**CWID:** 50321021

**Name:** Rama Krishna Kamma

**Assignment:** Assignment 1

**Date:** 01-31-2023

**Instructor:** Dr. Pooja Rani

**Subject Details:** CSCI 573.01B Big Data Computing and Analytics

**Number of Pages:** 17

**Research Interest:** Map Reduce, Hadoop, Hive, Spark, Data Analytics.

**Main Paper Title:**

MapReduce: Simplified Data Processing on Large Cluster.

**Abstract:**

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown  
in the paper. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program’s execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any  
experience with parallel and distributed systems to easily utilize the resources of a large distributed system. Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use hundreds of MapReduce programs have been implemented and upwards of one thousand MapReduce jobs are executed on Google’s clusters every day.

**Conclusion:**

The MapReduce programming model has been successfully used at Google for many different purposes. We attribute this success to several reasons. First, the model is easy to use, even for programmers without experience with parallel and distributed systems, since it hides the details of parallelization, fault-tolerance, locality optimization, and load balancing. Second, a large variety of problems are easily expressible as MapReduce computations. For example, MapReduce is used for the generation of data for Google’s production web search service, for sorting, for data mining, for machine learning and many other systems. Third, we have developed an implementation of MapReduce that scales to large clusters of machines comprising thousands of machines. The implementation makes efficient use of these machine resources and therefore is suitable for use on many of the  
large computational problems encountered at Google. We have learned several things from this work. First, restricting the programming model makes it easy to parallelize and distribute computations and to make such computations fault tolerant. Second, network bandwidth is a scarce resource. A number of optimizations in our system are therefore targeted at reducing the amount of data sent across the network: the locality optimization allows us to read data from local disks and writing a single copy of the intermediate data to local disk saves network bandwidth. Third, redundant execution can be used to reduce the impact of slow machines, and to handle machine failures and data loss.

1. **Abstract:**

The distributed system architecture used by contemporary user-facing applications deployed in datacenters exacerbates the latency needs of their constituent microservices. Because these apps require a long transition latency (order of 100 s) to wake up from a deep CPU idle state, existing CPU power-saving approaches reduce their performance. In order to avoid the system entering deep package C-states while all CPU cores are idle, server manufacturers advise only enabling shallow core C-states for idle CPU cores.

Georgia Antoniou, Haris Volos, Davide B. Bartolini, Tom Rollet, Yiannakis Sazeides, Jawad Haj Yahya, "AgilePkgC: An Agile System Idle State Architecture for Energy Proportional Datacenter Servers", 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), pp.851-867, 2022.

1. **Abstract:**

Modern datacenters host user-facing apps that use a variety of services with strict latency requirements (30-250 s) and exhibit unpredictable request patterns. Due to the lengthy transition time from a deep CPU core idle power state as a result of these characteristics, present energy-conserving approaches are ineffective when processors are idle. While other studies suggested management strategies to reduce this inefficiency, we attack it head-on with AgileWatts (AW), a novel deep CPU core C-state architecture designed for datacenter server processors aimed at latency-sensitive workloads. On the basis of three fundamental concepts, AW significantly lowers the transition latency from deep CPU core idle power states while maintaining the majority of their power savings.

Jawad Haj Yahya, Haris Volos, Davide B. Bartolini, Georgia Antoniou, Jeremie S. Kim, Zhe Wang, Kleovoulos Kalaitzidis, Tom Rollet, Zhirui Chen, Ye Geng, Onur Mutlu, Yiannakis Sazeides, "AgileWatts: An Energy-Efficient CPU Core Idle-State Architecture for Latency-Sensitive Server Applications", 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), pp.835-850, 2022.

1. **Abstract:**

Numerous distributed computing applications in data centers are built on the Fork-Join framework. In this study, we create a method called ForkMV to calculate the mean and variance of request response times for Fork-Join queuing networks (FJQNs) with any request fanout degrees and both short- and long-tailed service time distributions (i.e., the number of Fork nodes). ForkMV is specifically able to estimate the mean and variance of request response time for a FJQN with any particular service time distribution of practical interests, accurately enough to provide efficient resource allocation for data center applications.

