# Parallel Color Coding and Graph Partitioning Enabling Subgraph Counting for Massive Graphs

Zhao Zhao, Maleq Khan, V.S. Anil Kumar, Madhav V. Marathe

Virginia Tech

## Summary


### Motivation & Challenges

Subgraph/template counting has been widely applied in many areas, say biochemistry, neurobiology, ecology and engineering, e.g.:

- Motif counting in protein-protein network
- Cascade frequency in blog/posts network
- Information cascades in recommendation network

**Challenges in subgraph counting:**

- Running time is exponential in the template size.
- Parallel implementation is difficult, due to the backtracking process in the subgraph counting.
- Previous work are limited in graphs with thousands of nodes, due to the high computational cost and memory usage.

### Our Approach

We propose a parallel algorithm called **ParSE,** to estimate the number of occurrences of a template in very large graphs using color-coding and graph partitioning.

**Features:**

- Can handle graphs with millions of nodes.
- Deal with more generalized and larger templates.
- Estimation error is controllable.

**Basic steps of ParSE:**

- Partition the graph, as well as split the template.
- Use color coding to count the number of sub-template embeddings in each partition.
- Calculate the number of template embeddings in the whole graph, by aggregating the sub-templates' countings.

### Results

Algorithm is tested on:

- million-nodes social contact graphs, random graphs
- various templates

The results showing that our algorithm has:

- High precision in approximation.
- Good scaling to processor, and template size.
- Large speed up over sequential color-coding algorithm.


## The Problem

The problem is to count the number of *non-induced* subgraphs of an undirected graph $G(V, E)$, which are isomorphic to a given template $T(V_T, E_T)$, as shown in Fig. 1.

- *non-induced subgraph*: A subgraph $H(V', E')$ which is *isomorphic* to the template $T$ (there is a bijection $f: V_T \rightarrow V'$ such that if $(u, v) \in E_T$ then $(f(u), f(v)) \in E'$);
- *induced subgraph*: $(u, v) \in E_T$ *i.f.f.* $(f(u), f(v)) \in E'$).

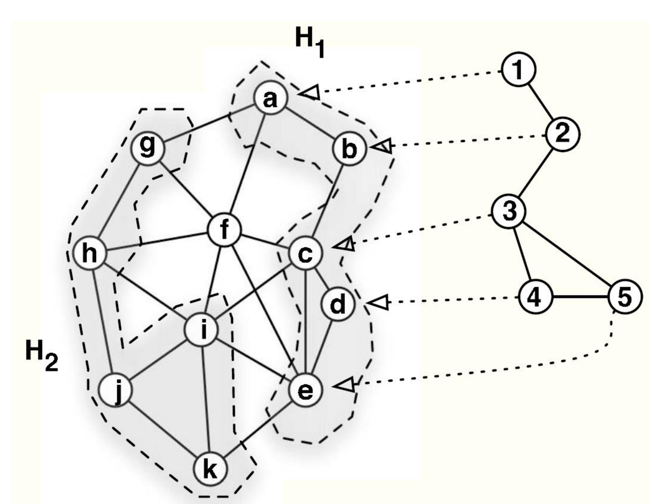

Figure 1: Non-induced and induced occurrences of the template, in which $H_1$ is both an induced and non-induced subgraph, and $H_2$ is only a non-induced subgraph.

## Color Coding Technique

Color-coding is an approximating algorithm to estimate the subgraph embeddings $emb(T,G)$ for a given template $T$ and graph $G$, by counting the colorful embeddings $C$. All the vertices in a "colorful" embedding has distinct color. The procedure of color coding is briefly given below:

1. For $i$ from 1 to $N=O[(e^k \cdot \log 1/\delta)/\varepsilon^2]$ perform the following steps, such that the approximation satisfies:

$$\Pr[|Z - emb(T,G)| > \varepsilon \cdot emb(T,G)] \leq \delta$$

(*Here Z is the estimated number of embeddings. k is the template size, ε and δ are parameters to control the error.*)

a) Color each vertex of $G$ uniformly at random with a color from $\{1,\dots,k\}$.
b) Count $X_i$, the "colorful" embeddings of $T$ in $G$.

