Research Statement

January 27, 2021

Roger Allan Pearce, Ph.D.

My research over the past decade at LLNL has focused on co-design research to enable large scale data analytics. From algorithms and analytics, to High Performance Computing (HPC) frameworks for irregular asynchronous parallelism, to system software for mapping data-structures to persistent memories, to persistent memory hardware evaluations, my research contributions have spanned the full vertical system stack required to enable data analytics at scale. In tandem with my published research efforts, I have produced mission impact in related applied research projects, and it is this applied experience that I draw on to shape the research needs, goals, and directions for my research group. I believe this gives me a unique perspective for how future HPC technologies can be leveraged for data-science applications. I have conducted this research as the PI, Co-PI, and Co-Investigator of multiple internally and externally funded R&D projects, supervising three postdoctoral researchers and over twenty students. It is this perspective, along with mission drivers, that motivates me to design the next-generation data-science platform for HPC systems.

My team has pushed the bleeding edge of HPC graph analytics on the two main HPC graph benchmarks and competitions. Our contributions on the *DARPA Graph Challenge* have received three *Champion* awards (2017, 2018, 2019) for the largest *triangle counting* and *k-truss* calculations in published literature. Similarly, our contributions to the *Graph500* demonstrated the largest *Breadth-First Search* in published literature. Along this research journey, my work has been published in five high-impact *IEEE/ACM Supercomputing* papers, and I have been invited to talk at multiple high-impact venues, including *GraphFest* at the National Security Agency (NSA). These contributions required co-design research at both the algorithms and systems level – it is this intersection of algorithms and systems where I see next-generation data-science platform innovation occurring.

Graph Algorithms & Analytics: The foundation of my team's HPC graph algorithms and analytics is an asynchronous vertex-centric runtime I developed as a Ph.D. student and Lawrence Scholar for both shared [22] and distributed [20, 18] memory. My approach forces algorithm designers to "think like a vertex" (a phrase coined by Google's Pregel team), and to shun traditional bulk-synchronous programming (BSP) for a fully asynchronous one-sided programming model. As a Ph.D. student, I demonstrated this approach using many of the traditional textbook graph algorithms. More recently, my team has blazed a path through more impactful algorithms and analytics:

Labeled Graph Pattern Matching: Searching for small templates or motifs in a large labeled graph is an important task in many network science use cases, including *cyber security* and *social media analysis*. My team has developed *PruneJuice* [13, 10, 1], a set of graph pruning algorithms that recursively remove vertices and edges that can be proved to not be a member of an input template, leaving only subgraph structures belonging to the template. This work has been demonstrated to scale to the largest graph datasets currently in literature and can search for templates with complex high-order structure (e.g., nested cycles) – again, pushing the sate of the art by multiple orders of magnitude. For this project I supervised *Dr. Tahsin Reza* who was a Ph.D. student at the time at the University of British Columbia, and I joined his dissertation committee. Dr. Reza is now a postdoctoral researcher in my group investigating interactive graph pattern matching and applying motif search for a graph embedding application as part of a *graph learning* research effort.

Triangle counting & K-Truss: Counting the triangles (3-cycles), in a graph is an important building block for network analysis used by many calculations including *clustering coefficient*. Intuitively, this is a measure of *how many of my friends are also friends*. *K-truss* is decomposition of a graph where edges are recursively removed if they fail to be a member of $k-2$ triangles. K-truss was developed by Jonathan Cohen at NSA more than a decade ago, and has captured the attention of the HPC Graph community due to his high-impact in network science as method to decompose a graph into tightly cohesive subgraphs. Triangle counting and k-truss are benchmarks on the *DARPA Graph Challenge*, and my research has delivered the largest demonstrations of triangle counting [12, 3] and k-truss [9], by multiple orders of magnitude. This work required innovation at both

the algorithms and HPC systems level, and achieved a search of over one quadrillion triangle queries using more than one million processors on an IBM BG/Q system at LLNL [3].

Additional Graph Algorithms Contributions: In keeping with the theme of large graph processing on HPC, my team also developed algorithms for Vertex Eccentricity [7], Breadth-First Search for Distributed-GPU architectures [8], and Dynamic (on-line) Graphs algorithms [6, 15]. Each of these is built on my asynchronous graph analytics runtime, and have been demonstrated at the largest scales known in the research literature.

HPC systems research for irregular communication and parallelism: Perhaps surprising to many of the organizers of the *Graph500* and *GraphChallenge*, they both turned out to be algorithmic and systems challenges in the beginning – not a hardware benchmark as originally described. During the first few cycles of the Graph500, multiple orders of magnitude improvement were achieved on the same hardware (IBM BG/P at the time). The GraphChallenge is still seeing algorithmic improvements; however, the Graph500 has been well optimized for traditional CPU clusters.

