2006

Web workload analysis and session characterization using clustering

Deepak Jha
West Virginia University

Web Workload Analysis and Session Characterization using Clustering

by

Deepak Jha

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Dr. Katerina Goseva Popstojanova, Chair
Dr. Jagannathan Vasudevan
Dr. Arun A. Ross

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2006

Keywords: Clustering,Principal Component Analysis, Log Analysis, User Sessions

Abstract

Web Workload Analysis and Session Characterization using Clustering

by

Deepak Jha

Web servers have a significant presence in today's Internet. Corporations want to achieve high availability, scalability, and consistent performance for respective Web systems, maintaining high customer service standards. Web Workload characterization and the analysis of Web log files are the basis on which Web server modeling for efficiency, scalability and availability can be planned. This thesis analyzes the Web access logs of six public Web sites: Department of Computer Science and Electrical Engineering at West Virginia University, West Virginia University, three NASA IVV servers, and Clarknet server. In addition, three private NASA IVV servers are also analyzed.

We characterize sessions using several attributes such as number of request per session, session length in time units, number of bytes transferred per session, and number of erroneous requests per session. We use clustering, as unsupervised learning methods, to classify Web server sessions. Unlike most other studies which were focused on building user profiles based on their navigational patterns, we use session attributes as basis for clustering. We also study the effectiveness of the Principal Component Analysis on session classification based on clustering.

Acknowledgments

This thesis is a result of contributions from a large number of people, who have, in the course of time, helped me understand, support and finish my research. First and foremost I want to thank my advisor Dr. Katerina Goseva Popstojanova for her untiring guidance, support and patience. I would also like to acknowledge the support guidance and valuable suggestions of my committee members Dr. Arun Ross and Dr. V. Jagannathan.

Its true that theory cannot feed without data, and so is my thesis. The credit for regular help in data access for my thesis goes to David Krovich, Lane Department of Computer Science and Electrical Engineering (LDCSEE), WVU, David Olsen, WVU Web Services, WVU, and Brian Kesecker, NASA IV & V Facility, Fairmont,West Virginia. I also want to acknowledge the financial support from the NASA office of Safety and Mission Assurance (OSMA), Software Assurance Research Program (SARP) managed through the NASA IVV Facility . Special thanks goes to my contemporaries Fengbin Li , Amit Sangle, Xuan Wang and Ajay Deep Singh, without whom the work would have not be fun.

Finally, I would like to thank and appreciate the patience of my parents and sister, back in India, who understood and supported me throughout my Masters' program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Web, as we know it today, has developed tremendously from the era of Intranets, LANs and small networked groups. Couple of decades earlier, no one would have imagined a streamlined infrastructure of clients and servers (Web Servers, Application Servers, Database Servers, etc.) working in tandem to accomplish a complex network of applications up and running. Most of the corporations these days have at least a part of this network implemented. This follows with a high demand of availability of such systems with scalable capacity and performance for a better and efficient service provision for the end users of these systems. It has thus become increasingly important to diagnose these servers, integral part of these systems, for patterns and detect regularities as well as anomalies, to keep them 'healthy'.

This boom of data related to web activities on these systems channeled a new area of data analysis or rather data excavation and methods. These were grouped under a generic category of "Web Data Mining".

1.1 Web Workload Characterization

Web Workload characterization [3, 4, 5, 6, 7] have been studied, however these studies were done prior to the year 2000 and the Web system implemented at the time were different. These systems were traditional Web servers which were information oriented. Their main objective was to provide information to end users, mostly static information. Web servers

since then have changed a lot in that they have multiple objectives of supporting E-commerce functions such as transaction support, state transition support and persistent and reliable data storage[8]. These changes in the Web server functionality over time and technology guarantees the change in the Web workload characteristics, which in turn demands a new understanding of these Web workload characteristics.

Studies concentrating on Web workload characteristics[9, 10, 11, 12] in the recent years have provided interesting and important insight on E-commerce workloads. However these studies are less in number because of the scarce availability of real Web server workload data. Corporations are skeptical in lending their public Web server data because of security issues and public abuse. Though some of the public Web servers have maintained an easy access to their raw log files, most of the time they are either outdated or not sufficient to represent the actual domain of the study.

In the study of Web workload a request made by the client is the basic unit for the analysis. These requests over a period of time make the workload. The analysis of these requests is important for meaningful characterization of the workload, but studying the sessions based on these requests is important too. A session is a unit to identify activities by a single user. In some cases sessions give out more information than individual requests. Sessions are also useful in case of clustering, as it is easier to categorize and infer from sessions than a stream of requests. Commercial Web server workload is based on transactions and to understand the interaction between users and the Web system, sessions encapsulating this transaction are more suitable. Session features can improve server performance, e.g. session based scheduling algorithm can improve server responsiveness by almost 50%[13]. Session group characterization has also helped in tuning server performance and scalability[9, 14].

Session definition has been used by many studies[15, 16, 17, 18, 19, 20, 19, 21, 10] as an input to clustering for classification purposes. The basic concept used in majority of the studies consider session as a set of consecutive requests made by the client over a time restriction. Arlitt et. al.[9] uses 15 minutes of inactivity between successive requests by the same client to differentiate sessions, while Popstojanova et. al. [1] uses 30 minutes for the same differentiation. Arlitt et. al. though uses their server setup to timeout sessions after 15 minutes, which is not suitable for our studies as the time out condition is imposed

not derived from raw data, unlike the study done by Katerina et. al. where they use the untrained data set to derive an optimal session timeout condition of 30 minutes. Figure1.1 shows the plot of change in number of sessions with increasing threshold limit of session timeout[1]. After a 30 minute threshold value, the decrease in number of sessions is very minimal, hence suggesting that its almost optimal to select 30. Industry standard cited by Menasce et. al.[22] also suggests using a timeout limit of 30 minutes between consecutive requests to limit session boundaries.

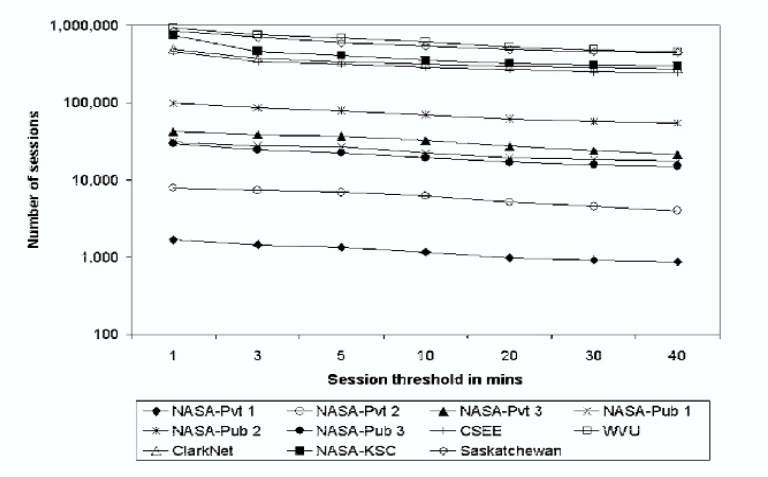

Figure 1.1: Effect of the session thresholds on the number of sessions[1]

A similar study done by Arlitt et. al. [2] uses similar concept to plot number of sessions with different timeout values. Figure1.2 shows the plot of Total sessions/Maximum active sessions with respect to idle timeout value in seconds. Notice how sharply the number of session reduces as the session timeout value is increased from 0 to 100 seconds.

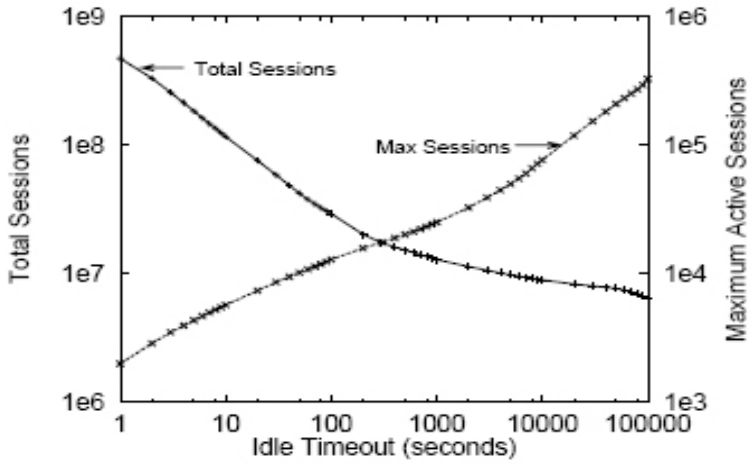

Figure 1.2: Effect of the session timeout values on the number of sessions[2]

1.2 Session Characterization

In the earlier work done, sessions were grouped in different ways in order to infer different type of information as dictated by the goal of the study. Menascè et al.[14] grouped sessions based on navigation patterns to improve the server resource management and optimize revenue. Arlitt et al. [9, 10] grouped session based on the resource usage for handling server performance and scalability issues.Arnoux et al.[15] tried to group sessions based on the navigations, selecting k navigations as the starting point of the dynamic clustering process. Their main objective was to find proper usage of this study and whether or not these are correlated.

Studies done in the area of session characterization and Web Usage Mining provide in-depth understanding of the methods session groups, their characteristics, session group analysis for related studies. However it is not practical to compare these studies as they differ in many aspects : a) how the sessions are defined and represented, b) what algorithms are used to cluster these sessions, c) the domain of the web site under study e.g. whether its an educational web site or a commercial web site or a simulated web site, and also d) what area of concern these studies have.

1.3 Session Parameter Clustering

When we do not have *a priory* knowledge of the user access patterns, unsupervised classification or clustering methods prove to be useful in analyzing the semi-structured log data of user accesses by categorizing them into classes of user sessions[23].

This thesis concentrates on clustering using K-means, and tries to implement principal component analysis on the server logs, but we limit ourselves to as many as four parameters. Most of the studies in recent years are done in the area of session clustering uses the client page viewing URL or/and time spent on a particular page[]. Another aim of our study is to develop a process which can complement the unsupervised learning of session characteristics. Studies are done in the area of unsupervised learning[23, 15, 24, 16, 17]. Major concentration is applied on first finding out navigational patterns and them some similarity/dissimilarity matrix to identify the pattern closeness. Clustering is then applied to search for groups with similar patterns and categorized thereafter.

Previous studies, though extensive in the area of selecting different session attributes for defining the session, did not concentrate on finding the relationship among different session groups resulting from selecting different criteria for session groupings. Menascè et al.[14] optimized revenue leaving server resource usage, by representing sessions with navigation patterns. Arlitt et al.[9, 10] on the other hand represented sessions with resource usage, discussing the server scalability issue without considering the revenue. These studies have hence one major drawback that they consider only one performance related problem at a time. To overcome this problem, several related problems should be analyzed in the same context. In order to have a complete understanding of a Web server we need to draw relationships between different session representations.

Another problem with previous studies is that they concentrate on the outer domain of session groups while neglecting how the inner session group characteristics are related and how they affect the session clusters. Arnoux et al. [15]. suggests *hybrid clustering method*, where Principal Component Analysis (PCA) is used for determining the correlations among the variables and then clustering the principal factors generated by the PCA. They use the *user navigation* to cluster groups of *homogeneous* navigations. Their main objective is to

analyze the relationship between structure of the web site and the log files. Their approach to use PCA and the principal factors for clustering is similar to what we have explored in this thesis. We further this study by exploring variables and their relationship and

Session based analysis also helps us to understand better the dynamic content caching issue. Session identification and study helps in deciding load balancers to direct requests to the proper server handling that session. To conserve resources, application servers time out sessions after 15 minutes of inactivity[9]. The only problem is that they have trained session identifiers. Also the way the application server and the web server are configured, it does not allow to cross-identify the requests in web server logs and the requests in application server logs belonging to the same session.

Issues of stability Issues of noise variables in corrupting the clusters. They propose that removing features with low variance values act as a filter resulting in a distance metric providing a more robust clustering.

A lot of work is going in the area of web log mining for user behaviorial pattern discovery, and web server performance issues. The scope of use of these studies is so vast and unbounded that there are still many areas which need more in depth study. In this thesis we characterize the web workload in terms of sessions based on several intra-session characteristics. A preliminary analysis on characteristics of detectable robots is also done Some emphasis is also done on finding HTTP error response characteristics to understand the behavior of intra-session parameters with respect to session characterization. This is done by comparing our result of clustering and principal component analysis with these error characteristics.

In this chapter we discuss an overview of the area of web log mining for useful information of web server access characterization, what are the components of this area, how the different modules of this area are related to each other, and what core modules are we interested in. The soul purpose of this chapter is to identify the concept of web server logging mechanism and not to indulge ourselves into the vastness of this area. Following sections describe a more detailed understanding of various components involved and their individual importance in our study.

1.4 Thesis Overview

This study analyzes Web access logs at nine public and private Web servers for four different organizations: a commercial public Web server, public server of Computer Science Department at West Virginia University, Public server of West Virginia University, and public and private servers of NASA IV&V Facility, Fairmont. Characterization of Web server workload is done at several different levels to better understand and postulate different theories. The system overhead because of the process involved in our study has also been discussed for practical purposes, helping in determining the resource usage and scalability of our method.

Chapter 2 discusses the work done in the area of Web workload characterization, and different methods such as Principal Component Analysis and Clustering applied to it. Chapter 2 also discusses in detail the methodology used by previous studies in the area of session formation and session characterization, but this is limited to the study we have done in later chapters. Chapter 3 expands the theoretical part of the thesis and explains fundamentals of Web Workload characterization and techniques used in this study. Definitions of PCA, Clustering and few other Clustering measures are explained in this chapter. An analysis of time cost for both PCA and K-Means Clustering is done from resource utilization point of view.

In chapter 4, explanation of the design method for the database and data pruning exercises are explained in detail. A detailed description of the tables and procedures used for data manipulation and analysis is given. Chapter 5 discusses the results and analysis followed by the conclusion and recommendations in chapter 6.

Chapter 2

Related work

This chapter discusses the work done in the field of web log analysis. The main aim is to discuss the work done in the area of unsupervised learning of user session characteristics. We also discuss the work done in the area of web usage mining[15], concentrating on web server session characterization, intra-session parameters and client and server side error characterization. This discussion is based on the methodology used and the area of concentration done by various researchers.

Recent research has explored web user session clustering as a means to understand user activities on a given web system. These studies though effective require a user input to define the number of clusters in advance or analyze a large hierarchy of clusters to find the optimal depth for describing user activity.

We still believe that different web traffic composition based on varied demographic may in fact require this kind of study where an optimal size of clustering is derived every time the traffic data is influenced by the demographics itself.

2.1 Web-server Log analysis

Web servers logs were studied in the past for many different reasons at many different levels. Some of them are being used to find the user statistics or the "scent" of the user[25]. It is sometimes also used to find the fingerprints of the HTTP server[26] of interest. Most of studies done using web logs have used the access pattern data in the logs, i.e. what page

is accessed, how much time has been spent on a particular page. There have been some studies done using the raw log variables to formulate sessions, which are further studied for characterization.

2.2 Web User Session Clustering

Most of the studies in the area of web usage mining are new, and web session clustering has become popular in the field of real application of clustering techniques recently[19]. There are many different tools available which offers a basic summary of web activity, like number of hits on a page, or demographic distribution of users and more. Most of these tool try to group user actions in predefined activities. A number of clustering approaches have been proposed which utilizes the web server logs to define a user action model which is then grouped with a clustering algorithm[17].

Shahabi et al.[17] utilizes the page viewing time as the primary feature for characterizing the user session. K-means is then used to cluster the sessions. Fu et al.[16] uses the page URLs to construct a hierarchy which is then used to categorize the pages. These categorizations are used to describe the page accesses and then clustered using BIRCH algorithm[27]. Banerjee et al.[18] utilizes the combination of time spent on a page and the longest common subsequences(LCS) to cluster the user sessions. The LCS algorithm is applied on all pairs of user sessions, and then this LCS path is reduced using page hierarchy in a generalized based approach called 'Concept-based Clustering'. This is basically a simpler form of Generalized-based Clustering approach, using only the topmost level of page hierarchy for the generalization. Based on similar work Wang et al. [19] considered measuring session similarity as the first step but they considered each session as a sequence and utilized the concept of sequence alignment from the field of bio-informatics to measure similarities between sequences of page accesses. Further they utilize dynamic programming to find the "*Best Matching*" between two session sequences.

Heer and Chi proposed a technique utilizing various information sources for creating user profile model, which are then grouped using Multi Modal clustering algorithm[25, 20]. Their method utilized content and structural data features in addition to the URLs, sequence

ordering and timing data contained in the logs. The drawbacks to these approaches is that while doing Partitional clustering, no methodology to find optimum number of clusters has been used. Also when following hierarchial clustering techniques the optimum or right level to decide number of clusters has to be done manually. The difference in our study and theirs is that we further our research on session parameter characterization for inter and intra-session behavior.

Table 2.1: Web Usage Mining Research Groups

Research Project	**Content**	**Structure**	**Usage**	**Session Clustering**
Menascè et al.[10]			*	K-Means
Arlitt et al.[5]			*	K-Means
Arnoux et al.[15]		*		PCA with Dynamic clustering method
Heer et al.[17]	*	*		Multi Modal Clustering (MMC)
Shahabi et al.[28]		*		K-Means,navigation path, cosine path vectors
Fu et al.[16]		*		Generalization-based clustering method, web sessions, generalized session
Banerjee et al.[18]				longest common subsequence
Wang et al.[19]		*		TURN, ROCK, CHAMELEON , page and session similarity using sequencing
Larsen et al.[24]		*		Hierarchical probabilistic clustering with Independent Component Analysis

Table2.1 explains the major work done in the area of Web Log Mining. It gives an overview of the area of concentration by different research groups in the area of Web Usage Mining, and also presents the detailed methodolgy applied to respective research. A similar table has been organized by Srivastava et al.[29], which gives an comparision of different research projects based on data source, data type, user type and site type. Table2.1 tries to update that list while keeping the objective restricted to this thesis work.

2.3 Contributions of our work

The major area of concentration in this thesis is to find the session characteristics related to the parameters defined in the log files such as request, time of request, number of bytes requested, number of error requests. We have tried to establish sessions and categorized them using unsupervised learning through clustering. Unlike most other studies which concentrated on building user profiles based on their navigational patterns we have used the raw web log parameters to base our session parameters, such as number of requests, length of a session or total bytes requested/transferred in a session. We try to extend the study done on session characteristics with respect to robots and errors. Our main goal is to find out the effectiveness of Principal Component Analysis in session characterization using clustering as a means of unsupervised learning.

Chapter 3

Background

3.1 World Wide Web

Hyper Text Transfer Protocol (HTTP) is a protocol used for interaction on the web. Any transaction on web starts with HTTP (RFC 2616 HTTP/1.1), be it a web site hosting personal information or a big scale e-tailer such as Amazon[1]. HTTP is a request/response stateless protocol which relies on specific methods for requests and responses. The basic methods used for request made by the client are GET, HEAD and POST, while there are some predefined responses for the servers to respond to the clients. Now lets take a look at the Requests sent by the clients over HTTP.

- GET

  This method retrieves the information from the file system on the hosting web server. Static HTML page will display the content while if it is a dynamic page i.e. a dynamic JSP page, the web server will process the JSP file and return the output desired by the application to the client's browser.

- HEAD

  This request is similar to GET but only in terms of functionality, the content returned by the web server to the client is not complete, it has only the header information involved, which might include the server's meta information like server headers, server

[1] http://www.amazon.com

response codes.

- `POST`

  POST is a request from client side which is directed towards the server and directs it to accept the information passed and use it for processing. Mostly these requests are initiated when involving scripting on the server side or CGI scripting. This requires all requests to have valid content-length while sending to the server side script.

3.2 Web logs

All web servers are configured to store logs of client accesses to the server on the web. These logs have the basic setup for capturing data like, where the client is coming from, i.e. IP address, what the client has requested from the web server, i.e. File name - resource demand, and many others. It is a way to make sure that every transaction on the server is not going unnoticed and can be retrieved in future for further analysis. These log formats can be customized but the standard logging mechanism remains the same. Hence we can say with enough confidence that the study we are doing, can be applied evenly on web server running different vendor software. The details of this are discussed in chapter 4.

3.2.1 Data source components

Web Mining has been studied for past several decades in wide variety of areas. For web mining purposes we have to define our web data source and its components, which might not be necessary in this study as it assumes an abstraction layer above these data source components. These components define the types of different data present. A look at what type of data is available on web can be summarized as shown in Figure 3.1.

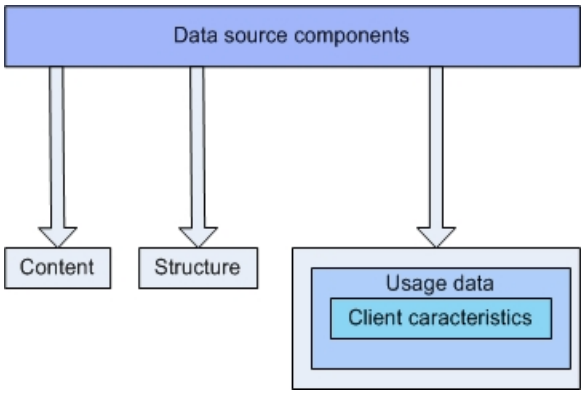

Figure 3.1: Data source components of a web system

Web page content

This is the actual data the web pages contain. The communication between a client and a server on the web encapsulates this data in packets . We will not be categorizing this in our study but it is necessary to know that what different types of contents are available and how they affect the intra-session characteristics.

Web page structure - Intra page structure

This is the organization of the data on the web. It can be explained with the tree structure of the data of an organization, specifying the pages and their hierarchies. This information can be useful as it gives us a look at the structure and what their privileges can be.

Usage data

This data consists of imprint made on the hosting server by user resource requests like IP address requesting the resource, or `User ID` (if the client supports and provides `identid` or `userid`), status code etc. It also logs in the date and time of the access. Apache has a set standard output format for this data, which is again user configurable. Almost all the prevalent web servers have similar data structure for logging, and it is beneficial for our study as we are not constrained by any particular web server. This is the actual data which we are going to concentrate on and base our studies on. Usage data is primarily used to

characterize clients and is analyzed in various ways detailed in chapter 5.

3.3 Multivariate analysis

Multivariate analysis is a major field where the multivariate data sets is being analyzed for patterns and behavior among the set of variables measured over a number of sample points. This behavior is closely related to how the parameters are correlated among themselves. The correlation of parameters governs what the final outcome of the analysis is. Multivariate analysis helps us to remove the need for doing multiple correlations among the variables. Here we have two aspects to consider, first one is the correlation of the variables in picture, also known as *R-mode* analysis and, the second aspect would be to find the relationship between the samples itself (in our case variables are the attributes of a session and the sample is the session itself). The latter approach is often referred to as *Q-mode* analysis.

3.3.1 Principal component analysis

Principal Component Analysis (PCA) is a multivariate analysis technique which helps in data dimension reduction. PCA utilizes the R-mode analysis approach and is probably the oldest ordination technique available.

Introduction

Principal Component Analysis(PCA) tries to find the principal components in the data set which are orthogonally related to each other, i.e. do not have any dependency among them. The final factors have zero vector product, because of this. Each factor (also called as principal factor) defines the variance among the data along that vector. These factors represent the object's properties and hence the variation of data can be explained with respect to these properties.

The first principal component is a single axis in space. When is projected each observation on that axis, the resulting values form a new variable. And the variance of this variable is the maximum among all possible choices of the first axis. The second principal component is another axis in space, perpendicular to the first. Projecting the observations on this axis

generates another new variable. The variance of this variable is the maximum among all possible choices of this second axis. The full set of principal components is as large as the original set of variables. But it is common place for the sum of the variances of the first few principal components to exceed 80% of the total variance of the original data. By examining plots of these few new variables, researchers often develop a deeper understanding of the driving forces that generated the original data[30].

Statistically, given a set of n parameters $x_1, x_2, ..., x_n$, the principal component analysis gives a set of factors $y_1, y_2, ..., y_n$ such that following statements holds true[31]:

1. The set of factors i.e. y's are linear combinations of x's:

$$y_i = \sum_{j=1}^{n} a_{ij} x_j \tag{3.1}$$

where a_{ij} is ***loading*** of variable x_j on factor y_i

2. The set of factors are orthogonal to each other, i.e. their inner product is zero:

$$\langle y_i, y_j \rangle = \sum_{k} a_{ik} a_{kj} = 0 \tag{3.2}$$

In simple terms that means all y_i's are uncorrelated.

3. The last property states that all y's are ascending ordered with respect to the amount of variance in resource demands for that particular factor. This is the most important property as it enables us to eliminate the high degree of dimensionality in the given data sets. First few factors can be used to classify the workload components[31].

The general theory of principal component analysis allows us to choose first few principal factors to explain almost $90-95\%$ of the total variation in the data, depending on the distribution of the data. These principal factors are hence useful in reducing the dimensionality problem, reducing the final data set dimensions to 2 or 3. This also helps the analysis as it is easier to visualize and represent the final output, which is one major attraction of PCA.

PCA : A step by step approach

The calculation of principal components is a lengthy task of finding correlations, *eigenvectors*, and principal factors. The process involved is straightforward though. The following

is a detailed explanation of our process of finding principal components from the raw data-sets. The 'R' procedure used in this thesis for PCA calculations has been developed by E James Harner[32].

1. **Mean and standard deviation of the data**

   First of all the data is used to calculate the mean and standard deviations of the properties of the data. If the data we have has n records and m parameters the representation of the means and standard deviations are given as follows:

$$\overline{X} = F(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_m)\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ji} \quad ; \quad for \quad j = 1 \quad to \quad m$$

$$S_X^2 = F(\sigma_{x_1}^2, \sigma_{x_2}^2, \ldots, \sigma_{x_m}^2)\sigma_{x_j}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ji} - \overline{x}_j)^2 \quad ; \quad for \quad j = 1 \quad to \quad m \tag{3.3}$$

2. **Zero mean normalization with unit standard deviation**

   The data we have is a multi-parametric in nature, where the scale of each parameter or property is different and not necessarily comparable. This nature of the data is handled by normalizing it. The simple mean of the corresponding parameters is not sufficient and simple comparison over the parameters values is not fruitful, hence the the data is pruned by normalizing the variable values with respect to the mean and the standard deviation of the data set. The normalized values can be represented as following:

$$X' = F(x_1', x_2', \ldots, x_m')x_j' = \frac{x_j - \overline{x}_j}{\sigma_{x_j}} \quad ; \quad for \quad j = 1 \quad to \quad m \tag{3.4}$$

3. **Correlation matrix of the variables**

   The correlation between the data set properties is calculated and a correlation matrix is being populated corresponding to it. The right diagonal elements of this matrix have a value of 1 (as the correlation is between a single property), while other elements of the matrix are the values representing the correlation coefficient between the two properties.

4. **Eigenvectors of the correlation matrix**

   Eigenvalues are first calculated of the correlation matrix, which is used to solve the equation to find *eigenvectors*. The equation defining the *eigenvector* can be shown as:

$$\mathbf{C}\mathbf{q}_1 = \lambda_1 \mathbf{q}_1 \tag{3.5}$$

   Where, $\mathbf{q}_1$ is the *eigenvector*,
   λ_1 is the first *eigenvalue*
   and, $\mathbf{C}$ is the correlation matrix.

5. **Principal factor calculations**

   Once *eigenvectors* are calculated, principal factors are derived by the product of *eigenvectors* to the normalized vectors. Sum of all principal factor values should be 1 for a given property, and sum of squares of the principal vector gives the percentage of variation explained by that principal factor.

Principal Component Analysis : A geometric explanation

This section explains the mechanism of principal component analysis based on geometric perspective. Consider a data set as a collection of m variables over a size of n points of observation. This yields a matrix of size n x m. So if we visualize this data set it might look like data points in a m dimension space, where each data-set is being positioned based on the m values associated with that data point.

In short, what PCA analysis does to this m dimensional data set is reorganize on the axis where the maximum variation of the data can be explained, and these new axis for the data set are known as principal components. These axis are orthogonal to each other, that is, there is no correlation among them. Figure 3.2 shows the relocation of the original axis ($X1$, $X2$) to fit on the new defined principal components axis, $PC1$ and $PC2$ respectively.

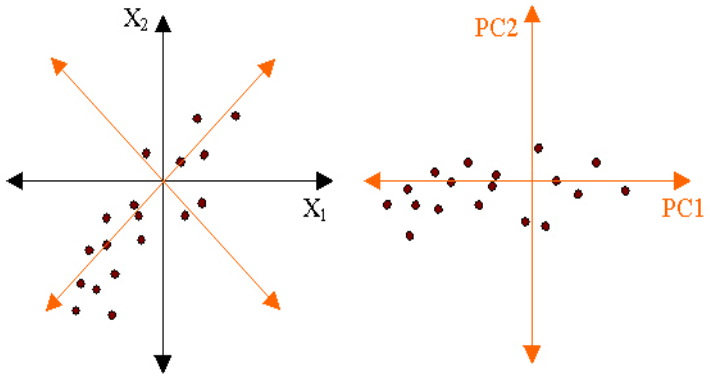

Figure 3.2: Geometric representation of PCA as re-projection along new coordinates

As we can see from the Figure 3.2, PCA constitutes a new set of axis, which are linear combinations of the original axis. It also aligns the original axis in the directions defining the maximum variation of the data. Hence we can say that PCA finds the new coordinate system for the data, which represents the internal variability of the data. It also helps in selection of first few axis which have the maximum combined total variation associated with them.

3.3.2 Clustering

Clustering is one of the age old methods to handle multivariate data for analysis purposes. For a low dimensionality problem of analysis does not produce a challenge as far as human mind is concerned, but anything above 2 or 3 dimensions can make the problem complex enough for what can be comprehended from it. Clustering is a process where a given set of data points are divided into groups or cluster of points. These individual clusters have no data points in it which share another cluster in the given cluster set. The data points in a given cluster are "more similar" to each other that to data points in other clusters. This "similarity" is decided by some measure of proximity bringing a set of data points closer in a cluster, while leaving other data points out.

Data in scientific studies tend to gain a higher dimensionality than what we observe in our daily life. Decisions made on the basis of few parameters in a given problem can work well but the same process cannot be followed if the parameters to decide upon increases dramatically. Humans have a natural tendency to group things into categories, which have entities having more likeness towards that category compared to other categories.

Introduction

Clustering is a type of classification imposed on finite set of objects, where these objects are bound together in relationships through a proximity matrix. In fact the proximity matrix is the one and only input to any given clustering algorithm[33]. The data points of a data-set can be visualized as objects in space where the proximity matrix defines the distance or some other form of relationship between them (e.g. Euclidian distance). Hence the proximity matrix can be defined as rows and columns corresponding to these data points, and having values of their intra-distance metric.

Cluster analysis is the process of organizing the data in groups, such that each group is a separate entity when compared with respect to the parameters defining the clusters. This means that the properties of a cluster in a set of clusters is unique to itself with respect to some of the parameters involved in cluster analysis. This similarity can be absolute or can have a degree attached to it as in the case of fuzzy clustering[34]. For example an absolute

relationship will require to have the data point value of either in or out (1 or 0) for a given cluster, while in other cases it might have a degree attached to it, 0.8 indicating a strong affinity towards the given cluster or 0.2 indicating the alleviating degree of bond with the given cluster.

Clustering classification

Classification of a clustering technique can be done in 2 top level denominations, i.e. Exclusive and Non-exclusive clustering techniques (see Figure 3.3). Exclusive clustering technique requires the final outcome of clusters to have no subsets shared among them, i.e. the intersection of any two resulting clusters should produce a null set. In case of a Non-exclusive clustering technique, there might be some overlap of data points or objects among clusters because of the nature of the objects' properties taken into consideration for the clustering technique. For example, if we are considering the source of an image file, it might happen that two or three different referrer might have referred the same file, so in case if we are clustering them based on that, that particular object might find place in two or three different clusters.

Exclusive clustering technique can be further subdivided into Extrinsic and Intrinsic classifications. Extrinsic classification takes the help of category labels on the objects and the proximity matrix, while in the case of Intrinsic classification it is done only with the help of proximity matrix and hence the label "Unsupervised learning".

