JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 9, pp. 91-100, September 2020
https://doi.org/10.9708/jksci.2020.25.09.091

# Design and Implementation of the Evaluation Framework for Decentralized Multimedia Streaming Services

Sangsoo Park*

*Associate Professor, Dept. of Computer Science & Engineering, Ewha Womans University, Seoul, Korea

[Abstract]

This paper presents an evaluation framework for prototyping multimedia streaming services including audio and video in a distributed and/or decentralized storage that can evaluate service quality and performance under various network conditions. The evaluation framework focuses on important indicators which measure and improve service quality by applying decentralized storage to multimedia streaming services that can mimic the scalability of the existing server-client software architecture and the issue of a single point of failure. The integrated framework not only measures performance indicators for evaluating the quality and performance of multimedia streaming on open source based multimedia content streaming services, but also adjusts network quality using network virtualization technology for comprehensive evaluations. The experimental results show that the integrated framework has low overhead in building and operating a decentralized storage with multimedia streaming services on a single host computer which validates the scalability of the developed framework.

▸ **Key words**: Multimedia streaming, Decentralized storage, Performance evaluation, Quality of service, Network virtualization

[요 약]

본 논문은 네트워크 품질에 대한 서비스의 품질과 성능을 평가할 수 있는 분산형/탈중앙 스토리지에 오디오와 비디오를 포함하는 멀티미디어 스트리밍 서비스의 프로토타입을 설계하고 구현한 통합된 평가 프레임워크를 제안한다. 본 논문의 평가 프레임워크는 기존 클라이언트·서버 구조의 확장성과 단일 장애 지점의 문제를 극복할 수 있도록 멀티미디어 스트리밍 서비스에 분산형 스토리지를 적용하였으며, 서비스 품질을 측정하고 향상 시키는 중요한 지표에 초점을 맞추었다. 특히 탈중앙 스토리지 기반의 멀티미디어 콘텐츠 스트리밍 서비스에서 스트리밍의 품질과 성능을 평가하기 위한 성능 지표를 측정할 수 있을 뿐만 아니라 네트워크 가상화 기술을 이용하여 네트워크 품질을 조정할 수 있다. 실험 결과에 따르면 제안된 프레임워크는 단일 호스트 컴퓨터에 탈중앙 스토리지 기반 스트리밍 서비스를 구축하고 운영하는 데 있어 오버헤드가 낮음을 보여주었으며, 평가 가능한 시스템 규모의 확장성을 검증하였다.

▸ **주제어**: 멀티미디어 스트리밍, 탈중앙 스토리지, 성능 평가, 서비스 품질, 네트워크 가상화

• First Author: Sangsoo Park, Corresponding Author: Sangsoo Park
*Sangsoo Park (sangsoo.park@ewha.ac.kr), Dept. of Computer Science & Engineering, Ewha Womans University
• Received: 2020. 08. 13, Revised: 2020. 08. 31, Accepted: 2020. 09. 01.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

## I. Introduction

넷플릭스, 아마존 비디오 등 인터넷 기반 멀티미디어 스트리밍 서비스는 기술 완성도와 사용 편의성이 높은 기술 중 하나이다. 대부분의 인터넷 스트리밍 서비스는 영상·음원 등 멀티미디어 스트리밍 소스를 공급하는 미디어 서버, 서버에서 미디어 스트림을 수신·유통하는 스트리밍 서버, 미디어 스트림을 수신·재생하는 스트리밍 플레이어로 구성된다.

기존 중앙 집중식 클라이언트·서버 서비스 구조에서는 스트밍 데이터의 트래픽 집중으로 병목 현상이 발생하여 이용 가능한 스트리밍 단말의 수가 제한된다. 이러한 병목 현상을 개선하기 위해 멀티미디어 캐시 서버를 도입할 수 있지만, 기본 클라이언트와 서버 구조의 한계를 극복하기는 어렵다. 이러한 한계를 극복하기 위해 P2P (Peer-to-Peer) 방식의 분산형 소프트웨어 구조에서 소프트웨어 구조를 완전히 바꾸고 개별 스트리밍 단말기의 네트워크 대역폭, 컴퓨팅 전력, 저장 공간을 활용하는 기술이 도입되었으며 이는 이론적으로 무한한 확장성을 갖는다.

본 논문은 P2P 네트워크 기반 분산형 스토리지 혹은 탈중앙 스토리지에 저장된 오디오 및 비디오를 포함한 멀티미디어 콘텐츠 스트리밍 시스템의 프로토타입을 설계 및 구현하고 P2P 네트워크에서 발생할 수 있는 다양한 네트워크 지연 시간 및 대역폭 제한에 대한 멀티미디어 스트리밍 서비스의 품질과 성능을 평가하는 프레임워크를 제안한다. 이를 위해 오픈소스를 활용한 탈중앙 스토리지 상에 멀티미디어 스트리밍 서비스를 필요에 따라 추가로 생성가능한 가상화 플랫폼에 구현하고, 가상화 플랫폼의 네트워크의 설정을 변경하여 다양한 네트워크 대역폭에 대해 멀티미디어 스트리밍 서비스의 품질과 성능을 측정할 수 있다.

제안된 평가 프레임워크를 위한 멀티미디어 스트리밍 서비스를 지원하는 프로토콜로는 Apple사의 멀티미디어 스트리밍 프로토콜 표준인 HLS (HTTP Live Streaming)가 사용되었다[1]. HLS는 기존 웹서버를 수정하지 않고 네트워크 품질에 따라 인코딩된 멀티미디어 콘텐츠와 적응적으로 재생해 웹 브라우저에서 조작이 가능하다는 장점이 있다[2].

또한 가상화 플랫폼으로는 도커 (Docker)를 사용하였으며 도커는 컨테이너라는 라이브러리, 시스템 툴 실행 파일, 소프트웨어 코드 등 소프트웨어를 실행하는 모든 것을 포함하는 단위로 구성된 소프트웨어를 패키징하는 플랫폼이다[3]. 컨테이너 생성 후 추가 또는 수정된 정보. 컨테이너에 추가로 저장해 여러 컨테이너를 신속하게 복제·실행할 수 있고, 생성 후 추가 또는 수정된 정보만 컨테이너에 저장하기 때문에 동일한 애플리케이션을 여러 별개의 사본으로 실행하는 데 적합하다. 특히 가상 네트워크와 같은 가상화 기법이 사용되기 때문에 본 논문의 평가 프레임워크에 적합하다.

