전성원
NaverLabs(네이버랩스) 수석연구원

클러스터 파일시스템 소개

전성원 NaverLabs(네이버랩스) 수석연구원

I. 클러스터 파일시스템 소개

방송을 위한 영상 데이터가 이제 모두 디지털화되면서 이 디지털 영상 데이터를 저장하고 또 컴퓨터로 처리하는데 필요한 스토리지가 점점 중요한 위치를 차지해 가고 있다. 더욱이 영상 데이터의 크기가 HD, UHD 시대로 바뀌면서 그 용량도 엄청나게 증가하고 있어 필요한 스토리지 용량과 함께 비용도 엄청나게 증가하고 있다. 뿐만 아니라 이런 데이터를 다양한 목적에 따라 다양한 컴퓨터에서 이용해야 하기 때문에 여러 컴퓨터에서 같은 데이터에 대한 접근이 가능한 공유 기능은 꼭 필요한 기능이라 할 수 있다. 이렇게 여러 컴퓨터에서 같은 데이터를 공유하여 사용할 수 있는 기능을 제공하는 파일시스템은 아주 다양한 방식과 종류가 있으며 각각의 특징이 있고, 구축 비용이나 관리 비용도 다양하다. 그러므로 이런 다양한 파일시스템의 각각의 기능과 그 특징을 잘 알고 사용자의 사용 목적에 따라 잘 선택해서 사용한다면 훨씬 효과적으로 사용할 수 있을 것이다.

II. 파일시스템의 분류

여러 컴퓨터에서 하나의 동일한 파일을 공유할 수 있는 파일시스템(File System)을 크게는 클러스터 파일시스템(Cluster File System)이라고 한다. 이 클러스터 파일시스템은 다시 디스크를 공유하는 방식인 공유 디스크 파일시스템(Shared-Disk File System)과 아무것도 공유하지 않고 네트워크를 통하여 모든 것이 처리되는 분산 파일시스템(Distributed File System)의 두 가지로 나누어 진다. 이제 각 방식의 특징과 해당하는 실제 파일시스템의 특징을 설명하도록 하겠다.

1. 공유 디스크 파일시스템

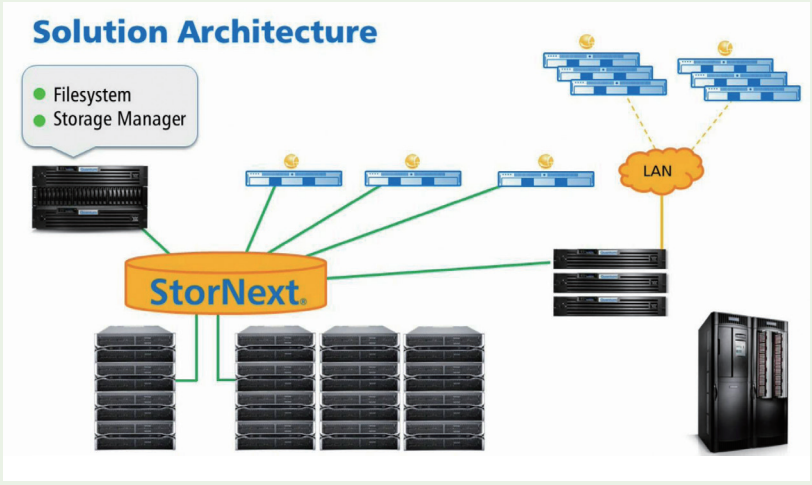
여러 서버에서 파일시스템에 실제 데이터를 저장하고 사용하는 모든 디스크를 공유하고, 모든 컴퓨터에서 각 디스크에 대한 블록 레벨의 직접적인 접근이 가능하다. 일반적으로 이런 디스크 공유 구성을 하기 위하여 고가의 SAN 스토리지를 사용하거나 인피니밴드(Infiniband)와 같은 고속의 전용 네트워크를 사용하여 디스크를 접근하는 경우가 대부분이다. 이 공유 디스크를 하나의 파일시스템으로 묶고 이 파일시스템을 사용하는 컴퓨터에서 마운트(mount)하여 사용하게 된다. 그러면 모든 컴퓨터에서 동일한 파일에 접근할 수 있게 된다. 파일을 사용할 때는 파일의 메타데이터 조회를 통하여 어떤 디스크에 접근해야 하는지 결정한 후에 해당 디스크를 직접 읽거나 쓰기 때문에 로컬 파일시스템과 유사한 성능을 가질 수 있는 장점이 있다. 그러나 이런 파일시스템은 일반적인 로컬 파일시스템과는 달리 여러 서버에서 발생하는 쓰기 작업에 대한 일관성 유지를 위하여 동시성 제어(Concurrency Control)를 하게 된다. 그러므로 이런 작업이 거의 필요 없는 읽기에 비하여 쓰기가 상대적으로 느린 단점이 있다. 그러므로 일부 클라이언트에서만 쓰기를 하고 대다수의 나머지 클라이언트는 읽기를 하는 환경에 적합하다 할 수 있다.

