

Web 服务计算组合流程 QoS 验证

开金字^{1,2} 缪淮扣^{1,2} 高洪皓³

(上海大学计算机工程与科学学院 上海 200444)¹ (上海市计算机软件评测重点实验室 上海 201114)²
(上海大学计算中心 上海 200444)³

摘 要 在满足功能需求的前提下,Web 服务能否赢得市场,主要取决于其服务质量(Quality of Services, QoS)。采用概率模型检验的方法判断 Web 服务计算组合流程的服务质量是否满足系统质量需求。依据用户访问 Web 服务产生的访问日志,采用聚类的方法,构建面向用户群组的 Web 服务流程的使用行为 QoS 模型。采用 QoS 扩展的状态图描述 Web 服务流程的 QoS 需求。然后,利用概率模型验证工具 PRISM 判断 Web 服务流程的 QoS 是否满足系统质量需求。验证面向用户群组的 Web 服务流程的使用行为质量模型是否满足系统质量需求,为服务流程结构的调整提供了依据。

关键词 Web 服务质量需求,面向用户群组的服务流程 QoS 模型,概率模型检验

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.027

Verification QoS of Web Services Compositional Processes

KAI Jin-yu^{1,2} MIAO Huai-kou^{1,2} GAO Hong-hao³

(School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China)¹

(Shanghai Key Laboratory of Computer Software Evaluating & Testing, Shanghai 201114, China)²

(Computing Center, Shanghai University, Shanghai 200444, China)³

Abstract Whether the Web services can win the market, on the premise of meeting the functional requirements, quality of service (QoS) becomes an important factor in judging its advantage. This paper adopted the technique of probability model checking to evaluate the QoS of the Web services computing combination process. In this paper, starting from the access log of user accessing a Web service, using the method of clustering, we constructed a user group-oriented QoS using model. As for the QoS requirement, we used the extended QoS state diagram to describe. And then, we used the simulation verification method in the tool of probabilistic model checking—PRISM to judge whether the behavior of the Web services computing combinational process satisfies its QoS requirement.

Keywords Web service quality requirements, User group-oriented QoS model of Web service compositional processes, Probabilistic model checking

1 引言

在软件工程中,需求分析的目的是“解决正确的问题”,而体系结构设计的目的是“正确地解决问题”。在传统意义上,两者都是发生在软件设计阶段,而在软件产品部署以后,对问题或解决方案的改变都需要一个迭代的、耗时的开发过程。然而,软件系统正越来越多地需要对变化的环境和变化的需求做出快速的响应,即动态地对运行时体系结构做出合适的调整。

现有的自适应方法大多都是被动式的自适应调整方法,即当软件系统出现问题以后才进行自适应调整。然而,对于一些系统,出现问题后再进行自适应调整可能为时已晚,一种

更理想的方法是主动式的自适应^[1]。

因此,在问题真正产生影响之前,通过定量模型检查^[2]技术对 Web 服务计算组合系统进行预测和分析,对 Web 服务系统进行主动式自适应调整具有很重要的指导意义^[3]。

本文采用定量模型检查技术对 Web 服务计算组合系统的服务质量进行预测和分析。

2 相关研究

针对 Web 服务计算组合流程 QoS 问题,已有一些研究结果。

关于 Web 服务流程中原子 Web 服务的 QoS 指标选择、量化、统一化及质量指标获取,文献[4]从时间、成本和可靠性

到稿日期:2015-02-11 返修日期:2015-03-25 本文受国家自然科学基金项目(60970007,61170044,61572306,61502294),上海市自然科学基金项目(15ZR1415200),上海高校青年教师培训资助计划(ZZSD13008)资助。

开金字(1976—),女,博士生,讲师,CCF 会员,主要研究方向为 Web 应用、服务计算、需求分析、软件形式化验证,E-mail:aykijy@shu.edu.cn; 缪淮扣(1953—),男,教授,博士生导师,CCF 高级会员,主要研究方向为软件工程、软件形式化方法;高洪皓(1985—),男,博士,讲师,CCF 会员,主要研究方向为模型检验、Web 应用、服务计算。

