International Conference on "Emerging Trends in Computer Engineering, Science and Information Technology"- 2015
Special Issue of International Journal of Electronics, Communication & Soft Computing Science and Engineering, ISSN: 2277-9477

# An Approach to Handle Dynamic Graph Partitioning

**Ms. Rupali J. Jagdale** **Prof. Mrs. S. M. Kamalapur**

*Abstract* — **Large Dynamic Graphs are the big data structures. Now days it is important to study the dynamic graphs as there are different graphs are prepared like facebook, twitter, network analysis. Large Dynamic graphs are difficult to analyze. More memory is required to load this kind of graphs. Large graphs consist of complex data structure. To view these graphs in understandable form one has to make the partitions of the graph. Making partition will create overlapping sub graphs. Overlapping sub graphs will contain repeating nodes or some common nodes. These partitions can easily be analyzed and used for further processing. Dynamic graph always gets the addition or deletion of the contents like edge insertion or deletion and node insertion and deletion. There are updations which can be reflected in the sub graphs. The proposed system works on the same kind of graphs where large graph is partitioned into sub graphs.**

*Key Words* - **Edge Insertion; Edge cuts; Graph Partitions; Large Dynamic graphs; Metanodes; Sub-graph.**

## I. INTRODUCTION

Large graphs are made up of millions of nodes and edges. They are so much complicated as edges and nodes are connected in complex manner. For large graphs, viewing a full layout of the whole graph may not provide a useful level of abstraction for users and can be visually overwhelming. Dynamic graphs often change with time as there is addition or deletion of nodes in graphs. As large dynamic graphs are of compound data structure, these graphs require too much processing of knowledge of a structure of the graph. More Memory is required to load such graphs. It is very complicated to analyze and predict about graph as number of nodes are maximum in large graphs. Large graph analysis begins with separation of the corresponding graph into many small parts called as sub graph. "Graph partitioning" refers to a family of computational problems in which the vertices of a graph have to be partitioned into more pieces while minimizing the number of the edges that cross the cut. As whole graph cannot be loaded into memory for processing at particular time one must make partitions of Large Graphs. Once graphs are partitioned for analyzing, many operations like updation of graphs after node/edge deletion/insertion can be done. These methods of updating large dynamic graphs also focus on sub graphs, that any node/edge insertion or deletion in sub graph should also be reflected in main large dynamic graph. Graph partitioning problems are of NP-hard problem type. Approximation algorithms can solve these problems related with partitioning. Graph partitioning is a common and frequent preprocess- step in many high-performance parallel applications on distributed- and shared-memory architectures. There are several parallel implementations of graph partitioning for distributed-memory architectures. For tree and grids there are no approximation algorithms that they can solve partitioning problem. Important applications of graph partitioning are scientific computing and task scheduling in multi-processor systems.

## II. LITERATURE SURVEY

There are different works which have focused on graph partitioning by different ways. Memory usage decrement is achieved in some of the cases. Yet the dynamic graph partitioning and its uptations is not achieved properly. Following methods are there which tried to achieve graph partitioning with less edge cuts.

Daniel Archambault et. Al. presented the hierarchical partitioning to explore the results of graph partitioning. [1]

Opening and closing of meta nodes to visualize required part of graph. Metanodes are useful parts of this implementation. Underlying graph topology is explained here. Which has worked to prepare the sub graphs using meta nodes.

Additional operations on graph hierarchy are not done here. Dynamically addition and deletion of nodes could not be focused. Hence there is problem of hierarchical graph partitioning.

Vladimir Batagelj et. Al.[2] has represented a method of graph partitioning using (X-Y) clustering and hybrid visualizations. Here graph clusters are formed. VHYXY has supported both node and edge representations and matrix-based representations visualizing graph prepared of different clusters. Practically graphs are not considered. Here social sites cannot be handled in the form of graphs.

Frank van Ham et. Al. [3] accessed the small world graphs. Natural cluster structures of the graphs are identified to represent them in the format of sub graphs. Results show the abstracted view of the large graph. Larger graph of billions of nodes and edges could not be handled here for making partitions.

G. Kasneci et. Al. [4] presented knowledge discovery on graph

based data. Social networking sites and knowledge sharing communities are considered for graph based data structure. Handling the dynamic properties of the graphs could not be achieved.

## III. MOTIVATION

Large dynamic graphs are difficult to load into memory. Hence partitioning of these graphs is required to analyze the whole graph in the form of small partitions called as sub graphs. Till now the static graph partitioning and updations are taken care and dynamic graph partitioning and updations are not considered. Hence the proposed work can make the sub graphs of large Dynamic Graphs and update them.

Previous methods used static graph partitioning. There was a fixed approach of partitioning a give graph. Many times percentage of the given dataset was used to show the sub graph. The problem was user could not provide the particular number of partitions they want. There was the problem that limited partitions were made there as per the 20% and 30 % limit was put up. There was requirement to get the perfect partitions as the dataset size is. Hence the proposed system working here to get as many sub graphs user wants.

## IV. IMPLEMENTATION DETAILS

### A. *Proposed system*

Figure 1. shows the working of proposed system. Function of each block is discussed below:

*1*) Input Graph Dataset
Reading the input graph dataset which will contain nodes and edges shown in the text format.

*2) Calculate total nodes and edges*
Calculate all the nodes and edges in the graph dataset. This can help to get an idea that how many nodes can be shown in the sub graphs.

*3) Prepare Dependent Sets*
Prepare the dependent sets of the each node where the node set will have all its connected nodes.

*4) Process Dependent Sets*
By comparing dependent sets and arranging them in ascending order we will prepare the partitions.

