Research Statement

Adwait Jog, Department of Computer Science

1. Overview

For decades, there has been a continuous need for high-performance and energy-efficient computing for a variety of real-world applications. Such applications are constantly emerging from a wide range of areas such as medicine, science, finance, engineering, social media, and gaming. To meet such needs, computer architecture research has gained tremendous attention in the community. This area has also shaped the computer hardware industry for decades and will continue to be important as future computers need to meet the never-ending demand of exascale computing (and beyond) on a very stringent power budget. However, meeting such a demand will not be easy given emerging design, cost, and power constraints. To this end, the computer architecture community is experiencing a paradigm shift in the way computing is performed – it is becoming more heterogeneous and sophisticated. Almost all computing platforms now incorporate some specialized devices (called as *accelerators*) alongside traditional multi-core processors (CPUs). Such a setup has shown to be very promising, however, it is *far* from reaching the optimal performance and efficiency demands. Furthermore, it is also *not* immune to various faults and malicious attackers, making them unreliable and vulnerable. Addressing such deficiencies is challenging but also has brought unique opportunities for computer architects (including my research group) to make a positive impact.

Research Trajectory. At William & Mary (W&M), I am leading an independent Insight Computer Architecture Lab (`https://insight-cal.github.io/`) with a vision to make heterogeneous computing more efficient, reliable, and secure. Specifically, my current focus has been on general-purpose (e.g., Graphics Processing Units (GPUs)) and custom accelerators (e.g., Automata Processors). Figure 1 shows my research journey at W&M. My Ph.D. dissertation (before 2015) research was mostly focused on improving the performance of GPUs. Since arriving at William & Mary in Fall 2015, along with my students, I have taken a trajectory that has expanded my research in four related research directions: a) continuing to improve system performance and overall throughput of GPUs, b) reducing/managing data movement and improving energy efficiency in GPUs, c) improving GPU security and reliability by carefully trading-off performance and energy, and d) exploring and improving the capability of custom accelerators and also infusing their key features into general-purpose GPUs to keep them relevant in the future years to come. A high-level overview of these research directions is also captured by W&M news [1, 2].

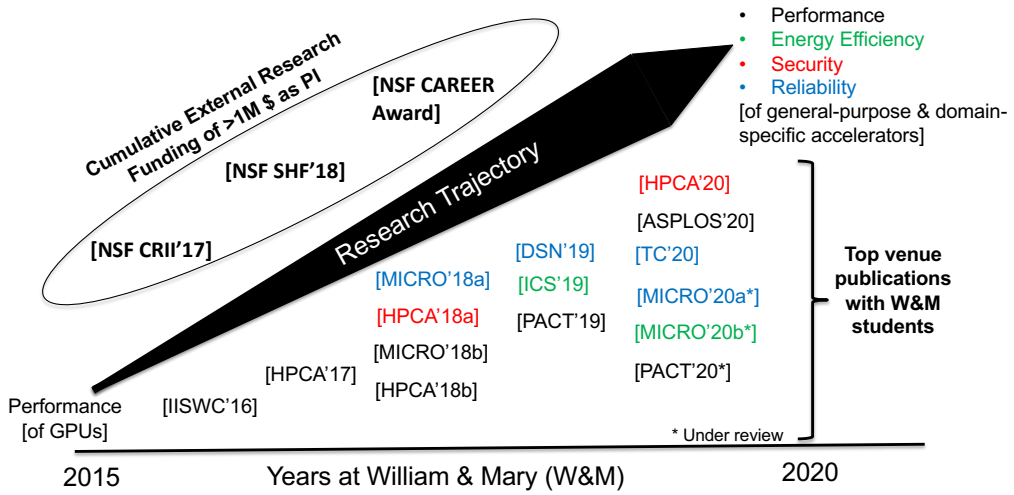

Figure 1: My research journey at William & Mary. Only major externally funded projects (including NSF CAREER Awarded Project) and top-tier publications with W&M students are shown.

I am currently advising four full-time W&M Ph.D. students, each leading one of the aforementioned research directions. Over the last five years, my students and I have worked closely and our projects have resulted in multiple peer-reviewed publications in top-tier venues such as ASPLOS, MICRO, HPCA, DSN, and ICS [3–17]. My first Ph.D. student, Haonan Wang, is expected to graduate in Aug 2020 and has accepted a tenure-track assistant professor position at San Jose State University (SJSU). Our group has also collaborated with industry, national labs, and other university researchers. For example, our work with AMD (one of the leaders in the GPU market) has led to publications [5, 18, 19] and US patent applications. Such collaborations have also enabled internships for students, including the ones at Intel and Pacific Northwest National Laboratory. I have also advised W&M undergraduate students (Charles Center and 1693 scholars) on their Honors thesis and summer projects, and have published papers with them [13, 14].

