Billion-Scale Graph Analytics

Toyotaro Suzumura¹, Koji Ueno, Charuwat Hounckaew, Hidefumi Ogata, Masaru Watanabe, and ScaleGraph Team

Principal Investigator for JST CREST Project
Research Scientist, IBM Research
Visiting Associate Professor, University College Dublin
Visiting Associate Professor, Tokyo Institute of Technology (2009/04-2013/09)
Outline

- Introduction
- Graph500 Benchmark
- ScaleGraph: Billion-Scale Graph Analytics Library
- Time-Series Analysis for Whole Twitter Network
- Summary
Large-Scale Graph Mining is Everywhere

- Cybersecurity
- Medical Informatics
- Data Enrichment
- Social Networks
- Symbolic Networks

Internet Map

Symbolic Networks:

Social Networks

Protein Interactions

Cyber Security (15 billion log entries / day for large enterprise)
Large-Scale Graph Processing System (2011-2018)

Data Source
- Sensors
  - Smart Meters
  - Smart Grid
  - GPS
  - SNS (Twitter)

Disaster Management
- Transportation, Evacuation, Logistics
- Energy・Power Saving
- Social Network Analysis

Large-Scale Graph Processing System
- Large-Scale Graph Visualization
- Real-Time Graph Stream Processing
  - Centrality
  - Shortest Path
  - Quickest Flow Problem
  - PageRank / RWR
  - Clustering
  - Semi-Definite Programming
  - Mix Integer Programming
  - Real-Time Stream Processing System
  - X10 Language
- Large-Scale Graph Library
- 100 Peta Flops Heterogeneous Supercomputer
- Large-Scale Graph Store
Graph500 is a new benchmark that ranks supercomputers by executing a large-scale graph search problem.

The benchmark is ranked by so-called TEPS (Traversed Edges Per Second) that measures the number of edges to be traversed per second by searching all the reachable vertices from one arbitrary vertex with each team’s optimized BFS (Breadth-First Search) algorithm.
Highly Scalable Graph Search Method for the Graph500 Benchmark

- We propose an optimized method based on 2D based partitioning and other various optimization methods such as communication compression and vertex sorting.
- We developed CPU implementation and GPU implementation.
- Our optimized GPU implementation can solve BFS (Breadth First Search) of large-scale graph with $2^{35}$ (34.4 billion) vertices and $2^{39}$ (550 billion) edges for 1.275 seconds with 1366 nodes (16392 cores) and 4096 GPUs on TSUBAME 2.0
- This record corresponds to 431 GTEPS


Vertex Sorting by utilizing the scale-free nature of the Kronecker Graph

2D Partitioning Optimization

![Graph 500 Logo]

Scalable 2D partitioning based CPU Implementation with Scale 26 per 1 node
TSUBAME 2.5 Supercomputer in Tokyo

**TOP 10 Systems - 06/2011**

1. **K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect**
2. **Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C**
3. **Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz**
4. **Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU**
5. **TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows**

**Complete Results - November 2011**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Machine</th>
<th>Owner</th>
<th>Problem Size</th>
<th>TEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NNSA/SC Blue Gene/Q Prototype II (4096 nodes / 65,536 cores)</td>
<td>NNSA and IBM Research, T.J. Watson</td>
<td>32</td>
<td>254,349,000,000</td>
</tr>
<tr>
<td>2</td>
<td>Lomonosov (4096 nodes / 32,768 cores)</td>
<td>Moscow State University</td>
<td>37</td>
<td>103,251,000,000</td>
</tr>
<tr>
<td>3</td>
<td>TSUBAME (2732 processors / 1366 nodes / 16,392 CPU cores)</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>36</td>
<td>100,366,000,000</td>
</tr>
<tr>
<td>4</td>
<td>Jugene (65,536 nodes)</td>
<td>Forschungszentrum Jülich</td>
<td>37</td>
<td>92,876,900,000</td>
</tr>
<tr>
<td>5</td>
<td>Intrepid (32,768 nodes / 131,072 cores) ANL</td>
<td></td>
<td>35</td>
<td>78,869,900,000</td>
</tr>
</tbody>
</table>
Our Scalable Algorithm continuously achieves 3rd or 4th place in the World since 2011/11

No.4 (2012/06)  No.3 (2011/11)  No.3 (2012/06)
Outline

- Introduction
- Graph500 Benchmark
- ScaleGraph: Billion-Scale Graph Analytics Library
- Time-Series Analysis for Whole Twitter Network
Our Motivation

- Create an open source **Highly Scalable Large Scale Graph Analytics Library** beyond the scale of billions of vertices and edges on Distributed Systems

- Symbolic Networks:
  - Protein Interactions
  - Cyber Security (15 billion log entries / day for large enterprise)

