Mining of Resource Usage Using Evoc Algorithm in Grid Environment

Kee Sim Ee, Chan Huah Yong, Fazilah Haron
School of Computer Sciences
Universiti Sains Malaysia, 11800 Minden Penang, Malaysia.
Tel:604-6534390, Fax:604-6573335
{sharon,hychan,fazilah}@cs.usm.my

***Abstract*- This paper addresses the new algorithm namely Evolving Clustering (Evoc Algorithm) which is an improved version of Evolving Clustering Method (ECM). The algorithm has been evaluated using three main criteria; that is dynamicity, accuracy and the ability to identify the stable cluster members. Our results show the improvements of the algorithm to process the data in an on-line mode in the evaluation of the algorithm's dynamicity and accuracy criteria compare to other existing clustering techniques. Furthermore, the stability evaluation was a success where we were able to identify the stable cluster members from the filtered stable clusters. However, the result was affected by three factors namely threshold value, stability value and stability hour.**

Keywords: evoc algorithm, evolving clustering, resource usage, grid.

1. INTRODUCTION

Grid computing or simply grid is a generic term given to technologies designed to make pools of distributed computer resources available on-demand. It has become one of the latest buzzwords in the IT industry. Grid provides wide-spread, dynamic, flexible and coordinated sharing of geographically distributed heterogeneous networked resources, among dynamic user groups. It is an innovative approach that leverages existing IT infrastructure to optimize computer resources and manage data and computing workloads [1].

The number of computing nodes and users participating in the grid environment is in increase and may reach up to thousands or millions in a grid environment. The abundance of these resources forges new problems, such as how to collect the massive amounts of evolving resources in real time and extract the useful information from them. Furthermore, at a glance, these resources are not ordered, random and chaotic where normal user is not able to easily discover any knowledge or meaningful information from them. These resources will be useful if

- Their implicit and underlying meaningful pattern can be extracted to form a new knowledge for advanced usage,
- The issues of dynamics and large amount of data are well processed in real time mode using dynamic clustering method.

In order to deal with these requirements, clustering is proposed as one of the best ways in terms of processing large set of raw data and turning these data into meaningful information. It is a discovery process that groups a set of data such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized [2]. This technique sorts through data to identify patterns and establish relationships.

By applying the **Evoc** algorithm in grid environments, the problem of processing massive amount of data in real time mode will be solved. Using these clustered data, we proposed a stability feature to identify the stable computers within a certain period of time which can give very useful information for the purpose of predicting and monitoring of resources.

II. RELATED WORK

The research that has been done in [3] suggested the epochal behavior of CPU usage exist, long and significant. According to Peter A. Dinda, the load traces might seem at first glance to be random and unpredictable might be having the structure that can be exploited and the results suggested that load traces do indeed have some structure in the form of clearly identifiable properties.

CURE [4] and CHAMELEON [5] are two of the popular algorithm for hierarchical clustering technique. However, several problems are found in attempting apply these algorithm to grid resources. The major problem for this technique is it is a rigid method where it

never revisits any cluster member once the clusters are constructed. Furthermore, the clustering process is a one-pass process where the data are clustered for once only without any continuously processing. This will cause the result's accuracy affected. Apart from that, in certain situations the unrelated cluster members sometimes might join together as one cluster for the sake to get the clustering process done for all data.

PAM (Partitioning Around Medoids) which was proposed by Kaufman and Rousseeuw [6] computationally is quite inefficient for large value of n and k; n is the number of data and k is number of clusters. According to the experimental result that was mentioned in [6], PAM works satisfactorily for small data sets only for around 100 objects in 5 clusters but does not work well for medium and large data sets. In grid environment provided with massive data which might reach until thousands of data, this algorithm for sure is not suitable as the value of n and k grows higher, the process will turn slow. Furthermore, since it
is not scale well for large set of data especially grid resources, thus inaccuracy of clustering result might happen as well.

CLARA (Clustering LARge Applications) algorithm which was proposed by Kaufman and Rousseeuw as well works better than PAM for large data sets but yet it is an off-line mode clustering. Thus, it is not in the consideration to become the chosen technique for our research work. CLARANS (Clustering Large Applications based upon RANdomized Search which combines PAM and CLARA algorithms) algorithm which was proposed by Ng and Han [7, 8] is better among two algorithms mentioned previously. Experimental results here shown to be more effective than both algorithms. However, same reason as CLARA, it does not perform the clustering process in real time mode which is not so suitable to be used for grid resources processing.

