FASCIA: Fast Approximate Subgraph Counting and Enumeration

George M. Slota and Kamesh Madduri

Department of Computer Science and Engineering
The Pennsylvania State University

Abstract

We present a new shared-memory parallel implementation of Alon et al.'s color-coding technique called FASCIA for the problems of approximate subgraph counting and subgraph enumeration.

- Subgraph counting is used in multiple domains, inlcuding bioinformatics, chemoinformatics, social network analysis, among many others
- We present multiple algorithmic improvements to the baseline color-coding technique for subgraph counting targeted at improving runtime, parallelization, and memory usage
- **Our method allows real-time count estimates of subgraphs up to seven vertices on networks with tens of millions of edges. Count estimates of up to twelve vertice subgraphs are obtained in minutes.**

Color-Coding Technique

Color-coding allows an approximation algorithm for subgraph counting that runs in about $O(m \cdot 2^k \cdot e^k \cdot)$ instead of $O(n^k)$ for the exact naïve algorithm, where m is the number of edges in the graph, n is the number of vertices, and k is the number of vertices in the subgraph.
The procedure for the algorithm is as follows:

- Randomly color every vertice in the graph with at least k colors
- Use a dynamic programming scheme to count the number of *colorful* subgraph embeddings, where colorful means each subgraph vertice has a distinct color
- Scale this count by the probability that any given embedding will be colorful
- Repeat the first three steps some number of times
- Average all determined counts and output result

Acknowledgements

Special thanks to the NSF, XSEDE, SDSC, PSU RCC

Test Environment

Reported runtimes were retrieved from a single node of Gordon at the San Diego Supercomputing Center. The graphs used came from the SNAP database, Virginia Tech Network Dynamics and Simulation Science Laboratory.

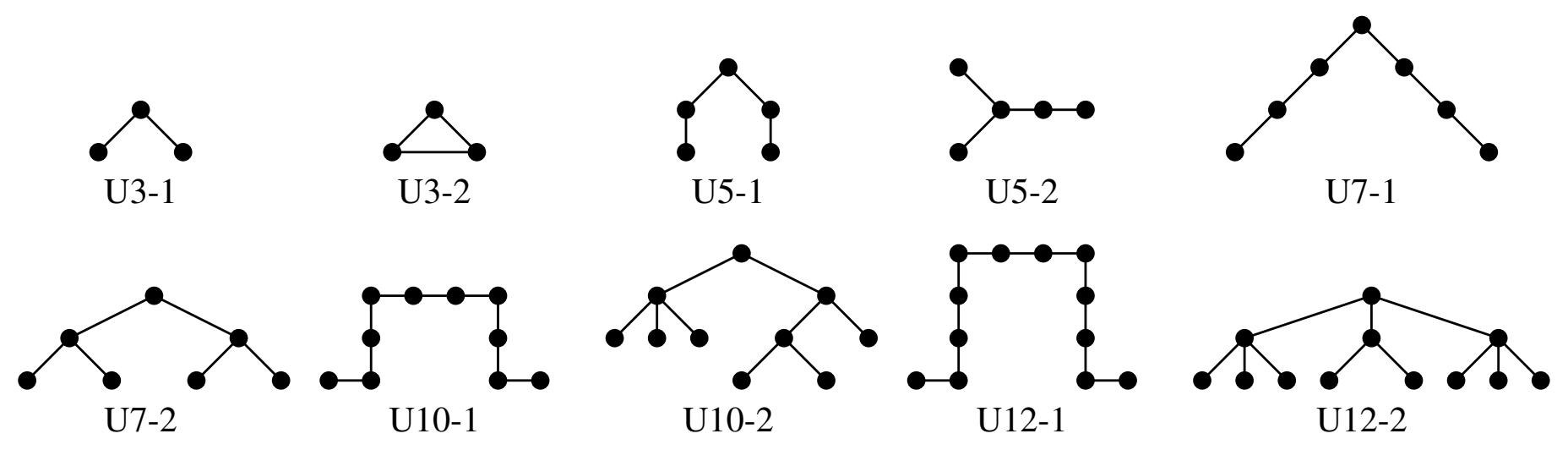

Figure 1: Templates used in experiments

Algorithmic Optimizations

- Multiple parallelization strategies
- Representation of subgraph colorings as single integers using a combinatorial indexing system
- Pre-computation of complex operations stored in cache-resident table
- Template partitioning scheme that allows up to $\frac{k-1}{k}$ reduction in total work performed

