HA-PVFS : 부분 복제를 통하여 데이터의 고가용성을 지원하는 클러스터 파일 시스템

심상만 김진석° 박성용
서강대학교 컴퓨터학과
yeuri@dcclab.sogang.ac.kr, {sabin°, parksy}@sogang.ac.kr

HA-PVFS : A Cluster File System for High Data Availability using Partial Replication

Sangman Sim, Jinseok Kim°, Sungyong Park
Dept. of Computer Science, Sogang University

요 약

최근 인터넷 서비스 공급 및 사용자의 폭발적인 증가로 클러스터 파일 시스템에 대한 관심이 높아졌다. 이에 많은 클러스터 파일 시스템이 개발되었지만, 데이터의 가용성 측면에서 문제가 있었다. 이 가용성 문제를 해결하기 위한 방안으로 패리티 서버를 사용하는 방법과, 전체 파일 시스템을 복제하는 방법이 소개 되었다. 그러나 이러한 방법들도 가용성을 완벽하게 지원하지 못하였으며, 추가적인 비용 또한 발생하였다. 본 논문에서는 파일 시스템에 대한 접근 패턴을 고려한 클러스터 파일 시스템, HA-PVFS를 제안한다. HA-PVFS는 파일 시스템에 존재하는 모든 파일이 아닌, 자주 사용되는 파일만을 접근 패턴을 분석하여 부분 복제함으로써 데이터의 가용성을 지원한다. 따라서 HA-PVFS를 통해 기존의 방법보다 낮은 비용으로 높은 가용성을 보장해 줄 수 있다.

1. 서 론

컴퓨터가 개발된 이래로 보다 나은 성능을 위한 노력은 끊임없이 이루어져 왔다. 특히 최근에는 단일 시스템의 성능 향상뿐만 아니라, 이미 존재하는 시스템들을 통합하여 마치 하나의 시스템처럼 활용하는 클러스터링에 관한 연구도 활발히 이루어지고 있다. 클러스터 시스템은 그 특성상 높은 유연성을 보이고, 확장성과 장애 허용성의 측면에서 장점을 지니고 있다. 이러한 특성이 응용된 분야 중 하나가 클러스터 파일 시스템이다.

한편, 클러스터 파일 시스템의 성능에 대한 요구가 높아져감에 따라 관련 연구가 진행되었고, 디스크의 성능 보다는 저장 서버의 입출력 대역폭이 전체 시스템의 성능을 좌우한다는 사실이 알려졌다. 따라서 여러 대의 저장 서버를 통해 입출력 대역을 증가시키는 연구가 진행되었고[1][2][3][4], 적은 구축비용과 높은 성능을 보이는 클러스터 파일 시스템이 개발되었다.

그러나 높은 성능을 구현하기 위해 파일을 분할한 뒤 여러 저장 공간에 저장함에 따라[4][5] 가용성의 문제가 나타났다. 분할 저장된 파일의 조각들 중 하나라도 문제가 발생하는 경우, 해당 파일을 정상적으로 읽을 수 없게 되는 것이다. 이 문제에 대한 과거의 해결 방안으로는, 전체 파일 시스템 복제 방식[4]과, RAID[6]에서 사용된 패리티 디스크 방식이 대표적이다. 그러나 이 방법들도 장애상황에 대한 완벽한 해결책이 되지 못했다.

본 논문에서는 파일 시스템에 대한 접근 패턴을 고려하여 부분 복제를 통해 데이터의 고가용성을 지원하는 클러스터 파일 시스템, HA-PVFS를 제안한다. HA-PVFS에서는 높은 빈도로 사용되는 파일들에 대해 중점적으로 가용성을 보장 해줌으로써, 기존의 방법에 비해 적은 비용으로 장애상황에 효과적으로 대처할 수 있다. HA-PVFS의 구현을 위해, 오픈 소스로 개발이 진행되고 있는 PVFS(Parallel Virtual File System)[7]을 확장하였다.

본 논문의 구성은 다음과 같다. 2장에서는 HA-PVFS의 기본 아이디어와 핵심 알고리즘을 살펴본다. 3장에서는 제안한 알고리즘의 타당성을 테스트 결과를 통해 살펴보며, 마지막으로 4장에서는 결론과 보완점에 대해 살펴본다.

