Software Rejuvenation - Do IT & Telco Industries Use It?

Javier Alonso[1], Antonio Bovenzi[2], Jinghui Li[3], Yakun Wang [3], Stefano Russo[2], Kishor Trivedi[1]
[1]Electrical Computer Engineering,
Duke University, Durham, NC, USA
javier.alonso@duke.edu, kst@ee.duke.edu
[2]Dipartimento di Informatica e Sistemistica (DIS)
Università di Napoli, Federico II, Napoli, Italia
{antonio.bovenzi, russo}@unina.it
[3]Huawei Technologies Co., Ltd
{jinghui.li, Wangyakun}@huawei.com

Abstract— ***Software rejuvenation has been addressed in hundreds of papers since it was proposed in 1995 by Huang et al. The growing number of research papers shows the great importance of this topic. However, no paper has studied yet software rejuvenation in the real world. This paper investigates to what extent software rejuvenation techniques are integrated in the IT and Telco solutions. For this purpose, it has been conducted an intensive search of different sources such as company's product websites, technical papers, white papers, US patents, and consultant surveys. The results show that IT and Telco companies develop software rejuvenation solutions to deal with software aging. The number of US patents addressing this issue confirms the interest of industry to develop mechanisms to deal with software aging-related failures. It has been observed that real software rejuvenation solutions mainly use time-based or threshold-based policies, while the US patents are focused on predictive approaches.***

Keywords; Software rejuvenation; Software aging; IT; Telco;

I. INTRODUCTION

Since Huang *et. al* [1] introduced software rejuvenation concept to deal with software aging phenomena in 1995 [2], hundreds of research papers have been published, especially since the emergence of the International Workshop on Software Aging and Rejuvenation (WoSAR) in 2008 [3].

The growing number of research papers shows the importance of this topic. Some of them describe software rejuvenation techniques adopted in real systems such as IBM Director [4] and OLTP DBMS [5]. However, to the best of our knowledge no study has yet analyzed to what extent software rejuvenation techniques are really applied in industry.

This study investigates the IT and telecommunications industry solutions that integrate software rejuvenation, and studies the proposed rejuvenation techniques, if any. In particular, the study focuses on identifying: 1) the *rejuvenation scheduling*, i.e., the method adopted to trigger the rejuvenation such as time-based [6], threshold-based [7] or prediction-based [8]; and 2) the *rejuvenation granularity target*, i.e., the software directly influenced by the rejuvenation, such as the application, the operating system, the virtual machine or the virtual machine monitor [9].

For this purpose, an intensive search has been conducted through available documentation such as companies' technical papers, whitepapers, US patents, products' websites, and consultant surveys. We have concentrated our research on solutions proposed in the last decade to give the most accurate snapshot of the state the art of software rejuvenation in the IT and telecommunications industry.

The results of the study show that companies develop proactive fault management mechanisms like software rejuvenation in order to deal with different types of software anomalies, especially software aging. Apart from the solutions already implemented, the number of US patents addressing this issue confirms this. A detailed examination of the solutions reveals that, software rejuvenation techniques adopted by the industry mainly use time-based or threshold-based scheduling strategies. Instead, the solutions described in US patents are focused on prediction-based approaches.

The rest of the paper is structured as follows: Section II describes the search criteria used to select and mine relevant sources of information; Section III presents the taxonomy used to classify software rejuvenation techniques; Section IV describes the industry solutions found, and summarizes the classification results. Section V concludes the paper by discussing the results of our study and the future work.

II. SOFTWARE REJUVENATION SEARCH PROCESS

An actual and complete snapshot of software rejuvenation solutions adopted by the industry is not easy to obtain due to several reasons. First, it is necessary to identify the relevant sources of information, such as companies' websites, patents archives, or white papers. Table I summarizes the most important sources considered in this study.

Second, the considered sources of information belong to different business areas (e.g., IT or telecommunications) and describe heterogeneous products such as operating systems, databases, web servers, and networking technologies. It emerged soon that no single terminology is used. It was found that terms that are very common in the scientific literature – such as software rejuvenation and aging-related keywords - were likely to leave out of search results most of the relevant documents. Hence it was necessary to widen the search criteria. Table II lists the main keywords used.

