ISSN: 0976-3104
SPECIAL ISSUE: (Emerging Technologies in Networking and Security (ETNS)

ARTICLE OPEN ACCESS

# A SURVEY ON MAPREDUCE USING FREQUENT SUBGRAPH MINING

**M. Gokilavani*, B. Anitha, R. Jayanthi**

*Dept. of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Elayampalayam, Tamilnadu, INDIA*

## ABSTRACT

*Graph mining approaches have become more popular, especially in certain domains such as bioinformatics, chemo informatics and social networks. In data mining applications, mining frequent sub graph from a large number of small graphs is an important task. For mining frequent sub graphs many algorithms have been proposed. To overcome this some distributed solution using Map Reduce is becoming important paradigm for computation on massive data. In experimented work, we investigate how to economically perform extraction of frequent sub graph over a large datasets using Map Reduce. However, as the real-world graph data grows in amount and mass, result could not be met. To overcome this, a few graph database-centric methods have been projected in real problem for solving FSM. however, a distributed solution using Map Reduce paradigm has not been explored extensively. Since Map Reduce is flattering the de-facto paradigm for computation on massive data, an efficient FSM algorithm on this paradigm is of huge demand. The efficiency of extracting the frequent subgraph is experimentally investigated over a large datasets using map reduce.*

**Published on:** 2nd -December-2016

KEY WORDS

*FSM,database-centric method,de-facto paradigm,mapreduce*

***Corresponding author: Email:** greenvani88@gmail.com; **Tel.:** +91 9442041624

## INTRODUCTION

The big data is a huge amount of data, when compared to other datasets. The large number of data is stored in memory, the stored data will be retrieved (wanted data)in that time we are facing the many number of problems. And also the problems resolved means, in problem solving operation using the data mining process. In data mining, large amount of data, the wanted data only extracted from big data, i.e. ,stored large amount of data. Some storage area only using the data mining concepts and also using the map reduce concepts. The map reduce concepts are mainly focused on the frequent set of items, the frequent path finding and then the map reducing process is executed. In map reduce process, having two type of functionalities. They are mapping and reducing functions used in map reduce process. In map reduce function of operation, the large amount of graph data will be divided into sub graphs, divided sub graphs are merged. For the use of finding the frequent set of path, time is reduced for retrieving the wanted set of data.

## BIGDATA

Big data is a big term for data sets so large or complex that conventional data dispensation applications are incompetent.In the bigdata Challenges are investigation, arrest, hunt, contribution, storage, relocate, apparition, and information isolation. The term often refers merely to the use of extrapolative analytics or other definite advanced methods to mine value from large dataset, and seldom to a particular size of data set. Correctness in big data may lead to more convinced judgment making. And enhanced decisions can mean greater operational competence, cost diminution and reduced risk. Analysis of data sets can find new correlations, to spot business trends, put a stop to diseases and so on. Scientists, business executives, practitioners of media and marketing and governments alike recurrently meet difficulties with large data sets in areas including Internet rummage around, economics and industry informatics. Work with big data is inevitably uncommon; most investigation is of personal computer size data, on a desktop personal computer or notebook that can switch the obtainable data set [13].

Big Data processing chiefly depends on equivalent programming models like MapReduce, as well as on condition that a cloud computing platform of Big Data services for the public. MapReduce is a batch-oriented similar computing model. There is still a convinced gap in presentation with relational databases. Civilizing the presentation of MapReduce and attractive the concurrent scenery of large-scale data processing have acknowledged a momentous amount of consideration with MapReduce similar programming being sensible to many machine knowledge and data mining techniques. Data mining techniques habitually need to scan through the instruction data for obtaining the statistics to solve or diminish model parameters. It calls for exhaustive computing to admittance the large-scale data habitually. To amplify the efficiency of algorithms, Chu et al. projected a universal-purpose parallel programming method, which is applicable to a more number of machine learning algorithms based on the usual MapReduce programming model on multi-core processors. Classical data mining algorithms are realized in the skeleton includes in the vicinity prejudiced linear regression, k-Means, naive Bayes, linear sustain vector machines, the self-regulating variable investigation, Gaussian discriminant study, anticipation maximization, and flipside-broadcasting neural networks.

