Implementation of GRAC Algorithm (Graph Algorithm Clustering) in Graph Database Compression

I Gusti Bagus Ady Sutrisna[1], Kemas Rahmat Saleh W[2], Alfian Akbar Gozali[3]
School of Computing
Telkom University
Bandung, Indonesia
[1]igustibagus.as@gmail.com, [2]bagindokemas@telkomuniversity.ac.id, [3]alfian@tass.telkomuniversity.ac.id

***Abstract*—Graph database is a representative of a data collection modeling into Node and Edge form. Graph database is one of implemented method of NoSQL (Not Only SQL), i.e. database system that is useful for data storage in a large number and is represented in the form of graph so that the data have high accessibility. However, the stored data in Graph database processing is not efficient yet in terms of data storage. The storage of million or billion nodes and edges requires compression. In this research, the conducted graph database compression uses GRAC (Graph Algorithm Clustering). The used Graph Database is the one which includes collaboration data among journal writers. In GRAC (Graph Algorithm Clustering), Hierarchical Clustering is used. It is a method that clusters Nodes into Cluster Nodes hierarchically. In the hierarchical cluster making, the strategy used is Agglomerative in which every node combined into a cluster. By applying GRAC (Graph Algorithm Clustering) using Hierarchical Clustering that forms hierarchical clusters, the lossless and well-compressed graph database will be resulted.**

Keywords—Graph Clustering, Compression

I. INTRODUCTION

Relational database is a system of data retrieval and storage which has been popular and dominant for almost three decades. A large number of applications use relational database to store the data. It can work better if the number of the data is minimal and if it has well-structured data [5]. However, when there is an increase in the data and its various processing, the relational data, with its strict scheme, is not suitable with the semi-structured data and even unstructured data. The case of the semi-structured data and unstructured data has flexibility in terms of data processing such as the social network with interconnected data.

One of the solutions to cope with the problem is by using Graph Database. It is an implementation method of NoSQL (Not Only SQL) i.e. the database system which is useful for storing data in a large number and is represented into a graph in the form of Nodes and Edge [1]. This is conducted because Node and Edge gives an opportunity to extract information inter-users. The strength of Graph database is in terms of data searching. It can be performed transversally in which every relation is represented by an edge that relates related nodes. As a result, the process can be efficient [1].

Graph database has inefficient storage since the Edge that represents all relations in Graph Database causes the large size and inefficiency. The large number of Edges becomes the research focus so that Graph database can be compressed into smaller size than before and more efficient.

From many algorithms of Graph Database Compression, GRAC algorithm is the one that is used to compress Graph database. Graph Database to be conducted is the data which includes the collaboration among academic journal writers. In GRAC (Graph Algorithm Clustering) the clustering that is used is Hierarchical Clustering.

Hierarchical Clustering is a method in clustering that will cluster the nodes into cluster node hierarchically [2]. In making the hierarchical cluster, the strategy used is Agglomerative in which every node combined into one cluster. To get an effective and efficient Agglomerative, the maximum distance among clusters is calculated. The distance is usually called Complete Linkage Clustering [5]. The index jaccard of every node is calculated to get the weight of the distance between nodes [3]. The use of Hierarchical Clustering is to form Cluster Node that has similar neighbor. The Cluster Node is then connected to the Cluster Edge and is achieved through greedy searching in every cluster node relation that abstracts Edge the most.

A well-compressed and lossless graph database can be resulted by applying GRAC (Graph Algorithm Clustering) using Hierarchical Clustering that forms hierarchical cluster.

II. THEORETICAL FOUNDATIONS OF THE STUDY

A. Graph Database

Graph data base is a database which uses graph structure with Nose, Edge and Property to represent and store data. Graph is a branch of mathematics that has application relation with many other disciplines.

978-1-4799-7752-9/15/$31.00 ©2015 IEEE

Graph theory is a branch of maths which is founded by Euler in the early 1736. Theoretically, the graph is formed by vertex (Node and Edge) that connects vertex (Node and Edge). Formally, a graph is a pair of elements of G=(V,E) in which V is a group of Nodes and E is a group of Edges that is formed by a pair of Nodes. The general way to draw a graph is by drawing a point for each vertex and joining the two points by a line. There is nothing considered irrelevant, it is only how to draw a point and a line. The most important thing is the information of which vertex pair forms an edge and which vertex does not. [1].

Graph database is a scheme-less database that uses the data graph structure with certain Nodes, Edges and Properties to depict the data. The Nodes describes the entity such as people, business etc. the properties show the related information that is connected to Nodes. On the other hand, Edge connects one Node to another [1].

