

대용량 영구 메모리 기반 실시간 빅데이터 검색 플랫폼 성능 분석

¹이은서, ^{2*}박동철

Performance Analysis of Real-Time Big Data Search Platform Based on High-Capacity Persistent Memory

¹Eunseo Lee, ^{2*}Dongchul Park

요약

다양한 빅데이터 기술의 발전은 많은 산업에 큰 영향을 미치고 있으며, 방대한 양의 데이터를 빠르게 처리하고 분석하기 위해 여러 연구가 진행되고 있다. 이러한 상황에서 인텔 차세대 대용량 영구 메모리 모듈이나 CXL과 같은 새로운 형태의 메모리와 컴퓨팅 기술이 크게 주목받고 있다. 그러나, 현존하는 대부분의 빅데이터 소프트웨어 플랫폼들은 여전히 기존의 전통적인 DRAM 환경을 기반으로 최적화되어 있으며, 특히 빅데이터 실시간 검색 플랫폼 관련 연구는 상대적으로 미흡한 실정이다. 본 연구에서는 차세대 영구 메모리인 인텔 옵테인 영구 메모리의 기본 성능을 평가하고, 옵테인 영구 메모리 기반 시스템에서 빅데이터 실시간 검색 플랫폼으로 유명한 Elasticsearch의 다양한 성능 분석 결과를 통해 대용량 영구 메모리의 효용성과 가능성을 검증한다. 본 논문은 대용량 영구 메모리 기반 시스템이 기존 DRAM 기반 시스템에 비하여 색인과 검색 측면에서 각각 1.45배, 3.2배의 성능 향상을 확인하였고, 이를 통해 고성능 I/O와 대용량, 비휘발성 등의 다양한 이점을 가진 차세대 영구 메모리가 Elasticsearch와 같은 빅데이터 검색 플랫폼에서 좋은 대안이 될 수 있음을 확인하였다.

Abstract

The advancement of various big data technologies has had a tremendous impact on many industries. Diverse big data research studies have been conducted to process and analyze massive data quickly. Under these circumstances, new emerging technologies such as high-capacity persistent memory (PMEM) and Compute Express Link (CXL) have lately attracted significant attention. However, little investigation into a big data “search” platform has been made. Moreover, most big data software platforms have been still optimized for traditional DRAM-based computing systems. This paper first evaluates the basic performance of Intel Optane PMEM, and then investigates both indexing and searching performance of Elasticsearch, a widely-known enterprise big data search platform, on the PMEM-based computing system to explore its effectiveness and possibility. Extensive and comprehensive experiments shows that the proposed Optane PMEM-based Elasticsearch achieves indexing and searching performance improvement by an average of 1.45 times and 3.2 times respectively compared to DRAM-based system. Consequently, this paper demonstrates the high I/O, high-capacity, and non-volatile PMEM-based computing systems are very promising for big data search platforms.

Keywords: Persistent Memory, Big Data, Optane, Elasticsearch, Search Engine

¹ 숙명여자대학교 컴퓨터과학과(les97@sookmyung.ac.kr)

^{2*} 교신저자 중앙대학교 산업보안학과 교수(dongchul@cau.ac.kr)

I. 서론

다양한 빅데이터 기술의 발전은 다양한 산업과 학계에 큰 영향을 미치고 있으며, 방대한 양의 데이터를 빠르게 처리하고 분석하기 위해 여러 연구가 진행되고 있다[1]. 이러한 상황에서 기존의 휘발성 메모리인 DRAM(Dynamic Random Access Memory)을 대체 또는 보조할 수 있는 차세대 대용량 영구 메모리 모듈인 인텔 옵테인 영구 메모리(Intel Optane Persistent Memory)[2]나, DRAM 모듈의 용량과 대역폭을 확장할 수 있도록 해주는 차세대 인터페이스인 CXL(Compute Express Link)[3] 등 새로운 형태의 메모리 기술이 주목을 받고 있다. 많은 기업들은 효율적으로 데이터를 처리하기 위해 여러 빅데이터 플랫폼을 사용하고 있지만 현존하는 대부분의 빅데이터 플랫폼들은 여전히 기존의 전통적인 DRAM 시스템을 기반으로 최적화되어 있다. 특히 국내에서 빅데이터 실시간 검색 플랫폼 관련 연구는 상대적으로 미흡한 실정이다.

본 논문에서는 벤치마크를 통해 차세대 영구 메모리인 인텔 옵테인 영구 메모리와 DRAM 간의 기본 성능 비교 평가를 제공한다. 또한, 영구 메모리 기반 컴퓨팅 시스템에서 빅데이터 실시간 검색 플랫폼으로 널리 쓰이는 Elasticsearch[4]의 성능을 다양하게 분석하여 빅데이터 검색 플랫폼에서 차세대 영구 메모리의 효용성과 가능성을 검증하고자 한다. 실험결과 기존 SSD (Solid State Drive) 대비 옵테인 영구 메모리의 고성능 I/O 를 통해 빠른 색인이 가능하며, DRAM 대비 대용량의 영구 메모리가 검색 응답 지연 시간을 감소시켜 기존 DRAM 기반 시스템보다 향상된 성능을 보여주었다. 대용량 데이터 검색 플랫폼의 특성상 색인 작업 시 많은 입출력이 필요하고, 데이터 검색 시에는 잦은 데이터 접근이 필요하다. 따라서 옵테인 영구 메모리의 고성능 I/O 와 대용량의 특성은 Elasticsearch 와 같은 엔터프라이즈 검색 플랫폼에 적합한 특성을 지니고 있을 뿐만 아니라, SAP HANA 와 같은 인메모리 데이터베이스나 Apache Spark 와 같은 인메모리 컴퓨팅 빅데이터 플랫폼의 경우처럼 대용량 메모리를 필요로 하는 빅데이터 플랫폼에도 큰 효용성이 있음을 검증한 연구가 존재한다[5][6].

본 논문에서는 가장 널리 알려진 빅데이터 검색 플랫폼인 Elasticsearch 에서 인텔 옵테인 영구 메모리의 효용을 검증하였으며, 색인이나 검색과 같은 다른 성능 특성을 필요로 하는 빅데이터 검색 플랫폼에서 다양한 성능 이점을 가진 차세대 영구 메모리가 좋은 대안이 될 수 있음을 확인하였다. 본 연구의 핵심 요약 결과는 다음과 같다.