FSM-H is distributed frequent subgraph mining method in excess of MapReduce. Given a graph database, and a fewest amount support threshold, FSM-H generates a unqualified set of frequent subgraphs[14]. To ensure completeness, it produces and retains all patterns in a partition that has a non-zero hold up in the map phase of the extracting, and then in the reduce phase, it decides whether a pattern is recurrent by aggregating its support computed in all partitions from dissimilar compute nodes. To triumph over the reliance among the states of a mining process, FSM-H runs in an repeated process fashion, where the output from the reducers of iteration i-1 is used as an input for the mappers in the iteration i. The mappers of iteration i produceing candidate subgraphs of size i (number of edge), and also calculate the local support of the candidate pattern. The reducers of iteration i then compute the true frequent subgraphs (of size i) by aggregating their.

## FREQUENT SUBGRAPH MINING

The FSM is to haul out all the frequent subgraphs, in a given data set, whose incident counts are above a individual threshold. The immediately forward idea behind FSM is to “grow” candidate subgraphs, in either a breadth first or depth first manner (candidate production), and then institute if the documented candidate subgraphs crop up recurrently enough in the graph data set for them to be painstaked interesting (support counting) [15]. The two main research issues in FSM are thus how to efficiently and effectively (i) produce the candidate regular subgraphs and (ii)bring to a close the frequency of occurrence of the generated subgraphs.

Effective candidate sub graph production given that the generation of photocopy or superfluous candidates is to defeat. Occurrence counting given cyclical comparison of candidate subgraphs with subgraphs in the input data, a process known as isomorphism inspection. FSM, in many compliments, can be viewed as an extension of Frequent Itemset Mining (FIM) popularised in the context of group rule mining. Consequently, many of the proposed solutions to addressing the main research issues completing FSM are based on comparable techniques found in the sphere of FIM.

The existing system involves mining frequent sub-graphs from the given graph database (A database with ‘N’ graphs). Here MapReduce approach is used which is a programming model that enables disseminated computation over massive data. In extraordinary, Iterative MapReduce is used here which can be defined as a multi theatrical execution of map and reduce function pair in a cyclic fashion, i.e. the production of the period i diminish is used as an input of the phase i + 1 mappers. An peripheral condition decides the standstill of the job. Graph isomorphism is also measured.

A elementary feature for truthful search based algorithms is that the taking out is absolute i.e. the mining algorithms are distinct to find all frequent subgraphs in the input data. Complete mining algorithms execute imaginatively only on bare graphs with a large amount of labels for vertexes and edges. Due to this completeness constraint these algorithms take upon physically extensive subgraph isomorphism assessment either overtly or absolutely substantial in a outstanding computational visual protuberance The usefulness of integrating constraint into the FSM process is liable by many aspects, including the belongings of the data and pruning cost. limitation based mining algorithms therefore necessitate to take into account the trade-off flanked by the pruning cost and any credible assistance[11].

- Before sending input graph data to nodes, they are not neutral For example, one node may be assigned with more big graphs and other node with small graphs.

- Nodes have to stay for Reduce phase and can start the process only after all the mapper processes are finished in all nodes.
- Overall time efficiency is poor.
- Candidate generation strategy poor.
- The mechanism for traversing the search space and low occurrence counting process.
- Single graph based FSM applied.
- Graph isomorphism is neither known to be solvable in polynomial time nor NP-complete.

## LITERATURE SURVEY

Penetrating plays an significant role in Map Reduce algorithm. It helps in the combiner phase (elective) and in the Reducer phase. Let us try to appreciate how penetrating works with the help of a lot of papers. The papers are includes, map reduce concepts and their operational details. Many process will based on map reduce operations, the following papers are surveyed.

In [1] authors Jeffrey Dean and Sanjay in the paper “Map reduce: Simplified Data Processing On Large Clusters” describe the Map Reduce and its a brainwashing model and an connected functioning for executing and producing large data sets. Users identify a map function that processes a key/value pair to produce a set of transitional key/ value pairs, and a reduce function that joins all intermediate values connected with the identical intermediate key, Many real world tasks are expressible in this model. Programs written in this functioning style are automatically parallelized and executed on a huge cluster of examination machines. The run-time arrangement takes care of the particulars of separating the input data, scheduling the program’s implementation crosswise a set of machines, managing machine failures, and behaviour the obligatory inter-machine announcement.