Memory Optimizations

- Smart storage and dynamic programming table initialization
- Careful template partitioning and organization
- Fast hash table which exploits random graph coloring to reduce collisions
- **These memory optimizations can demonstrate up to a 90% reduction in memory usage over the naïve approach**

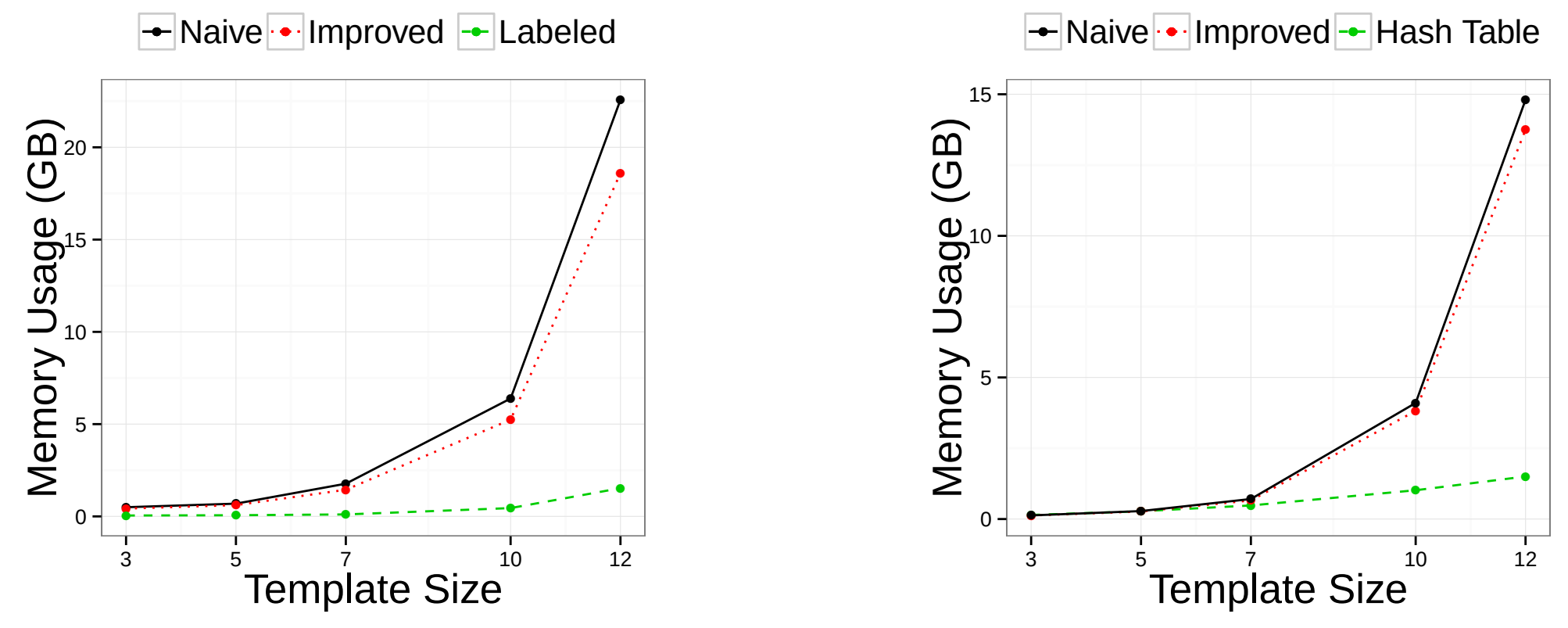

Figure 2: Memory usage of the naïve and improved table and hash table on the (a) Portland and (b) PA road networks

Runtimes and Parallel Scaling

Figure 3 demonstrates how FASCIA can produce real-time count estimates for templates up to 7 nodes in size on a large 33 million edge network. Template up to 12 nodes complete in a matter of minutes.