- 2세대 인텔 옵테인 영구 메모리 (200 시리즈) 메모리 모드의 읽기 워크로드 대역폭은 DRAM 과 약 1.36 배의 차이를 보였으며, 읽기-쓰기 1:1 혼합 워크로드의 대역폭은 DRAM 에 준하는 성능을 보였다.
- 차세대 영구 메모리를 고성능 스토리지로 사용하는 경우 DRAM 기반 시스템에 비해 약 1.45 배 빠른 색인이 가능하다.
- 차세대 영구 메모리를 대용량 메모리로 사용하는 경우 DRAM 기반 시스템에 비해 검색 요청에 대한 응답이 최대 약 3.2 배 빠르다.

본 논문의 2 장에서는 차세대 영구 메모리와 관련된 기존의 연구들에 대해 소개한다. 3 장에서는 본 연구에서 사용한 인텔 옵테인 영구 메모리와 Elasticsearch 에 대해 설명한다. 4 장에서는 다양한 성능 실험 결과에 대하여 설명한다. 먼저 차세대 영구 메모리의 성능 평가 결과를 보여주고 실시간 검색 플랫폼의 성능 평가를 위한 시스템 환경과 실험 구성을 설명한다. 각 벤치마크의 성능 측정 결과와 분석 내용을 보여주며 시스템 구성을 평가한다. 5 장에서는 결론을 맺는다.

II. 관련연구

Y. Wu[7]는 DBMS(Database Management Systems)에서의 옵테인 영구 메모리 성능 분석을 위해, 마이크로소프트 SQL(Structured Query Language) 서버를 통한 OLTP(Online Transaction Processing)와 OLAP(Online Analytical Processing) 데이터베이스 워크로드를 평가하였다. 비휘발성 메모리를 인메모리 버퍼와 영구 스토리지로 사용하는 각각의 경우에 대하여 영구

메모리를 이용해 많은 이점을 얻기 위한 권장 사항을 제안하였다. 향후 DBMS 설계에서 병렬화 수준, I/O 요청 크기, 페이지 배치 정책 등에 대한 새로운 고려가 필요함을 분석하였다.

M. Weiland[8]는 슈퍼컴퓨팅에서 메모리와 I/O 병목 현상을 해결하기 위한 방법으로 인텔 옵테인 영구 메모리를 사용하였다. 과학 애플리케이션을 이용하여 성능과 효율성, 사용성 측면에서 메모리 모드와 앱 다이렉트 모드를 분석하였다. 메모리 모드는 애플리케이션의 변경 없이도 기존의 DRAM 전용 시스템에서 메모리 용량으로 인해 제한되었던 시뮬레이션에 대해 동등한 성능을 나타낼 수 있음을 확인하였다. 또한, 앱 다이렉트 모드의 스토리지가 작업 시 생성되는 데이터 경합을 줄인다고 분석하였다.

S. Akram[9]은 다중 코어 플랫폼에서 Psearchy 엔진 기반의 텍스트 인덱싱을 위한 옵테인 영구 메모리의 활용을 분석하였다. 옵테인 영구 메모리와 DRAM, 블록 스토리지를 이용하여 다양한 하이브리드 시스템의 구성을 평가하고 분석하였다.

이와 같이 차세대 영구 메모리와 관련된 여러 선행연구를 조사하였으며, 대용량 영구 메모리에 여러 빅데이터 소프트웨어 플랫폼을 적용하는 다양한 연구가 있었지만, 2세대 인텔 옵테인 영구 메모리의 성능 평가 결과를 제시하거나, 기업에서 가장 널리 사용되고 있는 실시간 검색 엔진인 Elasticsearch 에서 차세대 영구 메모리를 이용하여 성능 벤치마크를 수행하고 시스템 구성 별 성능을 평가한 연구는 확인할 수 없었다.

III. 배경 지식

3.1. 인텔 옵테인 영구 메모리

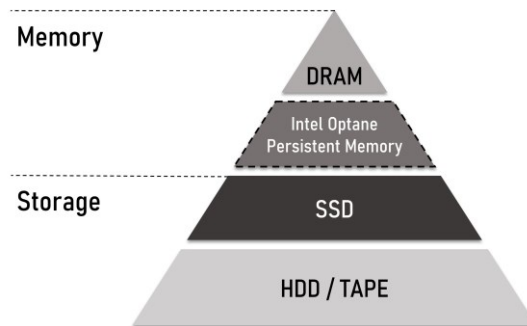


Figure 1. Memory Hierarchy
그림 1. 메모리 계층 구조 [10]

약 50여 년 동안 컴퓨터의 메인 메모리로 자리 잡고 있는 DRAM은 짧은 대기 시간의 장점이 있지만 높은 비용으로 용량이 제한적이며, 전원이 사라지면 데이터를 잃게 되는 휘발성의 속성을 가지고 있다. 블록 스토리지는 비용이 저렴하고 비휘발성의 속성을 가지고 있지만 데이터를 가져오는 속도가 DRAM에 비해 현저히 느리다. 이러한 상황에서 인텔은 DRAM을 대체, 또는 보조할 수 있는 차세대 대용량 영구 메모리 모듈인 인텔 옵테인 영구 메모리(Optane PMEM: Persistent Memory)를 발표하였다. [그림 1]은 인텔 옵테인 영구 메모리의 계층 구조를 보여준다. 인텔 옵테인 영구 메모리는 계층형 메모리와 향상된 스토리지를 지원하여 비용과 성능 측면에서 최적화가 가능하다[11].

