Piotr LATOSIŃSKI[1], Janusz SOSNOWSKI[1]

Warsaw University of Technology (1)

Monitoring Dependability of a Mail Server

Abstract. *This paper presents the methodology of monitoring a mail server in order to assess its dependability and detect various anomalies. It is based on collecting and analysing various events stored in system logs and continuous monitoring of system resource usage. A special program has been developed and practically verified to deal with these problems in the server handling mails within the Institute.*

Streszczenie. *Artykuł przedstawia metodykę monitorowania serwera pocztowego ukierunkowaną na ocenę jego wiarygodności oraz detekcję różnych anomalii. Bazuje ona na zbieraniu i analizie logów zdarzeniowych oraz monitorowaniu wykorzystania zasobów systemowych. Opracowano specjalny program, który wykorzystano w monitorowaniu serwera pocztowego Instytutu. (**Monitorowanie wiarygodności serwera pocztowego**).*

Keywords: event and performance monitoring, dependability, security, mailing system.
Słowa kluczowe: monitorowanie zdarzeń oraz wydajności, wiarygodność, system poczty elektronicznej.

Introduction

Dependability is a term which combines system features related to reliability, availability, safety, maintainability, etc. Dependability is gaining more and more attention in most computer systems ranging from those used in critical applications (e.g. banking, flight control, e-government) to simple systems used by ordinary people [1,2]. The classical dependability is targeted at handling errors (so as to block their propagation to failures - reactive approach) and scheduling efficient maintenance. In contemporary complex systems we have to take into account also other problems. In particular they relate to on-going system patches and updates, component based system design (including commercial-off-the-shelf components), etc. This leads to the necessity of introducing more general and extended notion of errors/failures i.e. anomaly (abnormal system behaviour) and resilience (the capability of adapting to changes). Hence new proactive approaches in dependability are gaining much interest; they involve some actions before a problem can appear. To resolve these problems we can base on runtime monitoring of the systems.

System behaviour can be observed from different perspectives e.g. user, administrator, application, operating system, etc. Most computer systems provide various logs comprising reports on their operation [2,3]. In general we can distinguish event and performance logs. Usually event logs comprise huge amount of data describing specific events which occur in the run time, the status of system components, operational changes related to start-ups or closings of services, configuration modifications, execution errors, etc. Performance logs give some view on system resource usage (CPU, RAM, discs, network, etc.) and load.

Most research on monitoring related to system availability and reliability at the hardware or operating system levels ([1,4,5] and references therein). Available monitoring tools are targeted at specific problems e.g. SPAM, cyber-attacks [6]. Flexible system analysis from the application perspective seems to be neglected. We have faced this problem in the case of the mail server used in the Institute. Hence we have adapted monitoring techniques to this perspective. To evaluate the quality of mail services as well as to identify normal operational profiles and anomalies we have developed a special tool SyslogAnalyser which uses uniquely defined regular expressions describing event classes related to various system behaviour images in different logs. Event log analysis has been complemented with performance log analysis based on collected data with standard *munin* program.

Section 2 describes basic features of the mail server including the space of possible monitoring. Section 3 and 4 present the developed monitoring schemes related to event and performance analysis (multidimensional approach), they are illustrated with practical results.

Basic features of the mail server

Analysing the operation of the mail server it is reasonable to have a broader view on the whole mailing system. Preparing a message for transmission a user composes an e-mail in his mail client agent (MUA – Mail User Agent), within his computer. Then the message is sent with SMTP protocol (Simple Mail Transfer Protocol) to the MTA agent (Mail Transfer Agent), which is most popular in used mail servers. The message is retransmitted further via several MTAs to the destination mail server, which knows where the target MDA agent (Mail Delivery Agent) is located; this agent supports the user mailbox. All the messages delivered to the user's mailbox can be available to the user, however the user has to retrieve them using MUA agent (in his computer) and POP3 (Post Office Protocol ver. 3) or IMAP (Internet Message Access Protocol) protocols from MDA agent in the email server and finally he can read them. In the mail scheme message routes are fully symmetric and the recipient can be the sender and can send a message in the same way.

