开源许可证及其检测工具研究

何东杰[1] 宋 昊[2] 王 琪[1] 匡翔宇[2] 刘为怀[1] 蒋丹妮[1]

[1](中国银联电子支付研究院电子商务与电子支付国家工程实验室 上海 201201)

[2](复旦大学计算机科学技术学院 上海 200433)

摘 要 开源许可证的不合理使用为企业留下了许多法律隐患,但主流的检测工具仅仅只能检测开源软件中的许可证名称、数量等,并不能给出风险分析。针对这些问题,对开源许可证进行了广泛研究,提出商业化风险、许可证兼容风险、专利侵权风险、产权归属风险、商标使用风险、服务提供风险等方面的法律风险,设计并实现了开源许可证兼容性和合法性检测系统。该系统能够从许可证版权信息、许可证详细信息、许可证兼容问题、许可证法律风险四个方面给出风险分析,为开源软件的合法使用提供了重要的参考依据。

关键词 开源 许可证 合法性 兼容性

中图分类号 TP3 **文献标识码** A DOI: 10.3969/j.issn.1000-386x.2018.06.005

A STUDY OF OPEN SOURCE LICENSE AND ITS DETECTION SOFTWARE

He Dongjie[1] Song Hao[2] Wang Qi[1] Kuang Xiangyu[2] Liu Weihuai[1] Jiang Danni[1]

[1](*National Engineering Laboratory for Electronic Commerce and Electronic Payment, China UnionPay Electronic Payment Research Institute, Shanghai 201201, China*)

[2](*School of Computer Science, Fudan University, Shanghai 200433, China*)

Abstract The unreasonable use of open source licenses leaves many legal hidden dangers for enterprises, but the mainstream detection tools can only detect the license names, quantities, etc. in open source software and cannot provide risk analysis. In response to these problems, we conducted extensive research on open source licenses and proposed legal risks in terms of commercialization risk, license compatibility risk, patent infringement risk, property ownership risk, trademark use risk, and service provision risk. We designed and implemented an open source license compatibility and legality detection system. The system provided risk analysis from license copyright information, license detailed information, license compatibility issues, license legal risk, and provided an important reference for the legitimate use of open source software.

Keywords Open source Licenses Legality Compatibility

0 引 言

随着开源软件的不断发展和完善,其地位也日益重要,很多中小型公司都使用开源软件,并将其产品提供到开源社区,以求从其他贡献者中获益,从而加速产品的完善、接受和普及。开源许可证伴随着开源软件而产生,开源许可证提供了软件的原作者对于开源软件使用的许可授权,从而使得任何人都能够自由地使用开源软件。

但是,开源许可证的组织也强调对于有开源许可证的软件要谨慎使用,避免产生法律问题。然而一些中小型的公司或者经验不足的工程师往往忽略了这些开源许可证的重要性,对开源软件进行简单包装后便进行商业化使用或销售[1],违反了开源许可证中的条款,为企业带来了法律上的问题。

因此,对现今主流的许可证条款进行研究,然后基于研究结果,给予软件开发者开源许可证中存在的法

收稿日期:2017-10-13。上海市科委项目(16DZ1100200);复旦—银联合作项目"中国银联 2016 年云计算基础数据研究"支持。
何东杰,硕士,主研领域:大数据,云计算,开源软件。宋昊,硕士。王琪,硕士。匡翔宇,硕士。刘为怀,硕士。蒋丹妮,硕士。

律风险及限制和许可证兼容性的提示,让开发者在软件开发之前将上述问题解决,具有重要的应用价值。但是,现今主流的检测工具仅仅只能检测开源软件中的许可证名称、数量等,并不能基于许可证的条款给出许可证的兼容性和法律风险的分析结果[5]。

针对上述问题,本文对现今主流的许可证的条款和许可证的法律效益进行了研究,将不同类型的法律风险和许可证的兼容性问题进行分类分析,基于研究设计并开发了开源软件许可证的检测工具 FINDLICENSE。

FINDLICENSE 能够从许可证兼容、开源发布软件、商业化发布软件、商业化服务提供等不同使用角度能够给出开源软件使用的风险提示,为开源软件的合法使用提供了重要的参考依据,使得企业在使用开源软件之前就能避免法律风险。

1　开源许可证及其检测工具

1.1　开源许可证及其权益

软件许可证是一种具有法律性质的合同或指导,目的在规范受著作权保护的软件的使用或散布行为。通常的授权方式会允许用户来使用单一或多份该软件的副本,因为若无授权而径自使用该软件,将违反著作权法给予该软件开发者的专属保护。效用上来说,软件授权是软件开发者与其用户之间的一份合约,用来保证在匹配授权范围的情况下,用户将不会受到控告。

开源许可证是软件许可证的一种特殊形式,其授权一般是针对软件的开放源代码,而不同于商业软件的运行文件,其允许用户在承认软件原作的著作权的基础上对软件源代码的使用、修改、私用等权益,以保证开源软件能够合法地被广大软件开发者自由使用和共享。

开源许可证的一般格式如图 1 所示。

```
XXX License
Copyright (c) [year] [fullname]
Permission is hereby xxx
The above copyright notice and this permission notice shall be included in xxx
xxx.
……
…
```

图 1　开源许可证的一般格式

开源软件的许可证的种类比较繁多和复杂,常用的有六个: AGPL、GPL、LGPL、BSD、Apache、MIT,表 1 列出了 GITHUB 上排名前十的许可证的使用比例。

