리눅스 PC 클러스터에서 ENBD를 이용한 SIOS

김태규[0] 김방현 김종현
연세대학교 전산학과
windcry@korea.com[0], legnamai@chol.com, jhkim@dragon.yonsei.ac.kr

Single I/O System Using ENBD on Linux PC Clusters

Tae-Kyu Kim[0] Bang-Hyun Kim Jong-Hyun Kim
Dept. of Computer Science, Yonsei University

요 약

클러스터 컴퓨터에서 접속된 모든 노드들의 디스크들을 통합 사용하기 위한 SIOS의 구현은 사용자 레벨과 파일 시스템 레벨, 그리고 디바이스 드라이버 레벨로 분류할 수 있다. 본 연구에서 제안하는 방법은 현재 공개되어 있는 소프트웨어 라이브러리만을 이용하여 리눅스 클러스터에서 SIOS를 구현하는 방법으로서, 확장 네트워크 블록 디바이스(ENBD: Enhanced Network Block Device)를 이용한 디바이스 드라이버 레벨의 하위 계층과 S/W RAID 및 NFS를 이용한 파일 시스템 레벨의 상위 계층으로 구성된다.

이 방법의 주요 장점은 현재 공개되어 있는 소프트웨어 라이브러리를 이용하기 때문에 구현이 용이하고 비용이 들지 않는다는 점이다. 그리고 하위 계층으로서 디바이스 드라이버 레벨의 ENBD를 이용하기 때문에 파일 시스템을 변경하지 않기 때문에 이전의 응용 프로그램에 대한 호환성이 높다. 또한, 상위 계층에서는 파일 시스템 레벨의 S/W RAID와 NFS를 이용함에 따라 디스크 배열 방식의 조정이 비교적 자유롭다. 또 다른 장점은 하위 계층과 상위 계층이 서로 독립적이기 때문에, 클러스터의 사용 목적에 따라 각 계층을 다양한 방법으로 변경할 수 있다는 것이다. Bonnie 벤치마크를 이용한 성능 측정 결과에 따르면, ENBD를 이용하여 RAID-5로 구성한 경우에 오버헤드가 높은 NFS를 사용했음에도 불구하고 비용이 많이 드는 다른 방법과 대등한 성능을 보였으며, 부분적으로는 더 높은 성능과 확장성을 가지는 것으로 나타났다.

1. 서 론

클러스터 컴퓨터는 저비용으로 고성능 병렬컴퓨팅 환경의 구현이 가능하다는 점에서 그 잠재력이 입증되어 왔다. 클러스터 컴퓨터에서 단일 시스템 이미지(SSI: Single System Image) 서비스는 성능, 편의성, 확장성, 그리고 신뢰성 측면에서 클러스터의 이용률을 향상시킬 수 있는 주요 방법이다[1][2]. 특히 I/O 중심의 응용을 고속으로 처리하기 위해서는 단일 I/O 공간(SIOS: Single I/O Space)을 구축하는 것이 필요하다. 즉, SIOS 프레임워크의 구축은 SSI 서비스를 제공하는 클러스터 환경에서 I/O 중심의 모든 응용들을 효과적으로 지원하는 기반 구조가 된다[3]. 그러나 현재 PC 클러스터 환경에서 SIOS에 관한 연구는 매우 미비한 실정이다. SIOS에 대한 최근 연구로는 디바이스 드라이버 레벨에서 SIOS를 구현한 K. Hwang의 연구[4]가 있다.

클러스터 컴퓨터에서 접속된 모든 노드들의 디스크들을 통합 사용하기 위한 SIOS를 구현하는 방법은 사용자 레벨과 파일 시스템 레벨, 그리고 디바이스 드라이버 레벨로 분류할 수 있다[5]. 본 연구에서 제안한 방법은 리눅스 클러스터에서 현재 공개되어 있는 소프트웨어 라이브러리만을 이용하는 방법으로서, 확장 네트워크 블록 디바이스(ENBD: Enhanced Network Block Device)를 이용한 디바이스 드라이버 레벨과 S/W RAID 및 NFS를 이용한 파일 시스템 레벨을 조합한 형태로 구성된다. 클러스터의 대표 노드는 다른 노드들에 있는 원격 디스크들을 ENBD를 이용하여 가상 디스크들로 만들고, 이들을 S/W RAID를 이용하여 하나의 메타 디스크(meta disk)로 만든다. 이렇게 생성된 메타 디스크를 다른 노드들이 NFS를 통하여 액세스할 수 있도록 함으로써 클러스터 내의 모든 노드들은 SIOS를 가지게 된다.

