# CLUSTERING OF CPU USAGE DATA IN GRID ENVIRONMENT USING EVOC ALGORITHM

***Chan Huah Yong, Kee Sim Ee, Fazilah Haron***
School of Computer Sciences
Universiti Sains Malaysia, 11800 Penang, Malaysia.
Email:{hychan, sharon, fazilah}@cs.usm.my

***ABSTRACT***

*Clustering is a process of organizing objects into groups whose members are similar in some way. It is one of the data mining techniques is an unsupervised learning. In a grid environment, the number of computing nodes and users may reach up to thousands or millions. The grid is said to be dynamic in that the behaviors and values of these resources change all the time. Hence, these data are not suitable to be processed in an off-line mode. The existing clustering techniques today however emphasize more on the data's behaviors categorization but not the data's stability. Furthermore, the normal clustering techniques are more suitable to be used for static data type in an off-line mode. This paper addresses these issues by presenting an* ***E****volving* ***C****lustering (****Evoc Algorithm)*** *which is an improved version of Evolving Clustering Method (ECM). We apply both methods on CPU usage to identify computers behaviors. 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 highly affected by three factors namely threshold value, stability value and stability hour.*

***Keywords: Clustering, Grid, Data Stability, Evolving Clustering (Evoc), Evolving Clustering Method (ECM)***

## 1.0 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

i) Their implicit and underlying meaningful pattern can be extracted to form new knowledge for advanced usage,

ii) 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, Evoc 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. The improved version of Evolving Clustering Method is used because the nature of this resource problem is to group the similar behavior of usage pattern in the real time mode for studies. 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.

## 2.0 MOTIVATIONS AND RELATED WORK

The research in [3] concluded that unforeseen computer behavior in the systems is common. A.K. Jain considers those unexpected behavior as a "bug" or if it's benign, it is called as a feature. However, the paper from [3] only describes the method to predict the computer behavior that it will behave within acceptable limits with the purpose of avoiding dramatic loss of life, property and money. This research is not related to our current research study. The research that has been done in [4] 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.

There is another research work [14] which discovers the grid resources usage patterns (CPU usage) using suffix tree method. This paper proposes to discover the CPU usage pattern from its historical usage data to better understand the CPU usage does show the behavior and thus its behavior for next level is believed can be predicted. Network Weather Service (NWS) [15] is one of the most popular resource prediction methods. NWS was developed for the use of schedule in a networked computational environment by using a series of forecasting methods to forecast a recourse performance and then choose the method that yields the lowest mean square error or lowest mean percentage error.

CURE [5] and CHAMELEON [6] 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. However, there are some good things in this technique as well. The hierarchical method better in handling any forms of similarity or distance and it is applicable to any attributes types. The advantage is suitable to be applied in grid resources but there are several arisen issues as mentioned in previous part that raise some arguments to their applicability in cluster the grid resources.

Partitioning clustering is a process where it partitions the data into k clusters. Given a set of objects and a partition criterion, partition the objects into clusters based on the similarity that the objects have with each other to match the specified criterion. One of the popular algorithms for partitioning technique is PAM (Partitioning Around Medoids). The cost iteration for PAM algorithm proposed by Kaufman and Rousseeuw [7] is $O(k(n-)^2)$ where its 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 [7], 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 [8, 9] 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 research in [10] proposed that cluster stability to be one of the criterions in determining the structure of the data. According to Jefferey, the cluster stability-method which was proposed by [11] is one of the methods in finding the optimal number of clusters. The technique samples a space of clustering for each choice of k, and uses a clustering similarity metric to generate a distribution of stability values. This distribution is then used to choose the most stable clustering. The author claimed that this method was used in their research work as it is an independent technique and furthermore does not make any assumptions as to cluster shape or density as some other methods do. The technique does not search for stable cluster members which are 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 being chosen in our research work for the cluster members' stability part.

