

데스크톱 가상화를 위한 파일 기반 스냅샷

김영철*, 이상민*, 김영균*

*한국전자통신연구원

e-mail : kimyc@etri.re.kr

File-based Snapshot for Desktop Virtualization

Young-Chul Kim*, Sang-Min Lee*, Young-Kyun Kim*

*Electronics and Telecommunications Research Institute

요 약

최근 기업, 학교 등에서 가상 데스크톱 환경을 구축하는 사례들이 확대되면서 보다 효율적이고 안정적인 서비스를 제공할 수 있는 적합한 스토리지 시스템 도입이 요구되고 있다. 가상 데스크톱 시스템은 대부분 가상 데스크톱의 기본 이미지를 스토리지 시스템에 저장하고 다수의 가상 데스크톱 스냅샷들이 기본 이미지를 공유하도록 함으로써 시스템 운영과 보안 그리고 자원 활용을 높이도록 구성한다. 이러한 가상 데스크톱 이미지는 파일 시스템에서 파일 기반으로 저장되고 관리된다. 따라서 파일 시스템에서 파일 기반의 스냅샷 기능이 필요하다. 하지만 일반 파일 시스템에서는 스냅샷과 같은 기능이 제공되지 않기 때문에 이미지를 QCOW2 등과 같은 파일 포맷으로 만들어서 스냅샷 기능을 사용할 수 있다. 하지만 스냅샷 기능을 수행하기 위해서는 파일 데이터를 읽고 쓰는 동작이 함께 수반되어야 하기 때문에 가상 데스크톱 성능에 커다란 영향을 미칠 수 있으며 가상 데스크톱 환경이 확장함에 따라 파일 데이터 입출력으로 인한 오버헤드로 서비스를 제공하기 어렵게 된다. VDI-FS 시스템은 가상 데스크톱 환경을 지원하기 위한 파일 시스템으로 기본 이미지와 스냅샷들 간의 메타데이터 및 데이터 공유를 제공하며 보다 효율적인 스냅샷 기능을 제공한다. 본 논문에서는 VDI-FS 시스템에 대한 소개와 스냅샷을 지원하는 방법에 대해 기술한다. 그리고 NFS 에서 QCOW2 를 이용한 스냅샷을 사용했을 때와 VDI-FS 시스템에서 스냅샷을 사용했을 때와의 성능 비교를 통하여 VDI-FS 시스템의 스냅샷 기능이 우수한 성능을 보임을 제시한다.

1. 서론

서버 가상화 기술에 이어 데스크톱 가상화 기술이 발전함에 따라 데스크톱 환경을 가상 데스크톱으로 대체하는 사례가 빠르게 확대되고 있다. 가상 데스크톱 환경을 구축함에 있어 가장 중요한 이슈 중에 하나는 효율적이고 안정적인 입출력을 제공할 수 있는 스토리지가 요구된다는 것이다. 여기에 더불어 가상 데스크톱 서비스를 효율적으로 운영하기 위해 필요한 스냅샷 기능과 같은 스토리지 기능이 제공되어야 한다.

일반적으로 가상 데스크톱 운영은 주로 몇 개의 가상 데스크톱 기본 이미지를 두고 다수의 가상 데스크톱들이 기본 이미지를 함께 공유하면서 변경된 내용만을 저장하도록 하는 쓰기 가능 스냅샷 또는 클론이라는 방식을 사용함으로써 통합 운영되는 스토리지 효율성을 제공하고자 한다. 이를 위해서는 VMware [7]의 VMFS 와 같은 특화된 파일 시스템에서 자체적으로 제공하는 스냅샷 기능을 사용한다. 하지만 일반 파일 시스템에서는 이를 위한 별도의 기능을 제공하지 않기 때문에, QCOW2 [5]와 같은 특정 파일 포맷으로 데이터를 저장함으로써 스냅샷 기능을 이용할 수 있다. 이러한 QCOW2 방식은 가상 데스크톱 환경이

확장될수록 파일 포맷을 유지하기 위한 데이터 입출력 오버헤드로 가상 데스크톱 서비스를 제공하기 어렵다.

