Volume 5, Issue 11, November 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Study of Iterative Mapreduce Techniques on Frequent Subgraph Mining

[1]Praveena[*], [2]Anitha, [3]Rohini
[1] Student, [2, 3]Assistant Professor
[1, 2, 3] Department of CSE, Vivekanandha College of Engineering for Women,
Thiruchencode, Namakkal, Tamilnadu, India

Abstract----Frequent subgraph mining (FSM) is an important task for exploratory data analysis on graph data especially when the graph is huge. In the recent years, many algorithms have been proposed to solve this task. These algorithms assume that the mining task's data structure is small enough to fit in the main memory in the systems. However, as the real-world graph data grows, both in quantity and size, such an assumption could not be met. To overcome this, FSM-H (Frequent Subgraph Mining using Hadoop) has been proposed which uses an iterative mapreduce based framework. This paper provide a general overview of the different papers related to an iterative mapreduce and gives the knowledge on different algorithms proposed for mining frequent subgraph which are basis for future research.

Keywords---- Frequent subgraph mining, iterative mapreduce, Hadoop

I. INTRODUCTION

In recent years, the "big data" phenomenon has engulfed a significant number of research and application domains including data mining, computational biology, environmental sciences, web mining, and social network analysis. In these domains, analyzing and mining of massive data for extracting novel insights has become a routine task. However, traditional methods for data analysis and mining are not designed to handle massive amount of data, so in recent years many such methods are re-designed and re-implemented under a computing framework that is aware of the big-data syndrome.

FSM-H distributed frequent subgraph mining method over Mapreduce. FSM-H generates a complete set of frequent subgraphs for a given minimum support threshold. To ensure completeness, it constructs and retains all patterns that have a non-zero support in the map phase of the mining, and then in the reduce phase, it decides whether a pattern is frequent by aggregating its support from different computing nodes. Iterative Mapreduce is used here which can be defined as a multi staged execution of map and reduce function pair in a cyclic fashion, i.e. the output of the stage i reducers is used as an input of the stage i + 1 mappers.

II. LITERATURE SURVEY

Gabriel Ghinita [2] is to mining frequent subgraph from a large collection of graph datasets. Applications are bio-informatics, social networks, and computer vision etc. To incur significant overhead for accessing the data as the number of disk input output is very high. In this paper two step approaches is proposed as filter and refinement uses mapreduce. In filter step the collection of graphs is partitioned among worker nodes and each worker nodes determines a set of locally frequent subgraphs on its local partition. In refinement step the union of all local candidate is processed where each candidate is evaluated across all partitions not only that where it is originated and only globally frequent subgraphs are retained. This approach can improve efficiency, reduces communication cost with low computational overhead. Filter and Refinement mapreduce works well for large datasets.

In [11] is to find the frequent subgraph discovery in large graph datasets. In this paper the proposed approach is density-based partitioning technique using mapreduce is to ensure load balancing and to limit the impact of parallelism and the bias of the tolerance rate. There are two instances such as 1)default partitioning method proposed by mapreduce 2) a density-based graph partitioning technique (DGP).DGP tends to balance graph density distribution in [12] each partition. The main advantage is to balance computational load and decrease the execution time. The load balancing problem is a scalability issue is addressed.

[4] Find all instances of a given sample graph in a larger data graph. The proposed algorithm that uses a single round of mapreduce and are able to detect all instances of a given sample graph. To mainly focuses on minimize the communication cost and computation cost and it can uses one round algorithm for arbitrary subgraphs. Efficient mapping schemes to minimize communication cost with efficient serial algorithms to be used at the reducers. The problem of

enumerating instances of a sample graph in a huge data graph and the technique derive a parallel algorithm of the same complexity as the serial algorithm.

Author [30] Mapreduce has emerged as a de facto programming paradigm for parallel computation on massive data sets. The main focus of this work is to give mapreduce algorithms for counting triangles which we use to compute clustering co-efficient. Mapreduce and its open source implementation Hadoop, have emerged as the standard platform for large scale distributed computation. In this paper we will give algorithms for computing one of the fundamental metrics for social networks, the clustering co-efficient, in the mapreduce framework. The proposed algorithm is sequential triangle counting algorithm can use any triangle counting algorithm as a black box and distribute the computation across many machines. The advantage is increased time efficiency.

Author in [6] proposed a framework that adopts breadth first search strategy to iteratively extract frequent subgraphs as mapreduce frequent subgraph extraction (MRFSE) like platform such as Hadoop. An iterative mapreduce frequent subgraphs is all frequent size-(i+1) subgraphs are generated based on frequent size- (i) subgraphs of the ith iteration using a single mapreduce job. In this paper mainly focus on parallelizing techniques directly using mapreduce does not yield good performance as it is difficult to balance the work load. The main advantages are efficiently extract frequent subgraphs and also it is scalable and efficient. A graph isomorphism testing is required to remove duplicates and finally each distinct subgraph is verified whether it is frequent by counting the number of occurrences in the graphs of the dataset.

