PVFS2 를 위한 파일 접근 로그 생성

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

Log Generation for File Access on PVFS2

Kwangho Cha, Hyeyoung Cho, Sungho Kim
Supercomputing Center
Korea Institute of Science and Technology Information

요 약

클러스터 시스템의 응용 분야가 다양화되고 복잡해짐에 따라, 대규모 클러스터 시스템을 보다 효율적으로 사용하기 위해서 실제 사용자의 이용 패턴을 예측할 수 있는 워크로드 분석의 필요성이 높아지고 있다. 워크로드 분석으로는 다양한 작업이 진행되는데 특히 파일 단위의 동적 접근 분석도 이에 포함된다. 본 논문에서는 실험용 병렬 파일 시스템으로 많이 보급된 PVFS2 에 파일 단위 접근 기록을 가능케하는 방안을 모색하고 이 기능의 활용 가능성을 살펴 보았다.

1. 서론

클러스터 시스템의 응용 분야가 다양화되고 복잡해짐에 따라, 대규모 클러스터 시스템을 보다 효율적으로 사용하기 위해서 실제 사용자의 이용 패턴을 예측할 수 있는 워크로드 분석이 필요하다. 특히 파일 시스템을 사용자의 이용 패턴에 따라 최적화함으로써 보다 나은 성능 향상을 가져올 수 있다[1,2]. 이러한 워크로드 분석에는 파일 단위 접근에 대한 동적 기록도 중요한 요소로 포함되고 있다[3].

PVFS(Parallel Virtual File System)는 클러스터 시스템의 확산과 함께 병렬 파일 시스템 연구에 활발이 이용되고 있으나 동적 워크로드를 분석하기 위한 별도의 도구를 제공하고 있지는 않다[4]. 본 연구에서는 PVFS2 가 제공하는 디버깅 도구에 기능을 추가하여 파일 단위 접근을 기록할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 PVFS2 의 현 상황과 한계점을 살펴보고, 3 장에서는 파일 단위 접근 로그 생성기의 구조를 설명한다. 4 장에서는 구현 환경, 생성된 예제 로그 및 활용 가능성을 설명하고, 마지막으로 5 장에서는 결론에 대하여 기술한다.

2. PVFS2 (Parallel Virtual File System ver. 2)

파일 시스템의 성능을 높이기 위하여 많은 병렬 파일 시스템이 취하는 방식이 RAID 0 처럼 파일을 쪼개서 (stripe) 서로 다른 볼륨에 저장하는 것이다. Clemson 대학에서 개발된 PVFS(Parallel Virtual File System) 역시 I/O 를 담당하는 복수 I/O 노드에 파일이 분산되어 저장되며 이에 대한 위치 정보를 관리하는 관리 노드가 별도로 존재한다. 이때 I/O 노드와 관리 노드의 역할을 하는 프로세스는 단일 노드 내에 위치 할 수 있다 [5,6].

PVFS 의 기능에 설치의 편리성, 이질적인 클러스터 시스템 지원 및 스토리지 및 네트워크를 위한 모듈화 기능 강화 등을 고려하여 추가 개발된 것이 PVFS 버전 2 이다. PVFS 가 TCP/IP 위주의 프로토콜을 사용하는 반면, PVFS2 는 클러스터 시스템용 고성능 네트워크인 미리넷과 인피니밴드를 위한 프로토콜의 지원도 포함하고 있다[4,7].

현재 PVFS2 는 IO 패턴 분석과 같은 동적인 로그 생성을 지원하지는 않고 있다. 다만 디버깅을 목적으로 하는 디버거를 부수적으로 제공하고 있다. 즉 그림 1 처럼 PVFS2 소스 코드중 원하는 위치에 디버깅문을 기술한뒤 그림 2 와 같이 런타임에 선택적으로 디버깅 로그를 생성할수 있다.

```
gossip_debug(GOSSIP_MKDIR_DEBUG,
        "creating dspace on coll_id %d\n",
        s_op->u.mkdir.fs_id);
```

(그림 1) PVFS2 의 예제 디버깅문

```
[c00]$ pvfs2-set-debugmask -m /mnt/pvfs2 all
Setting debugmask on all servers
[c00]$ pvfs2-set-debugmask -m /mnt/pvfs2 none
Setting debugmask on all servers
```

