Triangle Enumeration in Large-scale Graph

Chan Shing-Kit, Huang Huaxun
Outline

Part One: A brief introduction to triangle enumeration

Part Two: Research papers about triangle enumeration in large scale graph
A Brief Introduction to Triangle Enumeration

Chan Shing-Kit
What is triangle enumeration

- Give graph $G$, enumerate the triangles in $G$.
  - Sometimes we just want to count/approximate.
  - Enumeration important to answer questions like: “how many triangles incident to $v$?”
Practice

- People want simple parallelizable algorithms.
- Enumerating triangle in large graph brings a challenge.
Classic Algorithms

Search by spanning tree

1. Tree
   i. Build a rooted spanning tree for each component
   ii. If any tree edge contains a triangle, then return it and delete the edges forming the triangle

2. Triangle
   i. Repeat Tree until all triangles have been found

● Requires at most $O(e^{1.5})$
Classic Algorithms

Search by vertices

- $G$ contains a triangle if there exists a vertex $v$ and an edge $a$ between two vertices $u$ and $w$ ($u < w$) of $UA(V)$.
- For each vertex in $G$, find if there is an edge between the adjacent vertices
- $O(n^{5/3})$
Classic Algorithms

Matrix multiplication

- Let $M$ be the adjacency matrix
- $M^2_{uv} = 1$ if there exists a $w$ where $(u,w)$ and $(w,v)$ are in $E$
- Let $B = M^2$ and $M$
- $B_{uv} = 1$ if a triangle pass through $(u,w)$

$O(n^{\log 7})$
Why we need triangle enumeration

- Triangles are primary building blocks of many problems
- Recent work on social network wants a fast algorithm to enumerates the connections among people and analyze the underlying social process.

Social sciences: [Holland-Leinhardt70] [Coleman88] [Skvoretz90] [Portes98] [Burt04] [Welles etal10] [Faust10] [Szell-Thurner10]

Physical sciences: [Watts-Strogatz98] [Eckmann-Moses02] [Fagiolo07] [Milo etal10] [Son etal10] [Leskovec etal10] [Winkler-Reichardt13]

Algorithmics: [Becchetti etal08][Berry etal11] [Gleich-S12] [Rohe-Qin13]
Applications

Is a “Friend” a Friend? Investigating the Structure of Friendship Networks in Virtual Worlds

- Use balance of triangles as one metric in measuring the relationship among friends in social networks

Figure 1. Balanced and unbalanced triangles.
Why do traditional algorithms not fit

- Traditional algorithms focus on dense graph and complete enumeration of triangles
- However, real networks are usually huge but sparse graph and sometimes approximation is enough
- We also need a parallelizable algorithm

<table>
<thead>
<tr>
<th>Graph</th>
<th>[SNAP, LAW]</th>
<th>n</th>
<th>m</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>web-BerkStan</td>
<td></td>
<td>700K</td>
<td>6.6M</td>
<td>64M</td>
</tr>
<tr>
<td>flickr</td>
<td></td>
<td>1.8M</td>
<td>15M</td>
<td>550M</td>
</tr>
<tr>
<td>livejournal</td>
<td></td>
<td>5.2M</td>
<td>48M</td>
<td>300M</td>
</tr>
<tr>
<td>uk-union</td>
<td></td>
<td>132M</td>
<td>4B</td>
<td>450B</td>
</tr>
</tbody>
</table>
Part Two

Research Paper on Triangle Enumeration in Large-Scale Graph

Huang Huaxun
Find triangles in graph
Finding Anomalies

Spammer
Telemarketer
Fake User
The problem is... Enormity of Graphs!

1 billion users facebook
1 trillion pages Google
PTE: Enumerating Trillion Triangles On Distributed Systems

Ha-Myung Park*, KAIST; Sung-Hyon Myaeng, KAIST;
U Kang, Seoul National University
Outline

- Parallel algorithm to enumerate triangles in large-scale graph.
- Reduce searching space for triangle enumeration.
- Reduce memory space.
MapReduce

**Map step:** transforms each pair of the input to a set of new pairs.

**Shuffle step:** groups the pairs by key so that all values with the same key are aggregated together.

**Reduce step:** processes the values by each key separately and outputs a new set of key-value pairs.
3 Types of Triangle

