CHAPTER III

PROPOSED FRAMEWORK

This chapter contains two parts. The first part explains the preliminary study concerning user behavior analysis, exploring different usage behaviors for each server service, and demonstrating workload distribution. The second part focuses on consumption of hardware resources to propose an appropriate number of CPUs and memory units allocation for the system.

3.1 User Behavior Analysis

User behavior is applied with a visual data mining technique in two scenarios: user behaviors in accessing the server and user behaviors on the required data size from servers. The association rule is employed to predict user behavior for each type of workload service and time.

In the first step, data from proxy servers and web servers are collected. Figure 3.1 shows sample proxy server data. Figure 3.2 shows sample web server data. Typically, these access log files contain millions of records. Each record refers to a visit by a user to a certain web page served by the web and proxy servers. Data set to be used in this study were collected over two month periods, i.e., the log file run from 00:00:00 November 1 to 23:59:59 December 30 [38].

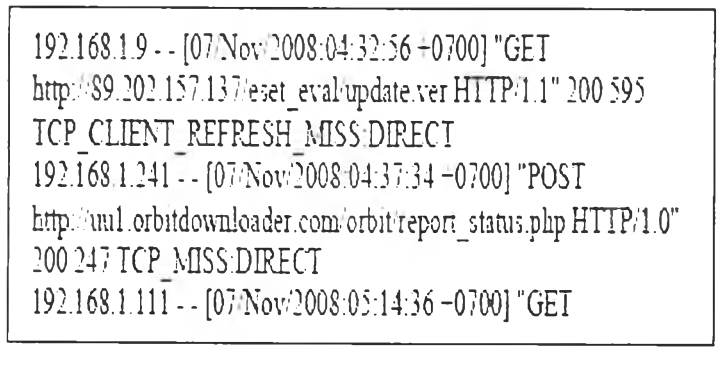

Figure 3.1: Example log file from proxy server

```
66.249.71.38 - - [30/Nov/2008:08:01:26 -0800] "GET /search.php
HTTP/1.1" 200 18676 "-" "Mozilla/5.0 (compatible;
Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.71.36 - - [30/Nov/2008:08:12:12 -0800] "GET /articles.php
HTTP/1.1" 200 18736 "-" "Mozilla/5.0 (compatible;
Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.71.37 - - [30/Nov/2008:08:53:31 -0800] "GET /robots.txt
HTTP/1.1" 404 327 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
```

Figure 3.2: Example log file from web server

The original format of data in server's log files were unsuitable for mining purpose. For this reason, a preprocessing step had to performed before the start of pattern discovery phase.

In the preprocessing step, data are selected to be used in the analysis and elimination of unwanted parts. After the preprocessing step, the information from server log will participates in the following calculations:

1) The number of user accesses in proxy servers and web servers are counted to determine the frequencies of days of the week and associate time, and

2) data size in proxy servers and web servers are calculated for each time period.

Table 3.1: Examples of user access to the proxy at 10:00 AM

	date	day	access
1	01/Nov/2008	Saturday	4496
2	02/Nov/2008	Sunday	591
3	03/Nov/2008	Monday	11398
4	04/Nov/2008	Tuesday	16094
5	05/Nov/2008	Wednesday	17428
6	06/Nov/2008	Thursday	54751
7	07/Nov/2008	Friday	24251
8	08/Nov/2008	Saturday	4485
9	09/Nov/2008	Sunday	255
10	10/Nov/2008	Monday	20334
11	11/Nov/2008	Tuesday	35897
12	12/Nov/2008	Wednesday	2786
13	13/Nov/2008	Thursday	1132
14	14/Nov/2008	Friday	13361
15	15/Nov/2008	Saturday	2263
16	16/Nov/2008	Sunday	5674
17	17/Nov/2008	Monday	10892
18	18/Nov/2008	Tuesday	13066
19	19/Nov/2008	Wednesday	19154
20	20/Nov/2008	Thursday	12610
21	21/Nov/2008	Friday	30621
22	22/Nov/2008	Saturday	8415
23	23/Nov/2008	Sunday	6280
24	24/Nov/2008	Monday	29769
25	25/Nov/2008	Tuesday	36077
26	26/Nov/2008	Wednesday	21785
27	27/Nov/2008	Thursday	25441
28	28/Nov/2008	Friday	19602
29	29/Nov/2008	Saturday	5769
30	30/Nov/2008	Sunday	7451

Table 3.2 : Examples of data size in the proxy at 10:00 AM

	date	day	datasize
1	01/Nov/2008	Saturday	30950855
2	02/Nov/2008	Sunday	1418847
3	03/Nov/2008	Monday	125886186
4	04/Nov/2008	Tuesday	127058844
5	05/Nov/2008	Wednesday	149523626
6	06/Nov/2008	Thursday	415356901
7	07/Nov/2008	Friday	403072403
8	08/Nov/2008	Saturday	20685033
9	09/Nov/2008	Sunday	3847972
10	10/Nov/2008	Monday	205890833
11	11/Nov/2008	Tuesday	444676852
12	12/Nov/2008	Wednesday	45661153
13	13/Nov/2008	Thursday	14877427
14	14/Nov/2008	Friday	276960334
15	15/Nov/2008	Saturday	19036607
16	16/Nov/2008	Sunday	68706292
17	17/Nov/2008	Monday	103866498
18	18/Nov/2008	Tuesday	214861431
19	19/Nov/2008	Wednesday	158531124
20	20/Nov/2008	Thursday	104668978
21	21/Nov/2008	Friday	373925535
22	22/Nov/2008	Saturday	47708838
23	23/Nov/2008	Sunday	30773571
24	24/Nov/2008	Monday	269297093
25	25/Nov/2008	Tuesday	321460936
26	26/Nov/2008	Wednesday	211576159
27	27/Nov/2008	Thursday	164382886
28	28/Nov/2008	Friday	232745432
29	29/Nov/2008	Saturday	49640746
30	30/Nov/2008	Sunday	62501883

Table 3.1 shows the examples of user access to proxy server during 01 Nov – 30 Nov at 10:00A.M. Table 3.2 shows the examples of data size (in byte) required by users in the proxy server for the same period. Both period of day, time, and data size required by the users in proxy and web servers were calculated. The average number of user accesses in the proxy server and web server for each period of day and time were plot accordingly.

