소규모 클러스터 시스템에서의 spNFS 성능 평가

차광호, 김성호, 이상동
한국과학기술정보연구원 슈퍼컴퓨팅센터
e-mail : khocha@kisti.re.kr

Performance evaluation of spNFS on a small scale cluster system

Kwangho Cha, Sungho Kim, Sangdong Lee
Supercomputing Center, Korea Institute of Science and Technology Information

요 약

분산되어 있는 스토리지 자원을 하나의 클러스터로 구성하여 분산 파일 시스템으로 구성하고자 하는 경우, 기존의 네트워크 파일 시스템만을 이용하기에는 여러 가지 제약이 존재한다. 특히 Parallel Striped Access 는 IO 데이터를 스토리지에 나누어 분산시키고 클라이언트가 직접 접근하는 방식으로 병렬 파일 시스템과 같은 HPC 용 특수 파일 시스템에서는 이미 사용되는 기법이나, 일반적인 시스템을 대상으로 한 표준안의 부재가 제약이 된다. pNFS(Parallel NFS)는 이러한 문제를 해결하기 위하여 제시되는 새로운 NFS 기술이다. 본 연구에서는 pNFS 의 연구 동향과 더불어 소규모 클러스터 시스템에서 나타나는 성능적 특징을 조사하였다.

1. 서론

IT 기술의 발달은 다양한 사용자층을 만들어 내면서 다양한 사용자 요구 사항을 만들어 내고 파생되는 데이터양도 날로 급증하고 있다. 이러한 시대적 상황을 반영하여 스토리지 및 파일 시스템 역시 다양한 형태로 변화를 꾀하고 있다. 네트워크 파일 시스템으로 대표적인 NFS 역시 Version 4 가 발표되면서 새로운 기능들을 제공하게 되었다[1]. 특히 pNFS(Parallel NFS)는 NFSv4 의 확장 기능으로 클라이언트가 스토리지나 데이터 서버에 직접 접근할 수 있도록 하는 것을 주요 골자로 하고 있다[2,3].

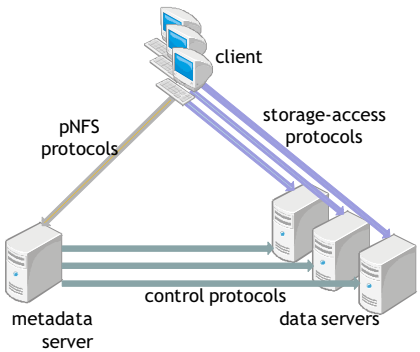
이러한 개념은 OSD(Object-based Storage Device) 기반 파일 시스템에서 이미 제안된 개념이지만 HPC 가 아니 일반 IT 분야에서도 필요성이 제기되어 표준화 과정을 거쳐 pNFS 의 주요 기능으로 정의되었다[4].

본 논문에서는 pNFS 의 연구 동향을 살펴보고 현재 프로토타입으로 제공되는 spNFS(Simple pNFS)의 성능을 소규모 클러스터 시스템에서의 확인하였다. 특히 스트라이핑 크기를 변경하면서 전체 성능의 변화를 살펴 보았다.

2. pNFS(Parallel NFS)

2.1. pNFS(Parallel NFS)의 기본 구조

pNFS 는 NFSv4.1 의 확장 기능으로 클라이언트의 스토리지에 대한 직접적이고 병렬화된 접근을 허용하여 기존의 NAS 시스템이 갖던 확장성 문제를 해결한 표준 프로토콜이다. pNFS 는 파일서버, OSD 및 기존의 블록기반 스토리지의 다양한 백엔드 스토리지 시스템을 지원한다[1,2,3]. Univ. of Michigan(CITI), EMC, IBM, NetApp, SUN, Panasas 등이 pNFS 커뮤니티에 참여하고 있으며 pNFS 의 기본 구성은 그림 1 과 같다.



