BOSS: Bandwidth-Optimized Search Accelerator for Storage-Class Memory

Jun Heo, Seung Yul Lee, Sunhong Min, Yeonhong Park, Sung Jun Jung, Tae Jun Ham, Jae W. Lee

ISCA' 21

Department of Computer Science, Dankook University

System Software Lab.

Suhwan Shin

09/04/2024

Contents

Introduction

- **Full-text search** is an important web service, and **inverted index** is a core data structure for document information management, but existing systems have **difficulty** cost-effectively meeting **large memory requirements**

- **Storage-Class Memory (SCM)** based memory pools are a **cost-effective** alternative for **capacity** expansion, but they suffer from **limited bandwidth** of the shared interconnect between SCM devices and host CPUs

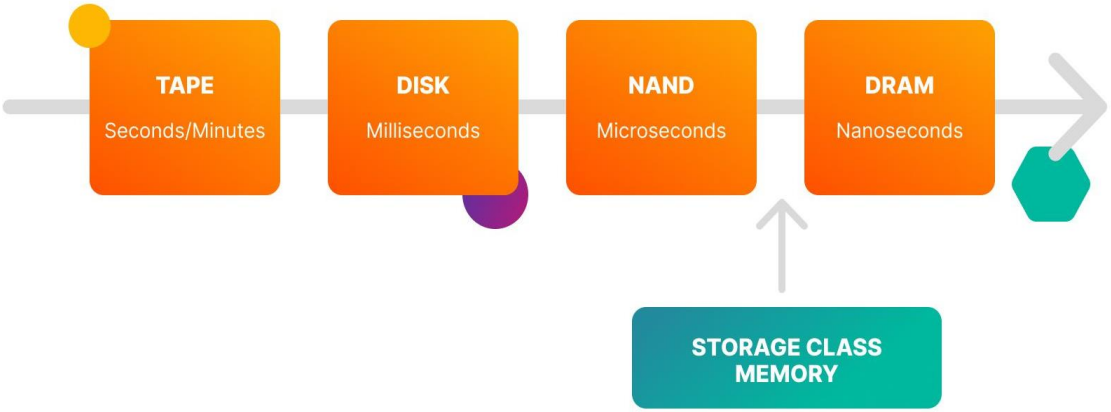
- **BOSS** proposes the **near-data processing (NDP)** architecture for inverted index search in SCM-based pooled memory, maintaining **high query processing** throughput even in **bandwidth - constrained environments**

Background - Storage Class Memory (SCM)

- SCM is a new tier in the memory hierarchy to bridge the gap between memory & disk (Memory: DRAM, Disk: HDD/SSD)

- SCM is non-volatile(persistent memory) and acts as another pool of server memory

- SCM Advantages
  - Lower latency than traditional NAND flash storage
  - Faster write-intensive workloads (ex. Online Transaction Processing)
  - Lower cost than DRAM storage



https://www.purestorage.com/kr/knowledge/what-is-storage-class-memory.html

Background - Inverted Index

- Definition
  - Inverted Index is a standard data structure for efficiently managing document information in specialized search engines
- Structure
  - A set of key-value pairs (called posting lists)
  - Key: Searchable term
  - Value: List of documents containing the term
- Challenges
  - Memory usage on large datasets
  - Efficient compression and decompression
  - Fast set operations

