

# 델타 기술을 기반으로 한 온라인 스토리지 마이그레이션 방법

## Online Migration Method for Storage based on Delta Technology

김진수, 송석일  
한국교통대학교

Jinsoo Kim, Seokil Song  
Korea National University of Transportation

### 요약

이 논문에서는 스토리지 교체, 복제, 데이터 센터 이전 등 다양한 응용에서 꼭 필요한 온라인 스토리지 마이그레이션 방법을 제안한다. 제안하는 온라인 스토리지 마이그레이션 방법은 전송속도를 높이기 위해서 전송 대상 파일들의 익스텐트 맵을 생성하고 물리주소에 따라 순차IO를 수행하여 파일들을 읽고 이를 타겟(target)으로 전송한다. 또한, 전송 중에 발생하는 소스(source) 측의 변경에 대해서 추적을 하고 이를 로그레코드로 기록하여 전송중의 변경을 타겟측에 반영할 수 있다. 이 논문에서는 제안하는 온라인 마이그레이션 방법을 실제로 구현하고 소스와 타겟의 일치성과 온라인 마이그레이션 수행 속도를 측정하는 실험을 수행하였다.

## I. 서론

온라인 마이그레이션(Online Migration)은 서비스를 중단하지 않고 스토리지를 원격에 복제(Replication)하거나 스토리지 교체 등을 수행하는데 있어 중요한 연산이다. 온라인 마이그레이션을 수행하기 위해서는 소스(Source) 스토리지의 내용이 지속적으로 변경하는 상황에서 소스 스토리지의 데이터를 일관성 있게 타겟(Target) 스토리지에 전송해야 한다.

대표적인 온라인 마이그레이션이 가능한 도구로는 RSync[1]가 있다. RSync는 리눅스 운영체제에 기본적으로 설치되어 있는 도구이며, 소스 디렉토리를 타겟 디렉토리에 동기화 시키는 기능을 수행한다. 동기화를 위해, RSync는 소스 디렉토리와 타겟 디렉토리의 파일들을 읽어 서 시그니처(Signature)를 생성하고 시그니처를 비교하여 델타(Delta)를 생성한다. 최종적으로 델타를 타겟 디렉토리에 적용하여 소스 디렉토리와 타겟 디렉토리를 일치시킨다.

RSync는 소스와 타겟 디렉토리간 동기화를 수행할 수 있지만, 이를 이용해서 온라인 마이그레이션을 수행하기 위해서는 지속적인 동기화를 수행해야 한다. 또한, RSync는 변경되는 파일의 시맨틱을 파악하지 못하기 때문에 이름이 변경된 파일이나 이동된 파일을 신규 파일로 인식할 수 있고, 이에 따라 재전송 동작을 수행하게 되므로 비효율성이 존재한다.

Aspera Sync[2]는 소스 디렉토리에 대한 스냅샷을 생성하여 소스 스냅샷 대비 타겟의 변경 내용을 추적하여 타겟 디렉토리에 대한 동기화를 수행할 수 있다. Aspera Sync는 스토리지의 변경이 발생할 때 마다 동작하여 동

기화를 수행할 수 있다.

기존 방법들(RSync, Aspera Sync 등)은 모두 타겟 디렉토리에 파일이 이미 있을 때 이를 동기화하는데는 적합하지만 최초로 소스 디렉토리의 파일들을 타겟 디렉토리로 이동하는데는 적합하지 않다. 최초로 소스 디렉토리의 파일들을 타겟 디렉토리로 이동시키기 위해서는 모든 파일에 대한 원격 파일 전송이 수행되어야 하며 전송 시 소스 디렉토리에 가해지는 변경을 타겟에 반영해야 한다.

기존 방법들을 이용해서 이를 수행하기 위해서는 먼저 소스 디렉토리의 파일들을 어떤 방법을 통해서 타겟으로 전송한 후 타겟과 소스의 차이를 찾아내어 다시 동기화를 시키는 과정이 수행되어야 한다. 변경이 지속되는 상황에서는 소스와 타겟을 완전히 일치시키기 위해서는 반복적인 동기화 과정이 필요하며 상황에 따라서는 끝나지 않을 수도 있다.

이 논문에서는 파일 시스템의 변경을 실시간으로 추적하는 델타 기술과 파일의 물리적 주소인 익스텐트 맵(extent map)을 이용하여 효과적으로 소스의 파일들을 타겟으로 전송하는 온라인 마이그레이션 기법을 제안한다. 제안하는 방법은 파일의 익스텐트 맵을 통해서 연속 읽기를 가능하게 하며 파일의 변경을 변경 연산단위로 추적하여 변경분을 빠르게 추적하여 전송이 가능하다.

## II. 제안하는 온라인 마이그레이션 방법

온라인 마이그레이션의 첫 번째 단계는 디렉토리 스냅샷의 생성이다. 디렉토리 스냅샷은 소스 디렉토리의 파일들에 대한 디렉토리 구조와 각 디렉토리에 포함된 파

일들의 하드링크들로 구성된다. 다음 단계는 생성된 디렉토리 스냅샷의 모든 하드링크들을 스캔하여 파일들에 대한 익스텐트 맵을 생성한다. 익스텐트 맵은 각 파일을 구성하는 익스텐트들의 하드디스크 상의 물리 주소를 나타낸다.