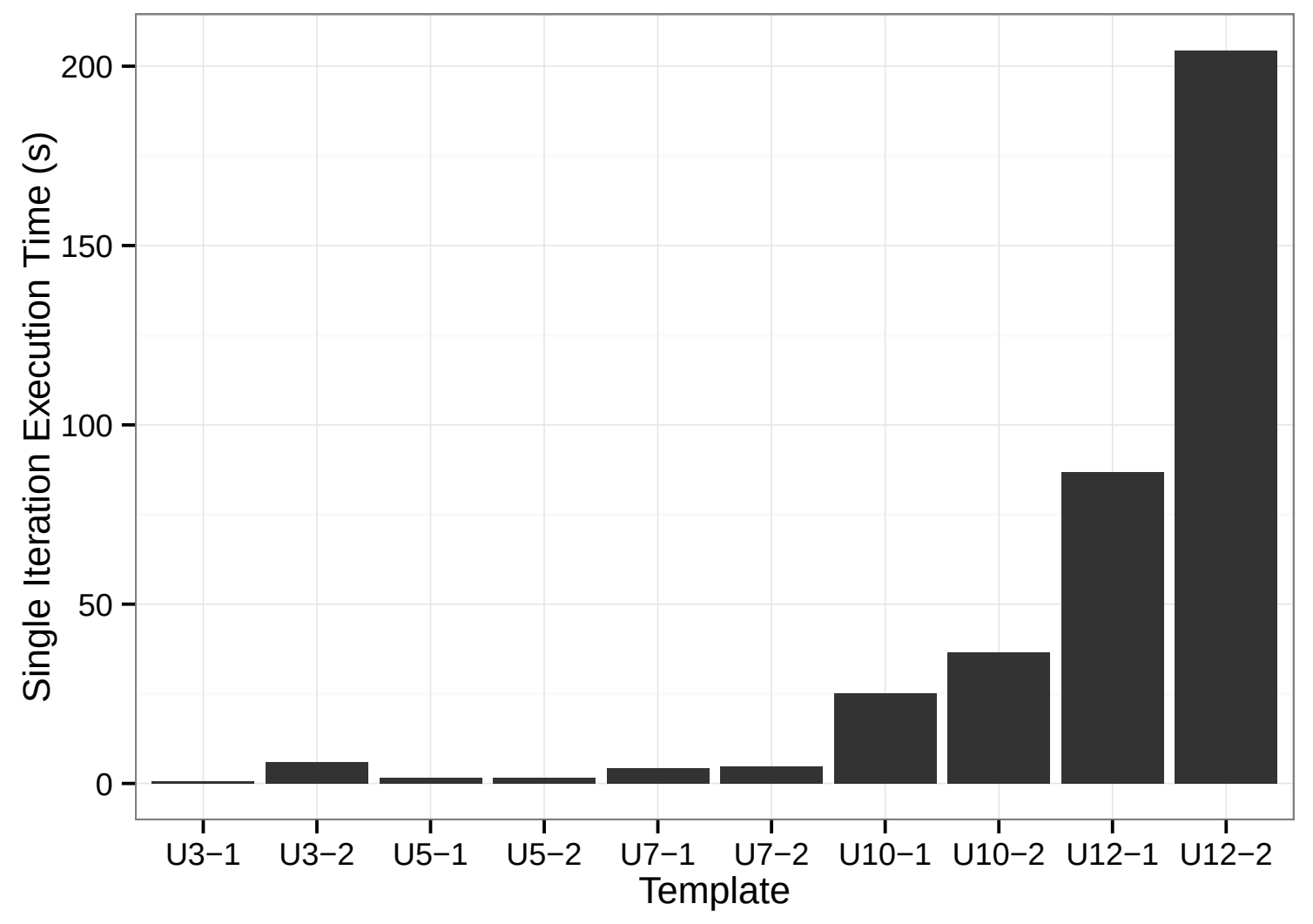

Figure 3: Runtimes on Portland network (n=1.6M, m=33M)

Using a 12 node template on a large network, we achieve 12× parallel speedup. On a smaller network we can parallelize multiple simultaneous counts, and achieve over a 6.5× parallel speedup.