For the k-nearest neighbor, the technique generally successful with the classifiers k=1. Just like the partitioning technique, the user needs to determine the value of parameter of K where k stands for the number of nearest neighbors to be taken into the consideration. Thus, it is not chosen by us to be applied on grid resources in our research work as a very large value of k can destroy the locality of the estimation. This is due to farther examples might be taken into the consideration in the decision making which class that data sample belongs to which will cause the inaccuracy of clustered result. This method leads to a high cost of computation because the distance of each query instance to all training samples needs to be computed. Apart from the above reasons, the k-nearest neighbor is a supervised learning method and the grid resources that we will process are dynamic. We believe that the unsupervised method of clustering is more suitable to process the grid resources instead of supervised technique. Thus, k-nearest neighbor is not in our choice.

The stability metric that was proposed by [9] is more towards in determining the number of clusters automatically instead of searching for stable cluster members which is not related to our research mode. Furthermore, this technique only works with both hierarchical and partitioning clustering algorithms that do not work in on-line mode. Thus, it is not chosen for our research work for the cluster members' stability part.

Evolving Clustering Method (ECM) which was proposed by Qun Song and Nikola Kasabov [10, 11] is an on-line, dynamic clustering which performs a one-pass, maximum distance-based clustering process without any optimization. From the experiments that were carried out in [12], the author concluded that ECM allows for unsupervised, lifelong, on-line modeling of evolving processes and it is much faster than the off-line clustering techniques.

The grid resources are categorized as dynamic type of data where the data values are changing from time to time. From the research [11], we found that ECM is more adaptive to the input stream of dynamic data. Furthermore, the created cluster for this technique is able to increase its size to adapt to the increasing size of data. The created cluster will not be updated when the cluster's radius, R_u equals to threshold values, *Dthr* [10, 11, 12].

Considering all the advantages of the ECM method, ECM is recommended to be applied in our system due to its characteristic that can work partially in on-line mode (Table 1). Meanwhile, there are some weaknesses that we need to consider for this algorithm as well when applying this technique on grid resources. ECM algorithm even though allows for cluster radius size increment to support the increasing size of data in on-line mode; however it is not able to minimize its size in on-line mode at the same time. Thus, we propose to take ECM as our clustering technique to process the grid resources, but there are some parts that still need to be improved to allow it to be suitable to be applied on grid resources. We will enhance this technique in this research by enabling it to increase and decrease the cluster size in on-line mode. Moreover, this algorithm will be included the stability feature which allows it to search for the stable cluster members in the stable clusters.

TABLE 1

COMPARISON OF CLUSTERING TECHNIQUES

Clustering Techniques		Adaptive to dynamic data stream?	Off-line or On-line mode?	Stability Features?	One-passed mode?
Hierarchical	Agglomerative	No	Off-line	Yes	Yes
	Divisive	No	Off-line		Yes
Partitioning	PAM	No	Off-line	Yes	Yes
	CLARA	No	Off-line		Yes
	CLARANS	No	Off-line		Yes
	k-means	No	Off-line		Yes
Supervised Learning	k-nearest neighbor	No	Off-line	No	Yes
On-line mode Clustering	ECM	Yes	Partially on-line	No	Yes

III. EVOC ALGORITHM

In this paper, we present the design of improved Evolving Clustering Method (ECM) algorithm and this will be applied in the process of clustering dynamic data (CPU usage) in a grid environment. We name this algorithm *Evoc* algorithm. This algorithm performs evolving and dynamic type of clustering process where the size of a cluster can be increased and decreased. The aim of this work is to show that *Evoc* algorithm can work better and more dynamic compare to the existing off-line and on-line mode of clustering techniques. In addition, we present experimental evidence to show that the performance of this algorithm is more dynamic and able to give the result of more stable clusters and instances.

Preliminaries

Before go for the further explanation of this clustering process, we will first begin with by defining the concept of *Evoc* algorithm more precisely.

- x_i *denotes the sample of data from the input stream of data, where i = 1,2,3,4... n data*
- Cc_j^k *denotes the cluster centre, where the k = 1, 2, 3, 4... n number of items in the cluster and j = 1, 2, 3, 4... n number of clusters.*
- C_j^k *denotes the cluster, where the k = 1, 2, 3, 4 ... n number of items in t he cluster and j = 1, 2, 3, 4 ... number of clusters.*
- Ru_j^k *denotes the cluster radius, where the k = 1, 2, 3, 4 ... n number of items in t he cluster and j = 1, 2, 3, 4 ... number of clusters.*
- R'_t *denotes the radius size for the previous period of time, t-1.*
- R' *denotes the radius size for current time, t.*
- A data point represents a computer node.

