NOHOST: 네트워크에 직접 연결이 가능한 플래시 기반 분산 저장서버

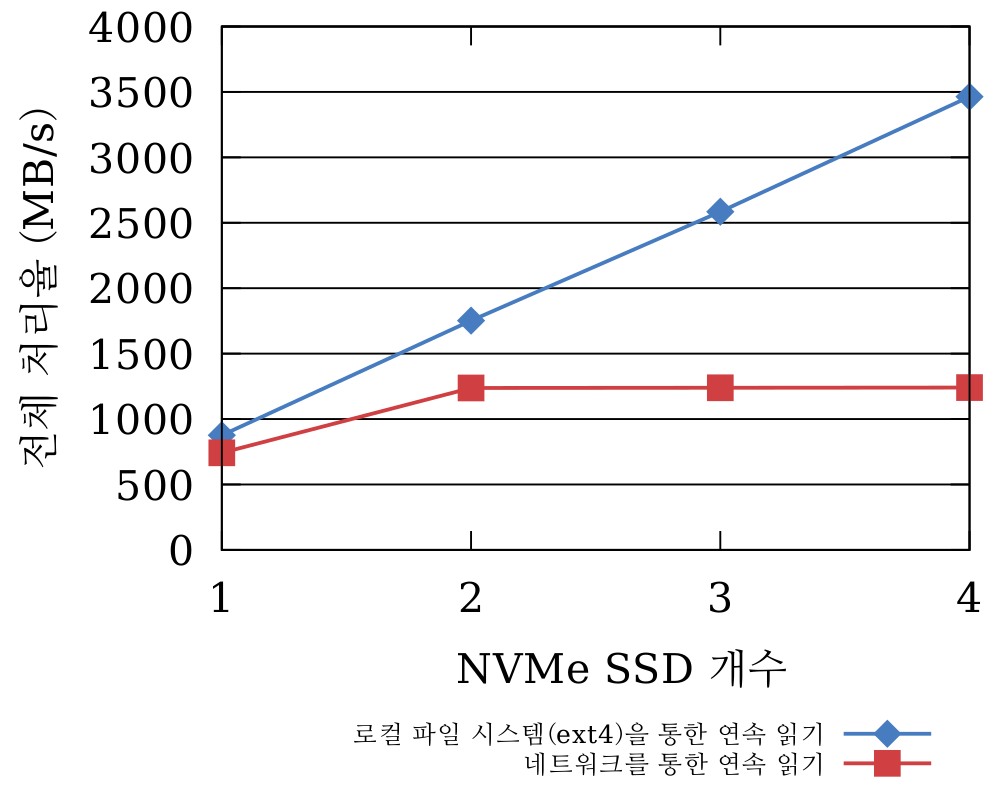
**최근 빅데이터 및 딥러닝 기반의 AI 기술의 발전으로 새롭게 생성되는 데이터의 양이 폭발적으로 증가하고 있다. 이러한 대량의 데이터는 데이터센터 내의 다수 저장서버들에 분산되어 저장된다. 현재 저장서버 구조들은 하드디스크(HDD) 기준으로 설계되었고, 따라서 SSD 사용시 SSD가 지닌 HDD보다 빠른 성능, 저전력성, 대용량이란 장점을 제대로 활용하지 못한다. 본 논문에서는 NOHOST라는 대용량 데이터 관리에 최적화된 분산 저장서버를 제안한다. NOHOST 시스템은 기존의 'x86 CPU + 대용량 DRAM + RAID 기반 저장장치 어레이' 기반의 저장서버를, 네트워크에 직접 연결된 다수의 SSD 구성으로 치환한다. 또한 각각의 NOHOST 노드는 단순하고 융통성 있는 Key-Value (KV) 인터페이스를 외부에 노출함으로써 다양한 응용 프로그램들이 효율적으로 데이터에 접근할 수 있도록 지원한다. 실제 SSD 프로토타입에 구현하고 평가한 결과, NOHOST는 기존 저장서버 대비 33% 수준의 낮은 공간 사용률 및 25% 수준의 낮은 전력 사용량을 요구했으며, 이와 함께 확장성 높고 우수한 I/O 성능을 제공하였다.**

# **1. 서론**

최근 빅데이터 기술과 딥러닝 기반 인공지능 기술의 빠른 발전, 많은 수의 사물 인터넷 기기로 인해 생성되는 데이터의 양이 폭발적으로 증가하면서 효율적인 데이터 저장 및 처리에 대한 요구 또한 함께 증가하고 있다.