External Research Funding. Over the course of my tenure at W&M, I have brought external research funding of over 1 million USD as a principal investigator. The source of this funding is primarily the National Science Foundation (NSF) – including a five-year $450,000 CAREER Award (details in the CV). This funding has continuously supported the needs of my research group. Our group has also received equipment support from NVIDIA and internal W&M grants.

2. Research Projects and Directions

In this section, I briefly summarize my major research contributions that were made being at William & Mary.

2.1. Improving GPU Performance and System Throughput

My first research thrust is to improve system performance and fairness of GPUs. This project is funded primarily by the NSF CAREER Award. There are two dimensions to this project. First, we focused on identifying and exploiting new sources of bandwidth, which is a critical performance bottleneck in GPUs. To this end, my W&M Ph.D. student Mohamed Assem Ibrahim and I focused on efficiently unlocking another potential source of bandwidth in GPUs, which we call as remote-core bandwidth. As also observed by prior works, a fraction of data required by one GPU core (i.e., L1 misses) can also be found in the private caches of other GPU cores. If such sharing of data can be detected and accessed efficiently, access to shared caches (L2) and main memory can be avoided, thereby alleviating the L2/main-memory bottleneck in GPUs. However, efficient detection and utilization of this remote core bandwidth presents several challenges; and we focused on these challenges. Specifically, we address: a) which data is shared across cores, b) which cores have the shared data, and c) how we can get the data as soon as possible. Our extensive evaluation across a wide set of GPGPU applications show that we can achieve significant improvement in performance because of the additional bandwidth received from the cores. This work is published at PACT 2019 [5]. Currently, we are also exploring how private caches can efficiently be shared across cores in order to improve performance further. This work is recently accepted at PACT 2020 [20] and another work is currently under-submission [19]. These works are in collaboration with AMD Researchers who provided useful industry perspective to the project.

Second, we focused on how multiple applications can share GPUs efficiently. To address this problem, we proposed new application-aware thread-level parallelism (TLP) management techniques such that all co-scheduled applications can make good and judicious use of all the shared resources. Our key idea is to turn the fairness problem into an optimization problem with the goal of maximizing a key metric that my W&M students (Haonan Wang, Fan Luo, Mohamed Assem Ibrahim) came up with. By tuning the parallelism-levels for optimizing this metric at runtime, we were able to improve the system throughput and fairness by 20% and 2$\times$, respectively, over a baseline where each application executes with a TLP configuration that provides the best performance when it executes alone. This work is published in HPCA 2018 [10].

2.2. Optimizations for Energy Efficiency and Data Movement in GPUs

My second research thrust is to reduce data movement across the GPU memory hierarchy. This project is funded by the NSF CRII and CAREER awards. Data movement is one of the most significant contributors to overall energy consumption and hence should be reduced. To this end, my student Haonan Wang and I considered load data value prediction and approximation techniques. We showed that the existing approximate value predictors are not optimal in improving the prediction accuracy as they do not consider

memory request order for value prediction. As a result, the overall performance and data movement reduction benefits are capped as it is necessary to limit the percentage of predicted values (i.e., prediction coverage) for an acceptable value of application-level error. To this end, we proposed a new Address-Stride Assisted Approximate Value Predictor (ASAP) for GPUs that explicitly considers the memory addresses and their request order information so as to provide high-value prediction accuracy. We evaluated ASAP on a diverse set of approximable GPGPU applications. The results showed that ASAP can significantly improve the value prediction accuracy over the previously proposed mechanisms at the same coverage, or achieve higher coverage (leading to higher performance/energy improvements) under a fixed error threshold. This work is published at ICS 2019 [7].

In another work, we focused on the idea of approximation at the memory-level. Memory system energy is a major component of the total energy consumption of GPUs, which are becoming an integral part of almost all computing systems. DRAM dynamic energy is highly dependent on how well row buffers are utilized. Our results show that several GPGPU do not reuse the data fetched in row buffers effectively leading to high energy costs. In this context, we developed value approximation techniques that can avoid accessing row buffers periodically while improving energy efficiency. This work is published at DSN 2019 [6].

