



## (12) 发明专利

(10) 授权公告号 CN 116302984 B

(45) 授权公告日 2025. 03. 21

(21) 申请号 202310151819.7

(22) 申请日 2023.02.10

(65) 同一申请的已公布的文献号

申请公布号 CN 116302984 A

(43) 申请公布日 2023.06.23

(73) 专利权人 深圳华为云计算技术有限公司

地址 518129 广东省深圳市龙岗区坂田街道万科城社区华为公司华为总部办公楼101

(72) 发明人 付求爱 万锐媛 赖源豪 吴身修  
黄治钧(74) 专利代理机构 深圳市深佳知识产权代理事务  
所(普通合伙) 44285

专利代理师 陈彦如

(51) Int.Cl.

G06F 11/3668 (2025.01)

G06F 11/07 (2006.01)

(56) 对比文件

CN 113238889 A, 2021.08.10

US 2020117587 A1, 2020.04.16

CN 115470025 A, 2022.12.13

CN 115543802 A, 2022.12.30

审查员 张美霞

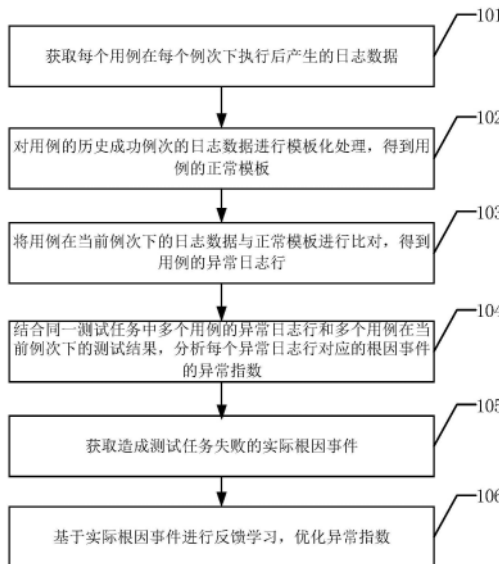
权利要求书2页 说明书13页 附图3页

## (54) 发明名称

一种测试任务的根因分析方法、装置及相关设备

## (57) 摘要

本申请实施例公开了一种测试任务的根因分析方法、装置及相关设备,用于提高根因分析的效率和准确性。本申请实施例方法包括:对用例的历史成功例次的日志数据进行模板化处理,得到用例的正常模板;将用例在当前例次下的日志数据与正常模板进行比对,得到用例的异常日志行;结合同一测试任务中多个用例的异常日志行和多个用例在当前例次下的测试结果,分析每个异常日志行对应的根因事件的异常指数,异常指数反应根因事件造成测试任务失败的可能性。



1. 一种测试任务的根因分析方法,其特征在于,包括:

对用例的历史成功例次的日志数据进行模板化处理,得到所述用例的正常模板,所述日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种,其中,所述测试执行机日志与所述产品运行日志为半结构化数据,所述调用链日志为结构化日志;

所述对用例的历史成功例次的日志数据进行模板化处理,具体包括:对用例的历史成功例次的测试执行机日志和/或产品运行日志通过日志异常检测和诊断模型进行处理;对所述用例的历史成功例次的调用链日志通过根因定位模型进行处理;其中,所述日志异常检测和诊断模型包括DeepLog模型,所述根因定位模型包括REPtrace模型;

将所述用例在当前例次下的日志数据与所述正常模板进行比对,得到所述用例的异常日志行;

结合同一测试任务中多个用例的异常日志行和所述多个用例在当前例次下的测试结果,分析每个所述异常日志行对应的根因事件的异常指数,所述异常指数反应所述根因事件造成所述测试任务失败的可能性。

2. 根据权利要求1所述的方法,其特征在于,计算每个所述异常日志行对应的根因事件的异常指数之后,还包括:

获取造成所述测试任务失败的实际根因;

基于所述实际根因进行反馈学习,优化所述异常指数。

3. 根据权利要求1或2所述的方法,其特征在于,在对用例的历史成功例次的日志数据进行模板化处理之前,还包括:

获取每个用例在每个例次下执行后产生的日志数据。

4. 根据权利要求1或2所述的方法,其特征在于,结合同一测试任务中多个用例的异常日志行和所述多个用例在当前例次下的测试结果,分析每个所述异常日志行对应的根因事件的异常指数,具体包括:

结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到所述多个用例对应的根因事件;

结合所述多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

5. 根据权利要求4所述的方法,其特征在于,结合所述多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,分析每个根因事件的异常指数具体包括:

根据多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,构建异质图;

通过高阶个性化页面排名HPPR得到每个所述根因事件的异常权重;

通过加权谱分析每个所述根因事件的异常指数。

6. 一种测试任务的根因分析装置,其特征在于,包括:

模板构建模块,用于对用例的历史成功例次的日志数据进行模板化处理,得到所述用例的正常模板,所述日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种,所述测试执行机日志与所述产品运行日志为半结构化数据,所述调用链日志为结构化日志;

所述模板构建模块,具体用于:对用例的历史成功例次的测试执行机日志和/或产品运

行日志通过日志异常检测和诊断模型进行处理;对所述用例的历史成功例次的调用链日志通过根因定位模型进行处理;其中,所述日志异常检测和诊断模型包括DeepLog模型,所述根因定位模型包括REPtrace模型;

异常比对模块,用于将所述用例在当前例次下的日志数据与所述正常模板进行比对,得到所述用例的异常日志行;

根因分析模块,用于结合同一测试任务中多个用例的异常日志行和所述多个用例在当前例次下的测试结果,分析每个所述异常日志行对应的根因事件的异常指数,所述异常指数反应所述根因事件造成所述测试任务失败的可能性。

7. 根据权利要求6所述的装置,其特征在于,还包括:

根因确定模块,用于获取造成所述测试任务失败的实际根因;

反馈优化模块,用于基于所述实际根因进行反馈学习,优化所述异常指数。

8. 根据权利要求6或7所述的装置,其特征在于,还包括:

数据获取模块,用于获取每个用例在每个例次下执行后产生的日志数据。

9. 根据权利要求6或7所述的装置,其特征在于,所述根因分析模块,具体包括:

日志聚类子模块,用于结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到所述多个用例对应的根因事件;