Finally, the evaluation of the effectiveness of the solutions represents a further issue. Empirical, simulative, or analytical results are often unavailable. In most cases, the information about the effectiveness of the solutions is biased and only available in the marketing documentation of the companies. However, this information cannot be easily assessed since there was no access to all the solutions found. A potentially good practice could be to analyze third party agency reports (e.g. IDG [10], Gartner [11], InfoTech [12], or ReadSoft [13]). However, the available reports are not exhaustive. The approach followed is based on a qualitative analysis of the solutions to determine their potential effectiveness to deal with software aging.

TABLE I. SOURCES OF DOCUMENTATION

Source	Description
www.google.com/patents	Engine to search for US patents
www.freepatentsonline.com	Engine to search for patent research
Company's websites	Web pages illustrating offered solutions, or containing white papers and documentations
Technology websites	News, articles and blogs on technology related to software rejuvenation
Third party consulting company' websites	Websites of consulting companies obtain possible surveys about proactive software recovery mechanisms

For all these reasons, a systematic procedure to identify and then classify software rejuvenation solutions was followed. The procedure is iterative: during the document review process, the keyword list is refined and new documents, and solutions emerge. The steps of the search procedure are the following:

1. Create a list of rejuvenation-related keywords (see Table II). The list helps *i)* selecting the most relevant documents and *ii)* highlighting the part of the document that may describe or may be related to software rejuvenation. Apart from the basic terms such as rejuvenation, software aging and memory leak, we use terms that embrace broader activities such as fault tolerance, proactive maintenance proactive fault treatment, as well as terms related to specific maintenance activities such as automatic restart/reboot, daily reboot, and process(es) recycler.
2. Create a keyword list containing the effects or the failures that an aging related bug can cause such as crash, hang, out of memory, slow response, performance degradation, thrashing, route flapping.
3. Query for the identified keywords in the selected document sources.
4. Examine in detail each solution and its impact on the considered system. If the solution prevents, or is a countermeasure, for at least one of the failures identified in Step 2, consider the solution for the classification process. Otherwise describe the technology in detail, if it can be useful for architecting rejuvenation strategies.

It is noteworthy that even if a solution under review includes the description of technologies that may be used for architecting rejuvenation strategies (e.g., monitoring infrastructures, trend and anomaly detection algorithms) these technologies are not included in the classification process. On the other hand, solutions designed explicitly to improve the rejuvenation have been included.

TABLE II. MAIN SEARCH KEYWORDS USED

Rejuvenation-related keywords	
• Software rejuvenation • Software aging • Preventive maintenance • Proactive maintenance • Recovery • Reboot • Process recreation	• Restart • Daily Reboot • Automatic restart/reboot • Proactive reboot • Proactive Recovery • Processes recycler • Process killing
Aging effects-related keywords	
• Memory leak • Hang up • Software bug • System crash • Performance degradation • Anomalies Forecasting/Prediction	• Failure Prediction • Slow response time • Thrashing • Resource exhaustion • Preventing Leak • Memory Management

III. SOFTWARE REJUVENATION TAXONOMY

As demonstrated by previous studies, software aging affects many kinds of long-running software systems ranging from safety- to business- critical domains such as networking systems [14], operating systems [15], databases [16], web servers [16], middleware [17], spacecraft's flight software [18] and virtualized environments [19]. Each system and environment requires different features to deal with software aging and other software faults. Hence, it is important to consider these differences to properly classify the solutions.

The classification of software rejuvenation solutions encompasses different aspects, which are described below.