마지막으로 P2P 네트워크 기반 분산형 스토리지를 지원하는 프로토콜로 IPFS (InterPlaneray File System)[4]를 사용하였다. IPFS는 중앙집중식 서버가 없는 피어 노드 간 통신을 통해 일부 노드의 연결이 끊겨도 인터넷에 연결된 컴퓨터를 다른 연결된 노드로 서비스할 수 있다는 장점이 있다. 특히 Go 프로그래밍 언어로 작성된 Go-ipfs 프로젝트에서 제공되는 IPFS 피오 노드 구성 애플리케이션을 적용하였다[5].

제안된 탈중앙 스토리지 기반 멀티미디어 스트리밍 성능 평가를 위한 프레임워크에서 다양한 네트워크 상태에 따라 스트리밍 서비스의 품질을 측정하고 단일 호스트 컴퓨터에서 수 십개에서 최대 200개까지의 피어 노드를 생성하여 연결하여 시스템 규모 면에서의 확장성을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서는 분산형 스토리지 상에서의 성능평가 관련 기존 연구들을 살펴보고 본 논문에서 제안된 프레임워크와 비교한다. 3장에서는 제안된 프레임워크의 세부 설계와 구현 방법 및 최종 통합을 기술하며 4장에서는 구현된 통합 프레임워크의 효용성을 검증하기 위해서 오버헤드 및 단일 호스트 컴퓨터에서 생성하는 피어 노드의 수에 따른 CPU 워크로드를 측정한 실험결과를 제시한다. 마지막으로 5장은 본 논문의 결론을 기술한다.

## II. Related Work

실시간 멀티미디어 데이터베이스 혹은 빅데이터 스트리밍 컴퓨터 플랫폼에 대한 연구는 기존 데이터베이스가 문자 및 숫자 등의 전통적인 데이터 유형에 대해 저장, 관리 및 검색을 지원하였던 것에 비해 이미지, 사운드 및 비디오와 같은 멀티미디어 미디어 데이터를 지원하기 위한 이론과 기술로 확장되어 왔으며, 본 논문이 대상으로 하는 멀티미디어 스트리밍 서비스의 근간이 되는 기술적인 연구를 포함하고 있다.

이러한 실시간 멀티미디어 데이터베이스 플랫폼 설계의 고려사항으로는 성능, 확장성, 동적 리소스 추가를 위한 유연성, 향후 리소스 사용 증가에 대한 예측 가능성이 필요하다[6]. 아파치 하둡 기반의 병렬 프로그래밍 프레임워크 기반의 대용량 미디어 데이터를 저장할 수 있는 효율적인 구조 및 분산 처리를 위한 HDFS (Hadoop Distributed Files Systems)[7-8]에 대한 확장된 연구가 존재하나, 대규모의 노드 기반 실시간 멀티미디어 제공에 한계점을 갖고 있다.

또한, 네트워크에서 활용도가 낮은 대역폭을 사용하여 지리적으로 분산된 클라우드에서 서로 다른 시간에 서로 다른 기간에 남은 대역폭을 멀티미디어 빅데이터 전송에 대한 수요를 충족시키기 위한 주문형 대역폭(Bandwidth-on-Demand)[9] 연구가 진행되고 있으나, 실제 적용되기 위해서는 대규모 분산 멀티미디어 서비스에 대한 성능평가가 필요하다.

P2P 네트워크 기반 분산 스토리지의 프로토타입을 구현하고 탈중앙 스토리지에 대한 서비스 품질을 평가하기 위한 다양한 관련 연구가 있다. [10]에서는 IPFS 기반의 P2P 스토리지는 데이터를 전달하기 위해 높은 처리량에 의존하도록 설계되었기 때문에 개인용 컴퓨터에는 부적절하다는 점에 착안하여 그 원인의 분석을 위해 데이터 처리량 문제를 분석하였다. IPFS에서 일반적인 다운로드 속도는 1MB/s 미만으로 알려져 있으나 IPFS 서비스를 위한 시스템 응용 프로그램이 시작 된 후 네트워크 대역폭이 즉시 700-800KB/s로 증가하여 시스템이 느리게 반응하고 그에 따라 사용자 경험이 저하된다는 결과를 제시하였다.

포그/에지 컴퓨팅 인프라에서 객체기반 스토리지 시스템에서의 성능 분석은 [11-12]에서 광범위하게 수행되었다. 해당 논문에서 채택한 기본 분산형 스토리지 플랫폼인 Rados [13], Cassandra [14], IPFS [15]에 대한 정량적인 분석을 수행했으며, 평가를 위한 소프트웨어 아키텍처로는 (1) 일반 클라우드에 구축된 기본 구성의 IPFS (2) 포그/엣지 인프라에 구축된 기본 구성의 IPFS, 그리고 마지막으로 (3) IPFS 및 포그/엣지 컨텍스트 독립형 확장 솔루션이 결합된 세 경우를 적용하였다.

플래시 군중 (Flash crow)는 웹 사이트가 인기 뉴스 에 언급된 후 예상치 못한 방문자 급증으로 인해 웹 사이트 서비스의 심각한 성능 저하를 설명하는 데 사용되는 용어로서, DNS 부하 배분 및 지역 복제 서비스와 같은 웹 서버에 대한 플래시 군중의 영향을 줄이기 위한 기술이 제안되었다[16]. 이 논문은 플래시 군중 동안의 파일 배포에서 IPFS 프로토콜의 성능을 분석하였으며, 특히 시간 경과에 따른 피어 노드에서의 다운로드 속도 변화를 분석하고 피어 수가 증가함에 따라 다운로드 성능이 어떻게 변하는지 제시하였다. 또한 블록체인과 IPFS를 기반으로 사물인터넷 보안 모델을 구축하고 평균 지연시간과 처리량 측정을 통해 제안된 모델의 성능을 평가한 연구도 존재하는데 이때 실험에 사용된 노드 수는 각각 6, 8, 12개로 제한된다[17].

