SDIO에서의 디스크 이질성 지원

김호진[01], 황인철[1], 김동환[2], 맹승렬[1], 윤현수[1]
한국과학기술원 전산학과[1], 한국전자통신연구원[2]
{hojin[0], ichwang}@camars.kaist.ac.kr, dhkim76@etri.re.kr, {maeng, hyoon}@camars.kaist.ac.kr

Disk Heterogeneity Support on Single Disk I/O

Hojin Ghim[01], In-Chul Hwang[1], Dong-Hwan Kim[2], Seungryoul Maeng[1], Hyunsoo Yoon[1]
Division of Computer Science, Dept. of EECS, Korea Advanced Institute of Science and Technology[1]
Electronics and Telecommunication Research Institute[2]

요 약

클러스터 시스템의 규모가 커질수록 이질성 문제가 심각해지고 유지, 보수에 큰 비용이 소요되게 된다. 본 논문에서는 I/O 부시스템의 입장에서 이러한 문제를 해결하기 위해 다양한 종류의 디스크를 지원하도록 개선된 SDIO(Single Disk I/O: 단일 디스크 입출력)를 설명한다. 단일 디스크 입출력은 리눅스의 커널 모듈의 형태로 제작되었으며 클러스터 내의 다양한 디스크를 하나의 큰 디스크 장치처럼 보이도록 해주는 역할을 한다. 또한 다양한 성능의 디스크가 존재할 때 모든 디스크의 성능을 최대한 활용하도록 한다.

1. 서론

클러스터 시스템[1]은 저렴한 가격과 확장성으로 인해 큰 인기를 얻고 있다. 클러스터 시스템이 많이 쓰이게 되고 규모 또한 점차 커지면서 이질적인 하드웨어에 대한 지원이 필수적이게 되었다. 본 논문에서는 이질적인 디스크를 보유한 클러스터 시스템에서 SDIO[2]가 어떻게 이질적인 디스크를 활용할 것인가에 대한 해결 방안을 모색한다.

SDIO는 클러스터 시스템의 각 노드의 디스크를 묶어 하나의 큰 디스크로 보이게 하는 소프트웨어이다. 기존의 SDIO는 모든 디스크를 동일한 성능과 동일한 용량의 디스크로 간주하고 있다.

디스크의 이질성은 다양한 각도에서 정의될 수 있지만 본 연구에서는 디스크의 성능과 용량을 이질성의 기준으로 삼는다. 또한 성능이나 용량이 다른 디스크가 두 종류 이상 존재할 때의 시스템 환경을 이질적인 디스크 환경이라 부른다.

본 논문의 구성은 다음과 같다. 2장은 디스크의 이질성 지원에 관련된 타 연구를 소개한다. 3장에선 본 연구의 기본 플랫폼인 SDIO에 대해 설명한다. 4장에서는 디스크의 이질성에 따라 디스크 블락의 배치를 조정할 수 있는 알고리즘(SDIO-HETERO)을 설명한다. 5장에서 간단한 성능 측정 결과를 보이고, 6장에서 결론 및 향후 과제에 대해 논의한다.

2. 관련연구

2.1 AdaptRaid5

AdaptRaid5[3]는 널리 쓰이고 있는 RAID-5를 heterogeneous 디스크 환경에서 사용하도록 변형시킨 것이다. 따라서 RAID-5의 장점을 그대로 얻을 수 있으며, 다양한 용량의 디스크에서 용량 활용률을 최대화시킨다.

AdaptRaid5는 heterogeneous 디스크 환경의 모든 용량을 활용하면서 RAID-5와 비슷한 성능과 신뢰도를 보이고 있다. 그러나 단순한 계산으로 주소가 결정되는 RAID-5와 달리 addressing에 다중 레벨의 테이블을 필요로 하며 그 검색 알고리즘도 복잡해진다. 또한 디스크 블락 배치가 고정되어 disk configuration의 변동에 적응하기 어렵다.

2.2 Panda

Panda[4]는 대용량 과학 계산 어플리케이션을 위해 개발된 I/O 라이브러리이다. Panda는 collective I/O 형태의 API를 제공하며 원래 수퍼 컴퓨터에서 개발되었으나, 후에 클러스터 시스템에서도 사용할 수 있도록 개선되었다. collective I/O API를 통해 받는 데이터는 파일 시스템에서 보통 쓰이는 바이트 스트림이 아니고 다차원 배열의 데이터이다. Panda 라이브러리는 이 데이터를 I/O 노드에 적절히 분배하여 저장한다.

