ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36418

Running a Virtual Research Experience for Undergraduate (REU) Site in Computing Systems

Wei Zhang

Dr. Wei Zhang is the professor and Chair of the Department of Computer Engineering and Computer Science at the University of Louisville. He received his Ph.D. in Computer Science and Engineering from the Pennsylvania State University in 2003. Dr. Zhang served as an assistant/associate professor in Electrical and Computer Engineering at Southern Illinois University Carbondale (SIUC) from 2003 to 2010 and as an associate and full professor at Virginia Commonwealth University from 2010 to 2019. His research interests are in computer architecture, compiler, real-time computing, and hardware security. Dr. Zhang has led 9 NSF projects as the PI and has published 160+ papers in refereed journals and conference proceedings. He received the 2016 Engineer of the Year Award from the Richmond Joint Engineer Council, the 2009 SIUC Excellence through Commitment Outstanding Scholar Award for the College of Engineering, and the 2007 IBM Real-time Innovation Award.

Running a Virtual Research Experience for Undergraduate (REU) Site in Computing Systems

Abstract:

This paper summarizes our experiences in running the REU site in a virtual environment at the University of Louisville. This is our first year to run the REU site. While our original plan when we proposed this project was to have a traditional in-person program, the ongoing COVID-19 pandemic and the concerns about safety for both faculty mentors and students involved made us decide to run it virtually. While we had to cancel some in-person activities such as face-to-face meetings, tours, and social events, we also added virtual events such as private and group MS Team meeting, Slack chat rooms (channels), and online movie nights and discussions. Nine out of the ten research projects were conducted entirely virtually. For one project that involves hardware component, we managed to mail a hardware kit to the student so that she could still work on her project remotely. Student evaluations indicate that this virtual REU site program, though in its first year, was quite successful and satisfactory.

Introduction

The Research Experiences for Undergraduates (REU) Site in the Computer Science and Engineering (CSE) Department at the University of Louisville (UofL), funded by a recent grant from the National Science Foundation, ran for ten weeks from May to July 2021. Ten undergraduate students participated in intensive research with eight dedicated faculty mentors from the CSE Department. The central objective of this REU site project is to provide research and professional development opportunities to undergraduate students from underrepresented groups (including women, underrepresented minorities, and persons with disabilities) and academic institutions with limited research programs.

The research focus of this REU Site was on computer systems, which broadly concentrated on the design and development of hardware and software necessary to create platforms that meet users' computing requirements. The REU Site projects include a broad range of topics, ranging from processor architecture to storage systems, and from the small-scale (mobile and edge devices) to the large-scale systems (e.g., data centers). All the research projects in this REU site require students to share common knowledge on computing systems architecture and performance and/or energy evaluation, enabling students to understand each other’s work and broaden their perspectives of different computer systems. Within this CSR scope, our site’s general research goals are to advance energy efficiency, time predictability, performance, as well as security & privacy, scalability, and sustainability of computer systems.

Participants

The REU program sought to recruit a diverse group of students, including students from underrepresented groups in computer science (including women and underrepresented minorities), students from outside of the host institution, and students from institutions with limited research. The program successfully recruited 10 students and was successful in regard to diverse recruitment, recruiting the following groups of students:

- Participants from outside of the host institution: 8 of 10
- Participants from underrepresented groups: 3 of 10
- Participants from institutions with limited research: 5 of 10

Geographically, four of the students are from Kentucky. The six out-of-state students include one from each of the six states: California, New Jersey, Pennsylvania, Indiana, Ohio, and Arkansas. These areas span across from the East Coast and the Midwest to the West Coast, covering three different time zones including Eastern, Central, and Pacific. Given that California is 3 hours behind the EST, we tried to make online events asynchronous whenever possible. For our group meetings and other synchronous events, we tried to schedule them in the early afternoon so that the student from California could join us at a relatively convenient time.

Research Projects and Mentors

This REU site program involved eight faculty from the CSE Department at UofL who served as mentors. Two of the faculty mentors supervised two students each, involving them in two different components of the proposed projects. Each of the other six faculty mentors supervised one student each. The topics of the projects are listed in Table 1. This list was also published on our REU student application website for students to choose from. We asked each student to identify the top 3 projects they would like to involve in their application. We then examined students' demands and were able to match students' interests with the proposed projects for all ten participants. In most cases, students were able to participate in a project that was their first or second choice.

