December 1987

UILU-ENG-87-2274
CSG-78

COORDINATED SCIENCE LABORATORY
College of Engineering

USAGE ANALYSIS OF USER FILES IN UNIX

Murthy V. Devarakonda
Ravishankar K. Iyer

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-87-2274 (CSG 78)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION AT&T NASA
6c. ADDRESS (City, State, and ZIP Code) 1101 W. Springfield Avenue Urbana, IL 61801	7b. ADDRESS (City, State, and ZIP Code) AT&T NASA Langley Research Center Info Systems Hampton, VA 23665 Middletown, N.J. 07738
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AT&T NASA 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AT&T 1-5- 37411 NASA-NAG-1-613
8c. ADDRESS (City, State, and ZIP Code) See 7b.	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Usage Analysis of User Files in Unix

12. PERSONAL AUTHOR(S)

Devarakonda, M. V. and Iyer, R. K.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	1987 December	42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Measurements, UNIX, file usage, statistical analysis, user-behavior

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper presents a user-oriented analysis of short term file usage in a 4.2 BSD UNIX environment. The key aspect of this analysis is a characterization of users and files, which is a departure from the traditional approach of analyzing file references. Two characterization measures are employed: accesses-per-byte (that combines fraction of a file referenced and number of references) and file size. This new approach is shown to distinguish differences in files as well as users, which can be used in efficient file system design, and in creating realistic test workloads for simulations. A multi-stage gamma distribution is shown to closely model the file usage measures. Even though overall file sharing is small,

over

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

some files belonging to a bulletin board system are accessed by many users, simultaneously and otherwise. Over 50% of users referenced files owned by other users, and over 8% of all files were involved in such references. Based on the differences in files and users, suggestions to improve file system performance were also made.

USAGE ANALSYIS OF USER FILES IN UNIX®

Murthy V. Devarakonda
Ravishankar K. Iyer

November, 1987

Computer Systems Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1101 W. Springfield Avenue
Urbana, IL 61801

ABSTRACT

This paper presents a user-oriented analysis of short term file usage in a 4.2 BSD UNIX environment. The key aspect of this analysis is a characterization of users and files, which is a departure from the traditional approach of analyzing file references. Two characterization measures are employed: accesses-per-byte (that combines fraction of a file referenced and number of references) and file size. This new approach is shown to distinguish differences in files as well as users, which can be used in efficient file system design, and in creating realistic test workloads for simulations. A multi-stage gamma distribution is shown to closely model the file usage measures. Even though overall file sharing is small, some files belonging to a bulletin board system are accessed by many users, simultaneously and otherwise. Over 50% of users referenced files owned by other users, and over 8% of all files were involved in such references. Based on the differences in files and users, suggestions to improve file system performance were also made.

1. Introduction

The study of a user's file usage is important for efficient file system design. In addition to providing useful statistics for the measured system, such a study also provides measures and distributions which may be valuable for testing simulation-based models. This paper describes a user-oriented analysis of file usage in 4.2 BSD UNIX® running on a VAX®-11/780 at the University of Illinois. Traces of file-related system calls — read, write, open, close and other calls with their arguments — were collected on 5 different days. The data is analyzed to characterize file usage.

This analysis quantifies a typical user's file usage in a login session and the usage of a typical file in all login sessions, which is a departure from the traditional approach of analyzing file references. A measure of file usage referred to as *accesses-per-byte* is introduced. This measure combines fraction referenced and number of references to a file. Using this measure, two types of usage characterizations are defined. A typical user's file referencing behavior is quantified by the average accesses-per-byte made to referenced files in a login session, the average size of referenced files, and the number of files referenced. This characterization is referred to as a *user characterization*. The usage of a typical file is quantified by the average of accesses-per-byte made over all login sessions, the average file size, and the number of login sessions that referenced the file. This characterization is referred to as a *file characterization*.

Files are then categorized according to the UNIX file type (regular or directory files), the ownership, and the type of use (read-only, temporary, etc.); and users by the amount of file I/O during a login session. Based on empirical distributions and on

---

UNIX is a Trademark of AT&T Bell Laboratories
VAX is a Trademark of Digital Equipment Corporation

analysis of variance, the user and file characterizations are shown to quantify the variability in file usage across the file and user categories. Thus, we establish a systematic approach to quantify a user's file usage in detail, and show that the analysis distinguishes nonuniformity in file usage.

The other results from the study are the following. Almost all user-owned files are completely referenced. User-owned files are usually small and are not referenced many times in a login session, but heavy users' files are larger and are referenced several times more than those belonging to light users. Even though overall file sharing is small, some files belonging to the bulletin board (Notes) system were accessed by many users (simultaneously and otherwise). A surprisingly large number of users (about 50%) are found to reference files belonging to other users; some group programming efforts and system utilities (such as *finger*) were the reasons for this result.

The next section discusses the related work in this area. Section 3 describes the data and its collection method. Sections 4 through 7 discuss the user and file characterizations in detail. In section 8, we briefly speculate on how the results might be used in file system design. Summary and conclusions appear in section 9.

2. Related Work

Related work can be categorized as the long term and short term file usage studies. The long term studies analyze data from once-a-day scans of the file system. The scans of the file system record whether or not a file is referenced on a day. Consequently, the studies such as [Smith 81] and [Satyanarayanan 81] do not quantify how heavily a file is used during a day. A comprehensive review of long-term studies can be found in [Satyanarayanan 81].

The short term studies analyze traces of disk I/O requests or system calls. Based on traces of disk I/O requests from two IBM batch systems, in [Porcar 82], an approach for shared file migration assuming a Markov chain model for the file usage is described. In the model, each state corresponds to a node in a computer network. In calculating model parameters, aggregate referencing behavior of all users is used. As the analysis in this paper will demonstrate such an assumption is not valid in general. Some users can vary significantly from the norm in their referencing characteristics. Consequently, model parameters can also vary for these users, and thus affect the validity of the overall model in a dynamic sense. Since no attempt was made to validate the Markov model itself, the impact of user variability on the results is unknown. Another study of short term file access [Ousterhout 85], mainly analyzes disk cache performance.

The study closely related to the present one is that in [Floyd 86a] and [Floyd 86b]. Using short term file access data from a 4.2 BSD UNIX environment, the author provides distributions of measures such as fraction referenced, file-open time, inter-open time, and number of references per file. This broad analysis of references to all types of files, also brings out the value of a short term file usage study. As the author points out, an important issue, which may enchance the value of this work is an in-depth analysis of file usage activity by user.

None of the short term studies explicitly quantify a typical user's file usage. As will be shown, user-based and file-based measures quantified in this paper are useful in bringing out differences in users (and in files), and these differences can be important in evaluating an existing system. The work presented here is unique in the following respects:

- The notion of *how heavily a file is used* is quantified.
- A typical user's file usage as well as usage of a typical file by all users are quantified.
- The above two ways of characterizing file usage are shown to distinguish nonuniformity in file usage.
- Properties specific to file categories (e.g. user-owned, notes files,[1] and others) and user categories (light and heavy) are evaluated.
- Analysis of variance methods are used to evaluate the relative influence of the user and file categories on usage characterization measures.

3. Data Description

The data analyzed in this paper has been collected from a VAX-11/780 running 4.2 BSD UNIX. The system is used by the faculty and graduate students of the Department of Computer Science, University of Illinois at Urbana-Champaign, for text editing, sending and receiving mail, and for research programming. About 400 logins were recorded per day, but at any time the system only has a maximum of 40 users.

File-related system calls and their arguments were traced on a continuous basis. The data collected was at the system call level rather than at disk I/O level, because the intent of this paper is to analyze users' file usage that is not influenced by the caching policy or by the level of multiprogramming in the system. The data was collected from 8:00 a.m. to 12:00 midnight on Monday through Friday, each day being selected from a different week. The hours capture the typical working hours of most users. We chose the five days of data collection randomly from five different weeks so that the data represents a good sample of users' activity. From the trace, the following data for each file-login session combination was obtained:

[1] Notes files are described later in the paper

User identification data:

- user id
- login process id

File specific data:

- file id (inode, device, and *usage* numbers)
- file size
- file owner's id
- file type information

File usage data:

- number of reads
- average bytes read in each read call
- number of writes
- average bytes written in each write call

Time stamps:

- software clock value at the first and last call

In UNIX, the real user id and login process id together identify a login session. For the sake of simplicity, the word *user* is used to mean a login session of a user in this paper. The inode (which contains the disk addresses of the data blocks) and device numbers, that are provided by UNIX, do not uniquely identify a file in the trace because inodes can be re-used. To combat this problem, each inode-device number pair was complemented by a usage number.

The data analyzed is limited to users' data files and to files belonging to the Notes[2] file system. Specifically, it was decided not to include calls to command files and system

[2] Notes is a multi-topic bulletin-board-like system. Messages for a topic are stored together in one file; users can selectively read messages and can also add new messages. See [Essick 84] for more details. Similar bulletin-board systems are available on most computers, and in some installations, the system is known as News and has a slightly different implementation.

files (operating system related log, database, and library files) in this analysis. The exclusion was achieved by filtering out calls to files owned by the system identifiers *root* and *bin*. The reasons for the exclusion are detailed below.

Command files are the load modules containing executable programs. Once execution of one of these files begins, the virtual memory system is responsible for making pages of the program available in memory. Paging behavior of programs has been extensively studied elsewhere, and it is not our objective to duplicate this work.

