Computational Memory Solution

White Paper

Computational Memory Solution: Smarter Memory to Enhance Data Analytics System

Memory System Architecture, Memory System Research

Table of Contents

Introduction

As the scale of data reaches the order of zettabytes and the demand for instant business insight surges, companies invest aggressively in enhancing their data analytics systems to process a massive volume of data at low latency. Such data-intensive workloads put substantial pressure on the underlying analytics systems, requiring unprecedentedly high-bandwidth and high-capacity memory.

However, conventional data infrastructures based on processor-centric architectures are inefficient to address such demands. Specifically, processor-centric architectures lack memory scalability and cannot provide sufficient memory bandwidth and capacity required for higher performance. Such limited memory scalability incurs significant costs to build a high-performance cluster because more servers with high-end CPUs are needed to provide the required bandwidth and capacity. Furthermore, current data warehouse and analytics systems consume substantial energy to transfer enormous data between memory and processors. Such significant data traffic has become a performance bottleneck in today's data infrastructures.

Computational Memory Solution (CMS) has been designed to overcome all the above drawbacks related to conventional data infrastructures and to address the demands for high-bandwidth and high-capacity memory. In contrast to the traditional processor-centric approach, CMS is a data-centric computing solution based on Near-data Processing that leverages exceptionally high internal bandwidth and capacity to accelerate memory-intensive workloads. CMS also saves power by reducing data traffic between memory and processors.

Ultimately, CMS 1) delivers high-performance for various memory-intensive workloads, 2) provides cost-effective scalability in system performance and memory capacity, and 3) reduces the total cost of ownership (TCO) for modern data analytics systems.

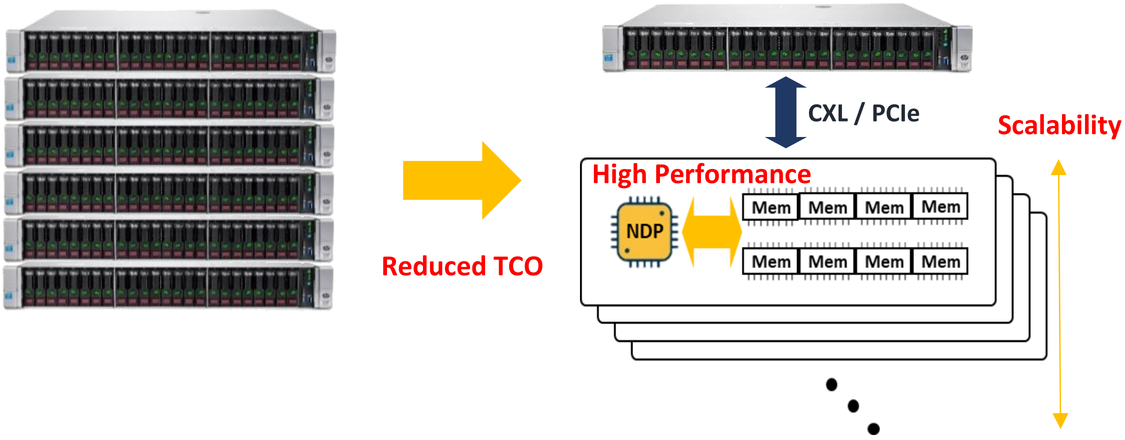

Figure 1 CMS delivers high performance, scalability, and reduced TCO

Part1: Near Data Processing Opportunities in Data Analytics

Near-data Processing (NDP) refers to processing data in proximity to the memory where data is stored. Due to its proximity to memory, NDP can efficiently accelerate data-intensive workloads by leveraging extremely high bandwidth to memory. In particular, NDP is most effective for workloads requiring relatively simple yet parallelizable computation that processes massive data.

Why Data Analytics?

Many existing data analysis applications are data-intensive, requiring high memory bandwidth and capacity. However, current compute-centric systems running these applications are inefficient in handling such workloads because their architectures are optimized for compute-intensive workloads, whose characteristics are fundamentally different from memory-intensive ones. The mismatch between characteristics of memory-intensive data analytics applications and attributes of compute-centric systems causes considerable inefficiencies and costs in building high-performance data analytics systems.