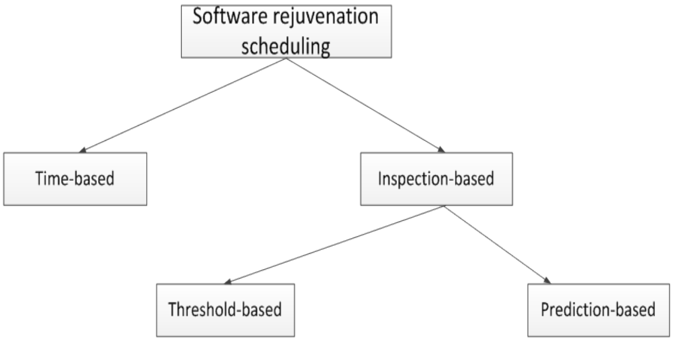

Figure 1. Software rejuvenation scheduling strategies

Rejuvenation scheduling: The first classification is based on the method used to trigger the rejuvenation. Software rejuvenation scheduling methods can be divided into two broad categories, (see Fig. 1): Time-based and Inspection-based. The former includes rejuvenation scheduling triggered at predetermined time intervals [6]. The software rejuvenation approaches in which the triggering depends on the data collected at runtime belong to the second category. The inspection-based approaches can be subsequently divided in two categories: threshold-based approaches and prediction-based approaches. Threshold-based approaches are based on monitoring the aging effects and triggering the software rejuvenation when a specific threshold is exceeded [7]. In prediction-based approaches, a prediction method is applied to estimate the time to the exhaustion of resources or the time to failure caused by the software aging [8].

Rejuvenation target: The second type classification is based on the granularity of target software directly influenced by the rejuvenation. We consider the following targets granularities adapted from [9]: Application component (CMP), Application (APP), Operating System (OS), Virtualization (VRT), i.e., Virtual Machine (VM) and Virtual Machine Monitor (VMM), and Physical node (PHY). We will classify each software rejuvenation strategy according to these six granularity levels. This will allow us to know which software target is considered critical in terms of availability and prone to suffer software aging. Fig. 2 presents the six rejuvenation granularity targets as well as corresponding references.

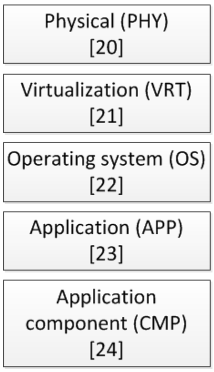
Physical (PHY) [20]
Virtualization (VRT) [21]
Operating system (OS) [22]
Application (APP) [23]
Application component (CMP) [24]

Figure 2. Software rejuvenation granularities

IV. Industry and US Patents classification

For those Industry solutions and US patents that matched the search criteria, relevant information is described. After that, the results of the classification, based on taxonomies described earlier, are presented.

A. *List of solutions*

The solutions found have been divided between Industry solutions and US patents, since the industry solutions are based on real software already in use. In the case of patents, it was not possible to determine if the design described was already implemented or it is just an idea for future systems.

1) *Industry solutions*

a) Alcatel-lucent - Switches OmniSwitch: Alcatel-lucent has developed an automatic and proactive mechanism in its switch family *OmniSwitch*. This mechanism allows an automatic reboot of the switch at configurable time epochs. The command is called *reload working* [25].

b) Avaya - Servers and Media Gateways: Avaya has implemented in its Servers and Media gateways a proactive recovery mechanism based on different escalated levels of rejuvenation [26]. This solution implements two types of rejuvenation scheduling: Inspection-based for the top granularity target, and time-based for the rest of the levels.

c) Apache web server: The solution implemented by Apache has been extensively described in several research papers [16],[27]. Apache kills and recreates each child process upon reaching a certain condition, based on the maximum number of requests served by the child process [28].

d) IBM - Tivoli IBM xSeries Director: The IBM xSeries director presents a software rejuvenation solution that monitors the resources of the system, estimates the time to resource exhaustion, and applies the rejuvenation at different levels (i.e., application, process group, or operating system) [4]. This solution was the result of a joint project between Trivedi's Duke research group and IBM.

e) Microsoft – IIS web server: Internet Information Server, the webserver of Microsoft, implements a worker recycling similar to the Apache solution. The main difference is that IIS recycling can be configured for: Time-based, Inspection-based (number of requests, total virtual memory usage threshold, used on memory), and on demand scheduling by the administrator or the web application [29].

f) Oracle – DBMS: Oracle Database servers implement a software rejuvenation mechanism in the ORACLE database resident connection pool, implemented in the DBMS_CONNECTION_POOL package. The approach is similar to Apache and IIS recycling. The database resident connections are restarted based on two scheduling policies: Time-based (Time to live for a pooled session) and inspection-based (number of times a connection has been taken and released to the pool) [30].

g) JBoss - DBCP: JBoss web application server implements a software rejuvenation mechanism to prevent Data Base Connection Pool (DBCP) leaks. The DBCP is responsible to create and manage a pool of Database connections between JBoss and the Database. The connections of the pool are recycled and reused because it is more efficient than opening a new connection. However, the connection has to be explicitely closed by the applications to allow JBoss to reuse it. If the application fails to do this, the connection pool can be exhausted. To avoid that situation, JBoss implements a solution to mitigate the consequences of DB connection leaks. Proactively JBoss tracks and recovers the abandoned connections. The recovery process is triggered when the number of available connections becomes small. A connection is deemed abandoned based on the time period that it has been idle. Both parameters can be configured [31].