表 1　许可证使用情况排名表

排名	许可证	所占百分比
1	MIT	42.84%
2	Other	14.68%
3	GPLv2	11.96%
4	Apache	11.19%
5	GPLv3	8.88%
6	BSD 3-clause	4.53%
7	Unlicense	1.87%
8	BSD 2-clause	1.70%
9	LGPLv3	1.30%
10	AGPLv3	1.05%

在法律上,一般采用知识产权法对软件进行保护,与软件相关的知识产权包括著作权、专利权、商标权、商业秘密权、反不正当竞争权。开源软件许可证的在法律上通常被认为是合同,一般都遵照《合同法》进行判别,即遵照合同本身声明的权益进行判别。

在现今主流的开源许可证中,声明的条款涉及软件权益的主要包括商业用途、分发、修改、专利授权、私用、公开源码、放置许可协议与版权信息、使用网络分发、使用相同协议、声明变更、承担责任、使用商标等 12 项。

1.2　开源许可证使用方法及检测工具

若要在项目中使用某种开源许可证,一般有三种使用方法: 在“LICENSE”文件使用、在源代码注释中使用、在 readme 文件中使用。

使用“LICENSE”文件,需要在项目的根目录下创建一个“LICENSE”的文件,表明整个项目置于此许可证的声明下; 使用注释的形式声明在源代码中,表明仅此代码文件置于此许可证的声明下; 在 readme 文件中放置许可证的全文条款,表明 readme 文件中声明的整个软件置于此许可证的声明下。

因此,若要检测开源许可证,可以根据“LICENSE”文件、源代码注释中、readme 文件中等查看软件许可证,但是这种方法过于浪费人力,而现在网络上有提供许多自动化工具来检测许可证。The Binary Analysis Tool 可用于审计开源软件的内容,其中包括了许可证的内容; FOSSology 可以用于许可证和版权检测; The Open Source License Checker 可以用于检查和分析来自开源包的许可证信息; Spago4Q 是一款免费开源软件质量平台,可以检测出项目中的许可证; Apache Rat 是一个发布的审计工具,专注于软件许可证,告知你使用准确性,以提升许可证使用的效率。

但是,一般的许可证检测工具的结果都是罗列出项目中许可证的种类、数量以及许可证所在的目录信

息,并不能检测出许可证之间的兼容性问题和法律风险问题。

2 开源许可证的法律风险分析

主流开源许可证中,声明的条款主要包括商业用途、分发、修改、专利授权、私用、公开源码、放置许可协议与版权信息、使用网络分发、使用相同协议、声明变更、承担责任、使用商标等 12 项。

根据开源许可证的法律依据和效用,参考不同开源许可证的条款,在使用开源软件时,为避免产生法律问题,主要需要考虑以下几方面的问题:

- 开源要求
- 修改声明
- 专利允许
- 许可证兼容
- 商标使用
- 网络分发

根据以上六方面的问题分析,我们将开源许可证的法律风险分为六个方面:商业化风险、许可证兼容风险、专利侵权风险、产权归属风险、商标使用风险、服务提供风险。

2.1 商业化风险

商业化风险,是指开源许可证要求开源软件的任何衍生作品都要开源发布,因此使用此类许可证下的软件进行商业版本发布都必须开源,这与商业化的利益要求不符,会导致商业化风险。

不同许可证的要求大致分为三类:不强制开源;作为库引用不强制开源;强制开源。表 2 中开源一列列出了常见许可证的分类。其中,强制开源的许可证多是一些较为严格的许可证。如 GPL、LGPL、EPL 和 MPL 等,发布软件时必须提供源代码;作为库引用不强制开源的许可证,仅仅只有 LGPL 的不同版本,在商业化上有实际价值,但只能作为库来引用,不能直接使用任何源代码;不强制开源的许可证,包含了 BSD、Apache、MIT 等常见许可证。这类许可证是比较宽松的许可证,在其他方面往往也限制较少,对商业使用来说,这些许可证是比较友好的。

表 2 常见开源许可证条款声明情况表

许可证	开源	专利	商标	声明变更	网络分发开源	使用相同协议
Academic Free License v3.0		√	×	√		
GNU Affero General Public License v3.0	√	√		√	√	√
Apache License 2.0		√	×	√		
Artistic License 2.0		√	×	√		
BSD 2-clause "Simplified" License						
BSD 3-clause Clear License		×				
BSD 3-clause "New" or "Revised" License						
Creative Commons Attribution 4.0		×	×	√		
Creative Commons Attribution Share Alike 4.0		×	×	√		√
Creative Commons Zero v1.0 Universal		×	×			
Eclipse Public License 1.0	√	√				√
European Union Public License 1.1	√	√	×	√	√	√
GNU General Public License v2.0	√			√		√
GNU General Public License v3.0	√	√		√		√
ISC License						
GNU Lesser General Public License v2.1	√			√		√
GNU Lesser General Public License v3.0	√	√		√		√
LaTeX Project Public License v1.3c	√			√		
MIT License						
Mozilla Public License 2.0	√	√	×			√
Microsoft Public License		√	×			
Microsoft Reciprocal License	√	√	×			√
SIL Open Font License 1.1						√
Open Software License 3.0	√	√	×	√	√	√
The Unlicense						
Do What The F* ck You Want To Public License						
zlib License				√		