## 3.0 EVOLVING CLUSTERING (EVOC ALGORITHM) OVERVIEW

We have chosen Evolving Clustering Method (ECM) [12, 13] in our research work as the main clustering technique to be improved. It is one of the evolving clustering that can work partially in on-line mode to other off-line mode clustering techniques, for instance hierarchical and partitioning clustering. On-line mode means that the data streams keep on coming and changing, whereas off-line mode the data is well gather and won't change for the single pass processing. The off-line mode techniques are not suitable for on-line mode processing because it would need multiple pass processing. ECM is more suitable to be used to process the grid resources due it its ability of process the evolving type data by maximizing the cluster size. However, the ECM algorithm only allows for the cluster size to increase in an on-line mode while decreasing cluster size in an off-line mode. We have incorporated the ECM with the following features:

i) Ability to minimize the cluster radius size in an on-line mode
ii) Ability to identify the stable cluster members

### 3.1 The Proposed Flow of Clustering Process in Grid Environment

We divide the clustering process into two main stages. During the first stage, the system will collect the CPU usage data from all related computers. In average the data will be calculated at every hour to identify a data point which can be used as a cluster point during the clustering process. (Fig. 1)

After the data has been collected, the data will be processed and clustered using the improved Evolving Clustering (Evoc Algorithm), Evolving Clustering Method (ECM). Clusters are generated according to their similar behaviors at every hour and these clusters will be used in the second part which is instance's stability identification process.

In order to identify the stable computer for a certain period of time, the stability feature will be used. Cluster size that changes less than 20% from the previous time, *t-1* is regarded as a stable cluster. A stable computer node is only being considered as stable if it is belongs to a stable cluster. The purpose of this stage is to identify the stable cluster members which are computers in this case. We believe that this part is able to give a better and more accurate result of computers' stable behavior.

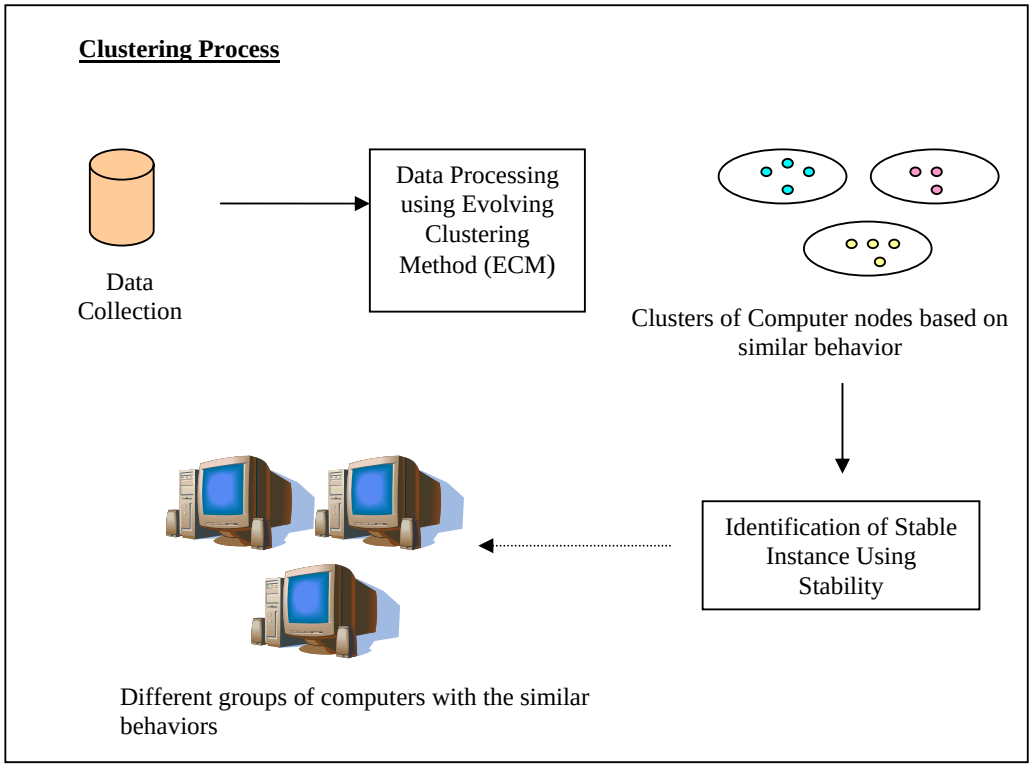


Fig. 1: The Flow of Clustering Process in Grid Environment

### 3.2 Definition

The concepts of our Evoc algorithm are defined as follows:

