내 장애성을 갖는 분산 인메모리 블록 스토리지
Fault-tolerant Distributed In-memory Block Storage

문 정 주, 정 창 명, 송 석 일
한국교통대학교

Jeongju Moon, Changmyeong Jeong, Seokil Song
Korea National University of Transportation

요약

이 논문에서는 클러스터의 다수 노드의 메모리를 블록 스토리지로 가상화하는 분산 인-메모리 스토리지 기술을 개발한다. 이때 클러스터를 구성하는 어떤 노드가 고장이 나더라도 지속적으로 스토리지에 접근할 수 있는 내장애성을 갖도록 한다. 또한, 실험을 통해서 개발한 분산 인-메모리 스토리지의 성능을 입증한다.

I. 서론

리눅스의 ramdisk [1]나 ramfs [2]와 같은 메모리 스토리지는 주기억 장치의 일부를 블록 스토리지로 사용할 수 있도록 하는 기술이다. 메모리 스토리지는 순차 IO (Sequential Input/Output) 는 물론 랜덤 IO (Random I/O)의 성능을 주기억장치에 근접하게 제공할 수 있다.

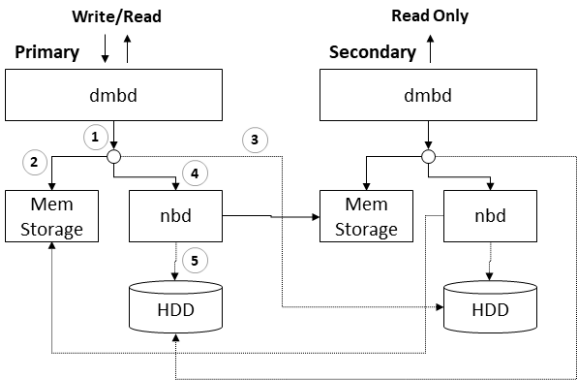
최근 주기억장치의 가격이 내려가면서 대용량의 데이터를 주기억장치에서 저장 및 처리하기 위한 인-메모리 (in-memory) 스토리지 및 데이터 처리 프레임워크들이 제안되고 있다. 특히, Apache Ignite [3] 와 Alluxio [4] 는 클러스터 환경에서 다수 노드의 메모리를 가상화하여 확장성과 내 장애성이 있는 인-메모리 저장소를 제공한다. 하지만, Apache Ignite 나 Alluxio는 POSIX 호환 스토리지가 아니기 때문에 전용 API를 이용해서 데이터를 저장하거나 읽을 수 있다. 따라서 기존의 응용을 Apache Ignite 나 Alluxio를 기반으로 운용할 수 없다. 반면, ramdisk 나 ramfs 는 기존 응용에서 바로 사용 가능하지만, 확장성 있는 스토리지를 제공하기가 어렵다.

이 논문에서는 클러스터의 다수 노드의 메모리를 블록 스토리지로 가상화하는 기술을 개발한다. 이때 클러스터를 구성하는 어떤 노드가 고장이 나더라도 지속적으로 스토리지에 접근할 수 있는 내장애성을 갖도록 한다.

Ⅱ. 제안하는 분산 인-메모리 블록 스토리지

이 논문에서 제안하는 분산 인-메모리 블록 스토리지 (DMBD, Distributed Memory Block Storage Device) 의 구조가 그림 1에 나타나 있다. 그림은 두 노드로 DMBD를 구성한다고 가정한 것이다. DMBD를 구성하는 노드들 중 하나를 Primary 노드라 하고 이를 제외한 나머지 노드를 Secondary 노드라한다. Primary 노드는 DMBD를 이용해 스토리지 서비스를 제공하는 노드를 말하며 Secondary 노드는 Primary 노드에 문제가 발생할 경우 스토리지 서비스를 연속해서 제공하게 된다.

DMBD는 메모리 스토리지 (Mem Storage) 와 nbd를 기반으로 설계 되었다. 메모리 스토리지로는 ramdisk나 ramfs 등 어떤 것을 사용해도 문제가 없다. 이 논문에선는 BestIO[5]를 이용한다. BestIO는 비휘발성 메모리인 nvdimm과 하드 디스크를 이용하여 메모리 스토리지를 영속성 있게 하는 특성을 가진다. nbd[6] 는 원격의 스토리지를 로컬의 블록 장치처럼 사용할 수 있도록 해주는 리눅스에서 제공하는 가상 장치 드라이버이다. DMBD는 적절한 매핑 기법을 기반으로 DBMD에 전달되는 블록 IO를 로컬의 메모리 스토리지인 Mem Storage 또는 원격의 Mem Storage에 전달하여 처리하게 된다.