备注 "√"表示许可证明确要求,"×"表示协议明确禁止,留空表示对于该条款许可证未声明

对于强制开源的许可证,使用时要慎重,因为违反开源许可证被起诉已经有许多成功的判例。如 2007 年 2 月,Skype 被起诉违反了 GPLv2 许可协议,自由软件基金会(FSF)称,Skype 是基于 Linux 的 WSKP100 WiFi VoIP 电话,而 Skype 并未基于 GPL 协议提供其 Linux 产品的源代码。德国一审法庭调查后认定事实

确凿,宣判 Skype 违反了协议规定。

2.2　许可证兼容风险

许可证兼容风险,是指开源软件引用了其他多个开源软件,而这些引用的开源软件可能是不同的许可证,此时开源软件本身的许可证和引用的开源软件的许可证可能会存在兼容性问题。这其中包含了两方面的问题: 第一,开源软件本身的许可证与引用的开源软件的许可证相同权益的条款是否有冲突以及最终的许可证应该如何放置; 第二,开源软件本身的许可证是否允许引用使用了其他许可证下软件。

开源软件协议从使用限制强弱上来看,可以分为三大类: 放任型、弱保护型、强保护型。一般来说,强限制协议可以向下兼容弱限制协议,这意味着软件最终许可证取决于强限制协议。但限制条件完全对立的两个协议则无法兼容。图 2 说明了常见开源软件协议之间的条款兼容性情况。

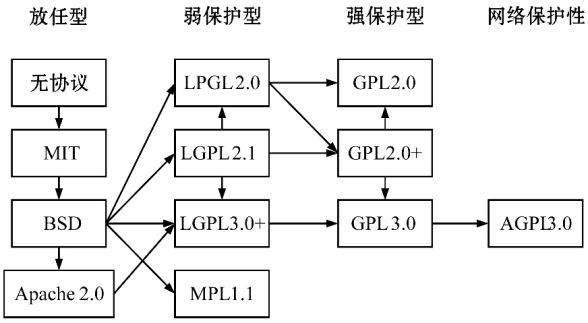

图 2　常见开源软件协议条款兼容性指示图

箭头从 A 框到 B 框代表,A 框和 B 框中的协议是兼容的,即两种开源软件可以组合使用,且最终 License 取决于 B 框中协议。而如果两个框之间没有单向的箭头贯通,即意味着两个框中的协议不兼容,即两种开源软件不可以组合使用。

举例说明,如 MIT -> BSD -> Apache -> LGPLv3 -> GPLv3.0 是单向通路,通路上的任意两个及以上的开源软件都可组合使用,软件最终 License 取决于通路上箭头最末端 GPLv3.0 协议。MPL <- BSD -> Apache是一个双向链路,链路两端的 MPL 和 Apache 协议是不兼容的,无法组合使用。

对于开源软件本身的许可证是否允许引用其他许可证下的软件可以由表 2 中使用相同协议条款判别。其中,有相同许可证要求的许可证在分发软件时,必须使用相同的许可证发布。无相同许可证要求且不强制开源的许可证,可以设置任何类型的许可证,且不会存在法律问题。

2.3　专利侵权风险

专利侵权风险,是指开源软件的衍生作品申请了专利,而许可证条款不承认衍生作品申请专利或者衍生作品申请的专利归原作者所有,这种情况也将产生严重的法律问题。

常见的开源许可证对于专利的要求如表 2 中的专利一列所示。由表 2 可知,许可证的专利授权有三种情况: 明确专利授权,许可证允许贡献者进行专利申请; 不提供专利授权,许可中明确声明它不授予贡献者专利授权; 未明确,许可证中没有提到专利许可。

明确授予专利许可权的许可证包括表 2 专利一列中列出的 12 种许可证,这些许可证可以申请专利,并且在专利法保护之下。但是,有一部分许可证同样要求开源,这部分许可证申请专利也无太大作用。四个许可证明确表示不提供专利授权,对于不承认专利的许可证不可申请衍生作品的专利,不然会产生侵权问题,所以应尽量避免选择此类许可证; 未明确专利授予问题的许可证,包括 GPL2.0、LGPL2.1 和 MIT 许可证等,在合同法范围内未声明权益,可以认为承认衍生作品的专利权,所以可以对衍生作品申请专利。值得注意的是,对于衍生作品专利的申请,只能针对自己修改的部分,不能包含原作品部分。另外,部分开源许可证要求衍生作品中对原作品的任何修改或添加都要进行声明,否则不承认衍生作品的专利权。

著名的 Robert 一案可以看出这一条款的重要性。Robert Jacobsen 研发出了名为“解码博”的软件,放置在 SourceForge 供其他人无偿下载。KAM 开发了名为“解码指挥官”的软件。在开发过程中,下载了 Robert 的软件套件,并将其中的定义文档和其他部分程序纳入到了“解码指挥官”的软件之中,向美国政府申请了一项专利。之后由于纠纷,Robert 一纸诉状将 KAM 送上了法庭。最终,上诉到美国联邦法院,判决 KAM 侵犯版权,理由是“开源协议”是一种著作权协议,违反协议就是侵权行为。

2.4　版权归属风险

版权归属风险,是指开源许可证明确声明了基于原作品的衍生作品的修改部分必须进行明确声明,才能承认其具有对修改部分的版权,而如果未对此部分进行声明便进行了商业使用,会产生版权归属不明的情况,由此可能会产生法律问题。

表 2 修改声明一列列出了常见许可证的修改声明要求。为防止产生法律问题,对于使用了明确要求声明修改的开源许可证的软件,在进行商业化时,必须对修改部分进行声明,以明确版权归属。

