REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
24-01-2023	Final Report	17-Apr-2017 - 16-Apr-2019

4. TITLE AND SUBTITLE
Final Report: Next-Generation All Flash Big Data Parallel Processing Engine for Mobile Computing

5a. CONTRACT NUMBER
W911NF-17-1-0208

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611103

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Central Florida
12201 Research Parkway, Suite 501

Orlando, FL 32826 -3246

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
70126-NC-RIP.1

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jun Wang
UU	UU	UU	UU		19b. TELEPHONE NUMBER 407-823-0449

RPPR Final Report
as of 25-Jan-2023

Agency Code: 21XD

Proposal Number: 70126NCRIP **Agreement Number: W911NF-17-1-0208**
INVESTIGATOR(S):

Name: Jun Wang
Email: jun.wang@ucf.edu
Phone Number: 4078230449
Principal: Y

Organization: **University of Central Florida**
Address: 12201 Research Parkway, Suite 501, Orlando, FL 328263246
Country: USA

DUNS Number: 150805653 EIN: 592924021
Report Date: 16-Jul-2019 Date Received: 24-Jan-2023
Final Report for Period Beginning 17-Apr-2017 and Ending 16-Apr-2019
Title: Next-Generation All Flash Big Data Parallel Processing Engine for Mobile Computing
Begin Performance Period: 17-Apr-2017 **End Performance Period:** 16-Apr-2019
Report Term: 0-Other

Submitted By: Jun Wang Email: jun.wang@ucf.edu
Phone: (407) 823-0449

Distribution Statement: 1-Approved for public release; distribution is unlimited.

STEM Degrees: **STEM Participants:**

Major Goals: We have acquired a massive big data and big compute instrument which enables us to conduct basic research in computer memory, computer storage and parallel computing areas. The enabled research will make potential contributions to our existing research projects and foster future DARPA related projects relating to big computing and big data.

We phased out an out dated Dell power-edge cluster and installed a new 10-node Silicon Mechanics Rackform R353.v6 GPU cluster in our college data center room. In addition, we purchased several machines with different storage capacity and API as clients in our research laboratory and connect remotely to the cluster.

Accomplishments: Instrument actually acquired
The new instrument is built and performed on the 10-node GPU server cluster named CASS with a SSD storage array. The CASS c l u s t e r headnode machine is configured with CPU: 2 x Intel Xeon Silver 4110, 2.1GHz (8-Core, HT, 2400 MT/s, 85W) 14nm
RAM: 192GB (12 x 16GB DDR4-2666 ECC Registered 1R 1.2V RDIMMs) Operating at 2666 MT/s Max. Nine serve nodes are configured with CPU: 2 x Intel Xeon E5-2650v4, 2.2GHz (12-Core, HT, 30MB Cache, 105W) 14nm
RAM: 128GB (8 x 16GB DDR4-2400 ECC Registered 1R 1.2V DIMMs) Operating at 2400 MT/s Max plus four NVIDIA GeForce GTX Titan XP Graphics Card, 12GB GDDR5X cards. In total, our new CASS distributed GPU cluster is equipped with 36 NVIDIA GeForce GTX Titan XP Graphics Cards, 232-core CPU 2x Intel Xeon (16+24*9), 1.344TB RAM, 40 TB Flash SSD direct attached storage and an Mellanox SX6005 SwitchX-2 FDR InfiniBand Switch, 12-port QSFP, 2PS, Short Depth, Power-to-Port Airflow.
In addition, we purchased several storage cloud clients and big data big compute servers as various terminals to connect to our new cluster to mimic various computation-intensive and data intensive applications and programs.

RPPR Final Report
as of 25-Jan-2023

Training Opportunities: how many undergraduate or graduate students have been trained or allowed to use the equipment/instrumentation,
>> About 150 undergraduate students and 20 graduate students have been trained by taking courses offered by the PI and his colleagues at both the Electrical and Computer Engineering department and Computer Science Department. Specific classes are EEL 6760 Data Intensive Computing, CDA 5601 Advanced Computer Architecture, and EEL 6762 Performance Analysis and Evaluation of Computer Systems at the graduate level, EEL 4798 Massive Storage and Big Data, EEL 4768 Computer Architecture at the undergraduate level. About three Ph.D. students successfully defended their dissertation. About ten graduate students have access to the instrument to conduct their ongoing research in big data.