In [2] authors Jie Tang, Jimeng Sun,Chi Wang and Zi Yang in the paper“ Social Influence Analysis in Large-scale Networks” illustrate the big public networks, nodes are prejudiced by others for an assortment of reasons. For example, the colleagues have brawny pressure on one’s work, while the friends have strong influence on one’s every day life. How to make a distinction the social influences from different angles (topics). How to enumerate the strength of those social influences and the model on real large networks and implement to address these fundamental questions, Topical Affinity Propagation (TAP) to model the topic-level social authority on hefty networks.

In [3] authors U Kang, Charalampos in this paper “Pegasus: A Peta-Scale Graph Mining System - Implementation and Observations” describe PEGASUS, an unlock foundation Peta Graph taking out library which performs attribute graph removal everyday jobs such as compute the diameter of the graph, manipulative the radius of every node and finding the associated mechanism. As the amount of graphs reaches quite a few Giga-, Tera- or Peta-bytes, the predictability for such a documentation grows too. To the brilliant of our knowledge, PEGASUS is the primary such documentation, implemented on the pinnacle of the HADOOP display place, the unbolt foundation version of MAPREDUCE. Numerous graph extracting operations (PageRank, ghostly clustering, diameter judgment, associated components etc.) are fundamentally a recurring matrix-vector multiplication.

In [4] authors Siddharth Suri Sergei in this paper “Counting Triangles and The Curse of The Last Reducer” describe the grouping coefficient of a node in a social network is a elementary estimate that quantifies how tightly-knit the community is in the region of the node. Its calculation can be reduced to counting the number of triangles incidence on the meticulous node in the network. In case the graph is too far above the ground to fit into storage, this is a non-trivial job, and foregoing researchers showed how to guesstimate the clustering coefficient in this circumstances. A different avenue of research is to carry out the computation in parallel, distribution it across many machines. In recent years Map Reduce has emerged as a de facto indoctrination paradigm for parallel totalling on huge data sets. The main sympathy of this work is to give Map Reduce algorithms for including triangles which we use to subtract cluster coefficients.

In [5] authors Rasmus pagh in this paper “Colorful Triangle Counting And A Map reduce implementation ”describe a new randomized algorithm for including triangles in graphs. The underneath gentle circumstances the calculation of new randomized algorithm is strappingly concerted around the factual figure of triangles. Completely if p >= max (log n / t, log n /Sqrt(t)) , where n, t, i denote the numeral of vertices  G, the figure of triangles G, the most advanced digit of triangles an edge of  G  is embarrassed then for any invariable  > 0 our unbiased approximation T is determined around its prospect. Ultimately present a Map Reduce implementation of new randomized algorithm.

In [6] authors Foto N. Afrati in this paper “Enumerating Sub graph Instances Using Map-Reduce” describe to and find all instances of a given “sample” graph in a larger “data graph,” using a solitary about of map reduce. For the standard sample graph, the triangle, we augment upon the superior known such algorithm**.** Inspect the universal case, bearing in mind together the announcement cost sandwiched between mappers and reducers and the whole functioning cost at the reducers. To diminish statement cost, **we** take advantage of the techniques of for computing multiway joins (evaluating conjunctive query) in a solitary map-reduce surrounding. Each methods are shown for translating illustration graphs into a union of conjunctive queries with as hardly any queries as probable.

In [7] authors  Bahman Bahmani in this paper “Densest Sub graph In Streaming And Map reduce” they are studied the difficulty of judgment dense sub graphs, a original primordial in pretty a few statistics administration applications, in streaming and Map Reduce, two computational models that are all the time more being adopted by significant data processing applications. A simple algorithm that make a small number of passes over the graph and obtains a (2+i) rough calculation to the densest sub graph. The obtained several extensions of this algorithm: for the case when the sub graph is prescribed to be more than a convinced size and when the graph is absorbed To the best of our knowledge, these are the sample algorithms for the densest sub graph problem that truly scale yet over demonstrable guarantees.