根因分析子模块,用于结合所述多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

10. 根据权利要求9所述的装置,其特征在于,

所述根因分析子模块,具体用于根据多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,构建异质图;通过高阶个性化页面排名HPPR得到每个所述根因事件的异常权重;通过加权谱分析每个所述根因事件的异常指数。

11. 一种测试任务的根因分析装置,其特征在于,包括收发模块和处理模块;

所述收发模块用于执行如权利要求1至5任一项所述的方法涉及的收发操作,所述处理模块用于执行如权利要求1至5任一项所述的方法涉及的处理操作。

12. 一种计算机可读存储介质,其特征在于,其上存储有计算机程序,所述计算机程序被通信装置执行时,使得所述通信装置执行如权利要求1至5中任一项所述的方法。

13. 一种计算机程序产品,包括程序,其特征在于,当所述程序被处理器执行时,实现权利要求1至5任一项所述的方法。

## 一种测试任务的根因分析方法、装置及相关设备

### 技术领域

[0001] 本申请涉及软件测试技术领域,尤其涉及一种测试任务的根因分析方法、装置及相关设备。

### 背景技术

[0002] 在软件开发中,调试一直是一个耗时、耗力且乏味的过程。软件产品能够提供各类服务,在产品上线之前,每个服务都会由研发或测试人员写上千的测试用例,来针对该服务的不同功能进行测试。测试用例定义了测试预期结果,每次用例执行复核预期则成功,否则失败,服务如需上线则需要成功的通过所有测试用例。目前对失败用例的根因分析仍较为依赖人工,然而服务越多,失败用例的数量越大,需要投入大量人力分析用例失败的原因,分析成本高,耗时长,可能在测试周期内都无法完成,从而影响产品发布的节奏。为了减少调试过程的开销,研究人员提出了一些智能化的手段进行用例根因分析方法,但当前这些方法大多只利用了失败例次单个日志,而单个数据源有时候只反映了失败表象,或是存在多个异常信息从而无法准确锁定根因。

[0003] 因此,如何提供一种测试任务的根因分析方法,提高根因分析的效率和准确性,是本领域技术人员亟待解决的技术问题。

### 发明内容

[0004] 本申请实施例提供了一种测试任务的根因分析方法、装置及相关设备,用于提高根因分析的效率和准确性。

[0005] 第一方面,本申请提供了一种测试任务的根因分析方法,包括:

[0006] 对用例的历史成功例次的日志数据进行模板化处理,得到用例的正常模板;

[0007] 将用例在当前例次下的日志数据与正常模板进行比对,得到用例的异常日志行;

[0008] 结合同一测试任务中多个用例的异常日志行和多个用例在当前例次下的测试结果,分析每个异常日志行对应的根因事件的异常指数,异常指数反应根因事件造成测试任务失败的可能性。

[0009] 本实施例中,通过结合每个用例历史多次成功执行的日志数据构建用例的正常模板,并根据当前例次下的日志数据与正常模板进行比对,得到用例在当前例次下的异常日志行;获取同一测试任务中多个用例的异常日志行,再结合每个用例在当前例次下的成功或失败结果,从而分析得到每个异常日志行对应的根因事件导致测试任务失败的可能性。本实施例所提供的方法,从当前测试用例的历史数据提取日志正常模式用来检测日志差异,且能根据同一测试任务下成功或失败用例的日志差异来排除软件正常变更导致的日志差异噪声,减少了失败用例根因定位人工分析成本,提升了服务快速迭代效率。

[0010] 一种可能的实现方式中,计算每个所述异常日志行对应的根因事件的异常指数之后,还包括:

[0011] 获取造成所述测试任务失败的实际根因;

[0012] 基于所述实际根因进行反馈学习,优化所述异常指数。

[0013] 本实施例中,在用户确认后,可通过反馈学习机制不断提升根因分析准确率。可以理解的是,本方法可引入基于反馈机制学习,例如通过获取经过用户确认的实际根因事件,来构建学习排序模型,使得推荐根因排序更加符合专家经验。

[0014] 一种可能的实现方式中,在对用例的历史成功例次的日志数据进行模板化处理之前,还包括:

[0015] 获取每个用例在每个例次下执行后产生的日志数据。

[0016] 一种可能的实现方式中,所述日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种。

[0017] 本实施例中,单个数据源有时候只反映了失败表象,或是存在多个异常信息无法锁定根因,因此可以结合执行机测试日志、调用链日志和产品运行日志等多源数据进行故障分析。

[0018] 一种可能的实现方式中,对用例的历史成功例次的日志数据进行模板化处理,具体包括:

[0019] 对用例的历史成功例次的测试执行机日志和/或产品运行日志通过日志异常检测和诊断模型进行处理;

[0020] 对所述用例的历史成功例次的调用链日志通过根因定位模型进行处理。

[0021] 本实施例中,针对测试执行机日志和产品运行日志这类半结构化日志,可以用基于系统日志使用深度学习方法做异常检测和诊断模型(DeepLog模型)实现模板化处理和差异诊断;针对调用链日志这类结构化日志,可以用REPtrace模型实现模板化处理和差异诊断。

[0022] 一种可能的实现方式中,结合同一测试任务中多个用例的异常日志行和所述多个用例在当前例次下的测试结果,分析每个所述异常日志行对应的根因事件的异常指数,具体包括:

[0023] 结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到所述多个用例对应的根因事件;

[0024] 结合所述多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

[0025] 本实施例中,可以进一步的对异常日志行中的冗余信息进行模板化处理,从而计算异常日志行之间的文本相似度,将相似度大于阈值的异常日志行模板聚类为同一个根因事件。

[0026] 一种可能的实现方式中,结合所述多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,分析每个根因事件的异常指数具体包括:

[0027] 根据多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,构建异质图;

[0028] 通过高阶个性化页面排名HPPR得到每个所述根因事件的异常权重;

[0029] 通过加权谱分析每个所述根因事件的异常指数。

[0030] 本实施例中,计算异常指数的方法可以具体为:构建“用例-根因事件”的异质图,通过高阶个性化网页排名(high-order personalized pagerank,HPPR)得到每个根因事件

的异常权重,在通过加权谱分析计算每个根因事件类的异常分数。

[0031] 第二方面,本申请实施例提供一种测试任务的根因分析装置,包括:

[0032] 模板构建模块,用于对用例的历史成功例次的日志数据进行模板化处理,得到所述用例的正常模板;