Exclusive, Intrinsic classification can be subdivided into Partitional and Hierarchical classification based on the type of structure imposed on the data. Hierarchical classification divides the data points in nested sequence, while Partitional classification divides them into single partitions. Hierarchical methods are good for smaller data sets where the resulting dendrograms provide the analyst a good view on how the data is clustered or split at different levels of proximity. If the number of patterns increase above hundreds, dendrograms are deemed useless.

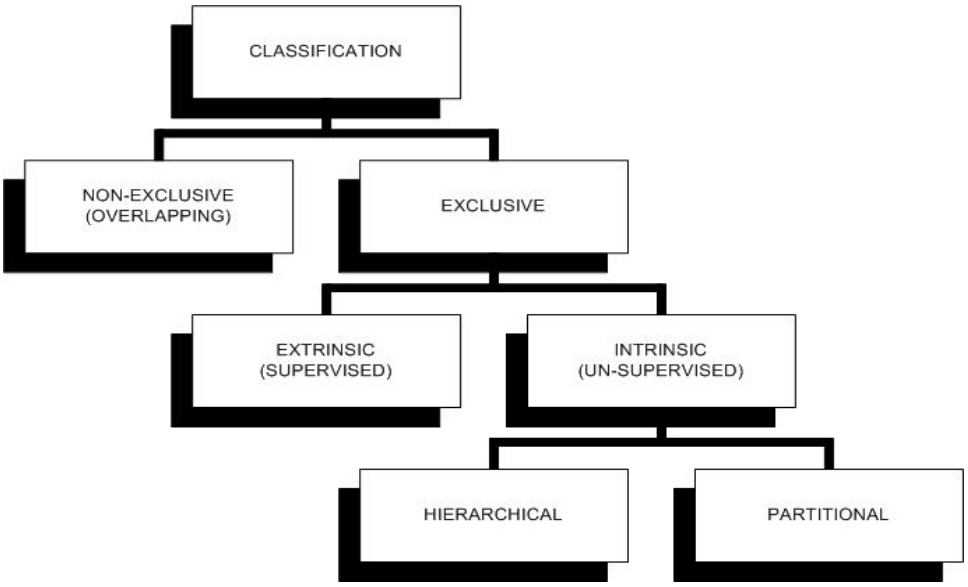

Figure 3.3: Cluster classification

Why clustering?

The concern in our case was which method to use for our analysis. Clustering seemed the most appropriate technique as we did not want to corrupt the data by "teaching" it, rather we want to perform unsupervised classification. Clustering has a long history of applications in various fields such as image processing, biometrics etc. One problem with unsupervised learning is that most of the clustering algorithms create clusters even if there are none present in the given data set[35]. This creates a huge analytical error as the data behavior shown through clustering would not represent the exact behavior of the given data set.

This problem can be solved by doing the stability test for the data to be clustered. This stability testing can be done in many different ways, such as sampling based methods. These methods are based on a common idea that if a partition captures the structure of a data set, this partition should be stable with the perturbation of the data set.

Advantages of clustering on large data-sets

- Data reduction.

  This can be accomplished by replacing the coordinates of each data point of a cluster with the coordinates of that cluster's reference point[34].

- Minimizing storage.

  Once we have data reduction, the storage requirements for the data also reduces. We can do some data pruning using techniques such as Principal Component Analysis (PCA). Its not only easier to reduce dimensionality but it also helps us in reducing the data size, ultimately reducing the storage requirements.

- Easier data handling.

  Clustering allows us to categorize and shift data points in well defined clusters. This can help us in our analysis as it creates an easier channel to focus through.

- Proved and well researched technique.

  Clustering has been used for a long time. In some form or the other clustering has been used as long as the early human started thinking.

3.3.3 K-Means Clustering

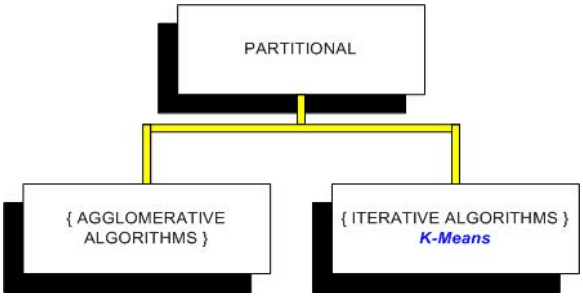

Figure 3.4: K-Means clustering

Agglomerative or Divisive algorithms are based on selecting all data points at once and assigning them to either n or 1 cluster respectively. One can say that they either have a top-down or bottom-up approach of clustering the data set. For example in case of agglomerative clustering techniques the whole data set is divided into the smallest possible cluster sizes with all possible combination of clusters, may be each data point constituting a cluster of its own. This is followed by merging these atomic clusters into larger clusters. While divisive algorithms work the same way, they approach the problem from top and process through the bottom till desired clusters are formed.

K-means clustering is one of the clustering methods in which k points are selected as center of clusters, and the data points are located around these centers such that the average squared distance from those data points to the assigned centers is minimum. The proximity matrix used for this is basically the Euclidian distances between data points or objects in the study. The Euclidian distance is defined as the shortest path between two points x y along a chosen 2-D or more general n-D space i.e. Euclidian space R^n and can be explained by the following mathematical formula:

$$d = |\mathbf{x} - \mathbf{y}| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{3.6}$$

Based on this we can define the distance $d_{a,b}$ between two points X_a, X_b in the session log as[10],

$$d_{a,b} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(C_a[i,j] - C_b[i,j])^2} \tag{3.7}$$

Approach to K-means clustering

The k-means algorithm which we use in our studies is based on the algorithm developed by Hartigan and Wong (1979) [36]. It is an algorithm for clustering N data points into K disjoint subsets S_j containing N_j data points so as to minimize the sum-of-squares criterion.

$$J = \sum_{j=1}^{K}\sum_{n\epsilon S_j} |x_n - \mu_j|^2 \tag{3.8}$$

where x_n is a vector representing the nth data point and μ_j is the geometric centroid of the data points in S_j. It does not achieve a global minimum of J over the assignments. This algorithm uses discrete assignment rather than a set of continuous parameters, hence the minimum it reaches cannot be even labeled as local minimum[30]. Except for Lloyd-Forgy method of K-means clustering algorithms, k clusters will always be returned if a number is specified. If an initial matrix of centers is supplied, it might be possible that none of the points are closest to one or more of those centers, which is an inherent error of Hartigan-Wong methodology of K-means algorithm[37].

Cluster Validity

Jain and Dubes proposed a validity index for clusters made by CLUSTER [33] algorithm. We used the same validity index to understand the "compactness" and "isolation" of clusters among other clusters. The validity index is the ratio of minimum squared distance over all properties from one cluster to another, i.e. inter-cluster distance, and average distance of all cluster points to the centroid of that cluster summed over all the properties, i.e. intra-cluster distance.

$$S_k = \frac{\overset{min}{\underset{l \neq k}{l}} \sum_{j=1}^{d} (m_j^{(k)} - m_j^{(l)})^2}{\{\frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{d} (x_{ij}^{(k)} - m_j^{(k)})^2\}^{1/2}} \tag{3.9}$$

where n_k is the number of patterns in cluster k,

$m_j^{(k)}$ is the cluster center for cluster k, along feature j

d is the number of properties,

and $x_{ij}^{(k)}$ is the value of the jth feature for the ith pattern belonging to cluster k.

Large values of S_k indicates the clusters have good isolation factor and are highly compact.

Clustering Efficiency

Menasce et al [10] have proposed measures to calculate the clustering algorithm efficiency. They defined two random variables, the average intra-cluster distance $\tilde{d}_k$, and inter-cluster distance between cluster i and j for $i \neq j$. The average intra-cluster distance for cluster k can be represented in terms of Euclidian distance:

$$\tilde{d}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \sqrt{\sum_{j=1}^{d} (x_{ij} - m_j^k)^2} \tag{3.10}$$

where n_k is the number of patterns in cluster k;

d is the number of properties under study,

x_{ij} is the i, jth value of data-matrix,

i is the data point and j represents the property,

and m_j^k is the centroid vector value for kth cluster.

We calculate the sample mean $\bar{d}$, sample variance σ^2_{intra}, and sample coefficient of variation C_{intra} for the intra-cluster distance.

$$\bar{d} = \frac{1}{k}\sum_{j=1}^{k} \tilde{d}_k \tag{3.11}$$

$$\sigma^2_{intra} = \frac{1}{k-1}\sum_{j=1}^{k}(\tilde{d}_k - \bar{d})^2 \qquad k > 1 \tag{3.12}$$

$$C_{intra} = \sigma_{intra}/\bar{d} \tag{3.13}$$

The sample mean $\overline{D}$, sample variance σ^2_{inter}, and sample coefficient of variation C_{inter} of the inter-cluster distance is computed using the equation.

$$\bar{D} = \frac{1}{k(k-1)/2}\sum_{j=1}^{k}\sum_{j=i+1}^{k} \tilde{d}_k \tag{3.14}$$

$$\sigma^2_{inter} = \frac{1}{k-1}\sum_{j=1}^{k}(\tilde{d}_k - \bar{d})^2 \qquad k > 1 \tag{3.15}$$

$$C_{inter} = \sigma_{inter}/\bar{D} \tag{3.16}$$

The purpose of these metrices is to measure the effectiveness of the clustering process. The clustering is suppose to reduce the intra-cluster variance while maximizing the inter-cluster variance[10]. This can be achieved optimally if all the clusters are made of single data-point. But a good representation of the web logs can be done only if we have small number of distinct clusters, such that the intra-cluster variance is small and inter-cluster variance is large. Menasce et al[10] also suggests using two ratios: β_{var}, the ratio of intra and inter-cluster variance, and β_{cv}, the ratio of intra and inter-cluster coefficient of variation. The smaller these ratios are, better are the clusters.

The ratios β_{var} and β_{cv} can hence be represented mathematically in equations 3.6 and 3.5

$$\beta_{var} = \frac{\sigma^2_{intra}}{\sigma^2_{inter}} \tag{3.17}$$

$$\beta_{cv} = \frac{C_{intra}}{C_{inter}} \tag{3.18}$$

Menasce et al[10] plotted these variables against varying k values. Figure 3.5 plots the values of intra-cluster coefficient of variation, inter-cluster coefficient of variation, and their ratio, i.e. C_{intra},C_{inter}, and β_{cv} against an increasing value of k. It is interesting to note that C_{intra} remains constant while the values of C_{inter} increases as the value of k increases. We will see a change in that observation of our results later.

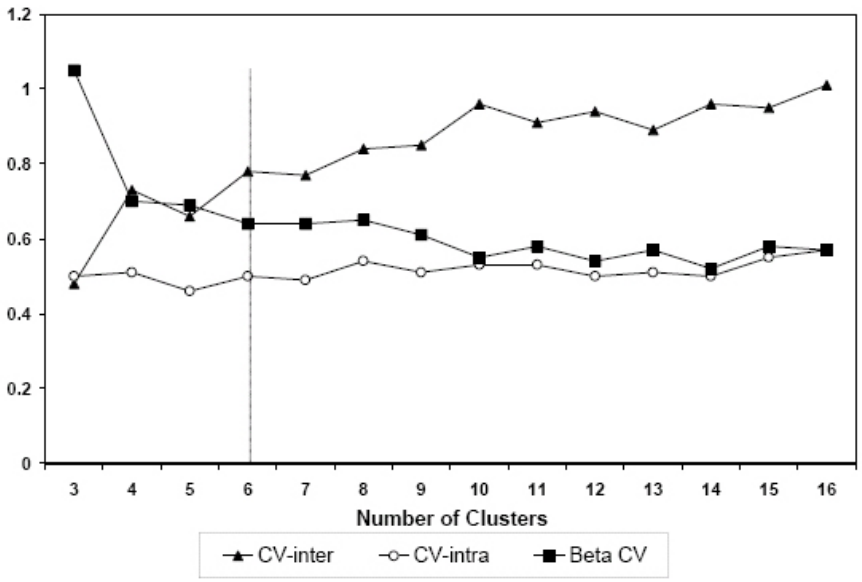

Figure 3.5: Inter and intra cluster coefficients of variation and β_{cv} vs. k.

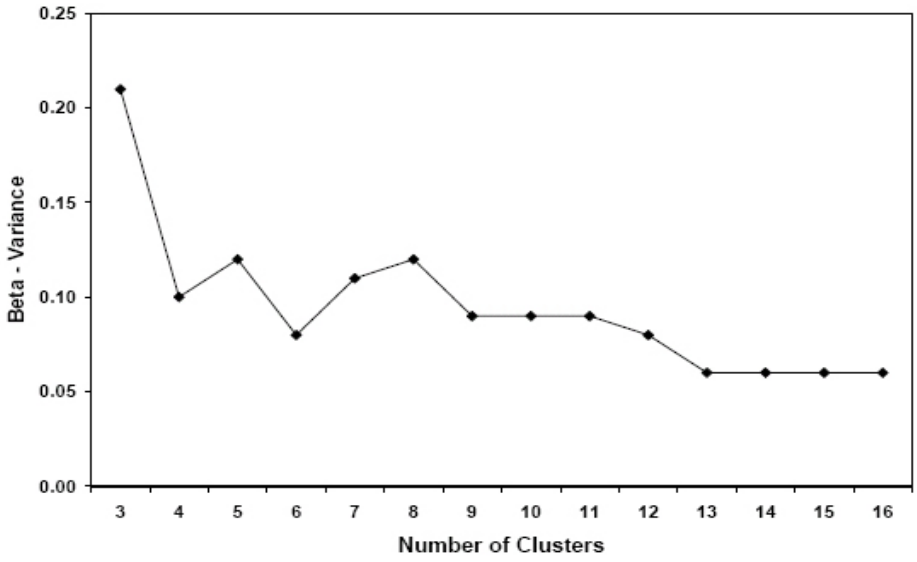

Figure 3.6: β_{var} vs. k.

Their observation was that k value from 3 to 6 shows a sharp decrease in the value of β_{cv}, after which it stabilizes. Furthermore β_{var}, refer figure 3.5, reaches a local minimum at the same vale of k at 6.

Chapter 4

Design & Approach

This chapter discusses the need to design a relational database for log storage. Later on it discusses the details of session and intra-session parameters and how they are formed. This chapter is organized by first explaining the setup, followed by the log file description. The flow of data processing is explained next with the output tables containing our processed data for final queries. Lets take a look at our experimental setup first.

4.1 Experimental setup

Our main concern in designing this system was to handle large amounts of data on a weekly basis. We also needed to design it in such a fashion so as to make the whole system scalable in various aspects, like number of users accessing the central database, increase in analysis data repository, complex data manipulations in post-database analysis, and many more issues like this. There were differences in this selection process which were due to the fact that certain products were not the only options in its category. The ease of use and prior knowledge of some systems and products were other deciding factors.

Our process of design setup takes an approach of 3 layer architecture, where we have a log file server as the backend layer of the design system, database server acting as the middle layer and our frontend tier is composed of applications which gives us on the fly query results or saved csv/xls files on storage workstation depending on user requirement. Note that this layer mechanism is not similar to 3 tier implementation mechanism in case of application

development , i.e. such as used in a typical J2EE or .NET architecture, rather its based on storage mechanism coupled with functionalities.

This architecture boasts all three components in a modular structure which gives us a greater flexibility of tool integration and also helps us in frontend development. A brief overview of the whole setup is summarized in Figure 4.1. Other than this basic setup, we have R, a open source statistical application which has been used extensively to draw most of our graphs and plots. Microsoft Excel was also used to plot graphs and tables for final results and analysis. Oracle 10g was used for the database creation, Toad was used as a front end application to access it.

Figure 4.1 explains the logic behind the design and the experimental setup.

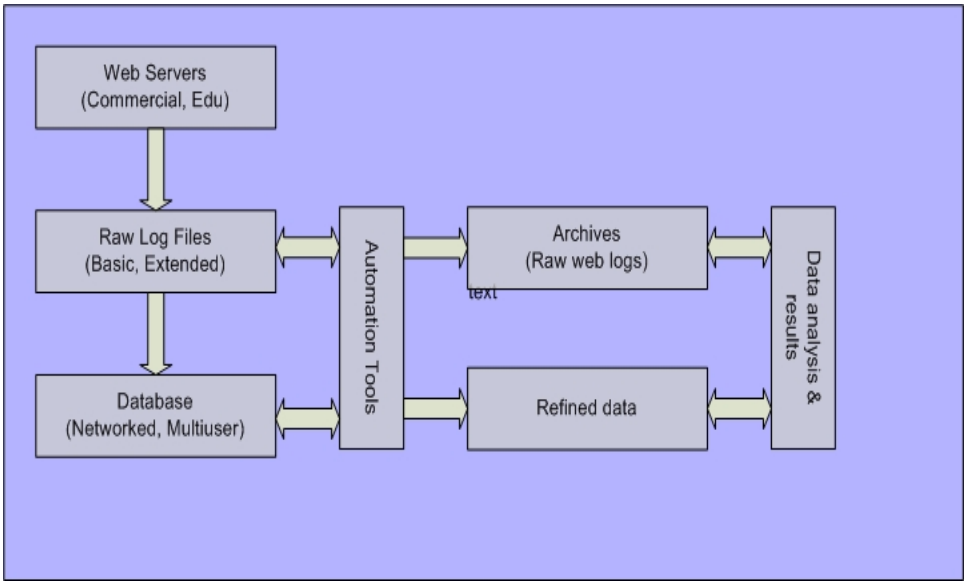

Figure 4.1: Data extraction process

The server logs which were studied in this project were from 9 different web sites:

- Clarknet - Sometimes also referred as CNET, a commercial internet service provider.
- CSEE - Lane Department Computer Science and Electrical Engineering department at West Virginia University.
- WVU - West Virginia University

- Three NASA public web servers - NASA-Pub1, NASA-Pub2, and NASA-Pub3
- Three NASA private web servers - NASA-Pvt1, NASA-Pvt2, and NASA-Pvt3

4.2 Log file storage and access log format

Most of our log files are generated by servers running Apache web server, except for WVU Web Services Web-logs which is running Microsoft's IIS web server.

We have a dedicated server, with enough capacity to store our raw data logs, which are furnished by various Web server administrators. Our plan is to capacitate this server to host number of software applications for better and faster performance. We have, other than the server, individual workstations, which are used for the front end application services.

Lets start with an example which illustrates the basic structure of a web log and explains what are the key features to be explored. A simple, unaltered standard web log[38] for an Apache web server hosting any kind of web service can be represented as:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

where common is the variable name given for this particular log format. This association of variable 'common' with the given format can be used to log events into output files. We can also use literal characters to log them into the output files as it is. There are any flavors of Web servers in the market but all of them follow the Common Log Format (CLF)[38]. A sample log file entry produced in CLF looks like this:

```
157.182.209.1 - john [15/Nov/2005:12:04:3 -0500] ‘‘GET /apache_gb.gif HTTP/1.0’’ 200 2326
```

This can be mapped onto the fields these variables represents as follows:

```
RemoteHost Identity Authorization [Timestamp] ‘‘Request Line’’ Status Bytes
```

Now lets take a look at the meaning of these percent directives and what piece of information each of those log in the access logs.

- `RemoteHost` - The IP address of the client or the remote host requesting to the web server. This directive can be alternated to log the host name instead of the host IP address.
- `Identity` - The RFC 1453 identity of the client determined by the "identd" on the client machine. If the value is not available then a "-" (hyphen) is recorded instead.
- `Authorization` - The userid of the client requesting the document as determined by HTTP authentication. This results in a "-" (hyphen) if the page requested is not password protected.
- `[Timestamp]` - The time stamp when the server finished processing the request from the client. The format used is `[dd/mmm/yyyy:hh:mm:ss zone]`. The logs used in this thesis have a granularity of 1 second.
- `Request Line` - This contains the HTTP method used (i.e. `GET`, `POST`), resource requested (i.e. `/apache_gb.gif`), and the protocol used by the client along with its version (i.e. `HTTP/1.0`). The general format looks like ``‘‘GET /apache_gb.gif HTTP/1.0’’``
- `Status` - The status code returned by the server in response to the client's request. `2XX` level codes represents successful requests, `3XX` level codes are used for redirection purposes, `4XX` level codes are request error codes (client side), while `5XX` level are server side errors.
- `Bytes` - This is the bytes transferred between client and the server.

Revisiting the example given above, the log entry indicates that a password protected page was requested by `john`, requesting the file `/apache_gb.gif`, from the IP address `157.182.209.1`, was successfully (notice the response code `200`) fulfilled by the server on November 15, 2005 at 4 minutes and 3 seconds past noon, eastern time (`-0500` GMT).

4.3 Methodology

In this section we will discuss how we process the raw log files and store them in tables. It also discusses in detail about the procedures and methods we used, resulting in final tables for analysis purposes.

Server	Time period	Log files
WVU	02/15/04-02/29/04	4
Clarknet	08/28/95-09/03/95	1
CSEE	02/15/04-02/29/04	8
NASA-Pvt1	04/06/04-08/30/04	20
NASA-Pvt2	04/06/04-08/30/04	20
NASA-Pvt3	04/06/04-08/30/04	20
NASA-Pub1	04/06/04-08/30/04	20
NASA-Pub2	04/06/04-08/30/04	20
NASA-Pub3	04/06/04-08/30/04	20

Table 4.1: Server information

First of all lets take a look at the servers and data we have analyzed. Table 4.1 shows the details of the servers and time period we have used for our analysis purposes. It also shows how many log files we had for each server. All the NASA servers were studied for a period of 20 weeks, while in case of WVU and CSEE servers we used 2 weeks data. Clarknet was the only server with old data.

What happens to log files?

Log files are processed through a compiled Java unit which automates the process of raw data parsing and inserting into the installed database. This process is the longest in terms of time, among all processes. Log files are simple text files before they are processed, but after getting into the database they are stored in the Oracle native data format. After this process completes, the data is ready to be queried. But for our purposes we have built another module(stored PL/SQL procedures), which are used to preprocess data and recreate user suitable data tables.

All these processes are automated but not web enabled. Many factors, such as cost, time and maintainability prevented us from implementing it. Future tasks can include these

automation as one of the options for better process control.

The old raw log files are now stored inside the database and have a more structured representation in terms of fields and meaning of those fields. We have a single, big table for the initial data input which, stores all the raw data. The reason behind a single table to store all data was to keep the end user queries simple and efficient at the same time. Our approach is based more on a data warehousing principal, where we have report oriented data storage rather than object oriented data storage keeping the efficiency as the main motive.

4.3.1 Data table

Our data table has an extra field "session-id" along with all other fields in the standard raw log file. We will refer this table as "Data table" throughout this document. We have developed couple of PL-SQL procedures which sit inside the data base and are used to populate this "session-id" field according to the records present in the Data table. This gives us or first look at making of sessions and later on a separate "Session table".

Separate data tables are created for various different log files from different servers and different time period as well. Table 4.2 describes our Data table parameters, we have in our database. Notice the session id field which is assigned through a stored PL-SQL procedure based on heuristics developed by Postojanova et al[1]. The session id assignment is based upon many different parameters and assumptions.

A data table has 'Record Id' as a Primary key and also has a sequence associated with it. This sequence is used later on to create and populate our Session table. The Figure 4.2 shows screen shot of the Data table we have in our database.

RECORD_ID	IP_ADDR	IDENT_ID	USER_ID	ACCESS_TIME	REQ_METHOD	REQ_URI	REQ_URI_LEN	PROTOCOL	PROT_VERSION	CGI_ERROR_TXT	STAT_CODE	BYTES_TRFD	TMSTMP_CREATED	SESSION_ID	LOG_NAME
4566194	[redacted]	-	-	30-DEC-	GET	[redacted]	1	-	-		404	14067	07-NOV-	605	WVU_24DEC_31DEC_B
4566195	[redacted]	-	-	30-DEC-	GET	[redacted]	1	-	-		200	16384	07-NOV-	146275	WVU_24DEC_31DEC_B
4566196	[redacted]	-	-	30-DEC-	GET	[redacted]	1	-	-		200	1169	07-NOV-	149404	WVU_24DEC_31DEC_B

Figure 4.2: DATA EXTRACTION PROCESS - The Data Table Generation

A session is defined as a set of consecutive requests made by the same client[1]. These requests are made by the client visiting a single Web site. A session is started when a

[1]A client is one single source unlike a user which can be masked behind a proxy.

client requests a resource from the Web site. Once a request is received by the Web site the server responds by a response to the client, which generates multiple requests for embedded resources. These internal requests also combine together and form a part of the session. With subsequent client requests, the session grows as in the case of a online shopping Web site where a client might request pages related to product browsing, product details, and finally a product purchase. The steps illustrated are simplified operations for which there might be multiple requests based on the implementation architecture of that particular Web site.

Our work assumes that no human user can have a session with two consecutive requests more than 30 minutes apart. Our algorithm to separate such sessions checks for the time stamp and IP address for comparison purposes. The sessions formed are sorted according to the time stamp of the starting request of individual sessions.

Field	Meaning
RECORD_ID	Unique Record ID
IP_ADDRESS	Unique IP Address of the client
IDENT_ID	Identity of client according to "identd" on clients machine.
USER_ID	User id determined by HTTP authentication
ACCESS_TIME	Time when server finishes processing of request
REQ_METHOD	First part of request line, method used by the client
REQ_URI	Second part of the request line, resource requested by the client
REQ_URI_LEN_ERROR	Flag set for any abnormal length of resource requested by client
PROT_VERSION	Third part of the request line, protocol used for communication
CGI_ERROR_TXT	Catch CGI errors by logging the error text generated
STAT_CODE	Status code sent by the server back to the client
BYTES_TRFD	Size of the object returned by the server in terms of bytes
TMSTMP_CREATED	Time when this log was logged into our database
SESSION_ID	Session id of the record
LOG_NAME	Unique identification for the raw log files input in the database

Table 4.2: Request parameter explained

4.3.2 Session table

Session id field in the data table is populated once the algorithm calculates and assigns appropriate session number to each request record. As described above the sessions are

formed using the time cutoff between two consecutive requests from the same client. Once the session id field is populated we use a stored PLSQL procedure to calculate information about individual sessions like session id, total number of requests, session length in seconds etc. This information is stored in the session table generated by another PLSQL stored procedure.

Once the Data table is created, and session id's are populated, we create another table, Session table which, effectively contains the summarized data according to sessions created in the Data table. Effectively a Session table stores information regarding a session,i.e. its characteristics like session id, bytes transferred, number of requests, etc. It also contains the individual count of all the 400 and 500 error level counts, with total counts as well. Session table has 'Session Id' as the primary key and its unique. In fact the way our Session table is made, the whole record itself is unique as it organizes the data according to each session, as they are created, so even if the users are repeated inside the Session table, their Session id won't. This means, our session table might have two sessions belonging to the same user, but the session id will be different. This might occur because of a user browsing the same Web site after a gap of 30 or more minutes.

A detailed description of the Session table is given in table 4.3. This details all the fields we calculate and populate in our Session table. The figure 5.35 below shows the actual database setup for the session table.

SESSION_ID	SESSION_COUNT	SESSION_LEN	BYTES_TRFD	ERR_400_COUNT_ALL	ERR_500_COUNT_ALL	E_CNT_400	E_CNT_400	E_CNT_401	E_CNT_402	E_CNT_403	E_CNT_404	E_CNT_405	E_CNT_406	E_CNT_407	E_CNT_408	E_CNT_409
572	52	+000000000 00:00:23	166184	9	0	0	0	0	0	0	9	0	0	0	0	0
573	49	+000000000 00:00:04	125894	6	0	0	0	0	0	0	6	0	0	0	0	0
574	147	+000000000 00:00:39	124763	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.3: DATA EXTRACTION PROCESS - The Session Table Generation

Field	Meaning
SESSION_ID	Unique session ID
REQUEST_COUNT	Total number of requests in that session
SESSION_LENGTH	Total duration of the session in seconds
BYTES_TRFD	Total number of bytes transferred in the session
ERR_400_COUNT_ALL	Number of requests in the session having status code starting with 4
ERR_500_COUNT_ALL	Number of requests in the session having status code starting with 5
E_CNT_400	Number of requests in the session having status code 400
E_CNT_401	Number of requests in the session having status code 401
E_CNT_402	Number of requests in the session having status code 402
E_CNT_403	Number of requests in the session having status code 403
E_CNT_404	Number of requests in the session having status code 404
E_CNT_405	Number of requests in the session having status code 405
E_CNT_406	Number of requests in the session having status code 406
E_CNT_407	Number of requests in the session having status code 407
E_CNT_408	Number of requests in the session having status code 408
E_CNT_409	Number of requests in the session having status code 409
E_CNT_410	Number of requests in the session having status code 410
E_CNT_411	Number of requests in the session having status code 411
E_CNT_412	Number of requests in the session having status code 412
E_CNT_413	Number of requests in the session having status code 413
E_CNT_414	Number of requests in the session having status code 414
E_CNT_415	Number of requests in the session having status code 415
E_CNT_416	Number of requests in the session having status code 416
E_CNT_417	Number of requests in the session having status code 417
E_CNT_500	Number of requests in the session having status code 500
E_CNT_501	Number of requests in the session having status code 501
E_CNT_502	Number of requests in the session having status code 502
E_CNT_503	Number of requests in the session having status code 503
E_CNT_504	Number of requests in the session having status code 504
E_CNT_505	Number of requests in the session having status code 505
SESSION_START_TIME	The second value a session starts

Table 4.3: Server intra-session parameter values and error codes

These error code counts are one of the important aspects of Session table with regard to our study but we have also included other fields such as total number of request per session, bytes transferred and length of the session, to capacitate other studies on the same set of data.

4.3.3 Frontend applications and scripts for server access

There are few compiled/stored PL-SQL procedures which are used to access data and modify according to the requirements. Few other procedures are used to create sessions and Session table. Below is a screen shot of TOAD [2], a user interface to Oracle database. What we have used is a non commercial version found at ToadSoft[3]. This site also has supported softwares for MySQL and SQL Server.

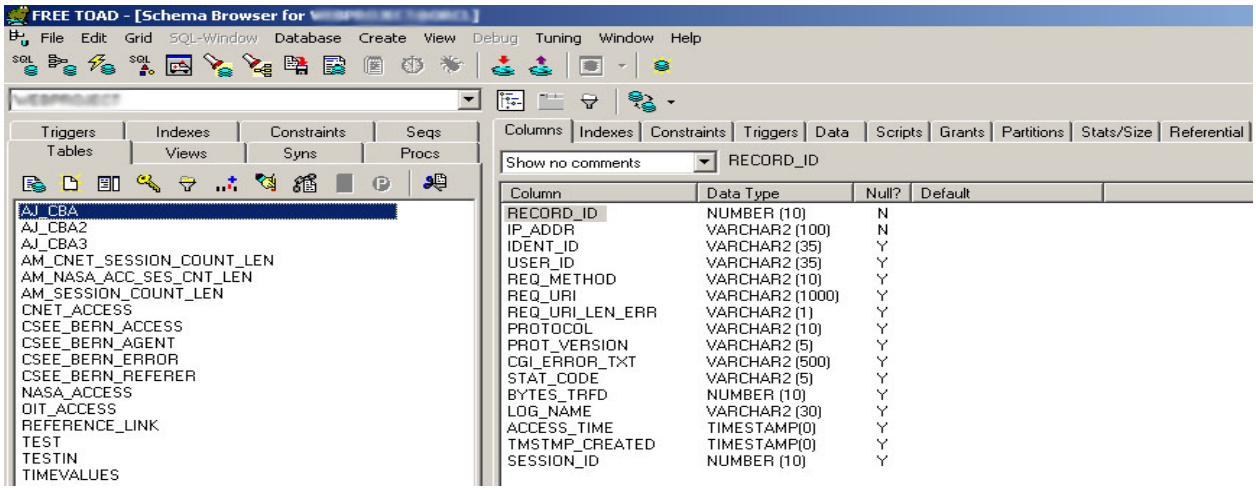

Figure 4.4: TOAD- GUI for Oracle database(tables)

Java modules

Java modules are primarily used to process the raw data to log them into the database. We have also used Java modules to transform data into different format so that it is suitable to process them for different applications like, graph plotting application R. There are few other java modules which are used for data testing and minor updates in the datafiles. We utilized OOPS concept to structure our Java modules, so they can be easily extended or cut off as required. The degree of flexibility is so great that once we created our code for access log analysis , job of building modules for analysis of other logs became piece of pie.

