI->mp4	AVI->Flv
0	100MB	95초	72초
1	100MB	112초	140초
2	100MB	980초	780초

표 3을 살펴보면 AVI 포맷의 300MB 원본 동영상을 균등 분할하여 mp4와 Flv 포맷으로 트랜스 코딩 할 때 노드 0의 트랜스 코딩 소요시간이 가장 작다. 하지만 노드 2의 트랜스 코딩 소요시간이 mp4, Flv 포맷 각각 980초, 780초로 가장 크기 때문에 원본 동영상에 대한 전체 트랜스 코딩 소요시간은 각각 980초, 780초이다.

300MB 원본 동영상을 차등 분할 할 경우 트랜스 코딩 소요시간은 표 4와 같다.

표 4. 트랜스 코딩 소요시간(차등 분할)
Table 4. Transcoding time(Non-uniform split)

포맷	노드	할당량	소요시간
AVI->mp4	0	153MB	145초
	1	132MB	148초
	2	15MB	145초
AVI->Flv	0	187MB	135초
	1	96MB	134초
	2	17MB	150초

표 4를 살펴보면 AVI ->mp4 트랜스 코딩의 경우, 노드 0, 1, 2의 성능에 따라 300MB 원본 동영상을 153MB, 132MB, 15MB로 차등 분할하여 트랜스 코딩 한다. 이때 트랜스 코딩 소요시간은 거의 동일하다. 또한, AVI ->Flv 트랜스 코딩의 경우, 노드 0, 1, 2의 성능에 따라 300MB 원본 동영상을 187MB, 96MB, 17MB로 차등 분할하여 트랜스 코딩 한다. 이때 각 노드간의 트랜스 코딩 소요시간 차는 AVI ->mp4에 비해 크지만 거의 동일하다. 각 노드간의 트랜스 코딩 소요시간 차를 줄이기 위해서는 보다 더 정확한 노드 성능 측정 방법이 필요하다.

원본 동영상 300MB를 실시간 재생하기 위해 Windows Azure 애플레이터의 이기종 노드에 분산 트랜스 코딩하면서

측정한 트랜스 코딩 소요시간을 비교하면 그림 19와 같다.

그림 19를 살펴보면 본 논문의 제안기법인 차등 분할 기법이 균등 분할 기법에 비해 약 6배 우수한 성능을 보인다. 따라서 제안한 기법을 사용하면 실시간 재생 서비스가 가능함을 알 수 있다.

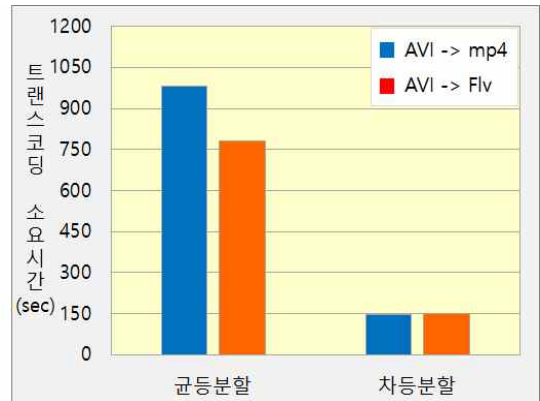


그림 19. 실시간 트랜스 코딩 결과(300MB)
Fig. 19. Result of Real-time Transcoding(300MB)

V. 결론

본 논문에서는 Windows Azure 기반의 N-스크린 서비스를 위한 실시간 재생 서비스 기법을 제안하였다. 이 기법은 원본 동영상을 각 노드의 성능에 따라 차등 분할한다. 그리고 차등 분할한 재생 블록을 해당 노드에 할당하여 트랜스 코딩함으로써 트랜스 코딩 소요시간을 단축한다. 본 논문에서는 상용화된 클라우드 시스템인 Windows Azure 기반에서 제안한 기법의 성능을 검증하였다. AVI 포맷 300MB 원본 동영상을 분할하여 Windows Azure의 이기종 노드에 분산 할당하여 mp4, Flv 포맷으로 트랜스 코딩하면서 트랜스 코딩 소요시간을 측정하였다. 그 결과 본 논문의 제안기법인 차등 분할 기법이 균등 분할 기법에 비해 약 6배 우수한 성능을 보였다. 따라서 제안한 기법을 사용하면 실시간 재생 서비스가 가능함을 알 수 있었다.