In [5] the problem is designing efficient and scaling approaches for frequent subgraph discovery in large clusters. The proposed approach is a large scale and fault-tolerant of subgraph mining by means of a density based partitioning technique using mapreduce to build balanced partitions of a graph database over a set of machines. The Mapreduce framework was designed so that node failures are automatically handled by the framework. The main advantages are reliable and scalable in order to improve the efficiency of the subgraphs. Using this approach, it can decrease the subgraph mining complexity knowing that the time complexity of the subgraph mining process is proportional to the size of the input data. The impact of chunk size to the accuracy is to increase parallelism and it is to reduce the chunk size.

Wei Wang et al [25] is to apply a novel frequent subgraph mining algorithm to three graph representations of protein three dimensional (3D) structures. The existing system is to identify several hundred common subgraphs equivalent to common packing motifs found in the majority of proteins in the family. We also use the counts of motifs extracted from proteins in two different SCOP families as input variables in a binary classification experiment using Support Vector Machines. Recurring substructures in a group of compounds with similar biological activity can be identify the representing these compounds as undirected graphs find the frequent subgraphs. The recurring substructures can indicate chemical features responsible for compounds' activities. The proposed method is the application of the frequent subgraph mining algorithm to protein structures represented as graphs. The goal of this investigation was to identify frequent subgraphs common to all (or the majority of) proteins belonging to the same structural and functional family.

In [1] Frequent subgraph is used to mining graph data using several frequent subgraph mining methods. Relational databases have their own space as well computing constraints when it comes to storing large databases. In this paper the proposed algorithm is an object oriented frequent subgraph mining (OO-FSG) database db4o to store graph data and it is used to mine the frequent subgraphs. The main issue in OO-FSG is for large data sets that cannot fit in main memory. Using object oriented databases over relational systems had an advantage in performance and scalability. The db4o is used to store large graph data sets thus eliminating the constraint of memory graph datasets also overcomes the space constraints.

Hayes [30] in this paper mainly focus on the challenges for mapreduce on big data and the mapreduce has been as one of the key enabling approaches for meeting continuously increasing demands on computing resources imposed by massive data sets. Mapreduce challenges are grouped into four categories are: 1) data storage (relational databases and No SQL stores), 2) Big data Analytics (machine learning and interactive analytics), 3) online processing, 4) security and privacy. The main advantage reliability is achieved by reassigning any failed node's job to another node. A well known open source mapreduce implementation is Hadoop which implements mapreduce on top of the Hadoop distributed file system (HDFS)[3].

[3] proposed an Iterative Mapreduce is used here which can be defined as a multi staged execution of map and reduce function pair in a cyclic fashion, i.e. the output of the stage i reducers is used as an input of the stage i + 1 mappers. Graph isomorphism is also considered. In addition, before nodes are assigned with graphs for Map process, the graphs are balanced such that all the nodes get correct number of graphs with nodes count. For example, two small graphs are given to Node A and one big graph is given to Node B. So, the map processes are completed in fewer intervals in all the nodes so that reduce phase can be started immediately. The advantages are before sending input graph data to nodes, they are balanced. For example, two nodes are equal number of nodes and edges .Nodes complete the mapper process in less intervals so that Reduce phase can be started with minimum delay. Overall time efficiency is increased. The above papers limitations can overcome by this iterative mapreduce based on frequent subgraph mining.

III. CONCLUSION

Frequent subgraph mining using mapreduce has attracted plenty of attention but much less attention has been given to mining frequent subgraph mining in an iterative mapreduce .This paper surveys different research papers that proposed various algorithms which are basis for future research in the field of graph mining. This paper explains

different application areas where the frequent subgraphs are used. Identifying frequent subgraphs efficiently from large datasets and the interesting subgraphs from the graph data sets are the challenging tasks in the field of frequent subgraph mining.

REFERENCES

[1] B. Srichandan and R. Sunderraman, "*Oo-FSG: An object-oriented approach to mine frequent subgraphs*," in Proc. Australasian Data Mining Conf., 2011, pp. 221–228.

[2] X. Xiao,W. Lin, and G. Ghinita, "*Large-scale frequent subgraph mining in Mapreduce*," in Proc. Int. Conf. Data Eng., 2014, pp. 844– 855.

[3] Mansurul A.Bhuiyan and Mohammad Al Hasan ,"*An Iterative Mapreduce Based Frequent Subgraph Mining Algorithm* ",in IEEE Transaction on Knowledge and Data Engineering, Vol.27,No.3,March 2015.

[4] F. Afrati, D. Fotakis, and J. Ullman, "*Enumerating subgraph instances using map-reduce*," in Proc. IEEE 29th Int. Conf. Data Eng., Apr. 2013, pp. 62–73.

[5] Sabeur Aridhi, Laurent d'Orazio , Mondher Maddouri and Engelbert Mephu Nguifo " *A large-scale and fault-tolerant approach of subgraph mining using density-based partitioning*", 30 Nov 2012.

[6] Wei Lu,Gang Chen,Anthony K.H.Tung,Feng Zhao " *Efficiently extracting frequent subgraphs using mapreduce* " in IEEE Conf. on Big data, 2013.

