Mobile Users are not Static Users on the Move

João Garcia[1], **João Leitão**[1], **Paulo Ferreira**[1]

[1]INESC ID Lisboa / Technical University of Lisbon
Rua Alves Redol 9, Apartado 13069, 1000-029 Lisboa, Portugal

Abstract. *This paper presents a measurement study of file system usage for a group of mobile devices users in different locations. The goal of this study is to dispel the idea that the behaviour of mobile systems can be modelled as being homogenous across various locations.*
The study shows that all volunteers access statistically significantly different sets of data as they move across locations. These results question the traditional way of evaluating mobile computing systems, which typically model the behavior of a user in different locations as being similar to her behavior in a single location for a longer period.

1. Introduction

Due to advances in mobile computer technology in the last decades, it is common for users to own several computing devices and use them to work in distinct locations. For instance, users frequently work on a desktop at the office, on a laptop at home or at a meeting location abroad, on a PDA in a public transport, etc.

In order to work on different devices and at different locations, the user usually has to explicitly transfer files between devices, an operation that can become very complex and cumbersome due to the huge amount and size of files that might be required to perform each task. Furthermore mistakes in selecting the data to transfer may lead to unnecessary storage and communication costs, or to situations where she cannot perform her work when relevant files were (mistakenly) not selected. One could imagine that a simple solution such as completely replicating the file system could address this challenge. However such a solution is not feasible as the storage capacity available varies with the type of device. Furthermore, the growth in the average file size may make such an approach extremely inefficient from the communication and energetic points of view.

In such a context, it is important to understand how people work, in order to be able to develop adequate solutions for this problem. In particular, there have been a series of proposals for automatic file hoarding between devices [Kuenning 1997, Satyanarayanan 2002, Tait et al. 1995, Andersen et al. 1994], which try to automate file transfer among devices, whereas the user only has to notify the hoarding system when she intends to stop using (and disconnect from) one device, thereby triggering a process of file selection and transfer to another mobile device. The evaluation of such systems has been based on modelling user behaviour by extracting traces from the operations performed by a user working on a single machine. Specifically, these systems are designed and evaluated by following one of two behavior models: i) uniformly distributed data accesses during a session,e.g. [Terry et al. 1995]; or ii) generation of multiple sessions by partitioning a trace from a static location and replaying it as several widespread sessions, e.g. [Kuenning 1997]. In the first case, user behavior on multiple devices is modeled by

doing multiple read and/or write accesses on a sequence of data elements where each element is randomly selected from the whole global dataset. In the second case, researchers use traces of previous work on a single location, slice them into multiple consecutive periods and assume these can be used to represent a sequence of work sessions in different locations and/or devices.

This paper presents a measurement study of file system usage for a group of users of mobile devices across different locations. In particular, we are mostly concerned with understanding how different are the sets of files that users access across different locations.

To this end we have developed a file access monitor software, which monitors the files accessed by a laptop user in his work sessions. We present preliminary results of a study we performed with a set of laptop users in order to better understand how people work in different locations. Namely, we analyzed how variable the accessed file sets are. To the best of our knowledge no previous work has studied or characterized this aspect of mobile device users.

Our results validate our intuition that users effectively perform different tasks in different locations. In particular, the set of files that a user accesses at a given location are very distinct from the set of files she accesses in other places, which reflects the fact that different tasks require access to different information and resources, and consequently, to different files.This study suggests that the design and evaluation of mobile computing systems should take into account behaviour adaptations due to location changes. This is specially relevant for systems that try to predict future file system accesses based on the recent access patterns in order to make decisions about which data to replicate on certain devices.

The rest of the paper is organized as follows: In Section 2 we describe the software we developed to perform the study, and provide a complete description on the conditions in which it was conducted. Section 3 characterizes the subjects that were involved in our study, presents the results we have obtained, and a discussion on their relevance. Finally, Section 4 concludes the paper proposing some directions for future work.

2. Tools and Methodology

In this section, we overview the architecture and operation of the software used to perform the study. Finally we describe the environment and conditions in which the study was performed and give some intuition on the data preprocessing.

2.1. File Access Monitor

The software we developed aims at creating a log containing user-initiated file system accesses with information on the corresponding location and used application. Furthermore, we group file accesses in *sessions*. We consider that a session is a set of file system accesses performed at a single location from the moment the computing device is turned on until it is disconnected.

We decided to perform the study using laptop users. It could be interesting to study the behavior of users in additional devices which are easier to use on the move, such as a PDA, the large amount of information that we require to store on disk (i.e. the

file access logs) could make such a study impractical on many smaller devices. Moreover, we also aimed at minimizing the impact of the monitoring software to the behavior of the user. Additionally, increasing he diversity of devices would almost certainly increase the diversity of user behaviour. Therefore, if the results using only laptops are significant, it is likely that apply to a more diverse set of devices.

