Event log based dependability analysis of Windows NT and 2K systems

Cristina Simache, Mohamed Kaâniche, Ayda Saidane

Event Log based Dependability Analysis of Windows NT and 2K Systems

Cristina Simache, Mohamed Kaâniche, and Ayda Saidane
LAAS-CNRS
7 avenue du Colonel Roche
31077 Toulouse Cedex 4 — France
{crina, kaaniche, asaidane}@laas.fr

Abstract

This paper presents a measurement-based dependability study using event logs collected during about 3 years from 133 Windows NT and 2K workstations and servers interconnected through a LAN. We focus on the identification of machine reboots, the classification of their causes, and the evaluation of statistics characterizing the uptimes, downtimes, and the availability of the Windows NT and 2K machines.

1. Introduction

The analysis and assessment of computer systems based on data collected during operation provide valuable information on actual error/failure behavior, and can be used to identify system bottlenecks, to quantify dependability measures and to verify assumptions made in analytical models. In most commercial systems, error and failure data can be obtained from the event logging mechanisms offered by the operating system. Event logs include a large amount of information about the occurrence of various types of events; some of these events are issued from the normal activity of the target systems, whereas others are recorded when errors and failures affect local or distributed resources. The latter events are particularly useful for dependability analysis.

Event-log-based dependability analysis of computer systems has been the focus of several research papers [1-3, 5, 8, 9]. While various types of systems have been studied, including mainframes and largely deployed commercial systems, only a few studies addressed Windows NT or Windows 2K systems [4, 6, 10]. To the best of our knowledge, none of published studies addressed the dependability analysis of both types of systems based on data collected in the same environment. In [4] several analyses are presented based on event logs collected over a six month period from 70 Windows NT mail servers. Similar analyses are presented in [10] based on event logs collected over a four month period from 503 Windows NT servers running in a production environment. An interesting discussion of Windows NT dependability related problems and how Windows 2K has been designed to cope with some of these problems is presented in [6].

In this paper, we present a measurement-based dependability study using event logs collected during an observation period of about 3 years, from 131 Windows NT and 2K workstations and servers interconnected through the LAAS local area network. This network is composed of a large set of Unix, Windows NT and 2K workstations and servers. The event logs recorded on these machines are collected and analyzed at a regular basis. The results presented in [8] concern Unix systems. In this paper, we focus on Windows NT and 2K systems. The identification of useful trends from large event logs is a time consuming task that requires thorough manual analyses. At this step of our study, we have focused on the identification of machine reboots, the analysis of their causes through the classification of the events logged within a time window before the reboots, and the evaluation of statistical measures characterizing the distribution of reboots and the availability of the corresponding machines.

The paper is organized as follows. Section 2 presents the event logging mechanism offered by the Windows operating system and the data collection strategy that we developed based on this mechanism. Section 3 presents the results obtained from the data and Section 4 concludes the paper. In the following, Windows NT and Windows 2K will be referred to as NT and 2K respectively.

2. Data Collection

2.1. Event logging

Event logging is used by computer systems to record the occurrence of significant events: error reports, system alerts, and diagnostic messages. For NT and 2K, it is implemented as a system service that runs in the background and waits for processes running on the local (or a remote) system to send it reports of events [7]. Each event report is stored in a specific event log file. There are three event log files:

- The *security log* contains events generated by the system security and auditing processes.
- The *system log* contains events generated by system components, including drivers and services. It is used primarily to store diagnostic messages that are useful for troubleshooting abnormal conditions, or to find

problems unnoticed by the users. For example, a driver has failed to load, the operation of a device has failed, an I/O error has occurred, etc.
- The *application event log* stores event reports not involving security auditing and system component event reporting. It is commonly used to report internal errors that occur during the execution of an application, such as failing to allocate memory, being unable to access object, aborting the transfer of a file, etc.

The only native facility giving the user access to event logs is *Event Viewer*. The data displayed by *Event Viewer* is formatted according to the following fields:
- *Event type*: denotes the event severity level (error, warning, information, success audit or failure audit).
- *Date and time*: the date and time the report was logged.
- *Source*: the name of the source that reported the event.
- *Category*: source-specific classification of the event.
- *Event*: source-specific event identification (Event ID).
- *User*: name of the user account that generated the event.
- *Computer*: name of the computer that reported the event.

Also, we can display a description of the event, its cause, and where it occurred. However, such a description is not always available.