하지만 각 노드의 성능에 따라 차등 분할하는 원본 동영상의 크기가 달라지기 때문에 노드간의 트랜스 코딩 소요시간 차가 발생한다. 따라서 각 노드간의 트랜스 코딩 소요시간 차를 줄이기 위해서는 보다 더 정확한 노드 성능 측정 방법에 대한 연구가 필요하다.

참고문헌

- [1] Won Joo Lee, Jung Pyo Lee, and Y. I. Yoon, "A Design and Implementation of N-Screen Emulator Based on Cloud," Journal of The Korea Society of Computer and Information, Vol. 18, No. 3, pp. 11-18, Mar. 2013.
- [2] Y. I. Yoon, B. Kim, "N-Screen Service Standardization Based on Platform Type," KSCI Review, Vol. 20, No. 1, pp. 1-9, June 2012.
- [3] H. Y. Lim, C. H. Kim, W. J. Lee, C. H. Jeon, "Distributed Encoding Scheme for N-Screen Service in Cloud Computing," Korea computer Congress 2012, Communications of the Korean Institute of Information Scientists and Engineers, Vol. 39, No. 1, pp. 16-17, June 2012.
- [4] E. Ohwovoriolè and Y. Andreopoulos, "Rate-distortion performance of contemporary video codecs: Comparison of Google/WebM VP8, AVC/H.264, and HEVC TMuC," LENS Symp., London, Sept. 2010.
- [5] Encoding.com, 2012.7, <http://www.encoding.com/>
- [6] <http://azure.microsoft.com/en-us/documentation/articles/fundamentals-introduction-to-azure/>
- [7] KTH, "Smart Contents Creation and Service Technology based on N-Screen," Final Report, Apr. 2013.
- [8] Thilina G., J. Qiu and G. Fox, "Twister4Azure : Iterative MapReduce for Azure Cloud," Cloud Computing and Its Applications, CCA 2011, ANL, Chicago, IL. April, 12-13, 2011.
- [9] T. H. Keum, W. J. Lee, C. H. Jeon, "Design and Implementation of a Monitor for Hadoop Cluster," Journal of The Institute of Electronics Engineers of Korea, Vol. 49-CI, No. 1, pp. 8 -15, Jan. 2012.
- [10] Thilina, G., B. Zang, T. L. Wu, and J. Qiu, "Portable Parallel Programming on Cloud and HPC: Scientific Applications of Twister4Azure," In Proceedings of the forth IEEE/ACM

International Conference on Utility and Cloud Computing (UCC 2011), Melbourne, Australia, Dec. 2011.

저자 소개



이 원 주

1989: 한양대학교 전자계산학과 공학사.
 1991: 한양대학교 컴퓨터공학과 공학석사.
 2004: 한양대학교 컴퓨터공학과 공학박사.
 현 재: 인하공업전문대학
 컴퓨터정보과 교수.
 관심분야: 병렬처리시스템, 모바일컴퓨팅,
 성능분석, 클라우드컴퓨팅,
 N-스크린 서비스
 Email: wonjoo2@inhac.ac.kr



임 현 용

2010: 인하공업전문대학
 컴퓨터정보과 전문학사
 2013~현재: 한양대학교
 컴퓨터공학과 공학석사
 관심분야: 성능분석, 클라우드컴퓨팅,
 모바일 컴퓨팅,
 N-스크린 서비스,
 Email: heonyong.lim@gmail.com