3个方面描述了 workflow 中原子任务的 QoS 属性。文献[5]在衡量 Web 服务质量时选用了 5 项指标,分别是 Web 服务 w_s 的执行代价、任务 ti 在 Web 服务 w_s 上的执行时间、Web 服务 w_s 的负载、Web 服务 w_s 的可靠性和 Web 服务 w_s 的信誉度。文献[6]将 Web 服务的 QoS 定义为响应时间、服务费用、可靠性和可用性,并将原子 Web 服务的质量指标映射为一系列的随机变量,利用随机变量的数学期望和方差来度量各指标取值,并依据各项指标对 Web 服务产生的正负影响对这些指标值进行归一化处理,然后依据 Web 服务计算组合流程的控制结构计算 Web 服务计算组合流程的 QoS 指标值。

关于 Web 服务计算组合流程中原子服务的 QoS 指标值的获取和工具也有一些研究成果。文献[5]提出了一种扩充了查询代理和 Web 服务 QoS 管理器的 Web 服务 QoS 管理体系结构,查询代理响应用户服务请求,反馈结果是一类功能属性相同、非功能属性有差异的 Web 服务,Web 服务 QoS 管理器用于对 Web 服务 QoS 的各项指标进行监控获取。具体而言,是通过对任务 ti 和 Web 服务 w_s 之间的通信数据进行监测,并记录与 w_s 的 QoS 有关的信息。对 Web 服务器访问日志进行统计和分析,获取 QoS 指标值。

这些方法都是基于功能或非功能性需求驱动的,与传统的软件相比,面向服务的软件系统已经逐渐从以往关注软件系统的功能与性能,转向更多地关注软件服务系统代表的业务带来的价值。在电子商务市场如此自由开放的环境下,如何以 Web 服务代表电子商务的业务活动的价值为导向,通过对 Web 服务系统资源的调整及流程的优化实现 Web 服务业务价值最大化的目标,成为电子商务和软件工程共同关注的焦点。因此,本文采用基于价值的软件工程和基于价值的需求工程的思想,提出了一种价值作为衡量 QoS 的指标。

关于 Web 服务流程行为模型构建,Ricca 等开发了网站分析工具 ReWeb^[7],其可以通过对网站的页面与链接进行分析,生成 UML 表示页面导航、对象状态图。文献[8]中 Lucca 和 Tramontana 等设计了一种用于 Web 应用进行逆向工程的工具 WARE,从粗粒度和细粒度两个层面对 Web 应用进行分析,给出 UML 的类图、序列图等。这些方法都是对 Web 应用进行静态整体分析,产生的结果不适用于用户使用服务行为分析。文献[10]提出从单个用户访问 Web 服务器日志出发,挖掘 Web 服务器访问日志,以概率时间自动机形式动态构造组合 Web 服务系统行为模型。文献[11]提出了从概率统计的角度对 Web 服务组合的可靠性进行验证的方法。虽然上述文献是围绕 Web 应用展开的,但 Web 服务作为 Web 应用的一种实现技术,上述研究成果在 Web 服务组合系统中同样适用。受上述文献的启发,考虑到从不同用户群组使用系统行为的行为存在差异,从不同群组不同的系统使用行为角度去建模能真实地反映系统运行时的行为模型,有利于依据 Web 服务流程的运行质量对 Web 服务流程进行调整,本文从用户访问 Web 服务器日志出发,采用聚类的方法构建面向用户群组的 Web 服务计算组合流程使用行为 QoS 模型。

关于 Web 服务流程的质量需求描述问题,概率模型检验中的需求性质采用概率时序逻辑公式进行描述,但这种描述方式如同阳春白雪,其描述和理解需要有较高的数学专业知识,虽对机器可读、易处理,但对使用这种描述语言的工作者

