Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/070630

K-Medoids based Clustering of PlanetLab's Slice-Centric Data

*Aun Haider**

Department of Electrical Engineering, University of Management and Technology, Lahore, Pakistan

Received: 21 Mar. 2013, Revised: 21 Jul. 2013, Accepted: 24 Jul. 2013
Published online: 1 Nov. 2013

Abstract: This paper investigates the application of widely used K-Medoids based clustering algorithm on data collected through CoMon facility for the PlanetLab testbed. The averaged values of various metrics in passively collected slice-centric data has been considered for clustering purposes. Various groups of slices, depicting similar resource usage patterns have been identified in original data set. These clusters have been represented in reduced dimensional space formed by first two principal components of original data set. In order to capture variations in pattern of resource usage by various slices at a PlanetLab node, clustering of standard deviations of various metrics have also been carried out. Further, combining averaged and standard deviation, clustering has also been performed on index of dispersion computed from the original data set. It has been found that K-medoid based clustering can effectively split the original data space into various sub-spaces of different resource usage behaviour of slices. Thus, it can lead to better resource management and control in publicly available testbeds.

Keywords: PlanetLab, Slice-Centric CoMon data, Resource management, Principal Component Analysis (PCA), K-medoids based clustering & Cluster validation.

1 Introduction

PlanetLab, an overlay on the Internet, provides experimental testbed services to world wide community of researchers. Since its launching in mid 2002, it has been extensively used to evaluate a diverse set of network services, including content distribution, anycast, Distributed Hash Tables (DHTs), robust DNS, Peer-to-peer, measurement and analysis, anomaly and fault diagnosis, [1, 2].

Recently, some issues relating to conflicts arising in PlanetLab usage have been pointed in [3]. These include high frequency measurement probes, illegal content distribution and excessive usage of resources such as bandwidth. In order to address these issues and thus to avoid the abusage of PlanetLab resources, log files have been extensively used by administration. All of the conflicts and security issues are currently being handled by PlanetLab Central (PLC), that acts as a trusted intermediary between node hosting sites and researchers. Basically, the operation of PLC is dependent upon three key mechanisms: isolation provided by VServer, an auditing mechanism provided by PlanetFlow and secure remote boot mechanism that allows to inspect a node even if its kernel has been compromised, [3]. However, PLC does not attempt to prevent the occurence of problems but is responsive when complaint is reported. It records only per-flow byte and packet counts for auditing purposes, indicating that a given packet belongs to which slice.

CoMon is a monitoring system that has been designed for providing necessary information about operation of all of the PlanetLab nodes. It has been inspired by the CoDeeN Content Distribution Network. Since its launch in August 2004, with its freely available data, it has provided sufficient monitoring, community aided problem identification, login trouble shooting and as an aid to node selection for conducting experiments, [4]. It runs two daemons, i.e. node and slice-centric, on each node of the PlanetLab. The slice centric daemon is simple and reports an aggregated consumption of resources within each slice, [5]. Whereas the node-centric daemon reports a larger number metrics which can be classified as: (i) OS-provided, such as CPU utilization and Memory consumption etc (ii) passively measured, such as number of slices in memory and resources hogs and (iii) actively measured quantities such as amount of memory pressure and TCP/UDP failure rates for local DNS servers. Thus,

* Corresponding author e-mail: aun@acm.org

the data collected through CoMoN has wealth of information which can be used to understand the state of affairs at nodes or slices across the PlanetLab.

It is important to notice that with an increase in number of PlanetLab nodes as well as the users, the amount of raw monitoring data generated by CoMon has also been significantly increased. Hence, making the manual use of monitoring data a quite challenging task and thus requiring to use efficient data processing techniques for intelligent decision making. For instance, it can lead us to better resource usage control as well as detecting the irresponsible users. A first step in this direction can be an efficient clustering of the data collected. Generally, clustering refers to a process of organizing data into homogeneous groups or clusters, where an attempt is made to maximize similarity between objects included in the same group and also maximize dissimilarity between objects included in different groups. It can also be regarded as an unsupervised classification of patterns in the data. Despite of its hardness, due to combinatorial nature, it has been widely employed in various disciplines [6].

Clustering can be classified as crisp or fuzzy in nature. In crisp clustering each data object is included in exactly one cluster, whereas in fuzzy clustering each data object can have varying degree of membership to several or all if the clusters. Clustering can also be carried out on the basis of mixture models where data is assumed to be generated by several parameterized distributions, [7]. Further, one can find a huge body of literature related to clustering; as according to [8] there exists more than three digit number of clustering algorithms. A basic purpose of most of the clustering techniques is to minimze the inter-cluster distance and maximize the intra-cluster distance. Whereas, one can find several definitions of distance in [9].

This paper presents K-Medoids based crisp clustering of resource usage monitoring data captured by CoMon facility of PlanetLab. It does not attempt to present a comparison between a large number of clustering algorithms available in litrature. For representation of clusters, the dimensions of data has been reduced by Principal Component Analysis (PCA). Euclidean distance has been employed as a metric to measure distance between various data points, while forming the clusters. Major contributions of this paper include: development of an analytical model for resource usage by various slices, clustering and identification of resource usage pattern among various slices at a PlanetLab node [1].

The rest of this paper is organized as follows. An overview of Principal Component Analysis technique for data analysis and K-Medoids based clustering has been presented in Section 1. Services provided by PlanetLab to manage and monitor the resource usage by various slices have been discussed in Section 2. The analysis of slice-centric CoMon's data has been presented in the Section 3. Finally, some conclusions have been drawn in Section 5.

[1] This manuscript is an extended version of paper published in 15th IEEE International Multitopic Conference 2012.

1.1 Principal Component Analysis

The basic idea in Principal Component Analysis, [10], is to transform the higher dimensional space data, consisting of many interrelated variables, into a lower dimensional space data; while attempting to retain maximum possible variation in the original data set. It performs coordiante transformation by mapping the input data onto set of axes spanned by eigenvectors.