- Internet Map

- Social Networks
Existing Graph Analytics Libraries

- **Single Node**
  - igraph (R package)
  - GraphLab/GraphChi (Carnegie Mellon University and Start-up, C++)

- **Distributed Systems**
  - **MPI-based libraries**
    - PBGL (Parallel Boost Graph Library, C++) [Gregor, Oopsla 2005]
    - ParMetis (dedicated for parallel graph partitioning, C+), etc
  - **Hadoop-based libraries**
    - Apache Giraph (Pregel Model, Java)
    - PEGASUS (Generalized Iterative Sparse Matrix Vector Multiplication, Java CMU), etc
  - **GPS** (Graph Processing System - Pregel Model, Stanford, Java + NIO)
Programming models that offer performance and programmer productivity are very important for conducting big data analytics in Exascale Systems.

HPCS languages are an example for such initiatives.

It is very important for having complex network analysis software APIs in such languages.
**Aim** - Create an open source **X10-based Large Scale Graph Analytics Library** beyond the scale of billions of vertices and edges.

**Objectives**
- To define concrete abstractions for Massive Graph Processing
- To investigate use of X10 (i.e., PGAS languages) for massive graph processing
- To support significant amount of graph algorithms (E.g., structural properties, clustering, community detection, etc.)
- To create well defined interfaces to Graph Stores
- To evaluate performance of each measurement algorithms and applicability of ScaleGraph using real/synthetic graphs in HPC environments.

**URL:** [http://www.scalegraph.org/](http://www.scalegraph.org/)
Programming Language X10

X10 is a new parallel distributed programming language being developed by IBM Research.

- X10 aims at improving the productivity of highly parallel and distributed applications.
  - Enables scalable programming for parallel distributed environment, where many multicore SMP chips and GPGPUs are interconnected.

- X10 adopts APGAS (Asynchronous Partitioned Global Address Space) programming model.
  - Can manage multiple machines as a global memory space partitioned into “Places”.
  - Can create lightweight asynchronous “Activities”.
  - Supports creation and reference of activities and objects in remote places.

- X10 supports various execution environments.
  - Can run both on Java execution environments and native environments.
  - Provides development tools integrated into Eclipse.

- X10 is being developed as an open source project.
  - See http://x10-lang.org/ for more information.

Credit: X10 Overview by Vijay Saraswat (IBM Research)
Features of ScaleGraph

- XPregel framework which is based on Pregel computation model proposed by Google
- Optimized collective routines (e.g., alltoall, allgather, scatter and barrier)
- Highly optimized array data structure (i.e., MemoryChunk) for very large chunk of memory allocation
- Rich graph algorithms (e.g., PageRank, spectral clustering, degree distribution, betweenness centrality, HyperANF, strongly-connected component, maximum flow, SSSP, BFS)
- We achieved running PageRank, spectral clustering, degree distribution on huge Twitter graph with 469M of users and 28.5B of relationships

ScaleGraph Software Stack

- User Program
- Graph Algorithm
- X10 Graph Processing System
- BLAS for Sparse Matrix
- File IO
- Third Party Library (ARPACK, METIS)
- ScaleGraph Base Library
- X10 Standard Lib Team
- MPI
- X10 & C++
The scale-28 graphs we used have $2^{28} \approx 268$ million of vertices and $16 \times 2^{28} \approx 4.29$ billion of edges.
The scale-28 graphs we used have $2^{28}$ (≈268 million) of vertices and $16 \times 2^{28}$ (≈4.29 billion) of edges.
The scale-28 graphs we used have $2^{28}$ ($\approx$268 million) of vertices and $16 \times 2^{28}$ ($\approx$4.29 billion) of edges.
Outline

- Introduction
- Graph500 Benchmark
- ScaleGraph: Billion-Scale Graph Analytics Library
- Time-Series Analysis for Whole Twitter Network
Understanding time-series nature of large-scale social networks (e.g. separation of degree, diameter, clustering, ..)

Crawled the entire Twitter follower/followee network of **826.10 million vertices and 29.23 billion edges.** How could we analyze this gigantic graph?

Supercomputers
Crawling Billion-Scale Twitter Follower-Followee Network

- with Twitter API (v1.0* ) from Jul. 2012 to Oct. 2012 (around 3 months).
- begin with top 1,000 users*1 with the largest number of followers
- according to breadth-first search along the direction of follower

Crawled Data Set

- **We stopped our crawling at depth 29**
  - Because the user after depth 26 was less than 100.
  - Finally, we collected **469.9 million user data**.