Results Dissemination: how has the equipment/instrumentation contributed to scientific advancement,
>> The PI and his colleagues have developed several AI-enhanced solutions which will be used in the real data center and cluster machines, warehouse-scale computer and supercomputers in both industry and national labs. The research outcomes have been published in quality international technical conferences such as IEEE/ACM The International Symposium on Computer Architecture (ISCA) 2020, the IEEE 2020 International Conference on Distributed Computing Systems (ICDCS 2020), The IEEE International Conference on Computer Design ICCD 2018, IEEE Transactions on Distributed Systems, The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20) AAAI, etc. More specifically, the PI's team has developed several intelligent I/O pattern forecasting algorithms and integrated into the big data & learning machines with generalpurpose GPU cards. Current results indicate our intelligent solutions could accelerate I/O intensive program execution speed by about 33% to 70%.

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: jun wang
Person Months Worked: 3.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Jian Zhou
Person Months Worked: 3.00 **Funding Support:**
Project Contribution:
National Academy Member: N

RPPR Final Report
as of 25-Jan-2023

Partners

,

I certify that the information in the report is complete and accurate:
Signature: Jun Wang
Signature Date: 1/24/23 4:44PM

Next-Generation All Flash Big Data Parallel Processing Engine for Mobile Computing

Final Report

March 2022

PI Jun Wang, jun.wang@ucf.edu

University of Central Florida, Orlando, FL, USA

1. Project Summary

We have acquired a massive big data and big compute instrument which enables us to conduct basic research in computer memory, computer storage and parallel computing areas. The enabled research will make potential contributions to our existing research projects and foster future DARPA related projects relating to big computing and big data.

We phased out an out dated Dell power-edge cluster and installed a new 10-node Silicon Mechanics Rackform R353.v6 GPU cluster in our college data center room. In addition, we purchased several machines with different storage capacity and API as clients in our research laboratory and connect remotely to the cluster.

2. Instrument actually acquired

The new instrument is built and performed on the 10-node GPU server cluster named CASS with a SSD storage array. The CASS cluster headnode machine is configured with CPU: 2 x Intel Xeon Silver 4110, 2.1GHz (8-Core, HT, 2400 MT/s, 85W) 14nm

RAM: 192GB (12 x 16GB DDR4-2666 ECC Registered 1R 1.2V RDIMMs) Operating at 2666 MT/s Max. Nine serve nodes are configured with CPU: 2 x Intel Xeon E5-2650v4, 2.2GHz (12-Core, HT, 30MB Cache, 105W) 14nm

RAM: 128GB (8 x 16GB DDR4-2400 ECC Registered 1R 1.2V DIMMs) Operating at 2400 MT/s Max plus four NVIDIA GeForce GTX Titan XP Graphics Card, 12GB GDDR5X cards. In total, our new CASS distributed GPU cluster is equipped with 36 NVIDIA GeForce GTX Titan XP Graphics Cards, 232-core CPU 2x Intel Xeon (16+24*9), 1.344TB RAM, 40 TB Flash SSD direct attached storage and an Mellanox SX6005 SwitchX-2 FDR InfiniBand Switch, 12-port QSFP, 2PS, Short Depth, Power-to-Port Airflow.

In addition, we purchased several storage cloud clients and big data big compute servers as various terminals to connect to our new cluster to mimic various computation-intensive and data intensive applications and programs.

Table 1 below includes a list of major instruments we purchased and installed in data center and research laboratory.