Here, we are primarily concerned with the analysis of users' files. The usage patterns of the system files can be substantially different from that of users files — system files are usually referenced only in part, although (sometimes) heavily. An example is the file that contains users' passwords and other related information, */etc/passwd*. As it will become apparent in the subsequent sections, users tend to access their own files in entirety. Thus, the inclusion of the system files in our analysis can significantly distort the overall results.

Further, the referencing patterns of the system files can depend on the specific implementation of the operating system. For example, in Version 4.3 of the Berkeley UNIX the password file is searched by hashing, where as a sequential search is employed in 4.2 version. In SUN Microsystems UNIX, most system databases are implemented using centralized server processes. Given that the referencing patterns of system files are different from user files and that the referencing patterns of the system files can change from one implementation of UNIX to another, we believe that the system files should be studied separately.

The user files, by their very nature, are independent of implementation. Therefore, the analysis of the user files can be of considerable value in creating a synthetic workload that is useful for any system. It should be emphasized that the key issue in this paper is methodology, and the method is equally applicable to the analysis of the system files.

In summary, the data used in this study is traced from a university research environment, and consists of file-related system calls to system-independent files,[3] namely the users' data files and notes files.

4. File Usage Characterization

In this section, we introduce two types of characterizations of file usage. A *user characterization* quantifies how a user uses an average (referenced) file in a login session, and a *file characterization* quantifies how a file is used by an average user in the measurement period. Alone, neither the user characterization nor the file characterization fully captures the many-to-many relationship between users and files. For instance, the user characterization does not show file sharing among users, but the file-based approach does. On the other hand, the file characterization does not show variability in users, which the user-based approach quantifies. In addition, as will be shown later, the two ways of characterizing the same data allow us to quantify the nonuniformity in file access.

A key measure central to both characterizations is what will be referred to as the number of *accesses-per-byte* (*APB*). Given a login session s and a file f, the APB for the specified file in the login session is defined as:

[3]Indirect references to directories for file name translation are also excluded. The argument for the exclusion is similar to the one given for the system files. This indirect use of directories is quite different from the normal usage, and the implementation can change from one system to another. Consequently, these indirect references should be studied separately, as is done in [Floyd 86b].

$$Accesses\_Per\_Byte[s,f] = \sum_{i=1}^{NumOpens} FR[s,f,i] \qquad \text{Eq. 4.1}$$

where, $FR[s,f,i]$ is the fraction of the file referenced in *ith* open of the file, and *NumOpens* is the number of opens made to the file in the login session. Intuitively, the measure shows how many times a file is completely referenced by a user in a login session, and thus quantifies *how heavily a file is referenced*. As it will be seen, this measure allows us to clearly classify who are heavy users in the system.

If the fraction referenced, for a given file, is always 1.0, then the APB shows number of references made to the file. However, if only one reference is made to the file in a login session, then the APB, in common with other file access studies [Porcar 82; Floyd 86a], measures the fraction referenced. But unlike these studies, accesses-per-byte (as it combines fraction referenced and number of references) also provides information on how heavily a file is used in a given period of time. Our data shows that in nearly 92% of references, the referenced files is accessed in entirety. For files not referenced in entirety but referenced many times, such as operating system related log and database files, the APB (in Eq. 4.1) should be calculated for each record of the database.

We considered alternatives to the accesses-per-byte, such as accesses per logged-in minute and accesses per day, but found them not to reflect a user's file usage characteristics. For example, accesses per logged-in minute may depend on the system load. If a user's login session occurred when the system load is high, then the user's accesses per minute measure can be significantly lower than what it would be if the user were logged-in at a low system load. Thus, accesses per minute may be more reflective of the system usage than a user's file usage. Another point of importance in this regard is that, as will be shown later in the paper, if a user's total file I/O in a login is high then

(a) the user's file I/O rate is also *likely* to be high, and (b) the user's accesses-per-byte is also *likely* to be high. Consequently, if a user's APB is high it is *likely* that the user's file I/O rate is also high. So, since the accesses-per-byte measure reflects file I/O rate to a large extent without actually being influenced by the system load, we chose to use it as the characterization measure of a user's file usage.

The other alternative, accesses-per-day, may encompass too much of a user's activity, and thus it may suppress the variability in usage. For example, a user may login several times during a day, doing different things in each login, and these differences will be averaged out in accesses-per-day.

One can ask: why analyze file usage by user and by login session? Most current literature does not do so. For example, the study in [Porcar 82] assumes that all users are alike. As we will show, the distributions of file usage measures can be heavily skewed by a few, but significant number of heavy users. In such a case it is invalid to assume uniformity among users. In fact, in analyzing user behavior, we found that users can indeed behave differently in different login sessions. Thus, it was considered statistically sound to treat each login session separately. Finally, one application of this analysis, synthetic workload creation, needs user-based as well as file-based analysis.

Based on the accesses-per-byte measure and a few other parameters, we define the user and file characterization measures:

User Characterization: Each user is characterized by the average number of accesses-per-byte made to referenced files, the average size of the referenced files, and the number of files referenced in a login session. Mathematical definitions[4] for the characterization

[4] *Notation:* In the mathematical expressions, *accesses_per_byte[i,j]* denotes accesses per byte made to jth referenced file by ith user. A "*" in the place of an index indicates a quantity obtained by averaging over

measures of *ith* user with N_i files follows:

$$accesses\_per\_byte[i,*] = \frac{1}{N_i}\sum_{j=1}^{N_i} accesses\_per\_byte[i,j]$$

$$file\_size[i,*] = \frac{1}{N_i}\sum_{j=1}^{N_i} file\_size[i,j]$$

$$num\_of\_files[i,*] = N_i$$

File Characterization: Each file is characterized by the average number of accesses-per-byte made by all logins in the measurement period, its average size, and the number of users of the file. Mathematical definitions for the characterization measures of *jth* file with M_j users follows:

$$accesses\_per\_byte[*,j] = \frac{1}{M_j}\sum_{i=1}^{M_j} accesses\_per\_byte[i,j]$$

$$file\_size[*,j] = \frac{1}{M_j}\sum_{i=1}^{M_j} file\_size[i,j]$$

$$num\_of\_users[*,j] = M_j$$

4.1. Distributions of the Characterization Measures

In this subsection, distributions of the user and file characterization are provided, with intuitive explanations for the results. Statistical models to fit the distributions are also provided. Figures 4.1 and 4.2 show the distributions and the multi-stage gamma functions (g's in the figures) model the distributions. Mean and quartiles of the distributions appear in Table 4.1 and Table 4.2, where the parenthesized values are the standard deviations of the parameters across the five days of measurement.

the index. Similar notation is employed for other measures

Representativeness of data is evident from small standard deviations.

As seen in Figure 4.1, distributions of the user-based measures are skewed towards small values, and they also have long tails. This is also evident from the fact that mean values are larger than their median values but are smaller than third quartiles. It implies that even though there are many light users, a significant number of heavy users also exist. Since these heavy users make severe demands on the system, all users can experience poor response times when a heavy user is active (assuming shared resources). From a file system designer's viewpoint it is important to differentiate these heavy users so that the file system can be designed to adapt to different workloads. From a performance evaluator's viewpoint, such a characterization helps to accurately evaluate the system performance under heavy and light loads.

The user-based file size distribution (Figure 4.1) shows two peaks, the second peak occurs near 14K bytes. However, the other measures show little difference between the users with mean file size greater than 14K and those with mean file size less than 14K. A further examination reveals that the users belonging to the former group referenced mostly notes files, which are considerably larger than the other files. This group accounts for about 45% of the total users.

Distributions of the file-based measures (Figure 4.2) have even longer tails than distributions of the user-based measures. For instance, the mean of the file-based accesses-per-byte is larger than its 3rd quartile. The file-based file size distribution (Figure 4.2) shows dominance of small files in a UNIX environment. About 80% of all files are smaller than 10K bytes. Studies of long term file reference patterns (for example, [Smith 81] and [Satyanarayanan 81]) reported similar file size distributions.