Evoc Algorithm: Clustering Process

We propose the Evolving Clustering Method (ECM) [10, 11, 12] to solve this problem which collect large volumes of data and turn them into useful knowledge. The basic principle of this clustering technique is that the process starts with an empty set of clusters. The cluster radius, R_u is initially set with the value of 0 and the cluster centre, C_c is located when a new cluster is created. With the incoming data, some existing clusters will be updated or some other actions will be taken on them. The cluster's size may evolve in three different situations:

- Update the cluster size by updating its cluster radius, R_u
- Do nothing to the cluster size but a new member will be put in that cluster and
- When the data does not fit in any of the existing clusters; a new cluster will be created.

However, some modifications have been made to ECM algorithm enable the cluster size to shrink as well. Thus, our proposed Evoc algorithm (which is the improved version of ECM) able to work in two major ways which are shrink and enlarge the cluster size [13].

For every new data point coming in, the checking needs to be done to see if there is any same data point/computer node stay in any existing cluster. If there is none same data point stay in any existing cluster, then the clustering process can be preceded. Else if the same data point already exists in any existing already created cluster, then the data has to be removed from that cluster. For instance, in previous period of time computer node A' already stayed in Cluster A, and thus when the new point value of computer node A coming in, the previous computer node A' has to be removed from the cluster A and the re-clustering process will be going on for the new point.

Evoc: Stability Methodology

After gone through the clustering process in the first level, there will be many created clusters according to different computer's behaviors in every hour. However, until this level, we cannot make any conclusion about the stable data from this information. Thus, we propose the stability methodology [13] to get to identify the stable cluster as well as those stable instances which are stable and staying in the stable cluster.

Since the clustering process is dynamic, it is hard to detect the stable cluster as well as the stable computer

nodes in the process. The clustering process will only show the result the groups of the similar behavior of computer nodes in the certain period of time but not the stable computer nodes. Thus, the stability methodology is been added into the original ECM algorithm as extension for the clustering process to get to know the stable computer nodes which is stay in the stable cluster as well.

In terms of stability, a cluster can only be considered as stable when its radius changing is not obvious from time to time. A cluster's stability can be calculated by using equation (6)

$$\text{Stability} = \frac{|R_{-t} - R_t|}{R_{-t}} \times 100\% \quad (6)$$

where R_{-t} stands for radius in previous hour, *h-1* and R_t stands for the current hour's cluster radius size. The less the value of the stability, the more stable that cluster is. In our research, we set the value of <20% for a cluster to be considered as stable. Any cluster with the stability value which is <20% at time *t* will classified as stable cluster.

From the aspect of the instances / computer nodes, it can only considered as stable if it stays in the same cluster for certain period of time or more continuously for instance 3 hours and the associated cluster(s) during this period of time is stable (which is <20% as mentioned above), otherwise the cluster member is deemed unstable.

IV. EXPERIMENTAL RESULT AND DISCUSSIONS

In this section, we will present our improved ECM algorithm, which is Evoc algorithm in terms of its dynamicity, stability as well as its accuracy compared to supervise learning based on clustering. In the following sections, we will discuss the experimental results and works. In the experiments, we use different threshold values and compare the results with several others existing clustering methods. The experiments were carried out on three aspects which are i) dynamicity ii) the ability to identify stable cluster member and iii) the accuracy of this unsupervised learning method compare to supervised learning clustering method.

Evaluation of Dynamicity

The experiments were carried out using 4 different threshold values, 0.03, 0.05, 0.07 and 0.09. We plot the changes of the cluster radius size in using threshold value 0.05 (Figure 1).

In the experiment with threshold value of 0.05, we found that when the 6th hour is reached, cluster 1, C1 from ECM algorithm (upper line, blue color) encounters error in its cluster size and the following data stream created in new cluster 5 causes error in categorizing the data. However, Evoc algorithm (lower line, red color) still works well until the end of the experiment. This shows that Evoc algorithm is more dynamic compared to ECM when both algorithms are run on-line continuously without stopping. For Evoc algorithm, the C1 cluster shows a radius size value of 0 after the 5th hour. This is because the cluster no longer has any cluster member and its cluster radius is reduced to zero. Thus, the cluster 1, C1 was deleted from the cluster list after that.

