

클라우드 스토리지를 위한 오픈 소스 기반 분산 파일 시스템의 성능 평가

이세호, 김지홍, 엄영익
성균관대학교 정보통신대학
e-mail : {loadic, jjilong, yieom}@ece.skku.ac.kr

Performance Evaluation of Open Source Based Distributed File System for Cloud Storage

Seho Lee and Yong Ik Eom
College of Information and Communication Engineering,
Sungkyunkwan Univ.

요 약

최근 클라우드 컴퓨팅 기술은 기존의 서버, 데스크톱 컴퓨팅 환경을 빠르게 변화시키며, 차세대 인터넷 서비스의 핵심 분야로 부각되고 있다. 클라우드 컴퓨팅 기술 중 특히 저비용, 안정성, 확장성, 무결성 그리고 보안성을 가지고 있는 클라우드 스토리지 서비스가 각광 받고 있다. 이에 본 논문은 클라우드 스토리지 기반 기술인 분산 파일 시스템에 관해서 살펴보고, 오픈소스 기반의 분산 파일 시스템인 MooseFS, XtremFS, GlusterFS, Ceph 등을 이용하여 시스템 구축 및 성능 측정을 수행 하였다. 수행 결과 Postmark에서는 GlusterFS, MD5SUM에서는 XtremFS가 가장 좋은 성능을 보여주었다.

1. 서론

클라우드 컴퓨팅(Cloud Computing)은 서버, 네트워크, 스토리지, 운영체제, 응용프로그램 등의 IT 자원을 모아 서비스 형태로 제공하는 컴퓨팅 스타일이다. 클라우드 컴퓨팅 기술은 기존의 서버, 데스크톱 컴퓨팅 환경을 빠르게 변화시키며, 차세대 인터넷 서비스의 핵심 분야로 부각되고 있다[1]. 따라서 구글, IBM, Amazon 등 해외의 IT사업자들은 대규모의 투자와 이미 보유한 기반 기술들을 활용하여 다양한 클라우드 서비스 및 플랫폼 기술 개발에 앞장서고 있다. 국내의 경우 삼성, KT, SK 등 대기업에서 많은 투자와 플랫폼 개발을 위해 노력 하고 있다.

이런 성장은 대용량 데이터 사용량 증가로 인한 비용증가 문제, 스토리지 공간 부족이라는 문제를 가져왔다. 위의 문제를 해결하기위해 클라우드 스토리지 서비스에 분산 파일 시스템 기술이 도입되었다[2]. 분산 파일 시스템은 저비용, 공간의 확장성 그리고 시스템의 안정성 등을 보장하여 주기 때문에 클라우드 스토리지 서비스에 적합하다. 현재 수많은 분산 파일 시스템이 존재하지만 실제로 클라우드 스토리지에 사용되는 시스템은 많지 않은 상황이다[3]. 이런 상황을 해결하기 위해 오픈 소스기반의 분산 파일 시스템인 MooseFS, XtremFS, GlusterFS, Ceph 등을 이용하여 실험환경을 구축하였다. 이를 통해 소요시간, 메모리와 하드 사용량을 측정 하였다. 이때 Postmark에서는 GlusterFS가 MD5SUM에서는 XtremFS가 가장

좋은 성능을 보여주었다.

본 논문의 구성은 다음과 같다. 2장에서 일반적인 분산 파일 시스템과 성능 평가에 사용 된 분산 파일 시스템에 대해 살펴본다. 3장에서는 실험 환경에 대해 소개하고, 각각의 분산 파일 시스템을 비교를 통해 평가한다. 마지막으로 4장에서는 결론 및 향후 연구에 대하여 설명 한다.

2. 분산 파일 시스템

분산 파일 시스템은 컴퓨터 네트워크를 통해서 여러 사용자에게 동일한 파일을 접근 할 수 있는 공간을 제공하는 시스템이다. 또한, 여러 사용자가 다수의 서버에 저장된 데이터를 읽고, 쓰면서 데이터를 공유할 수 있는 기능을 제공한다[4]. 클라우드 스토리지를 위한 분산 파일 시스템은 저비용 스토리지 서버를 이용하여 스토리지 비용을 최소화 하고, 장애에 대한 효율적인 통제를 제공한다. 추가적으로 동시에 같은 데이터에 접근하는 경우를 해결하기 위한 동시접근제어 기술을 제공한다.