본 논문의 접근 방법과 유사한 시뮬레이션에 의해 실제 네트워크의 환경을 모방하여 물리적 장치 및 실제 배포 없이도 최소 비용으로 테스트하는 방법이 제안되었으며, [18]에서는 소프트웨어 정의 네트워크 기반의 DASH 비디오 스트리밍 서비스에 대해 시뮬레이션 플랫폼이 적용 되었으나, 클라이언트·서버 소프트웨어 구조에 기반하여 확장성이 제한된다.

본 논문의 통합 평가 플랫폼은 수백 개의 노드로 확장할 수 있는 P2P 네트워크 기반 탈중앙 스토리지에 저장된 오디오 및 비디오를 포함하여 멀티미디어 콘텐츠 스트리밍 시스템의 프로토타입을 구축하고 평가하는 등 여러 면에서 기존의 연구와 차별된다. 특히 가상화 기술을 사용하여 P2P 네트워크에서 발생할 수 있는 다양한 네트워크 대기 시간 및 대역폭 제한에 대한 멀티미디어 스트리밍 서비스의 품질 및 성능을 측정할 수 있다.

## III. The Proposed Framework

### 1. Decentralized Storage Node Generator

#### 1.1 Docker-Based IPFS Peer Node

도커는 컨테이너라는 라이브러리, 시스템 툴 실행 파일, 소프트웨어 코드 등 소프트웨어를 실행하는 모든 것을 포함하는 단위로 소프트웨어를 패키징하는 구조를 갖는다. 각 컨테이너 객체에 대해 운영 체제를 별도로 운용하지 않기 때문에 서비스 응용을 신속하게 복제·실행할 수 있다는 장점이 있다[3]. 도커 이미지에는 컨테이너 객체를 실행하는 데 필요한 파일 및 설정이 포함되어 있으며, 특정 목적을 위해 구성된 이미지를 실행하는 동안 컨테이너 생성 후 추가 또는 수정된 정보가 컨테이너에 저장된다. Fig 1과 같이 하나의 이미지에서 복수의 컨테이너를 신속하게 복제·실행할 수 있으며, 실행 후 사용하는 정보가 컨테이너 객체마다 다르더라도 생성 후 추가되거나 수정된 정보만 컨테이너에 저장되므로 다수의 피어 노드를 생성하여 연결하는 시스템 상에서 구현되는 P2P 네트워크 기반 탈중앙 스토리지의 설계 및 구현에 적합하다.

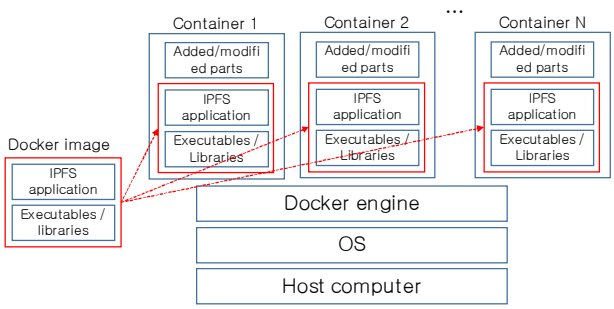



Fig. 1. Docker Image and Container Structure: an Example of IPFS Application

IPFS는 인터넷에 연결된 컴퓨터를 위한 분산 P2P(Peer-to-peer) 파일 시스템으로 Fig. 2와 같이 중앙 집중식 서버가 없는 피어 노드의 P2P 통신으로 인해 서버 장애나 클라이언트 간의 인터넷 장애로 접속이 차단되면 서비스가 불가능한 서버-클라이언트 구조와 달리 일부 노드의 연결이 끊겨도 연결된 다른 노드에 P2P 서비스를 이용할 수 있는 탈중앙 스토리지 시스템이다[19].

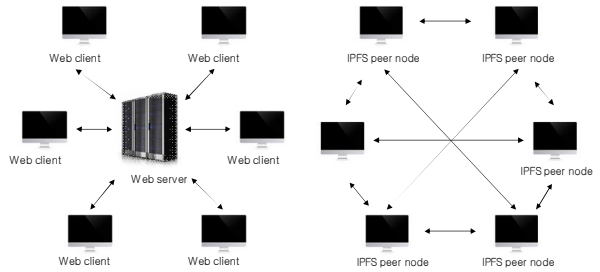



Fig. 2. Web Server-Client Structure and IPFS P2P Structure Comparison

본 논문의 탈중앙 스토리지 프레임워크는 Go 프로그래밍 언어에 기반의 IPFS 피어 노드를 구성하는 애플리케이션을 탑재하고 있는 도커 이미지 (ipfs/go-ipfs:latest)를 적용하여 IPFS 피어 노드에 필요한 파일이 저장될 위치인 ipfs_data 환경 변수를 설정하고 도커 명령인 도커를 사용하여 IPFS 피어 노드를 구성하였다 [4-5].

### 1.2 Multiple Peer Node Generation in IPFS

IPTB (InterPlanetary TestBed)는 단일 호스트 컴퓨터 내에 샌드박스 처리된 IPFS 피어 노드의 클러스터를 생성 및 관리할 수 있게 하며 도커의 가상 네트워크로 연결된 IPFS 네트워크의 사용이 가능하다 [20-21]. 즉, 하나의 호스트 컴퓨터에 여러 개의 피어 노드를 생성할때 때 호스트 컴퓨터와 피어 노드 사이에 네트워크가 생성되어 연결되며, 해당 네트워크를 통해 해당 호스트 컴퓨터에 생성된 피어 노드들을 제어할 수 있다. 특히 기본적으로 임의의 수의 서로 연결되지 않은 IPFS 피어 노드를 생성하여 모든 피어 노드 간에 연결하거나 선별된 일부 피어 노드만 연결할 수 있는 등 기능을 제공하기 때문에 본 논문의 프레임워크에 적용되었다.

Fig. 3의 예는 0부터 4까지 5개의 노드를 생성하고 노드 0과 노드 4를 연결하고 데이터 공유의 기본 접근 방법인 해시 값을 사용하여 노드 4에서 IPFS 네트워크의 데이터 접근하는 방식을 기술한다.