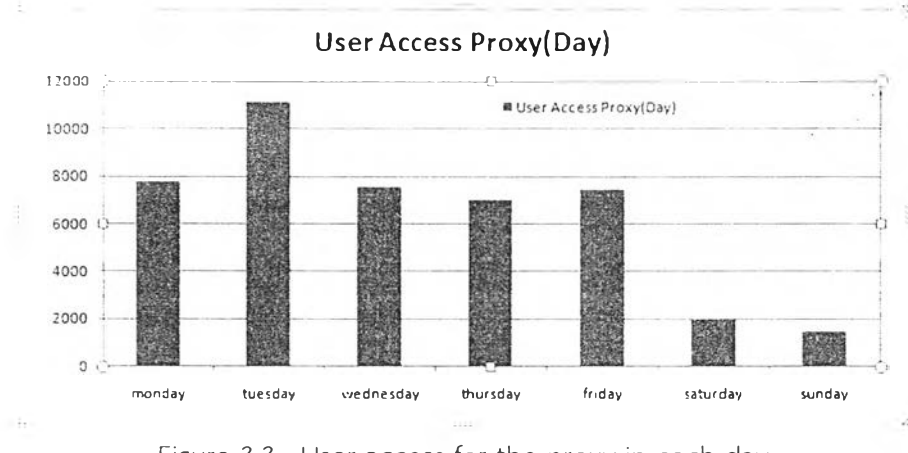

Figure 3.3 : User access for the proxy in each day.

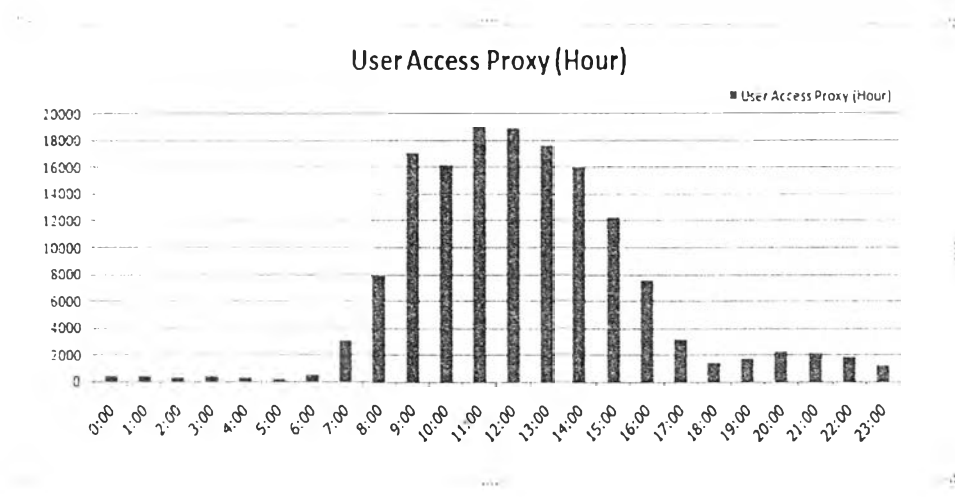

Figure 3.4 : User access for the proxy in each hour.

Figure 3.3 shows user access behavior for the proxy server in each day. The accesses are lower on weekends than weekdays. In particular, Tuesday seems to have the highest access. Figure 3.4 shows user access behavior of the proxy server. It can be seen that during 07.00 to 16.00, user accesses are more frequent than other times and during 11.00 – 12.00 has the highest level of user access in proxy server.

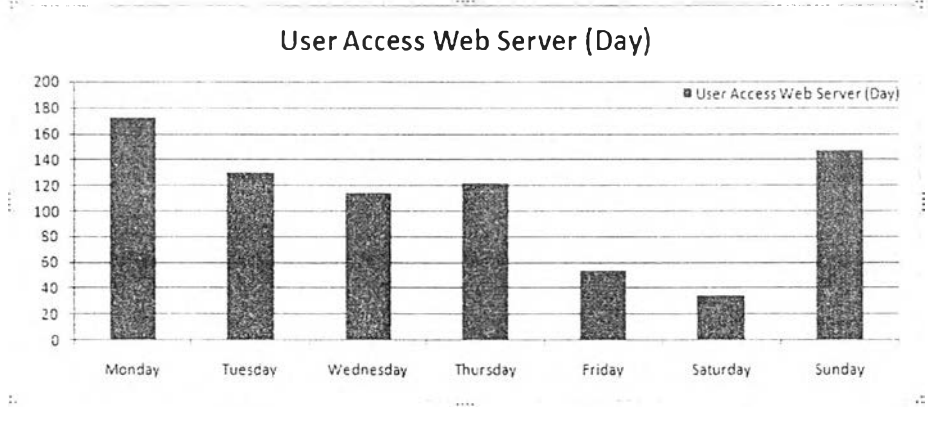

Figure 3.5 : User access for the web server in each day.

Figure 3.5 shows user access behavior for the web server in each day. The accesses are the lowest on Saturday than the rest of the week. Note that Monday has the highest access.

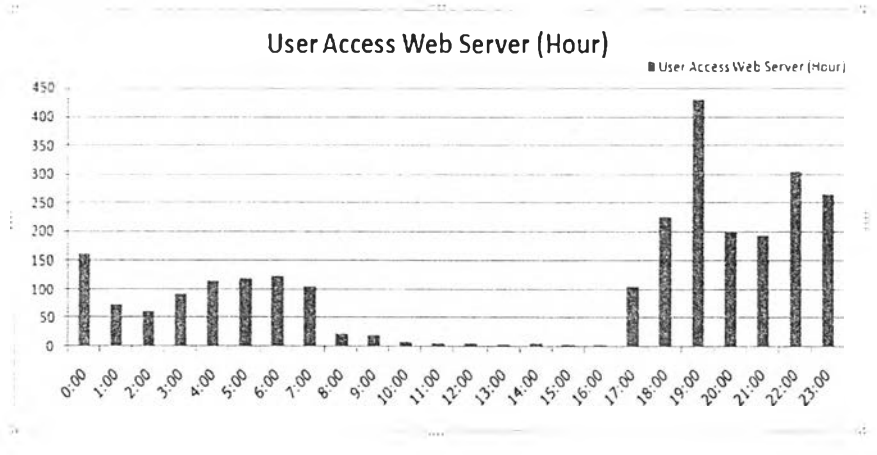

Figure 3.6 : User access for the web server in each hour.