Mathematically, computation of princiapl components can be summarized as follows. Assume that $\mathbf{X} \in \mathbf{R}^{n\times m}$ is the original data matrix containing m rows and n number of columns, that is normalized to zero mean and unit variance matrix $\mathbf{Y} \in \mathbf{R}^{n\times m}$. Next step is to compute the covariance matrix of $\mathbf{Y}$, i.e. $\mathbf{R} = 1/(n-1)\mathbf{Y^T Y}$. Then find the Singular Value Decomposition (SVD) of $\mathbf{R}$, which can be expressed as $\mathbf{R} = V\Lambda V^T$. Whereas, Λ is a diagonal matrix containing eigen values of $\mathbf{R}$ in decsending order and columns of V are eigen vectors of $\mathbf{R}$. A transformation matrix $P \in \mathbf{R}^{m\times p}$ can be constructed by selecting p number of eigen values. This selection of p defines the number of principal components that will employed in the analysis of data in reduced dimensions space. One popular method to select the number of principal components is to ensure that cummulative variance should be $\geq 90\%$. Multiplication of $\mathbf{P}$ with $\mathbf{Y}$ will give us a reduced dimension data matrix $\mathbf{Z}$; where columns of $\mathbf{P}$ are called loadings and elements of $\mathbf{Z}$ are termed as scores. These scores can be transformed back into the original data space as $\mathbf{Y} = \mathbf{ZP}^T$. The difference between original data and the reconstructed data will form the residual error.

The first principal component indicates the maximum amount of variation or energy present in original data in the direction of first eigenvector. The remaining variations in data are orderly captured by subsequent principal components. Hence, principal components are in descending fashion according to capturing of amount of variations in the original data. In this paper we have employed first two principal components of data to represent clusters.

1.2 k-medoid based Clustering

In order to find k clusters in n number of data points, using medoids, Partitioning Around Medoids (PAM) algorithm has been developed in [11]. It is one of the earliest implementations of medoids based partition algorithms. Whereas, a medoid can be defined as a representative object of a data set, or a cluster within a data set, which is most centrally located within that group

of data. Although, similar to concept of means or centroids, a medoid is always a member of data set.

The clustering algorithms based on medoids possess some very useful properties, [12]: such as robustness to outliers, independence to order in which data is examined, invariance to translations and orthogonal transformations of data points and handling of large data sets. The operation of PAM can be summarized in the following steps [12]:

–Select k representative objects, i.e. medoids, arbitrarily.
–Compute the total cost of replacement for all pairs of data objects O_m and O_p, $\mathbf{C}_{mp}$; where O_m is currently selected as medoid and O_p is not selected.
–Select the pair O_m, O_p which corresponds to $\min_{O_m,O_p} \mathbf{C}_{mp}$. If minimum $\mathbf{C}_{mp}$ is negative, replace O_m with O_p and go to step 2.
–Otherwise for each non-selected object, find the most similar medoid.

However, PAM does not work satisfactorily for large data sets as its complexity for single iteration is $\mathbf{O}(k(n-k)^2)$. For dealing with larger data sets, Clustering Large Applications (CLARA) algorithm has been proposed in [11]. It draws multiple samples of data and apply PAM on each sample to generate best clustering as its output. Its complexity, for a sample size of s, can given as $\mathbf{O}(ks^2 + k(n-k))$. A weakness of CLARA is that there is no systematic way to select the sample size s. However, a simple heuristic used is to select 5 samples of size $(40+k)$, [11]. The performance of CLARA has been improved by Clustering Large Applications upon Randomized Search (CLARANS) algorithm, [12]. It arbitrarily picks one of the k medoids and attempts to replace it by another data object that has been randomly chosen among $(n-k)$ data objects. In this paper, due to smaller size of data, we have only employed the basic K-medoids based clustering as performed by PAM implementation in R statistical package: http://www.r-project.org/.

2 Resource Management in PlanetLab

For resource management purposes, PlanetLab provides three basic services to users: Slice Creation Service, Brokerage Service, Monitoring Service, and Auditing service, [2]. The slice creation service is provided by plconf and require no special privileges, wherein the node owner creates a resource pool and assign it to plconf at the time of bootup. The brokerage service is mostly provided by Sirius, [13]. It performs function of admission control to resource pool available at PlanetLab node. The plconf set aside a part of resources to be used by Sirius. Other market based resource brokerage services, such as Bellagio and Tycoon, [2], may also be available at some of the PlanetLab nodes. For monitoring of a node, CoStat has been used. It is a low-level instrumentation that has access to /proc files. Services such as CoMon, collects and process the data collected, [4]. A simplified relationship among various PlanetLab services has been depicted in Fig. 1. Among

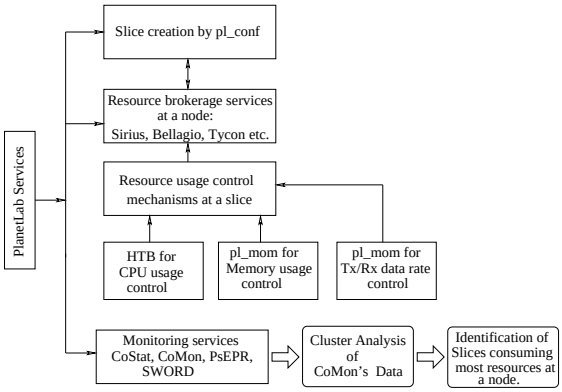


Fig. 1: An overview of services offered by PlanetLab.

these services, this paper concentrates only on data collected through the monitoring services provided by the CoMon facility. It is important to note that from CoMon data, one cannot guess the underlying brokerage services being provided by a PlanetLab node.