“绿坝-花季护航”用于不良图像过滤的主要文件 cximage.dll、CImage.dll、xcore.dll 和 Xcv.dll 均来自

OpenCV。OpenCV 采用的是 BSD 许可证，当商业软件使用 BSD 许可证的开源程序时，需要在软件版权信息中加上 BSD 许可证声明，以及自己变更部分的声明，绿坝并未在其软件中加入这一声明。2010 年初，美国加州 Cybersitter 软件公司向当地法院提起诉讼，称“绿坝-花季护航”软件抄袭了该公司的近三千行代码，要求索赔 22 亿美元。

2.5 商标使用风险

商标使用风险，是指如果开源软件的衍生作品使用并申请了商标权，但原开源软件的许可证条款声明了不承认商标权，这将会导致衍生产品的商标不合法，不具有法律担保性。

常见的开源许可证对于商标的使用条款有两种：一种是明确不承认使用商标，另一种是没有条款针对商标进行声明。常见许可证对于商标使用的声明情况如表 2 商标一列所示。

其中，明确不提供商标使用的许可证有 11 种。但是，值得注意的是，这些不承认商标权的许可证，有 6 种是不要求使用相同许可证的。因此，只要在衍生作品中不使用原来的许可证，就可以使用商标，其他的不承认商标权的许可证则不能使用商标。对于许可证条款中没有明确商标使用的，有 16 种，这些许可证下的衍生作品则可以使用商标。

2.6 服务提供风险

由于云计算的兴起，使得软件可以不通过传统的互联网或光盘 release 分发，而以“不分发软件，为客户在云上提供网络服务”的模式使用软件，这种服务模式虽不受强制开源的要求，但是受网络分发条款限制。

服务提供风险，是指软件在云上提供网络服务的软件是某些开源许可证下的软件的衍生或者原生作品，但是这些开源许可证不允许衍生作品闭源进行云服务的提供，因此，此种使用方式会产生法律上的风险。

常见的开源许可证对于网络分发的条款有两种：一种是明确声明网络分发必须开源，另一种是没有条款针对网络分发进行声明。常见许可证对于网络分发的声明情况如表 2 网络分发开源一列所示。其中，明确声明网络分发必须开源的只有三种，这三种许可证即使在商业中以服务形式也无法使用，而没有声明网络分发的许可证，可以闭源并以云服务的形式供商业客户使用。

因此，对于有开源许可证的软件，首先阅读许可协议与版权信息条款，按照许可证要求将开源许可证放置于软件之中；然后，参照表 2 和图 1，分别从开源要求、修改声明、许可证兼容、专利允许、商标使用、网络分发六个方面分别去衡量商业化风险、许可证兼容风险、专利侵权风险、产权归属风险、商标使用风险、服务提供风险，在符合许可证条款的条件下，使衍生作品避免法律纠纷和许可证兼容问题。

3 FINDLICENSE 工具

基于上述分析结果，我们的工具 FINDLICENSE 能够提供以下功能：查找开源软件中包含的所有许可证和权益信息、指出开源软件中包含的不同许可证兼容问题、给出开源发布软件的风险提示、给出商业化发布软件风险提示、给出作为服务提供使用软件风险提示、给出其他权益风险提示、自定义开源许可证添加、自定义许可证风险添加等。

3.1 设计概述

图 3 展示了 FINDLICENSE 工具的整体架构。它使用 python 语言实现，使用 FLASK 作为后端，HTML 作为前台，用户只需将待检测软件的源代码压缩包，在 WEB 界面上传至后台或者传入源代码托管仓库地址进行扫描，因此是与平台无关的。它包含了一个输入：待检测的开源软件源代码压缩包文件或者开源软件源代码所在的仓库地址。包含一项输出：开源软件许可证的分析报告，分析报告现在以网页 HTML 的形式给出，报告包含了开源软件许可证兼容问题、开源发布软件风险、商业化发布软件风险、作为服务提供使用软件风险、其他权益风险以及文件中包含的许可证详细的官方信息。

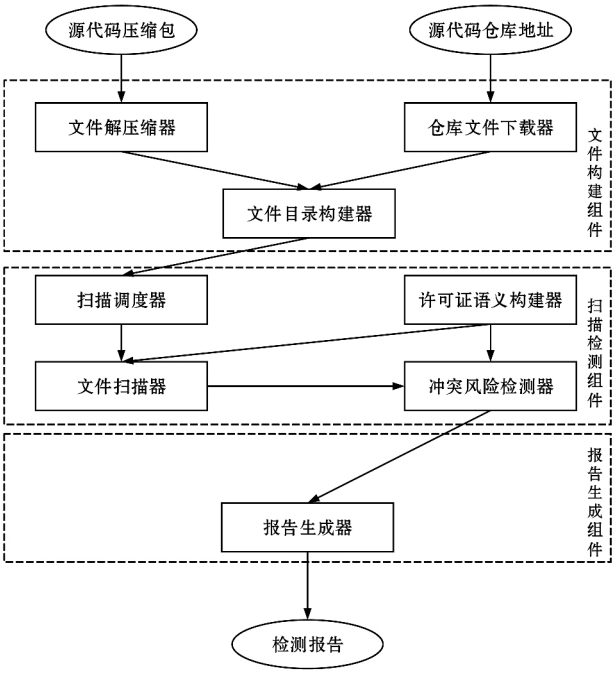

图 3 FINDLICENSE 工具架构图

我们的 FINDLICENSE 遵循模块化设计的思路,工具包含了三个主要模块: 文件构建组件、扫描检测组件、报告生成组件。文件构建组件负责将上传的源代码压缩包解压到指定位置或者根据仓库地址将其下载到指定位置,然后分析文件结构和文件信息,构建文件扫描的树状图。扫描检测组件负责根据库中许可证构建扫描的语义表,然后根据语义表进行按照文件构建组件给出的结构图进行扫描,匹配许可证和版权信息,再根据定义的许可证冲突和风险,对扫描结果进行匹配。报告生成器根据扫描检测组件扫描检测的结果,结合库中许可证的信息,以 HTML 格式的文件将扫描结果输出,并返回给用户。

