A Software Toolkit for Accelerating Graph Algorithms

David Bader (PI), Oded Green (Georgia Tech); Viktor Prasanna, Rajgopal Kannan (USC)

Architectures Thrust: Hierarchical Identify Verify Exploit (HIVE)

Overview

Problem: Seamless acceleration of static and streaming graph analytics-based DoD applications.

Background

Current practice: Programmers are required to exploit low-level hardware primitives, which is time consuming and results in limited portability.

Innovation

Application of genetic algorithms with unique expertise in HW/SW co-design (modeling, analysis and optimization). Experience of large-scale parallel graph algorithms and data structures for dynamic graphs.

Partition-centric Programming Model

- Set of graph partitions on each node
- Stream out-edges: data to remote memory (partitions)
- Stream in-edges: use local data to compute
- Lock-free parallel computation and communication
- Message grouping: reduce traffic
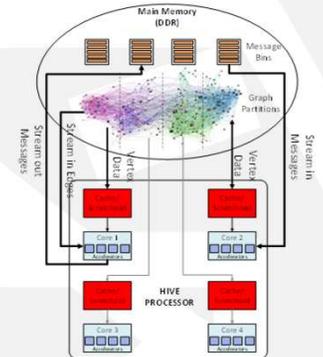
Hornet

- Dynamic graph data structure resembling CSR
- Generalized data structure with good cache performance
- High insertion & deletion rates
- Quickly identifies empty edge blocks
- Achieves good thread load balance
- Enables programmer to interact with graph building blocks instead of hardware primitives.
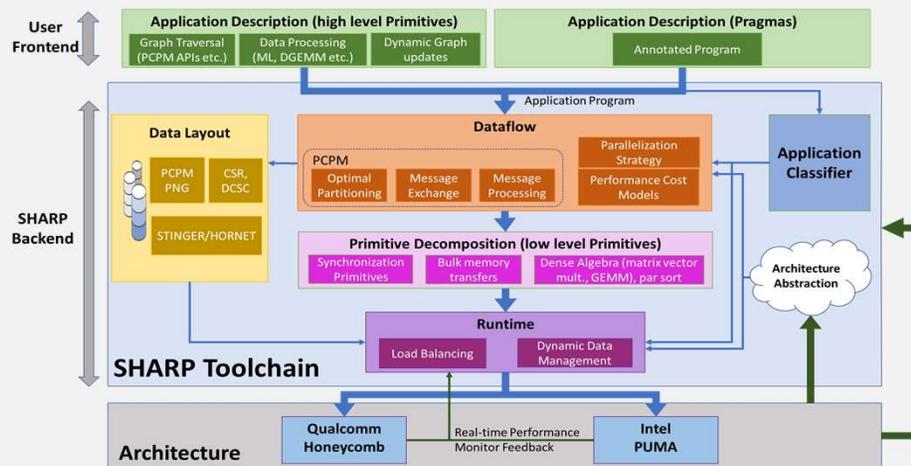
Impact: Roadmap For 1000× efficiency

Processing efficiency		SHARP software	HW support for primitives	HW energy efficiency
1000×	=	20×	10×	5×

Current acceleration using SHARP
- **20× SW** speedup over state-of-the-art on Xeon
- Subset of optimizations deployed currently
- Experiments on Xeon – minimal hardware support for graph processing

On TA1 architectures → 1000× processing efficiency
- **10× hardware** speedup
- Additional Dataflow optimizations
- **5× energy** savings per edge traversed

Six published papers at USENIX ATC, HPEC, HiPC

Georgia Institute of Technology

USC University of Southern California

THE ELECTRONICS RESURGENCE INITIATIVE