(그림 1) pNFS 의 기본 구성

pNFS 는 클라이언트와 메타데이터 서버간 정보전달에 사용되는 pNFS 프로토콜과 클라이언트와 데이터 서버(또는 스토리지)간의 데이터 전송에 관여하는 스토리지 접근 프로토콜로 구성된다. pNFS 프로토콜은 NFSv4 에 일부 오퍼레이션이 추가된 형태로서 Layout 과 데이터 서버의 위치를 질의하고 확인하기 위하여 사용된다. 스토리지 접근 프로토콜은 NFS 나 SCSI 등이 사용되며 클라이언트가 데이터 서버에 어떻게 접근하는가를 기술한다. 메타 데이터 서버에 존재하는 각 파일에 대한 메타 데이터는 파일의 위치, 스트라이핑 파라메터, 그 외 토큰등이 포함된 레이아웃과 디렉토리에 대한 파라메터 및 파일 어트리뷰트등을 보유하고 있다. 파일의 삭제등과 같은 오퍼레이션은 클라이언트가 메타데이터 서버를 통하

여 수행하게 되며 이때 콘트롤 프로토콜을 통하여 동기화를 수행하게 된다. 현재까지는 분산 파일 시스템의 특성을 배려하고 유연성을 고려하여 콘트롤 프로토콜에 대해서는 특별하게 제한하고 있지 않다[2,3,5,6].

2.2. Layout Type

가. File Layout

스토리지 서버에 분산 파일 시스템이 설치되어 있고, 클라이언트는 각 스토리지 서버의 스트라이프된 파일 시스템 데이터를 접근하는 시스템에서 사용된다. 기본적으로는 NFSv4 를 스토리지 접근 프로토콜로 권장하고 있으나 일부 프로토 타입의 경우, 스토리지 서버에 설치된 분산 파일 시스템을 접근할 수 있는 프로토콜이 사용되기도 한다. 레이아웃 정보에는 데이터 서버 목록, 스트라이프 사이즈, 각 NFS 핸들등이 포함된다. 이러한 File Layout 정보는 간결하고 파일이 갱신되더라도 변경되지 않기 때문에 해당 파일이 광범위하게 공유되더라도 동기화 오버헤드없이 많은 클라이언트에서 캐쉥될 수 있다[2,3].

나. Block Layout

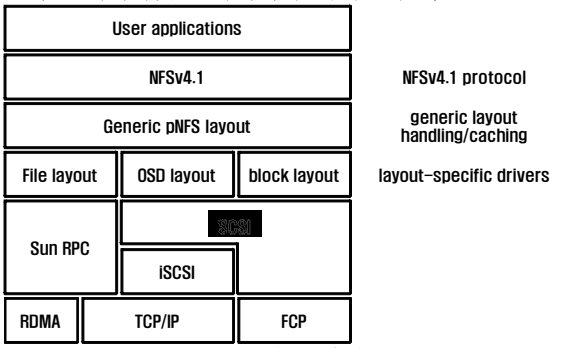
클라이언트가 블록/볼륨 디바이스에 접근하는 기능을 목표로 설계되고 있다. 이때 사용되는 프로토콜로는 iSCSI 나 FC(T11 표준)의 사용을 기본으로 한다. 레이아웃 정보에 실제 스토리지 블록이 사용되기 때문에 파일의 갱신시 메타 데이터의 변경이 필요하므로 File Layout 과 달리 메타 데이터의 빈번한 갱신이 요구된다[2,3,5]

다. Object Layout

클라이언트가 OSD 에 접근하는 기능을 목표로 설계되었다. SCSI Object 프로토콜(T10)을 이용하여 스토리지에 접근하고 레이아웃의 경우 Object ID 를 사용하나, File Layout 과 유사한 특징을 갖는다[2,3,6].

2.3. pNFS 클라이언트

그림 2 는 pNFS 클라이언트의 구조를 보여 주고 있다. NFSv4.1 하부에 레이아웃 캐쉥과 같은 일반적인 레이아웃 드라이버가 위치하고 레이아웃 타입별로 세분화된 레이아웃 드라이버가 위치한다[2,3].



(그림 2) pNFS 의 클라이언트 구조

2.4. WAN 환경에서의 pNFS

pNFS 의 적용 분야를 확대시키는 다양한 연구가 있었는데 대용량 IO 이외에 소규모 IO 에서도 성능 저하가 나타나지 않도록 경계 값을 이용하는 하이브리드 형태의 pNFS 가 그 예이다[7,8,9]. 또한 최근에는 WAN 을 사이에 두고 원격지의 파일 시스템에 접근하는 수단으로 pNFS 를 사용한 사례가 보고되고 있어 향후, 사용 분야가 더욱 확대될 것으로 예상된다[10,11].