2.1 MooseFS

MooseFS[5]는 폴트 톨러런스(fault-tolerance)를 제공하는 네트워크 분산 파일 시스템이다. 이 시스템은 몇 개의 퍼져있는 서버의 데이터를 유저에게 하나의 자원공간으로 보이게 지원한다.

MooseFS는 메타데이터서버, 청크서버, 메타로거, 클라이언트로 구성되어 있다. 메타데이터서버는 전체 파일 시스템을 관리하고, 모든 파일의 메타정보를 메타로거에 기록

† 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0025971)

한다. 청크서버는 데이터를 저장하고, 선택적으로 복제된 파일을 나누어 저장한다. 청크서버의 구성은 제한 없이 확장이 가능하다. 메타로거는 메타데이터서버로부터의 메타정보를 주기적으로 저장한다. 클라이언트는 퓨즈(FUSE)를 통해 유저영역에 마운트하고, 메타데이터서버를 통해 위치 정보를 획득한 뒤 청크서버에서 데이터를 가져온다.

MooseFS는 안전성과 확장성을 위한 특징을 가진다. 첫째, 데이터의 복사 본을 여러 곳에 나누어 저장하여 높은 안정성을 제공한다. 둘째, 새로운 컴퓨터나 디스크를 부착하게 되면 쉽게 저장 공간을 확장 할 수 있다. 마지막으로 파일을 쓰거나 접근하고 있는 동안에도 규칙적으로 스냅샷(Snapshot)하여 시스템 구성을 유지한다.

2.2 XtreamFS

XtreamFS[6]는 클라우드를 위한 객체기반의 분산 파일 시스템이다. 이 시스템은 로컬영역에 분산된 파일 서버를 가지고 있는 것이 아닌, 광역의 공간에 서버가 분산되어 있다는 가정으로 설계 되었다.

XtreamFS는 MRC(Metadata and Replica Catalog), DIR(Directory Service), OSD(Object Storage Device) 서버와 클라이언트로 구성되어 있다. 먼저 디렉터리 트리 정보와 파일 이름과 같은 메타데이터를 저장하고 있는 MRC가 있다. DIR은 XtreamFS의 모든 서비스를 위한 정보를 가지고 있다. MRC는 스토리지 서버들의 위치를 DIR을 통해 확인한다. OSD는 파일의 객체를 임의적으로 저장한다. 클라이언트는 퓨즈 파일 시스템을 사용하여 마운트하고, OSD에 직접 접근하여 파일을 읽고, 쓴다.

XtreamFS는 안정성과 확장성 그리고 장애를 효율적으로 극복하는 특징을 가지고 있다. 첫째, 폴트 톨러런스를 위해 객체를 복제하고, 메타정보를 캐쉬한다. 둘째, 클라이언트는 리눅스와 윈도우즈 환경 모두를 지원하는 이점이 있다. 마지막으로는 파일을 교차적으로 복제하여 자동적인 장애극복(Failover)이 가능하다.

2.3 Ceph

Ceph[7]는 용량을 페타바이트 수준으로 손쉽게 확장할 수 있는 확장성, 가변적인 워크로드를 효과적으로 처리할 수 있는 성능, 그리고 강력한 신뢰성을 목표로 설계된 분산 파일 시스템이다.

Ceph는 MDS(Metadata Server), OSD, MON(Monitors Cluster), 클라이언트로 구성 되어 있다. MDS는 데이터의 위치와 새 데이터를 저장할 위치를 관리하고, 분산 메타데이터를 캐시 및 동기화하는 작업을 한다. OSD는 데이터와 메타데이터를 객체로 저장하고, 블록 할당과 같은 로우 레벨의 파일 스토리지 관리를 한다. MON는 OSD가 고장 나거나 새 장치가 추가되면 MON는 이를 감지하고 올바른 클러스터 맵을 관리한다. 클라이언트는 리눅스 커널의

파일 시스템 소스 자체 내에서 구현되어있다. 클라이언트는 메타데이터서버에서 메타정보를 얻어오고, OSD에 직접 파일을 읽고, 쓰는 작업을 수행한다.

Ceph는 기존의 파일 시스템과는 차별적인 확장성, 회복 능력, 자원관리의 특징을 가진다. 첫째, 이 파일 시스템은 간단하게 OSD를 추가하고, 이를 유연하게 확장할 수 있는 기능을 가지고 있다. 둘째, 서버 구성 요소들은 각각의 클러스터를 이루고 있다. 모든 데이터가 OSD클러스터들 사이에서 복제되어 강한 안정성을 제공하고, 메타데이터 서버 클러스터는 병렬 처리를 통한 빠른 복구 능력을 제공한다. 마지막으로 메타데이터 서버는 현재 작업량에 따라 OSD간의 저장 공간 조절과 효율적인 서버 자원 사용량을 조절 할 수 있다.