提出了更高的要求,而这项技能是一般的软件从业人员所不具备的。因此,对于 Web 服务流程的质量需求描述,本文采用带标记的有向图进行描述,将 Web 服务映射为有向图中的节点信息,将 Web 服务计算组合流程的控制结构映射为有向图中的边信息,将 QoS 指标值及其他约束映射为节点或边上的标记来描述。然后采用文献[10]提出的覆盖准则产生 Web 服务流程的 QoS 需求性质,并用 PCTL 来表示。

本文第 3 节通过聚类的方法对 Web 服务器访问日志进行分析,构建了面向群组的 Web 服务流程用户使用行为 QoS 模型;第 4 节给出扩展 QoS 的需求描述方式,并通过一个案例展示了这种方法用于验证 Web 服务流程质量是否满足需求;最后总结全文。

3 面向用户群组的 Web 服务流程使用行为 QoS 模型构建

3.1 形式化模型描述

根据 Web 服务访问日志对用户行为进行处理时,考虑从当前状态转移到下一个状态时的概率,也即当前状态只影响到下一个状态的转移概率,符合马尔可夫特性,因此,使用马尔可夫模型作为面向用户群组的服务使用行为模型的模型。考虑到 Web 服务流程会拥有不同的用户群组,从用户群组使用角度构建的系统行为模型能够客观真实地反映 Web 服务流程提供服务时被使用的行为模型及服务质量,每个群组的转移转移构成了一个状态转移空间,多个用户群组则构成了多个状态转移空间,因此,采用带回报期望的马尔可夫决策过程作为面向用户群组的 Web 服务流程使用行为 QoS 模型。

定义 1(面向用户群组的 Web 服务使用行为 QoS 模型) 面向用户群组的 Web 服务使用行为 QoS 模型是一个带回报期望的马尔可夫决策过程模型 P ,是一个 7 元组。

$$P = \langle S, init, \Sigma, T, AP, L, \Pi \rangle$$

其中, S 是非空的状态集合,用于表示 Web 服务组合中的 Web 服务; $init \in S$ 是非空的初始状态,用于表示初始的 Web 服务; Σ 是非空的标签集合,用于表示不同的用户群组; $T \subseteq S \times \Sigma \times Distr(S)$ 是状态转移关系,用于表示不同用户群组使用服务时的服务转移关系, $Distr(S)$ 是同一用户群组使用不同 Web 服务构成的样本空间 S 上的概率分布函数, $Distr(S) \rightarrow [0, 1]$; AP 是 Web 服务对应的原子命题集合; $L: S \rightarrow 2^{AP}$ 是标记函数,用于标记状态与原子命题的幂集之间的对应关系;回报期望函数 Π 的作用是产生 Web 服务质量值,回报期望值的值域是实数域。

马尔可夫决策过程 P 从某个状态出发选定某个用户群组执行状态迁移时,马尔可夫决策过程退化为马尔可夫链 C , C 的一次运行 r 是一条无限的状态序列, $r = \langle s_1, s_2, \dots \rangle$,对于每个 $i(0 \leq i < n)$ 存在 $\tau \in Distr(S)$ 使得 $(s_i, a) \rightarrow \tau, \tau(s_{i+1}) \geq 0$,并且 $\sum_{s_{i+1} \in S} \tau(s_{i+1}) = 1, \pi: \Pi \times S \rightarrow \hat{U}$ 产生 Web 服务 QoS 值。

3.2 模型构建

3.2.1 Web 访问日志中每个用户的使用行为处理

当用户访问 Web 服务器时,服务器容器会按照 server.xml 中的配置记录所有用户访问该服务器的行为以及服务器提供服务时的行为,生成相应的 Web 服务访问日志。为了构建面向用户群组的 Web 服务流程使用模型,首先应对日志中

每个用户的使用行为记录进行处理,包括访问日志清洗、用户识别、Web 服务建模、服务迁移建模、服务迁移概率建模。

(1)访问日志清洗。其是从原有的 Web 服务器访问日志中去掉与用户行为无关的信息。本文主要通过日志记录抽取用户请求的 Web 服务序列构建 Web 服务计算组合流程用户使用模式,因此本文从原有的 Web 访问日志记录中删除与 Web 服务无关的请求记录。

