

클라우드 환경의 오픈소스 기반 분산 파일 시스템 분석 및 비교†

김건우, 김지홍, 엄영익
성균관대학교 정보통신대학

e-mail:{kkw0528, jjilong, yieom}@ece.skku.ac.kr

An Analysis and Comparison of Open Source Based Distributed File System for Cloud Environment

Keonwoo Kim, Jeehong Kim and Young Ik Eom

College of Information and Communication Eng., Sungkyunkwan Univ.

요 약

클라우드 컴퓨팅이 많은 관심을 받고 발전하면서 여러 IT선도업체에서 클라우드 컴퓨팅 기술 개발에 많은 투자를 하고 있다. 이러한 클라우드 컴퓨팅 환경에서는 대부분의 데이터를 서버에 저장한다. 이러한 이유로 클라우드 환경에서 사용되는 파일 시스템은 기존의 파일 시스템 보다 많은 데이터를 저장하게 된다. 이에 따라 많은 데이터를 처리하기 위해서 클라우드 환경에서는 분산 파일 시스템을 활용하고 있다. 또한 분산 파일 시스템은 네트워크상의 여러 스토리지 서버에 데이터가 분산되어 저장되기 때문에 데이터의 관리뿐만 아니라 성능, 장애 허용, 보안 등의 요구사항을 만족해야 한다. 본 논문에서는 XtreamFS, Ceph, GlusterFS, MooseFS 등의 분산 파일 시스템들을 기능적 측면에서 살펴보고, 각 분산 파일 시스템을 본 논문에서 제안하는 기능적 평가요소 측면에서 비교하고 평가한다.

1. 서론

최근 클라우드 컴퓨팅이 각광을 받고 있다. 또한 2009년부터 2012년까지 가트너가 선정 한 10대 전략 기술[4]에 클라우드 컴퓨팅이 선정되었다. 이러한 클라우드 컴퓨팅 환경에서는 클라이언트의 대부분의 데이터를 서버에 저장한다[1]. 따라서 대용량의 데이터를 효율적으로 저장하고 관리하기 위한 비용이 증가 하게 된다. 이에 따라 대용량의 데이터를 다수의 서버에 분산시켜서 저장하고 관리 할 수 있는 분산 파일 시스템은 클라우드 컴퓨팅에서 중요한 부분을 차지하게 되었다. 이러한 분산 파일 시스템은 데이터를 네트워크상의 여러 스토리지 서버에 분산시켜 저장한다[1]. 그리고 분산된 데이터를 접근할 때에는 데이터를 마치 로컬에 있는 데이터처럼 접근 할 수 있는 파일 시스템이다[1]. 현재 여러 오픈 소스 기반의 분산 파일 시스템이 제공되고 있으며, 대표적인 오픈 소스 기반의 분산 파일 시스템으로 GFS[5], HDFS[7], XtreamFS[2,9], Ceph[3,10], GlusterFS[8,11], MooseFS[7,1] 등이 있다.

클라우드 환경을 구축함에 있어서 다양한 분산 파일 시스템 중, 적합한 분산 파일 시스템을 선택하는 것은 중요하다. 현재 분산 파일 시스템의 비교는 데이터 처리량이나 지연시간 등의 성능위주의 비교[2,8]와 장애 허용 여부, 보

안 등의 기능적 측면에서의 비교[6,7]가 수행 되고 있다. 분산 파일 시스템의 성능은 분산 파일 시스템을 선택함에 있어서 중요한 척도이다. 하지만 분산 파일 시스템은 네트워크상에 스토리지 서버들이 분산되어 있기 때문에, 기존의 로컬 파일 시스템보다 고려해야 할 사항이 많다[1,6,7]. 이에 따라 분산 파일 시스템의 기능적인 요소에 대한 평가도 중요하다. 분산 파일 시스템의 고려해야 할 기능적인 평가요소로는 장애 허용, 데이터 무결성, 확장 용이성, 동시성 제어, 보안 및 접근 제어 등이 있다. 이러한 기능적인 평가요소들은 성능적인 요소와 더불어 분산 파일 시스템을 선택함에 있어서 중요한 척도가 될 수 있다.