The software was developed for the Windows XP and Windows Vista operating systems as a user service. We chose these operating systems because they are the most used therefore enlarging the population from which we could gather volunteers.

As depicted in Figure 2.1, our software has three main components namely: i) a *File System Monitor*, that detects all accesses to the file system; ii) a *Network Monitor*, that infers changes to location by detecting connectivity changes; and finally, iii) a *Log Manager*, that integrates and records all tracked activity. These components are described in some detail in the following paragraphs.

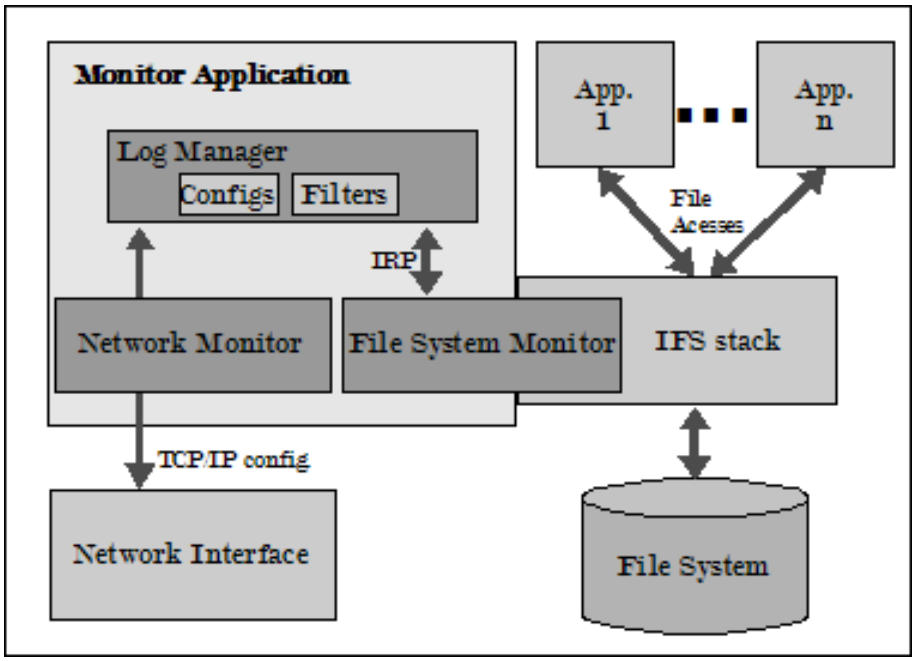

Figure 1. Overview of the File Access Monitor.

2.1.1. File System Monitor

This component is responsible for monitoring file system accesses and it is composed itself by a user-level component called the *observer* and a specially designed filter deployed in the *Installable File System* of NTFS. The *observer* starts its operation by reading a configuration file and initializing the filter in order to limit the information it extracts to a set of directories provided by the user, which contain their personal files and configurations.

Whenever a file is accessed the filter verifies if it belongs to one of the previously selected directories. In that case it reports the access to the observer which forwards the access record to the Log Manager to be further processed and stored in a persistent log.

2.1.2. Network Monitor

This component is responsible for extracting information regarding the user's current location. Ideally one would rely on an external source for providing the location of the user, such as a GPS device. Unfortunately, most laptops nowadays are not equipped with such devices making that approach unfeasible. To extract runtime information about the location of the user without being excessively intrusive, we rely on the current default gateway as an indicator of the current location. This is a reasonable solution as most users currently work within the access of a network, which includes a default gateway (and its IP address), enabling us to distinguish between different locations.

Our application associates a location with each gateway's IP address. Whenever the user connects to a gateway, if no association is known, the user is inquired about its current location. The user has then the opportunity to either choose a previously defined location name or to provide the system with a new one. This association is registered into a specific log through the Log Manager component. Because the user may sometimes work offline, whenever the network monitor detects that no gateway is defined it asks the user to provide its current location. This enables us to further improve the accuracy of the results extracted during the test phase. Whenever a change occurs to the current gateway, that event is registered through the log manager component.

Notice that this mechanism can still provide false information, as more than a location could use the same gateway, for instance, it is common for private network gateways to have an IP address in the 192.168.1.0/194 range. We believe that the impact of this to our study is still minimal, as the consequence of such wrong assignment is to merge the records of more than one location. Provided that other locations are registered, differences between the set of files accessed will still be found.