공유 디스크 파일시스템은 파일시스템의 메타데이터 관리나 동시성 제어를 위한 락(lock)을 관리하는 방식에 따라 크게 별도의 메타데이터 제어기를 두어 처리하는 방식과 모든 서버에 분산하여 처리하는 대칭형 분산 방식 두 가지 방식이 있다. 전자의 경우 전체적인 구조가 단순하고 중요 메타데이터의 동기화 문제가 적어 성능적인 장점이 있지만, 모든 메타데이터 처리가 제어기에 집중되면 병목이 될 수 있다. 후자의 경우 성능에 있어서 병목은 없지만, 분산된 메타데이터의 동기화 과정이 전자의 경우에 비하여 느릴 수 있는 단점이 있다.

1) 메타데이터 제어기 방식

SAN 또는 인피니밴드로 공유된 디스크를 모든 클라이언트가 접근 가능하고 파일시스템 전체의 메타데이터를 관리하는 메타데이터 제어기가 따로 있어서 모든 파일시스템의 file 메타데이터 정보와 해당 file의 할당된 디스크 위치 및 블록 정보를 관리한다. 클라이언트는 file에 접근할 때 메타데이터 제어기에서 해당 file의 메타데이터 operation을 통하여 file의 data가 저장되어 있는 디스크와 블록 정보를 받아 직접 공유 디스크를 블록 레벨로 접근하여 data를 읽고 쓰게 된다. File을 처리하는데 있어서 메타데이터 부분의 처리는 로컬 파일시스템에 비하여 차이가 나지만, 실제 data의 접근은 로컬 파일시스템과 거의 비슷한 방식을 사용하게 된다. 일반적으로 이 방식의 파일시스템은 전체 파일에 대한 메타 데이터의 조회가 쉬워 HSM(Hierarchical Storage Management)이나 ILM(Information Life-cycle Management) 시스템과 연동이 수월하다.

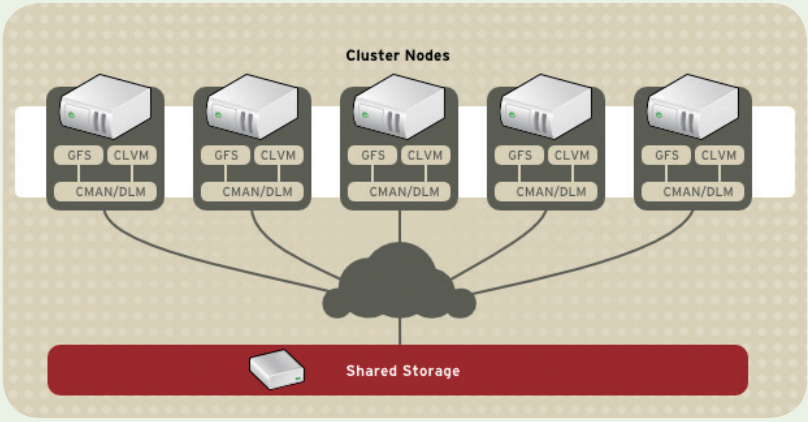
대표적인 파일시스템으로는 Quantum의 StorNext File System(SNFS), IBM의 General Parallel File System(GPFS)가 있다. 일례로 StorNext File System의 전체적인 구조는 다음과 같으며, 파일시스템과 스토리지를 관리하는 Metadata controller를 볼 수 있다.