[7] S.-Q.Wang,Y.-B.Yang,Y.Gao,G.P.Chen and Y.Zhang," *Mapreduce-based closed frequent itemset mining with efficient redundancy filtering*", in Proc.IEEE 12th Int. Conf. Data Mining Workshops,2012,pp.49—453.

[8] Pang-Ning Tan, Michael Steinbach, Vipin Kumar" *Introduction to data mining, Pearson Education"*, book.

[9] S. Hill, B. Srichandan, and R. Sunderraman, "*An iterative Mapreduce approach to frequent subgraph mining in biological datasets*", in ACM Conf. Bioinformat., Comput. Biol. Biomed., 2012,pp. 661–666

[10] B.-S. Jeong, H.-J. Choi, M. A. Hossain, M. M. Rashid, and M. R.Karim, "*A Mapreduce framework for mining maximal contiguous frequent patterns in large DNA sequence datasets*," IETE Tech. Rev., vol. 29, pp. 162–168, 2012.

[11] Sabeur Aridhi, Laurent d' Orazio,Mondher Maddouri and Engelbert Mephu Nguifo, " *A novel mapreduce-based approach for distributed frequent subgraph mining*", 9 May 2014.

[12] J.Dean and S.Ghemawat,"*Mapreduce: Simplified data processing on large clusters*", Commun.ACM, vol.51, pp.107-113, 2008.

[13] M. Kuramochi and G. Karypis, "*Frequent subgraph discovery*," in Proc. Int. Conf. Data Mining, 2001, pp. 313–320.

[14] S. Chakravarthy and S. Pradhan, "*Db-FSG: An SQL-based approach for frequent subgraph mining*," in Proc. 19th Int. Conf. Database Expert Syst. Appl., 2008, pp. 684–692.

[15] S. N. Nguyen, M. E. Orlowska, and X. Li, "*Graph mining based on a data partitioning approach*," in Proc. 19th Australasian Database Conf., 2008, pp. 31–37.

[16] Y. Tao, W. Lin, and X. Xiao"*Minimal mapreduce algorithms*", In SIGMOD Conference, pages 529–540, 2013.

[17] Wang Lam, Lu Liu, STS Prasad ,Anand Rajaraman ,Zoheb Vacheri ,AnHai Doan ," *Muppet: Mapreduce Style Processing of Fast Data*" ,2012.

[18] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin. Incoop "*MapReduce for Incremental Computations*", In SOCC, pages 7:1–7:14, 2011.

[19] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears." *Mapreduce Online*", In NSDI, pages 313–327, 2010

[20] B. Bahmani, R. Kumar, and S. Vassilvitskii, "*Densest subgraph in streaming and mapreduce*," Proc. VLDB Endow.,vol. 5, no. 5, pp. 454– 465, Jan. 2012.

[21] J. Huan, W. Wang, and J. Prins, "*Efficient mining of frequent subgraphsin the presence of isomorphism*," in Proceedings of the Third IEEE International Conference on Data Mining, ser. ICDM '03. Washington ,DC, USA: IEEE Computer Society, 2003.

[22] Coble, J., Rathi, R. Cook, D., Holder, L." *Iterative Structure Discovery in Graph-Based Data*",. the International Journal of The Artificial Intelligence,2005.

[23] Nijssen and J. N. Kok. "*The Gaston Tool for Frequent Subgraph Mining*", Proc. Int'l Workshop on Graph-Based Tools, 127(1):77–87, 2004.

[24] C. Borgelt. "*On Canonical Forms for Frequent Graph Mining*", In 3rd Int'l Workshop on Mining Graphs, Trees, and Sequences, pages 1– 12Porto, Portugal, October 2005.

[25] Jun Huan, Wei Wang et al "*Mining Protein Family Specific Residue Packing Patterns From Protein Structure Graphs*" ,International Conference,Vol.55,pp.324-332.

[26] Morales, A. Gionis, and M. Sozio. "*Social content matching in the mapreduce*", PVLDB, pages460-469, 2011.

[27] F. Chierichetti, R. Kumar and A. Tomkins "*Max-Cover in Mapreduce*", In WWW, pages 231-240, 2010.

[28] S. Ranu and A. K. Singh, "*Graphsig: A scalable approach to mining significant subgraphs in large graph databases*," *ICDE*, 2009.

[29] Q. He, Q. Tan, X. Ma and Z. Shi, "*The high-activity parallel implementation of data pre-processing based on Mapreduce*," Proc. of the 5th International Conference on Rough Set and Knowledge Technology, 2010.

[30] *Siddharth Suri Sergei Vassilvitskii* "*Counting Triangles and the Curse of the Last Reducer*", Proc. of the IEEE 10th 2014 World Congress on Services (SERVICES 2014), Alaska, USA, June 27-July 2, 1014

[31] Jan Ramon," *Graph Mining*"

[32] http://www.tutorialspoint.com/map_reduce.htm

[33] Lect12_Graph Mining