In a graph, the degree of a Node x is the number of Edge that is connected to the Node. It occurs since in undirected graph, an Edge can direct out of the Node or into a Node and the degree of undirected node can be multiplied by 2. The average degree of a graph is a result of the number of edges in a graph divided by the number of the nodes. This calculation can be used to know the density of a graph. In undirected node, the average degree can be calculated.

B. GRAC (Graph Algorithm Clustering)

GRAC (Graph Algorithm Clustering) that is used is hierarchical clustering. Hierarchical clustering is one of many methods in clustering algorithm. It is also called Hierarchical Cluster Analysis (HCA)-a method to make hierarchical data from clusters. [5]. The strategies in using this method is Agglomerative and Divisive.

Agglomerative is a kind of "bottom up" forming strategy i.e. every data unit forms the cluster itself and join into one cluster with another hierarchically [4]. In contrast to Agglomerative, divisive is the opposite. Divisive is a "top down" strategy in which one cluster forms into many other clusters hierarchically.

Generally, the change of this joining and splitting is found through greedy. In agglomerative strategy, there are three formulas in deciding the distance between clusters i.e. maximum (also called complete linkage clustering), minimum (also called single linkage clustering) and maximum (UPGMA). In GRAC implementation, Hierarchical Clustering uses Agglomerative and Complete Linkage Clustering Strategy.

In the above examples, six data {a}, {b}, {c}, {e} and {f} will be formed hierarchically. The data will be combined into one cluster based on the proximity. The six data will be one Cluster {a,b,c,d,e,f} which are formed hierarchically.

The examples above will be applied in Graph Database. Graph database is database representation in the form of graph. In normal Graph, Node is represented by circle and connecting line of two nodes which is usually called Edge. Every node will have ID Node. In Cluster Graph, there are Cluster Node entity which is circular (in every entity there is node or other cluster nodes) and Cluster Edge which is a connecting line between two Cluster Nodes.

If there is graph $G = (V, E)$ in which $V = \{v_0, \dots, n_n)$ is a group of Nodes and $E \subseteq V \times V$ is a group of Edges then cluster graph $G' = (V', E')$ is a graph of power set of Cluster Nodes $V' \subseteq P(V)$ that is connected one to another by Cluster Edge $E' \subseteq V' \times V'$. Therefore, Cluster Graph consist of power set from Node and power set from Edge Graph G. If two cluster Nodes are connected by a cluster Edge, all Nodes in the first Cluster Nodes are connected to all Nodes in the second Cluster.

C. The Implementation of GRAC (Graph Algorithm Clustering)

The main purpose of GRAC algorithm implementation is to make cluster node and cluster edge which are resulted from the resulted cluster using hierarchical clustering. A Cluster Graph is formed before getting the Graph Database Compression. It consists of Cluster Nodes and Cluster Edge.

The first step in implementing the GRAC algorithm is to form a cluster consisting Node. A Cluster Node that has these nodes has similar neighbor. Therefore, the cluster node formation is gained through hierarchical clustering. However, hierarchical clustering is possible to be conducted after the jaccard Index is calculated.

To conduct Hierarchical Clustering, the matrix should be defined to calculate the distance between clusters. In this case, the distance between clusters is decided by the similarity of neighboring nodes. Two nodes that have similar neighbor are considered having minimal distance.

Meanwhile, the two nodes that have only few similar neighbors have great distance. The calculation between these nodes is done using Jaccard Index or Jacard Similarity Coefficient. [3]

The mechanism of Jaccard Index is calculating the similarity and the differences between two set. In this case, the compared items are set of neighboring Node in two nodes. For example, if there are two nodes a and b that have neighboring set A and B, the jaccard value of Node a and b is :

$$\boxed{J(a,b) = 1 - \frac{|A \cap B|}{|A \cup B|}}$$

Fig. 1. Jaccard index method

After arranging the array of Jaccard Index, the next step is to conduct hierarchical clustering. In Agglomerative Hierarchical Clustering, each singular node is considered a Custer. The pair of cluster that has the least score of jaccard, x and y are combined into a cluster of xy. Since the Hierarchical Cluster in this research uses complete linkage clustering method, the jaccard value of xy cluster on other clusters (for example z cluster) is gained from the biggest value between J(x,z) and J(y,z).

This is conducted repeatedly so in the end there will be only one cluster. If n is the number of existing Node, the looping (repetition) which is conducted to result into one cluster is n-1.