2. 관련 연구

SIOS를 지원하는 클러스터에서 각 노드는 I/O 장치의 물리적인 위치를 알지 못하더라도 지역 디스크나 원격 디스크를 액세스할 수 있다[2]. SIOS를 구성하는 디스크들은 고유의 I/O 주소가 배정되어, 사용자에게는 단일 주소 공간을 가지는 하나의 가상 디스크처럼 보인다. 그림 1은 클러스터 사용자 관점에서 SIOS에 의한 시스템 구성 개념을 보여주고 있다.

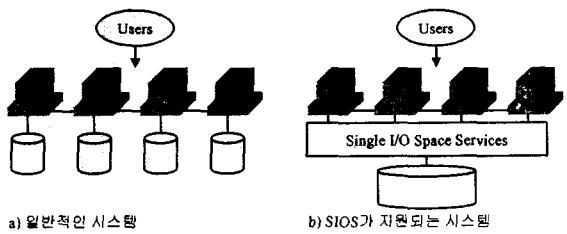

그림 1. SIOS 서비스의 개념

PC 클러스터에서 SIOS를 구현한 최근 연구로는 디바이스 드라이버 레벨의 CDD(Cooperative Disk Driver)를 이용한 K. Hwang의 연구가 있다[4]. 이 연구에서는 서버가 별도로 존재하지 않는 분산 디스크 시스템을 구성하기 위하여 커널 수준에서 동작하는 CDD를 이용하여 SIOS를 제공한다. CDD는 이를 위하여 가상 디스크를 이용한 원격 디스크 액세스 기능과 데이터 일관성 유지 기능, 그리고 S/W RAID와 같은 디스크 배열 기술 등을 포함하고 있다(그림 2). 그러나 이 방법은 추가적인 비용이 들고 시스템 확장성에 한계가 있다.

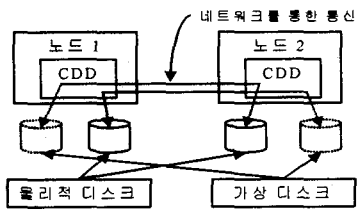

그림 2. CDD를 이용한 SIOS

가상 디스크를 이용하여 원격 디스크를 액세스할 수 있는 방법을 제공하는 다른 연구로는 P. Machek의 네트워크 블록 디바이스(NBD: Network Block Device)가 있다[6]. NBD는 지역 클라이언트에 하드 디스크나 파티션과 같은 블록 디바이스를 시뮬레이션 하는 액세스 모델을 제공하며, 이것은 실제 물리적인 저장장치를 가지고 있는 원격 서버에 네트워크를 통해 연결된다. 이렇게 제공된 가상 디스크는 클라이언트에게는 로컬 디스크 파티션처럼 보이지만, 실제는 단지 원격 디스크로의 통로 역할만 한다. 실제 액세스 요구와 데이터 블록들이 네트워크 상에서 통신되고 있음에도 불구하고, NBD 계층은 모든 상세 동작을 숨기기 때문에 클라이언트는 가상 디스크를 로컬 디스크처럼 사용할 수 있다. 그림 3은 이와 같은 NBD의 기본 개념을 보여주고 있다. 본 연구에서 이용한 ENBD는 리눅스에서 NBD의 확장된 기능을 제공하는 공개된 소프트웨어 라이브러리이다.