Figure 3.6 shows user access behavior for the web server. It can be seen that during 08.00 to 16.00, user accesses are less frequent than other times. However, according to the graph in Figure 3.7, the access is peak at 19.00.

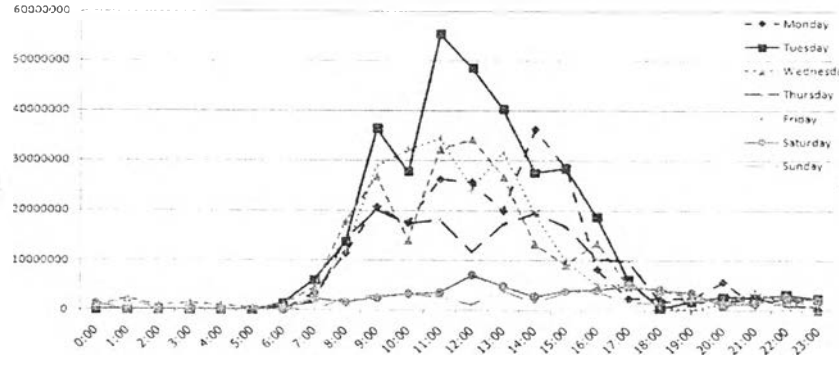

Figure 3.7 : Workload for the proxy server.

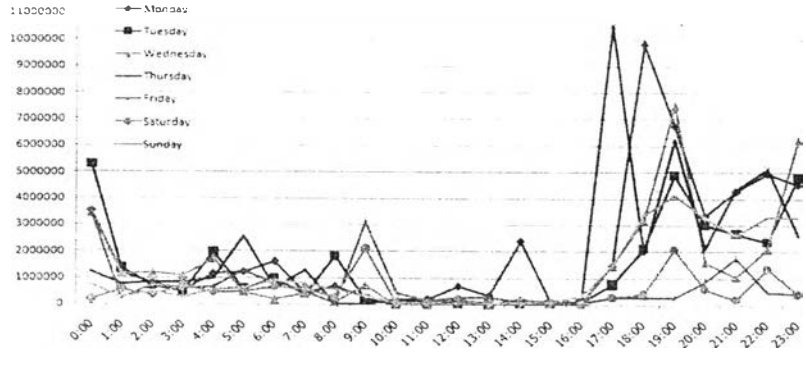

Figure 3.8 : Workload for the web server.

In Figures 3.7 and 3.8, days of the week and time are plotted against the data sizes requested by the users to proxy and web servers, respectively. From Fig. 3.7, it can be seen that during the period 07.00 to 16.00, user data requirements are more frequent than other time for each day of the week. Nevertheless, the requirements are lower on weekend. Tuesday seems to have the highest data size requested during 07.00-16.00. In Fig. 3.8, the requirements during 10.00 to 16.00 are less than other time, but after 16.00, more data are requested. It can be seen that the data requirements peak on Thursday during 17.00 to 19.00.

Based on the above data, association rules for predicting user behaviors in each server is applied [39].

From preprocessing step, the number of user accesses in proxy servers and web servers and, the frequencies of both the day of the week and the time are computed according to step (1) and (2) given earlier. Then, levels of access are categorized into 5 levels, namely '1' for low level; '2' for medium low level; '3' for medium level; '4' for medium high level; and '5' for high level. Here, the levels of access are assumed to be uniformly distributed.

3.1.1 Association Model for Analyzing User Behavior

The relationship in the form of the left-hand side to the right-hand side *(LHS → RHS)* is applied for extracting rules. The extracted rules for *LHS* are based on daily 1-hour period.

Let *D1, D2, ..., D7* be days of week and *T1,T2, ...,T24* be times of the day. However, the *RHS* is restricted as follows. Let *L1, L2, L3, L4, L5* be the levels of user access for the *RHS* that can be predicted based on the term on the *LHS*. Therefore, a rule *(Di, Tj) → Lk* is created, where *Lk* occurs most frequently in the rows.

For each rule of the form *LHS → RHS*, define the *supp* and *conf* as follows:

$$conf(LHS, RHS) = \frac{count(LHS, RHS)}{count(LHS)} \qquad (3.1)$$

Such as $conf(day, time \rightarrow level) = \frac{count(day, time \text{ and } level)}{count(day, time)}$ (3.2)

$$\sup(LHS, RHS) = \frac{count(LHS, RHS)}{count(All)} \quad (3.3)$$

Such as

$$\sup(day, time \rightarrow level) \quad \sup(\text{day, time} \rightarrow \text{level}) = \frac{count(day, time \ and \ level)}{count(All)} \quad (3.4)$$

Table 3.3 shows examples of rules for predicting the access levels on Monday and Tuesday at 10:00 AM. Confidence and support value are used for rule selections. Because plenty of rules are generated, some rule selections criteria are established:

1) Select the rule with maximum confidence.

2) Select the rule with maximum support if confidence value is equal.

3) Select the rule that happens first when confidence and support values are equal.

Table 3.3 : Examples of rules for prediction

rule	Conf (%)	Sup (%)
Monday , 10:00 AM → Low	50	0.28
Monday , 10:00 AM → Medium Low	25	0.14
Monday , 10:00 AM → Low	50	0.28
Monday , 10:00 AM → Medium	25	0.14
Tuesday , 10:00 AM → Medium Low	25	0.14
Tuesday, 10:00 AM → Medium	50	0.28
Tuesday, 10:00 AM → Low	25	0.14
Tuesday, 10:00 AM → Medium	50	0.28

From Table 3.3, some prediction rules include level of user access from the proxy server on Monday at 10:00 AM is "Low", while level of user access on Tuesday at 10:00 AM is "Medium."

