Chapter 4

Workload Characterization

Workload characterization can be used to study the operation of a system to gain insight that can be used to make decisions in the model design process. In this chapter, we examine the workload of the CDF system to determine the day, time interval, and feature set that will be included in our model. After the interval for the model has been identified, we study the workload during this period for the purpose of understanding the current operation of the system, as this aids in model design and policy decisions.

Section 4.1 explains the data reduction and analysis techniques and tools that were used. A general workload overview for the full data collection period is presented in Section 4.2. In Section 4.3, we determine that Thursday December 9^{th} from 1:00 pm to 5:00 pm is the interval on which our model will be based. In Section 4.4, we examine the workload during this four hour interval to determine the feature set for our model in Section 4.5. Our examination of the workload in Section 4.4 also provides insightful information about about the current operation of the CDF system. In Section 4.6, we examine the components of the workload that may influence design decisions for the model. Concluding remarks for this chapter are presented in Section 4.7.

The appendix for this chapter, Appendix A4, contains a mass of tables and figures that provide additional information for the workload characterization study in this chapter. The appendix is provided for those readers who require a more detailed examination of the data analyzed in our study.

4.1 Data Reduction

In studies of distributed systems, the amount of data that is collected can be very cumbersome, unless suitable tools and techniques are employed to reduce and to interpret the data. Unfortunately there was no single tool that was capable of performing all the measurement, reduction, and analysis that was needed for this thesis. Table 4.1 summarises the tools that were used for different phases of our study.

Task	Tools
Data Collection	see Table 3.6
Data Analysis	C-shell script, UNIX commands, C programming language, maple, jgraph, SAS, cluster analysis
Model Design	C-shell script, UNIX commands, C programming language, maple, cluster analysis
Simulation	CSIM would be suitable

Table 4.1: Tools Employed in this Study

A series of C-shell scripts and programs written in the C programming language were used to reduce the data collected by the dynamic data collection script. The C-shell scripts used a `foreach` loop to cycle through each host for which data have been collected. Specialized C programs were used to create `jgraph` [1] code for each host. The `jgraph` program was used to translate this code into the postscript graphs that are presented throughout this chapter.

C-shell scripts were also used in the reduction of the process accounting records. The main script cycled through each unprocessed accounting file and used the UNIX `acctcom` command to convert each file into ASCII form. A series of specialized C programs were used to determine resource averages for different views of the data [2] and to create input files for the SAS cluster analysis routines.

No data reduction was required for the static data collection. If a model were to be built, configuration parameters for the model could be taken directly from these data.

[1] jgraph is a simple graph description language program that converts ASCII data files to postscript or LaTeX formats.

[2] The resource usage fields are explained in Table 3.4.

4.2 General Analysis

The purpose of this section is to provide an overview of the workload in the CDF system over the full duration of the data collection period. This characterization is used to determine fluctuation and affinity in the workload, and to highlight any potential problem areas in the system. A better general understanding of the workload will aid in the selection of the interval on which the model will be based.

In the discussion that follows, data from the `top` command were examined for individual hosts. As there were too many hosts in the CDF system to present data for each host, the tables in the following three subsections contain a representative subset of the hosts. The file server (marvin), the compute server (eddie), and dev were included because of their specialized functions in the CDF system. A few workstations that had high (clique, puck, titania), medium (prefect, tea), and low load (setbasis, zarquon), and two workstations that had short periods of extremely high load (snout, tsp) were also chosen. A graphical presentation of the `top` data for these hosts with some further interpretation is provided in Section A4.1 of Appendix A4.

4.2.1 UNIX Load Averages

The UNIX load average is a measure of the average number of jobs waiting in the run queue. High load averages may indicate that response times experienced by users are not acceptable. Acceptable UNIX load averages recommended by the UNIX tuning guides are two for client workstations, and five for servers.

In Table 4.2, we present statistics for the UNIX load averages on 12 hosts in the CDF system over the full duration of our four day collection period. The mean UNIX load average for the servers (marvin and eddie) was substantially higher than that of the client workstations. The servers also had a higher standard deviation, indicating that they had more fluctuation in their workload than the client workstations over the four hour collection period.

Periods of high load for the servers generally occurred during the early morning hours,[3] and from approximately 12:00 pm to 6:00 pm each afternoon. The early morning load was

[3]Throughout this thesis, "early morning" will be used to refer to the interval from 12:00 am until approximately 5:00 am. System activity typically predominated user activity in the CDF system during this period.

Host	min	max	mean	med.	st. dev.
(a) marvin	0.20	14.72	3.83	3.48	1.64
(b) eddie	0.12	10.37	2.49	1.96	1.82
(c) dev	0.00	4.29	0.61	0.21	0.81
(d) clique	0.00	4.30	0.60	0.44	0.54
(e) puck	0.00	2.60	0.34	0.19	0.42
(f) titania	0.03	2.37	0.73	0.68	0.33
(g) snout	0.00	2.73	0.47	0.23	0.55
(h) tsp	0.00	3.66	0.71	0.32	0.86
(i) prefect	0.00	2.35	0.30	0.21	0.32
(j) tea	0.00	2.00	0.32	0.21	0.35
(k) setbasis	0.00	1.41	0.19	0.11	0.24
(l) zarquon	0.00	1.78	0.17	0.12	0.20

Table 4.2: UNIX Load Average statistics for 12 CDF Hosts

caused by system activity, while the afternoon load was caused by user activity.

Two of the hosts listed in Table 4.2, snout and tsp, had high UNIX load averages for a period of approximately one hour. These loaded periods were each caused by a single long-running process. Processes that placed such a high load on the system were rare, but can be represented by the distribution-based model that we design in this thesis.

We examined the `top` data to determine if there was a difference in the workload on the 44 monochrome client workstations, compared to the 20 coloured client workstations. A significant difference might indicate that these hosts should have separate workload definitions in a system model. Table 4.3 shows that the mean load average for the coloured and monochrome client workstations (which does not include marvin, eddie, and dev) was quite similar. 20 coloured workstations had a mean load of 0.31, while the mean UNIX load average for the 44 monochrome workstations was 0.28. This difference is not significant, and understandably so, since the only major variation between the coloured and monochrome client workstations is the main memory size.

Combined Hosts	min	max	mean	med.	st. dev.
64 workstations	0.00	12.07	0.29	0.16	0.38
20 coloured	0.00	12.07	0.31	0.17	0.40
44 monochrome	0.00	9.64	0.28	0.16	0.37

Table 4.3: UNIX Load Average statistics for Groups of CDF Workstations

4.2.2 CPU Utilization

Subtracting the percent idle time that is provided by the `top` command from 100% yields the percent utilization of the CPU. We show simple statistics for the percent utilization of the CPU during our four day collection period in Table 4.4.

The CPU utilization on marvin (an average of 81.49%) was extremely high during the four day collection period, especially during the early morning hours when it was running the `earlymorn` system script. The highest percent utilization reached on eddie was 93.20%, which might indicate a cause for concern. The average CPU utilization was quite acceptable at 34.95%,[4] however, indicating that eddie's powerful CPU subsystem (4 CPUs, each running at 40 MHz) was adequately keeping up with processing requests.