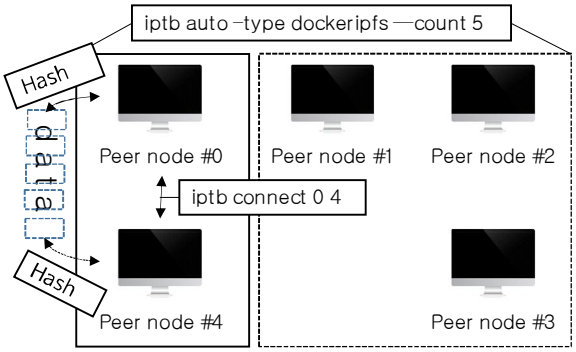



Fig. 3. IPFS Network Configuration in the Proposed Framework

이를 위해 제안된 평가 프레임워크는 Fig. 3의 예제에서 다음의 순서로 IPFS 피어 노드를 구성하고 가상 네트워크를 연결하며 데이터를 공유하고 접근한다.

(1) ipfs/go-ipfs 이미지 기반 컨테이너를 5개 생성
(2) 연결되지 않은 전체 피어 노드의 수행을 시작
(3) 0번 피어 노드에서 데이터를 IPFS에 추가: 현재 피어 노드들이 연결되어 있지 않기 때문에 0번 피어 노드에서만 접근이 가능한 상태임
(4) 0번 피어 노드에서 추가된 데이터의 해시 값을 공유
(5) 호스트 컴퓨터에서 0번과 4번 피어 노드를 연결
(6) 4번 피어 노드에서 IPFS 네트워크에 공유된 해시 값을 사용하여 데이터에 접근

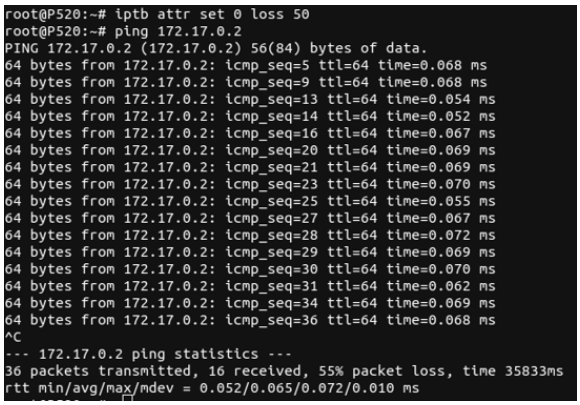
### 1.3 Network Bandwidth Adjustment Method

TC (Traffic Control)은 리눅스 운영 체제에서 네트워크 트래픽 제어 기능을 제공하는 도구로서 응용 수준에서 데이터 비트 전송률을 제어할 수 있으며 커널 수준에서 패킷 전송 스케줄링 정책을 설정할 수 있다[22]. 본 논문의 프레임워크는 다양한 네트워트 상태에 대한 시뮬레이션을 위해 IPTB 도커 플러그인의 TC를 통합하고 Table 1의 명령 수준에서 각 피어 노드의 네트워크 상태를 제어하는 방법을 사용하였다.

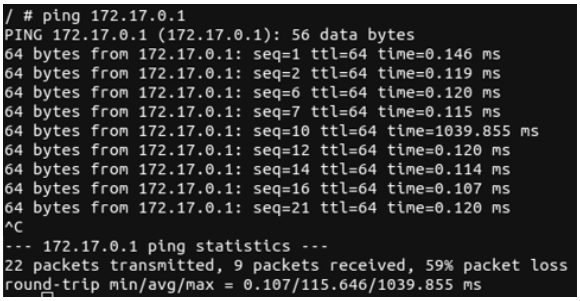
Table 1. IPTB Docker Plugin TC Control Items

| Item | Function |
|---|---|
| latency | Packet Delay Time (ns) |
| bandwidth | Network Bandwidth (Mbps) |
| jitter | Packet Jitter (ns) |
| loss | Packet drop rate (0-100) |

본 논문에서는 IPTB 도커 플러그인에 통합된 TC의 동작을 검증하기 위해 Fig. 3의 환경에서 다음의 실험을 수행하였다. 실험에서는 먼저 TC가 IPTB의 가상 네트워크에 연결된 피어 노드와 호스트 컴퓨터 사이에서 그리고 피어 노드와 피어 노드 사이에서 정상적으로 작동하는지 확인하기 위해 수행하였다. ping 명령문을 통해 패킷 손실률을 확인할 수 있기 때문에 이에 기반하여 손실률이 설정되지 않았을 때 호스트 컴퓨터와 피어 노드 사이에 ping을 수행함으로써 손실율이 0%임을 확인하였다.



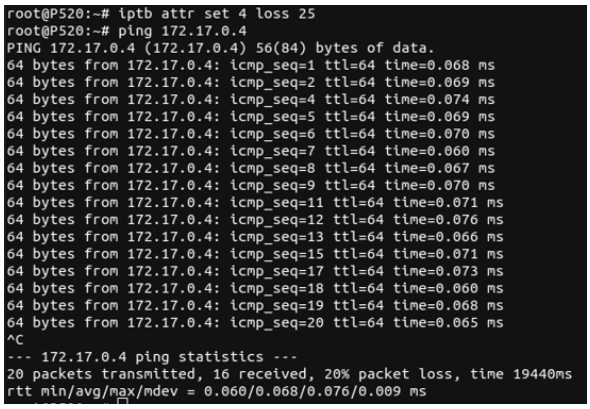

(a) Node 0 to 1



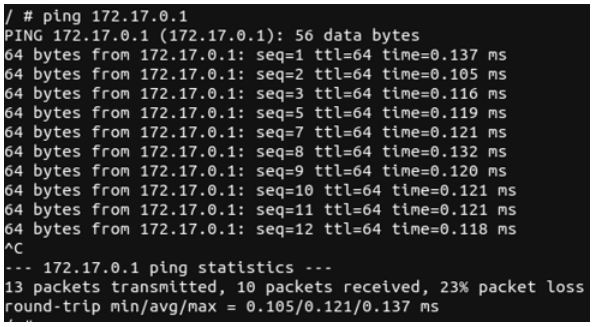

(b) Node 1 to 0

Fig. 4. Experimental Results for the Packet Loss Rate between Node 0 and Node 1

패킷 손실률이 설정되었을 때의 실험을 위해 Fig. 4, Fig. 5와 같이 피어 노드 1의 손실률은 50%로 피어 노드 4의 손실 항목은 25%로 설정하였다. Fig, 4와 같이 ping 명령에서 호스트 컴퓨터에서 피어 노드 0까지의 패킷 손실률은 55%로 그 반대의 경우 59%로 나타났다. 또한 피어 노드 0과 4의 패킷 손실률은 각 방향에 따라 20%, 23%로 측정되어 TC 손실 항목의 설정에 따라 가상 네트워크에서 패킷 손실이 정상적으로 동작하는 것을 검증하였다.