[0033] 异常比对模块,用于将所述用例在当前例次下的日志数据与所述正常模板进行比对,得到所述用例的异常日志行;

[0034] 根因分析模块,用于结合同一测试任务中多个用例的异常日志行和所述多个用例在当前例次下的测试结果,分析每个所述异常日志行对应的根因事件的异常指数,所述异常指数反应所述根因事件造成所述测试任务失败的可能性。

[0035] 一种可能的实现方式中,还包括:

[0036] 根因确定模块,用于获取造成所述测试任务失败的实际根因;

[0037] 反馈优化模块,用于基于所述实际根因进行反馈学习,优化所述异常指数。

[0038] 一种可能的实现方式中,还包括:

[0039] 数据获取模块,用于获取每个用例在每个例次下执行后产生的日志数据。

[0040] 一种可能的实现方式中,所述日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种。

[0041] 一种可能的实现方式中,所述根因分析模块,具体包括:

[0042] 日志聚类子模块,用于结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到所述多个用例对应的根因事件;

[0043] 根因分析子模块,用于结合所述多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

[0044] 一种可能的实现方式中,

[0045] 所述根因分析子模块,具体用于根据多个用例对应的根因事件和所述多个用例在当前例次下的测试结果,构建异质图;通过高阶个性化页面排名HPPR得到每个所述根因事件的异常权重;通过加权谱分析每个所述根因事件的异常指数。

[0046] 第三方面,本申请实施例提供一种测试任务的根因分析装置,包括收发模块和处理模块;

[0047] 所述收发模块用于执行如第一方面的方法实施例中的收发操作,所述处理模块用于执行如第一方面的方法实施例中的处理操作。

[0048] 第四方面,本申请实施例提供一种计算机可读存储介质,该计算机可读存储介质存储有计算机程序或指令,该计算机程序或指令用于在由一个或多个计算机执行时使得一个或多个计算机实施上述各方面中任意一方面任意可能的实施方式的方法。

[0049] 第五方面,本申请实施例提供一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行上述各方面中任意一方面任意可能的实施方式的方法。

[0050] 本申请第六方面提供一种芯片装置,包括处理器,用于与存储器相连,调用该存储器中存储的程序,以使得该处理器执行上述第一方面中的任一种实现方式。

[0051] 上述第二方面至六方面提供的方案,用于实现或配合实现上述第一方面提供的测试任务的根因分析方法,因此可以与第一方面达到相同或相应的有益效果,此处不再进行赘述。

## 附图说明

- [0052] 图1为本申请实施例所提供的测试任务的根因分析方法的方法流程图；  
[0053] 图2为本申请实施例所提供的测试任务的根因分析方法的应用示例图；  
[0054] 图3为本申请实施例所提供的测试任务的根因分析装置的结构示意图；  
[0055] 图4为本申请实施例所提供的测试任务的根因分析装置的结构示意图。

## 具体实施方式

[0056] 为了使本申请的目的、技术方案及优点更加清楚明白,下面结合附图,对本申请的实施例进行描述,显然,所描述的实施例仅仅是本申请一部分的实施例,而不是全部的实施例。本领域普通技术人员可知,随着新应用场景的出现,本申请实施例提供的技术方案对于类似的技术问题,同样适用。

[0057] 本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或模块的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或模块,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或模块。在本申请中出现的对步骤进行的命名或者编号,并不意味着必须按照命名或者编号所指示的时间/逻辑先后顺序执行方法流程中的步骤,已经命名或者编号的流程步骤可以根据要实现的技术目的变更执行次序,只要能达到相同或者相类似的技术效果即可。本申请中所出现的单元的划分,是一种逻辑上的划分,实际应用中实现时可以有另外的划分方式,例如多个单元可以结合成或集成在另一个系统中,或一些特征可以忽略,或不执行,另外,所显示的或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元之间的间接耦合或通信连接可以是电性或其他类似的形式,本申请中均不作限定。并且,作为分离部件说明的单元或子单元可以是也可以不是物理上的分离,可以是也可以不是物理单元,或者可以分布到多个电路单元中,可以根据实际的需要选择其中的部分或全部单元来实现本申请方案的目的。

[0058] 在软件开发中,调试一直是一个耗时、耗力且乏味的过程。软件产品能够提供各类服务,在产品上线之前,每个服务都会由研发或测试人员写上千的测试用例,来针对该服务的不同功能进行测试。测试用例定义了测试预期结果,每次用例执行复核预期则成功,否则失败,服务如需上线则需要成功的通过所有测试用例。目前对失败用例的根因分析仍较为依赖人工,然而服务越多,失败用例的数量越大,需要投入大量人力分析用例失败的原因,分析成本高,耗时长,可能在测试周期内都无法完成,从而影响产品发布的节奏。

[0059] 智能运维(artificial intelligence for it operations,AI0ps),是指将人工智能(artificial intelligence,AI)应用于运维领域,基于已有的运维数据通过AI的方式来解决传统运维没办法解决的问题。AI0ps主要包括问题发现、问题根因定位、问题修复三个阶段,本发明方案致力于解决测试场景下的失败用例根因定位问题。

[0060] 微服务架构与开发运营(development and operations,DevOps)模式下导致失败用例数量突增,需要投入大量人力分析用例失败的原因,分析成本高,耗时长,可能在测试

周期内都无法完成分析,从而影响产品发布的节奏。当前自动化分析能力已无法满足快速、准确的质量反馈要求。

[0061] 以某云服务为例,云服务包括上百个微服务,上万个接口,且需要按周/按天/按需发布,每个微服务都会有研发或测试人员所写的上千测试用例,针对该微服务不同功能(可视作用户请求)进行测试。测试用例定义了测试预期结果,每次用例执行符合预期则成功,否则失败,可视为用例级标签。用例执行过程会产生并保留用例执行机日志(测试日志)、调用链日志、产品运行日志等数据。服务如需上线则需要成功通过所有测试用例,目前对失败用例的分析仍较为依赖人工,面对海量微服务与快节奏开发带来的大量失败用例,无法满足快速质量反馈。因此需要有基于用例执行产生的关联数据以数据驱动的智能化手段快速定位用例失败的根因,做到真正帮助研发人员在开发阶段快速定位并修复问题。

[0062] 目前业界已有探索以数据驱动的智能化手段进行用例根因分析的方法,例如:

[0063] 1) 采用基于关键词和正则表达式的日志分析方法,通过收集测试用例关联产生的日志数据,基于关键词和正则表达式的方式识别日志中出现的异常日志,进行失败用例故障定位。

[0064] 这种方法存在以下缺陷:日志规范不统一,日志模式多样,不同厂家有不同的规范与专业术语,对分析人员专业技能要求高;日志数据量大,每天日志产生约GB/TB量级,分析工作耗时长;日志数据半结构化或非结构化,规则不明显,很难预设正则表达式提取关键信息;服务快速变更可能导致手工设置规则失效。

[0065] 2) 在加载工程代码时,在工程代码中注入日志输出字节码;当运行包含工程代码的测试脚本时,进入测试脚本的每个方法前通过日志输出字节码记录方法调用日志,并根据方法调用日志形成方法调用层次的日志文件;最后根据日志文件中的方法调用层次分析运行失败的测试脚本,定位运行失败的原因。其本质上采用的是一种传统的基于调用链定位失败原因方式,通过将当前失败调用链日志与历史最近一次成功调用链日志做人工对比,找出差异点,并根据人工经验识别其中导致用例失败的原因。

[0066] 这种方法存在以下缺陷:使用了用例最近一次运行成功版本的调用日志与当前版本的调用日志做对比,忽略了用例可以存在多种正常调用层次,从而导致用例失败原因的误诊断。此外,该算法不具备对识别的失败原因按照根因可能性排序的功能,无法优先推荐最可能导致用例失败的原因,降低了其分析的有效性;人工对比找差异效率低,但十分依赖人工经验,需要人工识别哪些差异是噪声、表象或是潜在根因。

[0067] 3) 基于有监督的测试日志根因分析方法,通过人工标注失败原因,训练机器学习模型,推荐问题根因。

[0068] 该方法的不足在于:日志异常模式多样,日志中多含有噪声;需要大量人工标注数据来训练模型;对智能算法性能要求高,可解释性弱。

[0069] 为了解决上述问题,本申请提供了一种测试任务的根因分析方法。本申请实施例所提供的测试任务的根因分析方法所应用的场景是:在测试任务失败的情况下,该测试任务中每个用例在当前例次下的状态(每个用例在当前例次下是成功还是失败)是已知的。

[0070] 请参阅图1,图1为本申请实施例所提供的测试任务的根因分析方法的方法流程图,包括:

[0071] 101,获取每个用例在每个例次下执行后产生的日志数据。

[0072] 可以理解的是,在一个测试任务中包括若干个测试用例,且每个测试用例在会按照测试需求进行多次测试,即在多个例次下执行该测试。在每个例次下执行后,可能会执行成功,也可能会执行失败,无论最终执行的结果是成功还是失败,执行过程会产生并保留用例执行机日志、调用链日志、产品运行日志等数据。

[0073] 本实施例中,为了便于后续对测试任务的失败用例进行根因分析,可以获取每个用例在每个例次下执行后产生的日志数据,当然,也可以只获取需要进行分析的用例的相关日志数据。该日志数据可保存在相关存储设备中,后续若需要对哪个用例进行分析,可直接调用该相关日志数据。在另一种实现方法中,相关日志数据可以保存在测试系统中,当出现测试任务失败的情况时,再从系统中调取该测试任务的每个用例在每个例次下执行后产生的日志数据。

[0074] 进一步的,由于单个数据源有时候只反映了失败表象(如400返回码),或是存在多个异常信息无法锁定根因,因此可以结合执行机测试日志、调用链日志和产品运行日志等多源数据进行故障分析,即本实施例中,日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种。

[0075] 102,对用例的历史成功例次的日志数据进行模板化处理,得到用例的正常模板。

[0076] 可以理解的是,每一个测试任务的多个用例需要进行执行多次,且多次执行均成功的情况下,该测试任务才算成功,若用例在某一例次下执行失败,则表示在当前例次下,该测试任务失败。因此,当测试任务失败时,先获取该测试任务每个用例的历史成功例次的日志数据,基于日志内容与上下文不变量的方法,通过学习日志内容位置、频率与语义,分离语义模板与变量,降低数据分析维度;根据不同数据结构特点(树、序列)基于历史测试成功数据对正常态建模,得到每个用例的正常模板。

[0077] 本实施例中,通过对与用例关联的各个数据源(调用链日志、测试执行机日志、产品运行日志)的历史数据进行分析,不依赖业务知识建立正常数据模型(模板),形成与用例对应的正常模板,即单用例画像,以便于后续将正常模板与当前例次下的日志数据进行对比,得到数据差异。

[0078] 在一种可能的实现方法中,该步骤102具体包括:

[0079] 1021,对用例的历史成功例次的测试执行机日志和/或产品运行日志通过日志异常检测和诊断模型进行处理;

[0080] 1022,对用例的历史成功例次的调用链日志通过根因定位模型进行处理。

[0081] 本实施例中,针对测试执行机日志和产品运行日志这类半结构化日志,可以用基于系统日志使用深度学习方法做异常检测和诊断模型(DeepLog模型)实现模板化处理和差异诊断;针对调用链日志这类结构化日志,可以用REPtrace模型实现模板化处理和差异诊断。可以理解的是,不同形态的日志数据,可以分别对应相应形态的正常模板,即针对某一个测试用例而言,可以包括测试执行机日志正常模板、产品运行日志正常模板和调用链日志正常模板。

[0082] 103,将用例在当前例次下的日志数据与正常模板进行比对,得到用例的异常日志行。

[0083] 可以理解的是,基于步骤103中通过历史成功日志数据构建的正常模板,自动化比较当前例次成功或失败用例的日志与其相应形态日志的正常模板,得出各个用例(包括本

次测试)的差异(DIFF)日志行列表。

[0084] 请参阅图2,图2为本申请实施例所提供的测试任务的根因分析方法的应用示例图。

[0085] 例如,成功用例T1的DIFF日志行列表包含L3一个日志行,失败用例T2的DIFF日志行列表包含L2和L4两个日志行,失败用例T3的DIFF日志行列表包含L1,L5和L6三个日志行。

[0086] 本实施例中,对比测试用例在失败时与正常模板逻辑、参数差异,识别当前异常态用例与正常表现时的差异点,得到差异检测结果。

[0087] 104,结合同一测试任务中多个用例的异常日志行和多个用例在当前例次下的测试结果,分析每个异常日志行对应的根因事件的异常指数,异常指数反应根因事件造成测试任务失败的可能性。