Table 4.1 : Means and Quartiles of User Characterization Measures

measure	mean	median	III quartile
accesses-per-byte	1.57 (0.06)	1.34 (0.04)	1.78 (0.11)
file size	14.57k (1.318)	9.75k (0.433)	24.12k (2.96)
number of files	27.94 (2.09)	15.60 (1.14)	33.55 (3.34)

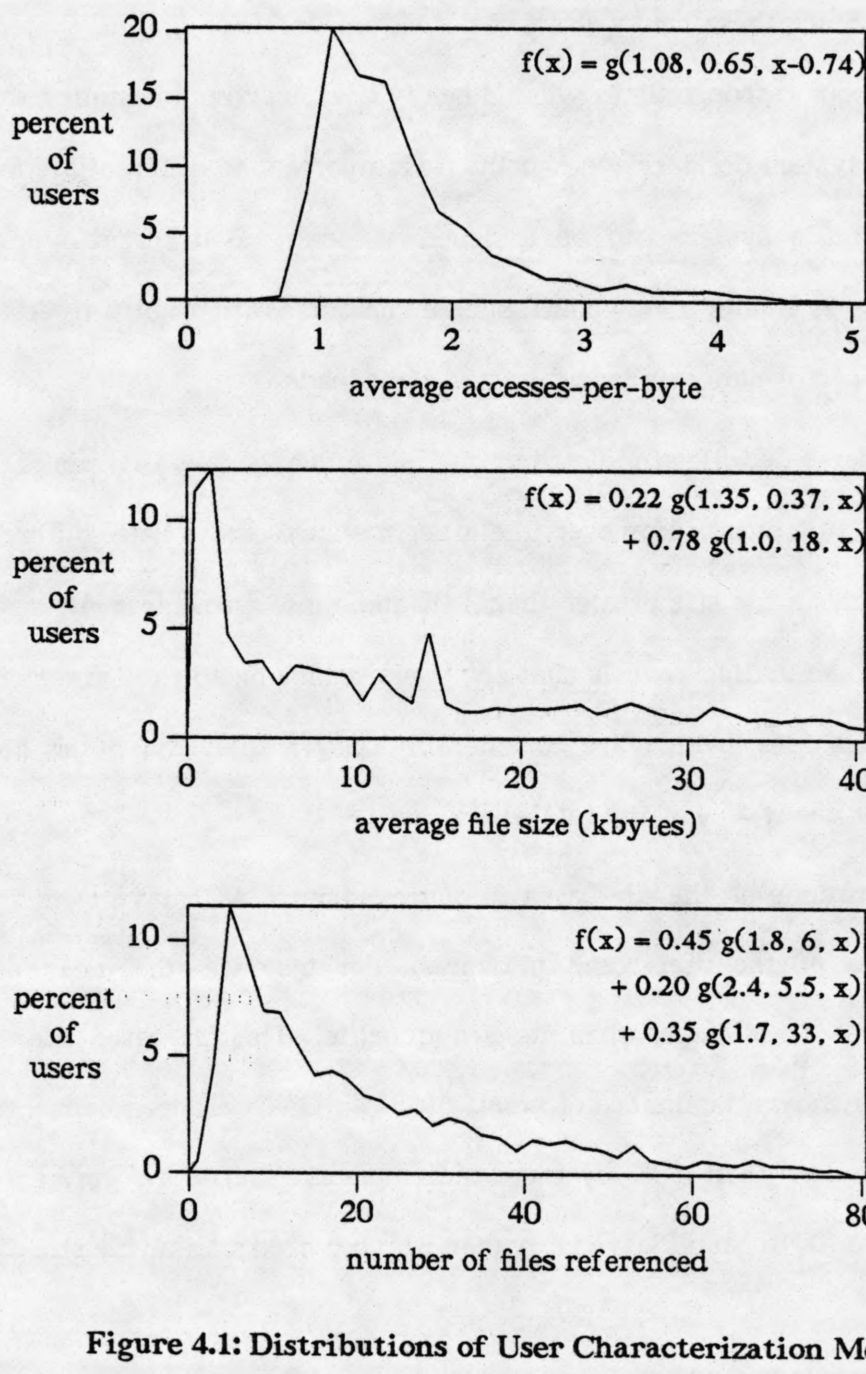
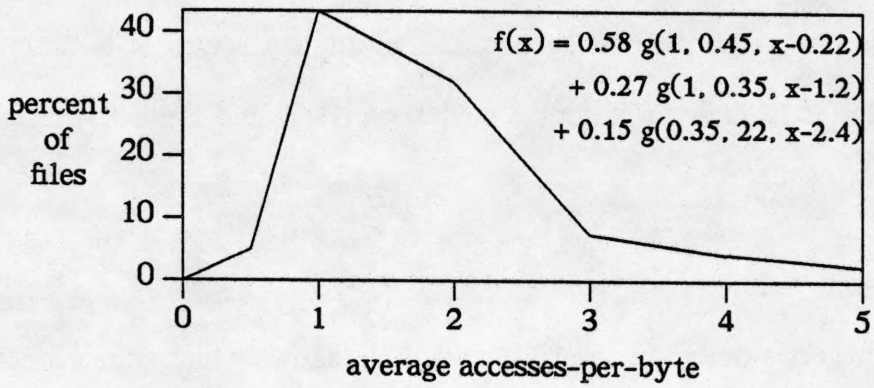
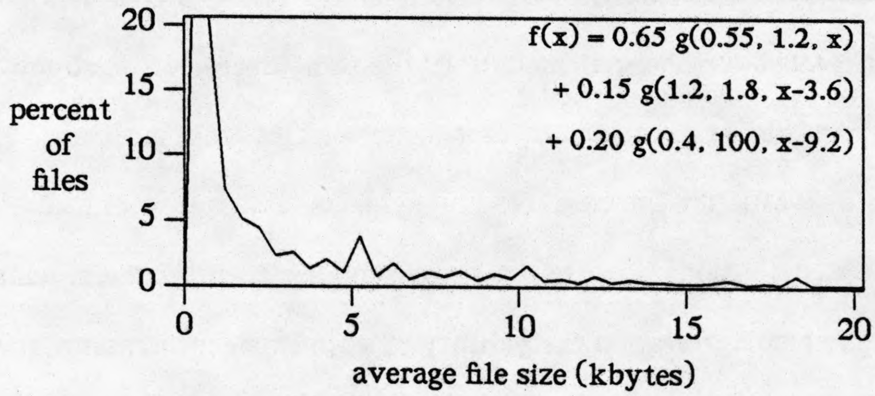


Figure 4.1: Distributions of User Characterization Measures

Table 4.2: Means and Quartiles of File Characterization Measures

measure	mean	median	III quartile
accesses-per-byte	2.35 (0.09)	1.66 (0.12)	2.00 (0.00)
file size	11.38k (1.54)	1.42k (0.22)	7.03k (0.76)
number of users	2.00 (0.11)	1.00 (0.00)	1.4 (0.55)





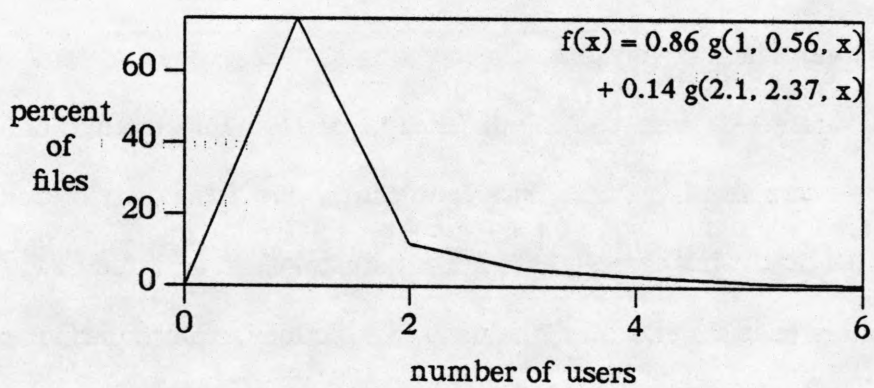


Figure 4.1: Distributions of File Characterization Measures

Owing to the long tails and multiple modes, the empirical distributions are modeled by multi-stage gamma distributions. The probability density functions appear in figures 4.1 and 4.2 as:

$$f(x) = \sum_{i=1}^{N} w_i \, g(\alpha_i, \theta_i, x - s_i)$$

where w_i is the weight, and s_i is the offset of the *ith* stage. N is the number of stages. Sum of all w_i is 1. G is the gamma distribution [Hogg and Tanis 83] function:

$$g(\alpha, \theta, y) = \frac{1}{\Gamma(\alpha)\theta^{\alpha}} y^{\alpha-1} e^{\frac{-y}{\theta}} \quad 0 \leqslant y < \infty$$

The Kolmogorov-Smirnov test [Daniel 78] shows that the multi-stage gamma distribution model the empirical distributions at over 99% confidence level. We could not fit multi-stage exponential models to the same degree of accuracy. Clearly, single stage exponentials are not valid representations of the measures. Most analytical performance evaluation studies of file systems assume workload parameters have exponential distributions because the system models then become numerically tractable. However, our results question the validity of such exponential assumptions.

In summary, distributions of the user and file characterization measures follow multi-stage gamma distributions. Hence, single stage exponential models appear to be invalid for these — a result of significance in performance evaluation. Also, there are some heavy users and large files that significantly effect the distributions, which clearly demonstrates that using aggregates is not satisfactory. In an attempt to further quantify the differences in users and files, the next two sections explore various categories of files and users.

5. Effects of File Categorization

So far we have obtained distributions of the user and file characterizations. How these characterizations change with different *file categories* is brought out in this section. In particular, we examine how a user uses files belonging to different categories, and how a file belonging to a given category is used in all login sessions. Further, a comparison of the corresponding measures of the user and file characterizations shows nonuniformity in file access. For the purposes of this study, files are categorized using the following orthogonal criteria:

1. **UNIX file type:** A file may be a directory (DIR) or a regular file (REG). This criterion groups the files according to the implicit use of the files in the operating system.

2. **Ownership:** A file of the notes file system belongs to NOTES type, a user-owned and owner-referenced file belongs to USER type, and a user-owned nonowner-referenced file belongs to OTHER type.

3. **Type of Use:** A file whose contents are only read during a login session belongs to RDONLY class. A file that is either nonexistent before or truncated to zero size before writing belongs to NEW class. A file that is nonexistent before and deleted after use is a temporary (TEMP) file. A file that is neither RDONLY nor NEW nor TEMP belongs to RD_WRT class.