2.4 GlusterFS

GlusterFS[8]는 페타바이트 규모의 크기와 수천 개의 클러스터를 다룰 수 있는 분산 파일 시스템이다. Ethernet 및 Infiniband 네트워크에서 NAS용 파일시스템으로 개발되었다. 이 시스템은 클러스터로 모여 디스크와 메모리 자원을 함께 구성하고, 하나의 글로벌 네임스페이스 안에서 데이터들이 관리된다. GlusterFS는 모듈형 구조로 관리자가 사용자의 요구에 맞게 모듈을 썬아 갈 수 있게 한다.

GlusterFS는 서버와 클라이언트 2가지 요소로 구성되어 있다. 서버는 스토리지 브릭(bricks)으로 구성 되어 있고, 로컬 파일 시스템이 하나의 볼륨으로 사용된다. 클라이언트는 서버와 TCP/IP, InfiniBand 또는 SDP(Session Description Protocol)등의 프로토콜을 선택하여 연결한다. 퓨즈 파일 시스템을 사용하여 마운트 하거나 libglusterfs 라이브러리를 사용하여 직접 서버에 접근하게 된다.

GlusterFS는 파일 기반의 미러링, 복제 기능을 제공하고, 분산된 메타데이터 모델을 사용한다. 마지막으로 볼륨들의 추가, 삭제 등이 자유롭게 이루어진다.

3. 분산 파일 시스템 성능 평가

3.1 실험 환경

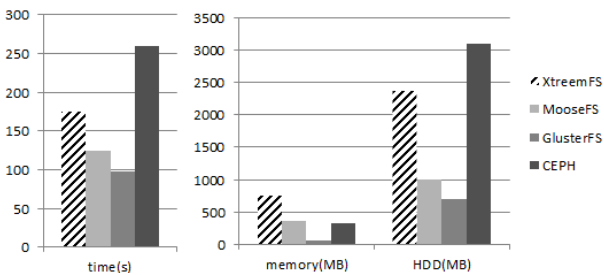
실험 환경은 표1과 같이 4대의 PC를 사용하여 실험 하였다. 메타 정보와 데이터를 관리하는 메타데이터용 서버 PC 1대, 파일을 저장하는 OSD 서버 PC 2대 그리고 분산 파일 시스템을 마운트 하여 실험을 진행한 클라이언트 PC 1대를 사용하여 실험 하였다. 이때 GlusterFS는 따로 메타데이터 관리 서버가 존재하지 않아 OSD 서버 2대로만 구성하였다. 성능을 비교할 분산 파일 시스템은 오픈소스로 제공되는 분산 파일 시스템인 MooseFS, XtreamFS, Ceph 그리고 GlusterFS를 이용하여 시스템을 구축하고, 클라이언트에 각 파일 시스템을 마운트 한 뒤 Postmark Bench, MD5SUM을 수행 하면서 성능을 측정하였다. 측정 요소는 클라이언트에서의 처리시간, OSD에서의 메모리와 하드 사용량이다.

<표 1> 실험 환경

| 구성 요소 | 시스템 사양 |
|---------|---|
| 메타 서버 | Intel Quad CPU Q6600 2.4GHz, 2GB RAM, 500MB HDD |
| OSD 서버1 | Intel Quad CPU Q8300 2.5GHz, 4GB RAM, 1TB HDD |
| OSD 서버2 | Intel Quad CPU Q6600 2.4GHz, 3GB RAM, 1TB HDD |
| 클라이언트 | Intel Core i5 CPU M560 2.67GHz, 4GB RAM, 650GB HDD |

