데이터 보호를 위한 파일시스템 기반의 스냅샷

엄현철○, 장한을, 노재춘
세종대학교 컴퓨터공학과
mayaabba@naver.com○, slpenbi@gmail.com, jano@sejong.ac.kr

Filesystem-based Snapshot for Data Protection

Hyun Chul Eom○, Han Eul Jang, Jaechun No
Dept. of Computer engineering, Sejong University

파일시스템 스냅샷은 파일시스템에 저장된 데이터들에 대한 특정 시점의 이미지를 생성하고 기록할 수 있는 유용한 기법이다. 파일시스템 스냅샷을 이용하기 위해서는 파일시스템이 자체적으로 스냅샷 기능을 제공해야 하지만 현재 리눅스에서 일반적으로 사용되는 파일시스템들은 스냅샷 기능을 제공하지 않는다. 따라서 파일시스템 스냅샷을 사용하기 위해서는 사용 중인 파일시스템에 스냅샷 기능을 직접 추가하거나 스냅샷 기능을 제공하는 특별한 파일시스템을 선택해야 한다.

본 논문에서는 특정 파일시스템에 종속되지 않고 모든 파일시스템에서 공통적으로 사용 가능한 파일시스템 스냅샷 라이브러리인 PSnap(Portable Snapshot Library for Filesystem)을 설계하고 구현한다. 기존의 파일시스템 스냅샷들은 파일시스템의 자료구조와 데이터 블록 처리 알고리즘에 종속되어 서로 다른 파일시스템으로 이식이 불가능하고 스냅샷 정보를 기록하기 위해서 원본 파일시스템의 자료구조를 변경하는 경우 원본 파일시스템과 호환이 안 된다는 문제가 있다.PSnap은 파일시스템에 비종속적인 PSnap_API와 종속적인 처리를 담당하는 PSnap_Interface를 분리하여 PSnap_Interface가 구현된 모든 파일시스템들이 동일한 PSnap_API를 호출함으로서 파일시스템에 스냅샷 기능을 추가할 수 있게 한다. 또한 파일시스템의 자료구조를 변경하지 않고 원본 파일시스템과의 호환성을 유지함으로서 기존에 사용 중인 파일시스템에도 스냅샷 기능을 추가할 수 있다.

그림 1은 파일시스템과 PSnap의 관계를 보인다. 파일시스템은 스냅샷 기능을 추가하기 위해서 PSnap_API를 사용한다. PSnap_API는 모든 파일시스템에서 공통적으로 이용 가능한 스냅샷 정보 관리 기법과 Redirect-on-Write 스냅샷 알고리즘을 사용하여 스냅샷을 처리하고, 실제 파일시스템의 자료구조에 접근해야 할 필요가 있을 때는 PSnap_Interface를 호출하여 간접적으로 처리한다. PSnap_Interface는 특정 파일시스템의 자료구조에 접근하고 수정하는 방법을 정의한다.파일시스템 스냅샷은 스냅샷 처리를 위해서 파일의 데이터 블록을 직접적으로 조작하기 때문에PSnap_API를 사용하기 위해서는 다음과 같은 PSnap_ Interface들이 정의되어야 한다.

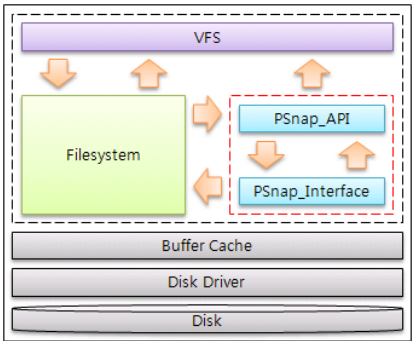

그림 1. 파일시스템과 PSnap의 관계

- psnap_interface_init_cmap : 파일시스템의 데이터 블록 개수를 읽어서 스냅샷 정보 관리에 사용되는 PSnap_Countmap의 크기를 결정하고 초기화한다.
- psnap_interface_copy_data : 스냅샷 파일이 생성될 때 원본 파일의 메타데이터를 스냅샷 파일로 복사한다.
- psnap_interface_set_cmap : 스냅샷 파일이 생성될 때 원본 파일이 사용 중인 데이터 블록에 맵핑된 PSnap_Countmap 엔트리들의 값을 1씩 증가시킨다.
- psnap_interface_chk_cmap_mdfy_blk : 맵핑된 PSnap_Countmap 엔트리 값이 1이상인 데이터 블록이 수정될 때 스냅샷 처리를 한다.