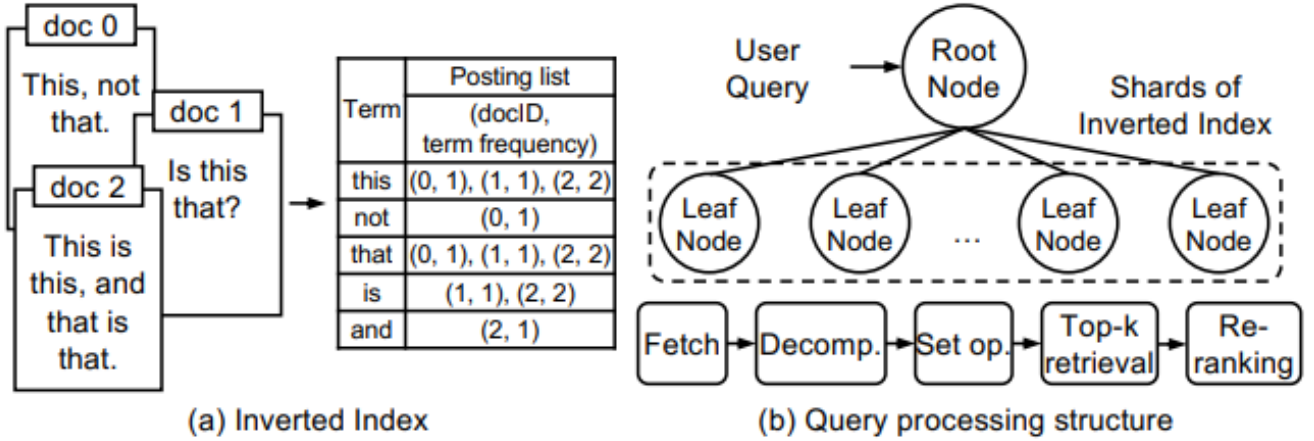


Fig. 1. Query processing using an inverted index

Background - IIU (State of the art)

- IIU: A hardware accelerator specifically designed for inverted index search
- Key functions
  - Decompression
  - Set operations (intersection, union)
  - Score calculation
- Advantages
  - Provides performance specialized for inverted index search compared to general-purpose CPUs or GPUs
- Limitations
  - Performance not optimized for SCM-based memory systems
  - Frequent random memory accesses due to binary search-based intersection algorithm
  - Unnecessary data search due to lack of pruning mechanisms
  - Support limited to specific compression schemes

Motivation 1

- Advantages & Constraints of SCM-based memory pools
  - Advantages
    - High capacity: Offers much larger capacity than DRAM (e.g., Intel Optane DCPMM supports up to 512GB per channel)
    - Cost-effectiveness: Lower cost per bit compared to DRAM, suitable for large-scale data processing
    - Scalability: Capacity can be scaled almost infinitely through memory disaggregation without additional CPU sockets
  - Constraints (Bandwidth)
    - SCM devices have lower bandwidth than DRAM (about 3-6 times difference)
    - Shared interconnect bandwidth to the host CPU is lower than DRAM channels, resulting in a decreased bandwidth-to-capacity ratio

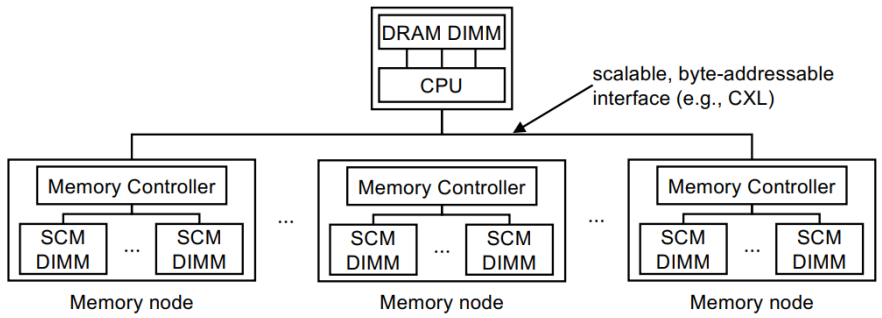


Fig. 2. SCM-based pooled memory architecture

Motivation 2

- Limitations of existing accelerators in SCM environments
  - Designed assuming high-bandwidth DRAM-based memory systems
  - Not optimized for SCM's lower bandwidth and longer latency
  - For example, in the case of IIU
    - Binary search-based intersection algorithm causes frequent random accesses in SCM
    - Union algorithm retrieves more data from memory than necessary
    - Limited support for specific compression schemes