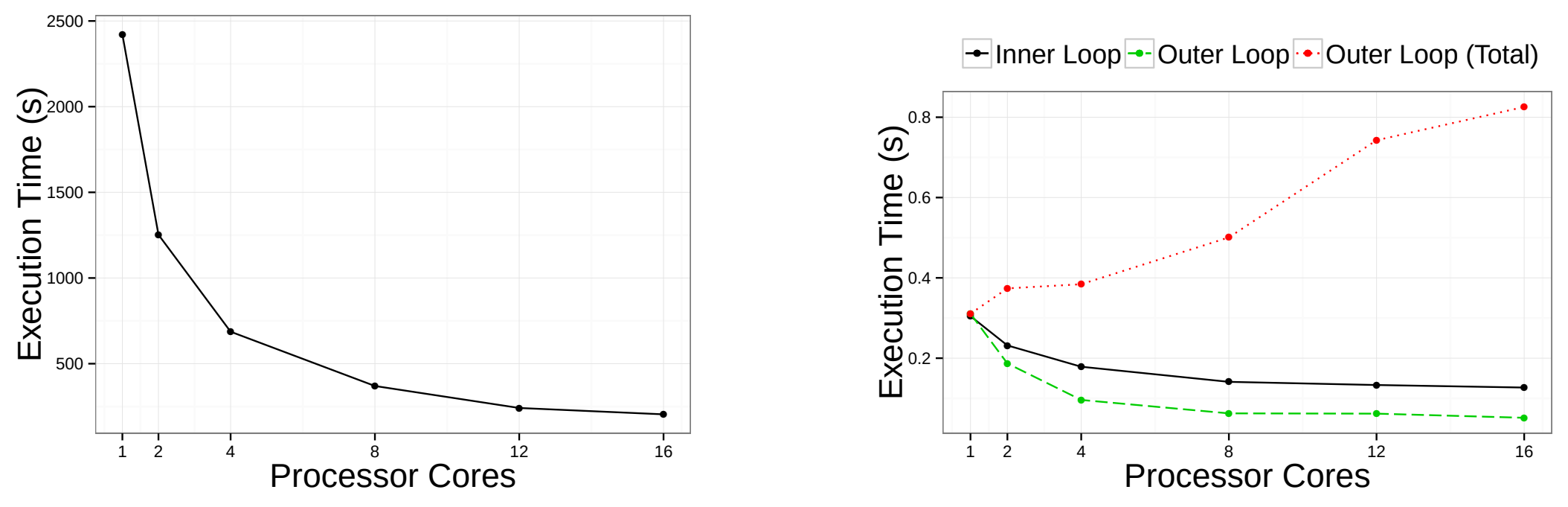

Figure 4: Parallel scaling of (a) inner loop with U12-2 template on Portland and (b) both inner and outer loop with U7-2 template on Enron email network (n=33K, m=180K)

Error Bounds

Testing demonstrates that for even a modest sized network, error is very small after few iterations. Error decreases with increasing network size but increase with increasing template size.