[0088] 可以理解的是,多个用例可能是由同一原因导致失败,如环境配置、同一段代码Bug等。通过对统一测试任务的多用例关联分析,可以在无标签下识别共同根因,并将相同失败根因的用例聚合推荐,减少重复问题分析量。

[0089] 在同一测试任务中,在获取了该测试任务中每个用例在当前例次下的测试结果,且将每个用例在当前例次下的日志数据与正常模式进行比对后,可以得到每个异常日志行与测试结果之间的关联性。例如,在当前例次下,成功用例T1的DIFF日志行列表包含L3一个日志行,说明该用例T3在当前例次的日志数据中L3与正常模板的L3存在差异,但是这个差异并不影响测试结果,则L3对应的根因事件的异常指数低,表示L3对应的根因事件造成任务失败的可能性低。若存在多个失败用例的DIFF日志行列表包括同样的日志行,那么该日志行对应的根因事件的异常指数高。该异常指数可以用过0至1之间的小数表示,用于体现根因事件导致测试任务失败的百分比。

[0090] 在一种可能的实现方法中,该步骤104具体包括:

[0091] 1041,结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到多个用例对应的根因事件;

[0092] 1042,结合多个用例对应的根因事件和多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

[0093] 本实施例中,可以进一步的对异常日志行中的冗余信息进行模板化处理,从而计算异常日志行之间的文本相似度,将相似度大于阈值的异常日志行模板聚类为同一个根因事件(E)。

[0094] 请参阅图2,例如:若L3和L4的相似度大于阈值,因此可以将L3和L4聚类为同一根因事件E3;L2和L5的相似度大于阈值,因此可以将L2和L5聚类为同一根因事件E2。L1、L6与其他日志行之间的相似度均低,因此分别对应根因事件E1、E4。

[0095] 具体的,将文本相似度高于阈值的异常日志行聚类,可采用TFIDF(term frequency-inverse document frequency)向量实现。TF-IDF是一种用于信息检索与数据挖掘的常用加权技术,用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。

[0096] 在一种可能的实现方法中,该步骤1042具体包括:

[0097] 10421,根据多个用例对应的根因事件和多个用例在当前例次下的测试结果,构建异质图;

[0098] 10422,通过高阶个性化页面排名HPPR得到每个根因事件的异常权重;

[0099] 10423,通过加权谱分析每个根因事件的异常指数。

[0100] 本实施例中,计算异常指数的方法具体包括:构建“用例-根因事件”的异质图,通过高阶个性化网页排名(high-order personalized pagerank,HPPR)得到每个根因事件的异常权重,在通过加权谱分析计算每个根因事件类的异常分数。

[0101] 如图2所示,各用例的日志行进行聚类后,构建异质图,该异质图中各根因事件节点之间的箭头表示了该根因事件对应的日志行的位置顺序,例如,用例T2包括E2,E3节点,其中E2节点对应日志行L2,E3节点对应日志行L4,L4的位置在L2之后,因此E3指向E2。同理,在用例T3中,E1对应日志行L1,E2对应日志行L5,E4对应日志行L4,那么箭头指向为:E4指向E1和E2,E2指向E1。

[0102] 在构建了“用例-根因事件”的异质图后,通过HPPR迭代求解根因事件的异常权重,对于任意根因事件E,定义频谱(ep、ef、np、nf),其中ef为执行了E且执行失败的测试用例数,ep为执行了E且执行成功的测试用例数,nf为没有执行E且执行失败的测试用例数,np为没有执行E且执行成功的测试用例数。

[0103] 基于程序频谱的动态缺陷定位方法具有较好的定位效果,例如Dstar2方法、Ochiai方法、Goodman方法、Sørensen方法、Jaccard方法、RussellRao方法、M2方法、Dice方法等。具体的,本申请实施例中可以采用Dstar2方法、Ochiai方法和M2方法。可以理解的是上述方法对应的公式为本领域常用技术,此处不再进行赘述。

[0104] 用例-根因事件异质图权重分数主要考虑:

[0105] a) 用例权重:如果一个失败用例只包含很少根因事件,那么这个用例需要排查的异常范围更小,在计算异常分数时权重应该更高。如失败用例T2只包含E2和E3两个根因事件,那么E2和E3的权重应该更大;

[0106] b) 根因事件权重:如果一个根因事件被多个权重高的失败用例覆盖,那么这个根因事件更可能是根因,权重也应该高。如E2同时被失败用例T2和T3覆盖,更可能是根因;

[0107] c) 时间权重:因为因果关系具有时间顺序,所以发生时间早或是含有异常关键字的根因事件权重更高,该异常关键字可以通过相关技术人员根据经验进行预先设置;

[0108] d) 异构日志可以设置不同权重。如产品运行日志比测试执行机日志能反映更加底层的程序执行逻辑,因此产品运行日志的DIFF日志行对应的根因事件应该更有可能是根因。

[0109] 在设定权重后,通过加权谱分析计算每个根因事件类的异常分数。谱分析异常分数公式反映根因事件导致测试失效的比例,例如由于E2因为同时出现在T2和T3的DIFF日志行里,且没有出现成功用例T1里,所以最终得分最高。

[0110] 例如,通过上述方法,计算得到根因事件对应的异常分数如下表所示:

表 1 本申请实施例中的根因事件对应的异常分数表

|        | 根因事件 | 异常分数 |
|--------|------|------|
| [0111] | E1   | 0.4  |
|        | E2   | 0.9  |
|        | E3   | 0.2  |
|        | E4   | 0.6  |

[0112] 在得到所有根因事件对应的异常指数后,可以推荐失败用例T2和T3的TOP根因事件E2供用户确认。

[0113] 105,获取造成测试任务失败的实际根因事件。

[0114] 106,基于实际根因事件进行反馈学习,优化异常指数。

[0115] 本实施例中,在用户确认后,可通过反馈学习机制不断提升根因分析准确率。

[0116] 可以理解的是,本方法可引入基于反馈机制学习,例如通过获取经过用户确认的实际根因事件,来构建学习排序模型,使得推荐根因排序更加符合专家经验。排序学习是一个有监督的机器学习过程,对每一个给定的查询—文档对,抽取特征,通过日志挖掘或者人工标注的方法获得真实数据标注。然后通过排序模型(LambdaMart),使得输入能够和实际的数据相似。