▶▶ 그림 1. DMBD 의 구조

그림 1에서 DMBD는 상위로부터 전달되는 블록 IO (①) 에 대해서 매핑을 수행한다. 매핑을 통해서 블록 IO를 지역의 Mem Storage (②) 에 전달할지 nbd를 통해서 원격의 Mem Storage (④) 에 전달할지 결정한다. 두 경우 모두 블록 IO를 복제하여 ②의 경우 복제한 블록IO를 원격의 HDD (③) 로 전달하여 기록한다. ④ 의 경우에는 복제한 블록 IO를 로컬의 HDD (⑤) 에 기록하게 된다. 복제한 블록 IO를 원본 블록IO가 수행되는 노드를 제외한 다른 노드에 전달하여 저장하는 것은 노드의 고장이 발생하여 접근이 어려울 때에도 스토리지를 지속적으로 접근하여 데이터를 사용할 수 있도록 하기 위한 것이다.

Ⅲ. 성능평가

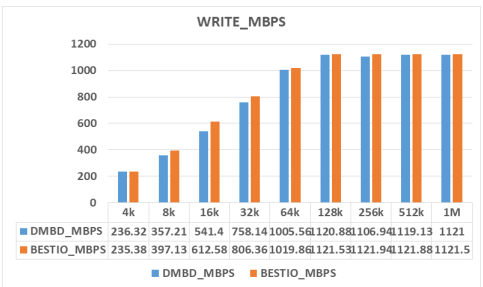
이 논문에서는 단일 노드에 BestIO를 설치하여 생성한 스토리지와 다수 노드에 BestIO를 설치하고 이를 DMBD를 이용하여 가상화한 스토리지간의 성능을 비교하였다. 성능평가에 사용된 하드웨어 및 소프트웨어 환경은 표 1과 같다.

표 1. 성능평가 환경

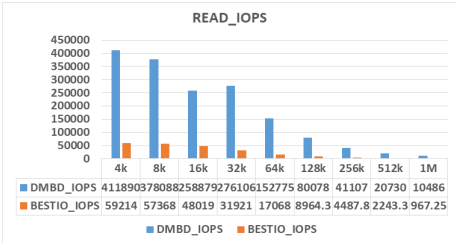
구분	내용
소프트웨어	리눅스 Fedora 26, Kernel Ver. 4.13.5 vdbench
하드웨어	10G 이더넷 CPU : 32코어 RAM : 32G HDD : 111G
스토리지	DMBD : 총 10G (각 노드에 5G 할당) BestIO : 단일 노드 10G
워크로드	4k~1M 크기의 랜덤 IO 총 5G 의 IO 발생

그림 2에서 5는 write 연산과 read 연산 각각에 대하여 표 1 의 워크로드에 따라 실험을 수행한 결과를 IOPS와 MBPS 별로 나타낸 결과이다. 이 그림에서 볼 수 있는 것처럼 Write 의 경우에는 BestIO 단일의 스토리지 성능과 DMBD 의 성능이 거의 유사하거나 BestIO가 다소 높은 것을 볼수 있다. 하지만, Read 의 경우 BestIO 단일 스토리지에 비해서 DMBD 의 성능이 더 우수한 것을 볼수 있다.

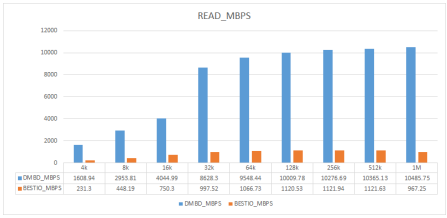
▶▶ 그림 2. Write 연산의 IOPS (BestIO vs. DMBD)



▶▶ 그림 3. Write 연산의 MBPS (BestIO vs. DMBD)



▶▶ 그림 4. Read 연산의 IOPS (BestIO vs. DMBD)



▶▶ 그림 5. Read 연산의 MBPS (BestIO vs. DMBD)

Ⅳ. 결론

이 논문에서는 클러스터의 다수 노드의 메모리를 블록 스토리지로 가상화하는 분산 인-메모리 스토리지 기술을 개발하였다. 블록 IO에 대한 복제를 수행하고 원본 블록IO를 제외한 노드에 복제한 블록IO를 기록하여 노드 고장에도 지속적으로 스토리지 서비스가 가능하도록 하였다. 실제 구현을 통해서 단일 노드에서의 BestIO 스토리지와 제안하는 DMBD를 이용하여 생성한 분산 인-메모리 스토리지간의 read/write 연산의 성능을 비교하였다. 비교 결과 read 연산의 경우 DMBD가 더 높은 성능을 보였으며 write 연산의 경우에는 유사한 성능을 보임을 알 수 있었다.

Ⅴ. 감사의 글

본 연구는 중소기업청의 창업성장기술개발사업의 일환으로 수행하였음. [S2495411, 고속 데이터 처리를 위한 NVDIMM 기반 인-메모리 블록 스토리지 기술개발]

▌참 고 문 헌▐

[1] Koutoupis, P., The linux RAM disk. LiNUX+ magzine, 2009, pp.36-39.
[2] https://wiki.debian.org/ramfs
[3] https://ignite.apache.org/
[4] https://www.alluxio.org/
[5] 전태인, 비휘발성 메모리를 이용한 DRAM과 HDD의 하이브리드 저장장치, 석사학위논문, 한국교통대학교, 2017
[6] https://nbd.sourceforge.io