Reliability Engineering and System Safety 147 (2016) 142–155

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Multi-objective optimization of IT service availability and costs

Sascha Bosse *, Matthias Splieth, Klaus Turowski

Magdeburg Research and Competence Cluster for Very Large Business Applications, Faculty of Computer Science, Otto von Guericke University Magdeburg, P.O. Box 4120, 39016 Magdeburg, Germany

ARTICLE INFO

Article history:
Received 30 April 2015
Received in revised form
15 October 2015
Accepted 7 November 2015
Available online 19 November 2015

Keywords:
IT service management
Availability
Reliability
Redundancy allocation problem
Cost optimization

ABSTRACT

The continuous provision of highly available IT services is a crucial task for IT service providers in order to fulfill service level agreements with customers. Although the introduction of redundant components increases availability, the associated cost may be very high. Therefore, decision makers in the IT service design stage face a trade-off between cost and availability in order to define suitable service level objectives. Although this task can be seen as a redundancy allocation problem, the existing definitions in this area are not transferable to IT service design due to the assumption of independent component failures, which has been identified as unrealistic in IT systems.

In this paper, a multi-objective redundancy allocation problem for IT service design is defined. Therefore, a Petri net Monte Carlo simulation is developed that estimates the availability and costs of a specific design. In order to provide (sub)optimal solutions to an IT service redundancy allocation problem, two meta-heuristics, namely a genetic algorithm and tabu search, are adapted. The approach is utilized to optimize the IT service design of an application service provider in terms of availability and cost to demonstrate its feasibility and suitability.

1. Introduction

The importance of IT services is ever increasing. On one hand, trends such as Cloud Computing bring millions of consumers in contact with IT services. On the other hand, even internal IT organizations are commonly understood as IT service providers in order to effectively manage costs and the business value of IT [1]. Service or Operational Level Agreements (SLAs/OLAs)[1] document the quality of service that is to be expected by an IT service consumer, for instance, for the service availability [2].

Availability is one of the most crucial quality aspects for customers [3], and can be defined as the likelihood that a service is able to provide its function at a certain point in time [4]. Although service availability is seen "at the core of customer satisfaction and business success" [5, p. 127], even the big IT companies are suffering severe service disruptions that last hours or even days (e.g. Amazon [6], Apple [7] and Microsoft [8]). In August 2013, a five minute inaccessibility of the Google services led to a 40% decrease in internet traffic and an estimated revenue loss of over US-$500,000 for Google alone [9]. However, smaller enterprises are also affected by unavailability: 134 companies that have been studied by the Aberdeen Group each suffered on average a revenue loss of more than one million US-$ in 2012 due to IT downtime [10].

The two basic approaches to increase system availability are the introduction of more reliable components and the implementation of redundancy mechanisms [11]. However, the associated cost of these approaches may not be justified by the availability improvements. In addition to this, the special characteristics of software components limit their reliability [12]. The balancing of availability and cost with respect to desired or existing service level objectives is one of the core activities in IT Service Management, and is described in well-known frameworks such as the ISO 20000 (Service Continuity and Availability Management) [13], CoBIT 5 (Managing Availability and Capacity) [14] and the IT Infrastructure Library (ITIL) (Availability Management).

In the 2011 version of ITIL, service availability is characterized as an essential service quality attribute immediately influencing customer satisfaction [5]. Additionally, SLA violations in the operation phase can lead to penalty costs and loss of reputation for the IT service provider [15]. Since design changes due to insufficient service quality in the operation phase (reactive measures) can be very costly, measures to achieve sufficient IT service availability should be considered in the service design stage (proactive measures) [5,16]. Nevertheless, the lack of feasible supporting tools for high availability design is also noticed [5]. This is mainly caused by the fact that classical analytical availability/reliability models that have been successfully applied in other domains assume independent component failures [17]. This

* Corresponding author.
E-mail addresses: sascha.bosse@ovgu.de (S. Bosse), matthias.splieth@ovgu.de (M. Splieth), klaus.turowski@ovgu.de (K. Turowski).

[1] In the following, the term SLA(s) is used for both SLAs and OLAs.

http://dx.doi.org/10.1016/j.ress.2015.11.004
0951-8320/

assumption is unrealistic in modern IT systems due to the presence of inter-component dependencies, thus rendering results obtained from these models useless for decision support [18].

Examples of these inter-component dependencies are common cause failures and imperfect switching. In the former case, even heterogeneous components can be subject to the same fault under certain conditions [12]. Imperfect switching describes the phenomenon that a redundant component may not cover the failure of an active component due to problems in the switching process [19].

In addition, operator interaction may be required for component recovery or switching, leading to the fact that operator errors are a major cause for IT service unavailability [20,21]. However, these errors are not represented in classical availability models.