A file category[5] is defined as a specific combination of UNIX file type, ownership, and type of use. For example, REG-USER-RDONLY refers to user-owned regular files that are used in a read-only mode in a login session. If the context is clear, a shorter name (e.g., while discussing regular files, REG-USER-RDONLY may be abbreviated as USER-RDONLY) is used to reference a file category.

[5] Note that how a user uses a file is the basis for the ownership and type of use classifications. Consequently, a file can be in more than one class. An examination of the data shows that about 5% of the files belong to more than one category. In developing file characterization, we consider such multiple occurrences of a file as occurrences of multiple files.

5.1. User Characterization by File Category

This section discusses how a user uses files belonging to different categories, and the next section discusses how a file belonging to a given category is used in all logins. Table 5.1 shows the mean values of the user characterization measures by file category. (Figure A.1 shows distributions of the user characterization measures for selected file categories.) For example, an average user's usage of a REG-USER-RD_WRT file is characterized by 3.46 accesses-per-byte and 19796 bytes of file size. On an average, 2.1 REG-USER-RD_WRT files are referenced in a login session. About 45% of logins reference files of this category.

An average user's usage of REG-USER files: An average read-write file is about ten times larger than an average read-only file, and is accessed 3 times as much. This is because, in UNIX, read-only files contain mostly default options, electronic mail

Table 5.1: Averages of User Characterization Measures by File Category

file category			characterizing measures			%users using the category
file type	owner	type of use	accesses-per-byte	file size	files	
DIR	USER	RDONLY	3.33	803	2.8	68%
	NOTES	RDONLY	2.41	6248	1.0	8%
	OTHER	RDONLY	2.28	1198	2.5	70%
REG	USER	RDONLY	1.38	1909	5.8	100%
		NEW	2.30	11323	4.0	40%
		RD_WRT	3.46	19796	2.1	45%
		TEMP	2.00	9233	9.7	60%
	NOTES	RDONLY	0.54	49856	10.1	53%
		RD_WRT	1.77	20254	5.7	38%
	OTHER	RDONLY	1.52	4280	3.0	51%

messages, and user defined type declarations. Therefore, the read-only files are usually small and are rarely modified. On the other hand, read-write files contain program source code, object modules, or text. As a result, they are relatively large and are frequently updated. These statistics indicate that migration or prefetching an entire file may be a more efficient strategy for all REG-USER files. Specifically for read-write files, a delayed write-back policy is worth considering, because these files are heavily used in a login session. However, reliability requirements may dictate regular write-backs to nonvolatile storage (disk), but during heavy usage periods, these write-backs can cause response time degradation [Johnson 87]. Thus, it is preferable to improve memory reliability instead of frequent write-backs [Georgiou 87].

An average user's usage of REG-NOTES files: Read-only and read-write files are the largest and the next largest (49856 and 20254 bytes). On an average, only 54% of a NOTES file is read in a login session. Even read-write files are not fully accessed (accesses-per-byte is 1.77). In contrast to the above, migration or a complete prefetch of these files is inadvisable as it would waste file buffer space as well as communication bandwidth. Thus, different policies are suggested for different file categories.[6]

An average user's usage of directories: As expected, an average USER or OTHER directory referenced in an average login session is only about 1K bytes. A user accesses directories two to three times as heavily as REG-RDONLY files, but the number of directories referenced is only half as many as regular files. This indicates that even a small per-user directory-cache can achieve very high hit ratios, and is worth investigating.

[6]Current implementations of UNIX use a single policy for all files.

Probability that an average user references a file category: The last column in Table 5.1 gives the probability that a user references a file of a certain category.[7] For example, the probability that a user reads one or more NOTES files is 0.53. Note that the categories are not mutually exclusive.

An average user's usage of other users' files: The last column of Table 5.1 also shows that there is a measurable degree of sharing[8] apart from NOTES files. Seventy percent of logins read directories and 51% read regular files that belong to other users. This unexpectedly large amount of sharing comes from two sources: first, there are a few research groups developing large software systems (e.g. a programming environment), and individuals involved in such projects share type-declaration files; secondly, UNIX provides utilities (e.g. *finger*) which enable a user to obtain information about another user by reading this other user's file (e.g. *.plan*). Interestingly enough, an average user accesses other users' files just as heavily as his own read-only files.

5.2. File Characterization by File Category

This subsection discusses how a file belonging to a given file category is used in all login sessions. Table 5.2 shows the mean values of file characterization measures by file category. (Figure A.2 shows distributions of the file characterization measures for selected file categories.) For example, an average REG-USER-RD_WRT file is characterized by 4.30 accesses-per-byte, and 17443 bytes of file size. On an average, a REG-USER-RD_WRT file is referenced in 1.4 logins. Files of this category constitute about 4.7% of all files.

---

[7] The last column of Table 5.1 shows that only 69% of users (i.e. 31% of users do not) read their own directories. At first it might seem improbable, but note that about 32% of users make file I/O less than 10K bytes (see section 6), and that our analysis does not include directory references made while translating a file name into an inode number.

[8] Does not necessarily imply simultaneous use.

Table 5.2: Averages of File Characterization Measures by File Category

file category			characterizing measures			%files in the category
file type	owner	type of use	accesses-per-byte	file size	logins	
DIR	USER	RDONLY	3.55	713	1.70	7.8%
	OTHER	RDONLY	2.21	708	3.43	3.4%
REG	USER	RDONLY	1.81	4524	1.83	21.5%
		NEW	2.54	11164	1.08	9.8%
		RD_WRT	4.30	17443	1.40	4.7%
		TEMP	2.00	12393	1.00	38.7%
	NOTES	RDONLY	0.80	31514	5.54	6.5%
		RD_WRT	2.68	19410	4.53	3.3%
	OTHER	RDONLY	2.36	8639	2.14	4.6%

The last column of Table 5.2 gives the breakdown of files into file categories. About 75% of files are regular files that are user-owned and -referenced, and an additional 7% are directories of the same category. A little less than 10% of files are NOTES files. Over 4.6% of files are nonowner-referenced user files. These percentages show that, although most files are exclusively referenced by their respective owners, a significant portion (nearly 15%) of files are shared. Dominance of read-only files is also apparent: About 72% of all the permanent files are referenced in a read-only mode.

Accesses-per-byte and file size appear in Table 5.2 as well as in Table 5.1, and the corresponding entries in both tables exhibit certain similarities. This issue will be further discussed in the next subsection. Here, the key issue is file sharing, we comment on three types of sharing among users.

Sharing via notes files: From the logins measure of Table 5.2, it can be seen that an average NOTES file is read in 5.54 login sessions. Considering that nearly 150 different users use the system every day (at a rate of about 2.7 logins per person), one would expect a typical NOTES file to be used in more logins than this. A visual examination of the data reveals the presence of several special purpose NOTES files (such as a NOTES file exclusively used by a small research group) that influenced the characterization.

Simultaneous sharing via notes files: A separate analysis of notes file usage for a single day showed that over 2% of notes files are shared simultaneously by two or more users. One file had 4 simultaneous users at one time, and another file had 2 simultaneous users on 16 occasions during a day. Note that 22% of notes files had 3 or more (not necessarily simultaneous) users during the day, and nearly 10% of these notes files had 2 or more simultaneous users. These results indicate that a few notes are heavily shared.

In the previous subsection, it was observed that a typical user does not access notes files heavily, but here we showed that a few notes files are extensively shared (simultaneous and otherwise). These results may have some implications when considering a distributed environment. For example, the results, when applied to such an environment suggest that the notes files (instead of being duplicated or buffered at each node) should probably be supported using centralized *servers* similar to what is done with the password files in SUN Microsystems UNIX.

Sharing via users' files: Table 5.2 also shows that a OTHER class (nonowner-referenced user class) file has 2.14 users. This result complements a related observation from the previous subsection, which indicates that an average login session references 3.0 files of the OTHER class. Thus, between the two, the user and file characterizations well quantify the degree of file sharing.

As the results indicate, in a single processor system, users do take advantage of the ability to access other users' files, which shows the value of integrating single-user workstations into a unified system. However, since the usage of the OTHER class of files is less frequent than the rest of the file categories, performance optimization for the OTHER files may not be a real concern. Thus, a simple scheme such as SUN NFS may be adequate, and extensive migration policies may be unnecessary in these situations.

5.3. Comparison of User and File Characterization Measures

Since the user characterization describes a typical user's usage of an average file, and the file characterization describes the usage of a typical file by an average user, the extent to which these characterizations are similar shows the uniformity in file usage. This point is brought out when tables 5.1 and 5.2 are compared with each other. Even though both tables display a similar trend, significant differences can be observed. The file characterization measures are reflective of heavy users, and the user characterization measures are typical of light users. For instance, accesses-per-byte measure in Table 5.2 (i.e. in the file characterization) is larger than in Table 5.1 (i.e. in the user characterization). In particular, the difference is about 35% for REG-USER files, and it is over 50% for read-write notes files. The reason for these results is that a heavy user tends to reference a large number of files, and consequently his activity influences the file characterization considerably. On the other hand, a majority of logins in the measured system are light, and consequently the user characterization reflects their behavior.

File sizes of REG-USER files also follow the pattern of the accesses-per-byte measure, but the NOTES files are an exception. For example, file size of a read-only NOTES file is about 50K bytes in Table 5.1, whereas in Table 5.2 it is only about 30K bytes. An explanation is that a few large NOTES files are read by many users, but since these files constitute only a small percentage of all NOTES files they do not influence the file characterization much. However, it implies that high throughput as well as fragmentation avoidance is needed for large files.

