Goal-based Dependability Analysis and Improvement Framework for Network Storage Systems

Hua Wang
birch_wh@163.com

Ke Zhou
k.zhou@hust.edu.cn

College of Computer Science and Technology
Huazhong University of Science & Technology, Wuhan, China
Wuhan National Laboratory for Optoelectronics, China

Abstract

On the basis of deep research on dependability of computer system, we extend the notion of dependability attribute, which is the dependability analysis object of traditional dependability theory, and present the multidimensional dependability object. We put forward a scientific and practical goal-based dependability analysis and improvement framework for computer systems. Aiming at the characteristics of network storage systems, we devise a reusable database for dependability analysis and improvement. According to the framework and the reusable database, a user can be directed and elicited to perform the analysis and improvement towards a network storage system.

1. Introduction

Dependability issues of computer systems are gaining more and more attention. The International Federation for Information Processing Working Group10.4 defines dependability as "the trustworthiness" of a computing system, which allows reliance to be justifiably placed on the services it delivers [1]. On the basis of research on the dependability of computer systems, how to discover the factors affecting dependability, so as to attain system dependability by some methods, has come to be a hot topic. Currently correlative research institutions and groups have gained certain achievements on computer system dependability whose focus is in the macroscopical layer, giving the definition of the basic concepts and taxonomy of computer system dependability, named as dependability tree, which includes dependability attributes, threats, means to attain dependability [2][3][4]. Research on computer system dependability is a systematic subject, which can't deviate from the affected factors of computer systems such as application types and stakeholders [5][6].

Network storage is a new direction of the storage technology, which is to gain higher storage goal by combining network technologies and storage technologies. Nowadays dependability research on network storage field is constrained within storage data dependability, which focuses on providing methods for determining which data protection techniques should be applied to each application and how to set the configuration parameters for these techniques and the resources they use [7][8][9][10]. Few available literatures have been found that aim at dependability research on the whole network storage system.

On the basis of deep research on dependability of computer system, this paper presents a scientific and practical goal-based dependability analysis and improvement framework. Aiming at the characteristics of network storage system, we devise a reusable database for dependability analysis and improvement. According to the framework and the reusable database, a user can be directed and elicited to perform the analysis and improvement towards a network storage system. At the same time, we put forward the notion of multidimensional dependability object, the extension on dependability correlative concepts provides good reference for various dependability analysis for computer systems.

The remainder of the paper is organized as follows: Section 2 discusses the corresponding concepts of dependability theory, including dependability object, threats, means to attain dependability, within which

978-0-7695-3187-8/08 $25.00 © 2008 IEEE
DOI 10.1109/NAS.2008.49

extends the notion of dependability object. Section 3 presents a goal-based dependability analysis and improvement framework. Section 4 illustrates dependability analysis and improvement for network storage systems. By using a reusable database, a user can be elicited to perform the analysis and improvement towards a network storage system. The last section gives the conclusion and future research directions.

2. Fundamental concepts of dependability

2.1. Dependability object

Currently, analysis object about dependability issues of computer system is dependability attributes [2][3][4], however, research on dependability attributes cannot depart from the affecting factors such as application types, stakeholders and environment. Through introducing the multidimensional dependability object to replace simple dependability attributes, dependability analysis can be directed more easily and the analysis result can be more concrete and effective.

Aiming at inside disciplines and characteristics of computer system dependability, target object of dependability analysis can be defined as four dimensional vector: dependability analysis scope, dependability attributes, stakeholders and application types.

Dependability analysis scope is used to confine the bound or range of dependability analysis of computer systems.

Dependability is an integrative concept that encompasses many dependability attributes, which are used to describe system dependability from different facets. There are many compositions of dependability attributes. The popularly accepted definition is that dependability encompasses the following basic attributes: availability, reliability, safety, confidentiality, integrity and maintainability [2][3][4]. According to system characteristics, a group of appropriate dependability attributes can be induced. In the condition of fixed system cost, contradictory may exist among different dependability attributes, for example, improving security may lead to a performance decrease, fault tolerance is opposite to testability.