(a) Node 0 to 4




(b) Node 4 to 0

Fig. 5. Experimental Results for the Packet Loss Rate between Node 0 and Node 3

## 2. Integrated Evaluation Framework

본 논문에서는 기술된 도커 기반 단일 호스트 컴퓨터에서 다수의 탈중앙 스토리지 시스템인 IPFS 피어 노드를 생성하기 위한 오픈소스 프로젝트인 IPFS, IPTB, TC를 통합하였으며, 멀티미디어 스트리밍 서비스와 서비스 품질 측정을 위해서 본 논문의 프레임워크에서 통합된 다수의 IPFS 피어 노드에 Fig. 6과 같이 Apple사에서 제시한 HLS 스트리밍의 품질을 평가하는 척도인 KPI 성능 지표를 지원하는 HLS 기반 멀티미디어 스트리밍 플레이어와 이를 구동하기 위한 웹 인터페이스가 통합하였다[23].

통합 프레임워크에서는 웹 인터페이스에서 개별 피어 노드 그룹으로 나눌 수 있도록 생성될 피어 노드를 임의의 개수의 그룹으로 나누고, 네트워크 품질이 낮은 피어 노드 그룹과 네트워크 품질이 높은 피어 노드 그룹을 그룹화하여 KPI 성능 지표 측정을 가능하게 구현하였다. 이를 위해 각 그룹에 대한 튜플 (그룹 이름, 그룹에 포함된 피어 노드 수, 최소 네트워크 대역폭 및 최대 네트워크 대역폭)을 설정할 수 있다. 이때 각 그룹에 대해 HLS 스트리밍이 가능한 IPFS 피어 노드를 생성하고, 지정된 네트워크 대역폭 범위 내에서 임의의 대역폭 값을 생성 및 설정한다.

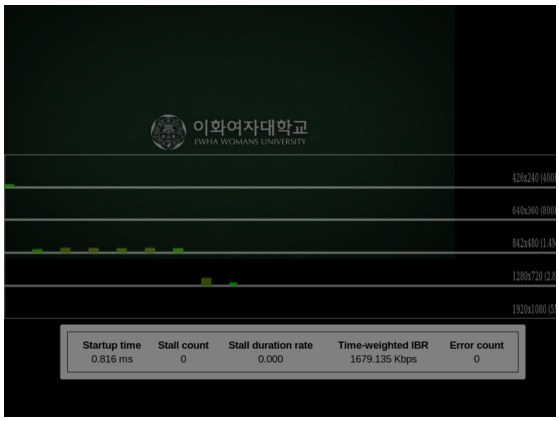



Fig. 6. HLS Performance Tool with KPI index


Fig. 7. Web Interface for the Integrated Evaluation Platform

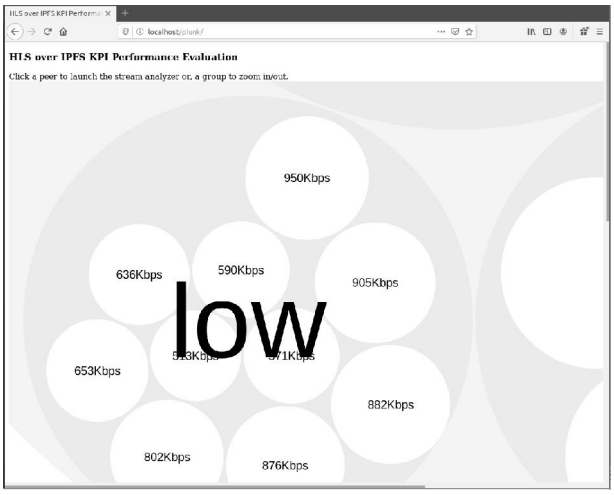



Fig. 8. An Example of the Peer Node Group in 'low' Network Bandwidth zoomed in

생성된 노드 목록은 향후 KPI 성능 지표를 지원하는 HLS 기반 멀티미디어 스트리밍 웹 기반 플레이어에 전달되고 웹 인터페이스에서 각 그룹에 대해 피어 노드가 설정한 대역폭을 확인할 수 있다. 예를 들어, Fig. 7, Fig. 8과 같이 'low', 'medium', 'high'의 3개 피어 노드 그룹에 대해 (각각 네트워크 대역폭 범위: 500Kbps~1000Kbps, 500~2500Kbps, 3EA(1000~100Kbps)를 설정할 수 있다.

생성된 IPFS 클러스터의 피어 노드는 소속된 네트워크 대역폭에 비례하여 원의 크기를 조정하는 인터페이스가 구현되어 각 그룹의 피어 노드는 원형으로 그룹화하고, 그룹 내의 피어 노드에 마우스 클릭 한 번만으로 액세스할 수 있도록 확대/축소 기능을 구현하였다. 이러한 원 형태의 웹 인터페이스의 구현을 위해 d3.js 기반 확대 가능한 원형 패키지 [24-25]을 적용하였다.

## IV. Experimental Results

본 논문에서 제안된 탈중앙 스토리지 기반 멀티미디어 스트리밍 성능 평가 프레임워크의 효과를 검증하기 위해 Table 2와 같은 환경에서 적용하여 다음의 실험을 수행하였다.

개발된 통합 성능 평가 프레임워크가 실제로 적용되려면 프레임워크가 실행되는 호스트 컴퓨터에서 서로 연결된 실행 가능한 도커 기반 사설 IPFS 클러스터를 구축하는 데 필요한 시간이 허용 가능한 범위 내에 있는지 확인해야 할 필요가 있다.