*5) Prepare Partitions*
Process on each dependent set and prepare the overlapped partitions of the graph. Also there is need to get the intersection of some of the dependent sets.

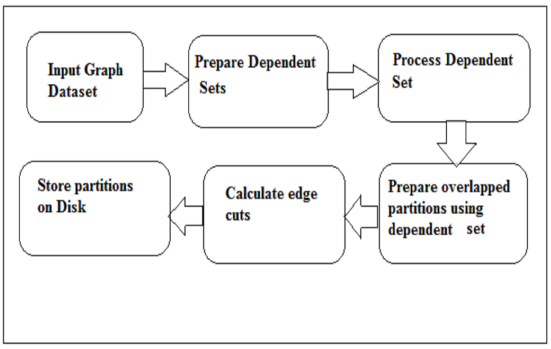


Fig.1. Block Diagram of Proposed System

*6) Store the Partitions*
Storing the partitions will store the dataset files of the sub graphs in the memory itself. So whenever required sub graph can be used.

### B. *Experimental Setup*

All Experimentation is performed using Pentium processor and 2 GB RAM. The required operating system is windows 7(32 bit or 64 bits) with JDK1.7update 10 and Netbeans 8.0.2. The proposed system works on the graph datasets, in the format of a text file and provides the output in the text file.

#### A. *Dataset*

To analyze the performance of the proposed graph partitioning method following data sets are used DBLP Data Set: It is a database of Computer Science publications where author and co-author relationship is shown using graphs.

## VI. AVAILABLE RESULTS

Table II. Available Results

| No. | Nodes | Edges | Partitions | Edge Cuts |
|---|---|---|---|---|
| 1 | 18 | 29 | 5 | 14 |
| 2 | 6 | 23 | 2 | 11 |
| 3 | 8 | 45 | 3 | 39 |

Above results show the limited edge cuts in dynamic graph partitioning. Previous approaches could satisfy the problem of partitioning, but there were many edge cuts as shown in following graph.

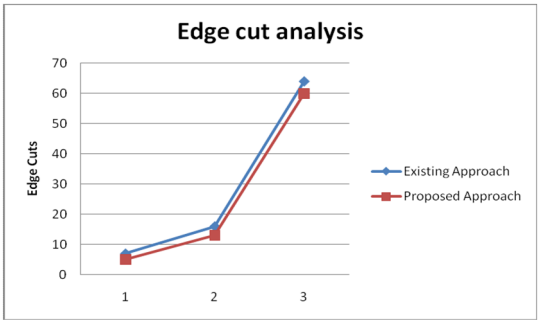


Fig. 2. Result analysis

## CONCLUSION

The existing graph partitioning methods are useful to partition the static graphs. Many of them have used hierarchical graph partitioning i.e. tree structures for partitioning of graphs. Proposed work has focused on dynamic graph partitioning. Dependent set of each node are processed to get the partitions of the main large graph. This is useful to solve the problem of insufficient memory to load the large graph dataset. Here graph is partitioned using the set theory.

## REFERENCES

[1] D. Archambault, T. Munzner, and D. Auber, "Grouseflocks: Steerable Exploration of Graph Hierarchy Space," IEEE Trans. Visualization and Computer Graphics, vol. 14, no. 4, pp. 900-913, July/Aug. 2008

[2] V. Batagelj, W. Didimo, G. Liotta, P. Palladino,and M. Patrignani, "Visual Analysis of Large Graphs Using (x,y)-Clustering and Hybrid Visualizations," Proc. IEEE Pacific Visualization Symp. (PacificVis), pp. 209-216, 2010.

[3] F. van Ham and J.J. van Wijk, "Interactive Visualization of Small World Graphs," Proc. IEEE Symp. Information Visualization(InfoVis), pp. 199-206, 2004.

[4] 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.

[5] P. Eades and Q. Feng, "Multilevel Visualization of Clustered Graphs," Proc. Symp. Graph Drawing, pp. 101-112, 1997.

[6] V. Batagelj, W. Didimo, G. Liotta, P. Palladino, and M. Patrignani, "Visual Analysis of Large Graphs Using (x,y)-Clustering and Hybrid Visualizations," Proc. IEEE Pacific Visualization Symp. (PacificVis), pp. 209-216, 2010.

[7] M.L. Huang and Q.V. Nguyen, "A Space Efficient Clustered Visualization of Large Graphs," Proc. Fourth Int'l Conf. Image and Graphics, pp. 920-927, 2007.

[8] P. Vaz de Melo, L. Akoglu, C. Faloutsos, and A. Loureiro, "Surprising Patterns for the Call Duration Distribution of Mobile Phone Users," Proc. European Conf. Machine Learning and Knowledge Discovery in databases (ECML-PKDD), pp. 354-369, 2010

[9] "A Partitioning Method for Large Graph Analysis", Miss. Shital Deshmukh1, Prof. S. M. Kamalapur,IJAIEM PUBLICATION, ISSN2319 – 4847 , May 2014

[10] "A Study to Handle Dynamic Graph Partitioning", Rupali Jagdale, Mrs. S. M. Kamalapur, IJAIEM PUBLICATION, ISSN2319 – 4847, October 2014

**Rupali J. Jagdale** received the B.E. degree in Information Technology from K.B.Thakare College of Engineering, Nashik, Savitribai Phule Pune University in 2012. Now pursuing M.E.degree in Computer Engineering from K. K. Wagh Institute of Engineering Education & Research, Nashik, Maharashtra, India.

**Prof. S. M. Kamalapur,** working as Assosiate Professor in K. K. Wagh Institute of Engineering Education & Research, Nashik, Maharashtra, India.