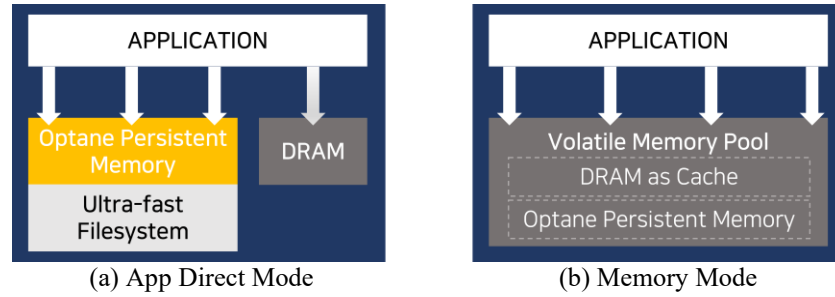


Figure 2. The Operating Modes of Intel Optane PMEM
 그림 2. 인텔 옵테인 영구 메모리 운영 모드 [12]

인텔 옵테인 영구 메모리는 앱 다이렉트 모드(App Direct Mode)와 메모리 모드(Memory Mode)의 두 가지 작동 모드를 제공한다. [그림 2]는 옵테인 영구 메모리의 운영 모드를 나타낸다. 먼저 앱 다이렉트 모드는 인텔 옵테인 영구 메모리를 고성능의 계층형 비휘발성 스토리지로 사용하는 옵션이다. 스토리지로 사용하는 경우 기존 블록 스토리지의 인터페이스를 사용하여 추가적인 수정을 하지 않아도 동작하며, 애플리케이션을 수정하여 커널을 우회하고 직접적인 Load/Store 작업을 할 수 있도록 만드는 것도 가능하다[13]. 앱 다이렉트 모드는 영구 메모리 인식 파일 시스템을 사용하는 운영 체제나 가상 환경이 필요하다. 또 다른 작동 모드인 메모리 모드에서는 인텔 옵테인 영구 메모리를 계층형 메모리로 사용한다. 해당 옵션에서는 영구 메모리가 대용량의 휘발성 메모리로 동작하고, DRAM은 L4 캐시(cache) 역할을 한다. 전원이 사라지면 휘발성의 속성을 가진 영구 메모리는 데이터를 잃게 된다[14].

3.2. 실시간 검색 플랫폼 : Elasticsearch

Elasticsearch는 아파치 루씬(Apache Lucene) 라이브러리 기반의 오픈 소스 검색 엔진이다. JSON(JavaScript Object Notation) 기반의 문서를 색인 또는 검색하거나 분석할 수 있다. 많은 양의 데이터를 빠르게 색인하고 검색할 수 있어 빅데이터 처리나 로그 모니터링, 문서 검색 등 여러 분야에 사용된다[15]. 색인 작업을 통해 원본 데이터를 검색어의 토큰(token)으로 변환하여 검색 가능하도록 변경한다. 모든 유형의 데이터에 대해 근실시간(Near Realtime, NRT) 검색과 분석을 제공한다[16].

Elasticsearch의 클러스터 구조는 [그림 3]과 같다. 클러스터에는 색인과 검색 작업에 참여하고 색인 데이터를 저장하는 노드(Node)가 존재하며, 각 노드에는 Elasticsearch의 인덱스를 이루는 샤드(Shard)가 존재한다. 노드가 여러 개 존재하는 경우 인덱스는 여러 노드에 분산 저장된다. 노드는 역할에 따라 마스터 노드, 데이터 노드, 인제스트 노드, 코디네이트 노드 등으로 구분할 수 있다[15]. 마스터 노드는 전체 클러스터를 관리하는 역할을 하고 데이터 노드에는 실제 색인 데이터가 저장된다. 환경 설정 시 노드에게 부여할 역할을 설정할 수 있으며 하나의 노드에 여러 역할을 할당하는 것도 가능하다. 클러스터 운영 시 여러 노드를 클러스터링 하여 목적에 맞게 구성하는 것이 일반적이지만, 단일 노드에서도 모든 작업을 수행하도록 할 수 있기 때문에 테스트 환경에서는 단일 노드로 구성해도 충분하다. 본 연구에서는 하나의 물리 서버를 사용하였기 때문에 단일 노드로 구성하여 실험을 진행하였다.

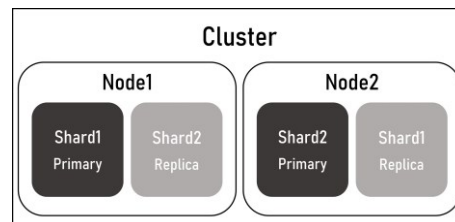


Figure 3. Elasticsearch Cluster Architecture
 그림 3. Elasticsearch 클러스터 구조

[그림 4]는 Elasticsearch의 인덱스 구조를 보여준다. 하나의 노드 안에는 여러 개의 인덱스가 존재하며, 인덱스는 색인된 데이터들이 나뉘어 저장되어 있는 물리적인 공간인 샤드가 여러 개 존재하는 형태로 이루어져 있다[15]. 이러한 형태는 데이터가 작업 수행 시 여러 샤드에서 병렬 처리될 수 있도록 해주기 때문에 성능과 처리량을 향상시킬 수 있다. 샤드는 원본 데이터를 저장하고 요청을 받아들이는 역할을 하는 프라이머리 샤드(Primary Shard)와 프라이머리 샤드의 복제본인 레플리카 샤드(Replica Shard) 두 가지로 구분된다. 레플리카 샤드는 프라이머리 샤드의 데이터를 복제하여 데이터 손상 시 장애에 대응할 수 있도록 한다. 프라이머리 샤드와 레플리카 샤드는 서로 다른 인덱스에 저장되어 데이터의 안정성을 보장한다. 샤드는 여러 개의 세그먼트(Segment)로 구성되어 있다. 세그먼트는 색인된 데이터가 존재하는 파일로 불변의 특성을 가지고 있으며, 데이터는 빠른 검색을 위한 역색인(Inverted Index) 구조로 저장되어 있다.