In the case of our Institute we have one mail server (bolek). This server is based on a virtual machine configured within the hardware platform: IBM pSeries 550 (9133-55A) - physical memory: 32 GB; physical processors: 2 x 4-core IBM Power5+ 1.6 GHz; Hitachi disc array 4TB. It is virtualized with IBM PowerVM virtualization (Logical Partitions). The server bolek is running in Logical Partition (LPAR): - allocated memory: 2 GB; allocated CPU: 0.2 processing units (1/5 of 1 physical core); virtual processors: 1 with SMT (simultaneous multithreading) enabled (2 logical CPUs); operating system: AIX 5.3 64-bit; allocated 3 logical discs (50 GB each).

The mail server handles all mailboxes of the staff and some students. It performs all actions needed in sending or receiving messages, moreover it performs backups (and in the case of crashes recovery). For this activity it uses CPU, RAM, disc and network resources. This activity can be evaluated indirectly via event and performance logs collected in the server. To assure dependable and resilient operation of the mail server we have to monitor its operation taking into account such issues as detection or prediction of arising problems, evaluation of system resource usage and trends, characteristics of operational profiles, etc. Hence an important issue is collecting and analysing event and performance logs, which is discussed in the sequel.

In Unix systems we have many event logs generated by syslog or other programs. We concentrate on the following logs: auth.log (gives the information on correct or erroneous

user loggings, connection crashes, etc.), daemon.log (gives some information related to cooperation with DNS server), mail.log (comprises useful information on mail system operation, in particular messages generated by the sendmail daemon), and last.log (it is created from the utmp binary files with last command and provides information on login and logouts of the users). The first three logs are provided by syslog.

The authentication log contains information from such applications like Secure-Shell daemon (sshd), su program or some mail daemons. Reading this file we can get information on time, when users log in or when they try to do it, but there is no information about how long they are logged in the system nor when they log out. There are also a few other entries, such like a message on disconnection from a client or other minor internal errors.

The daemon log is used mostly by dns server, which provides information on various minor errors related to updating dns zone (when configuration does not allow changing local zones, while hosts try to do it) or to resolving some non-local zone, which are configured incorrectly. There are also a few entries from others daemons (such like xntpd), and they inform on the current situation (e.g. Synchronization lost or The daemon is started).

The mail log stores information on the whole mail traffic, which is generated by mail server applications, e.g. sendmail, pop3d and imapd. In particular it includes information from which server and to which server the message is sent, and the information on logged-in and logged-out users. It does not contain information on the message transmission path, this information is comprised within message headers. The complete information on the mail transmission path (including delays) can be extracted from the messages stored in the mailbox. However here we touch the problem of legal data protection (can be resolved by anonymity procedures).

The last.log is not the syslog file. It relates to standard *nix commands, which report time of users log-ins and log-outs. To create last.log file from the outputs of these commands we need to pipe their outputs to this file (e.g.: `last > last.log`). This log additionally contains user and terminal names (optional host name) as well as the information on system reboots and halts (shutdowns).

We have developed a special application SyslogAnalyser which derives synthetic characteristics of the mail operation. Moreover derived statistics can characterize activities of different users, domains of users, incoming and out coming mailing traffic in the system, etc. To get better view on the server operation we collect basic performance parameters (related to CPU, RAM, disc, network resources, etc.) using standard munin program composed of munin-node agent (located in the monitored system) and munin-master (communicating with munin-node via network). The munin-node uses available local tools providing performance information. The munin-master uses RRDtool (to store data and generate plots) and a www server.