Table 2. Experimental Setup

| Item | Specification | Performance Index |
|---|---|---|
| CPU | Intel Xeon W-2123 | # of Cores: 4<br># of Threads: 4<br>Cache: 8.25MB<br>Op. Freq.: 3.60Ghz |
| RAM | DDR4 PC4-21300 | Capacity: 192GB<br>Op. Freq.: 2666Mhz |
| Storage | Intel SSD 660P | Capacity: 1TB (QLC)<br>Sequential Access: 1800MB/s<br>Random Read/Write: 150K/220K IOPS |
| OS | 64-Bit Ubuntu Desktop Linux | Ubuntu Desktop 18.04.03 64-Bit |

이를 위해 Fig. 9와 같이 통합 성능 평가 프레임워크에서 IPFS 피어 노드의 수를 증가시키면서 플랫폼 구축에 필요한 시간을 측정하였다. 실험 결과에 따르면 최대 약 200개의 노드(우분투 리눅스와 도커 컨테이너의 한계로 인해 만들 수 있는 최대값)를 구축하는 데 최대 189초가 걸린 것으로 측정되었다. Fig. 9에서 실행 시간이 피어 노드 수에 비례한다는 것을 알 수 있듯이 최대 40개 노드에

대해 노드당 약 2초, 그 이상의 노드에서는 노드당 약 1초가 소요된 것으로 측정되어 탈중앙 스토리지 생성을 위한 단발성 오버헤드가 매우 낮은 것을 확인하였다.

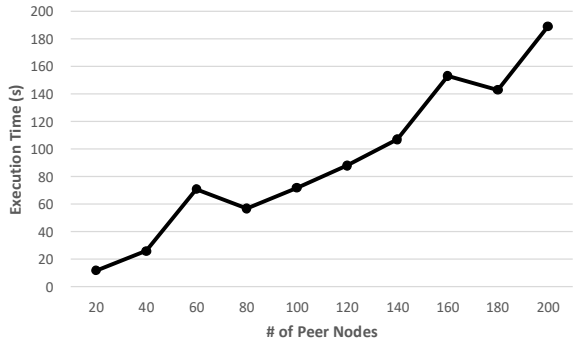



Fig. 9. The execution time of building an IPFS cluster according to the number of nodes in Docker

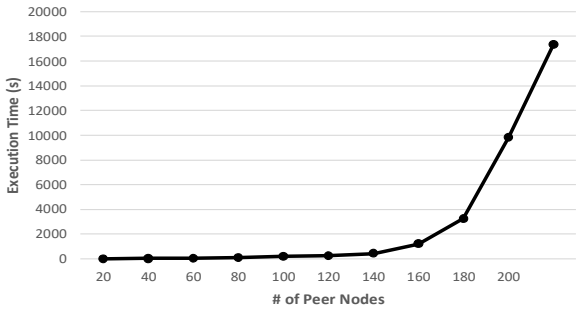



Fig. 10. The execution time of building an IPFS cluster according to the number of nodes in KVM

본 논문에서 대상으로 하는 하나의 호스트 컴퓨터에서 가상화 기술 기반의 다중 노드를 생성하는 방법으로는 도커 기술을 사용하였으며, 하이퍼바이저 기반의 KVM을 비교대상으로 하였다[26]. 하이퍼바이저 기반의 가상화 기술은 가상 노드간의 리소스를 서로 격리하고 사용량을 어느 정도 제어 할 수 있기 때문에 광범위하게 사용되지만, 빅데이터에 대해서는 오버헤드 문제가 존재하는 것으로 알려져 있다[27].

Fig. 10은 은 본 논문의 통합 성능 평가 프레임워크를 KVM 상에서 구동한 것으로 IPFS 피어 노드의 수를 증가시키면서 플랫폼 구축에 필요한 시간을 측정한 것으로 노드 수가 100개를 넘으면 오버헤드가 기하급수적으로 증가되어 본 논문의 도커 기반의 프레임워크에 비해 수십배의 오버헤드 증가가 발생함을 확인할 수 있다.

통합 성능 평가 프레임워크가 IPFS 클러스터를 구축하고, 배포가 완료되어 각 도커 컨테이너가 수행 중일 때 호스트 시스템의 자원 활용률 (CPU, 메모리)은 Fig. 11, Fig. 12, Fig 13과 같이 측정되었으며 측정은 시스템 모니터 도구 (gnone-system-monitor [21])를 사용하였다. 실험 결과에 따르면 메모리 사용량의 경우 200개 이상의 피어 노드를 사용했을 때에도 총 5.9GB (192GB의 3.1%)만 사용되었다. 또한, CPU 사용량의 경우 클러스터 구축 시 피어노드 수에 관계 없이 100% 활용도를 보여주는 기간이 있지만 이 기간은 Fig. 11과 같이 피어 노드당 1~2초 내외로 제한된다. 클러스터 설정 후 도커 컨테이너가 일상적 실행을 수행할 때 최대 200개의 피어 노드를 사용하더라도 Fig. 12와 같이 총 CPU 활용률이 40% 미만임을 확인할 수 있다.

