An Advanced Graph Processor Prototype

Vitaliy Gleyzer

GraphEx 2016
Graph Analysis at Scale

Interested in enabling advanced data analysis of large graphs in the embedded and data center environments.
Mathematical Foundation

- Graphs capture relationship information between entities
  - Molecular forces
  - Social interactions
  - Semantic concepts
  - Vehicle tracks

- Graphs can be fully expressed in the language of linear algebra
  - Represented as sparse matrices
  - Enable mathematic foundation for data analysis
  - Leverage existing linear algebra techniques and methods
  - Define a small set of well-defined mathematical operations
Graph Structure

Structured Graphs

- Contain inherent connectivity patterns
- Edges constrained via some physical phenomenology
- Can be processed efficiently via careful hand tuning and mapping

Unstructured Graphs

- No inherent structure
- Random distribution of edges
- No clear optimization for processing

Unstructured graphs are inherently difficult to process
Unstructured Graphs of Interest

Cross-domain Dataset Examples

**ISR**
- Intelligence information
- ~1K – 1M entities and connections

**Social**
- Relationships between individuals
- ~10M – 10B individuals and interactions

**Cyber**
- Network patterns
- ~1M – 1B network events

**Bio**
- Connectivity between brain regions
- ~1B – 1T regions and connections

**Scale**

Graphs of interests are large, unstructured and often follow a power-law distribution
Graph Analysis Application Stack

Hardware acceleration of a small number of well-defined mathematical operations enable an extensive analytic ecosystem.
Commercial HPC* Solutions

Graph algorithms run orders of magnitude slower on conventional processors

* High Performance Computing (HPC)
Commercial HPC System Limitations

• General-purpose processor architecture
  – Cache-based memory architecture
  – Vector-unit processing
  – Lack of application specialization

• Communication architectures
  – Insufficient cross-sectional bandwidth
  – End-to-end oriented reliable communication paradigm
  – Inefficient network utilization
Commercial HPC Performance vs. Power

- Insufficient performance for important DoD and commercial applications.
Graph Processor Requirements

• Scalable architecture to enable graph analysis application
  – Size, Weight and Power (SWaP)

• Provide computational throughput required for real-world graph application

• Native support for all GraphBLAS primitives
  – Access to expert analytic community
Novel Graph Processor Enabling Technologies

High Bandwidth Communication Network
- Multidimensional reliable toroid interconnect
- Randomized routing (US Patent No. 8,819,272)

Data/Algorithm Dependent Multiprocessor Mapping
- Efficient load balancing and memory usage (US Patent No. 8,751,556)

Graph Processor
- Up to 1M nodes
- >100x throughput
- >100x power efficiency

Cacheless Memory
- Optimized for sparse matrix processing access patterns

Accelerator-Based Architecture
- Dedicated VLSI computation modules (US Patent No. 8,751,556)
- Systolic sorting technology (US Patent No. 8,190,943)

Graph BLAS-Based Instruction Set
- Sparse matrix-based architecture

Custom Low-Power Circuits
- Full custom design for critical circuitry
Graph Processor Performance Projections

Architectures under development provide >100x performance improvement while scaling to DoD problems of interest.
Supported Sparse Matrix Operations

- The +, -, *, and / operations can be replaced with any arithmetic or logical operators
  - e.g. max, min, AND, OR, XOR, ...
- Instruction set can efficiently support most graph algorithms

<table>
<thead>
<tr>
<th>Operation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>C = A .* B</td>
<td>Matrix multiply operation is the throughput driver for many important benchmark graph algorithms. Processor architecture highly optimized for this operation.</td>
</tr>
<tr>
<td>C = A .± B</td>
<td>Dot operations performed within local memory.</td>
</tr>
<tr>
<td>C = A ./ B</td>
<td>Operation with matrix and constant. Can also be used to redistribute matrix and sum columns or rows.</td>
</tr>
</tbody>
</table>
Graph Processor Node Architecture

• Key attributes:
  – Accelerator-based reconfigurable architecture
    • High-performance optimized hardware for all instructions
  – Flexible memory arbitration for all modules
  – Ability to pipeline multiple accelerators together
    • Optimizes external memory access
  – Native hardware support for sparse matrix formats
  – Simple FIFO-based network interface
Early Concept Demonstration System

- 4-board COTS PCIe system
- 320 MTEPS
- Supports:
  - Up to 8 processing nodes
  - 1D toroidal interconnect (can be expanded to 2D)
  - Parallel sparse matrix-matrix operations (including multiplication and element-wise operations)
Large-Scale High-Performance FPGA Board System Development

- Scalable OpenVPX-based FPGA system
- Up to 40 TTEPS
- Board specifications:
  - 4 nodes
  - 32GB of SDRAM
  - 960 Gb/s I/O network bandwidth
- Supports:
  - Up to 256K boards and 1M processing nodes
  - Up to 6D network topology
  - Full GraphBLAS API
Technology Development and Demonstration Plan

COTS-based FPGA Prototype
- 320M TEPS*

Custom FPGA Processor
- 2,560M TEPS

Custom FPGA Rack
- 10G TEPS

SWaP-Optimized ASIC
- 5T TEPS

ASIC Processor
- 100G TEPS

Data Center: FY19-21
- 100T TEPS

Embedded: FY19-21

* Traversed Edges Per Second (TEPS)
** MA Green High Performance Computing Center (MGHPCC)
Summary

• Graph processing is critical to many commercial, DoD, and intelligence community applications

• Conventional processors perform poorly on graph algorithms
  – Architecture is poorly match to computational flow

• MIT LL has developed a novel sparse matrix processor architecture optimized for graph processing
  – Numerous innovations enable highly efficient graph computing
  – Orders of magnitude higher performance projected versus conventional supercomputers

• MIT LL is developing a Graph Processor Prototype using FPGA technology
  – Future ASIC version expected to deliver significantly higher performance and power efficiency to enable ultra large scale applications