Research Statement

January 20, 2021

Roger Allan Pearce, Ph.D.

The volume of data currently generated both by science and security applications and by the modern internet-connected human experience has surpassed our ability to process and understand at adequate levels of fidelity. *Graph analytics*, algorithms and methods used to understand the often complex relationship amongst data, are powerful tools used in a variety of science and security applications that is particularly challenging at scale on large distributed systems. While often considered in security or intelligence application contexts, graph analytics are used in diverse scientific domains including *systems biology*, *genomics*, *machine learning*, *social sciences* and *chemistry*. In engineering applications, graphs are used to model *wide area networks*, *power grids*, *utility distribution*, and the increasingly complex relationship between *cyber-physical systems*. When deep historical or longitudinal analysis is required, the volume of data often requires heavy triage or filtering that can impede deep analysis. The promise of using High Performance Computing (HPC) for such analysis is that a unified picture of a large distributed dataset is possible; however, tools to tackle enterprise-level datasets are still in research. While the so-called Cloud Computing environment has dominated the space of *Big Data*, graph analytics remain a key computation that is challenging "in the cloud", which is due in large part to the interdependent nature of graph computations. Many real-world graph datasets are difficult, if not impossible, to partition into embarrassingly parallel work units fit for cloud computing, which is where the promise of HPC and Exascale Computing arises: Graph analysis is the data-science problem that *requires* HPC. Graph analysis demands a high-speed, low-latency interconnect to facilitate tightly coupled parallel computations over a distributed dataset – just like traditional HPC scientific computing.

My research over the past decade at LLNL has focused almost entirely on co-design research for large scale data analytics. From algorithms and analytics [1, 3, 5, 8, 12, 11, 9, 10, 16], to HPC frameworks for expressing asynchronous parallelism [7, 18, 20], to system software for mapping data-structures to persistent memories [4, 6], to hardware evaluations [2, 14], my research contributions have spanned the full vertical system stack required to enable data analytics at scale. In tandem with these published research efforts, I have produced mission impact in related applied research areas that cannot be published, and it is this applied experience that I draw on to shape the research goals and directions for my teams. I believe this gives me a unique perspective for how future HPC technologies can be leveraged for data-science applications. **It is this perspective, along with mission drivers, that motivates me to design the next-generation data-science platform.**

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. My team has previously demonstrated HPC solutions on a variety of graph algorithms at scales 2-3 orders of magnitude beyond competing approaches, leading in HPC community benchmarks such as the Graph500 and DARPA Graph Challenge. However, no disruptive impact will be achieved if the average data scientist cannot access HPC solutions via an interactive environment such as a Python module. Enabling data scientists to compose complex analytic tasks, transform data, interact with summary reports from within the safety and familiarity of the Python ecosystem, all while the true data crunching tasks are performed on a distributed system, is absolutely crucial to accomplish many of today's EDA tasks related to mission applications.

A vertically integrated HPC solution requires R&D at multiple levels of the system stack from a team dedicated to ensuring the stack is sufficiently integrated. Viewed top down, from the user-interface to the HPC systems software, my three primary research thrusts are summarized in paragraphs (a), (b), and (c).

(a) Composition and expression of complex graph analytic workflows via interactive user-interface (UI): Analysts accustomed to working with data fully resident on their workstation will be asked to operate on distributed data so large that their workstation can only view small extractions and summary statistics. My team is designing a client-server architecture that preserves an interactive data exploration environment on the analyst's client, while dispatching data analysis tasks to the distributed system. The client interface will provide a diverse set of composable high-level operations. Its development will be guided by direct interactions with data scientists at LLNL and external partners, working on mission-relevant problems. Research questions to be investigated include: *What level of abstraction provides the best UI experience in terms of composing and customizing complex graph analytic workflows? How to optimally express the often-complex relationship between graph analytics and structured data frames?*

(b) Distributed algorithms and data structures for hybrid graph and relational data: My recent experience working with LLNL and external USG domain experts revealed the arduous challenges in transforming a real, semantically rich relational dataset into a form capable of being ingested into distributed graph algorithms. This process is commonly called Extract, Transform, Load (ETL) and is a significant challenge at HPC scales that has largely been ignored by the broader HPC research community. Because I aim to support hybrid graph and relational data, where rich metadata needs to be ingested alongside the graph topological information, I intend to develop technologies to seamlessly ingest and transform datasets from common existing data archive technologies, bridging HPC with databases and frames.

Research questions to be investigated towards graph-enabled data analytics include: *What data structures best combine graph topological information and data frame metadata? What are the most efficient ways to tie graph topological algorithms with operations on vertex and edge metadata for joint analytics? What communication abstractions can support both future HPC systems on the road to Exascale and commodity cloud systems?*

(c) System architecture and software for emerging persistent memory infused HPC systems: Emerging on many new distributed systems is node-local and rack-local persistent memory, currently in the form of PCIe attached NVRAM and, soon, byte-addressable memory attached NVDIMMs. This fast-growing capacity, extending the reach of expensive and power-hungry main-memory DRAM, is ideal for staging data between analysts' interactive operations and storing incremental state, should they choose to role back a computation to a previous state (a common desire in EDA). My team's prior research has shown that emerging persistent memory devices are well-suited for data intensive computing tasks including graphs, and my team has developed memory allocators and runtimes that allow applications to transparently operate out of persistent memory. In support of the project's goal of providing an interactive client experience, this project is developing new techniques to provide snapshotting, versioning, and staging of persistently allocated data structures. HPC has traditionally treated global storage as a centralized POSIX file system requiring expensive hardware to run distributed filesystems such as Lustre. However, such heavyweight file systems are not necessary for most data science tasks, as demonstrated by the use of HDFS in Spark and Hadoop clusters. I will investigate a lightweight object management system (not a filesystem) based on peer-to-peer persistent memory devices using content addressable storage techniques, common in the Cloud environment.

Expected Impact on Cyber (Physical) Security

My 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, economics consideration, 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" [15]. Temporal graph-based analysis techniques can be used to detect common attack profiles and to define threat profiles for suspicious activity. [13, 19]

I will continue working closely with cyber security experts and practitioners to ensure the research products of this project 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] Ivy Peng, **Roger Pearce**, and Maya Gokhale. On the memory underutilization: Exploring disaggregated memory on hpc systems. In *2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 183–190. IEEE, 2020.

[3] Tahsin Reza, Matei Ripeanu, Geoffrey Sanders, and **Roger Pearce**. Towards interactive pattern search in massive graphs. In *Proceedings of the 3rd Joint International Workshop on Graph Data Management Experiences; Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA'20, New York, NY, USA, 2020. Association for Computing Machinery.

[4] 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.

[5] **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.

[6] 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.

[7] 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.

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

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

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

[11] **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.

[12] 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.

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

[14] Manu Shantharam, Keita Iwabuchi, Pietro Cicotti, Laura Carrington, Maya Gokhale, and **Roger Pearce**. Performance evaluation of scale-free graph algorithms in low latency non-volatile memory. In *IEEE IPDPS*, 2017.

[15] 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.

[16] 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.

[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.