Table 3.4 : Prediction model of proxy server

No.	rule	Conf (%)	Sup (%)
1	Monday ,12:00 AM → Low	100	0.56
2	Monday ,01:00 AM → Low	100	0.56
...	.....	...	...
10	Monday ,09:00 AM → Medium Low	75	0.42
11	Monday ,10:00 AM → Low	50	0.28
12	Monday ,11:00 AM → Medium Low	100	0.56
13	Monday ,12:00 PM → Medium Low	100	0.56
14	Monday ,13:00 PM → Low	100	0.56
15	Monday ,14:00 PM → Medium Low	50	0.28
16	Monday ,15:00 PM → Medium Low	50	0.28
...	.....	...	...
168	Sunday ,23:00 PM → Low	100	0.69

Table 3.4 shows total association prediction model for proxy server with confidence and support values.

3.1.2 Correlation between User Access and Workload

It is interesting to know how much the two variables, user access and their data size requirements in the server, are correlated. A simple linear correlation is employed for the explorations.

Let "x" be defined as the number of user accesses (independent variable) and "y" as data size requirements (dependent variable). A simple linear correlation equation is in the form $y = a + bx$, where a, b are calculated from the following equations:

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \quad (3.5)$$

$$a = \bar{y} - b\bar{x} \quad (3.6)$$

when x and y are the average value of x and y, respectively.

The correlation coefficient measures the strength and direction of the linear relation between two variables. The correlation coefficient can be computed by the following formula:

$$R = \frac{n[\sum_{i=1}^{n}(x_i y_i)] - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n}(x_i)^2 - (\sum_{i=1}^{n} x_i)^2}\ \sqrt{n\sum_{i=1}^{n}(y_i)^2 - (\sum_{i=1}^{n} y_i)^2}} \qquad (3.7)$$

R^2 (coefficient of determination) denotes the strength of the linear association between x and y. In other words, it represents the percentage of data that are the closet to the line of best fit. For example, R^2 =0.986 means that 98.6 % of the total variation in Y can be explained by the linear relation between x and y. The coefficients (a,b) and R^2 of the relationship between user access and data size for each day of the week can be displayed as in Table 3.5.

Table 3.5: The coefficients of the correlations between user access and data size for each day of the week

Day	a	b	R^2
Monday	-554.008	11.576	0.986
Tuesday	319.220	12.065	0.973
Wednesday	-2163.298	12.360	0.978
Thursday	5193.521	9.269	0.909
Friday	-5318.320	12.832	0.970
Saturday	4544.345	9.573	0.738
Sunday	2534.031	8.933	0.647

From Table 3.5, the values of R^2 range from 64.7% to 98.6%. For 5 days of the week, R^2 values are higher than 90%. This implies that the regression line represents the data very well. In other words, the linear relation is a good representation of the relationship between the number of user access and data size requirements.

3.1.3 Workload Distribution

In managing heterogeneous workloads in virtualized systems, it is necessary to know the distribution of the workload for every server service. In this research, data logs were collected from two servers to find the distribution. First, data size was descendingly sorted and plotted. Figures 3.9 and 3.10 show the data size required by users in the proxy server and the web server. The y-axis of both graphs denotes user-required data sizes, whereas the x-axis represents the frequency of the required data sizes. In the proxy server, there are a number of data sizes over 200,000 byte (200KB), but most of the sizes required by users are less than 200,000 bytes (200KB). For the web server, there are a number of dat sizes over 50,000 bytes (50KB), but most of the sizes required by users are less than 50,000 bytes (50KB).

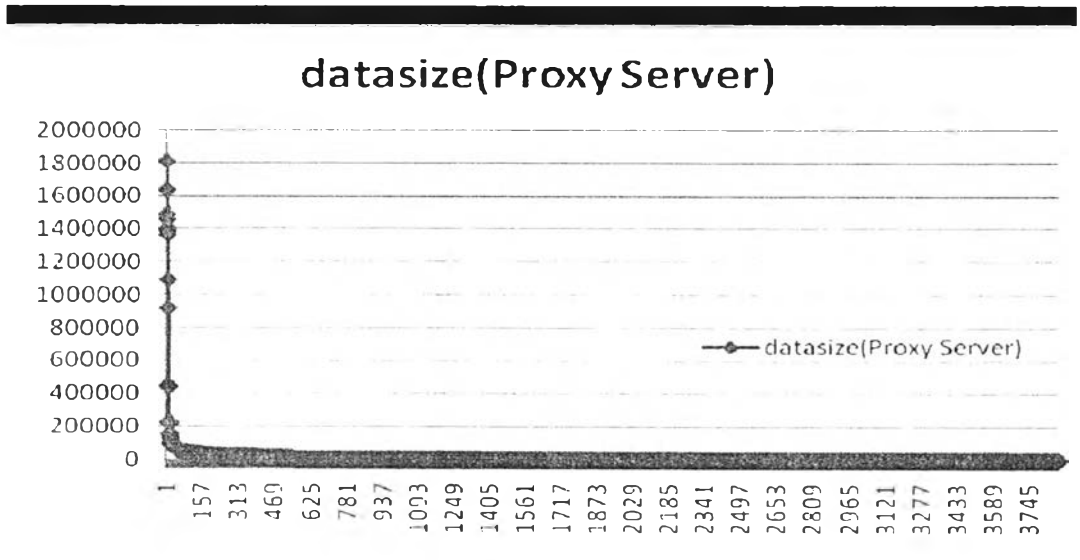

Figure 3.9: Data size (sorted by descending size) in Proxy server over 24 hour period.