The CPU usage at a PlanetLab node is controlled by fair share scheduling and work conserving reservations provided by overlaying a token bucket filter on the top of the standard Linux Scheduler. For further details, see [2] and its references. On local scale, Planetlab monitors resources consumed by pl_mom watchdog daemon. It resets a slice consuming most of the physical memory when swap has almost filled. Similarly the sustained traffic rate, bandwidth, is also limited by pl_mom. It allows each slice to send a quota of bytes per day at a node's maximum cap rate, and imposes much smaller limit if slice exceeds its quota. However two weaknesses exists in this approach: (i) some sites for PlanetLab nodes pay on the basis of total amount of traffic generated per month, they need to control the sustained rate of traffic rather than the peak rate; pl_mom operates on per slice basis and cannot be controlled by sites (ii) PlanetLab nodes do not cap the incoming bandwidth, thus giving a possibility to saturate the bottle neck by downloading a large amount of data. Thus, overall resource allocation to various slices at a PlanetLab node is a complicated process involving resource brokerage, CPU scheduling and pl_mom daemon.

The combined effect of all these mechanisms, as indicated in Fig. 1, is reflected in the slice-centric data provided by CoMon. Wherein, it is not trivial to filter out the effect of each mechanism for resource control and allocation. Thus, it is reasonable to treat the resource management mechanisms as a block box and resort to identify the slices consuming most of the allocated

resources. One method to achieve this goal is to employ clustering algorithms on the collected data to form groups of slices with similar resource usage characteristics. It will help to identify the slices consuming too high resources. After forming clusters, it would be easier to take corrective actions on various slices to maintain fairness in resource usage.

3 CoMon's Data Analysis

Currently, PlanetLab consists of more than 1100 active nodes which are being constantly monitored after every 5 minutes. Hence, data is collected from each node for 288 times every day. This data has been made avilable publicly. Generally each PlanetLab node has a different number of slices. It stems from the fact that each user selects its own set of nodes for creating slices to conduct experiments. Although, some services, e.g. SWORD [14], are available to users for aid in node selection process. However the role of these services is advisory only.

In this setup, there can be two possible ways to carry data analysis: (i) consider resource usage of various slices at a particular node in one day or for longer periods of time (ii) consider set of all slices in Planetlab and determine that how much resource are being consumed by each of them at each node. The first approach will tell us which slice is consuming most of the resources at a particular node, whereas the second approach will indicate an overall resource usage pattern of a particular slice over the entire PlanetLab. But it will require a centralized entity for global advertisement of this resource usage information by a particular slice. Then each node can have its own policy to deal with slices consuming too much resources.

3.1 Arrangement of Data

For our analytical studies, we have employed the following eight parameters available in slice centric CoMon's data: transmit bandwidth (Kb/s) over last 1 and 15 minutes, receive bandwidth (Kb/s) over last 1 and 15 minutes, physical and vsirtual memory used (MBytes), % CPU and % Memory usage by each slice. Raw data from CoMon is parsed to filter out these metrics. This fileterd data can be arranged in a matrix that can have the following two forms.

– Columns of data matrix represents various slices running at a node. The rows will represent samples of a single metric taken at an interval of 5 minutes. For a node running m number of slices, the dimension of data matrix will be $288 \times m$. In this arrangement of data, each slice will be acting as a variable whose samples are represented by rows. It has been represented in Fig. 2. Each column of data matrix will be representing a time series of a particular metric in the data for that slice. Thus, we will be requiring a different matrix for each of the each eight metrics under analysis.
– One can take time average of several metrics and arrange them in columns for various slices of the node as rows of matrix. However, temporal characteristics of each metric will be lost due to averaging process. One solution to this loss of information would be to compute the standard deviation of the averaged data and use it in clustering as well.

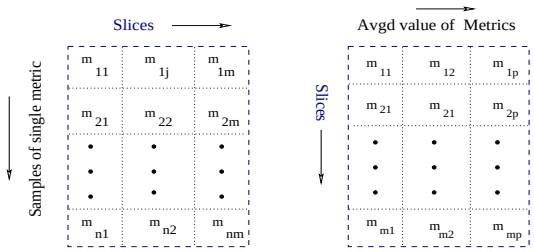


Fig. 2: Two arrangements of fileterd data in matrix form.

In this paper we have adopted the second approach and collected one week's CoMon data. Then we computed arithmetic mean over each day, for each of the eight metrics under study. We have focussed on resource usage pattern of 60 slices, giving a data matrix of 60x8, in one of the PlanetLab's node. The clustering has been performed by using K-medoids algorithm.

3.2 Cluster Validation

The next step after clustering is to determine the quality or vaildatation the clusters. For this purpose, a large number of cluster validation techniques exist in literature, [6]. In this paper we have employed the widely used Silhouette Index for cluster validation. It can be defined as follows: For a given cluster Z_j with $j \in (1,2,...,c)$ the Silhouette method assigns the $i-$th sample of of Z_j a quality measure known as Silhouette width which has been defined as follows:

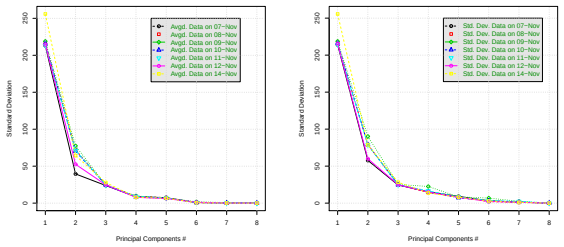
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{1}$$

where $a(i)$ is the average distance between $i-$th sample and all samples in Z_j and $b(i)$ is the minimum average distance between the $i-$th sample and all samples in Z_k for $k \in (1,2,...,c)$ with $k \neq j$. It indicates that value of silhouette width varies between -1 and 1. The value of 1 indicates that sample has been included in appropriate cluster, whereas -1 indicates the misclassification, [15].