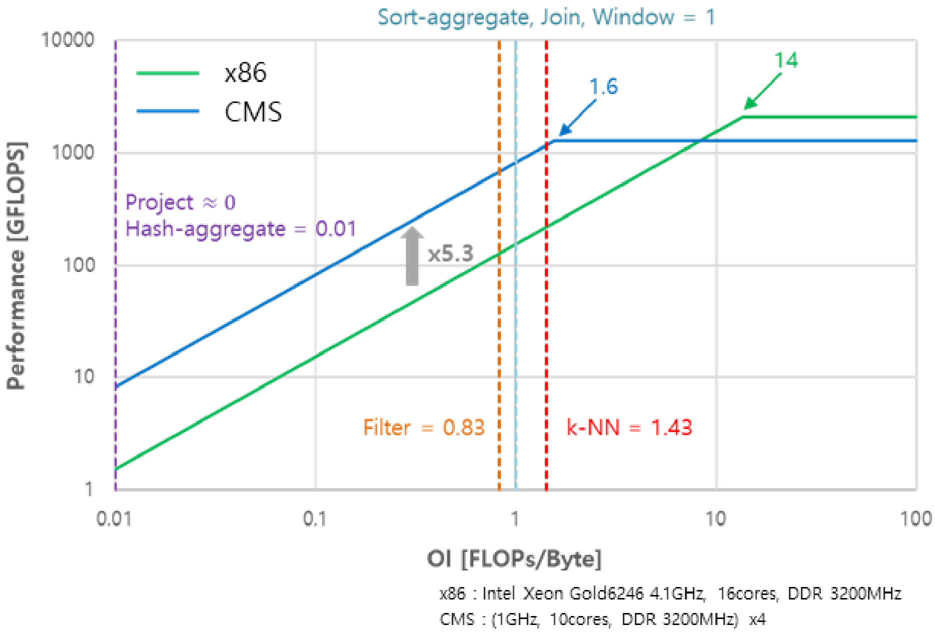

Figure 2 Roofline analysis for representative data analysis operations: project, filter, aggregate, join, window, and k-NN. The X-axis is Operational Intensity (OI), and the y-axis is the anticipated performance in GFLOPs. The figure indicates that CMS offers 5.3 times higher aggregate memory bandwidth than an Intel Xeon Gold6246 server, and therefore, can deliver up to 5.3 times performance improvement for all the operations.

Typical data analysis queries involve simple yet highly parallelizable operations dealing with large datasets, such as project, filter, aggregate, or join. These operations feature low

Operational Intensity (OI), an ideal characteristic for Near-data Processing. Figure 2 presents the result of the Roofline analysis[1] for representative SQL operations (project, filter, aggregate, join and window) and a machine learning function (k-NN). All of them, including many other SQL operations, are memory-bound with OIs significantly lower than the ridge point of modern CPUs. Therefore, providing high memory bandwidth is crucial to enhance their performance.

Indeed, our experiments with SQL queries provided evidence that numerous SQL queries are memory-bound when filtering, aggregating, or joining large datasets. As presented in Figure 3, we measured the memory bandwidth utilization for TPC-DS queries running on our data analytics cluster. We observed that the system's memory bandwidth utilization was often at the maximum when CPUs were busy processing queries.

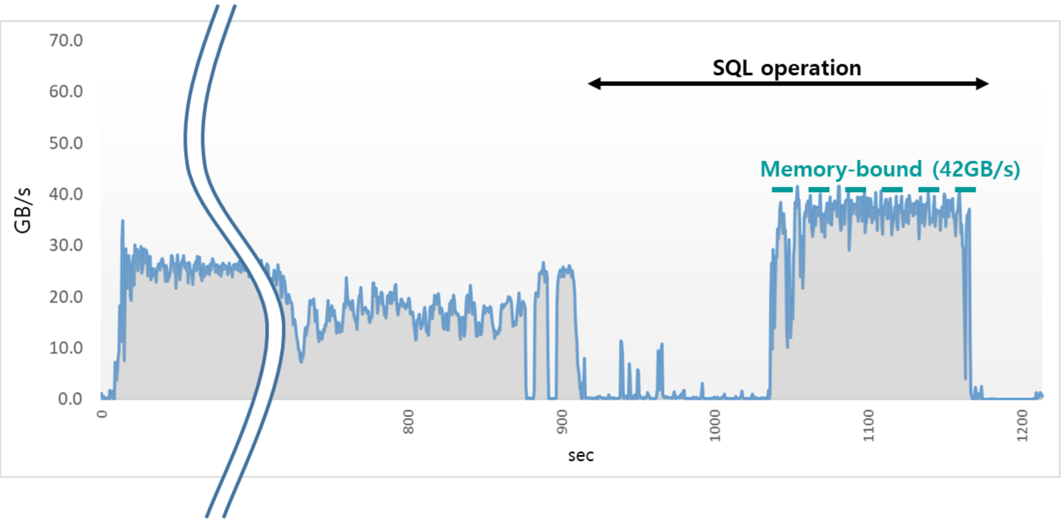

Figure 3 Memory bandwidth hits the maximum when running SQL queries. In this experiment, we ran TPC-DS Q26 on an Intel Silver 4114 server with peak memory bandwidth of 76.8GB/s, whose sustainable memory bandwidth measured by Stream Benchmark is 39 ~ 45 GB/s. Only 50% of the total CPU cores were used to run the query.

Furthermore, we experimented with a k-NN algorithm to see how much we could accelerate the algorithm using compute resources available for an x86 server. As shown in Figure 4, k-NN performance immediately flattened as the number of threads increases. Due to the low OI of k-NN, its performance is bounded by the memory bandwidth once the number of threads has reached 7, which corresponds to mere 25% utilization of the total compute resources.