Stakeholders conduct the action of dependability analysis. Stakeholders include developers, maintainers, users, managers, etc. Different stakeholders may focus on different dependability attributes. A developer may be more concerned about availability and reliability of a system, a maintainer may be more interested in maintainability, while the manager pays attention to safety and confidentiality. Different stakeholders have different weight constitutions of dependability attributes.

Application types are the applicable circumstances in which systems exist. Different applicable systems have different dependability requirements. Real-time workloads require rapid response time, non real-time systems may require higher throughput rate. Different application types have different weight constitutions of dependability attributes. We must define appropriate anticipated values aiming at certain application requirements and stakeholders' desire for different attributes.

The process for defining the dependability object is illustrated as Figure 1. The first step is to confirm the dependability analysis scope for a specific system. The second step is to ascertain the constitution of dependability attributes. The third step is to make sure of the concrete stakeholders and the weight constitution of dependability attributes. The fourth step is to define weight constitution of dependability attributes towards a specific application type. The fifth step is to synthesize the previous two groups of weight and form the synthesis weight of dependability attributes. It is important of weight definition because the synthesis weight will affect setting the target value of each dependability attribute directly.

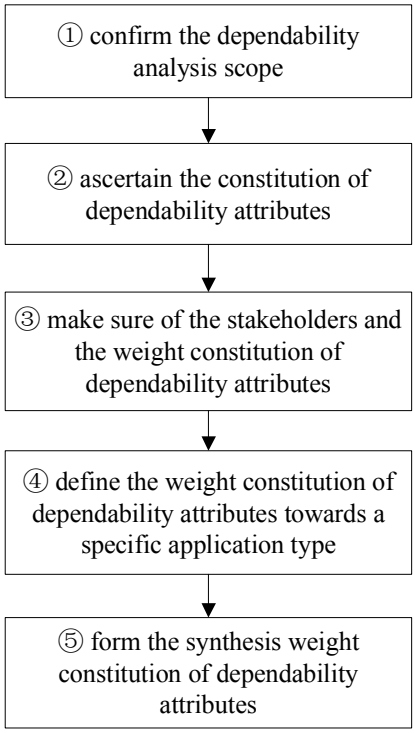

Figure 1. The process for defining a dependability object

Suppose that the system dependability is represented as D, which encompasses n attributes: A_1, A_2, …, A_n. To a specific stakeholder or an application type, according to the priority of each attribute, the weight factors can be expressed as W_1, W_2, …, W_n, the weight constitution of dependability attributes can be shown as Figure 2.

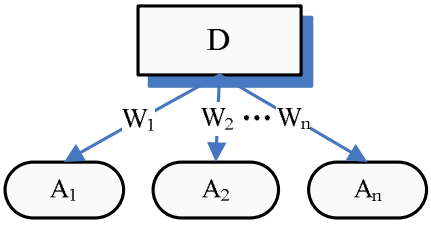

Figure 2. Weight constitution of dependability attributes

The relationship among weight factors can be denoted as: $\sum_{i=1}^{n} W_i = 1$, the synthesis weight also satisfies the equation.

2.2. Threats

According to the different appearance levels, the threats of dependability are the failures, the errors and the faults [2][3][4][11].

A failure is a deviation of the external state of a system from the correct one. It lies in the operational layer. Failures have many classifications, in terms of domain, it can be classified into content failure and time failure, In terms of severity, it can be divided into harmless failure, medium harmful failure and catastrophic failure.

An error is a deviation of the internal state of a system from the correct value. It lies in the information processing layer. Only when there is information processing, such as data or signal processing, may errors occur. An error is often neglected by users, but it will lead to system failure directly.

A fault arises in the devices of the system or is produced by human beings. It lies in the physical layer. A fault can be caused by natural reasons, such as hardware fault and network fault. It can also be caused by man-made factors, which may be produced during software development or operational process. According to the modality of origin, it can be divided into malicious fault, deliberate fault, accidental fault and incompetence fault. According to the activity state, it can be divided into dormant fault, latent fault and active fault.