그림 2는 PSnap에서 사용하는 스냅샷 정보 관리 기법을 나타낸다. PSnap은 스냅샷 정보를 관리하기 위해서 파일시스템의 최상위 디렉토리에 PSnap_Countmap 파일을 생성한다. PSnap_Countmap은 파일시스템의 모든 데이터 블록에 대해 1:1로 맵핑되는 엔트리들로 구성된다. 각각의 엔트리들은 0으로 초기화되며 자신과 맵핑된 파일시스템 데이터 블록이 공유되고 있는 횟수를 나타낸다.PSnap이 추가된 파일시스템에 데이터 블록1, 2, 3을 가지는 파일 A를 생성한 후 파일 A에 대한 스냅샷 파일 A'를 생성하는 시점에 PSnap_Countmap은 갱신된다. 파일 A가 포함하는 3개의 데이터 블록이 스냅샷 파일A'에 의해서 공유되기 때문에 데이터 블록1,

"본 논문은 Seoul R&BD Program(10557) 지원을 받아 수행되었음"

2, 3에 맵핑된 PSnap_Countmap 엔트리 1, 2, 3의 값이 1씩 증가한다. 이후 파일 A에 대한 수정 작업이 발생하면 PSnap은 PSnap_Countmap을 참조하여 데이터 블록 1, 2, 3이 공유되고 있음을 확인하고 스냅샷 처리를 한다. 스냅샷 처리 후 데이터 블록 1, 2, 3에 대한 공유가 감소하기 때문에 PSnap_Countmap 엔트리 1, 2, 3의 값이 1씩 감소한다. PSnap은 PSnap_Countmap을 접근할 때 VFS에서 제공하는 함수를 사용함으로서 파일시스템 독립성을 유지한다. 또한 스냅샷 정보 관리를 위해서 원본 파일시스템의 자료구조를 변경하지 않기 때문에 원본파일시스템과의 호환성도 유지할 수 있다.

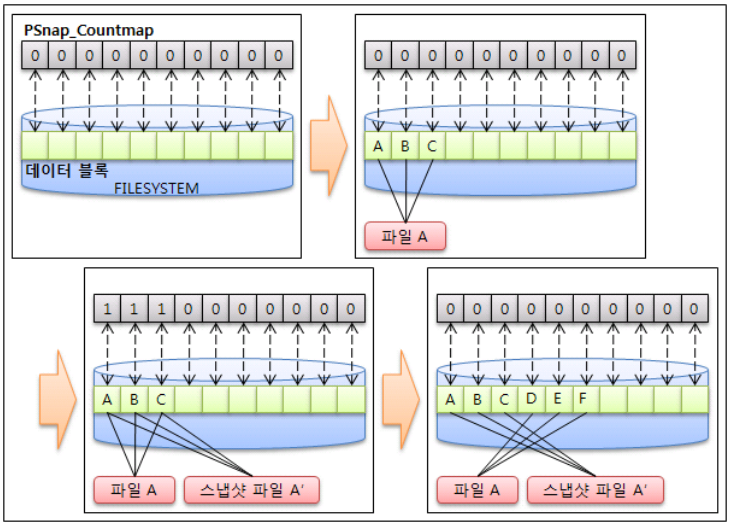

그림 2. 스냅샷 정보 관리

현재 PSnap은 리눅스 커널 2.6.28.2에서 EXT2 파일시스템과 EXT3 파일시스템을 지원한다. 그림 3은 구현된 PSnap을 이용하여 EXT2 파일시스템과 EXT3 파일시스템에서 성능을 측정한 결과이다. 성능 측정은 CPU : Intel(R) Core(TM)2 Duo CPU E6300 @ 1.86GHZ, RAM : Samsung DDR2 1GB PC2-6400 x 1, HDD : Seagate ST3750330AS SATA2/750GB/ 7200RPM/32M 환경에서 원본 파일시스템과PSnap이 추가된 파일시스템에 4KB 단위로 2GB까지 WRITE, READ, REWRITE를 수행하였다.