(2)用户识别。当 Web 服务计算组合流程复杂且拥有大量并发访问用户时,正确识别 Web 服务用户对于分析 Web 服务用户使用模式非常重要。用户识别错误将会影响到会话分析以及对于 Web 服务用户使用模式的分析。本文采用 Web 访问日志记录中的客户端 IP 地址来区分 Web 服务的用户。

(3)Web 服务建模。在 Web 服务流程用户使用行为模型建模过程中,Web services 在实现时可能包含多个操作,用户在使用某个 Web services 时,实际上是在使用该 Web services 操作列表中的某个具体的操作,这样对一个 Web service 进行 QoS 度量,其实质是对这个 Web service 中的几个操作对应的服务操作的 QoS 的组合 QoS 进行度量,其 QoS 求解方法与多个仅包含单一操作方法的原子 Web 服务的组合求解方法相同,因此以多个仅包含单一操作的原子 Web 服务组装的服务组合流程为例对 Web 服务流程进行建模。在 Web 服务建模时,将原子 Web 服务中的操作命名为服务名对原子 Web 服务进行建模。

同时,用户在使用原子 Web 服务的操作时,通常会根据具体需求输入不同的实参值,这样就造成了用户访问同一个 Web 服务但 URL,不同的效果,但这些服务操作名相同、形参相同、实参不同的 URL,其服务都由同一个 Web 服务提供,因此,对 URL 进行抽象,忽略 URL 中参数部分,从而抽取出现应的 Web 服务。特殊地,对于无法提供的用户访问服务请求,将其抽象为失败服务。

(4)服务迁移建模。服务迁移建模用于描述用户在使用 Web 服务流程中使用 Web 服务的迁移序列。可以通过识别用户会话的方式,按照会话中访问 Web 服务的先后次序对用户使用行为模式中的服务迁移建模。用户会话是指用户对服务器的一次有效访问,一个用户会话中含有该用户在一段时间内对 Web 应用的连续访问的 Web 服务。

(5)服务迁移概率建模。服务转移概率取决于用户访问服务的统计次数,结合用户会话分析中的有效会话和无效会话(服务器返回状态码为 404 的会话),根据公式 $p_{ij}^k = \frac{C(s_i, s_j)}{C(s_i, \cdot)}$ 进行计算,其中, p_{ij}^k 表示标识为 k 的用户使用 Web 服务计算组合流程时从服务 s_i 到服务 s_j 的转移概率, $C(s_i, s_j)$ 表示用户访问服务 s_i 后访问服务 s_j 的统计次数, $C(s_i, \cdot)$ 表示用户访问服务 s_i 后访问所有服务的统计次数总和, $p_{ij}^k \in [0, 1]$ 且 $\sum_{i=1, j=1}^{i=n, j=n} p_{ij}^k = 1$ 。

对 Web 服务器访问日志进行上述步骤的处理,可以得到每个用户使用 Web 服务的行为表。

定义 2(用户使用 Web 服务的行为表) 用户使用 Web 服务的行为表 M 由 m 行、 $(n^2 + 1)$ 列组成, m 行分别记录从服务器访问日志中分析出的 m 个用户的行为, n 表示分析出的 n 个服务, $(n^2 + 1)$ 列分别由 uid 列和 t_{ij} ($i \leq n, j \leq n$) 列构成。

uid 表示用户的唯一标识号, t_{ij} 表示从服务 s_i 到服务 s_j 的转移,表格元素 M_{ij}^k 表示第 k 行列名为 t_{ij} 的元素值, $M_{ij}^k = p_{ij}^k$, 如果 $M_{ij}^k = 0$,则表示服务 s_i 到服务 s_j 不存在转移,可用“—”简化表示,如果 $M_{ij}^k = 1$,则表示服务 s_i 到服务 s_j 之间存在稳定的转移关系。用户使用 Web 服务的行为表示例如表 1 所列。

