



(12) 发明专利

(10) 授权公告号 CN 112073269 B

(45) 授权公告日 2021.09.28

(21) 申请号 202010961881.9

审查员 郑杰

(22) 申请日 2020.09.14

(65) 同一申请的已公布的文献号
申请公布号 CN 112073269 A

(43) 申请公布日 2020.12.11

(73) 专利权人 腾讯科技(深圳)有限公司
地址 518057 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72) 发明人 陈金龙

(74) 专利代理机构 广州三环专利商标代理有限公司 44202
代理人 熊永强 杜维

(51) Int. Cl.
H04L 12/26 (2006.01)
H04L 29/08 (2006.01)

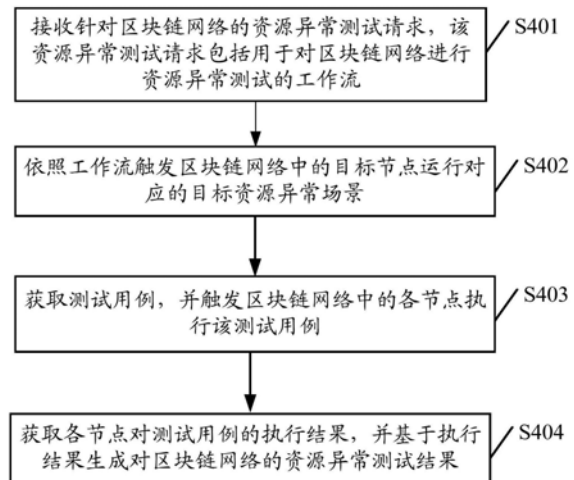
权利要求书2页 说明书15页 附图9页

(54) 发明名称

区块链网络测试方法、装置、服务器及存储介质

(57) 摘要

本申请实施例公开了一种区块链网络测试方法、装置、服务器及存储介质,其中方法包括:依照 workflow 触发区块链网络中的目标节点运行对应的目标资源异常场景,获取测试用例,并触发区块链网络中的各节点执行该测试用例。进一步地,可以获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。采用这样的方式,可以模拟运行资源异常场景,并在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。



1. 一种区块链网络测试方法,其特征在于,包括:

接收针对区块链网络的资源异常测试请求,所述资源异常测试请求包括用于对所述区块链网络进行资源异常测试的工作流,所述区块链网络包括至少一个节点;所述工作流包括目标资源异常场景的运行数据;

依照所述工作流触发所述区块链网络中的目标节点运行对应的目标资源异常场景,所述目标节点为所述至少一个节点中的一个或者多个;一个资源异常场景对应一个模拟执行脚本,所述目标节点通过所述工作流包括的运行数据运行所述目标资源异常场景对应的模拟执行脚本,以运行所述目标资源异常场景;

获取测试用例,并触发所述区块链网络中的各节点执行所述测试用例;

获取所述各节点对所述测试用例的执行结果,并基于所述执行结果生成对所述区块链网络的资源异常测试结果。

2. 如权利要求1所述的方法,其特征在于,当所述目标节点的数量为1时,所述目标节点对应的目标资源异常场景包括多个不同时序运行的资源异常场景;

当所述目标节点的数量为n时,n个目标节点中任一目标节点对应的目标资源异常场景包括一个资源异常场景或者多个不同时序运行的资源异常场景,所述n为大于1的整数。

3. 如权利要求1所述的方法,其特征在于,所述工作流还包括所述目标节点的地址信息;所述依照所述工作流触发所述区块链网络中的目标节点运行对应的目标资源异常场景,包括:

获取与所述目标资源异常场景关联的模拟执行脚本;

依照所述目标节点的地址信息向所述目标节点发送所述模拟执行脚本;

基于所述目标资源异常场景的运行数据向所述目标节点发送资源异常运行指令,所述资源异常运行指令用于指示所述目标节点运行所述模拟执行脚本,并且所述资源异常指令包括运行结束时间。

4. 如权利要求3所述的方法,其特征在于,所述目标资源异常场景的运行数据包括所述目标资源异常场景的运行开始时间和运行结束时间,所述资源异常运行指令是在检测到系统时间到达所述运行开始时间时发送的。

5. 如权利要求1所述的方法,其特征在于,所述基于所述执行结果生成对所述区块链网络的资源异常测试结果,包括:

根据所述各节点对所述测试用例的执行结果确定所述区块链网络中运行异常的节点数量;

依照所述区块链网络中运行异常的节点数量对所述区块链网络进行拜占庭共识分析,得到分析结果;

