Benchmarking Graph-Processing Platforms: A Vision
(A SPEC Research Group Process)

Yong Guo, Ana Lucia Varbanescu, Alexandru Iosup,
Claudio Martella, Theodore L. Willke

Delft University of Technology, University of Amsterdam,
VU University Amsterdam, Intel Corporation
The data deluge: large-scale graphs
Platform diversity

- Platform: the combined hardware, software, and programming system that is being used to complete a graph processing task.

![Platform Logos](image)
What is the performance of these platforms?

Performance Metrics

Graph Diversity

Algorithm Diversity

- Graph500
  - Single application (BFS), Single class of synthetic datasets

- Few existing platform-centric comparative studies
  - Prove the superiority of a given system, limited set of metrics

Our vision: a benchmarking suite for graph processing across all platforms
A Call to Arms

• Defining workloads
• Understanding the metrics, datasets, and algorithms used in practice: fill in our survey http://goo.gl/TJwkTg

• Evaluating and reporting on various platforms

Join us within the SPEC RG Cloud Working Group
http://research.spec.org/working-groups/rg-cloud-working-group.html
Our Vision for Benchmarking Graph-Processing Platforms

• Methodological challenges
  1. Evaluation process
  2. Selection and design of performance metrics
  3. Dataset selection
  4. Algorithm coverage

• Practical challenges
  5. Scalability of evaluation, selection processes
  6. Portability
  7. Result reporting

Selection and Design of Performance Metrics for Graph Processing

- Raw processing power
  - Execution time
  - Actual computation time
  - Edges/Vertices per second

- Resource utilization
  - CPU, memory, network

- Scalability
  - Horizontal vs. vertical
  - Strong vs. weak

- Overhead
  - Data ingestion time
  - Overhead time

- Elasticity (?)

Challenge 2. Metric selection
## Dataset Selection: Application Domains

- Number of vertices, edges, link density, size, directivity, etc.

<table>
<thead>
<tr>
<th>Graphs</th>
<th>#V</th>
<th>#E</th>
<th>d</th>
<th>D̄</th>
<th>Directivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1 Amazon</td>
<td>262,111</td>
<td>1,234,877</td>
<td>1.8</td>
<td>4.7</td>
<td>directed</td>
</tr>
<tr>
<td>G2 WikiTalk</td>
<td>2,388,953</td>
<td>5,018,445</td>
<td>0.1</td>
<td>2.1</td>
<td>directed</td>
</tr>
<tr>
<td>G3 KGS</td>
<td>293,290</td>
<td>16,558,839</td>
<td>38.5</td>
<td>112.9</td>
<td>undirected</td>
</tr>
<tr>
<td>G4 Citation</td>
<td>3,764,117</td>
<td>16,511,742</td>
<td>0.1</td>
<td>4.4</td>
<td>directed</td>
</tr>
<tr>
<td>G5 DotaLeague</td>
<td>61,171</td>
<td>50,870,316</td>
<td>2,719.0</td>
<td>1,663.2</td>
<td>undirected</td>
</tr>
<tr>
<td>G6 Synth</td>
<td>2,394,536</td>
<td>64,152,015</td>
<td>2.2</td>
<td>53.6</td>
<td>undirected</td>
</tr>
<tr>
<td>G7 Friendster</td>
<td>65,608,366</td>
<td>1,806,067,135</td>
<td>0.1</td>
<td>55.1</td>
<td>undirected</td>
</tr>
</tbody>
</table>

### The Game Trace Archive
- https://snap.stanford.edu/
- http://www.graph500.org/
- http://gta.st.ewi.tudelft.nl/
Graph-Processing Algorithms

- Literature survey of metrics, datasets, and algorithms
  - 10 top research conferences: SIGMOD, VLDB, HPDC ...
  - Key word: graph processing, social network
  - 2009–2013, 124 articles

