International Journal of Computer Applications Technology and Research
Volume 3– Issue 7, 457 - 459, 2014

A Dependent Set Based Approach for Large Graph Analysis

Shital Deshmukh
University of Pune,
KKWIEER,
Nashik, India.

S. M. Kamalapur
University of Pune,
KKWIEER,
Nashik, India

Abstract: Now a day's social or computer networks produced graphs of thousands of nodes & millions of edges. Such Large graphs are used to store and represent information. As it is a complex data structure it requires extra processing. Partitioning or clustering methods are used to decompose a large graph. In this paper dependent set based graph partitioning approach is proposed which decomposes a large graph into sub graphs. It creates uniform partitions with very few edge cuts. It also prevents the loss of information. The work also focuses on an approach that handles dynamic updation in a large graph and represents a large graph in abstract form.

Keywords: Clustering, Graph Partitioning, Large Graph, Sub Graph.

1. INTRODUCTION

Large graph is one which consists of hundreds to thousands of nodes and millions of edges. Web graphs, social networks, recommendation systems are some examples of large graph. As it is a complex data structure such graphs require excessive processing, more memory for storage and knowledge of a pattern of the graph. It is very difficult to comment on exact size and pattern of a large graph as it changes with time. In large graph analysis the first step is to divide the input graph into number of small parts called as sub graph as whole graph cannot fit into memory for processing at given time and second step is graph summarization which finds the strong connected component i.e a node which is connected to maximum nodes in the sub graph. All such components are then used to maintain connection between different sub graphs by using hierarchical representation. For the first step many serial and parallel graph partitioning methods like spectral bisection, multilevel partitioning, and incremental partition are proposed so far.

Graph partitioning problem complexity is NP complete. For any graph partitioning method to be the best or efficient it must answer following questions:

1. What is the threshold value of partition for given graph?
2. How the connection between sub graphs is maintained?

Some algorithms fail to answer both the questions. For example spectral bisection method produces excellent partitions but connection between sub graphs is difficult to maintain as it is matrix based approach and partitions are stored in matrix form, Multilevel partitioning method is a K –way partitioning method which does not provide threshold value for number of partitions to be produced. The proposed method focuses on both the aspects i.e threshold value and connection between sub graphs.

For the second step, CEPS summarization method is commonly used which uses random walk with restart concept to find connected component/vertex of a graph but it's a matrix based approach so it is not scalable for large graph. The proposed graph partitioning method calculates this connected component/vertex while producing partitions of a large graph. One more issue in large graph analysis is graph size changes with time because information in social network, web graphs which best explains large graphs changes with time. So, dynamic updation like addition or deletion of information in produced sub graphs should also be handled.

So proposed system focuses on implementation of two methods one is graph partitioning and other for dynamic updations in large graph.

Section 2 focuses on literature review, section 3 explains block diagram, algorithms of the proposed approaches, data sets for the proposed method and results are briefed in section 4, and section 5 concludes the paper.

2. LITERATURE REVIEW

This chapter covers related work done on large graph analysis i.e different graph partitioning methods.

2.1 Graph Partitioning Methods

The graph partition problem is , Let graph G = (V, E), with V vertices and E edges, it is possible to form sub graphs or partitions of G into smaller components with some properties also called as k-way partitioning which divides the vertex set into k smaller components or sub graphs. A good partition is one in which the number of edge cuts are less and uniform graph partition is one which divides graph into equal size sub graphs.

Spectral bisection partitioning [8] method is a matrix based approach in which for a given a graph with adjacency matrix A, where A_{ij} gives an edge between node i and j, and Degree matrix D, is a diagonal matrix, in which each diagonal entry of a row i, d_{ii}, represents the degree of node i. The Laplacian of matrix L is defined as $L = D – A$, then a partition for graph $G = (V, E)$ is defined as a partition of set V into disjoint sets U, and W, such that cost of cut $(U, W)/ (|U|\cdot|W|)$ is minimum. The second smallest eigenvalue (λ) of L gives a lower bound of the optimal cost (c) of partition where $c \geq \lambda/n$.