[0117] 本申请实施例中,通过用例各个形态日志(包括测试执行机日志、产品运行日志、调用链日志)的历史成功例次数据分别学习该用例自主学习每个形态的日志的正常模式,自动化比较当前例次成功或失败用例的日志与其相应形态日志的正常模式,得出异常日志片段列表,再关联多态日志异常分析根因,不依赖专家经验,以无监督方式自动推荐失败用例细粒度根因,减少失败用例根因定位人工分析成本,提升服务快速迭代效率。

[0118] 本申请还提供了一种测试任务的根因分析装置。请参阅图3,图3为本申请实施例所提供的测试任务的根因分析装置的结构示意图,包括:

[0119] 模板构建模块301,用于对用例的历史成功例次的日志数据进行模板化处理,得到用例的正常模板;

[0120] 异常比对模块302,用于将用例在当前例次下的日志数据与正常模板进行比对,得到用例的异常日志行;

[0121] 根因分析模块303,用于结合同一测试任务中多个用例的异常日志行和多个用例在当前例次下的测试结果,分析每个异常日志行对应的根因事件的异常指数,异常指数反应根因事件造成测试任务失败的可能性。

[0122] 本实施例中,通过结合每个用例历史多次成功执行的日志数据构建用例的正常模板,并根据当前例次下的日志数据与正常模板进行比对,得到用例在当前例次下的异常日志行;获取同一测试任务中多个用例的异常日志行,再结合每个用例在当前例次下的成功或失败结果,从而分析得到每个异常日志行对应的根因事件导致测试任务失败的可能性。本实施例所提供的根因分析装置,从当前测试用例的历史数据提取日志正常模式用来检测日志差异,且能根据同一测试任务下成功或失败用例的日志差异来排除软件正常变更导致的日志差异噪声,减少了失败用例根因定位人工分析成本,提升了服务快速迭代效率。

[0123] 本申请实施例所提供的根因分析方法,用例的日志模板正常模式是数据驱动的,不需要或只需要少量业务知识介入,使得不依赖或弱依赖测试人员经验、专家经验,实现可自动化比较异常点的效果;同时,采用了针对同一个测试任务,进行跨用例同时段的用例根因聚类,可大量的减少由于同一类根因导致的大批量失败用例分析数,提升分析效率;另外,通过获取实际根因事件,可基于反馈学习持续改善失败根因推荐结果,结合专家经验提供正反馈,持续性的提升分析准确率。

[0124] 在一种可能的实现方法中,还包括:

[0125] 根因确定模块304,用于获取造成测试任务失败的实际根因;

[0126] 反馈优化模块305,用于基于实际根因进行反馈学习,优化异常指数。

[0127] 本实施例中,在用户确认后,可通过反馈学习机制不断提升根因分析准确率。可以理解的是,本方法可引入基于反馈机制学习,例如通过获取经过用户确认的实际根因事件,来构建学习排序模型,使得推荐根因排序更加符合专家经验。

[0128] 在一种可能的实现方法中,还包括:

[0129] 数据获取模块306,用于获取每个用例在每个例次下执行后产生的日志数据。

[0130] 可以理解的是,在一个测试任务中包括若干个测试用例,且每个测试用例在会按照测试需求进行多次测试,即在多个例次下执行该测试。本实施例中,在对每个用例的历史成功例次的日志数据进行处理之前,首先需要获取每个用例在各个例次下执行后产生的日志数据。

[0131] 在一种可能的实现方法中,日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种。

[0132] 可以理解的是,单个数据源有时候只反映了失败表象,或是存在多个异常信息无法锁定根因,因此可以结合执行机测试日志、调用链日志和产品运行日志等多源数据进行故障分析。

[0133] 在一种可能的实现方法中,根因分析模块303,具体包括:

[0134] 日志聚类子模块,用于结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到多个用例对应的根因事件;

[0135] 根因分析子模块,用于结合多个用例对应的根因事件和多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

[0136] 本实施例中,可以进一步的对异常日志行中的冗余信息进行模板化处理,从而计算异常日志行之间的文本相似度,将相似度大于阈值的异常日志行模板聚类为同一个根因事件。

[0137] 在一种可能的实现方法中,

[0138] 根因分析子模块,具体用于根据多个用例对应的根因事件和多个用例在当前例次下的测试结果,构建异质图;通过高阶个性化页面排名HPPR得到每个根因事件的异常权重;通过加权谱分析每个根因事件的异常指数。

[0139] 本实施例中,计算异常指数的方法可以具体为:构建“用例-根因事件”的异质图,通过高阶个性化网页排名(high-order personalized pagerank,HPPR)得到每个根因事件的异常权重,在通过加权谱分析计算每个根因事件类的异常分数,具体请参阅上述方法实施例中的步骤10421至10423的相关描述,此处不再进行赘述。

[0140] 本申请实施例还提供了一种测试任务的根因分析装置。请参阅图4,图4为本申请实施例所提供的测试任务的根因分析装置的结构示意图,根因分析装置用于执行图1所示的实施例中终端设备执行的步骤,具体请参考上述方法实施例中的相关介绍。

[0141] 本实施例中的根因分析装置包括收发模块401和处理模块402。

[0142] 通信装置400包括收发模块401和处理模块402。

[0143] 收发模块401可以实现相应的通信功能,收发模块401还可以称为通信接口或通信单元。处理模块402用于执行处理操作。

[0144] 可选的,通信装置400还可以包括存储模块,该存储模块可以用于存储指令和/或数据,处理模块402可以读取存储模块中的指令和/或数据,以使得通信装置实现前图1所示的方法实施例。

[0145] 该通信装置400可以用于执行上文方法实施例中终端设备所执行的动作。该通信装置400可以为终端设备或者可配置于终端设备的部件。收发模块401用于执行上述方法实施例中终端设备侧的接收相关的操作,处理模块402用于执行上述方法实施例中终端设备侧的处理相关的操作。

[0146] 可选的,收发模块401可以包括发送模块和接收模块。发送模块用于执行上述图1所示的方法实施例中终端设备的发送操作。接收模块用于执行上述图1所示的方法实施例中终端设备的接收操作。

[0147] 需要说明的是,通信装置400可以包括发送模块,而不包括接收模块。或者,通信装置400可以包括接收模块,而不包括发送模块。具体可以视通信装置400执行的上述方案中是否包括发送动作和接收动作。