VDI-FS 시스템은 가상 데스크톱 환경에서 요구되는 스냅샷 기능 및 우수한 입출력 성능과 확장성을 제공하는 시스템으로 개발되었다.

본 논문은 다음과 같이 구성된다. 2 장에서는 VDI-FS 시스템의 전체 구조에 대해 소개하고, 3 장에서는 VDI-FS 시스템에서 제공하는 쓰기 가능 스냅샷에 대해 설명한다. 4 장에서는 QCOW2 와의 입출력 성능을 비교하고 마지막으로 5 장에서 결론을 맺는다.

2. VDI-FS 시스템

VDI-FS 시스템은 가상 데스크톱 환경을 지원하는 파일 시스템으로 메타데이터 서버, 데이터 서버 그리고 클라이언트로 구성되며, 파일 메타데이터를 접근하는 경로와 파일 데이터를 접근하는 경로가 분리된 비대칭 분산 파일 시스템 구조를 갖는다. 그리고 이들 구성 요소들 간에는 네트워크로 연결된다. 이러한 VDI-FS 시스템은 국내에서 개발된 GLORY-FS [1], MAHA-FS [2]와 동일한 구조를 갖는다.

VDI-FS 메타데이터 서버는 파일 시스템 메타데이터 처리, 자원 관리, 데이터 서버 및 클라이언트 모니터

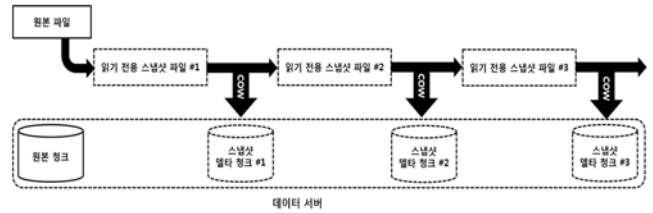
터링 및 장애 복구 등을 수행한다. 또한 메타데이터 서버는 전체 파일 시스템을 논리적인 볼륨 단위로 나누어 구성하고 관리한다.

VDI-FS 데이터 서버에는 파일 데이터가 분산되어 저장된다. 데이터가 저장되는 단위는 볼륨, 디렉토리 또는 파일 별로 지정된 청크(Chunk) 단위로 분할되어 서로 다른 데이터 서버들에 분산되어 저장된다. 또한 이렇게 저장된 청크들은 장애 복구 및 부하 분산을 위하여 다시 서로 다른 데이터 서버에 복제된다. 청크를 복제하는 방식은 설정에 따라 동기식 또는 비동기식으로 이루어질 수 있다.

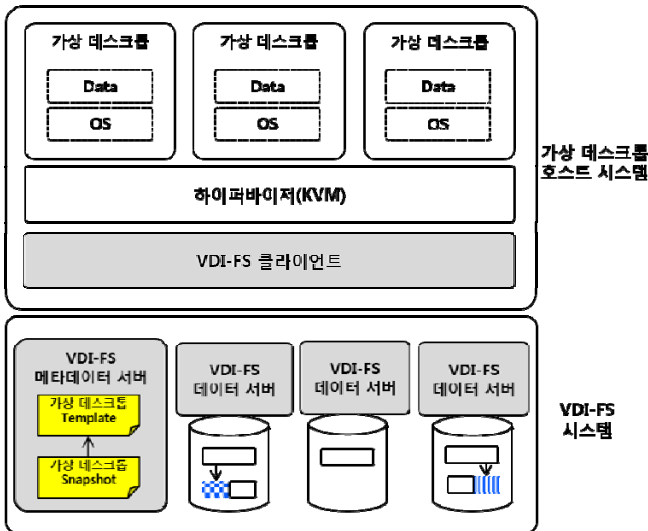
VDI-FS 클라이언트는 FUSE(Filesystem in Userspace) [4]를 기반으로 구현되었다. FUSE 는 커널 모듈과 사용자 라이브러리로 구성되며, 사용자 수준의 파일 시스템을 구현할 수 있도록 함으로써 POSIX 표준 파일 인터페이스 사용할 수 있는 기능을 제공한다. VDI-FS 클라이언트는 메타데이터 서버에서 관리하는 볼륨 단위로 마운트하며 네임스페이스 처리 및 파일 입출력을 로컬 파일 시스템처럼 수행할 수 있다. 따라서 클라이언트에서 파일 입출력을 수행하기 위해서는 먼저 메타데이터 서버로부터 파일의 메타데이터와 파일 데이터의 청크 레이아웃 정보를 받아온 후에, 해당 청크가 저장된 데이터 서버로부터 직접 입출력을 수행하게 된다.