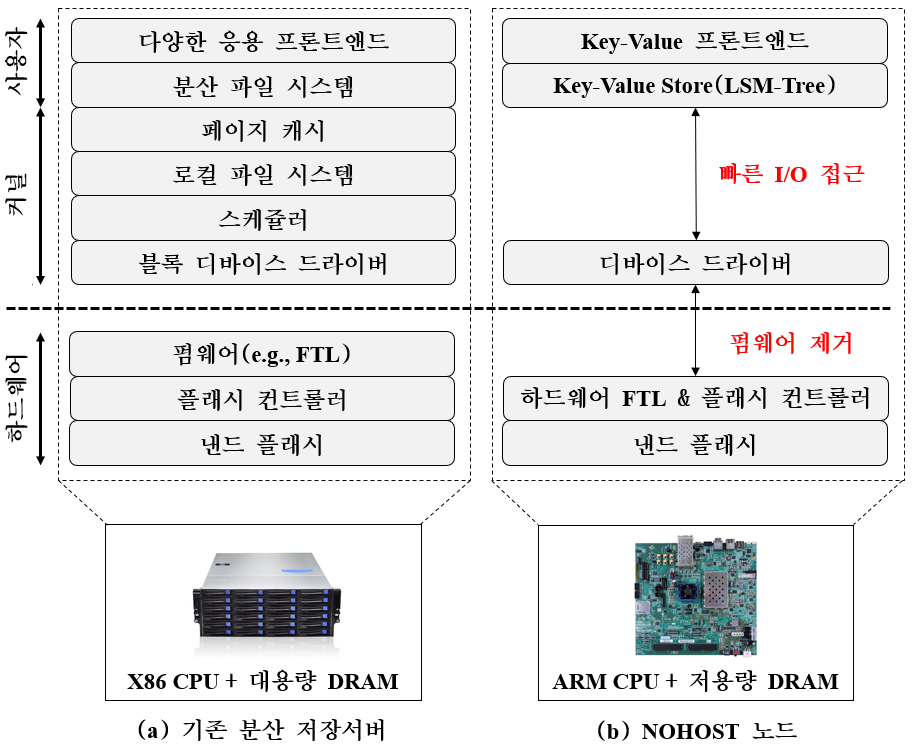
기존의 데이터센터는 데이터를 다수의 x86 기반 서버에 분산하여 저장하며, 응용 프로그램들은 네트워크를 통해 저장서버의 데이터에 접근하게 된다. 이러한 x86 기반 저장서버들은 Xeon과 같이 다수의 코어가 포함된 CPU와 대용량의 DRAM을 사용하기 때문에 매우 큰 공간(예, 6U의 랙(Rack) 공간)을 차지할 뿐만 아니라 많은 양의 전력(예, 800W)을 소모한다. 또한 기존 저장서버는 하드디스크(HDD)의 낮은 임의 읽기 및 쓰기 성능과 제한된 용량 문제를 극복하기 위해 대용량의 버퍼 메모리와 다수의 저장장치를 묶어 사용하는 RAID 기술을 사용한다.

상기 저장서버는 HDD에 기반하고 있어 SSD가 제공하는 우수한 성능(1-6GB/s)과 대용량의 저장공간(128TB)을 효율적으로 사용하지 못한다. 또한 다수의 SSD들의 최대 성능을 네트워크로 인해 100% 활용하지 못할 뿐만 아니라 불필요한 전력 소모 및 공간 낭비를 유발 시킨다. Fig. 1은 1개의 10G 이더넷 포트를 가지는 x86 기반 Xeon 서버의 읽기 성능을 서버에 연결된 약 880MB/s의 읽기 성능을 가진 NVMe SSD의 개수에 따라 나타낸 것이다. Fig.1에서 보여지듯 SSD 증가에 따른 성능 증가는 네트워크 성능으로 인해 한계가 있으며, 다수 SSD들의 최대 성능 또한 충족시키지 못한다. 즉, 고성능 CPU 및 대용량 DRAM을 사용하는 기존의 구조는 SSD를 사용하기에 효율적이지 못함을 보여준다.