2.1.3. Log Manager

The Log Manager is the core of our file access monitor software and is responsible for the management of the permanent record storage. These records are divided in four categories: Known Locations, Location, Application, and File Access. Each category relies in a specific file to maintain its records.

The *known location* records maintain the association between gateway IP addresses and *location name*. The *location name* is a label provided by the user as discussed previously.

The location log records changes in the location of the user. Entries are added whenever the device is switched on or off, and when the network monitor detects a change in the gateway. The log keeps information concerning the time in which the event occurred and information that identifies the current location according to information in the known location log.

The log manager also tracks the applications used by the user at each moment. To this end, the log manager periodically polls the foreground application and adds an entry whenever the application that owns the focus changes. The log manager registers an entry in a specific log indicating the time, the process name and the id of the current foreground

application. We employed a small period of 500 ms for this polling process. This allows us to further eliminate records of file accesses performed by background applications that are not related with the behavior of the user. We provide more detail on this further ahead in this section.

Finally, the log manager has to maintain information about the user-initiated file accesses. This information is provided by the file system monitor component, and contains the information within I/O Request Packets (IRP) generated by application file system calls. Each entry contains several fields describing the type of record, the time of the access, process identifier of the application that generated the IRP, the type of the operation, and the full name of the accessed file.

Exceptions Preliminary tests to our software showed an abnormal high number of file accesses that did not correspond to the behavior of the user. This was mostly due to the operation of background applications that scan files in the hard disk. This is the case of applications that inspect or index files content (*e.g.* Google Desktop) or anti-virus applications (*e.g.* Norton anti-virus, BitDefender, etc), and also due to directories content scan performed by Windows Explorer File Manager when a user browses its directories. Such applications could provide incorrect results as file accesses would not be dependent on user activity. To overcome this, we added an additional filtering mechanism to the log manager component. Upon startup, our monitor reads from a configuration file a set of names of excluded executables. Whenever a file access event is processed by the log manager it simply discards accesses generated by those applications. Furthermore, we compiled an extensive list of applications that scan the hard disk and included them as exceptions in the configuration file that was distributed to the volunteers that participated in our study.

We also noticed that the file access log contained several sequential entries for each file registered. This phenomena happens due to sequential IRP requests to read a whole file. Although this additional entries did not compromise the accuracy of our results, they resulted in additional consumption of hard disk space by log files, which could be cumbersome to the user. To overcome this, we added a filtering rule to the log manager to avoid sequential entries for a single file.

Finally, files that were accessed by our monitoring application were also not registered in the logs. This was achieved by registering our application as an exception in the configuration file distributed to the users.

2.2. Test Conditions

Using the software described above, we performed a test, by asking to a set of test subjects to run our monitor in their laptops. In this section we offer some details on the test subjects, and describe how the test was conducted.

2.2.1. Test Subjects

As test subjects we tried to find people that used a single laptop as main work tool and that regularly moved between more than one location, to perform work there. We contacted

several people by using mailing lists: computer science students, information technology professionals, researcher and professors, and post-graduate students.

In more detail, we issued a request for volunteers for a study on how people used their laptops, that required a single laptop installed with either Windows XP or Vista operating systems, and on which the user worked at several locations (at least 2 or 3 different locations in a single week). We did not provide any further information on the goals of the test to avoid inducing volunteers to adapt or change their behavior.

2.2.2. Test Duration

The study was executed for a period of 10 days. During these 10 days the test subjects had our software installed in their laptops and used them in regular conditions. Our goal here was not to subject users to a long test period, and at the same time capture enough information concerning several locations where the user worked to have significant data. We also wanted to ensure that the test period would include a weekend, as this is a typical week period when users perform their work in different locations (for instance, at home).

None of our volunteers gave any indication that the test period was extensively long, nor that it would not be enough to capture information in the several places where each one of them worked.

2.2.3. Test Setup and Data Retrieval

We provided an installer for our monitoring application to all users. Furthermore, we helped most of users to adequately adapt the basic configuration file provided with the software.

After the conclusion of the test period, the users were instructed to send the log files to us, to uninstall the software and furthermore were given some details on the goal of the study. All users agreed with the use of their data after an anonymization process.

2.3. Processing Results

The first operation we performed upon the reception of the files by the users was to anonymize their data. To this end we used a Java application that replaced all filenames with a number code in all logs. Furthermore, we aggregated all accesses performed to a single file in a session to a single record, to simplify the data analysis. Moreover, as several volunteers used wireless networks, which connectivity was not constant, we also aggregated sessions in the same location that were separated by a small session where no gateway was detected (for this we considered a disconnection period below 5 minutes).