2.2. Data collection strategy

Data collection is performed once every month and consists of two main steps:
- Identification of all machines of the network to be included in the data collection process;
- Backup of Application and System event logs to a dedicated machine used for data processing.

The identification of NT and 2K machines is based on the analysis of the `hosts.org_dir` master table maintained by the NIS+ server, in which all IP devices connected to the network are declared. From this table, we select all NT and 2K systems, without considering laptops and systems that have Linux as a second operating system. In this manner, we take into account the frequent evolution of the systems connected to the network and their configuration during the data collection period. Such evolution mainly results from system administration and maintenance activities (connection of new machines, upgrade of OS versions, modification of shared services and resources configuration, temporary disconnection of some machines from the network, etc.).

The event logs collected from each machine are concatenated into a single file that is sorted chronologically. Only the new events compared with the last collection are included in the file containing the data associated with the corresponding machine.

The data collected using this strategy corresponds to a three year observation period (January 1999, January 2002). It is noteworthy that data collection from 2K machines started only on September 2000. Figure 1 plots the evolution of the number of NT and 2K systems per month during this period. The number of systems connected to the network progressively increased with some local decreases that are due to the disconnection of some machines from the network. The total number of machines monitored during our study is 133: 76 running NT and 57 running 2K.

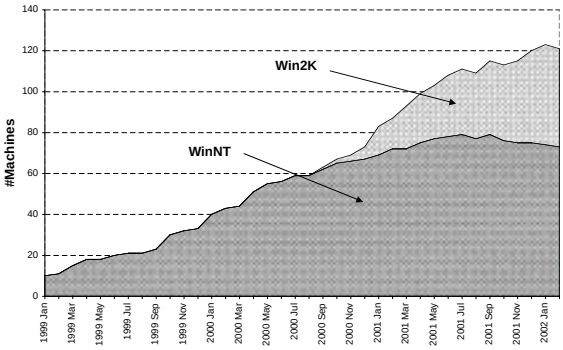

Figure 1. Number of machines per month

2. Data processing

Data processing consists of: 1) extracting from the log files the information that is relevant to the dependability analysis of the target system and 2) evaluating statistical measures to identify significant trends. The log files contain a large amount of information that is not always easy to categorize. In this paper, we focus on the identification of machine reboots and evaluation of statistics characterizing: a) the distribution of reboots (per machine, time), b) classification of reboot causes, c) the distribution of uptime and downtime associated with reboots, and d) availability assessment of machines.

3.1. Identification of reboots

This consists in identifying in the log files the events that are informing upon the shutdown and the restart of the machine, as well as the downtime associated with the corresponding reboot. A reboot may be identified using the events notifying that the *Event Logging service* was stopped and restarted. In the default system configuration, this service is started at the system boot. Once started, it can't be stopped otherwise than by system shutdown.

Typical reboot scenarios with the corresponding events logged by the system are explained in the following.

When a clean shutdown of the system is performed, the event 6006 is recorded informing that the *Event Logging service* is shutting down. The event recorded at the *Event Logging service* startup is 6005. When the system is booted, event 6009 is also logged (just before event 6005) to indicate the operating system version.

Another event to pay close attention to is 6008 that is recorded when a dirty shutdown (also called "blue screen") occurred. The description part of this event contains the "last alive" system time stamp. However, under certain conditions, the system can't record a 6006 or a 6008 event, only 6009 and 6005 are recorded in the log files.

Considering the scenarios above, we developed an algorithm for identifying machine reboots from the

collected log files. It is based on the sequential parsing and matching of each message in the collected log files to one of the events mentioned above. The system shutdown is associated with the events 6006 or 6008, if any, otherwise with the event 6009; the end of reboot is indicated by 6005 event.

This algorithm implemented in Perl, allowed us to detect 11845 reboots from the log files collected from 131 machines of LAAS network for an observation period of about 3 years. However, in our data collection environment, a large number of the NT and 2K machines are used as personal machines. Most of these machines are managed by the system administrators. Although the users are asked to not power-off their machines when they leave the office, some of them do not follow this recommendation. In this context, it is necessary to filter out from the identified reboots those that correspond to such scenarios, before performing any dependability analysis based on the collected data. A careful analysis of the 11845 reboots initially identified by our algorithm led us to detect 2241 reboots (1917 for NT and 324 for 2K machines) corresponding to such scenarios. These reboots were not considered in our study. The results presented in the following are based on the 9604 remaining reboots.