2. 고가용성 지원을 위한 HA-PVFS

2.1. 기본 아이디어와 제반 가정

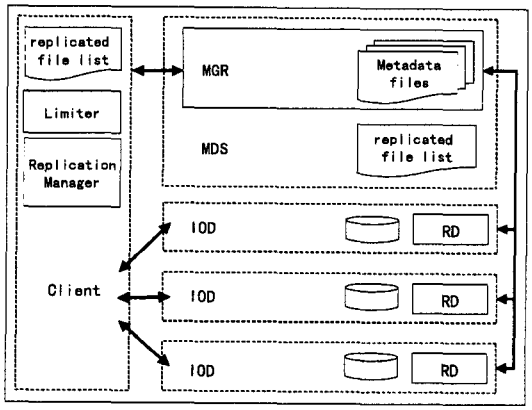
기존의 클러스터 파일 시스템들은 시스템의 전체 파일에 대해서 가용성을 지원하였었다. 그러나 실제 파일 시스템에서는 특정 파일들만이 높은 빈도로 사용되고, 특히 상업적 용도의 클러스터의 경우 그런 지역성이 더욱 심하다[8][9]. 따라서 본 논문에서는, 높은 빈도로 사용되는 파일들에 대해서만 가용성을 보장해 주는 것이 더 효율적이라고 판단하였다. 또한 대상 시스템이 높은 지역성을 보이는 인터넷 서비스용이라고 가정하고 알고리즘의 효율성을 위해 지역성은 지수분포를 따른다고 가정한다.

2.2. HA-PVFS 기본 구조

HA-PVFS의 기본 구조는 [그림 1]과 같다. HA-PVFS에서는 기본적으로 클라이언트에서 파일 시스템에 대한 접근 패턴을 분석하고 특정 파일이 자주 사용된다고 판단되면, 해당 파일의 복제를 메타 데이터 관리 서버(MDS)에 존재하는 MGR(Metadata server)에 요청하는 구조를 갖는다. 복제는 MGR에 의해서 이루어지며, 복제된 파일의 이름은 복제 파일 리스트(replicated file list)에 추가되어, 클라이언트와 메타 데이터 관리 서버에서 각각 관리된다. 본 논문에서는 접근 패턴을 분석하여 복제할 파일을 선정하는데 중요한 요소가 되는 복제 관리자와 한정자에 대해서 초점을 맞추며 다음은 이에 대한 설명이다.

2.3. 복제 관리자

복제 관리자는 가용성 지원 정책의 직접적인 명령권을 가진 주체이다. 복제 관리자는, 실시간으로 들어오는 접근 정보와 한정자가 지정해준 한계치(L)를 통해 파일 시스템에 대한 접근 패턴을 예측하고, 자주 사용되는 파일에 대해 복제를 요청한다.



[그림 1] HA-PVFS의 기본 구조

한편, 접근 패턴을 정확히 예측하기 위해 과거의 정보가 필요한데, 현실적인 비용을 고려해 부분적인(가장 최근의) 지역성만을 조사하고, 이것을 측정시간 Δt 라고 한다. Δt 는 다음과 같이 구한다.

$$\Delta t = \arg\min_{\Delta t}(L - \sum_{F} S(f, \Delta t))$$

<식 1>

여기서 함수 S는 Δt로 지역성을 조사했을 때 나타나는 파일 f의 접근 빈도를 말하고, F는 전체 파일 집합을 의미하며 f는 F의 원소를 의미한다.

2.4. 한정자

한정자는 복제 관리자에 의해 호출되며, 한계치 L값, 즉 복제할 용량을 재조정하는 일을 한다. 한편, 시스템에 대한 접근이 지역성을 보인다고 해도, 여러 요소에 의해 그 영역에 변동이 있다. 따라서 임계치 값 θ (0과 1사이의 실수)를 두고, 접근 빈도가 있는 파일들 중 θ 만큼의 파일들에 대해서만 가용성을 지원해 준다.