2) *US patents*

Due to a lack of space it is not possible to describe the details of all the US patents found during our search. Furthermore, the US patents' descriptions usually are ambiguous or too general about the solution. Technical details are often not disclosed. For this reason, a brief description of the patents is presented, in order to contextualize the solution proposed. Amazon -US patent 7610214 [32]- presents two types of forecasting methods to deal with seasonal data anomalies. AT&T US patent 7881189 [33] describes a threshold-based rejuvenation mechanism to deal with long post-dial delays in VoIP infrastructures. Cisco US patent 6993686 [34] presents a mechanism to detect misbehavior in network devices, and recover from them. IBM US patent 6978398 [35] is focused on a predictive fail-over mechanism to deal with performance degradation due to hardware and software failures. IBM US patent 2008/0163004 A1 [36] is focused on application level. The authors present a fail-over application mechanism on a single server. A secondary application is created to substitute the failed application. IBM US patent 6993458 [37] is focused on prediction mechanism for different purposes. The authors present a mechanism to preprocess data before forecasting. The preprocessing categorizes the data into different types. Each type is then analyzed and used by the forecasting method. A Bayesian network model to predict critical events based on log information is proposed in IBM US patent 7895323 [38]. This solution is focused on clustered environments. Intel US patent 7702966B2 [39] presents a mechanism to predict application errors, based on past error notifications. The applications are required to notify their errors. Microsoft US patent 8117505 [40] describes a method to contextualize the resource status with different information about the usage of the resource. Motorola US patent 7227845 B2 [41] presents a mechanism to restore the communication link of a base station. Oracle/Sun US patent 7543192 [42] presents a mechanism to predict failures based on a learning process which tries to correlate failures with monitoring data. Siemens US patent 2006/0156299A1 [43], Siemens US patent 2007/0250739A1 [44], Siemens US patent 7475292B2 [45], Siemens US patent 8055952 [46], and Siemens US patent 2011/0072315A1 [47] are closely related to each other. They propose, in general, a rejuvenation mechanism for clustered computer systems. The main idea across the patents is to use user-related metrics like response time to trigger the rejuvenation.

3) *Classification results*

Table III presents the results of the Industry and US patents classification following the taxonomy described in Section III. Two classifications have been conducted by two of the authors separately. Then, the classifications were compared and in the cases of disagreement, a detailed analysis and discussion was conducted to reach a consensus. Note that several solutions classify into both time-based and inspection-based scheduling approaches. This indicates that the solutions propose a configurable rejuvenation approach or a combination of two different types of scheduling.

As for the rejuvenation granularity, the classification process has been harder because the solutions do not explain in detail how the rejuvenation is conducted. In fact, in several cases the description only indicated *node restart* or *system reboot*. This information is not enough to determine physical or OS rejuvenation target since both rejuvenation targets apply to this simple description. In these cases, we have classified the rejuvenation target of the solution as “unknown” (UNK). The ambiguity about the rejuvenation granularity target is especially evident in the US patent descriptions.

One of the questions the study is interested in is to identify if there is any predominant type of rejuvenation scheduling mechanism used by the industry or proposed in the patents. Fig. 3 summarizes the results. It is observed that threshold-based solutions are, in general, predominant. It is noticed that real solutions (Industry in Fig. 3) are mainly focused on threshold-based and time-based. Just one real solution exploiting prediction-based rejuvenation, was found. It was implemented in IBM director in xSeries. However, the administrators are able to disable this feature and, then, only time-based rejuvenation is applied. By contrast, it is observed that many patents propose prediction-based mechanisms. This indicates that the industry has interest in predictive approaches; however it is still not psychologically ready to adopt predictive software rejuvenation solutions, especially in a closed-loop manner.