2.3. Improving GPU Security and Reliability

My third research thrust is to develop techniques that can improve GPU security and reliability at low performance and energy cost. This project is funded by the NSF CORE grant. As GPUs are becoming default accelerators in many domains, including accelerating security-related tasks, its security is becoming important. In our HPCA 2018 paper [11], my W&M Ph.D. student Gurunath Kadam and I focused on a popular GPU attack that reveals vulnerability in one of the important memory bandwidth optimization technique – intra-warp coalescing. This attack steals AES cryptographic private keys by exploiting the correlation between the number of coalesced accesses and execution time. To this end, we proposed a series of defense mechanisms to alleviate such timing attacks by carefully randomizing the coalescing logic, thereby trading-off performance for improved security. We found that the combination of our security mechanisms offers 24- to 961-times improvement in the security against the correlation timing attacks with 5 to 28% performance degradation. We have extended this work to retain security benefits while reducing this degradation to less than 1% for different scenarios. The key idea behind this new approach is to make the correlation close to zero by shaping the memory requests such that always a constant number of requests are issued. This new work is recently published at HPCA 2020 [4]. Both these works are in collaboration with Danfeng Zhang (Assistant Professor at Penn State) who provided useful input from the security-side.

In collaboration with Prof. Evgenia Smirni at W&M, I am also focusing on GPU reliability. In our MICRO 2018 [9] and TC 2020 [21] papers, we focused on pruning fault sites in order to gain a comprehensive understanding of the effect of faults on the applications. Specifically, we discussed how GPU application-specific properties can be used to achieve highly accurate estimates of applications' resilience. Our mechanisms provided orders of magnitude reduction in analysis time potentially leading to faster protection solutions. In another work (currently under submission [22]), we extended this line of work to also consider multiple inputs. We are also working on low-overhead protection mechanisms for GPUs (currently under submission [15, 18]).

2.4. Efficient Domain-specific Computations on Automata Processors and GPUs

Along with my student Hongyuan Liu, we focused on the execution of domain-specific computation models like non-deterministic finite automata (NFAs). First, we show that they can be accelerated on Automata Processor (AP), however, it is unable to handle larger automata programs without repeated reconfiguration and re-execution. To achieve high throughput, our MICRO 2018 paper [8] proposes for the first time architectural support for AP to efficiently execute large-scale applications. We find that a large number of existing and new NFA based applications have states that are never enabled but are still configured on the AP chips leading to their underutilization. With the help of careful characterization and profiling-based mechanisms, we predict which states are never enabled and hence need not be configured on AP. Furthermore, we develop SparseAP, a new execution mode for AP to efficiently handle the mispredicted NFA states. Our detailed simulations across 26 applications from various domains show that our newly proposed execution model for AP can obtain 2.1$\times$ geometric mean speedup (up to 47$\times$) over the baseline AP execution.

Second, we show that NFAs perform poorly on GPUs. In our recent ASPLOS 2020 paper [3], we show that NFA acceleration on GPUs has two challenges. First, NFA computation involves a significant volume of irregular off-chip memory accesses to retrieve the matchset (containing which alphabets are acceptable for a given NFA state) and the NFA topology information. This is because this information is stored very inefficiently in a huge transition table in the off-chip global memory. Second, NFA computation when mapped to a highly threaded environment of GPUs leads to severe under-utilization – not all threads are active at all times due to the fact that its corresponding NFA states demonstrate variable active ("hot") and in-active ("cold") properties. To address these two problems we first analyze the edges and the state matchset information and observe that the transition table is highly sparse and contains redundant entries making it infeasible to keep it on-chip. To this end, we propose a new way to store and access matchset and topology information such that it can be accessed from on-chip resources as much as possible. Second, by understanding the active behavior of NFA states, we find the opportunity of improving the core utilization by intelligently mapping only hot states to dedicated threads while other not so active states are processed on-demand. Overall, across 16 NFA applications, the best of our schemes improve the throughput on average by 26.5$\times$ over prior work.

We hope that our work will make GPUs efficient in executing domain-specific computations (e.g., Automata). To enable this quickly, we have also open-sourced all our code (received all possible validation badges from ASPLOS 2020 Artifact Evaluation Committee) and available at: `https://github.com/insight-cal/gpunfa-artifact`. Both these works are in collaboration with Sreepathi Pai (Assistant Professor at Rochester) who provided useful input from the compiler and software-side.

3. Future Plans

I believe there are pros and cons of both general-purpose and domain-specific accelerators. My long-term future research will be focusing on enhancing and extracting goodness of both these types to develop more capable, reliable, and secure accelerators and heterogeneous platforms. I will continue to seek external funding to support my research agenda. In the near-term, I will focus on the following two research themes – they have already led to full length grant proposals (under submission) to different funding agencies.