Unfortunately, increasing memory bandwidth usually involves upgrading to high-end servers or introducing more servers to the analytics cluster, incurring high costs. In fact, low-OIs and high-bandwidth requirements of typical data analysis workloads suggest the need for a new solution that scales memory bandwidth more efficiently.

[1] Williams, S., Waterman, A.; Patterson, D. (2009). Roofline: An insightful visual performance model for floating-point programs and Multicore Architectures.

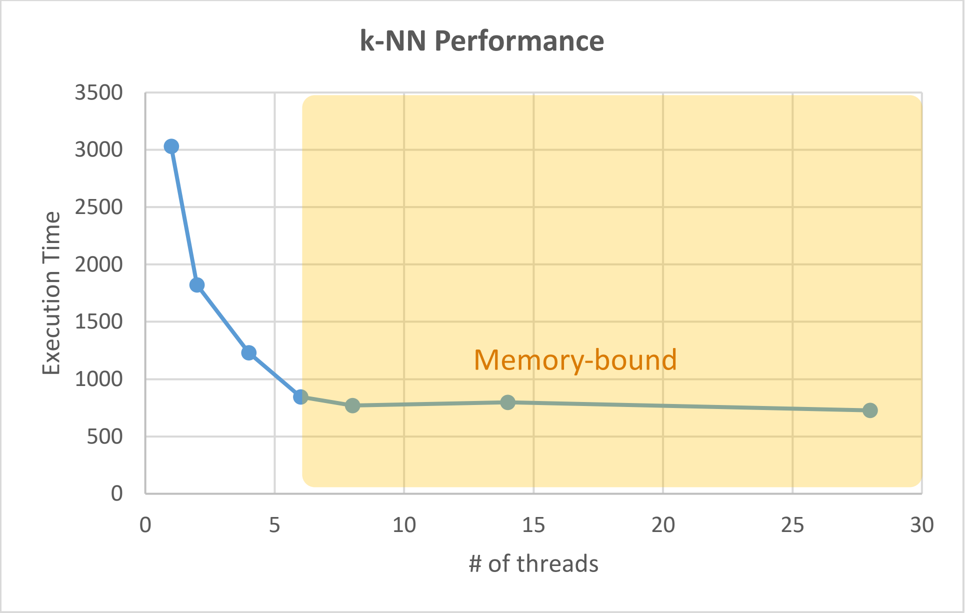

Figure 4 k-NN is memory-bound, and using more than 25% of the total compute resources does not help improve its performance. This experiment involves processing a batch of 128 k-NN queries, each of which calculates and compares distances to 100,000 256-dimension samples and selects the k nearest ones. Xeon E5-2690 v4 is used for this experiment.

Given the above results, a plausible solution is to introduce CMS to the analytics system. In the following sections, we introduce CMS and analyze its benefits for high-performance data analytics system.

Part2: Computational Memory Solution

Computational Memory Solution (CMS) offers a scalable card-type memory composed of an NDP core and large capacity memory, as shown in Figure 5. CMS provides high-performance for memory-intensive workloads by Near-data Processing that leverages ultra-high internal memory bandwidth. It also offers cost-effective scalability in performance and capacity. Specifically, CMS allows customers to scale their systems by simply inserting additional CMS cards into their servers via PCIe or Compute Express Link (CXL). Since customers can augment their analytics cluster using additional CMS cards with fewer servers, yet experience equivalent performance, they can significantly reduce the total ownership cost. In particular, we expect our CMS solution integrated via Compute Express Link (CXL) will deliver unprecedented performance and memory scalability for large-scale data analytics systems.

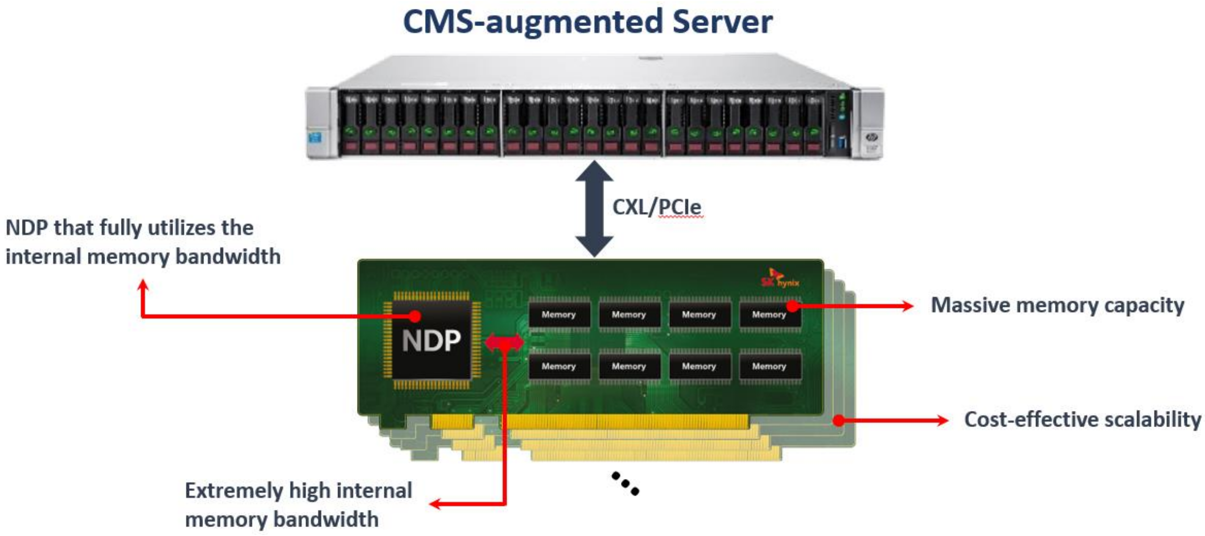
CMS-augmented Server
CXL/PCIe
NDP that fully utilizes the internal memory bandwidth
NDP
Memory
SK hynix
Massive memory capacity
Cost-effective scalability
Extremely high internal memory bandwidth