Table 1. List of the REU projects in computing systems at the UofL in Summer 2021

Mentor	Title of the Project
Dr. Wei Zhang (PI)	Reducing Performance Variability for Inference Execution at the Edge
Dr. Nihat Altiparmak (co-PI)	Energy Efficient OS Components for Emerging Data Storage Devices
Dr. Dar-Jen Chang	Graph Databases and Graph Analytics
Dr. Anup Kumar	Cybersecurity Challenges in Internet of Things (IoT) Applications
Dr. Adrian Lauf	Embedded Systems and Accelerators
Dr. Olfa Nasraoui	Explainable Machine Learning Algorithms
Dr. Juw Won Park	Large Scale Document Inversion using GPU Parallelization
Dr. Hui Zhang	Enabling Data-Intensive Computing on the Web with NSF XSEDE and MRI

This paper also includes two sample projects as described below to provide some more details of the research projects hosted on this REU site. A complete list of project descriptions provided to REU students can be seen at our project website at https://engineering.louisville.edu/academics/departments/computer/reu-site-summer-research-program-computer/.

Sample Research Project 1 – Energy Efficiency of Storage IO Completion Mechanisms: Various innovations have taken place in the storage subsystem within the past few years, including wide adoption of the Non-Volatile Memory Express (NVMe) interface for new generation storage devices [1], the corresponding multi-queue request submission/completion capability (blk-mq) implemented in the

Linux block IO layer [2-3], and new IO scheduling algorithms specifically designed for blk-mq [4-6]. We have also seen the emergence of new NVM based storage technologies such as Intel's Optane Solid-State Drive (SSD) based on their 3D XPoint technology [7], Samsung's Z-SSD with their SLC based 3D NAND technology [8], and Toshiba's XL-Flash design using a similar technology as Z-SSDs [9-10]. All these innovations enable a new generation of Ultra-Low Latency (ULL) SSDs that are broadly defined as providing sub-10 μs data access latency [11]. This new level of ULL IO performance questions the suitability of traditional interrupt-based IO completion mechanisms [12]. Both industry and academia suggested the use of polling-based IO completion methods for improved latency in such devices [13-20]. However, the IO completion mechanism's impact on energy efficiency has not been investigated.

In this REU project, the undergraduate student will investigate the effect of existing IO completion mechanisms on the system's overall energy efficiency, specifically for ULL storage performance, and develop a new, energy-efficient dynamic polling technique. For this research, the student will be provided an Intel Optane SSD [7] attached to a dedicated server and an Onset HOBO Power meter [21] connected to the power supply of the server. Using this experimental setup, the student will initially measure, analyze, and compare the impact of the following IO completion mechanisms on the system's overall energy efficiency, which will pave the way for energy-efficient storage stack designs:

1. Interrupt Based IO Completion: For traditional storage devices such as HDDs, IO latency is on the order of milliseconds. This is enough time for the CPU to perform a context switch and run other tasks, where completion of the request is handled asynchronously by the use of hardware interrupts (IRQ). When the storage device completes a request, the operating system is notified of the interrupt, performs a context switch, and executes the Interrupt Service Routine (ISR). Using interrupts is preferable for high IO latency devices; however, the overhead of switching and handling interrupts can be significant for low latency devices. If the rate of interrupts is exceedingly high, then it can even overload the system by creating a livelock situation [22].
2. Polling Based IO Completion: For high-performance devices, such as ULL SSDs with IO latencies on the order of microseconds, the cost of context switching, task scheduling, and interrupt handling generally outweighs the benefit given to another task. In polling, the kernel continuously queries for completed IO requests without switching to another task, and handles completions immediately, which has been supported since the Linux kernel version 4.4. By avoiding context switching, in addition to reduced overhead and improved IO latency, there are other performance benefits such as not polluting hardware and memory caches (TLB) and eliminating the execution of ISR [12]. However, continuously polling for IO completion is expected to put significant pressure on the CPU, possibly increasing energy consumption.
3. Hybrid Polling Based IO Completion: Instead of continuously polling for IO completion, hybrid polling offers a trade-off between interrupt and polling-based IO completion handling, in which the polling process sleeps between polls. This will reduce the CPU activity and energy consumption, but how to determine the sleep amount is a very important research question. Hybrid polling has been supported since the Linux kernel version 4.10 and current designs either use a user-specified static sleep time, or a system-determined static sleep time [17].