본 논문에서는 XtreamFS, Ceph, GlusterFS, MooseFS에 대한 기능적인 요소의 비교 평가를 실시한다. 2장에서 각 분산 파일 시스템에 대해서 알아보고, 3장에서 본 논문이 제안하는 기능적인 비교요소에 대해서 알아본다. 4장에서는 본 논문이 제안하는 기능적인 요소에 대한 각 분산 파일 시스템의 비교 및 평가를 수행하고, 마지막으로 5장에서는 결론을 기술 한 후 논문을 마친다.

2. 관련연구

기존의 분산 파일 시스템의 비교는 수행시간, 처리량, 지연시간 등의 성능 위주의 비교 및 평가와 장애 허용 여부, 보안 등의 기능적 측면에 대한 비교가 수행 되어 왔다 [2,6,7,8]. John H. Howard는 AndrewFS와 NFS를 수행시

† 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2011-0020520)

간, CPU 사용 효율, 디스크 사용 효율 등의 실험을 수행하고, 이에 대한 비교 평가를 했다[2]. Gideon Juve는 NFS, GlusterFS, PVFS2, S3, 로컬 파일 시스템 등의 파일시스템에 대해서 과학적 애플리케이션의 수행시간과 워크 플로우 비용에 대한 실험을 수행하고, 이에 대한 비교 평가를 했다[8]. 분산 파일 시스템의 기능적인 요소에 따른 비교를 수행한 연구로는 [6,7]이 있다. 분산 파일 시스템은 데이터가 네트워크상의 서버에 분산되어 저장되기 때문에 서버의 장애나 데이터의 손실 등의 문제가 생길 수 있다. 이에 따라 장애 허용 여부, 데이터 무결성, 보안 등의 기능적인 요소가 중요하다. 이러한 이유로 성능적인 요소만으로 어떤 분산 파일 시스템이 좋다고 판단하기에는 부족함이 있다.

2.1 분산 파일 시스템 기술

2.1.1 XtremFS

XtremFS[9]는 객체기반의 POSIX 분산 파일 시스템이다. 파일을 고정된 크기의 객체 단위로 나누어 스토리지 서버에 분산시켜 저장한다. 리눅스, 솔라리스, 맥 OS X, 윈도우즈 등의 운영체제를 지원한다. 객체를 여러 서버에 복제해서 저장하므로 하나의 서버에서 장애가 발생해도 복구할 수 있으며, 또한 동시성 제어와 RAID0 스트라이핑을 지원한다. 또한 보안을 위해서 X.509를 이용한 인증과 SSL을 지원한다. XtremFS는 DIR(디렉토리 서비스), MRC(메타데이터 서버), OSD로 구성된다. DIR서버는 파일시스템의 각 서버에 대한 관리를 하는 역할을 한다. 또한 MRC서버에서 데이터 객체가 어느 서버에 위치해 있는지 알아내기 위해서 DIR 서버를 사용한다. MRC서버는 파일에 대한 메타데이터에 대한 정보를 저장하며, OSD는 객체 데이터를 저장하고 클라이언트는 OSD에서 데이터를 읽고 쓰게 된다. XtremFS는 설치가 간편하고, 서버의 확장이 용이하다는 장점이 있다.

2.1.2 Ceph

Ceph[10]는 객체기반의 POSIX 분산 파일 시스템이며, 리눅스 운영체제를 지원한다. 리눅스 커널 2.6.34부터 커널에 포함되었다. XtremFS와 마찬가지로 RAID0 스트라이핑을 지원한다. Ceph는 클러스터 모니터, 메타데이터 서버, OSD로 구성된다. 클러스터 모니터는 클러스터 노드들의 장애를 감시하는 역할을 하고, 메타데이터 서버는 파일에 대한 메타데이터와 inode 등의 정보를 저장한다. 그리고 OSD에는 실제 데이터가 저장된다. 또한 데이터를 각 OSD에 분산 저장하는 CRUSH와 부하 조절을 위한 RADOS를 이용한다.

2.1.3 GlusterFS

GlusterFS[11]는 파일기반의 POSIX 분산 파일 시스템이다. 리눅스, 맥 OS X, NetBSD, FreeBSD, 오픈 솔라리스 등의 운영체제를 지원한다. 서버들은 Infiniband RDMA 또는 TCP/IP방식을 이용하여 연결되며, 파일단위로 복제 및 스트라이핑을 수행한다. GlusterFS와 다른 분산 파일

시스템의 가장 큰 차이점은 메타데이터 서버가 별도로 존재하지 않고, 각 데이터 서버들에 분산되어 있다는 점이다. 따라서 메타데이터 서버에 장애가 발생했을 때 생기는 문제점을 해결할 수 있으며, 큰 규모의 시스템에서 생길 수 있는 메타데이터 서버에 의한 병목현상을 방지할 수 있다.