- **Collect two kind of user data by crawling for 3 months**
  - 1. **User profile**
    - Include user id, screen_name, description, account creation time, time zone, etc.
    - The serialized data size is **91GB**
  - 2. **Follower-friend**
    - Adjacency list of followers and friends
    - The compressed (gzip) data size is **231GB**

- **To perform the Twitter network analysis**
  - **Apache Hadoop** for large-scale data processing
  - **HyperANF** for approximate calculation of degree of separation and diameter
    - Lars Backstrom*1 also use HyperANF for Facebook network analysis

*1: “Four degrees of separation” ACM Web Science 2012
Explore Twitter Evolution (1/2)
- Transition of the number of users

- Total user count (left fig.)
  - Twitter started at June 2006 and rapidly expanded from beginning of 2009.
  - Haewoon Kwak *1 studied Twitter network on July 2009

- Monthly increase of users (right fig.)
  - Twitter users increase, but it seems a little unstable...

*1: “What is Twitter, a social network or a news media?”
Explore Twitter Evolution (2/2)
- Transition of the number of users by regions-

- Classify 131 million users by “Time zone” property under 6 regions
  - Africa, Asia, Europe, Latin America and Caribbean (Latin), Northern America (NA), Oceania
  - Only 131 million user correctly set one’s own “Time zone”

- Massive change of ratio of users by region
  - Asia users : 8.30% => 20.8% (12.5% up)
  - NA users : 54.4% => 40.4% (14.0% down)

<table>
<thead>
<tr>
<th></th>
<th>July 2009</th>
<th></th>
<th>October 2012</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># users</td>
<td>ratio (%)</td>
<td># users</td>
<td>ratio (%)</td>
</tr>
<tr>
<td>Africa</td>
<td>0.13M</td>
<td>0.66</td>
<td>1.27M</td>
<td>0.96</td>
</tr>
<tr>
<td>Asia</td>
<td>1.65M</td>
<td><strong>8.30</strong></td>
<td>27.4M</td>
<td><strong>20.8</strong></td>
</tr>
<tr>
<td>Europe</td>
<td>3.01M</td>
<td>15.1</td>
<td>19.8M</td>
<td>15.1</td>
</tr>
<tr>
<td>Latin</td>
<td>3.80M</td>
<td>19.0</td>
<td>28.5M</td>
<td>21.6</td>
</tr>
<tr>
<td>NA</td>
<td>10.9M</td>
<td><strong>54.6</strong></td>
<td>53.1M</td>
<td>40.4</td>
</tr>
<tr>
<td>Oceania</td>
<td>0.45M</td>
<td>2.29</td>
<td>1.52M</td>
<td>1.15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>19.9M</strong></td>
<td>100</td>
<td><strong>131M</strong></td>
<td>100</td>
</tr>
</tbody>
</table>

Characteristic of Twitter network also change?

Monthly increase of users by region
Monthly Increase of Users by Regions
Degree Distribution: **Unexpected value in in-degree distribution**

- "Scale-free" is one of the features of a social graph
- **Unexpected value in in-degree distribution**
  - at $x=20$ due to Twitter recommendation system
  - at $x=2000$ due to upper bound of friends before 2009
Reciprocity: decline from 22.1% to 19.5%

- Reciprocity is a quantity to specifically characterize directed networks. Traditional Definition:
  \[
  r = \frac{L^{\leftrightarrow}}{L}
  \]
  \(L^{\leftrightarrow}\): # of edges pointing in both directions
  \(L\): # of total edges

- As a result, **Twitter network reciprocity decline from 22.1% to 19.5%**

<table>
<thead>
<tr>
<th></th>
<th>July 2009</th>
<th>October 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td># of users</td>
<td>41.6 M</td>
<td>465.7 M</td>
</tr>
<tr>
<td># of edges</td>
<td>1.47 B</td>
<td>28.7 B</td>
</tr>
<tr>
<td><strong>Reciprocity</strong></td>
<td>22.1% *1</td>
<td>19.5%</td>
</tr>
</tbody>
</table>

*1: “What is Twitter, a social network or a news media?”
How many edges do celebrities have in Twitter network? → Only 0.06% celebrities control most of edges

- 93% users have less than or equal to 100 followers
- 99.94% users have less than or equal to 10,000 followers
- However, their followers count are only 11% of total followers count
- But still... 57.6% of total followers count
- Only 0.06% celebrities control most of edges in Twitter network
Degree of Separation and Network Diameter (1/3)

- Both degree of separation and diameter are measures to characterize networks in terms of scale of graph.

