SIOS 기반의 저장 장치를 사용하는 클러스터 시스템의 결함 회복 성능 평가

유택근, 장윤석
대진대학교 컴퓨터공학과
e-mail:{tacgeun, cosmos}@daejin.ac.kr

Performance Evaluation of the Roll-back Recovery on the Cluster System with SIOS

Taek-Geun Yu, Yun-Seok Chang
Dept of Computer Engineering, Dae-Jin University

요 약

클러스터 시스템에서 결함이 발생하였을 때, 결함 회복 성능은 매우 중요한 설계 요소가 된다. 단일 입출력 공간(SIOS)을 저장 장치로 사용하는 클러스터 시스템에서, 각 노드들의 결함허용정보를 주기적으로 저장하는 roll-back 결함회복 기법을 사용하는 경우, 결함 회복 성능은 SIOS가 제공하는 입출력 병렬성과 깊은 관계가 있다. 본 연구에서는 클러스터 시스템의 SIOS 구성에 참여하는 노드 수에 따른 결함 회복 성능을 HPL 벤치마크를 통하여 여러 환경에서 평가하고, 그 결과를 분석하였다. 성능 평가 수행 결과, 클러스터 시스템은 SIOS 구성에 참여하는 노드의 수가 증가할수록 우수한 결함 회복 성능을 보인다. 따라서 SIOS를 결함허용정보 저장 장치로 사용하는 클러스터 시스템을 설계할 경우, SIOS 구성에 참여하는 노드 수가 클러스터 시스템의 결함 회복 성능을 결정하는 데에 중요한 요소가 됨을 알 수 있다.

1. 서론

클러스터 시스템의 각 노드에서 동시에 실행되는 모든 프로세스들은 메시지 전송을 통하여 상호간에 통신을 수행함으로써 동기적인 프로세스 수행이 가능하도록 한다[1]. 그러므로 하나 이상의 노드에서 결함이 발생될 경우에 클러스터 차원에서 전역적인 일관성을 유지하고 최소한의 비용으로 결함 회복(Rollback recovery)을 수행하기 위하여, 각 노드들은 자신이 수행하고 있는 프로세스에 대한 정보, 즉 결함허용정보(Checkpoint)를 저장 장치에 주기적으로 저장하여야 한다[2].

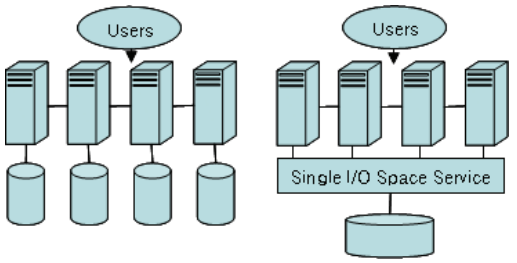
이러한 결함허용정보는 대개 파일 서버나 서버에 연결되어 있는 지역 디스크와 같은 집중된 저장 장치를 사용하여 저장하고, 결함이 발생될 경우, 저장 장치에 저장된 결함허용정보를 사용하여 회복을 수행한다. 그러나 서버를 사용하지 않는 다중 컴퓨터를 기반으로 하는 클러스터 시스템에서는 각 노드에 연결된 지역 디스크들을 이용하여 분산 RAID를 구현하고 여기에 단일 입출력 공간(SIOS : Single IO Space)의 저장 공간을 구성하여, 결함허용정보를 저장하는 방법이 서버를 이용하는 방법에 비하여 효율적이다[3][4]. 따라서 SIOS을 저장 장치로 사용하는 클러스터 시스템에서 결함 회복 성능은 SIOS가 제공하는 입출력 병렬성과 깊은 관계가 있다.

따라서 본 연구에서는 SIOS 기반의 저장 장치를 사용하는 클러스터 시스템의 결함 회복 성능에 SIOS 구성에 참여하는 노드 갯수가 미치는 영향을 평가하기 위하여 클러스터 컴퓨터 시스템에 SIOS를 구현하고, HPL(High-Performance Linpack) 벤치마크와 MPI를 이용하여 SIOS 참여 노드 수에 따른 결함 회복 성능을 비교, 분석하였다. 그리고 다양한 성능 평가 결과들을 제시함으로써 SIOS를 기반으로 하는 클러스터 시스템의 결함 회복을 위한 저장 장치를 설계할 때에 활용할 수 있도록 하였다.