Host	min	max	mean	med.	st. dev.
(a) marvin	24.30	99.80	81.49	85.10	17.09
(b) eddie	6.00	93.20	34.95	31.90	19.45
(c) dev	0.70	100.00	28.62	9.60	35.34
(d) clique	7.50	76.60	21.84	21.20	9.47
(e) puck	1.20	100.00	17.85	10.20	23.61
(f) titania	1.00	100.00	12.16	9.00	12.84
(g) snout	1.40	100.00	27.65	9.60	36.25
(h) tsp	1.00	100.00	30.61	10.90	37.58
(i) prefect	1.40	100.00	11.53	9.00	12.44
(j) tea	1.30	90.60	13.13	9.40	13.87
(k) setbasis	1.10	52.60	7.97	6.40	6.81
(l) zarquon	0.80	58.70	7.82	6.30	6.57

Table 4.4: CPU Utilization statistics for 12 CDF Hosts

In general, on all hosts periods that had a high UNIX load average also had a high percent utilization of the CPU. This is anticipated, as the load average is a measure of how many jobs are waiting for the CPU; the more jobs that are waiting for the CPU, the more likely it is to have been highly utilized.

[4]The percent CPU utilizaton reported for eddie represents the total capacity of the 4-processor system. An average of 13.54% of the percent utilization on eddie was recorded as %spin by the `top` command; this represents the unusable wasted CPU time that was a result of the SunOS 4.1.2 kernel locking mechanism.

4.2.3 Memory Usage

The `top` command provides four different memory statistics: active, free, available, and locked memory. These statistics can be equated as follows:

$$Available\ \ Memory = Active\ \ Memory + Free\ \ Memory \quad (4.1)$$

$$Total\ \ Memory = Available\ \ Memory + Locked\ \ Memory \quad (4.2)$$

The free memory is the amount of memory kept free by the kernel for short term use. This memory is dynamically allocated by the kernel when new data structures are needed. When the free memory becomes too low, the paging algorithm is activated. The active memory is used as a disk buffer cache. The locked memory is reserved by the kernel and had little size variation over the collection interval. The total memory on each host did not fluctuate at all during the collection period.

Table 4.5 shows simple statistics for the active memory on 12 hosts, as a percentage of the available memory. In general, as the system becomes busier, the disk buffer cache fills, and more memory becomes active. The amount of active memory was especially high on the workstations during the day when X11 sessions are active.

Host	min	max	mean	med.	st. dev.
(a) marvin	74.02	99.82	97.60	99.14	3.61
(b) eddie	20.43	99.03	91.52	96.44	10.69
(c) dev	58.62	99.49	85.76	93.47	12.15
(d) clique	79.17	99.20	95.62	97.24	3.55
(e) puck	45.64	99.36	93.08	96.53	7.52
(f) titania	63.54	99.03	91.97	96.16	8.58
(g) snout	75.07	98.64	95.37	96.85	4.07
(h) tsp	36.36	98.97	95.30	96.82	5.15
(i) prefect	74.69	98.71	95.49	96.40	3.30
(j) tea	74.22	99.28	94.62	96.71	4.83
(k) setbasis	59.87	99.30	90.81	94.92	8.64
(l) zarquon	70.61	98.17	94.19	95.74	4.52

Table 4.5: Active Memory Percentage statistics for 12 CDF Hosts

Several hosts had low active memory usage in the early morning hours, particularly on Monday morning. When pages of memory are not used for a long period of time, they are put back into the free memory page list, thus increasing the amount of free memory in the

system. The low active memory usage on most Monday mornings is likely a consequence of a reduced user workload in the system at this time [DiM96].

As shown by the standard deviation values in Table 4.6, there was little fluctuation in the available memory as a percentage of the total memory during the collection period. On the coloured workstations there was an average of 88.96% of the total memory available, while on the monochrome workstations there was an average of 86.52% of the total memory available. This difference was simply because the monochrome workstations had a smaller main memory size (12 MB) than the coloured workstations (16 MB). After locking the required memory for the kernel, this left a smaller percentage of available memory on the monochrome machines.

Combined Hosts	min	max	mean	med.	st. dev.
64 workstations	59.65	92.03	87.30	87.09	1.68
20 coloured	64.66	92.03	88.96	89.03	1.18
44 monochrome	59.65	89.71	86.52	86.72	1.25

Table 4.6: Available Memory Percentage statistics for CDF Workstations

4.3 Interval Selection

The model that is designed in this thesis is intended to be used for a load sharing or capacity planning study. This intended use should be used to guide the selection of the period that will be characterized by the model. In either a load sharing or capacity planning study, periods when the system is operating under peak load are of uppermost concern. Load sharing is not worthwhile unless there are periods of intense load on at least a subset of the hosts in the system; accordingly, the model should capture the behaviour of the system during busy periods. To increase the capacity of a system, it is necessary to know how the system operates when it is at its current maximum capacity; thus, periods of heavy load are also desirable for this type of study.

4.3.1 Selecting A Day

The goal of this section is to determine which day of data collection will be included in the model. The data from the `top` command, presented in Section 4.2, did not reveal a particular day that was substantially busier than the others. Any day during the collection

period would be suitable for our study, as the entire week experienced heavy user usage.

Figure 4.1 shows the combined number of unique users that were on all CDF hosts and that were on eddie at 30 minute intervals.[5] The peak number of unique users usually occurred in the early afternoon. The daily peak is quite similar for each day of the collection period. The number of users in the system does not necessarily correspond to the resource demands of the system, but since this study is concerned with user behaviour, it is desirable to choose an interval in which the user activity is high.

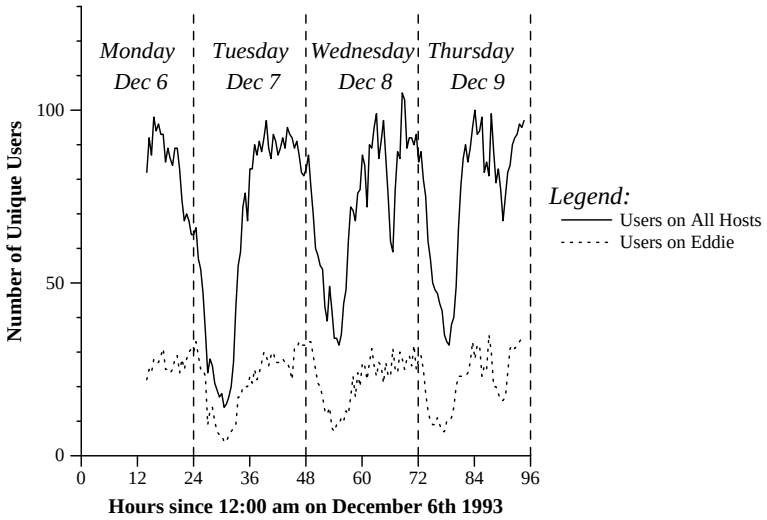


Figure 4.1: Number of Unique Users existing at 30 Minute Intervals

Wednesday was not chosen for further analysis because some process accounting records were missing on this day because of the down period of the compute server on Wednesday afternoon. Additional data from the various "stat" commands were collected on both Tuesday and Thursday, making either of these two days a desirable choice. Thursday was chosen primarily because there were several assignments due in different courses on the following day (Friday), thus providing a diverse user population. Also, the data collection from the "stat" commands was more complete for Thursday, as the `nfsstat` data files for the servers on Tuesday were unsuccessfully collected.