[그림 1] 참조 : StorNext youtube 동영상 http://www.youtube.com/watch?v=62VzKrueOYY

2) 대칭형 분산 방식

이 방식은 로컬 파일시스템처럼 공유 디스크에 공유 파일시스템을 구축하고 사용하는 클라이언트 간의 동시성 제어 및 동기화를 위하여 클러스터로 묶어 관리한다. 클러스터 관리자 및 분산 락 관리자 등의 관련 S/W들은 서버 간의 네트워크 통신 프로토콜을 이용하여 클러스터 멤버 간에 동시 디스크 쓰기 작업을 관리하므로 추가 메타데이터 제어기가 필요하지 않다.



[그림 2] 참조 : https://access.redhat.com/documentation/fr-FR/Red_Hat_Enterprise_Linux/5/html/Cluster_Suite_Overview/images/DLM_Overview.png

전성원
NaverLabs(네이버랩스) 수석연구원

대표적인 파일시스템으로는 Red Hat의 Global File System(GFS2), Oracle Cluster File System(OCFS2) 등이 있다. [그림 2]는 Global File System 2의 S/W 및 H/W 구성도이다.

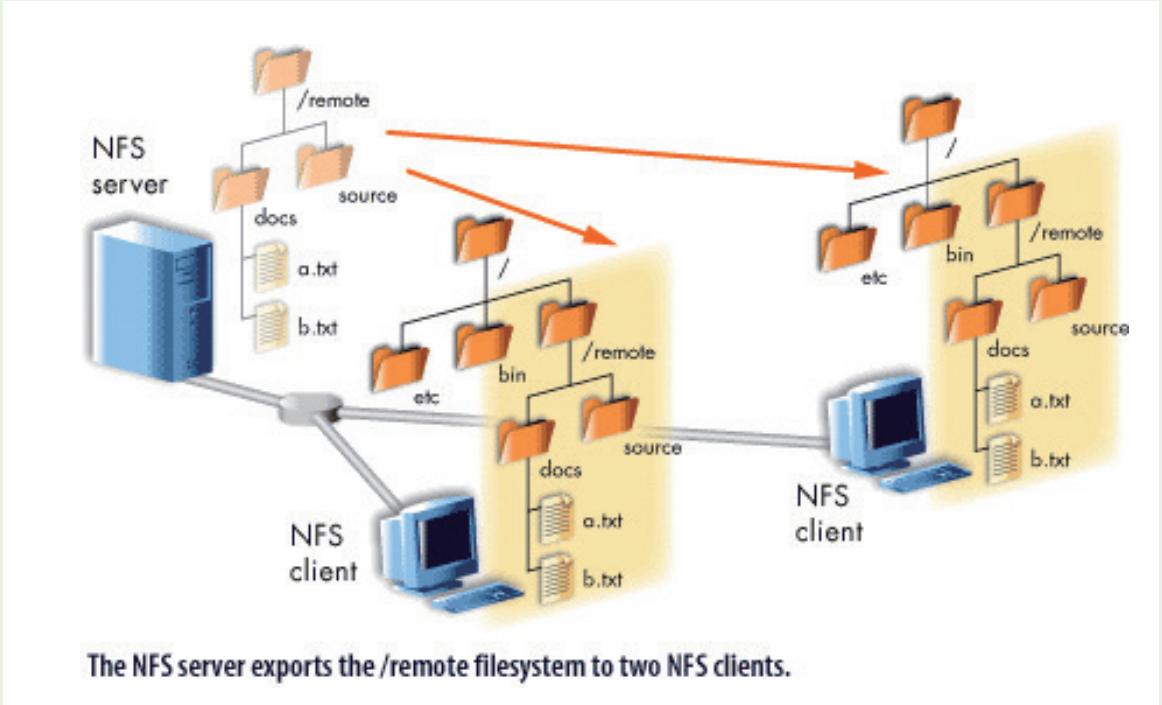
2. 분산 파일시스템

분산 파일시스템은 스토리지를 블록 레벨로 공유하는 것이 아니라 네트워크 프로토콜(protocol)을 이용하여 접근하게 된다. 그래서 구현 방식이나 프로토콜에 따라 파일시스템의 특성 차이가 많아 각각의 분산 파일시스템의 특징을 살펴보는 것이 좋다. 분산 파일시스템은 다시 서비스하는 서버의 수를 기준으로 두 가지 분류로 나뉠 수 있다. 하나는 고성능 단일 서버를 사용하는 분산 파일시스템인 Network File System(NFS), Common Internet File System(CIFS)가 있고 저가의 다수의 서버를 묶어 사용하는 Google File System(GFS), Hadoop Distributed File System(HDFS), Inktank의 Ceph, 네이버의 OwFS 등이 있다.