Panda는 여러 노드에서 대용량의 데이터를 동시에 읽고 쓰는 collective I/O에 알맞게 만들어져 있다. collective I/O API를 사용하는 어플리케이션에서 아주 높은 성능을 보일 수 있다. 그러나 collective I/O는 블락 단위로 작업하는 디스크 레벨에서는 제공할 수 없는 API이다. 따라서 일반적인 사용에는 적합지 않다.

2.3 RIO

RIO[5](Randomized I/O)는 멀티미디어 데이터를 위한 스토리지 시스템이다. 여러 개의 디스크를 사용하여 멀티미디어 데이터 요구에 대한 동시 처리량을 높이고 또한 최대 지연 시간을 보장함으로써 실시간 전송을 지원하는 것을 목적으로 한다.

RIO는 원본 데이터와 복제본의 두 가지 측면에서 데이터 allocation을 수행한다. 원본의 저장 위치는 무작위로 결정하고, 복제본의 저장 위치는 디스크의 성능을 나타내는 지표 중 하나인 BSR[6]을 사용하여 결정한다. BSR은

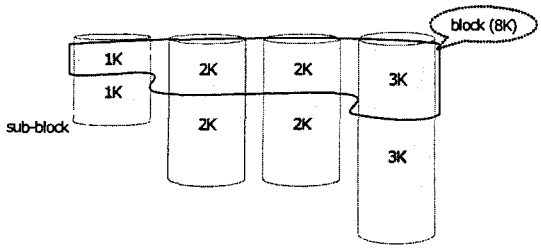

그림 1 디스크와 블락과 서브블락의 관계

Bandwidth-Space Ratio의 약자로서 용량에 비해 높은 성능을 보이는 디스크에서 높게 나타나는 척도이다.

RIO는 덩어리가 크고, 쓰기보다는 읽기가 주 작업인 데이터에 대해 좋은 성능을 보인다. 그러나 randomization 된 각각의 블락의 위치를 기록하기 위한 매핑 테이블의 크기가 필연적으로 커지게 되므로 성능의 장애 요소가 된다. 또한 작은 데이터나 메타데이터를 많이 사용하는 파일 시스템 작업에서는 좋은 성능을 낼 수 없다. 두 가지 종류의 디스크만을 사용할 수 있으므로 heterogeneity의 지원이 제한적이다.

3. SDIO(Single Disk I/O)

SDIO는 클러스터 환경에서 여러 노드에 장착되어 있는 디스크들을 하나의 디스크처럼 사용하도록 해주는 SSI 서비스의 일종이다. SDIO는 장치 드라이버 수준에서 구현되어 있으며 응용 프로그램에서 하나의 디스크를 사용하듯이 SDIO를 사용하면 실제 데이터는 클러스터 시스템에 흩어져 있는 디스크에 저장된다.

이처럼 하나의 가상 디스크를 제공함으로써 응용 프로그램은 데이터의 실제 위치에 신경쓸 필요없이 전체 클러스터 시스템의 디스크를 모두 사용할 수 있다. 또한 여러 개의 디스크를 병렬 작업에 활용하여 하나의 디스크에서보다 더욱 높은 성능을 얻을 수 있으며, 데이터를 중복 저장하여 필연적으로 발생할 수밖에 없는 디스크의 장애에도 불구하고 데이터의 유실을 방지할 수 있다.

SDIO는 장치 드라이버 수준에서 구현되어 있으므로 블락 단위의 Application Programming Interface(API)에서 투명성이 보장되므로 응용 프로그램이 보기에 물리적인 하드디스크와 완전히 같다. 따라서 SDIO에서 제공하는 가상 디스크에 일반적인 디스크에 사용되는 파일시스템을 수정 없이 사용하는 것이 가능하다. 파일시스템을 사용하는 응용 프로그램들도 마찬가지로 소스코드 수정이나 재 컴파일이 필요없으므로 이진 코드 수준의 호환성이 제공된다.

4. SDIO-HETERO

SDIO-HETERO는 본 절에서 설명하는 알고리즘을 사용하여 디스크 블락의 배치를 결정한다. 디스크 블락의 배치는 시스템 초기 설치시에만 수행할 수 있다.

SDIO-HETERO에서 하나의 블락은 여러 개의 서브블락으로 구성되며, 각 서브블락은 서로 다른 노드의 다른 디스크에 존재할 수 있다. 모든 블락의 크기는 같지만 블락을 구성하는 서브블락의 크기와 수는 다를 수 있다. 서브블락의 크기는 디스크의 특성에 따라 결정된다. 디스크와 블락, 서브블락의 관계를 그림 1에서 설명하고 있다.