3.3 Clustering Results

3.3.1 Principal Components of Data

In order to observe variability in data matrix, we have computed Principal Components for both averaged and standard deviation of the fileterd data. The standard deviation of various components has been plotted in Fig. 3. We have performed K-Medoids based clustering on



(a) (b)

Fig. 3: Standard deviation of various principal components computed for averaged data (a) and its standard deviation (b).

complete data matrix. However, these clusters cannot be visualized due to higher dimensions of data. Hence we will represent clusters using first two principal components of data.

3.3.2 Clustering based on Averaged Data

The clustering results for three days averaged data are presented in Fig. 6. Each column represents results for one day of CoMon's data. The clusters have been represented in the plane formed by first and second principal components of the data. The five clusters formed from the first day of observation are shown in Fig. 6. (a). The Sillhouette index was computed and has been plotted in the Fig. 6. (d). Also, it indicates that cluster membership for each slice under study. From this figure, it can be observed that cluster # 2 contains a largest number of slices, whereas cluster 5 contains only one slice (# 33). Also, cluster 2 has largest value of average Silhouette index.

By selecting two or three metrics of ineterest, one can represent the results of K-Medoids clustering in a plane or in 3-dimensions. For instance, we have considered % CPU usage, % Memory usage and data transmission rates as three most crucial resources in PlanetLab and are represented along each axis. Thus, we have represented the clusters in Figs. 6. (g) and 6. (j), for transmitted and received bandwidth respectively. These two representations are almost similar except for one slice (# 39). Next we draw similar plots for the data of next two days. It can be observed that the membership of clusters does not change drastically with each day. The average sillhouette index slightly improved from 0.41 to 0.45 for data collected on second day and onward. Again the Cluster # 2 has largest number of slices with an average Sillhouette index close to 0.74. The cluster # 5 has only only slice (# 33).

Correlating the cluster diagrams with 3-D representations, it can be observed that slice # 33 is consuming most of the bandwidth resources. Whereas the slice # 54 and 25 are consuming the most of the CPU and memory resources. Its important to note that PlanetLab node act as a substrate of resources and allocate a fixed amount of CPU and Memory to each slice. Hence % CPU and % Memory usage are reflecting the consumption in the allotcated quota for each slice. However, the transmit and receive bandiwdth are can partially controlled by PlanetLab through pl_mom, [2]. It is important to note that desipte large resource consumption by individual slices (within their quotas), the PlanetLab Node might be underloaded.

3.4 Clustering based on Standard Deviation of Data

The averaging of time series data of CoMon's measured metrics will conceal its time variations. Thus clustering of averaged data will be representative of the groups of slices having same mean resources consumed in a single day. However, it might be important to identify slices having most variations in the resource usage at a PlanetLab node. Therefore, in order to cluster data on the basis of its variations, we have computed standard deviations of all of the eight metrics and new clustering was performed.

The results for clustering of the standard deviation of CoMon's data for three days have been shown in Fig. 7. The clusters formed on plane formed by first and second principal components have been shown in Figs. 7 (a), 7 (b) and 7 (c). Whereas the corresponding Sillhouette plots are given in Figs. 7 (d), (e) and (f) respectively. From these it can be observed that, as in the case of the averaged data, almost half of the slices lie in the cluster # 2. These slices have similar variations in resource requirements.

The 3-D represenations of clusters formed on the basis of standard deviation of measured metrics are shown in Figs. 7 (g), 7 (h) and 7 (i) for received bandwidth and in Figs. 7 (j), 7 (k) and 7 (l) for transmit bandwidth, repectively. Looking at these diagrams it can be found that resource requirements of slices included in cluster # 2 do not change much. Thus, indicating that many slices do not change their resource requirements dramatically during their life time. The slices having highest variations can also be easily identified. It is interesting to observe that the slices whose average values

are highest also have highest highest value of standard deviation for the same metric.

3.5 Variations in Cluster membership

In order to quantify changes in cluster membership of various slices for each day of data measurement, Jaccard index has been employed, [16]. It measures the similarity between two sample sets (clusters) and is defined as a ratio between sizes of intersection and union between them; i.e. for a data set D with two subsets A and B, Jaccard index is defined as $J = \frac{A \cap B}{A \cup B}$. The Jaccard index for averaged and standard deviation of data has been plotted in Figs. 4 and 5, respectively. It can be seen that

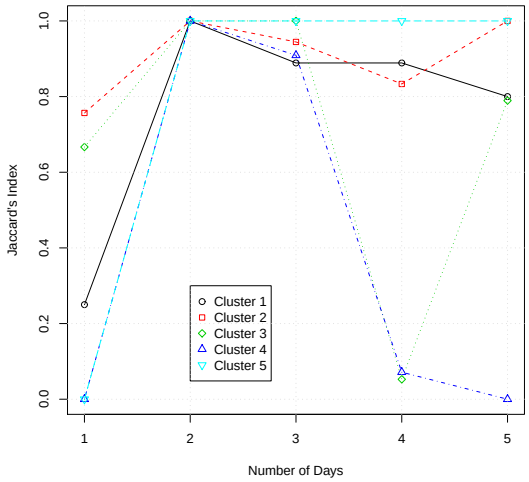


Fig. 4: Jaccard index plot for k-medoids clustering performed on the averaged data.

except for day 1, the clusters 1, 2 and 5 for averaged data, Fig. 4, have high values of Jaccard index. It indicates that not many slices in these clusters change their membership. However, for the cluster 3 and 4 the index drops to low values after passage of day 3. The same pattern can be see from Sillhouette index plots. Similar trend in the change in cluster membership has been maintained in Fig. 5.