One common hypothesis is that interrupt-based IO completion would be more energy-efficient than polling since in polling the CPU is continuously querying for completion. However, considering ULL SSDs where the performance reaches millions of IO operations per second due to their rich internal parallelism, handling millions of interrupts per second may easily falsify this hypothesis. When we also consider the existing static hybrid polling options, which IO completion method is the most energy-efficient for ULL IO performance becomes an interesting open research question that we would like to initially answer in this REU project. Our next research direction is developing a new, dynamic hybrid polling technique that can automatically adjust its polling interval based on observed IO performance changes, which requires new research in efficient and accurate workload characterization. This project will pave the way for energy-efficient IO completion methods for emerging storage devices.

Sample Research Project 2 – Reducing Performance Variability for Inference Execution at the Edge: In order to avoid the communication overheads of transmitting massive amounts of data and to improve privacy and real-time analysis, the current trend is to move computation away from data centers towards the edge of the network. For Machine Learning (ML) applications, while training can be done at the cloud, there is an increasing push to move inference execution, especially deep learning, to the edge. A recent study by Facebook [23] shows that mobile inference performance (e.g., the convolutional neural network layer) exhibits significant variability, even across the same device running the same software stack. This variability, caused by both hardware and software, is harmful to applications that demand real-time inference such as object detection in autonomous driving, and surveillance video processing and analytics.

Current research on computer architecture support for ML at edge devices has mainly focused on measuring the average inference latency in a static environment where the ML model is running with little interference from other applications. However, this is not the case in real environments. Applications running concurrently on the same processor can cause significant performance variation in the inference latency. In this project, the student will first evaluate and analyze the benchmark behaviors of the MLPerf Inference. Based on workload characteristics, a particular architectural configuration can be selected as the baseline to execute the inference independently and efficiently. The student will then quantitatively study how executing multiple applications at the same time increases contention on shared resources such as memory and the on-chip network and studies their impact on inference latencies. Resource partition and management techniques will be investigated to reduce the performance variability for inference benchmarks without significantly impacting the overall performance of different applications running on the edge.

With the end of Moore's law, it is an exciting time to study new computer architectures for emerging applications such as deep learning. The research conducted in this project can increase our understanding of the machine learning inference code and its interaction with other applications on mobile and edge devices, as well as their implication on computer architectural design. The application-aware resource management techniques will be essential to provide high and predictable performance to enable real-time inferences at mobile and edge devices.

Research Training and Seminars

Before the students began to do their research projects, the PI Dr. Zhang and co-PI Dr. Altiparmak provided students with intensive research training in the first week of this program. The topics included computer architecture fundamentals and research, and Operating Systems basics and components. Since most students would use Linux in their research projects, we also offered a complete Linux command line tutorial, and how to install Ubuntu and programming in C and Linux. At the end of the first week, we also introduced topics on how to read research papers, how to write research papers, how to give a research talk, and how to prepare a research poster.

We also hosted a virtual weekly seminar between the 2nd and the 9th week. PIs and the faculty mentors presented various computer systems research areas in embedded and multicore systems; mobile and extensible distributed systems; cloud and data-intensive processing systems; machine learning; and memory, storage, and file systems. We also covered various computation, storage, and acceleration techniques, benchmarks and research tools, and hardware/software tradeoffs. In addition, faculty mentors shared their personal research journeys with participants, which could provide students insights on the joy and methods for doing research in computing systems or computer science broadly. The post-program survey showed that the students were better motivated to do research after their REU experiences this summer. Eight out of ten participants indicated that they would plan to pursue a graduate study program after graduation.

Social Events

Probably due to the remote learning in most of the year in 2020 and the spring of 2021, students seemed to be reasonably prepared and ready for online training, virtual collaboration, and conducting research work remotely. While we had planned a variety of social events such as the program-wide picnic, the end-of-the-project banquet, and three field trips to local museums and technology centers, all these activities had to be canceled since we decided to run this site virtually as the pandemic still continued. To encourage student-to-student communications, we added social channels on Slack for participants to connect with each other and to share common questions. We also compiled a list of movies with a theme on computer science or generally math and sciences. However, the 3 hours of difference in time zones made it hard for us to schedule the movie nights as a synchronous event. Instead, we suggested that students could watch the movie on Thursday nights at their convenient time, and then join the movie discussions on Friday afternoon at the end of the group meeting.