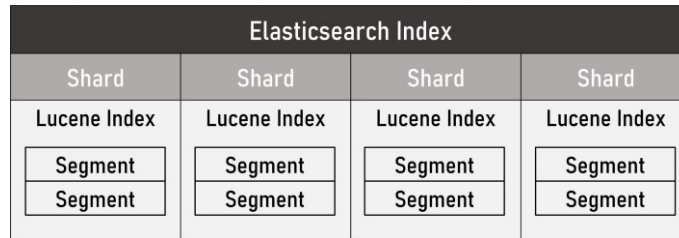


Figure 4. Elasticsearch Index Architecture

그림 4. Elasticsearch 인덱스 구조

IV. 성능 평가

4.1. 인텔 옵테인 영구 메모리 성능

[그림 5]와 [그림 6]은 2세대 인텔 옵테인 영구 메모리 모듈을 각각 앱 다이렉트 모드와 메모리 모드로 사용하였을 때의 대역폭을 보여준다. 영구 메모리로 동작하는 앱 다이렉트 모드의 경우 DRAM에 비하여 읽기, 읽기-쓰기 1:1 워크로드 모두 낮은 대역폭을 보였다. 그러나 대용량의 휘발성 메모리로 동작하는 메모리 모드인 경우 읽기 워크로드에서는 DRAM과 약 1.36 배의 차이로 앱 다이렉트 모드에 비해 높은 대역폭을 나타냈고, 특히 읽기-쓰기 혼합 워크로드에서는 DRAM에 준하는 대역폭이 측정되었다. [그림 7]은 메모리 모드로 동작하는 영구 메모리의 지연 시간을 보여준다. 요청이 들어오면 메모리 컨트롤러는 먼저 DRAM 캐시를 확인하고, 데이터가 존재하면 DRAM과 동일한 응답 시간을 보인다. 데이터가 존재하지 않을 경우 영구 메모리로부터 데이터를 읽어와 영구 메모리의 지연 시간이 나타난다. 따라서 메모리 컨트롤러가 예측할 수 있는 데이터 패턴이 나타나는 경우 더 높은 캐시 적중률을 보여 빠른 응답 시간이 측정되며, 임의 접근이 많은 워크로드에서는 성능 변화가 있다[17].

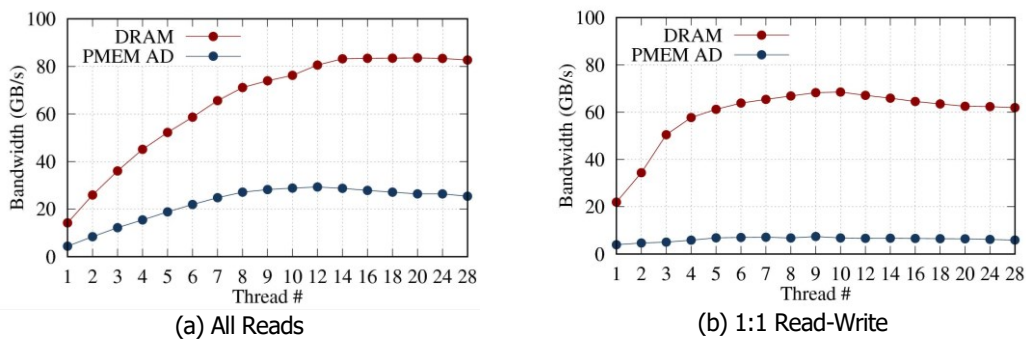


Figure 5. The App Direct Mode Bandwidth of Intel Optane PMEM

그림 5. 옵테인 영구 메모리 앱 다이렉트 모드 대역폭

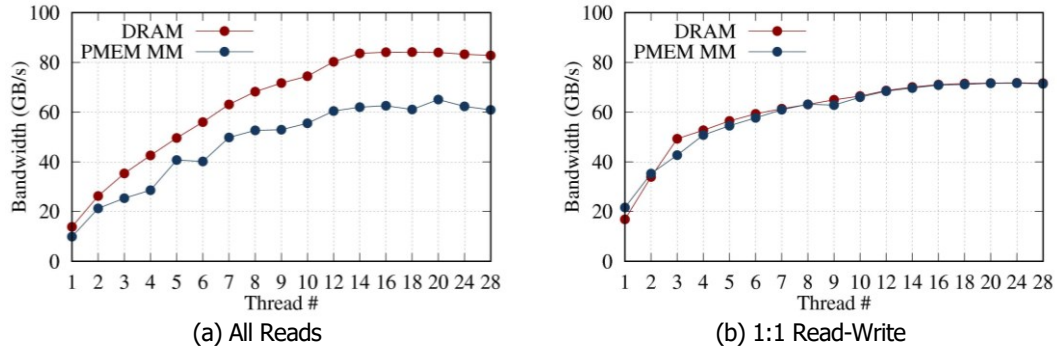


Figure 6. The Memory Mode Bandwidth of Intel Optane PMEM

그림 6. 메모리 모드 대역폭

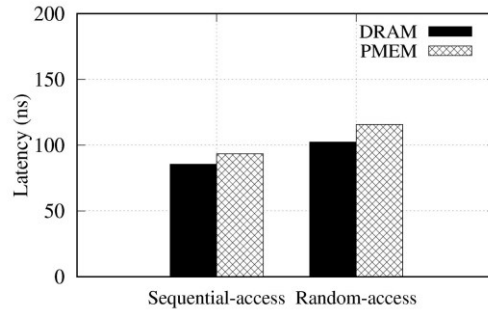


Figure 7. The Memory Mode Latency of Intel Optane PMEM

그림 7. 메모리 모드 지연 시간

4.2. Elasticsearch 실험 구성

본 연구에서는 차세대 대용량 영구 메모리에서 실시간 검색 플랫폼인 Elasticsearch의 색인과 검색 성능을 평가하고 분석하여, Elasticsearch에서 효율적으로 데이터 처리를 할 수 있는 시스템 구성 정보를 제공한다. [Table 1]과 [표 2]는 본 연구에서 사용한 컴퓨팅 시스템의 사양과 구성을 보여준다. 2개의 인텔 옵테인 영구 메모리 128GB 모듈을 사용하여 총 256GB의 대용량 영구 메모리를 이용한 시스템과, 전통적인 휘발성 메모리인 DRAM을 이용한 시스템을 각각 구성하였다. 차세대 영구 메모리 기반 시스템은 영구 메모리를 고성능 스토리지로 사용하는 애플다이렉트 모드(PMEM-AD)와 대용량 메모리로 사용하는 메모리 모드(PMEM-MM)를 이용한 환경으로 구성하였다. 차세대 영구 메모리 기반 컴퓨팅 시스템 환경에서는 2개의 SATA SSD를 RAID0 방식으로 구성하여 스토리지로 이용하였다. 영구 메모리가 고성능 스토리지의 역할을 하는 PMEM-AD 환경의 경우 PMEM과 RAID로 구성된 스토리지에 1:1 비율로 데이터가 저장되도록 하였다.