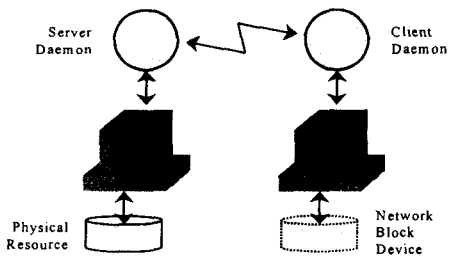

그림 3. NBD의 개념

3. ENBD를 이용한 SIOS의 구현

본 연구에서는 리눅스 클러스터에서 현재 공개되어 있는 소프트웨어 라이브러리인 ENBD와 S/W RAID 및 NFS를 이용하여 SIOS를 구현하였다. ENBD는 디바이스 드라이버 레벨의 하위 계층에서 동작하고, S/W RAID와 NFS는 파일 시스템 레벨의 상위 계층에서 동작한다. ENBD는 가상 디스크를 이용하여 원격 디스크를 액세스할 수 있게 하고, S/W RAID는 서버에서 클러스터에 분산되어 있는 디스크들을 하나의 메타 디스크로 묶는 역할을 한다. 그리고 NFS는 서버에서 생성된 메타 디스크를 클라이언트들이 액세스할 수 있게 해주며, 다중 액세스를 위하여 데이터 일관성을 유지할 수 있게 한다.

SIOS를 구성하기 위한 과정은 다음 순서대로 진행된다. 먼저 클러스터의 서버 노드는 ENBD를 이용하여 다른 노드들의 원격 디스크들을 가상 디스크들로 만들고, S/W RAID를 이용하여 가상 디스크들을 하나의 메타 디스크로 구성한다. 이렇게 구성된 메타 디스크는 단일 주소 공간을 가지게 되고, NFS를 통하여 다른 모든 클라이언트 노드들이 사용할 수 있게 된다. 결과적으로 클러스터 내의 모든 노드들은 SIOS를 가지게 된다.

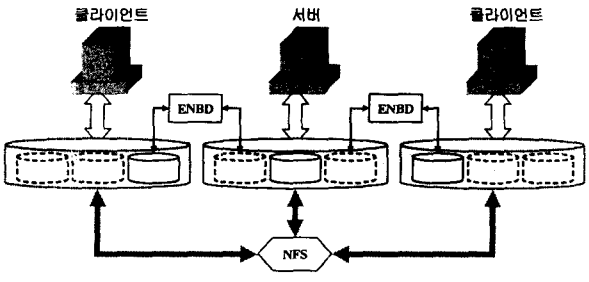

그림 4. ENBD와 S/W RAID 및 NFS를 이용한 SIOS

클러스터 컴퓨터에 접속된 서버 노드와 클라이언트 노드들은 SIOS에 접근하는 방법에 차이가 있다. 서버 노드는 메타 디스크를 직접 액세스하여 SIOS에 접근하지만, 클라이언트 노드는 NFS를 통하여 메타 디스크를 간접 액세스하여 SIOS에 접근한다. 그림 4는 ENBD와 S/W RAID 및 NFS를 이용한 SIOS의 구성을 보여주고 있다.

4. 구현 방법 및 성능 분석

이 절에서는 본 연구에서 제안한 SIOS 구현 방법과 CDD 방식[4]의 특징을 비교하고, 현재 공개된 소프트웨어 라이브러리들만을 이용하여 SIOS를 구현하였음에도 불구하고 성능 측면에서도 대등하다는 것을 입증하기 위하여 성능 분석 결과를 제시하였다.

4.1 구현 방법

CDD 방식은 리눅스 PC 클러스터 환경에서 가상 디스크와 S/W RAID를 사용하여 SIOS를 구현하였다는 측면에서는 본 연구와 유사하지만 아래의 표 1과 같은 차이점이 있다.