2.1.4 MooseFS

MooseFS[12]는 객체기반의 POSIX 분산 파일 시스템이다. 리눅스, FreeBSD, 솔라리스, 오픈 솔라리스, 맥 OS X 등의 운영체제를 지원한다. 동적으로 새로운 스토리지 서버를 추가할 수 있으며, 데이터 복제본을 각 스토리지 서버에 저장할 수 있다. 따라서 한 스토리지 서버에서 장애가 발생하여도 정상적으로 서비스를 수행할 수 있다. MooseFS는 마스터 서버, 체크 서버, 메타로거 서버로 구성된다. 마스터 서버는 전체 파일시스템을 관리하며, 각 파일의 메타데이터 등의 정보를 저장한다. 체크 서버는 실제 파일에 대한 데이터를 저장하며, 다른 체크 서버들과 동기화 한다. 메타로거 서버는 마스터 서버의 메타데이터에 대한 로그를 저장하며, 주기적으로 마스터 서버에 저장된 메타데이터를 저장한다. 마스터 서버에서 장애가 발생하면 메타로거 서버가 마스터 서버의 역할을 하게 된다.

3. 제안 평가요소

3.1 장애 허용(Fault tolerance)

분산 파일 시스템은 구성된 서버 중 한 서버에서 장애가 발생되어도 정상적으로 서비스를 제공해야 한다. 이를 위해서 분산 파일 시스템은 복제 데이터를 각 서버에 나누어 저장한다. 이에 따라 한 서버에서 장애가 발생할 경우 다른 서버의 데이터에 접근해서 정상적으로 서비스를 제공할 수 있어야 한다[6].

3.2 데이터 무결성(Data integrity)

분산 파일 시스템에서 데이터를 저장 시에 장애나 알 수 없는 문제로 잘못된 데이터가 저장될 경우가 생기게 된다. 이 경우 분산 파일 시스템은 데이터의 무결성을 위해서 오류를 검출하고 복구할 수 있어야 한다[1,6].

3.3 확장성(Scalability)

분산 파일 시스템은 시스템의 확장이 용이해야 한다. 새로운 스토리지 서버의 추가가 간단해야 하며, 서버의 추가를 위해서 서비스가 중단되는 일이 없어야 한다[1].

3.4 동시성 제어(Concurrency control)

분산 파일 시스템에서는 동시에 다수의 사용자가 접근 가능하다. 그러므로 같은 데이터에 대해서 동시에 접근하는 상황에 대한 제어가 필요하다[1,6].

3.5 보안 및 접근제어(Security and Access control)

분산 파일 시스템에서는 관리자에 의해 사용자의 데이터가 유출될 수 있으며, 악의적인 해킹으로 인한 해킹이 우려된다. 또한 다수의 사용자가 다수의 서버의 데이터에 접근한다. 이러한 이유로 분산 파일 시스템에서의 사용자 인증 및 접근 허용 권한나현수 관리가 중요하다[1,6].

3.6 단일 고장점(Single point of failure)

분산 파일 시스템 상에서 한 지점에 장애가 발생하면 시스템 전체가 동작하지 않는 부분을 말한다. 이와 같은 이유로 분산 파일시스템은 스토리지 서버 이외에도 다수의 메타데이터 서버나 마스터 서버를 구성할 수 있어야 한다[7].

4. 분산 파일 시스템 비교 및 평가

본 장에서는 본 논문에서 제안한 분산 파일 시스템 평가요소를 바탕으로 비교 및 평가를 수행한다. 비교 및 평가할 대상은 XtreamFS, Ceph, GlusterFS, MooseFS이다.