**Fig. 1 제한된 네트워크 성능으로 인한 기존 서버 성능의 병목 현상**

본 논문에서는 기존 ‘x86 CPU + 대용량의 DRAM + RAID 저장장치’ 구조의 분산 저장서버가 가지는 공간과 전력 측면의 단점과 낮은 성능 확장성 문제를 해결하고 빅데이터의 저장 및 처리에 최적화된 NOHOST라는 시스템을 제안한다. NOHOST는 ‘ARM CPU + FPGA + 낸드 플래시’로 구성된 장치에 이더넷 인터페이스를 붙여 네트워크에 직접 연결하는 방식으로 기존의 저장서버를 대체한다. NOHOST의 각 노드는 네트워크에 직접 연결되기 때문에 개별 노드의 성능을 병목 없이 활용할 수 있으므로 높은 성능 확장성을 제공한다. NOHOST는 ARM 기반 보드로 구성되어 있기 때문에 기존 x86 기반 저장서버의 다양한 기능을 ARM 기반으로 이식 해야 할 필요성이 있다. 우리는 이러한 기술적 문제점을 (1) 불필요한 소프트웨어 계층(예, 로컬 파일 시스템, Flash Translation Layer(FTL), 분산 파일 시스템)의 제거, (2) 간결한 Key-Value(KV) 인터페이스를 이용한 데이터 관리, (3) ARM CPU에 맞는 가벼운 플래시 관리 알고리즘 개발을 통해 극복했다.



**Fig. 2 기존 분산 저장서버와 NOHOST 노드의 비교**

본 논문에서 제안한 NOHOST는 기존 x86의 하드웨어 및 소프트웨어를 재구성 함으로 낮은 공간 사용률(기존 대비 최소 33%)과 전력 사용량(기존 대비 최소 25%), 유연한 성능 확장성을 제공한다.

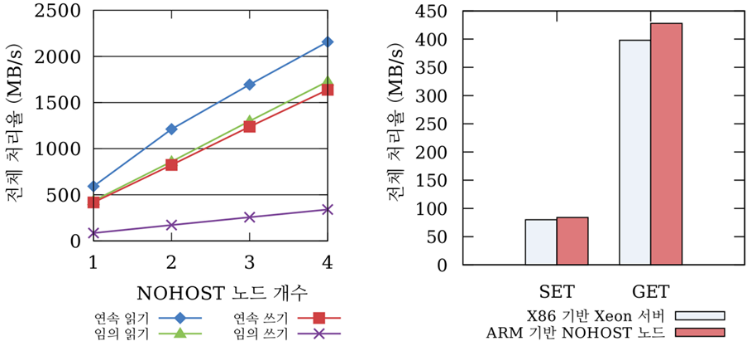
# **2. NOHOST 시스템의 구성**

Fig. 2는 기존 분산 저장서버와 NOHOST 노드의 하드웨어 및 소프트웨어 구성을 비교하고 있다.

NOHOST의 하드웨어 구성은 ARM CPU와 FPGA, 낸드 플래시로 구성된다. NOHOST는 FTL을 FPGA 기반 하드웨어로 구현하였으며, 이는 플래시 페이지 주소의 논리-물리 매핑과 배드 블록 관리를 수행하며 상위 계층에 블록 디바이스 인터페이스를 제공한다. 따라서 낸드 플래시를 이용하는 상위 응용은 낸드 플래시 관리를 할 필요가 없어져 소프트웨어 설계가 단순해 지고 예측 가능하게 되며, 소프트웨어 FTL의 부재로 성능 상의 이득을 얻을 수 있다.