The relationship among the above three threats can be described as a transform chain: a failure is generated by an error and an error by a fault. Perceived factors affecting system dependability are system failures, which are induced by faults. A fault is the triggering-off source of the mechanism. So in order to avoid system failures and to improve system dependability, we need to decrease or avoid the emergence of system faults at root.

2.3. Means to attain dependability

The means to attain dependability encompasses four types: fault prevention, fault tolerance, fault removal and fault forecasting [2][3][4].

Fault prevention runs through every phase of system establishment, such as requirement analysis, system design and system implementation. It emphasizes taking some measures to prevent the occurrence or introduction of faults. Fault prevention can be aimed at software or hardware.

Fault tolerance means to avoid service failures in the presence of faults. When there are active faults, we should take measures to prevent system from stepping into failure state, so as to ensure system to provide the service that meets the request.

Fault removal means that when a fault occurs, we should detect it by verification, then diagnose and correct it, ultimately reduce the number and severity of faults.

Fault forecasting is the estimation that aims at future failure's occurrence probability and the consequence by dependability specification, allocation and evaluation on different system components.

3. Dependability analysis and improvement framework for computer systems

From the viewpoint of dependability theory of computer system, the dependability analysis framework encompasses three elements: dependability object, threats, means to attain dependability, which is shown as Figure 3. Dependability object is extended from simple dependability attributes into a four dimensional vector: analysis scope, dependability attributes, stakeholders and application types.

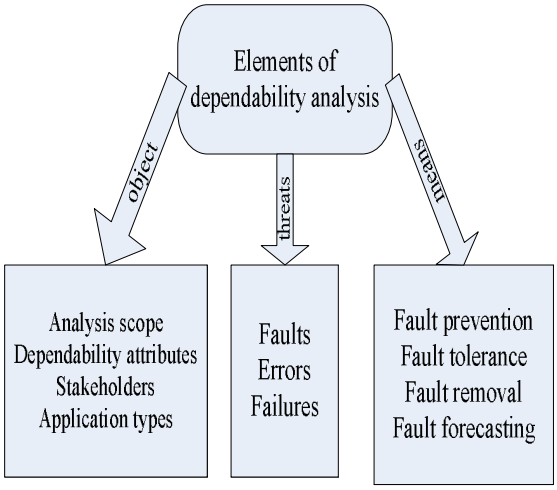

Figure 3. Elements of the dependability analysis framework

On the basis of research on dependability theory of computer system, we present a goal-based dependability analysis and improvement framework, which is illustrated as Figure 4.

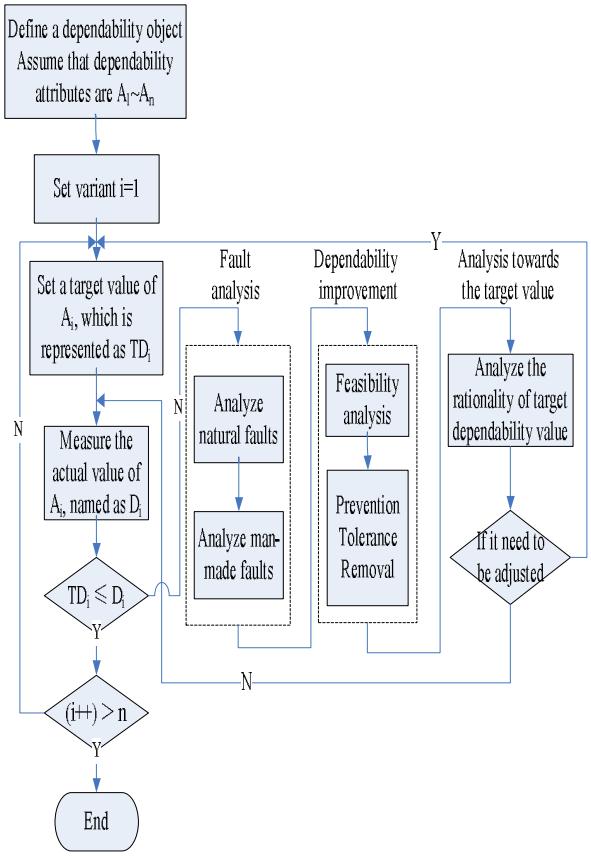

Figure 4. The goal-based dependability analysis and improvement framework for computer systems