4.3.2 Selecting A Time Interval

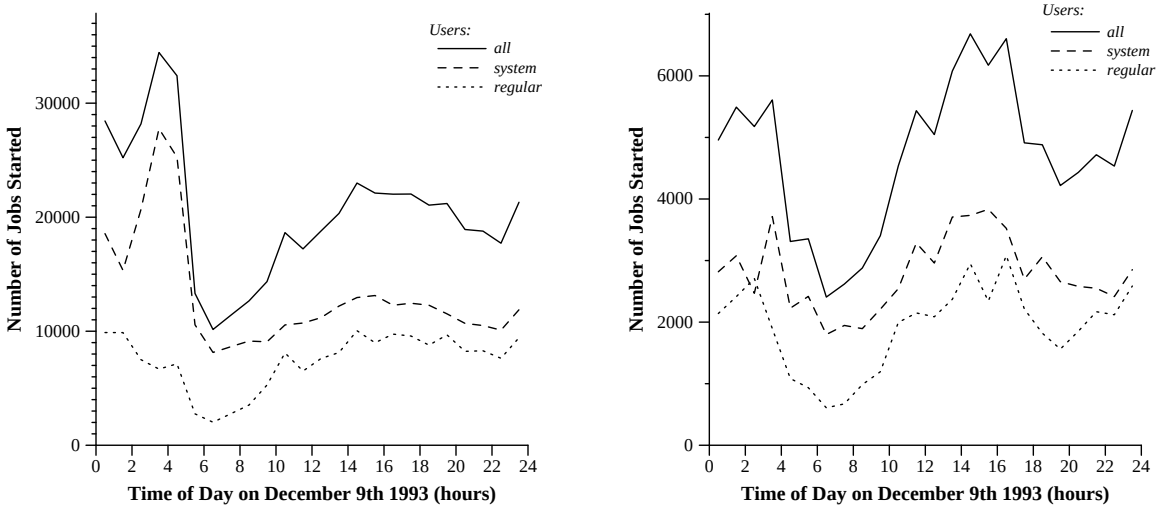
The goal of this section is to determine a specific time interval on Thursday from which the model will be derived. Since the model being designed in this thesis is a static workload

[5] Figure 4.1 was derived using data from the `ru` command. As the `ru` command experienced some drifting, the time of each observation had to be adjusted by 30 seconds.

model that does not vary with time, it is important to select an interval that has relatively constant activity.

When determining the how large or how small of an interval should be chosen, there are several tradeoffs that must be considered. The longer the interval that is chosen, the more time that will be required to process the data in the construction of the model. If too short of an interval is chosen, there many not be enough data to construct a representative distribution-driven model.

The graphs in Figure 4.2 show the number of jobs created on eddie and on all workstations combined. A single point is plotted at the mid-point of each one-hour interval. Separate curves have been drawn to indicate the number of jobs created by system users (root, nobody, sys, news, daemon), by regular users (all other users), and by all users combined. Since user activity is of primary concern in this thesis, a period that has constant, high regular user activity should be chosen.



(a) all hosts combined (b) eddie.cdf only

Figure 4.2: Job Creations in 60 Minute Intervals

As Figure 4.2 shows that the period from 12:00 pm to 8:00 pm on each afternoon had a large number of user jobs created per minute, we chose to study this interval in more detail. We provide graphs and a further examination of the 12:00 pm to 8:00 pm interval in Section A4.2 of Appendix A4. Using this examination, we chose 1:00 pm to 5:00 pm as our study interval. In general we found that activity dropped after 5:00 pm. The period before 1:00 pm was not included in the selected interval because both the load and utilization on eddie and marvin were lower before this time. The four hour interval that we have selected contains sufficient data to construct the distributions for the model, yet the amount of data

within this time interval is still manageable.

4.4 Analysis of Selected Interval

When analyzing a distributed environment, it should be perceived as a series of connected components, such as workstations, software, peripherals, and networks. Each of these components should be broken into smaller measurable components. A client workstation, for example, is composed of memory, CPUs, caches, software, busses, disks, and other peripheral devices. The effect of each component on the overall performance of the system should be taken into consideration.

According to [JK95], the analysis of a distributed system can be performed from a *high-level* or *low-level* perspective. A low-level perspective considers each system and subsystem, whereas a high-level perspective considers the entire network of systems. Ideally both perspectives should be considered when analyzing the performance of a system.

In this section, we examine the current operation of the disks, memory, CPUs, and network in the CDF environment. The measurements for this analysis were extracted for each component using the commands that were listed in Table 3.7. The analysis in this section provides an understanding of the current operation of the CDF system, and will be used for feature set extraction in Section 4.5. In the process of analyzing these components, we discovered insightful information about the current operation of the CDF system and potential areas for performance improvement. We present these comments, which might be beneficial to system administrators, in Section A4.3 of Appendix A4. The discussion in this section is a summary of these findings.

4.4.1 Disk

As most systems usually have an overloaded disk [Sun93], we carefully examined the utilization of the disks in the CDF system using the data collected by the `iostat` command.

The workstations' disks were generally not highly utilized as the average percent utilization for the 33 client workstation disks that were examined was 9.467%.[6] Only clique, which had a percent utilization of 28.194% on its local disk, had a heavily loaded disk. This high utilization for clique's disk was a consequence of its excessive virtual memory usage.

[6] The tuning guide states that disks with a percent utilization that is greater than 30% may be a cause for concern.

We found that the percent utilization of the news disk on eddie and the disks that were scanned by the `backup` command on marvin was quite high, indicating that I/O was in contention.

4.4.2 Memory

Our analysis of the virtual memory usage in the CDF system (using data collected from the `vmstat` command) showed that memory was not in contention on the servers and on the client workstations, with the exception of clique. Only clique experienced excessive swapping (i.e., whole processes transferred to disk) and paging (i.e., pages of memory transferred to disk) activity. The average number of processes that were blocked waiting for resources on clique was high; accordingly, the virtual memory activity was related to the number of processes on this host and to their high memory demands.

4.4.3 CPU

We examined data from the `top` command, over the 1:00 pm to 5:00 pm interval, to determine the CPU utilization on the CDF hosts. The only client workstation that appeared to be problematic was diamond who had an average percent CPU utilization of 100% during the four hour period.[7]

The average percent CPU utilization on eddie and marvin was higher than on the workstations (except for diamond). The average percent CPU utilization on marvin was particularly high at 73.02%, suggesting that the CPU on this host was often in contention. The average combined CPU utilization for the four CPUs on eddie was 52.74%; an average of 25.48% of this utilization was wasted CPU time that was spent spinning on the SunOS 4.1.2 kernel's mutex lock (%spin reported by the `top` command).

4.4.4 Network

The analysis of the data from the `netstat` and `nfsstat` commands showed that the network was not in contention in the CDF system. The percentage of error on the input packets (an average of 0.00010% for 40 client workstations) was substantially lower than the recommended 0.025% [Ste90]. In addition, the combined average percentage of the network collisions for 42 hosts in the CDF system was 2.407%, which is lower than the 5%, which is

[7] A percent CPU utilization of 80% or greater indicates that the CPU may be in contention [Sun93].

characteristic of a lightly loaded Ethernet.