<table>
<thead>
<tr>
<th>Class</th>
<th>Examples</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Statistics</td>
<td>Diameter, PageRank</td>
<td>16.1</td>
</tr>
<tr>
<td>Graph Traversal</td>
<td>BFS, SSSP, DFS</td>
<td>46.3</td>
</tr>
<tr>
<td>Connected Component</td>
<td>Reachability, BiCC</td>
<td>13.4</td>
</tr>
<tr>
<td>Community Detection</td>
<td>Clustering, Nearest Neighbor</td>
<td>5.4</td>
</tr>
<tr>
<td>Graph Evolution</td>
<td>Forest Fire Model, PAM</td>
<td>4.0</td>
</tr>
<tr>
<td>Other</td>
<td>Sampling, Partitioning</td>
<td>14.8</td>
</tr>
</tbody>
</table>

Platforms we have evaluated

• Distributed or non-distributed
• Graph-specific or generic

BFS: results for all platforms, all graphs

Key Findings From the Study of 6 Platforms

- Performance is function of (Dataset, Algorithm, Platform, Deployment)
  - Previous performance studies may lead to tunnel vision

- Platforms have their own drawbacks (crashes, long execution time, tuning, etc.)
  - Best-performing is not only low response time
  - Ease-of-use of a platform is very important

- Some platforms can scale up reasonably with cluster size (horizontally) or number of cores (vertically)
  - Strong vs weak scaling still a challenge—workload scaling tricky

Thank you for your attention! Comments? Questions? Suggestions?

- Join us, join the SPEC Research Group
- Fill in our survey on Big Data Use Cases  http://goo.gl/TJwkTq
- Ask about other results

Yong Guo
Yong.Guo@tudelft.nl
http://www.pds.ewi.tudelft.nl/yong/
Parallel and Distributed Systems Group
Delft University of Technology
Experimental Setup

- **DAS4**: a multi-cluster Dutch grid/cloud
  - Intel Xeon 2.4 GHz CPU (dual quad-core, 12 MB cache)
  - Memory 24 GB
  - 1 Gbit/s Ethernet network

- **Size**
  - Most experiments take 20 working machines
  - Up to 50 working machines

- **HDFS** used here as distributed file systems
Design of a Survey

• Survey on Big Data Use Cases
  http://goo.gl/TJwkTg

• In total, 33 questions
• 15-20 minutes avg to finish the survey, at most 40 minutes

• Includes 4 major parts
  • Institution profile—Who processes Big Data?
  • System infrastructure—On what infrastructure?
  • Data collection, storage, and process—Which workloads, metrics, etc.? What are the challenges?
  • Framework stacks—Who do they rely on?
Our method: A benchmark suite

- Identifying the performance aspects and metrics of interest
- Defining and selecting representative datasets and algorithms
- Implementing, configuring, and executing the tests
- Analyzing the results
Selection of algorithms

A1: General Statistics (STATS: # vertices and edges, LCC)
  • Single step, low processing, decision-making
A2: Breadth First Search (BFS)
  • Iterative, low processing, building block
A3: Connected Component (CONN)
  • Iterative, medium processing, building block
A4: Community Detection (CD)
  • Iterative, medium or high processing, social network
A5: Graph Evolution (EVO)
  • Iterative (multi-level), high processing, prediction
Scalability: BFS on Friendster

- Using more computing machines/cores can reduce execution time
- Tuning needed, e.g., for GraphLab, split large input file into number of chunks equal to the number of machines
The CPU utilization: computing node

- YARN and Hadoop exhibit obvious volatility
- The CPU utilization of graph-specific platforms is lower
The percentage of overhead time of generic platforms is smaller.

The percentage of overhead time is diverse across the platforms, algorithms, and graphs.
Additional Overheads
Data ingestion time

- Data ingestion
  - Batch system: one ingestion, multiple processing
  - Transactional system: one ingestion, one processing

- Data ingestion matters even for batch systems

<table>
<thead>
<tr>
<th></th>
<th>Amazon</th>
<th>DotaLeague</th>
<th>Friendster</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS</td>
<td>1 second</td>
<td>7 seconds</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Neo4J</td>
<td>4 hours</td>
<td>6 days</td>
<td>n/a</td>
</tr>
</tbody>
</table>