TABLE III. CLASSIFICATION OF INDUSTRY AND US PATENTS

Solution/US Patent	Rejuvenation Scheduling	Rejuvenation Target
Alcatel-lucent	Time-based approach	OS
Avaya Servers & Media Gateways	Time-based & Inspection-based (Thresholds)	PHY, OS, APP, and CMP
Apache webserver	Inspection-based (Thresholds)	CMP
IBM-Tivoli IBM director xSeries	Time-based & Inspection-based (Prediction-based)	OS, and APP
Microsoft IIS	Time-based & Inspection-based (Thresholds)	CMP
ORACLE DBMS	Time-based & Inspection-based (Thresholds)	CMP
JBoss – DBCP	Inspection-based (Thresholds)	CMP
Amazon US7610214	Inspection-based (Prediction)	UNK
AT&T US7881189	Inspection-based (Thresholds)	UNK
CISCO US6993686	Inspection-based (Thresholds)	UNK
IBM US6978398	Inspection-based (Prediction)	UNK
IBM US2008/0163004 A1	Inspection-based (Threshold & Prediction)	APP
IBM US6993458	Inspection-based (Prediction)	UNK
IBM US7895323	Inspection-based (Prediction)	UNK
Intel US 7702966B2	Inspection-based (Prediction)	APP
Microsoft US8117505	Inspection-based (Prediction)	UNK
Motorola US7227845B2	Inspection-based (Thresholds)	UNK
Oracle/Sun	Inspection-based (Thresholds)	UNK
Siemens US 2006/0156299A1	Inspection-based (Thresholds)	UNK
Siemens US2007/0250739A1	Inspection-based (Thresholds)	UNK
Siemens US7475292B2	Inspection-based (Thresholds)	UNK
Siemens US8055952	Inspection-based (Thresholds)	UNK
Siemens US2011/0072315A1	Inspection-based (Thresholds)	UNK

Fig. 4 presents the classification results of the industry solutions and patents based on the rejuvenation target. There is no clear observed preference. It seems that the rejuvenation target is defined in an ad hoc manner based on the product architecture and characteristics. The application (APP) and application component (CMP) target are predominant with respect the other targets. However, the number of solutions analyzed is not representative to generalize any conclusion. Due to the fact that patents' descriptions are often ambiguous or too general, it is impossible to determine without any doubts which rejuvenation target is addressed.

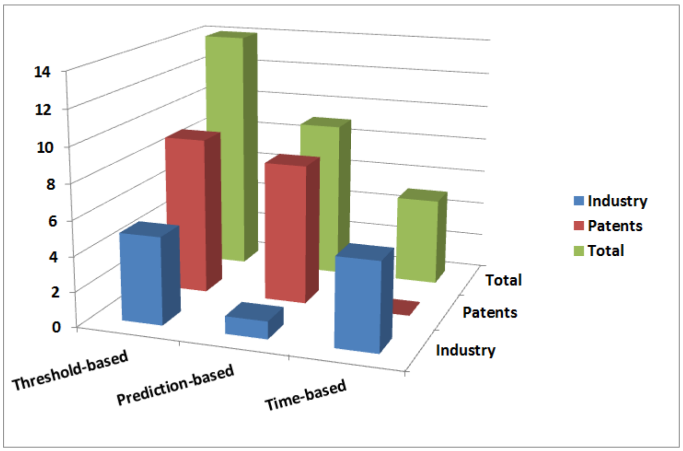

Figure 3. Industry and US patents scheduling classification

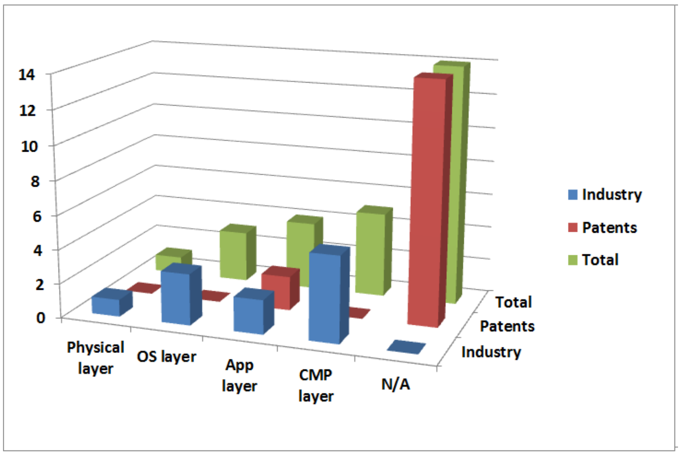

Figure 4. Industry and US patents rejuvenation target classification

V. DISCUSSION AND FUTURE WORK

The study conducted about IT and Telco industry solutions of the last decade leads to:

- Software rejuvenation is a mature technique in the industry, since it is used across different software systems. However, different names are used instead of the term rejuvenation (e.g., proactive recovery and proactive/preventive maintenance).
- The time-based and threshold-based scheduling approaches are predominant in real scenarios being simpler and deterministic compared with prediction-based scheduling. Prediction-based rejuvenation is also addressed in industry since we have found many patents focusing on this. However, these prediction-based techniques are not yet considered mature and stable enough to be adopted in the field.
- There is no clear trend in terms of the rejuvenation target. The real solutions mainly address the rejuvenation of the applications and the application components. However, due to the ambiguity of some of the solutions, especially of the patents, it is not possible to fully analyze this aspect without field experience.
- Based on the system type studied, it is observed that similar process recycling mechanisms are adopted in client-server systems (e.g., Oracle DB, Apache HTTP Server and Microsoft IIS). However, it seems that rejuvenation solutions are in general defined and used in an ad-hoc manner.

To generalize this study it is necessary to examine a richer and more detailed bunch of industry solutions. Furthermore, it will be refined the search criteria in order to improve the body of knowledge of rejuvenation solutions analyzable. Finally, it will be investigated the differences, if any, between the rejuvenation solutions developed across different types of industries.

ACKNOWLEDGMENT

The work by J. Alonso and K. S. Trivedi is supported in part by Huawei Technologies Co. Ltd. The work of A. Bovenzi has been partially supported by the Italian industrial research project "Iniziativa Software CINI-Finmeccanica"

REFERENCES

[1] Y. Huang, C. Kintala, N. Kolettis, and N. Fulton, "Software rejuvenation: analysis, module and applications," in *Proc. 25th Int'l. Symp. Fault-Tolerant Computing, 1995,* pp.381-390.

[2] M. Grottke, R. Matias, and K. Trivedi, "The fundamentals of software aging," in *Proc. IEEE 1st Int'l. Workshop on Software Aging and Rejuvenation, 2008.*

[3] D. Cotroneo, R. Natella, R. Pietrantuono, and S. Russo, "Software aging and rejuvenation: Where we are and where we are going", in *Proc. IEEE 3rd Int'l. Workshop on Software Aging and Rejuvenation, 2011.*

[4] V. Castelli, R. E. Harper, P. Heidelberger, S. W. Hunter, K. S. Trivedi, K. Vaidyanathan, and W. P. Zeggert, "Proactive Management of Software Aging", *IBM Journal of Research & Development, Vol. 45, No. 2, March 2001, pp. 311-332.*

[5] K. J. Cassidy, K. C. Gross, and A. Malekpour, "Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers", in *Proc. the 32nd Int'l. Conf. Dependable Systems and Networks, 2002, pp.478-482.*

[6] K. Vaidyanathan and K. Trivedi, "A comprehensive model for software rejuvenation", *IEEE Trans. Dependable Secure Computing, Vol. 2, No 2, 2005, pp. 124–137.*

[7] J. Alonso, L. Silva, A. Andrzejak, P. Silva, and J. Torres, "High-available grid services through the use of virtualized clustering", in *Proc. the 8th IEEE/ACM Int'l. Conf. Grid Computing, 2007, pp. 34–41.*

[8] J. Alonso, J. Torres, J. Berral, and R. Gavalda, "Adaptive on-line software aging prediction based on machine learning", in *Proc. the 40th Int´l Conf. Dependable Systems and Networks, 2010, pp. 507–516.*

[9] J. Alonso, R. Matias, E. Vicente, A. Maria, and K. Trivedi, "A comparative evaluation of software rejuvenation strategies", in *Proc. the IEEE 3rd Int'l Workshop on Software Aging and Rejuvenation, 2011.*

