Chaos, Solitons & Fractals 69 (2014) 246–252

Contents lists available at ScienceDirect

Chaos, Solitons & Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

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

Analysis of Linux kernel as a complex network

Yichao Gao, Zheng Zheng *, Fangyun Qin

School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 17 June 2014
Accepted 14 October 2014
Available online 9 November 2014

ABSTRACT

Operating system (OS) acts as an intermediary between software and hardware in computer-based systems. In this paper, we analyze the core of the typical Linux OS, Linux kernel, as a complex network to investigate its underlying design principles. It is found that the Linux Kernel Network (LKN) is a directed network and its out-degree follows an exponential distribution while the in-degree follows a power-law distribution. The correlation between topology and functions is also explored, by which we find that LKN is a highly modularized network with 12 key communities. Moreover, we investigate the robustness of LKN under random failures and intentional attacks. The result shows that the failure of the large in-degree nodes providing basic services will do more damage on the whole system. Our work may shed some light on the design of complex software systems.

1. Introduction

Many natural and technological systems, such as the Internet [1–3], social networks [4–6], biological networks [7–9], and airline networks [10–12], can be represented as complex networks. These networks consist of a set of nodes representing entities of the systems connected by edges indicating relationships between entities. In 1950s, Erdős and Rényi introduced the famous ER random network model [13], which has guided the research on complex networks for decades. Since the small-word network model [14] and scale-free network model [15] were proposed at the end of last century, it is found that many real-world networks are actually complex networks associated with small-world properties and power-law degree distributions. In recent years, the study of complex networks has attracted various scientific communities, such as network modeling [16–18], synchronization [19,20], cascading failures [21,22], evolutionary games [23–25], optimization [26], network traffic [27,28] and epidemic spreading [29,30]. One interesting research direction is to explore software systems in the framework of complex network theory [31–33].

As the centerpiece of information world, software systems play more and more important roles in our daily life. How to build high quality software systems is one of our major concerns. In the last decades, the software engineering, whose objective is to provide methodologies and tools to efficiently design and build software systems [34], has been widely introduced. However, it remains a challenge to model software system due to its high complexity. Software systems are typical man-made systems which can be represented as networks. Valverde et al. [35] presented the first evidence for the emergence of scale-free and small-world structure in software networks. Cai and Yin [36] proposed the software mirror graph to record the runtime information of a software system under given environments. Gorshenev and Pis'mak [37] investigated the evolution of software systems and proposed a model representing the natural selection principle behind software evolution. The studies of software using graph methods are helpful to predict defects of software systems and to develop large software systems with well-designed structures.

Among various software systems, operating system (OS) is the foundation of other kind of software systems.

* Corresponding author.
E-mail address: zhengz@buaa.edu.cn (Z. Zheng).

http://dx.doi.org/10.1016/j.chaos.2014.10.008
0960-0779/

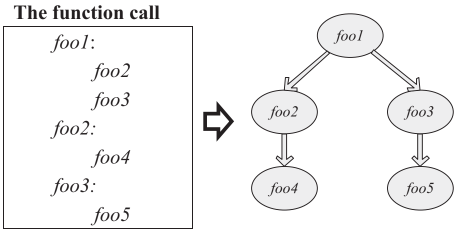

Fig. 1. An illustration of function calls.

It provides interfaces to software and hardware, allowing users to communicate with computers. There are a lot of popular OSs such as Microsoft Windows, Linux, OS X, and UNIX, in which Linux is a well-known open source OS.[1] Thus, ever since it was first released by Linus Torvalds in 1991, Linux has been arousing millions of volunteers' interests to add, change and improve the system. Nowadays, it is a matured OS widely utilized, ranging from cell phones to supercomputers. In this paper, we will study the Linux system from the network point of view. For the high complexity of Linux, we focus on the kernel, which is the innermost part of Linux.

The rest of this paper is organized as follows. Section 2 presents LKN's topological properties to investigate the underlying design principles, followed by the analysis of LKN's community structures and its relationship with kernel's functional modules in Section 3. In Section 4, the robustness of LKN is discussed. Section 5 concludes this study.