4.5 Feature Set Extraction

In this section, we use workload characterization to select the feature set that will be included in the model of the CDF system. The feature set was selected based on the available features at the chosen level of model representation, the intended use of the model, and the workload characterization (see Section 4.4) of the CDF system.[8]

The resources used by jobs were chosen as the level of model representation. This decision was made because this information was readily available from the process accounting records, and because if load sharing was implemented in the model, it would be straightforward to implement a placement policy on a per job basis.

The six job features listed below were provided by the process accounting records. A detailed description of each feature can be found in Table 3.4.

1. System CPU Time (seconds)
2. User CPU Time (seconds)
3. Number of Disk Blocks Read or Written
4. Kcore Memory (cumulative KB/minute)
5. Real Time (seconds)
6. Number of Characters Transferred

The total CPU time (user and system CPU time) and the number of disk blocks used by each job were chosen for the feature set. The workload characterization presented in Section 4.4 showed that the CPU and disk were often overloaded on the servers, and on some individual workstations.

Since the network was generally lightly loaded (2.407% of transmissions resulted in collisions), extensive modelling of this subset of the system is not really necessary.

There was little paging activity on most hosts, indicating that the amount of main memory on the hosts was sufficient. On the host that had excessive paging activity (clique), the memory constraint could have been overcome by more intelligent job placement by the user (i.e., users should execute memory-intensive jobs on the compute server).

[8] When there are several variables that are closely related, Factor Analysis is another useful technique for feature set extraction.

SunOS 4.X provides the **Kcore** field instead of a field for memory size in the process accounting records. Since the **Kcore** field is a measure of the total cumulative amount of kilobyte segments of memory that were used by a process during each minute of run time, it is not suitable as a feature for the model.

Process accounting records also provide the real (elapsed) time of each process. The real time depends on the resource requirements of a process, network delays, and the user interaction (user think time) with the process. It cannot be considered a resource, but it can be used to estimate the average user think time between bursts of CPU requests.

The number of characters transferred refers to the number of characters (in kilobytes) that are transferred to or from devices. This feature was not included because the source and destination of the characters could not be determined, and it is beyond the level of representation to be used in the model.

4.6 Characterization for Model Design

In this section, we use workload characterization to study various components of the system that provide insight into model design decisions.

4.6.1 Process Behaviour

In this section, we analyze the behaviour of processes that comprised the workload of the CDF system from 1:00 pm to 5:00 pm on Thursday. A better understanding of resource requirement distributions can aid in load sharing policy and model construction decisions.

During the four hour interval from 1:00 pm to 5:00 pm on Thursday, there were 87444 processes for all users (including system users) recorded in the process accounting records. Table 4.7 shows simple statistics for each resource consumed by these processes. Examining the combined simple statistics for all data points is a useful tool that can quickly provide insight into massive amounts of data. A description of each simple statistic displayed in Table 4.7 can be found in Law and Kelton [LK91] or in most general statistical references.

The median value for each resource was lower than the mean, indicating that all resource distributions had tails to the right. The skewness and coefficient of variation (CV) can be used to determine the shapes and general distribution family for each resource [LK91].

Table 4.8 shows resource correlations for these 87444 processes. Correlation values that are close to 1.0 indicate high positive correlation, values that are close to -1.0 indicate

Resource	Min	Max	Mean	Median	St Dev	Skew	Kurt	CV
Real Time (sec)	0.02	80213.33	160.25	0.82	1706.00	21.62	666.70	10.65
System CPU (sec)	0.00	2093.87	0.55	0.08	10.29	130.84	23350.37	18.75
User CPU (sec)	0.00	1764.27	0.73	0.02	18.51	55.48	3797.51	25.34
Total CPU (sec)	0.00	3352.54	1.28	0.12	25.54	63.22	5575.11	19.96
Disk Blocks	0	41704	9.57	0	299.22	113.07	13551.17	31.25
Terminal IO	0	1.824xE9	179955.9	514	1.116xE7	135.81	20012	62.04
Kcore Mem (KB/min)	0.00	306417.20	40.67	0.50	1418.00	135.03	26052.54	34.87

Table 4.7: Resource Statistics for all jobs by all users in 1-5pm interval

high negative correlation, while those close to 0.0 show no correlation. The three CPU measurements were highly positively correlated amongst themselves, with the correlation between the user CPU time and the total CPU time being the highest at 0.9402. This shows that most processes used similar ratios of user CPU time and total CPU time.

Resource	Real	SCPU	UCPU	CPU	Blocks	TTY IO	Kcore
Real	1.0000	0.2488	0.2664	0.2933	0.0217	0.0718	0.1753
SCPU		1.0000	0.5350	0.7907	0.2593	0.4583	0.8454
UCPU			1.0000	0.9402	0.0241	0.2044	0.6644
CPU				1.0000	0.1220	0.3328	0.8222
Blocks					1.0000	0.0329	0.0937
TTY IO						1.0000	0.2472
Kcore							1.0000

Table 4.8: Resource Correlations for all jobs by all users in 1-5pm interval

There was also high correlation between the Kcore memory usage and CPU usage. The more CPU cycles that a process uses, the larger the cumulative Kcore memory statistic will be. The system CPU time had the highest correlation of 0.8454 with Kcore memory. Figure 4.3 shows the joint distribution of the Kcore memory and total CPU time. This graph indicates the regression analysis line with a 95% confidence limit provided by SAS. Figure 4.3 (a) shows the full range of data, while Figure 4.3 (b) expands a small region near the origin. The high correlation is evident by the general pattern of points spanning from the lower left to the upper right corner of these graphs.

The resources to be included in the model, total CPU and disk block usage, had a low positive correlation of 0.1220. To better understand the relationship between these two resources, a scatter plot of total CPU and disk block usage is shown in Figure 4.4. Figure 4.4 agrees with the joint CPU and disk usage reported by Leland and Ott [LO86], which reveals

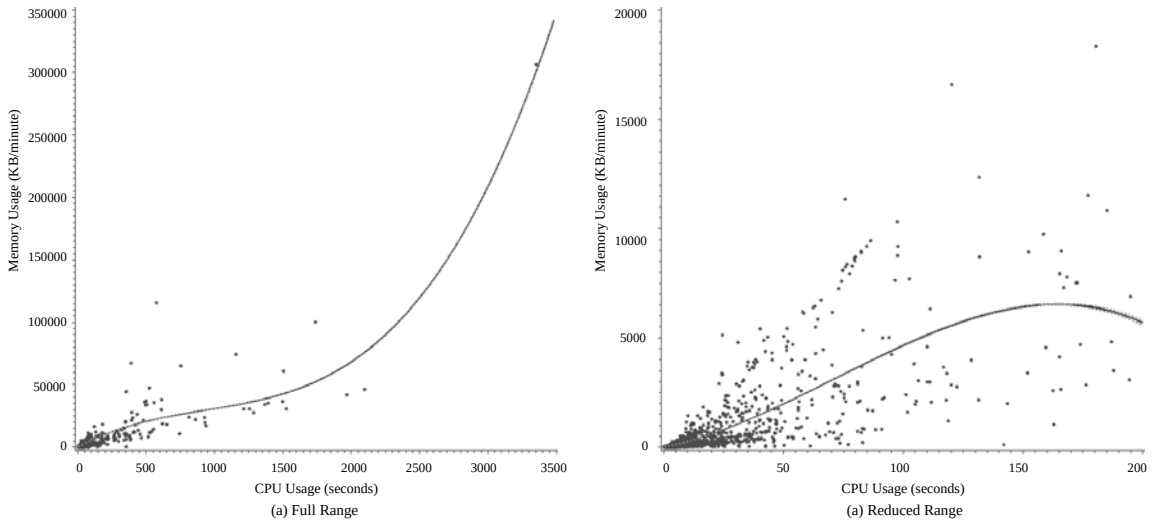


Figure 4.3: Joint Probability Distribution of CPU and Memory Usage

three different types of processes. The majority of processes were small "ordinary" processes that are close to the origin in the scatter plot. There were also CPU hogs and disk hogs along either axis, but no processes were both disk and CPU hogs.