In [8] author Jun Huan in this paper “Mining Protein Family Specific Residue Packing Patterns From Protein Structure Graph” report on the submission of the recurrent sub graph withdrawal algorithm to protein structure correspond to as graphs. The aspire of this consideration was to be memorable with normal subgraphs widespread to each and every one (or the mainstream of) proteins belonging to the indistinguishable structural and well-designed family in the SCOP catalogue and explore these sub graphs as folks specific amino acid set down signature of the necessary family unit. even though protein graphs are talented this submission has be converted into possibly will or maynot, appreciation to every highly developed rationalization of the habitual subgraph withdrawal algorithm in employment in this paper.

In [9] authors Bay Vo and Bac Le  in this paper “OO-FSG: An Object-Oriented Approach to Mine Frequent Subgraphs” present a fresh algorithm for withdrawal  alliance rules. The progress  the algorithm which scans database one time only and use Tidset to estimate the support of comprehensive item set faster. A tree structure called GIT-tree, an glasshouse of IT-tree, is developed to store database for extracting frequent item sets from hierarchical database. Fresh algorithm is often more rapidly than MMS_Cumulate, an algorithm extracting frequent item sets in hierarchical database with more bare minimum supports, in tentative databases.

In [10] authors authors Srichandan B and R. Sunderraman in this paper “OO-FSG: An Object-Oriented Approach to Mine Frequent Subgraphs”  describe a Frequent sub graph mining (FSG) and has all time been more popular issue in data mining. Several frequent subgraph removal methods have been superior for taking out graph data. However, most of these are chief memory algorithms in which scalability is a most well-liked issue. A hardly any algorithms have opted for a relational approach that provisions the graph data in relational table.

## COMPARITIVE ANALYSIS

| S.NO | Paper Title | Algorithm | Advantages | Disadvantages |
|---|---|---|---|---|
| 1 | Map Reduce: Simplified Data Processing on Large Clusters | Map Reduce | Execution time | Worker node crash |
| 2 | Social Influence Analysis in Large-scale Networks | Topical Affinity Propagation (TAP) | Efficiency and Effective | Scalability |
| 3 | PEGASUS: A Peta-Scale Graph Mining System Implementation | Generalized Iterated Matrix-Vector multiplication | Decrease running time | Computation transparency |

| | | | | |
|---|---|---|---|---|
| | and Observations | (GIM-V) | | |
| 4 | Spectral Analysis for Billion-Scale Graphs: Discoveries and Implementation | Eigen solver (HEIGEN) | Accuracy | Convergence trouble |
| 5 | Counting Triangles and the Curse of the Last Reducer | Sequential triangle counting | No triangle is lost | Increase disk space |
| 6 | Colourful triangle counting and a map reduce implementation | Map Reduce colourful triangle counting | Total count is scaled | Low efficiency |
| 7 | Enumerating Sub graph Instances Using Map-Reduce | Partition Algorithm | Lowering the number of reducer | Communication and computation cost |
| 8 | Densest Sub graph in Streaming and Map Reduce | Largest densest sub graph | Achieve quality and performance | Scalability |
| 9 | Mining Protein Family Specific Residue Packing Patterns From Protein Structure Graphs | Delaunay tessellation | Computational competence | Robust |
| 10 | OO-FSG: An Object-Oriented Approach to Mine Frequent Sub graphs | Frequent pattern withdrawal | Stability, computation load | Chunk amount and duplication factor |

## MAPREDUCE

MapReduce is a programming model that enables dispersed computation over large amount of data. The model provides two nonfigurative operations: map, and reduce. Map corresponds to the "map" operation and reduce corresponds to the "fold" function in functional programming. Based on its role, a member of staff node in MapReduce is called a mapper or a reducer. A mapper takes a collection if (key, value) pairs and apply the map process on each of the pairs to manufacture an subjective number of (key,value) pairs as halfway output. The reducer aggregates all the values that have the comparable key in a minimize list, and applies the reduce operation on that list. It also writes the output to the output file. Iterative MapReduce: Iterative MapReduce can be defined as a multi staged execution of map and reduce function pair in a returning fashion, i.e. the output of the stage i reducers is used as an input of the juncture i + 1 mappers. An exterior situation decides the standstill of the job. Pseudo code for iterative MapReduce algorithm is obtainable in **[Figure- 1]**.