Table 1 shows the break up of these reboots for NT and 2K machines, the length of the data collection period and the number of machines included in the collection. The reboots are grouped into two categories:

- *Clean shutdown Reboots* identified by the sequence of events 6006—6009—6005 or corresponding to the reboots logged for new machines the first time they are connected to the network.
- *Abnormal Reboots* including all other reboots (e.g., those preceded by a "blue screen").

Abnormal reboots represent 38.1% of all reboots. We obtain almost the same percentage for NT machines (38.02%) and 2K machines (38.39%).

	Period (months)	# Machines	# Reboots	# Clean Shutdown Reboots	# Abnormal Reboots
NT	37	76	7213	4470	2743
2K	17	57	2391	1473	918

Table 1. Data collection period, number of machines and reboots for NT and 2K

3.2. Distribution of reboots per machine

The number of reboots observed during the data collection period constitutes a large sample of data on which significant statistical analyses can be performed. However, these reboots are not uniformly distributed among the machines. This is illustrated by the machine reboot rate statistics (number of reboots per hour) given in Table 2. Such variability is explained by differences with respect to the configuration of these machines, the types of software running on them and the user workload.

	Min	Maxi	Average	Median	Std Dev
NT	$2.1\ 10^{-4}$	$2.7\ 10^{-2}$	$6.1\ 10^{-3}$	$4.7\ 10^{-3}$	$4.8\ 10^{-3}$
2K	$1.8\ 10^{-3}$	$9.0\ 10^{-2}$	$1.3\ 10^{-2}$	$8.0\ 10^{-3}$	$1.5\ 10^{-2}$

Table 2. Machine reboot rate statistics (per hour)

The impact of the user's behavior may be observed from another perspective as illustrated in Figure 2 which plots the number of reboots as a function of the hour of day when reboots occurred. This figure shows that the majority of reboots occur during normal working hours. The graph has two peaks: one corresponds to 9AM, the second at 4PM. The first peak includes all reboots that are generally done during the morning by the system administrator to solve the problems that occur during the night. The second peak is likely to be related to software aging problems that are solved by software rejuvenation. It is noteworthy, that there are no planned preventative actions by the system administrators to reboot the machines at regular basis to avoid such problems.

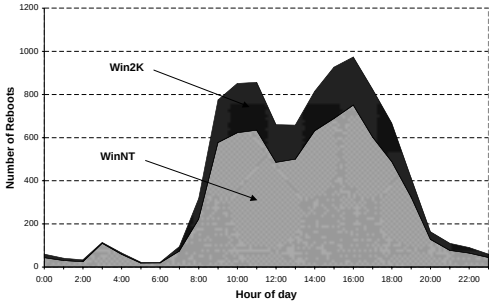

Figure 2. Number of reboots *vs* hour of day

3.3. Reboot cause analysis

The classification of reboot causes is based on the analysis and categorization of the events logged on the system within a given time window before the reboot. A manual analysis of the collected data led us to classify events based on the following three fields: event source, event ID and event type. We call this *a vector*. All events having the same vector values have a similar description. The reboot causes can be inferred from the analysis of this description. We identified 898 different vectors that we classified into 7 classes. Valuable information for the interpretation of event descriptions was obtained from the Microsoft Knowledge Base[1] web site. Unfortunately, as we were not able to retrieve from this site all information needed to analyze all events, some of them remained unclassified. The 7 event classes are defined below.

- *Normal:* gathers events issued from the normal activity of the system and the applications. Most of these events have *Information* as event type.
- *Reboot:* gathers events reported upon the occurrence of a reboot as defined in the reboot identification algorithm.
- *Application Failure (AppF)*: includes events indicating the occurrence of errors during applications execution.

[1] http://support.microsoft.com

- *System Failure (SysF)*: includes events recorded when a system software or hardware component raises an error.
- *Network (Net)*: all events tied to a network service or network card are included in this class.
- *Install/Configuration (I/C)*: two types of events are included: a) those notifying the start/shutdown of an application or a service, the success of an installation, a successful upgrade (all of these have *Information* as event type) and b) those notifying the occurrence of problems during and installation or upgrade operations (most of these events are of types *Warning* or *Error).*
- *Unknown:* includes events that couldn't be classified.

Based on the above classification, we defined an algorithm to categorize the cause of each reboot considering the events recorded within a time window before the reboot and their classes. In particular, when another reboot is recorded during the time window, the cause of the reboot is denoted as *"Rsucc"* corresponding to successive reboots. When several events with different classes are recorded in the window, the cause of the reboot is assigned to the most significant one.