스냅샷 지원 파일 포맷으로 저장해야 한다.

VDI-FS 는 파일 기반 스냅샷으로 읽기 전용 또는 쓰기 가능 스냅샷을 제공한다. 읽기 전용 스냅샷은 스냅샷을 생성할 시점에 파일 상태를 보관함으로써 스냅샷을 생성한 이후에 원본 파일이 계속해서 갱신되더라도 스냅샷 생성 시점의 파일로 복귀할 수 있도록 제공한다. 또한 읽기 전용 스냅샷은 시간 상에서 계속해서 생성할 수 있으며 각각의 스냅샷은 생성 시점의 원본 파일 상태를 보관하게 된다. 반면에 쓰기 가능 스냅샷은 읽기 전용 스냅샷과 같이 생성되며, 스냅샷 생성 시점의 원본 파일과 메타데이터를 공유하지만 읽기 전용 스냅샷과는 달리 원본 파일과는 독립적으로 스냅샷에 대한 읽기 또는 쓰기가 가능하다. 그리고 원본 파일에 대한 갱신은 쓰기 가능 스냅샷에 반영되지 않으며, 쓰기 가능 스냅샷에 대한 갱신은 원본 파일에 반영되지 않는다.



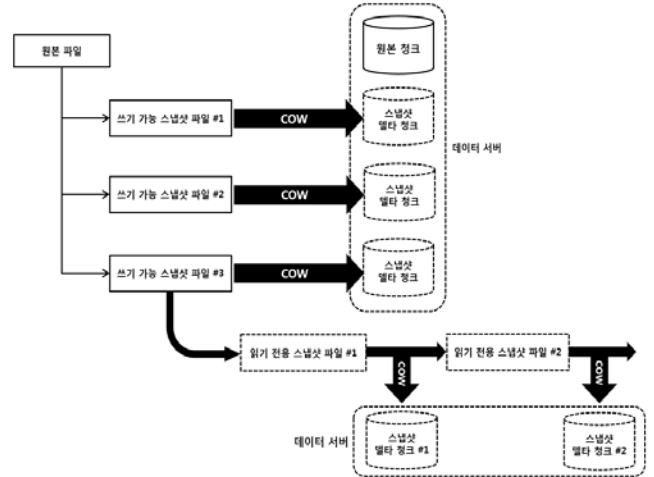
(그림 2) 읽기 전용 스냅샷



(그림 1) VDI-FS 시스템

3. 파일 기반 쓰기 가능 스냅샷

파일 시스템에서는 일반적으로 파일 시스템 또는 볼륨 수준에서 스냅샷을 제공하며 파일 메타데이터와 데이터를 함께 관리하는 구조를 갖는다. 스냅샷을 구현하는 방법으로는 일반적으로 COW(Copy-On-Write), ROW(Redirect-On-Write) 등을 사용한다. 파일 기반 스토리지에서 가상 데스크톱 이미지 또는 가상 디스크 이미지는 파일로 저장되고 관리된다. 이때 파일은 주로 RAW 또는 QCOW2 포맷이 사용된다. 따라서 파일 기반의 스냅샷을 제공하기 위해서는 QCOW2 와 같은