Table 1. Server Configurations

표 1. 서버 사양

| Name | Configurations |
|---------------|---|
| CPU | 3 rd Gen Intel Xeon Scalable Gold 6300 @ 2.0GHz (28 Cores) |
| RAM | DDR4 2933 REG/ECC 64GB (16GB × 4) |
| Motherboard | ASRock Rack ASPR SPC621D8-2L2T |
| Storage (SSD) | Samsung 970 EVO M.2 NVMe 1TB, Seagate Barracuda 1TB (500GB × 2) |
| PMEM | Intel Optane Persistent Memory 200 Series 256GB (128GB × 2) |
| OS | Ubuntu 20.04.1 LTS |
| Software | Elasticsearch v7.17.0, Rally v2.3.1 |

Table 2. System Configuration
표 2. 시스템 환경 별 구성

| | DRAM | Persistent Memory | SSD |
|---------|--------------|-------------------------|------------------|
| DRAM | 64GB | - | NVMe 1TB |
| PMEM-AD | 64GB | 256GB (App Direct Mode) | SATA 1TB (RAID0) |
| PMEM-MM | 64GB (Cache) | 256GB (Memory Mode) | SATA 1TB (RAID0) |

성능 평가를 위해 Elastic 에서 제공하는 매크로벤치마킹(Macrobenchmarking) 프레임워크인 Rally[18] 벤치마크를 사용하였다. Rally 는 Elasticsearch 의 성능 벤치마크 결과를 제공하여 성능 결과를 비교하거나 성능 문제를 발견할 수 있도록 해준다. 트랙(track)이라고 하는 벤치마킹 시나리오를 선택하여 성능 평가를 수행할 수 있다. 본 연구에서는 Elastic 웹 사이트에서 생성된 로그를 기반으로 HTTP 접근 로그를 색인하고 검색하는 eventdata 트랙[19]을 사용하여 성능 평가를 수행하였다.

[표 3]은 각 컴퓨팅 시스템의 Elasticsearch 성능 평가를 위한 실험 환경을 보여준다. Elasticsearch 에서는 시스템에서 사용 가능한 메모리의 50%를 힙 메모리(Heap Memory)로 제공하고, 메모리의 양이 64GB 이상이더라도 힙 메모리를 32GB 이하로 설정하도록 권장하고 있다[20]. 실제로 실험을 진행한 결과 힙 메모리를 30GB 까지 증가시켰을 때 힙 메모리의 크기에 따라 성능이 증가하였지만, 힙 메모리를 30GB 보다 더 많이 할당하더라도 성능이 증가하지 않거나 오히려 감소하는 경향을 보였다. 따라서 본 연구에서는 전체 시스템의 메모리 양에 관계없이 힙 메모리에 30GB 만을 할당하고 캐시가 가능하도록 여유 공간을 두었다.

Elasticsearch 와 같이 I/O 의 속도가 성능에 영향을 미치는 플랫폼에서는 스토리지의 I/O 속도에 따라 병렬 처리 요청을 효율적으로 처리할 수 있는지 여부가 달라진다[21]. 따라서 시스템에서 효율적으로 데이터를 처리할 수 있도록 적절하게 최적화하는 것이 필요하다. 실험을 진행한 결과 본 연구에서 사용한 스토리지의 경우 프라이머리 샤드(Primary Shard)의 개수가 8 개일 때 평균적으로 가장 높은 처리량을 나타냈고, 그 이상으로 증가시키는 경우 스토리지 속도에 비해 요청되는 병렬 처리의 양이 증가하여 모든 요청을 즉시 처리하기 어렵게 되어 성능이 감소하였다. 따라서 성능 평가에서 가장 높은 성능을 확인할 수 있도록 프라이머리 샤드의 개수를 8 개로 설정하고 실험을 진행하였다. 마찬가지로, 단일 호출에서 수행하는 작업의 수를 조정하여 색인 성능을 최적화할 수 있는 Bulk[22]의 경우 서버의 CPU 코어 개수인 28 개로 조정하였을 때 가장 높은 성능을 보였기 때문에 해당 값으로 설정하였다.

Table 3. Elasticsearch Configurations
표 3. Elasticsearch 성능 실험 환경 설정

| | |
|--------------------------|------|
| JVM Heap Memory | 30GB |
| Number of Primary Shards | 8 |
| Bulk Indexing Clients | 28 |

4.3. 색인 성능

색인 성능 벤치마크 실험은 10 억 개의 이벤트(event)를 색인하여 디스크에 약 200GB 의 인덱싱 데이터를 생성한다. 각 시스템에서의 Elasticsearch 색인 성능을 확인할 수 있다. 성능 분석을 위해 동일한 환경에서 벤치마크를 실행하고 수행 시간, 평균 처리량, 자원 사용량을 측정하였다.

[그림 8]은 각 환경에서의 색인 벤치마크 결과를 보여준다. 평균 처리량이 높을수록 같은 시간 동안 많은 데이터를 색인할 수 있음을 의미한다. 색인 성능의 경우 영구 메모리를 계층형 스토리지로 사용한 PMEM-AD 환경에서 가장 높은 처리량을 나타냈다. PMEM-AD 환경은 DRAM 환경에 비하여 약 1.45 배 높은 색인 처리량을 보였다. 영구 메모리를 대용량 메모리로 사용한 PMEM-MM 환경과 DRAM 환경의 경우 주목할 만한 성능 차이가 나타나지 않았으며, 색인 데이터가 저장되는 스토리지의 사용률이나 가비지 컬렉션(Garbage Collection)의 작동 여부에 따라 성능이 크게 변화하였다. 그럼에도 DRAM 환경보다 더 높은 지연 시간이 측정되는 메모리와, 더 낮은 입출력 속도를 가진 스토리지를 사용하는 PMEM-MM 환경에서의 성능 평가