[2]http://www.quest.com/toad/
[3]http://www.toadsoft.com/

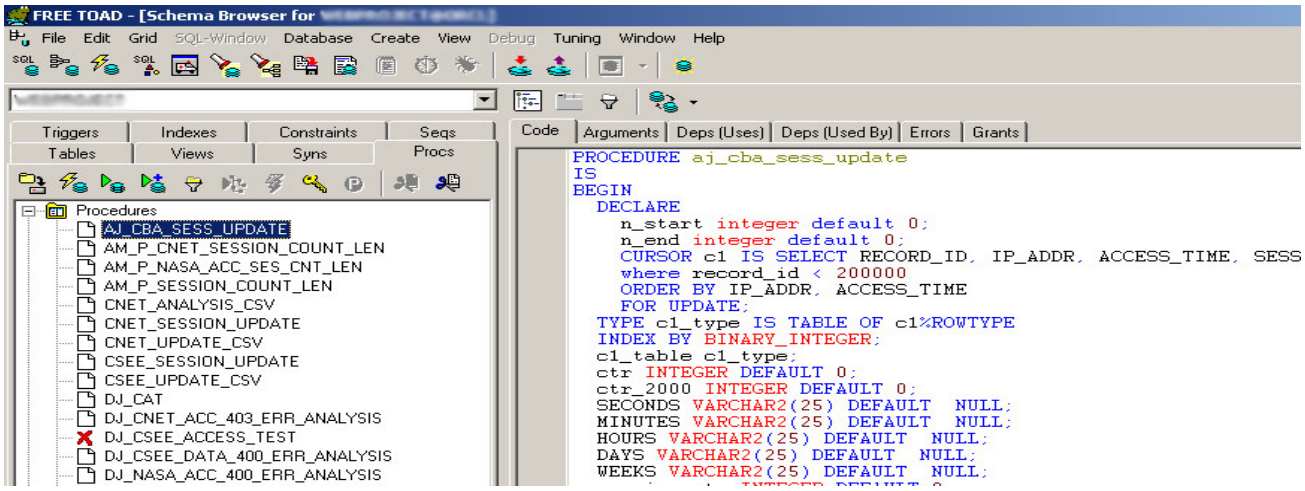

Figure 4.5: TOAD- GUI for Oracle database(procs)

PL-SQL procedures

These are primarily used to modify data and create different database tables, i.e. Data tables and Session tables. Benefit of having Stored PL-SQL procedures is manifold; it is easy to access from external applications, or languages like Java, also it is a performance booster as compared to external methods of data manipulations inside a database. Above all it gives us a high degree of flexibility in terms of usage and portability among different users or even machines. Some of our stored PL-SQL procedures are built to process the Data table and output another table,Session table, by measuring the IP addresses, their respective sessions(which, as explained earlier are created inside the Data table using another stored PL-SQL procedure and error counts for different 4XX and 5XX error codes .

Graphing applications

Primarily we were using Microsoft Excel for all our graphs. As our needs grew in terms of data size so did the incapacity of Excel to handle it efficiently. We used R, an open source software for plotting and managing data analysis, for its unique qualities and for its capacity to handle lager data files. Data files larger than 65465 lines of data, limit which Excel cannot exceed, can be easily handled by R.

Chapter 5

Data Analysis & Results

This chapter discusses the results and analysis of We Workload Characterization for web sessions. It also summarizes the results for robot characterization trying to distinguish well behaved robots from non robot sessions. The results are divided in two sections, the first one takes a look at the HTTP error distributions for different servers, while the second part concentrates on session characteristics. HTTP error characteristics are

The results are divided in three sections, the first one takes a look at the HTTP error distributions for different servers, while the second part concentrates on session characteristics, and the third part expands on robots characteristics we have explored.

Analysis objectives

HTTP error characteristics are studied to understand better the behavior of those servers for

- Better file management.
- Improved request-response based server performance.
- Distinguishing robots from non robot sessions.

Our focus in this work will adhere to file management and server performance issues, though some work is done in the area of robot characterization.On the other hand study of sessions is done to understand session characteristics, which can also be applied to explain the results from other parts of web server workload characterization studies, done by my colleagues.

The intra session parameters which are listed in table 4.2 are the session metrics used for our analysis. Number of requests, session length, bytes transferred and number of errors per session are the four major intra session characteristics we have analyzed. The main aim of collecting these session characteristics is to

- Explain how intra session variables behave independently.
- What intra session variables are related and how they affect each other.
- Does clustering and PCA help in determining these relationships.
- How does these intra-session variables help in categorizing robots from non-robots session.

5.1 RAW data for server session parameters

Table 5.1: Server session parameter statistics

Server	Total number of requests	Total distinct users	Total number of sessions	Total bytes transferred	Total bytes transferred per week
WVU	37,870,087	169,251	487,637	96,953,286,815	48,476,643,408
Clarknet	1,654,855	90,503	139,740	14,454,810,366	14,454,810,366
CSEE	2,509,790	37,322	100,069	210,449,778,907	105,224,889,453
NASA-Pvt1	22,623	123	921	496,614,847	24,830,742
NASA-Pvt2	92,112	158	4,544	169,610,450	84,80,522
NASA-Pvt3	489,004	328	23,907	2,297,296,733	114,864,836
NASA-Pub1	92,541	10,345	18,443	9,424,545,924	471,227,296
NASA-Pub2	731,504	17,157	57,889	6,988,408,844	349,420,442
NASA-Pub3	108,200	7,273	15,850	4,794,183,943	239,709,197

Table 5.1 gives us the information about each servers load statistics and the time period, which we have used to collect the logs. Both WVU and CSEE have logs for 2 weeks, while Clarknet has logs for one week. NASA servers did not have heavy traffic load and hence we gathered data for 20 weeks instead. WVU logs were the largest in terms of number of requests, total distinct users and, total number of sessions, strangely having a low bytes

transferred value compared to CSEE server. This may be attributed to the fact that CSEE server hosts many large downloadable files i.e. assignments in PDF, applications, and availability of personal space for each student. The frequency of these downloads is much higher than those available on WVU public server. This can be explained easily as the total distinct users are 4 times more in WVU though CSEE has almost double the data transfer in two weeks.

All these servers have recent data except Clarknet, but for a variety in the data sets and incorporate a commercial server we have used it for our analysis purposes. All the NASA server data is gathered for the same time frame, from April till August 2004, for better comparisons, while CSEE and WVU data is taken from the starting months of 2004.

5.2 Correlation coefficient analysis of intra-session parameters

Correlation coefficient between two parameters shows how they are linearly defined. If the relation between two variables is non-linear, correlation coefficient might not be an answer to associate different variables. Figure 5.1 and 5.2 shows the correlation coefficients of intra session variables, one having Error counts and the other without it. The x axis shows the pair of variables with y axis showing the correlation coefficient value between them. Following are the definition of the acronyms :

RQPS, Requests per session

SL, Session length in seconds

BT, Bytes transferred and

EPS, Errors per session

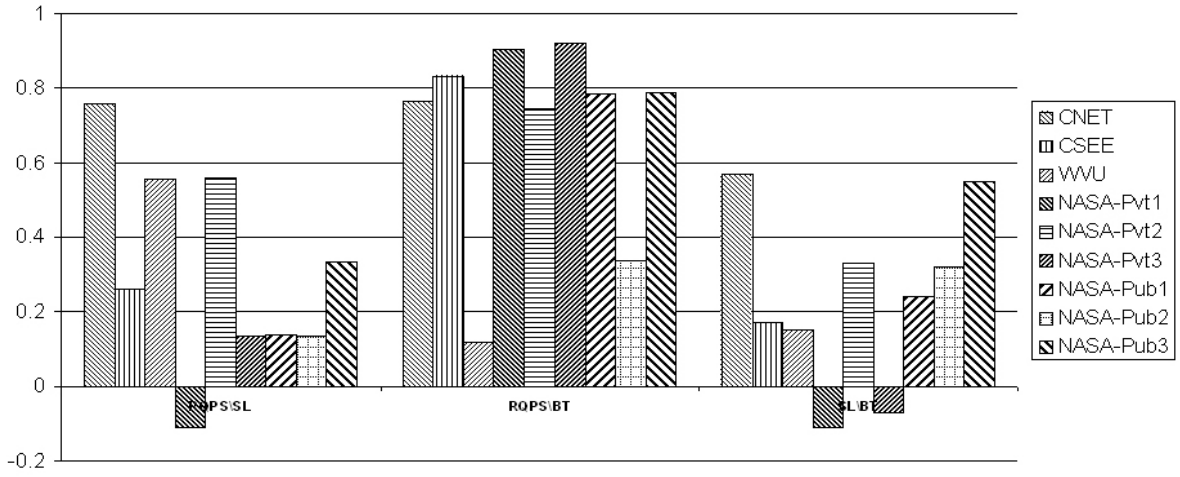

Figure 5.1: Correlation coefficient between intra-session variables without error count

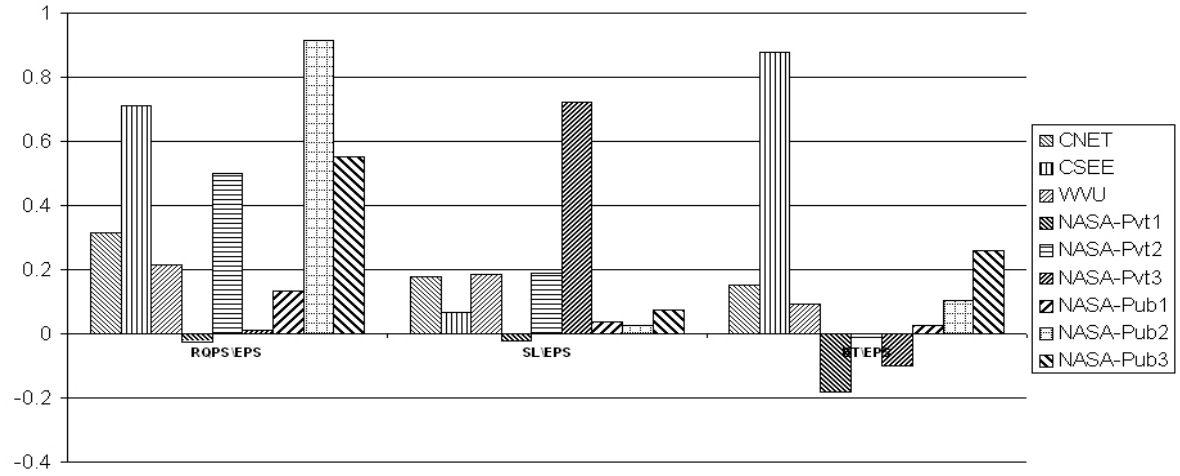

Figure 5.2: Correlation coefficient between intra-session variables with error count

As we can see from Figure 5.1 that Requests per Session and Bytes Transferred are highly correlated to each other in almost all the servers, specially in NASA Private servers and NASA-Pub1 and NASA-Pub3 servers. Another general trend we see is that variables are less correlated with the Number of errors parameter compared to other parameters, though CSEE and NASA-Pub2 server shows a high affinity between Requests per Session and Errors per Session, while in case of Session Length and Error relationship only NASA-Pvt3 server

shows a high correlation. In case of CSEE server Bytes transferred and Errors per Session are highly correlated.

As seen earlier in PCA plots, majority of the servers have Requests per Session and Bytes Transferred per Session as the 2 dominant variables explaining almost all of the data variance. This is in accordance to the fact that correlation between these two variables is high among majority of the servers. This suggests that either of the two variables can predict the behavior of the data set to a great extent.

Overall if we see, with respect to the servers, Clarknet is highly correlated among all parameters except Errors per Session. Similarly if we see figure 5.2, in case of NASA-Pvt1 server Error count is not positively correlated to any of the variables.

5.3 Clustering of sessions with multivariate data

We have used the standard K-Means clustering technique to process data into sets of clusters, remembering that there has been no data normalization technique applied yet, to the data. It is an attempt to classify data through this process and see how the data behaves. We have used this behavior to compare it with the manual inspection of data we did earlier, like how the robots in a particular data set are extracted.

5.3.1 Cluster distribution function

Lets start with an example of clustering plots of Clarknet server for a varying cluster size k. This is provided just as an example of how a clustering plot looks like. Figure 5.3 shows an actual plot of Clarknet server, where clusters for different k size i.e. 5,10,15,20.

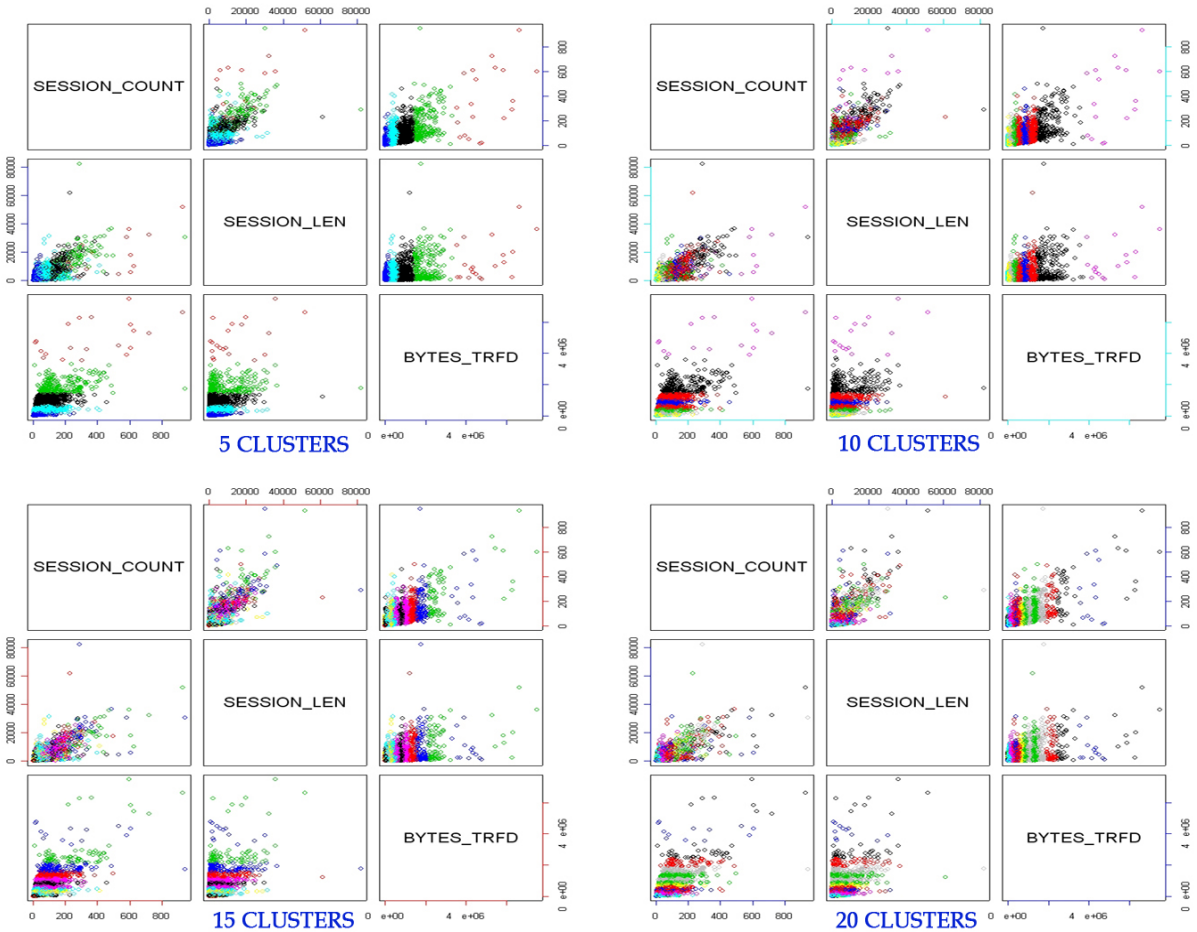

Figure 5.3: K-means Clustering example for 5,10,15 and 20 clusters

Notice how the complexity of clusters increase as we increase the value of k. We can also see that in contrary to trained data set clusters, which are almost always well separated, these clusters have overlapping data points. Its apparent that more the dimensions, more complex the clusters are. Some parameters such as *Bytes Transferred* in this case, has an upper hand in determining the structure of the cluster, even when the values of k change.

5.3.2 Cluster verification and quality estimation

The clustering has been done with following standards:

- Number of clusters are 5
- Number of iteration done are 15

- Four variables listed below are used as the primary clustering data set properties.
- There is no manipulation in the scale of the variables in observation.

The Cluster validity index provided by Jain and Dubes [33], has been used to estimate the cluster composition. Menasc et al[10] proposed means to explore the quality of clustering process, which helps in determining how many clusters i.e. k we should select for our analysis.

The distribution of validity ratios and coefficients for different cluster sizes are plotted for all servers, shown below. This helps us in understanding the number of cluster selection for "almost" optimal k-means clustering exercise. We try to see the variation among the four ratios i.e. Coefficient of variation of intra-cluster distance, Coefficient of variation of inter-cluster distance, β_{cv} , β_{var}, to understand how they behave when number of cluster is varied.

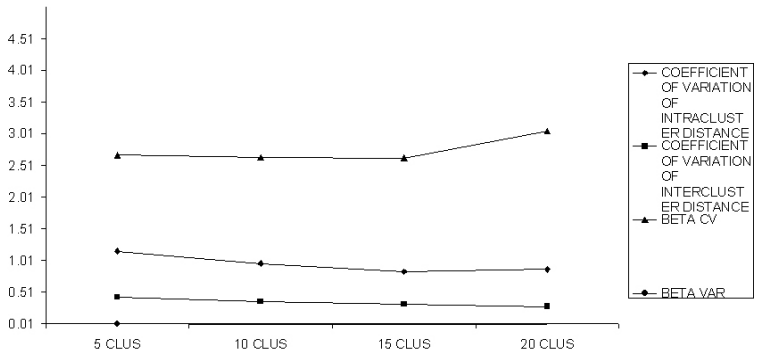

Figure 5.4: Clarknet cluster validity ratios

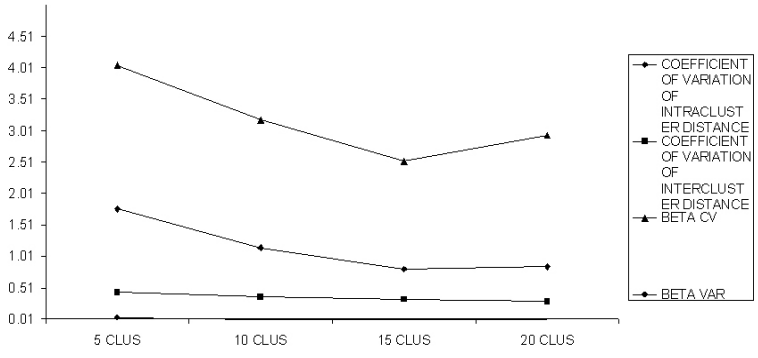

Figure 5.5: CSEE cluster validity ratios

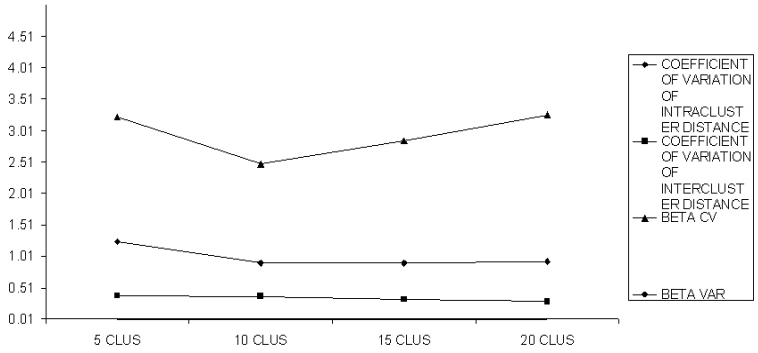

Figure 5.6: WVU cluster validity ratios

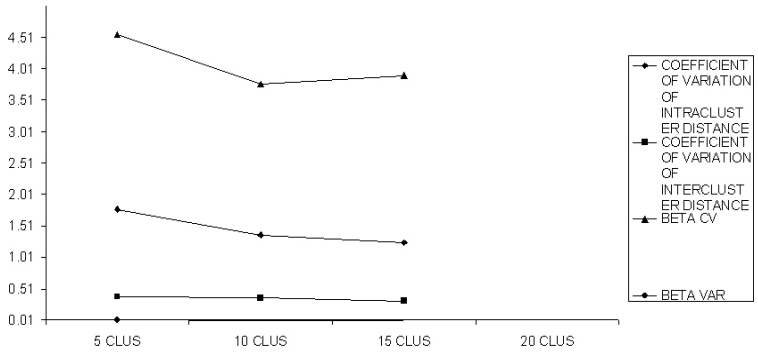

Figure 5.7: NASA-Pub1 cluster validity ratios

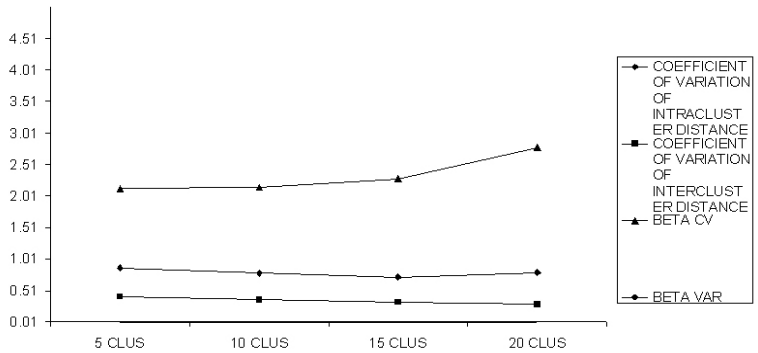

Figure 5.8: NASA-Pvt1 cluster validity ratios

After looking at figures 5.4 to 5.8, which show the distribution of ratios for all 9 servers, we can say that in almost all servers, the ratios decrease convincingly when cluster size is

increased from 5 to 10 and then to 15. Based on that, we can say that cluster size 10 or or near to 10 yields almost optimal results.

We also compared this result with results obtained by Menasce et al[10], refer figure 3.5. The difference in our result is that the value of C_{inter} remains constant while C_{intra} decreases with an increasing value of k. We also observed that in case of NASA servers, which have less number of data points, show a better result at lower k value than higher k values. This is normal to expect as in case of smaller data sets, a high value of clusters means that the data has been partitioned forcefully, leaving the properties such as inter-cluster variance and intra-cluster variance behave differently.

When comparing the values of β_{cv} with figure 3.6 along different servers it maintains the behavior and is always almost constant at a given value, though a decreasing trend is shown near a k value of 15. Another interesting fact we notice here is that in all the servers the value of β_{cv} decrease minimally till the k value is 15, and then there is a small increase in the value. This is most promising result regarding selection of the value of k for clustering.

Validity index

Next we discuss about the cluster validity index, which is already explained in chapter 3. The validity index is used to gauge the quality of clusters being formed. Lets look at figures 5.9 and 5.10, which shows the plot of validity index for varying k values.

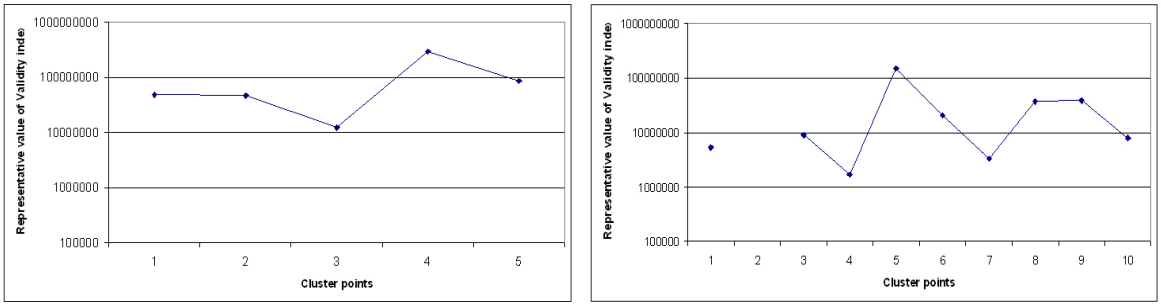

Figure 5.9: Validity index plot for 5 and 10 clusters for CSEE

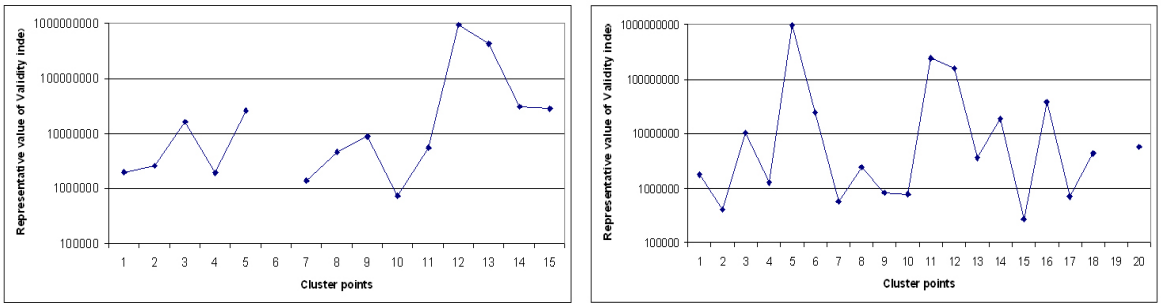

Figure 5.10: Validity index plot for 15 and 20 clusters for CSEE

We observed that for a increasing k value the validity index decreases for almost all the clusters except some. As we can see that the number of clusters in high validity value zone decrease as we increase the k value. We also observed that as we increase the k value from 15 to 20 the validity index value increase for some of the clusters. This behavior was seen in almost all the servers. As discussed later, with certain confidence we can say that a k value of 10 or 15 is better suited for our analysis than either 5 or 10.

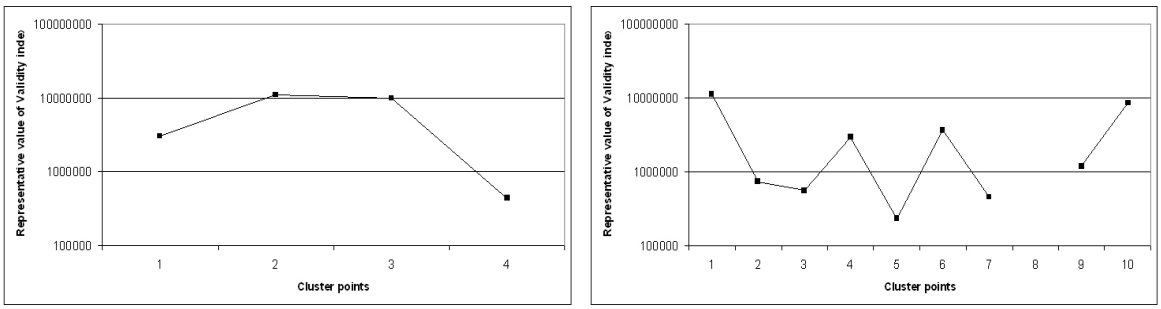

Figure 5.11: Validity index plot for 5 and 10 clusters for NASA-Pub2

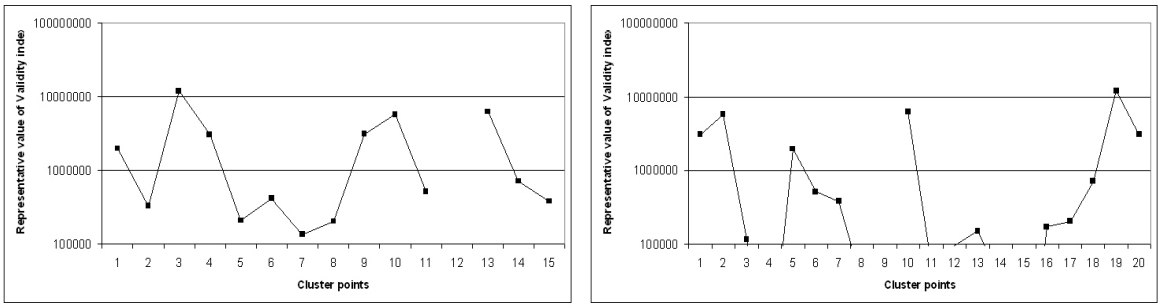

Figure 5.12: Validity index plot for 15 and 20 clusters for NASA-Pub2

As we can see from figures 5.11 and 5.12, index values for most of the clusters decrease as the value of k is increased. We also observe that when the value of k is increased from 15 to 20 the index value disperse, though the decreasing trend of index values for majority of the clusters is not disturbed.

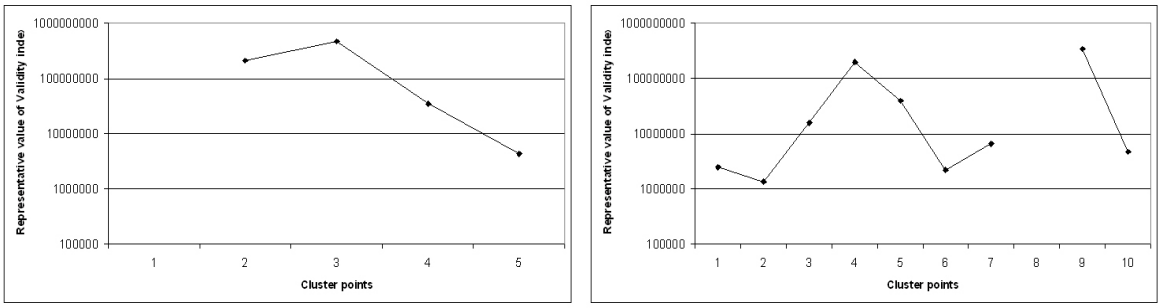

Figure 5.13: Validity index plot for 5 and 10 clusters for WVU

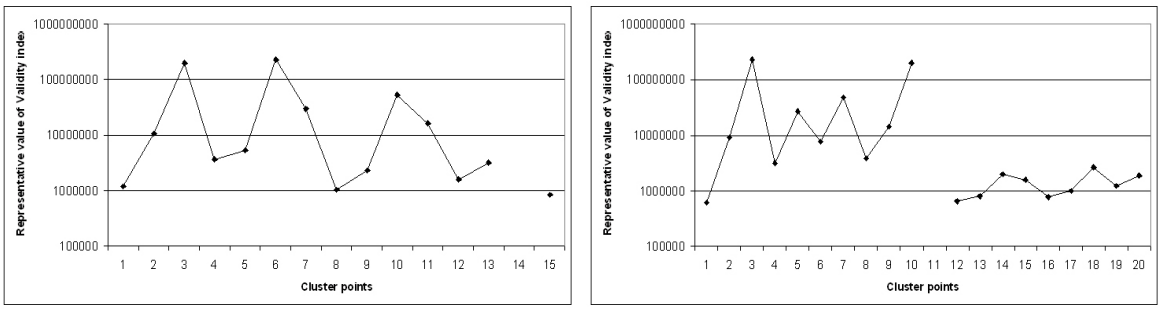

Figure 5.14: Validity index plot for 15 and 20 clusters for WVU

Cluster centroid values

Now lets take a look at the cluster centroid values for CSEE server. The table 5.2 and 5.3 gives.