전자의 경우 우리가 주변에서 일반적으로 볼 수 있는 대표적인 파일시스템으로 대부분의 시스템에서 표준으로 제공하고 있어 사용이 편리하다. 그리고 대부분의 엔터프라이즈 환경에서는 고가용성 등의 다양한 추가 기능을 제공하는 NAS(Network Attached Storage)를 주로 이용하고 있다. 후자의 GFS, OwFS 같은 경우 공개되어 있지 않으므로 일반적으로 사용할 수가 없다. 그러므로 Open Source로 누구나 사용할 수 있는 HDFS와 Ceph에 대하여 알아보도록 한다.

1) Network File System(NFS), Common Internet File System(CIFS)

NFS는 1984년에 Sun에서 개발한 프로토콜로 1989년부터 사용하게 되었다. 그리고 현재 모든 Unix 계열의 System에서 기본 파일시스템 중 하나로 지원하고 있다. NFS는 단순한 클라이언트/서버 방식으로 서버는 자신의 로컬 파일시스템의 일부를 NFS로 외부에 공유하고 다수의 NFS 클라이언트는 이 공유된 파일시스템을 클라이언트에 마운트하여 사용하게 된다.



[그림 3] 참조 : http://tuxapuntes.com/2014/10/conociendo-nfs-network-file-system/

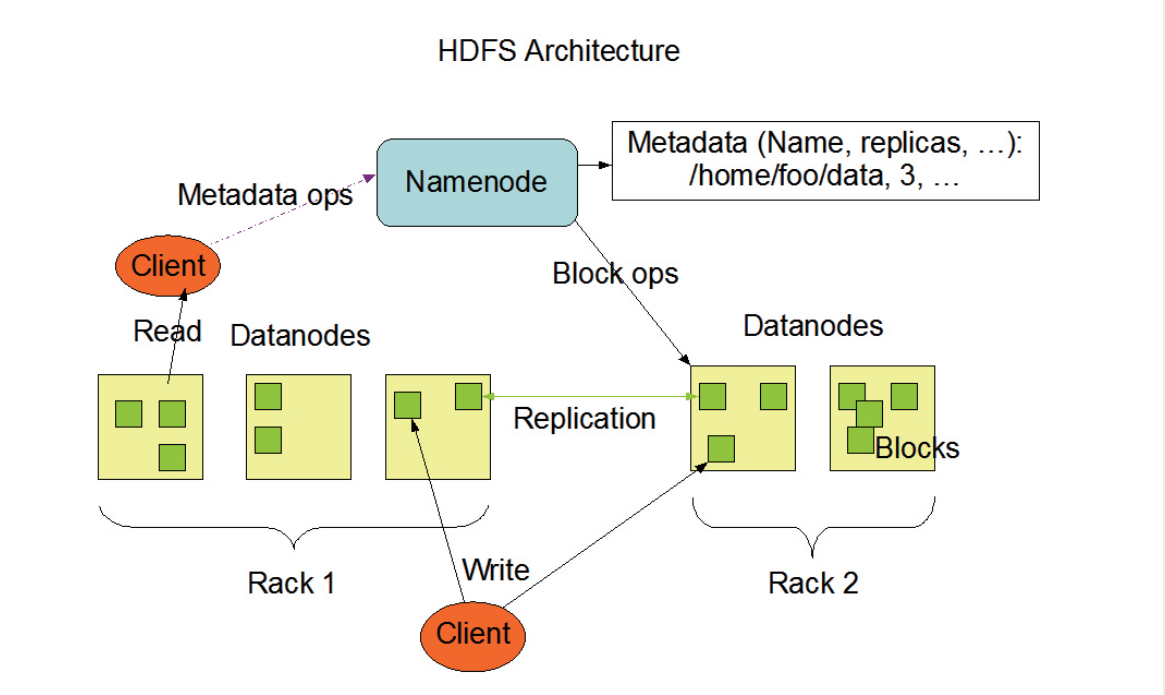
CIFS는 Server Message Block(SMB)이라고도 불리며 파일이나 프린터, 등을 공유할 수 있는 프로토콜로 Microsoft Windows 네트워크 환경에서 주로 사용된다. CIFS는 MS Windows OS에 기본으로 장착되어 있으며, 점점 지원하는 OS도 많아지고 있다.