Central to all distributed data structures and algorithms is the need for efficient communication, traditionally managed in HPC by MPI. However, MPI by itself is ill suited for sparse and irregular communication patterns common to many data processing tasks. My team has built a communication framework specifically for graph and irregular algorithms and have demonstrated excellent scalability [5]. The techniques included in this work are aimed at accelerating the transport of tiny messages sparsely sent between distributed processors, and is one of the key innovations enabling the graph algorithms previously discussed. We have abstracted these techniques into a reusable C++ library and its currently being used by new projects investigating large scale approximate nearest-neighbor search (e.g., K-nearest neighbor graphs) and also an astronomy application.

System software for mapping data structures to persistent memories: It is my view that persistent memory (NVM, NVRAM) will fundamentally change how computational resources are managed for data-hungry applications. Towards this goal, my prior research has shown that NVRAM is well suited for memory-hungry applications like Graph analytics [22, 20]. The largest experiment on the *Graph500*, at the time of this writing, is my team's work using the NVRAM devices in each compute node of *Sierra* at LLNL to store 70-trillion edges that were too large for main-memory (DRAM).

System software remains the largest impediment to achieving this vision of using persistent memory for large scale data analytics. I have investigated kernel-level [16] and user-level [4] memory map accelerators for data science applications. Memory mapping (mmap) is a mechanism by which the operating system (e.g., Linux/Unix) virtually maps a file's contents into the address space of a process, and reads/writes pages to the file using the page fault mechanism. Such mmap mappings can be larger than the physical main-memory of the system, allowing applications to address datasets larger than physical main-memory (often referred to as out-of-core or external memory).

Under my mentorship, my postdoc Keita Iwabuchi has developed *Metall* [2], a persistent memory allocator designed to provide developers with an API to allocate custom C++ data structures in both block-storage and byte-addressable persistent memories. *Metall* relies on a file-backed mmap mechanism to map a file in a filesystem into the virtual memory of an application, allowing the application to access the mapped region as if it were regular memory. Our approach allows an application to transparently create, detach, and reattach to persistent data structures without heavyweight serialization. Traditionally serialization techniques continue to have their palace for portable data archive reasons; our work is aimed at enabling lightweight manipulation of persistently stored data structures and not full replacement for portable serialization.

Exploratory Data Analytics (EDA) at HPC-Scale (LLNL R&D FY21-FY23, $1.5M)

Exploratory Data Analytics (EDA) is often the first step used by data scientists when faced with a new dataset or analytic task, and it specifically aids in hypothesis generation and evaluation. The de facto standard among a large percentage of data scientists is *Jupyter* notebooks (i.e., interactive Python), in which relatively small

datasets can be manipulated using popular tools such as *NumPy*, *SciPy*, *Pandas*, or *NetworkX* on a desktop or laptop environment, possibly connected with a small compute cluster backend via *Apache Spark*. This paradigm is limited by poor performance and scalability, yet many scientific and security data analysis tasks demand algorithms that require many unstructured analysis phases over significant data scales. **I am the PI of this new-start three year R&D project that cuts across the full HPC algorithms and system stack to design the next-generation exploratory data analytics (EDA) platform for HPC.**

Expected Impact on Cyber (Physical) Security

My future looking research agenda has been shaped by close hands-on interactions with subject matter experts in Cyber Physical Security. I have had direct interactions with practitioners at LLNL/DOE and external USG agencies. As a computer science researcher motivated to investigate HPC tools for national security data science problems, my standard set of questions for domain-experts include: *What capabilities do you lack? Can you process the data you collect in its entirety? How could HPC impact your mission?*

Distributed instrumentation has permeated almost every modern complex engineered system. From smart meters on the smart grid, to host-based endpoint sensors on an enterprise computer network, to distributed sensors in a large scale scientific instrument installation, the volume of interconnected instrumentation data generated today is astounding. Quoting the 2015 ASCR report on cyber security, "*one way to capture interdependencies between security-relevant events is through a graph*" [17].

Protecting and optimizing the power grid has been identified as a *grand challenge in cyber physical networks.* Reported in a 2012 NITRD report, "Each year $\$18 - 20$ billion is lost by under-optimizing energy networks, and an estimated $\$49$ billion lost in annual outages" [21]. Additionally, a lack of tools to process the collected data in its entirety has also been identified: "We also need scalable tools and methods for understanding, designing, and conducting tradeoffs between various characteristics of the network, including suitability for use, economic considerations, and performance" [21].

Aggregate events and dynamics on complex engineered systems is of particular strategic importance [21]. Temporal graph analytics can enable insight into the dynamics of cascading failures or the propagation of malicious actors attempting to disrupt or infect a network. It is not a lack of high-fidelity data that impedes these forms of analysis from being routine, it is the lack of scalable tools that prevents all but the heroic analysts from performing enterprise-wide temporal analysis.

The goal of graph-enabled analytics on host-based data is to detect distributed-coordinated and *low-and-slow* attacks. Distributed-coordinated attacks require a wide view across the full enterprise network, a view not possible from individual siloed sensor's independent view. Similarly, *low-and-slow* attacks require deep longitudinal data stores that are infeasible without significant data storage and indexing.