The eigenvector (*V*) corresponding to *λ*, which is called as Fiedler vector, bisects the graph into only two sub graphs based on the sign of the corresponding vector entry. To do the division into a larger number of sub graphs is usually achieved by repeated bisection, but this does not always give satisfactory results which is a drawback of the method also minimum cut partitioning fails when the number of sub graphs to be formed, or partition sizes are unknown.

Multilevel partitioning method is analogous to multigrid method to solve numerical problems. Karypis and Kumar has proposed k-way graph partitioning known as METIS [4] which is based on multilevel partitioning in which the proposed method reduces the size of the graph by collapsing vertices and edges, partitions the graph into smaller graph, and then uncoarsen it to construct a partition for the original graph. The drawback is the graph partitions are stored in adjacency matrix, as it uses static data structure to store partitions node or edge addition or deletion in sub graphs (partitions) at run time is not possible.

To execute several scientific and engineering applications parallel, requires the partitioning of data or among processors to balance computational load on each node with minimum communication. To achieve this parallel graph partitioning there are many algorithms like geometric, structural, spectral & refinement algorithms are proposed. One of such method is parallel incremental graph partitioning [2] in which recursive spectral bisection-based method is used for the partitioning of the graph which needs to be updated as the graph changes over time i.e a small number of nodes or edges may be added or deleted at any given instant. The drawback of the method is initial partition is to be calculated using linear programming based bisection method.

The Proposed approach focuses on uniform partitions creation with no loss of information.

3. IMPLEMENTATION DETAILS

3.1 Block Diagram of the System

The following figure explains the Block Diagram of Proposed System:

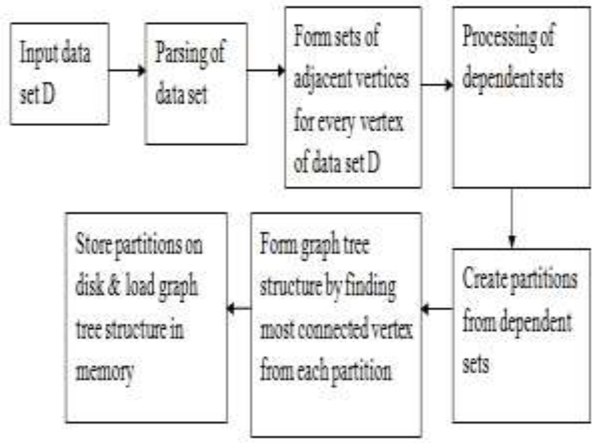
Input data set D
Parsing of data set
Form sets of adjacent vertices for every vertex of data set D
Processing of dependent sets
Create partitions from dependent sets
Form graph tree structure by finding most connected vertex from each partition
Store partitions on disk & load graph tree structure in memory

Fig 1. Block Diagram of Proposed Method

Here the Input to the system is a data set D consisting of Set of connected vertices. The proposed system will directly form sub graphs i.e partitions of input data which is different from previous work.

3.2 Proposed Approach with Example

Dependent Set: For a given vertex dependent set is set of all vertices connected to it.

Consider the following graph G:

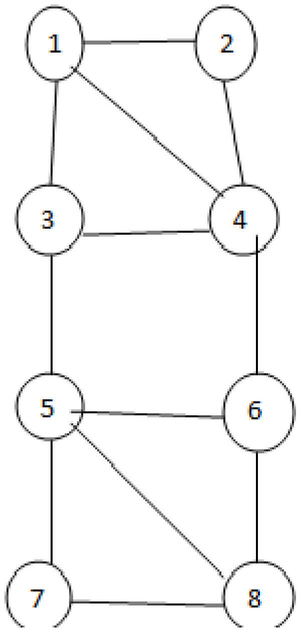

Fig 2. Undirected Graph

The dependent sets are:
Vertex 1 = {2, 3, 4}
Vertex 2 = {1, 4}
Likewise calculate all dependent sets
After calculating all dependent sets generate uniform partitions of given graph by processing dependent sets.