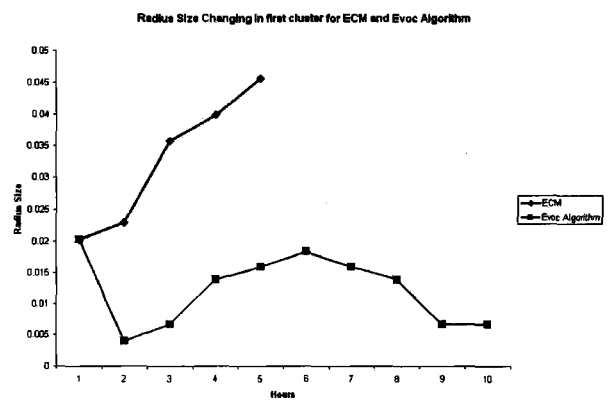

Fig. 1.The changes in radius size of the first cluster, C1 for ECM and Evoc algorithm with threshold value = 0.05.

We can deduce from Figure 1 that the ECM stops the clustering process after the 5th hour. Moreover, we also can derive that the cluster radius for ECM keeps on increasing as it only allows the cluster size to increase in an on-line mode. ECM allows the cluster radius to decrease in off-line mode. For Evoc algorithm, the line shows the ups and downs trend in its cluster radius. This shows that Evoc algorithm is able to perform the process of maximizing and minimizing the cluster radius in on-line mode. In addition, the result also proofs that Evoc algorithm is more dynamic.

The results have shown that Evoc algorithm is much more dynamic than the original Evolving Clustering Method (ECM) algorithm. Evoc algorithm can handle and process massive amount of data without any significant error rate. ECM algorithm cannot deal with continuous data stream for a long period of time if the constraint optimization is not done in the off-line mode to control the cluster radius size. In short, Evoc

algorithm is more suitable for on-line clustering process compared to ECM algorithm or any other existing clustering techniques.

Evaluation of Accuracy

The experiments for evaluating the accuracy are done in two scenarios. First scenario uses symmetric data while the second scenario uses random data. Five experiments were carried out to evaluate the accuracy for three clustering techniques using different value of *k* for k-means and threshold value for k-nearest neighbor and Evoc algorithm.

Out of 5 experiments that were conducted, Evoc has the error rate of ~9.58% whereas k-means error rate reaches up to ~20.15%. The less error rate proves that our algorithm is much more dynamic as well as more accurate. The higher error rate for k-means technique happens when the *k* value gets higher. This is one of the weaknesses for this technique. The user has to fix the *k* value before the clustering process is starts and sometimes it is difficult to predict value *k*. In this case, if the determined value of *k* is not 'right', it might cause some unrelated data points to join together and affects the clustering result. For Evoc algorithm, the accuracy goes down beyond the higher value of threshold value as well. However, its accuracy does not decrease as drastically as k-means technique (Figure 2).

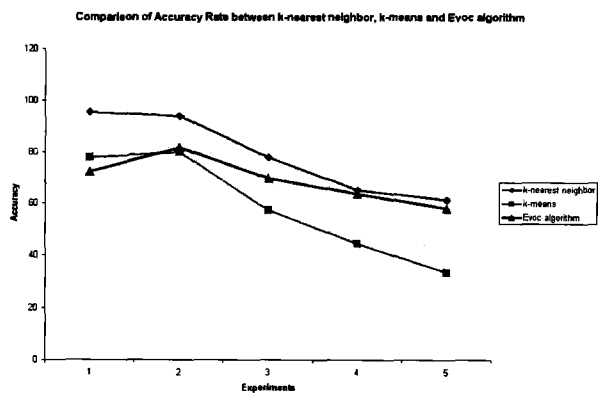

Fig. 2.The Comparison of Accuracy Rate between k-Nearest Neighbor, k-Means and Evoc Algorithm for Symmetric Data Set

In summary, we can see that Evoc algorithm accuracy rate is not as good as supervised learning clustering, that is k-nearest neighbor. However, if compared to unsupervised learning clustering technique which is k-means, Evoc algorithm performs better. Even though Evoc algorithm is categorized as unsupervised learning, it is still better than existing unsupervised clustering techniques.