NFS와 CIFS는 로컬 파일시스템과 동일한 기능을 그대로 지원하기 때문에 사용하기 편리하며, 모든 응용 프로그램을 사용하는데 있어서 호환성 문제가 없는 장점을 가지고 있다. 그러나 하나의 서버에 있는 파일들을 여러 클라이언트에서 공유하여 사용하기 때문에 서버의 성능에 많은 영향을 받게 된다. 그러므로 서비스하는 서버가 병목될 수 있으며, 또 서버 장애 시 서비스가 중단될 수 있다. 현재 엔터프라이즈 환경에서는 NFS 서비스를 하면서 고가용성 등의 다양한 추가 기능을 제공하는 NAS(Network Attached Storage)를 주로 이용하고 있다.

2) Hadoop Distributed File System(HDFS)

GFS는 Google이 전 세계의 web 데이터를 수집하여 저장하고 분석을 하는데 사용하는 파일시스템이다. GFS는 전체 파일시스템의 기능을 모두 구현한 것이 아니라 Google에서 사용하는데 꼭 필요한 기능만을 구현하고 구현의 편의성이나 이식성 등을 고려하여 API를 이용하여 사용하도록 하였다. 그리고 GFS 내에 저장된 data를 빠르게 분석하기 위한 맵-리듀스(map-reduce)라는 병렬 분석 기능에 일부 초점이 맞춰져 있다. GFS는 다수의 저가의 서버가 가지고 있는 로컬 디스크를 묶어 하나의 커다란 파일시스템을 구성하여 상업적으로도 사용할 수 있다는 것을 보여 주었으며, 이후 많은 분산 파일시스템에 영향을 주었다. 그러나 GFS는 Google에서만 사용하고 있어 오픈되어 있지 않아 Apache Hadoop 프로젝트에서 Google이 발표한 GFS 논문을 토대로 GFS와 같은 기능을 하는 Hadoop Distributes File System을 Java를 이용하여 개발하였다. 이 HDFS는 Open Source로 누구나 사용할 수 있다.

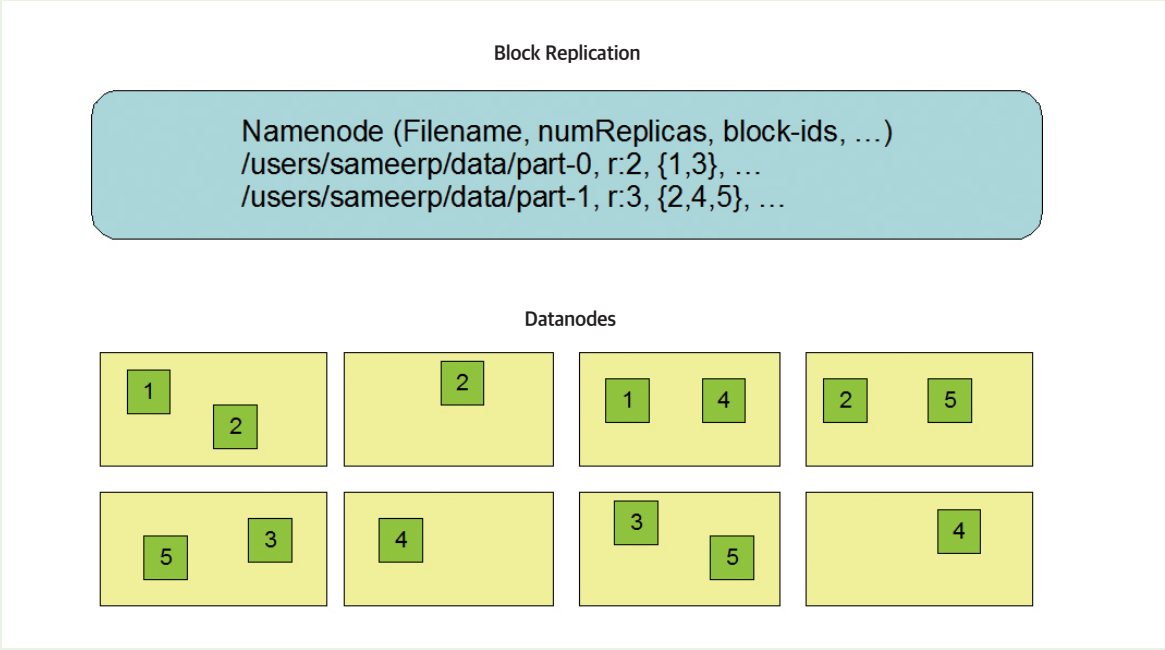
HDFS는 NameNode라고 하는 파일시스템 메타데이터를 관리하는 서버와 DataNode라고 하는 파일 데이터를 저장하고 서비스하는 서버로 나누어진다. 클라이언트는 NameNode로부터 파일에 대한 메타데이터 요청을 하면 NameNode가 파일 데이터가 저장된 DataNode 정보를 전달하고 클라이언트는 해당 파일 데이터가 있는 DataNode로 접근하여 파일 데이터를 읽거나 쓰는 방식이다. 그 구성도는 아래와 같다.