2. Partition the $N$ samples above into $O(\log 1/\delta)$ sets, and let $Y_j$ be the average of the $j$ set. Output the median $C$ of $(Y_1,\dots,Y_t)$.
3. Since the possibility that an embedding to be colorful is $k!/k^k$, the number of actual embeddings can be estimated as $Z = C \cdot k^k/k!$.

## ParSE

ParSE deals with the template which can be split into two sub-templates by a "cut-edge" $(u, v)$. We let $u$ and $v$ to be the roots of the two sub-templates $T_1$ and $T_2$. We first count the number of sub-template embeddings rooted from each vertex $w$ in the graph. Then we will aggregate the sub-template countings to obtain the number of template embeddings in the graph. In the following we use $C(w, u, T_i, S_i)$ to denote the number of colorful embeddings of sub-template $T_i$ with root $u$ lying on $w$, specifying the color set $S_i$. Fig. 2 is an example.

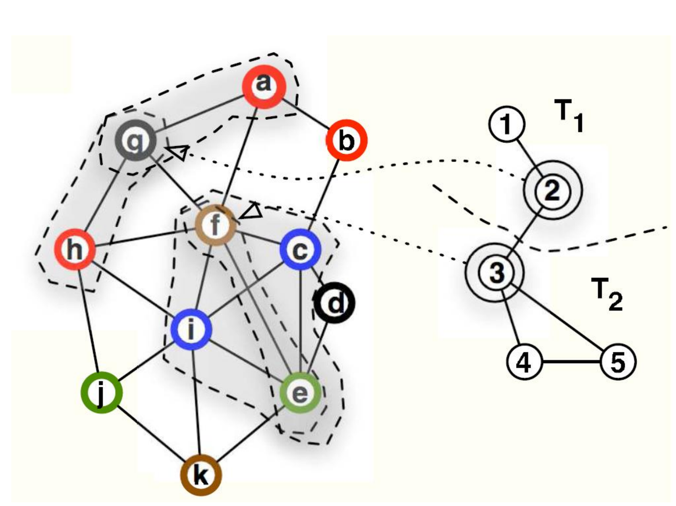

Figure 2: Illustration of the dynamic programming step of color coding. Template $T$ is partitioned into two subgraphs $T_1$ and $T_2$, with roots 2 and 3, respectively. We have $C(g, 2, T_1, S_1 = \{\text{black, red}\}) = 2$ and $C(f, 3, T_2, S_2 = \{\text{brown, blue, green}\}) = 2$. So the colorful embeddings of $T$ located at edge $(g, f)$ is $C(g, 2, T_1, S_1)C(f, 3, T_2, S_2) = 4$.

### ✓ Overview of ParSE

The high-level peudo-code of **ParSE** is given below:

1. Partition $G(V, E)$ and assign processors.
2. Partition $T$ into $T_1$ and $T_2$, let $\rho(T_i)$ denote the root of $T_i$
3. Assign each node $v$ in $V$ a random color from $\{1,\dots,k\}$.
4. For each processor $q$ and each partition $G_p$ assigned to it, do
5. For each node $v$ in $core(G_p)$, each set $S_i \subset \{1,\dots,k\}$, $|S_i| = |T_i|$, $i = 1, 2$, do
6. Compute $C(v, \rho(T_i), T_i, S_i)$
7. For each edge $e=(u,v) \in E$, do
8. Compute $C(e) = \sum_{S_1,S_2} C(u, \rho(T_1), T_1, S_1)\, C(v, \rho(T_2), T_2, S_2)$
9. $+\, C(v, \rho(T_1), T_1, S_1)\, C(u, \rho(T_2), T_2, S_2)$
10. where the sum is over all $S_1 \cup S_2 = \{1,\dots,k\}$.
11. $X = \sum_e C(e)/\beta$,
12. Repeat line 3–11 until the average of $X$ reaches the precision requirement.