또한, 이 절의 시작 부분에서, 시스템 접근 패턴의 지역성은 지수분포를 따른다고 가정하였다. 지수분포는 하나의 λ 라는 인수로 정의되며, 추정 또한 매우 쉽다. 특히 지수분포의 판단 최우 추정기(Maximum Likelihood Estimator)는 측정된 표본의 평균이다. <식 2>는 지수분포의 MLE를 나타낸다.

$$\lambda = \frac{n}{\sum_{i=1}^{n} X_i}$$

<식 2>

여기서 X_i는 복제 파일 리스트 내의 파일들을 접근 빈도에 따라 정렬했을 때 해당 파일의 인덱스를 의미한다. 이렇게 구한 추정치를 통해서 예상되는 입력분포를 예측해 내면, θ 와의 관계를 통해서 목적하는 곳의 정보를 파악해 낼 수 있다. <식 3>은 이들의 관계식이다.

$$\theta = \int_0^{x_\theta} \lambda e^{-\lambda t} dt$$
$$= 1 - e^{-\lambda x_\theta}$$
$$x_\theta = -\frac{1}{\lambda} ln(1-\theta)$$

<식 3>

한편 한정자가 얻고자 하는 값은 한계치 L의 변동량 인데, 이 값을 찾아내는 것은 앞으로 복제 파일 리스트에 들어올 파일의 용량을 예측하는 일로 확률적인 문제가 된다. 이때 이 변위를 크게 한다면 그만큼 실제 분포를 추정해내는 시간이 줄어들게 되는 대신 안정값에 이르렀을 때의 진동이 우려되고, 반대로 아주 작게 변위를 설정 하는 경우, 적응시간이 길어지는 단점이 생긴다.

HA-PVFS에서는 몇 가지 실험을 통해 확률적 크기의 차이에 현재의 평균한계치를 곱해 줌으로써 변위를 취하는 중간 정도의 방법을 선택하였다. [식 4]가 이를 나타낸 식이다.

$$\Delta L = (F_r(x_c) - F_e(x_\theta))L$$
$$= (1 - f_r(x_c) - (1 - f_e(x_\theta)))L$$
$$= (f_e(x_\theta) - f_r(x_c))L$$

<식 4>

이때 함수F는 해당 분포의 CDF를 나타내고 아래 첨자로 표시된 e와 r은 추정한 분포와 실제 복제 파일 리스트에 존재하는 데이터만을 이용해서 얻게 된 분포 함수 이다. 다시 각각을 수식으로 정의 하면 다음과 같다.

$$f_e(x_\theta) = \lambda e^{-\lambda x_\theta}$$
$$f_r(x_c) = \frac{y}{N}$$

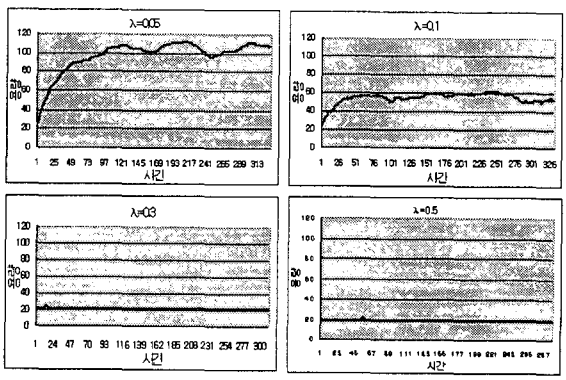
<식 5>

y는 복제 파일 리스트의 파일들을 접근 빈도에 따라 내림차순으로 정렬했을 때, 첫 번째 인덱스부터 더한 파일의 용량이 기존 한계치 L값의 θ 비율이 되는 시점의 파일 인덱스가 가지는 접근 빈도를 이야기 한다.

위의 값들을 모두 구하면 <식 4>에 의해서 한계치의 변위를 구하게 되고 기존 한계치에 더해줌으로써 새로운 한계치를 구하게 되어 시스템의 복제 정책이 갱신된다.