Because some users executed the test for more than 10 days, we decided to truncate all logs so that all contain information regarding exactly 10 days. Finally, we combined the information from the distinct log files to generate a table for each user, that stated the number of accesses to each file (observed for that user) in each of the observed locations.

3. Experimental Results

In this section, we characterize the volunteers that participated in the study, present several descriptive statistics of the results, and provide a brief statistical analysis. We conclude this section with a discussion on the accuracy and significance of the results.

3.1. Characterizing the Test Subjects

Initially, 17 volunteers answered our request. From these, 4 reported issues with the software deployment, namely problems related with the startup or shutdown of their laptops and incompatibilities with their network infrastructure, typically due to the use of virtual private networks. We also lost contact with 2 of the initial volunteers, that we assumed had quit the experience. Finally, the results reported by 2 volunteers only contained empty logs, that we believe were due to interference of third party software with our file access monitor.

Therefore, we successfully gathered results from 9 volunteers. 4 of these were computer science professionals (2 academic researchers and 2 consultants), and 5 students (4 undergraduate students from computer science and 1 master student from another scientific area).

3.2. Overall Statistics

Table 1. Number of accessed files per user.

User	1	2	3	4	5	6	7	8	9
No. of Files	3599	1935	436	359	15778	2548	2319	3416	1704

	Minimum	Average	Maximum
No. of Files	359	3566	15778

Table 1 summarizes the number of accessed files during the test period. The average of files accessed in a period of 10 days is surprisingly high: 3566 distinct files, which indicates an average of 357 distinct files each day. Furthermore, additional observations have shown that, in each session, a user accesses, on average, 342 distinct files.

We believe that such high numbers are mostly due to frequent access to Internet sites where currently rendering a single page of content requires accessing many files. These results may be related with the target devices used in this study: laptops. In future work we would like to extend the study to other mobile devices and compare the amount of files accessed by users, which could provide additional relevant insights for the development of mobile computing applications and protocols.

Concerning the number of different locations captured in our study, the number is typically low (as depicted in Table 2). In fact, most users only use two distinct locations: their home and their work place. Note that in these results we have opted to ignore locations to which the users did not attribute a label, as these were probably locations were the users used their laptop very briefly (for instance to check their email). Once again, these results can be entwined with the nature and typical use of laptops. Smaller devices

Table 2. Number of distinct locations per user.

User	1	2	3	4	5	6	7	8	9
No. of Locations	2	2	2	4	3	3	2	2	3

	Minimum	Average	Maximum
No. of Locations	2	2,5556	4

such as cellphones and PDAs would probably be used in additional locations. However, that use could not be related with work, being therefore a poor target for instance, to the use of hoarding mechanisms.

Table 3. Percentage of files accessed at a single location.

	1	2	3	4	5	6	7	8	9
Total	76,2	99,5	81,9	57,2	38,1	87,1	29,1	67,9	88,7
Loc. 1	61,4	98,2	58,5	0,0	0,0	82,0	0,3	48,9	0,2
Loc. 2	14,8	1,3	23,4	0,1	0,1	3,7	28,8	19,0	48,5
Loc. 3				35,1	38,1	1,4			39,9
Loc. 4				22,0					

Minimum	Average	Maximum
29,1074	69,514	99,4832

The most relevant result of our study concerns with the high percentage of files that are only accessed at a single location. In fact, this result is a strong evidence that the behavior of people varies with their location. On average, the percentage of files which is accessed by a user at a single location is 69,51%. In summary, files that are accessed in a given location are not representative of the files that will be accessed in other locations. From this we can conclude that it is very hard to predict the behavior (and the files that will be accessed) by a user in a given location, based on information retrieved in another location (see Table 3).

Table 4. Percentage of file accesses that could not be associated with a location.

	1	2	3	4	5	6	7	8	9
%	28,8	0,1	0,0	8,2	0,0	13,6	4,4	0,0	0,3

Notice that some of the data we retrieved from our test subjects was not labeled with a location. This happened mostly because users did not provide a label to an offline location (*e.g.* a location where no gateway was present). However, the percentage of accessed files that could not be associated with a location is, on average, very low. In Table 4 we present the percentage of files in this condition for each user. With the exception of

user number 1 the percentage of accesses to files to which no location could be attributed is below 13.6%.

3.3. Statistical Analysis

One could argue that the results presented above regarding files that are uniquely accessed in a single location are only a result of sampling bias. The rational for this is that the sample of files accessed by each user at a given location is random in nature, and that further observations of that user at that location could provide other samples in which the set of files accessed between each location would be much more similar.