[0148] 例如,通信装置400用于执行如下方案:

[0149] 处理模块402,用于对用例的历史成功例次的日志数据进行模板化处理,得到用例的正常模板;

[0150] 处理模块402,还用于将用例在当前例次下的日志数据与正常模板进行比对,得到用例的异常日志行;

[0151] 处理模块402,还用于结合同一测试任务中多个用例的异常日志行和多个用例在当前例次下的测试结果,分析每个异常日志行对应的根因事件的异常指数,异常指数反应根因事件造成测试任务失败的可能性。

[0152] 一种可能的实现方式中,

[0153] 收发模块401,用于获取造成测试任务失败的实际根因;

[0154] 处理模块402,还用于基于实际根因进行反馈学习,优化异常指数。

[0155] 一种可能的实现方式中,

[0156] 收发模块401,还用于获取每个用例在每个例次下执行后产生的日志数据。

[0157] 一种可能的实现方式中,日志数据包括测试执行机日志、产品运行日志和调用链日志中的一种或多种。

[0158] 一种可能的实现方式中,对用例的历史成功例次的日志数据进行模板化处理,具体包括:

[0159] 对用例的历史成功例次的测试执行机日志和/或产品运行日志通过日志异常检测和诊断模型进行处理;

[0160] 对用例的历史成功例次的调用链日志通过根因定位模型进行处理。

[0161] 一种可能的实现方式中,结合同一测试任务中多个用例的异常日志行和多个用例在当前例次下的测试结果,分析每个异常日志行对应的根因事件的异常指数,具体包括:

[0162] 结合同一测试任务中多个用例,将文本相似度高于阈值的异常日志行聚类为同一个根因事件,得到多个用例对应的根因事件;

[0163] 结合多个用例对应的根因事件和多个用例在当前例次下的测试结果,分析每个根因事件的异常指数。

[0164] 一种可能的实现方式中,结合多个用例对应的根因事件和多个用例在当前例次下的测试结果,分析每个根因事件的异常指数具体包括:

[0165] 根据多个用例对应的根因事件和多个用例在当前例次下的测试结果,构建异质图;

[0166] 通过高阶个性化页面排名HPPR得到每个根因事件的异常权重;

[0167] 通过加权谱分析每个根因事件的异常指数。

[0168] 本申请实施例还提供一种包括指令的计算机程序产品,当其在计算机上运行时,使得该计算机执行如上述图1所示的实施例的通信方法。

[0169] 本申请实施例还涉及一种计算机可读存储介质,包括计算机可读存储介质存储有计算机程序或指令,计算机程序或指令在由一个或多个计算机执行时,实现如图1所示实施例的方法步骤。

[0170] 本申请实施例还涉及一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行如图1所示实施例的方法步骤。

[0171] 本申请实施例还涉及一种芯片装置,包括处理器,用于与存储器相连,调用该存储器中存储的程序,以使得该处理器执行上述图1所示的实施例的方法。

[0172] 本申请的主要应用场景可以为测试分析领域和AIOps的根因分析(root cause analysis,RCA)领域,适用于不同环境、不同产品形态的测试定位场景,如云化服务场景、嵌入式场景等。本申请方案所用方法具备通用性、扩展性和可移植性,只要产品/服务/工具的测试用例能关联上多态日志信息,即可使用本申请去做分析,以达到快速定位失败根因的目的,大量提升分析效率、节省人工分析成本。本申请未来可能从以下几个方面进行扩展:

[0173] (1) 构建故障知识图谱,在基于知识图谱进行根因诊断和推导:根因分析的总体思路是在异常事件发生时,系统收集信息并生成该异常事件的知识图谱,在图的基础上运用演绎推理和归纳推理等方法来对事件根因进行分析。

[0174] (2) 与有监督方式进行融合:本申请前期检测出的差异点可作为有监督智能分析方法的输入,构造文本特征,接入有监督智能分析系统。

[0175] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0176] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦

合或通信连接,可以是电性,机械或其它的形式。

[0177] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0178] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0179] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