Figure 5 Computational Memory Solution

Reference System for Experiment and Analysis

Although CMS can enhance any system that runs memory-intensive workloads, this paper examines the benefits of CMS for a real-time data analytics system comprising an in-memory database and Apache Spark. As presented in Figure 6, the reference system used for our experiment consists of LightningDB[2] as a storage engine and Apache Spark as a compute engine.

[2] Lightning DB - DRAM/SSD optimized Real-time In-memory DBMS (https://lightningdb.io/)

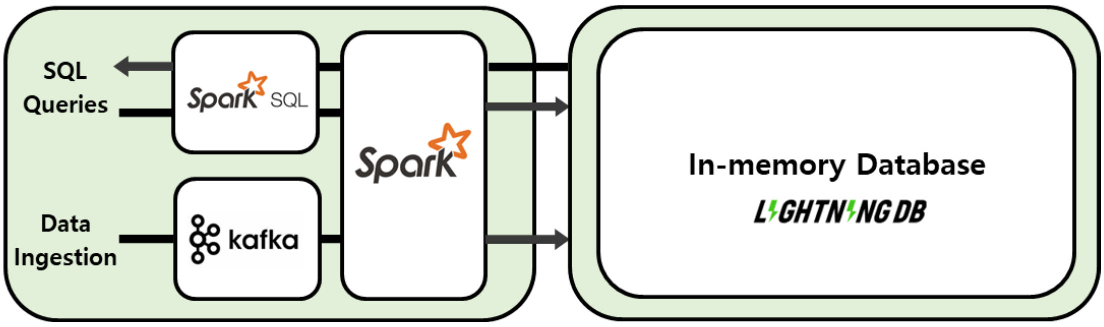

Figure 6 Reference real-time analytics system

We uploaded the input datasets into the reference in-memory database (LightningDB) before querying the data for all our experiments. In later sections, we use the measurements collected from this reference system to project the performance improvement CMS can deliver to in-memory databases and Apache Spark, respectively.

CMS-augmented In-memory Database

In-memory databases offer minimal response time by fetching data directly from memory. We chose LightningDB as our reference in-memory database because it also supports pushdown filtering and aggregation to reduce the data traffic to the upper compute engine. As Figure 7 presents, CMS augments the reference in-memory database cluster with CMS cards that can accommodate much more data and accelerate pushdown filtering and aggregation.