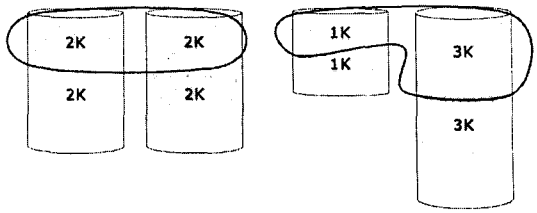

그림 2 서브블락을 그룹지어 하나의 블락을 만든다.

다음에 설명하는 디스크 블락 배치 알고리즘은 각 블락이 어떤 서브블락으로 구성되며, 각 서브블락이 어느 노드의 어느 디스크에 위치하는지를 결정하는 과정이다.

1. 시스템에 포함된 모든 디스크에 대해 서브블락의 크기를 정한다. 모든 디스크에서 서브블락에 접근하는데 필요한 시간이 동일하도록 서브블락의 크기를 정한다.
2. 서브블락들이 구성하는 블락은 SDIO-HETERO에서 기본 작업 단위가 된다. 대부분의 디스크 작업 단위가 4KB이거나 그 약수이므로 기존 파일시스템이나 어플리케이션들은 4KB의 배수의 단위로 디스크에 접근하도록 최적화되어 있는 경우가 많다. 따라서 SDIO-HETERO에서도 블락을 4KB의 배수로 맞추면 기존 최적화 기법이 그대로 적용된다.
   블락의 크기를 4KB의 배수로 맞추기 위해 먼저 모든 디스크의 서브블락의 크기의 합계가 4KB의 배수가 되도록 한다. 서브블락의 크기는 다음과 같이 조정할 수 있다.
   가. 모든 디스크의 서브블락을 두 배 한다.
   나. 디스크 중 BSR이 가장 큰 디스크의 서브블락 크기를 한 단계(디스크 하드섹터 크기) 줄인다. 서브블락의 크기를 줄일 수 있는 최소크기는 정해져 있는데 BSR이 가장 큰 디스크에서 서브블락 크기를 줄이면 그에 따른 접근 시간 변화가 가장 작으므로 밴드위쓰의 활용률이 가장 적게 줄어든다.
3. 서브블락을 여러 개의 그룹으로 나눈다. 이 때 모든 그룹은 속한 서브블락의 크기의 합계가 같아야 한다. 한 그룹으로 묶인 서브블락들이 곧 하나의 블락을 이루게 된다. 같은 크기의 그룹으로 묶기가 불가능한 경우 2.가 단계를 한번 더 거치고 3 단계를 수행한다.
   그림 2는 그룹으로 묶은 결과를 예시한 것이다.

모든 디스크 접근의 단위는 블락이고, 한 번의 블락 접근에 여러 개의 서브블락에 대한 접근이 동시에 일어나게 된다. 이 때 접근되는 서브 블락의 크기가 디스크에 따라 다르므로 디스크마다 한 번에 접근하는 데이터 크기가 달라지게 된다. 이로 인해 서로 다른 성능의 디스크를 동시에 사용하여도 모든 디스크의 성능을 최대한 활용할 수 있게 된다.

5. 성능 평가

성능 평가에 사용된 장비는 다음과 같다.

기종	A	B
CPU	Intel Pentium IV 1.8GHz	Intel Pentium III 850MHz
메모리	512MB	1.5GB
하드디스크	IBM Deskstar 120GXP	SAMSUNG SpinPoint 40GB
네트워크	3Com Etherlink III 3C59X	3Com Gigabit

100메가바이트를 네트워크를 통해 읽고 쓰는 간단한 실험 결과 데이터 전송률은 다음과 같이 나타났다.

기종	A	B
읽기	38652 KB/s	9039 KB/s
쓰기	26067 KB/s	6822 KB/s

위 결과에 따르면 A는 B에 비해 읽기에서 약 4.3배, 쓰기에서 약 3.8배의 성능을 보인다. 따라서 다음과 같은 세 가지 구성으로 실험을 실시하였다. 각 구성은 하나의 그룹만을 가지고 있고 이 그룹은 네 개의 서브블락을 포함한다. 다음 표는 구성별로 각 노드에 적용된 서브블락 크기를 나타낸다.