Equipment	Quantity	Price (USD)
Computer Server With local RAID configuration, SSD and GPU from Silicon Mechanics Inc	10	150,000
Mellanox SX6005 SwitchX-2 FDR InfiniBand Switch, 12-port QSFP for a side fast storage interconnection network		5,000
Supermicro Storage Server	2	9,600
A VWS-1542881-DPN - Deep Learning DevBox (X299, 1x i7-7980x, 16GBx8 DDR4, 1x 256GB OS SSD, 1x 4TB HDD SW RAID5, 4x Titan Xp)	1	14,600
A GPU Accelerated Server	1	13,000
Total		199,200

Table 1: Acquired Instrument

3. Research Thrusts Enabled by Acquired Big Data Computing Instrument

The Input/Output wall problem, namely, growing disparity between off-chip memory and disk drive transfer rates is a long-standing research challenge. It becomes even more severe in today's big data and big compute era. Despite the data storage technologies evolve rapidly in recent years, the increasing heterogeneity and diversity in machines and workloads, coupled with the continued data explosion, exacerbate the speed gap between computing and second storage. There is an increasing need to develop a high-performance, and cost-effective architecture for emerging large-scale and diverse applications.

Enabled by our acquired new instrument, my research laboratory is able to conduct basic research to use Artificial Intelligence (AI in brief) to improve computer systems architecture, and develop new computer systems architecture for AI applications. This project entails several research thrusts as detailed below.

3.1 Using AI to Develop Learned Storage Systems Architecture

There is a trend to apply machine learning techniques to improve the performance of core storage components. Typically, a storage system is shared by many concurrent workloads. Because of this, it's challenging for contemporary machine learning algorithms to learn the hidden patterns from the entangled traces. We develop a novel temporal-aware sequence classification to mine the correlation between I/O requests and represents the addresses with multidimensional vectors that shows better spatial locality. We can efficiently split and clean the entangled I/O trace. By integrating with Recurrent Neural Network (RNN), we greatly improve the cache hit ratio for several concurrent file access workloads.

To classify the I/O sequence, one key is to identify appropriate features. Researchers from Google apply k-means classification to addresses to category the sub I/O sequences and

feed them to individual RNN models [1], [2]. The classification of CPU to memory accesses is well matched with the segment classification function of k-means. First, the CPU time is multiplexed by the applications using the time-sharing technique, in a given time segment we can assume there is only one application accessing the memory. Second, the physical memory addresses shared by the applications are segmented by the operation system. However, these two assumptions are not applied to external I/O access classification. To solve this problem, we develop a fine-grained Temporal Aware Sequence Classification (TAC) algorithm to find the sequence classifications by using a sliding time window. We aim to discover the hidden relation being buried in address domain, and project our trace records to a new multi-feature domain by considering the record correlation hidden in the time series. The convergence algorithm is being developed to implement a projection protocol taking both address and time series info of data into consideration.

Due to the specific internal data structures of various kinds of applications, there exists repeated access patterns in the trace. However, modern operation systems generally execute a mixture of a large number of I/O workloads concurrently. These concurrent workloads interweave their I/O sequences stochastically. This is substantially different to the DRAM workloads. Because, the DRAM workloads mainly share the DRAM bandwidth based on the timesharing mechanism in a multi-core CPU. However, the external I/O workloads share the resources through the hardware interrupts. In the consolidated I/O sequence, the amount of the noise requests substantially exceeds that of the original requests. Therefore, it is important and difficult to first discover access sequence of each individual workload from an entangled trace.

We hypothesize the correlation of I/O requests is implicit in I/O traces. Specifically, two or more requests are supposed to be strongly correlated if there have been accessed together in short time intervals repeatedly. According to this observation, we split the consolidated I/O sequence into multiple sub I/O sequences based on the address correlation. Thus, the requests within a sub sequence have a significant higher correlation than those across subsets. On the contrary, the interlaced trace is segregated based on the division of the address been requested. And each trace split is likely to be belonging to one or a few correlated transactions and with reduced noise.

Our TAC is an unsupervised learning-based approach to solve the problem. The idea is to split the address sequence and disentangle the multi-thread effects in I/O request trace according to their temporal correlation. TAC share part of the idea from both graph processing and clustering algorithms. Comparing to clustering RNN [1], TAC makes no assumption about the spatial locality of addresses. Instead, it scrutinizes the low-level I/O trace based on correlated addresses information from their time stamps. Given specific complexity and feasibility, we develop two distinct protocols to implement TAC.

1). Statistical-based I/O Classification

To deal with M addresses and N potential groups, we maintain a M×N matrix as the probability table. Each entry in the table represents the probability that specific address is affiliated with the target group. The objective is to maximize the probability that a cluster of addresses being correlated could be assigned to a same group.