- $x_i$ denotes the data from the input stream of grid resources, where $i = 1,2,3,4\ldots n$ of data
- $Cc_j^k$ denotes the cluster centre, where $k = 1, 2, 3, 4\ldots n$ number of items in the cluster and $j = 1, 2, 3, 4\ldots n$ number of clusters.
- $C_j^k$ denotes the cluster, where $k = 1, 2, 3, 4 \ldots n$ number of items in the cluster and $j = 1, 2, 3, 4 \ldots$ number of clusters.
- $Ru_j^k$ denotes the cluster radius, where $k = 1, 2, 3, 4 \ldots n$ number of items in the cluster and $j = 1, 2, 3, 4 \ldots n$ number of clusters.
- $R_{\_t}$ denotes the radius size for the previous period of time, $t\text{-}1$.
- $R_t$ denotes the radius size for current time, $t$.
- A data point represents a computer node.

Evoc algorithm can be divided into two main phases, where the first phase (section 3.3) place the data stream into the appropriate cluster and the second phase (section 3.4) calculates the clusters stability and provides the result of stable instance which is the stable computer node.

### 3.3 Clustering Process

The basic principle of Evoc algorithm is that the process starts with an empty set of clusters. The cluster radius, $R_u$ is initially set to 0 and the cluster centre, $C_c$ is located when a new cluster is created. With the next coming in 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:

i) Update the cluster size by updating its cluster radius, $R_u$

ii) Do nothing to the cluster size but a new member will be put in that cluster and,

iii) When the data does not fit to any of the existing clusters; a new cluster will be created.

Some modifications have been made to ECM algorithm to suit the dynamicity of grid environment. We called the improved ECM algorithm as **Evo**lving **C**lustering (Evoc). Evoc algorithm allows:

- A more dynamic way of clustering for a longer period of time.
- The identification of stable cluster members from the generated clusters

Our proposed Evoc algorithm works both ways; shrink and enlarge the cluster size. Each new in coming data point will be checked whether there exist any duplicate in any of the existing cluster. If there is no duplicate, the data point will be placed in any of the existing cluster then only the clustering process can proceed. Otherwise, the data point has to be removed. For example, a previous computer node $A\__t$ exist in Cluster K, and thus when the following next point of computer node A comes in, the previous computer node $A\__t$ has to be removed from the cluster K and the re-clustering process will continue for the next point.

Before removing the previous point $A\__t$ from the cluster, there are three conditions that need to be taken into consideration. $A\__t$ point needs to be determined whether it is the outermost point in that cluster.

- If $A\__t$ is the outermost point for that cluster, check if the point $A\__t$ has the Extended Distance Value (Equation 3)
- If the point $A\__t$ has the Extended Distance Value, then the point will be removed, cluster size will be shrunk and the radius size will be calculated based on the following equation:

$$Ru_{j-1} = 2\big(s(i, j)\big) - d(i, j) \qquad (1)$$

- Else if the point $A\__t$ does not have the Extended Distance Value, then the point will be removed directly from the cluster without any changes to the value of the radius.
- If $A\__t$ is not the outermost point in the cluster, then this point can be removed from the cluster without affecting the cluster and the radius size.

After the point $A\__t$ has been removed from the cluster, the subsequent point $A$ will go through the clustering process. However, if point $A\__t$ is not found in any of the existing cluster, the subsequent point $A$ will proceed with the clustering process only, that is without removing any of the old instances from the clustering process.

There is another situation where the $A\__t$ is removed from the cluster and $A\__t$ is the last member to be removed. In this case, the cluster radius size will become zero or it does not exist any longer.

The clustering continues with the next data point in the data stream. The ***Euclidean Distance, d (i, j)*** between the new data point, $\boldsymbol{x_i}$ an all existing cluster centers, $\boldsymbol{Cc_j}$ will be calculated.(as shown in equation 2)

$$d(i, j) = \| \chi_i - Cc_j \|, j = 1,2,3,4...n \qquad (2)$$

The result will be used to determine which cluster data point $\boldsymbol{x_i}$ belongs to.

If there is a cluster $\boldsymbol{C_m}$ with its centre $\boldsymbol{Cc_m}$**;** cluster radius $\boldsymbol{Ru_m}$ and the minimum distance value *min d (i , j)* (which is between the cluster centre for this cluster and the new incoming point $\boldsymbol{x_i}$) is $\leq \boldsymbol{Ru_m}$**,** then the current data point $\boldsymbol{x_i}$ is considered belongs to this cluster $\boldsymbol{C_m}$.

