The Future of Advanced (Secure) Computing

DataSToRM: Data Science and Technology Research Environment

Mr. Vitaliy Gleyzer
MIT Lincoln Laboratory
5 March 2018
Advancing the State of Big Data Analytics: Raw Data to Insight

1. **What new insight can be gained from the data?**
2. **What new information should be collected?**
3. **Can different data help improve human understanding?**
4. **How can data be stored/collected more efficiently?**
5. **How can the insight be presented to improve human cognition?**
6. **Can analytics be modified to allow efficient implementation?**
7. **Are there new application areas enabled by new analytics?**
8. **How can existing analytics be accelerated?**

**Big Data Application**
- Presentation
- Data
- Analytics
- Technologies
Large-Scale Graph Applications Today

**Applications**
- Drug Discovery
- Personalized Healthcare
- Fraud Detection
- Recommender Systems
- Functional Brain Mapping
- Cyber Data Analysis

**Development Environment**

**Hardware Platform**
- Lincoln Laboratory Supercomputing Center

**Graph Analysis Frameworks and Databases**
- neo4j
- Project Pegasus
- Giraph
- MATLAB
- GraphX
- D4M
- Pregel

**Algorithms**
- PageRank
- Centrality
- Walktrap
- InfoMap
- K-Truss
- BFS
- MST

**Visualization Toolkits**
- Gephi
- Tulip
- KeyLines
- graphistry

**Diverse, quickly evolving ecosystem**
Advancing the State of Big Data Analytics: Challenges

• Technology moves quickly
  – New algorithms and analytic techniques
  – New storage solutions
  – New processing technologies
  – New database technologies and frameworks
  – New applications
• New framework adoption is a serious investment
• How to leverage new technologies?
• How to enable co-design opportunities?
• How to integrate disparate communities to enable co-design?

Keeping up with big data technology is challenging
Different communities are attempting to unify and standardize interface and languages in the ecosystem.
Different communities are attempting to unify and standardize interface and languages in the ecosystem.
Unifying Principles for Big Data Graphs

• 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
### DataSToRM: Data Science and Technology Research Environment

<table>
<thead>
<tr>
<th>Applications</th>
<th>Threat Detection</th>
<th>Sentiment Analysis</th>
<th>Recommender Engine</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Analysis Kernels</td>
<td>Community Detection</td>
<td>Classification</td>
<td>Centrality Analysis</td>
<td>...</td>
</tr>
<tr>
<td>API</td>
<td>GraphBLAS (Semi-ring Linear Algebra API)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td><img src="image1" alt="Graph Processor" /></td>
<td><img src="image2" alt="LSI" /></td>
<td><img src="image3" alt="AMD" /> <img src="image4" alt="NVIDIA" /> <img src="image5" alt="intel" /> <img src="image6" alt="XILINX" /></td>
<td></td>
</tr>
</tbody>
</table>

Hardware acceleration of a small number of well-defined mathematical operations enable an extensive analytic ecosystem.
GraphBLAS Overview

• Five key operations
  \[ A = S^{NxM}(i,j,v) \quad (i,j,v) = A \quad C = A \oplus B \quad C = A \otimes C \quad C = A \odot B = A \oplus \otimes B \]

• Can be used to build 12 GraphBLAS standard functions
  buildMatrix, extractTuples, Transpose, mXm, mXv, vXm, extract, assign, eWiseAdd, eWiseMult, apply, reduce

• Can be used to build a variety of graph utility functions
  Tril(), Triu(), Degreed Filtered BFS, ...

• Can be used to build a variety of graph algorithms
  K-Truss, Jaccard Coefficient, Non-Negative Matrix Factorization, ...

• That work on a wide range of graphs
  Hyper, multi-directed, multi-weighted, multi-partite, multi-edge

Unifying interface for backend graph processing
Lincoln Laboratory Technologies Targeting Large-Scale Graph Analytics

Dynamic Distributed Dimensional Data Model (D4M)
Data analysis framework based on associative array algebra
- Concise language for complex graph analytics
- Mathematical closure
- Linear Algebra underpinning

Graph Processor
Novel graph processing architecture
- Scalable graph processing hardware architecture
- Unprecedented performance
- Native linear algebra instruction set

Graph Algorithms

D4M

GraphBLAS

GraphBLAS
Standard API for graph analytics using Sparse Linear Algebra primitives

Graph Processor

LLSC

Lincoln Laboratory Super Computing Center (LLSC)
State-of-the-art super computing environment
- Heterogeneous processing capabilities
- Ideal technology integration environment

• Simple hardware agnostic API
Graph Processor Matrix Multiply Performance

Highly efficient graph processing technology that is 100s to 1000s of times more efficient compared to traditional architectures.
Collaboration Opportunities

• Developing graph algorithms in the language of linear algebra
  – Community detections, subgraph isomorphism, subgraph matching, etc.

• Developing graph algorithms that can scale to datasets with billions to trillions of vertices
  – Sparsity-aware, distributed memory algorithms

• Identifying or developing new technologies to leverage the linear algebra abstraction
  – Compilers, optimizer, hardware accelerators, etc.

• Integrating GraphBLAS backend as part of popular frameworks
  – e.g., Apache TinkerPop, Neo4j, ElasticSearch, etc.