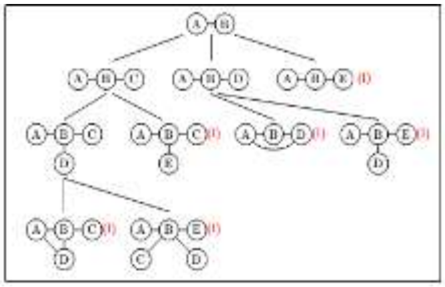

**Fig. 1: Map Reduce Mining**

………………………………………………………………………………………………………………

```
Algorithm: Iterative MapReduce():While(Condition)
Execute MapReduce Job
 Write result to DFS
Update condition
// G is the database
// k is initialize to 1
Mining Frequent Subgraph(G, minsup):
Populate F1
```

```
while Fk 6= ∅
Ck+1 = Candidate generation(Fk, G)
 forall c ∈ Ck+1
 if isomorphism checking(c) = true
support counting(c, G)
if c.sup ≥ minsup
 Fk+1 = Fk+1 S{c} k = k + 1
 return Si=1···k−1 Fi
```

In this paradigm [12], the mining task starts with recurrent patterns of size one (solitary edge patterns), denoted as F1. Then in every one of the iterations of the at the same time as loop , the method gradually more finds F2, F3 and so on until the entire recurrent pattern set (F) is obtained. If Fk is non-empty at the end of an iteration of the above while loop, from every one of the frequent patterns in Fk the removal method creates probable candidate recurrent patterns of mass k+1 . These candidate pattern are represented as the set C. For each of the candidate patterns, the extracting method operates the pattern's support against the dataset G. If the holdup is well-built than the smallest amount support doorstep (minsup), the agreed pattern is recurrent, and is stored in the set Fk+1 . Before bear including, the method also ensures that dissimilar isomorphic forms of a fashionable candidate patterns are amalgamated and only one such reproduction is processed by the algorithm. Once all the persistent patterns of size k + 1 are obtained, the while loop is continued. Thus each repeatation of the while loop obtains the set of recurrent patterns of a fixed size, and the development continues until all the recurrent patterns are obtained. The FSM algorithm earnings the union of Fi : 1 ≤ i ≤ k − 1.

## POSSIBILE SOLUTION

The following **[Table- 1]** describes investigational result for obtainable and predictable system investigation. The table contains search node, recurrent sub graph mapping node count and average recurrent sub graph amalgamation sub graph details are shown. The table contains search node, recurrent sub graph amalgamation node count and touchstone recurrent sub graph amalgamation sub graph details are shown.

**Table:1. Frequent Sub-Graph Mining Performances Analysis(Weight of Node)**

| S.NO | Search Node | Mapping Sub Frequent Graph Node Count (n) | Average of Mapping Frequent Sub graph Node [%] |
|---|---|---|---|
| **1** | **200** | **155** | 77.5 |
| **2** | **250** | **220** | 88.00 |
| 3 | 300 | 272 | 90.66 |
| 4 | 350 | 322 | 92.00 |
| 5 | 400 | 383 | 95.75 |
| 6 | 450 | 429 | 95.33 |
| 7 | 500 | 468 | 93.60 |
| 8 | 550 | 523 | 95.05 |
| 9 | 600 | 578 | 96.33 |
| 10 | 650 | 633 | 97.74 |

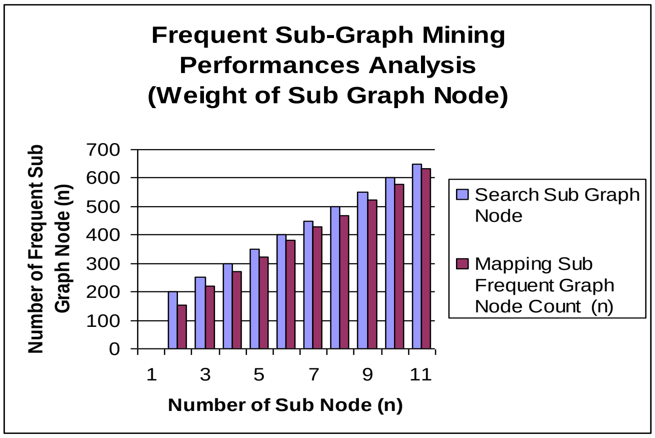

**Fig. 2: Frequent Sub-Graph Mining Performances Analysis (Weight of Node)**

……………………………………………………………………………………………………………………

The following [**Figure- 2]** describes investigational result for existing and proposed system psychiatry. The figure contains search node, frequent sub graph mapping node count and average recurrent sub graph mapping sub graph details are shown.