We observe that strong correlated cells are likely to be accessed by one thread in succession, thus tend to appear in short intervals in the entangled trace. Therefore, we give a big weight to strengthen the virtual link between each pair of addresses which emerge in a fixed-length time window and update the probability table according to those links. Specifically, the probability that a cell is affiliated with a group is determined by the probabilities that the cells appeared in

previous time window are affiliated with that group. Given an increasing number of accesses to a certain cell, its correlated cells will obtain a higher chance to be accessed together and thus make a larger impact than others.

2). Graph-based I/O Classification

To address the issues of the statistical-based approach, we propose graph-based approach which procedure is similar but essentially different. Instead of binding the addresses with a probability matrix, we adopt a graph-based approach which maps the addresses to points in a N-dimensional linear space (typically set to 3 to 5). Upon the arrival of the requests, the points of correlated cells apply attractive force to each other while the uncorrelated ones apply repulsive force. As the points moving in the space, they will gradually form clusters while those clusters will be separated by repulsion. In this process, the repulsive force serves as regularization and makes the points more equally distributed.

To gauge the effectiveness and efficiency of our TAC, we conducted a comprehensive set of experiments, and collected results about the cache hit ratio of I/O accesses from our working SSD Emulator based prototype. We implemented the FTL module using different cache and prefetching algorithms. Regarding the learned I/O prefetcher module, we first implemented our TAC algorithm to split mixed workloads. Afterwards, we employ the LSTM model to predict the future I/O. We used both synthetic and real-world traces and divided traces into training part and testing part. the number of files grows, the classification results will follow the same principles. Obviously, as the attractive and repulsive force move the dots, the addresses from the same file gradually form clusters, and addresses from different files are effectively separated.

We demonstrate the effectiveness of our TAC algorithm in classifying the entangled I/O workloads [3]. We collect the I/O trace under a dedicated multithread workload. We then label all I/O request with the corresponding file id. TAC is an unsupervised classification algorithm that classify the I/O address in a way that the address belonging to one file is in the same cluster and vice versa. Without loss of generality, we show 8 files and thirty-two files classification in 3-dimensional linear space respectively. By using the TAC, we split the workloads into multiple clean I/O streams. We then use the RNN model to predict the future I/O. We split the collected I/O trace into training and testing datasets. We use the trained models to do the I/O prefetch and study the cache hit ratio, and compare the performance of the traditional LRU cache, the default LSTM (RNN) prefetcher, and the TAC + LSTM (TAC) prefetcher. We use 104 files and use 0.1% (in x1) and 1% (in x10) respectively. RNNx1 prefetch the first candidate predicted by the RNN model. RNNx10 prefetch the top 10 candidates predicted by the RNN model. TAC is the RNN with temporal-aware classification sitting in front. TACx1 prefetch the first candidate of the RNN prefetcher. TACx10 prefetch the top-10 candidates of the RNN prefetcher. LRUx1 use LRU cache and reserves the same amount of cache space as the RNNx1. LRUx10 use LRU cache and reserves the same amount of cache space as the RNNx10. Given new learned I/O prefetcher, we improve the cache hit ratio by 5-6 times for modern storage architecture.

3.2 ArchSampler: Architecture-Aware Memory Sampling for In-Memory Applications [4]

With the explosive rate of data growth, the limited Scalability of the DRAM technology defies the performance potentials for in-memory applications. Fortunately, emerging non-volatile memory (NVM) technologies, such as Phase-Change Memory (PCM) and Memristor, are promising candidates for replacing DRAM. Emerging NVMs are very dense, hence promise large capacities. Additionally, NVMs are non-volatile, thus enable persistent applications and

byte-addressable files. Both, density and persistency, are key enablers for in-memory applications. On the other side, emerging NVMs are slower than DRAM, thus optimizing for locality and avoiding contentions are key aspects to unlock the NVM performance.

