Applied Mathematical Modelling 39 (2015) 5681–5692

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Service reliability modeling of distributed computing systems with virus epidemics

Yan-Fu Li [a], Rui Peng [b,*]

[a] Ecole Centrale Paris – SUPELEC, Paris, France
[b] Dongling School of Economics and Management, University of Science & Technology Beijing, Beijing, China

ARTICLE INFO

Article history:
Received 21 November 2012
Received in revised form 17 December 2014
Accepted 12 January 2015
Available online 29 January 2015

Keywords:
Service reliability
Distributed computing system
Virus epidemics
Continuous-state model
Differential equations
Universal generating function

ABSTRACT

Distributed computing (DC) system is widely implemented due to its low setup cost and high computational capability. However, it might be vulnerable to malicious attacks like computer virus due to its network structure. The service reliability, defined as the probability of fulfilling a task before a specified time, is an important metric of the quality of a DC system. This paper attempts to model and compute the service reliability for the DC system under virus epidemics. Firstly, the DC system architecture is modeled by an undirected graph whose nodes (i.e. computers) have a continuous-state model representing its computational capability. Then a set of epidemic differential equations are formulated and solved to obtain the state dynamics of each node under the virus epidemics. A universal generating function (UGF) based approach is proposed to calculate the service reliability of DC system. Numerical results show the effectiveness of the proposed method. The sensitivity analysis on the model parameters, the comparison with centralized computing system and the optimization of defense level parameter are also conducted.

1. Introduction

Distributed computing (DC) system [1] is a collection of multiple autonomous computers that can communicate through a computer network to solve a large computational task. The purpose of the DC system is to coordinate the use of shared resources and provide communication services to the users [2]. Comparing to the centralized computing system, the DC system possesses many advantages, such as high performance, low setup cost, and potential for enhanced reliability [3–5]. Therefore, the DC system has gained increasing popularity in many application fields such as distributed software/hardware system [4,6], distributed power generation [7], distributed sensor system [8], and etc.

Like many other computing systems, the service quality of the DC system is of high concern to the majority of practitioners. Service reliability, which measures the capability of a system to accomplish its tasks on time, is a very important metric of DC system service quality [9]. Many research works have been devoted to modeling and analysis of the reliability (including service reliability) of DC systems [9–13]. However, most of the previous research works have focused on the failures caused by the 'unintentional' defects embedded in the DC hardware infrastructure and the installed software. In practice, external factors such as infective computer virus become widely spread in the current computer networks [14]. In this paper, we focus on the type of virus which can reproduce themselves and infect other computers in the network. If the virus

[*] Corresponding author. Tel.: +86 13051540519.
E-mail address: pengrui1988@ustb.edu.cn (R. Peng).

http://dx.doi.org/10.1016/j.apm.2015.01.036
0307-904X/© 2015 Elsevier Inc. All rights reserved.

successfully parasitizes one computer, it will rapidly copy itself, consume the computing resources (e.g. CPU and memory) of the host, and attempt to infect other healthy computers via network connections (e.g. email, FTP transfer, message exchange, etc). This process will repeat on other infected computers and may eventually lead to a great loss of computational capability of the whole DC system if the situation is not attended to.

Protecting the DC system against the virus attacks becomes an increasingly important issue [6] and this type of protection is clearly different from the protection against ‘acts of nature’ or ‘accidents’ [15]. For example, the CPU breakdown in a computer usually will not affect the operations of other computers connected in the same network. In the literature of reliability research, many studies have been devoted to intentional attack protections by designing protection strategies for different systems (e.g. power substations, defense systems, etc) [16–20], but few have investigated the attacks with epidemic characteristics, such as computer virus [21]. In the field of epidemiology modeling, some research works have addressed the virus spreading issue in computer networks, but the emphasis is on the speed and range of the spreading [21–23]. To bridge the gap, in this work the service reliability of the DC system is modeled and computed under the virus epidemics with the consideration of possible system noises.

The rest of this paper is organized as follows. In Section 2, the general model of virus epidemics is proposed: the continuous-state model is used to describe the computational capability of each node in the DC system and the epidemic differential equations are set and solved to obtain the time-dependent state index. In Section 3, service reliability is defined based on the virus epidemic model and the universal generating function (UGF) technique is adopted for computing service reliability. Section 4 illustrates the proposed model on a numerical example with (1) the sensitivity analysis on the defense level parameter and the processing speed coefficient, (2) the comparison with centralized computing system and (3) the optimization of defense level parameter. Section 5 concludes this study with some possible future research directions.

2. Modeling of virus spreading in distributed computing systems

Notations

Ω	range of continuous state $\Omega = [0, 1]$, where 0 indicates the perfect functioning state and 1 indicates the complete failure state
T	system time
N	total number of nodes in the computer network
G	the undirected graph representing the computer network
V	the set of nodes in the computer network
v_i	node i in the computer network
L	the set of communication channels in the computer network
l_{ij}	the communication channel that links node v_i and v_j
$\mu_i(t)$	the state index of node i at time t
Ψ_i	the neighborhood set of node v_i
E_i	the set of subtasks distributed to node i
δ_i	defense parameter at node v_i
ξ_k	percentage of the raw data in sub-task k
d_k	the amount of data related to subtask k
$\theta_{ki}(t)$	data processing speed of subtask k by node v_i at time t
α_{ki}	processing speed coefficient which links the processing speed to the node state
$R_{ki}(T)$	probability that all the transmission and processing operations of subtask k assigned to node i can be finished by time t
K	total number of subtasks

In this work, the DC system is modeled as an undirected graph $G = (V, L)$, where $V = \{v_i | 1 \leqslant i \leqslant N\}$ is the set of computers (nodes), and $L = \{l_{ij} | 1 \leqslant i \leqslant N, i < j \leqslant N\}$ is the set of communication channels (links) connecting the nodes. It is noted that many authors have assumed homogeneous elements (no difference between nodes and links) in DC system [9,10]. However, for the virus epidemic modeling, nodes are usually treated as infectious components while the links are treated as the non-infectious channels for virus spreading [12,23].

2.1. A continuous-state reliability model of individual nodes

Markov chain is one of the conventional approaches for modeling DC system reliability with a number of discrete states [13], where each node has two states: online (functioning state) or offline (failure state) and the transition diagram is established to model the system state changes. However, the size of Markov state space grows exponentially with the increase of the number of nodes and the degradation states [24]. Moreover, when a node is under virus infection, it will not completely lose its computational capability in a short time. Once a virus successfully resides onto a node, it attaches itself to some executable files. Its code will be executed when one user attempts to launch an infected program. After the execution of