Implementation of High Performance Computing on Network Analysis

Xin-Zeng Wu*, Aiichiro Nakano†‡

*Information Sciences Institute, University of Southern California, Marina del Rey, California 90292-6611
†Department of Computer Science, University of Southern California, Los Angeles, California 90089-0781
‡Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089-0484
{xinzengw, anakano}@usc.edu

Abstract—Network science research area has a large demand of analyzing graphs with large number of nodes and edges. The single CPU computer has limitations on the computing source and storage space. However, the network structure distributions are discrete and the mathematical models often suffer from these discretization when the discretization gap are large. To minimize these errors, larger networks have to be considered. This is the point for the parallel computing to get involved.

I. INTRODUCTION

Complex systems can be represented as networks of locally-interacting entities or nodes. Local interactions among nodes in a complex network can lead to an astounding array of global behaviors. Examples include viral outbreaks and social contagions in social networks, cascading failures in the power grid and financial networks, synchronization of coupled oscillators, opinion dynamics and consensus formation in human groups.

A node’s local view of network structure, however, may be systematically different from the global ground truth, which may affect global phenomena. Social scientists have identified one source of bias—the friendship paradox—which states that, on average, nodes have fewer connections, or smaller degree, than their neighbors. Recently, more interesting variations of the paradox were discovered. The strong friendship paradox states that most nodes have fewer connections than most of their neighbors. Unlike the original friendship paradox and its generalizations to attributes other than degree, the strong friendship paradox does not arise trivially as a result of sampling from skewed distributions.

We also investigated the strength of the strong friendship paradox microscopically to predict the amount of information distortion a node with a given connectivity experiences. Our recent work shows that the mathematical models for predicting the strength of the strong friendship paradox require a measure beyond the connectivity preference between two nodes mentioned above.

In this paper, we will focus on parallelizing the simulation of the generalized friendship paradox, and we may also parallelize our theoretical model calculation if possible.

II. METHODS

The simulation procedure of generalized friendship paradox is done by randomly selected node and assign it to be active on a given network or graph. Then, we can count the number of nodes with over a half of its neighbor are activated, these nodes are observing the paradox. When the graph becomes large, this calculation becomes heavy and slow. To parallelize the algorithm, we will add the following modification on the procedure.

- Break the full graph into separate subnetworks, each subnetwork should be in similar size and handles by different processors.
- An algorithm will determine the clustering behavior and community structure in the network to minimize the inter-subnetwork edge. This will also reduce the information passing time.
- The inter-subnetwork edges will be stored in the adjacency list.
- Randomly activate a certain percentage of node (Assign the attribute of the node from 0 to 1)
- The activation of nodes is then be observed by the nodes connected to them.
- The inter-subnetwork information passing is handles by querying the adjacency lists.
- Each core reports the number of nodes that observe the paradox.
- Combine the results from each core to the final result.

III. EXPECTED RESULTS

In this project we expect to reproduce the results in [1] and [2] by using parallel computing techniques in HPC. The divide an conquer algorithm will be implemented by rewriting the program that was used in the papers. Similar result graphs will be generated, and the time order of the two algorithms will be compared.

ACKNOWLEDGMENT

This research was supported by DARPA and ISI. Computation for the work described in this paper was supported by the University of Southern California’s Center for High-Performance Computing.

REFERENCES