The results of Figure 3 are obtained by analyzing the events logged one hour before each reboot. Similar trends are obtained with a 20 minute window.

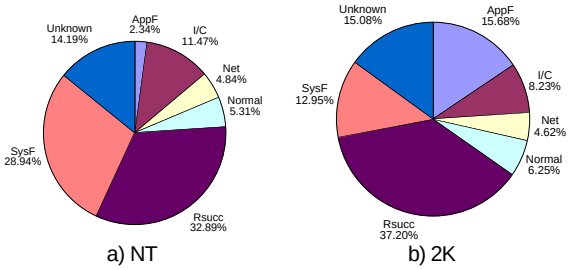

Figure 3. Distribution of reboot causes

As shown in Figure 3, a large number of reboots could not be categorized (~15% for NT and ~14% for 2K). Either the description of the events logged by the system was not available or the description, although available, is not easy to interpret. However, the number of classified reboots is large enough (6105 for NT and 2019 for 2K) and significant conclusions can be derived.

The large percentage of successive reboots (~37% for NT and ~33% for 2K) suggests that very often, more than one reboot was needed to put the system in a stable state after the occurrence of errors or after installation or configuration operations. Note that 55.5% of successive reboots for NT machines, and 50.7% for 2K machines, include at least 3 reboots. A more detailed investigation of successive reboots showed that only one third of these correspond to abnormal reboots (i.e., not preceded by a clean shutdown).

A large proportion of NT and 2K reboot causes can be traced to System failures (12.95% and 28.94%, respectively). Moreover, NT seems to be more vulnerable to application related problems than 2K. This could be the results of the enhancement introduced in 2K compared to NT to prevent applications from corrupting the in-memory operating system or stop responding to requests for service in a timely manner [6].

3.4. Uptime and Downtime evaluation

Similarly to the approach used in [4], machine uptimes and downtimes are estimated as follows:

- For each reboot, the timestamp of the end of reboot and of the event immediately preceding the reboot are recorded (this would be the last event logged by the machine before it goes down);
- Each downtime estimate is obtained by the time difference between the timestamp of the end of reboot and the timestamp of the event preceding the reboot;
- Each uptime estimate corresponds to the time interval between two successive downtimes.

Table 3 presents the uptime and downtime statistics estimated after coalescing all successive reboots observed within an hour into a single reboot event.

	NT		2K	
	Uptime	**Downtime**	**Uptime**	**Downtime**
Min	1 hour	1 sec	1 hour	1 sec
Max	9.3 months	2.3 months	4.5 months	1.3 months
Average	9.6 days	18.7 hours	7.1 days	9.3 hours
Median	4.1 days	4.91 hours	2.91 days	1.4 hours
Std Dev	15.5 days	2.7 days	11.8 days	1.8 days

Table 3. Uptime and downtime statistics

Considering the uptimes, the estimated median values are relatively low (~ 4 days for NT and ~3 days for 2K), however the standard deviation suggests a large variation of machine uptimes. The maximum value is about 9.3 months for NT and 4.5 months for 2K. Considering downtimes, the median value is around 4.9 hours for NT and 1.4 hours for 2K. However, high downtimes values are also observed for some machines that have been temporary disconnected for maintenance.

The results discussed above are not surprising for an academic environment characterized by a frequent evolution of system configurations and applications, and where a large number of machines are used as personal workstations. Nevertheless, further investigation of our data considering only NT servers revealed that the uptime and downtime estimates are in the same range of those reported in [4] and based on event logs from 70 Windows NT mail servers. For the sake of comparison, both results are reported in Table 4.

	NT servers only		Results from [4]	
	Uptime	**Downtime**	**Uptime**	**Downtime**
Min	1 hour	92 sec	1 hour	1 sec
Max	2 months	19.4 hours	2.8 months	15.7 days
Average	12.7 days	1.2 hours	11.8 days	1.9 hours
Median	5.3 days	22.4 min	5.54 days	11.4 min.
Std Dev	16.06 days	2.7 hours	15.6 days	15.8 hours

Table 5. Uptime & downtime statistics for NT servers only and comparison with results from [4]

3.5. Availability evaluation

The availability (A_i) and unavailability ($\overline{A_i}$) measures for each machine are derived from the uptime and downtime estimates using the following formulas:

$A_i = \sum uptime_i / \sum (uptime_i + downtime_i)$ and $\overline{A_i} = 1 - A_i$

Usually, availability is expressed as a percentage value and unavailability is represented as an amount of downtime per year (e.g., in number of days per year).