The next section introduces a user categorization, and discusses how the user categorization explains the nonuniformity in file access.

6. Effects of User Categorization

Based on logical file I/O done, we categorize users as casual, light, medium, heavy, and very-heavy. The logical file I/O of a user is the total number of bytes read from or written via the read and write system calls in a login session. Mathematically, it is:

$$File\_IO = ReadCalls * AvgReadSize + WriteCalls * AvgWriteSize$$

Table 6.1 shows the percentage of users in each user category. Note that the system usage is fairly heavy: Over 42% of users have done file I/O in excess of 100K bytes per login session.

Tables 6.2 and 6.3 show the user and file characterizations by user category. For the sake of brevity, the measures are shown only for the USER, NOTES, RDONLY, and RD_WRT file classes. Figure B.1 shows distributions of the user-based measures for user-owned files and for heavy and light users.

Table 6.1: User Categories by File I/O

user category	file I/O range	percent of users
casual	less than 1K bytes	8.7%
light	1K - 10K	23.5%
medium	10K - 100K	25.1%
heavy	100K - 1,000K	33.8%
very-heavy	1,000K or more	8.9%

Table 6.2: Averages of the User Characterization Measures by User Category

measure	user category	values by file category			
		USER		NOTES	
		RDONLY	RD_WRT	RDONLY	RD_WRT
accesses-per-byte	casual	1.01	-	0.03	-
	light	1.06	1.67	0.29	-
	medium	1.22	2.12	0.55	1.26
	heavy	1.45	3.46	0.61	1.93
	v-heavy	2.46	6.06	0.75	2.03
file size	casual	158	-	24271	-
	light	354	10505	23743	-
	medium	1558	12064	46580	21554
	heavy	2829	18794	58419	19607
	v-heavy	5266	41777	62761	23320
number of files	casual	2.30	-	1.00	-
	light	3.32	1.06	1.4	-
	medium	4.93	1.90	3.50	2.23
	heavy	7.32	1.88	13.4	6.01
	v-heavy	12.33	3.52	23.9	10.34

Table 6.3: Averages of the File Characterization Measures by User Category

measure	user category	values by file category			
		USER		NOTES	
		RDONLY	RD_WRT	RDONLY	RD_WRT
accesses-per-byte	casual	1.02	-	-	-
	light	1.06	1.52	0.60	-
	medium	1.24	2.29	0.64	1.50
	heavy	1.53	3.43	0.75	2.58
	v-heavy	3.10	8.20	0.82	2.80
file size	casual	153	-	-	-
	light	357	8316	18217	-
	medium	1875	13650	47157	23323
	heavy	4086	16218	31511	16269
	v-heavy	7133	28994	42213	21155
number of users	casual	1.58	-	-	-
	light	1.43	1.29	2.18	-
	medium	1.42	1.22	2.18	1.92
	heavy	1.47	1.27	4.49	3.66
	v-heavy	1.25	1.21	2.50	2.26

A significant result from Table 6.2 is that the user characterization measures (i.e., APB, file size, and number of users) follow file I/O done by the user. For instance, a very-heavy user's usage of USER-RDONLY files is three to twelve times larger than that of a light user.[9] So, we conclude that the heavy usage can be quantified using any of the following measures: total file I/O, average accesses-per-byte, mean file size, or the number of files.

The blank entries in Table 6.2 are owing to the absence of certain file categories in the referenced files of a user category. For example, a casual user does not reference any read-write files. This information is part of a casual user's characterization. Turning now to Table 6.3 (the file characterization), it can be seen that accesses-per-byte and file size measures follow the same trend as in Table 6.2 (the user characterization).

Interestingly, a comparison of tables 6.2 and 6.3 shows smaller differences in the user and file characterization measures than in section 5.3, where user categories were not used. For example, differences in accesses-per-byte of REG_USER files are now about 8% compared to over 35% differences noticed in section 5.3. Similarly, differences in file sizes of REG-USER-RDONLY files are now about 35% compared to 120% earlier.[10] This closeness between the user and file characterization shows uniformity in file access among users of a user category. Recall that in section 5.3, the differences between the user and file characterizations were attributed to the nonuniformity in file access, and it was claimed that the user categories would reduce the nonuniformity. By making the users more uniform in each category, we have reduced the nonuniformity in each user category, thus, providing support to the claim made. These patterns are also apparent in

[9]The only exception to this pattern is that a heavy user's NOTES-RD_WRT files are smaller than a medium user's files of the same category. This exception is partly responsible for the secondary peak in the file size distribution of Figure 4.1.

[10]Once again, an exception to this pattern is the size of the NOTES files.

figure B.2, which shows distributions of the file characterization measures for user-owned files and for heavy and light users.

6.1. Correlation Between a User's Total File I/O and I/O Rate

Earlier in this section, the total file I/O done by a user was used to group users into heavy and light users. One could argue that a user's file I/O rate may be more significant than the total file I/O. Here, we show the correlation between a user's average file I/O rate and total file I/O. In Figure 6.3, each user is denoted by a dot based on the user's file I/O rate and total file I/O done in a login session. A user's file I/O rate (bytes per second) is the average number of bytes read or written in a unit of login time. As shown

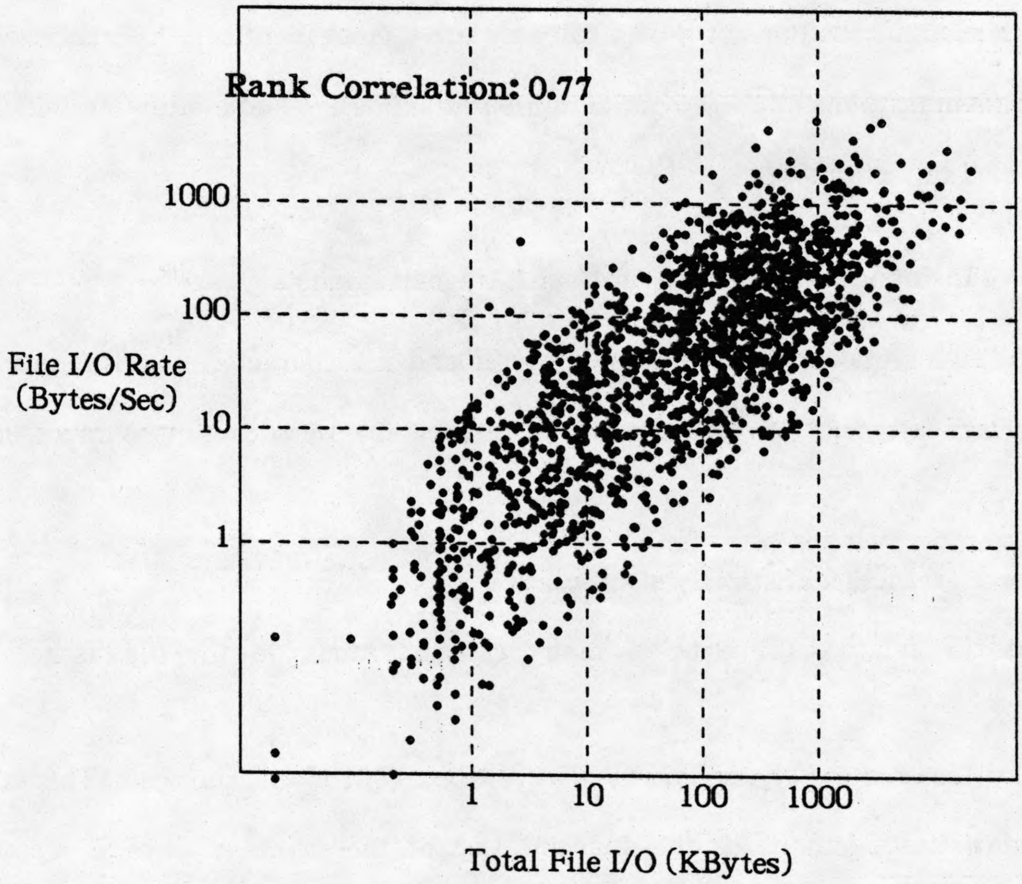


Figure 6.3: Users' Access Rate versus Total File I/O

in the figure, the coefficient of (Spearman's) rank correlation [Mendenhall and Sincich 84] for the two measures is 0.77. The rank correlation quantifies the relationship between the ranks of two quantities, and it shows how well high values of one measure correspond to high values of the other, without assuming a linear relationship between the two. A coefficient value of 1.0 implies a perfect correlation. Given that a coefficient value of 0.77 was observed, we can conclude that it is unlikely that a user categorization based on file I/O rate would be considerably different from the one based on total file I/O.

In summary, an average user's characterization measures (average accesses-per-byte, average file size, and number of files) follow total file I/O done by the user. Also, the user and file characterizations of a user category are almost identical, differences are as small as 8%. Applications of these results in file system design and evaluation will be (briefly) discussed in section 8.

7. The Relative Influence of the File and User Categorizations

In the last two sections, differences in the user and file characterization measures across file and user categories were quantified. In this section, we address two important questions:

- Are these differences statistically significant?
- What is the relative influence of many categorizations on the file usage measures?

