Cost Efficient Large-Scale Graph Analytics

Dr. Joseph Schneible
Applications of Graph Analysis

- Social Networks
- WWW
- Medicine
- Natural Language
- Cybersecurity
- Homeland Security
- Local Government
OUTLINE:

- Graph Analysis
- System Design
- Performance
How to bring meaning to this?
## Example Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>Find Influential Nodes Within a Network</td>
</tr>
<tr>
<td>Community Detection</td>
<td>Find Dense Sub-graphs</td>
</tr>
<tr>
<td>Belief Propagation</td>
<td>Perform Inference on a Graph</td>
</tr>
</tbody>
</table>
PageRank trends linearly with degree

- Anomalous nodes are above this trend line
- Used to find mastermind of 9/11 attacks
- Can be applied to biological networks, etc
OUTLINE:

- Graph Analysis
- System Design
- Performance
Goals

Affordability

Time Efficiency

Customizability

Meaning
System Approach

- All of these are affected by design choices:

  - Commodity Hardware
  - Memory Efficiency
  - GPU Utilization
  - I/O Efficiency
  - Graph Construction
### Challenges

<table>
<thead>
<tr>
<th>Parallelization</th>
<th>Memory Limitations</th>
<th>Irregular Graph Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Edges or Vertices?</td>
<td>▪ GB of RAM and vRAM</td>
<td>▪ Many Nodes with Few Connections</td>
</tr>
<tr>
<td></td>
<td>▪ TB Graphs</td>
<td>▪ Few Nodes with Many Connections</td>
</tr>
</tbody>
</table>

- Edges or Vertices?
- GB of RAM and vRAM
- TB Graphs
- Many Nodes with Few Connections
- Few Nodes with Many Connections
Parallelization Strategies

- **Edge-wise Distribution:**
  - One Operation per Edge
  - Memory for Temporary Data Structures
  - Even Load Balance

- **Vertex-wise Distribution:**
  - Multiple Operations per Vertex
  - Uneven Load Balance
Load Balancing Graph Analysis on the GPU

- High degree vertices will dominate computation time
- Created multiple kernels
- Threshold between high and low degree
Out-of-Core Graph Processing

- Divide graph into intervals (sets) of vertices
- Gather associated in-edges into a shard
- Order edges in shards such that out-edges are located together in windows
Compression

- Power-law distribution is common in natural graphs
- Compression scheme exploits distribution
Task Analysis

■ Use:
  ■ Graph Meta-data
  ■ Performance Models
  ■ Micro-benchmarks

■ To:
  ■ Divide work between CPU and GPU
  ■ Divide work between kernels
OUTLINE:

- Graph Analysis
- System Design
- Performance
PageRank Performance

**PageRank:**
- 5 Iterations

**LiveJournal Graph:**
- Vertices: 4.6 Million
- Edges: 77.4 Million

**FUNL Desktop System**
- 9.5 Seconds

**Spark Cluster**
- 110.4 Seconds

**FUNL Desktop System:**
- GPU — GeForce GTX TITAN, 2688 CUDA Cores, 928MHz, 6GB vRAM
- CPU — Core i7, Quad Core, 3.40GHz
- RAM — 16GB (4x4GB), 1333MHz
- Storage — HDD, 180MB/s

**Spark Cluster:**
- System: AWS EC2 m1.large
- Number of Nodes: 10
- Network: Moderate Performance
Belief Propagation Performance

**FUNL/Quad Core CPU:**
- GPU — GeForce GTX TITAN, 2688 CUDA Cores, 928MHz, 6GB vRAM
- CPU — Core i7, Quad Core, 3.40GHz
- RAM — 16GB (4x4GB), 1333MHz
- Storage — HDD, 180MB/s

**16 Core Server:**
- CPU — Xeon E5-2690, 16 Cores, 2.9GHz
- RAM — 64GB, 1600MHz
- Storage — HDDx6, RAID0, 690MB/s

- https://github.com/GraphChi
Advantages

■ Unique Features:
  ■ Large scale Graph Analysis on the GPU
  ■ Task Analysis for Efficient Parallel Processing (on CPU’s and GPU’s)
  ■ UI and Interactive Visualization to bring Meaning to Big Data

■ Benefits:
  ■ Big Data Graph Analysis on a Budget
  ■ Customizability
  ■ Ease of Use (You don’t have to be a data scientist)
  ■ Reduction in Infrastructure and Energy Needs
Big Data Appliance for Graph Analytics

- Gain insight by discovering unknown relationships in big data.
- Achieve a competitive advantage without a large budget.
- Ease adoption with a small footprint solution and customizability.

Graph analytics solution that supports pattern discovery and inferencing on large scale data sets.

Purpose-built to solve big data graph problems with commodity hardware.

Data-center-friendly appliance with a suite of graph algorithms and flexibility to add custom solutions.
Point of Contact

Joe Schneible
Enterprise Software Solutions
Engineering Group Manager

Email: jschneible@technicacorp.com
LinkedIn: linkedin.com/in/jschneible

Technica Corporation
22970 Indian Creek Dr., Suite 500
Dulles, VA 20166

703.662.2000

technicacorp.com