(그림 3) 쓰기 가능 스냅샷

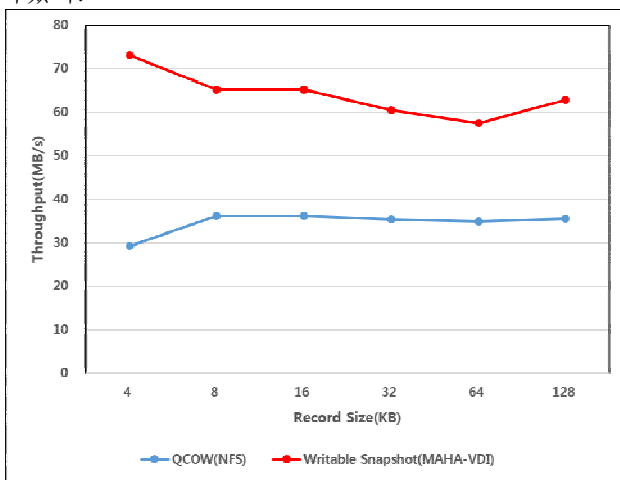
VDI-FS 메타데이터 서버에서는 원본 파일의 메타데이터 정보를 유지하고 있기 때문에 스냅샷이 원본 파일의 메타데이터 정보를 공유할 수 있도록 할 수 있다. 반면에 QCOW2 는 스냅샷에 대한 메타데이터 정보가 파일 데이터로 저장되어 있어서 메타데이터를 접근하거나 변경하기 위해서는 데이터 입출력이 수반되어야 한다.

VDI-FS 쓰기 가능 스냅샷에서는 데이터 갱신이 발생하였을 때 갱신되는 데이터에 대해 원본 청크로부터 스냅샷 메타 청크라는 별도의 청크로 COW 를 수행하여 저장하게 된다. 이때 수행되는 COW 는 데이터 서버 수준에서 수행되며 클라이언트의 입출력을 수반하지 않는다. 또한 COW 는 미리 지정되거나 또

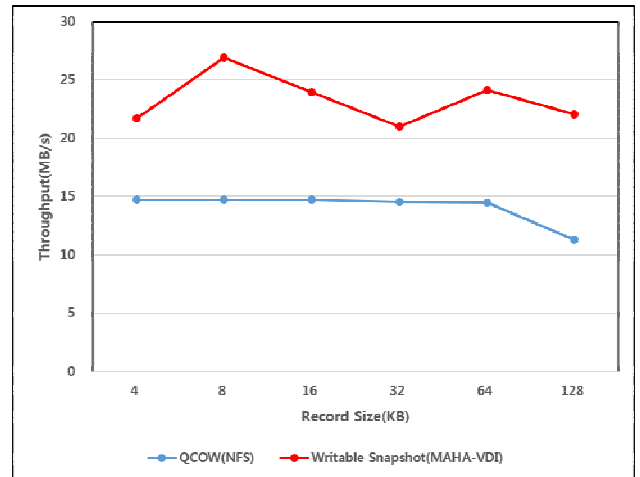
는 워크로드에 따라 결정된 익스텐트(Extent) 단위로 이루어지게 되며, 스냅샷 델타 청크에 저장된 갱신 데이터에 대해서는 비트맵(Bitmap)을 유지함으로써 추후 입출력을 수행할 때 원본 데이터 청크로부터 할 것인지 아니면 스냅샷 델타 청크로부터 할 것인지를 결정할 때 사용하게 된다. 이와 같이 VDI-FS 쓰기 가능 스냅샷은 COW 를 위하여 데이터를 읽어서 갱신한 후에 다시 데이터를 내려 쓰는 이중 작업이 필요하지 않기 때문에 보다 빠른 데이터 입출력을 제공할 수 있게 된다. 이러한 메타데이터 공유와 데이터 COW 는 가상 데스크톱 성능에 상당한 영향을 미칠 수 있으며 가상 데스크톱 적용 여부에 중요한 잣대가 된다.