[10] International Data Group @ONLINE (Sept. 2012) URL http://www.idg.com/

[11] Gartner Inc. Technology research @ONLINE (Sept. 2012) URL http://www.gartner.com

[12] InfoTech–Research group @ONLINE (Sept. 2012) URL http://www.infotech.com

[13] ReadSoft @ONLINE (Sept. 2012) URL http://www.readsoft.com

[14] Cisco Systems Inc. “Cisco security advisor: Cisco Catalyst memory leak vulnerability,” Document ID: 13618, 2000, @ONLINE (Sept. 2012) http://www.cisco.com/warp/public/707/cisco-sa-20001206-catalyst-memleak.pdf

[15] D. Cotroneo, N. Natella, R. Pietrantuono, and S. Russo, “Software aging analysis of the Linux operating system,” in *Proc. IEEE Int’l. Symp. Software Reliability Engineering, 2010. pp. 71-80.*

[16] A. Bovenzi, D. Cotroneo, R. Pietrantuono, and S. Russo, “On the aging effects due to concurrency bugs: A case study on MySQL”, *to appear in Proc. IEEE Int’l. Symp. Software Reliability Engineering, 2012.*

[17] L. Silva, H. Madeira, and J. G. Silva, “Software aging and rejuvenation in a SOAP-based server”, in *Proc. 5th IEEE Int’l. Symp. on Network Computing and Applications, 2006.*

[18] M. Grottke, A. P. Nikora, and K. S. Trivedi, An empirical investigation of fault types in space mission system software”, in *Proc. the 40th Int´l. Conf. Dependable Systems and Networks, 2010, pp. 447-456.*

[19] K. Kourai and S. Chiba, “Fast software rejuvenation of virtual machine monitors” *IEEE Trans. on Dependable and secure computing, Vol. 8, No. 6, Nov./Dec. 2011, pp. 839-851.*

[20] K. Vaidyanathan, R. E. Harper, S. W. Hunter, and K. S. Trivedi, “Analysis and implementation of software rejuvenation in cluster systems”, in *Proc. ACM SIGMETRICS Int’l Conf. Measurement and modeling of computer systems, 2001, pp. 62–71.*

[21] K. Yamakita, H. Yamada, and K. Kono, “Phase-based reboot: Reusing operating system execution phases for cheap reboot-based recovery”, in *Proc. the 41st Int’l. Conf. Dependable Systems and Networks, 2011, pp. 169-180.*

[22] A. Pfiffer, “Reducing system reboot time with kexec” @ONLINE (Sept 2012). URL. http://devresources.linux-foundation.org/andyp/kexec/whitepaper/kexec.pdf

[23] R. Matias, P. J. F. Filho, An experimental study on software aging and rejuvenation in web servers, in *Procs. the 30th Annual Int’l Computer Software and Applications Conf. Vol. 01, 2006, pp. 189–196.*

[24] G. Candea, S. Kawamoto, Y. Fujiki, G. Friedman, and A. Fox, “Microreboot - a technique for cheap recovery”, in *Proc. the 6th Symp. on Opearting Systems Design & Implementation, 2004, pp. 31–41.*

[25] Alcaltel-Lucent - OmniSwitch CLI Reference Guide @ONLINE (Sept. 2012) http://enterprise.alcatel-lucent.com/docs/?id=19012 (pp. 53-4)

[26] Avaya – Avaya Servers and Media Gateways – Software failure recovery @ONLINE (Sept. 2012) http://downloads.avaya.com/css/P8/documents/100018347

[27] L. Li, K. Vaidyanathan, and K. S. Trivedi. “An Approach for Estimation of Software Aging in a Web Server” *in Proc. Intl. Symp. Empirical Software Engineering, 2002.*

[28] Apache HTTP server – MPM worker @ONLINE (Sept. 2012) http://httpd.apache.org/docs/2.4/mod/worker.html

[29] Microsoft IIS 6.0 server @ONLINE (Sept. 2012) http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/6f66808c-230c-4b0e-a922-42318a3095e1.mspx

[30] ORACLE DBMS_CONNECTION_POOL @ONLINE (Sept. 2012) http://docs.oracle.com/cd/B28359_01/appdev.111/b28419/ d_connection_pool.htm