Table 1: A high level description of ParSE

❖ Here $\beta$ is the number of cut-edges in $T$, for which the template is isomorphic to itself.

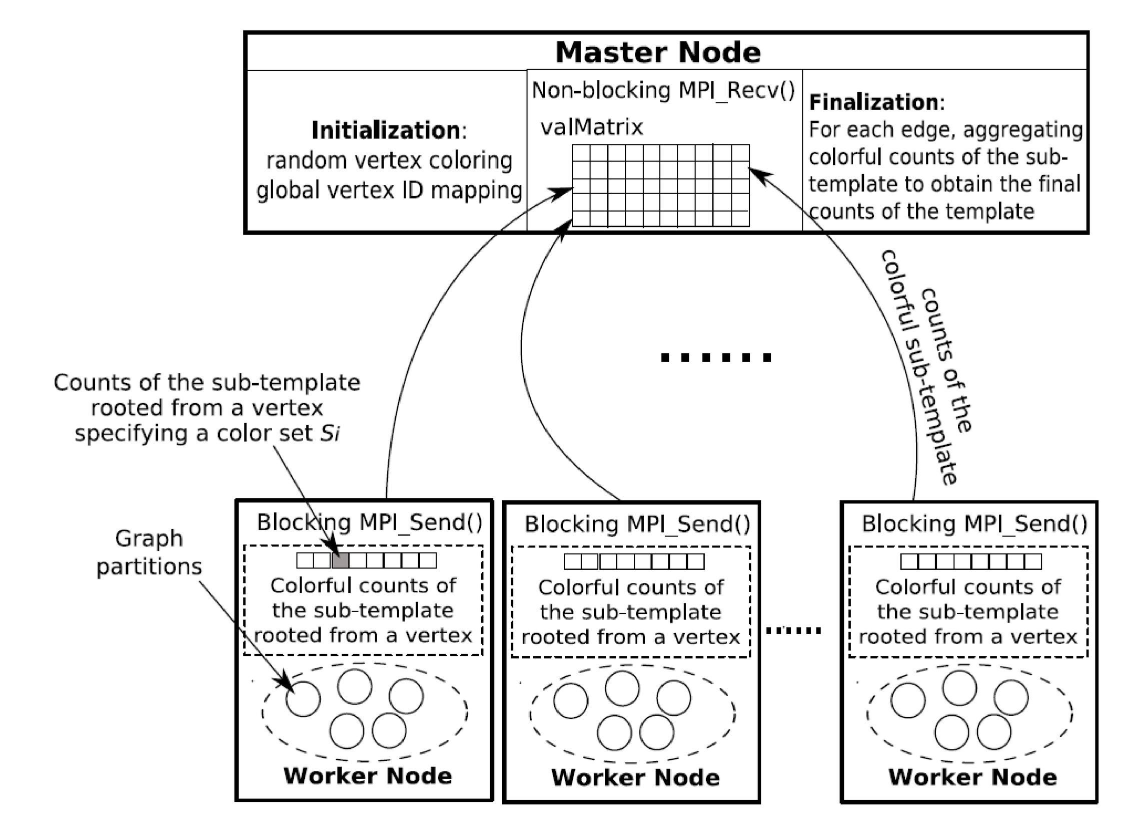


Figure 3: A schematic description of ParSE

### ✓ Cover-based Graph Partitioning

*Several notations:*

- $G_p(V_p, E_p)$: Graph partition.
- $N_r(v)$: $N_r(v) = \{u : d(u, v) \leq r\}$, where $d(u, v)$ is the distance between $u$ and $v$.
- $core(G_p)$: $core(G_p) = \{v : N_r(v) \subset V_p\}$

❖ $G$ is partitioned to a number of $G_p$ s.t.:

$$i)\ \bigcup_{1 \leq p \leq P} core(G_p) = V$$

$$ii)\ \forall p_1 \neq p_2,\ core(G_{p_1}) \cap core(G_{p_2}) = \phi$$

❖ We let $r$ equal to the radius of the $T_i$, so that the counting of the sub-template rooted from each vertex in $core(G_p)$ can be done locally in $G_p$.