Motivation 3

- Need for improving bandwidth efficiency

1. Inefficient use of SCM's limited bandwidth can lead to performance bottlenecks

2. Potential performance degradation due to shared interconnect bandwidth constraints when scaling memory pools

3. Need to optimize bandwidth usage by reducing unnecessary data movement and computation

4. Necessity to minimize data movement between host CPU and accelerator through Near-Data Processing (NDP) paradigm

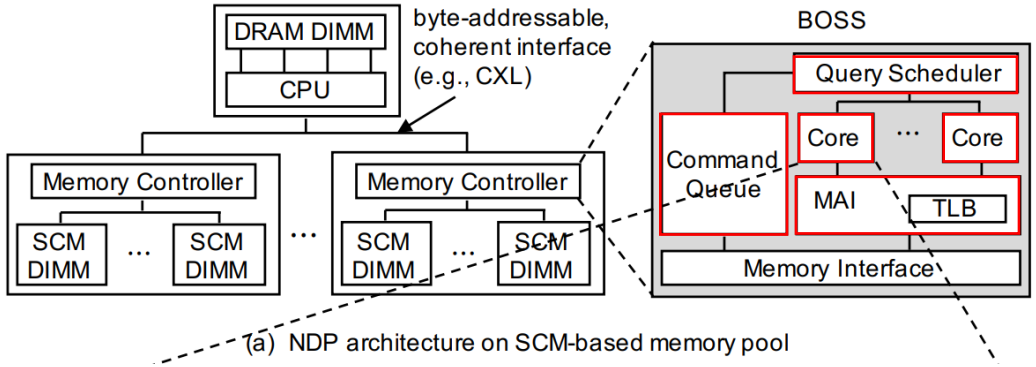
5. Need for internal SCM bandwidth savings through early termination techniques, multi-term query optimization, etc.

BOSS Strategies

- Strategies for bandwidth savings in BOSS
  - Near-data Processing
    - Perform search operations near SCM DIMMs to minimize data movement with the host CPU
  - Top-k Selection
    - Transfer only top k results to CPU instead of entire result set, reducing bandwidth usage
  - Early Termination
    - Skip unnecessary data evaluation at block and document levels
  - Multi Query Processing Optimization
    - Implement pipelined intersection to eliminate need for intermediate data storage
  - Programmable Decompression Module
    - Support various compression schemes to achieve optimal compression ratio

Design - Architecture

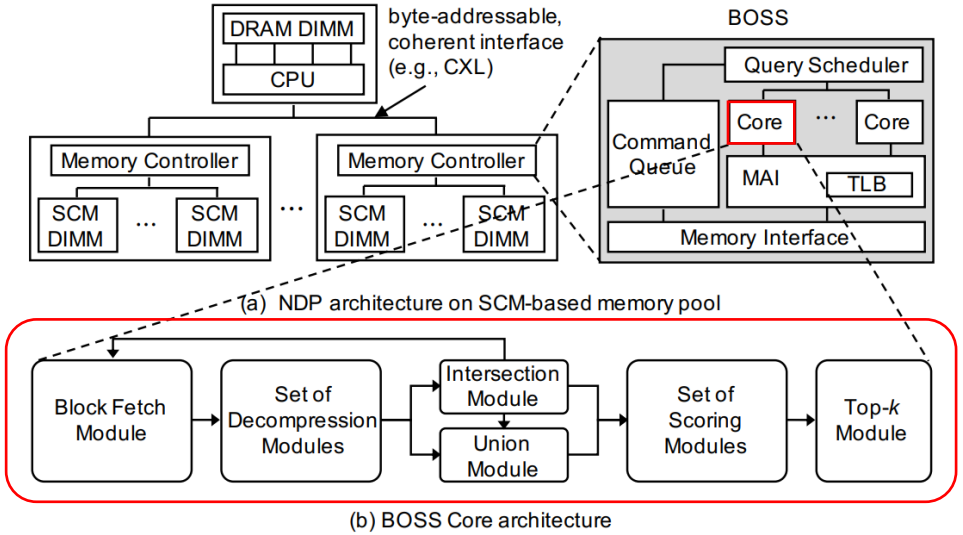
- BOSS Architecture
  - Deployed in SCM memory controllers
  - Composed of multiple BOSS cores and peripherals
  - Query processing managed through command queue and query scheduler
  - SCM memory access handled via Memory Access Interface (MAI)