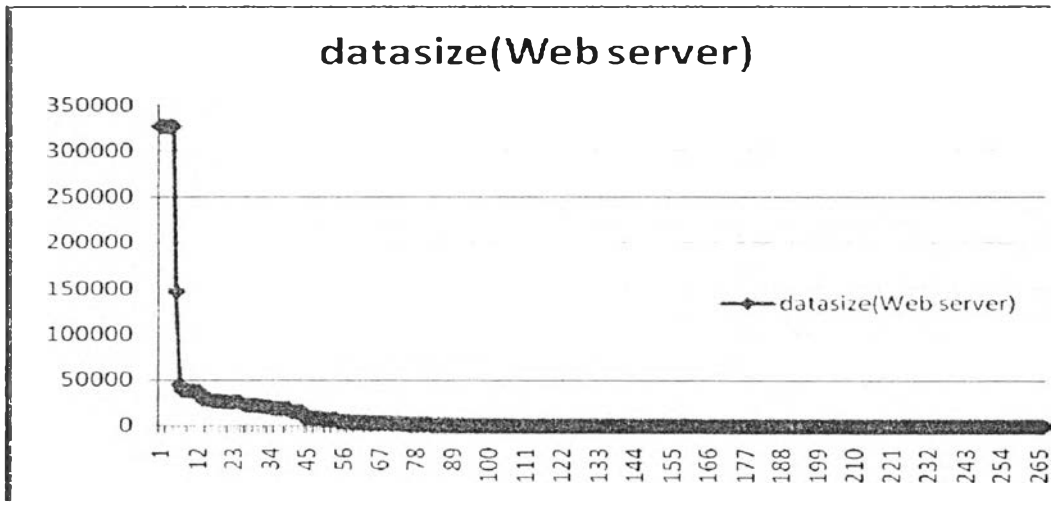

Figure 3.10: Data size (sorted by descending size) in Web server over 24 hour period.

There are two parameters in Pareto distribution: $\hat{\alpha}$ and $\hat{x}_m$ as shown in Formula (2.6) and (2.8) that are used to estimate the values of the cumulative distribution function in Formula (2.3).

Figure 3.11 shows the data sizes in proxy server over a 24 hour period (cumulative percent) and Figure 3.12 shows the Pareto distributive function generated using the above Pareto distribution equations on the real data. It can be seen that both graphs from Figures 3.11 and 3.12 are similar.

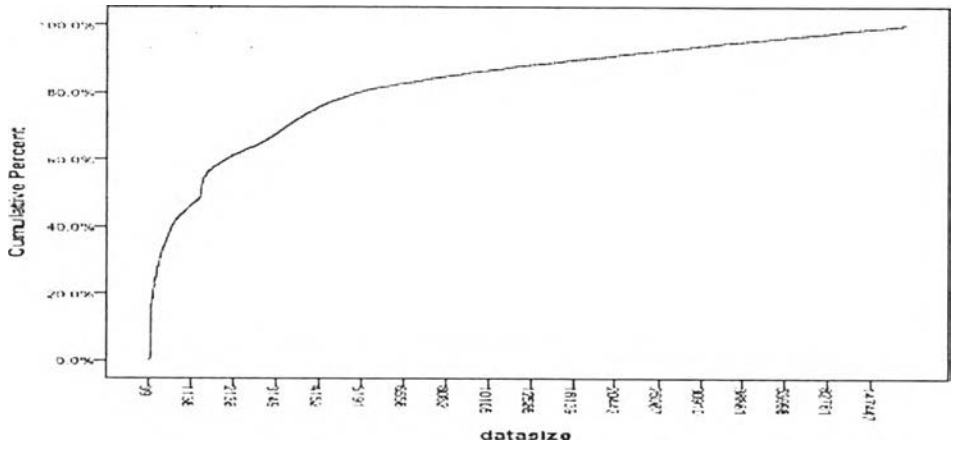

Figure 3.11: Data size in proxy server over a 24 hour period (cumulative percentage).

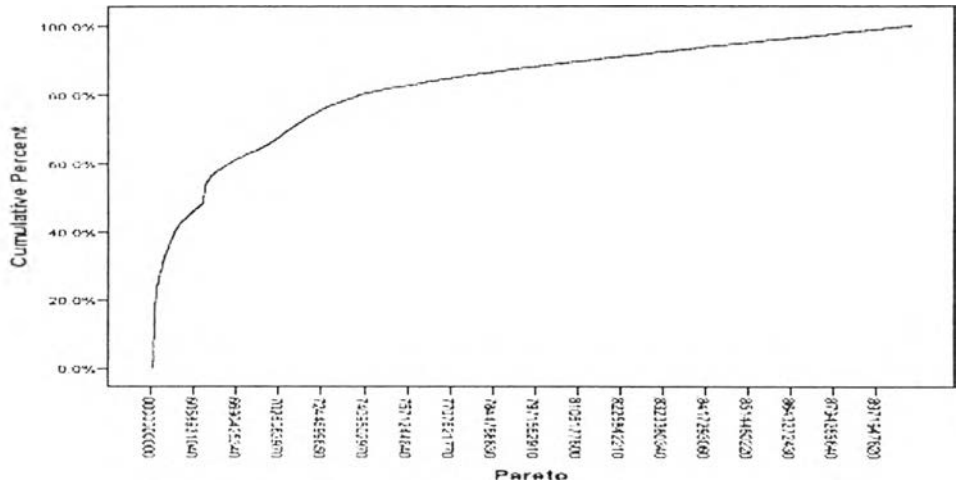

Figure 3.12: Pareto distribution from proxy server over a 24 hour period (cumulative percentage).