Table 5 presents the availability statistics for NT and 2K machines. It can be seen, that 2K machines exhibit better availability than NT. Indeed, the average unavailability of a NT machine is about 23.3days/year, while for a 2K machine it is about 16.7days/year. This result is explained by the fact that downtime values measured for Windows NT machines are higher than those observed for Windows 2K (see Table 3). The average and median availability values are relatively modest. This is due to the fact that some machines have very low availability values as indicated by the minimum values in Table 5. When considering only machines offering shared services to the network users, the average availability is 99.60% for NT and 99.8% for 2K, corresponding to an average unavailability of 34.69 hours/year and 17.43 hours/year, respectively.

	Minimum	Maximum	Average	Median	Standard Deviation
NT	65.67%	99.85%	93.62%	95.22%	5.72%
2K	50.03%	100%	95.42%	98.05%	8.72%

Table 5. Availability statistics

4. Conclusion

In this paper, we presented the results of a dependability-related study based on event logs collected during a three year period from 133 Windows NT and 2K machines. We focused on the identification of machine reboots and the characterization of these reboots from different perspectives. Based on the classification of events recorded within one hour before the reboot, the classification of reboot causes was performed. Several statistics characterizing reboot causes, uptime and downtime estimations as well as the availability of the NT and 2K systems are presented. It is important to note that the statistics presented in this paper and the trends observed are intimately related to the environment from which the data was collected, and also to the behavior of the users. It is clear that while event logs provide useful insights into the dependability of the corresponding systems, there is still a need for enhancing the accuracy and completeness of the information included in the logs.

This work is still in progress. Additional investigations are being carried out to analyze error propagation among machines and to assess, from the end user perspective, the availability of the main services offered by the network, taking into account the results obtained for Windows NT and 2K systems as well as on Unix systems.

Acknowledgement: This paper has benefited from fruitful discussions with Karama Kanoun. The authors are indebted to the reviewers for their helpful comments.

This work is partially supported by the European Community (Project IST-1999-11585: DSoS — Dependable Systems of Systems).

References

[1] M. F. Buckley, D. P. Siewiorek, "VAX/VMS Event Monitoring and Analysis", 25th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-25), (Pasadena, CA, USA), pp. 414-423, IEEE Computer Society, 1995.

[2] J. Gray, "A Census of Tandem System Availability Between 1985 and 1990", IEEE Transactions on Reliability, vol. R-39, pp. 409-418, 1990.

[3] R. K. Iyer, D. Tang, "Experimental Analysis of Computer System Dependability", in Fault-Tolerant Computer System Design, D. K. Pradhan, Ed., Prentice Hall PTR, 1996, pp. 282-392.

[4] M. Kalyanakrishnam, Z. Kalbarczyk, R. K. Iyer, "Failure Data Analysis of a LAN of Windows NT Based Computers", 18th IEEE Symp. on Reliable Distributed Systems (SRDS-18), (Lausanne, Switzerland), pp. 178-187, 1999.

[5] I. Lee, R.K. Iyer, D. Tang, "Error/Failure Analysis Using Event Logs from Fault Tolerant Systems", 21st IEEE Int. Symposium on Fault Tolerant Computing (FTCS-21), (Montreal, Canada), pp. 10-17, 1991.

[6] B. Murphy, B. Levidow, "Windows 2000 Dependability", Supplement of Int. Symp. on Dependable Systems and Networks (DSN-2000), New-York, USA, pp. D20-D28, 2000.

[7] J.D. Murray, Windows NT Event Logging, O'Reilly, ISBN 1-56592-514-9, 1998.

[8] C. Simache, M. Kaâniche, "Measurement-based Availability Analysis of Unix Systems in a Distributed Environment", The 12th Int. Symp. on Software Reliability Engineering (ISSRE-2001), (Hong Kong, China), pp. 346-355, IEEE Computer Society, 2001.

[9] A. Thakur, R. K. Iyer, "Analyze-NOW — An Environment for Collection & Analysis of Failures in a Network of Workstations", IEEE Transactions on Reliability, vol. 45, pp. 561-570, 1996.

[10] J. Xu, Z. Kalbarczyk, R. K. Iyer, "Networked Windows NT System Field Failure Data Analysis", Proc. 1999 IEEE Pacific Rim Int. Symp. on Dependable Computing (PRDC-1999), (Los Alamitos, CA), pp. 178-185, 1999.