表 1 用户使用 Web 服务的行为表示例

uid	t_{00}	t_{01}	t_{02}	t_{03}	t_{10}	...	t_{33}
μ_1	—	0.75	0.25	0.27	—	...	—
μ_2	—	0.70	0.30	0.21	—	...	—

3.2.2 面向用户群组的 Web 服务使用行为处理

通过 Web 服务访问日志进行分析处理得到了用户使用 Web 服务行为表,以此表为基础,采用聚类技术对 Web 服务用户进行划分,构建面向用户群组的 Web 服务使用行为模型。具体步骤包括:确定 Web 服务用户特征;计算用户行为的相似度;使用 K-means 聚类方法对用户进行划分,构建面向用户群组的 Web 服务使用行为模型。

(1)确定 Web 服务用户的特征是采用聚类方法划分用户的首要任务。将用户使用 Web 服务时的服务迁移概率作为用户划分的特征。用户特征的提取为计算用户访问 Web 服务时行为的相似度做好了准备工作。

(2)采用欧氏距离来计算不同用户之间的相似度。欧氏距离的计算公式如下:

$$d(u_i, u_k) = \sqrt{\sum_{i=1, j=1}^{i=n, j=n} (p_{ij}^i - p_{ij}^k)^2}$$

其中, $d(u_i, u_k)$ 表示用户 u_i 与用户 u_k 之间的距离,距离越大表明这两个用户在使用 Web 服务组合流程时行为模式差异越大,反之,则表示这两个用户的行为模式相似度较高。

(3)利用 K-means 聚类方法对 Web 用户进行划分。

K-mean 方法是聚类思想中的一个经典方法。K-means 聚类算法采用迭代的方法使簇内个体之间具有较高的相似度,同时保持簇间的相似度较低;此外,该算法相对简单,具有较高的实现效率。算法展示了使用 K-means 对 Web 服务用户聚类划分的算法。

算法 1 使用 K-means 算法对 Web 服务用户聚类划分

输入:划分的用户组数 k ,以及用户使用 Web 服务的行为表

输出:划分后的 k 组用户

步骤 1 随机选择 k 个用户作为 k 组用户的中心。

步骤 2 对每个用户,计算它与 k 个组的中心的相似程度。将该用户划入与他最相似的那个组中,直到所有用户都有属于各自的组。

步骤 3 更新各个用户组的中心值。

步骤 4 重复步骤 2、步骤 3,直到没有任何变化后算法停止。

对用户使用 Web 服务的行为表进行上述步骤的处理,可以得到划分为不同用户群组的 Web 服务使用行为表,从而可以构建不同用户群组使用 Web 服务的行为模型。

对于 QoS 的衡量,由于本文采用基于价值的需求工程的思想,将 Web 服务系统产生的业务价值作为衡量 QoS 的指标。将 Web 服务系统产生的业务价值是否能满足系统价值需求作为判断 Web 服务系统是否进行自适应调整的依据。

通常,影响系统业务价值的因素有很多,有直接的影响因素,也有间接的影响因素,有可量化的影响因素,也有不可量化的影响因素。例如,在网上购物系统中,销售额可以转化为利润来度量系统的业务价值,它是一个直接的、可量化的影响因素;而一个商品的正面评论会吸引更多的客户来购买,而负

面评论则会降低客户的购买欲,因此商品评论对系统业务价值的影响是间接的,而且往往是难以量化的。因此,业务价值是特定于系统的,它需要由业务专家通过市场分析、业务分析、客户分析等综合性的分析来定义。

因此,在考虑 Web 服务系统的业务价值模型的定义时,本文对 Web 服务系统的业务价值模型做了一定的简化,只考虑其可量化的因素,将对应于系统中各个原子服务所产生的利润作为 QoS 价值指标。