The list of movies we used this summer is listed in Table 2. While we understand that the virtual "movie nights" cannot fully replace the in-personal social activities, it seems to help motivate students as all these movies were themed in computer science or generally math and sciences. One of the participants provided the following feedback – "The movies were a nice relevant touch to the overall goal of the program, and seeing them within light of what is trying to be accomplished here also inspired me in some ways of pursuing higher education and exploring new ideas."

Table 2. List of movies for the asynchronous virtual movie nights.

Week	Movie
1	The Imitation Game
2	A Beautiful Mind
3	Good Will Hunting
4	The Theory of Everything
5	Moneyball
6	2001: A Space Odyssey
7	Hidden Figures
8	Steve Jobs
9	Pirates of Silicon Valley
10	WarGames

Overall Evaluation of the REU Program

All ten participants completed a pre-program survey of questions on their current knowledge, perceptions, and attitudes related to computer science on their first day of the program in May. They answered the same questions in a post-survey at the conclusion of the program, as well as an evaluation of the program, on the final day of the program in July. The response rate was 100% for both instruments. Overall, respondents were highly satisfied with the REU Program *[1 = strongly disagree to 5 = strongly agree]*, and all 10 would recommend the program to a friend.

I was satisfied with the REU program.	**4.80 / 5 (SD = 0.42)**
I am confident I will use what I learned this summer to be successful in my future coursework.	**4.70 / 5 (SD = 0.67)**
I am confident I will use what I learned this summer to be successful in my future career.	**4.80 / 5 (SD = 0.42)**

Satisfaction with Program Components

Respondents evaluated six different components of the REU program with respect to their level of personal satisfaction *[1 = strongly disagree to 5 = strongly agree]*. The highest and lowest rated program components, based on average scores across respondents, are noted in the adjacent boxes.

Strongest Satisfaction

-The faculty in the REU program are experts in their fields. **5/5** (st.dev. = 0)

-The faculty and facilitators of the REU program cared about me and my learning. **4.7/5** (st.dev.= 0.48)

-The academic elements of the REU program enhanced my learning. **4.7/5** (st. dev.= 0.48)

-The faculty in the REU program were excellent academic mentors. **4.6/5** (st. dev.=0.52)

Weakest Satisfaction

-The relationships I developed with peers at the REU program enhanced my overall experience. **3.5/5** (st. dev.=1.27)

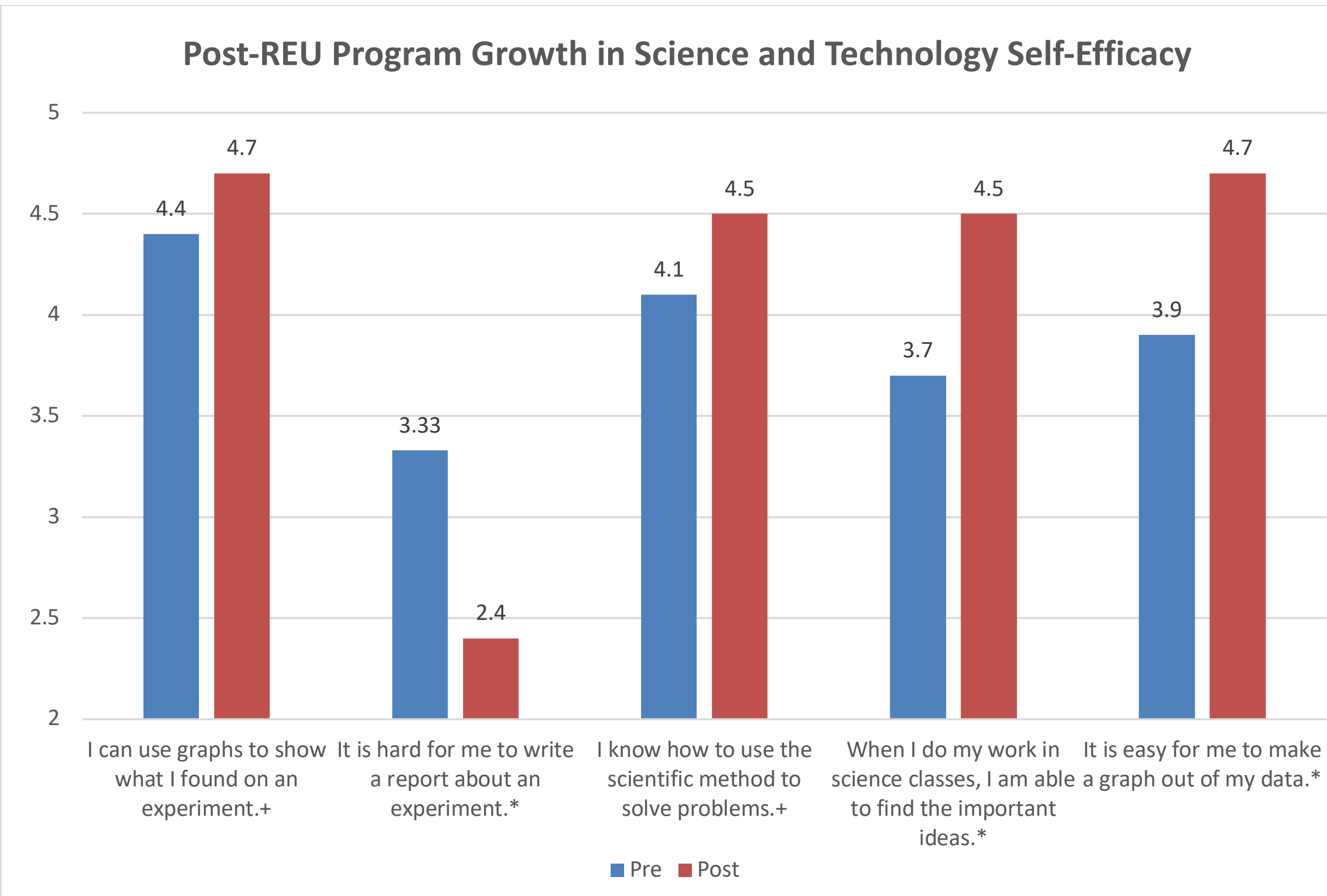
-I was satisfied with the non-academic elements of the REU program. **4.1/5** (st. dev.=0.74)