결과가 DRAM 환경과 비교하여 비슷한 성능이 나타나는 것은 Elasticsearch 색인 작업에서 나타나는 대용량 메모리의 이점 때문이다. Elasticsearch 는 색인 작업 시 데이터를 빠르게 검색할 수 있도록 여러 캐시를 사용한다. 운영 체제 레벨에서 관리되는 페이지 캐시(page cache)는 색인되는 데이터의 내용을 메모리에 저장하여 같은 데이터에 대한 요청이 일어나는 경우 디스크 접근을 줄이고 빠르게 처리할 수 있도록 해준다. 색인이 진행되면서 페이지 캐시의 양이 늘어나 한정된 메모리가 가득 찰 경우 새로운 데이터를 위해 페이지를 효율적으로 교체하는 작업이 필요하다[23]. [그림 9]는 색인 벤치마크가 수행되는 동안 PMEM-MM 환경과 DRAM 환경에서의 캐시 메모리 양의 변화를 보여준다. 64GB 의 메모리를 가진 DRAM 환경의 경우 최대 약 21GB 정도의 캐시 데이터가 저장되는 반면, 256GB 대용량 메모리를 가진 PMEM-MM 환경의 경우 약 188GB 의 캐시 데이터를 저장하는 것을 확인할 수 있다. PMEM-MM 환경에서는 충분한 메모리로 인해 새로운 색인 데이터를 위한 페이지 교체 작업을 DRAM 보다 적게 수행하더라도 새로운 데이터에 대한 캐시를 유지할 수 있어 성능 측면에서 이점을 얻을 수 있다.

Elasticsearch 의 색인 성능은 스토리지의 I/O 성능에 가장 크게 영향을 받으며, 차세대 영구 메모리를 고성능 스토리지로 사용하여 색인 데이터를 저장하도록 하는 경우 색인 작업에서 높은 성능을 얻을 수 있다. 또한 차세대 영구 메모리를 대용량 메모리로 사용하는 경우에도 DRAM 환경에 준하는 색인 성능을 낼 수 있음을 확인하였다.

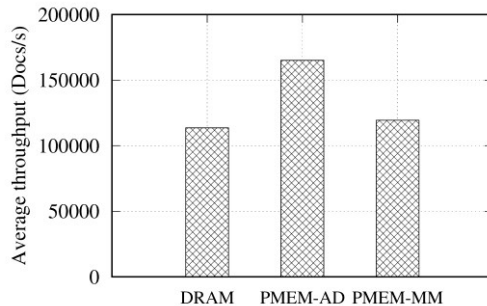


Figure 8. Average Indexing Throughput
그림 8. 색인 평균 처리량

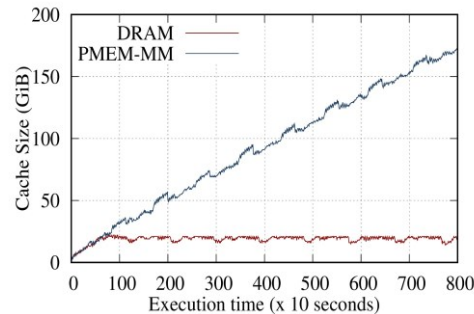


Figure 9. Cache Size
그림 9. 캐시 메모리 크기

4.4. 검색 성능

검색 성능 벤치마크 실험은 색인 벤치마크 성능 실험에서 진행했던 색인 데이터의 검색 요청에 대한 성능을 확인할 수 있다. 본 연구에서는 18000 초 동안 요청을 수행한 결과에 대한 지연 시간을 분석하였다. 검색 벤치마크에서는 색인 작업이 동시에 수행되지 않는다.

[그림 10]은 각 환경에서 무거운 워크로드를 요청하였을 때 쿼리 별 지연 시간을 보여준다. 빠른 지연 시간을 보일수록 요청을 빠르게 처리할 수 있음을 나타낸다. 검색 벤치마크 성능 측정 결과가 가벼운 워크로드에서는 지연 시간에서 눈에 띄는 차이가 나타나지 않았지만, 무거운 워크로드에서는 대용량 메모리를 가진 PMEM-MM 환경에서 평균적으로 1.09 배 빠른 지연 시간을 보였다. 이는 색인 시 PMEM-MM 환경에서 더 많은 데이터를 캐시로 저장할 수 있어, PMEM 의 메모리 지연 시간이 DRAM 의 지연 시간보다 낮음에도 불구하고 캐시가 되지 않은 데이터에 대한 검색에 대해서는 더 적은 용량의 메모리를 가진 DRAM 환경이나 PMEM-AD 환경보다 빠르게 요청을 처리할 수 있기 때문이다. PMEM-AD 환경은 같은 용량의 메모리를 사용함에도 불구하고 DRAM 환경보다 더 큰 지연 시간이 측정되었다. 이는 앱 다이렉트 모드 동작 방식으로 인하여 나타나는 것인데, PMEM 의 앱 다이렉트 모드에서는 메모리 접근 요청이 DRAM 을 거치지 않고 PMEM 에 독립적으로 작동하여 페이지 캐시의 효과를 얻기 어렵기 때문이다. PMEM 을 빠른 스토리지로 인식하여 색인 데이터를 저장하는 PMEM-AD 환경에서는 DRAM 환경보다 색인 시 저장되는 페이지 캐시의 양이 상대적으로 적기 때문에 요청을 처리하는데 더 긴 시간이 소요되었다.

Elasticsearch 는 메모리의 양이 많을수록 많은 캐시 데이터를 이용할 수 있기 때문에 검색

요청에 더 빠르게 응답할 수 있다. 따라서 차세대 영구 메모리를 계층형 메모리로 사용하는 경우 대용량의 메모리가 이점이 되어 검색 작업에서 효율적인 데이터 처리가 가능하며, DRAM의 크기보다 큰 용량의 데이터가 대용량 메모리에 캐시 되어 있는 상황에서 요청을 처리하는 경우 DRAM 환경에 준하는 지연 시간을 나타내는 것을 확인하였다.