3. spNFS (Simple pNFS)

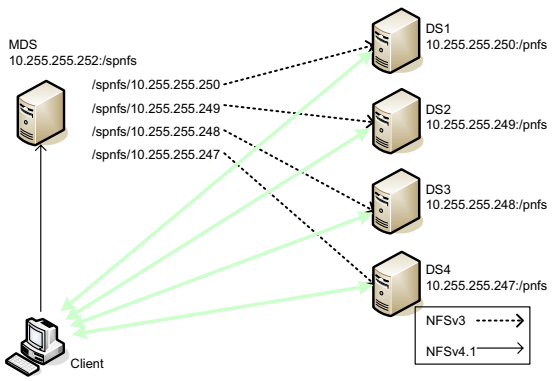
미시간대학교(Univ. of Michigan)에서 개발한 초기 pNFS 프로토타입은 파일 레이아웃을 지원하는 형태로서 PVFS(Parallel Virtual File System)를 스토리지 서버에 설치하고 pNFS 클라이언트는 PVFS 레이아웃 드라이버를 이용하여 스토리지 서버에 접근할 수 있도록 하였다. 실제로 동작하는 pNFS 프로토타입이라는 의미를 갖기는 하나 구조가 복잡하다는 단점이 있다[7,8].

이와 달리 단순한 구성을 지향하는 pNFS 프로토타입으로 NetApp.에서 개발중인 spNFS(Simple pNFS)가 있다. 간단한 파일 시스템 연산만을 제공하는 대신 pNFS 환경을 보다 쉽게 구성하는데 중점을 두고 있다[12].

표 1 은 spNFS 의 테스트 환경을 구축하는데 사용된 소규모 클러스터 시스템의 구성을 보여 주고 있으며 실제 환경 설정은 그림 3 과 같다.

<표 1> 테스트 베드 구성 요소

1 Client, 1 MDS, and 4 Data Servers	
CPU	Intel Pentium 4 XEON 2.8
Memory	1GB
HDD	80GB SATA
Networks	Fast Ethernet, Gigabit Ethernet
OS	Linux - 2.6.25 spnfs patch



(그림 3) spNFS 테스트 베드

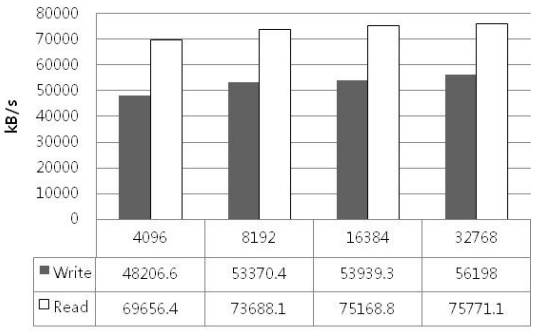
4. 성능 실험

본 장에서는 대표적인 IO 벤치마크 도구인 Bonnie 를 이용하여 spNFS 에서 수행한 성능 측정 결과를 설명한다.

4.1. Stripe 크기 변경에 따른 성능 변화

분산 파일 시스템이나 RAID 의 경우에 분할 및 저장 단위인 stripe 의 크기가 전체 성능에 영향을 주게 된다. 본 장에서는 stripe 크기의 변화에 따른 성능 변화를 살펴 보았다.

spNFS 를 사용하기 위해서는 파일 시스템 관련 데몬인 spnfsd 가 메타 데이터 서버에서 실행되어야 하고 이때 사용되는 spnfsd 용 환경 설정 파일을 수정하여 스트라이핑 크기(stripe-size)를 변경할 수 있다.



	4096	8192	16384	32768
■ Write	48206.6	53370.4	53939.3	56198
□ Read	69656.4	73688.1	75168.8	75771.1

(그림 4) Stripe-size 별 성능 변화

그림 4 는 스트라이핑 크기를 4KB 에서 32KB 까지 증가시키면서 bonnie 테스트의 블록 쓰기와 블록 읽기를 수행한 결과이다. 스트라이핑 크기가 증가할수록 향상된 결과를 보여 주었으나 8KB 일 때 개선 폭이 가장 크다는 것을 알 수 있다. 즉 8KB 이상의 스트라이핑 크기에서는 개선 폭이 크지 않아 이미 포화 구간에 들어선 것으로 볼 수 있다.