To address such question, we performed an additional statistical analysis to compare the samples (files accessed at each one of a user's locations)[1]. To this end, we applied the Wilcoxon test to compare pairs of data samples. The null hypothesis of this test is that both samples are similar enough to be considered as being part of the same population (*e.g.* all files in the laptop hard disk). Therefore, the test's resulting *p-value* is the probability that both samples come from the same population.

In our case, the population are the files a user accesses at a given location and the samples are the observations we performed. If the test succeeds (resulting in a high *p-value*), this would mean that the population of files users access at different locations are the same, validating the null hypothesis. Our samples are represented as vectors, one for each one of a user's locations, where each entry represents a file and the value associated with that entry is the number of sessions in which that file was accessed at that location. To compensate for the distinct number of sessions per location, we normalized the data by dividing the number of accesses to each file for a given location by the total number of sessions captured at that same location.

Table 5. Wilcoxon test results.

	1	2	3	4	5	6	7	8	9
Test Result	-20,3	-43,0	-13,9	-4,7	-151,4	-54,154	-37,5	-47,8	-35,95
p-value	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000

Table 5 summarizes the results of this test. As one can see, the resulting *p-values* are all below 0.001. Considering the typical significance levels of 0.05 and 0.01 the Wilcoxon test fails and therefore, the null hypothesis is rejected, meaning that the samples extracted in different locations belong indeed to different populations, *i.e.* the set of files the user accesses in each location is significantly different.

Finally, because both user 4 and 5 used their laptops at more than 2 locations, the Wilcoxon test is not applicable to them. Therefore, we applied the Friedman test for k-variables combination to analyze their data, resulting in the following results:

- User 4 - Test with 3 degrees of freedom
  test statistics = 5024,238, p-value $<$ 0,001
- User 5 - Test with 3 degrees of freedom
  test statistics = 42031,021, p-value $<$ 0,001

[1]These results were calculated using the SPSS application.

In both cases the test rejects the null hypothesis for typical significance levels of 0.05 and 0.01. Again, this means that the probability that the groups of files these users access at different locations is the same is below 0.1%.

This has significant implications for the evaluation of distributed systems namely for mobile computing environments, and in particular for replication and caching mechanisms, such as file hoarding. If user behavior varies significantly between locations, it is important to gather traces from multiple sessions in multiple locations in order to have a realistic evaluation basis. For example, taking our result of the average number of unique files (files accessed at a single location) being around 65%, is a clear indicator that file traces from one location are a bad predictor of behavior in future locations.

4. Conclusion and Future Work

In this paper we have presented preliminary results from a study on the behavior of mobile device users, more specifically, concerning the set of files users access in different locations. This was demonstrated both by descriptive statistics, the percentage of files that are unique to a location and by the tests comparing the traces on different locations. Results support our initial intuition that files accessed by a mobile device user in distinct scenarios are not co-related. This result is very significant, as it goes against the typical models used to design and evaluate mobile computing solutions, in particular hoarding solutions.

Notice that our study was not performed in perfect conditions. The main points against these results are the small number of test subjects that have participated in the study, and the use of the network gateway as an indicator for the location of the user. However, the study was conducted by taking special care not to bias its final result in the direction of our original hypothesis. As future work we would like to extend this study to more users and other mobile devices. We also hope to be able to produce a file access model that could benefit the scientific community by providing an adequate test bed for hoarding solutions. Moreover, we plan to evaluate several existing hoarding solutions using the traces extracted in this study, which we hope will provide additional insight on the accuracy of these solutions.

References

Andersen, B., Jul, E., Moura, F., Guedes, V., and de Gualtar, C. (1994). File system support for semiconnected operation in AMIGOS. In *2nd USENIX Symposium on Mobile and Location-Independent Computing*.

Kuenning, G. H. (1997). *SEER: Predictive File Hoarding for Disconnected Mobile Operation*. PhD thesis, University of California at Los Angeles.

Satyanarayanan, M. (2002). The evolution of coda. *ACM Transactions on Computer Systems*, 20(2):85–124.

Tait, C., Lei, H., Acharya, S., and Chang, H. (1995). Intelligent file hoarding for mobile computers. In *Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 119–125, Berkeley, California, United States. ACM.

Terry, D., Theimer, M., Petersen, K., Demers, A., Spreitzer, M., and Hauser, C. (1995). Managing update conflits in bayou, a weakly connected replicated storage system.

In *Proc.\ 15th ACM Symposium on Operating Systems Principles*, pages 172–183, Cooper Mountain Resort, Colorado (USA).

Acknowledgements

The authors wish to thank Prof. Rodrigo Rodrigues from the Max Plank Institute for Software Systems for his insights and comments and also thank all the volunteers who took part in this study.