Towards Scalable Graph Analytics on Time Dependent Graphs*

Suraj Poudel
Department of Computer and Information Sciences
University Of Alabama at Birmingham
spoudel@uab.edu

Roger Pearce
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
rpearce@llnl.gov

Maya Gokhale
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
maya@llnl.gov

1. INTRODUCTION

We present an approach to model distributed time-dependent graphs by annotating partitioned graph topology with temporal metadata, and analyze the scalability of our approach on an HPC system.

Edge parameters in time dependent graphs vary as a function of time. Graph analytics in such context takes into consideration the fact that the actual view of the graph changes with time. As an example, single source shortest path (SSSP) in time dependent graphs analyses optimal arrival time with varying start time of graph traversal and/or varying waiting time at each node during traversal. Analytics in time dependent graphs have been applied mostly sequentially in various applications[1, 2]. In this work, we evaluate the graph analytics in large distributed graphs on HPC systems.

This work shows a) an increased capability of extended HavoqGT to allow time dependent graph analytics and b) a preliminary scalability study of the framework with various use cases.

2. TEMPORAL METADATA

The Center for Applied Internet Data Analysis (CAIDA)[3] datasets are representative of a time-varying graph of communicating vertices (IPs). It consists of more than 8TB of publicly available anonymized passive traffic traces and packet header traces. All our experiments are conducted with an 1-hour capture period, roughly 190GB of data. Packets were abstracted as flows based on aggregating packets within 0.0001 sec into a single flow. Using this approach on our experimental data, we accumulated nearly 1.35 billion flows which represents a time-dependent graph.

*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. (LLNL-ABS-676013).

3. EDGE ANNOTATION

The edge metadata are annotated using a two pass approach: 1) ingest the links between two communicating IP's to form a partitioned-graph topology [4]; 2) annotate the edges in the partitioned-graph topology with the corresponding metadata (flow data). Step 1 uses the *delegate* partitioning with asynchronous visitor model to create the partitioned-graph topology.

In Step 2, the asynchronous visitor model[5] in HavoqGT pushes the visitor containing locally read metadata (flow data) to the partition owning the source ip (vertex) of the flow (edge) and annotates the edge. HavoqGT broadcasts edge metadata to delegates in case of hub vertices and registers the metadata to corresponding edge.

On completion of this process, the partitioned-metadata resides in conjunction with the partitioned-graph topology. Also, the partitioned temporal-metadata complements the asynchronous visitor model to model analytics on distributed time-dependent graphs. We successfully created graph and annotated edges with 1.35 billion flows on a Linux HPC Cluster at LLNL.

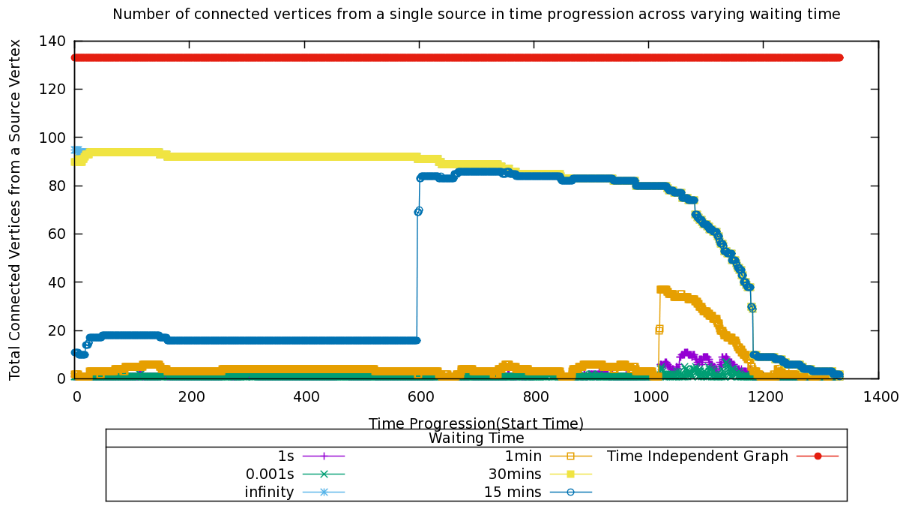
4. TIME-DEPENDENT GRAPH MODEL

Multiple temporal-metadata per edge models a time dependent graph. Let $M\left(t_s, t_e\right)$ be the set of metadata such that $t_s \leq \mathit{start\_time}(m_i \in M\left(t_s, t_e\right)) \leq t_e$ and $E\left(t_s, t_e\right)$ be the set of edges that has been annotated with $m_i \in M\left(t_s, t_e\right)$, then $G\langle t_s, t_e\rangle\left(V, E\left(t_s, t_e\right)\right)$ represents a graph in time interval (t_s, t_e). So, different set of graph topology with differently annotated edges can be seen for $0 \leq t_s < t_e$.