Given a situation where there exist three clusters and each of their cluster radius size, cluster centre size, Euclidean distance are showed as follow:

Table 1: Generated Result for the Euclidean Distance Calculation

| ***J*** | ***$X_1$*** | ***$Cc_j$*** | ***$Ru_j$*** | ***d(1, j)*** |
|---|---|---|---|---|
| 1 | 0.05 | 0.05 | 0.01 | 0.00 |
| 2 | 0.05 | 0.02 | 0.05 | 0.03 |
| 3 | 0.05 | 0.09 | 0.57 | 0.04 |

From the Table 1 above, we can see that the data sample, $x_1$ is having the minimum value of Euclidean Distance for cluster 1 ( $j$=1 ) which is $d(1,1)$ = 0.00, and that minimum value is $d(1, 1) \leq Ru_1$ ($d(i, j)) \leq Ru_m$, where m represents the number of the cluster,

$$d(i,m) = \min_j d(i, j) = \min_j(\| \chi_i - Cc_j \|), j = 1,2,3,...n \qquad (3)$$

then it is regarded that the current sample $x_1$ belongs to the cluster $C_1$, which is the first cluster.

In this case, neither a new cluster nor any existing cluster will be updated. The sample point $x_1$ will be added into the cluster $C_1$ as a new member in that cluster.

In the case when the minimum value of Euclidean Distance is higher than cluster radius size where $d(i, m) \geq Ru_m$. Thus, for further process, the Extended Distance Values, $s(i, j)$ for all the existing cluster and the sample point will be calculated using equation (4).

$$s(i, j) = d(i, j) + Ru_j, j = 1,2,3...n \qquad (4)$$

The same process applies to the Euclidean Distance calculation, the minimum value of the Extended Distance Values, equation (5) will be taken as the decision for that sample point which cluster it would fit in.

$$s(i, j) = d(i, j) + Ru_j = \min_j s(i, j), j = 1,2,3...n \qquad (5)$$

There are two possible situations that may occur:

- When the minimum value of Extended Distance Values (5), $s (i, k) \geq 2Dthr$, $k$ represents the number of cluster, then that certain sample point $x_i$ does not belong to any existing clusters. A new cluster will be created in the same way as described above.
- When the minimum value of Extended Distance Values (5), $s (i, k) \leq 2Dthr$, then that data point $x_i$ belongs to $\mathbf{C_a}$. The cluster with radius, $Ru_a$ will be increasing its value and the cluster $C_a$ will be updated by moving its cluster centre, $Cc_a$. The new updated radius $Ru_a^{new}$ is set to be equal to

  $$s(i, a) / 2 \qquad (6)$$

  and the new cluster centre, $Cc_a^{new}$ is located on the line connecting input data point $x_i$ and the old cluster centre $Cc_a$. The distance from the new centre $Cc_a^{new}$ to the sample point $x_i = Ru_a^{new}$ (refer to $x_1$ in Fig. 2).