Prathyusha Enganti, Todd Rosenkrantz, Lin Sun, Zhijun Wang, Hao Che, Hong Jiang, "ForkMV: Mean-and-Variance Estimation of Fork-Join Queuing Networks for Datacenter Applications\*", 2022 IEEE International Conference on Networking, Architecture and Storage (NAS), pp.1-8, 2022.

1. **Abstract:**

The widespread use of distributed architecture creates new difficulties for maintenance and operation. The workload for monitoring increased significantly as the number of system nodes and microservices grew rapidly. Human maintenance is incompetent due to the enormously complicated relationships between the monitoring items. Data fragmentation and distant storage make it challenging to maintain the conventional maintenance mode. These issues with conventional operation and maintenance exist: 1) The operation and maintenance are dispersed as a result of the group/provincial two-level maintenance structure. This makes it impossible to properly manage the entire network's business support, and the mechanism for scheduling network problems and faults as a whole is not efficient. 2) The entire network monitoring system was developed for many businesses, with distributed monitoring data and retroactive monitoring techniques, which causes a challenge in positioning issue for various businesses. 3) Traditional maintenance ignores the full client experience in favor of a single system and one business. 4) Processing cross-domain or cross-layer problems or faults on a single system takes a long time and is slow, making it impossible to locate faults precisely and recover from them quickly. This study suggests a pinpoint-based end-to-end intelligent monitoring system.

Lanying Shi, Huan Liang, Wensheng Yao, Jingxiang Chen, Chunhua Chen, Yong Chen, Chengwei Yang, Mengxia Chen, Yiquan Jiang, Jiangang Tong, Man Li, Hongming Qiao, "An end-to-end intelligent monitoring system based on pinpoint", 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), pp.676-679, 2022.

1. **Abstract:**

Since of the thousands of servers linked to the network, traditional tree network topologies with their technological infrastructure are unable to develop, expand, and scale because they cannot provide enough bandwidth and consistent latency performance. This incident presents a difficult problem in terms of being able to give its clients better service. The data center network topology is moving in the direction of managing massive amounts of data and preparing for development (scalability). Then the data center network emerged, which had a more significant impact on society, the economy, and daily life. In this method, the network keeps changing and a new network model appears at the appropriate historical time. It's critical to take into account the data center network information for upcoming initiatives in this area.

Antonio Cortés Castillo, "BCube Connected Crossbar and GBC3 Network Architecture: an Overview", 2022 31st Conference of Open Innovations Association (FRUCT), pp.37-44, 2022.

1. **Abstract:**

Search engine query processing can be divided into CPU compute operations and disk I/O operations. The main goal of traditional search engine caching techniques was to minimize disk I/O operations. However, the bottleneck of query processing switches from disk I/O to computation with the emerging trend of employing solid state drives (SSDs) rather than hard disk drives for search engine storage. In this paper, we design a three-level compact caching structure that is appropriate for SSD-based search engines. It includes caches for (a) doculets, which are incomplete documents that contain previously seen snippets, (b) fragments, which are compact data structures that contain snippet metadata, and (c) ranked docID lists, which index the top-k documents pertinent to a query. This three-level compact caching technique intends to lessen the CPU bottleneck caused by routine computing tasks.

Rui Zhang, Pengyu Sun, Jiancong Tong, Ruirui Zang, Heng Qian, Yu Pan, Rebecca J. Stones, Gang Wang, Xiaoguang Liu, Yusen Li, "Three-level Compact Caching for Search Engines Based on Solid State Drives", 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), pp.16-25, 2021.