4 QoS 需求描述与验证

4.1 QoS 需求描述

考虑到定量概率模型检验器 PRISM 强大的数值分析能力,本文利用 PRISM 判断 Web 服务流程是否满足系统质量需求。

可从不同角度对 Web 服务流程的需求进行描述,如从静态目标元素进行需求分析的需求目标模型,从需求目标元素的动态行为进行分析的状态图。模型检验的方法是对系统行为状态是否需求进行验证,需要建立需求目标元素的动态行为模型,因此,本文采用 QoS 扩展的状态图来描述系统质量需求。QoS 扩展的状态图实质是带标记的有向图,Web 服务映射为有向图中的节点信息,Web 服务流程的控制结构映射为有向图中的边信息,将 QoS 指标值映射为节点或边上的标记。

采用概率模型检验方法在系统行为模型上检验用概率时序逻辑公式描述的系统需求性质是否成立,因此在建立了系统质量需求模型后,还需要将其转化为相应的概率时序逻辑描述性质公式。本文采用文献[10]提出的覆盖准则产生 Web 服务流程的 QoS 需求性质,并用 PCTL 来表示。

4.2 案例分析

为了说明本文所提出的 Web 服务流程 QoS 验证方法,以形式化方法研究实验室开发的形式化模型验证项目作为案例进行展示。图 1 展示了 Web 服务流程 QoS 评估控制流,由 7 个原子 Web 服务组成,分别是初始化服务、日志分析服务、用户体验评分捕获服务、系统使用行为建模服务、需求建模服务、需求性质生成服务、QoS 概率模型检验服务,用 ws_i ($i=0, 1, 2, \dots, 6$) 表示。其中,圆圈表示 Web 服务,有向箭头表示 Web 服务组合顺序,符号“*”代表箭尾的服务全部完成后才能进入箭头指向的服务。

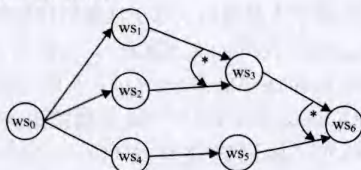


图 1 QoS 评估 Web 服务流程控制流

实验环境是 100M 的局域网通信系统,网络拓扑为星型结构,共有 30 个结点,每个结点都是单 CPU(Intel Pentium 4 2.8GHz,内存 2GB)结构,用 Windows 2003 Server 操作系统。将设计模型验证服务流程部署在其中一个结点上,采用 Apache Tomcat 作为服务器容器,以 AXIS2.0 作为服务引擎以支持服务运行,其他结点作为客户端向系统发出服务请求。采用 apache bench 对每个服务结点进行性能测试。

对该 QoS 评估 Web 服务流程的需求描述如下:按照每

项验证任务 QoS 设置为 1 个质量评价单位,迭代 480 次,该服务流程的 QoS 值是否达到 299。

利用 PRISM 对模型进行检验的结果如图 2 所示。

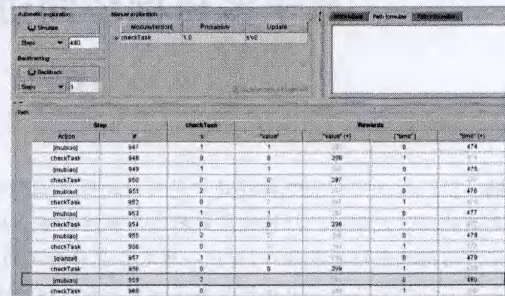


图 2 PRISM 模拟验证结果

可以看到,在 480 次迭代后,完成任务调度的 Web 服务价值回报值达到 299,位于价值期望 $[300-5, 300+5]$ 的范围内。

结束语 本文从用户访问 Web 服务产生的访问日志出发,采用聚类的方法,构建面向用户群组的 Web 服务 QoS 使用行为模式。针对 Web 服务计算组合流程的 QoS 需求描述,本文采用扩展的 QoS 状态图来描述,通过覆盖准则和转化算法产生 Web 服务流程的 QoS 需求性质,并用 PCTL 来表示。然后,采用概率模型检验的方法验证 Web 服务流程的 QoS 是否满足需求。