- **Definition**
  - **Degree of Separation:**
    - Average value of the shortest-path length of all pairs of users.
  - **Diameter:**
    - Maximum value of the shortest-path length of all pairs of users.
  - *Note: unreachable pairs are excluded from calculation*

\[\begin{align*}
(A, B) &= 1 \\
(A, C) &= 1 \\
(B, A) &= \infty \\
(B, C) &= 1 \\
(C, A) &= \infty \\
(C, B) &= 1
\end{align*}\]

Degree of Separation : 1  
Diameter : 1
Experimental environment

- Using an approximate algorithm named HyperANF [Paolo, WWW’12] on TSUBAME 2.0 (Supercomputer at TITECH)
  - TSUBAME 2.0 Fat node
    - 64 cores, 512 GB memory, SUSE Linux Enterprise Server 11 SP1
  - HyperANF Parameters
    - We set the logarithm of the number of registers per counter to 6 in order to reduce an error.
- Four times executions
  - Degree of Separation
    - take a average of 4 calculation
  - Diameter
    - take a minimum value of 4 calculation
    - because HyperANF guarantee lower bound of diameter
  - Each execution on 2012 took more than 42,000 sec.
Degree of Separation and Network Diameter (3/3)

- **Degree of Separation**
  - Only a little difference between ‘09 and ’12 in spite of the lapse of three years.

- **Diameter**
  - Diameter of 2012 is much larger than the one of 2009.

- **Cumulative Distribution**
  - In 2009
    - 89.2% of node pairs whose path length is 5 or shorter
    - 99.1% pairs whose it is 6 or shorter.
  - In 2012
    - 85.2% pairs whose it is 5 or shorter
    - 94.6% pairs whose it is 6 or shorter.

<table>
<thead>
<tr>
<th></th>
<th>Degree of Separation</th>
<th>Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2009</td>
<td>2012</td>
</tr>
<tr>
<td>1st</td>
<td>4.39</td>
<td>4.48</td>
</tr>
<tr>
<td>2nd</td>
<td>4.46</td>
<td>4.65</td>
</tr>
<tr>
<td>3rd</td>
<td>4.53</td>
<td>4.54</td>
</tr>
<tr>
<td>4th</td>
<td>4.62</td>
<td>4.71</td>
</tr>
<tr>
<td>Result</td>
<td>4.50</td>
<td>4.59</td>
</tr>
</tbody>
</table>
Computing Degree of Separation with ScaleGraph on Distributed Systems

The scale-28 graphs we used have $2^{28} \approx 268$ million of vertices and $16 \times 2^{28} \approx 4.29$ billion of edges.

![Strong-scaling result of HyperANF (scale 28)](image-url)
Degree of Separation and Diameter for Time-Evolving Twitter Network

![Graph showing the degree of separation and diameter over time. The x-axis represents months from May 2005 to August 2013, with markers for Oct 2006, Feb 2008, Jul 2009, Nov 2010, Apr 2012, and Aug 2013. The y-axis on the left represents degrees of separation, while the y-axis on the right represents diameter.]
# Classifying Degree of Separation by Spoken Language

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>Portuguese</th>
<th>Japanese</th>
<th>Turkish</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Users</td>
<td>64,927,267</td>
<td>22,456,938</td>
<td>20,279,402</td>
<td>10,402,846</td>
<td>10,743,511</td>
</tr>
<tr>
<td>Follow ratio to</td>
<td>64%</td>
<td>58%</td>
<td>89%</td>
<td>57%</td>
<td>51%</td>
</tr>
<tr>
<td>its own language</td>
<td>(Follow ratio to English)</td>
<td>31%</td>
<td>36%</td>
<td>9%</td>
<td>39%</td>
</tr>
<tr>
<td># of Nodes for DOS</td>
<td>60,708,434</td>
<td>21,152,308</td>
<td>19,682,116</td>
<td>9,638,906</td>
<td>8,964,888</td>
</tr>
<tr>
<td># of Edges for DOE</td>
<td>2,266,838,184</td>
<td>1,098,723,999</td>
<td>1,394,986,423</td>
<td>271,513,323</td>
<td>177,419,512</td>
</tr>
<tr>
<td>Average Degree</td>
<td>37.33</td>
<td>51.94</td>
<td>70.87</td>
<td>28.16</td>
<td>19.79</td>
</tr>
<tr>
<td>Degree of Separation (Average path length between two users)</td>
<td><strong>4.625</strong></td>
<td><strong>4.253</strong></td>
<td><strong>4.014</strong></td>
<td><strong>4.340</strong></td>
<td><strong>4.699</strong></td>
</tr>
<tr>
<td>Diameter (Lower bound value)</td>
<td>42</td>
<td>23</td>
<td>27</td>
<td>39</td>
<td>22</td>
</tr>
</tbody>
</table>
Concluding Remarks

- **Graph500, ScaleGraph, and Social Network Analytics**

  - Project information and Documentation
  - Source code distribution / VM Image

- **Ongoing/Future Work**
  - Other domains: RDF Graph, Human Brain Project (EU)
  - More temporal web analytics on our whole Twitter follower-followee network and all the user profile as of 2012/10
  - Performance comparison w/ other graph libraries
  - Performance Evaluation on Amazon EC2