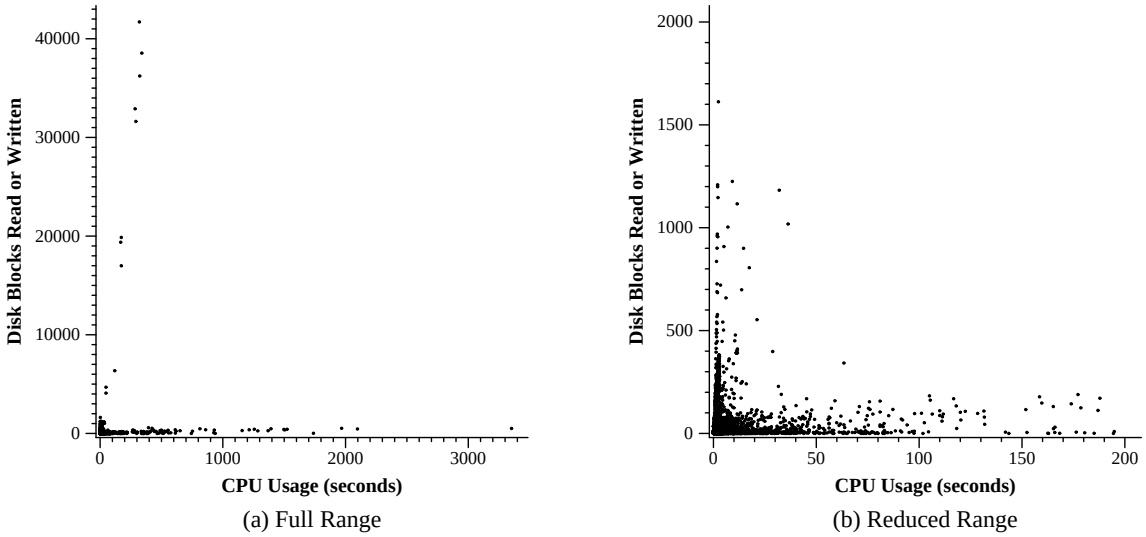
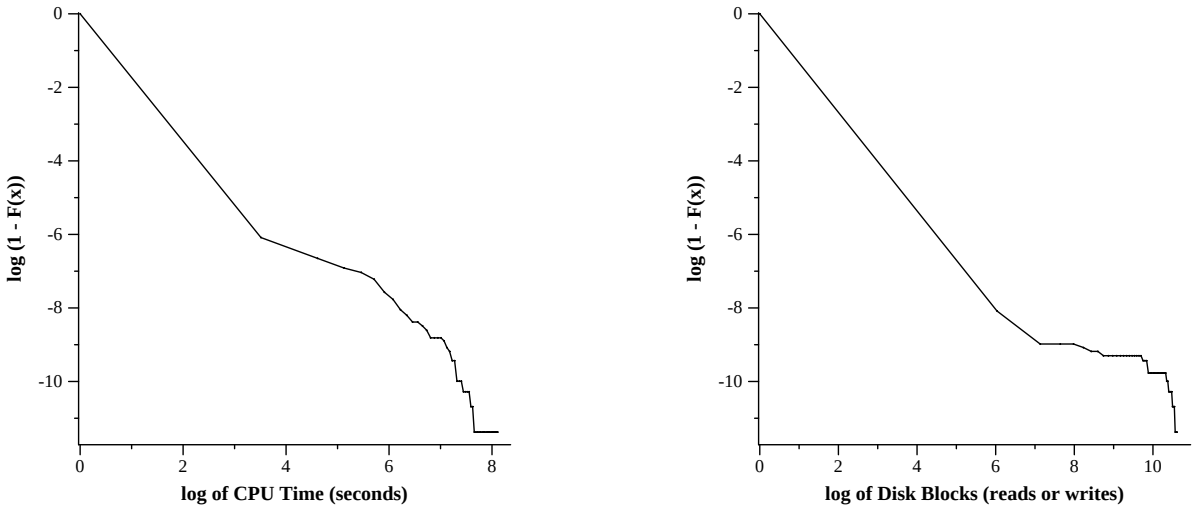


Figure 4.4: Joint Probability Distribution of CPU and Disk Usage

The distributions for CPU usage and disk block usage both had very long tails to the right. The probability density functions (pdf) and cumulative distribution functions (cdf) do not provide much insight into the data, as the data were severely skewed. A log-log plot of the pdf or cdf plot can be used to flatten out the data for increased visibility. Figure 4.5 shows the cdf plots for (a) CPU usage and (b) disk usage with a log-log scale.

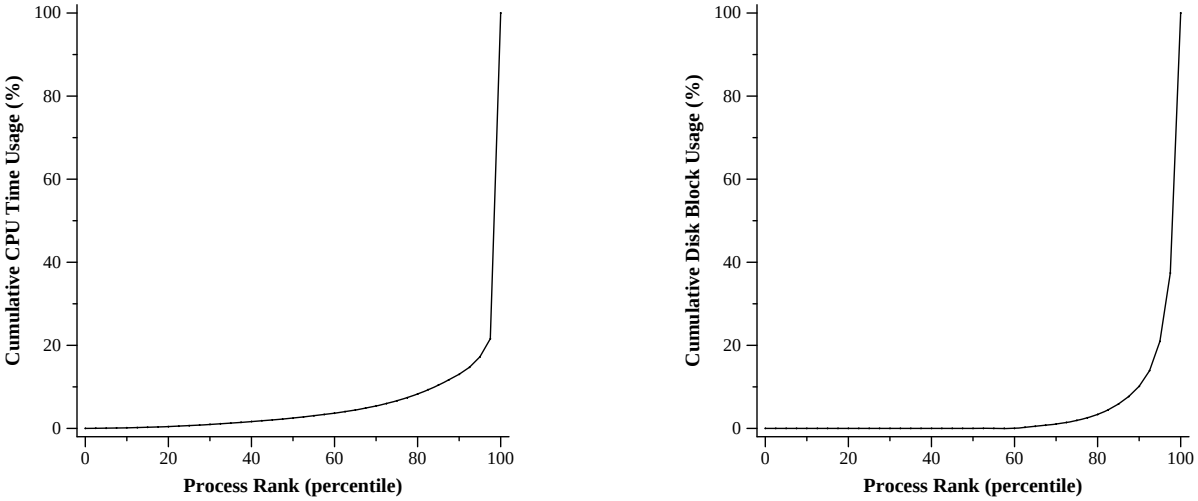


(a) CPU time (b) disk blocks

Figure 4.5: Cumulative Distribution Function with Log-Log Scale

In load sharing studies, the processes that have the highest resource demands are of primary importance because these processes may be candidates for remote placement. Processes that do not consume substantial resource amounts would not merit the overhead of remote placement. The graphs in Figure 4.6 rank the processes in terms of their resource usage, and show the cumulative percentage of the overall resource consumption for each process rank (percentile). In Figure 4.6 (a) the 97.5 percentile of processes accounted for 21.58% of all CPU usage, meaning that the remaining 2.5 percentile accounted for 78.42% of all CPU usage. Figure 4.6 (b) shows that the final 5 percentile of disk block usage ranked processes accounted for 79.03% of all disk block usage. As in [LO86], a small percentage of the largest processes accounted for a large fraction of the overall CPU usage. This was also true of the disk block usage.