2. 관련연구

2.1 단일 입출력 공간(SIOS : Single IO Space)

SIOS는 클러스터 시스템에서 개별 노드에 연결된 I/O장치 또는 저장장치의 물리적인 위치에 대한 지식 없이 어떤 노드에서도 공통으로 접근 가능하게 하여 주는 기능이다. SIOS를 구성하는 디스크들은 고유의 I/O 주소가 배정되어, 사용자에게는 단일 주소 공간을 가지는 하나의 가상 디스크처럼 보인다[5]. 그림 1은 클러스터의 사용자 관점에서 SIOS에 의한 시스템 구성 개념을 보이고 있다.



a) 일반적인 시스템 b) SIOS가 지원되는 시스템

그림 1. SIOS 서비스의 개념

SIOS를 지원하여 주는 접근 방식에 따라 분류하면 사용자 레벨(User level)의 구현 방법과 파일 시스템 레벨(File system level)의 구현방법, 그리고 장치 드라이버 레벨(Device driver level)의 구현 방법으로 구분할 수 있다. 사용자 레벨의 구현 방법으로는 Clemson 대학의 PVFS(Parallel Virtual File System)와 Argonne 국립 연구소의 RIO(remote I/O) 프로젝트가 있고, 파일 시스템 레벨의 구현 방법으로는 UC Berkeley 대학의 xFS(serverless network file system)와 Sun Microsystems 연구소의 Solaris MC 프로젝트가 있으며, 장치 드라이버 레벨의 구현 방법으로는 DEC 사의 Petal 프로젝트와 USC 대학의 CDD(Cooperative Disk Driver)가 있다. 또한 가상 디스크 방식을 이용한 원격 디스크 액세스에 관한 다른 연구로 P. Machek의 네트워크 블록 디바이스(NBD: Network Block Device)가 있다[6].

2.2 Checkpoint와 Recovery

결함허용정보(Checkpoint) 란, 실행중인 프로세스 수행시간 선상의 한 점에서의 프로세스 상태 정보로, 하나의 프로세스가 정상적으로 수행을 완료하기 위해 오랜 시간을 요구하는 경우, 수행중인 프로세스의 상태 정보는 결함 허용 프로그램에 의해 안정된 저장 장치에 저장된다. 이후 시스템에 결함이 발생하게 되면, 해당 프로세스는 초기 상태로부터 재실행되는 것이 아니라, 저장 장치에 있는 결함허용정보를 이용하여 프로세스의 상태를 복원하는 결함 회복 과정을 수행한다. 결함 회복 후의 프로세스는 프로세스가 일시 중단되었다가 재개되는 형태로 이전의 작업을 계속 수행할 수 있다.

이와 같은 결함 허용 구조를 지원하는 시스템에서는 수행중인 프로세스의 상태 정보를 주기적으로 저장하므로 결함이 발생하였을 때 마지막으로 결함허용정보를 저장했던 시점으로부터 프로세스의 상태를 복원하여 수행하는 것을 결함 회복이라고 한다[7][8].

현재까지 여러 연구를 통하여 다양한 결함 허용 구조들이 설계되고, 이들을 지원하는 프로그램 라이브러리들이 개발되었다. 그 중 가장 대표적인 Libckpt[8]는 분할된 결함허용정보와, 증가(Incremental) 결함허용정보 저장 기능과 사용자에게 투명한 결함허용정보를 제공한다.

3. 실험환경

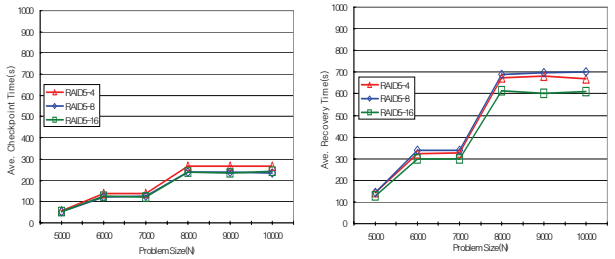
본 연구에서는 Libckpt 라이브러리를 이용한 프로그램 함수를 HPL 벤치마크 프로그램에 첨가함으로써 시스템에 추가적인 부하를 가하지 않고 결함허용정보를 저장하는 알고리즘을 구현하여 Linux 클러스터 환경에서 결함허용과 결함 회복을 수행할 수 있는 벤치마크 프로그램을 구성하였다. 클러스터 시스템의 결함 회복 성능은 메시지 기록에 소요되는 시간을 포함하여 결함허용정보를 저장하고 결함 회복을 수행하는 데에 걸리는 시간과 입출력 처리량을 통하여 직접적으로 나타낼 수 있다.

벤치마크를 실행시키기 위한 클러스터 시스템은 1000Mbps의 네트워크 대역폭을 가지는 이더넷으로 4개, 또는 16개의 노드들을 연결하여 구성하였다. 벤치마크에서 사용된 클러스터 노드의 구성은 <표 1>과 같다.