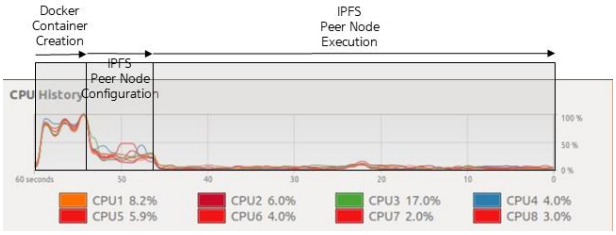



Fig. 11. Number of Peer Nodes: 40

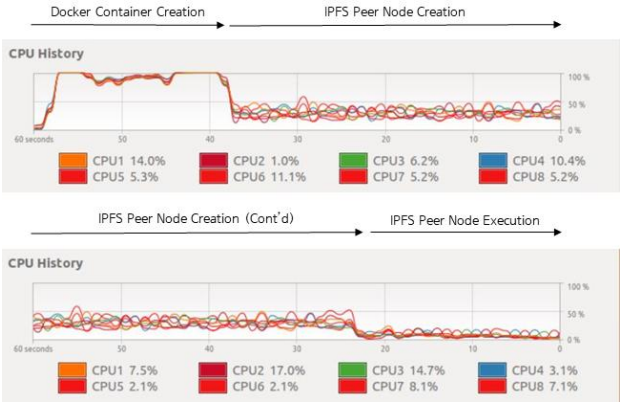



Fig. 12. Number of Peer Nodes: 100

실험 결과에 따르면 개발된 통합 성능 평가 프레임워크는 탈중앙 스토리지 구성을 위한 IPFS 피어 노드 생성 시 노드 별로 1~2초 이하가 소요되며, 최대 200개의 피어 노드로 IPFS 클러스터를 수행할 경우 운영 클럭 3.6GHz로 4코어 CPU일 때 CPU 사용률이 40% 이내이며 최대 메모리 사용량이 6GB 이내인 것으로 확인되었다. 이는 리눅스 운영체제의 네트워크 브리지 기능을 사용하여 일반 호스트 컴퓨터에서 수행되는 도커 가상 네트워크를 여러 호스트 컴퓨터에서 동시에 수행할 때 매우 낮은 오버헤드를 갖음으로써 제안된 통합 평가 프레임워크의 확장성을 검증한다.

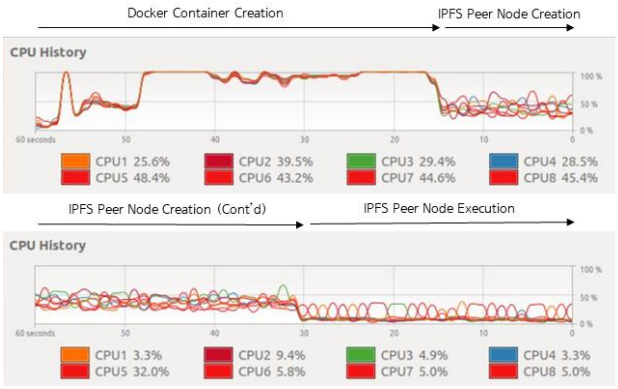



Fig. 13. Number of Peer Nodes: 200

## V. Conclusions and Future Work

본 논문은 오픈소스를 활용하여 탈중앙 스토리지를 구축하였으며, 구축된 스토리지에 멀티미디어 콘텐츠를 스트리밍할 수 있는 소프트웨어 통합하여 가상화 플랫폼에서 상에서 구현하였다. Apple사의 HLS 멀티미디어 스트리밍 서비스의 성능 지표인 KPI를 지원하는 멀티미디어 스트리밍 플레이어의 통합을 통해 재생 품질에 따라 비트 전송률로 인코딩된 미디어 콘텐츠 부분을 재생하는 과정에서의 품질 변화를 감지할 수 있다. 또한, 가상화 플랫폼에서 IPFS 클러스터가 구현되기 때문에 가상 네트워크의 구성을 변경함으로써 다양한 네트워크 대역폭에 대한 서비스의 품질과 성능을 측정할 수 있다.

실험을 통해 제안된 통합 평가 프레임워크를 단일 호스트 컴퓨터 상에서 구축하는 데 필요한 시간이 IPFS 피어 노드당 1~2초 이내임을 확인하였으며, CPU와 메모리 등 시스템 수준의 오버헤드가 매우 낮기 때문에 다양한 피어 노드 구성과 다양한 피어 노드 수 뿐이 아니라 다양한 워크로드에 대한 성능 평가가 가능하다.

본 논문의 향후 연구로는 탈중앙 스토리지와 블록체인 기술을 결합하여 멀티미디어 스트리밍 성능평가를 통해 P2P 네트워크 기반의 상용 스트리밍 서비스에서의 각 참여 단말기의 네트워크에 대한 리소스 기여도 평가로 확장될 수 있다[28].

본 논문이 대상으로 한 P2P (Peer-to-Peer) 방식의 탈중앙 스토리지는 개별 스트리밍 단말기의 네트워크 대역폭과 컴퓨팅 파워 및 저장 공간을 활용하여 이론상으로 무한의 확장성을 갖도록 한다. 반면 P2P 방식의 스트리밍 서비스를 이용하는 사용자는 스트리밍 서비스를 이용하는데 비용을 지불하면서 동시에 사용자 단말기의 자원이 다른 사용자의 서비스를 위해 사용될 수 있다. 즉, P2P 네트워크가 스트리밍 서비스 기술의 확정성의 한계를 극복하였지만 P2P 네트워크에 참여하는 사용자 단말기의 자원이 사용되는 부분에 대한 적절한 비용에 대한 보상이 이루어지지 않는 다른 문제점이 발생할 수 있다[29-30].

블록체인 기술은 근본적으로 P2P 네트워크 기반의 소프트웨어 구조를 갖으며 암호화폐 토큰과 연동이 될 경우 사용자 단말기의 네트워크 대역폭과 컴퓨팅 파워 및 저장공간 사용에 따른 토큰을 지급함으로써 P2P 네트워크에 참여하는 사용자에 대한 보상 문제를 해결할 수 있을 것으로 기대된다. 그러나 현재까지의 블록체인 기술은 사용자 단말기의 자원의 활용과 그에 따른 스트리밍 서비스 품질에 대한 평가 기술이 미흡하여 음원 사용에 따른 라이센스비의 결재 등에 사용되고 있는 수준에 머물러 있어 본 논문의 탈중앙 스토리지 기반 멀티미디어 스트리밍 성능 평가를 위한 프레임워크에 블록체인 기술을 통합하게 된다면 P2P 네트워크 기반 스트리밍 서비스에서의 사용자 리소스 기여도를 평가하는데 적용될 수 있을 것으로 기대된다.

## ACKNOWLEDGEMENT

This research was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by Ministry of Culture, Sports, and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2019 (R2018030393, Development of Blockchain-based Web Content Creation and Distribution Platform Technology).

## REFERENCES

[1] HTTP Live Streaming: Apple Developer Documentation, *https://developer.apple.com/documentation/*

[2] S. Hesse, JavaScript HLS client using Media Source Extension, *https://github.com/video-dev/hls.js*

[3] James Turnbull, "*The Docker Book*", 2017.