After Cluster node is gained, the next step is finding every possible combination of cluster nodes pair from Cluster Node which is in the set of Cluster Node. The combination will be connected to a Cluster Edge. A cluster node of x can be paired with Cluster Node of y if every Node in X possesses Edge with every Node in y. If x and y is paired, a cluster Edge will be formed between the two cluster nodes.

After all Cluster Edge are gained, the next step is finding greedy searching. It is called greedy since the checking is conducted from Cluster Edge that abstracts Edge the most to the one which only abstracts a single Edge. The searching can be conducted by looping all the Cluster Edges that have been arranged before. For every Cluster Edge, the checking is conducted to two cluster nodes which are connected. If all Edges, connecting every Node in the first cluster node to each node in the second cluster nodes, have not been abstracted before, the cluster Edge is considered final. If there is even one Edge between Nodes in the first Cluster Graph with the Node is the second Cluster Graph that has been abstracted by a Cluster Edge, the Cluster Edge is skipped and the checking on the next cluster is conducted.

III. Research Methodology

The purpose of this research is to make a system that implement Graph Database Compression technique in data storing process based on the dataset from SNAP (Stanford Network Analysis Project). The input of this system is collaboration data of scientific journal writing. The data is represented into Node and Edge in which the entities are described as Node and the way the entities connect to each other described as Edge. The data are in the form of text that represent the undirected graph. The following is the description of the system to be developed :

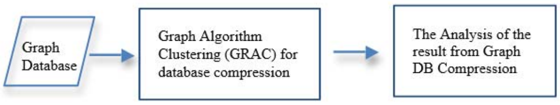
Graph Database
Graph Algorithm Clustering (GRAC) for database compression
The Analysis of the result from Graph DB Compression

Fig. 2. General description of the system

From the figure 2, the detailed processes are as follows:

1. The system uses the input from collaboration data of scientific journal writing of SNAP (Stanford Network Analysis Project) which stored later in the form of graph.

2. The system conducts a transformation into Graph Database Compression using GRAC (Graph Algorithm Clustering).

3. The system analyzes the transformation from Graph Database into Graph Database Compression.

IV. Discussion

A. Testing Scenario

For this testing there will be five dataset in the form of graph with various average degree. The five graphs has various comparison of Node and Edge numbers. Each graph will be processed by applying GRAC algorithm. The implementation result will be analyzed to know the compression result of Graph Database. After GRAC is implemented into five graph input, five cluster graphs are resulted. The next step will be comparison between Node and Cluster as well as Edge and Cluster Edge. Besides that, there will be average degree claculation. After that the comparison is conducted and the result is analyzed. The compression level of Cluster Graph that are resulted will be measured by parameter of Edge Reduction and database compression rate. The use of GRAC (Graph Algorithm Clustering) in Graph Database Compression will reduce the number of Edge in a graph significantly. This occurs because the compilation of single node into hierarchy of Cluster Node. As a result for every Cluster Node forming, some Edges can be abstracted into only one Cluster Edge. The followings are the visualization of Edge decreasing and Cluster Edge forming.

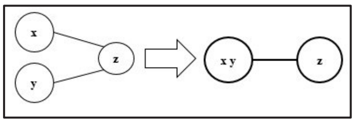

Fig. 3. The example of edge decreasing

In the formation of Graph Database Compression, the level of compression can be calculated using the comparison of graph Database size before and after the compression is conducted. By deciding the file size in advance and getting the compressed file, the analysis of Compression rate can be gained.

The database performance is tested using the Query Read on the former database and the latter database which has been converted into Cluster Graph. Query Read that is conducted is the searching of neighbor Node with the depth of 1-5. The measurement of each depth is conducted to five former nodes that are decided randomly. The length of time from five neighbors search for each depth is then averaged. The result of the time measurement is used to compare graph performance. After that, the performance tersting will be conducted on Normal Graph and Cluster Graph. This performance testing is conducted by doing the Read Query to find the neighborhood Node from the depth of 1 to the depth of 5. From this testing, the difference of the time needed to do the Query in Normal Graph and Cluster Graph can be seen.

The other result of Read Query can also be used to check whether the Cluster Graph, resulted from GRAC method, has the same information as the original graph. The checking of Loseless compression can be done by checking the number of neighbors from the Query result. If the Query output on original graph and output of Query on Cluster graph are always the same, it can be concluded that the compression that is done is Loseless Compression. Here are the visualization of Query Read from the depth of 1 to the Depth of 5.