Session Count	Session Length	Bytes Transferred	Percentage of points
22	458	46,725	86.686
213	7,359	364,078,847	0.004
232	4,891	8,031,693	0.424
125	1,796	703,533	9.046
101	2,779	114,068,921	0.021
332	4,856	15,498,150	0.181
165	2,584	2,014,819	2.475
163	2,044	28,311,846	0.129
194	3,057	45,327,385	0.103
194	3,770	4,603,673	0.926

Table 5.2: Centroid values for CSEE server for 10 clusters

Session Count	Session Length	Bytes Transferred	Percentage of points
184	2587	1434461	2.255
149	2506	2385259	1.138
247	3803	14605326	0.151
15	371	22380	77.242
194	3057	45327385	0.103
213	7359	364078847	0.004
122	1826	753694	5.111
198	3605	3931849	0.640
247	4442	8959424	0.233
83	1253	292423	12.383
177	4327	5972132	0.554
72	4136	151948432	0.008
120	1875	88815915	0.012
183	2363	31373295	0.077
362	4718	22335368	0.082

Table 5.3: Centroid values for CSEE server for 15 clusters

We have utilized the concept of Probability Distribution Function to find out the distribution of variables in different clusters. We also try to compare how this distribution changes when we increase the cluster k size from 10 to 15. The criteria behind selecting only these two variations was due to the previous finding about the quality of clusters. We found that clusters with k values of 10-15 have better representation of the whole data set compared to other k values.

When plotting the distribution of clusters for NASA Pub2 server for 10 cluster size, we found that cluster 4 has some sessions with large Number of Requests but the small session length, with high amount of bytes transferred. For cluster number 5 we again found that Number of requests and Bytes transferred are less while the session length is really high.

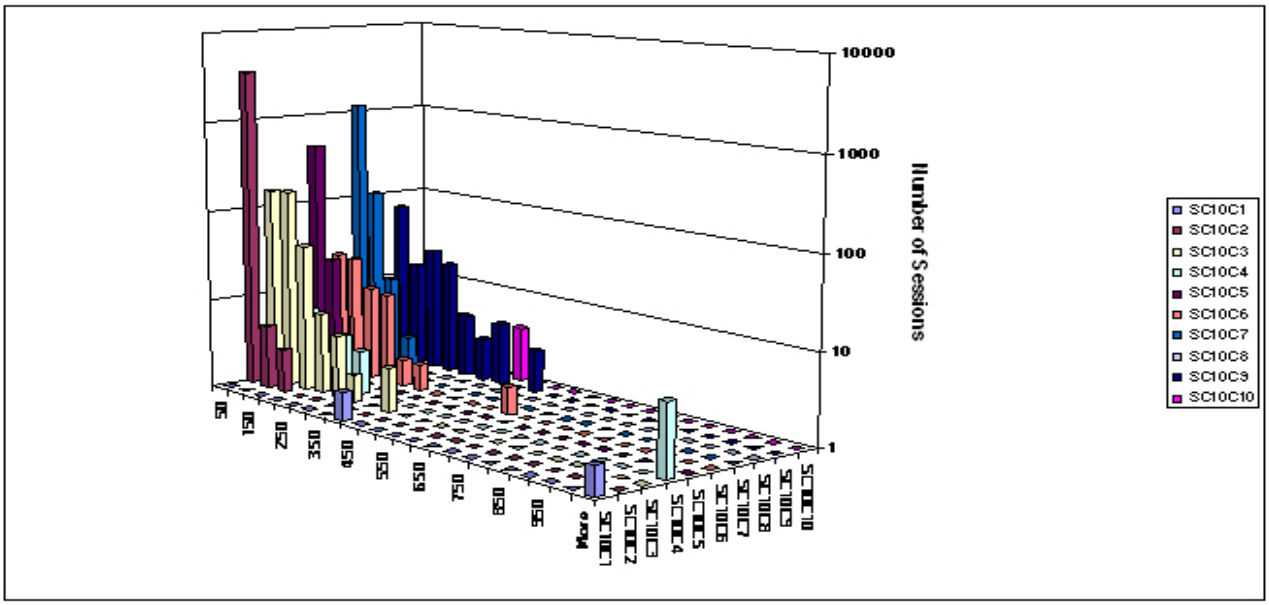

Figure 5.15: Nasa-Pub2 - Session distribution with respect Session Count(SC) for 10 Clusters

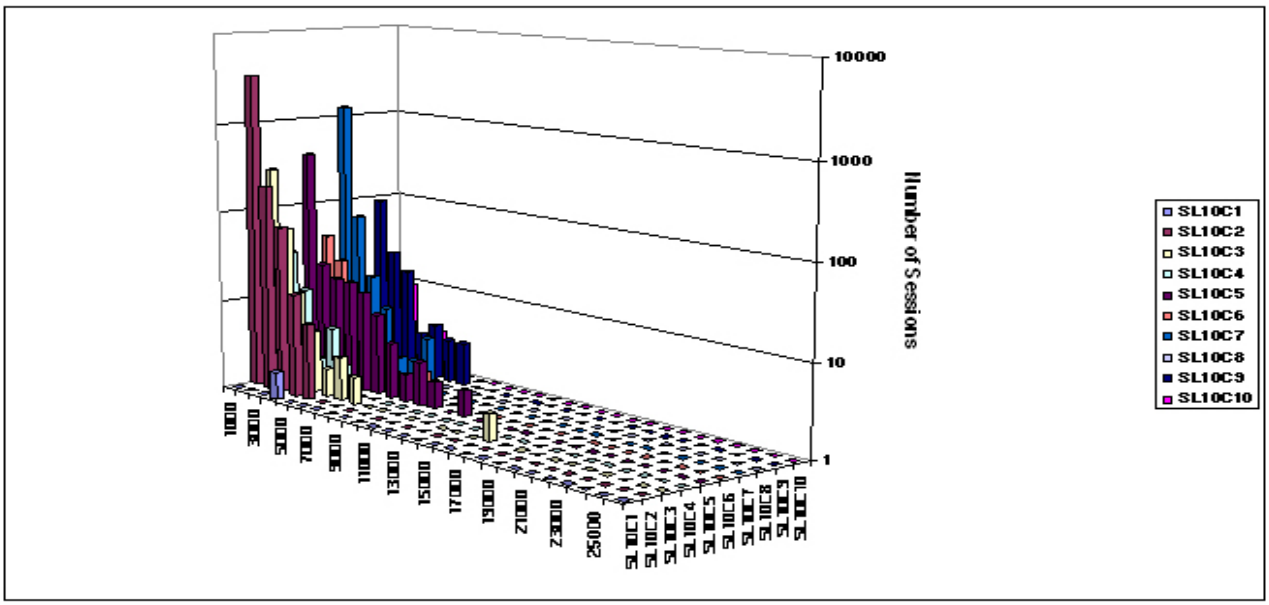

Figure 5.16: Nasa-Pub2 - Session distribution with respect to Session Length(SL) for 10 Clusters

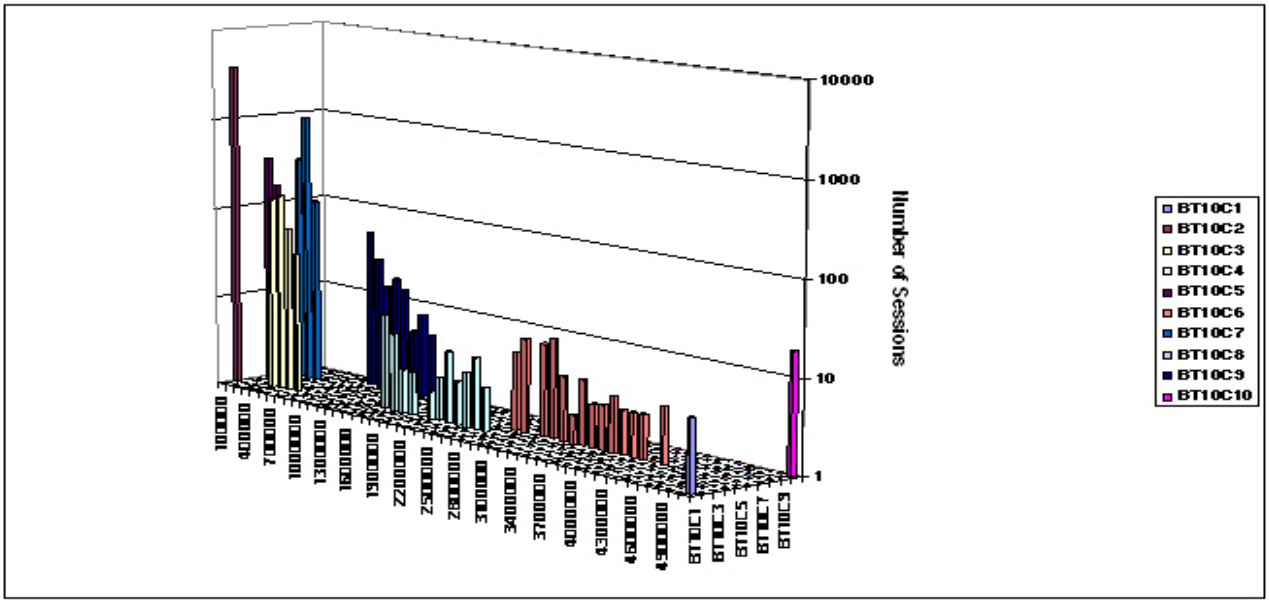

Figure 5.17: Nasa-Pub2 - Session distribution with respect to Bytes Transferred(BT) for 10 Clusters

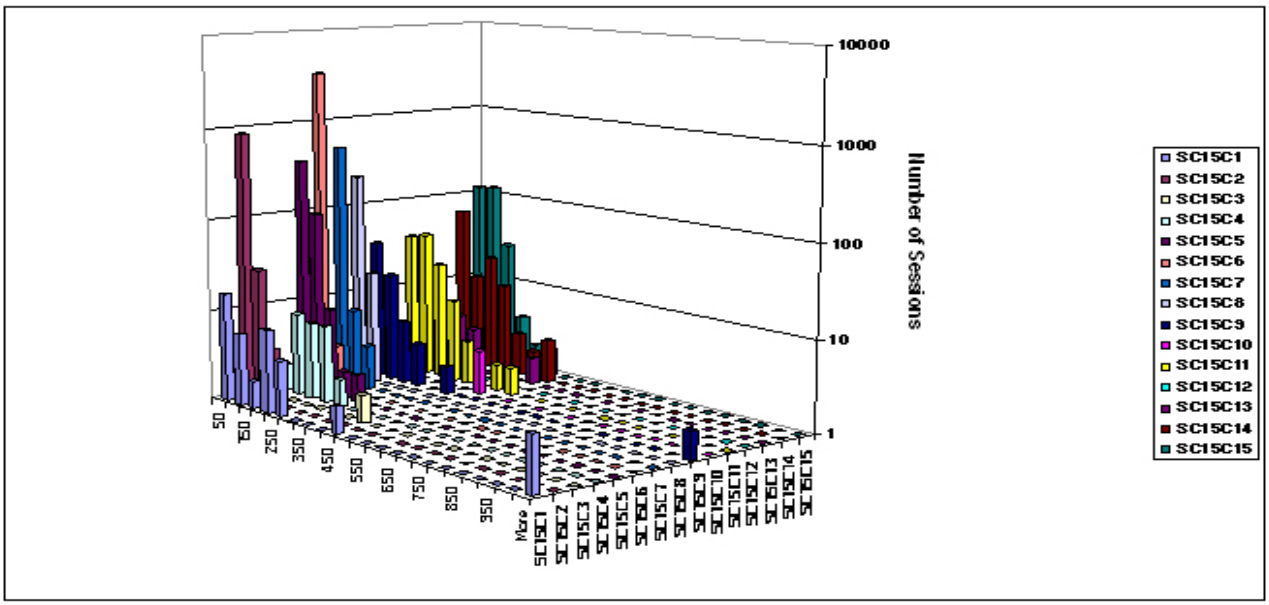

Figure 5.18: Nasa-Pub2 - Session distribution with respect Session Count(SC) for 15 Clusters

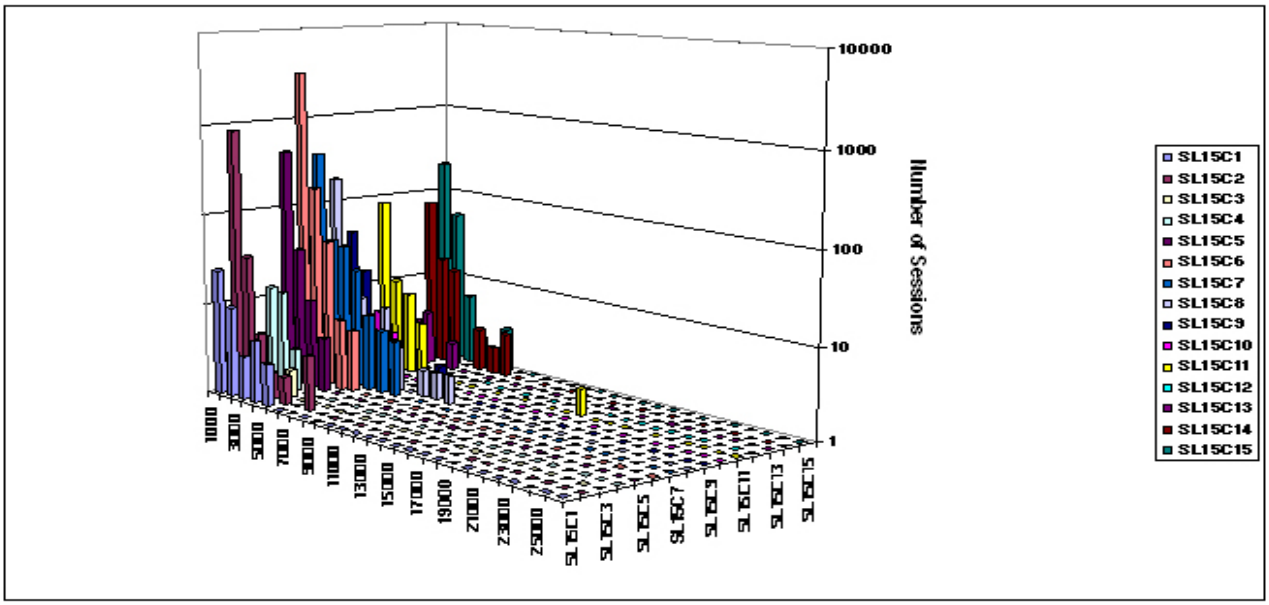

Figure 5.19: Nasa-Pub2 - Session distribution with respect to Session Length(SL) for 15 Clusters

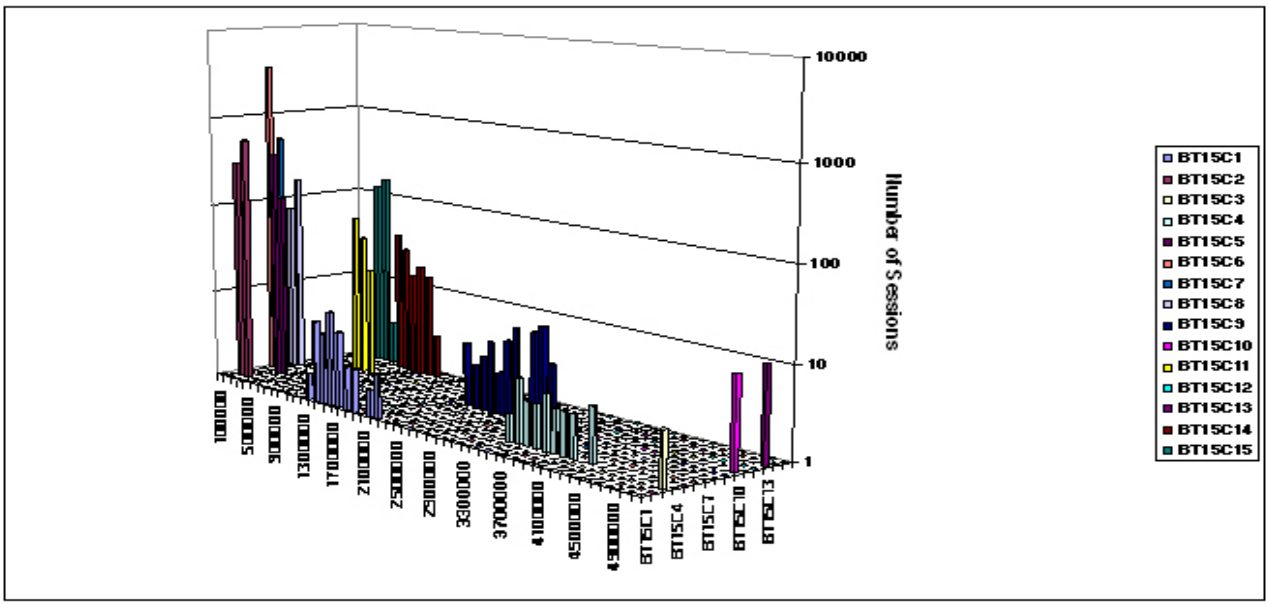

Figure 5.20: Nasa-Pub2 - Session distribution with respect to Bytes Transferred(BT) for 15 Clusters

We then plotted the distribution of clusters for a cluster size of 15. In cluster number 4, we found that sessions have less number of requests and a small session length but the bytes transferred are really high. While in cluster number 11 its quite the opposite, where the number of bytes transferred is less with higher values for the other two parameters. Cluster number 1 has small session lengths with larger number of requests and bytes transferred.

In case of CSEE, figures 5.21 and 5.22 show the distribution of clusters for 10 and 15 k value. We observed couple of sessions with hight amount of bytes transfer but has smaller session count and session length values. There was no clear pattern which suggested that all the clusters are small and short with a high data transfer but almost 5-6 clusters out of 15 have similar behavior, plots for WVU also suggest the same behavior.

One important observation we made was that the relationship between session count and session length is directly proportional for most of the clusters, whether it be a k value of 10 or 15.

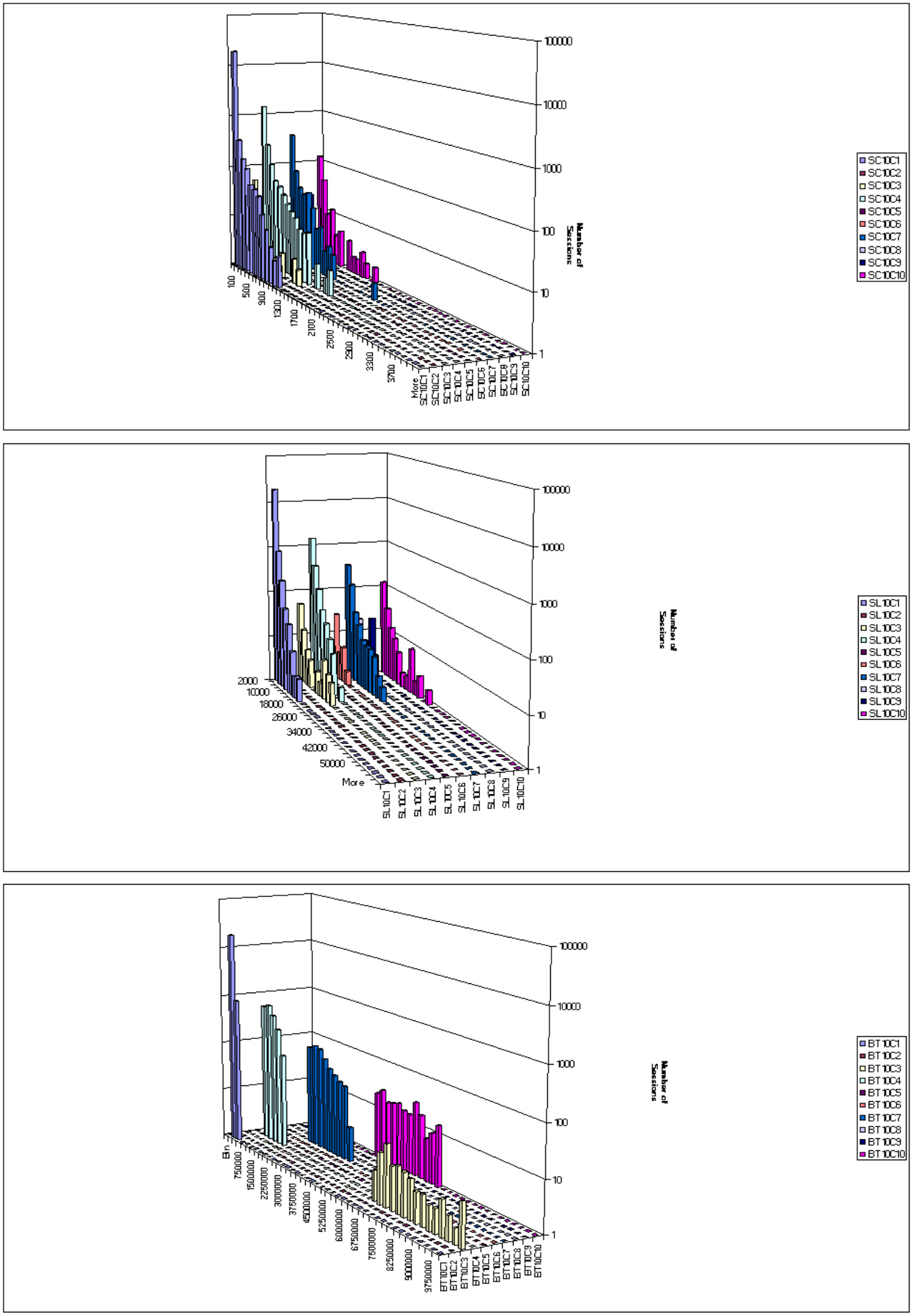

Figure 5.21: CSEE - Session distribution with respect to three variables, Session Count(SC), Session Length(SL), and, Bytes Transferred(BT) for 10 Clusters

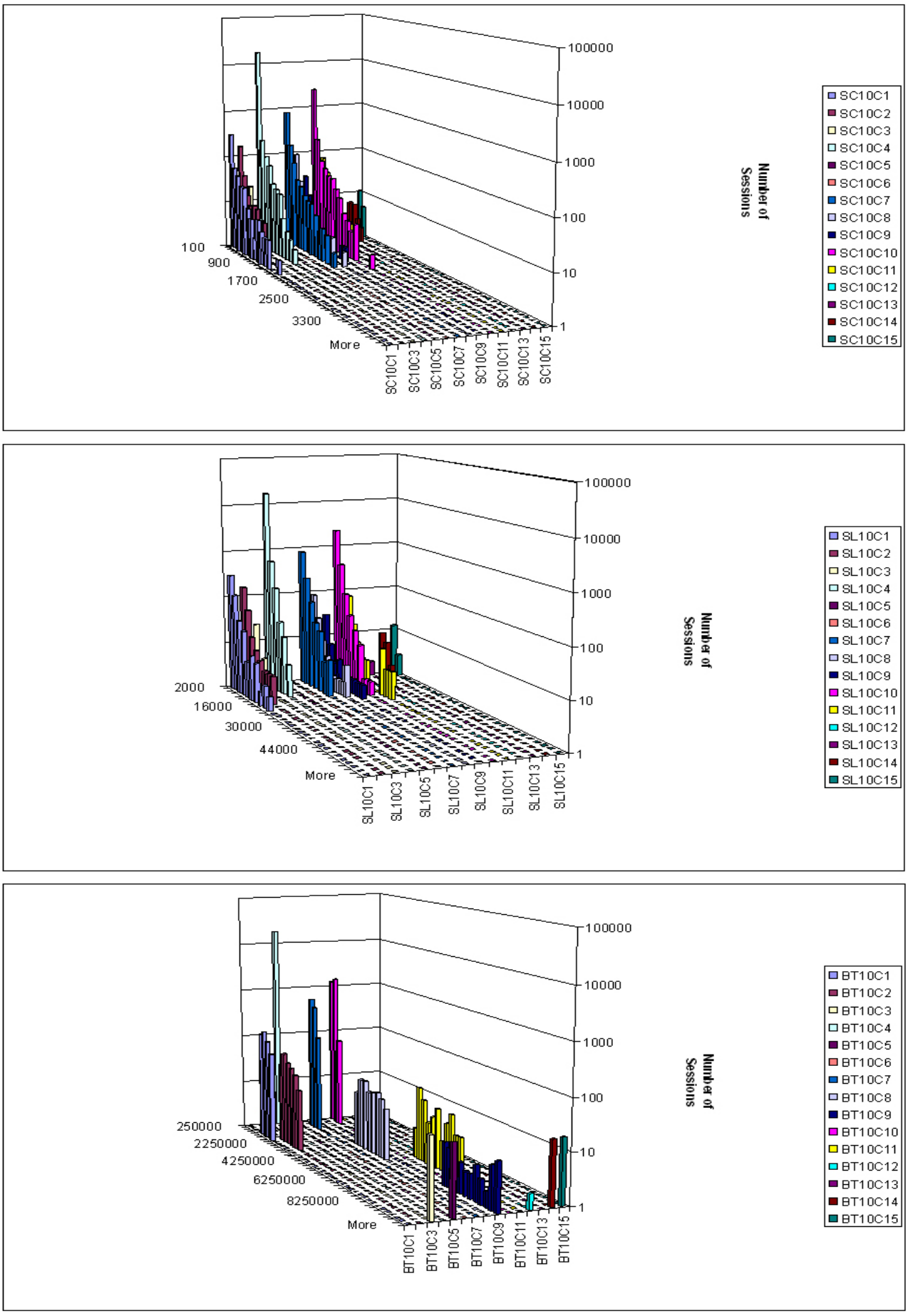

Figure 5.22: CSEE - Session distribution with respect to three variables, Session Count(SC), Session Length(SL), and, Bytes Transferred(BT) for 15 Clusters

5.3.3 Range distribution of different cluster size for Raw data clusters

We plotted the ranges of clusters done by *K-means* with 4 different cluster sizes, 5, 10, 15, and, 20. Figure 5.23 and 5.24 shows the ranges of each cluster in Clarknet, CSEE, and, WVU server, each for different cluster sizes.

- Number of requests parameter is interesting to observe as maximum value of it remains constant even if the cluster size is increased from 5 to 20.
- One cluster has distinctive small range. Interestingly enough the change in number of clusters from 5 to 15 or even changing across the three parameters it seems to hold the property of least range cluster.
- Another interesting observation is that as the number of clusters grows from 15 to 20 almost all the ranges across the three variables seem to have same minimum value. We assume that this can be due to less number of data points in the server data set. The low number of data points is forcing the K-means algorithm to break data points.
- We can also see that almost all servers have one cluster with not more than 5 data points in case of 5 clusters. Increasing the number of clusters doesn't change this phenomenon, there is an increase in clusters with number of data points less than 5.

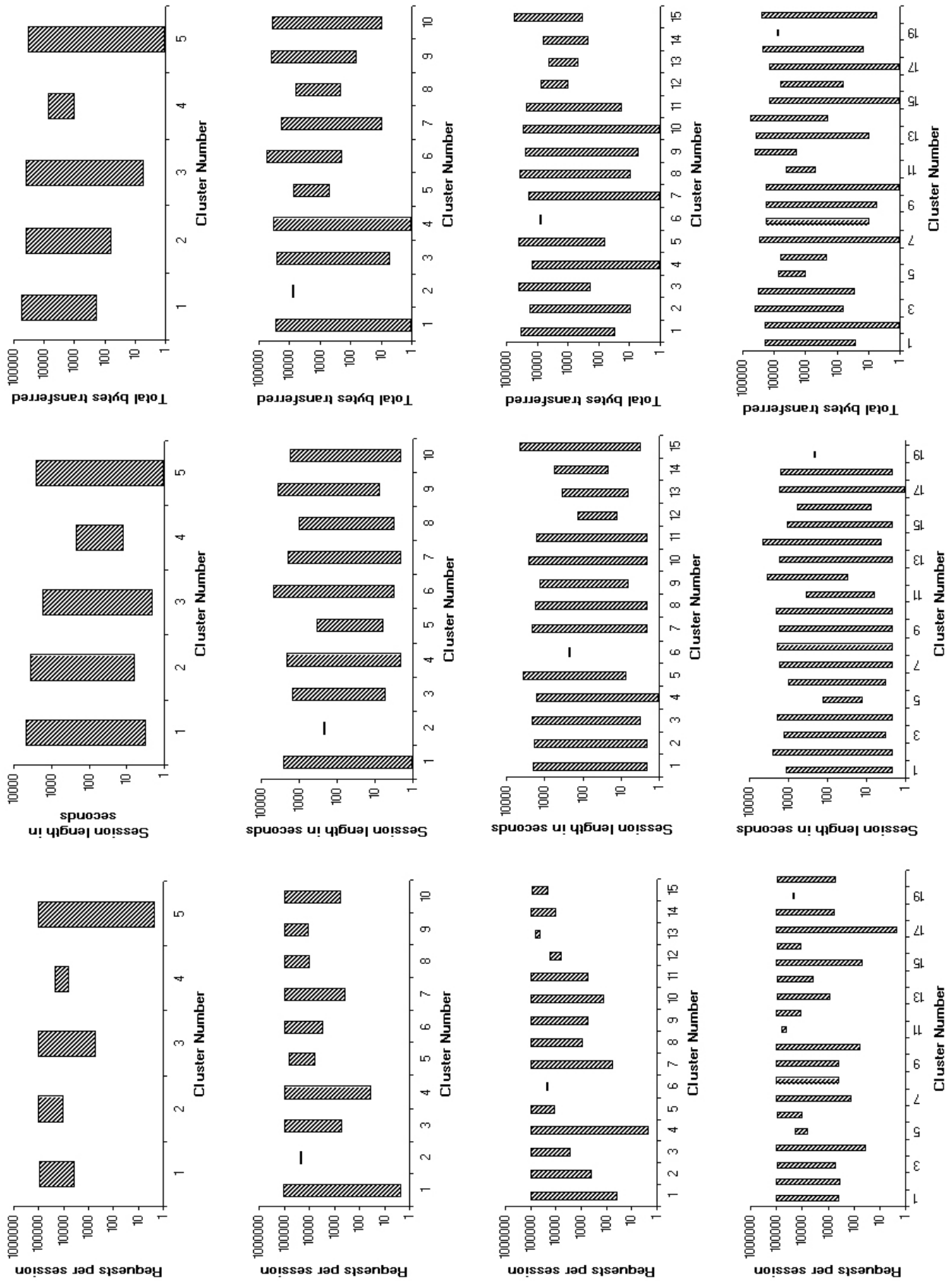

Figure 5.23: Boxplot of ranges of clusters for 5, 10, 15 and, 20 clusters : NASA Pub2

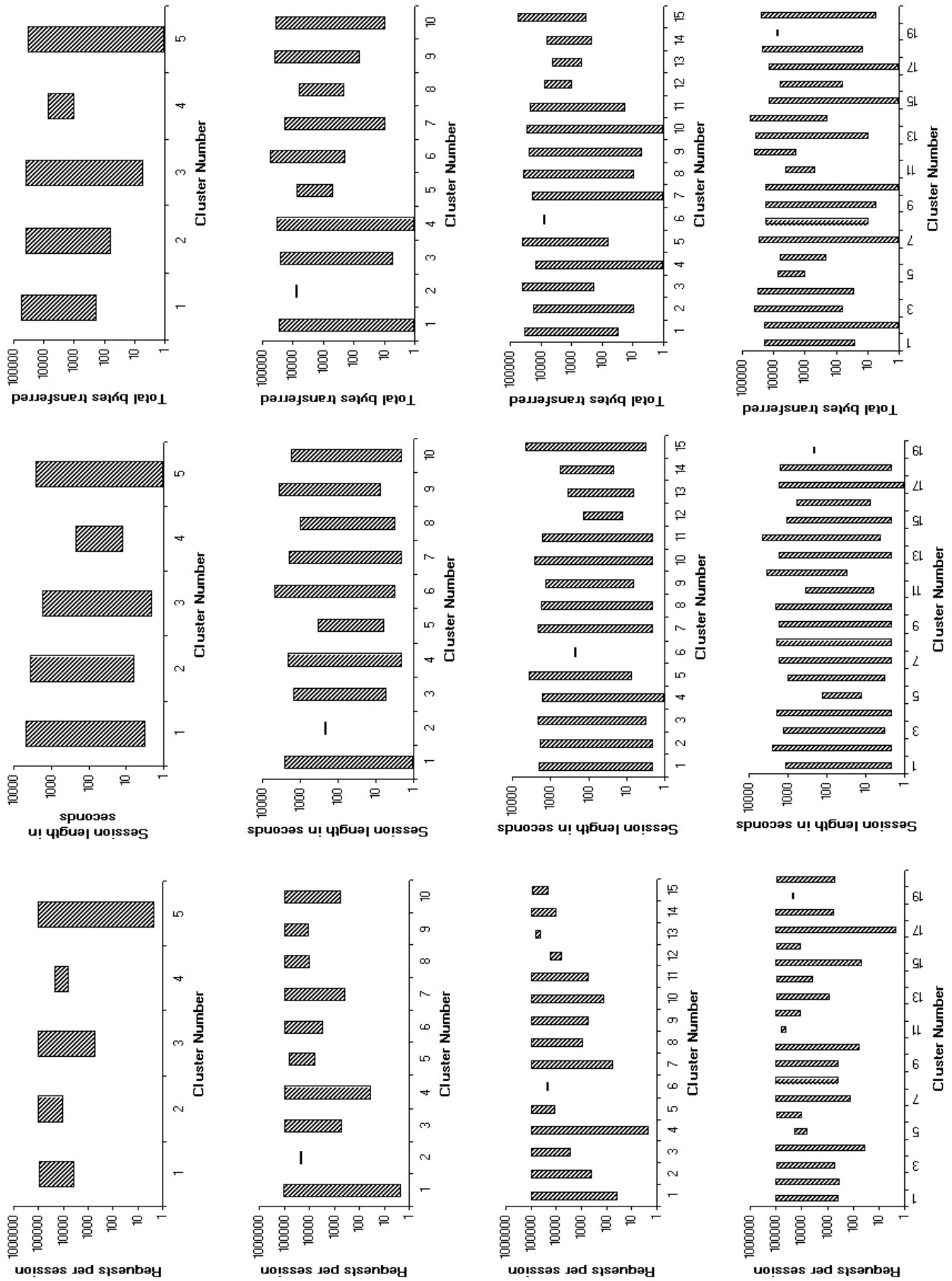

Figure 5.24: Boxplot of ranges of clusters for 5, 10, 15 and, 20 clusters : CSEE

If we look at the variable Request per session, we can see that the upper bound for all the ranges remains pretty much the same, while this is not true in case of other two variables i.e. session length in seconds and total bytes transferred. The reason behind this can be due to the fact that :

- Either the number of high values are more than number of low values in Clarknet for requests per session.
- It can also be true that requests per session might play a lesser role in deciding the cluster distribution compared to other two variables.