[4] Go Implementation of IPFS: the InterPlanetary FileSystem, *https://hub.docker.com/r/ipfs/go-ipfs/*

[5] Go Open Source Programming Language, *https://golang.org/*

[6] S. Mahmoudi, M. Belarbi, A. Mohammed, M. Said, G. Belalem, and P. Manneback, "*Multimedia Processing Using Deep Learning*

*Technologies, High-Performance Computing Cloud Resources, and Big Data Volumes. Concurrency and Computation: Practice and Experience*", 2020. DOI: 10.1002/cpe.5699

[7] S. Arsh, A. Bhatt, and P. Kumar, "Distributed image processing using Hadoop and HIPI," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2673-2676, 2016. DOI: 10.1109/ICACCI.2016.7732463

[8] Raghavendra Kune, Pramod Kumar Konugurthi, Arun Agarwal, Raghavendra Rao Chillarige, and Rajkumar Buyya, "XHAMI - extended HDFS and MapReduce interface for Big Data image processing applications in cloud computing environments," Softw. Pract. Exper. Vol. 47, No. 3, 2017. DOI: 10.1002/spe.2425

[9] A. Yassine, A. A. N. Shirehjini and S. Shirmohammadi, "Bandwidth On-demand for Multimedia Big Data Transfer across Geo-Distributed Cloud Data Centers," IEEE Transactions on Cloud Computing, DOI: 10.1109/TCC.2016.2617369

[10] Y. Chen, H. Li, K. Li and J. Zhang, "An improved P2P file system scheme based on IPFS and Blockchain," 2017 IEEE International Conference on Big Data (Big Data), pp. 2652-2657, 2017. DOI: 10.1109/BigData.2017.8258226

[11] B. Confais, A. Lebre and B. Parrein, "An Object Store Service for a Fog/Edge Computing Infrastructure Based on IPFS and a Scale-Out NAS," 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), pp. 41-50, 2017. DOI: 10.1109/ICFEC.2017.13

[12] B. Confais, A. Lebre and B. Parrein, "Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures," 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 294-301, 2016. DOI: 10.1109/CloudCom.2016.0055

[13] S. A. Weil, A. W. Leung, S. A. Brandt and C. Maltzahn, "RADOS: A Scalable Reliable Storage Service for Petabyte-scale Storage Clusters," Proceedings of the 2nd International Workshop on Petascale Data Storage: Held in Conjunction with Supercomputing '07, 2007. DOI: 10.1145/1374596.1374606

[14] A. Lakshman and P. Malik, "Cassandra: A Decentralized Structured Storage System," SIGOPS Operating Systems Review, vol. 44, no. 2, pp. 35-40, Apr. 2010. DOI: 10.1145/1773912.1773922

[15] J. Benet, "IPFS - Content Addressed Versioned P2P File System," Protocol Labs Inc. Tech. Rep., 2014.

[16] B. Zhang, A. Iosup, J. Pouwelse and D. Epema, "Identifying, analyzing, and modeling flashcrowds in BitTorrent," 2011 IEEE International Conference on Peer-to-Peer Computing, pp. 240-249, 2011. DOI: 10.1109/P2P.2011.6038742

[17] Z. Wang, X. Dong, Y. Li, L. Fang and P. Chen, "IoT Security Model and Performance Evaluation: A Blockchain Approach," 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), pp. 260-264, 2018. DOI: 10.1109/ICNIDC.2018.8525716.

[18] A. Ahmad, A. Floris and L. Atzori, "Timber: An SDN-Based Emulation Platform for Experimental Research on Video Streaming," IEEE Journal on Selected Areas in Communications, Vol. 38, No. 7, pp. 1374-1387, 2020. DOI: 10.1109/JSAC.2020.2999683

[19] Narayan Prusty, "*Building Blockchain Projects*," Packt Publishing, 2017.

[20] InterPlanetary TestBed, *https://github.com/ipfs/iptb*

[21] IPTB Plugins for IPFS, *https://github.com/ipfs/iptb-plugins*

[22] Tc(8) - Linux man page, *https://linux.die.net/man/8/tc*

[23] S. Park, H. Choi, and I. Kim, "Open Source Based Web Tool for Multi-Bitrate HTTP Live Streaming Performance Evalution," The 5th International Conference on Next Generation Computing (ICNGC), 2019.

[24] Mike Bostock, D3.js: Data-Driven Documents, *https://d3js.org/*

[25] Plunker, Zoomable Circle Packing, *http://plnkr.co/edit/Gel6gWwDOVwT8miuhfD5?p=preview*

[26] W. Felter, A. Ferreira, R. Rajamony and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 171-172, 2015. DOI: 10.1109/ISPASS.2015.7095802

[27] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu and W. Zhou, "A Comparative Study of Containers and Virtual Machines in Big Data Environment," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 178-185, 2018. DOI: 10.1109/CLOUD.2018.00030

[28] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung and M. Song, "Distributed Resource Allocation in Blockchain-Based Video Streaming Systems With Mobile Edge Computing," in IEEE Transactions on Wireless Communications, Vol. 18, No. 1, pp. 695-708, 2019. DOI: 10.1109/TWC.2018.2885266

[29] N. Barman, G. C. Deepak and M. G. Martini, "Blockchain for Video Streaming: Opportunities, Challenges, and Open Issues," Computer, Vol. 53, No. 7, pp. 45-56, 2020. DOI: 10.1109/MC.2020.2989051

[30] K. Nandakumar, N. Ratha, S. Pankanti, A. Pentland and M. Herlihy, "Blockchain: From Technology to Marketplaces, Computer, Vol. 53, No. 07, pp. 14-18, 2020. DOI: 10.1109/MC.2020.2990776

## Authors

Sangsoo Park received the BS degree in Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1998, and the MS and PhD degrees in Computer Science & Engineering from Seoul National University, Seoul, Korea, in 2000 and 2006, respectively. Dr. Park joined the faculty of the Department of Computer Science & Engineering at Ewha Womans University, Seoul, Korea, in 2009. He is currently an Associate Professor in the Department of Computer Science & Engineering, Ewha Womans University. He is interested in real-time embedded systems and system software.