Overall, participants were most satisfied with the REU faculty and academic elements of the program. Participants unanimously strongly agreed that REU faculty were experts in their fields. They also reported high satisfaction with the level of care shown by faculty, the academic elements of the program, and the mentorship of the faculty. Weaker levels of satisfaction were reported related to relationships made with peers and in regard to non-academic elements of the program. These weaker scores are not surprising given the program was moved to a remote offering due to circumstances surrounding the COVID-19 pandemic. Participants interacted with peers only virtually. While virtual movie nights and other online activities sought to foster informal interactions and social connections between participants, this is achieved more easily in face-to-face environments.

Impact of the REU on Participants' Knowledge and Perspectives on Computer Science
Growth in Science and Technology Self-Efficacy

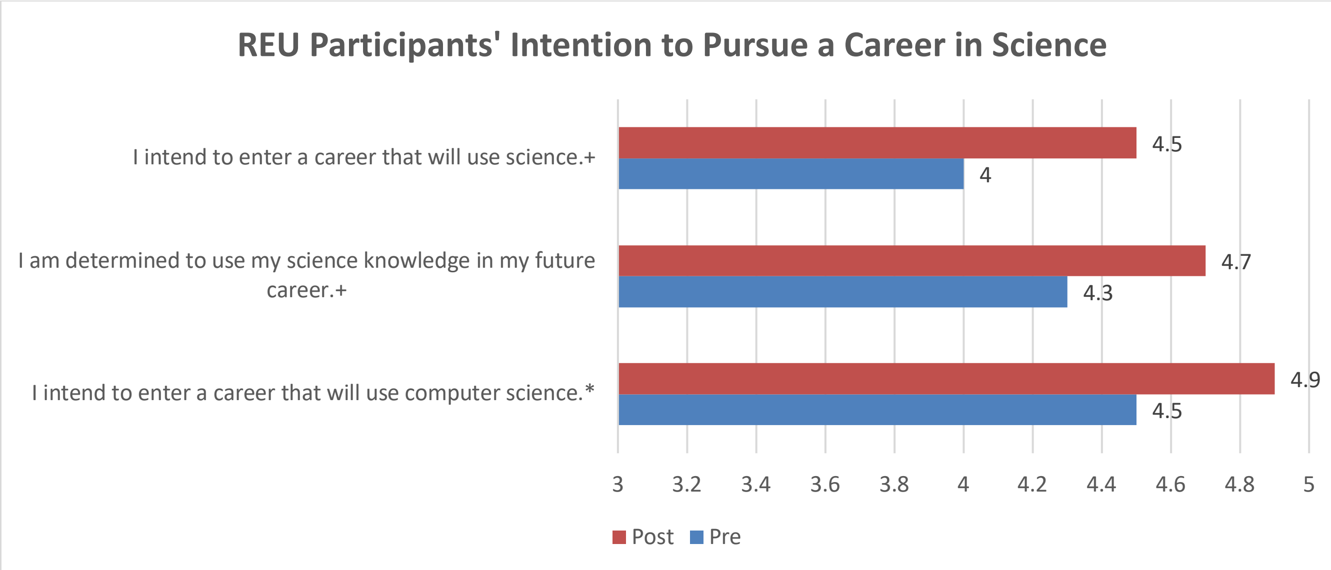
Participants were asked to indicate their level of agreement (1=strongly disagree to 5=strongly agree) on a series of items adapted from the Self-Efficacy in Technology and Science (SETS) instrument on both the pre- and post-surveys. Higher scores indicate a participant has a higher degree of self-efficacy.