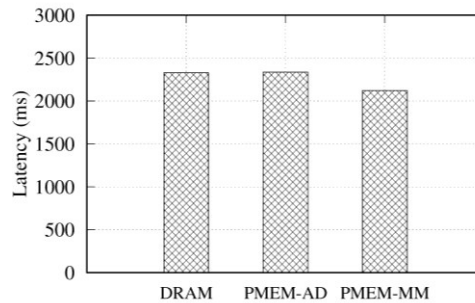


Figure 10. Searching Latency

그림 10. 검색 지연 시간

4.5. 색인/검색 혼합 성능

색인/검색 성능 벤치마크 실험은 색인 벤치마크 성능 실험이 동시에 수행될 때 색인 데이터에 대한 검색 요청 결과를 측정한다. 테스트 환경이 아닌 상황에서 Elasticsearch를 사용할 때 색인과 검색 작업을 함께 수행하는 경우가 일반적이기 때문에 보다 현실적인 벤치마크 결과를 확인할 수 있다. 과거에 색인이 일어났던 데이터(Relative)와 현재 색인 작업이 일어나고 있는 데이터(Current)에 대한 검색 요청의 지연 시간을 각각 측정하였다. 색인과 검색이 동시에 이루어지는 경우 검색 요청에 대한 지연 시간은 동일한 색인 처리량을 기준으로 비교하는 것이 좋다. 본 연구에서는 색인 작업 시 세 가지 환경 모두 도달할 수 있는 평균 처리량인 100,000 Docs/s를 목표 처리량으로 설정하였다.

[그림 11]과 [그림 12]는 100,000 Docs/s를 목표 처리량으로 설정하였을 때 Relative와 Current의 지연 시간을 나타낸다. 성능 측정 결과 Current의 지연 시간은 환경에 따라 뚜렷한 차이가 나타나지 않은 반면, Relative의 지연 시간은 무거운 워크로드 기준으로 PMEM-MM 환경이 DRAM 환경에 비하여 약 3.2배, PMEM-AD 환경에 비하여 약 5.14배 빠른 지연 시간으로 가장 높은 성능을 보였다. 이는 이미 색인이 일어났던 데이터에 대한 요청인 Relative의 경우 캐시의 영향을 받기 때문에, 대용량 메모리로 많은 캐시 데이터를 가지고 있는 PMEM-MM 환경에서 더 빠르게 응답할 수 있기 때문이다. PMEM-AD 환경에서는 상대적으로 캐시의 효과를 얻기 어려워 가장 긴 지연 시간이 나타났다. 검색 벤치마크 실험 결과와 마찬가지로 무거운 워크로드에서 대용량 메모리의 장점이 두드러지게 나타났다.

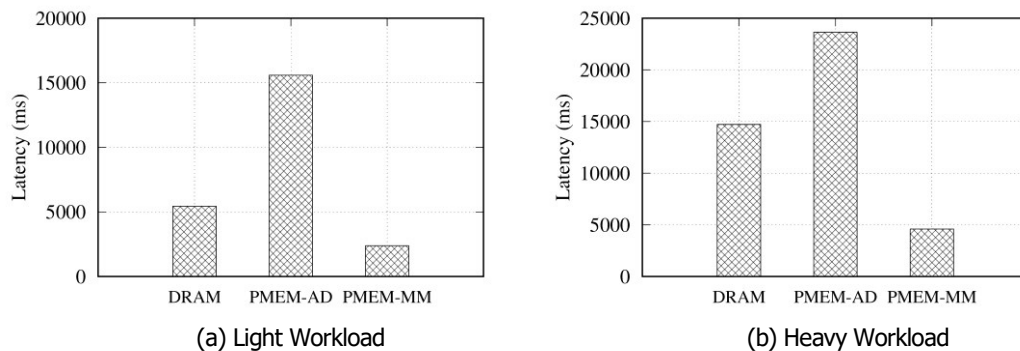


Figure 11. Indexing/Searching Latency (Relative)

그림 11. 색인/검색 혼합 지연 시간 (Relative)

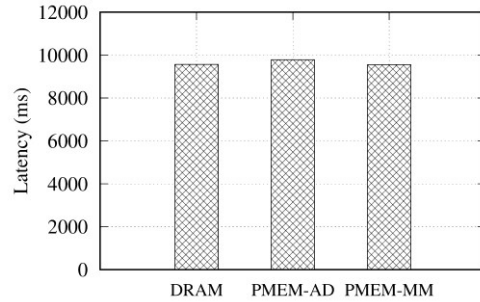


Figure 12. Indexing/Searching Latency (Current)
그림 12. 색인/검색 혼합 지연 시간 (Current)

같은 컴퓨팅 환경에서는 목표 처리량이 증가할수록 요청에 대한 처리가 늦어지는 것을 확인할 수 있었다. [그림 13]은 목표 처리량에 따른 지연 시간의 변화를 보여준다. 동일한 환경에서 목표 처리량을 50,000 Docs/s 와 150,000 Docs/s 로 각각 설정하여 벤치마크를 수행하고 지연 시간을 측정하였다. 가벼운 워크로드와 무거운 워크로드 모두 목표 처리량이 증가하면 지연 시간이 길게 나타나는 것을 확인하였다.

실시간 검색 엔진에서 색인 작업과 검색 작업을 동시에 수행하는 경우 검색 성능 면에서 차세대 영구 메모리의 대용량 메모리로 이점을 얻을 수 있음을 알 수 있다. 또한, 색인 처리량과 검색 지연 시간에는 상충 관계가 존재하며 효율적인 시스템을 위해서는 균형이 필요하다.

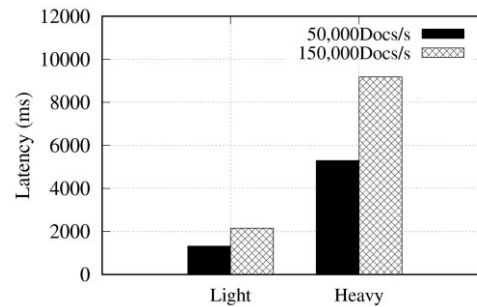


Figure 13. Latency Change over Target Throughput
그림 13. 목표 처리량에 따른 지연 시간 변화

V. 결론

Table 4. Summary of Performance Evaluation Results

표 4. 성능 실험 결과 요약정리

| | Indexing Throughput (Docs/s) | Searching Latency (ms) | Indexing/Searching Latency (ms) |
|------------------|------------------------------|------------------------|---------------------------------|
| DRAM | 113,669 | 2,328.96 | 14,715.4 |
| PMEM-AD | 165,077 | 2,336.79 | 23,647.6 |
| PMEM-MM | 119,442 | 2,121.35 | 4,598.55 |
| Best Performance | PMEM-AD (1.45 x) | PMEM-MM (1.09 x) | PMEM-MM (3.2 x) |