We employ the *analysis of variance* (ANOVA) [Box 78] for this purpose. This is a well known statistical method for the quantification of the effects of several *factors* (e.g., file categorization criteria) on a *response variable* (e.g., accesses-per-byte). A linear dependency between the response variable and the factors is assumed, as in the following example:

$$Y = A + B + C + A\&B + A\&C$$

where A, B, and C are the factors and T is the response variable. A&B and A&C represent the interaction effects of A combined with B and C respectively. ANOVA decomposes the sum of square variations in Y (denoted by SST) into sum of square components of the terms on the right hand side of the model equation (SSA, SSB, and so on), and a residual error (SSE). The ratios, SSA/SST, SSB/SST, ..., and SSAC/SST, show the relative influence of the terms. The fraction SSE/SST represents unknown variations in the dependent variable. From the sum of square components, *significance levels* for the model and for each factor of the model are derived. The smaller the significance levels, the better the fit. For each measure, using mean values, an ANOVA model was obtained at better than 0.0001 level of significance. The model was analyzed using SAS, the Statistical Analysis System [SAS 85a; SAS 85b].

Table 7.1: ANOVA models for the file characterization measures and percent sum of squares contributions from the factors

source of variations (factors)	model for accesses-per-byte	model for file size	model for users
file_type	3%	23%	7%
ownership	19%	11%	50%
type_of_use	19%	-	4%
user_type	17%	-	11%
file_type&ownership	2%	14%	8%
ownership&type_of_use	-	36%	1%
user_type&file_type	-	16%	-
user_type&ownership	21%	-	-
user_type&type_of_use	13%	-	-
user_type&file_type&ownership	6%	-	18%
R-Square	0.78	0.74	0.89

Each column in Table 7.1 shows an ANOVA model for a characterization measure — a nonblank entry implies the presence of the corresponding categorization (or an interaction of categorizations) in the measure's ANOVA model. For instance, an ANOVA model for accesses-per-byte is shown below:

$$accesses\_per\_byte = file\_type + ownership + type\_of\_use + user\_type$$
$$+ file\_type\&ownership + user\_type\&ownership$$
$$+ user\_type\&type\_of\_use + user\_type\&file\_type\&ownership$$

The relative influence of the categorizations are shown as percent sum of squares explained by each categorization (or an interaction of categorizations). A large percentage implies a heavy influence. As the results indicate, the variations in the characterization measures are statistically significant.

We find that the user type has the largest influence on accesses-per-byte. Alone, user type contributes 17% to variations in accesses-per-byte, and interaction terms involving user type contribute an additional 40% to variations in accesses-per-byte. Ownership of a file and type of use also figure significantly in explaining the variations in accesses-per-byte.

File type and ownership determine the file size. File type and ownership contribute 48% to variations in file size, and the interaction terms involving file type or ownership with other categorizations contribute the remaining 52%. The number of users of a file is mostly determined by its ownership. Ownership alone contributes about 50% to variations in the number of users, and the interaction terms involving ownership contribute an additional 27%.

(The effects of the categorizations on user characterization measures were also analyzed for statistical significance and relative influence. The results are shown in Table C.1 of Appendix C.)

8. Implications of the Results

Throughout this paper we have obtained numerous results on both user and file characteristics, and discussed specific implications of these results. This section highlights important results and discusses possible implications for efficient file system design and evaluation.

A. Synthetic Workloads for File System Evaluation

The measures and distributions from this study can be used to develop a synthetic file access workload for evaluating the file system of a stand-alone or a networked system. Such a workload generator has been developed, and is described in [Barrington 86]. Briefly, the workload generator first populates disk(s) with files using the file size distribution of the file characterization. Next, the generator simulates several logins. Using a UNIX process, each login is simulated with specific file usage characteristics (average APB, average file size and number of files) that are taken from the user characterization. Actual read and write calls are issued to the simulated files, according to the distributions of the file characterization measures of the user type (heavy or light). Apart from recreating the measured file access characteristics, the generator can also produce a heavy or a light file access load by selecting a certain ratio of users from various categories (i.e. light, heavy, and so on). The information on sharing among users (via notes and user files) and file I/O rate is also useful in making the synthetic workload realistic. This synthetic workload is being used to evaluate file system performance and to evaluate some of the new policies discussed below.

B. Towards File System Design

Our study shows that the user-owned files are almost always completely referenced, but many notes files are rarely referenced in entirety, and they are quite large. These results suggest the use of different prefetch policies for different file categories. The fact that there is a large variability in file size distribution may have some implications for networked systems also. These results suggest the use of file transfer protocols that can efficiently transfer small amounts (few tens of bytes) as well as large amounts (few ten thousands of bytes), which is unlike, for example, TCP/IP.

This study also shows that only user-owned read-write files and heavy users' files are also likely to be referenced heavily. The heavy referencing suggests a limited use of a delayed write-back policy for these classes of files. Since regular write-backs can be a source of response time degradation (particularly, during heavy usage periods), such a policy coupled with recent improvement in memory reliability can be considerably beneficial. Further, the results point towards a way to improve the file replacement policy by combining the LRU policy with a selection criterion based on the category of a buffered file and the current status of its user (heavy or light). Such a replacement policy may increase file buffer hit ratios, without significantly impairing the response to other files and users, since our results show that these other files are unlikely to be referenced more than once.

The results on file size show that 80% of files are 10K bytes or smaller, implying that the translation of a file name into an inode number can be an important performance issue (as it was also pointed out in [Floyd 86a]) for the measured system.

It can be easily addressed with a small cache of name-to-inode mappings (as it is done in 4.3 version of Berkeley UNIX and in [Floyd86b]). Further, since an average user-owned directory is even smaller than 1024 bytes, a per-user directory cache of a few kilobytes might capture most references to directories.

The results on sharing may have some additional implications to how notes files are implemented in networked systems. It was observed that a typical user does not access notes files heavily (APB is about 0.54), but a few (about 20% of) notes files are extensively shared (simultaneous and otherwise). These results suggest that the notes files, instead of being duplicated or buffered at each node, should probably be supported using centralized servers similar to what is done with the password files in SUN Microsystems UNIX.

It should be noted that the Berkeley UNIX [Quarterman 85] addresses some, but not all the issued raised here. For example, from 4.2 version onwards, Berkeley UNIX uses a large disk block size to improve file reads from disk, and a sophisticated scheme to avoid disk space fragmentation [Mckusick 85] that could result from a large disk block size. As a policy, UNIX uses only a single block read-ahead [Ritchie and Thompson 78] (4.2 and 4.3 BSD versions only make the implementation efficient), and in that way, UNIX deals somewhat with the uncertainity of whether a file will be referenced in entirety or not. It is worthwhile to examine how these schemes compare with what we suggest here in future networks that may consist of 100's or 1000's of workstations as well as many superminis and file servers ([Devarakonda 85] and [Satyanarayanan 85]).

9. Summary and Conclusions

Based on the short term file access data collected from a 4.2 BSD UNIX, this paper quantified a typical user's file usage in a login session and the usage of a typical file in all login sessions. This approach is a departure from the traditional way of analyzing file references without actually characterizing either a user or a file. Two characterization measures were employed: accesses-per-byte (which combines fraction of a file referenced and number of references) and file size. It was shown that this new approach distinguishes differences in files as well as users. The multi-stage gamma were shown to model the file usage measures, which implies that the user demands cannot be assumed to be a single-stage exponential in performance evaluation.

Files and users belonging to various categories (based on ownership, type of use, UNIX file type, and file I/O) showed significant differences in their usage characteristics. More than 50% of users referenced files owned by other users, and over 8% of all files were involved in such references. Some group programming efforts and system utilities (such as *finger*) are the reasons for this result. Significant simultaneous sharing occurred only to notes files, and that too involved only about 3% of all notes files.

Finally, the file and user characteristics measured here have been used to generate a synthetic file access workload to evaluate file system design. Based on the differences in files and users, suggestions to improve file system performance were also made. As with any case study, caution is advised when using specific numerical results of this paper for other environments. More studies on other UNIX and non-UNIX systems are suggested, so that a wide range of such results are available.

ACKNOWLEDGEMENTS

The first author acknowledges the support and encouragement of Professor Roy H. Campbell. We thank Sharon Peterson, Luke Young, and Rick Eichemeyer for a careful proof reading of this paper. This research was supported in part by AT&T Metropolitan Networks Grant Number 1-5-13411, and in part by NASA Grant Number NAG-1-613.

REFERENCES

[Barrington 86] T. Barrington, "A Synthetic Workload Generator Based on the User-Oriented Analysis of File Usage," *EE 441 Project Report*, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, 1986.

[Berkeley UNIX 84] *UNIX Programmers Manual: Reference Guide*, 4.2 Berkeley Software Distribution, Virtual Vax-11 Version, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, March, 1984.

[Box 78] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters*, John Wiley & Sons, 1978.

[Daniel 78] W. W. Daniel, *Applied Nonparametric Statistics*, Houghton Mifflin Co., 1978.

[Devarakonda 85] M. Devarakonda, R. Mcgrath, R. Campbell, and W. Kubitz, "Networking a Large Number of Workstations Using UNIX United," *Proc. of 1st Intl. Conf. on Computer Workstations (IEEE)*, Nov. 1985, pp. 231-239.

[Essick 84] R. B. Essick IV, "Notesfiles: A UNIX Communications Tool," *Technical Report UIUCDCS-R-84-1165*, University of Illinois at Urbana-Champaign, Urbana, 1984.