Although finding statistically significant improvement in self-efficacy is difficult to measure with such a small *n (n = 10)*, the analysis nevertheless shows participant growth in a number of categories. The graph below shows growth across topics. Because of the small n, growth at both the 0.1 and 0.05 levels are reported. Growth statistically significant at the 0.1 level is somewhat likely to gain statistical significance at the 0.05 level when combined with future cohorts to expand the sample size. Differences statistically significant at the 0.1 level are indicated with a plus sign (+) following the question, while differences statistically significant at the 0.05 level are indicated with an asterisk (*) following the question. The differences represent growth ranging from 7.5% ("I can use graphs to show what I found on an experiment") to 23.25% ("It is hard for me to write a report about an experiment") of the total 5-point scale. Average reported self-efficacy was higher in a large number of additional categories, but this growth was not statistically significant.



Increase in Likelihood to Pursue a Science Career- SETS

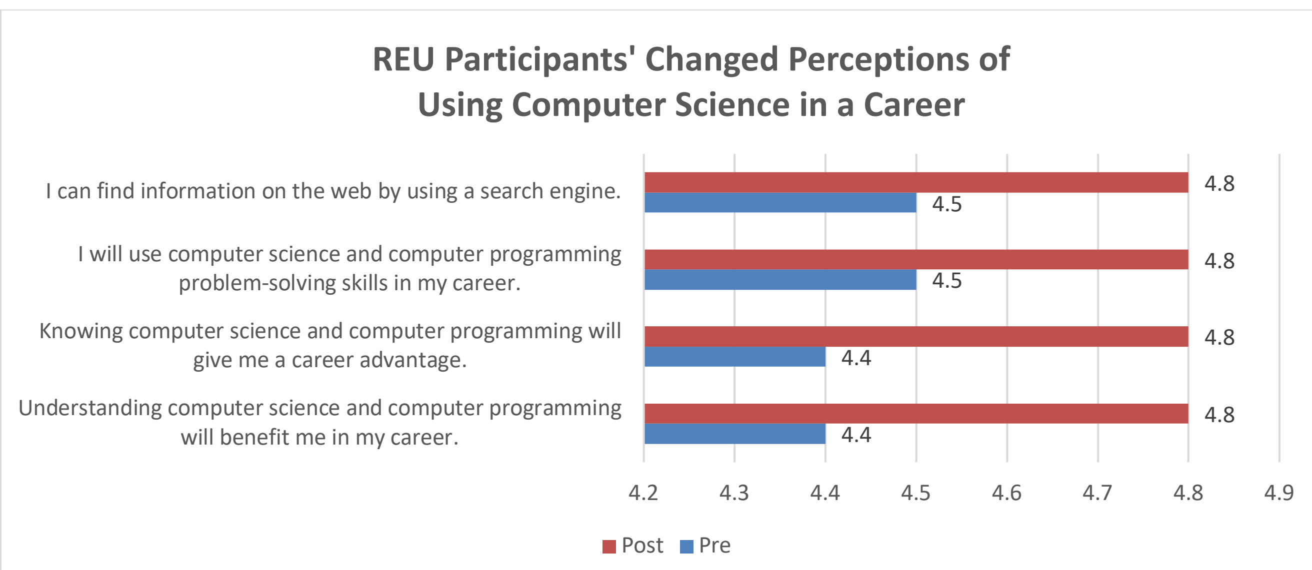
Participants were asked to indicate their level of agreement (1=strongly disagree to 5=strongly agree) on additional items in the Self-Efficacy in Technology and Science (SETS) instrument related to their intended future careers. On three items, participants showed statistically significant change in their reported intentions to pursue a science career. Specific questions and amount of growth are shown in the chart below.