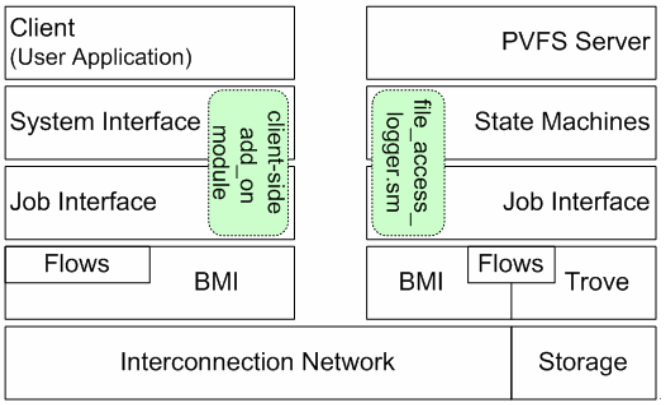
(그림 2) PVFS2 의 디버깅 실행 및 종료 예제

3. 파일 단위 접근 로그 생성기

앞서 설명한 것처럼 PVFS 는 단일 파일 데이터를 여러 IO 서버에 일정한 크기로 쪼개어서 저장하며 이러한 조각을 디스트리뷰션(Distribution)이라고 부른다[4]. PVFS2 는 이 디스트리뷰션 단위로 서비스를 수행하므로 각 서버들이 현재 서비스되고 있는 데이터의 파

일 차원의 정보를 유지하지 않는다. 이러한 이유에서 PVFS2 클라이언트는 VFS 에서 File Open 과 File Close 를 요청하더라고 서버측으로는 아무런 행동도 취하지 않고 있다. 물론 메타테디터 서버에 파일 관련 정보를 수정하기는 하나 각 IO 서버의 실제 작업 패턴을 파악하기 위해서는 메타데이터 서버의 도움없이 각 IO 서버들이 파일접근에 대한 로그를 기록할수 있어야 한다.

본 장에서는 이러한 현 상황에서 PVFS2 의 모든 서버들이 파일 차원의 정보를 수용하고 이를 기록하도록 하기 위해 수정 개발된 내용에 대하여 설명한다. 그림 3 은 파일 단위 접근 로그생성기의 구조를 보여준다. 현 PVFS2 의 구조에 클라이언트측에는 커널 모듈을 수정하였고, 서버측에는 상태기계(State Machine)와 이를 수용하기 위한 관련 루틴을 수정하였다.



(그림 3) File 접근 로그 생성기의 구조

3.1. 클라이언트 측 추가 모듈

PVFS2 클라이언트는 수행하고자 하는 IO 오퍼레이션에 대하여 서버측에 서비스를 요청하는 'pvfs2-client' 데몬과 VFS의 요청을 감지하여 'pvfs2-client' 데몬에게 이를 전달하는 'pvfs2.o' 모듈이 존재한다.

본 연구를 위하여 우선 'pvfs2.o' 모듈이 File Open 과 Close 호출시 이를 'pvfs2-client' 데몬에 전달하도록 하였다. 그후 'pvfs2-client' 데몬은 Open mode 와 같은 관련 정보와 서버에서 로그를 생성할 상태기계를 기술하여 각 서버에 서비스를 요청하게된다. 즉 기존의 클라이언트는 File Open 과 Close 시 서버에 아무런 요청도 수행하지 않았으나 파일 단위 접근 로그 생성기에서는 파일 개방 정보를 각 서버에 전달하도록 하였다.

3.1. 서버측 측 상태 기계

PVFS2 서버측에는 파일 단위 접근 기록을 전담할 상태 기계를 추가하였다. 별도의 상태 기계의 추가없이 로그의 생성도 가능하나, 향후 기록되어야 하는 정보량의 증가와 세부적인 통계 기능등이 추가될 가능성이 있으므로 로그 생성을 전담하기 위한 상태 기계를 새로 추가하였다.

추가된 상태 기계는 파일 접근에 대한 정보를 PVFS2 자체 디버거인 'gossip_debug'를 이용하여 비교적 간단하면서도 시간 기록면에서는 자세하게 정보를 기록할수 있도록 하였다.

4. 구현

본 장에서는 구현 환경과 결과에 대하여 간략하게 설명한다. 구축이 용이한 소규모 클러스터 시스템을 대상으로 개발 및 테스트가 진행되었다.

4.1. 구현 환경

표 1 은 구현에 사용된 소규모 클러스터 시스템의 하드웨어 및 소프트웨어 구성을 보여준다.

<표 1> 클러스터 시스템 구성요소

1 Client, 1 Management node, and 3 IODs	
CPU	Intel Pentium 4 XEON 2.8
Memory	1GB
HDD	80GB IDE
Networks	Fast Ethernet
OS	Linux - 2.4.22
PVFS	PVFS2 - 1.4.0

4.2. 실험 결과

표 2 는 클라이언트들에서 동시적으로 파일을 생성하면서 그동안 생성된 로그중 파일 접근에 대한 로그만을 기록한 것이다.

표와 같이 현재 개방하는 파일의 용도 및 해당 서버가 IO 서비스에 관여하고 있는 파일수, 해당 서버의 로컬 시간에 기반을 둔 접근 시각등이 기록되고 있다.