1. **Abstract:**

In the preceding decade, people were concerned about the dangers of adopting cloud computing. A wholly novel idea, it raised more questions than it answered. Recently, there has been greater talk about the negative effects of not using the cloud. The likes of Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP), among others, have built complex cloud systems that are driving the cloud agenda and supplying unique, creative solutions to meet the needs of contemporary organizations. When it comes to processors, which are the brains of the cloud, hyperscale data centers are more and more turning to specialized chips like GPUs (Graphics Processing Units), FPGAs (Field-Programmable-Gating Arrays), and ASICs. This is also regarded as a step in the artificial intelligence (AI) discovery process.

Rohitha Pasumarty, Raja Praveen, Mahesh T R, "The Future of AI-enabled servers in the cloud- A Survey", 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp.578-583, 2021.

1. **Abstract:**

We all travel everyday over large digital lakes, rivers, and oceans that are filled with social media, e-commerce, streaming video, e-mail, cloud documents, web pages, traffic flows, and network packets. Amorphous data flows backed by continuous streams that defy conventional ideas of type and dimension make up this digital hyperspace. The mathematics of hypergraphs, hypersparse matrices, and associative array algebra may elegantly express, explore, and change the unstructured data of digital hyperspace. In order to provide the fundamental operations for graph analytics, database operations, and machine learning, this study investigates a novel mathematical notion called the semilink, which combines pairs of semirings. The current version of the GraphBLAS standard supports hypergraphs, hypersparse matrices, the math necessary for semilinks, and efficiently executes operations on graphs, networks, and matrices.

Jeremy Kepner, Timothy Davis, Vijay Gadepally, Hayden Jananthan, Lauren Milechin, "Mathematics of Digital Hyperspace", 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp.263-271, 2021.

1. **Abstract:**

Storage workloads (such as key-value stores and databases) often employ a synchronous protocol that places network and server stack latency on the crucial route of request processing in order to assure data permanence. Although the storage cost of the server stack has been reduced because to the introduction of fast and byte-addressable persistent memory (PM), networking is still a major contributor to the end-to-end latency of request processing. By transferring some of an application's compute onto the network (such as caching results for read requests), emerging programmable network devices can minimize network latency. However, for update requests, the client still needs to continually delay on the server to commit the updates. In this paper, we describe PMNet, a programmable data plane (e.g., switch or NIC) with PM for data persistence in the network, and introduce in-network data persistence, which expands the data-persistence domain from servers to the network. Incoming update requests are logged by PMNet, and clients are directly acknowledged rather than needing to wait for the server to commit the request. The logged requests serve as redo logs for the server to recover in the event of a failure. We put PMNet into practice on an FPGA and test its performance with typical PM workloads like key-value stores and PM-backed programs.

Korakit Seemakhupt, Sihang Liu, Yasas Senevirathne, Muhammad Shahbaz, Samira Khan, "PMNet: In-Network Data Persistence", 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), pp.804-817, 2021.

1. **Abstract:**

One of the most used and significant web services is search. The most common data structure used by full-text search engines is the inverted index. Inverted index search has recently seen the emergence of specialized hardware accelerators with substantially greater throughput than the traditional CPU or GPU. Less focus has been placed on using inverted index to relieve the burden on memory capacity. Making a main memory with a terabyte capacity costs substantially more using the traditional DDRx DRAM memory system. A far more affordable option for increasing memory capacity is a shared memory pool made up of storage-class memory (SCM) devices. However, due to the constrained bandwidth of both SCM devices and the shared link to the memory, this SCM-based pooled memory offers new issues. The first near-data processing (NDP) architecture for inverted index search on SCM-based pooled memory that maintains high query processing throughput in this bandwidth-constrained context is BOSS, which we propose. By utilizing early-termination search algorithms, minimizing the footprint of intermediate data, and adding a programmable decompression module that can choose the best compression strategy for a given inverted index, BOSS lessens the effects of the poor bandwidth of SCM devices. In addition, BOSS has a top-k selection module built into the hardware to significantly lower host-accelerator bandwidth consumption. BOSS outperforms production-grade search engine library Apache Lucene, which utilizes 8 CPU cores, by a geomean speedup of 8.1 on a variety of sophisticated query types while consuming 189 percent less energy on average.