If we look at the second graph in Figure **??**, we find that requests per session fits perfectly for a principal component and hence explains most of the variation of the Clarknet data, thus refuting our second hypothesis.

5.3.4 Clustering the raw data

Figures 5.25, and 5.26 show the clustering of all four parameters for CSEE, and WVU server. Looking at these figures, we can see that total bytes transferred regulates the clusters to align along it. The cluster demarkation changes as the value of bytes transferred increases. We also observed one interesting phenomenon, at higher bytes transfer values, total error counts are small. Its only those sessions with smaller bytes transfer values, that the error counts are high. This might be the result of

a. All the error containing sessions are small sessions as they end abruptly after the error message.

b. All those sessions with high bytes transfer values does not have a high percentage of secured pages. This can be further verified if the distribution of type of error is plotted against those sessions with higher bytes transfer values and low error counts or vice-versa.

We strongly believe that this alignment of clusters with respect to the values of bytes transferred is due to the fact that the scale of bytes transferred is approximately 10^3 times larger than the other two parameters i.e. *Requests per session* and *session length*. This suggests that normalization should have been done before clustering for a better cluster formation.

CSEE Servers

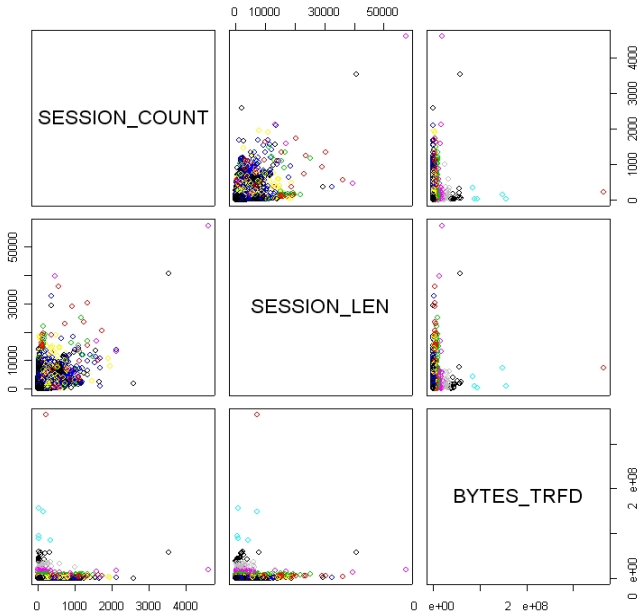

Figure 5.25: CSEE : Session clustering with raw data

WVU WEB SERVICES Servers

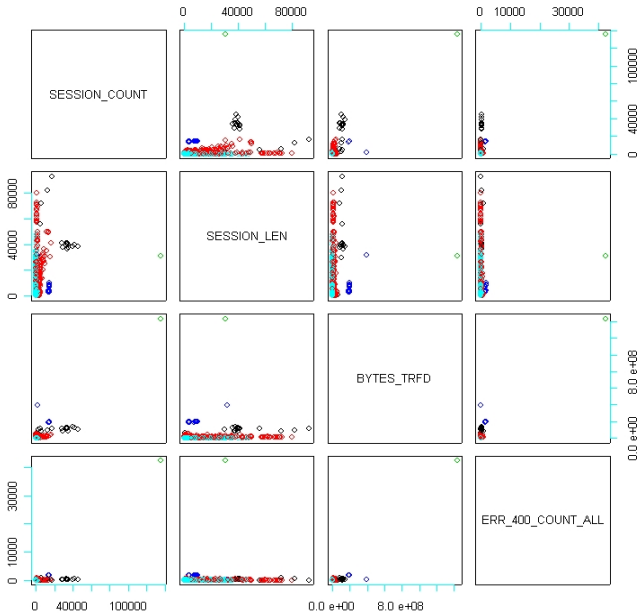

Figure 5.26: WVU : Session clustering with raw data

NASA Server Following are the clustering figures done on all the four variables together, on the raw data (without normalizing it). We can see that there is one data point which is exceptionally coming out as a single cluster. The statistics for that session is provided below

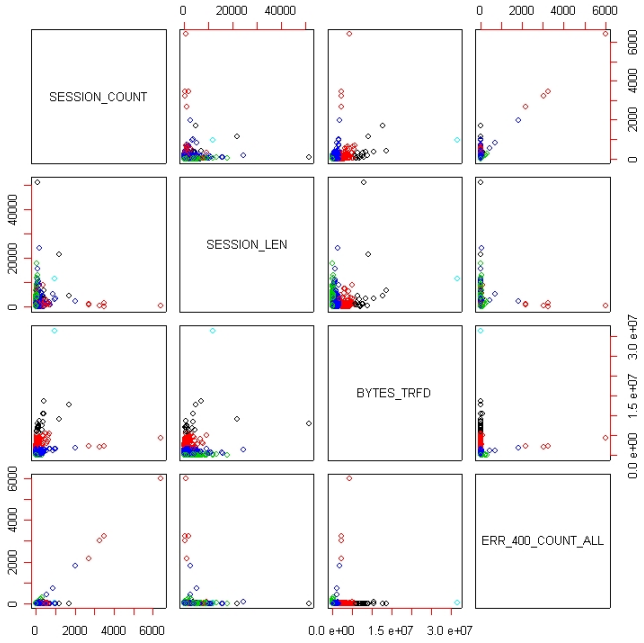

Figure 5.27: NASA-Pub2 : Session clustering with raw data

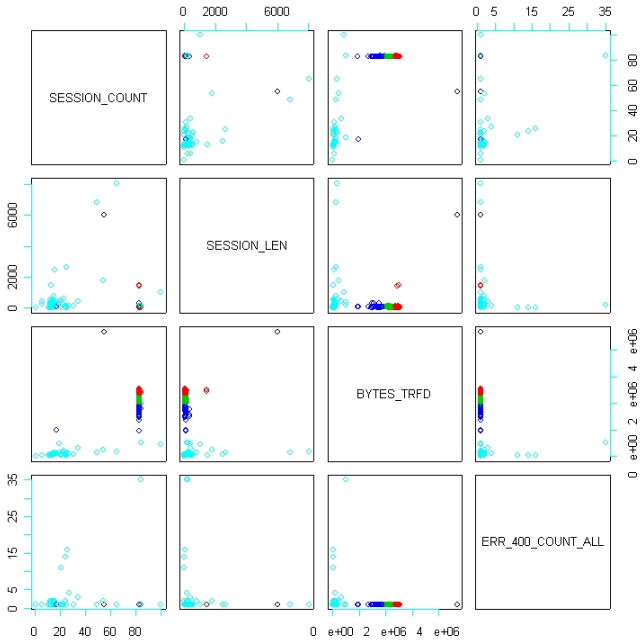

Figure 5.28: NASA-Pvt1 : Session clustering with raw data

The same analysis has been done on 3 parameters i.e. Session count, Session length, and Bytes transferred (leaving all the Error counts) and plotted in a 3 dimensional scatter-plot. This study also tries to vary the cluster numbers for the same data sets. We have used clustering with 5, 10, 15, and 20 clusters with 30 iterations in each of them. The number of iterations is important if the value set for it is very low, it might bring a difference in the results. A value of 30 ensures that there is no more chance of improvement in the clustering algorithm for a given data set.

Following are the data results for all the four servers , NASA, CSEE, CNET & WVU.

CSEE 3 Parameter 3-d plot

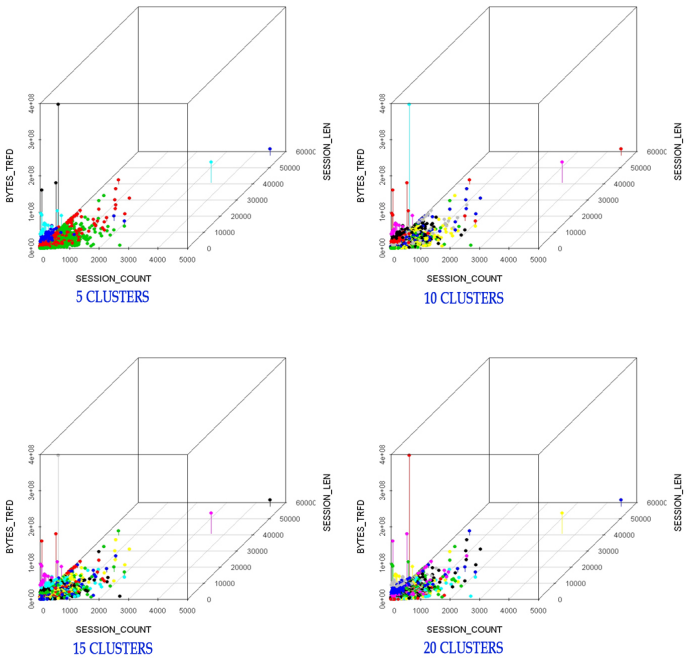

Figure 5.29: CSEE : Session clustering

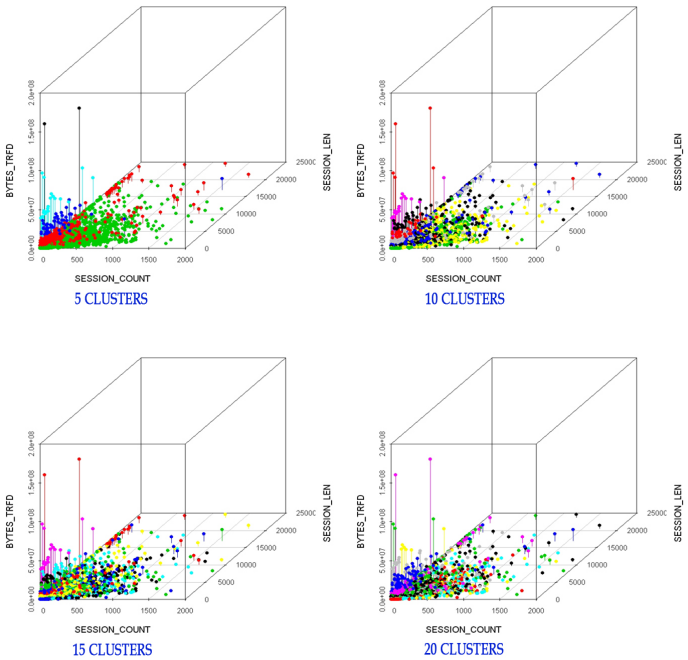

Figure 5.30: CSEE : Session clustering 250% expanded

WVU 3 Parameter 3-d plot

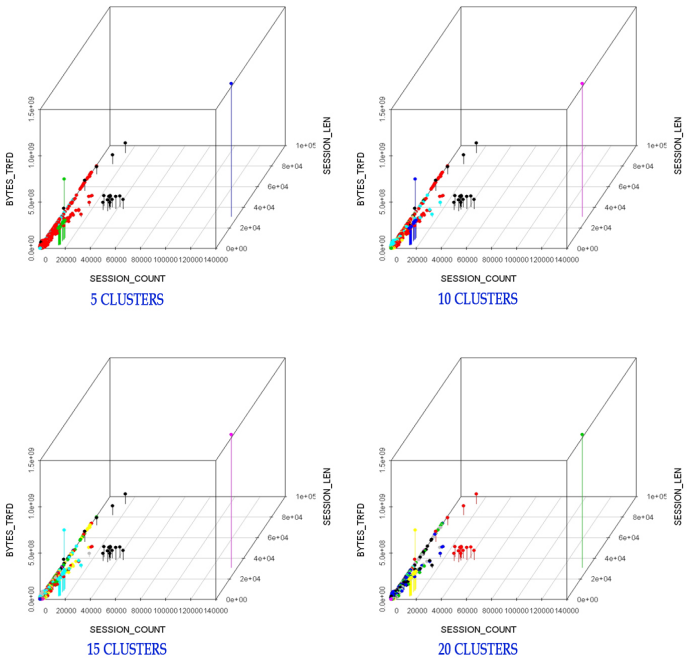

Figure 5.31: WVU : Session clustering

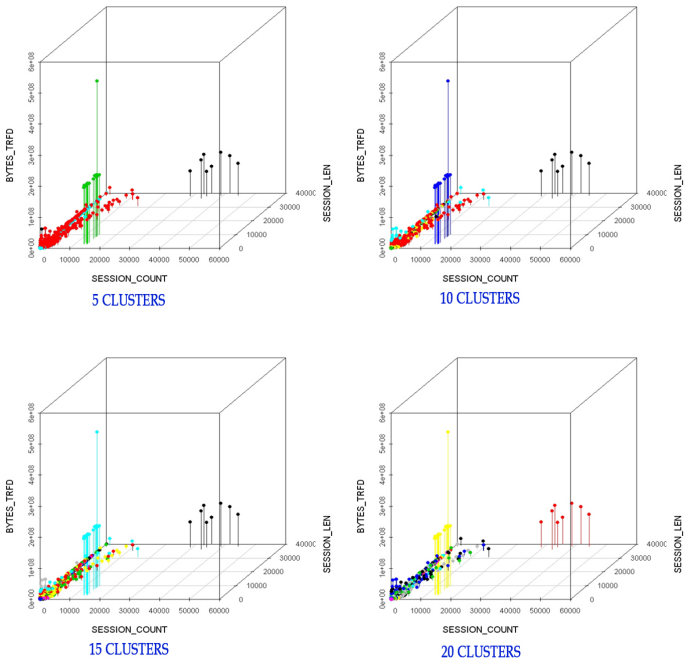

Figure 5.32: WVU : Session clustering 250% expanded

NASA 3 Parameter 3-d plot

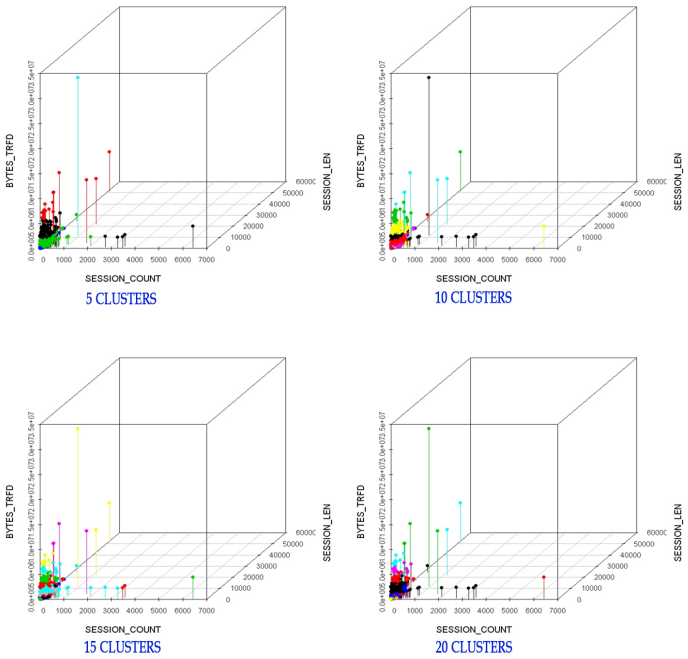

Figure 5.33: NASA-Pub2 : Session clustering

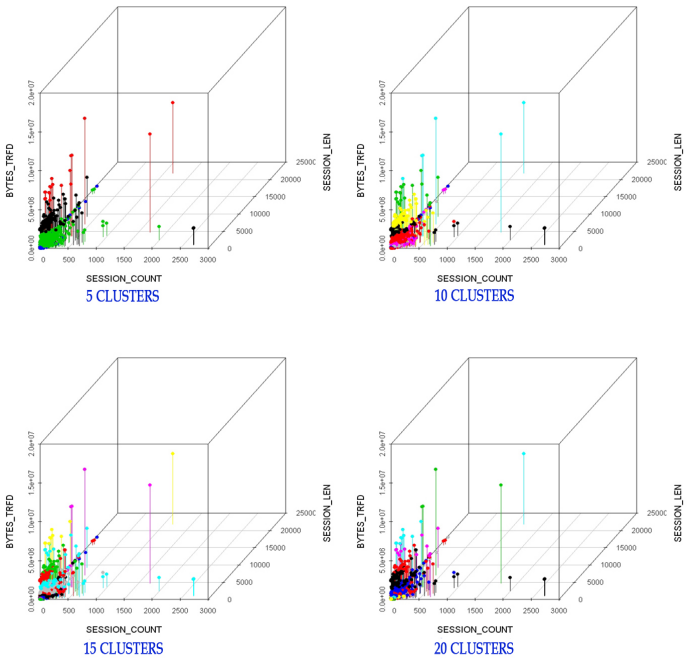

Figure 5.34: NASA-Pub2 : Session clustering 250% expanded

5.4 Principal component analysis of sessions

This section discusses the results we have from calculating the principal component data for the sessions we created. We also try to cluster the Principal Component Analysis data to find more about the effectiveness of Principal Component Analysis over raw data clustering. We have plotted principal components with respect to all the four variables and tried to find the "inter" and "intra" relationship of those variables. Lets start by first explaining what Principal Component Analysis is and how it is applied on our data sets.

5.4.1 Principal component analysis for data normalization

Data sets with many variables have often pair or more than a pair of variables which govern the same behavior of the whole data set. This induces the unnecessary redundancy of parameters which can represent similar variation of the data set. With more than one such parameter in our data set, we can take the advantage of PCA's ability to reduce the dimensionality and drain out unnecessary variables to better represent the data set behavior. We have used four parameters for our analysis which does not require necessarily a dimension reduction but figuring out couple of redundant behavior is important when we start clustering our data to overcome the resource and time overload on the clustering process. This analysis is the first step in our study where we pair different combinations of intra-session parameters and try to analyze :

1. how these parameters behave independently.
2. what is the relationship between these parameters if there is any existing.
3. how their behavior change with different server data sets.

Plot of principal components

Lets take a look at the NASA Public servers first. Figures 5.35,5.36 show the PCA analysis of NASA-Pub2 . The axis are principal components, and the parameter vectors are drawn to give a visual representation of their relationship with each other and also the principal components. Notice that each figure uses only the first two principal components as their axis. We direct our program[32] to utilize only the first two PC as they account for almost 95% of the variation of the data set, in almost all the server data sets.

The major points noted are:

1. Number of Requests per Session and Bytes Transferred in NASA-Pub1, NASA-Pub3, NASA-Pvt1, Clarknet and WVU behave almost identical, and they both contribute towards the maximum variation of data. NASAPvt3 behaves similar to NASA-Pub1 but one distinct feature observed in this server was that Session length and Total Error Count control the second maximum variation of data set along the second Principal Component. So its quite obvious that retaining any one of the variables, ones whose vector are more aligned to the Principal Component vectors, is a better idea.

2. In case of NASA-Pub2 Bytes Transferred coincides with explaining the maximum variation while Number of Requests per Session has second maximum variation. In case of CSEE the behavior is exactly opposite where Number of Requests per Session along Principal Component 1, while Bytes Transferred controls the second maximum variation of the data along principal Component 2.

3. In NASA-Pvt2 server number of Requests per Session explains maximum variation on the other hand Bytes Transferred does not show any inclination towards either of the two Principal Components.

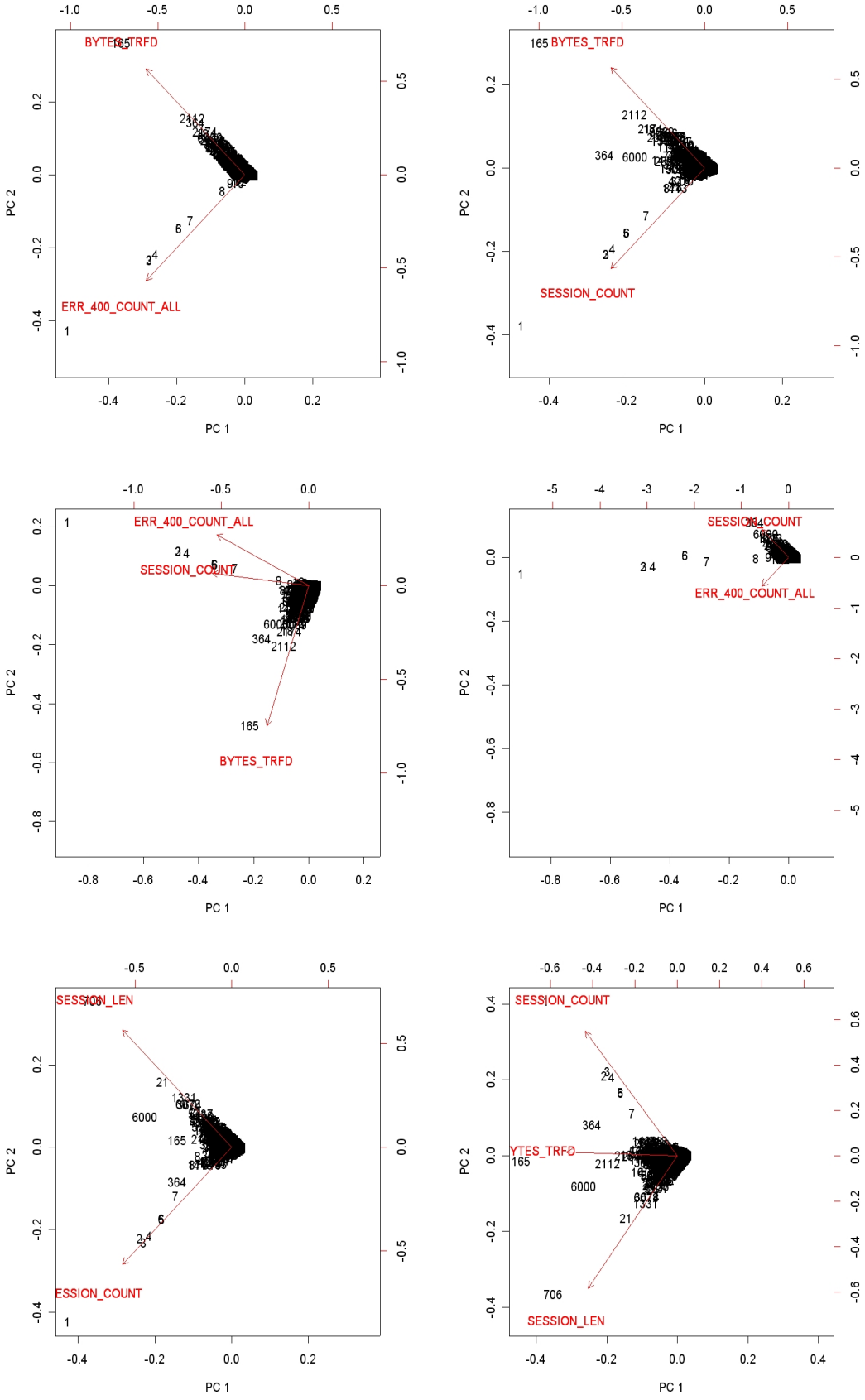

Figure 5.35: NASA-Pub2 *a*

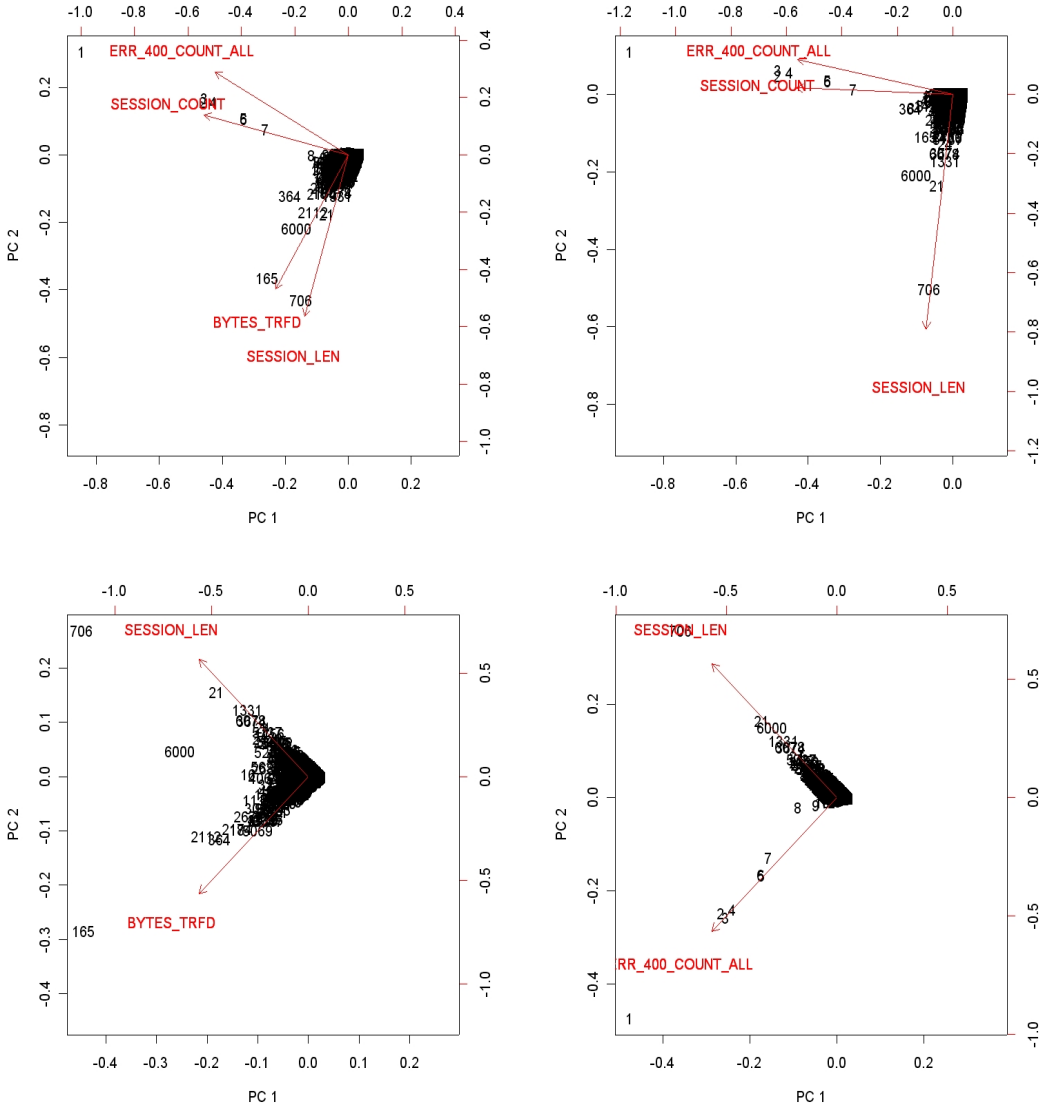

Figure 5.36: NASA-Pub2 *b*

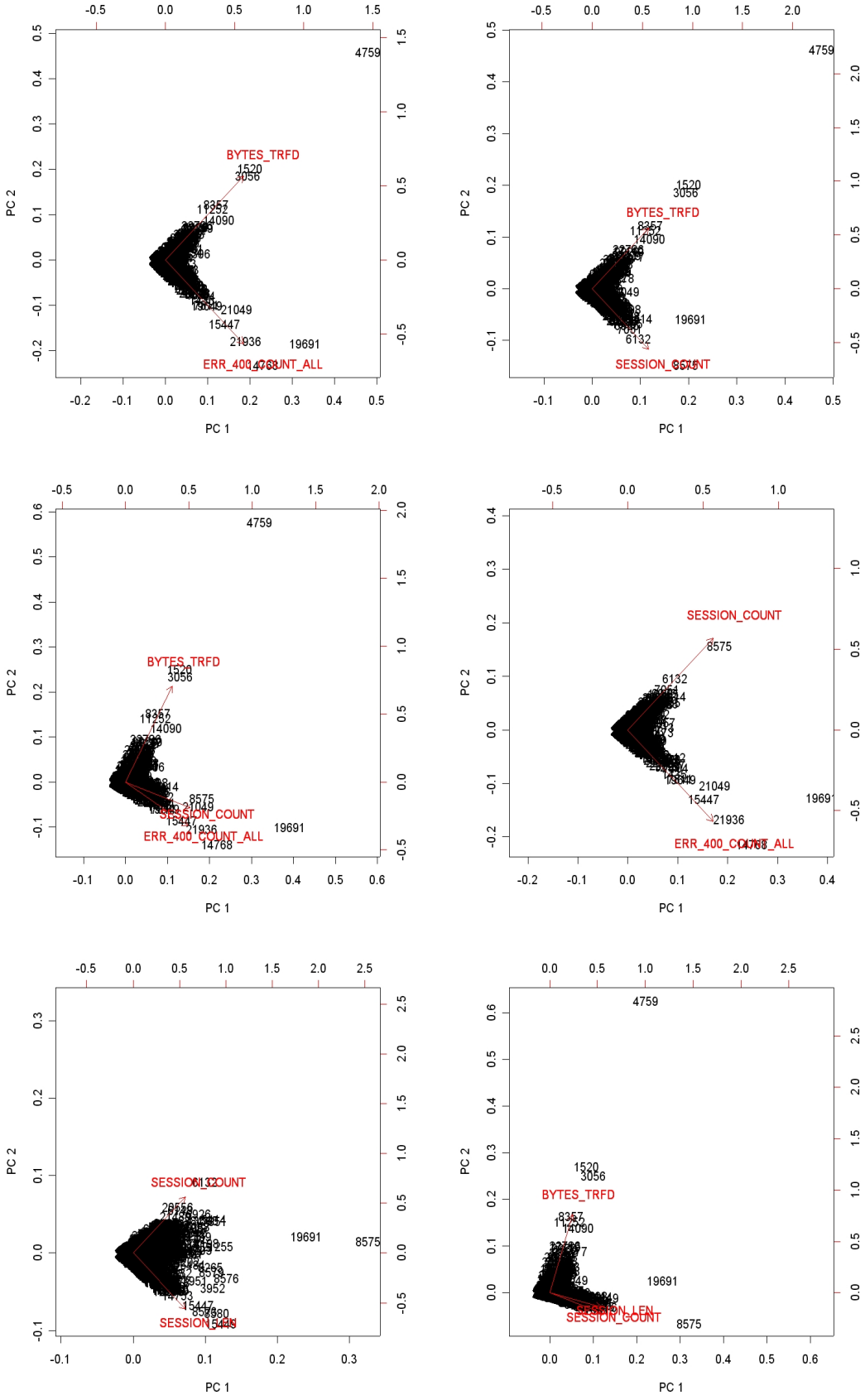

Figure 5.37: CSEE *a*

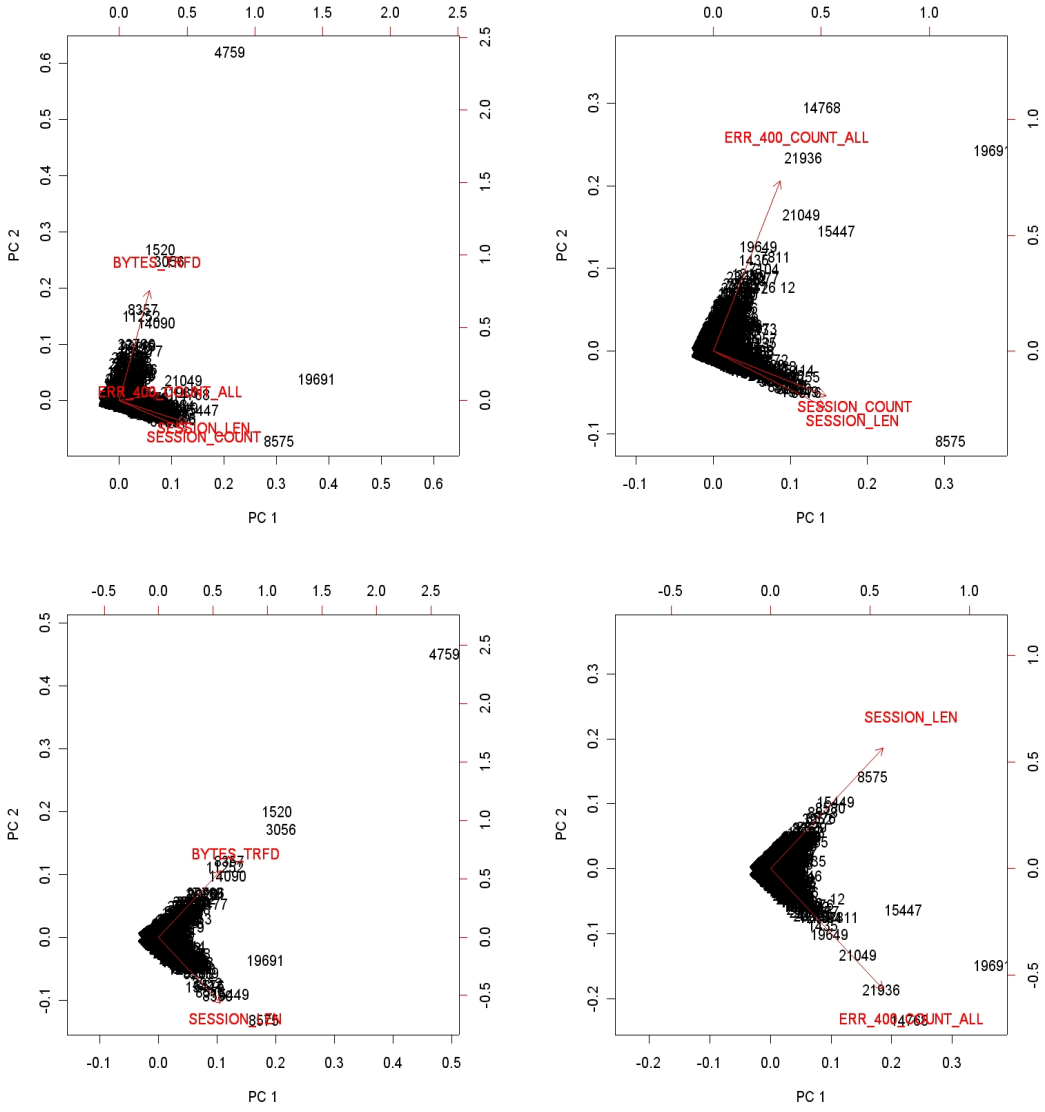

Figure 5.38: CSEE *b*

If we look at the WVU data, we can see clearly that the variation of the data set is not very high in either of the two principal component directions. This means that the data set is very highly correlated and its difficult to find a parameter which governs most of the variation seen.