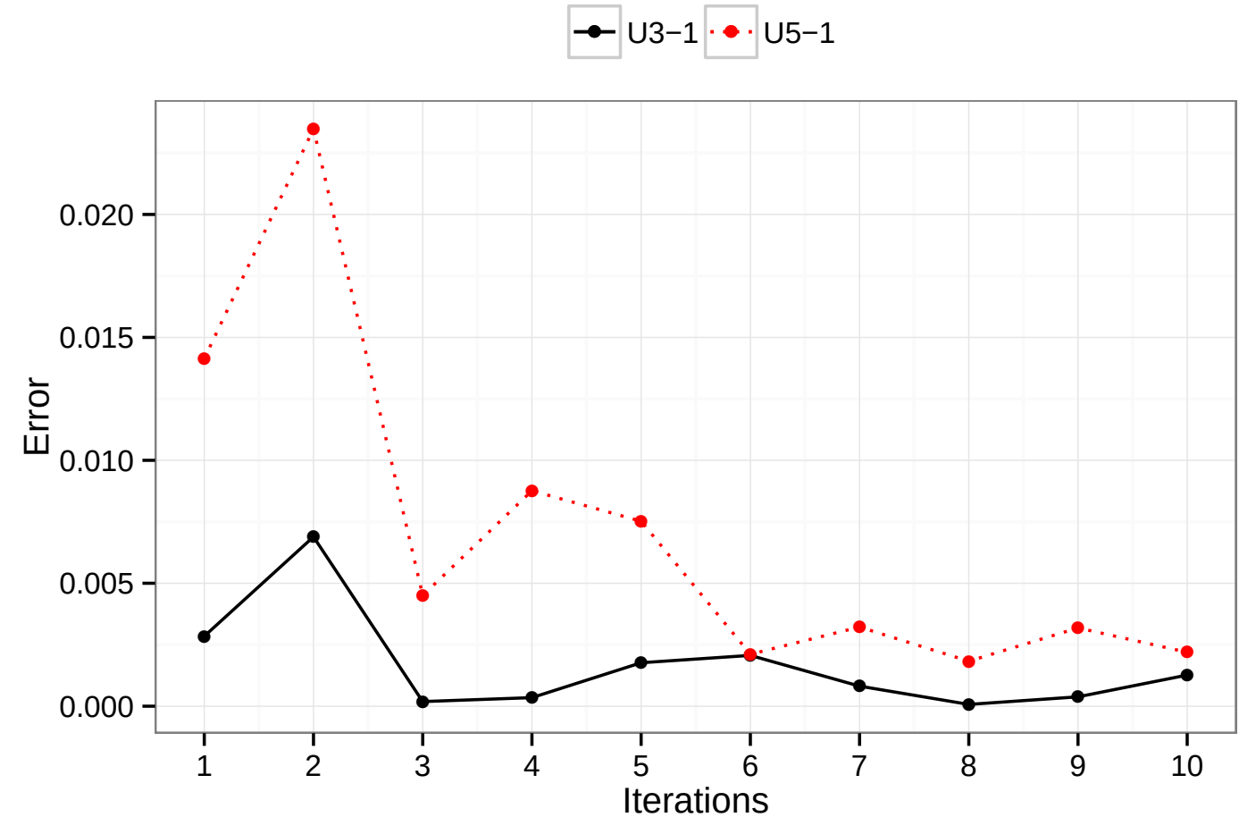

Figure 5: Error on the Enron email network

Motif Finding

Motif finding is determining frequently occuring subgraphs by comparing counts between networks, and has found importance within the field of bioinformatics. Below we demonstrate counts for all 11 possible 7 vertice templates on 4 biological networks. All counts were calculated in a few minutes with under 1% error.