Event monitoring and analysis

During normal operation of workstations or servers a large amount of events is registered in the logs. The formats of registered events have some loosely defined general scope; in particular we can distinguish various data fields comprising specific information in textual or numerical form with some specific brackets, etc. Some fields can be considered as parameters. Typically we have time stamp (the time of event registering), name of the event source (e.g. host name, application program, process PID, etc.), a text describing the related event problem, severity of the problem, etc. However, events of different classes can be stored in different log files (e.g. security events specifying authorisation problems, user login and logout events). The included texts can be very general of low information value or more specific. Nevertheless their meaning can be better interpreted after gathering some practical experience within a longer observation time period. Basing on this experience we have developed SyslogAnalyser. Having analysed the structure and information contents of all collected event logs (within half a year) we have defined classes of semantically similar events for various logs. This is a complex log abstracting problem [3], which we have resolved by exhaustive specification of event classes using regular expressions. These expressions are used during generation of the statistics. Some of them define specific errors or anomalies. Their practical usefulness was proved by detecting some critical situations including maintenance flaws, characterizing mail traffic, etc.

In the analysed system we deal with auth, daemon, mail and last logs. The first three logs have to some extent similar structure which can be described by the following regular expression (consistent with perl notation):

(\w{ 3 } (?:\d|)\d \d\d:\d\d:\d\d) (\w+) ([a-zA-Z\|]+:
[a-zA-Z\|]+) (? : ([^ \ [:] +) (? : \ [(\d+) \]) ?:) ? (. *)

Text characters are interpreted directly; beyond them we can use meta-characters and character classes. Basic meta-characters are as follows: ^ - specification of the word beginning, $ - denotes characters at the end of a word, . – denotes any character, […] - denotes any character from the list within the brackets, we can also include here ranges of characters e.g. [a-zA-Z] - denotes any letter of lower or upper case, [^…] - denotes any character excluding those in the bracket following ^, (..) - denotes a group of characters, | - denotes alternative, * - denotes 0 or more repetitions of the succeeding character or a group, similarly the following meta-symbols ? and +, relate to 0 or a single repetition, and one or more repetitions, respectively; {..} - denotes a specified number (in the bracket) of repetitions of the preceding character or a group. Typical character classes are preceded with \, so \d denotes any decimal digit, \w any character starting a word, \S denotes white space character (blank, space tab of new line character). For illustration we give a complete event generated by syslog:

Sep 27 21:51:40 bolek mail:debug pop3d[2187482]: pop3 service init from 194.29.168.97

It can be correlated with the general expression format as follows:

(\w3 (?:\d|)\d \d\d:\d\d:\d\d) - Sep 27 21:51:40 – *time stamp*
(\w+) - bolek – *host name*,
(([a-zA-Z\|]+:[a-zA-Z\|]+) - mail:debug – *event source and priority*
[^\[:]+)(?:\[(\d+)\])?: - pop3d[2187482]: - *name of the program and PID*
(.*) - pop3 service init from 194.29.168.97 – *message text*

The structure of events generated by the last program is a little bit different and can be described by the following expression:

(\w+) +(\~|\S*?) +?(\S+)? +(\w{ 3 } \d\d \d\d:\d\d) (? : still logged in.|(\d\d:\d\d|.*?) +\(((? :\d+\+)?\d\d:\d\d) \)) ?

Having analysed event logs we have identified various event classes which characterise the server operation in particular the mail program. For the considered mail log we have identified 52 event classes generated by sendmail (type distribution: 24 - info, 18 - notice, 2-debug, 2-alert, 3 - warning, 3 - error). The programs responsible for access protocols (pop3, pop3d, pop3ds, imapd, impads) generated 27 event classes to the mail log and 2 classes to auth.log. Syslog and other programs generated 61 different event classes stored in auth and security logs. Regular

expressions describing these classes are included into the SyslogAnalyser and facilitate providing appropriate statistics by correlating specified class with appropriate data structures. SyslogAnalyser comprises various data structures for counting specific situations. The basic structure includes general information such as: information on senders, receivers, linked messages (previous and next), reasons of rejecting messages by sender, receiver, additional information (e.g. information obtained from other server on non-existing user), etc.