In this project, we study the impact of memory contentions and architecture-oblivious implementations on the performance of sampling based in-memory approximation. Sampling has become an imperative technique used to accelerate big data processing, especially in today's emerging in-memory computing. However, we observe multiple of times slow-down for naive and default implementations of in-memory data sampling. Accordingly, we develop ArchSampler, an architecture-aware sampling library. The main idea is to exploits free choice of data samples to dynamically select which bank as a host to serve memory requests. Hence, ArchSampler enables efficient and high performing sampling through employing its knowledge of the NVM architectural details to maximize data locality and avoiding inter-thread contentions. Our evaluation shows that ArchSampler can achieve up to 1.62 speed up (1.20 on average) for different in-memory applications.

3.3 A Garbage Collection friendly Hash Indexing for Multi-stream SSDs within A Unified Memory-Storage Hierarchy

Flash-based solid state drives (SSDs) are increasingly adopted in unified memory-hierarchy. To eliminate implicit bulk data copy/swap between devices, unified memory hierarchy provides a single memory address space for all memories. Thus, the processors can directly access structured data in SSDs. Thanks to the byte-addressability supported by modern SSDs, most modern in-memory databases can benefit from unified memory hierarchy with little modifications. This inevitably incurs significant performance overhead and extra I/O traffic. Hash tables is one of the most widely used indexing schemes in in-memory databases, due to its excellent average query performance. Conventional hash indexing schemes are mostly designed for DRAM without considering the distinct hardware property when NAND flash is used in unified memory hierarchy. Most importantly, the random write workloads generated by hash indexing can hardly benefit from the optimization designed for NAND Flash SSDs, such as multi-stream technology.

However, existing hash schemes has not addressed to take the advantage of Multi-stream SSDs. To address this problem, we develop a Log Structured Hash Table (LSHT in brief), to reduce the Garbage Collection overhead cause by existing hash schemes in NAND flash memory. The idea behind LSHT is to separate the hot and cold data by their life time. Group data with similar lifetime and store them into same physical storage blocks. LSHT can efficiently improve the Garbage Collection performance of Multi-stream SSDs. We have implemented LSGH in commercial Muiti-stream SSDs and test it using real world traces, which demonstrate the efficiency this new scheme. We have also released the source code of LSHT for public use.

3.4 ApproxSSD: Data layout Aware Sampling on an Array of SSDs [5]

Execution of analytic frameworks on sample data sets is the current trend in response to increasing data size and demand for real-time analysis. Additionally, high-performance, energy-efficient Solid-State Drive (SSD) arrays are the primary storage subsystem for parallel data analysis systems. To exploit the benefits of SSD arrays when executing sample data set analytics, several key areas must be considered. First, due to logical to physical address translation, random data choice in data sampling jobs can cause unbalanced workloads among SSDs in the array. Second, after the data choice, existing task schedulers in data analysis

frameworks can introduce non-negligible resource contentions resulting from the suboptimal Input/Output (I/O). The performance of SSDs is unpredictable because of their varying maintenance costs at runtime, which renders them hard to be managed by the scheduler. With the trend towards sample set data analytics and the use of SSDs, it is increasingly important to ensure balanced workloads and minimize resource contention. Without addressing these areas, sample-set data analytics on SSDs will continue to suffer from performance inefficiencies. In this project, we propose ApproxSSD to perform on-disk layout-aware data sampling on SSD arrays. This proposed framework leverages data selection and task scheduling to improve the performance of many applications. ApproxSSD decouples I/O from the computation in task execution. This avoids potential I/O contentions and suboptimal workload balances. We have developed an open-source prototype system of ApproxSSD in Scala at Github. Our evaluation shows that ApproxSSD can achieve up to 2.7 times speed up at 10% sampling ratio under an example sampling workload when compared to Spark, while simultaneously maintaining high output accuracy.