<표 1> 클러스터 노드 구성

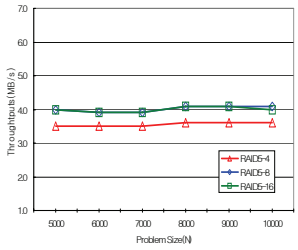
구성요소		규 격
CPU		Intel Pentium-Ⅳ 1.7GHz
RAM		512Mbyte DDR SDRAM
HDD	Model	Seagate ST380021A
	Capacity	80 GB
	Spindle speed	7200 rpm
NETWORK		1000Mbps RTL8169

각 노드에서 수행되는 운영체제는 리눅스 커널 2.4.20 버전의 Red-Hat 9.0이며, SIOS를 제공하기 위하여 각 노드에 연결되어 있는 디스크에 1.5GB 용량을 할당하여 전체 SIOS 용량을 최대 24GB로 설정하였으며, 단일 입출력 공간을 이루기 위하여 최소 3개에서 최대 16개까지의 디스크들을 분산 RAID로 구성하였다.

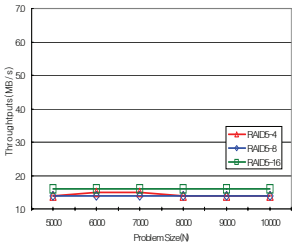


(A) 4(2x2)Node Checkpoint Time

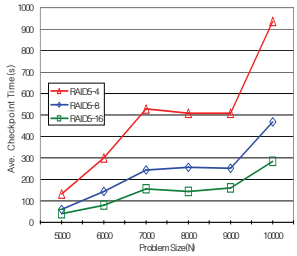
(B) 4(2x2)Node Recovery Time



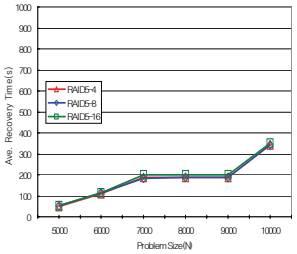
(C) 4(2x2)Node Checkpoint I/O Throughput



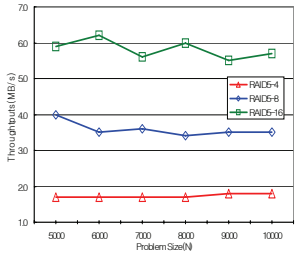
(D) 4(2x2)Node Recovery I/O Throughput



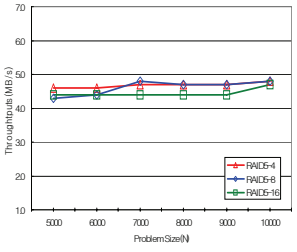
(E) 16(4x4)Node Checkpoint Time



(F) 16(4x4)Node Recovery Time



(G) 16(4x4)Node Checkpoint I/O Throughput



(H) 16(4x4)Node Recovery I/O Throughput

그림 2. SIOS의 부하에 따른 I/O 성능

SIOS 구현은 리눅스 프리웨어인 ENBD와 S/W RAID 및 NFS를 이용하여 SIOS를 구현하였으며, 이 소프트웨어들은 모두 공개 소스이다. 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 디스크(Meta Disk)로 묶는 기능과 RAID 시스템에서 제공하는 결함 허용 기능을 자체적으로 제공한다. 그리고 NFS는 서버에서 생성된 메타 디스크를 클라이언트들이 액세스할 수 있게 해주며, 다중 액세스를 위하여 데이터 일관성을 유지할 수 있게 한다.

본 실험은 HPL 벤치마크의 입력에서의 부하(Problem size)값을 5000에서 10000까지 변화시키면서 수행되었으며, 하나의 단위 부하가 처리되었을 때마다, Libckpt 함수를 사용하여 결함허용정보를 저장하도록 하였다. 그리고 4노드와 16노드로 구성된 클러스터 시스템에서 각각 SIOS에 참여하는 노드의 수를 4개, 8개, 그리고 16개로 증가시키면서 10회씩의 벤치마크를 실행하였다.

4. 실험 및 결과 분석

그림 2는 각각 4개와 16개의 노드를 가지는 SIOS 기반의 클러스터 시스템의 부하 증가에 따른 결함 회복 성능을 나타내고 있다. 이는 결함허용정보를 저장할 때 걸리는 시간(CT : Checkpoint Time)과 입출력 처리량(CTP : Checkpoint IO Throughputs) 그리고 결함 회복에 걸리는 시간(RT : Recovery Time)과 입출력 처리량(RTP : Recovery IO Throughputs)로 나타낼 수 있다

4노드와 16노드 클러스터 시스템 모두 부하 변화에 따른 I/O 처리량에 대한 변화는 거의 없었으며, 부하가 증가하면 결함허용정보의 크기 증가로 결함허용정보 입출력 시간이 증가하였다. 경우에 따라서, 부하가 증가하여도 결함허용정보의 크기가 같아서 입출력 시간이 거의 동일한 경우도 발생되었다.