下一阶段将根据需要对 QoS 的指标进行扩充,并利用对 Web 服务计算组合流程 QoS 验证的定量结果进行分析,采用线性回归分析方法预测 Web 服务计算组合流程整体的 QoS 性能,采用非线性回归分析方法预测服务组件的 QoS 性能,为依据 QoS 性能指标对 Web 服务组合流程中的失效服务进行分析与定位做准备。

参考文献

- [1] 陈碧欢. 基于需求和体系结构的软件系统自适应方法[D]. 上海:复旦大学,2014
Chen Bi-huan. Requirements- and Architecture-based Self-adaptation of Software Systems[D]. Shanghai: Fudan University, 2014
- [2] Wang Zhen-zhen. Survey of Model Checking [J]. Computer Science, 2013, 40(Z6): 1-14
- [3] Calinescu R, Ghezzi C, Kwiatkowska M, et al. Self-adaptive software needs quantitative verification at runtime[J]. Communications of the ACM, 2012, 55(9): 69-77
- [4] Cardoso J, Sheth A, Miller J, et al. Quality of service for workflows and Web service processes[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2004, 1(3): 281-308
- [5] 范小芹, 蒋昌俊, 王俊丽, 等. 随机 QoS 感知的可靠 Web 服务组合[J]. 软件学报, 2009, 20(3): 546-556
Fan X Q, Jiang C J, Wang J L, et al. Random-QoS-Aware reliable Web service composition[J]. Journal of Software, 2009, 20(3): 546-556
- [6] 刘阳. 基于概率模型验证的服务流程建模与验证[D]. 上海: 上海大学, 2012

(下转第 147 页)

修改系统对应用程序进行检测,主要关注重新打包的应用程序,其中 DroidMOSS^[6]通过指印生成技术来检测第三方 Android 应用市场中重新打包的应用;DNADroid^[10]检测 Android 应用程序中的相似性并分析了相似性行为;PiggyApp^[11]检测现存 Android 市场中 piggybacked 的应用程序;DroidRanger^[12]检测现存官方和第三方 Android 市场中的恶意程序,但未根据重新打包应用的变化进行准确定位并细粒度地检测出应用程序中相关类、方法、接口以及域的信息改变情况从而利于进一步的自动分析。后者通过权限使用分析来确定越权行为,其中 Stowaway^[13]提取出 Android APIs 到权限之间的映射,并且据此检测应用程序中的越权行为;PScout^[2]分析了 Android 系统中的权限映射等。

本文的工作则关注于国内第三方市场重新打包应用程序的现存情况,对国内主流五大第三方 Android 应用市场中 572 个应用程序系统地进行了对比检测分析,具体有效地检测出现在 Android 第三方应用市场重新打包应用程序的所占比例、修改分类,以及修改后存在的越权和权限滥用行为,针对不同情况进行了相应分析。

结束语 本文设计了一套针对重新打包应用程序的安全检测系统,并对国内五大主流 Android 第三方应用市场中重新打包应用程序进行了安全检测及分析。分析结果显示,市场中 29.75%~36.28%的应用程序是重新打包过的。其中,78.84%的应用程序存在资源以及代码修改,19.58%的应用对权限进行了修改,而这些应用程序对官方市场同款的应用的标签或者名字进行了替换,对程序逻辑进行了修改。在对权限进行过修改的应用中,45.95%存在越权行为,27.03%存在滥用权限行为。第三方市场需要更严格地检测与分析这些存在的现象和问题,规范并保证第三方应用市场的安全。

目前针对第三方 Android 应用市场中重新打包应用程序的越权行为还未考虑如何从 Android 系统的角度来检测权限是否将用户使用权限修改为 ROOT 权限以及更好地协调用户选取权限与应用功能运行效果。本文工作未深入涉及这些问题,而这些问题将作为本文未来的研究工作。

参 考 文 献

[1] AppBrain[OL]. <http://www.appbrain.com/stats/number-of-android-apps>
 [2] Au K W Y, Zhou Yi-fan, Huang Zhen, et al. D. PScout; analyzing

(上接第 123 页)