(a) NDP architecture on SCM-based memory pool

Design - Core

- BOSS Module
  - a) Block fetch module
    - Application of **Early Termination (ET)** algorithm for union queries
  - b) Decompression module
    - Reconfigurable structure supporting **various compression schemes**
  - c) Intersection module
    - **Pipelined** processing for **multi-term queries**
  - d) Union module
    - **Skipping** unnecessary document evaluations
  - e) Scoring module
    - Okapi BM25 metric, **Pre-computation** to reduce runtime calculation
  - f) Top-k module
    - **Priority queue**-based implementation



Evaluation

- Comparison
  - Apache Lucene, IIU (Intel Optane DCPMM, 8 Core)
  - DRAMSim2 Simulator

- Workload
  - CC-News

    (Contains news articles crawled from news sites through CommonCrawl)
  - ClueWeb

    (Crawled by Hetrix web crawler and made public by CMU)

TABLE I
HARDWARE METHODOLOGY

Host Proessor	
Core	Intel Xeon Scalable Processor 8280M @ 2.70GHz
L1 $	32KB I-cache, 32KB D-cache
L2 $	28MB (Private)
L3 $	38.5MB (Shared, Unified)
Hosts Memory System	
Organization	DDR4 2666 ECC REG, 6 channels, 384GB Intel Apache Pass Memory, 6 channels, 1.5TB
Bandwidth	140.76 GB/s (23.46 GB/s per channel) 39.6 GB/s (6.6 GB/s per channel)
BOSS Configuration	
BOSS	8 BOSS Cores @ 1.0GHz
BOSS Core	1 Block fetch module, 4 Decompression modules, 1 Intersection module, 1 Union module, 4 Scoring modules, 1 Top-*k* module
BOSS Memory System	
Organization	SCM, 4 channels
Bandwidth	Read bandwidth: 25.6GB/s (sequential), 6.6GB/s (random) [70] Write bandwidth: 2.3GB/s [70]

TABLE II
QUERY TYPES

Type	Number of Terms	Operation
Q1	1	A
Q2	2	A AND B
Q3	2	A OR B
Q4	4	A AND B AND C AND D
Q5	4	A OR B OR C OR D
Q6	4	A AND (B OR C OR D)

Evaluation

- Multi-core throughput & Bandwidth utilization

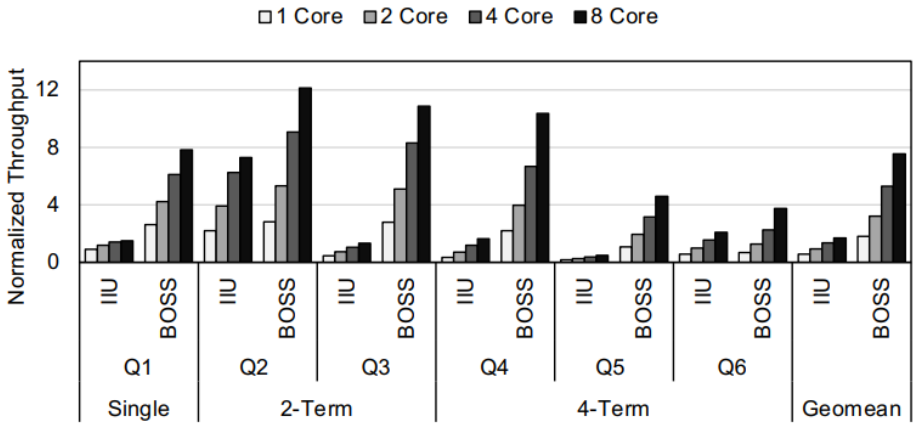


Fig. 9. Multi-core throughput analysis (ClueWeb12) (normalized to Lucene with 8 cores on SCM)