4 Some Related Work

In [17] an extensive study to understand and characterize the resource usage by PlanetLab, has been presented. It has analyzed the six years (2005 to 2010) of data collected from CoMon facility of PlanetLab. It has been

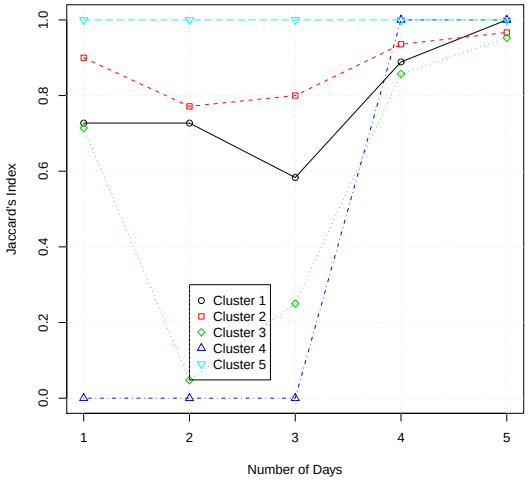


Fig. 5: Jaccard index plot for k-medoids clustering performed on standard deviation of the averaged data.

reported that only 3% of all slices can account for more than 80% of all CPU usage in PlanetLab. Similarly, only around 4% of all slices consume more than 80% of all memory resources. It has also been found that unlike the compute clusters, where users consume resources greedily, the PlanetLab users are not agressive in resource consumption. Thus, resource consumption depicts the bimodal distribution along various axes. Two major resource allocation systems, i.e. pair-wise bartering and central banking, have also been investigated. Also, it has been reported that both resource allocation systems would handle only a small proportion of total usage of resources in PlanetLab.

In Sharp [18] various users can trade their resources using tickets which can be issued, delegated and redeemed in a cryptographically secure manner. Several mechanisms for trading resources in an econmomical way have been proposed in literature, such as: Millenium [19], Mirage [20] and Tycoon [21]. In general, these systems attempts to maximize the benefit delivered to users by providing a method to express the value of their resources.

A resource discovery system, SWORD [22], has also been deployed in PlanetLab [1]. In this system the users specifies the desired resources in XML and submit queries to a service which attempts to first locate and then allocate the resources to users.

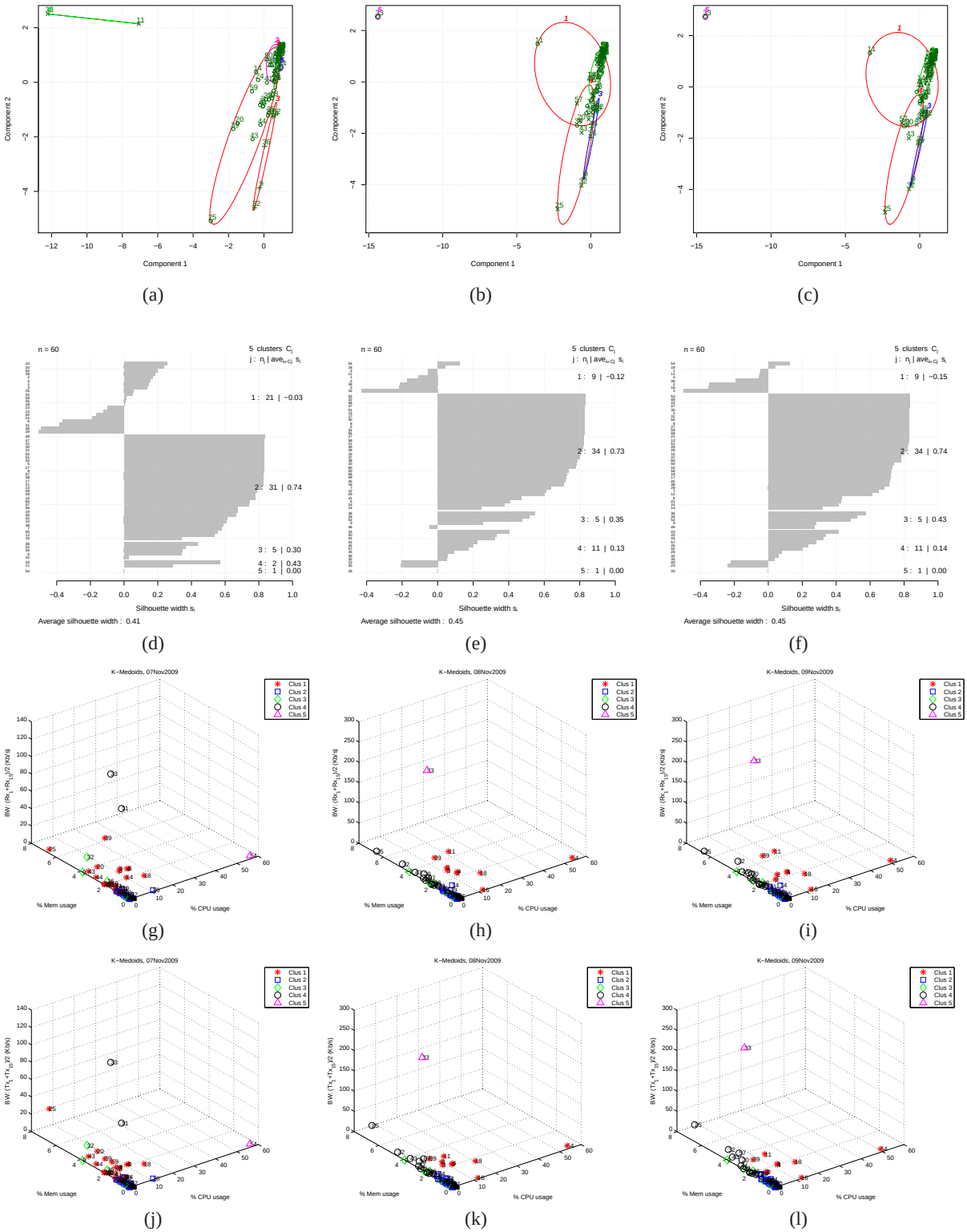


Fig. 6: (a), (b) and (c) show K-Medoids clustering for data collected (60x8); corresponding silhouette are also shown; number of slices considered are 60, for which nummber of clusters selected are 5.