许可证的规范使用方法是在“LICENSE”文件使用、在源代码注释中使用、在 readme 文件中使用,但是考虑到不规范使用的情况,FINDLICENSE 是进行全文本扫描,可以扫描出开源软件源代码中多类型应用许可证。

3.2　实　现

文件构建组件。包含文件解压缩器、仓库文件下载器、文件目录构建器。其中,文件解压缩器实现对 ZIP、RAR、GZ、BZ2 等格式的文件进行解压缩。仓库文件下载器实现从远程仓库到本地的复制,目前仅支持 GIT 的仓库。文件目录构建器对于整个文件夹进行遍历,然后构建文件夹的树形结构,以 JSON 数据形式保存。

扫描检测组件。包含了扫描调度器、文件扫描器、许可证语义构建器、冲突及风险检测器。文件扫描器根据许可证语义构建器提供的关键字,对扫描调度器指定的文件进行语义匹配,匹配到对应的关键字之后,将关键字的位置及许可证或者版权的标示保存到结果中。实际扫描中,我们的问题相当于给定的长度为 n 的文本,和模式集合 $P\{p_1,p_2,\cdots,p_m\}$,找到文本中的所有匹配模式的位置,而 AC 算法可以在 $O(n)$ 时间复杂度内找到文本中的所有目标模式,而与模式集合的规模 m 无关,因此我们使用 AC 算法进行实际的文件扫描。

AC 算法的算法思想如下: 对于模式集合 $P\{$he,she,his,hers$\}$,模式 s(he) 的非前缀子串 he,实际上却是模式(he),(he)rs 的前缀。如果目标串 target $[i\cdots i+2]$ 与模式 she 匹配,同时也意味着 target $[i+1\cdots i+2]$ 与 he、hers 这两个模式的头两个字符匹配,所以此时对于 target $[i+3]$,我们不需要从 target $[i]$ 进行比较,而直接将 target $[i+3]$ 与 he、hers 两个模式的第 3 个字符比较,然后直接向后继续执行匹配操作。

根据上述的 AC 算法,我们文件扫描器对每个文件中提供的关键字进行语义匹配,匹配到对应的关键字之后,将关键字的位置及许可证或者版权的标示保存到结果中,然后继续扫描,直到整个文件扫描结束,最后将扫描的结果返回给给扫描调度器。

扫描调度器根据文件目录构建器给出的文件结构,按照深度优先的策略,调度文件扫描器进行扫描,并将文件名称和扫描的结果以字典的形式保存在字典中,供冲突及风险检测器使用。在实际调用中,单线程的调度扫描速度过慢,因此,为加快扫描速度,我们借鉴了分布式系统中 MapReduce 的思路,将扫描阶段进行了分解,具体执行流程如图 4 所示。

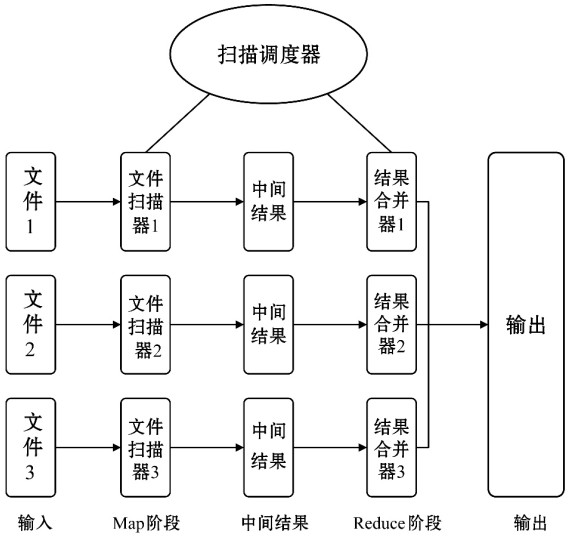

图 4　文件扫描调度执行流程图

扫描开始时,我们扫描调度器会启动多个文件扫描器的进程,按照图 4 中由最底层叶子节点开始,向上分别对不同的文件进行扫描,待所有文件扫描器的进程将所有文件扫描结束后,产生中间结果; 尔后,扫描调度器会启动多个结果合并器的进程,以文件路径作为键,扫描结果作为值,将扫描结果进行合并,得到输出。

许可证语义构建器,首先在许可证目录中将所有许可证逐个按照内容进行切分。切分完成后,将许可证名称和对应的切分内容按照键和内容的形式构建字典,待所有许可证都切分并且构建完毕之后,将字典返回给文件扫描器供实际扫描过程中使用。

冲突及风险检测器。首先根据我们的研究结果构建了许可证风险库和不兼容库,实际使用中,用户可以对这个库进行自定义的添加、修改、删除等操作。风险库包含了商业化风险、专利侵权风险、产权归属风险、商标使用风险、服务提供风险,风险库中包含了此种风险的名称和其对应的许可证的键。兼容库中包含了所有强许可证(即衍生作品必须使用本许可证并且要求