3.3 Dependent Set Based Graph Partitioning Algorithm

The proposed graph partitioning method consists of following steps:

1. Read the input data set D.
2. Parse the data set i.e arrange it in one order (Ascending / Descending).
3. Calculate the sets of adjacent vertices for every vertex from input data set. These sets are called as dependent sets.
4. Calculate the size of each dependent set, process and analyze the sets to calculate threshold value of number of partitions
5. Calculate the partitions for sets by considering largest set first till all the vertices of data set does not get covered in any of the partition.
6. Store these partitions and dependent sets on the disk.
7. Form a hierarchical representation of all partitions by taking most connected vertex of every partition.
8. Store this representation that is tree on the disk.

3.4 Algorithm for Dynamic Large Graph Analysis

This section explains the working of proposed algorithm to perform operations on large graph dynamically. The steps of the algorithm are:

1. Traverse the graph – Tree from Super Graph i.e root node till the Leaf Super Node of required partition.
2. Select the Leaf Super Node, its corresponding partition will be loaded in memory and shown on the system.
3. Now add or delete any node or edge in the partition.
4. Once the updation is done the store the partition again on the disk and update the corresponding leaf node information.

4. Result

4.1 Data Set

To analyse the performance of the proposed methods following data sets are used

1. DBLP Data Set: It is a database of Computer Science publications which represents an authorship graph in which every graph node represents an author and the edge represents co-author relationship.

2. Social Networks: Twitter
http://www.socialcomputing.asu.edu/dataset/Twitter

4.2 Expected Results

Following table shows the expected results of proposed graph partitioning method on given data:

Table 1. Expected Results

No. of Nodes	**No. of Edges**	**No. of Partitions**
8	11	2
10	17	2
12	22	3
25	100	4
50	300	5

5. CONCLUSION

The paper concludes that, the main issue in large graph analysis is to decompose it into sub graph. The existing graph portioning methods requires excessive processing and some are not scalable for large graph. The proposed graph partitioning method addresses the issue of limited main memory by storing the partitions on the disk. All existing approaches work on static large graph the method proposed here also addresses dynamic updation in large graph.

6. REFERENCES

[1] C. Faloutsos, K.S. McCurley, and A. Tomkins, “Fast Discovery of Connection Subgraphs,” Proc. ACM 10th Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD), pp. 118-127, 2004.

[2] Chao-Wei Ou and Sanjay Ranka “Parallel Incremental Graph Partitioning”IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 8, NO. 8, AUGUST 1997

[3] C.R. Palmer and C. Faloutsos, “Electricity Based External Similarity of Categorical Attributes,” Proc. Seventh Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 486-500, 2003

[4] G. Karypis and V. Kumar, “Multilevel Graph Partitioning Schemes,” Proc. IEEE/ACM Conf. Parallel Processing, pp. 113-122, 1995.

[5] G. Kasneci, S. Elbassuoni, and G. Weikum, “Ming: Mining Informative Entity Relationship Subgraphs,” Proc. 18th ACM Conf. Information and Knowledge Management (IKM), pp. 1653- 1656, 2009

[6] H. Tong and C. Faloutsos, “Center-Piece Subgraphs: Problem Definition and Fast Solutions,” Proc. 12th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp. 404-413, 2006.

[7] J.F. Rodrigues Jr., H. Tong, A.J.M. Traina, C. Faloutsos, and J.Leskovec, “Large Graph Analysis in Gmine System” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 1, JANUARY 2013

[8] Stephen T. Barnard and Horst D. Simon.A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. In Proceedings of the sixth SIAM conference on Parallel Processing for Scientific Computing, pages 711–718, 1993.