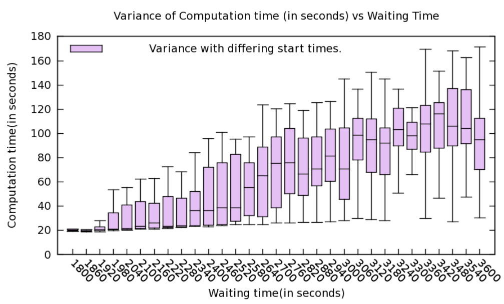
5. EXPERIMENTAL RESULTS

The connectivity of time-dependent graphs can vary greatly as a function of the *start* time and *waiting* time; the changes in connectivity for a CAIDA graph is shown in Figure (a). In contrast to the time flattened graph, time dependent graph captures dynamic nature of the real world graph. The variance seen in Figure (a) propagates further to higher level algorithms such as time-dependent betweenness centrality, as shown by Figure (b).

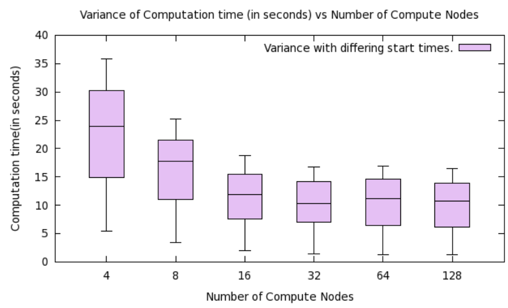
The variability of the connectivity correlates to the computation time as shown in Figure (b). For each waiting-time value, different starting-times are shown by the variance bars. As the waiting-time is increased, the traversal com-



(a) Plot of number of connected vertices from a single source with respect to the change in time at varying waiting time. Traversal from an example source in aggregate version of this graph has 133 connected vertices while in time dependent graph shows significant variation in its connectivity.



(b) Plot of variance of the computation time of exact betweenness centrality for start time within first 20 minutes of the test data with respect to waiting time period from half-an-hour to the full hour.



(c) Plot of variance of computation time across various start time with respect to the number of compute nodes used to process the time dependent exact betweenness centrality.

plexity increases as the number of eligible paths increases.

Finally, Figure (c) shows the result of strong scaling of the betweenness centrality with an infinite waiting time (infinite waiting time yields a larger workload, ideal for scalability tests). The computation time decreases with an increasing number of compute nodes, and scales near-linearly up to approximately 32 compute nodes. The strong-scaling limit after 32 nodes is not uncommon for distributed graph analytics.

6. CONCLUSION

We present an approach for modeling time-dependent graphs in distributed memory for HPC systems, by extending the capabilities of HavoqGT to represent temporal edge metadata. To evaluate our approach, we implemented temporal versions of SSSP and Betweenness Centrality. We demonstrate near-linear strong-scaling on a HPC Linux cluster up to 32 compute nodes, using real temporal CAIDA internet connectivity data.

7. REFERENCES

[1] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," *Journal of the ACM (JACM)*, vol. 37, no. 3, pp. 607–625, 1990.

[2] A. K. Ziliaskopoulos and H. S. Mahmassani, "Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications," *Transportation research record*, pp. 94–94, 1993.

[3] CAIDA, "The CAIDA UCSD Anonymized Internet Traces," 2008.

[4] R. Pearce, M. Gokhale, and N. M. Amato, "Faster parallel traversal of scale free graphs at extreme scale with vertex delegates," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, (Piscataway, NJ, USA), pp. 549–559, IEEE Press, 2014.

[5] R. Pearce, M. Gokhale, and N. M. Amato, "Multithreaded asynchronous graph traversal for in-memory and semi-external memory," in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '10, (Washington, DC, USA), pp. 1–11, IEEE Computer Society, 2010.