开源) 的键和兼容风险名称。冲突及风险检测器根据扫描调度器提供的字典,在库中对风险和兼容性规则进行匹配,匹配到风险或者兼容性问题后便将许可证的键和风险类型保存在字典中,待所有扫描调度器提供的字典扫描完毕后,将结果的返回给报告生成器。

报告生成组件仅包含报告生成器一个组件,报告生成器使用了 FLASK,将扫描的结果填充到 HTML 模板的指定位置,并将最终生成的报告以 HTML 的形式返回给客户。

4 评 估

我们对于 github 前 20 的项目进行了测试。测试分别使用 FOSSology[6]、The Open Source License Checker、Spago4Q、Apache Rat 以及我们自主开发的 FINDLICENSE,我们从三方面对我们的工具进行评估:第一,扫描涵盖结果的全面性,即许可证的检测工具为用户提供的结果涵盖许可证版权信息、许可证详细信息、许可证兼容问题、许可证法律风险四个方面的程度。第二,对于整个开源软件许可证扫描的准确性,即许可证检测工具是否能够将所有的许可证都检测出来,比率如何。第三,扫描的时间,即对于同一软件,许可证扫描工具从用户提交数据到生成结果所需要的时间。测试中,我们使用的系统环境及软件版本如表 3 所示。

表 3 测试系统环境表

名称	版本
处理器	Intel Core i7 2600 3.4 GHz
硬盘	SATA3 20 GB 5400 转
内存	DDR2 1 GB
操作系统	Ubuntu 14.04 64 位
Jdk	8u131
Python	2.7
Mysql	server-5.7
Tomcat	8.5.14
Apache Httpd	2.4.25
GCC	6.3
FOSSology	2.6
Spago4Q	2.4
The Open Source License Checker	3.0
Apache Rat	0.12

4.1 扫描涵盖结果的全面性

我们从许可证版权信息、许可证详细信息、许可证兼容问题、许可证法律风险四个方面去衡量许可证涵盖结果的全面性,每个方面各占 25%,不同软件许可证检测工具结果涵盖程度如图 5 所示。

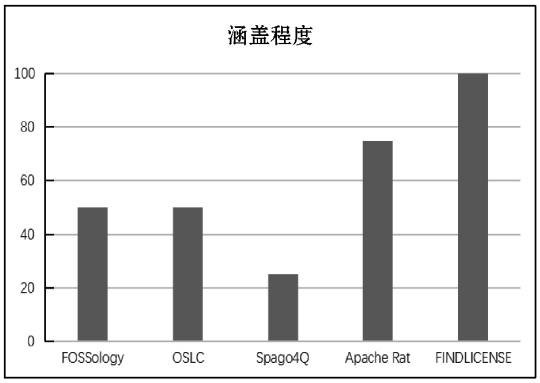

图 5 不同软件许可证检测工具结果涵盖程度

图5 中,FOSSology 和 The Open Source License Checker (OSLC) 只能够检测许可证版权信息和许可证详细信息,没有许可证兼容问题和许可证法律风险这两个方面,所以只有 50%; Spago4Q 仅能够提供许可证版权信息,所以只有 25%; Apache Rat 能够提供许可证版权信息和许可证详细信息,但对于许可证兼容问题,只能提供强许可证使用正确性提醒,所以总值 75%; 而我们的 FINDLICENSE 工具包含了许可证版权信息、许可证详细信息、许可证兼容问题、许可证法律风险四个方面,所以涵盖程度为 100%,也是五个工具中功能和涵盖程度最强大的。

4.2 许可证扫描的准确性

许可证扫描的准确性,我们对于 gitbub 上排名前 20 的项目源代码分别使用 FOSSology、The Open Source License Checker、Spago4Q、Apache Rat 以及我们自主开发的 FINDLICENSE 进行扫描,扫描结果的准确性如图 6 所示。

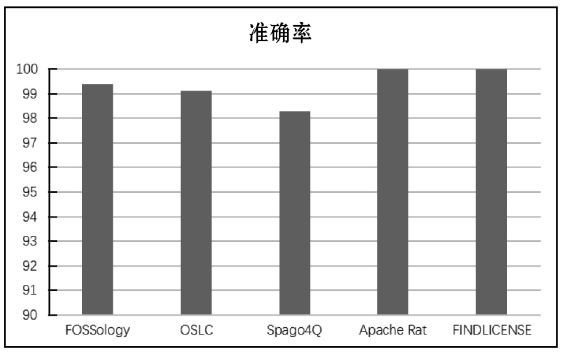

图 6 不同软件许可证检测工具检测准确率

检测中,我们都没有自定义用户库,都是使用软件默认库进行扫描。从图 3 可以看出,Apache Rat 以及

我们自主开发的 FINDLICENSE 扫描的准确率最高，能够达到 100%，FOSSology 和 The Open Source License Checker 能够达到 99% 以上，Spago4Q 能够达到 98% 以上。分析其数据差异的原因，由于我们自主研发的 FINDLICENSE 和 Apache Rat 都不仅针对库中的许可证进行扫描，并且匹配 LICENSE、COPYLEFT、COPYRIGHT 等敏感字信息，所以扫描的准确率很高。而 FOSSology 和 The Open Source License Checker 的默认库涵盖程度很高，所以扫描的准确率也能够达到 99% 以上，但是对于最新的小众的许可证扫描就不尽理想。Spago4Q 的默认库相对来说涵盖程度略低，达到 98% 左右，同样对于最新的小众的许可证扫描不尽理想。

