Research Report:
Scaling Computation of Graph Structured Data with NScale

Konstantinos Xirogiannopoulos, Stephen Mark Herwig, Bor-Chun Chen, Mingchao Shao
University of Maryland, College Park
{kostasxirog91,smherwig,bcsiriuschen,Shaomc}@gmail.com

February 23, 2015

In this day and age, the already vast amounts of data being generated that we have to deal with are still increasing in size by the second. The ”Big Data” buzzword keeps becoming more relevant not only in computer science but in nearly all sciences, and with good reason. The more data in a specific domain increases in size the more valuable it is considered. There may exist incredibly useful insight in the data that remains untapped until analyzed. Researchers in the field of Database Systems have been on the hunt for a fast, efficient, and scalable way we can analyze very large volumes of data.

Structuring interconnected data for scalable analysis

Data can be structured in various ways, always depending on the use case. One of the ways that has been gaining popularity is structuring data as a property graph of nodes connected by edges. The property graph holds many advantages over other schematic ways of representing data. If you think about it, nearly everything in the real world can intuitively be seen as a graph of objects (nodes) that interact, relate to, or connect with other nodes (via edges). A few examples include social networks, citation networks, biological networks, IP traffic networks. Graphs provide adjacency of data without the need for indices. This is part of the reason social networks have adapted graphs into their systems, as querying them is significantly more efficient for specific types of queries, like finding the friends of the friends of a specific user.

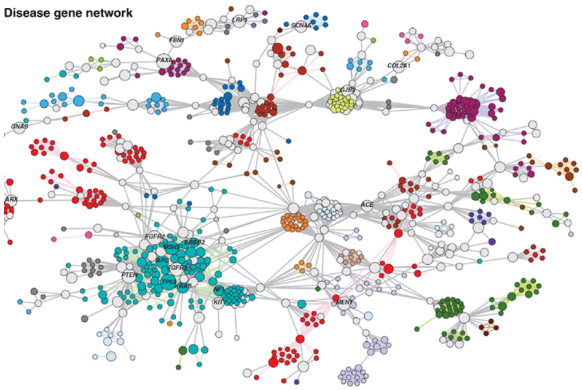

Figure 1: Many kinds of networks. In the above network, the nodes are genes, and genes are connected if they are implicated in the same disorder.[4]

Graphs are easily comprehensible and provide a very representative model of the real world. This has led researchers to investigate ways of obtaining deep insights from large-scale graph structured data. Led by Google’s work on Pregel [2], researchers have been looking into graph computation frameworks [1, 2, 3], that would exploit the index-free adjacency provided by graphs for scalable large-scale analytics.

NScale

We are going to talk about a project underway here at the University of Maryland, called NScale[6, 7], a processing framework for analyzing large graphs in a distributed fashion. NScale is work led by PhD candidate Abdul Quamar and faculty Professors Amol Deshpande and Jimmy Lin.
Analytics frameworks like the aforementioned ones are what is called "vertex-centric" as the algorithms that can be written and executed in these frameworks start from a single node (vertex) and by using either message-passing techniques or shared memory, are able to converge to apply computations on the data stored as properties in the graph. What sets NScale apart is that it uses a "subgraph-centric" approach to graph processing. So instead of a node being the center of a computation, now we have a smaller part of the whole graph being the main protagonist of the computation. This allows for optimizations that simply allow us to leverage larger amounts of data for graph analytics.

Using NScale, users have the ability to apply computations only on select sub-portions of a graph, and therefore scale neighborhood-centric analytics to graphs larger than could be analyzed using a vertex-centric model.

Why subgraph-centric?

A wide variety of analytical procedures that are typically applied to graph data have to do with finding patterns in the relationships inside the graph, and doing so by analyzing the relationships (subgraphs) that are happening in close proximity to a vertex.
An example of such analytics includes finding Local Clustering Coefficients (LCC). In the Social Network example, this translates into finding the set of friends of a user, who are also friends. Another example would be identifying and clustering the social circles of a particular node, as subgraphs. More examples include social recommendations using PageRank and neighboring subgraphs, or counting motifs of subgraphs that appear many times.

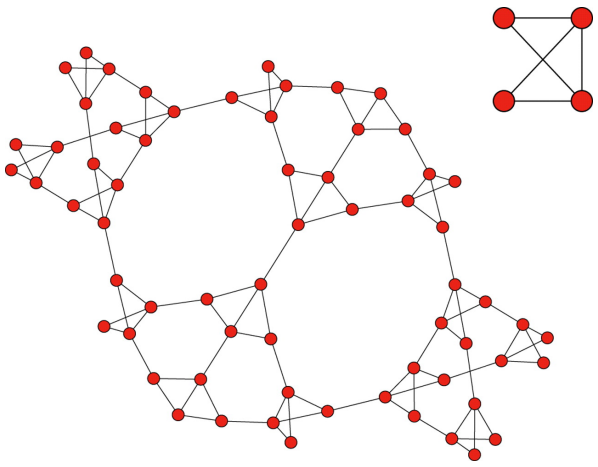

Figure 2: Recurrent subgraph (motif) in larger network [5]

To conclude, NScale will allow users to specify subgraphs or neighborhoods as the scope of computation, which entails scalable analytics on very large graphs. Being able to analyze graphs

realistic to today's volume and velocity, and being able to effectively extract insight from them will open up great opportunities for more useful analytics. Such insight can have a substantial impact on any organization, as well as open new horizons in the research field of data management systems.

References:

[1] Avery, Ching. "Giraph: Large-scale graph processing infrastructure on hadoop." Proceedings of the Hadoop Summit. Santa Clara (2011).

[2] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

[3] Low, Yucheng, et al. "Graphlab: A new framework for parallel machine learning." arXiv preprint arXiv:1408.2041 (2014).

[4] Goh, Cusick, et. al. "The human disease network."
Proceedings of the National Academy of Sciences of the United States (2007)

[5] Simon DeDeo, David C. Krakauer "Dynamics and processing in finite self-similar networks." J.R. Soc. Interface: 10.1098 (2012)

[6] Abdul Quamar, Amol Deshpande, and Jimmy Lin. "NScale: Neighborhood-centric largescale graph analytics in the cloud" (2014)

[7] Abdul Quamar, Amol Deshpande, and Jimmy Lin. "NScale: Neighborhood-centric Analytics on Large Graphs" VLDB 2014