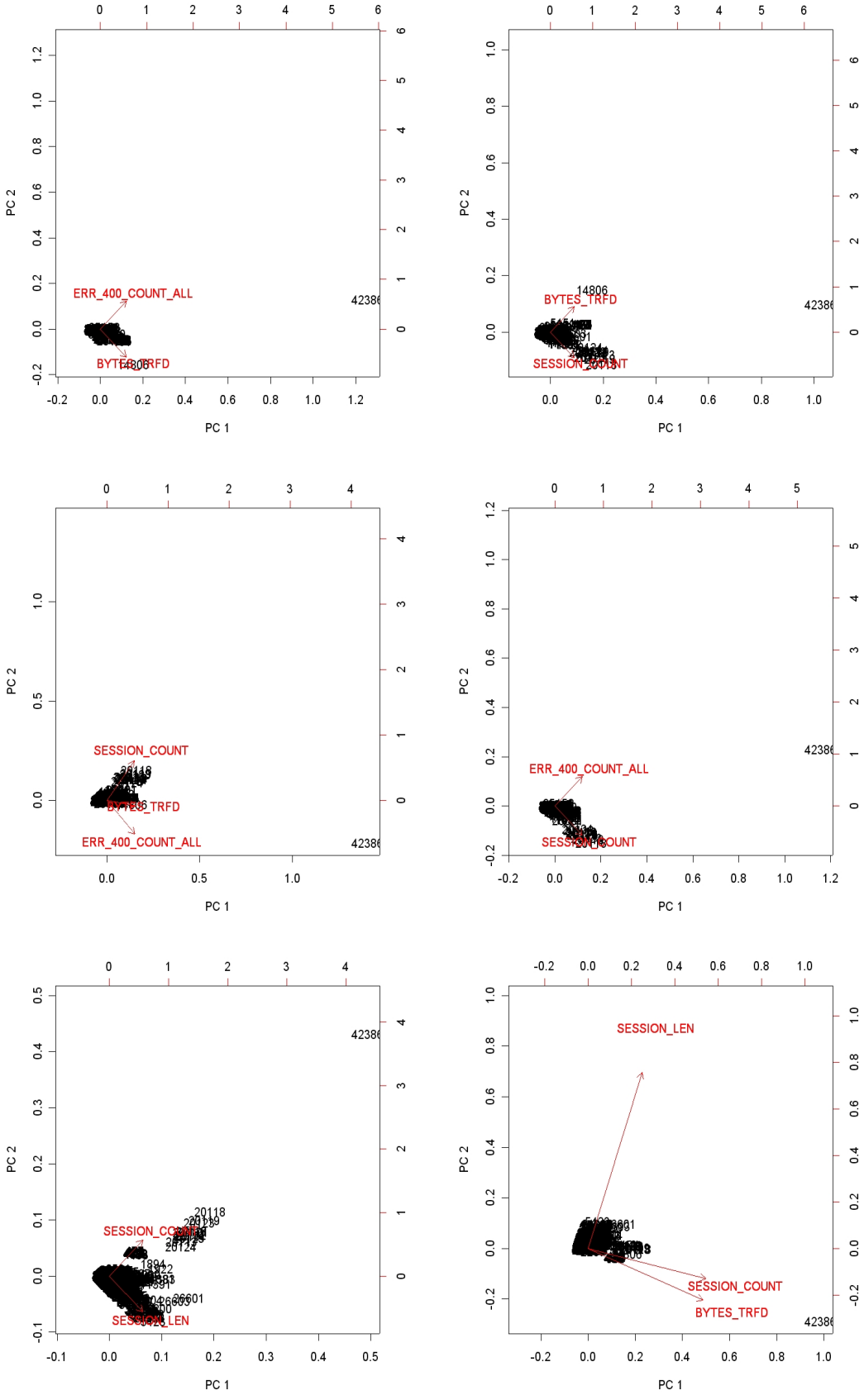

Figure 5.39: WVU *a*

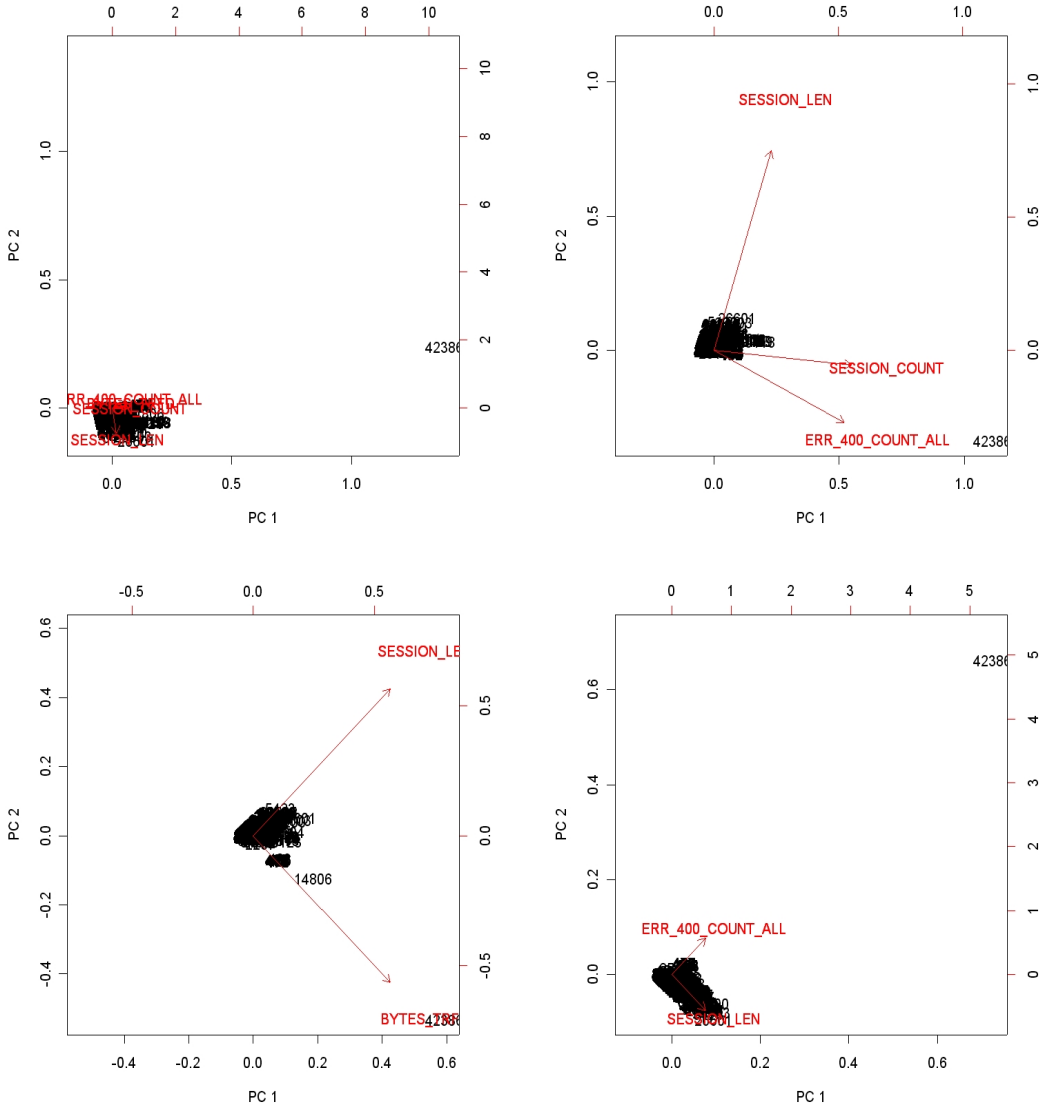

Figure 5.40: WVU *b*

Aim of this analysis is to find out if PCA helps in improving the clustering quality or not. Figure 5.41,5.42 refers to the plot of clusters for Clarknet server, and the data set used is the normalized PCA data. Each principal factor value is used to cluster, and the value of k used is 5,10,15 and 20.

Clustering plot for 10 **and** 15 **cluster size**

Lets take a look at clustering plots of principal components, notice how the axis of data points have changed when compared to the clustering plots of raw data points. As we saw

earlier that Principal Component Analysis decreased the total variation among the data points, it helps to explain the shift of the data points for the principal factors.

Figures 5.41 and 5.42 are plots of clusters for Clarknet data set for 10 and 15 cluster size. Figures 5.43, 5.44, 5.45 show the clustering plot of principal factors for a cluster size of 5. As we can see that the variation of the data is minimum among all the variables. These plots not only help us in understanding the variation of the data with respect to the given variables but also shows the behavior of the clusters as the cluster size is increased.

Clarknet

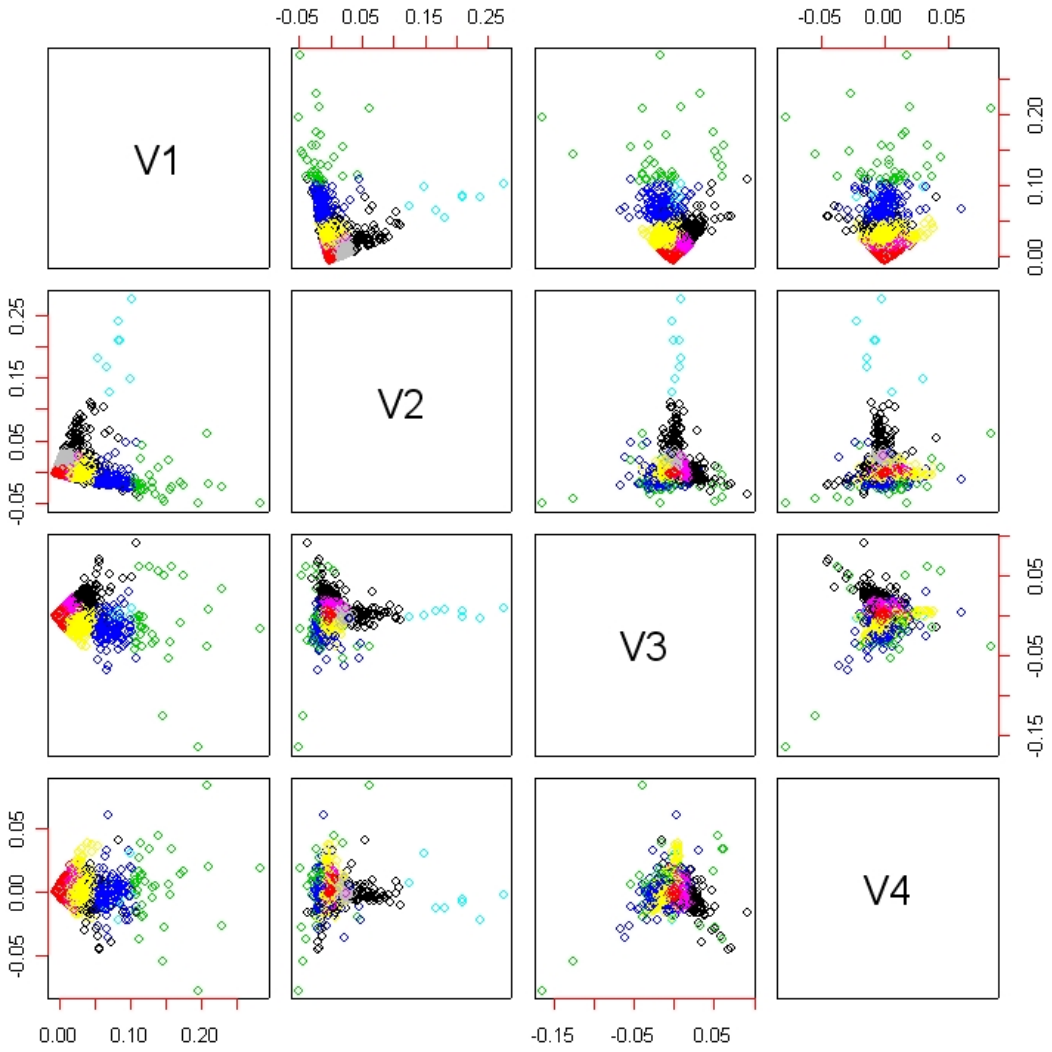

Figure 5.41: Clarknet : Clustering with principal factors for 10 clusters

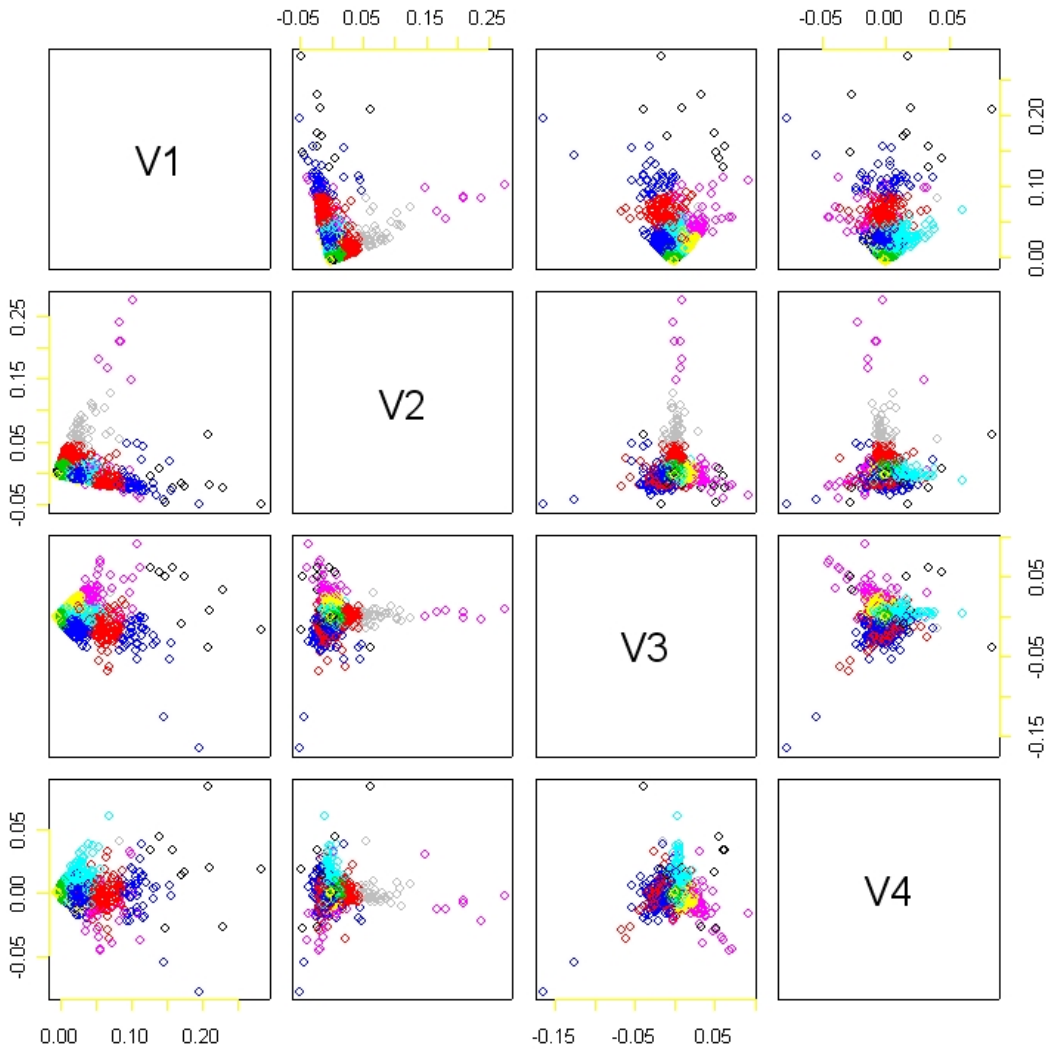

Figure 5.42: Clarknet : Clustering with principal factors for 15 clusters

CSEE

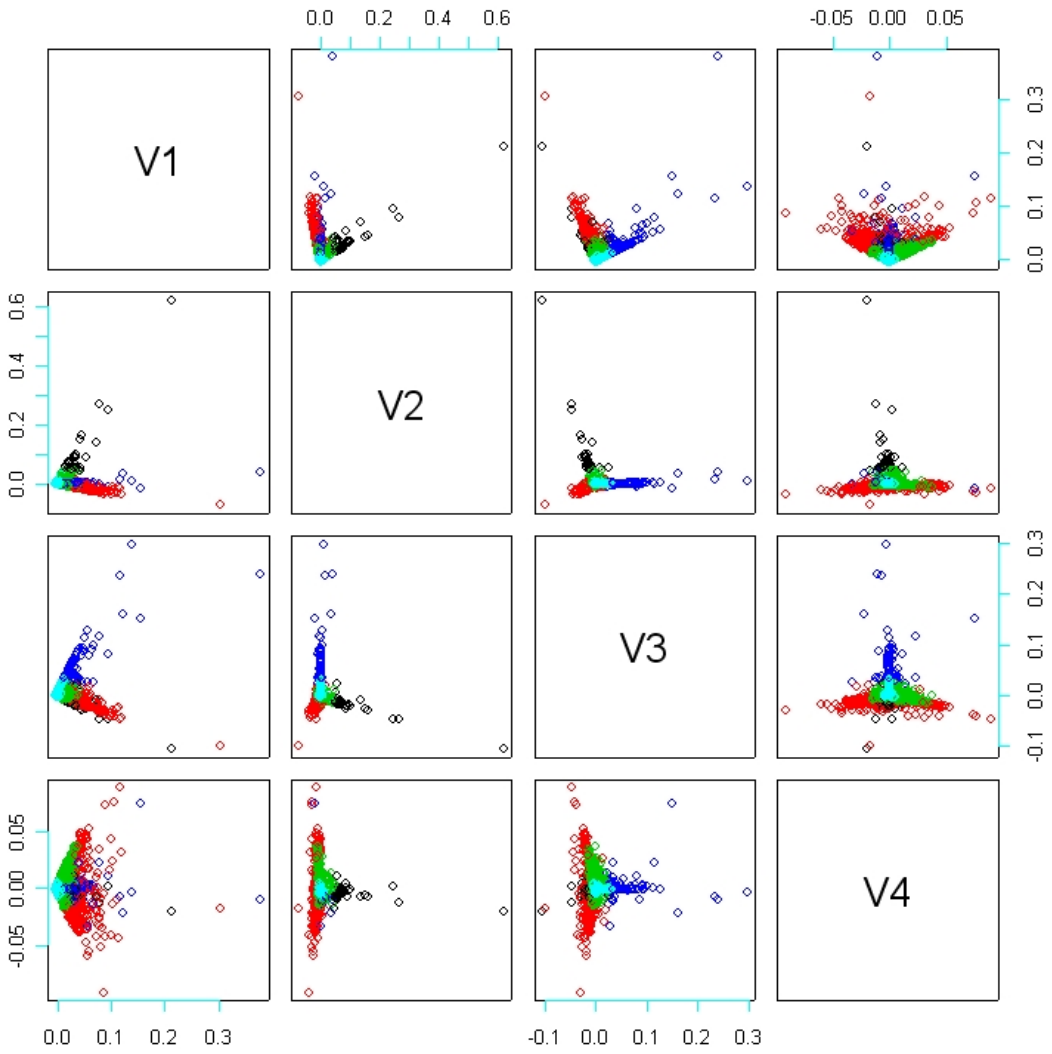

Figure 5.43: CSEE : Clustering with principal factors for a cluster size of 5

WVU

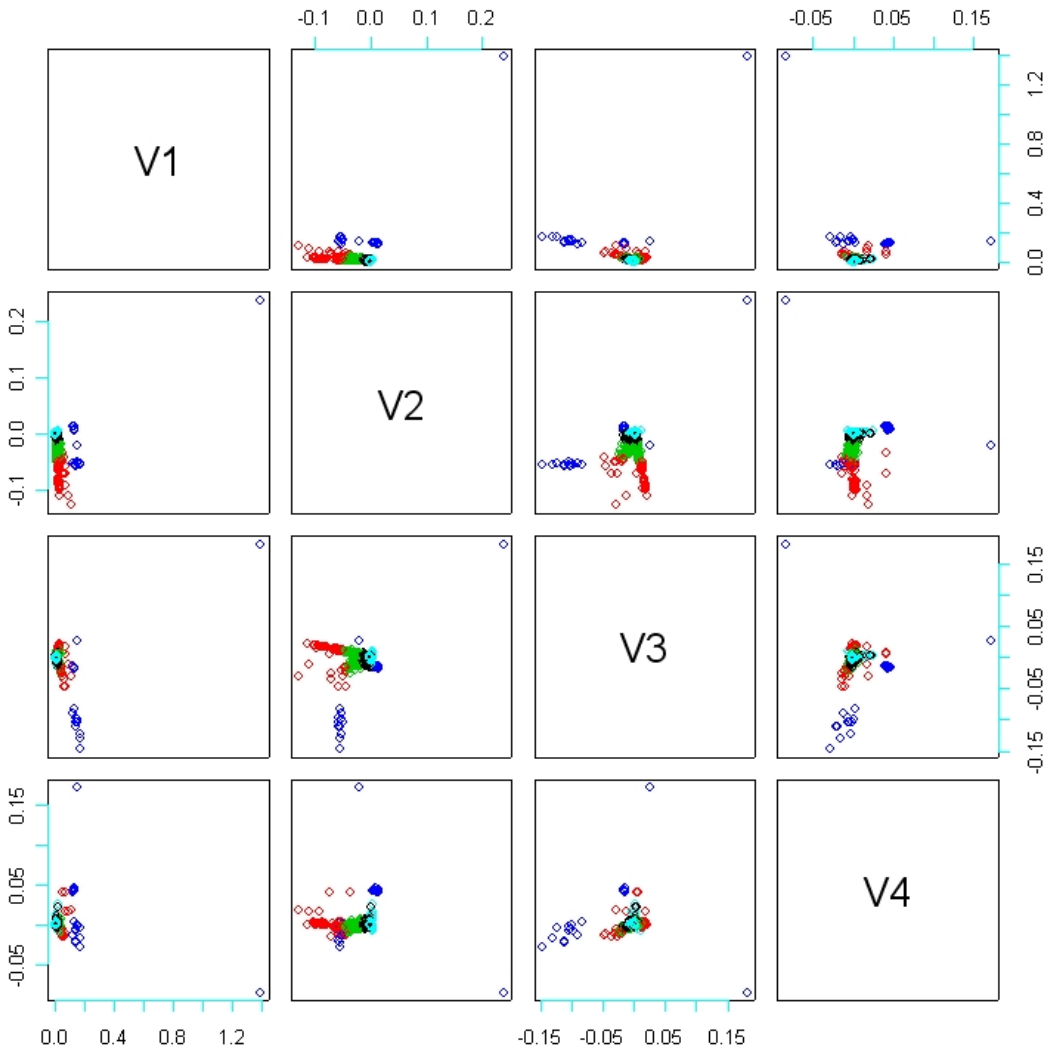

Figure 5.44: WVU : Clustering with principal factors for a cluster size of 5

NASA

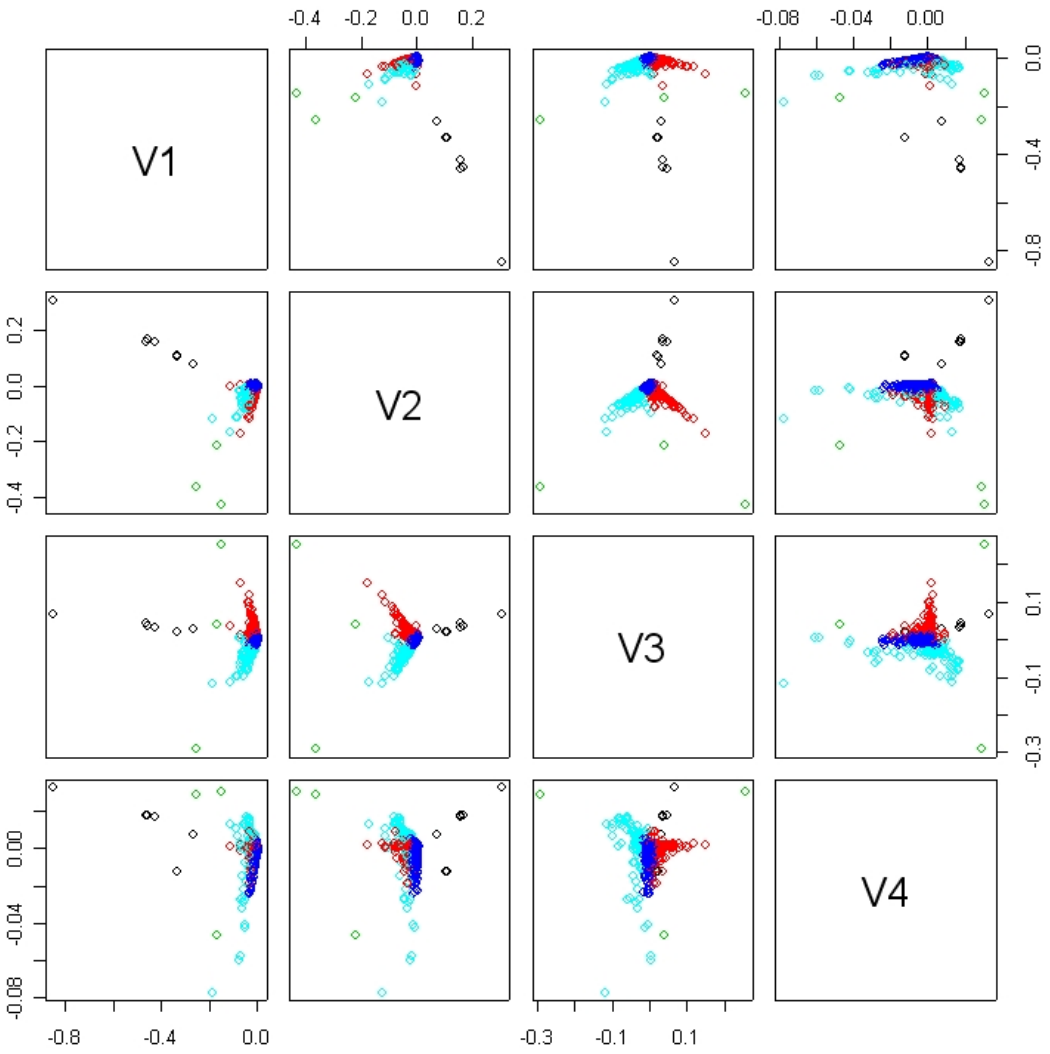

Figure 5.45: NASA-Pub2 : Clustering with principal factors for a cluster size of 5

5.4.2 Cluster quality estimation with PCA

Figure 5.46 to 5.50 shows the variation of coefficients among raw data and data which has been normalized by PCA. The objective here is to see wether PCA changes the values of ratios or not. Following observations were made,

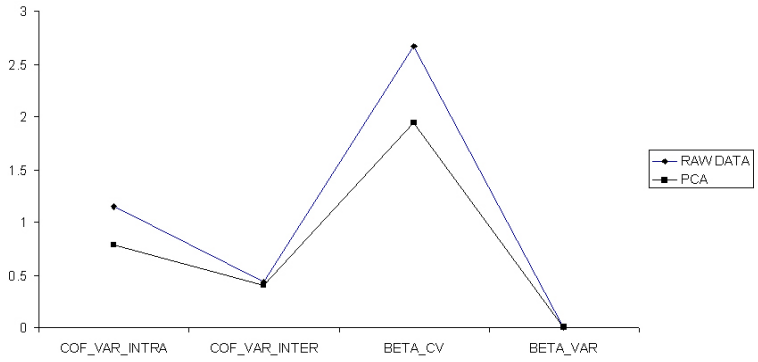

Figure 5.46: Clarknet validity ratios for PCA and raw data

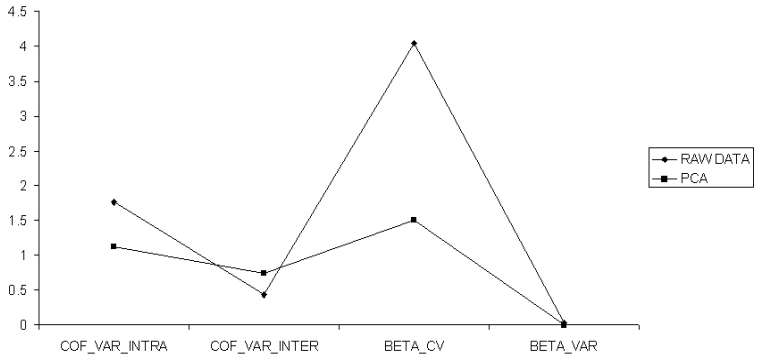

Figure 5.47: CSEE validity ratios for PCA and raw data

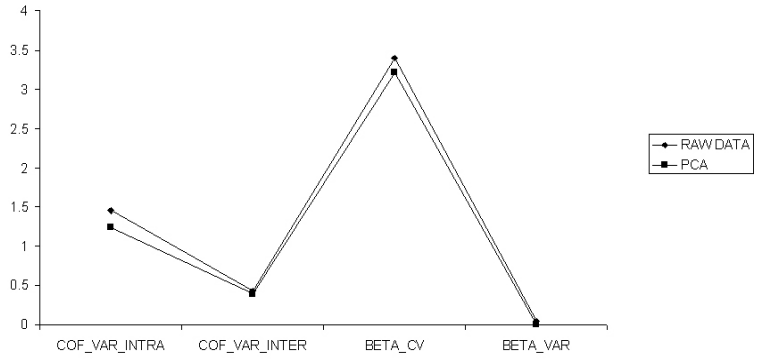

Figure 5.48: WVU validity ratios for PCA and raw data

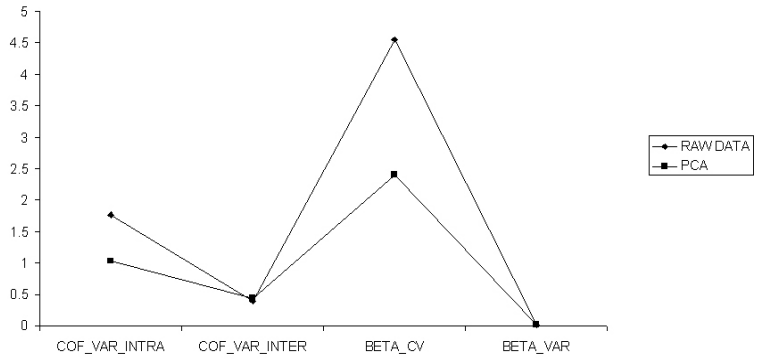

Figure 5.49: NASA-Pub1 validity ratios for PCA and raw data

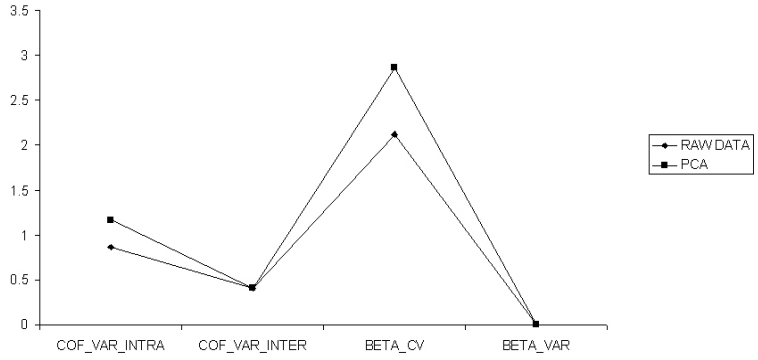

Figure 5.50: NASA-Pvt1 validity ratios for PCA and raw data

The Private NASA servers does not show a decreasing trend for β_{cv}, as the values of β_{cv} should decrease for PCA data. This is because the coefficient of intra-cluster variance increases for all NASA private servers, while coefficient of inter-cluster variance remains unchanged. On the other hand, all other servers have an expected result, where the coefficient of intra-cluster variance decrease for PCA set of data. We also found that the reason for decreasing trend of β_{cv} is attributed to the fact that values of coefficient of variation of intra-cluster distance decreases while the values of inter-cluster distance does not vary much, in fact it is almost constant for all the servers.