2. Topological properties of LKN

Linux kernel consists of thousands of functions which collaborate with each other through function calls. For example, Fig. 1 illustrates the function call relationship among 5 functions: *foo1* calls *foo2* and *foo3*, *foo2* calls *foo4*, and *foo3* calls *foo5*. With functions as nodes and function calls as edges, the relationship among the 5 functions can be represented as a directed network. Note that, Linux is updated almost every 2 months and there have been hundreds of releases until now. In this research, we select the latest version (v3.12.6) as our object. We build up the LKN as a directed complex network and focus on its largest weakly connected component, which consist of 9334 nodes.

In LKN, the in-degree k_{in} of a node indicates the number of functions calling it, while the out-degree k_{out} represents the number of functions it calls. Table 1 shows some basic features of the LKN. To comparatively illustrate the features of the LKN, a randomized network with the same number of nodes and edges is constructed. During the process of randomization, self-connections and duplicated edges are avoided. From Table 1, it is clear to see that the LKN consists of 9334 nodes and 26841 edges and it is a sparse network with $k_{in} = k_{out} = 2.8756$. In Table 1, C is the clustering coefficient, which is used to qualify a network's cluster tendency. It evaluates the extent to which the nodes in the neighborhood of a certain node are connected. Fagiolo [38] extended the clustering coefficient in a directed network as:

Table 1
Comparisons between LKN and a randomized network. n: number of nodes; m: number of edges; $k_{in,max}$: maximum in-degree; $k_{out,max}$: maximum out-degree; $\langle k_{in}\rangle$: average in-degree; $\langle k_{out}\rangle$: average out-degree; C: clustering coefficient; R: reciprocity parameter; E: the efficiency. The data of randomized network are averaged by 10 realizations.

	LKN	Randomized network
n	9334	9334
m	26841	26841
$k_{in,max}$	415	11
$k_{out,max}$	92	11
$\langle k_{in}\rangle$	2.8756	2.8756
$\langle k_{out}\rangle$	2.8756	2.8756
C	0.0253	0.0003
R	−0.0003	−0.0001
E	0.0017	0.1071

$$C = \frac{1}{2n}\sum_{i=1}^{n}\frac{\sum_j\sum_k(a_{ij}+a_{ji})(a_{ik}+a_{ki})(a_{jk}+a_{kj})}{(k_i^{in}+k_i^{out})(k_i^{in}+k_i^{out}-1)-2\sum_j a_{ij}a_{ji}} \quad (1)$$

where k_i^{in} and k_i^{out} are the in-degree and out-degree of node i respectively, and $a_{ij} = 1$ if there exists an edge from i to j, otherwise $a_{ij} = 0$. We can find that the clustering coefficient of LKN is $C = 0.0253$, which is 84 times larger than the value ($C = 0.0003$) of the randomized network. Fig. 2 exhibits the correlation between clustering coefficient C and degrees (both k_{in} and k_{out}), we can find that the clustering coefficients are negatively correlated with k_{in} and k_{out}. It implies that the neighborhoods of the nodes with small degrees tend to cluster together.

In Table 1, R is a reciprocity parameter, which is used to qualify the symmetry of a network [39]. It can be formulated as follows:

$$R = \frac{\sum_{i\neq j}^{n}(a_{ij}-\bar{a})(a_{ji}-\bar{a})}{\sum_{i\neq j}^{n}(a_{ij}-\bar{a})^2} \quad (2)$$

where

$$\bar{a} = \frac{\sum_{i\neq j}^{n}a_{ij}}{n(n-1)} \quad (3)$$

Here, $a_{ij} = 1$ if and only if there exists an edge from i to j, otherwise $a_{ij} = 0$. The maximum of R is 1, implying that each edge in the network has a reverse mate. A small R indicates the high asymmetry of the network. Table 1 shows that the reciprocity parameter of LKN is $R = -0.0003$, which is smaller than the value ($R = -0.0001$) of a randomized network. Actually, LKN is an extremely asymmetric network without edge reversing each other, because there is no mutual function call between two functions in the kernel. As a result, it is hard to find a path back in LKN even though there are some loops. In the table, E evaluates the efficiency of information translation [40]. It is defined as

$$E = \frac{1}{n(n-1)}\sum_{i\neq j}\frac{1}{d_{ij}} \quad (4)$$

[1] Linux is open source software with GNU General Public License which provides users the freedom to run, copy, distribute, study, change and improve the software. Its source codes are available on www.kernel.org.