Jun Heo, Seung Yul Lee, Sunhong Min, Yeonhong Park, Sung Jun Jung, Tae Jun Ham, Jae W. Lee, "BOSS: Bandwidth-Optimized Search Accelerator for Storage-Class Memory", 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), pp.279-291, 2021.

1. **Abstract:**

In terms of quality (as determined by the word error rate, or WER), endpointer latency, and other factors, end-to-end (E2E) models have been proven to beat state-of-the-art conventional models for streaming speech recognition. The model still has a considerably higher partial latency than a traditional ASR model since it still has a tendency to delay predictions near the conclusion. We consider using a technique called FastEmit  to encourage the E2E model to emit words early in order to solve this problem. Naturally, reducing delay causes a deterioration in quality. To solve this, we investigate substituting Conformer layers , which have demonstrated promising gains for ASR, for the LSTM layers in the encoder of our E2E model. In order to enhance quality, we also consider conducting a 2nd-pass beam search.

Bo Li, Anmol Gulati, Jiahui Yu, Tara N. Sainath, Chung-Cheng Chiu, Arun Narayanan, Shuo-Yiin Chang, Ruoming Pang, Yanzhang He, James Qin, Wei Han, Qiao Liang, Yu Zhang, Trevor Strohman, Yonghui Wu, "A Better and Faster end-to-end Model for Streaming ASR", ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.5634-5638, 2021.

1. **Abstract:**

Luiz André Barroso, "A Brief History of Warehouse-Scale Computing", IEEE Micro, vol.41, no.2, pp.78-83, 2021.

1. **Abstract:**

We develop DCMA, a cutting-edge multicast membership management technique for data center networks, in this research. In contrast to conventional protocols like IGMP/MLD, DCMA makes better and easier use of developing software defined networking (SDN) technology and data center multicast application characteristics to manage multicast members. To reduce the size of the forwarding table in switches, the multicast application master sends the membership to the SDN controller, who then allocates the group addresses in a coordinated manner. In specifically, we build both a batch method and an incremental approach to assign the multicast group addresses by articulating the problem of multicast group address allocation and capturing the interaction between forwarding entries in various switches.

Yang Cheng, Dan Li, Jing Zhu, Hongnan Liu, Kai Chen, Jianping Wu, "Managing Multicast Membership for Software Defined Data Center Network", 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), pp.1-5, 2020.

1. **Abstract:**

For data center networks, we develop DCMA, a unique multicast membership management technique, in this research. In contrast to conventional protocols like IGMP/MLD, DCMA makes better and easier use of developing software defined networking (SDN) technology and data center multicast application characteristics to manage multicast members. To reduce the size of the forwarding table in switches, the multicast application master sends the membership to the SDN controller, who then allocates the group addresses in a coordinated manner. In specifically, we build both a batch method and an incremental approach to assign the multicast group addresses by articulating the problem of multicast group address allocation and capturing the interaction between forwarding entries in various switches. K-Means, an unsupervised learning model, has numerous difficulties, including temporal complexity, choosing the ideal number of clusters (representing the classes), and figuring out the centroid values of the clusters. The goal of all clustering techniques is to minimize the mean square error between population centers within the same class. These techniques are frequently improved by employing centroids as recommendations; however they have K-Means constraints. The majority of the described work uses K-Means clustering to categorize workload classes for the application of clustering cloud log traces. There is, however, no published work that categorizes data center scaling classes. This study proposes a unique approach based on random variable model transformation for analyzing the features of workloads and datacenter setups.

Tariq Daradkeh, Anjali Agarwal, Marzia Zaman, Nishith Goel, "Dynamic K-Means Clustering of Workload and Cloud Resource Configuration for Cloud Elastic Model", IEEE Access, vol.8, pp.219430-219446, 2020.