[그림 4] 참조 : http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

전성원
NaverLabs(네이버랩스) 수석연구원

하나의 파일은 기본적으로 64MB 단위의 청크(chunk)로 나뉘어져 서로 다른 서버에 저장이 되며, 서버 장애에 대비하여 기본적으로 3개의 서로 다른 서버에 같은 청크를 3벌 각각 저장한다. 그림에서 녹색 블록이 청크를 나타낸다.



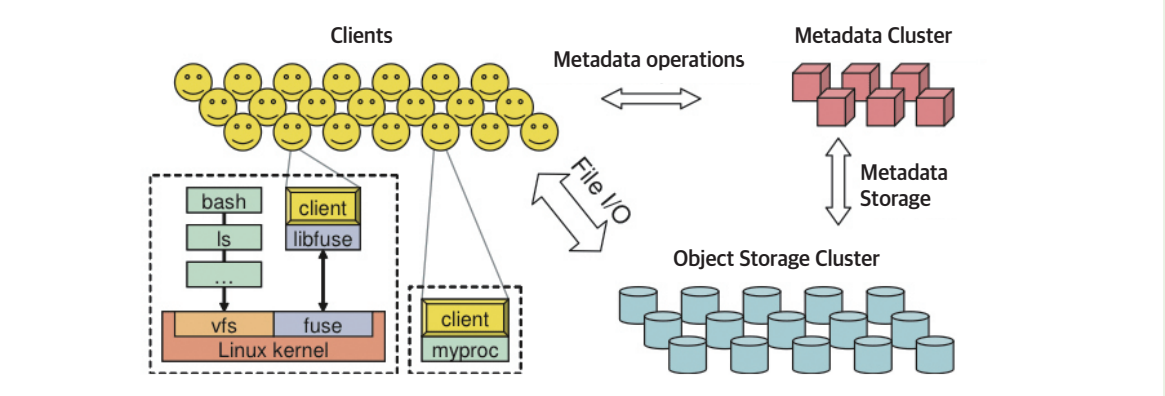
[그림 5] 참조 : http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

이 파일시스템의 특징은 청크 크기가 64MB이므로 파일이 이 크기보다 커야 효율적으로 저장할 수 있다. 그리고 모든 파일시스템의 메타데이터를 NameNode의 메모리에서 처리하므로 파일이 많아지는 경우 메모리가 부족해질 수 있다. 그러므로 GFS나 HDFS는 수 100GB에서 수 TB 이상의 커다란 파일에 적합하며, 파일의 수가 많거나 크기가 적은 파일은 저장하기에는 적합하지 않다. 그리고 파일시스템의 표준으로 정의된 모든 기능을 다 지원하지 않으며, API로 사용하도록 되어 있어 기존 응용 프로그램들과는 호환되지 않는다. 다만 최근 Fuse 등의 기능을 이용하여 Unix 계열의 OS의 경우 마운트를 할 수 있지만, 파일시스템의 모든 기능을 다 사용할 수 있는 것은 아니다.