### ✓ Template Enumeration

**Goal:** The process of counting the number of colorful sub-template embeddings rooted from each vertex $v \in core(G_p)$, i.e., $C(v, \rho(T_i), T_i, S_i)$, is shown in Fig. 4.

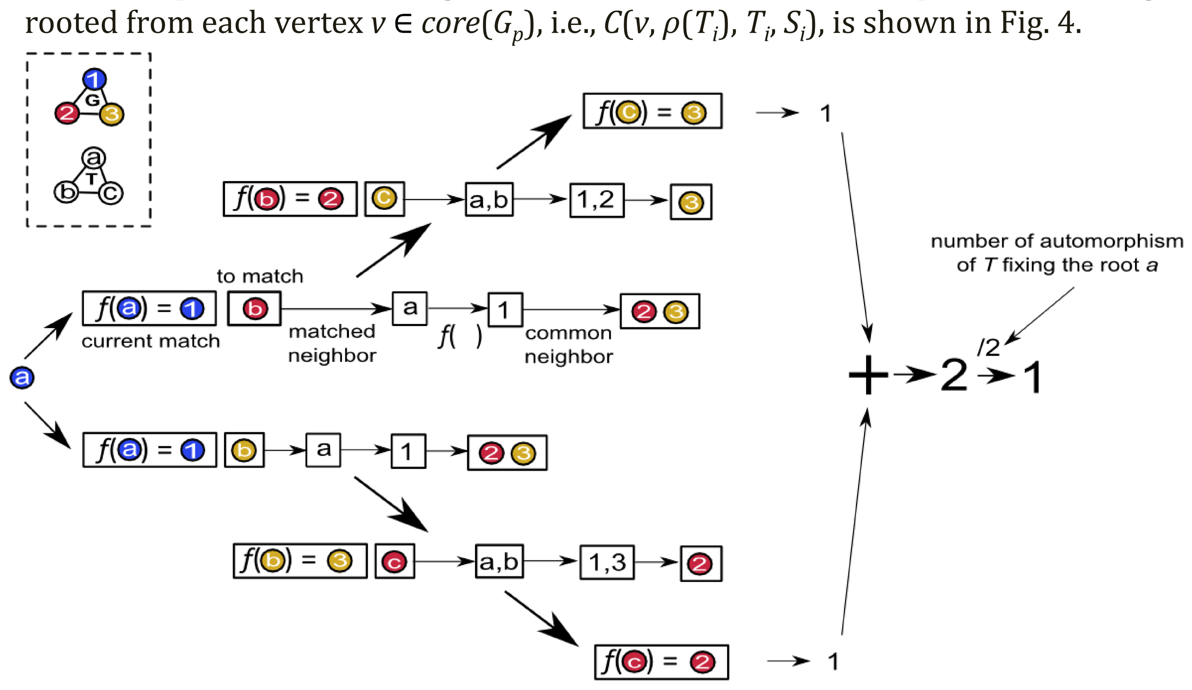


Figure 4: An example of colorful template counting

### ✓Running Time

The total running time of ParSE can be bounded by:

$$O\left(\frac{e^k \log 1/\delta}{\epsilon^2}\left(\frac{n}{Q}\Delta^{k'} + (n+m)k^{k'}\right)\right)$$

❖ Here $P$ is the number of partitions, $Q$ is the number of processors, $k'$ is $max(|T_1|, (|T_2|)$. And we suppose $rP/Q < k^{k'}$.