For the second scenario (Figure 3), results from the first and second experiment using k-means technique (lower line, red color) show the highest accuracy among the three techniques. This is because the *k* value is pre-defined in both experiments and it is suitable for random data. However, Evoc algorithm (upper line, black color) starts to display highest accuracy among three techniques from the third until fifth experiments. K-means's accuracy plunges drastically (Figure 2). This could be due to the real number of cluster that should be created is less than the pre-defined *k* value in each experiments and this causes some unwanted clusters to be created. However, both techniques which are k-nearest neighbor (middle line, green color) and Evoc display the accuracy rate which is quite close to each other which can be seen in Figure 3.

For both experiments, we found interesting information of error gap for the techniques that we have used on. Error gap is the different between the highest value and the lowest value of accuracy result that was obtained. Taken first experiment as example, even though the k-nearest neighbor is the best method among all the three methods that shows the highest accuracy, its error gap value is higher than Evoc algorithm. The k-nearest neighbor has an error rate within ~95.5% - ~61.3% where the error gap is around ~34.2%. For Evoc algorithm, it only has an error gap around ~13.8% for error rate within ~72.3% - ~58.5%. K-means has the highest value of error gap among the three methods where its error gap is ~44.7%. This can be seen from the decrease of accuracy in Figure 2.

Evoc algorithm is the most stable method from the results shown in all experiments. The smallest error gap for the Evoc algorithm can become one of the main factors in choosing this algorithm for the clustering process as it is able to give better result with low error compared to the other two methods, k-nearest neighbor and k-means. The second experiment (Figure 3) also showed the similar trend where the Evoc algorithm still has the lowest error gap among three clustering techniques where it has an error gap of ~15.3% compared to k-nearest neighbor (~17.7%) and k-means (~45.5%). K-means still gives the highest error gap result and it is not so suitable to be used in processing dynamic data.

The overall result shows that Evoc algorithm performs better than the k-nearest neighbor and k-means clustering techniques. Overall, it can be concluded that Evoc algorithm is more suitable to be used in clustering random data set.

Stability

Stability is a new feature that is proposed for the clustering technique. The results from the experiments have successfully shown that Evoc algorithm is able to identify stable cluster members/computers that are filtered from a pool of stable clusters. However, a cluster member's stability is affected by three main factors which are threshold value, stability hour and stability value.

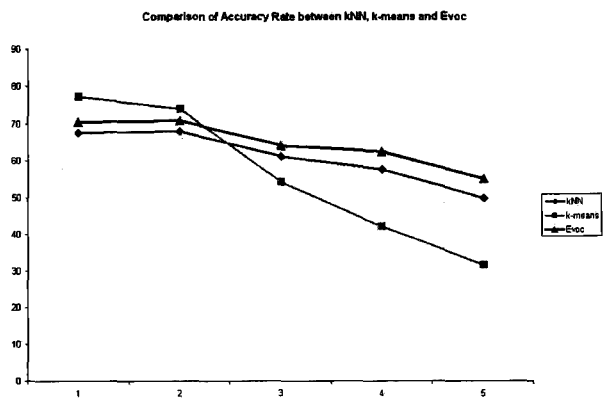

Fig. 3.The Comparison of Accuracy Rate between k-Nearest Neighbor, k-Means and Evoc Algorithm for Random Data Set

In an experiment where the stability value is a variable and stability hour is static, the lower threshold value produces more number of stable cluster members. We make the conclusion that when threshold value is lower; this leads to more creation of clusters. When there are a bigger number of clusters, more stable clusters can be identified.

In the experiment where the stability value is static and the stability hour is a variable, we can say that the higher the threshold value, the more the number of stable cluster members can be identified. This situation happens when the threshold value is higher; this leads to a lesser number of clusters created. When the number of clusters is small, the behavior range (in percentage) for these clusters becomes bigger. For instance, when 5 clusters are created, the range of behavior for each cluster takes up 20%. However, if only 2 clusters are created, the range of behavior for each of these 2 clusters becomes higher. In this case, each cluster's behavior range occupies 50% and more cluster members will be able to stay in these 2 clusters. This may also allow some cluster members to stay for a longer period of time in one of these created clusters. Thus, more stable clusters and stable cluster members will be identified in this case.

In addition to that, the stability hour is also one of the factors that affect the result of cluster members' stability. We found that the higher the value of the stability hour, the lower the number of the stable cluster members that can be generated. Normally, in the grid resources, parameters such as CPU usage that we are using in our work are very "active" and the value changes from time to time. Thus, the CPU usage is hardly stable within a longer period of time (higher Stability Hours). This leads to a smaller number of stable cluster members that can be identified. However, this factor is also affected by the threshold value. In the experiment that was conducted, there is no stable cluster members identified for the threshold value 0.01 but several cluster members are identified for the threshold value of 0.05. We can conclude that a higher threshold value gives a bigger impact to the stability result.