1. **Abstract:**

Modern servers' overall performance is critically dependent on I/O performance. The data transfer between CPUs, main memory, and devices has become a significant performance barrier due to the development of extremely high-speed I/O devices. Since I/O devices can't directly access processor side caches, the main memory is typically employed as an intermediary buffer between the processor and I/O devices. By allowing I/O devices to use Last Level Cache (LLC) as the intermediary buffer, Data Direct I/O (DDIO) technology seeks to lower the memory bandwidth use. According to the findings of our experiments, DDIO can totally reduce memory bandwidth usage even when running programs that are resource-intensive for networks or storage. However, DDIO is frequently disregarded when modeling the I/O subsystem using architectural simulators, which can lead to incorrect predictions regarding the I/O and memory subsystem of current and future large-scale computer systems. In this essay, we give a thorough introduction to the DDIO technology used in Intel server processors. Then, using the gem5 simulator, we demonstrate our model of the cycle accurate I/O subsystem that can simulate DDIO. We evaluate our model by comparing its output to a real computer system and checking it against the reference model gem5.

Mohammad Alian, Yifan Yuan, Jie Zhang, Ren Wang, Myoungsoo Jung, Nam Sung Kim, "Data Direct I/O Characterization for Future I/O System Exploration", 2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp.160-169, 2020.

1. **Abstract:**

Databases, graph processing, genome research, encryption, and hyper-dimensional computing are just a few of the key application domains where bulk bitwise operations, or bitwise operations on enormous bit vectors, are common. In conventional systems, data transportation between the compute units (such as CPUs and GPUs) and the memory hierarchy is a performance and energy efficiency bottleneck for bulk bitwise operations. Data movement through the entire memory hierarchy can be significantly reduced through in-flash processing, which has a strong potential to speed up bulk bitwise operations, particularly when the processed data does not fit in main memory. We point out two main drawbacks of the most recent in-flash processing method for large-scale bitwise operations. It is not designed to account for the highly error-prone nature of NAND flash memory, which makes it I unreliable and (ii) fall short of fully utilizing the bit-level parallelism of bulk bitwise operations that could be enabled by leveraging the distinct cell-array architecture and operating principles of NAND flash memory. We suggest Flash-Cosmos, a novel in-flash processing technique that considerably improves the performance and energy efficiency of bulk bitwise operations while offering high reliability. Flash-Cosmos stands for "Flash Computation with-O ne-S hot Multi-O perand S ensing." Modern NAND flash chips may readily handle the following two fundamental processes introduced by Flash-Cosmos: I Multi-Wordline Sensing (MWS), which allows for bulk bitwise operations on several operands (tens of operands) with a single sensing operation, and (ii) E nhanced S LC-mode P rogramming (ESP), which enables reliable computation inside NAND flash memory.

Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, Onur Mutlu, "Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory", 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), pp.937-955, 2022.

1. **Abstract:**

Modern applications need more and more data, which increases the cost of computation in traditional processor-centric computing systems. More than 60% of the energy used by current systems can be attributed to the transfer of huge amounts of data over memory channels with limited bandwidth between compute components (such as CPUs and GPUs) and memory devices (such as DRAM). The processing-in-memory (PIM) paradigm reduces (and in some cases eliminates) the requirement to transfer data between memory and the processor in order to reduce these expenses.

Geraldo F. Oliveira, Juan Gómez-Luna, Saugata Ghose, Onur Mutlu, "Methodologies, Workloads, and Tools for Processing-in-Memory: Enabling the Adoption of Data-Centric Architectures", 2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp.261-266, 2022.

1. **Abstract:**

Device-level ECC, which protects against media errors, is supplemented by system checksums and cross-device parity in production storage systems. Due to device firmware faults, computers can identify and recover from data corruption thanks to system-level redundancy (e.g., reading data from the wrong physical location). Software-only solutions of system-level redundancy suffer from direct access to NVM, leaving them with little alternative except to sacrifice data protection for severe performance costs. The updating and verification of system-level redundancy are tasks we suggest offloading to TVARAK, a brand-new hardware controller paired with the last-level cache. TVARAK makes it possible to effectively safeguard data from these memory controller and NVM DIMM firmware issues. TVARAK is effective, as demonstrated by a simulation-based evaluation of seven data-intensive applications.

