

# Small Random Write 환경에서의 NFS 와 iSCSI 에 대한 연구

최찬호, 김신규, 엄현상, 엄현영  
서울대학교 컴퓨터공학과

e-mail : {chchoi, sgkim, hseom, yeom}@dcslab.snu.ac.kr

## A Study on NFS and iSCSI in Small Random Write Intensive Applications

Chanho Choi, Shin-gyu Kim, Hyeonsang Eom, H Y. Yeom  
School of Computer Science and Engineering Seoul National University

### 요 약

클라우드 시스템에서 사용되는 스토리지 서버는 다수의 워크로드가 혼재됨으로 인하여 상당수의 쓰기 연산이 Small Random 한 특성을 가지게 된다. 좀 더 높은 성능을 위해서 이런 특성에 적합한 스토리지 서버를 구축하는 것이 필요하며 이를 위해 본 논문에서는 자주 사용되는 스토리지 프로토콜인 NFS 와 iSCSI 를 비교하여 어떤 쪽이 Small Random Write 에 더 적합한지 실험을 통해 알아 보았다. 결과적으로 Small Random Write 들은 캐시에 합쳐지는 효과에 상당한 영향을 받으며, 이런 캐시 효과가 더 효율적인 것은 iSCSI 임을 확인하였다.

### 1. 서론

최근 클라우드 서비스가 활성화됨에 따라서 스토리지 서버에 대한 관심도 높아지고 있다. 스토리지 서버는 다른 시스템에 네트워크 스토리지 장치를 제공하여 스토리지 자원을 공유함으로써 Live Migration 과 같은 기능이 활성화 될 수 있도록 돕는다.[2] 이런 네트워크 스토리지 서비스 방식은 다소 다른 특징들을 가지고 있는데 대표적으로 NFS 와 iSCSI 가 많이 활용된다.[1]

위와 같이 클라우드 시스템이 스토리지 서버를 사용할 경우 다수의 워크로드의 요청이 섞여서 Random Write 요청이 디스크에 보내지게 되며, 이는 Sequential Write 에 비해 크게 성능이 떨어지게 된다.[3] 하드 디스크를 사용하는 이상 이는 감수해야 할 단점이지만 스토리지 서비스 방식에 따라서도 성능이 달라지므로 이를 최소화하는 것이 바람직하다.

따라서 본 논문에서는 Small Random Write 에 대해서 NFS[5]와 iSCSI[4]의 성능 차이를 알아보는 실험을 하고 분석하였다..

### 2. NFS vs. iSCSI

NFS 는 설정이 간단하고 사용이 편리하다는 면에서 자주 사용되던 방식이다. 서버 측에서 특정 폴더를 NFS 서버로 열어두면 클라이언트 측에서 이 폴더를 자신의 특정 위치로 마운트해서 사용하는 메커니즘을 가지고 있다. 따라서 서버 측의 파일 시스템을 그대로 사용하게 되며 클라이언트 측에서는 실제 파일 시

스템이 어떤 것인지 모르는 상태로 NFS 를 사용하게 된다. 이 시스템에 클라이언트에서 쓰기 명령이 들어 오게 되면 NFS 클라이언트에서 캐싱을 해두었다가 특정 조건이 만족되면 그 동안 버퍼에 캐싱해두었던 정보들을 서버로 전송하게 된다.

반면에 iSCSI 는 기존의 SCSI 프로토콜을 네트워크를 통해서 이용할 수 있도록 만든 것으로써 설정은 다소 복잡하지만 보안성이나 성능 측면에서 우수하다고 알려져 있는 방식이다. iSCSI 는 target 측에서 일정 디스크 볼륨을 지정해서 iSCSI target demon 을 이용해서 서비스하고, initiator 측에서는 이 볼륨을 새로운 블록 디바이스로 인식할 수 있다. NFS 와는 다르게 이렇게 인식된 iSCSI target 볼륨에 자유롭게 파일시스템을 설치하고 마운트하여 사용할 수 있게 된다. iSCSI 도 역시 NFS 와 같이 쓰기 요청에서 캐싱을 하게 되는데 이는 설치한 파일 시스템에서 이용하는 캐싱 정책에 의존한다.