표 1. CDD 방식과 본 연구의 비교

	CDD 방식	본 연구
구조	분산형	중앙 집중형
구현 모듈	일체형	조립형
확장성	제한적	높음
S/W 종류	비공개	공개
이식성	낮음	높음
설치	커널 재컴파일 필요	용이

먼저 구조 측면에서 보면, CDD 방식은 분산형이기 때문에 노드들에게 SIOS 서비스를 제공하기 위한 모든 기능들이 CDD에 포함되어 있다. 따라서 CDD 방식의 구현 모듈은 일체형으로 되어 있어 SIOS의 크기나 내부 구성을 변경하는 것이 어렵다는 문제점이 있다. 반면에 본 연구는 클라이언트-서버 형태를 가지는 중앙 집중형이므로 서버에 SIOS 서비스 기능이 집중되어 있고, 구현 모듈이 상위 계층과 하위 계층으로 분리되어 있기 때문에 SIOS의 확장 및 축소가 용이하다. 또한 CDD 방식에서 SIOS 서비스가 CDD를 포함한 클러스터 내의 노드들에게만 제공되지만, 본 연구의 방법에서는 NFS를 통하여 서버와 연결 가능한 모든 컴퓨터들에게 SIOS를 제공할 수 있다.

SIOS 구축 측면에서 보면, CDD 소프트웨어 라이브러리는 비공개되어 구축이 어렵지만, 본 연구는 현재 공개되어 있는 소프트웨어 라이브러리만을 사용하기 때문에 비용을 들이지 않고 쉽게 SIOS를 구현할 수 있다. 이식성 측면에서도 CDD 방식은 클러스터 내의 모든 노드에 CDD 소프트웨어를 설치한 후에 커널 재컴파일이 필요하지만, 본 연구에서는 ENBD 소프트웨어만 설치하면 된다.

4.2 성능 분석

본 연구에서 제안한 클러스터 시스템의 성능을 비교하기 위하여 CDD 연구와 유사한 환경에서 실험하였다. 클러스터는 8개의 PC 노드로 구성하였고, 각 노드는 스위칭 허브를 통하여 100 Mbps Fast-Ethernet으로 연결하였다. 그리고 노드의 운영체제는 리눅스 커널 2.4.20-8의 Red-Hat 9이며, SIOS를 위하여 각 노드의 하드 디스크에 1 GB 용량을 할당하여 전체

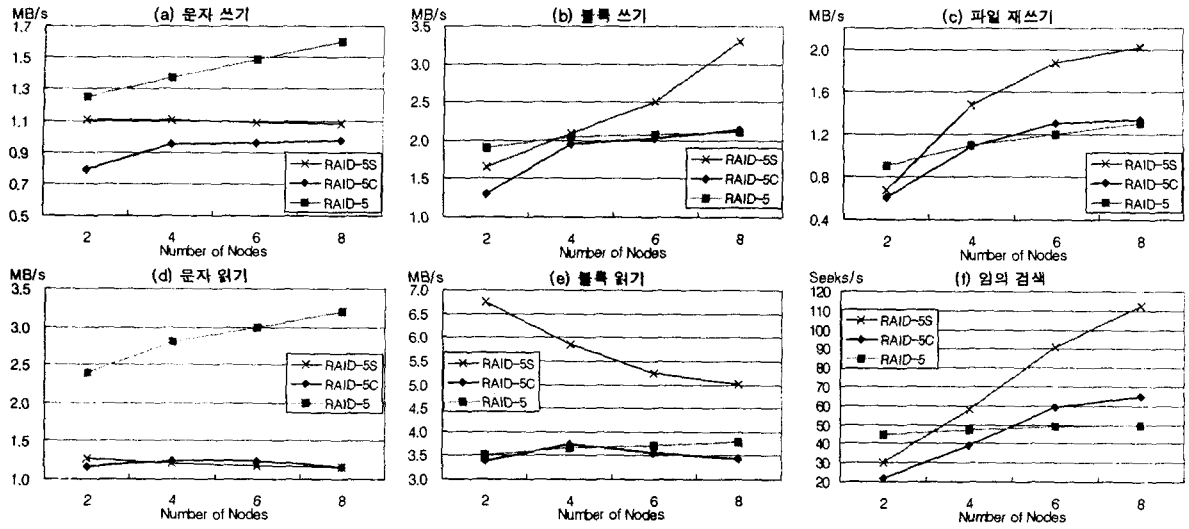

그림 5. 노드 수에 따른 Bonnie 벤치마크 결과

SIOS 용량을 8 GB로 설정하였다. 실험은 선형적 확장성(scalability) 분석을 위하여 SIOS에 포함되는 노드의 수를 2, 4, 6, 8개로 증가시키면서 RAID-5를 구성하여, Bonnie 벤치마크를 각각 10회 실행하였다.