## Experiments

We perform the experiments using the following graphs and templates:

| | Graph | Number of Nodes | Average Degree |
|---|---|---|---|
| Synthetic Social Contact Networks | NRV | 151,783 | 164 |
| | Miami | 2,092,147 | 50 |
| Random $G(n,p)$ graph | GNP50 | 50,000 | 20 |
| | GNP100 | 100,000 | 20 |

Figure 5: Datasets used in the experiment.

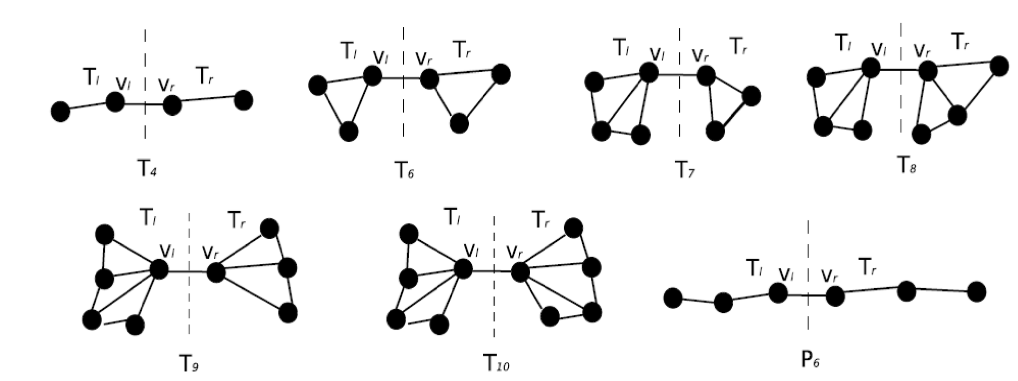

Figure 6: Templates used in the experiment.

### ✓ Approximation Quality

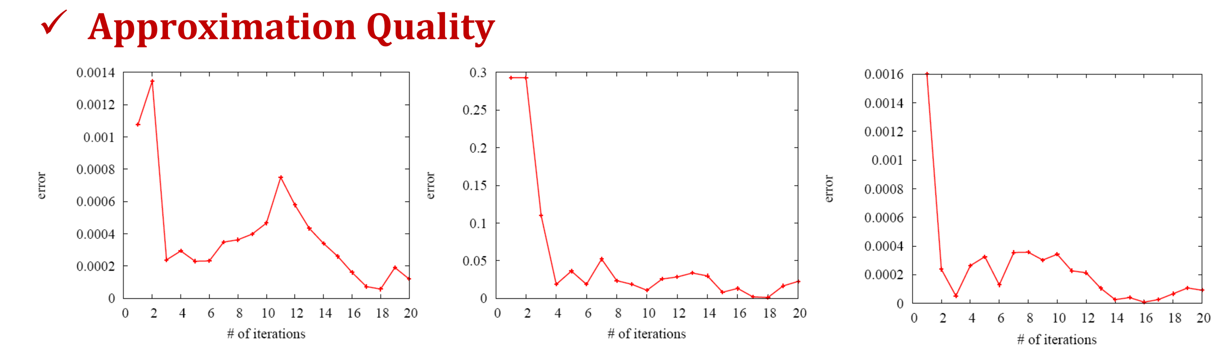

Figure 7: Error for counting $T_4$ on GNP100, $T_6$ on GNP100, and $T_4$ on NRV, from left to right.

### ✓ Speed up over Huffner's Sequential Color-coding

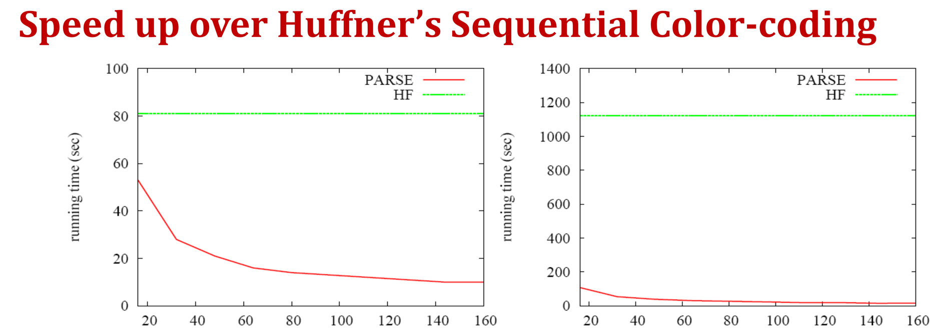


Figure 8: Running time for $T_4$ and $P_6$, conducted on GNP50.