There is a group of data structures with counting capabilities. For example they handle counting the number of bytes in emails, number of logins to MDA, number of senders/receivers local within the University, Institute, department and external, number of student/ staff users, the number of deferred mails in the queue (lack of place in the addressed mailbox), the number of rejected mails (e.g. due to lacking addressee on the target sever, empty messages, the number of connections with pop3d, pop3ds, imapd, imapds protocols, the number of logins and logouts of each user, etc. Moreover for alerts we have a separate data structure which stores the message text and the number of its occurrences e.g. "….domain of sender address ilom-alert@194.29.168.100 does not exist", "Fatal mailbox error user=…unexpected changes to mailbox, try restating", "..relaying denied". For daemon, last and auth logs the number of data structures is much more limited due to the specificity of comprised there information. In particular auth log is the basis to create data structures comprising the lists of correct and incorrect login tries, last.log provides timestamps of logins and logouts (including the user terminals) as well as a list of detected restarts.

Tracing logins we have identified some attack tries (for a short time the registered number of logins tries from a specified address exceeded 51 000 - with failed password) while for other legitimate users (total number of users 222) for two months failed passwords did not exceed 500. Here it is worth noting that the monitoring agent of munin master logs to the server every 5 minutes (about 2888 connections within 24 hours). Sporadically we observed invalid user login tries. Within daemon logs some not critical events of alert class have been detected. We have identified a few losses of logs (overwriting) due to a lack of some elasticity in the archiving scheme (periodic with reserved buffers based on average log file sizes).

A relatively large number of users (mostly new users) base on non-ciphered transmissions, so it is reasonable to make them conscious of such possibility. Observing mail traffic we have identified some periods of low activity e.g. summer vacations and some sporadic significant increases, for example at the end of a semester (students communicate frequently with the staff), and at end of October – significant number of emails generated by external users caused by a fault in developed application related to a large project. Analysing traffic profiles we can get also information of potential spam attacks, etc. From the last log we have identified 85 actions of system closings and restarts. In 33 cases this was related to the server administrator planned activity (33 shutdowns and 33 reboots). The remaining 19 cases related to system crashes for which only reboot events were registered.

We have also analysed characteristics related directly to the mail traffic such as average mail size (fig.1) calculated for each day separately, distribution of mails sent to the University domain or outside, distribution of mails attributed to the staff or students, distribution of used protocols pop3, pop3 with ssl, imap and imap with ssl, etc. The most popular protocol is imap. It has been observed that most emails have been sent without protection. Average daily traffic of the mails to the University domain (pw) was over 0.15 (maximal day average 0.5) mails per s, the outside mails ranged up to 0.05. These numbers are averaged over 24 day hours, the mail rates for active (working) hours are 4-5 times higher.

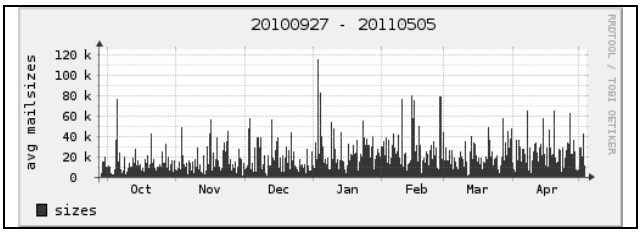

Fig.1. Average mail size in bytes

Resuming the collected statistics characterize the operational profile of the server and facilitate to identify (and diagnose) various problems, moreover they can allow predicting some of them also. The most important features (statistics) of logs are presented in synthetic reports.

Resource monitoring

Performance related information can be collected using various counters [2,6]. They are correlated with performance objects such as processor, physical memory, cache, physical or logical discs, network interfaces, server of service programs (e.g. web services), I/O devices, etc. For each object many counters (variables) are defined characterising its operational state, usage, activities, abnormal behaviour, performance properties, etc. These counters provide data useful for evaluating system dependability, predicting threats to undertake appropriate corrective actions, etc. Here arises the problem of selecting appropriate variables for monitoring so as to get representative image of the system operation at low time overhead. Beyond standard counters we can define application oriented counters. Selected data on performance is collected by munin program (http://munin-monitoring.org/Wiki/Native_ssh). In the sequel we give some illustrative results.