[Floyd 86a] R. A. Floyd, "Short Term File Reference Patterns in a UNIX Environment," *Technical Report 177*, University of Rochester, March 1986.

[Floyd 86b] R. A. Floyd, "Directory Reference Patterns in a UNIX Environment," *Technical Report 178*, University of Rochester, March 1986.

[Georgiou 87] C. J. Georgiou, S. L. Palmer, and P. L. Rosenfeld, "An Experimental Coprocessor for Implementing Persistent Objects on an IBM 4381," *Second International Conference on Architectural Support for Programming Languages and Operating Systems*, Palo Alto, California, October 5-8, 1987.

[Hogg and Tanis 83] R. V. Hogg, and E. A. Tanis, *Probability and Statistical Inference*, Macmillan Publishing Co., 1983.

[Johnson 87] T. D. Johnson, J. M. Smith, E. S. Wilson, "Disk Response Time Measurements," *Winter 1987 USENIX Technical Conference*, Washington, D. C., January 1987.

[Mckusick 84] M. K. Mckusick, W. J. Joy, S. J. Leffler, and R. S. Fabry, "A Fast File System for UNIX," *ACM Transactions on Computer Systems*, Vol. 2, No. 3, August 1984, pp. 181-197.

[Mendenhall and Sincich 84] W. Mendenhall, and T. Sincich, *Statistics for the Engineering and Computer Sciences*, Dellen Publishing Company, San Francisco, California, 1984.

[Ousterhout 85] J. Ousterhout, H. Da Costa, D. Harrison, J. Kunze, M. Kupfer, and J. Thompson "A Trace-Driven Analysis of the UNIX 4.2 BSD File System," *Proc. of Tenth Symp. on Operating Systems Principles*, Dec. 1985, pp. 35-50.

[Porcar 82] J. M. Porcar, "File Migration in Distributed Computer Systems," *Ph. D. Thesis*, University of California, Berkeley, CA, July 1982.

[Quarterman 85] J. S. Quarterman, A. Silberschatz, and J. L. Peterson, "4.2BSD and 4.3BSD as Examples of the UNIX System," *ACM Computing Surveys*, Vol. 17, No. 4 (Dec. 1985).

[Ritchie and Thompson 78] D. M. Ritchie and K. Thompson, "The UNIX Time-Sharing System," *Bell System Tech. J.*, Vol. 57, No. 6, Part 2, July-August 1978.

[SAS 85a] *SAS User's Guide: Basics*, Version 5, SAS Institute Inc., Cary, NC 27511, 1985.

[SAS 85b] *SAS User's Guide: Statistics*, Version 5, SAS Institute Inc., Cary, NC 27511, 1985.

[Satyanarayanan 81] M. Satyanarayanan, "A Study of File Sizes and Functional Lifetimes," *Proc. of Eight Symp. on Operating Systems Principles*, Dec. 1981, pp. 96-108

[Satyanarayanan 85] M. Satyanarayanan, J. Howard, D. Nichols, R. Sidebotham, A. Spector, and M. West, "The ITC Distributed File System: Principles and Design," *Proc. of Tenth Symp. on Operating Systems Principles*, Dec. 1985, pp. 35-50

[Smith 81] A. J. Smith, "Analysis of Long Term File Reference Patterns for Application to File Migration Algorithms," *IEEE Trans. on Software Engineering*, Vol. SE-7, No. 4 (July 1981).