The stability value plays an important role in producing the stability results. The higher the stability value, the more the number of stable clusters can be identified. This is due to fact that the total turnover rate for the cluster radius size is very high. When cluster member moves in and out from the cluster, the cluster radius size changes accordingly. Since most of the clusters radius size changes are quite high, a high stability value enables more clusters to satisfy this criterion which indirectly indicates more stable cluster members will be identified.

Further research is required on these three factors to determine the best combination of these three values in finding the stability of a cluster member. The combination of these three values should be placed in Evoc algorithm in order to enhance the stability feature.

V. FUTURE WORK AND CONCLUSION

This research work is driven by the objective to produce a better ECM algorithm for clustering technique. This improved algorithm is called Evoc algorithm. It able to process data in a more dynamic way and give the result of stable cluster member which is cannot be performed in any of the other existing clustering algorithms.

The Evoc algorithm has been evaluated using three main criteria; that is dynamicity, accuracy and the ability to identify the stable cluster members. Our results show the improvements of the algorithm to process the data in an on-line mode when evaluating algorithm's dynamicity. Evoc algorithm can handle and process massive amount of data without any significant error rate. From the experiment, we can conclude that the Evoc algorithm is more dynamic than ECM algorithm.

The second experiment, the Evoc algorithm also gives more accurate result for random data sets

(accuracy: ~64.60%) compare with symmetric data sets (accuracy: ~60.83%) and able to identify stable cluster members from a pool of stable clusters. However, we also found that the cluster members' stability is affected by three main factors which are the threshold value, the stability value and the stability hours.

The proposed new add-in feature for the clustering technique which is stability identification feature cannot be found in any other existing clustering techniques. Most of the existing clustering techniques nowadays group similar behavior of a group of data but are not able to identify the stable cluster members from those clusters. The proposed feature allows the identification of the stable cluster member apart from giving the result of the behavior of the clustered data. We believe that this feature can be applied in other domains/problems, for instance to study computer behaviors daily patterns identification for the purpose of predicting the resource usage in grid environment.

REFERENCES

[1] Foster, I. and C. Kesselman., The Grid: Blueprint for the New Computing Infrastructure. Morgan Kaufman Publishers, Inc., 1998.

[2] Chen, M.S., Han, J. and Yu, P.S. (1996). Data Mining: An Overview from database perspective. *IEEE Transactions on Knowledge and Data Eng.*, 866-883.

[3] Dinda, Peter A. (1998) *The Statistical Properties of Host Load*, Proceedings of the Fourth Workshop on Languages, Compilers.

[4] Guha, S., Rastogi, R. and Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 73--84, New York, 1998

[5] KARYPIS, G., HAN, E-H., and KUMAR, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, COMPUTER, 32, 68-75.

[6] Kaufman, L. and Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley and Sons.

[7] Jain, A.K. and Dubes R.C. (1988). *Algorithms for Clustering Data*, Prentice Hall, 1988.

[8] Ng, Raymond T. and Jiawei Han (2002). Member, IEEE Computer Society, *CLARANS: A Method for Clustering Objects for Spatial Data Mining*, IEEE Transactions on Knowledge and Data Engineering, Vol 14, No 5, September / October 2002, pg 1003 – 1016

[9] Ben-Hur, A., Elisseff, A., and Guyon, I. *A Stability Based Method for Discovering Structure in Clustered Data*, in Proceedings of the Pacific Symposium on Biocomputing (PSB2002), January 2002, Kaua'I, HI.

[10] Qun Song, Nikola Kasabov (2000). *Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS): On-line Learning and Application for Time-Series Prediction,* Proc. 6th International Conference on Soft Computing, 696-701, Iizuka, Fukuoka, Japan.

[11] Nikola Kasabov, Qun Song (2002). *DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction* Fuzzy Systems, IEEE Transactions on Volume 10, Issue 2, April 2002 Page(s):144 – 154

[12] Nikola Kassabov (2003). *Evolving Connectionist Systems, Methods and Applications in Bioinformatics, Brain Study and Intelligent Machine*, Springer-Verlag.

[13] Kee, Sim Ee, Chan, Huah Yong and Fazilah Haron (2006). *Clustering of CPU Usage in Grid Environment using Evoc Algorithm*, Malaysia Journal of Computer Science, Volume 19, No.2.