Dependability analysis and improvement of computer system encompasses four steps, which are described as follows:

The first step: define dependability object, set up a target dependability value, measure the actual dependability value, then compare the two values and decide if it is necessary to improve dependability. First, according to Figure 1, make certain all dimensions of the dependability object, which includes analysis scope, dependability attributes, stakeholders and application types. Let A_1~A_n represent dependability attributes, we perform the following processes towards the ith attribute repeatedly ($1 \leq i \leq n$): According to the requirements of users and the specific application type, on the basis of considering economic and technological feasibility, we set up a target value for the dependability attribute, which is represented as TD_i. We measure the actual value of the dependability attribute, named as D_i. We compare the previous two values, if $TD_i \leq D_i$, then dependability of the attribute satisfies request, we should turn to the next attribute and perform the process repeatedly. Otherwise dependability of the attribute does not meet the request, we need to take actions to improve dependability, and turn to the next step.

The second step: analyze the reasons that influence dependability, the reasons may be natural faults or man-made faults, analyze their effect on dependability and decide the regions for improvement.

The third step: on the basis of fault analysis, start feasibility analysis for dependability improvement, considering the synthesis weight and improvement cost, decide if we should take any action. If it is necessary, we will improve system dependability by means of fault prevention, fault tolerance, fault removal, and so on.

The fourth step: on the basis of fault analysis and dependability improvement, analyze if the target value for the dependability attribute need to be adjusted. For example, if the difference between TD and D of a dependability attribute is very much, at the same time, there is a narrow improvement space and the cost is very high, then the target value has been set too high. On the contrary, if fault reasons are numerous and are easy to be removed, the difference between TD and D is not very much, then the target value is too low. If any of the previous situations occurs, we need to adjust the target value. No matter in which situation, we have to return to the corresponding activity point, which is shown as Figure 4.

We can implement dependability analysis and improvement for a whole system by following the previous steps.

4. Dependability analysis and improvement for network storage systems

According to the previous dependability analysis and improvement framework, aiming at the characteristics of network storage systems [12], we can pick-up the reusable items of network storage systems from the aspects of dependability object, fault types, dependability improvement, and build a reusable database for dependability analysis and improvement for network storage systems, the detail content is shown as Table 1.

Table 1. A reusable DB for dependability analysis and improvement for network storage systems

class	subclass	reusable items
Dependability object	Dependability attributes	Reliability, availability, security, response time, throughput rate
	Analysis scope	Hardware, software and network Storage systems and nodes
	Stakeholders	Developers, maintainers, users, managers, etc.
	Application types	Real-time workloads, non real-time workloads
Fault types	Natural reasons	Hardware faults, network faults, software faults
	Man-made reasons	Error operations, malicious intrusions
Dependability improvement	Fault prevention	Improving device's reliability to avoid fault Improving storage software's dependability
	Fault tolerance	Redundant component design
	Fault removal	Diagnosing and recover Reducing the recover time

The process of dependability analysis and improvement for network storage systems is shown as Figure 5. It encompasses the following steps: ① Towards each attribute, set a target value, measure the actual value, compare the two values.②Fault analysis. ③ Dependability improvement. ④ Analyze the rationality of the target value. Towards each step, we can utilize the available content in the reusable database, or elicited by the reusable database to customize towards a specific system. After the completion of the process, we should extract reusable content to reinforce the database or do some modification to perfect the database, so as to provide better service for future reuse.