Detecting *Advanced Persistent Threats (APT)* remains a key challenge problem for cyber physical security. "Unlike automated broad-range attacks, APTs are human-driven infiltrations, perpetrated over long periods of time, customized for the targeted organization after some intelligence analysis, possibly on open sources, and can even leverage unknown exploits to infiltrate vulnerable systems" [14]. Temporal graph-based analysis techniques can be used to detect common attack profiles and to define threat profiles for suspicious activity. [11, 19]

I will continue working closely with cyber security experts and practitioners to ensure my research products are positioned to make a measurable impact on national-interest cyber physical security challenges.

References

[1] Tahsin Reza, Hassan Halawa, Matei Ripeanu, Geoffrey Sanders, and **Roger Pearce**. Scalable pattern matching in metadata graphs via constraint checking. *ACM Transactions on Parallel Computing (TOPC)*, 8(1):1–45, 2021.

[2] Keita Iwabuchi, Lance Lebanoff, Maya Gokhale, and **Roger Pearce**. Metall: A persistent memory allocator enabling graph processing. In *2019 IEEE/ACM 9th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*, pages 39–44. IEEE, 2019.

[3] **Roger Pearce**, Trevor Steil, Benjamin Priest, and Geoffrey Sanders. One quadrillion triangles queried on one million processors. In *2019 IEEE High Performance extreme Computing Conference (HPEC)*. IEEE, 2019.

[4] Ivy Peng, Marty McFadden, Eric Green, Keita Iwabuchi*, Kai Wu, Dong Li, **Roger Pearce**, and Maya Gokhale. Umap: Enabling application-driven optimizations for page management. In *2019 IEEE/ACM Workshop on Memory Centric High Performance Computing (MCHPC)*, pages 71–78. IEEE, 2019.

[5] Benjamin Priest, Trevor Steil, Geffrey Sanders, and **Roger Pearce**. You've got mail: Building missing asynchronous communication primitives. In *GrAPL 2019: Workshop on Graphs, Architectures, Programming, and Learning*, 2019.

[6] Scott Sallinen, **Roger Pearce**, and Matei Ripeanu. Incremental graph processing for on-line analytics. In *IEEE IPDPS*, 2019.

[7] Keita Iwabuchi, Geoffrey Sanders, Keith Henderson, and **Roger Pearce**. Computing exact vertex eccentricity on massive-scale distributed graphs. In *IEEE CLUSTER*, 2018.

[8] Yuechao Pan, **Roger Pearce**, and John D Owens. Scalable breadth-first search on a gpu cluster. In *IEEE IPDPS*, 2018.

[9] **Roger Pearce** and Geoffrey Sanders. K-truss decomposition for scale-free graphs at scale in distributed memory. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2018.

[10] Tahsin Reza, Matei Ripeanu, Nicolas Tripoul, Geoffrey Sanders, and **Roger Pearce**. Prunejuice: pruning trillion-edge graphs to a precise pattern-matching solution. In *Supercomputing*, 2018.

[11] Timo Schindler. Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats. *arXiv preprint arXiv:1802.00259*, 2018.

[12] **Roger Pearce**. Triangle counting for scale-free graphs at scale in distributed memory. In *IEEE HPEC*, 2017.

[13] Tahsin Reza, Christine Klymko, Matei Ripeanu, Geoffrey Sanders, and **Roger Pearce**. Towards practical and robust labeled pattern matching in trillion-edge graphs. In *IEEE CLUSTER*, 2017.

[14] Mirco Marchetti, Fabio Pierazzi, Michele Colajanni, and Alessandro Guido. Analysis of high volumes of network traffic for advanced persistent threat detection. *Computer Networks*, 109:127 – 141, 2016. Traffic and Performance in the Big Data Era.

[15] Scott Sallinen, Keita Iwabuchi, Suraj Poudel, Maya Gokhale, Matei Ripeanu, and **Roger Pearce**. Graph colouring as a challenge problem for dynamic graph processing on distributed systems. In *ACM/IEEE Supercomputing*, 2016.

[16] Brian Van Essen, Henry Hsieh, Sasha Ames, **Roger Pearce**, and Maya Gokhale. Di-mmapâĂŤa scalable memory-map runtime for out-of-core data-intensive applications. *Cluster Computing*, 18(1), 2015.

[17] Sean Piesert. Ascr cybersecurity for scientific computing integrity – research pathways and ideas workshop. June 2015.

[18] **Roger Pearce**, Maya Gokhale, and Nancy M Amato. Faster parallel traversal of scale free graphs at extreme scale with vertex delegates. In *ACM/IEEE Supercomputing*, 2014.

[19] John R Johnson and Emilie A Hogan. A graph analytic metric for mitigating advanced persistent threat. In *2013 IEEE International Conference on Intelligence and Security Informatics*, pages 129–133. IEEE, 2013.

[20] **Roger Pearce**, Maya Gokhale, and Nancy M Amato. Scaling techniques for massive scale-free graphs in distributed (external) memory. In *IEEE IPDPS*, 2013.

[21] NITRD LSN workshop report on complex engineered networks. September 2012.

[22] **Roger Pearce**, Maya Gokhale, and Nancy M Amato. Multithreaded asynchronous graph traversal for in-memory and semi-external memory. In *ACM/IEEE Supercomputing*, 2010.