본 논문에서는 2 세대 인텔 옵테인 영구 메모리(Optane PMEM)의 성능 평가 결과를 제공하였다. 또한, 빅데이터 검색 플랫폼인 Elasticsearch 에 적용하여 빠른 데이터 처리가 가능함을 보이고, 작업에 따라 가장 효율적으로 차세대 영구 메모리를 이용할 수 있는 방법을 제공하였다. Elasticsearch 색인 작업 시에는 차세대 영구 메모리를 계층형 스토리지로 사용하여 고성능 I/O 의 이점으로 기존 DRAM 기반 시스템에 비해 약 1.45 배 빠른 색인이 가능하며, 대용량의 메모리가

색인 성능 향상에도 영향을 미치는 것을 확인하였다. 검색 작업 시에는 대용량의 메모리가 검색 요청을 DRAM 기반 시스템에 비해 약 3.2 배까지 빠르게 처리할 수 있으며, 이전에 색인된 데이터에 대한 무거운 워크로드일수록 더 효과적으로 요청을 처리할 수 있었다 ([표 4] 참조).

본 논문은 고성능 I/O와 대용량 메모리 등의 다양한 성능 이점을 가진 차세대 영구 메모리가 작업마다 다른 성능 특성을 필요로 하는 Elasticsearch와 같은 검색 플랫폼에서 좋은 대안이 될 수 있음을 확인하였으며, 향후 새로운 형태의 메모리가 나타나더라도 본 논문에서 제공한 정보를 통해 효율적으로 실시간 검색 플랫폼에 적용하여 성능을 향상시킬 수 있을 것으로 기대한다.

VI. 감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터지원사업의 연구결과로 수행되었음 (IITP-2023-2018-0-01799).

VII. 참고문헌

- [1] Z. Zheng, P. Wang, J. Liu, and S. Sun. (2015). Real-time big data processing framework: challenges and solutions. [Online]. Available: <https://www.naturalspublishing.com/files/published/v6910010rnl56m.pdf>.
- [2] Intel Optane Persistent Memory, <https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/overview.html>.
- [3] Compute Express Link (CXL), <https://www.computeexpresslink.org/>.
- [4] Elasticsearch, <https://www.elastic.co/>.
- [5] Scale Up Memory for SAP HANA Performance, <https://www.intel.com/content/www/us/en/big-data/partners/sap/intel-it-sap-hana-persistent-memory-infographic.html>
- [6] H.J. Oh. (2022). The performance analysis of big data processing platforms using next-generation large-capacity persistent memory (Master's thesis). Sookmyung Women's University, Department of Computer Science.
- [7] Y. Wu, K. Park, R. Sen, B. Kroth, and J. Do, "Lessons learned from the early performance evaluation of Intel optane DC persistent memory in DBMS," in Proc. of the 16th International Workshop on Data Management on New Hardware (DaMoN), No. 14, pp. 1–3, June. 2020.
- [8] M. Weiland, et al. "An early evaluation of Intel's optane DC persistent memory module and its impact on high-performance scientific applications," in Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), No. 76, pp. 1-19, Nov. 2019.
- [9] S. Akram, "Exploiting Intel Optane Persistent Memory for Full Text Search," ISMM 2021, pp. 80-93, June. 2021.
- [10] Intel Corporation. (2019). Memory Tiering: Improving Data Management (White Paper). Available: <https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/memory-tiering-improving-data-management-paper.html>.
- [11] Intel Optane Persistent Memory 200 Series Brief, <https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/optane-persistent-memory-200-series-brief.html>.
- [12] "Intel Optane DC Persistent Memory Readies for Widespread Deployment," Intel Newsroom, October 30, 2018. Available: <https://newsroom.intel.com/news/intel-optane-dc-persistent-memory-readies-widespread-deployment/>.
- [13] Izraelevitz, Joseph, et al. "Basic performance measurements of the intel optane DC persistent memory module," arXiv preprint arXiv:1903.05714. 2019.
- [14] Y.S. Lee, Y.J. Woo, and S.I. Jung, "Trend of Intel Nonvolatile Memory Technology," Electronics and Telecommunications Trends (ETRI), Vol. 35, No. 3, pp. 55-65. June. 2020.
- [15] S.H. Park, et al. (2021). Practical Know-how for Elasticsearch Operations from the Basics. Insight. pp 1-20, pp 75-107.
- [16] What is Elasticsearch?, <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/elasticsearch->

- intro.html.
- [17] I. B. Peng, M. B. Gokhale, E. W. Green, "System Evaluation of the Intel Optane Byte-addressable NVM," In Proceedings of the International Symposium on Memory Systems. 2019, pp. 304-315.
- [18] Rally, <https://github.com/elastic/rally/>.
- [19] Rally-eventdata-track, <https://github.com/elastic/rally-eventdata-track>.
- [20] Important Elasticsearch configuration, <https://www.elastic.co/guide/en/elasticsearch/guide/current/heap-sizing.html>.
- [21] E. Lee, H. Oh, and D. Park, "Big data processing on single board computer clusters: Exploring challenges and possibilities," IEEE Access, vol. 9, pp. 142551-142565, 2021.
- [22] Elasticsearch Bulk API, <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/docs-bulk.html>.
- [23] S. Lim, and D. Park, "Efficient stack distance approximation based on workload characteristics," IEEE Access, vol. 10, pp. 59792-59805, 2022.

저자소개



이은서 (Eunseo Lee)

2021 년 2 월 숙명여자대학교 컴퓨터과학과 학사
2023 년 2 월 숙명여자대학교 컴퓨터과학과 석사

관심분야 : 빅데이터 처리 플랫폼, 스토리지 시스템, NVM



박동철 (Dongchul Park)

2017 년 8 월 미국 인텔 연구소 수석연구원
2023 년 2 월 숙명여자대학교 소프트웨어학부 조교수
2023 년 3 월 ~ 현재: 중앙대학교 산업보안학과 부교수

관심분야 : 빅데이터 플랫폼, 스토리지, 클라우드, NVM 등