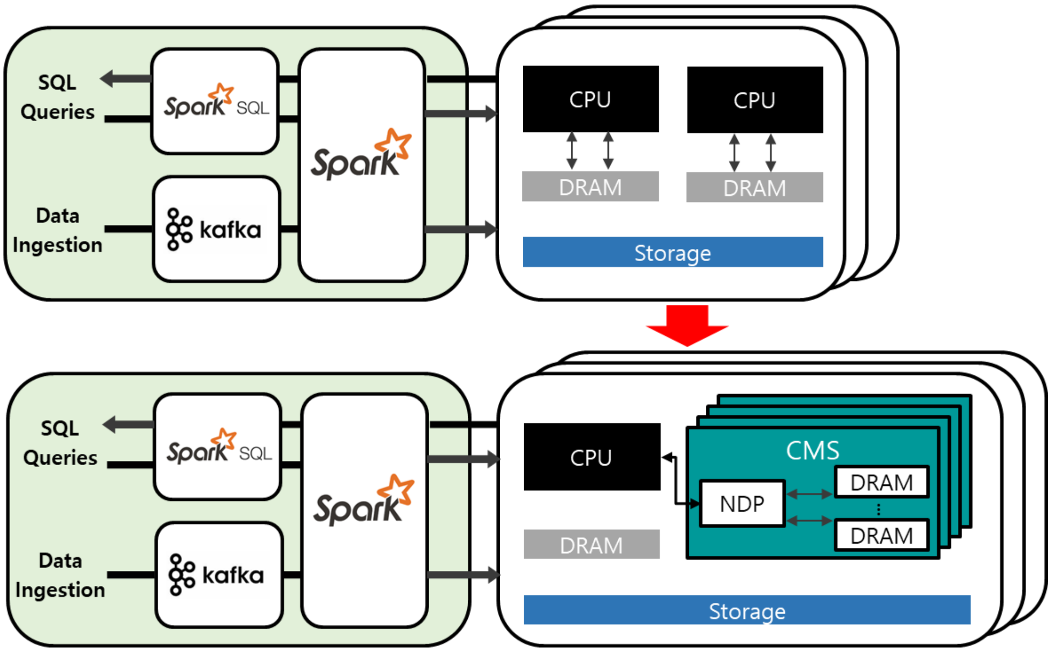

Figure 7 Reference in-memory database integrated with CMS cards

CMS-augmented Apache Spark System

Apache Spark is a distributed computing engine that processes big data sets using the MapReduce programming model. Similar to the case for in-memory databases, CMS augments the Apache Spark cluster with CMS cards that can store significantly larger datasets and accelerate SQL queries. Enlarged memory capacity also enables keeping frequently accessed datasets in memory through Spark's caching feature, further enhancing the query response time.

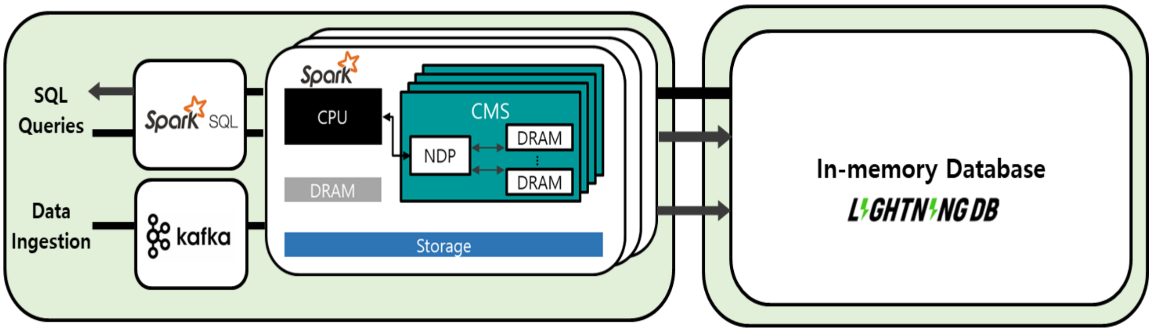

Figure 8 Reference Apache Spark system integrated with CMS cards

Part3: Use Case Study 1: In-memory Database

Amid rapidly increasing demands for real-time analytics, many applications employ in-memory databases to attain minimal response time by fetching data directly from memory rather than from disks or SSDs. On top of their fast response time, many latest in-memory databases feature pushdown filtering or aggregation as performance optimization by allowing the data analytics compute engine, such as Apache Spark, to push filtering or aggregation operations down to the underlying in-memory database. Such pushdown optimization coupled with short response time enhances the entire data analytics system by significantly reducing the amount of data transferred to the compute engine at extremely low latency.

Albeit their significant advantages for data analytics at scale, in-memory databases have to overcome several limitations to be adopted more widely. First of all, in-memory databases require substantial memory capacity to prevent disk-spill from degrading the response time. Since memory is much more expensive than disks or SSDs, storage-based databases are often preferred over in-memory databases when accommodating a large volume of data. Furthermore, handling pushdown filtering or aggregation operations requires non-negligible compute resources already occupied with managing other database requests.

CMS addresses these problems by offering scalability both in capacity and performance. According to our analysis, as more CMS cards are integrated into our reference in-memory database, the predicate pushdown performance of the system increases almost linearly while the memory capacity scales in proportion to the number of cards. For instance, an in-memory database server integrated with four CMS cards offers four times larger memory capacity and a similar increase in the predicate pushdown performance for the NYC Taxi Benchmark queries without requiring additional CPU resources. This result suggests that CMS could enable customers to build a high-performance in-memory database cluster with significantly fewer servers, thereby substantially reducing the total ownership cost.