Increase in Likelihood to Pursue a Science Career- Science Motivation Questionnaire

Items adapted from the Science Motivation Questionnaire II instrument further assessed participants' motivation to engage in STEM for career outcomes. Participants responded on the pre- and post-surveys to 5 Likert-type items in terms of their thinking relating to how frequently they view stem-career elements as important to their futures. Responses included never (1), rarely (2), sometimes (3), often (4), and always (5).

Participants showed statistically significant growth at the 0.1 level in four of the questions. These results are reported below.



Concluding Remarks

This paper summarizes our first-year experience in running this REU site program in a virtual environment. It was a hard decision for the project team to make by considering the ever-changing pandemic situations – a mix of an optimistic view due to the availability of vaccines in the spring of 2021 and a more cautious view due to the emergence of the Delta Variant, mostly abroad at the time when we made the decision. We also considered the COVID-19 related travel regulations such as the quarantine requirement for out-of-state travelers since six of our REU participants were from other states. We hoped that the virtual REU site would provide students a meaningful research experience in a safe manner.

The student evaluation of our REU site program is mostly very positive, which is encouraging. However, we believe we cannot generalize the success of the virtual REU experience. We believe this is also partially due to the nature of the research projects we proposed, which can be conveniently done in a virtual environment (except for one project for which we can mail a hardware kit but can still be done remotely). For REU projects that require physical access to research equipment and facilities, it could be much harder. Also, if we still need to run this REU site virtually again next summer, we will explore academic and non-academic activities to increase peer-to-peer interactions among REU participants.

Acknowledgment

The author would like to thank all collaborators who are involved in the REU site project (Dr. Altiparmak as a co-PI, Drs. Chang, Kumar, Lauf, Nasraoui, Park, H. Zhang as faculty mentors, and Dr. Jennifer Anderson as the evaluator). The author would also like to acknowledge the support from the National Science Foundation with the award 2050925.

References

[1] NVM Express Base Specification, rev. 1.4. https://nvmexpress.org/wp-content/uploads/NVM-Express-1_4-2019.06.10-Ratified.pdf.
[2] Matias Bjørling, Jens Axboe, David Nellans, and Philippe Bonnet. 2013. Linux Block IO: Introducing Multi-queue SSD Access on Multi-core Systems. In Proceedings of the 6th International Systems and Storage Conference (Haifa, Israel) (SYSTOR '13). ACM, New York, NY, USA, Article 22, 10 pages. http://doi.acm.org/10.1145/2485732.2485740.
[3] Arash Tavakkol, Juan Gómez-Luna, Mohammad Sadrosadati, Saugata Ghose, and Onur Mutlu. 2018. MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices. In 16th USENIX Conference on File and Storage Technologies (FAST '18). USENIX Association, Oakland, CA, 49–66. https://www.usenix.org/conference/fast18/presentation/tavakkol.
[4] Kyber multiqueue I/O scheduler. https://patchwork.kernel.org/patch/9672023/.
[5] Mq-deadline multiqueue I/O scheduler. https://github.com/torvalds/linux/blob/master/block/mq-deadline.c.
[6] Budget Fair Queueing [n.d.]. Budget Fair Queueing (BFQ) Storage-I/O Scheduler. http://algo.ing.unimo.it/people/paolo/disk_sched/.
[7] Intel Optane SSD 900P Series Product Brief. https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/optane-ssd-900p-brief.pdf.
[8] Samsung SZ985 Z-NAND SSD. https://www.samsung.com/us/labs/pdfs/collateral/Samsung_Z-NAND_Technology_Brief_v5.pdf.
[9] 2019. Goke 2311 SSD. https://www.bloomberg.com/press-releases/2019-09-30/goke-2311-series-drives-toshiba-memory-xl-flash-based-nvme-ssds.