4.3　扫描的时间

许可证扫描的时间，我们分别使用了三种不同大小的开源软件: 大型软件、中型软件、小型软件。大型软件我们使用了如 MYSQL 代码在 GB 级的，中型软件我们使用如 bootstrap 在 MB 级别的，小型软件我们使用如 jquery 在 KB 级别的分别进行检测，检测结果如图 7 所示。

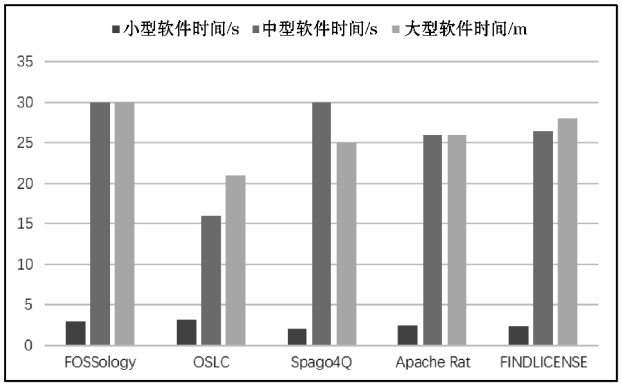

图 7　不同软件许可证检测工具扫描时间图

从图 7 我们可以看出，针对小型软件的扫描速度来看，五种软件相差不大，最快的是 Spago4Q，最慢的是 OSLC，所有软件平均时间在 3 s 左右。中型软件扫描最快的是 OSLC，最慢的是 Spago4Q 和 FOSSology，Apache Rat、FINDLICENSE 基本相同，居于中间位置，所有软件平均时间是 23 s 左右。而对于大型软件来说，差别相对较大，最快的 OSLC，时间大约在 2 min，而最慢的是 FOSSology，时间大约在 30 min，所有软件平均时间在 27 min 左右。

分析其原因，我们总计的扫描时间 = 软件启动时间 + 解压缩时间 + 实际扫描时间。对于小型软件来说，OSLC 因为是 java 语言开发的，程序启动相对较慢，所以软件启动时间很长，而小型软件的扫描时间本来就短，所以 OSLC 最慢，而 Spago4Q 最快。对于中型软件来说，因为扫描平均时间在 23 s 左右，此时软件启动时间所占比例就很小，主要是解压缩时间和实际扫描时间。而中型软件文件数量不大，所以解压缩时间也相对较短，软件的总扫描时间取决于实际扫描时间大小，而 OSLC、Apache Rat、FINDLICENSE 的软件许可证库是从文件中直接读取，而 Spago4Q 和 FOSSology 是从第三方数据库工具中读取，所以相对来说，OSLC、Apache Rat、FINDLICENSE 快于 Spago4Q 和 FOSSology。对于大型软件来说，有三点原因: 第一，因为 OSLC 要用户手动将文件先解压才能扫描，而大型软件文件数量很庞大，解压缩耗费时间巨大，而 OSLC 省去了解压缩时间，所以最快。第二，扫描要基于许可证的库进行匹配，而 FOSSology 和我们自己的工具 FINDLICENSE 库相对较全，但是扫描起来也相对耗时，而 OSLC、Spago4Q、Apache Rat 三者库较少，所以扫描速度相对较快。第三，我们的工具 FINDLICENSE、Spago4Q、FOSSology 为方便用户使用，都是以 B/S 的形式为用户提供服务，所以扫描时需要调用工具，相对直接扫描的工具，也会相对耗时。

综合上述分析，我们的工具 FINDLICENSE 在扫描涵盖结果的全面性、许可证扫描的准确性方面都居于第一位，虽然扫描的速度居于中间位置，但许可证的检测对于实时性要求不高，所以也处于可以接受的程度，未来可以做进一步优化。同时，我们的工具可以以服务的形式部署在云服务器上，方便用户使用。

5　结　语

本文对开源许可证进行了广泛的研究，并针对许可证的兼容问题和法律风险，提出了商业化风险、许可证兼容风险、专利侵权风险、产权归属风险、商标使用风险、服务提供风险等方面的风险，为开源许可证风险检测提供了重要的理论依据。然后，根据这个理论，设计并实现了开源许可证兼容性和合法性检测工具 FINDLICENSE。该工具能够从许可证版权信息、许可证详细信息、许可证兼容问题、许可证法律风险四个方面给出风险分析，使得软件开发者在开发软件之前就能知晓开源许可证中存在的法律风险及限制和许可证兼容性的问题，填补了开源许可证检测工具在此方面的空白，具有重要的应用价值。之后，我们从扫描涵盖结果的全面性、许可证扫描的准确性、扫描的速度三方面对我们开发的工具和其他开源许可证检测工具进行了评估，测试并证明了我们的工具 FINDLICENSE 在扫描涵盖结果的全面性、许可证扫描的准确性方面都居于第一位，扫描速度居于中间位置，继而证明了我们的工具的实用价值。

(下转第 53 页)

参 考 文 献

[1] 周颖伟,庄达民,吴旭,等. 显示界面字符编码工效设计与分析[J]. 北京航空航天大学学报,2013,39(6):761-765.

[2] 张磊,庄达民. 飞机界面设计颜色匹配性[J]. 北京航空航天大学学报,2009,35(8):1001-1004.

[3] 张磊,庄达民. 人机显示界面中的文字和位置编码[J]. 北京航空航天大学学报,2011,37(2):185-188.