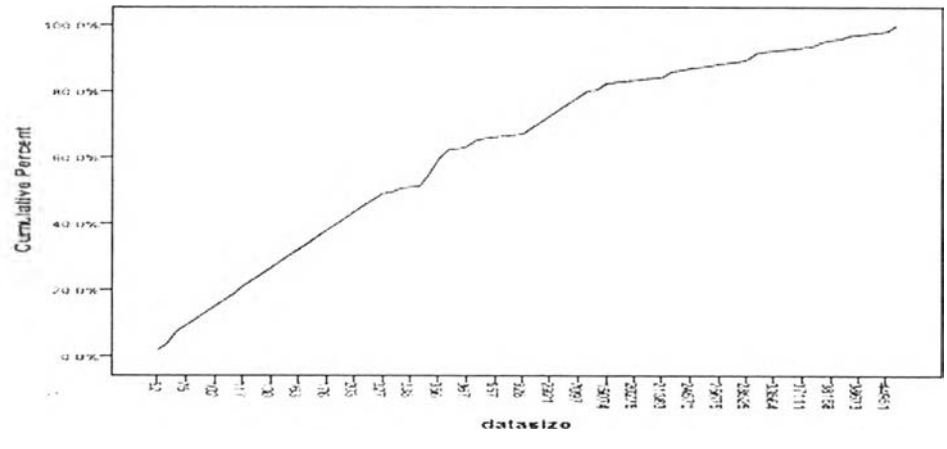

Figure 3.13: Data size in web server over a 24 hour period (cumulative percentage).

Figure 3.13 shows the data size in the web server over a 24 hour period (cumulative percent) and Figure 3.14 shows Pareto distributive function generated using the aforementioned Pareto distribution equations on the real data. It can be seen that both graphs from Figures 3.13 and 3.14 are similar. The concept of server virtualization is to utilize existing resources, as well as the management of heterogeneous workload performance. However, designing schedules to manage different types of workloads in the system, one needs to know the pattern of workload distribution of the system. In fact, the pattern of data size distribution in

the web, including data requested by users, data transmitted through the network, and data stored on servers exhibit a heavy tails distribution [40].

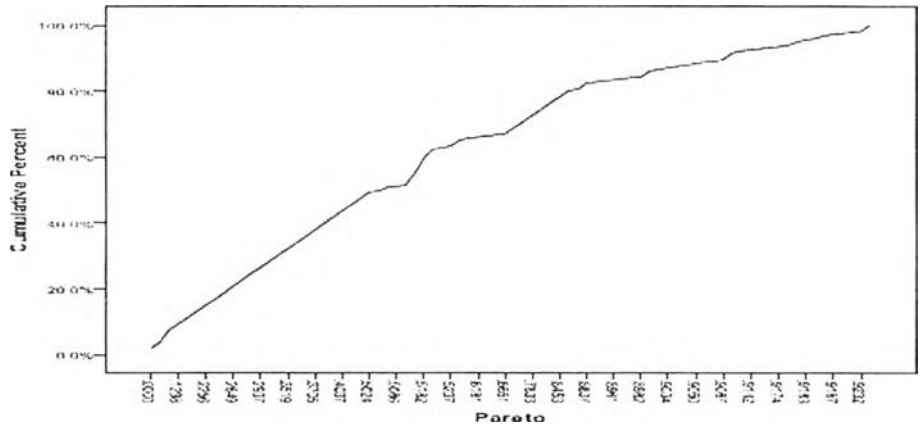

Figure 3.14 Pareto distribution from web server over a 24 hour period (cumulative percentage).

This section of the experiment shows file sizes requested by users in both proxy server and web server and also describe the Pareto distribution. Moreover, improving management workload come into the virtualized system. Understanding user behaviors may help better performance of heterogeneous workload management.

3.2 Consumption of Hardware Resources Analysis

The objective of this step is to explore consumption of hardware resources and to propose an appropriate numbers of CPUs and memory units to be allocated for the system. This will be carried out through simulation runs.

3.2.1 Simulation for Resource Analysis and Prediction

In this section, server virtualization simulation was set for analyzing CPU and memory behavior. This simulation using the same data log file as multi-clients user behavior simulation to send request to servers.

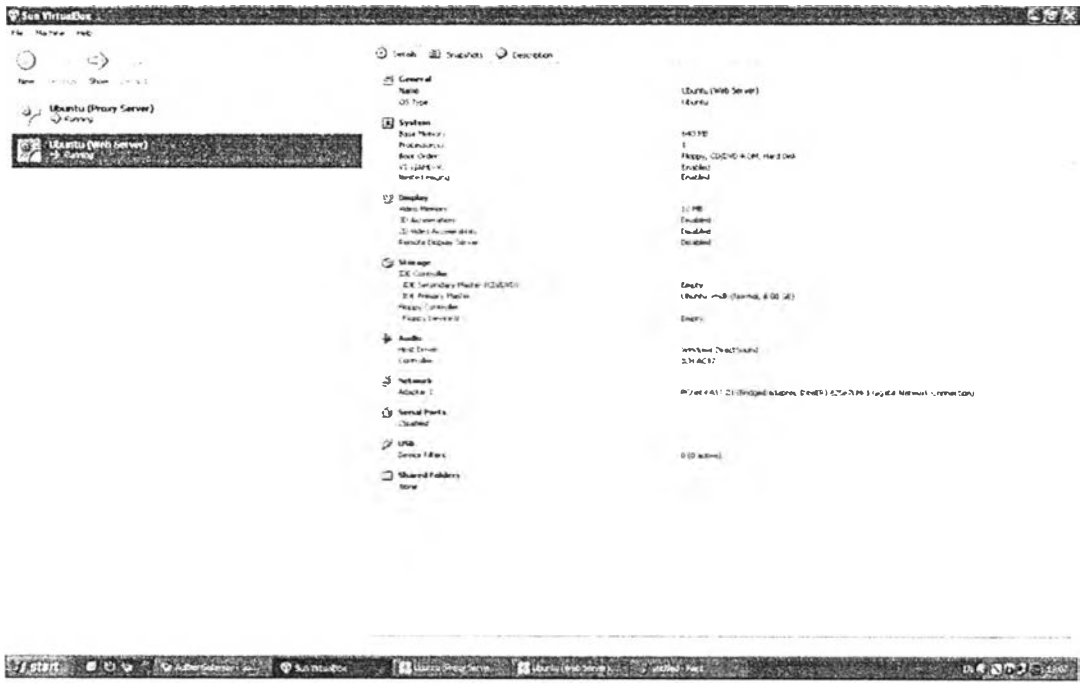

Figure 3.15: Program VirtualBox for simulation server.