[10] Shigeo (Jeff) Ohshima. Scaling Flash Technology to Meet Application Demands. https://flashmemorysummit.com/English/Conference/Keynotes.html.
[11] Gyusun Lee, Seokha Shin, Wonsuk Song, Tae Jun Ham, Jae W. Lee, and Jinkyu Jeong. 2019. Asynchronous I/O Stack: A Low-latency Kernel I/O Stack for Ultra-Low Latency SSDs. In 2019 USENIX Annual Technical Conference (USENIX ATC '19). USENIX Association, Renton, WA, 603–616. https://www.usenix.org/conference/atc19/presentation/lee-gyusun.
[12] Jisoo Yang, Dave B. Minturn, and Frank Hady. 2012. When Poll is Better than Interrupt. In Proceedings of the 10th USENIX Conference on File and Storage Technologies (San Jose, CA) (FAST '12). USENIX Association, USA.
[13] Adrian M. Caulfield, Arup De, Joel Coburn, Todor I. Mollow, Rajesh K. Gupta, and Steven Swanson. 2010. Moneta: A High-Performance Storage Array Architecture for Next-Generation, Non-Volatile Memories. In Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '43). IEEE Computer Society, USA. https://doi.org/10.1109/MICRO.2010.33
[14] Assaf Eisenman, Darryl Gardner, Islam AbdelRahman, Jens Axboe, Siying Dong, Kim Hazelwood, Chris Petersen, Asaf Cidon, and Sachin Katti. 2018. Reducing DRAM Footprint with NVM in Facebook. In Proceedings of the Thirteenth EuroSys Conference (Porto, Portugal) (EuroSys '18). Association for Computing Machinery, New York, NY, USA, Article 42, 13 pages. https://doi.org/10.1145/3190508.3190524
[15] Hyeong-Jun Kim, Young-Sik Lee, and Jin-Soo Kim. 2016. NVMeDirect: A User-space I/O Framework for Application-specific Optimization on NVMe SSDs. In 8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage '16). USENIX Association, Denver, CO. https://www.usenix.org/conference/hotstorage16/workshop-program/presentation/kim
[16] Jiuxing Liu and Bulent Abali. 2009. Virtualization Polling Engine (VPE): Using Dedicated CPU Cores to Accelerate I/O Virtualization. In Proceedings of the 23rd International Conference on Supercomputing (Yorktown Heights, NY, USA) (ICS '09). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1542275.1542309
[17] Damien Le Moal. 2016. I/O Latency Optimization with Polling. https://events.static.linuxfound.org/sites/events/files/slides/lemoal-nvme-polling-vault-2017-final_0.pdf.
[18] S. Swanson and A. M. Caulfield. 2013. Refactor, Reduce, Recycle: Restructuring the I/O Stack for the Future of Storage. Computer 46, 8 (2013), 52–59.
[19] Z. Yang, J. R. Harris, B. Walker, D. Verkamp, C. Liu, C. Chang, G. Cao, J. Stern, V. Verma, and L. E. Paul. 2017. SPDK: A Development Kit to Build High Performance Storage Applications. In 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). 154–161. https://doi.org/10.1109/CloudCom.2017.14.
[20] Jie Zhang, Miryeong Kwon, Donghyun Gouk, Sungjoon Koh, Changlim Lee, Mohammad Alian, Myoungjun Chun, Mahmut Taylan Kandemir, Nam Sung Kim, Jihong Kim, and Myoungsoo Jung. 2018. FlashShare: Punching Through Server Storage Stack from Kernel to Firmware for Ultra-Low Latency SSDs. In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18). USENIX Association, Carlsbad, CA, 477–492. https://www.usenix.org/conference/osdi18/presentation/zhang.
[21] Onset HOBO® UX120-018 Data Logger (datasheet). Onset. https://www.onsetcomp.com/datasheet/UX120-018.
[22] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. 2018. Operating Systems: Three Easy Pieces (1.00 ed.). Arpaci-Dusseau Books.
[23] Carole-Jean Wu, David Brooks, Kevin Chen, Douglas Chen, Sy Choudhury, Marat Dukhan, Kim Hazelwood, Eldad Isaac, Yangqing Jia, Bill Jia, Tommer Leyvand, Hao Lu, Yang Lu, Lin Qiao, Brandon Reagen, Joe Spisak, Fei Sun, Andrew Tulloch, Peter Vajda, Xiaodong Wang, Yanghan Wang, Bram Wasti, Yiming Wu, Ran Xian, Sungjoo Yoo, Peizhao Zhang. Machine Learning at Facebook:

Understanding Inference at the Edge. In Proc. of the IEEE International Symposium on High Performance Computer Architecture (HPCA), 2019.