根据所述分析结果生成对所述区块链网络的资源异常测试结果。

6. 如权利要求5所述的方法,其特征在于,所述资源异常测试结果是依据所述分析结果和所述区块链网络中目标节点运行对应的目标资源异常场景的运行日志确定的。

7. 如权利要求6所述的方法,其特征在于,所述依照所述工作流触发所述区块链网络中的目标节点运行对应的目标资源异常场景之后,所述方法还包括:

监控所述区块链网络中的目标节点运行对应的目标资源异常场景的运行状态;

基于监控结果生成目标节点运行对应的目标资源异常场景的运行日志。

8. 如权利要求1所述的方法,其特征在于,所述触发所述区块链网络中的各节点执行测试用例,包括:

将所述测试用例在所述区块链网络中进行广播,以使得所述区块链网络中的各个节点接收所述测试用例,并返回针对所述测试用例的执行结果。

9. 一种区块链网络测试装置,其特征在于,包括:

通信模块,用于接收针对区块链网络的资源异常测试请求,所述资源异常测试请求包括用于对所述区块链网络进行资源异常测试的工作流,所述区块链网络包括至少一个节点;所述工作流包括目标资源异常场景的运行数据;

处理模块,用于依照所述工作流触发所述区块链网络中的目标节点运行对应的目标资源异常场景,所述目标节点为所述至少一个节点中的一个或者多个;一个资源异常场景对应一个模拟执行脚本,所述目标节点通过所述工作流包括的运行数据运行所述目标资源异常场景对应的模拟执行脚本,以运行所述目标资源异常场景;

所述处理模块,还用于获取测试用例,并触发所述区块链网络中的各节点执行所述测试用例;

所述处理模块,还用于获取所述各节点对所述测试用例的执行结果,并基于所述执行结果生成对所述区块链网络的资源异常测试结果。

10. 一种服务器,其特征在于,所述服务器包括处理器和存储装置,所述处理器和存储装置相互连接,其中,所述存储装置用于存储计算机程序,所述计算机程序包括程序指令,所述处理器被配置用于调用所述程序指令,执行如权利要求1-8任一项所述的方法。

11. 一种计算机存储介质,其特征在于,该计算机存储介质中存储有程序指令,该程序指令被执行时,用于实现如权利要求1-8任一项所述的方法。

区块链网络测试方法、装置、服务器及存储介质

技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种区块链网络测试方法、一种区块链网络测试装置、一种服务器及一种计算机存储介质。

背景技术

[0002] 区块链网络是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用,本质上是一个去中心化的数据库,也即是一串使用密码学方法相关联产生的数据块。

[0003] 在区块链网络正式发布之前,通常会进行异常测试,该异常测试包括资源异常测试。传统针对区块链网络的资源异常测试,往往采用手工+脚本的方式执行命令,具体地,通过人为事先规划好模拟路径,等到了预定时间点,由人工触发模拟路径的执行,从而完成对区块链网络的资源异常测试。这样的资源异常测试方式使得测试准确率低下,耗时且人工成本大。

发明内容

[0004] 本申请实施例提供了一种区块链网络测试方法、装置、服务器及存储介质,可以针对区块链网络精准模拟多种资源异常场景,并在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

[0005] 一方面,本申请实施例提供了一种区块链网络测试方法,该方法包括:

[0006] 接收针对区块链网络的资源异常测试请求,该资源异常测试请求包括用于对区块链网络进行资源异常测试的工作流,区块链网络包括至少一个节点;

[0007] 依照工作流触发区块链网络中的目标节点运行对应的目标资源异常场景,目标节点为上述至少一个节点中的一个或者多个;

[0008] 获取测试用例,并触发区块链网络中的各节点执行测试用例;

[0009] 获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。

[0010] 另一方面,本申请实施例提供了一种区块链网络测试装置,该区块链网络测试装置包括:

[0011] 通信模块,用于接收针对区块链网络的资源异常测试请求,该资源异常测试请求包括用于对区块链网络进行资源异常测试的工作流,区块链网络包括至少一个节点;

[0012] 处理模块,用于依照工作流触发区块链网络中的目标节点运行对应的目标资源异常场景,目标节点为至少一个节点中的一个或者多个;

[0013] 处理模块,还用于获取测试用例,并触发区块链网络中的各节点执行测试用例;

[0014] 处理模块,还用于获取各节点对测试用例的执行结果,并基于执行结果生成对区

区块链网络的资源异常测试结果。

[0015] 相应地,本申请实施例还提供了一种服务器,该服务器包括输出设备、处理器和存储装置;存储装置,用于存储程序指令;处理器,用于调用程序指令并执行上述的区块链网络测试方法。