3) Inktank Ceph

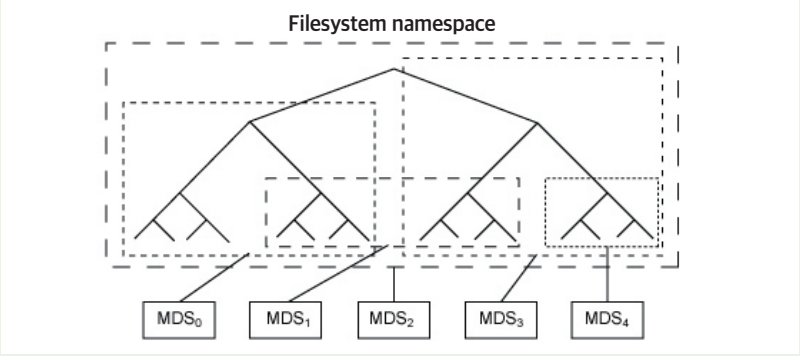
Ceph는 분산 파일시스템으로 메타데이터를 관하는 방식이 독특하다. 메타데이터 서버를 이용해 전체 파일시스템의 네임스페이스와 메타데이터를 관리하는 것은 다른 분산 파일시스템과 비슷하지만 메타데이터 서버들이 클러스터 형태로 동작하며, 동적으로 부하 정도에 따라 메타데이터 별로 담당하는 네임스페이스 영역을 조정할 수 있다는 것이 특징이다. 이를 통해 부하가 일부에 집중되는 경우에도 쉽게 대처할 수 있으며 쉽게 메타데이터 서버를 확장할 수 있다. 그뿐만 아니라 다른 분산 파일시스템과 달리 POSIX와 호환되며 마운트를 지원하여 좀 더 로컬 파일시스템과 가깝게 사용할 수 있다는 것이 장점이다.

Ceph의 유연한 메타데이터의 관리 기능은 기존 분산 파일시스템이 가지고 있는 메타데이터 관리의 어려움을 극복하는 한가지 방법을



[그림 6] 참조 : http://docs.openstack.org/havana/config-erence/content/ceph-rados.html

제시했다고 볼 수 있다. [그림 7]과 같이 여러 메타데이터 클러스터에서 디렉터리 트리의 일부분을 여러 메타데이터 서버가 분담하여 서비스하는 개념을 보여 주고 있다. MDS_2는 전체 네임 스페이스(name space)를 처리하지만 현재 많은 요청이 있는 하위 트리에 대하여 MDS_0, MDS_1, MDS_3, MDS_4가 각각을 담당하여 처리하는 경우를 설명하는 것이다.



[그림 7] 참조 : http://140.120.15.179/Howto-Install/CEPH/VirtualStorageAndUsbHdd.html

Ceph를 주목해야 하는 이유는 Linux 커널 소스(kernel source)에 포함돼 있다는 것이다. 그래서 Linux의 발전과 더불어 미래에 Linux의 주력 파일시스템이 될 수 있는 가능성이 있다는 것이다. 뿐만 아니라 인터넷의 클라우드 스토리지와 같이 REST API도 지원하며 Swift나 Amazon S3의 REST API와도 호환이 가능하다.

III. 마치며

앞에서 대표적인 몇 가지 클러스터 파일시스템에 대한 특징을 알아보았다. 이 이외에도 많은 클러스터 파일시스템이 존재하고 사용되고 있다. 필자의 생각으로는 일반적으로 방송에서 사용하는 영상을 편집하거나 변환하는 작업은 기존 로컬 파일시스템과 동일한 기능을 제공하는 공유 디스크 파일시스템이나 NAS를 이용한 NFS, CIFS를 이용하는 것이 좋아 보인다. 특히 영상 작업을 위한 읽기/쓰기 성능이 필요한 경우 공유 디스크 파일시스템이 더욱 유리할 것이다. 그러나 이런 파일시스템들은 대부분 구축 및 유지 비용이 많이 들어가게 된다. 이에 반해 저렴한 서버를 사용하는 분산 파일시스템은 훨씬 저렴한 비용으로 구축할 수 있다. 그러나 분산 파일시스템은 특징에 따라 장단점이 있으므로 사용하려는 목적에 맞게 잘 선택하여 사용하지 않으면 활용도가 떨어지게 된다. 그러므로 분산 파일시스템의 특징을 잘 알고 사용 목적에 선택하여 사용해야 한다. 추가로 다수의 서버를 묶어 운영해야 하므로 운영 측면에서도 좀 더 많은 경험과 지식이 필요한 부분이 있다. 그렇지만 동영상 서비스의 스토리지로 분산 파일시스템을 비용 측면에서는 한번 고려해볼 만하다고 생각한다.