APPENDIX A

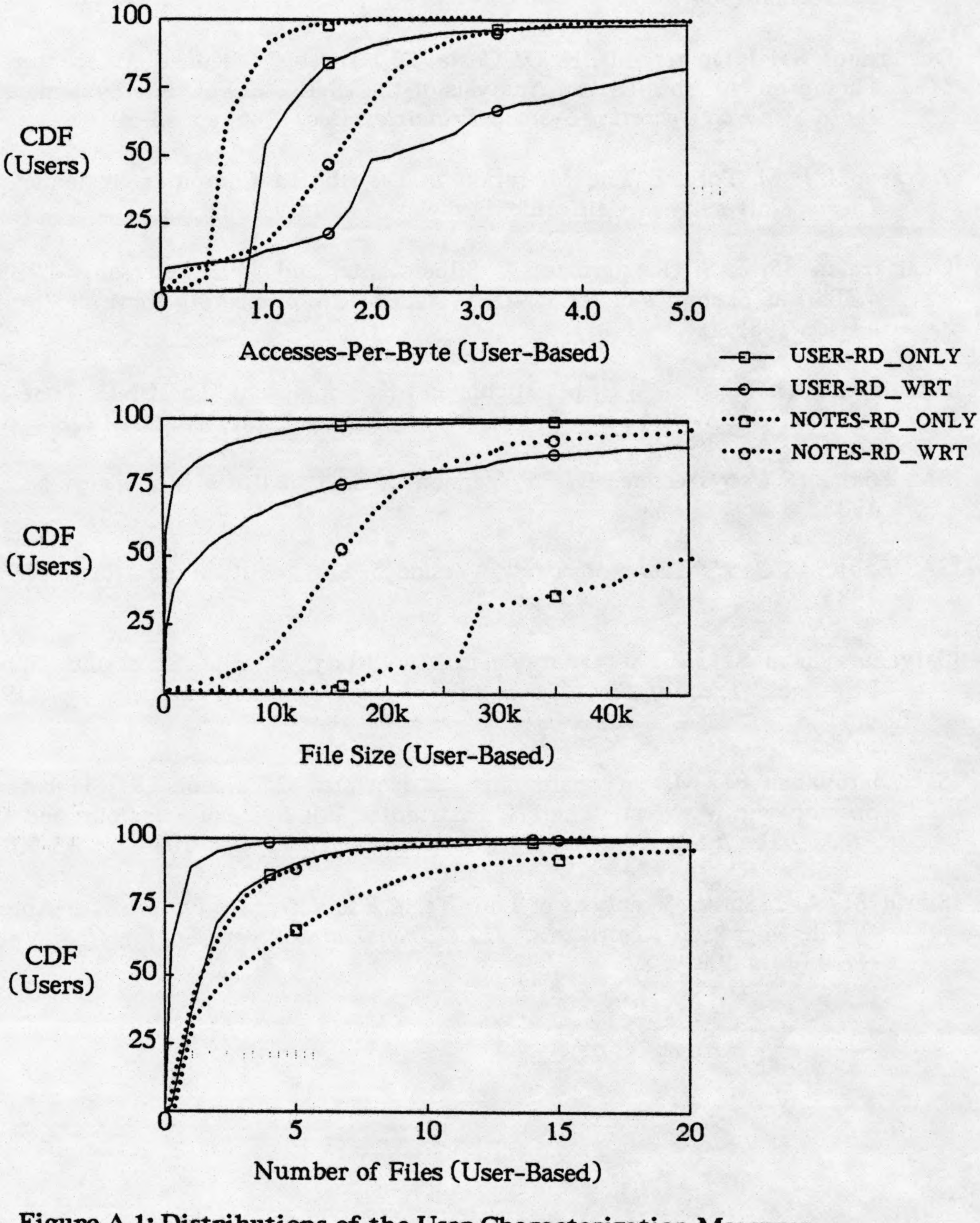


Figure A.1: Distributions of the User Characterization Measures

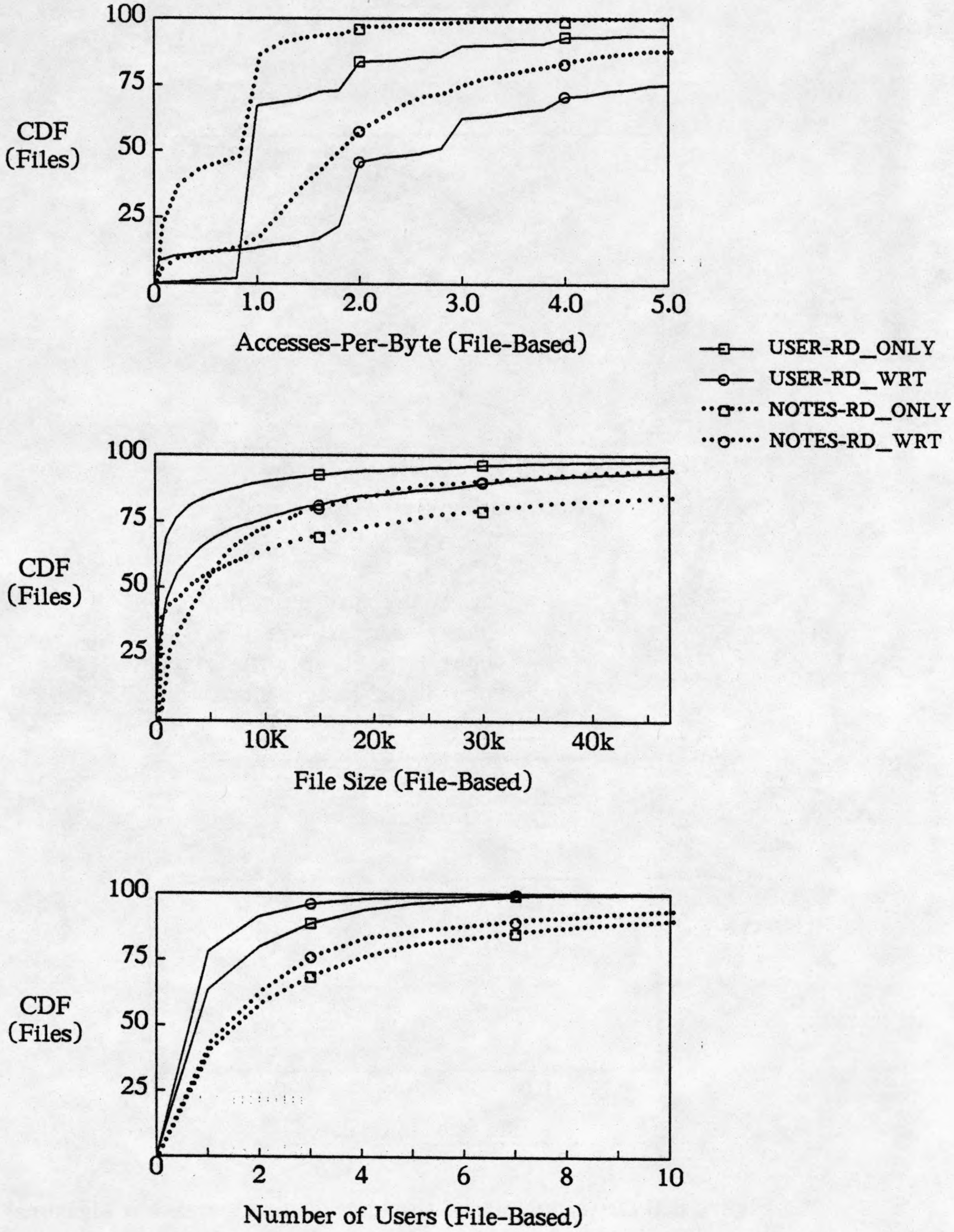


Figure A.2: Distributions of the File Characterization Measures

APPENDIX B

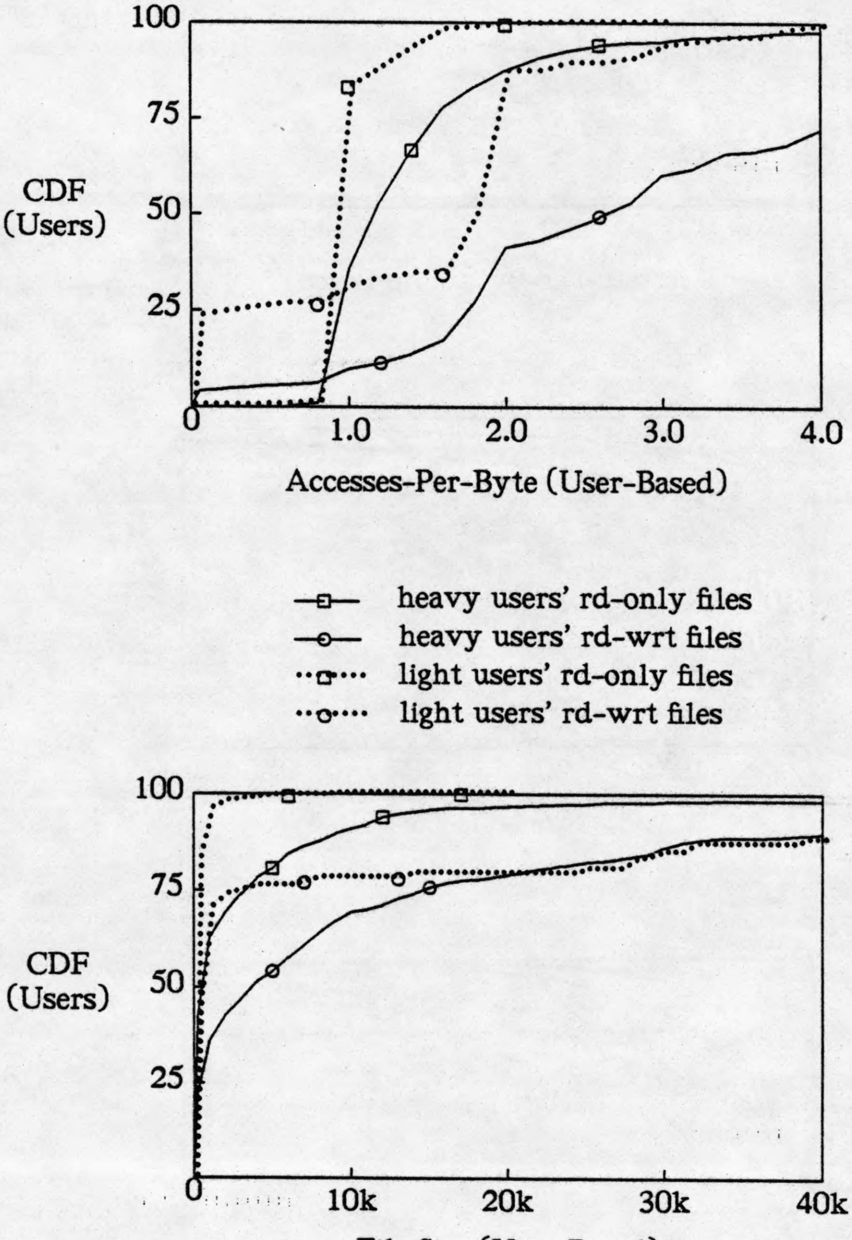


Figure B.1: Distributions of the User Characterization Measures

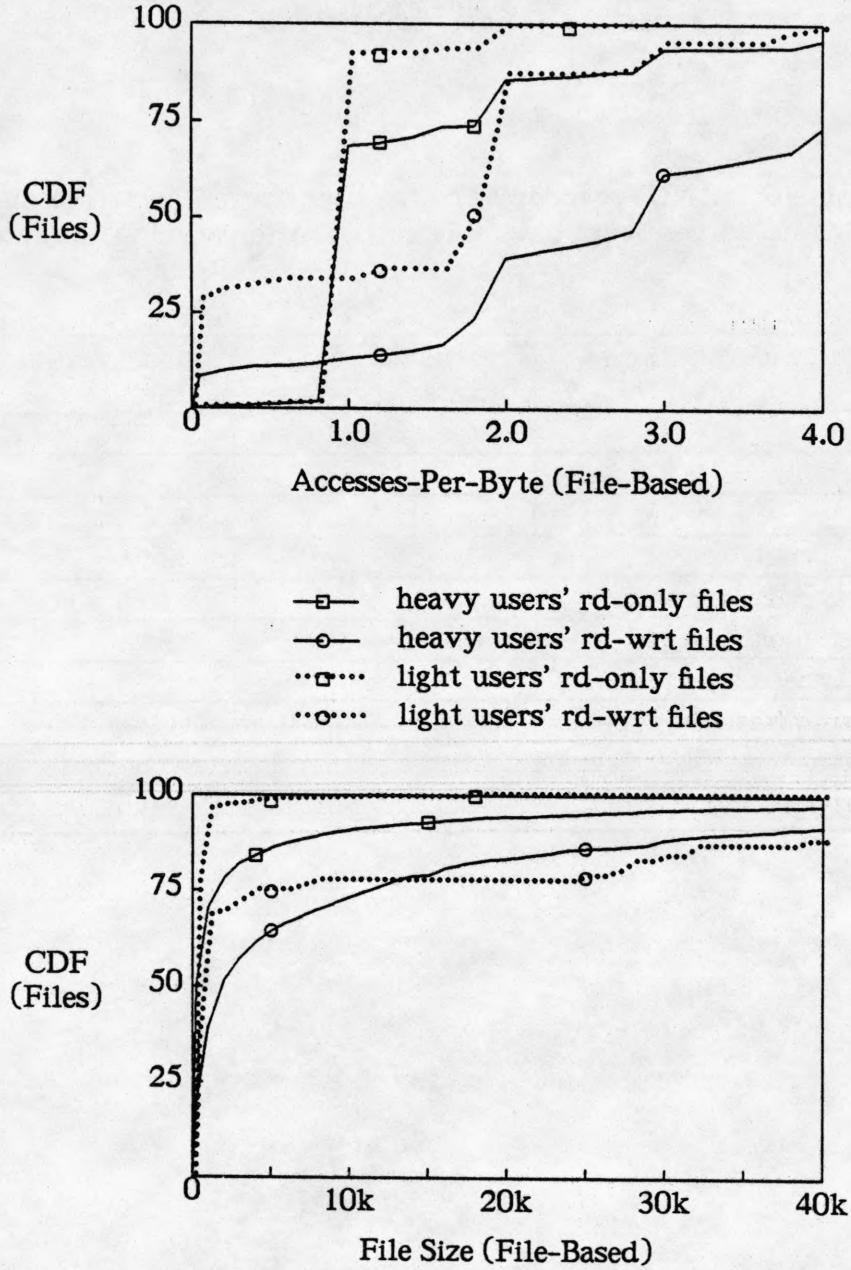


Figure B.2: Distributions of the File Characterization Measures

APPENDIX C

Table C.1: ANOVA models for the user characterization measures and percent sum of square contributions from the factors.

source of variations (factors)	model for accesses-per-byte	model for file size	model for files
file_type	8%	25%	8%
ownership	11%	3%	-
type_of_use	11%	5%	16%
user_type	50%	11%	34%
file_type&ownership	7%	15%	6%
ownership&type_of_use	-	34%	-
file_type&user_type	13%	7%	-
ownership&user_type	-	-	6%
type_of_use&user_type	-	-	30%
R-Square	0.62	0.83	0.85