4. 성능평가

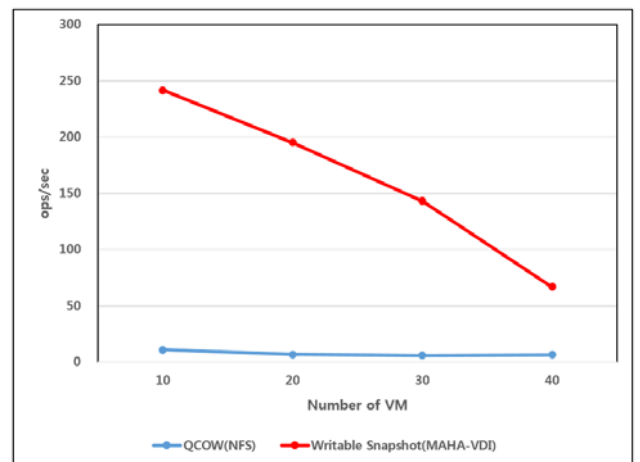
성능평가는 NFS [3]와 VDI-FS 시스템 상에 가상 데스크톱을 구성하고 가상 데스크톱 내에서 파일 시스템 입출력 벤치마크 도구인 IOzone [7]을 이용하여 순차/랜덤 입출력 성능을 시험하였다. 가상 데스크톱은 10 개의 호스트에 각각 4 개씩 생성하였고, 각각의 가상 데스크톱에서 IOzone 을 수행하였다. VDI-FS 는 메타데이터 서버, 데이터 서버 각각 1 대로 구성하였으며, NFS 는 서버 1 대로 구성하였다. 호스트의 하이퍼바이저는 KVM 이고, 6G 메모리와 인텔 Xeon 2.4GHz 4 core CPU 를 사용하였다. 가상 데스크톱에는 각각 1G 메모리와 1 개의 CPU 를 할당하였다. 그리고 가상 데스크톱에는 리눅스 CentOS 6.5 를 설치하여 시험을 수행하였다.



(그림 4) 가상 데스크톱 순차 쓰기 성능



(그림 5) 가상 데스크톱 순차 읽기 성능



(그림 6) 가상 데스크톱 랜덤 쓰기 성능

성능시험 결과 VDI-FS 환경에서 쓰기 가능 스냅샷을 이용한 가상 데스크톱이 NFS 환경에서 QCOW2 를 이용한 가상 데스크톱 보다 우수한 입출력 성능을 보였다. VDI-FS 의 경우 데이터 서버를 확장함에 따라 성능이 선형적으로 향상될 수 있는 반면 NFS 의 경우에는 확장에 한계가 있으며 이것은 전체 시스템의 성능 저하를 초래할 수 있다.

5. 결론

본 논문에서는 점차 확대되어 가는 가상 데스크톱 환경에서 요구되는 입출력 성능을 제공할 수 있는 VDI-FS 시스템을 소개하고 VDI-FS 시스템에서 가상 데스크톱을 운영하기 위해 필요한 스냅샷 기능인 쓰기 가능 스냅샷을 제안하였다. 그리고 기존에 이용되고 있는 QCOW2 포맷과의 성능 비교를 통하여 VDI-FS 시스템에서 스냅샷을 이용한 가상 데스크톱의 입출력 성능이 보다 우수함을 보였다.

Acknowledgments

※ 본 연구는 지식경제부 및 한국산업기술평가관리원의 IT 산업융합원천기술개발사업의 일환으로 수행

하였음. [10041730, 10,000 사용자 이상 동시 접속
가상 데스크톱 서비스를 지원하는 클라우드 스토
리지용 파일 시스템 개발]

참고문헌

- [1] Youngsoo Min, Keuntae Park, Myunghoon Cha, Kisung Jin, Sangmin Lee, Youngkyun Kim, Myungjoon Kim. "Glory-FS: A Design and Implementation of Distributed File System for Large Scale Global Internet Service," Proc. in 4th International Conf. on New Exploratory Technologies, 2007.
- [2] 김영창, 김동오, 김홍연, 김영균, 최완. "MAHA-FS: 고성능 메타데이터 처리 및 랜덤 입출력을 위한 분산 파일 시스템," 정보처리학회지, 제 2 권 제 2 호, 2013.
- [3] B. Callaghan, B. Pawlowski, P. Staubach. NFS version 3 protocol specification. RFC 1813, Network Working Group, 1995.
- [4] FUSE: Filesystem in Userspace. <http://fuse.sourceforge.net>
- [5] The QCOW2 Image Format. <http://www.gnome.org/~markmc/qcow-image-format.html>
- [6] VMware. <http://www.vmware.com>
- [7] KVM: Kernel Based Virtual Machine. <http://www.linux-kvm.org>
- [8] IOzone Filesystem Benchmark, <http://www.iozone.org>