## CONCLUSION

The paper achieves solving the task of frequent sub graph mining on a distributed platform like Map Reduce is challenging for various reasons. An FSM method is proposed which computes the support of a candidate sub graph pattern over the entire set of input graphs from a set of graphs (Graph Database).‘N’ number of nodes is given with their capabilities. Then the graph details with vertex count are also given. Then graph with minimum vertex count and maximum vertex are found. Then the difference between maximum and minimum is also found out. Then all the groups are grouped such that a) minimum vertex to minimum vertex + 1/3rd of difference ‘$G^{a}$’, b) minimum vertex + 1/3rd of difference to minimum vertex + 2/3rd of difference ‘$G^{b}$’ and c) the remaining ‘$G^{c}$’. Then the nodes are classified as 1) low capability are assigned with ‘$G^{a}$’ graphs, 2) medium capability with ‘$G^{b}$’ graphs and high capability with ‘$G^{c}$’ graphs. Thus the paper presented a novel iterative MapReduce based frequent subgraph mining algorithm, called FSM-H. The proposed system shows the performance of FSM-H over numerous graph records. This paper shows that FSM-H is significantly better than the existing method.

### CONFLICT OF INTEREST

The authors declare no conflict of interests.

### ACKNOWLEDGEMENT

None

### FINANCIAL DISCLOSURE

None

## REFERENCES

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

[2] Agrawal R and R. Srikant. [1994] Fast algorithms for mining association rules in large databases, in Proc. *20th Int. Conf. Very Large Data Bases*, pp. 487–499.

[3] Bahmani B, R Kumar, S Vassilvitskii.[2012] Densest subgraph in streaming and mapreduce,” *Proc. Very Large Data Bases Endow*, 5(5): 454–465

[4] Chaoji V, M. Hasan, S. Salem, and M. Zaki.[2008] “An integrated, generic approach to pattern mining: Data mining template library, *Data Min. Knowl. Discov. J*, 17( 3): 457–495.

[5] Cook DJ, LB Holder, G Galal, R Maglothin. [2001]Approaches to parallel graph-based knowledge discovery, *J Parallel Distrib. Comput.*, 61: 427–446,

THE IIOAB JOURNAL ISSN: 0976-3104

[6] Dean J, S. Ghemawat.[ 2008] Mapreduce: Simplified data processing on large clusters," Commun. ACM, 51: 107–113.

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

[8] Huan J, W Wang, D Bandyopadhyay, J Snoeyink, J Prins.[ 2004] Mining protein family specific residue packing patterns from protein structure graphs," in *Proc. Int. Conf. Res. Comput. Mol Biol*.pp. 308–315

[9] Jie Tang, Jimeng Sun,Chi Wang and Zi Yang. [2010] Social Influence Analysis in Large-scale Networks, in *Proc. Int Conf Comput Aspects Soc Netw*, pp. 487–490.

[10] Kang U, C E Tsourakakis, C Faloutsos. [2009] Pegasus: A petascale graph mining system implementation and observations, in Proc. 9th *IEEE Int. Conf. Data Mining* , pp. 229–238.

[11] Nijssen S, J Kok.[2004] A quickstart in frequent structure mining can make a difference" in Proc. 10th ACM SIGKDD *Int. Conf. Knowl. Discov*. DataMining, pp. 647–652.

[12] Pagh R, CE Tsourakakis,[ 2012] Colorful triangle counting and a mapreduce implementation, *Inf. Process. Lett.* 112(7): 277–281

[13] Srichandan B, R. Sunderraman, Oo-FSG:[2011] An object-oriented approach to mine frequent subgraphs," in Proc. Australasian Data Mining Conf221–228.

[14] Suri Sand S. Vassilvitskii.[2011] Counting triangles and the curse of the last reducer, *in Proc. 20th Int. Conf. World Wide Web*, pp. 607–614.

[15] wang C, S. Parthasarathy.[ 2004 ]Parallel algorithms for mining frequent structural motifs in scientific data, in *Proc. 18th Annu. Int. Conf. Supercomput*, 31–40.