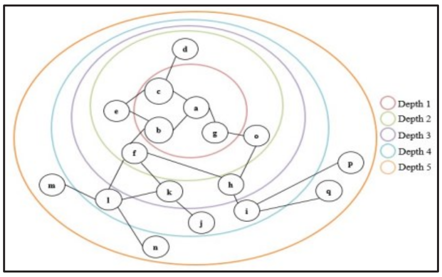

Fig. 4. The example of read query from the depth of 1 to the depth of 5

B. Data input specification

The graph data that are used in this research are collected from SNAP website. There are five graphs that consist of collaboration data among scientific journal writers that are found in Arxiv.org database. The five graphs are from various fields i.e. Astro Physics, Condensed Matter, General Relativity, High Energy Physics, and High Energy Physics Theory.

TABLE I. INPUT FILE AND GRAPH SPECIFICATION

File Graph	Original Size(KB)	Compressed Size (KB)	Compression Rate
AstroPh	5.160	1.337	74.08%
HepPh	2.949	465	84.23%
CondMat	2.363	818	65.38%
HepTh	643	263	59.09%
GrQc	344	111	67.73%

From table 1 it can be observed that the five graphs have numbers combination of node, Edge and various average degree. This system is expected to be implemented on graph with large number of Node and Edge.

C. Implementation Analysis of GRAC(Graph Algorithm Clustering) and Edge Reduction

After GRAC is implemented into five input graphs, five cluster graphs are resulted. Each file of AstroPh, HepPh CondMat, HepTh and GrQc has various numbers. Node is succesfully reduced by Cluster Node as the Edge which is successfully reduced by Cluster Edge.

In the implementation of GRAC, it can be observed that the numbers of Edge decrease since they form Cluster Edge. This also occurs because the single Node accumulation into hierarchical Cluster Node so that for each Cluster Node formation, some edges can be abstracted into only one Cluster Edge.

From the experiments on five graphs, this Algorithm can decrease the number of Edge at maximum into 76,89%. This number is achieved in HepPh database that has average degree of 1,74. The fewest reduced Edge is on HepTh database i.e. 42,90%, not more than 50%. The average degree of HepTh is 5,26 which has the same smallest degree. The followings are the result of GRAC implementation and Edge Reduction in the form of table.

TABLE II. GRAC IMPLEMENTATION ANALYSIS RESULT

File Graph	Normal Graph		Cluster Graph		Edge Reduction
	Node	Edge	Cluster Node	Cluster Edge	
AstroPh	18771	198050	10955	85902	56,62%
HepPh	12006	118489	6030	27379	76,89%
CondMat	23133	93439	11154	45530	51,27%
HepTh	9875	25973	3519	14829	42,90%
GrQc	5241	14484	1988	5926	59,08%

In order to see comparison of Node and Cluster Node numbers in visualization, the comparison graphics will be made. The Comparison graphics consist of two kinds i.e the comparison of Node and Cluster Node numbers as well as comparison of Edge and Cluster Edge numbers. Here are the result:

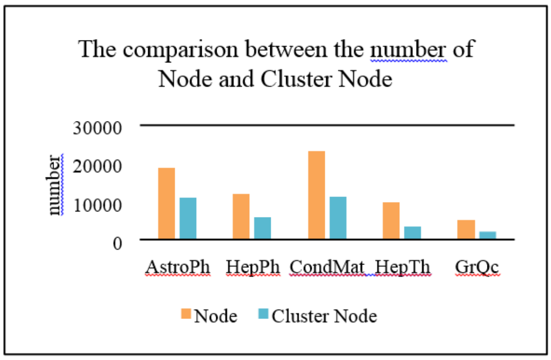

Fig. 5. Comparative Analysis of Node and Cluster Numbers

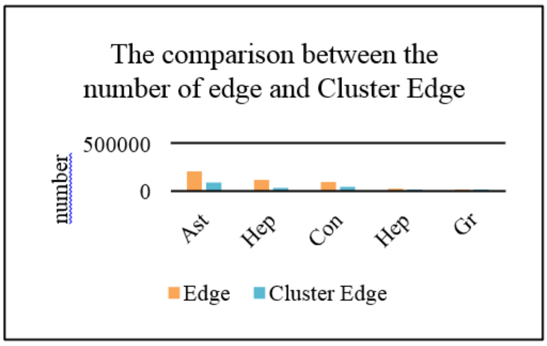

Fig. 6. Comparative Analysis of Edge and Cluster Node Numbers

Each Cluster Graph is then written into Cluster Graph file. Each cluster graph file is divided into two parts. The first part consists of Cluster Node. Every line consists of integer which is index cluster node as well as list of Nodes and Cluster Nodes in the Cluster Node. The second part consists of Cluster Edge Data that connects a Node or Cluster with Node or other Cluster Nodes.