노드	노드1(A)	노드2(A)	노드3(B)	노드4(B)
small-homo	4KB	4KB	4KB	4KB
large-homo	16KB	16KB	16KB	16KB
hetero	16KB	16KB	4KB	4KB

성능 평가는 데이터 읽기와 쓰기를 반복했을 때 읽기와 쓰기의 시간당 평균 데이터 전송률을 척도로 이루어졌다. 데이터에 대한 읽기와 쓰기는 일정한 레코드 크기의 데이터를 연속적으로 읽어서 총 읽은 양이 100메가가 될 때까지 진행하여 읽기의 시간당 전송률을 구하고, 같은 방법으로 쓰기의 시간당 전송률을 구한다. 읽기와 쓰기를 번갈아 4번 반복하여 처음 반복의 전송률은 버리고 나머지 세 번의 전송률의 평균을 계산하였다. 쓰기를 수행할 때 한 번의 레코드를 저장할 때마다 디스크에 동기화를 수행하여 메모리 캐시의 영향을 줄이도록 했다. 또한 벤치마크 프로그램을 수행하는 노드는 디스크에 접근하지 않는 다섯 번째 노드를 사용하여 공정한 결과가 나오도록 했다.

그림 3은 그 결과를 나타낸다. 이질성 지원이 없는 small-homo와 large-homo는 거의 비슷한 성능을 보이고 있고, 이질성 지원이 포함된 hetero는 읽기에서 약 2배, 쓰기에서 1.4~1.8배의 성능을 보이고 있다.

small-homo와 large-homo에서는 모든 노드에서 같은 양의 데이터를 접근하기 때문에 A노드는 B노드가 작업을 끝낼 때까지 기다려야 하는 반면 hetero에서는 그룹 내 서브블락에 대한 접근시간을 일치시킴으로써 모든 노드가 기다리는 시간 없이 최대한의 I/O 밴드위쓰와 네트웍 밴드위쓰를 사용하기 때문이다.

6. 결론 및 향후 과제

클러스터 시스템은 규모가 커지고 유지보수가 진행될수록 동질 노드만으로 구성하는데 큰 비용을 필요로 하게 된다. 본 연구는 이러한 기존 클러스터 시스템을 완전히 대체하지

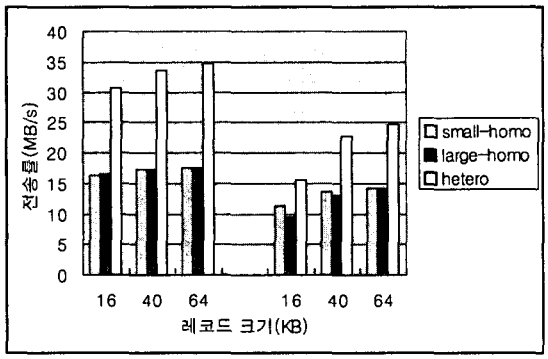

그림 3 데이터 전송률(왼쪽: 읽기, 오른쪽: 쓰기)

않고 새로운 구성품만을 추가함으로써 합리적인 비용에 성능을 높일 수 있는 가능성을 제시한다.

에서는 이질적인 디스크에 블락을 적절히 분배할 수 있는 알고리즘을 제시한다. 또한 제시된 알고리즘을 구현하는데 관련된 사항을 다루었으며, 구현된 SDIO-HETERO의 성능을 실험을 통해 확인하고 있다.

SDIO-HETERO는 디스크의 가능한 이질성 중 성능을 중점적으로 처리하고 있다. 이에 따라 모든 디스크의 용량을 전부 사용하지는 못하고 있다. 따라서 이렇게 남는 디스크 용량을 성능에 영향을 주지 않는 한에서 유용하게 사용하는 방법을 찾는 것이 향후 과제로서 남겨져 있다.

7. 참고문헌

[1] R. Buyya, "High Performance Cluster Computing: Architectures and Systems, Volume 1", Prentice Hall PTR, 1999

[2] 황인철, 김동환, 김호진, 맹승렬, 조정완, "단일 디스크 입출력을 위한 커널 모듈 프로토타입의 설계 및 구현", 한국정보과학회 2003년도 추계학술발표논문집, 2003

[3] T. Cortes and J. Labarta, "Extending Heterogeneity to RAID level 5", Proceedings of the General Track: 2001 USENIX Annual Technical Conference, pp119-132, 2001

[4] Y. E. Cho, M. Winslett, S. Kuo, J. Lee and Y. Chen, "Parallel I/O for Scientific Applications on Heterogeneous Clusters: a Resource-Utilization Approach", Proceedings of the 13th International Conference on Supercomputing, pp253-259, 1999

[5] J. R. Santos and R. Muntz, "Performance Analysis of the RIO Multimedia Storage System with Heterogeneous Disk Configurations", Proceedings of the 6th ACM International Conference on Multimedia, pp303-308, 1998

[6] A. Dan and D. Sitaram, "An Online Video Placement Policy Based on Bandwidth to Space Ratio (BSR)", Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, pp376-385, 1995