The generated plots of performance measures can be presented in different time perspectives (daily, weekly, monthly). We have found that CPU usage is relatively low, typically up to 15% averaged per day (it is attributed in 45% to system, 45% to user and 10% to I/O waiting processes). Average load (i.e. average process queue length in the scheduler) is typically 0.2-1.4 (in Christmas time 0-1.2) with average values in the range 0.55-0.85 (this relates to two logical CPUs of the virtual machine). At the end of Oct. a spike of average load 2.1 was observed related to erroneous burst of emails (compare mail queue in fig.3a). Fig. 2a presents RAM usage (y axis range 0-2.4 GB) for swap (the highest dark shadowed layer on the plot), cache, inuse and pinned (lowest white layer) memory areas. Three small negative pulses relate to system reboots.

In fig.2 we can observe a few days break at the end of Sept.; this related to halting the collection of monitored data (caused by a connection problem). Some other short breaks have been observed and diagnosed as the main switch hang-up (recovered with reset) or partial (local) disconnections in the network. They were not reported directly in the server event logs (could be detected in user computers). Fig. 2b relates to network traffic and presents the characteristics of connections per second (averaged per day): requests (out coming from the server), accepts (coming to the server), established, and closed connections, respectively in the order from the lowest to the

highest plot. The y axis is logarithmic from 0.005 to 4. The sum of incoming and out coming connections is higher than established connection (because some connections have been rejected due to TCP protocol). The visible fluctuations to lower values relate to low activity on weekends. The high pulse at the end of October relates to some anomaly described further on. Fig. 3a and 3b give a long term (10 months) and monthly (4 weeks) plots of sendmail queue, respectively. Typically the queue is in the range 1 to 4 with some spikes up to 11 (fig. 3b). However two anomalies have been detected: queue length up to 300 on June and up to 3000 at the end of October (fig. 3a). The first one has been caused by some disturbance in the sendmail processes. This was not noticed by the administrator, our observation was a trigger to manually cancel the queue. The second one resulted from some inconsistency of one user (using Windows for mails) with the server, in consequence a mail sent to a long list of recipients (involved in a big project) generated multiple responses from the absent user workstation resulting in some infinite loop of emails. The mail server was not adapted to this situation. The detected anomaly initiated some modification in the server. The range of y axis of fig. 3a is 0-3500, so the normal queue values are close to the x axis.