3.1. A Holistic Approach to Bandwidth Management in GPUs

In order to keep GPUs relevant in the coming years, it is not only important to improve GPU performance and energy-efficiency but also keep them protected against any potential information leakage. To this end, my future plan contains three main research components. First, we will focus on improving existing on-chip bandwidth utilization by rethinking the on-chip memory hierarchy. We will consider a wide-range of on-chip bandwidth sources (based on spatial, temporal, and value locality) and develop novel techniques to manage those sources. Second, we will develop new communication management techniques to connect various on-chip bandwidth sources in an efficient way. We will design novel interconnect designs that are area and power-efficient while providing high throughput. Third, we will develop low-overhead solutions to reduce or stop sensitive information leakage through existing and new bandwidth optimizations. Particularly, we will focus on analyzing security-bandwidth trade-offs and developing shared resource management techniques to concurrently improve GPU security and efficiency. We will be also considering these issues in the context of other accelerators for several domains (e.g., for machine learning, automata processing, and bioinformatics).

3.2. Efficient Execution of Domain-specific Computations on General Purpose Accelerators

Along the lines of System-on-chip (SoC), hosting several domain-specific architectures in one unified system, while choosing which ones to use for a certain application will likely remain an open research problem. Instead of developing accelerator for each domain, my vision is to extract and combine important domain-specific computations and make necessary hardware/software changes to the general-purpose accelerator – i.e., develop a domain-aware GPU. To realize this vision, my plan is to focus on: a) finding and accelerating a common computing paradigm that captures the essence of important workloads (e.g., computation graphs, graph analytics), and b) empowering general purpose accelerators (e.g., GPUs) to execute such

common computing paradigms efficiently. Altogether, the benefit of such an approach is two-fold: a) better utilize the existing general-purpose von Neumann hardware and reuse its mature features (e.g., virtualization, software/debugging support), and b) reinforce the progress of domain-specific eco-system by providing programmers and architects a more predictable environment and reusable toolchain.

To conclude, I am able to form a solid computer architecture research program at W&M and planning to leverage this foundation for further flourishing my independent research group.

References

My advisees are marked with *. Underlined Students are from William & Mary.

[1] "Computer science: As Moore's Law slows down, the Insight Architecture Lab accelerates." https://www.wm.edu/news/stories/2020/computer-science-as-moores-law-slows-down,-the-insight-architecture-lab-accelerates.php.

[2] "Using a CAREER award to advance the Insight Computer Architecture Lab." https://www.wm.edu/news/stories/2018/using-a-career-award-to-advance-the-insight-computer-architecture-lab.php.

[3] Hongyuan Liu*, S. Pai, and A. Jog, "Why GPUs are Slow at Executing NFAs and How to Make them Faster," in *the Proceedings of 23rd International Conference on Architectural Support for Programming Languages and Operating Systems **(ASPLOS)**, Lausanne, Switzerland*, pp. 251–265, *Acceptance rate: 86/479 ∼18%*, March 2020.

[4] Gurunath Kadam*, D. Zhang, and A. Jog, "BCoal: Bucketing-based Memory Coalescing in GPUs," in *the Proceedings of 26th International Symposium on High Performance Computer Architecture **(HPCA)**, San Diego, CA*, pp. 570–581, *Acceptance rate: 48/248 ∼19%*, Feb 2020.

[5] Mohamed Assem Ibrahim*, Hongyuan Liu*, O. Kayiran, and A. Jog, "Enhancing Bandwidth Utilization via Efficient Inter-core Communication in GPUs," in *the Proceedings of 28th International Conference on Parallel Architectures and Compilation Techniques **(PACT)**, Seattle, WA*, pp. 258–271, *Acceptance rate: 26/126 ∼21%*, Sept 2019.

[6] Haonan Wang* and A. Jog, "Exploiting Latency and Error Tolerance of GPGPU Applications for an Energy-efficient DRAM," in *the Proceedings of 49th International Conference on Dependable Systems and Networks **(DSN)**, Portland, OR*, pp. 362–374, *Acceptance rate: 54/252 ∼21%*, June 2019.

[7] Haonan Wang*, Mohamed Assem Ibrahim*, S. Mittal, and A. Jog, "Address-Stride Assisted Approximate Value Prediction in GPUs," in *the Proceedings of 33rd International Conference on Super Computing **(ICS)**, Phoenix, Arizona*, pp. 184–194, *Acceptance rate: 45/193 ∼23%*, June 2019.

[8] Hongyuan Liu*, Mohamed Assem Ibrahim*, O. Kayiran, S. Pai, and A. Jog, "Architectural Support for Efficient Large-Scale Automata Processing," in *the Proceedings of 51st International Conference on Micro-Architecture **(MICRO)**, Fukuoka, Japan*, pp. 908–920, *Acceptance rate: 74/351 ∼21%*, Oct 2018.