An interesting fact remains that in almost all servers value of β_{var} remains constant. Also the good part is that the value of β_{var} is very small for all the servers. One fact, which remains to be investigated further is, why CSEE shows a increase in the value of coefficient

of variation of inter-cluster distance with PCA data.

5.5 HTTP error code characterization

This section discusses about how the HTTP error codes are distributed among different servers, and what are the possible reasons behind it. In this section we also try to deduce what types of errors are predominant and how do they behave in different servers.

5.5.1 RAW data for HTTP error characterization

Before we head into the details, lets take a look at some general statistics about the 9 data-sets involved in this study. Table 5.4 shows the total number of 4XX level and 5XX level error counts in the given data-sets. These error codes are specifically chosen for either security analysis or server file management analysis.

Server	Total Errors	Total 4XX	Total 5XX	Total 400	Total 401	Total 403	Total 404	Total 405	Total 500	Total 501	Total 502	Total 503
WVU	337,351	331,226	6,125	43,296	113	8,534	276,523	2,306	6,110	15	0	0
Clarknet	36,773	36,502	271	0	0	1,467	35,035	0	271	0	0	0
CSEE	73,828	73,055	773	230	2,987	4,015	63,577	2,164	423	7	56	287
NASA -Pvt1	337	337	0	0	146	0	191	0	0	0	0	0
NASA -Pvt2	267	267	0	0	0	0	267	0	0	0	0	0
NASA -Pvt3	4,066	4,066	0	0	364	4	3,697	0	0	0	0	0
NASA -Pub1	4,623	4,623	9	212	0	110	4,134	158	0	9	0	0
NASA -Pub2	35,694	35,476	218	1,027	304	143	33,520	375	2	216	0	0
NASA -Pub3	2,938	2,929	9	23	0	16	2,707	175	0	9	9	9

Table 5.4: HTTP error distribution

All the private NASA servers have zero 400 error responses (bad requests), while all NASA public servers have at least 1% or more of total 4XX errors. NASA private servers also don't have any 5XX errors. Only CSEE and NASA-Pub3 servers have 502 and 503

errors i.e. up-server bad response, server unavailable, meaning these servers either acted as proxies/gateways or had problems with the free resources. The resource problem may can be a result of traffic overloading over the network. CSEE server might have high network traffic, as it is also one of the busiest servers with respect to requests per day. With a busy server, most of responses map to unavailable servers.

Other than NASA-Pvt1 server almost all servers have 83%[1] or more 404 errors (page not found). NASA-Pvt1 server also has almost 43% 401 errors (unauthorized access). Table 5.1 shows that NASA-pvt1 server is the only server which has unexpectedly least number of sessions. Further analysis found that almost 77% of the bytes transferred on NASA-Pvt1 server were from PDF document downloads. NASA-Pub3 also shows an exceptional download of almost 91% bytes in documents(.PDF and ,.DOC) We found that NASA-Pvt1 server also has robots??what percentage and stuff?

5.5.2 HTTP error response codes characteristics

First of all we look at general distribution of errors. We concentrate on the data-set behavior with respect to number of error responses we got for all the servers. The reason for selecting these specific HTTP response codes was to include those which might help us study in the area of security related issues, such as brute force attack or denial of service attack. The reason being that in any of these cases there might be some pattern to the response codes in study.

- The total number of errors generated.
- Percentage errors (404, 401, 400) within a data-set.
- Percentage errors for a session.
- Number of errors generated per session

[1] All percentages are with respect to total number of errors

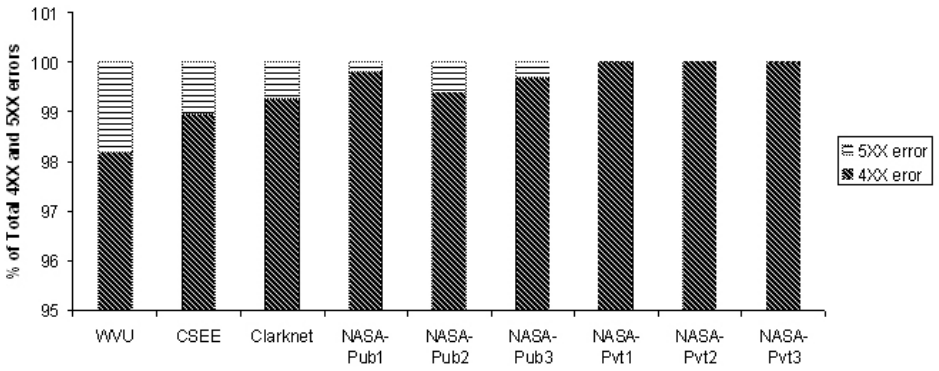

Figure 5.51: Distribution of 4XX and 5XX level

Figure 5.51 represents the percentage error distribution of 4XX and 5XX level errors among given data-sets. We can see that almost all the data sets have a major contribution of 4XX level errors towards the total errors , while 5XX level errors contribute not more than 2% in any of the data-sets. We can also see that all the private NASA servers have no 5XX level errors.

If we break the 4XX and 5XX level errors individually as shown in figure 5.52 and figure 5.53 , we can see following

- In almost all servers except NASA-pvt1, 404 level error constitutes more than 80% of total 4XX errors.
- CSEE, Clarknet and all NASA-pvt servers have no 400 errors, also All NASA-pvt servers are devoid of 405 errors.
- None of the NASA-pvt servers have any of 5XX errors.
- All NASA-pub servers have 501 errors unlike NASA-pvt servers and they also constitute almost all of the 5XX level errors they have.

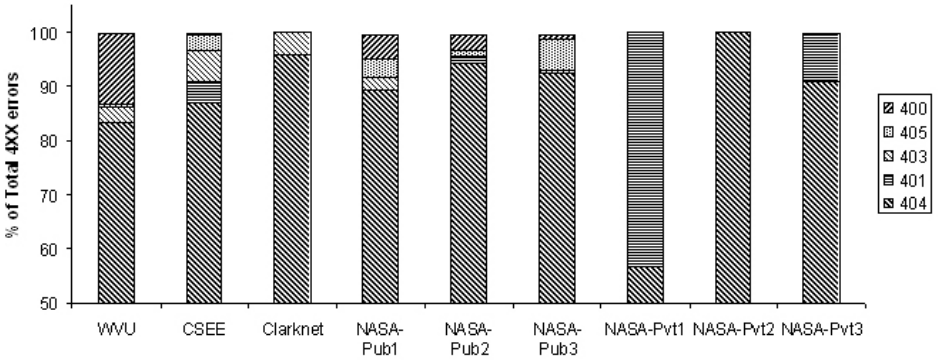

Figure 5.52: Distribution of 4XX level errors

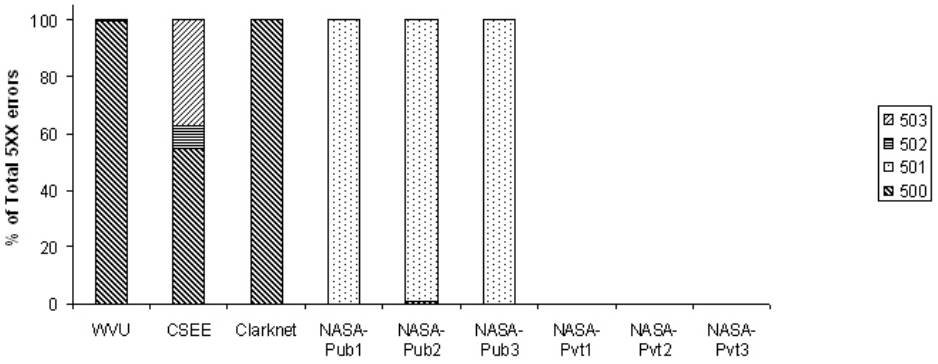

Figure 5.53: Distribution of 5XX level errors

Now lets take a look at the individual distribution of these error response codes in different servers. Figures 5.54 to 5.57 represents the distribution of HTTP error response codes for different servers. It seems 404 errors are predominant in almost all the servers, while bad requests are rare in Clarknet and WVU, CSEE and NASA public servers have sessions with large number of bad requests.

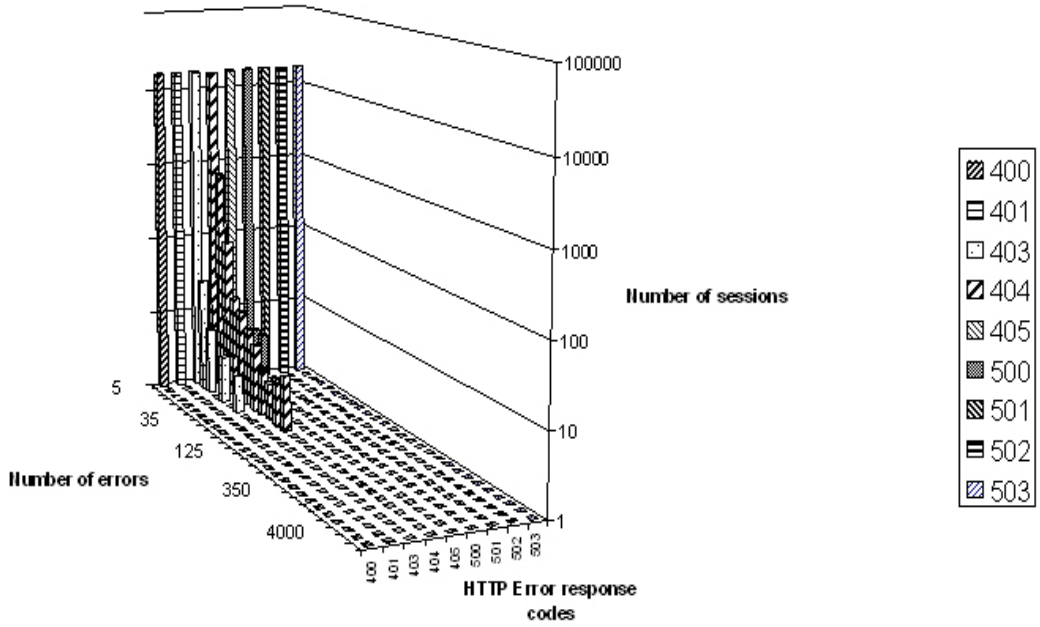

Figure 5.54: Distribution of HTTP error response codes in Clarknet

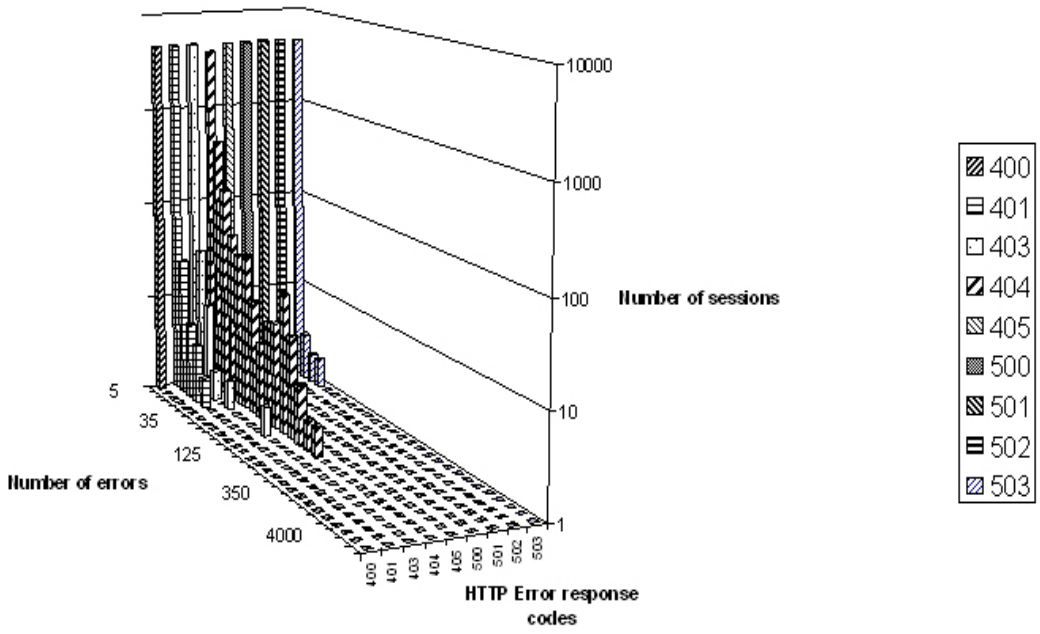

Figure 5.55: Distribution of HTTP error response codes in CSEE

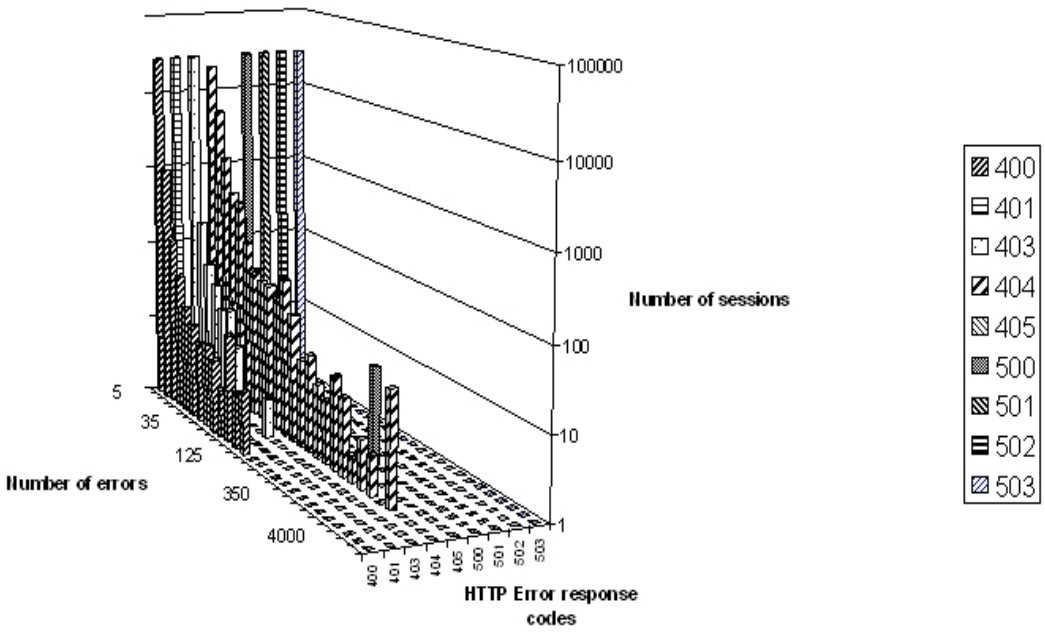

Figure 5.56: Distribution of HTTP error response codes in WVU

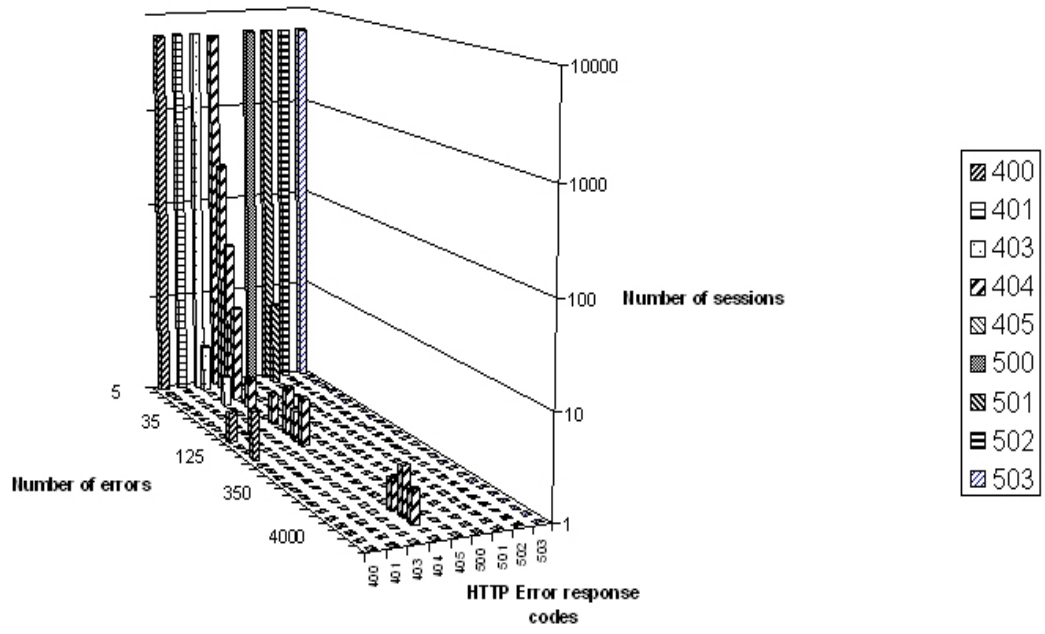

Figure 5.57: Distribution of HTTP error response codes in NASA-Pub2

5.5.3 Comparison between clusters with and without error count

1. The pattern of clusters looks same if compared based on any 2 parameters. When we consider Number of requests per session and bytes transferred, we can see that the in case of cluster size 5, clusters align almost in the same direction. The resulting pattern is almost identical.

2. It has to be seen that whether this pattern remains the same when we increase the cluster numbers. This can be done by plotting the clusters with all four parameters.

3. When the comparison was done with higher number of clusters for the same data, the outer-limit data points showed similar cluster characteristics i.e. the outer clusters were almost always similar. A closer inspection of inside data points reveals that as the number of cluster points were increased, the physical pattern of those clusters changes.

5.6 Sessions with robots

Now lets have a look at the sessions with respect to robots distribution in it. Figure 5.58 to 5.61 shows the distribution for each data set for one clustering exercise each. We can see that,

In case of data sets with 5 clusters, Clarknet, CSEE and WVU have a clear distinction of cluster having the majority of robots in it. In case of NASA servers the clusters having robots are distinct, but when the size of total clusters is increased to 10 or 15 the sessions containing robots are dispersed more acutely than other non-NASA servers. All the NASA private servers show a trend of distribution of robots among 2 or more clusters, and NASA-Pvt2 does not have any robots in sessions with at least one error. NASA public servers have a well defined robots characteristics. We also found that majority of the servers have at least one session with maximum robot percentage, and the cluster maintains this property for a varying size of k.

In case of data sets with 10 or more clusters, As the number of clusters are increased the distribution of robots becomes a little scattered, as was observed in couple of NASA servers, but this might be due the fact that these servers have a low amount of data points in them

and as we saw earlier a large size of k, i.e. 15 or 20 deliberates the cluster formation. This is the reason why session with robots are scattered when the k size is increased from 10 to 20. This study can also help us in choosing optimal number of sessions for robots characterization as we can see that data sets with 10 and 15 clusters have a better representation of robots session than with 5 clusters, while cluster size of 20 make the distribution skewed and hence difficult for us to analyze. In case of some NASA servers this might not be true because of low amount of raw data points in them.