This indicates that there were very few processes in the system that would be potential candidates for remote execution. This is not surprising as our earlier analysis showed that the number of regular user processes on all hosts that required at least 15 seconds of CPU time was an average of only 112.50 per hour during the 1:00 pm to 5:00 pm interval (see Section A4.2.1). When both system and regular users were included, there were an average of 150.25 processes that required at least 15 seconds of CPU time per hour during the 1:00 pm to 5:00 pm interval. Since there were 70 CDF hosts, there were approximately only two process per host per hour that consumed at least 15 seconds of CPU time! It is noteworthy that of these processes, very few were likely to have been processes that are actually suitable for remote execution. Many may have used large amounts of CPU because

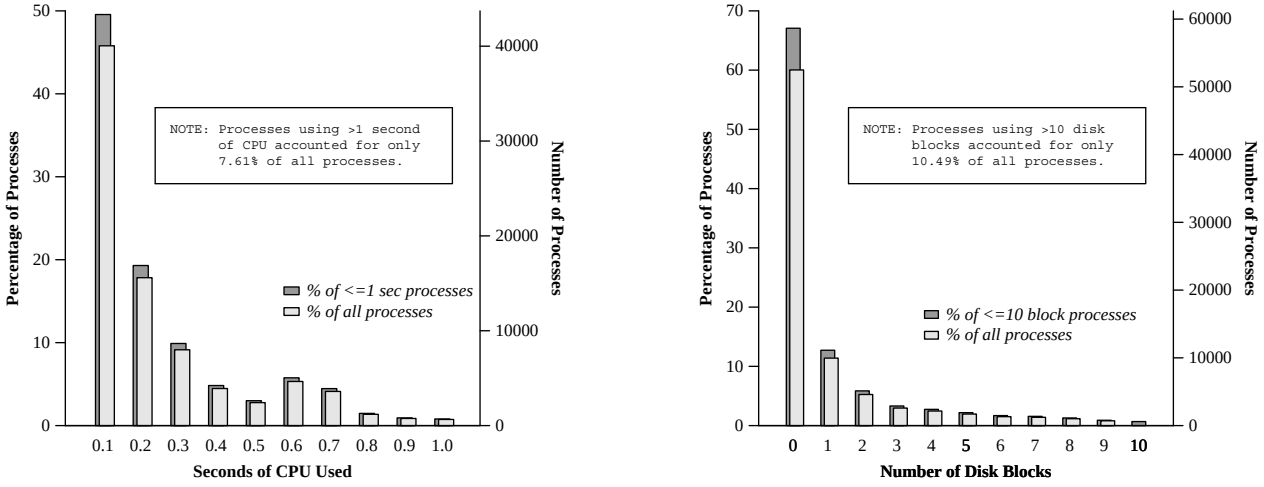


(a) CPU time (b) disk blocks

Figure 4.6: Resource Use by Process Rank

of their long-running nature, but placement of these processes on remote hosts would incur heavy disk overhead.

Figure 4.7 shows that the majority of processes in the system had very low resource usage. The percentage of processes in the system that required 1 second of CPU time or less was 92.39%. Of these processes, 49.56% required only 0.1 second or less. Overall, 45.79% of all processes required 0.1 second or less. If a load sharing policy were in place, the majority of processes should not even be considered. An initial placement policy that examines each job placed would clearly result in unnecessary processing overhead.



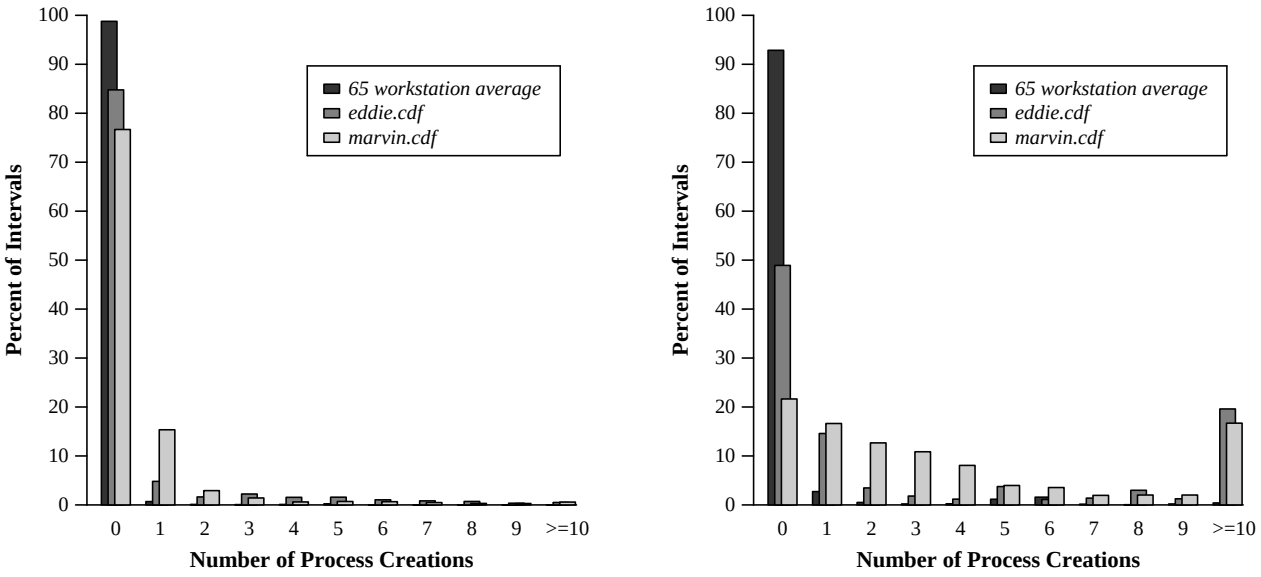
(a) CPU time (b) disk blocks

Figure 4.7: Breakdowns for Processes with Low Resource Requirements

Most load sharing studies focus on CPU usage because it is most indicative of load sharing gains. Disk usage graphs are shown in addition to CPU usage graphs in this section

to provide more insight into process and system requirements. Figure 4.7 (b) shows the percentage of processes that needed to read or write up to 10 disk blocks. It is quite remarkable that 60.03% of the processes in the system did not require the local disk or the file server disks at all. Only 10.49% of all processes used more than 10 disk blocks.

The graphs in Figure 4.8 are provided to understand the rate at which the system had to respond to process arrival requests (process creations). When one second time intervals were considered, there were frequently intervals in which no processes were initiated on the hosts in the system. For the client workstations, on average 98.77% of all one second intervals had no processes created, while this happened more infrequently on the servers: 84.77% on eddie and 76.68% on marvin. Marvin was more likely to receive one or two requests while eddie was more likely to create 3 to 9 processes in each one second interval. This indicates that the workload was more bursty on eddie, while processes were created at a steady rate on marvin. The regular rate of job creations on marvin is because of the more predictable behaviour of system scripts, while on eddie the bursty nature of job creations is related to the unpredictable nature of user requests.