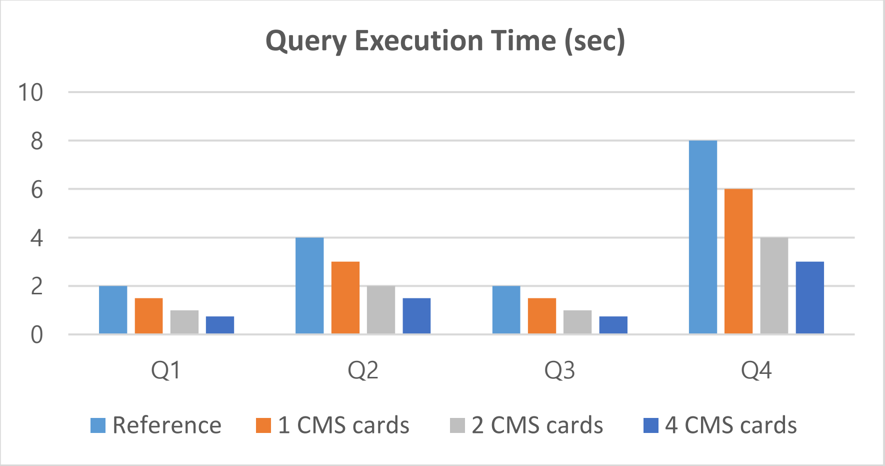

Figure 9 NYC Taxi Benchmark query execution time. Reference in-memory database comprises four Intel Xeon Gold 6140 servers with 170GB/s memory bandwidth for each. The original NYC Taxi dataset has been scaled down to fit in the cluster. CMS has been applied only to the in-memory database for

this analysis. The filter and aggregation pushed down to the in-memory database are accelerated by NDP cores, which filter and aggregate the data stored in their nearby memory.

Part4: Use Case Study 2: Apache Spark System

Apache Spark has become the most prevalent compute engine for large-scale data processing. The vast majority of data analytics companies are using Apache Spark in production to process an enormous number of data analysis queries to extract business insights. As data volume increases exponentially, the number of servers to form a Spark cluster also grows, resulting in significant overheads associated with managing numerous Spark tasks, such as task scheduling and shuffle IOs. In addition, the common practices of using small partition sizes and a large number of Spark tasks to maximize the CPU utilization are amplifying these overheads, making linear performance scaling of analytics systems challenging to achieve.

CMS enables customers to build a cluster with considerably fewer and more affordable servers that run potentially smaller numbers of tasks. Specifically, near-data processors of CMS accelerate memory-intensive SQL operations by leveraging its ultra-high internal memory bandwidth. In addition, a high aggregate memory capacity of multiple CMS cards allows using larger partition sizes, thereby fewer Spark tasks. This helps reduce task scheduling and shuffle IO overheads. Overall, acceleration of SQL operations via Near-data Processing combined with reduced scheduling and shuffle IO overheads leads to greater performance at a reduced total cost of ownership.

According to our analysis, TPC-DS query performance improves as we integrate more CMS cards into Spark servers of our reference data analytics system and configure Spark tasks to process larger data partitions. For instance, Q26 performance improves by 2.3x with a CMS card and 5.4x with 4 CMS cards. For the select TPC-DS queries presented in Figure 10, our analysis predicts CMS could improve their response time up to 7x.

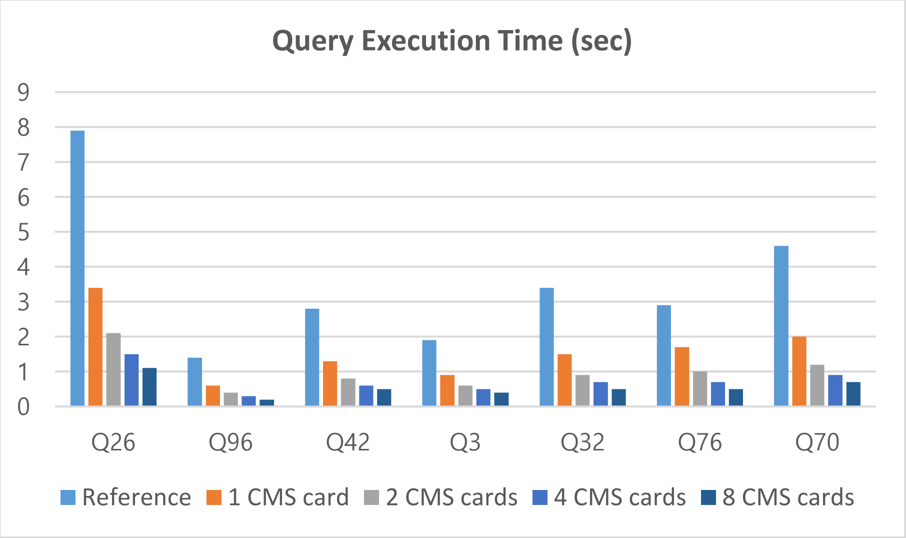

Figure 10 TPC-DS benchmark query execution time. TPC-DS dataset size has been chosen to fit in our reference in-memory database cluster. CMS has been applied only to the Spark cluster for this analysis. The in-memory database only provides the input data to the Spark cluster without pushdown filter or aggregate enabled. The Spark cluster comprises an Intel Silver 4114 server with 76.8GB/s memory bandwidth.