그림 5의 실험 결과에서 'RAID-5'는 CDD 방식의 결과를 나타내고, 'RAID-5S'와 'RAID-5C'는 본 연구에서의 서버와 클라이언트 결과를 각각 나타낸다. '문자 입출력'에서는 본 연구의 결과가 성능과 선형적 확장성이 CDD 방식에 비하여 다소 낮은 것으로 나타났다. 그러나 '블록 쓰기'와 '파일 재쓰기' 및 '임의 검색'에서는 'RAID-5S'가 CDD 방식에 비하여 높은 성능과 선형적 확장성을 가지는 것으로 나타났고, '블록 읽기'에서도 선형적 확장성은 떨어지지만 매우 높은 성능을 나타내었다. 또한 'RAID-5C'는 NFS의 오버헤드 때문에 'RAID-5S'에 비하여 전반적으로 낮은 성능을 나타내지만, '문자 입출력'을 제외한 항목에서는 노드의 수가 증가할수록 CDD 방식의 결과와 대등한 성능을 가지는 것으로 확인되었다.

5. 결론

본 연구에서는 리눅스 PC 클러스터에서 현재 공개된 소프트웨어 라이브러리만을 이용하여 SIOS를 구현하는 방법을 제시하였다. 이 방법의 주요 장점은 현재 공개되어 있는 소프트웨어 라이브러리만을 이용하기 때문에 구현이 용이하고 비용이 들지 않는다는 점이다. Bonnie 벤치마크를 이용한 성능 측정 결과에 따르면, ENBD를 이용하여 RAID-5로 구성한 경우에 오버헤드가 높은 NFS를 사용했음에도 불구하고 다른 유사한 연구에 비하여 뒤떨어지지 않는 성능을 나타내었으며, 부분적으로는 더 높은 성능과 선형적 확장성을 가지는 것으로 나타났다.

6. 향후 연구과제

본 연구의 클러스터 시스템에서 클라이언트 노드들은 NFS를 거쳐서 SIOS에 접근하기 때문에 비교적 낮은 성능을 나타내는데, 이 부분은 반드시 개선되어야 할 부분이다. 또한 ENBD는 TCP/IP 프로토콜을 이용하여 노드 간 통신을 하는데, 이미 많은 연구에서 언급된 바와 같이 TCP/IP는 높은 오버헤드를 가지고 있는 프로토콜이다. 최근 클러스터 시스템은 네트워크 안정성이 보장되는 소규모 지역 네트워크로 연결되는 경향이기 때문에 통신 안전성은 낮지만 고속의 속도를 지원하는 대체 통신 프로토콜을 사용할 필요가 있다.

7. 참고문헌

[1] K. Hwang and Z. Xu. Scalable Parallel Computing: Technology, Architecture, Programming. McGraw-Hill, New York, 1998.

[2] K. Hwang, H. Jin, E. Chow, C.L. Wang, and Z. Xu. "Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space," IEEE Concurrency Magazine, March 1999, pp. 60-99.

[3] R. Ho, K. Hwang, and H. Jin, "Design and Analysis of Clusters with Single I/O Space," Proc. 20th Int's Conf. Distributed Computing Systems (ICDCS 2000), pp. 120-127, Apr. 2000.

[4] K. Hwang, H. Jin, and R. Ho, "Orthogonal striping and mirroring in distributed RAID for I/O-centric cluster computing," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 1, pp. 26-44, Jan 2002.

[5] R. Ho, K. Hwang, and H. Jin, "Single I/O space for scalable cluster computing," Proc. 1th IEEE Computer Society Int's Workshop Cluster Computing, pp. 158-166, Dec. 1999.

[6] P.T. Breuer, A.M. Lopez, and A.G. Ares. The network block device. Linux Journal,(73), May 2000.