3. 성능 평가

본 장에서는 지금까지 설명한 HA-PVFS의 성능을 측정 하였다. 입출력을 담당하는 서버를 두 대 두고 한 대의 메타데이터 관리 서버를 두고 실험 환경을 구성하였다. 그리고 실제 입출력 작업 명령을 내리는 클라이언트를 분리하여 구성하였다. 본 논문이 제안한 가용성 지원은 실시간으로 정보를 분석해 변화에 적응하는 것이 특징이다. 따라서 다양한 접근 패턴에 대해 시스템이 입력분포를 정확하게 추정해 내는지 알아보았고, 추정한 입력분포(λ)를 통해 한계치를 어떻게 조정하는지 알아보았다. [그림 2]는 그 결과를 보여준다.



[그림 2] 다른 종류의 분포에 대한 한계치 변화

그래프에서 λ값이 작을수록 지역성이 작다는 의미이고, 여러 파일이 고르게 사용된다는 의미이다. 따라서 상대적으로 많은 수의 파일들에 대해서 가용성을 보장해 주어야 한다. 테스트 결과에서 λ 값이 0.05일 때, 한계치가 약 100메가로 가장 큰 것을 볼 수 있다.

또한 이 알고리즘의 목표는 사용되는 파일들 중 일정 비율(θ)의 파일에 가용성을 지원해 주는 것이다. 따라서 실험으로 측정된 복제 파일의 개수와 실제로 보장해 주어야 하는 파일의 개수와의 오차를 살펴보아야 한다. 여기서 오차란 실제 들어오는 접근의 분포를 안다고 가정했을 때 주어진 임계치(θ)에 의해 보장해 주어야 하는 파일의 수를 실험적으로 측정된 복제

파일의 수로 나눈 비율이고, 실제로 보장해 주어야 하는 파일의 수는 다음의 <식 6>으로 구할 수 있고 그 결과는 [표 1]과 같다.

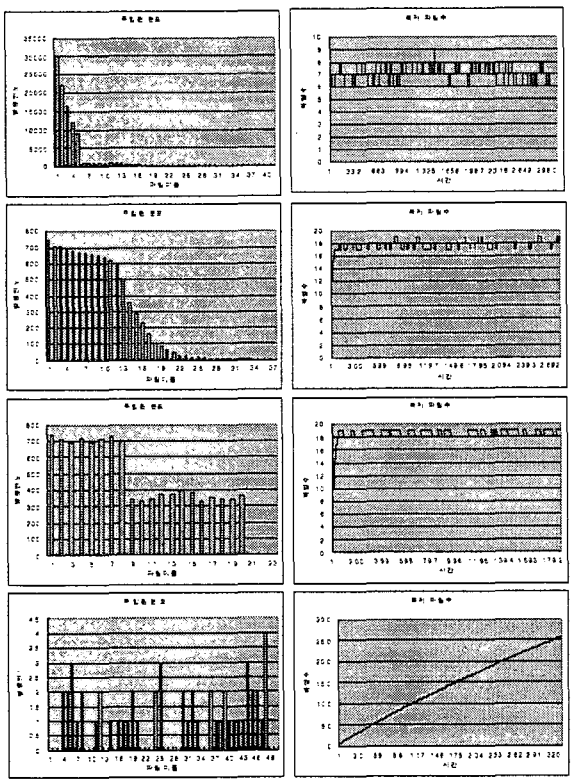
$$filenumber = \lfloor -\frac{1}{\lambda} ln(1-\theta) \rfloor \qquad \text{<식 6>}$$

[표 1] λ값에 따른 오차

	0.05	0.1	0.2	0.4	0.5
평균파일수	33.5	17.60	9.69	6.22	6.01
보장파일수	59.91	29.96	14.98	7.49	5.99
오차율	1.79	1.70	1.54	1.20	0.99

[표 1]에 나타난 결과를 보면 지역성이 커질수록 오차율이 줄어드는 것을 알 수 있다. 즉, 지역성이 높을수록 원래 의도했던 만큼의 가용성을 보장하고 있다. 반대로 지역성이 낮은 환경에서는 제안한 알고리즘이 상대적으로 부정확한 동작을 하고 있음을 알 수 있다.