Rajat Kateja, Nathan Beckmann, Gregory R. Ganger, "TVARAK: Software-Managed Hardware Offload for Redundancy in Direct-Access NVM Storage", 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), pp.624-637, 2020.

1. **Abstract:**

Due to their superior classification accuracy and network simplicity, prototype networks (PNs) are one of the most often used few-shot learning techniques. Test examples are grouped according to how far they are from class prototypes. The latency of transmitting data from memory to compute units is substantially higher than the PN computation time, notwithstanding the application-level advantages of PNs. The performance of PNs is thus constrained by memory bandwidth. This bandwidth bottleneck issue is solved by computing-in-memory by putting a portion of the compute units closer to the memory. In this paper, we offer a framework for computing prototypes and distance metrics inside the memory called CiM-PN. With CiM-PN, the Manhattan distance metric takes the place of the computationally demanding Euclidean distance metric.

Dayane Reis, Ann Franchesca Laguna, Michael Niemier, Xiaobo Sharon Hu, "A Fast and Energy Efficient Computing-in-Memory Architecture for Few-Shot Learning Applications", 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.127-132, 2020.

1. **Abstract:**

For high matching throughput, this article suggests REACT, a regular expression matching accelerator that may be integrated into a contemporary Solid-State Drive (SSD), together with a revolutionary data access scheduling technique. Particularly, REACT boosts the use of SSD and the level of internal memory parallelism for pattern matching processes. REACT also includes our data access scheduling technique. Modern SSDs in practice achieve great I/O speed by making use of the extensive internal parallelism at the system-level, despite the low-level flash exhibiting considerable delay. However, pattern matching is limited in its use of parallelism since the subblocks, which make up the input data and might be located in many flash pages, must be evaluated sequentially in order to properly process the input. This restriction may result in the accelerator being underutilized.

Won Seob Jeong, Changmin Lee, Keunsoo Kim, Myung Kuk Yoon, Won Jeon, Myoungsoo Jung, Won Woo Ro, "REACT: Scalable and High-Performance Regular Expression Pattern Matching Accelerator for In-Storage Processing", IEEE Transactions on Parallel and Distributed Systems, vol.31, no.5, pp.1137-1151, 2020.

1. **Abstract:**

Applications like social networks, medication development, and recommendation systems all heavily rely on graph analytics. Application performance is constrained by storage access time due to the high size of graphs, which may be larger than main memory can accommodate. Through methods like graph sharding and sub-graph splitting, out-of-core graph processing frameworks attempt to alleviate this store access constraint. Even with these methods, storage systems constitute a substantial performance barrier due to the requirement to retrieve data across many graph shards or sub-graphs. The solid state drive (SSD) framework we offer in this study, GraphSSD, is a full system solution for storing, accessing, and running graph analytics on SSDs. GraphSSD considers graph structure for deciding on graph architecture, access, and update techniques rather than viewing storage as a collection of blocks. In order to reduce the number of page accesses, GraphSSD substitutes the traditional logical to physical page mapping process in an SSD with a cutting-edge vertex-to-page mapping scheme. By reducing needless page movement overheads, GraphSSD also offers effective graph updates (vertex and edge alterations).

Kiran Kumar Matam, Gunjae Koo, Haipeng Zha, Hung-Wei Tseng, Murali Annavaram, "GraphSSD: Graph Semantics Aware SSD", 2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA), pp.116-128, 2019.