클라우드 시스템에서는 위와 같은 대표적인 스토리지 프로토콜들을 사용하여 스토리지 서버를 구성하게 되는데 다수의 가상 머신들이 스토리지 서버를 사용하게 되면 자연스럽게 Random I/O 가 발생하게 된다. 특히 메타 데이터에 접근하는 횟수가 늘어나게 되므로 대부분의 I/O 가 Small Random 의 특성을 가지게 된다. 따라서 클라우드 시스템에 적합한 스토리지 프로토콜이 어떤 것인지 확인하기 위해 Small Random I/O 벤치마크를 통해 실험함으로써 이를 알아볼 수 있다.

### 3. Benchmark

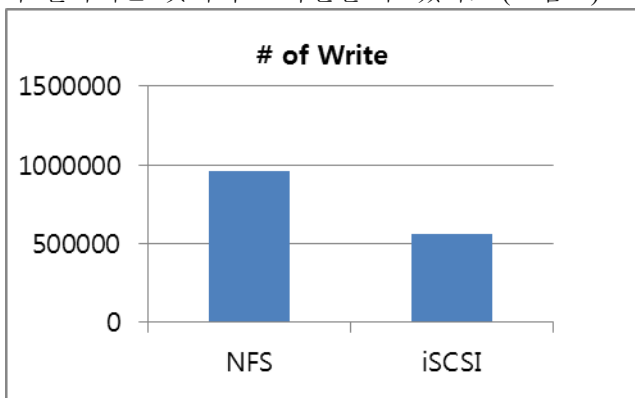
실험에 사용할 벤치마크로는 postmark[6]를 선정하였다. postmark 는 3 개의 구간으로 나뉘어져 있는데, 첫 번째에서는 생성 구간으로 옵션으로 주어진 폴더에 프로세싱에 사용될 다수의 폴더와 파일들을 생성한다. 두 번째 프로세스 구간에서는 옵션으로 주어진 트랜잭션과 반복 횟수만큼 지정된 사이즈에 맞게 각 폴더, 파일에 추가하는 쓰기 연산을 반복적으로 수행하게 된다. 이후 마지막 삭제 구간에서는 그 동안 수행된 폴더와 파일을 삭제하고 첫 번째 구간 수행 전으로 돌아가게 된다. 이 postmark 에 사용되는 대다수의 I/O 는 Small Random 의 특성을 가지게 되며 이는 우리가 하고자 하는 실험에 적합하다.

### 4. 실험 환경

- ✓ Host Machine
  - Intel Core2 Quad CPU Q9550 2.83GHz
  - 8GB RAM
  - Xen 4.0.1, Linux kernel 2.6.32.25
- ✓ Storage
  - 4 TB SATA 5900RPM Hard Disk
  - RAID 0