We also conducted an analysis that compares the performance of a multi-node CPU-only cluster with that of a single-node cluster augmented with CMS cards. Based on our analysis, a single-node cluster equipped with 2 CMS cards provides comparable performance to the four-node CPU-only cluster for the select queries we analyzed. This result implies that CMS could reduce the total cost of ownership for the Apache Spark cluster by decreasing the required number of servers to achieve target performance.

All the above analyses assume that input datasets have been loaded into the in-memory database of our reference data analytics system. In addition, the analysis is based on scaled-up partition size per task as more CMS cards are integrated into the Apache Spark cluster to reduce task scheduling and shuffle IO overheads.

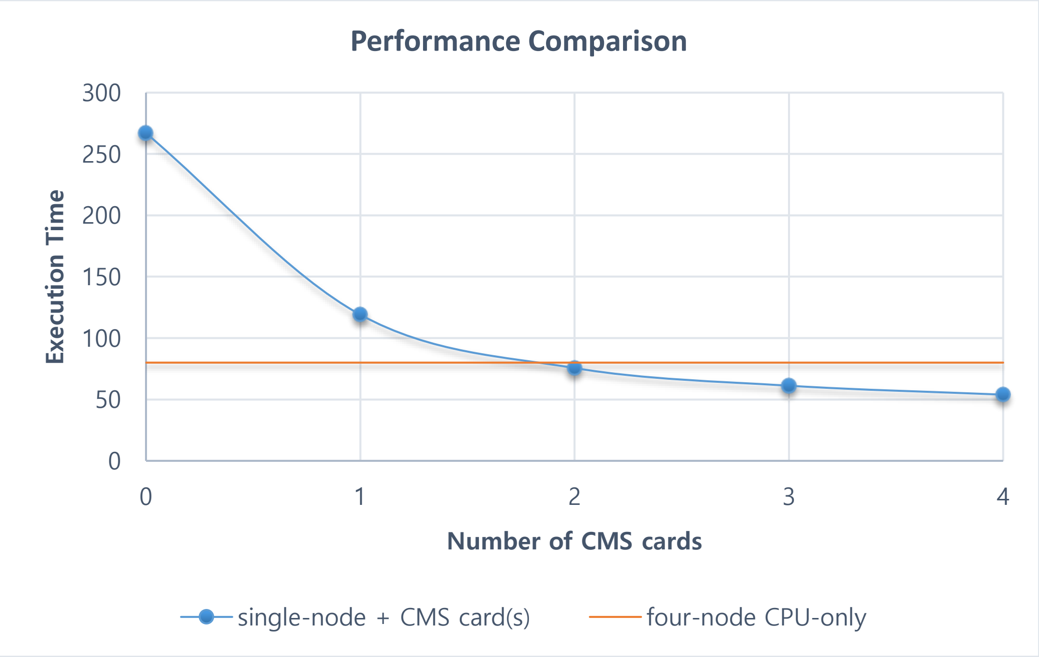

Figure 11 CMS reduces ownership costs by requiring fewer servers to form a Spark cluster. A single-node Spark cluster comprising an Intel Silver 4114 server (76.8GB/s memory BW) and two CMS cards shows comparable performance to a multi-node CPU-only cluster consisting of four Intel Xeon Gold 6140 servers (170GB/s memory bandwidth for each). The execution times for TPC-DS Q26 are used for this analysis.

Part5: Future Works

By using analytical performance modeling and measured results from our reference system, our analyses have demonstrated that CMS could be a cost-effective, high-performance solution that can enhance various modern big data applications. CMS is a technological breakthrough that could improve the performance of memory-intensive workloads, achieve cost-effective scalability in performance and memory capacity, and reduce ownership costs for data analytics systems.

Currently, a prototype for Computational Memory Solution targeting data analytics is being developed, and the solution is expected to be available to the market shortly.

New Initiatives (Memory Forest)

As the demand for big data and AI increases, data is growing explosively in both volume and variety. Such has led to the emergence of data-centric workloads that manipulate and analyze massive amounts of data. Consequently, the burden of data processing and energy consumption from data movement is becoming a critical issue for this rapidly growing segment. For the latest AI models (e.g. Google AI switch, Open AI GPT3) that require large-scale parameters, the energy cost of data movement is substantially higher than that of computation. In some popular Google applications, 62.7% of the total system energy is spent on data movement between CPU and main memory. This energy consumption due to data movement is expected to widen as the era of AI-based big data processing accelerates, rendering it essential to reduce data movement in order to improve performance and energy efficiency in data-centric computing systems.

The shift from a compute-centric to a data-driven era is an opportunity for SK hynix to take on a central role in the new ICT (Information & Communications Technology) industry. Having defined a more granular hierarchy for memory in each data processing stage, we are working to make servers and other systems more efficient with targeted solutions such as High-Bandwidth Memory (HBM), the multiprocessor-compatible Compute Express Link (CXL) interface, Processor-in-Memory (PIM) and Computational Memory Solution (CMS). Memory Forest shown in Figure 1 is our new initiative and slogan that encapsulates our strategy to build a memory-driven ecosystem with such technical expertise. Just like the lush, green forest it represents, the initiative will generate value from new memory systems and technologies to nurture a wider global ecosystem that produces ESG values for our customers and partners – essentially with Memory for the Environment (E), Society (S), and Tomorrow (T). This paper describes the CMS, one of SK hynix Memory Forest initiatives.