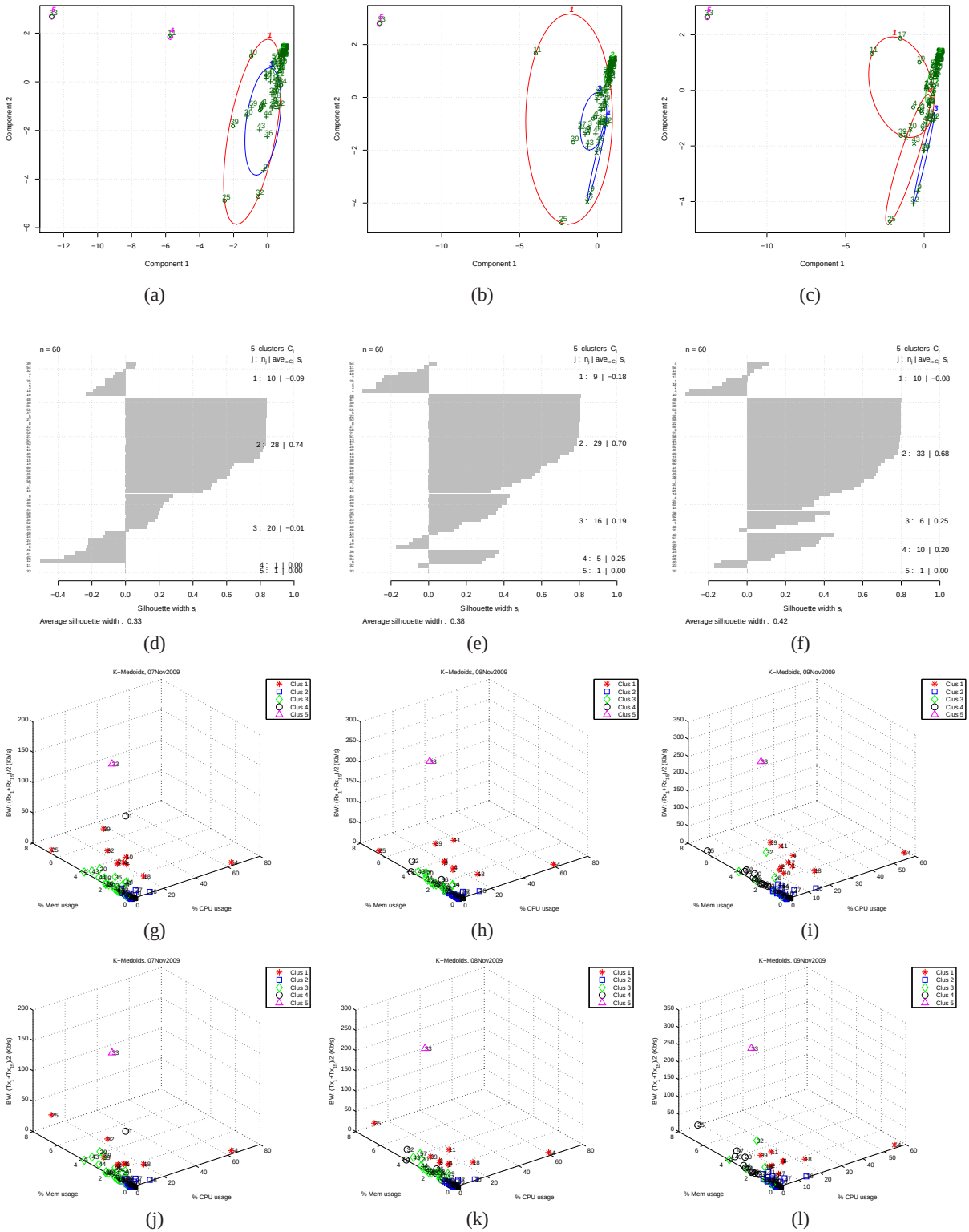
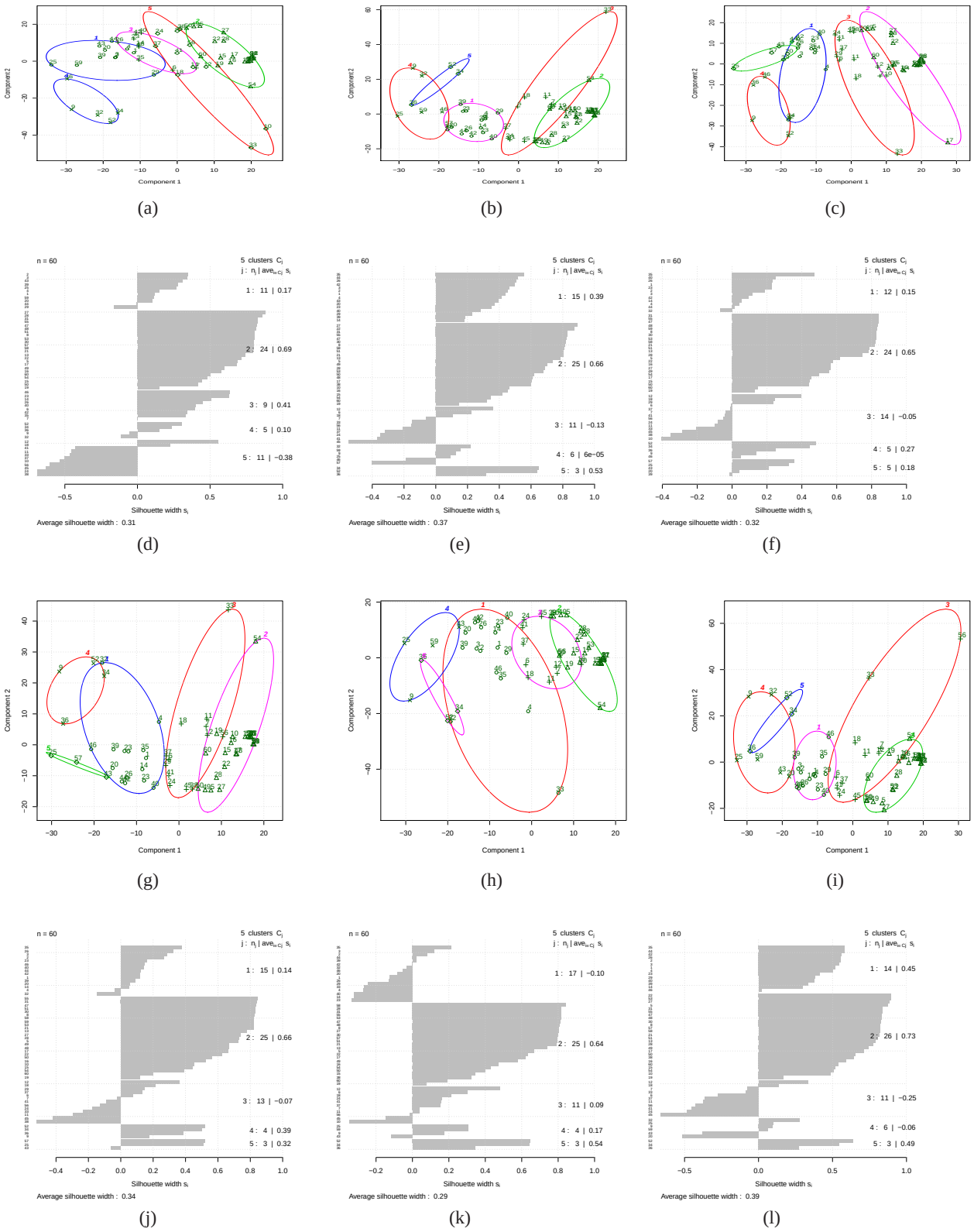