[31] JBoss Database connection Pool (DBCP) @ONLINE (Sept. 2012) http://docs.jboss.org/jbossweb/2.1.x/printer/jndi-datasource-examples-howto.html

[32] S. H. Dwarakanath, M. VanderBilt, and J. M. Zook, US patent 7610214, “Robust forecasting techniques with reduced sensitivity to anomalous data”, Amazon Technologies, 2005, @ONLINE (Sept. 2012) http://www.google.com/patents/US7610214

[33] P. Bajpay, H. Eslambolchi, J. McCanuel, and M. Rahman, US patent 7881189, “Method for providing predictive maintenance using VoIP post dial delay information”, AT&T, 2011, @ONLINE (Sept. 2012) http://www.google.com/patents/US7881189

[34] E. J. Groenendaal, P. J. Hanselmann, and G. M. Clendon, US patent 6993686 “System health monitoring and recovery”, Cisco Technology Inc., 2006, @ONLINE (Sept. 2012) http://www.google.com/patents/US6993686

[35] R. E. Harper and S. W. Hunter, US patent 6978398 “Method and system for proactive reducing the outage time of a computer system”, IBM Corp., 2005, @ONLINE (Sept. 2012) http://www.google.com/patents/US6978398

[36] S. R. Yu, US patent 20080163004 “Minimizing software downtime associated with software rejuvenation in a single computer system”, IBM Corp., 2011, @ONLINE (Sept. 2012) http://www.google.com/patents/US20080163004

[37] V. Castelli and P. A. Franaszek, US patent 6993458 “Method and apparatus for preprocessing technique for forecasting in capacity planning, software rejuvenation and dynamic resource allocation applications”, IBM Corp., 2006, @ONLINE (Sept. 2012) http://www.google.com/patents/US6993458

[38] M. Gupta, J. E. Moreira, A. J. Oliner, and R. K. Sahoo, US Patent 7895323 “Hybrid event prediction and system control”, IBM Corp., 2011, @ONLINE (Sept. 2012) http://www.google.com/patents/US7895323

[39] N. Chandwani, U. Mukherjee, C. Hiremath, and R. Dodeja, US patent 7702966 “Method and apparatus for managing software errors in a computer system”, Intel Corp., 2010, @ONLINE (Sept. 2012) http://www.google.com/patents/US7702966

[40] B. Sridharan, E. Nallipogu, A. Heddaya, M. R. Garzia, B. Levidow, US patent 8117505, “Resource exhaustion prediction, detection, diagnosis and correction”, Microsoft Corp., 2012, @ONLINE (Sept. 2012) http://www.google.com/patents/US8117505

[41] D. Ray, US patent 7227845, “Method and apparatus for enabling a communication resource reset”, Motorola Inc., 2007, @ONLINE (Sept. 2012) http://www.google.com/patents/US7227845

[42] K. Vaidyanathan, K. C. Gross, US patent 7543192, “Estimating the residual life of a software system under a software-based failure mechanism”, Sun Microsystems Inc., 2009, @ONLINE (Sept. 2012) http://www.google.com/patents/US7543192

[43] A. B. Bondi and A. Avritzer, US Patent 2006015699, “Inducing diversity in replicated systems with software rejuvenation”, Siemens Corp., 2009, @ONLINE (Sept. 2012) http://www.google.com/patents/US20060156299

[44] A. Avritzer, US patent 2007/0250739, “Accelerating software rejuvenation by communicating rejuvenation events”, Siemens Corp. 2010, @ONLINE (Sept. 2012) http://www.google.com/patents/US20070250739

[45] A. Avritzer, US patent 7475292, “System and method for triggering software rejuvenation using a customer affecting performance metric”, Siemens Corp., @ONLINE (Sept. 2012) http://www.google.com/patents/US7475292

[46] A. Avritzer and A. B. Bondi, US patent 8055952, “Dynamic tuning of a software rejuvenation method using a customer affecting performance metric”, Siemens Corp., @ONLINE (Sept. 2012), http://www.google.com/patents/US8055952

[47] A. Avritzer, US patent 2011/0072315, “System and method for multivariate quality-of-service aware dynamic software rejuvenation”, Siemens Corp., @ONLINE (Sept. 2012) http://www.google.com/patents/US20110072315