4노드 클러스터 시스템에서의 결함 회복을 수행할 때의 입출력 성능은 결함허용정보를 저장할 때와 비슷한 특성을 나타내지만, 그 성능은 50% 미만으로 감소하였다. 이는 버퍼 캐쉬(Buffer Cache) 및 디스크 캐쉬(Disk Cache)에 서의 입출력 동작 특성으로 인해 발생하는 것으로 추정된다. 그리고 SIOS 참여 노드 수의 변화가 결함 허용 및 결함 회복 성능에 그다지 영향을 미치지는 않고 있다.

16노드 클러스터 시스템에서는 SIOS 참여 노드 수가 증가함에 따라서 결함 허용 및 결함 회복 성능

이 증가되는 것을 볼 수 있다. 이는 클러스터 구성 노드 수에 따른 SIOS 참여 노드 수가 결함 허용 및 결함 회복 성능에 영향을 준다는 것을 나타낸다. 즉, 노드 수 보다 많은 수의 노드가 SIOS에 참여하게 되면 I/O 병목현상이 발생될 확률이 높아지고, 따라서 클러스터 노드 수가 SIOS 참여 노드 수보다 큰 상태에서만 SIOS 참여 노드 수의 증가로 인한 입출력 병렬성이 증가되는 효과가 나타날 수 있다.

5. 결론 및 향후 연구과제

본 연구에서 수행한 성능 평가 결과, SIOS 기반의 저장 장치를 사용하는 클러스터 시스템은 SIOS 참여 노드 수가 증가할수록 우수한 결함 회복 성능을 나타냄을 확인할 수 있다. 또한 클러스터 시스템 노드 수를 넘지 않은 범위에서 SIOS 노드 수를 구성할 때 병렬성 효과가 보다 높게 나타남을 확인했다. 따라서 SIOS 저장 장치를 사용하는 클러스터 시스템을 설계할 경우, SIOS에 참여하는 노드의 수는 클러스터 시스템의 결함 회복 성능을 결정하는데 중요한 요소가 된다.

본 연구에서는 4개의 노드와 16개의 노드로 클러스터 시스템을 구성하여 결함 회복 성능 평가를 수행하였으나, 차후에는 32개의 노드 이상의 클러스터 시스템에 대하여 SIOS 참여 노드 수에 따른 결함 회복 성능 평가를 수행할 예정이며, 앞에서 언급된 Recovery I/O 성능에 영향을 미치는 것으로 예상되는 버퍼 캐쉬와 디스크 캐쉬 크기에 따른 I/O 성능에 대한 연구도 함께 수행할 예정이다.

참고문헌

[1] K. Hwang and Z. Xu, "Scalable Parallel Computing," McGraw Hill, 2000.

[2] 장윤석 (2003.6). 분산 RAID 기반의 클러스터 시스템을 위한 분할된 결함허용정보 저장 기법. 정보처리학회논문지A 제10-A권 제2호, pp.123-130.

[3] K. Hwang, H. Jin, R. Ho and W. Ro, "Reliable Cluster Computing with a New Checkpointing RAID-x Architecture," Proceedings of 9th Workshop on Heterogeneous Computing, Cancum, Mexico, 2000.

[4] K Hwang, H. Jin and R. Ho, "Raid-x : A New Distributed Disk Array for I/O-Centric Cluster Computing," Proceedings of 9th High-Performance Distributed Computing Symposium, Pittsburgh, 2000.

[5] 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, pp.60-9, March 1999.

[6] 김태규, 김방현, 김종현 (2006.12). 클러스터 컴퓨터를 위한 단일 I/O 공간 서비스의 구현 및 성능분석. 정보처리학회논문지A 제13-A권 제6호, pp.517-524.

[7] M. Litzkow and M. Solomon. "Supporting Checkpointing and Process Migration Outside the Unix Kernel" In Usenix Conference Proceedings, San Francisco, Ca, January 1992, pages 283-290.

[8] James S. Plank, Micah Beck, Gerry Kingsley and Kai Li, "Libckpt : Transparent Checkpointing under Unix", Conference Proceeding, Usenix Winter 1995 Technical Conference, New Orleans, LA, January, 1995, pp. 213-223.