5. 결론

본 논문에서는 공개 소프트웨어로서 비교적 쉽게 구할 수 있는 병렬 파일 시스템인 PVFS2을 기반으로 사용자의 IO패턴을 분석하는데 있어서 중요한 파일 단위 접근에 대한 로그 생성기의 구현를 설명하였다.

PVFS2의 구조적인 특징으로 인하여 클라이언트와 서버 양측에 로깅을 담당할 모듈과 상태 기계를 추가하였다. 현재로서는 간단한 접근 정보를 기록하고 있으나 별도의 상태 기계를 이용하여 구현함으로서 향후 세부적인 기능의 확대 요구시 이를 반영하기 쉽도록하였다.

본 파일 접근 로그 생성기를 이용하여 사용자의 동적인 IO패턴에 대한 자료 수집이 가능할 것으로 예상되며 파일 단위 동시성 제어와 같은 부수적인 기능을 PVFS2에 구현하고자 하는 경우에도 이용 가능할 것으로 예상된다.

생성되는 정보를 보다 다양하게 확장하고 생성된 로그를 효과적으로 분석할수 있는 도구의 추가 개발을 계획중이며 대규모 클러스터에서 테스트를 통하여 오버헤드를 분석하고 이를 최소화하는 연구를 진행하고자 한다.

<표 2> 테스트 결과: 생성된 로그 예제

Meta Server: compute-0-2	[D 17:20:00.199599] [File open logger SM]: Write mode file open | Write Open 1 files. [D 17:20:00.942897] [File open logger SM]: Write mode file open | Write Open 2 files. [D 17:20:00.989541] [File open logger SM]: Write mode file close | Write Open 1 files. [D 17:20:02.326859] [File open logger SM]: Write mode file close | Write Open 0 files.
IO Server: compute-0-4	[D 17:24:38.048864] [File open logger SM]: Write mode file open | Write Open 1 files. [D 17:24:38.792172] [File open logger SM]: Write mode file open | Write Open 2 files. [D 17:24:38.838830] [File open logger SM]: Write mode file close | Write Open 1 files. [D 17:24:40.176128] [File open logger SM]: Write mode file close | Write Open 0 files.
IO Server: compute-0-5	[D 17:20:52.675363] [File open logger SM]: Write mode file open | Write Open 1 files. [D 17:20:53.418607] [File open logger SM]: Write mode file open | Write Open 2 files. [D 17:20:53.465240] [File open logger SM]: Write mode file close | Write Open 1 files. [D 17:20:54.802636] [File open logger SM]: Write mode file close | Write Open 0 files.
IO Server: compute-0-6	[D 17:24:27.894551] [File open logger SM]: Write mode file open | Write Open 1 files. [D 17:24:28.637798] [File open logger SM]: Write mode file open | Write Open 2 files. [D 17:24:28.684430] [File open logger SM]: Write mode file close | Write Open 1 files. [D 17:24:30.021842] [File open logger SM]: Write mode file close | Write Open 0 files.

참고문헌

[1] Drew Roselli, Jacob R. Lorch, and Thomas E. Anderson, "A comparison of file system workloads," Proc. of USENIX Annual Technical Conference, pp. 41~54, 2000

[2] E. Smirni, and D. A. Reed, "Workload characterization of input/output intensive parallel applications," Proc. of the Conference on Modeling Techniques and Tools for Computer Performance Evaluation, LNCS, Vol. 1245, pp. 169~180, Springer-Verlag, 1997

[3] Feng Wang, Qin Xin, Bo Hong, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Tyce T. McLarty, "File System Workload Analysis For Large Scale Scientific Computing Applications," Proc. of the 21st IEEE / 12th NASA Goddard Conference on Mass Storage Systems and Technologies, pp. 139~152, 2004

[4] Parallel Virtual File System version 2, http://www.pvfs.org/pvfs2

[5] John M. May, "Parallel I/O for High Performance Computing," Morgan Kaufmann, 2000

[6] W.B. Ligon III, and R.B.Ross, "Implementation and performance of a parallel file system for high performance distributed applications," Proc. of 5th IEEE International Symposium on High Performance Distributed Computing, pp 471 ~ 480, 1996

[7] Rob Latham, Neill Miller, Robert Ross, and Phil Carns, "A Next-Generation Parallel File System for Linux Clusters," Linux World, pp 56~59, Jan. 2004

[8] P. H. Carns, W. B. Ligon III, R. Ross, and P. Wyckoff, "BMI: a network abstraction layer for parallel I/O," Workshop on Communication Architecture for Clusters, Proc. of 19th IEEE International Parallel and Distributed Processing Symposium, 213a, 2005