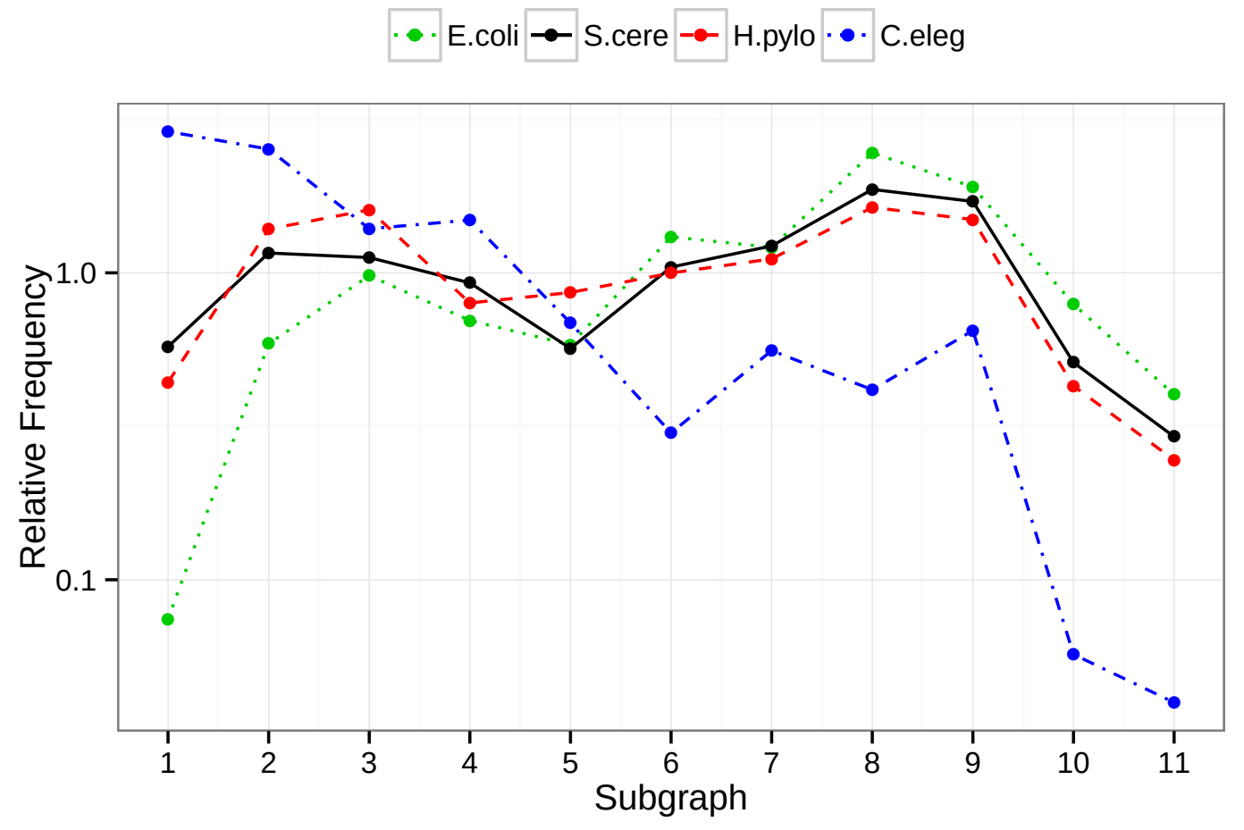

Figure 6: Motif finding on E. coli, S. cerevisiae, H. pylori, and C. elegans

Graphlet Degree Distributions

Similar to a regular degree distribution, a graphlet degree distribution is the number of vertices have a distinct number of subgraph embeddings. This distribution can be used to compare networks on the basis of local structural similatiry.

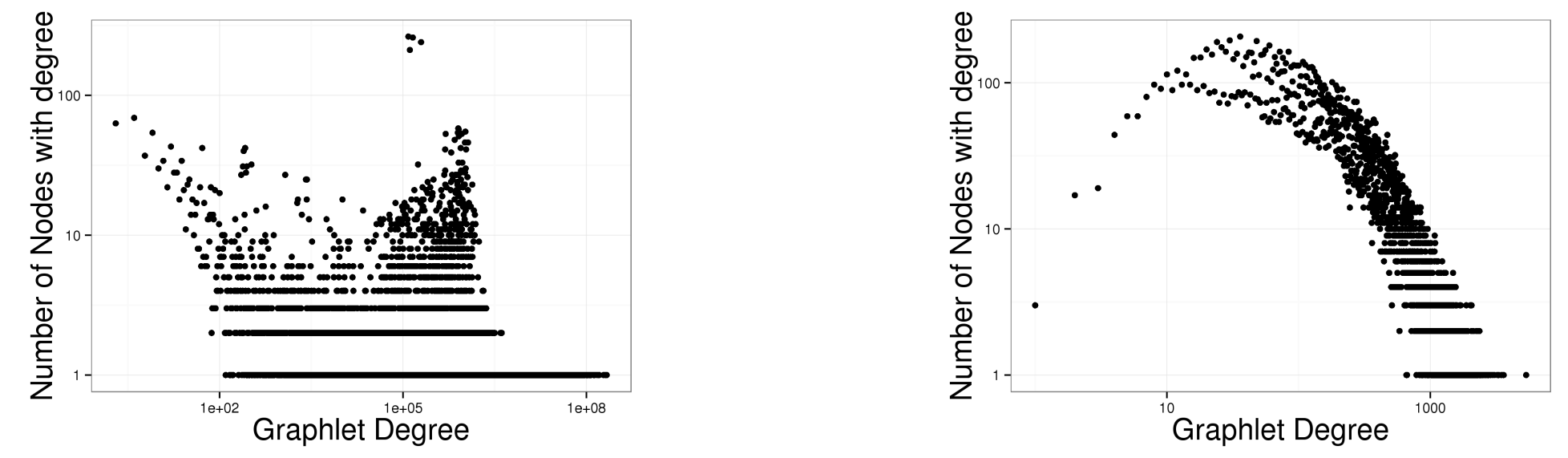

Figure 7: Graphlet degree distribution for template U5-2 on the Enron network and a random $G(n,p)$ graph of the same size and average degree

Conclusions

- Through algorithmic optimizations, FASCIA achieves considerable speedup compared to prior work while reducing memory consumption and parallel overhead
- FASCIA is especially useful for motif finding and calculating graphlet degree distributions