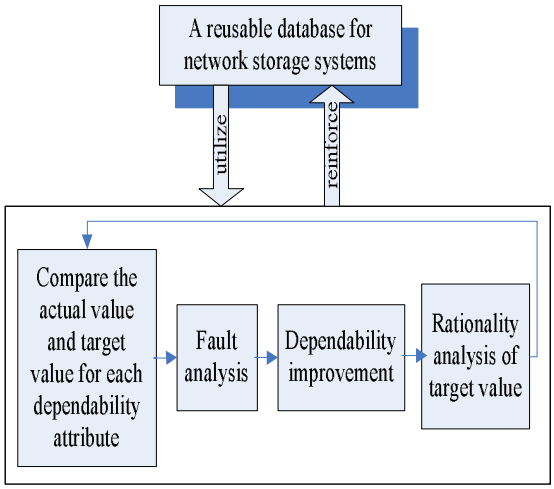

Figure 5. The process of dependability analysis and improvement for network storage systems

5. Conclusions

On the basis of deep research on dependability theory of computer systems, we extend the notion of traditional dependability object, which is dependability attributes, and present the multidimensional dependability object creatively, which not only encompasses dependability attributes, but also analysis scope, stakeholders and application types. Then we put forward a scientific and practical goal-based dependability analysis and improvement framework for computer systems. Aiming at the characteristics of network storage systems, we devise a reusable database for dependability analysis and improvement. According

to the framework and the reusable database, a user can be directed and elicited to perform the analysis and improvement towards a network storage system.

The future research is to develop a prototype system of dependability analysis and improvement framework for network storage systems. During the process for analyzing and improving dependability of typical network storage systems, through comparing with experiential result, we should modify and perfect the framework. On the other hand, in the process of fault analysis, we will apply analysis method of fault tree to improve the effect and quality of fault analysis.

Acknowledgements

We thank the reviewers for their helpful comments and insights. This work was supported by National Science Foundation of China under Grant No.60503059, National Basic Research Program of China (973 Program) under Grant No.2004CB318201, the Program for New Century Excellent Talents in University NCET-06-0650.

References:

[1] Victor Basili, Paolo Donzelli, Sima Asgari, "A Unified Model of Dependability: Capturing Dependability in Context", IEEE SOFTWARE, November/December 2004, pp. 19-25.

[2] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, January-March 2004, pp. 11-33.

[3] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, "Fundamental Concepts of Dependability", UCLA CSD Report no. 010028, LAAS Report no. 01-145, Newcastle University Report no.CS-TR-739.

[4] Jean-Claude Laprie, "Dependability of Computer Systems: Concepts, Limits, Improvements", 1995 IEEE, pp. 2-11.

[5] Jeffrey Voas, "Trusted Software's Holy Grail", Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), January 2003, pp. 6-9.

[6] Duy Huynh, Marvin V. Zelkowitz, Victor R. Basili, Ioana Rus, "Modeling dependability for a diverse set of stakeholders", Copyright © 2003 University of Maryland.

[7] Kimberly Keeton, Arif Merchant, "A Framework for Evaluating Storage System Dependability", Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN'04), 2004, pp. 877-886.

[8] Shravan Gaonkar, Kimberly Keeton, Arif Merchant, William H. Sanders, "Designing dependable storage solutions for shared application environments", Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN'06), 2006, pp. 371-382.

[9] Kimberly Keeton, John Wilkes, "Automatic design of dependable data storage systems".

[10] Kimberly Keeton and John Wilkes, "Automating data dependability", 10th ACM-SIGOPS European Workshop, Saint-Emilion, France, September 2002.

[11] Giuseppe Buja , Roberto Menis, "Conceptual frameworks for dependability and safety of a system", SPEEDAM 2006, International Symposium on Power Electronics, Electrical Drives, Automation and Motion, pp. S35–23--S35-28.

[12] Mike Mesnier , Gregory R. Ganger, Erik Riedel, "Object-based storage", IEEE Communications Magazine , August 2003, pp. 84-90.