[4] 吴志周,贾俊飞. 驾驶分心行为及应对策略研究综述[J]. 交通信息与安全,2011,29(5):5-9.

[5] Stelzer E M, Wickens C D. Pilots strategically compensate for display enlargements in surveillance and flight control tasks[J]. Human Factors, 2006, 48(1):166-181.

[6] Wickens C D, Alexander A L, Ambinder M S, et al. The role of highlighting in visual search through maps.[J]. Spat Vis, 2004, 17(4-5):373-388.

[7] 许志,唐硕. 装配式实时飞行视景仿真平台研究[J]. 计算机工程与设计,2005,26(12):3298-3300.

[8] 泮斌峰,唐硕. 可装配式飞行视景仿真系统的设计与实现[J]. 系统仿真学报,2007,19(4):768-771.

[9] 汪成为,高文. 王行仁,等. 虚拟现实技术理论与实现及应用[M]. 北京: 清华大学出版社,1996.

[10] 孙显营,熊坚. 车辆驾驶模拟器的发展综述[J]. 交通科技,2006(6):48-50.

[11] 于辉,赵经成. GL Studio 虚拟仪表技术应用与系统开发[M]. 北京: 国防工业出版社,2010.

[12] 谈卫,孙有朝. 面向显示界面工效研究的飞机座舱仿真系统[J]. 计算机系统应用,2016,25(8):41-47.

[13] 刘群,张燕军,李竹峰,等. 基于 MFC 对话框和 Vega Prime 的沉浸式跑步视景仿真[J]. 农业装备技术,2017,43(3):39-42.

[14] 万明,南建国. Vega Prime 视景仿真开发技术[M]. 北京: 国防工业出版社,2015.

[15] DiSTL. GL Studio 3.1 Users Manual[Z]. 2005.

[16] 孟晓梅,刘文庆. MultiGen Creator 教程[M]. 北京: 国防工业出版社,2005.

[17] 邵晓东,陈天鸿. Creator 建模艺术[M]. 西安电子科技大学出版社,2014.

[18] 徐恩,李学军,邹红霞,等. 基于 Creator/VP 的三维虚拟环境建模[J]. 系统仿真学报,2009,21(10):121-123.

[19] 刘卫东. 可视化与视景仿真技术[M]. 西安: 西北工业大学出版社,2012.

[20] 王孝平,董秀成,郑海. Vega Prime 实时三维虚拟现实开发技术[M]. 成都: 西南交通大学出版社,2012.

[21] 刘长征. Visual C++串口通信及测控应用实例详解[M]. 北京: 电子工业出版社,2013.

[22] 侯晓琴. Visual C++2005[M]. 北京: 清华出版社,2013.

[23] 梁森,王侃夫. 自动检测与转换技术[M]. 北京: 机械工业出版社,2012.

(上接第 35 页)

随着开源软件的不断发展,其地位必将超越商业软件发挥越来越重要的价值和作用,对计算机发展也将会产生深远的影响,希望我们的研究和工具能够为中国开源软件的发展尽绵薄之力。

参 考 文 献

[1] Kapitsaki G M, Charalambous G. Find your Open Source License Now![C]//2016 23rd Asia-Pacific Software Engineering Conference (APSEC), New Zealand, 2016.

[2] Engelfriet A. Choosing an Open Source License[J]. IEEE Software, 2009, 27(1):48-49.

[3] Singh P V, Phelps C. Networks, Social Influence, and the Choice Among Competing Innovations: Insights from Open Source Software Licenses[J]. Social Science Electronic Publishing, 2009, 24(3):539-560.

[4] Shani G, Gunawardana A. Evaluating Recommendation Systems[M]//Recommender Systems Handbook. 2011:257-297.

[5] Yin R K. Case study research: Design and methods[M]. New York: Sage publications, 2013.

[6] Gobeille R. The FOSSology project[C]//Proceedings of the 2008 International Working Conference on Mining Software Repositories, MSR 2008 (Co-located with ICSE), Leipzig, Germany, May 10-11, 2008.

[7] German D M, Manabe Y, Inoue K. Proceedings of the IEEE/ACM international conference on Automated software engineering, 2010[C]. Antwerp, 2010.

[8] Kapitsaki G M, Kramer F. Open Source License Violation Check for SPDX Files[C]//International Conference on Software Reuse. Software Reuse for Dynamic Systems in the Cloud and Beyond. Springer, Cham, 2015: 90-105.

[9] Obrenovic Z, Gasevic D. Open source software: all you do is put it together[J]. IEEE Software, 2007, 24(5):86-95.

[10] Wu Y, Manabe Y, Kanda T, et al. A Method to Detect License Inconsistencies in Large-Scale Open Source Projects[C]//Proceedings of the 12th Working Conference on Mining Software Repositories, MSR'15. 2015: 324-333.

[11] Hammouda I, Mikkonen T, Oksanen V, et al. Open source legality patterns: architectural design decisions motivated by legal concerns[C]//International Academic Mindtrek Conference: Envisioning Future Media Environments. ACM, 2010: 207-214.

[12] Bhattacharya J, Suman S. Analysis of popular open source licenses and their applicability to e-governance[C]//International Conference on Theory and Practice of Electronic Governance, Icegov 2007, Macao, China, December. DBLP, 2007: 254-257.

[13] Marti D. Reviews: Open source licensing: software freedom and intellectual property law[J]. Linux Journal, 2005, 129: 19.