Fig. 7: (a), (b), (c), (g), (h) and (i) show K-Medoids clustering for standard deviation of data collected (60x8); corresponding silhouette shown in (d), (e), (f), (j), (k) and (l); number of slices considered are 60, for which nummber of clusters selected are 5.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 8: (a), (b), (c), (g), (h) and (i) show K-Medoids clustering for Index of Dispersion (variance/mean) corresponding silhouette shown in (d), (e), (f), (j), (k) and (l); number of slices considered are 60, for which nummber of clusters selected are 5.

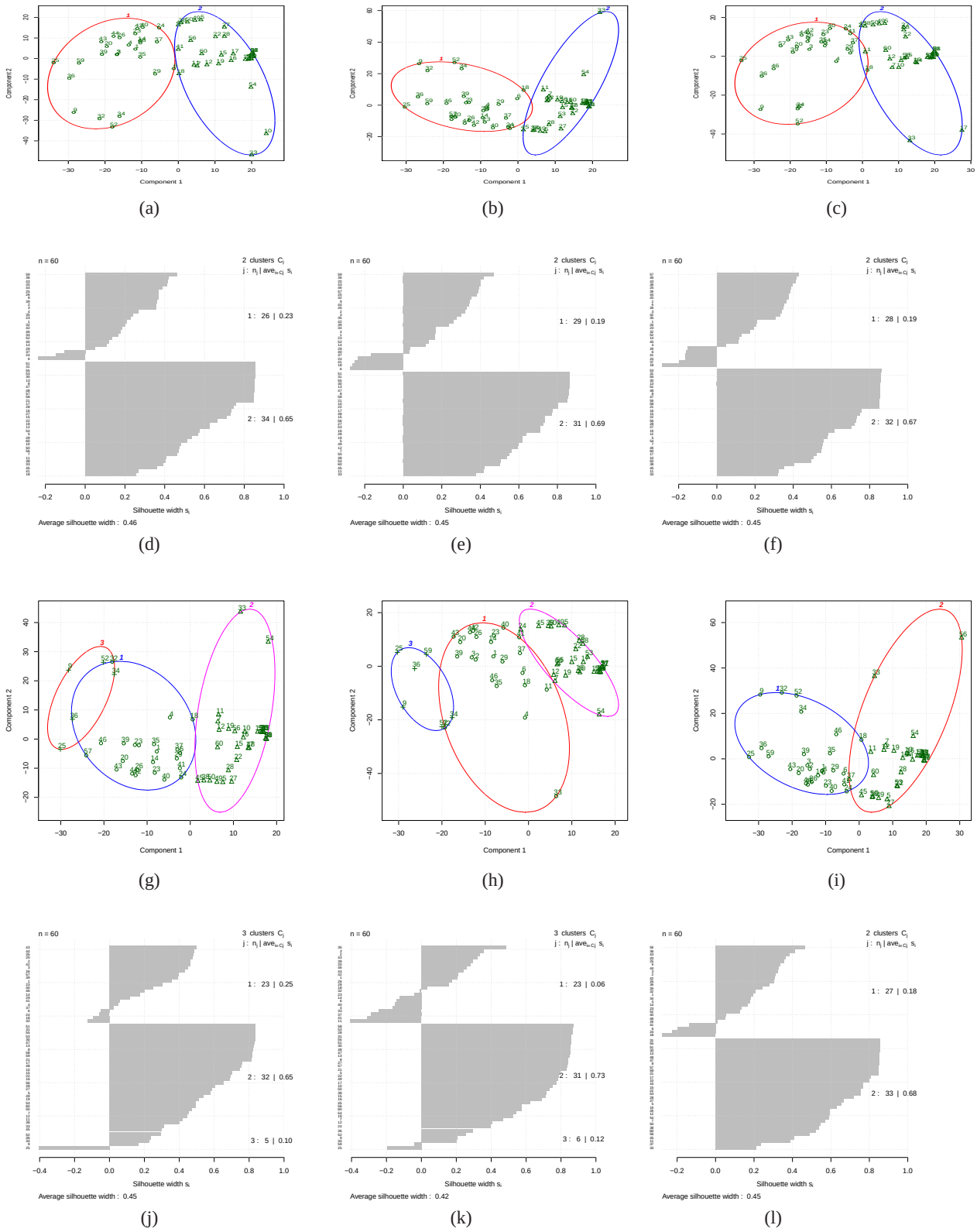


Fig. 9: (a), (b), (c), (g), (h) and (i) show K-Medoids clustering for Index of Dispersion (Variance/Mean) corresponding silhouette shown in (d), (e), (f), (j), (k) and (l); number of slices considered are 60, for which number of clusters are selected automatically by pamk.