NOHOST의 소프트웨어 구성은 (1) KV 프론트앤드, (2) KV Store(KVS), (3) 낸드 플래시를 위한 디바이스 드라이버로 구성된다. NOHOST는 낮은 성능의 ARM CPU에 맞는 플래시 관리 알고리즘을 위해 Log-Structured Merge Tree(LSM-Tree)[1] 기반 KVS를 구현하였다. NOHOST의 KVS는 기존 LSM-Tree와는 달리 실제 데이터의 이동 없이 메타 데이터만을 이동하는 Compaction, 임의 읽기 성능 향상을 위한 블룸 필터(Bloom Filter)의 사용, 메타 데이터 캐싱과 같은 최적화를 통해 실제 낸드 플래시의 최대 성능(읽기: 1.0GB/s, 쓰기: 430MB/s)을 활용한다. 또한 x86에 비해 ARM CPU에서 큰 부하를 가지는 메모리 복사를 줄이기 위해 미리 할당된 DMA 영역에 요청 데이터들을 유지하여 메모리 복사의 횟수를 최소화하고, 락-프리(Lock-Free) 큐의 사용이나 특정 CPU로 인터럽트 처리를 고정하는 등의 방법을 통해 ARM CPU에 최적화된 소프트웨어를 구현하였다.

KV 인터페이스로 구성된 NOHOST는 파일 시스템이나 데이터베이스와 같은 기존 응용과의 호환성을 위해 프로토콜 에뮬레이션을 제공한다.

**Fig. 3 (좌)노드 개수 증가에 따른 NOHOST 시스템의 성능, (우) X86 서버와 NOHOST 노드의 KV 연산 성능 비교**

해당 에뮬레이션은 읽기, 쓰기와 같은 연산을 KV연산으로 변경하는 어댑터를 클라이언트의 응용 프로그램에 부착하는 방식으로 구현되었다. 또한 클라이언트는 기존 이더넷 인터페이스를 통해 NOHOST의 노드와 통신할 수 있다.

# **3. NOHOST 실험 결과**

NOHOST는 4개의 1.3GHz ARM Core를 가진 CPU와 4GB의 DRAM, 10기가비트 이더넷으로 구성된 FPGA 기반 보드에 프로토타입으로 구현되었다. 실험을 진행한 클라이언트는 Xeon CPU와 32GB의 DRAM, 2개의 10기가비트 이더넷 포트로 구성되었다.

Fig. 3(좌)는 클라이언트가 NOHOST에 연속 읽기 및 쓰기와 임의 읽기 및 쓰기를 수행한 결과를 NOHOST 노드 개수에 따라 보여준다. Fig. 3(좌)에서 보여지듯 NOHOST는 높은 성능 확장성을 가진다.

Fig. 3(우)는 Xeon CPU, 32GB DRAM, 한 개의 NVMe SSD로 구성된 x86 저장서버에서 기존의 KV 기반 분산 소프트웨어를 수행하여 얻은 성능과, NOHOST 시스템의 단일 노드 성능을 임의 키에 대한 SET, GET 연산에 대해 비교한 것이다. NOHOST 노드는 낮은 CPU 성능에도 불구하고 기존 x86 서버 대비 약 5~10% 빠른 성능을 제공하는 것을 확인할 수 있다.

# **4. 결론**

기존 x86 기반 분산 저장서버는 대용량의 DRAM과 RAID 기반 저장장치로 인해 큰 공간과 많은 전력을 요구하고, 네트워크 성능의 제한에 따라 SSD들을 비효율적으로 활용하고 있다. 본 논문에서는 이러한 문제점을 해결하기 위해 작은 공간과 낮은 전력을 요구하는 ARM CPU와 낸드 플래시 기반 NOHOST 시스템을 제안한다. NOHOST는 하드웨어 FTL과 ARM에 최적화된 KV 소프트웨어를 통해 빠른 낸드 플래시의 성능을 활용하였고, 노드를 네트워크에 직접 연결함으로써 높은 성능 확장성을 제공하였다.

# **참고문헌**

[1] O’NEIL, P., CHENG, E., GAWLICK, D., & O’NEIL, E., “The log-structured merge-tree (LSM-tree)”, Acta Inf. 33, 4, 351–385, 1996