ID	Fullpath	Extent#	Logical Address	Physical Address	Length	Flags	Update
1	/a/a.txt	0	0	4	2	0000	
2	/a/a.txt	1	2	1	1	0001	
3	/b/b.txt	0	0	2	2	0000	U
4	/b/b.txt	1	2	6	1	0001	
5	/a/d/d.txt	0	0	0	1	0001	U

▶▶ 그림 1. 익스텐트 맵

익스텐트 맵의 엔트리들을 물리주소 (Physical Address) 를 기준으로 정렬한다. 익스텐트 맵의 정렬된 엔트리들을 순차적으로 읽어서 타겟으로 전송한다. 이와 동시에 파일시스템의 모든 변경연산을 추적하여 파일의 내용변경은 익스텐트 맵에 기록하고 구조변경 (디렉토리 변경, 파일 생성, 이름 변경, 삭제 등) 에 대해서는 로그 레코드를 생성하여 저장한다. 익스텐트 맵의 Update 컬럼은 추적한 변경연산이 변경인 경우 U (Update) 로 표시한다. 변경연산이 추가 (Append) 일 경우 엔트리를 추가하고 Update 컬럼을 A로한다. 변경연산이 축소 (Shrink) 일 경우 엔트리의 길이 (Length)를 조정한다.

전체 엔트리들을 타겟으로 전송하면 Update 컬럼이 U, A 인 엔트리들의 전체 엔트리 대비 비율을 계산한다. U, A 인 엔트리들의 비율이 주어진 값보다 작으면 해당 엔트리들에 대해서 타겟으로 전송을 수행하고 동시에 파일 변경에 대한 로그레코드를 생성하여 기록한다. 엔트리에 대한 전송이 끝나면 로그 레코드를 전송하여 이를 타겟에 반영하고 마이그레이션을 종료한다.

U, A 인 엔트리들의 비율이 주어진 값보다 크면 변경된 익스텐트가 매우 많다는 뜻이며 이 경우에는 처음 엔트리들을 전송하는 것처럼 U, A 인 엔트리들을 타겟으로 전송한다.

### III. 성능평가

이 논문에서는 제안하는 온라인 마이그레이션 방법의 성능을 입증하기 위해서 실제 구현을 하고 실험을 수행하였다. 실험은 온라인 마이그레이션의 수행시간과 온라인 마이그레이션 도중 소스에 변경 연산이 수행되더라도 이를 타겟에 일관성 있게 반영하는지를 측정하였다.

실험에는 500GB 의 랜덤하게 생성된 파일들을 사용하였으며 SATA 하드디스크와 1Gbps 의 대역폭을 갖는 네트워크 환경에서 수행하였다. 온라인 마이그레이션 도중에 소스에 일정량의 변경을 가했으며 변경한 내용이 타겟에 모두 반영되었는지 측정하였다. 실험결과 소스와

타겟이 정확히 일치함을 알 수 있었다. 또한, 전송 시간의 효율성을 보기 위해서 똑같은 데이터에 대한 scp 명령어 수행 결과와 비교하였다. 표 1에서 보는 것과 같이 제안하는 온라인 마이그레이션 방법이 scp에 비해서 2배 이상 빠른 것을 알 수 있었다. 속도가 개선된 이유는 익스텐트들을 물리주소로 정렬하여 하드디스크에서 읽어서 전송하므로 랜덤 IO가 최소화 되고 대부분의 읽기 연산이 순차 IO로 처리 되기 때문이다.

표 1. scp 명령어와의 전송속도 비교 결과

구분	소요시간 (분)
scp	131
제안하는 온라인 마이그레이션	56

### IV. 결론

이 논문에서는 스토리지 교체, 복제, 데이터 센터 이전 등 다양한 응용에서 꼭 필요한 온라인 스토리지 마이그레이션 방법을 제안하였다. 제안하는 온라인 스토리지 마이그레이션 방법은 전송속도를 높이기 위해서 전송 대상 파일들의 익스텐트 맵을 생성하고 물리주소에 따라 순차IO를 수행하여 파일들을 읽고 이를 타겟 (target)으로 전송하도록 하였다. 또한, 전송중에 발생하는 소스 (source) 측의 변경에 대해서 추적을 하고 이를 로그레코드로 기록하여 전송중의 변경을 타겟측에 반영할수 있게 하였다. 실험을 통해서 온라인 마이그레이션이 수행된후 소스와 타겟 스토리지의 일치성을 비교한 결과 정확히 일치함을 볼수 있었다. 또한, 온라인 마이그레이션 수행속도를 측정하는 실험에서도 기존의 리눅스 명령어인 scp에 비해서 2배 이상 수행속도가 더 빠름을 확인하였다.

### V. 감사의 글

본 연구는 중소기업청의 창업성장기술개발사업의 일환으로 수행하였음. [S2531573, 델타기술 기반 stop-free 다중 스냅샷 기술 개발]

### ■ 참고 문헌 ■

- [1] <https://rsync.samba.org/>
- [2] <http://asperasoft.com/software/synchronization/>