Analyzing the structure of Java software systems by weighted K-core decomposition

Weifeng Pan, Bing Li, Jing Liu, Yutao Ma, Bo Hu

PII: S0167-739X(17)32094-0
DOI: https://doi.org/10.1016/j.future.2017.09.039
Reference: FUTURE 3695

To appear in: *Future Generation Computer Systems*

Received date : 1 November 2016
Revised date : 27 July 2017
Accepted date : 16 September 2017

Please cite this article as: W. Pan, B. Li, J. Liu, Y. Ma, B. Hu, Analyzing the structure of Java software systems by weighted K-core decomposition, *Future Generation Computer Systems* (2017), https://doi.org/10.1016/j.future.2017.09.039

Analyzing the Structure of Java Software Systems by Weighted *K*-Core Decomposition

Weifeng Pan[a,*], Bing Li[b,c], Jing Liu[d], Yutao Ma[d], Bo Hu[e]

[a]*School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[b]*International School of Software, Wuhan University, Wuhan 430072, China*
[c]*Research Center for Complex Network, Wuhan University, Wuhan 430072, China*
[d]*State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China*
[e]*Kingdee Research, Kingdee International Software Group Co. Ltd., Shenzhen, Guangdong 518057, China*

Abstract

Statistical properties of un-weighted software networks have been extensively studied. However, software networks in their nature should be weighted. Understanding the properties enclosed in the weighted software networks can lead to better software engineering practices. In this paper, we construct a set of weighted software networks from real-world Java software systems and empirically investigate their topological properties by using weighted *k*-core decomposition. First, we investigate the static topological properties of the weighted *k*-core structure, and find that small value of the graph coreness is a property shared by many software systems, the distribution of weighted coreness follows a power law with an exponential cutoff, and weighted coreness and node degree are closely correlated with their spearman correlation coefficients larger than 0.94. Second, we analyze the evolving topological properties of the weighted *k*-core structure, including the graph coreness, size of the main core, and new members and vanishing members of the main core. Empirical results show that the graph coreness will keep relatively stable unless the system undergoes major changes, size of the main core keeps stable in its evolution, and new members or vanishing members of a main core are from or go to the shells very near the corresponding main cores. Finally, we apply the weighted *k*-core decomposition method to identify the key classes, and find that, compared with other nine approaches, our approach performs best in the whole set of subject systems according to the average ranking of the Friedman test. It can identify a majority of classes deemed important. This work could help developers to improve software understanding, propose new metrics for software measurement and evaluate the quality of the system in development.

Keywords: software networks, weighted *k*-core decomposition, program comprehension, static analysis

1. Introduction

Over the past few years, the study of complex networks has gained overwhelming popularity [1, 2, 3, 4]. It provides a unified perspective for studying various complex systems simply by modeling them as a network. Software systems, no matter object-orientation (OO) and structured programming, can be mapped to a network (or graph), also known as software network, where network nodes represent the software entities such as methods/attributes, classes/interfaces, or packages, and network edges (or links), couplings between them [5]. With the software becoming ever larger and complex, the idea of applying complex network theory to model large software and further to interpret their global statistical properties is viable [5, 6]. Great efforts have been made to understand the topological structure of software and many shared physics-like laws of software systems have been revealed such as scale-free [5, 7, 8, 9], small-world [5, 9, 10], and fractal properties [6].

k-core structure [11, 12, 13] is another interesting structural property that are not captured by scale-free, small-world, or other simple topological properties. An in-depth investigation of the *k*-core structures of software networks is very important for deeply understanding the inner characteristics of software systems [12, 13, 14]. Several related studies have been performed [12, 13, 14, 15]. However, one major limitation of these methods is that the software networks they used are unweighted, which does not conform to the reality of a piece of software [9, 16]. Another limitation of the existing methods is that the software systems they analyzed are mainly written in C++ language. Little attention has been paid to the analysis of *k*-core structure of weighted software networks extracted from Java software systems.

The objective of this paper is to explore the characteristics of *k*-core structure in weighted software networks extracted from Java software systems. First, we formally represent the topological structure of Java software at the class level of granularity using a weighted software network, which takes into consideration the coupling frequencies between classes as weights. Second, we introduce the *k*-core decomposition method for weighted complex networks proposed in [17] (hereinafter referred to as W_{k-core}) and use it to calculate the *k*-core structure of the weighted software network. W_{k-core} will partition the weighted software network into a layered structure which will be further measured by amount of relevant properties by statistical parameters. Our approach could potentially uncover some

*Corresponding author. Tel: +86 13588715082
E-mail: wfpan1982@163.com