1. **Abstract:**

For big data processing and cloud computing services today, storage systems are essential. Data processing and storage scalability are challenged by the constantly expanding size of computing and data analysis findings, which calls for more storage capacity. Additionally, today's storage systems are inefficient due to the growing complexity of the storage hierarchy and "passive" storage devices, which calls for the adoption of new storage technologies. In order to create an active cloud storage system where data may be handled on disk drives without data transportation, we investigate novel Ethernet linked drives with on-drive embedded CPU and DRAM in this study. These drives are software-defined storage-capable micro-storage servers. We test and assess on-drive data processing, such as data compression, aggregation, and erasure encoding, in addition to I/O activities, as these processes naturally enable data-intensive application.

Zhi Qiao, Shuwen Liang, Nandini Damera, Song Fu, Hsing-bung Chen, Michael Lang, "ACTOR: Active Cloud Storage with Energy-Efficient On-Drive Data Processing", 2018 IEEE International Conference on Big Data (Big Data), pp.3350-3358, 2018.

1. **Abstract:**

Large-scale storage systems frequently use key-value stores that are based on log-structured merge trees (LSM-trees). The fundamental cause is that big-data applications require great performance, which conventional relational databases are unable to deliver. LSM-tree-based key-value stores can handle high-throughput write operations and offer high sequential bandwidth in storage systems as high-throughput alternatives to relational databases. But when workloads are update-intensive, the compaction procedure results in write amplification and encounters poor write performance. To solve this problem, we develop a comprehensive key-value store called DStore that explores near-data processing (NDP) and on-demand scheduling for compaction optimization. The host-side and device-side subsystems' varied computational resources are fully utilized by DStore. According to the differing computing capabilities of the two sides, DStore separates the entire host-side compaction responsibilities into the aforementioned two-side subsystems. The gadget must, however, be equipped with an NDP model. The host and the device work together in parallel to complete the divided compaction duties. Key-value stores are made possible by the NDP-based devices' low latency and high bandwidth capabilities in DStore. DStore not only completes key-value store compaction but also enhances system performance. We use a variety of testbeds in our experiment and implement our DStore prototype on a real-world platform.

Hui Sun, Wei Liu, Zhi Qiao, Song Fu, Weisong Shi, "DStore: A Holistic Key-Value Store Exploring Near-Data Processing and On-Demand Scheduling for Compaction Optimization", IEEE Access, vol.6, pp.61233-61253, 2018.

1. **Abstract:**

A potential spintronic device called a spin switch (SS) uses the giant spin Hall effect, spin transfer torque, and dipolar coupling to achieve input-output isolation, compactness, low power consumption, and non-volatility. In this paper, we propose a novel device-to-architecture co-design for an in-memory computing platform based on coterminous SS (IMCS2) that can function as both non-volatile memory and reconfigurable in-memory logic (AND/NAND, OR/NOR, and XOR/XNOR) without the need for additional logic circuits on the memory chip. Using the shared memory peripheral circuits, the computed logic output may be read out as easily as a typical magnetic random access memory bit cell. The processing of data in memory using such inherent in-memory logic would significantly minimize the need for power-hungry and long-distance data transfer in the traditional von Neumann computer architecture.

Farhana Parveen, Shaahin Angizi, Zhezhi He, Deliang Fan, "IMCS2: Novel Device-to-Architecture Co-Design for Low-Power In-Memory Computing Platform Using Coterminous Spin Switch", IEEE Transactions on Magnetics, vol.54, no.7, pp.1-14, 2018.

1. **Abstract:**

The processor-memory data transfer bottleneck in computing systems can be reduced with the help of in-memory computing. Spintronics has drawn a lot of attention as a non-volatile memory technology, but recent research has demonstrated that its special features can also enable in-memory computation. We review the work done in this area and outline three new architectures that improve STT-MRAM by allowing it to evaluate transcendental functions, execute logic, arithmetic, and vector operations, all within memory arrays.

Shubham Jain, Sachin Sapatnekar, Jian-Ping Wang, Kaushik Roy, Anand Raghunathan, "Computing-in-memory with spintronics", 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.1640-1645, 2018.