Type-1 triangle

Type-2 triangle

Type-3 triangle
Triangle Enumeration in TTP

- Randomly color the vertices with $O(\sqrt{|E|/M})$ colors by hash function $\xi : V \to \{0, \cdots, \rho - 1\}$
- Get edge set $E_{ij}$ where color of two nodes are $i$ and $j$.
- $(i,j)$ problem: enumerate type-1 and type-2 triangles by enumerating triangles in $E'_{ij} = E_{ij} \cup E_{ii} \cup E_{jj}$
- $(i,j,k)$ problem: enumerate type-3 triangles in $E'_{ijk} = E_{ij} \cup E_{ik} \cup E_{jk}$
- There are $\binom{\rho}{2}$ $(i,j)$ problems, and $\binom{\rho}{3}$ $(i,j,k)$ problems.
- TTP fails to process a graph when the shuffled data size is larger than the total available space.
PTE (Pre-partitioned Triangle Enumeration)

Minimize shuffled data. Massive data are shuffled for generating subgraphs by the previous algorithms. How can we minimize the amount of shuffled data?

Minimize computations. The previous algorithms contain several kinds of redundant operations. How can we remove the redundancy?

Minimize network read. In previous algorithms, each subproblem reads necessary sets of edges via network, and the amount of network read is determined by the number of vertex colors. How can we decrease the number of vertex colors to minimize network read?
PTE (Pre-partitioned Triangle Enumeration)

Separating graph partitioning from generating subgraphs decreases the amount of shuffled data to $O(|E|)$ from $O(|E|^{3/2}/\sqrt{M})$ of the previous MapReduce algorithms.

Considering the color-direction of edges removes redundant operations and minimizes computations.

Carefully scheduling triangle computations in subproblems shrinks the amount of network read by decreasing the number of vertex colors.
**PTE\textsubscript{BASE}: Pre-partitioned Triangle Enumeration**

Given a direct graph, \( E_{ij} \) is a set of triangle where \( (\xi(u), \xi(v)) = (i, j) \) or \( (j, i) \).

**Observation:** The subproblems of TTP require each set \( E_{ij} \) of edges as a unit. If each edge set \( E_{ij} \) is directly accessible from a distributed storage, we need not shuffle the edges like TTP does.

**Partitioning strategy:** partition the whole graph into \( \rho + \binom{\rho}{2} = \frac{\rho(\rho+1)}{2} \) sets of edges, where \( \rho \) and \( \binom{\rho}{2} \) are for \( E_{ij} \) when \( i=j \) and \( i<j \).

Node is randomly colored.
PTE\textsubscript{BASE}: Pre-partitioned Triangle Enumeration

Map step:

PTE\textsubscript{BASE} transforms each edge \((u, v)\) into a pair \(((\xi(u), \xi(v)); (u, v))\), and the edges of the pairs are aggregated by the keys.

Reduce step:

Receives \(E^*_{ij}\) and emits it to a separate file in a distributed storage.

\(E^*_{ij}\) is set of edges \((u, v)\) where \((\xi(u), \xi(v)) = (i, j)\).
PTE\textsubscript{CD}: Reducing the Total Work

Implements on PTE\textsubscript{BASE} to minimize the amount of computations by exploiting color-direction.

Reduce $2-2/\rho$ times of computation effort than PTE\textsubscript{BASE}.
PTE\textsubscript{sc}: Reducing the Network Read

PTE\textsubscript{sc} further improves on PTE\textsubscript{cd} to shrink the amount of network read by scheduling calls of the function of PTE\textsubscript{BASE}.

We can prove that the maximum space takes $\frac{5}{6} \cdot |E|$ for any of the permutations.

<table>
<thead>
<tr>
<th>$E_{ij}$</th>
<th>$\Delta_{ijk}$</th>
<th>$\Delta_{ikj}$</th>
<th>$\Delta_{jik}$</th>
<th>$\Delta_{jki}$</th>
<th>$\Delta_{kij}$</th>
<th>$\Delta_{kji}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{ik}$</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{ji}$</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{jk}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{ki}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{kj}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation

Shuffled data space has been significantly decreased.
Evaluation

Running time reduce significantly.

![Graph showing running time reduction with number of machines. The graph plots running time (sec) on a logarithmic scale against the number of machines. Different lines represent different methods, with labels PTE\textsubscript{CD}, PTE\textsubscript{SC}, CTTP, and PTE\textsubscript{BASE}. The graph indicates a slope of approximately -0.94.]
Conclusion

- The author proposed PTE, a new distributed algorithm for enumerating triangles in an enormous graph, which is designed to minimize the amount of shuffled data, total work, and network read.
- PTE is experimentally evaluated using large real world networks. The results demonstrate that PTE outperforms the best previous distributed algorithms by up to 47×.
Q&A