(a) Creations in 1 Second Intervals

(a) Creations in 10 Second Intervals

Figure 4.8: Process Creations in Fixed Intervals

The number of jobs created in each ten second interval is shown in Figure 4.8 (b). Again, the number of creations is more constant on marvin, while peaks at 5 and 8 jobs created in ten second intervals are evident for eddie.

Figure 4.9 (a) shows the mean residual CPU usage for process ages in CPU seconds. The mean residual CPU usage is the average amount of CPU time remaining. The mean residual CPU usage shown in Figure 4.9 (a) is approximately linear. It can be used to determine at what age processes become candidates for migration. For more details about this type of analysis, the reader is directed to [LO86]. The mean residual disk block usage shown in Figure 4.9 (b) is not linear, indicating that predicting how much more disk usage a given process would require should be based on a higher order model.

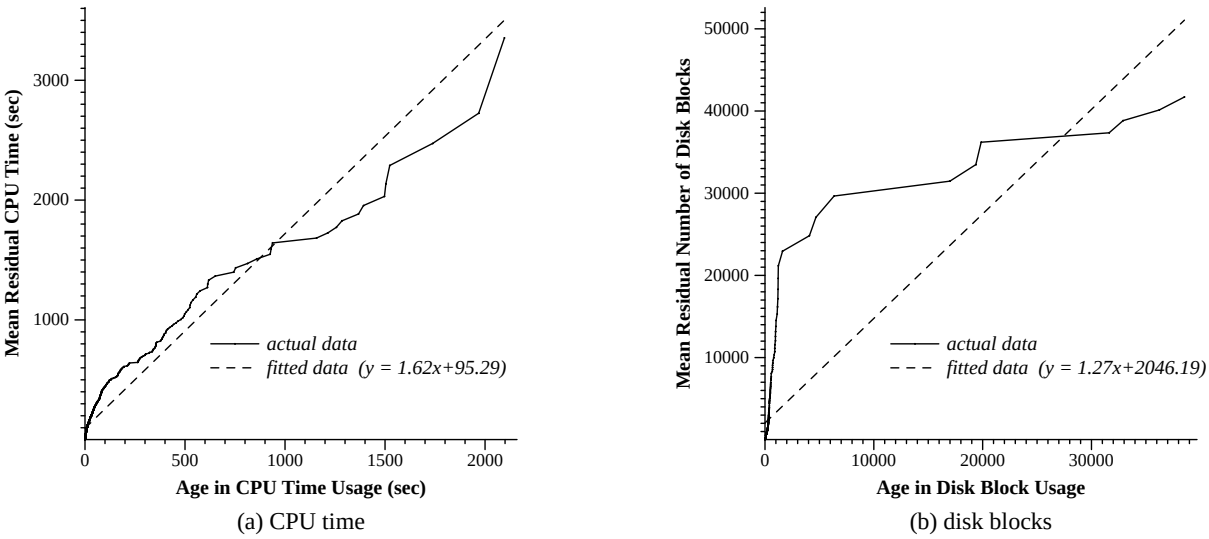


Figure 4.9: Mean Residual Resource Usage

In summary, few processes in the CDF system had very high resource usage, while the majority had exceptionally low resource usage. The results in this section support the load sharing recommendations cited by [Cab86] and [LO86]: expensive initial placement load sharing algorithms are not worth while based on the workload in this environment. If any load sharing is to be done, it should be based on the migration of long-running processes. The current workload in the CDF system does not justify load sharing. To study the effects of load sharing in this environment, artificial hot-spots of activity would have to be introduced, thus detracting from the overall validity of the study for this environment.

4.6.2 Command Usage

The available commands in a system can dictate the type of workload that is possible. The workload may exhibit cyclic per-academic-term or yearly behaviour, depending upon which

courses are offered and the software that is introduced or withdrawn from a system. Since world wide web browsers, for example, were not present in the CDF system during the time of our data collection, a more network-intensive workload could be expected if data were collected for the current CDF system that now provides access to the world wide web. As more complex software packages or applications are introduced in courses as the term progresses, or from year to year, the workload is expected to be influenced.

We provide a further discussion of some of the commands that were used in the CDF system in Section A4.4 of Appendix A4. We examine the most frequently used commands, and those commands with the highest CPU usage.

4.6.3 User Behaviour

The workload can be studied in terms of the impact that different types of users place on system resources. If groups of users that have distinct resource usage can be identified, they can be used as the primary components in the model. Such a model could easily be adapted in a capacity planning study; the count of users of a certain type could be incremented to predict the system's ability to accommodate this additional load.

Initially groups of users are chosen based on their known function in the CDF system. The pie chart in Figure 4.10 shows the percentages of commands that were given by each group of users. The **regular** users (shown in dark grey) include course users (39.90%) and staff users such as system administrators and faculty (2.28%). The **system** users (shown in light grey) include root (47.52%) and other system users, including "sys," "daemon," "news," "nobody," and "backup" users (10.30%).

Table 4.9 provides the mean and coefficient of variation (CV) for the total CPU usage and for the disk block usage of the four groups of users in Figure 4.10. In general, the resource means were low for the user groups, with the exception of the mean disk block usage for the system user group.

Examination of the commands ranked by average disk block usage showed that the 12 `backup` commands that are included in the system group, had exceptionally high resource usage. When these outliers were removed from the system group, Table 4.10 shows that the mean disk block usage for the system group is reduced by 78%. Table 4.11 shows the resource averages for each user name in the system group. Notice how much higher the average resource usage is for the `backup` command. The `backup` command is clearly an