D. *Compression Rate Analysis*

The size of five files have different results. The largest size of file is in Astroph of 5160 KB and the compressed size is 1337 KB. It has 74.08% Compression Rate. In HepPh the size is 2949 KB and the compression rate size is 465 KB. It has the largest size of compression rate of all other files (84.23%). In CondMat, the size is 2.3.63 KB and the compressed size is 818 KB. It has Compression rate size of 65.38%. In HepTh the size is 643 KB and the compressed size is 263 KB. It has the smallest compression rate size of 59.09%. In GrQc the size is 344 KB and the compressed size is 111 KB. It has 67.73% Compression Rate.

TABLE III. ANALYSIS RESULT OF COMPRESSION RATE

Graph File	Size (KB)	Numbers of nodes	Numbers of edges	Average Degrees
AstroPh	5.160	18.771	198.050	21,10
HepPh	2.949	12.006	118.489	19,74
CondMat	2.363	23.133	93.439	8,08
HepTh	643	9.875	25.973	5,26
GrQc	344	5.241	14.484	5,53

E. *Graph Performance Analysis*

The testing is conducted on Normal Graph and Cluster Graph by doing Query of neighbor searching with the depth of 1-5. For each depth, five experiments are conducted with different nodes that are chosen randomly. The time length of five experiments for each depth are then averaged.

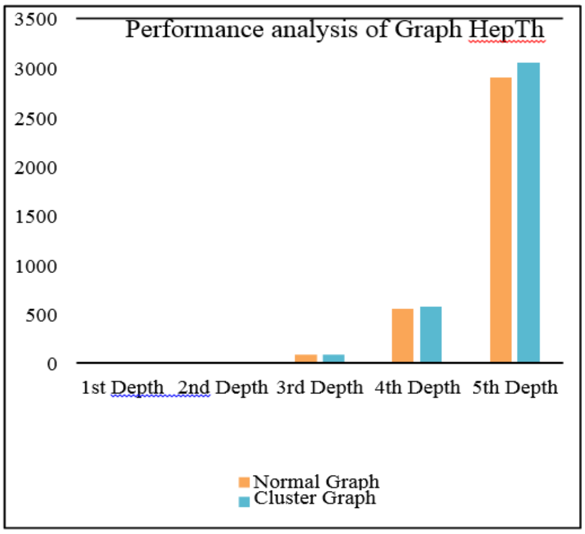

Fig. 7. The performance analysis of Graph HepTh

From figure 7, Query Graph execution in Depth 1 to Depth 5 can be conducted with sufficient time. The change can be seen in Depth 3 and Depth 4 in which the time becomes slower in terms of its processing. There is also a change in the early process in Depth 1 and 2.

F. *Testing Analysis of Lossless Compression*

To test whether the compression delete the information or not could be conducted with Read Query. If there are no item of information lost in the answer set, it means that the compression system is losless.

The results of the read query in this research are very satisfied. There are no item of information which is lost in the answer set.

V. CONCLUSION

Based on the analysis in this paper it can be concluded that database compression using the method could produce a lossless graph compression.

The testing analysis shows that the graph be compressed at maximum 84.23% and the average of compression result of the five files is 70.10% so that the graph database can decrease the size of dataset file. Edge Reduction decreases the Edge and is represented by Cluster Edge. By the decrease of Edge, the dataset can be more effective in Edge representation. The decrease of Edge can also influence the result of dataset compression. The performance result of time average of Query Read Graph Database Compression is slower than Graph database although it is not in a constant way.

REFERENCES

[1] I. Robinson, J. Webber, E. Eifrem, 2013. Graph Databases, O'Reilly.

[2] Zhang et al., 2012, Graph Degree Linkage: Agglomerative Clustering on a Directed Graph, ECCV 2012.

[3] Daniel Mullner, 2011, Modern Hierarchical, Agglomerative Clustering Algorithm,DMS 2011.

[4] Niwattanakul, Singthongchai, Naenudorn, Wanapu, 2013, Using of Jaccard Coefficient for Keyword Similarity, IMECS 2013 Vol I, Hong Kong.

[5] Jaccard's Coefficient, Kardi Teknomo's Page, [Online]. Available: http://people.revoledu.com/kardi/tutorial/Similarity/Jaccard.html. [Diakses 02 Januari 2014].

[6] Greedy algorithm, Princeton, [Online]. Available: http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Greedy_algorithm.html. [Diakses 02 Januari 2014]..