비지수 분포 형태의 접근 패턴에 대해서도 성능평가를 수행하였다. [그림 3]에서 좌측 단의 그래프가 입력 분포들이고, 우측 단의 그래프가 각 입력 패턴에 대해 복제된 파일의 개수이다. 세 번째 테스트 입력 패턴까지는 각각 6~7개, 17~18개, 18~19개의 파일 복제가 이루어져서 적절히 가용성을 보장하고 있음을 알 수 있다. 하지만 네 번째 테스트 입력 패턴, 즉 전체 파일이 고르게 사용되는 경우에는 복제되는 파일의 수가 계속 증가하는 것을 볼 수 있다. 이는 HA-PVFS가 접근 패턴이 높은 지역성을 갖는다는 전제하에 설계, 구현되었기 때문이며, 이 결과를 통해서 접근 패턴이 낮은 지역성을 갖는 시스템에 HA-PVFS를 적용하는 것은 적절하지 않다는 것을 알 수 있다.



[그림 3] 비지수 분포 접근 패턴에 대한 복제 파일의 수

4. 결론

기존의 가용성 지원을 위한 클러스터 파일 시스템들은 시스템에 존재하는 모든 파일을 그 대상으로 삼았다. 따라서 사용되지 않는 파일들에 대해서도 가용성 지원을 하였고, 불필요한 비용을 지불하였다.

본 논문에서는 파일 시스템의 접근 패턴을 고려하여 실제로 많이 사용되는 파일에 우선적으로 가용성을 지원해 주는 HA-PVFS를 제안하였다. 접근 패턴 분석을 위해 인터넷 서비스의 접근 패턴을 지수분포라고 가정하였고 판단 최우 추정기를 이용하였다.

테스트를 통해 제시한 알고리즘이 적절하게 동작하는지 알아보았다. 그 결과, 높은 지역성을 보이는 입력 패턴에 대해서 약간의 오차는 있었지만 옳은 방향성을 보이며 적절한 수의 파일을 복제해 가용성을 유지하는 모습을 볼 수 있었다. 그러나 다중 클라이언트 상황에 대한 테스트는 이루어지지 않았으며, 보완이 필요할 것으로 보인다. 또한, 가상으로 생성된 요청으로 테스트를 진행하였으므로 실제 인터넷 서비스를 운용하고 있는 클러스터 시스템에서 가용성 지원의 적절성을 다시 한 번 확인해 볼 필요가 있다.

참고문헌

[1]. P. Carns et al. "PVFS: A parallel file system for linux clusters." In Proceedings of the 4th Annual Linux Showcase and Conference, pp.317-327, 2000.

[2] Peter J. Braam et al., "The Lustre Storage Architecture", Cluster File System. Inc, Mar. 2003

[3] K. W. Preslan et al., "A 64 Bit, Shared Disk File System for Linux", Proceedings of the 16th IEEE Mass Storage Systems Symposium, pp.22-41, 1999.

[4]F. Schmuck et al. "GPFS:A Shared-Disk File System for Large Computing Clusters", Proceedings of the FAST Conference on File and Storage Technologies, pp.231-234, 2002.

[5] J. H. Hartman et al. "The Zebra Striped Network File System", ACM Transactions on Computer System(TOCS) Volume 13, Issue 3, pp.274-310, August 1995.

[6] D. A. Patterson et al. "A Case for Redundant Arrays of Inexpensive Disks (RAID), Proc. of the ACM Conference on Management of DATA(SIGMOD), pp.109-116, June 1988.

[7] P. Carns et al. "PVFS: A parallel file system for linux clusters." In Proceedings of the 4th Annual Linux Showcase and Conference, pp.317-327, 2000.

[8] L. Cherkasova et al"Dynamics and Evolution of Web Sites: Analysis, Metrics and Design Issues", Proceedings of the Sixth IEEE Symposium on Computers and Communications(ISCC), pp.64-71, 2001.

[9] L. Cherkasova et. al "Analysis of Enterprise Media Server Workloads: Access Patterns, Locality, Content Evolution, and Rates of Change", IEEE/ACM Transactions on Networking, Vol. 12, No. 5, pp.781-794, OCTOBER 2004.