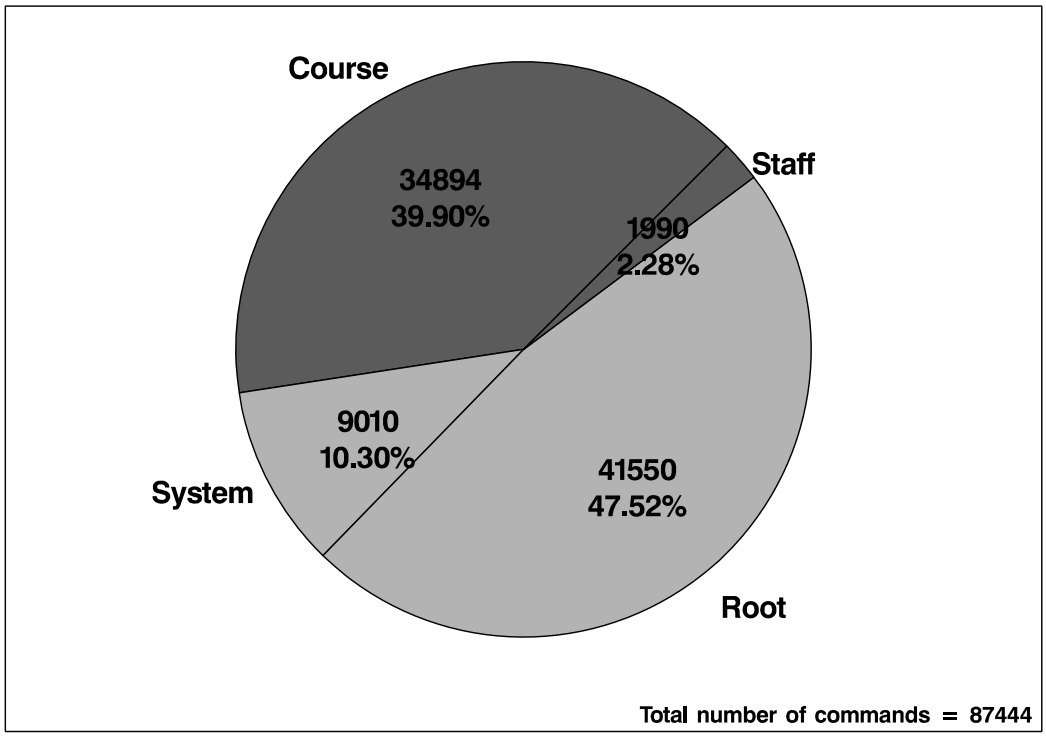


Figure 4.10: Number of Commands Used by Different Groups of Users

User Group	Number of Commands	Total CPU		Disk Blocks	
		Mean	CV	Mean	CV
Course	34926	1.40	10.68	4.35	8.77
Staff	1990	1.09	8.32	5.41	6.44
Root	41563	0.96	24.37	8.51	3.41
System	9023	0.65	12.45	35.32	26.21
Total	87502	1.11	17.11	9.54	31.35

Table 4.9: Resource Statistics for User Groups

outlier, and should be treated separately in a model.

Table 4.12 shows statistics for different groups of course users. The course users are subdivided according to the prefix of their user names (as explained in the appendix for Chapter 3). The resource averages are lower for the first year students (a100) than for students with accounts in higher level courses (a200-a2000). The difference, however, is not prominent. As in [Blu94], all academic course users have such similar resource usage that they should not constitute separate groups in the model.

The coefficient of variation for a group indicates how close the entries in a group are to its centroid (mean). Large coefficient of variation values indicate that members within a

User Group	Number of Commands	Total CPU		Disk Blocks	
		Mean	CV	Mean	CV
Course	34926	1.40	10.68	4.35	8.77
Staff	1990	1.09	8.32	5.41	6.44
Root	41563	0.96	24.37	8.51	3.41
System	9011	0.41	2.99	7.91	6.64
Total	87490	1.08	17.34	6.72	5.36

Table 4.10: Resource Statistics with 12 `backup` Commands Removed

User Group	Number of Commands	Total CPU		Disk Blocks	
		Mean	CV	Mean	CV
news	7917	0.37	2.30	8.92	6.27
sys	611	0.13	0.76	0.01	9.06
daemon	285	0.20	0.96	1.26	2.06
nobody	198	3.29	1.70	1.28	1.99
backup	12	181.18	0.72	20621.00	0.75
System	9023	0.65	12.45	35.32	26.21

Table 4.11: Resource Statistics for System User Groups

group do not fall close to the centre (or mean) of the group. It is desirable to make a model based on several groups that have low CV values, as these groups are easier to represent.

The goal in designing a model is to identify groups of users that have lower CV values on the feature variables than the combined data set does. Notice in Table 4.11 how much lower the CV values are for each group than they are for the overall system group that is shown in the last row of this table. Most improvement is a result of separating the `backup` command. The disk block CV for the system group is 26.21 with the 12 `backup` commands included, but reduces to 6.64 (in Table 4.10) when these 12 commands are removed.

The resource usage coefficient of variation values for the four user groups (with the `backup` command removed) shown in Table 4.10 indicate that this may not be the most suitable subdivision into groups. The disk block CV values for course, staff, and system users are larger than the overall disk block CV for all users (5.36), and the CPU CV value for root users is higher than the overall CPU CV for all users (17.34). An alternate method of grouping users, called cluster analysis, will be examined in the next two chapters to attempt to produce groups with lower CV values.

User Group	Number of Commands	Total CPU		Disk Blocks	
		Mean	CV	Mean	CV
a100	615	0.91	2.46	1.99	5.42
a200	7725	1.27	9.20	5.01	14.75
a300	3376	1.52	8.09	4.52	3.29
a400	6301	0.94	16.92	3.20	3.31
a2000	4423	2.56	7.98	5.50	3.63
g3	2873	1.26	9.34	3.25	2.37
g2	4510	1.80	11.79	4.71	3.27
g1	3411	1.13	8.94	5.48	5.96
g0	1498	0.31	13.67	1.28	11.98
g9	194	0.80	4.31	5.05	2.18
Course	34926	1.40	10.68	4.35	8.77

Table 4.12: Resource Statistics for Course User Groups

4.7 Summary

The approach taken in this chapter was to examine the workload for the full data collection period, and then to gradually narrow our focus to determine which interval would be included in our model. As the model designed in this thesis is a static model that does not include workload variations over time, an interval with uniform activity was sought. We chose the interval from 1:00 pm to 5:00 pm on Thursday December 9^{th} to be included in our model; this interval had consistently high load and user activity.

A detailed workload characterization of the CDF system for the 1:00 pm to 5:00 pm interval was carried out in Section 4.4. We studied the disk, memory, CPU, and network usage in this section by summarising the statistics provided by the various "stat" commands. Our analysis showed that the disk utilization on most hosts was acceptable. The `backup` command on marvin, however, did cause I/O contention on this host. The news disk on eddie also had higher than average disk utilization. The amount of memory on each host was sufficient, except for one workstation (clique) that was running a process that had high virtual memory demands. The CPU utilization was high on marvin, but reasonable on all other hosts. The network was not highly utilized.

In general, the CDF system was operating within its capacity, with the exception of the high disk and CPU utilization on marvin. The higher load on the file server suggests that it is overworked; consequently, the system may benefit from upgrading the hardware on this

host, or offloading some of its file server duties to other hosts. It may also be appropriate to designate a separate news server host, or to add more disks to eddie to handle the network news I/O requests.

The analysis in Section 4.4 was used in Section 4.5 to choose total CPU time and the number of disk blocks read or written as the features to be used in the model. In the process of examining the CDF workload in Section 4.4, we also became more knowledgeable about the current operation of the CDF system (specific results were presented in Section A4.3). This insight could be used to aid in model design decisions or to improve the performance of the current system

As a prelude to the model synthesis (to be outlined in Chapter 6), the behaviour of processes, commands, and users were examined in Section 4.6. This workload characterization showed that the CDF system had few commands that were extremely resource intensive, and few commands that would be suitable for remote placement in a load sharing environment.

Although the workload characterization in this chapter revealed that the CDF workload was not very resource-intensive, the techniques used for the workload characterization were very successful at providing insight into our workload data set. We feel that these techniques could also be extended to studies of workload data collected from other types of systems. As our workload characterization techniques are reasonably scalable, larger or smaller data sets could be examined. A word of caution is that when examining statistics from systems that have different operating systems and data collection commands, it may be necessary to interpret the fields returned by the statistical commands somewhat differently. Study the command manual pages carefully before beginning your analysis.

While this chapter primarily analyzed data that was collected by the "stat" commands, the next chapter (Chapter 5) will use cluster analysis to characterize the process accounting workload data.