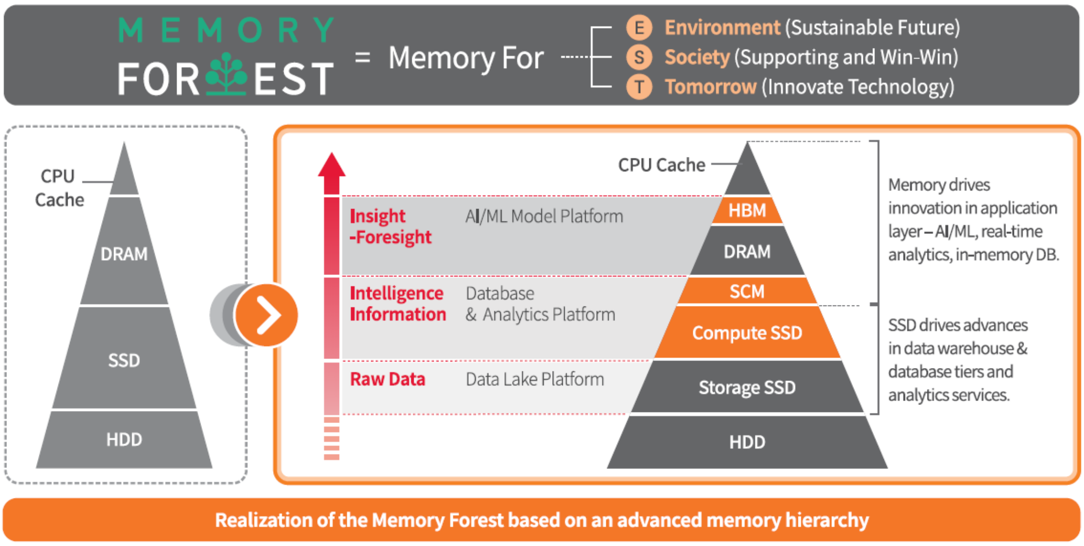
MEMORY FOREST = Memory For
E Environment (Sustainable Future)
S Society (Supporting and Win-Win)
T Tomorrow (Innovate Technology)
CPU Cache
DRAM
SSD
HDD
Insight -Foresight
AI/ML Model Platform
Intelligence Information
Database & Analytics Platform
Raw Data
Data Lake Platform
CPU Cache
HBM
DRAM
SCM
Compute SSD
Storage SSD
HDD
Memory drives innovation in application layer – AI/ML, real-time analytics, in-memory DB.
SSD drives advances in data warehouse & database tiers and analytics services.
Realization of the Memory Forest based on an advanced memory hierarchy

Figure 12 SK hynix Memory Forest Initiatives

Many researchers consider a departure from traditional CPU-centric computing systems, aka Von Neumann architecture, which involves complete separation of the computing and memory units. The work in adds extra computing units close to the memory to process the data locally. Processing in memory (PIM) is one of the solution that addresses the data movement issue by processing certain tasks inside memory blocks, resulting in improvements for both performance and energy efficiency. However, for some data-intensive workloads, a solution that can reduce inter-node communication by providing sufficient memory capacity and bandwidth to the processing unit is more suitable. In this research, we studied the architecture and use cases for the solution and implemented an FPGA PoC.

Legal disclaimer

The information contained in this document is claimed as property of SK hynix. It is provided with the understanding that SK hynix assumes no liability, and the contents are provided under strict confidentiality. This document is for general guidance on matters of interest only. Accordingly, the information herein should not be used as a substitute for consultation or any other professional advice and services. SK hynix may have copyrights and intellectual property right. The furnishing of document and information disclosure should be strictly prohibited. SK hynix has right to make changes to dates, product descriptions, figures, and plans referenced in this document at any time. Therefore the information herein is subject to change without notice.

About SK hynix Inc.

SK hynix seeks to propel the semiconductor industry forward with global tech leadership, and provide a future of greater value to stakeholders to create a better world with information and communication technology. As the world's third largest chipmaker with know-how and customer trust built over more than 38 years, SK hynix continues delivering on a comprehensive range of memory semiconductor solutions from DRAM and NAND Flash to CMOS image sensors.

The company's advanced memory technologies are driving critical innovations of the Fourth Industrial Revolution such as Big data, AI, Machine Learning, IoT, and Robotics. Moreover, SK hynix is aiming higher with the new "Memory Forest" initiative to quickly respond to future changes in the ICT ecosystem. With robust ESG management that accounts for value to the environment, societies, and future generations, SK hynix will continue to build competence and success around the globe.