Simulation of servers used VirtualBox running on Windows XP as the host operating system as shown in Figure 3.15. Then, two servers, proxy and web server were running on Ubuntu operating systems as shown in Figure3.16. On the client side, the program written in Microsoft Visual C Sharp would simulate multi-clients to request services from the two servers. System resources on the server were monitored, in particular, CPU and memory usage during various periods of time were collected.

Figure 3.16: Set-up server virtualization.

The simulation result shows that consumption of hardware resources is different by type of servers and the periods of time. In Figures 3.17 and 3.18, the CPU and memory consumption in the proxy server barely change during monitor periods.

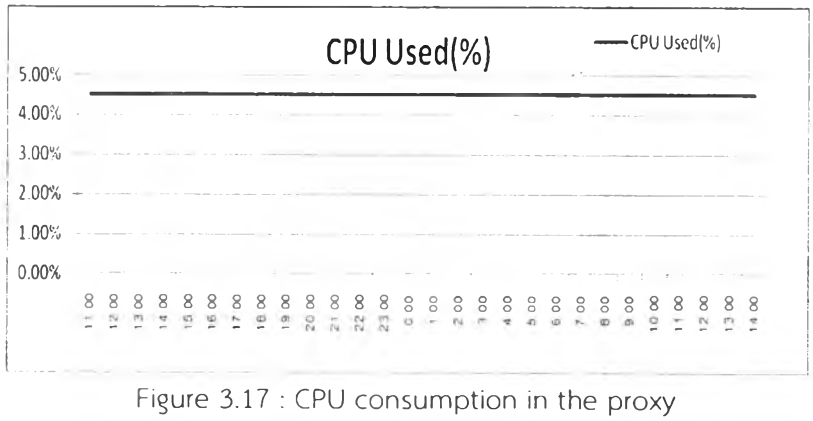

Figure 3.17 : CPU consumption in the proxy

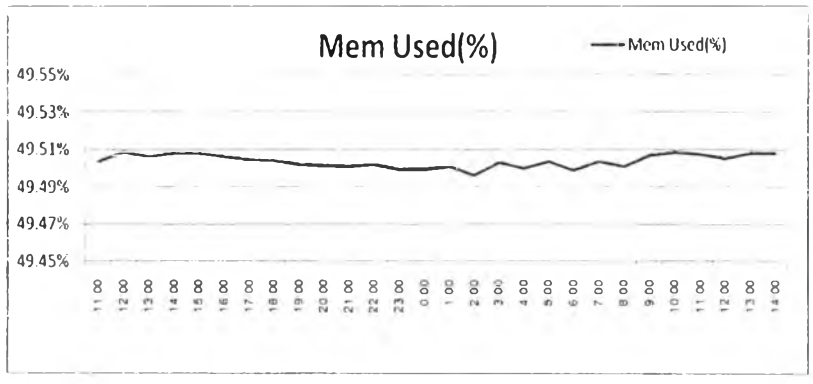

Figure 3.18: Memory consumption in the proxy

Figures 3.19 and 3.20 imply that resource consumption in the web server varies during different periods of time. In addition, the patterns of these two graphs demonstrate seasonal variations.

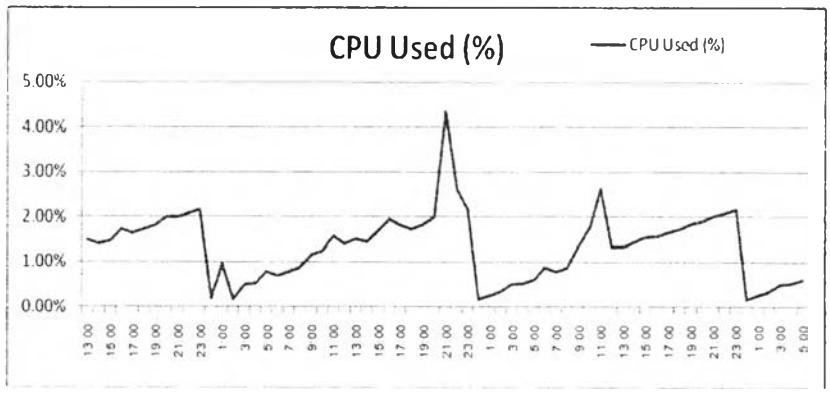

Figure 3.19: CPU consumption in the web server

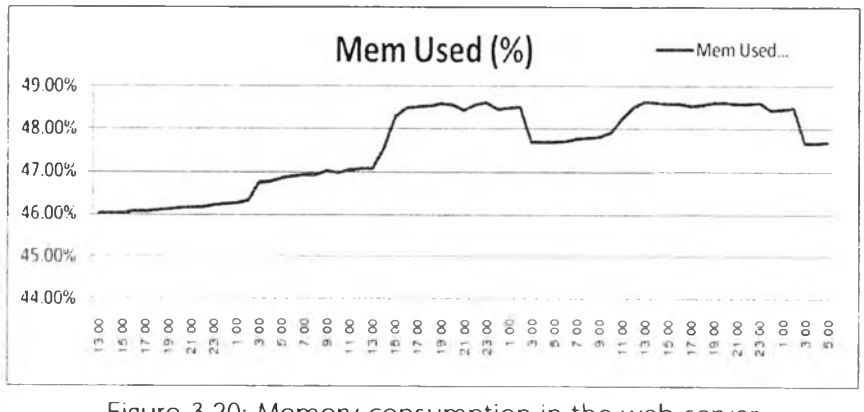

Figure 3.20: Memory consumption in the web server

However, the simulation has a lot of limitations such as the continuity for 24 hours period in the experiment, internet connection, and other factors that can affect CPU and memory usage. Their usage behavior on different servers and how to implement will be described in the next section.

3.2.2 Algorithm for Resources Prediction and Allocation

In this section, real behavior CPU and memory usage on 3 servers from Suan Sunandha Rajabhat University were used in the analysis. The experiment was conducted as above section. However, actual data from servers and boundary allocation policy were set-up. The algorithm consists of two steps. The first step predicts the amount of requested resources by applying double exponential smoothing method. In fact, it is rather difficult to make a precise prediction. In case of imprecision, some processed tasks may be interrupted due to insufficient resources, thereby response time is prolonged. This degrades user satisfaction although utilization may be maximum. The second step adjusts the predicted amount of resources to compromise utilization and system response time. The prediction is performed one hour in advance. The following variables are used in the algorithm. For any hour i, let

$c_{j,k}$: be the percentage of CPU usage at the j^{th} minute time and k^{th} second time interval.