5. 결론

병렬화된 스토리지에 대한 직접 접근은 IO 데이터를 스토리지에 나누어 분산시키고 클라이언트가 직접 접근하는 방식으로 병렬 파일 시스템과 같은 HPC 용 특수 파일 시스템에는 이미 사용되는 기법이나, 일반적인 시스템을 대상으로 한 표준안의 부재가 제약이 된다. pNFS(Parallel NFS)는 이러한 문제를 해결하기 위하여 제시되는 새로운 NFS 기술이다. 본 연구에서는 pNFS 의 연구 동향과 더불어 소규모 클러스터 시스템에서의 나타내는 성능적 특징을 조사하였다.

스트라이핑 크기를 4KB 에서 32KB 까지 증가시키면서 성능을 조사한 결과, 스트라이핑 크기가 증가할수록 향상된 결과를 보여 주었으나 8KB 일 때 가장 효율적인 성능을 보여 주었다.

참고문헌

[1] Brian Pawlowski, Spencer Shepler, Carl Beame, Brent Callaghan, Michael Eisler, David Noveck, David Robinson, Robert Thurlow, "The NFS Version 4 Protocol," Proc. of the 2nd International System Administration and Networking Conference, 2000, http://www.nluug.nl/events/sane2000/papers/pawlowski.pdf

[2] Garth Goodson, Sai Susharla, Rahul Iyer, "Standardizing storage clusters," ACM Queue, Vol.5(6), pp. 20 ~ 27, 2007

[3] "NFSv4 Wiki," Retrieved Aug. 28, 2008, from http://wiki.linux-nfs.org/wiki/index.php/Main_Page

[4] David Nagle, Denis Serenyi, Abbie Matthews, "The Panasas ActiveScale Storage Cluster - Delivering Scalable High Bandwidth Storage," Proc. of the 2004 ACM/IEEE conference on Supercomputing, pp. 53, 2004.

[5] David L. Black, Stephen Fridella & jason Glasgow (EMC). pNFS Block/Volume Layout (draft 09), June 11, 2008. http://tools.ietf.org/pdf/draft-ietf-nfsv4-pnfs-block-09.pdf.

[6] B. Halevy, B. Welch, J. Zelenka, Panasas, Object-based pNFS Operations (draft 09), June 11, 2008. http://tools.ietf.org/pdf/draft-ietf-nfsv4-pnfs-obj-09.pdf

[7] Dean Hildebrand, Lee Ward, Peter Honeyman, "Large files, small writes, and pNFS," Proc. of the 20th ACM International Conference on Supercomputing, pp. 116 ~ 124, 2006.

[8] Dean Hildebrand, Peter Honeyman, "Direct-pNFS: scalable, transparent, and versatile access to parallel file systems," Proc. of the 16th ACM International Symposium on High performance Distributed Computing, pp. 199 ~ 208, 2007

[9] Dean Hildebrand, Peter Honeyman, "Exporting Storage Systems in a Scalable Manner with pNFS," Proc. of the 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies, pp. 18 ~ 27, 2005.

[10] Dean Hildebrand, Marc Eshel, Roger Haskin, Phil Andrews, Patricia Kovatch, John White, "Deploying pNFS Across the WAN: First Steps in HPC Grid Computing," The 9th LCI International Conference on High-Performance Clustered Computing, http://www.linuxclustersinstitute.org/conferences/archive/2008/PDF/Hildebrand_98265.pdf

[11] R. Ananthanarayanan, M. Eshel, R. Haskin, M. Naik, F. Schmuck, R. Tewari, "Panache: a parallel WAN cache for clustered filesystems," ACM SIGOPS Operating Systems Review, Vol.42(1), pp. 48 ~ 53, 2008.

[12] Dan Muntz, Mike Sager, Ricardo Labiaga, "A Simple pNFS Server," Retrieved Aug. 28, 2008, from http://www.connectathon.org/talks08/dmuntz-spnfs-cthon08.pdf