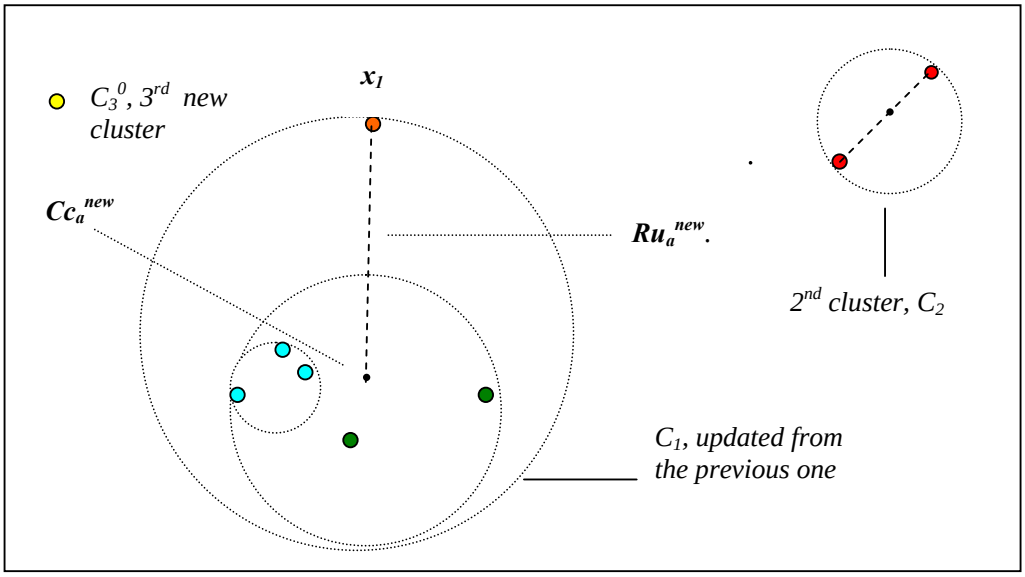


Fig. 2: A Brief Process of Updating Cluster Size [12]

### 3.4 Cluster Stability

Stability is one of the new features that we have proposed in our Evoc algorithm. This feature does not exist in the ECM algorithm. To identify a stable computer from the stable clusters, we need to go through two stages of data filtering process. However, before the clustering process starts, two important values need to be determined first which are stability value and stability hour.

- **Stability Value**: The value (in percentage) that measures the change in cluster radius. For instance, if the stability value is defined as 5%, any cluster radius that grows or shrinks less than 5% from the original size will be considered as stable.

- **Stability Hour**: The value that defines the required amount of time in hours for a cluster member to stay in the same cluster in order for it to be considered stable. If the stability hour is set to 3 hours, any cluster member that stays in the same cluster for more than this amount of time will be considered as stable.

A cluster member can only be considered as stable if both of the factors mentioned above are fulfilled. If one of the factors is not satisfied, the cluster member will be deemed as unstable. Clusters behaviors and their members stability is important system is meant for the monitoring and resources allocations.

Once the clustering process completes in the first phase, there will be many clusters created according to different computer's behaviors in every hour. However, up to this level, we cannot make any conclusion on the stability of the data based on this information. Thus, we propose the stability feature to identify the stable cluster members.

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 result from the clustering process will only show groups of similar behavior of computer nodes in an interval of time but not the stable computer nodes. Thus, the stability feature is being added into the original ECM algorithm to identify stable computer nodes that belongs to stable clusters.

A cluster's stability can be calculated by equation (7):

$$\text{Stability} = \frac{|R_{\_t} - R_t|}{R_{\_t}} \times 100\% \quad (7)$$

where $R_{\_t}$ stands for radius in the previous hour, *h-1* and $R_t$ stands for the current hour's cluster radius size. The smaller the value of the stability, the more stable the 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 <20% at time *t* will be classified as stable cluster.

The cluster members or computer nodes can only be considered as stable if it stays in the same cluster for a certain period of time or continuously for examples for 4 hours and the associated cluster(s) during this period of time is stable (which is <5% as mentioned above). Otherwise, the cluster member is deemed unstable. Both of these values are configurable according to the user's need.

### 3.5 Assumptions

Several assumptions have been made in designing the system:

- Any deleted cluster will not be created again. The cluster number will keep on increasing, *c1, c2, c3… cn.* As mentioned earlier, the same cluster will not be created once again.
- A cluster is considered to be stable depending on stability value which is pre-defined by the user, for instance 20%.
- A cluster member is considered to be stable if it stays in the same stable cluster continuously for or at least two hours. The stability hour is determined by the users.

The first assumption ensures that once a cluster is deleted, no similar cluster will be created. The proposed Evoc algorithm did not take this issue into consideration because we cannot determine the exact location of the cluster when it is created again. Thus, the first assumption is needed to overcome the weakness of the algorithm. We believe that further research is required to identify the most suitable values for the stability value and stability hour.

### 3.6 Threshold Value

A threshold is a limitation value that is used to measure whether certain action can be carried out or not. Threshold value normally is defined according to the need of evaluating different types of events, situations and requirements.

The smaller the value of the threshold, the more the cluster groups will be created. However, since the maximum value for a computer's CPU usage is 100%, the threshold value cannot exceed 0.5. A new cluster is only created if the Extended Distance Value is more than **2*Dthr***, where ***Dthr*** is the threshold value which is pre-defined before the clustering process has started. We use several sets of threshold value for the experimental purpose in order to determine which threshold value is better. In our research, we did not analyze deeper to identify the best threshold value for this Evoc algorithm.