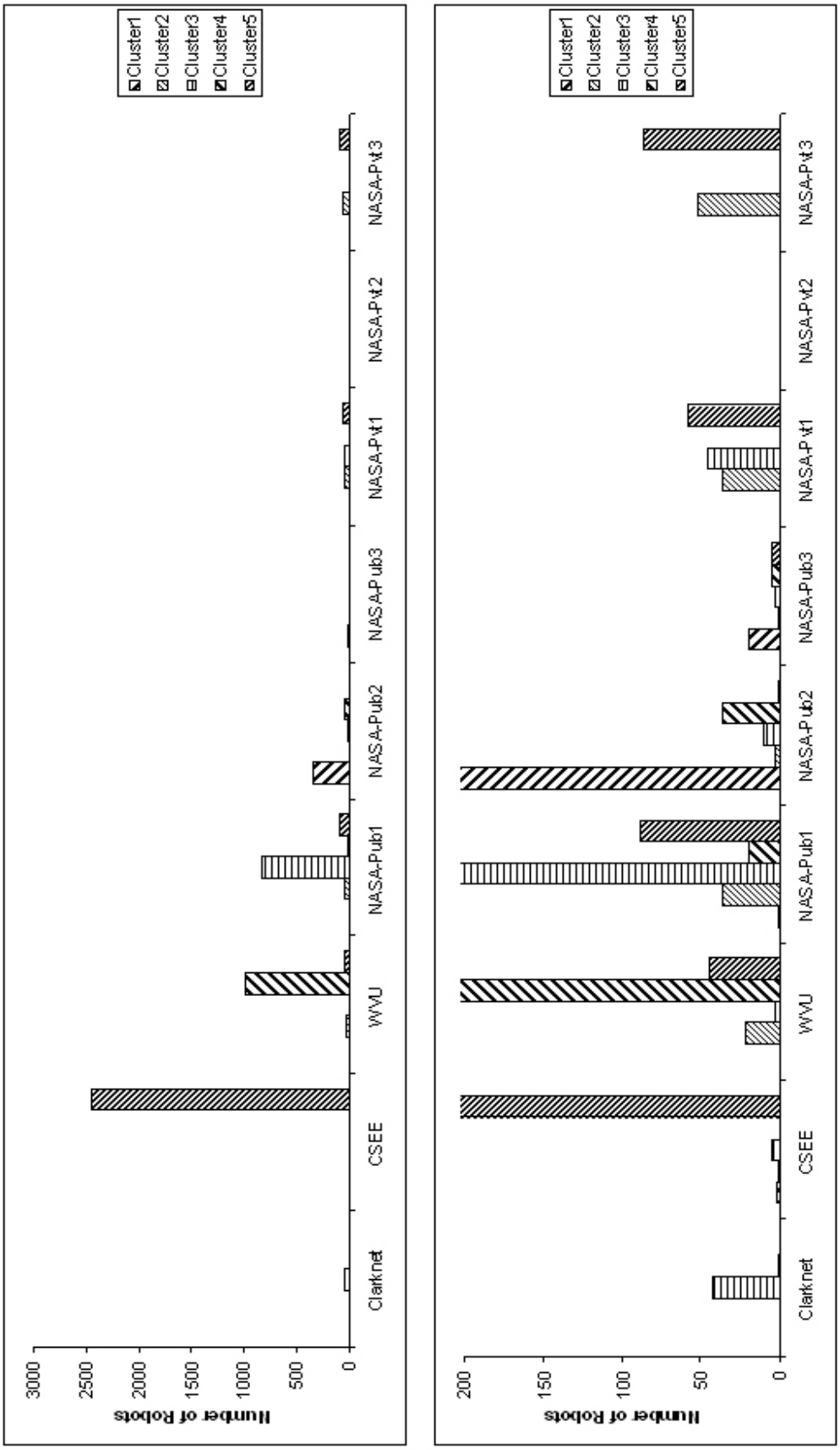

Figure 5.58: Robots distribution over sessions for 5 clusters

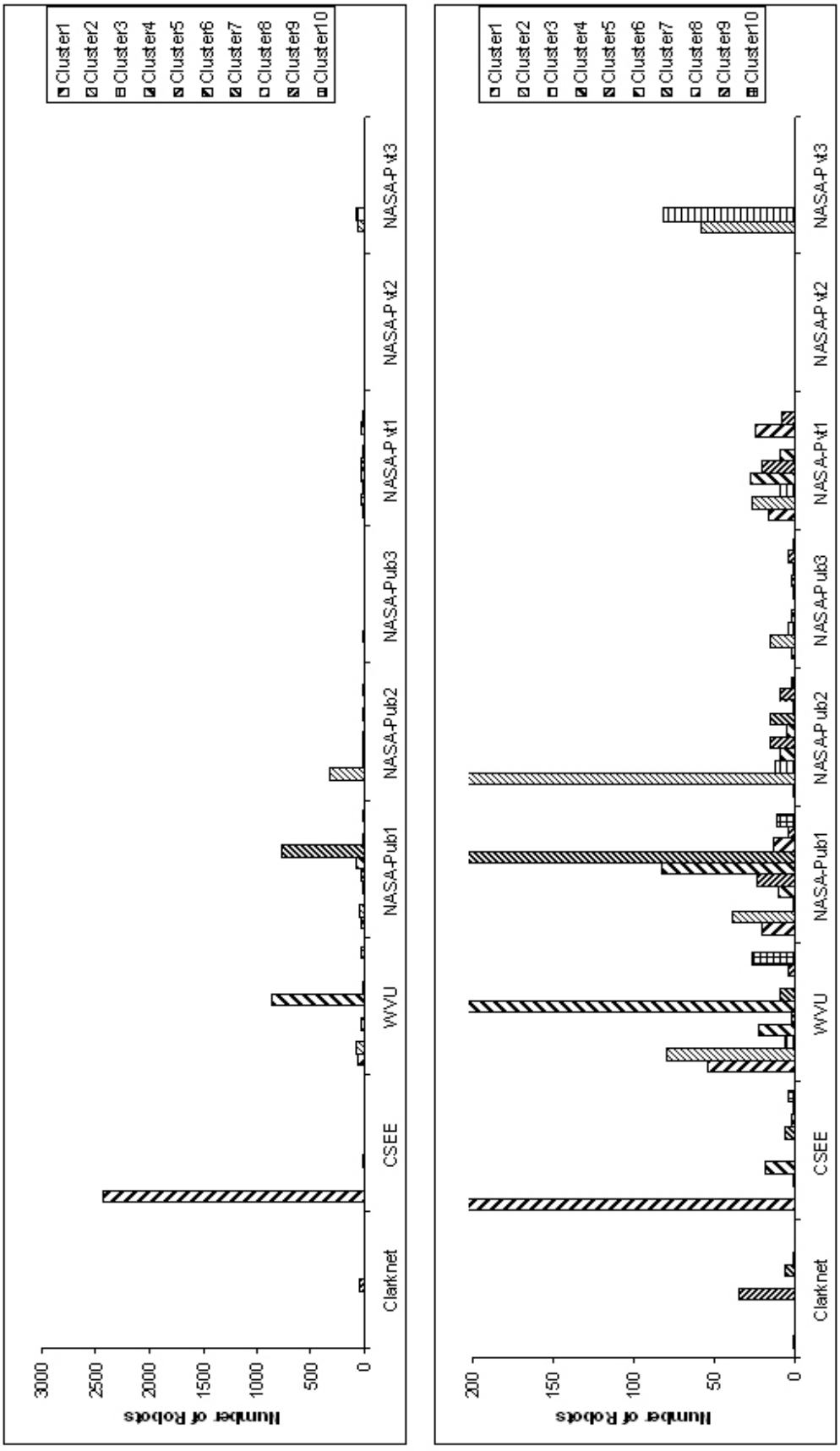

Figure 5.59: Robots distribution over sessions for 10 clusters

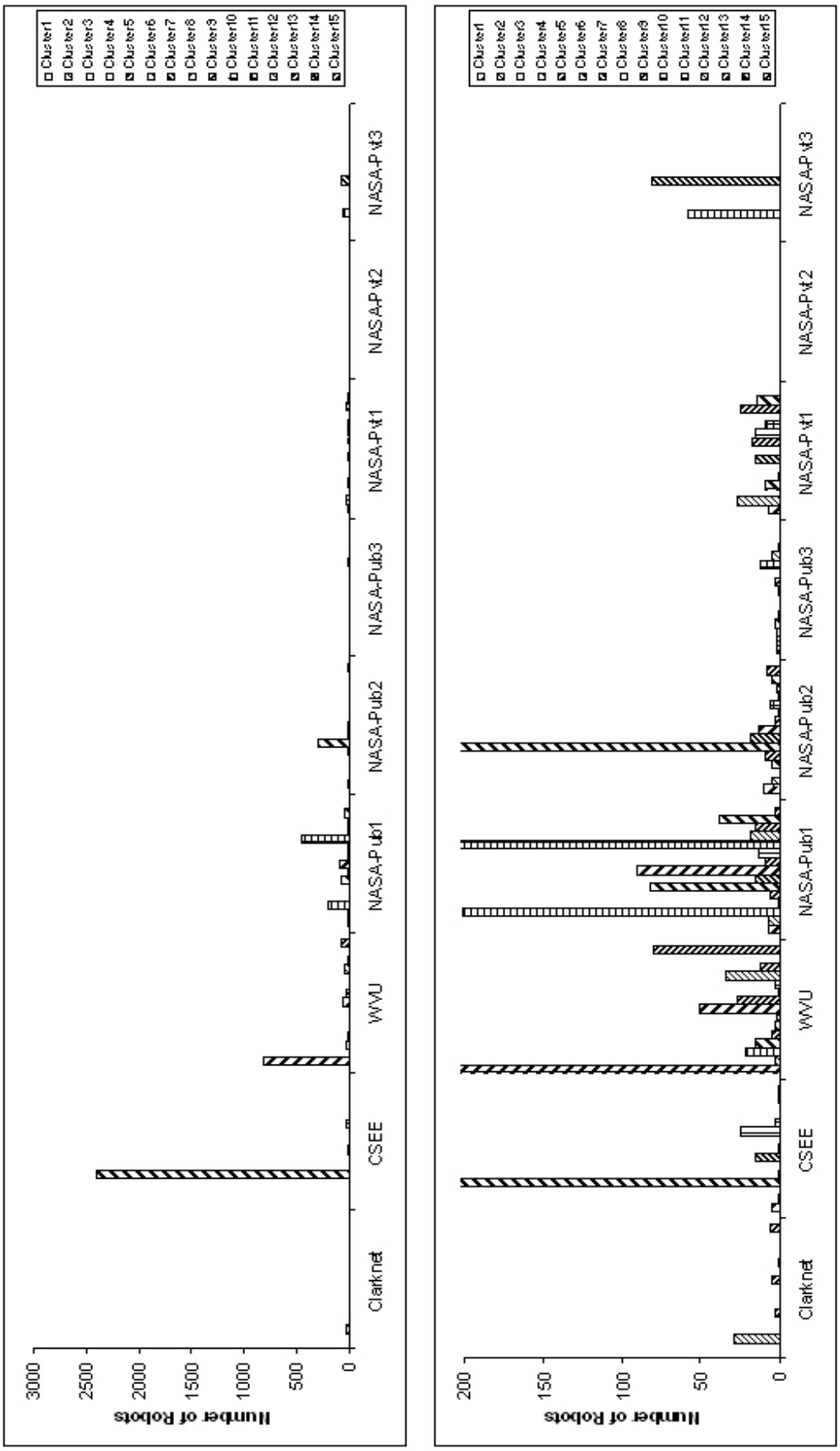

Figure 5.60: Robots distribution over sessions for 15 clusters

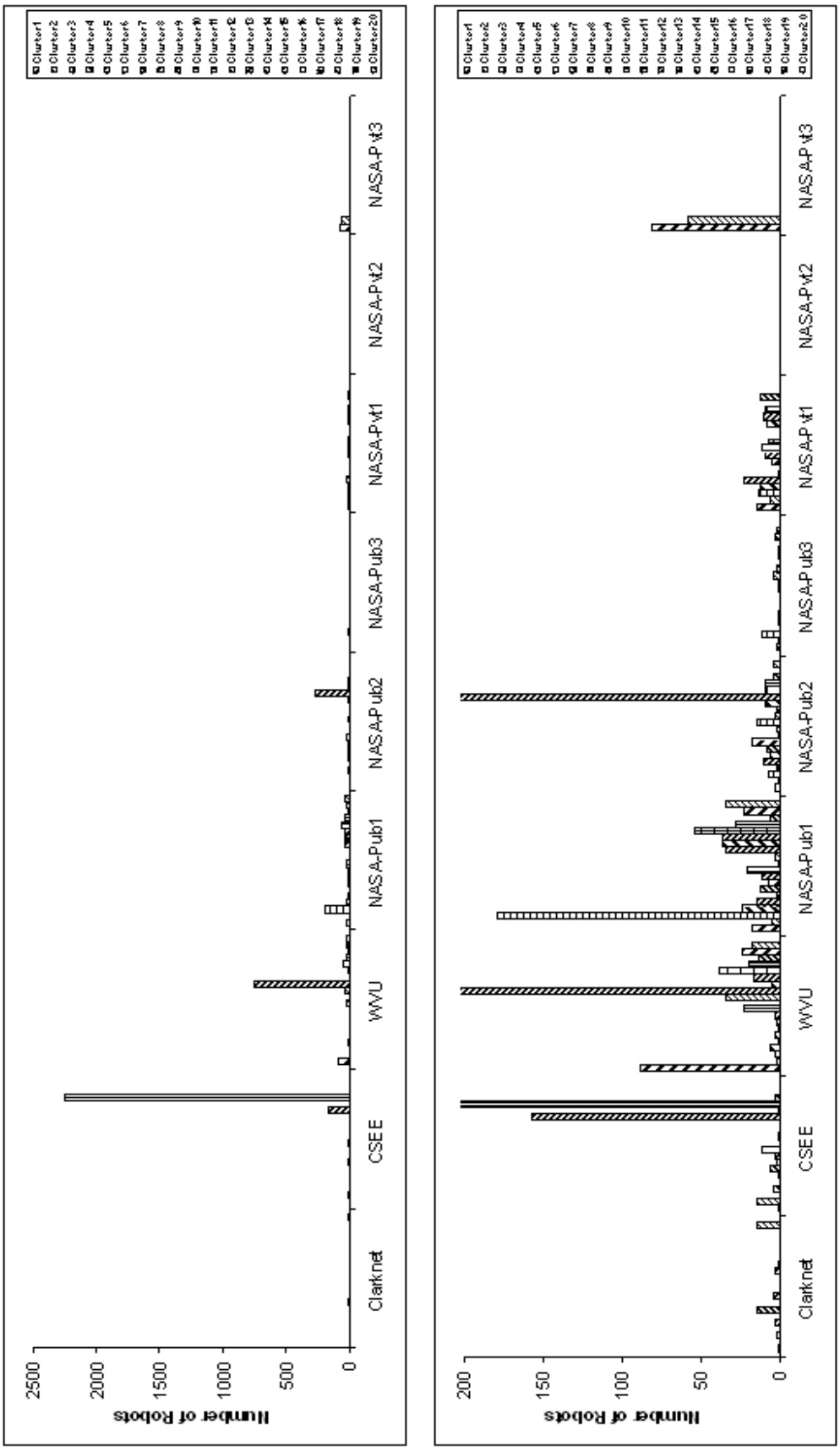

Figure 5.61: Robots distribution over sessions for 20 clusters

5.6.1 Robot session characteristic

Another good measure to find robots characteristics, is to plot them against the distribution of centroids of each clusters with respect to raw data. This gives us an idea of how the robots sessions behave with respect to the clusters and number of clusters as well. We have plotted the percentage values of each cluster with respect to the sum of centroid values for that cluster against the cluster number. A point to be noted is that the plot of robots is done with respect to total robots for all clusters.

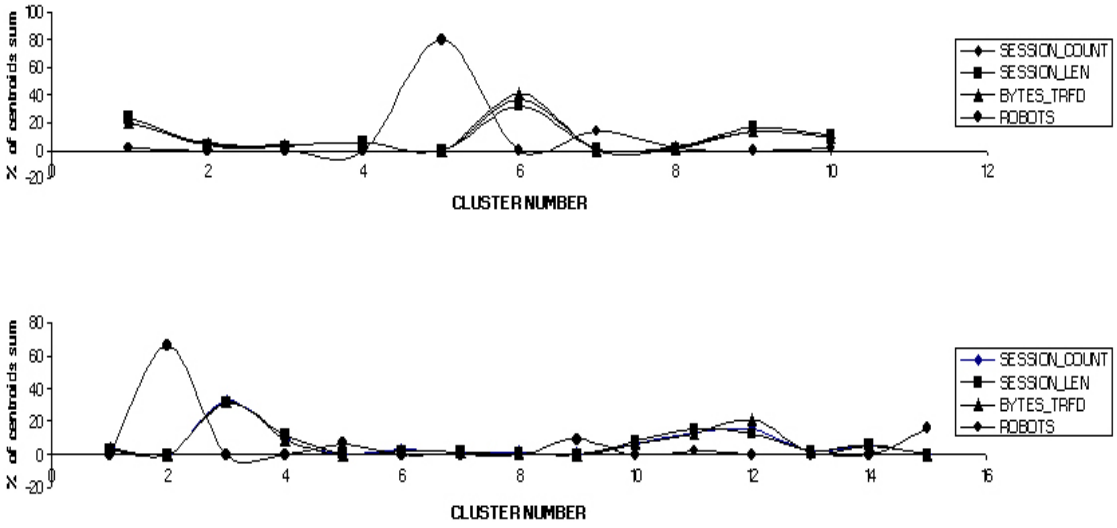

Figure 5.62: Clarknet : Distribution of robots over percentage of total centroid values

Figure 5.62 shows the distribution of robots over different clusters centroid sums. The general trend we found was that the robots usually have a session with least values of centroids, meaning that almost all of the data points in that cluster have lower values associated to them i.e. Less number of requests per session and less number of bytes transferred or even lower values of session length. In case of 5 clusters, cluster number 3 has more than 97% of robots, while the number of requests, session length, and bytes transferred constitute no more than 1.5% of their total values for that cluster. As we increased the number of clusters the distribution of robots settled down a little bit but we observed peaks irrespective of that. With a change in cluster numbers from 10,15 to 20, the peaks for robots changes to 89%,

65%, and 36% 31%. We can also see that as the cluster number increases from 15 to 20 the peaks are divided into 2 and hence making it obvious that the robots divide into two different clusters. We also observed that in case of 20 clusters, almost 4 clusters accounted for almost$5-10\%$ of the robots. We think clustering between 5-10 clusters yields the optimal results in this case.

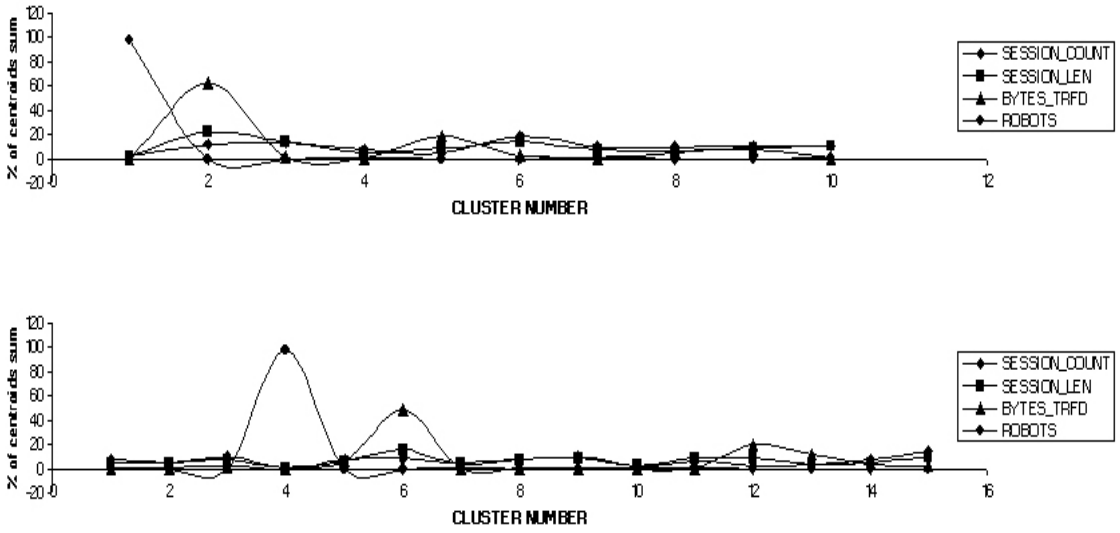

Figure 5.63: CSEE : Distribution of robots over percentage of total centroid values

Figure 5.63 shows the same trend as Clarknet. Almost all ($> 99\%$)the robots lie in session 5, for 5 clusters. It is interesting to know that in case of CSEE, the change of cluster numbers does not effect the percentage of robots in one cluster. In all the cases more than 90% of the robots are confined to a single cluster. This means clustering is effective in case of CSEE servers, keeping the robots together.

Figure 5.64 shows the robots distribution for NASA-Pub2 and NASA-Pvt1 servers. This follows the same trend as CSEE as the peak for robots does not get distributed over number of clusters. We also noticed that the robots sessions have very low centroid values, in fact these are the lowest values among all other cluster centroid values. We have plots of NASA public and private servers for 5 clusters. Space and Time constraints are the main factor of not including rest of the plots.

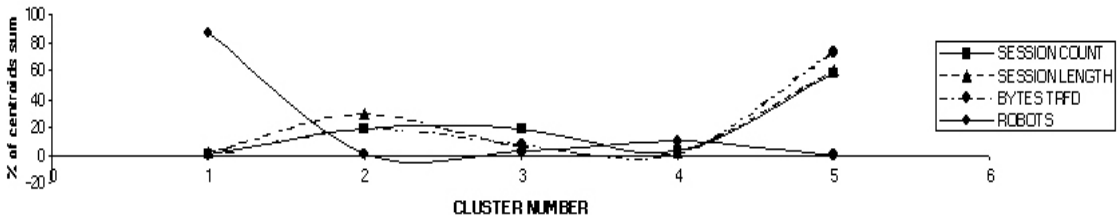

Figure 5.64: NASA-Pub2 : Distribution of robots over percentage of total centroid values for 5 clusters

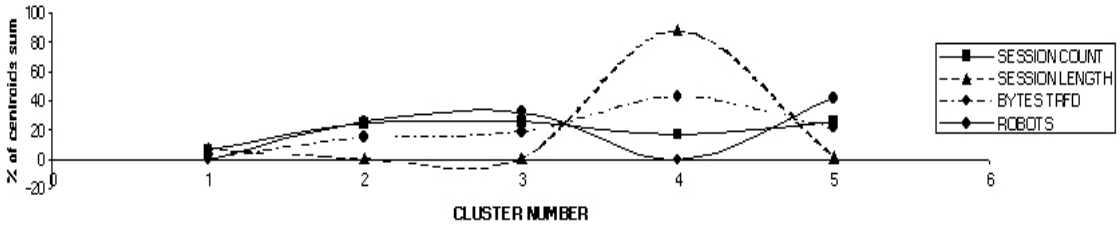

Figure 5.65: NASA-Pvt1 : Distribution of robots over percentage of total centroid values for 5 clusters

Conclusion

All the servers show the preservation of sessions with robots. The percentage of robots retained decreases, as the number of clusters is increased.

5.6.2 Robot session distribution

Figure 5.66 shows the distribution of percentage of sessions out of total sessions in clusters having maximum robots with respect to changing values of k i.e. as the number of clusters is increased from 5 to 20.

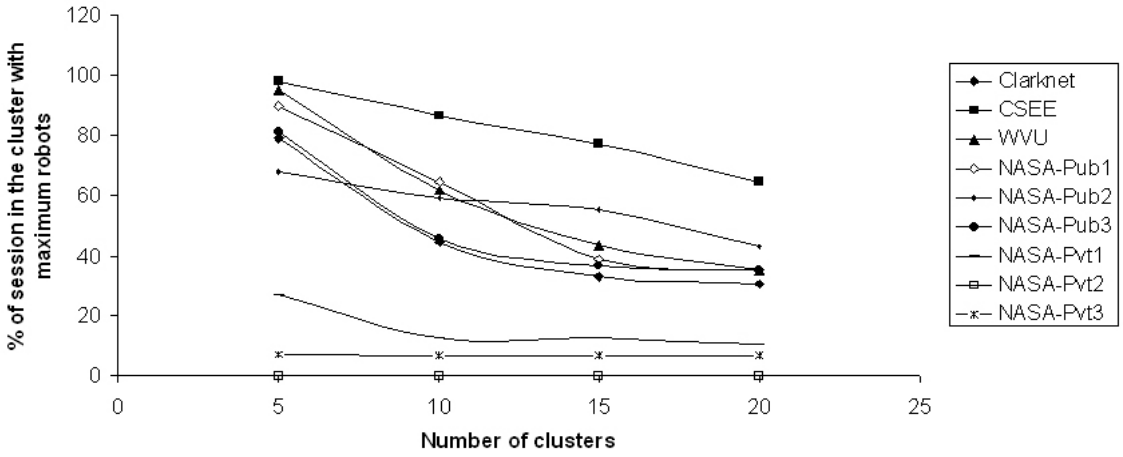

Figure 5.66: Variation in the value of percentage of sessions in the cluster with maximum robots

Clearly we can see that the retention power decreases as the cluster number is increased, this can be explained due to the fact that as the cluster number is increased the overall session's distribution gets skewed and hence the robots in those sessions will also follow the trend. But if we look closely we will find that in case of NASA private servers (NASA-Pvt2 does not have any robots in them), when increasing the size of clusters from 10 to 15, the old cluster's session distribution with maximum robots, remain the same as the new one.

In case of Clarknet, the request per session, session length and bytes transferred for the centroid determined by *K-Means* clustering vary as the number of clusters are increased, the parameter values for these centroids decrease with an increase in cluster numbers. Our objective is to find out the optimal cluster size for best robot-session representation. As we can see that robots always exist in clusters having the maximum amount of sessions with minimum centroid values. One reason can be that the robots have a smaller session length, with less number of requests per session and low bytes transfer value. Even if the robot sessions have large number of requests or long session length, either one of these , the bytes transferred in those sessions will have to a low value.

Finally we can say that almost all the servers, except NASA private servers, have maximum robots in clusters with maximum number of sessions in them. This is not to be confused with the finding that robots are also predominant in clusters with lowest values for

its centroids, as discussed in section 5.6.1.

Server	k Value	Percentage Sessions	Percentage Robots	Total Percentage Robots
Clarknet	5	78.97	97.67	0.25
Clarknet	10	44.59	81.39	0.21
Clarknet	15	32.88	67.44	0.18
Clarknet	20	30.48	34.88	0.10
WVU	5	95.05	93.50	2.02
WVU	10	61.96	81.07	1.75
WVU	15	43.53	75.61	1.63
WVU	20	35.18	71.75	1.55
CSEE	5	98.00	99.71	9.57
CSEE	10	86.68	98.74	9.48
CSEE	15	77.24	97.88	9.40
CSEE	20	64.67	91.68	8.80
NASA-Pub2	5	68.01	86.94	5.05
NASA-Pub2	10	59.25	81.98	4.77
NASA-Pub2	15	55.24	78.07	4.54
NASA-Pub2	20	42.97	71.02	4.13

Table 5.5: Robots distribution as percentage of total robots and total sessions for 5,10,15 and 20 cluster sizes.

Table 5.5 gives the percentage of sessions and percentage of robots for clusters having maximum robots in them. We limit ourselves to couple of servers, and try to figure out any pattern this follows.

As we can see in table 5.5, most of the servers have a god robot retention as the cluster size, k, is increased from 5 to 20, except Clarknet all servers follow the trend. Figure 5.66 shows similar information for all the servers and as explained earlier a value of 10 or 15 for k shows that the retention capacity of sessions for robots stabilize. This also supports our discussion earlier about the optimum number of k we should select.

5.6.3 Ranges and robots

Looking in the data, we also found that many of the cluster ranges are retained over the changing cluster sizes from 10 to 15 to 20, which is interesting as we also saw that those cluster ranges which have maximum robots were consistent in retaining the range when

cluster size was increased from 10 to 15 to 20. We also saw that most of the range's lower limit were retained irrespective of the cluster size change from 5 to 20.

Now comes the interesting part, if the range remains the same for cluster sizes ranging from 10 to 20, then the sessions with robots in those clusters should also remain the same but, as we saw in in the raw data(Note: I have data with robot counts for all servers and for all cluster sizes), it is not so.

Another interesting fact is that the session with robots have a range which covers the minimum and maximum for that cluster size, even then when cluster size is increased, some of those robot sessions disperse in different clusters. We also saw that in spite of this fact of dispersion, same cluster contained majority of the robots session in any cluster size.

Chapter 6

Conclusion

The session characterization has been done in past as already discussed but the use of Principal component analysis to analyze the quality of cluster being formed has not been explored. Earlier studies have concentrated upon user access navigation, while this thesis analyzes the intra-session parameters based on raw web log variables. We analyzed the intra-session parameters and then tried to find out how Principal component analysis helps in the clustering process. Our main aim for using Principal component analysis was to clean the data for clustering rather handle the dimensionality problem as we dealt with at the most four parameters. It is safe for us to say that Principal component analysis helps in "bettering" the sample data sets we chose. Some of our concentration has been on finding out how the HTTP error behaves and how it affects the normal clustering process. We have done some preliminary analysis on the behavior of robot sessions to characterize them.

We found that sessions change their cluster-membership as the cluster numbers are increased as it is obvious, but this change in cluster membership does not always follow the same pattern. Parameter with a high scale value like total bytes transferred dominates the clustering behavior by acting as one of the major variable to influence the data point alignment in clusters.

Number of requests per session and bytes transferred are closely related, also number of requests per session and Session length are closely related.

The Private NASA servers does not show a decreasing trend for β_{cv}. This might be because of the low number of data points in those servers. Cluster size of 10 to 15 seem to

give bet results in almost every study done in this thesis. A small cluster size is good for data sets with less number of data points. Higher cluster size tend to break the pattern in the data set which ultimately leads to a poor analysis.

The final conclusion we came up with is that to certain degree of confidence Principal Component Analysis definitely improves the quality of clusters. Clustering helps in unsupervised learning of behavior of data sets and is beneficial in finding out the characteristics of intra-session parameters. Number of errors in session is closely related to the number of requests per session but the relationship is not so strong as compared to the relationship between number of requests and session length.

Robots always exist in clusters with the largest range. This is because the cluster with largest range and eventually with lowest centroid values will also have the largest amount of data points in them, as provided in the table 5.5 on page 99. It also seems that Principal component analysis reduces the average intra-cluster distance and increased the inter-cluster distance hence increasing the quality of clusters.

As the number of clusters are increased the maximum robots session value also decrease. This rate of decrease though tend to stabilize once the cluster size reaches 15.

This thesis tries to cover most of the required exercises for analysis purposes, but as happens in every research, there is always gaps and holes to improve the work. Experiment with finer granularity for number of clusters will help selecting the optimal cluster size i.e. value of k. Validity and quality attributes can be studied further to make changes according to the data set they are applied to.

We also think that different clustering algorithms can be used to overcome some of the issues with K-Means clustering such as local minima problem and anomaly retaining capacity.

Appendix A

Table of Errors

Status Codes	Meaning
400	`Bad Request` Syntax of the request is wrong. Do not request again without modifying the request
401	`Unauthorized access` User Authentication is required, when the authorization requirements are provided by the client in the first place, this error code represents wrong credentials. In Apache access-logs, this is not exactly true. Apache documentation states that a 401 is generated as soon as a client requests a authorized page. Once it gets denied, another 401 is generated. We think that when somebody tries to access a secured page, that is not accessible, instead of showing the page Apache logs a 401 message and provides the user with the option of passing the required credentials. This again, if provided wrong creates another 401 message log
402	`Payment is required` Not used now. Its reserved for future use.
403	`Forbidden` There is no problem with the request, the server doesnt want the client to access the resource. For anonymity purposes, if server doesnt want to return reason for refusal, it should use 404 messages instead.
404	`File Not Found` No matching URI was found. If none of the error messages are applicable, this one is used instead, also when server wants to conceal the reason for not letting the access to the client.
405	`Method Not Allowed` The method requested is not allowed by the resource on which it is requested upon. Response should specify what are the allowable methods for the requested resource.
406	`Not Acceptable` Content characteristics are not acceptable by the accept headers sent in the request.
407	`Proxy Authentication Required` Similar to 401, but client should have authorization with a proxy.
408	`Request Timeout` Server wait time has expired before client could initiate request.
409	`Resource conflict` Request is incomplete because of a conflict in the current state of the resource requested.
410	`Resource not available` A permanent condition where the requested resource is not available at the server.
411	`Length required` The specified content length is required in the request content-length header field.
412	`Precondition failed` The precondition for the specified resource fails when evaluated on the server. This is requested by the client for getting only specific resource based on the precondition.
413	`Request entity too large` Server may close the connection to prevent further requests. In this case, the request entity is large than that understood by the server.
414	`Request URI Too Long` Request URI is longer then acceptable. A rare condition when client mistakes a GET and sends POST instead, URI-Black hole of redirection to itself (continuous loop), also when attacked, where server is using fixed length buffers for reading and manipulating the request-URI.
415	`Unsupported Media Type` Request format not supported.
416	`Requested Range Not Satisfiable` When request had Range-request header field defined and not in accordance with that of the request ed resources extent.
417	`Expectation Failed` Expectation given by the request in an Expect Request-Header field is not met by the server. Also when the server is proxy and server has knowledge (unambiguous) that the request could not be met by the next-hop server.
500	`Internal Server Error` Server had encountered some unexpected condition due to which it could not complete the request.
501	`Not Implemented` When requested method is not recognized by the sever and is not capable of supporting it for any resource it has.
502	`Bad Gateway` While acting as gateway or proxy, the server got a invalid response from the up server, it generates this error code.
503	`Service Unavailable` Temp overloading or maintenance of server.
504	`Gateway Timeout` In case of a gateway or a proxy server, if it did not receive timely response from the upstream server in order to complete the request.
505	`HTTP Version Not Supported` This version of HTTP is not supported, i.e. sometimes a server in not configured to accept request with `HTTP/1.1`

References

[1] Goseva-Popstojanova Katerina, Mazimdar Sunil, and Singh Ajay Deep, "Empirical study of session-based workload and reliability for web servers.," in *ISSRE*, 2004, pp. 403–414.

[2] M. Arlitt, "Characterizing web user sessions," *In Proceedings of the Performance and Architecture of Web Servers Workshop*, June 2000.

[3] Almeida Virgìlio, Bestavros Azer, Crovella Mark, and Oliveira Adriana de, "Characterizing reference locality in the www," in *DIS '96: Proceedings of the fourth international conference on on Parallel and distributed information systems*, Washington, DC, USA, 1996, pp. 92–107, IEEE Computer Society.

[4] Arlitt Martin and Jin Tai, "Workload characterization of the 1998 world cup web site," Tech. Rep., Internet Systems and Applications Laboratory, HP Laboratories Palo Alto), Palo Alto, CA 94304, USA, September 1999.

[5] Arlitt M. and Williamson C., "Web server workload characterization: The search for invariants," in *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems,*, Philadelphia, PA, USA, May 1996, pp. 126–137.

[6] Barford P., "Web server performance analysis," Tutorial at ACM SIGMETRICS, May 1999.

[7] Breslau Lee, Cao Pei, Fan Li, Phillips Graham, and Shenker Scott, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM (1)*, 1999, pp. 126–134.

[8] Gama G., Meira Jr. W., Carvalho M., Guedes D., and Almeida. V., "Resource placement in distributed e-commerce servers," *Evolving Global Communications Network (GLOBECOM 2001)*, vol. 3, November 2001.

[9] Arlitt Martin, Krishnamurthy Diwakar, and Rolia Jerry, "Characterizing the scalability of a large web-based shopping system," *ACM Trans. Inter. Tech.*, vol. 1, no. 1, pp. 44–69, 2001.

[10] Daniel A. Menascè;, Virgìlio A. F. Almeida, Rodrigo Fonseca, and Marco A. Mendes, "A methodology for workload characterization of e-commerce sites," in *EC '99: Proceedings*

of the 1st ACM conference on Electronic commerce, New York, NY, USA, 1999, pp. 119–128, ACM Press.

[11] Menascè; Daniel, Almeida Virgìlio, Riedi Rudolf, Ribeiro Flàvia, Fonseca Rodrigo, and Meira Jr. Wagner, "In search of invariants for e-business workloads," in *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, New York, NY, USA, 2000, pp. 56–65, ACM Press.

[12] Shi W., Wright R., Collins E., and Karamcheti V., "Workload characterization of a personalized web site — and it's implication on dynamic content caching," 2002.

[13] Chen H. and Mohapatra P., "Session-based overload control in qos-aware web servers," Proceedings of IEEE INFOCOM, 2002.

[14] Daniel A. Menascé;, Virgìlio A. F. Almeida, Rodrigo Fonseca, and Marco A. Mendes, "Business-oriented resource management policies for e-commerce servers," *Perform. Eval.*, vol. 42, no. 2-3, pp. 223–239, 2000.

[15] Arnoux Mireille, Lechevallier Yves, Tanasa Doru, Trousse Brigitte, and Verde Rosana, "Automatic clustering for the web usage mining," in 5^{th} *INt. Workshop on Symbolic and Numeric Computer Science*, 06902 Sophia Antipolis, France, 2003.

[16] Fu Y., Sandhu K., Shih M., and M. Asho, "Generalization-based approach to clustering of web usage sessions," in *Proceedings of WEBKDD 1999*, San Diego, LA, USA, August 1999, pp. 21–28.

[17] Heer Jeffery and Chi Ed H., "Mining the structure of user actvity using cluster stability," Tech. Rep., PARC(Palo Alto Research Center), Palo Alto, CA 94304, USA, 2005.

[18] A. Banerjee and J. Ghosh, "Clickstream clustering using weighted longest common subsequences," Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago, April 2001., 2001.

[19] Weinan Wang and Osmar R. Zaïane, "Clustering web sessions by sequence alignment," in *DEXA '02: Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, Washington, DC, USA, 2002, pp. 394–398, IEEE Computer Society.

[20] Heer Jeffery and Chi Ed H., "Seperating the swarm: Categorization methods for user sessions on the web," in *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, Minneapolis, MN, USA, April 2002, pp. 51–58, ACM Press.

[21] Chi Ed H., Rosien Adam, and Heer Jeffrey, "Lumberjack: Intelligent discovery and analysis of web user traffic composition," Proc. of ACMSIGKDD Workshop on Web Mining for Usage Patterns and User Profiles, ACM Press, Canada., 2002.

[22] Daniel A. Menascé;, Virgìlio A. F. Almeida, Rudolf H. Riedi, Flavia Ribeiro, Rodrigo C. Fonseca, and Wagner Meira Jr., "In search of invariants for e-business workloads," *ACM : Conference on Electronic Commerce*, pp. 56–65, 2000.

[23] Anupam Joshi and Raghu Krishnapuram, "On mining web access logs," in *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2000, pp. 63–69.

[24] Jan Larsen, Lars Kai Hansen, Anna Szymkowiak Have, Torben Christiansen, and Thomas Kolenda, "Webmining: learning from the world wide web," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 517–532, 2002.

[25] Heer Jeffery and Chi Ed H., "Identification of web user traffic composition using multi-modal clustering and information scent," in *Proceedings of the Workshop on Web Mining, SIAM Conference on Data Mining*, Chicago, IL, USA, April 2001, pp. 51–58.

[26] Calvin Ko Karl Levitt Dusting Lee, Jeff Rowe, "Detecting and defending against web-server fingerprinting," *Proceedings of the 18th Annual Computer Security Applications Conference(ACSAC'02)*, 2002, Computer Security Laboratory, University of California, Davis.

[27] Zhang T., Ramakrishnan R., and Livny M., "Web server performance analysis," Data Mining and Knowledge Discovery, 1997.

[28] Shahabi C., Zarkesh A. M., Adibi J., and Shah V., "Knowledge discovery from users web-page navigation," in *RIDE '97: Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications*, Washington, DC, USA, 1997, p. 20, IEEE Computer Society.

[29] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan, "Web usgae mining: Discovery and applications of usage patterns from web data," *SIGKDD Explorations*, vol. 1, no. Issue 2, pp. 12–23, January 2000.

[30] Mathworks, "Principal components analysis," .

[31] Raj K. Jain, *The Art of Computer Systems Performance Analysis : Techniques for Experimental design, Measurement, Simulation and Modelling*, Wiley, April 1991.

[32] James E Harner, "Professor and chair, department of statistics, west virginia university," 2005.

[33] Anil K Jain, Richard C Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.

[34] Vance Faber, "Clustering and the continuous k-means algorithm," *Los Alamos Science*, , no. 22, pp. 138–144, 1994.

[35] Guyon Isabelle." "Ben-Hur, Asa., *"Functional Genomics - methods and protocols (Detecting Stable Clusters Using Principal Component Analysis)"*, vol. "224", "Humana Press", "March" "2003".

[36] Hartigan J A and Wong M A, "A k-means clustering algorithm," *Applied statistics*, , no. 28, pp. 100–108, 1979.

[37] R Documentation, "A details description in r documentation," .

[38] Apache, "http://httpd.apache.org/docs-2.0/logs.html," .