	XtreamFS	Ceph	GlusterFS	MooseFS
파일 시스템 구조	MRC, DIR, OSD 서버	클러스터 모니터, 메타데이터 서버, OSD	메타데이터, OSD가 합쳐진 형태	마스터서버, 체크서버, 메타로거서버
단일 고장점	X	X	X	O
장애 허용	O	O	O	O (마스터서버 장애 시 X)
데이터 무결성	O 데이터복제, 동기화 통해 무결성 유지	O 데이터복제, RADOS 통해 무결성유지	O 데이터복제, 동기화 통해 무결성 유지	O 데이터복제, 동기화 통해 무결성 유지
확장성	O(동적)	O(동적)	O(동적)	O(동적)
동시성 제어	O (close-to-open consistency model)	O	O	O(read)
보안	O (X.509, SSL)	O (ssh)	O (ssh)	O (POSIX ACLs)

<표 1> 분산 파일 시스템 비교

표 1에서 각 분산 파일 시스템에 대해서 기능적인 요소에 대해서 비교를 하였다. 다른 분산 파일시스템과는 달리 GlusterFS는 메타데이터 서버가 각 스토리지 서버에 포함되어 있다. 이는 단일 고장점이나 장애 허용 측면을 고려한 구성이다. 반면 하나의 마스터 서버가 존재하는 MooseFS는 단일고장점이나 장애 허용 측면에서 좋지 않은 결과를 보이고 있다. 또한 저장된 데이터의 규모가 커지게 되면 병목현상이 발생할 가능성이 있다. 데이터 무결성에 대해서는 모든 분산 파일 시스템이 데이터 복제 및 동기화를 통해서 데이터 무결성을 유지하고 있다. 확장성 측면에서도 모든 분산 파일시스템이 스토리지 서버를 동적으로 추가하거나 제거 할 수 있다. 동시성 제어나 보안 및 접근 제어 측면에서는 모든 분산 파일 시스템이 지원하고 있으나 방식에는 다소 차이를 보인다.

5. 결론

본 논문에서는 현재 제공되고 있는 분산 파일 시스템 기술에 대해서 살펴보았다. 그리고 기존의 분산 파일 시스템들의 성능 위주의 비교에서 벗어나 기능적인 측면에서 비교를 수행 하였다. 각 분산 파일 시스템을 대상으로 기능적인 측면에서 비교를 수행한 결과 데이터 무결성, 동시성 제어, 보안등의 요소를 만족시키기 위해 이용하는 기술

적인 방법의 차이를 보인다. 이러한 차이 때문에, 클라우드 환경의 분산 파일 시스템에서는 좋은 성능뿐만 아니라 기능적인 요소 또한 중요하게 고려되어야 한다.

향후 클라우드 환경에서의 스토리지를 효율적으로 이용하기 위한 기술인 데이터 중복제거에 대한 연구를 할 것이다.

참고문헌

- [1] 민영수, 진기성, 김홍연, 김영균, “클라우드 컴퓨팅을 위한 분산 파일 시스템 기술 동향,” 전자통신동향분석, 제24권 제4호, pp.55-68, 2009 8월.
- [2] John H. H., Michael L. K., Sherri G. M., David A. N., Mahadev S., Robert N. S., and Michael J. W., “Scale and Performance in a Distributed File System,” in Proceedings of ACM Transactions on Computer Systems, Volume 6 Issue 1, pp.51-81, February 1988.
- [3] Sage A. W., Scoot A. B., Ethan L. M., Darrell D. E. L., and Carlos M., “Ceph: A Scalable, High-Performance Distributed File System,” in Proceedings of the 7th Symposium on Operating Systems Design and Implementation, pp.307-320, November 2006.
- [4] Gartner, Gartner Identifies the Top 10 Strategic Technologies for 2012, <http://www.gartner.com/>
- [5] Sanjay G., Howard G., and Shun-Tak L., “The Google File System,” in Proceedings of the 19th ACM Symposium on Operating Systems Principles, pp.29-43, October 2003.
- [6] Tran D. T., Subaji M., Eunmi C., Sangbum K., and Pilsung K., “A Taxonomy and Survey on Distributed File Systems,” in Proceedings of Fourth International Conference on Networked Computing and Advanced Information Management, pp.144-149, September 2008.
- [7] Songlin B. and Hao W., “The Performance Study on Several Distributed File Systems,” in Proceedings of 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp.226-229, October 2011.
- [8] Gideon J., Ewa D., Karan V., Gaurang M., Benjamin P. B., Bruce B., and Phil M., “Data Sharing Options for Scientific Workflows on Amazon EC2,” in Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, pp.1-9, November 2010.
- [9] XtreamFS official site, <http://www.xtreamfs.org/>
- [10] Ceph official site, <http://ceph.newdream.net/>
- [11] GlusterFS official site, <http://www.gluster.org/>
- [12] MooseFS official site, <http://www.moosefs.org/>