5 Conclusions

K-medoid based clustering has been performed on slice centric CoMon data, generated by one of the PlanetLab node, to identify the slices consuming the most of the resources. After performing clustering on complete data matrix, they have been represented by the first two principal components of data. Sillhouette index has been used to validate clusters and indicate slices belonging to various clusters. Further, in order to capture time behaviour of slices, the standard deviation of the averaged data has been also clustered. In each case clusters have been represented by 3-D space formed by %CPU, %Memory and Received/Transmitted data rates. During our period of observation, it has been found that most of the slices do not show much changes in resource usage pattern and almost half of them belong to one big cluster. The average sillhouette index of this cluster is also quite high. The technique presented in this paper is fairly general has a relevance to virtual network providers and cloud computing. It can be used to understand the nature resource usage pattern of various users, which can lead to better resource provisioning and revenue computation. For future work, it is worth to investiagte the state transition of various slices and find underlying Markov Chain Models for predictability of resource usage.

Acknowledgment

This work was done at National Institute of Information and Communications Technology (NICT), Hakusan Bunkyo-ku, Tokyo, Japan http://www.nict.go.jp/en/index.html. The author would like to thank Akihiro Nakao http://www.iii.u-tokyo.ac.jp/professor.php?id=386 for support to this work.

References

[1] PlanetLab. [Online]. Available: http://www.planet-lab.org/

[2] Larry Peterson, Andy Bavier, Marc E. Fiuczynski, and Steve Muir, "Experiences Building PlanetLab," *Unenix OSDI*, pp. 351–366, 2006. [Online]. Available: http://www.usenix.org/events/osdi06/tech/peterson.html

[3] Larry Peterson, "Understanding and Resolving Conflicts on PlanetLab," November 2008. [Online]. Available: http://www.cs.princeton.edu/~llp/policy.pdf

[4] KyoungSoo Park and Vivek Pai, "CoMon: A Mostly-Scalable Monitoring System for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006. [Online]. Available: http://comon.cs.princeton.edu/index.html

[5] "Cotop." [Online]. Available: http://codeen.cs.princeton.edu/cotop/

[6] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[7] Juha Vesanto and Esa Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, 2000.

[8] Georg Peters, Martin Lampart and Richard Weber, "Evolutionary Rough k-Medoid Clustering," *LNCS Transactions on Rough Sets VIII*, pp. 289–306, 2008.

[9] E. Deza and M. M. Deza, *Dictionary of Distances*. Elsevier, 2006.

[10] *Principal Coponent Analysis*, 2nd ed. Springer-Verlag, 2002.

[11] L . Kaufman and P. J. Rousseeeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1999.

[12] Raymond T. Ng and Jiawei Han, "CLARANS: A Method for Clustering Objects for Spatial Data Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.

[13] D. Lowenthal, "Sirius: A calendar service for planetlab," http://www.cs.uga.edu/∼dkl/ and http://www.planet-lab.org/services.

[14] D. Oppenheimer, J. Albrecht, D. Patterson, and A. Vahdat, "Distributed resource discovery on planetlab with sword," *WORLDs*, 2004, http://www.usenix.org/event/worlds04/tech/oppenheimer.html.

[15] N. Bolshakova, and F. Azuaje, "Improving Expression Data Mining Through Cluster Validation," *Proc. of 4-th IEEE EMBS Special Topic Conf.*, pp. 19–22, 2003.

[16] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. 2006, Pearson Addison Wesley.

[17] W. Kim, K. Y. Li, A. Roopakalu, and V. S. Pai, "Understanding and characterizing planetlab resource usage for federated network testbeds," *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pp. 515–532, 2011.

[18] Y. Fy, J. Chase, B. Chun, S. Schwab, and A. Vahdat, "Sharp: an architecture for secure resource peering," *Proc. of the 19th ACM symposium on Operating systems principles*, pp. 133–148, 2003.

[19] B. N. Chun and D. E. Culler, "User-centric performance analysis of market-based cluster batch schedulers," *Proc. of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 30–, 2002.

[20] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat, "Mirage: a microeconomic resource allocation system for sensornet testbeds," *Proc. of the 2nd IEEE workshop on Embedded Networked Sensors*, pp. 19–28, 2005.

[21] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, "Tycoon: An implementation of a distributed, market-based resource allocation system," *Multiagent Grid Syst.*, vol. 1, no. 3, pp. 169–182, August 2005.

[22] J. Albrecht, D. Oppenheimer, A. Vahdat, and D. A. Patterson, "Design and implementation trade-offs for wide-area resource discovery," *ACM Transactions on Internet Technology*, vol. 8, no. 4, pp. 1–44, September 2008.

Aun Haider received his B.Sc. and M.Sc. Electrical Engineering degrees from University of Engineering and Technology, Lahore Pakistan. He received his PhD degree in Electrical and Electronic Engineering from University of Canterbury, Christchurch New Zealand. His research interests are in Wireless Networks (LTE and LTE-A), Network Virtualization and Future Generation Networks. He has published several research articles in reputed International Journals and Conferences in the area of Electrical Engineering. Dr. Aun Haider is also Senior Member IEEE (http://www.ieee.org/index.html).