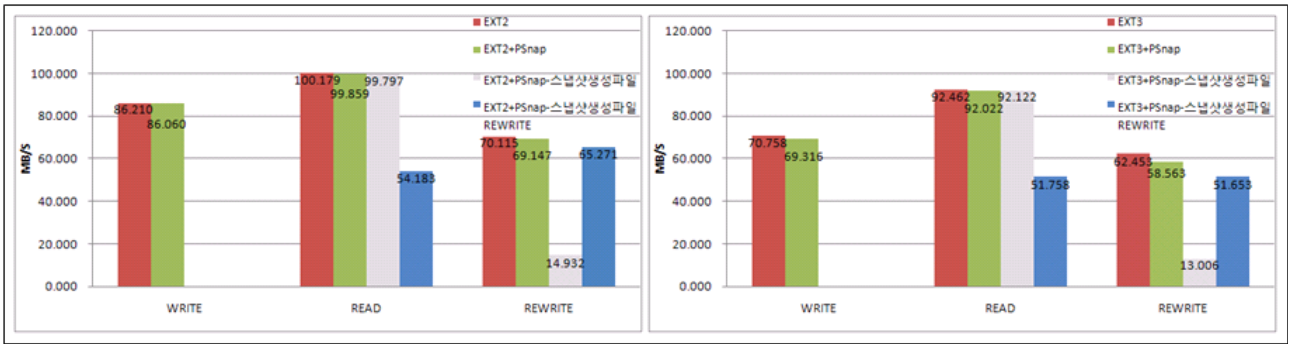

그림 3. EXT2+PSNAP과 EXT3+PSNAP의 성능 측정 결과

두 그래프는 유사한 형태를 가지는데 EXT2 파일시스템과 EXT3 파일시스템이 데이터 블록을 처리하는 방법은 다르지만 데이터 블록을 관리하는 방식이 동일하기 때문이다. 따라서 이후부터 그래프에 대한 설명은 EXT2 파일시스템과 EXT3 파일시스템을 원본 파일시스템으로 EXT2에 PSnap을 추가한 파일시스템과 EXT3에 PSnap을 추가한 파일시스템을 PSnap 파일시스템으로 표기한다. WRITE, READ, REWRITE 항목의 처음 두 막대는 일반 파일시스템과 스냅샷 파일시스템에서 스냅샷이 생성되지 않은 파일에 대한 성능을 나타낸다. PSnap 파일시스템에서 스냅샷이 생성되지 않은 파일의 WRITE, READ, REWRITE 성능은 그래프에서 보이는 것과 같이 일반 파일시스템과 유사한 성능을 가진다. READ 항목의 세 번째와 네 번째 막대는 PSnap 파일시스템에서 스냅샷이 생성된 파일과 스냅샷 생성 후 REWRITE에 의해서 스냅샷 처리가 발생한 파일에 대한 성능이다. 스냅샷이 생성된 파일에 대한 READ는 스냅샷 처리가 발생하기 전까지는 스냅샷이 생성되지 않은 파일과 유사한 성능을 보였지만 스냅샷 처리가 발생한 이후에는 Redirect-on-Write 스냅샷에 의해서 데이터 블록 단편화가 발생하여 성능의 저하를 보였다. REWRITE의 세 번째와 네 번째 막대는 스냅샷이 생성된 파일에 처음 발생하는 REWRITE와 그 이후에 발생하는 REWRITE의 성능이다. 첫 번째 REWRITE에서는 스냅샷 처리를 위해서 새로운 데이터 블록을 할당받아야 하기 때문에 급격한 성능 저하가 발생하였고 그 이후의 REWRITE는 데이터 블록 단편화에 의한 성능 저하를 보였다.

PSnap은 모든 파일시스템이 공통적으로 이용 가능한 스냅샷 정보 관리 기법을 사용하고 파일시스템에 종속적인 부분과 비종속적인 부분을 분리함으로서 파일시스템 스냅샷 기능을 제공하지 않는 파일시스템들이 공통적인 방법으로 파일시스템에 스냅샷 기능을 추가할 수 있게 한다. 또한 원본 파일시스템과의 호환성을 유지하기 때문에 기존에 운용중인 파일시스템에 스냅샷 기능을 추가하는 것도 가능하다. 앞으로 EXT2 파일시스템과 EXT3 파일시스템 이외에 다른 파일시스템에서 PSnap을 사용할 수 있도록 PSnap_Interface를 추가하고 Redirect-on-Write 스냅샷 처리 과정에서 발생하는 데이터 블록 단편화 현상을 최소화하기 위해서 개선된 데이터 블록 할당 기법을 연구할 예정이다.