## 4.0 EXPERIMENTAL RESULT AND DISCUSSION

We carried out experiments on the improved ECM algorithm, which is Evoc algorithm. The experimental are divided into three categories which are dynamicity, accuracy and stability. The purpose of the experiments is to evaluate the Evoc algorithm. The following are the summary of the experiments:

- **Experiment 1: 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 (Fig. 3). In the experiment with threshold value of 0.05, we found that when the $6^{th}$ hour is reached, cluster 1, C1 from ECM algorithm 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 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 $5^{th}$ 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.

  We can deduce from Fig. 3 that the ECM stops the clustering process after the $5^{th}$ 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.

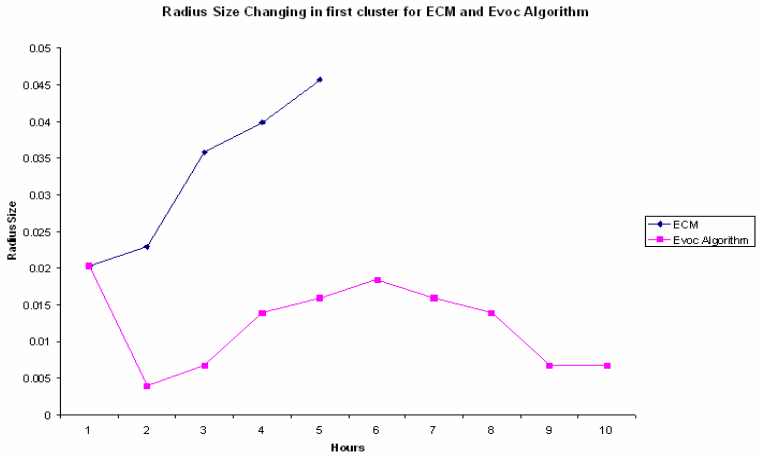


Fig. 3: The changes in radius size of the first cluster, C1 for ECM and Evoc algorithm with threshold value = 0.05.

- **Experiment 2: 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. K-means technique is sensitive to initial condition where different initial conditions may produce different results of clusters and the algorithm may be trapped in the local optimum. Of course, 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 (Fig. 4).

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.

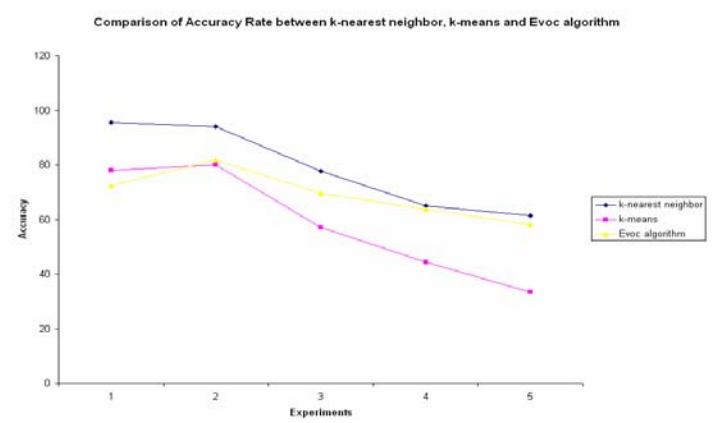


Fig. 4: The Comparison of Accuracy Rate between k-Nearest Neighbor, k-Means and Evoc Algorithm for Symmetric Data Set

For the second scenario, results from the first and second experiment using k-means technique 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 starts to display highest accuracy among three techniques from the third until fifth experiments. K-means's accuracy plunges drastically (Fig. 5). 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 and Evoc display the accuracy rate which is quite close to each other which can be seen in Fig. 4.

The overall result shows that Evoc algorithm performs better than the k-nearest neighbor and k-means clustering techniques (Fig. 4). Overall, it can be concluded that Evoc algorithm is more suitable to be used in clustering random data set.