3.5 A Correlation-Aware Page-Level FTL to Exploit Semantic Links in Workloads [6]

NAND Flash Solid State Disks (SSDs) are gaining tremendous popularity in today's storage market due to the unique erase-before-write feature of NAND flash, the Flash Translation Layer (FTL) in the SSD redirects the incoming writes to a free physical address and manages a logical to physical address mapping table. However, this induces significant performance degradation to the SSD. One of the main reasons is that current cache management in FTLs mainly focus on temporal or spatial locality. However, because of multiple levels of data buffers in the whole storage architecture, the locality of disk I/O is relatively low. What's more, the increasing capacity of SSD not only leads to mapping tables large in size, but also imposes high pressure on the efficiency of page-level address mapping. To overcome this limitation, we propose Correlation-Aware Page-level FTL, a.k.a CPFTL, which exploits I/O correlations in workload. First, a correlation-aware mapping table is developed based on the correlation in read operations. Second, we develop a correlation prediction table to support fast mapping entry lookup in correlation-aware mapping table. Third, because the data that has been flushed to the disk by the host buffer has low chances to get reused in a short timeframe, we developed separate read and write caches to improve the cache hit ratio. Finally, we develop a skew-aware dirty entry index to improve the "in-page" locality aware dirty cache update and thus reduce the garbage collection overhead. We developed an emulator and prototype, being open sourced at: https://github.com/janzhou/SSD-Emulator. Our experimental results show that CPFTL can reduce the average response time by 63.4% for read dominant workloads and 32.9% for transaction workloads.

4. Broad Impact

We summarize how DURIP awards are increasing university capacity to meet DoD research demands.

(1) how many undergraduate or graduate students have been trained or allowed to use the equipment/instrumentation,

>> About 150 undergraduate students and 20 graduate students have been trained by taking courses offered by the PI and his colleagues at both the Electrical and Computer Engineering department and Computer Science Department. Specific classes are EEL 6760 Data Intensive Computing, CDA 5601 Advanced Computer Architecture, and EEL 6762 Performance Analysis and Evaluation of Computer Systems at the graduate level, EEL

4798 Massive Storage and Big Data, EEL 4768 Computer Architecture at the undergraduate level. About three Ph.D. students successfully defended their dissertation. About ten graduate students have access to the instrument to conduct their ongoing research in big data.

2) how has the equipment/instrumentation contributed to scientific advancement,

>> The PI and his colleagues have developed several AI-enhanced solutions which will be used in the real data center and cluster machines, warehouse-scale computer and supercomputers in both industry and national labs. The research outcomes have been published in quality international technical conferences such as IEEE/ACM The International Symposium on Computer Architecture (ISCA) 2020, the IEEE 2020 International Conference on Distributed Computing Systems (ICDCS 2020), The IEEE International Conference on Computer Design ICCD 2018, IEEE Transactions on Distributed Systems, The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20) AAAI, etc. More specifically, the PI's team has developed several intelligent I/O pattern forecasting algorithms and integrated into the big data & learning machines with general-purpose GPU cards. Current results indicate our intelligent solutions could accelerate I/O intensive program execution speed by about 33% to 70%.

(3) has the equipment/instrumentation facilitated a technology breakthrough that has resulted in additional teaming with other DoD scientists/engineers.

>> Given the big data parallel storage the system, we set up an in-house working testbed for big data and learning systems tune-up. The PI Wang is looking for leading DoD related research projects about big data and learning storage and memory systems architecture. We have also been actively working with some of the PI's colleagues who are carrying out Navy research contracts such as Dr. Amro Awad entitled "Novel Hardware-Support for Ensuring Confidentiality and Integrity on Emerging Non-Volatile Memories State Active" from Naval Information Warfare Center Pacific.

References

[1] M. Hashemi, K. Swersky, J. A. Smith, G. Ayers, H. Litz, J. Chang, C. Kozyrakis, and P. Ranganathan, "Learning memory access patterns," arXiv preprint arXiv:1803.02329, 2018.
[2] L. Peled, U. Weiser, and Y. Etsion, "Towards memory prefetching with neural networks: Challenges and insights," arXiv preprint arXiv:1804.00478, 2018.
[3] J. Zhou, J. Wang. IOMeans: Classifying Multi-concurrent I/O threads using Spatio-Tempo Mapping. 2019 International Conference on High Performance Big Data and Intelligent Systems. Shenzhen, China.pp 249-254. IEEE catalog no.: CFP19512-PRT. ISBN: 978-1-7281-0466-9.
[4] J. Zhou, J. Wang. ArchSampler: Architecture-Aware Memory Sampling Library for In-Memory Applications. The 36th IEEE International Conference on Computer Design. IEEE International Conference on Computer Design (ICCD'18). October 7-10, 2018. Orlando, FL, USA.