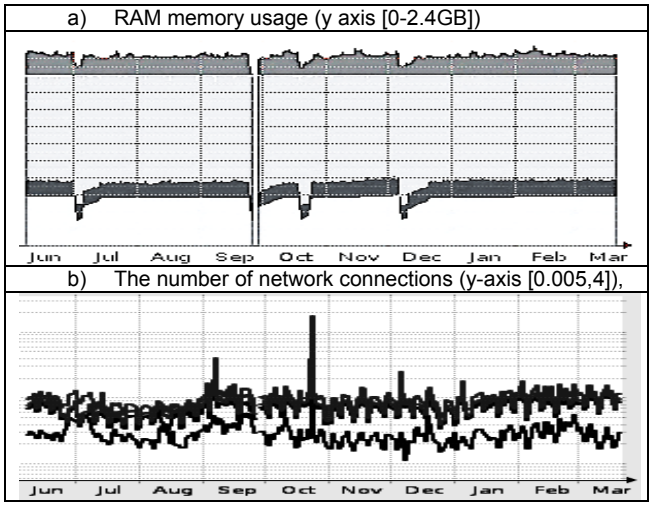

Fig. 2. Profiles of RAM and network usage

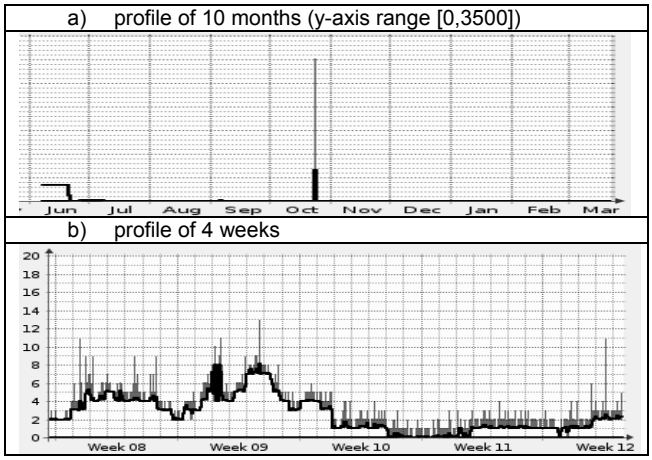

Fig. 3. Profiles of the mail queue in sendmail

Having monitored disc usage we observed some stable traffic on the basic disc with average (within 5 min samples) writes and reads 10-100 kB and 100-1000 kB per second, respectively. Two other discs are used for backups alternatively each weekend, with average writes about 340-350kB/s. Read operations from these discs occurred sporadically once per several months and related to recovery of some files (with averaged reads about 2-7 kB/s).

Conclusion

The developed SyslogAnalyser generates various statistics, filters interesting or neutral event records basing on the set of a reach and carefully specified event classes (over 130). They have been defined by regular expressions for various types of logs. Enhancing this analysis with performance monitoring we have got the capability of evaluating operational profiles, identifying normal operation conditions as well as detecting various errors and anomalies. Using regular expressions is also helpful in system diagnosis. For the identified anomalies we traced their sources and criticality (log correlations). They are reported (problem signatures) in the created data base.

As compared to other tools we assure multidimensional approach (correlation of various logs) and higher flexibility in identifying problems (including unknown), in particular fine-grained analysis can be adapted to new appearing problems. Such capability is not possible in commercial systems (e.g. Tivoli, Sawmill). Open source monitors are targeted at specific problems or logs e.g. SNORT (intrusion detection at the level of IP packets), Hyperic - checking specified performance thresholds, Sendmail analyser - is limited to mail log and delivers statistics from a specified list.

The analysis of the collected data allowed us to identify basic features of normal system operation and suggest potential signatures of anomalous behaviour (e.g. traffic bottlenecks, subsystem crashes, too long mail queues, backup and maintenance flows). It is worth noting that the developed approach allowed us to identify several anomalies which were not noticed directly by the system administrator or users. This confirms the usefulness of this approach. Currently we extend this approach to monitoring switches and the network infrastructure in the Institute.

Wydanie publikacji zrealizowano przy udziale środków finansowych otrzymanych z budżetu Województwa Zachodniopomorskiego.

REFERENCES

[1] Cinque M. et al., A logging approach for effective dependability evaluation of computer systems, *Proc. of 2nd IEEE Int. Conf. on Dependability*, 2009, 105-110.

[2] Król M., Sosnowski J., Multidimensional monitoring of computer systems, *Proc. of IEEE Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, 2009, 68-74

[3] Naggapan M., Vouk M. A., Abstracting Log Lines to Log Event Types for Mining Software System Logs, Proceedings of Mining Software Repositories, 2010, 114-117

[4] Yu Li, Zheng Z, Lan Z., Practical Online Failure Prediction for Blue Gene/P: Period-based vs. Event-driven, *Proceedings of the IEEE/IFIP International Conference DSN*, 2011, 259-264

[5] Salfiner F., Lenk M., Malek M., A survey of failure prediction methods, *ACM Computing Surveys*, vol. 42, no. 3, March 2010

[6] Ye N., Secure Computer and Network Systems, John Wiley& Sons, Ltd. ISBN 978-0-470-02324-2, Chichester, 2008

***Authors**: inż. Piotr Latosiński; prof. dr hab. inż. Janusz Sosnowski, Politechnika Warszawska, Instytut Informatyki, ul. Nowowiejska 15/19, Warszawa 00-665,*
E-mail: P.Latosinski@stud.elka.pw.edu.pl;
J.Sosnowski@ii.pw.edu.pl