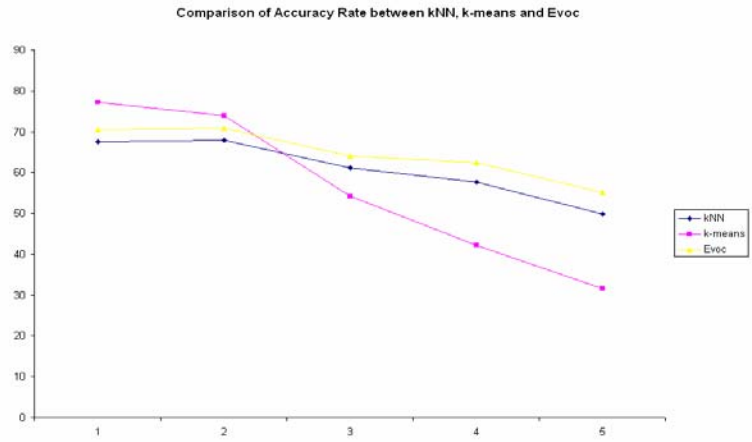


Fig. 5: Comparison of Accuracy Rate between k-Nearest Neighbor, k-Means and Evoc Algorithm for Random Data Set

- **Experiment 3: Evaluation of 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. 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.

## 5.0 CONCLUSION AND FUTURE WORK

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 is able to process data in a more dynamic way and give the result of stable cluster member which cannot be performed using 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 results for random data sets (accuracy: ~64.60%) compared with symmetric data sets (accuracy: ~60.83%) and it is 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 stability analysis that we have done so far is able to identify the stable cluster member for a certain period of time only. We suggest extending this feature so that it is able to give the result for a group of stable cluster members' stability patterns or trends which is actually a report that tells the activeness and inactiveness of that group of cluster members on daily basis. For instance, computer A, B, C and D are very active from 9.00am until 5.00pm but very inactive from 1.00am until 8.00am everyday. We believe that this information is very useful for resource allocation purpose. It allows a person to know the potential group of resources that is likely to be is available for job submission during which time.

## REFERENCES


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

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

[3] A.K. Jain, M.N. Murty and P.J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No. 3, Sept 1999. pp. 264-323.

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

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

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

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

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

[9] Raymond T. Ng and Jiawei Han, 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. pp. 1003 – 1016.

[10] Jeffrey Heer, Ed H. Chi., "Mining the Structure of User Activity using Cluster Stability", *In Proceedings of the Web Analytics Workshop, SIAM Conference on Data Mining*, 2002.

[11] 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)*, Kaua'I, HI, January 2002.

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

[13] Nikola Kasabov, Qun Song, *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. pp. 144–154

[14] Ooi Boon Yaik, Chan Huah Yong, Fazilah Haron, "CPU Usage Pattern Discovery Using Suffix Tree. 2nd International Conference on Distributed Frameworks for Multimedia Applications", *Distributed Frameworks for Multimedia Application 2006*, IEEE. Penang, Malaysia. May 15-17, 2006, pp. 14-21.

[15] Rich Wolski, Spring, N. and Hayes, J., "Predicting the CPU Availability of Time-shared Unix Systems on the Computational Grid", *Proceedings of 8tj High-performance Distributed Computing Systems Conference*, August, 1999.

## BIOGRAPHY

Chan Huah Yong is a senior lecturer at the School of Computer Sciences, and head of grid computing lab, in Universiti Sains Malaysia (USM). His research interests cover load balancing, resource management, optimization, prediction, grid computing, peer-to-peer computing, distributed agent technology, data mining, distributed and parallel database. He is actively involved in grid collaboration activities at the national level i.e. national grid initiative and at the international level i.e. PRAGMA.

Kee Sim Ee is a lecturer at the School of Art and Science, TAR College. She did her master thesis at USM on the clustering of CPU usage data in grid environment using EVOC algorithm. She has worked as a research officer on grid computing project at the grid computing lab, USM. She had published a few papers on her thesis work.

Fazilah Haron is a senior lecturer at the School of Computer Sciences, Universiti Sains Malaysia (USM). She is an active member of the parallel and distributed computing research group. She initiated the grid computing research in the department and has been involved in grid-based activities at the national and international levels. She is one of the key researchers in the pioneer grid project, the e-Science Grid which is funded by the Ministry of Science, Technology and Innovation, Malaysia. Her research interests include grid computing, parallel and distributed processing, and peer-to-peer computing.