[9] Bin Nie, Lishan Yang, E. Smirni, and A. Jog, "Fault Site Pruning for Practical Reliability Analysis of GPGPU Applications," in *the Proceedings of 51st International Conference on Micro-Architecture **(MICRO)**, Fukuoka, Japan*, pp. 749–761, *Acceptance rate: 74/351 ∼21%*, Oct 2018.

[10] Haonan Wang*, Fan Luo*, Mohamed Assem Ibrahim*, O. Kayiran, and A. Jog, "Efficient and Fair Multi-programming in GPUs via Effective Bandwidth Management," in *the Proceedings of 24th International Symposium on High Performance Computer Architecture **(HPCA)**, Vienna, Austria*, pp. 247–258, *Acceptance rate: 54/260 ∼20%*, Feb 2018.

[11] Gurunath Kadam*, D. Zhang, and A. Jog, "RCoal: Mitigating GPU Timing Attack via Subwarp-based Randomized Coalescing Techniques," in *the Proceedings of 24th International Symposium on High Performance Computer Architecture **(HPCA)**, Vienna, Austria*, pp. 156–167, *Acceptance rate: 54/260 ∼20%*, Feb 2018.

[12] X. Tang, A. Pattnaik, H. Jiang, O. Kayiran, A. Jog, S. Pai, Mohamed Assem Ibrahim*, M. Kandemir, and C. Das, "Controlled Kernel Launch for dynamic parallelism in GPUs," in *the Proceedings of 23rd International Symposium on High Performance Computer Architecture **(HPCA)**, Austin, TX*, pp. 649–660, *Acceptance rate: 50/224 ∼22%*, Feb 2017.

[13] Robert Risque* and A. Jog, "Characterization of Quantum Workloads on SIMD Architectures," in *the Proceedings of International Symposium on Workload Characterization **(IISWC)**, Providence, RI*, pp. 1–9, *Acceptance rate: 21/71 ∼29%*, Oct 2016.

[14] H. Zhao, Colin Weinshenker*, Mohamed Assem Ibrahim*, A. Jog, and J. Zhao, "Layer-wise performance bottleneck analysis of deep neural networks," *Architectures for Intelligent Machines **(AIM)** Workshop in conjunction with International Conference on Parallel Architectures and Compilation Techniques (PACT), Hillsboro, OR*, Sep 2017.

[15] Gurunath Kadam*, E. Smirni, and A. Jog, "Title is currently not public.," in *Submission to HPCA*, 2021.

[16] S. Mittal, R. Bishnoi, F. Oboril, Haonan Wang*, M. Tahoori, A. Jog, and J. Vetter, "Architecting SOT-RAM Based GPU Register File," in *the Proceedings of IEEE Computer Society Annual Symposium on VLSI **(ISVLSI)**, Bochum, Germany*, pp. 38–44, *Acceptance rate: 67/212 ∼32%*, July 2017.

[17] S. Mittal, Haonan Wang*, A. Jog, and J. Vetter, "Design and Analysis of Soft-Error Resilience Mechanisms for GPU Register File," in *the Proceedings of 30th International Conference on VLSI design and 16th International Conference on Embedded Systems **(VLSID)**, Hyderabad, India*, pp. 409–414, *Acceptance rate: 71/292 ∼24%*, Jan 2017.

[18] Lishan Yang, Bin Nie, A. Jog, and E. Smirni, "Title is currently not public.," in *Submission to IISWC*, 2020.

[19] Mohamed Assem Ibrahim*, O. Kayiran, Y. Eckert, G. H. Loh, and A. Jog, "Title is currently not public.," in *Submission to HPCA*, 2021.

[20] Mohamed Assem Ibrahim*, O. Kayiran, Y. Eckert, G. H. Loh, and A. Jog, "Analyzing and Leveraging Shared L1 Caches in GPUs," in *the Proceedings of 29th International Conference on Parallel Architectures and Compilation Techniques **(PACT)**, Atlanta, GA*, pp. TBD–TBD, *Acceptance rate: 35/135 ∼25%*, Oct 2020.

[21] Lishan Yang, Bin Nie, A. Jog, and E. Smirni, "Practical resilience analysis of gpgpu applications in the presence of single- and multi-bit faults," *IEEE Transactions on Computers, **(TC)***, pp. 1–14, 2020.

[22] Lishan Yang, Bin Nie, A. Jog, and E. Smirni, "Title is currently not public.," in *Submission to SIGMETRICS*, 2021.