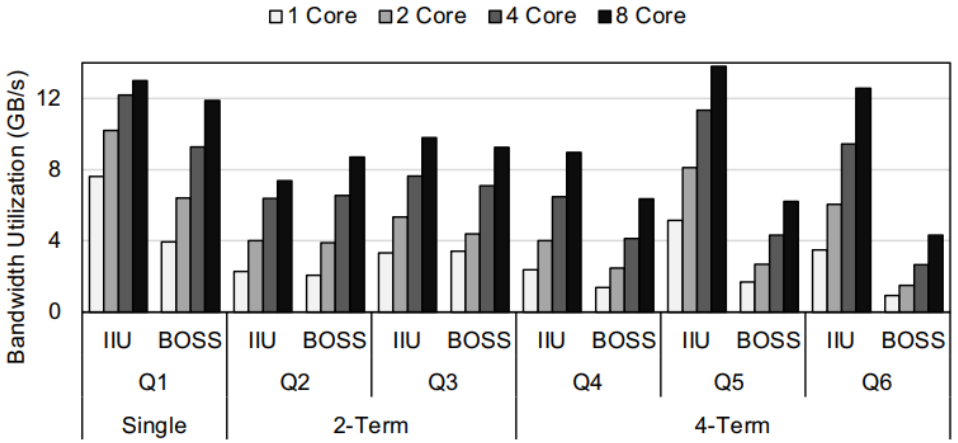


Fig. 11. Bandwidth utilization (ClueWeb12)

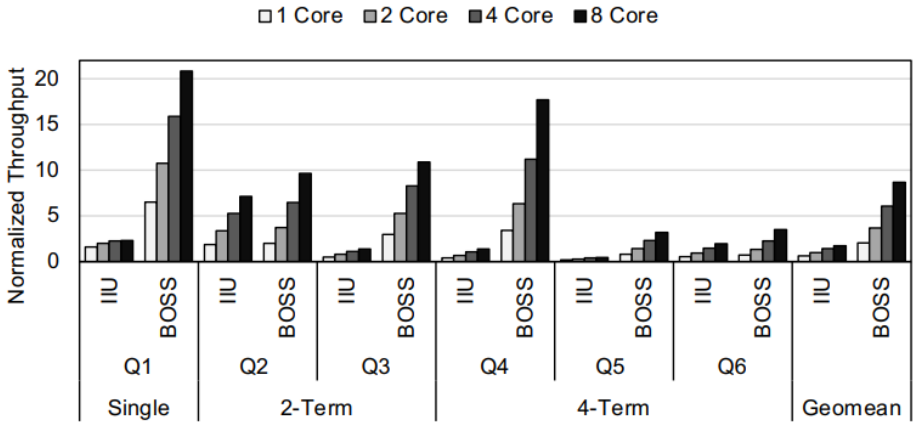


Fig. 10. Multi-core throughput analysis (CC-News) (normalized to Lucene with 8 cores on SCM)

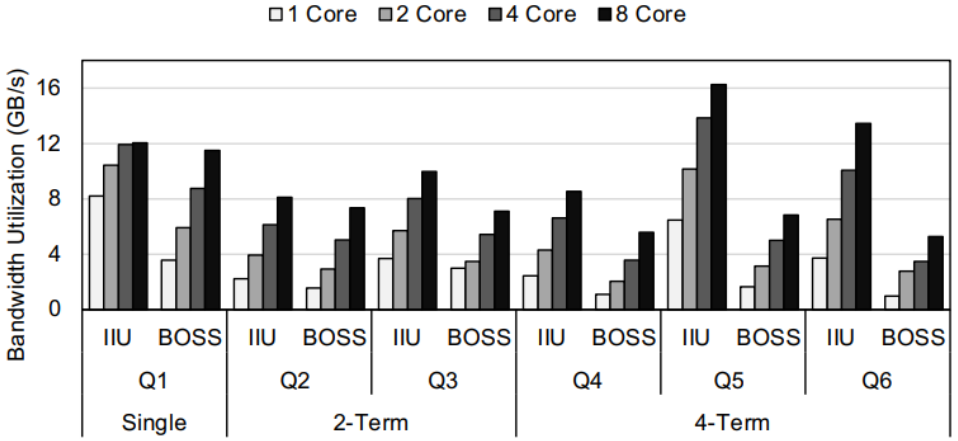


Fig. 12. Bandwidth utilization (CC-News)