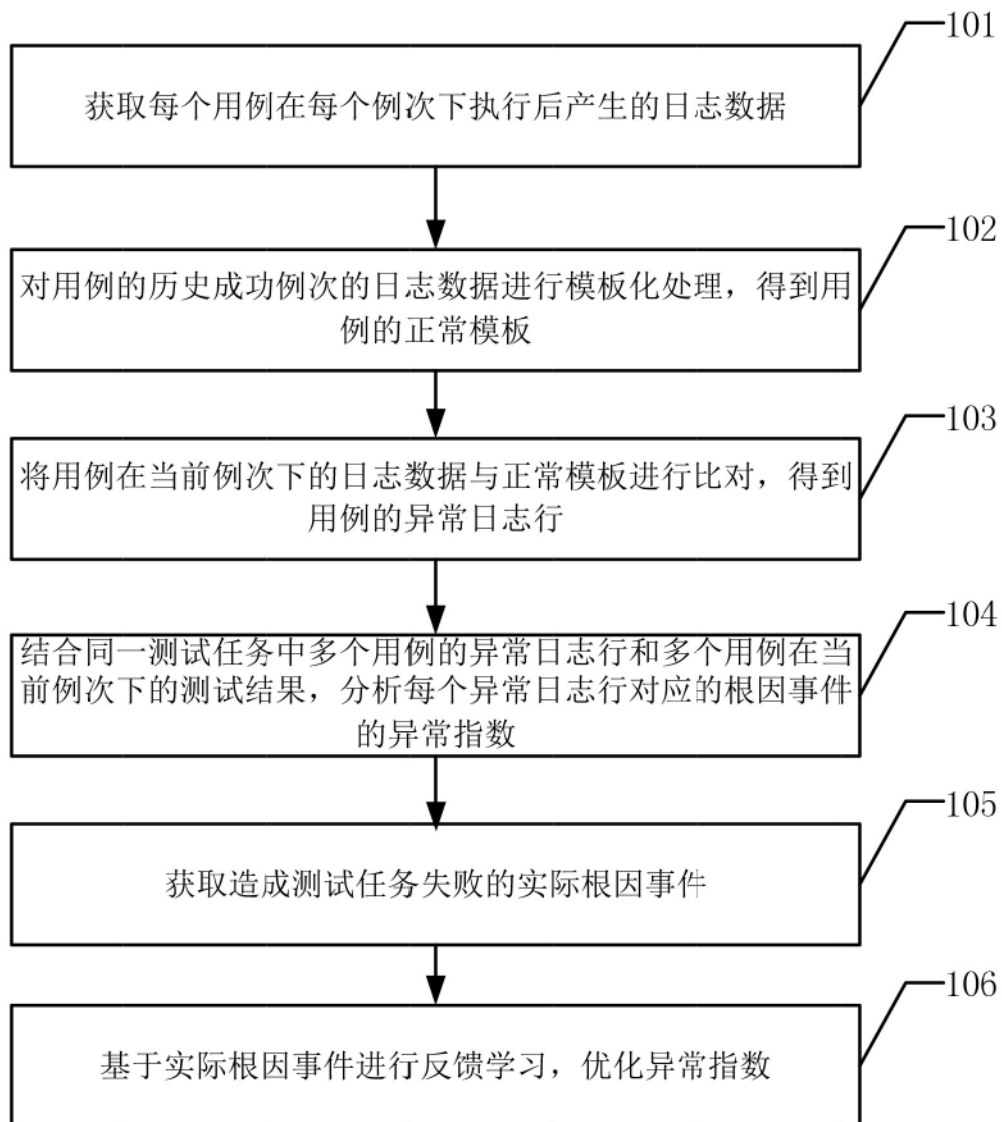


图 1

获取对同一测试任务的多个用例的异常日志行  
并根据异常日志行的相似度进行聚类

日志来源：测试执行机日志、产品运行日志、调用链日志

（成功）用例T1的DIFF日志行：L3

（失败）用例T2的DIFF日志行：L2, L4

（失败）用例T3的DIFF日志行：L1, L5, L6

对DIFF日志行采用TFIDF向量进行聚类

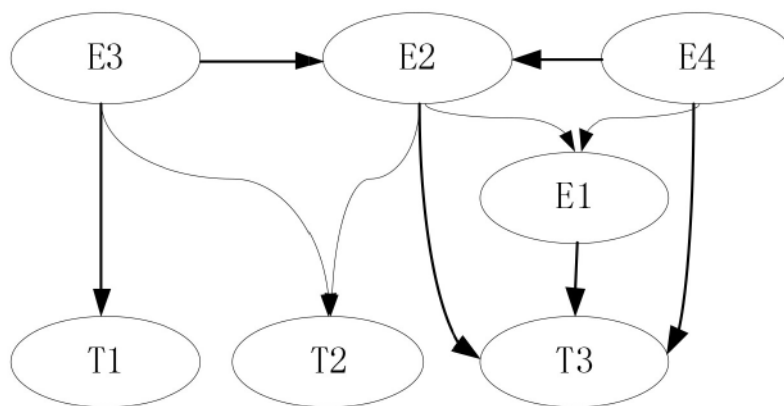
日志来源：测试执行机日志、产品运行日志、调用链日志

（成功）用例T1的根因事件：E3

（失败）用例T2的根因事件：E2, E3

（失败）用例T3的根因事件：E1, E2, E4

构建“用例-根因事件”的异质图，计算异常权重



采用HPPR迭代求解根因事件的异常权重，再进行谱分析来计算异常指数

图 2

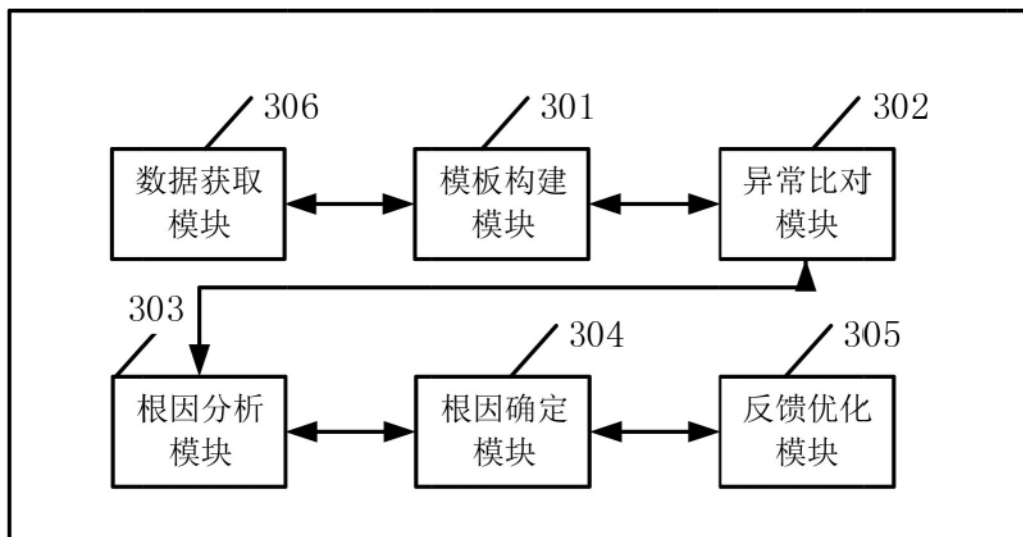


图 3

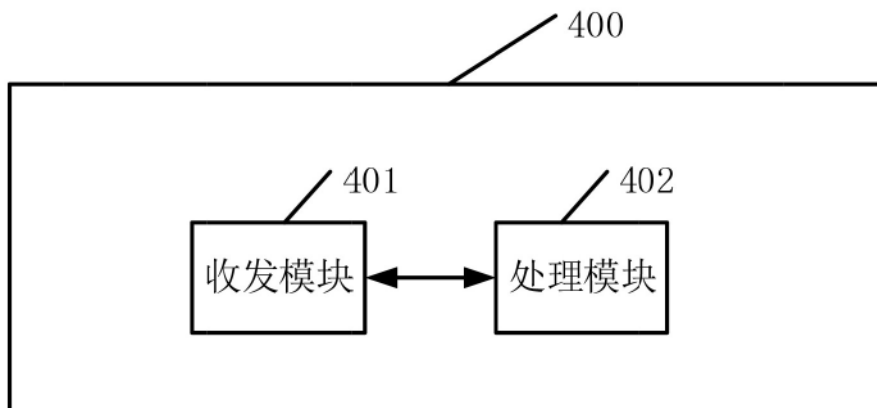


图 4