Co-design of streaming analysis for exascale cybersecurity
Area: Extreme Scale Data, Knowledge and Analytics

Jonathan Berry William Hart* Scott Hemmert Cynthia Phillips

Overview We posit that a future high-performance computing (HPC) system for scientific computing designed with intentional cybersecurity will require co-design of hardware, software, and algorithms for finding patterns or anomolies in infinite data streams. Real-time situational awareness in exascale systems will require several fundamental advances. First, the current stream of telemetry (system diagnostics) data, which is aimed at system resilience, should be augmented with information to aid cybersecurity. For example, any interaction between applications and the operating system should be recorded. Second, co-design of integrated computation and streaming engines should be performed to realize future HPC systems with integral cybersecurity. Third, streaming techniques should be developed to specifically support cyber situational awareness in exascale systems. These include both algorithms and theoretical models. A cybersecurity system may monitor the interaction between the parallel operating system, a dynamic mix of many independent programs simultaneously using dedicated subsets of processors, and data streaming from external equipment. Generally such security systems have limited memory and computing power relative to the input stream and therefore we must make decisions about what to compute and what to store.

We make the following assumptions:

- DOE and vendors have an opportunity to design the next extreme-scale systems for scientific computing with intentional cybersecurity.
- A cybersecurity system must monitor the HPC system all the time it is up. Therefore, we should consider algorithms for infinite streams.
- Streaming analysis can watch for the occurrence of specific complex patterns. The streaming algorithm should store as much partial-pattern information as possible to minimize the number of times it misses such cybersecurity-relevant patterns.
- The US investment in HPC hardware would be negated if scientists cannot trust the computational results. It may be worth investing a fraction of the cost of the HPC system to design hardware that better supports streaming-based cybersecurity.
- Even without special hardware, there is benefit to co-design the architecture, system software, and streaming algorithms. This co-design ensures that the streaming algorithm can protect against the most important vulnerabilities, and the system can deliver the most important data to the streaming algorithm.

Motivating Experience Our perspective is influenced by our recent work on both co-design of architecture and algorithms for near-future HPC systems, and modeling and processing infinite data streams.

Our team of architects and algorithm theorists produced a model for designing and analyzing algorithms to exploit the new memory features of the Trinity supercomputer. We designed a new sorting algorithm and showed with simulation that this new memory can lead to improved performance [1]. This work has earned Best Paper in the algorithms track at the IEEE IPDPS conference, May 2015.

We also introduced the first model and algorithm for infinite streams of relational data [3]. We were motivated by streaming cybersecurity data, such as network communication, which can be represented as a graph. Previous theoretical algorithms for graph models assumed finite streams of graph edges. Algorithms that pass over the data multiple times, writing and reading intermediate streams, can solve graph problems that are theoretically impossible with a single pass [5]. However, a cyber graph stream never ends while a network is up. Holding up a stream would drop information or create an insurmountable backlog.

*Proposed workshop attendee, {jberry,shemme,caphill,wehart}@sandia.gov, Sandia National Laboratories

We designed a model and an algorithm that requires global graph information with the cyber analyst in mind. We store as much as possible and process queries. We can effectively use the aggregate memory of an arbitrary number of processors to store edges. However, eventually and infinite stream will overwhelm any finite aggregate storage. Thus our algorithm gracefully removes edges according to an analyst predicate without dropping any edges. Our implementation of this algorithm identified opportunities to improve performance using heterogeneous hardware. Our algorithm currently handles only one input stream.

Required research The research required to realize the vision above includes co-designing the HPC hardware and software, and formulating algorithm questions to support intentional HPC cybersecurity. This involves determining which additional telemetry data would best support intentional cybersecurity, how those data are analyzed, and what infrastructure is necessary to support such collection and analysis.

Graph models of streaming data are interesting because they support a compact, discrete representation and can be used to model behavioral features associated with cyber threats.

Co-design of streaming algorithms with emerging architectural features will support the design of future exascale systems with inherent, scalable infrastructure for cyber situational awareness. Emerging architectural features such as many-core processors, near-memory [7], and hybrid memory cubes [9, 4] are affecting the design of future HPC already. We believe that these should also influence the co-design of built-in cyber telemetry. This long-term goal may involve not only hardware and algorithms, but data structure techniques such as write-optimized data structures (which allow data to float up and down as computation proceeds, minimizing block transfers from slower storage resources) [2]. Our objective will always be to store as much state as possible, while processing as much data as possible.

We will also conjecture potential usages of accelerators in this streaming model. Sandia's experience with Kokkos [6], the Kepler device [8] and graph algorithms has oriented us in the case of static data. Streaming presents a further challenge.

References

[1] Michael A. Bender, Jonathan Berry, Simon D. Hammond, K. Scott Hemmert, Samuel McCauley, Branden Moore, Benjamin Moseley, Cynthia A. Phillips, David Resnick, and Arun Rodrigues. Two-level main memory co-design: Multi-threaded algorithmic primitives, analysis, and simulation. In *Proc. 29th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Hyderabad, INDIA, May 2015.

[2] Michael A Bender, Martin Farach-Colton, Rob Johnson, Russell Kraner, Bradley C Kuszmaul, Dzejla Medjedovic, Pablo Montes, Pradeep Shetty, Richard P Spillane, and Erez Zadok. Don't thrash: how to cache your hash on flash. *Proceedings of the VLDB Endowment*, 5(11):1627–1637, 2012.

[3] Jonathan Berry, Matthew Oster, Cynthia A Phillips, Steven Plimpton, and Timothy M Shead. Maintaining connected components for infinite graph streams. In *Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pages 95–102. ACM, 2013.

[4] B Casper. Reinventing dram with the hybrid memory cube. blogs. intel. com/research/2011/09/hmc. php, 2010. *Research@ Intel*.

[5] Camil Demetrescu, Irene Finocchi, and Andrea Ribichini. Trading off space for passes in graph streaming problems. *ACM Transactions on Algorithms*, 6(1):6:1–6:17, December 2009.

[6] H Carter Edwards and Daniel Sunderland. Kokkos array performance-portable manycore programming model. In *Proceedings of the 2012 International Workshop on Programming Models and Applications for Multicores and Manycores*, pages 1–10. ACM, 2012.

[7] `http://www.hpcwire.com/2014/06/24/micron-intel-reveal-memory-slice-knights-landing/`.

[8] NVIDIA. Nvidia-kepler-gk110-architecture-whitepaper.

[9] J Thomas Pawlowski. Hybrid memory cube (HMC). In *Hotchips*, volume 23, pages 1–24, 2011.