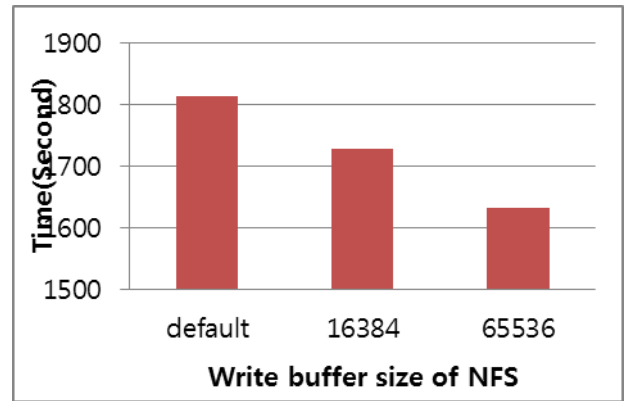
### 5. 실험 결과 및 분석

두 개의 다른 환경에서 실험하였을 때 NFS 에서의 postmark 의 수행시간이 iSCSI 에서 보다 약 1.5 배 가량 느리게 나온 것을 확인할 수 있었다. 이 같은 차이는 실제 디스크가 있는 스토리지 서버에서 측정하였을 때 볼 수 있는 쓰기 연산의 숫자에서 나온 것임을 알 수 있었다. (그림 1) 실제로 디스크에 적용되는 쓰기 연산의 수가 약 1.7 배 가량 많아지는 것에 비례하여 실행속도도 느려진 것을 확인하였다. 이러한 쓰기 연산의 수의 차이는 I/O Path 중에 얼마나 캐싱이 되는지에 따라서 달라질 수 있다. postmark 에 프로세스 대부분이 파일에 append 하는 연산으로 이루어져있고, 이러한 Small Write 들이 캐시에서 겹쳐지거나 이어져서 한번에 디스크로 보내진다면 수행 속도가 더 빨라지게 된다. 이와 같은 현상은 NFS 에서 Write Buffer Size 를 크게 설정하였을 때 프로그램의 수행속도가 빨라지는 것에서도 확인할 수 있다. (그림 2)



(그림 1) NFS 와 iSCSI 에서의 쓰기 연산 횟수

NFS 에서는 클라이언트에서 한번 캐싱이 일어난 후 서버로 보내져 파일시스템 캐시를 거쳐서 쓰기 연산이 수행된다. 반면 iSCSI 는 파일시스템 자체의 캐시에 크게 의존한다. 실험 결과로 미루어볼 때 NFS 와 같이 클라이언트에서 캐싱하는 방식보다 iSCSI 와 같은 방식이 더 효율적인 것을 알 수 있다.



(그림 2) NFS Write Buffer Size 에 따른 프로그램 수행시간의 변화

### 6. 결론

우리는 실제 클라우드 시스템 환경에서 Small Random Write 가 중요시 됨에 따라 자주 쓰이는 스토리지 프로토콜인 NFS 와 iSCSI 가 얼마나 이에 적합한지 알아보는 연구를 진행하였다. Small Random Write 는 캐시 효과에 의해서 디스크로 전달되는 쓰기 연산의 수를 효과적으로 줄일 수 있다는 것을 알 수 있었으며, 이러한 차이에 의해서 NFS 에서보다 iSCSI 에서 더욱 캐싱이 효과적으로 일어나며 이와 같은 면에서 유추할 때 NFS 보다 iSCSI 가 클라우드 시스템에 사용되는 스토리지 프로토콜로 적합하다고 볼 수 있다.

#### 감사의 글

이 논문은 2011 년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임. (No. 2011-0020521) 이 연구를 위해 연구 장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사 드립니다.

#### 참고문헌

- [1] W. Wang, H. Yeo, Y. Zhu, and T. Chong. "Design and development of Ethernet-based storage area network protocol". In Proceedings of the 12th IEEE International Conference on Networks (ICON), 2004.
- [2] W. Zeng, Y. Zhao, K. Ou, W. Song, "Research on Cloud storage architecture and key technologies," ICIS 2009.
- [3] S. Sivathanu, L. Liu, M. Yiduo, and X. Pu. "Storage management in virtualized cloud environment." Proc. IEEE 3rd Int. Conf. on Cloud Computing, pp. 204-211, 2010.

- [4] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, and E. Zeidner. Internet small computer systems interface(iSCSI). RFC 3720 (Proposed Standard), April 2004. URL
- [5] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, and D. Noveck. NFS version 4 protocol. RFC 3010, Network Working Group, December 2000.
- [6] J. Katcher. "Postmark: a new file system benchmark". Technical report TR-3022, Network Appliances, Oct 1997.