In order to provide a general modeling approach that overcomes the independent failure assumption, several approaches that are applicable for IT service availability estimation from design information were recently developed. The majority of these approaches are based on the modeling of the availability state-space that allows for the introduction of dependencies [22]. However, the capability of these approaches for decision support in availability management is questionable since these approaches require a high modeling effort and were never integrated with optimization procedures in order to suggest (sub)optimal design configurations.

Such approaches have been developed in the context of redundancy allocation problems (RAPs). Under this term, several reliability/availability models and optimization procedures are subsumed that can be utilized to optimize system design in terms of availability, cost and other constraints. Therefore, required subsystems and possible component choices for each subsystem are modeled so that (sub) optimal combinations of component choices can be identified. Since the combinatorial computation of availability in these approaches assumes independent component failures, they are not applicable for IT service design. Nevertheless, the developed optimization procedures are mainly based on flexible meta-heuristics such as evolutionary algorithms, which can be applied to a wide range of problems. Therefore, the question of whether or not these procedures can be integrated with IT service availability estimation methods for IT service design optimization arises.

The goal of this work is to provide decision support for IT service designers. Therefore, a redundancy allocation problem is defined that models the relevant aspects of IT service availability and costs depending on possible IT service designs. In the course of the paper, this problem is referred to as the ITRAP. In combination with a suitable IT service availability estimation method and solution algorithm, the ITRAP can be utilized to optimize availability and costs of an IT service based on design information.

A constructivist approach is followed in order to achieve this goal (cf. e.g. [23,24]). In Section 2, the related literature is presented to outline the relevance of the investigated problem. In this section, suitable approaches in the topics of IT service availability estimation and redundancy allocation optimization in order to develop a RAP for IT service design are identified as well. Based on the literature analysis, requirements for a RAP for IT service design are derived.

These requirements as well as the ITRAP artifact are presented in Section 3. This artifact is a framework designed in order to reach the goal of this work and consists of the problem definition, an availability and costs estimation method based on Petri net Monte Carlo simulation, and two adapted solution algorithms (a genetic algorithm and a tabu search). The artifact is evaluated by applying a prototypical implementation of the ITRAP to an IT service design optimization in a real-world use-case, an international application service provider, which is presented in Section 4. Section 5 concludes the paper by discussing the contribution of the paper as well as by providing an outlook to further research activities.

2. Related work

2.1. The redundancy allocation problem

In reliability/availability optimization, a redundancy allocation problem (RAP) can be utilized to determine suitable redundancy configurations for systems design. Therefore, it is instantiated with system design information, e.g. about reliability characteristics of possible component choices for required functional units of a system. On that basis, optimization algorithms identify (sub)optimal solutions in terms of availability, cost and other constraints. The first RAPs were defined during the 1960s and were mostly solved by exact solution methods such as linear or dynamic programming. In the last 25 years, more complex definitions were established in order to provide a more realistic model of the investigated systems. Due to the increased complexity of the optimization problem, more efficient solution algorithms, especially meta-heuristics, were applied. In the following, the development of RAP definitions and solution algorithms is briefly sketched. For more information of the RAP topic, one may refer to the literature reviews in [25–27].

Definitions. In 1962, Kettelle was one of the first researchers to describe and solve an optimization problem in which cost is to be minimized subject to an availability constraint [28]. The term redundancy allocation problem was first used by Fyffe et al. in 1968 [29]. Researchers utilized RAPs to maximize availability/reliability (e.g. in [29,30]), to minimize cost (e.g. in [28,31,32]), as well as for the multi-objective optimization of availability and cost (e.g. in [33–38]).

In Chern et al. [39], defined a RAP, which they proved to be a NP-hard optimization problem [39]:

(i) A system consists of s required subsystems (series–parallel system).
(ii) In a subsystem, a number of functionally equal components can be used in active redundancy.
(iii) A subsystem component can either be working or failed (binary-state); failures are independent and identically distributed in a subsystem (homogeneous redundancy).
(iv) The reliability of the system is to be maximized subject to linear constraints such as cost, weight, or volume.

In the subsequent years, this definition was further extended in order to provide a more realistic problem. Some literature examples for introduced characteristics are presented in Table 1.

Nevertheless, only a few works could be identified in the RAP literature that deal with failure dependencies and, therefore, are not using combinatorial approaches to estimate system availability. In [12], Chi and Kuo modeled common cause failures in

Table 1
Included characteristics in RAP definitions with exemplary references.

Characteristic	Description	References
Heterogeneous redundancy	Subsystem components may have different failure distributions	[33,34,40–42]
Passive redundancy	Decreased failure rate for passive components	[38,43–45]
Complex design	Subsystems can be arranged hierarchically/arbitrary	[46–48]
Multi-state components	Modeling of performance degradation	[31,49,50]
Uncertainty	Stochastic [44,51], fuzzy [11,37,52], fuzzy-random [53] and interval [43,47,54] input parameters	