Evaluation

- Single-core throughput & Bandwidth utilization

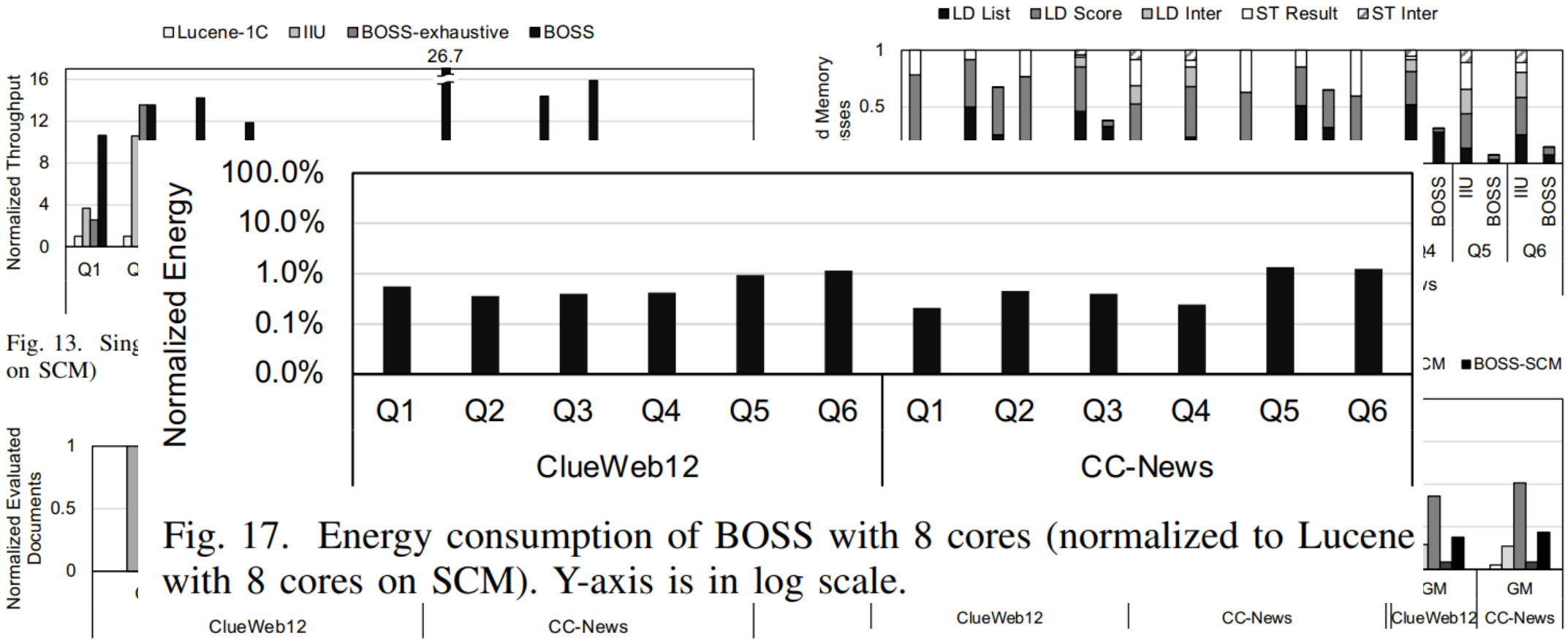


Fig. 13. Sing
on SCM)

Fig. 17. Energy consumption of BOSS with 8 cores (normalized to Lucene with 8 cores on SCM). Y-axis is in log scale.

Fig. 14. Normalized number of evaluated documents

Fig. 16. Performance comparison between Lucene, IIU, and BOSS on DRAM and SCM (normalized to Lucene with 8 cores on SCM)

Conclusion

- **BOSS** presents the first **NDP (Near-Data Processing) accelerator architecture** for inverted index search targeting the emerging **SCM-based** memory pool with **bandwidth constraints**

- BOSS applies synergistic strategies to overcome the **bandwidth challenges of SCM-based pooled memory**, including skip mechanisms, multi-term query optimization, a top-k selection hardware module, and a near-data processing paradigm

- Evaluation results show that BOSS achieves an **average 8.1x speedup** on various complex query types while reducing average **energy consumption by 189x** compared to Apache Lucene running on 8 CPU cores, demonstrating that BOSS is an **efficient alternative for inverted index search**

Thank you

Suhwan Shin

sshshin@dankook.ac.kr