$m_{j,k}$: be the percentage of memory usage at the j^{th} minute time and k^{th} second time interval.

$s_i^{(cpu)}$: be the overall smoothing value at the i^{th} hour. This has the same meaning as s_t in Section 3.2.2. The superscript *(cpu)* denotes that this overall smoothing value is for CPU.

$b_i^{(cpu)}$: be the trend smoothing value at the i^{th} hour. This has the same meaning as b_t in Section 3.2.2. The superscript *(cpu)* denotes that this trend smoothing value is for CPU.

$s_i^{(mem)}$: be the overall smoothing value at the i^{th} hour. This has the same meaning as s_t in Section 3.2.2. The superscript *(mem)* denotes that this overall smoothing value is for memory.

$b_i^{(mem)}$: be the trend smoothing value at the i^{th} hour. This has the same meaning as b_t in Section 3.2.2. The superscript *(mem)* denotes that this trend smoothing value is for memory.

$\bar{m}_i$: mean allocated memory units within the i^{th} hour.

$\bar{c}_i$: mean allocated CPU units within the i^{th} hour.

$\tilde{m}$: predicted amount of requested memory units in the next $(i+1)^{th}$ hour.

$\tilde{c}$: predicted amount of requested CPU units in the next $(i+1)^{th}$ hour.

u : compromising factor of utilization versus user satisfactions, $1 \leq u \leq 100$.

Double exponential smoothing is used in the prediction process because the prediction error of this method is less than that of the simple exponential smoothing. Details of how to predict CPU and memory requests are given in Algorithm 1.

Table 3.5. Algorithm for resource allocation and prediction

Algorithm 1 Predicting and allocating resources

Input : The resources usage in every k period $c_{j,k}$ and $m_{j,k}$.

Output : Resources allocated for next hour $\tilde{m}_i$, $\tilde{c}_i$

Step1: Let $T = 360$.

Step2: for $1 \leq i \leq 24$ do

Step3: Let $\bar{c}_i = \frac{1}{T}(\sum_{j=1}^{60} \sum_{k=1}^{6} c_{j,k})$

Step4: end for

Step5: for $1 \leq i \leq 24$ do

Step6: Let $\bar{m}_i = \frac{1}{T}(\sum_{j=1}^{60} \sum_{k=1}^{6} m_{j,k})$

Step7: end for

Step8: Compute $\tilde{c}_{i+1}$ by double exponential smoothing method as follows.

$$\tilde{c}_{i+1} = \propto \bar{c}_i + (1-\propto)(s_{i-1}^{(cpu)} + b_{i-1}^{(cpu)}) + \gamma(s_i^{(cpu)} - s_{i-1}^{(cpu)}) + (1-\gamma)b_{i-1}^{(cpu)}$$

Step9: Compute $\tilde{m}_{i+1}$ by double exponential smoothing method as follows.

$$\tilde{m}_{i+1} = \propto \tilde{m}_i + (1-\propto)(s_{i-1}^{(mem)} + b_{i-1}^{(mem)}) + \gamma(s_i^{(mem)} - s_{i-1}^{(mem)}) + (1-\gamma)b_{i-1}^{(mem)}$$

Step10: Adjust the prediction $\tilde{c}_{i+1}$ and $\tilde{m}_{i+1}$ by

$$\tilde{c}_{i+1} = \frac{100}{u}\tilde{c}_{i+1}$$

$$\tilde{m}_{i+1} = \frac{100}{u}\tilde{m}_{i+1}$$

3.2.3 Problem Scenario

The problem scenario is described as follows. A server is defined as a collection of homogeneous CPUs and memory units. A user submits a task consisting of a set of processes to the server. Some appropriate numbers of CPUs and memory units are allocated to execute these processes. However, any CPUs and memory units not allocated any processes are idle. If the received task requires more computing, resources, the server will turn on some idle CPUs and memory units to serve the request.

This research focuses on three related essential issues. The first issue concerns the estimation of number of requested CPUs and size of memory in the next hour. The second issue focuses on resource allocation in advance so that the amount of allocated resources is always larger than the requested resources within a defined constant value. The last issue pertains to the relation between maximum resource utilization and user satisfaction in terms of response time. As previously mentioned, the researches considered only how to improve the performance of the system but totally omitted the user satisfactory aspect. In this research, the term *utilization* refers to the state of system with no idle components at any time. Generally, system utilization and user satisfaction are controversial. To make a user satisfy with response time, more resources must be allocated. But more resources imply that some resources may not be fully used throughout the period of time and the energy consumption obviously increases. The problem scenario studied in this research can be formulated as follows.

Let $1 \leq j \leq 60$ and $1 \leq k \leq 6$ be the minute index in one hour and the time interval index of 10 seconds within one minute, respectively. Since there are 60 minutes in one hour, the value of j is between 1 and 60. For each minute,

there are six equal 10-second time intervals. Thus, the value of k is between 1 to 6. At any hour, define the following variables.

$c_{j,k}$: percentage of CPU usage at the k^{th} time interval of the j^{th} minute. The CPU time is measured in clock ticks or seconds. This percentage is measured by the ratio of number of deployed CPUs and total number of available CPUs in the server. Suppose a server has 10 CPUs and only 6 CPUs are executing the received tasks at the k^{th} time interval of the j^{th} minute. The other CPUs are idle. Hence the value of $c_{j,k}$ is 60%.

$m_{j,k}$: percentage of memory usage at the j^{th} minute and the k^{th} second interval. The percentage is measured by the ratio between the actual amount of memory units used and the total available memory units.

$\tilde{c}$: the predicted amount of requested CPU units in the next $(i+1)^{th}$ hour.

$\tilde{m}$: the predicted amount of requested memory units in the next $(i+1)^{th}$ hour.