Liu Yang. Probabilistic Model Checking-based Modeling and Verification for Service Flow[D]. Shanghai: Shanghai University, 2012
 [7] Ricca F, Tonella P. Web site analysis: structure and evolution [C]//Proceedings of the International Conference on Software Maintenance. San Jose, California, USA, 2000; 76-86
 [8] Tramontana P. Reverse Engineering Web Applications [C]//Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM05). Budapest, Hungary, 2005; 705-708
 [9] Belletini C, Marchetto A, Trentini A. WebUml; Reverse Engineering of Web Applications[C]//SAC. 2004; 1662-1669
 [10] 高洪皓. 基于概率模型验证的 Web 服务动态自适应配置[D]. 上海: 上海大学, 2012

the android permission specification[C]//Proceedings of the 19th ACM Conference on Computer and Communications Security. 2012; 217-228

[3] Aho A V, Lam M S, Sethi R, et al. Compilers; Principles, Techniques, and Tools[M]. Prentice Hall, 2006; 399-408
 [4] Gunasekera S. Android Apps Security[M]. Beijing: Publishing House of Electronics Industry, 2013; 37-53
 [5] Yang Bo, Tang Zhu-shou, Zhu Hao-jin, et al. Method of Android Applications Permission Detection Based on Static Dataflow Analysis[J]. Computer Science, 2012, 39(11A); 16-18
 [6] Zhou Wu, Zhou Ya-jin, Jiang Xu-xian, et al. DroidMOSS; Detecting Repackaged Smartphone Applications in Third-Party Android Marketplaces [C] // Proceedings of the 2nd ACM CO-DASPY. 2012; 317-326
 [7] Mitchell M, Tian Guang-yu, Wang Zhi. Systematic Audit of Third-Party Android Phones[C]//Proceedings of the 4th ACM Conference on Data and Application Security and Privacy. 2014; 175-186
 [8] Wu Lei, Grace M, Zhou Ya-jin, et al. The Impact of Vendor Customizations on Android Security[C]//Proceedings of the 20th ACM Conference on Computer and Communications Security. 2013; 623-634
 [9] Zhang Yuan, Yang Min, Xu Bing-quan, et al. Vetting Undesirable Behaviors in Android Apps with Permission Use Analysis [C]//Proceedings of the 20th ACM Conference on Computer and Communications Security. 2013; 611-622
 [10] Crussell J, Gibler C, Chen H. Attack of the Clones; Detecting Cloned Applications on Android Markets[C]//Proceedings of 17th European Symposium on Research in Computer Security. 2012; 37-54
 [11] Zhou Wu, Zhou Ya-jin, Grace M, et al. Fast, Scalable Detection of 'Piggybacked' Mobile Applications[C]//Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy. 2013; 185-195
 [12] Zhou Ya-jin, Wang Zhi, Zhou Wu, et al. Hey, You, Get off of My Market; Detecting Malicious Apps in Official and Alternative Android Markets[C]//Proceedings of the 19th NDSS. 2012
 [13] Felt A P, Chin E, Hanna S, et al. Android Permissions Demystified[C]//Proceedings of the 18th ACM Conference on Computer and Communications Security. 2011; 627-637

Gao Hong-hao. Dynamic Self-adaption Reconfiguration of Web Service using Probabilistic Model Checking [D]. Shanghai: Shanghai University, 2012

[11] 王晶, 戎玫, 张广泉, 等. 基于概率模型检测的 Web 服务组合验证[J]. 计算机科学, 2012, 39(1): 120-123
 Wang Jing, Rong Mei, Zhang Guang-quan, et al. Validation of Web Service Composition Based on Probabilistic Model Checking [J]. Computer Science, 2012, 39(1): 120-123
 [12] Filieri A, Ghezzi C, Tamburrelli G. Run-time efficient probabilistic model checking[C]//Proceedings of the 33rd International Conference on Software Engineering. ACM, 2011; 341-350
 [13] van der Meulen M J P, Strigini L, Revilla M A. On the effectiveness of run-time checks[M]//Computer Safety, Reliability, and Security. Springer Berlin Heidelberg, 2005; 151-164