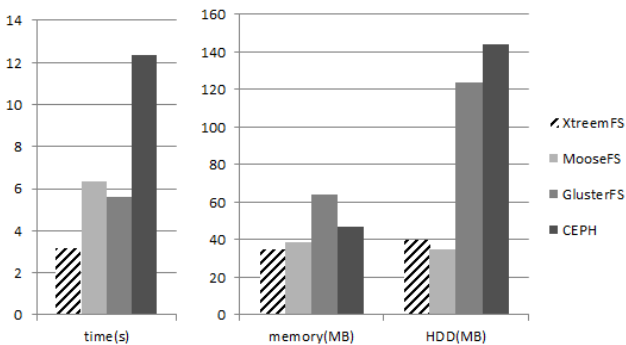
3.2 실험 결과 및 분석

그림 1은 Postmark를 이용한 성능 측정 결과이다. Postmark는 주어진 범위 내의 크기를 가지는 다량의 텍스트 파일들을 생성한 후, 랜덤하게 파일을 골라 그 파일에 대해 삭제, 읽기, 추가 등의 작업을 수행한다. Postmark를 수행한 결과 GlusterFS는 수행시간이 97초로 가장 빠르고, 메모리와 하드 사용량 또한 63MB, 709MB로 가장 적다. 반면 Ceph는 260초로 수행시간이 가장 길고, 메모리와 하드 사용량 또한 327MB, 3GB로 하드 사용량이 가장 크다.



(그림 1) Poskmark 수행결과

그림 2는 MD5SUM을 이용한 성능 측정 결과이다. MD5SUM은 주어진 파일을 가지고 128-bit MD5 해쉬 값을 계산하는 프로그램이다. MD5SUM을 수행한 결과 XtremFS는 3초로 수행시간이 가장 짧고, 메모리와 하드 사용량 또한 34MB, 40MB로 적다. 반면 Ceph는 12초로 가장 긴 수행시간을 보이고, 하드 사용량은 144MB로 가장 많다. 반면 메모리사용량은 46MB로 적게 나왔다.



(그림 2) MD5SUM 수행결과

두 실험을 통해 성능 상으로는 Ceph가 가장 좋지 않은 결과를 보여주었다. 평균적으로 MooseFS의 성능이 가장 고르게 나온 것을 확인 할 수 있었다. Ceph는 안정성을 위한 복제 정책 등을 위해 추가적으로 사용되는 오버헤드가 크게 작용하여 수행시간과 메모리와 하드 사용량이 다른 파일 시스템에 비해 크게나왔다. XtremFS는 CPU를 많이 사용하는 MD5SUM에서 빠른 처리 속도를 보여 주었다. GlusterFS는 작은 파일 단위를 여러 번 처리하는 Postmark에서 빠른 성능과 적은 메모리, 하드 사용량을 보여주었다. 이것은 각각의 분산 파일 시스템이 가지고 있는 특성들과 구성의 차이를 가지고 있어 각각의 실험 환경에 따라 다른 성능을 보이는 것을 확인 할 수 있었다.

4. 결론

본 논문에서는 최근 클라우드 스토리지에서 사용되고 있는 기술인 분산 파일 시스템 관하여 연구하였다[9]. 최근 오픈 소스 기반으로 연구되고 있는 MooseFS, XtremFS, GlusterFS와 Ceph를 사용하여 시스템을 구축하고 실험을 하였다. 분산 파일 시스템들은 각각의 실험 환경에서 다른 성능을 보여 주었다. 이는 각 분산 파일 시스템의 구성방식과 복제 정책 그리고 폴트 톨러런스를 위한 정책 등의 차이를 통해 발생하는 것임을 알 수 있다. 클라우드 스토리지를 위한 분산 파일 시스템을 선택 시에 위의 정책이나 구성방식 등이 주요한 요소로 작용할 것이다. 향후에는 여러 환경에서 최적의 성능을 제공할 수 있는 분산 파일 시스템을 위하여 동기화, 중복 제거 및 고속 캐싱 기술 등을 적용하여 분산 파일 시스템을 구현하여 볼 계획이다.

참고문헌

[1] M. D. Dikaiakos, D. Katarasos, P. Mehra, G. Pallis, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research," IEEE Internet Computing, 2009.

[2] S. Ghemawat, H. Gobioff, and S. Leung, "The Google File System," SOSP '03 Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, 2003.

[3] D. Fesehaye, R. Malik, and K. Nahrstedt, "A Scalable Distributed File System for Cloud Computing," Technical Report, University of Illinois at Urbana Champaign (UIUC), 2010.

[4] E. Levy and A. Silberschatz, "Distributed File Systems: Concepts and Examples," ACM Computing Surveys, 1990.

[5] MooseFS, <http://www.moosefs.org>

[6] XtremFS, <http://xtremfs.com>

[7] Ceph, <http://Ceph.newdream.net>

[8] Glusterfs, <http://www.gluster.org>

[9] H. Feel and M. Khafagy, "Search Content via Cloud Storage System," IJCSI, 2011.