### ✓ Time Cost of Various Steps of ParSE

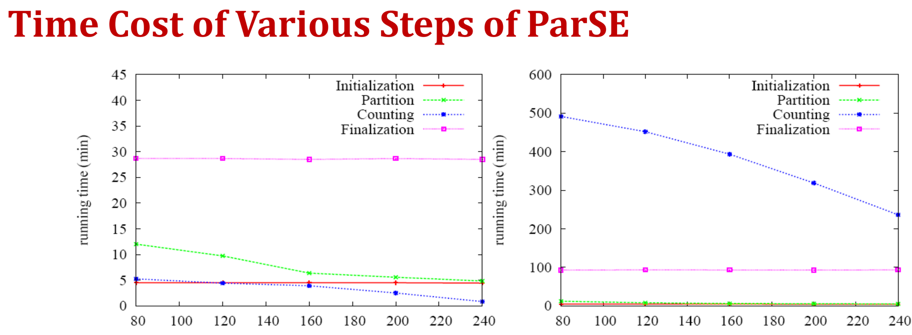


Figure 9: Time usage on various steps for $T_4$ and $T_6$, on NRV.

### ✓ Scaling of ParSE

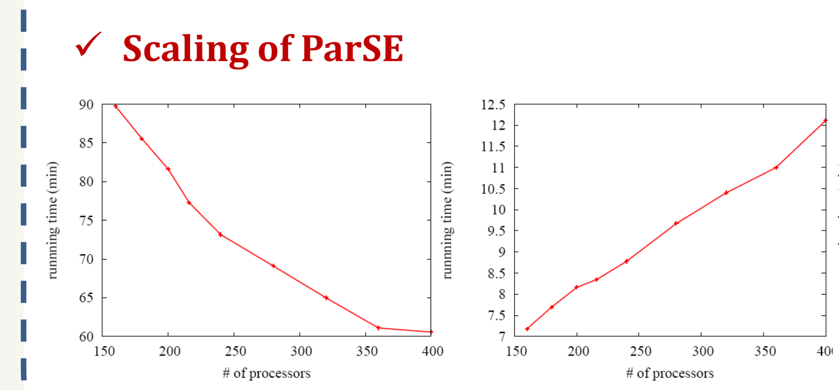

Figure 10: Strong and weak scaling on Miami.

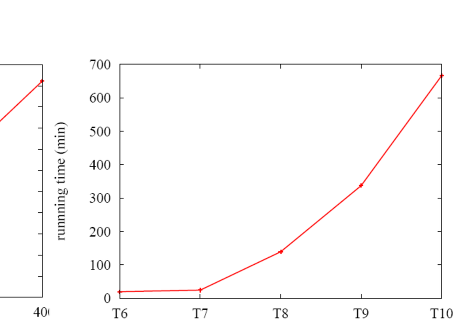

Figure 11: Time VS. Template on NRV

This work has been partially supported by NSF Nets Grant CNS-0626964, NSF HSD Grant SES-0729441, NIH MIDAS project 2U01GM070694-7, NSF PetaApps Grant OCI-0904844, DTRA R&D Grant HDTRA1-0901-0017, DTRA CNIMS Grant HDTRA1-07-C-0113, NSF NETS CNS-0831633, NSF CAREER 0845700, DHS 4112-31805, DOE DE-SC0003957 and NIH/CDC 1P01CD000284-01.