[0016] 相应地,本申请实施例还提供了一种计算机存储介质,该计算机存储介质中存储有程序指令,该程序指令被执行时,用于实现上述的区块链网络测试方法。

[0017] 相应地,根据本申请的一个方面,提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行上述提供的区块链网络测试方法。

[0018] 本申请实施例中,服务器可接收携带有工作流的资源异常测试请求,依照 workflow 触发区块链网络中的目标节点运行对应的目标资源异常场景,获取测试用例,并触发区块链网络中的各节点执行该测试用例。进一步地,可以获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。采用这样的方式,可以针对区块链网络模拟运行资源异常场景,并在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

附图说明

[0019] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0020] 图1是本申请实施例提供的一种区块链的结构示意图;

[0021] 图2a本申请实施例提供的一种区块链网络测试系统的示意图;

[0022] 图2b本申请实施例提供的一种区块链网络测试系统的示意图;

[0023] 图3是本申请实施例提供的一种区块链网络测试方案的示意流程图;

[0024] 图4是本申请实施例提供的一种区块链网络测试方法的示意流程图;

[0025] 图5是本申请实施例提供的一种工作流的示意图;

[0026] 图6是本申请实施例提供的一种资源异常测试报告的示意图;

[0027] 图7a是本申请实施例提供的一种资源监控视图的示意图;

[0028] 图7b是本申请实施例提供的一种业务监控视图的示意图;

[0029] 图8是本申请实施例提供的一种区块链产品从生成至发布的示意流程图;

[0030] 图9是本申请实施例提供的一种测试平台的架构示意图;

[0031] 图10本申请实施例提供的一种对区块链网络进行资源异常测试的执行框架示意图;

[0032] 图11是本申请实施例提供的一种区块链网络测试装置的结构示意图;

[0033] 图12是本申请实施例提供的一种服务器的结构示意图。

具体实施方式

[0034] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0035] 首先,对本申请实施例所涉及的一些关键术语进行介绍:

[0036] 内存溢出:简称OOM,是指应用系统中存在无法回收的内存或使用的内存过多,导致被测程序(区块链)运行所要用到的内存大于节点能提供的最大内存,最终导致系统阻塞,区块链服务异常终止。

[0037] 记账节点:又叫出块节点、主节点(Leader节点)。在区块链应用场景中,每个打包周期内,需要记账节点将若干交易列表打包成区块结构,并广播给区块链网络中的其他节点共识;记账节点也就是生成区块并发起区块提案的节点。

[0038] BFT(Byzantine Fault Tolerance,拜占庭容错)共识:即拜占庭容错共识。拜占庭容错技术是一类分布式计算领域的容错技术,假设分布式系统拥有 p 个节点,并假设整个分布式系统中拜占庭节点不超过 m 台($p \geq 3m+1$),则整个分布式系统可正常运行,即容许 $p/3$ 个节点出现异常,其中 p 和 m 均为大于1的整数。

[0039] 共识算法:就是解决对某一提案(目标,投票等各种协作工作),大家达成一致意见的过程。

[0040] P2P网络:对等网络,该网络中的节点角色既可以是客户端,也可以是服务器。即该网络中的每一台计算机既能充当网络服务的请求者,又对其它计算机的请求做出响应,提供资源、服务和内容。

[0041] REQ:软件研发提测流程管理平台,管理软件版本开发、提测、缺陷提交、回归测试、发布整个生命周期。

[0042] 随着科技时代的到来和移动互联网的发展,网络变革的步伐也愈来愈快,实现同一领域或多个领域的信息融合,为客户提供全方位信息化方案的过程亦面临着体系结构的改进、支撑重心转移等新的挑战。因此,区块链技术作为分布式账本的一种特定实现,凭借其存储和管理数据的天然优势,逐渐成为各个领域存储数据和交易数据的首选方式。

[0043] 区块链网络是一种分布式系统,分布式系统可以由多个节点通过网络通信的形式连接形成。其中,节点可以是接入网络中的任意形式的计算设备,如服务器、终端等计算机设备。服务器可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器。终端可以是智能手机(如Android手机、iOS手机等)、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表等,但并不局限于此。节点之间可以通过有线或无线通信方式进行直接或间接地连接,本申请在此不做限制。

[0044] 参见图1,是本申请实施例提供的区块结构(Block Structure)的一个可选的示意图,每个区块中包括本区块存储数据记录的哈希值(本区块的哈希值)、以及前一区块的哈希值,各区块通过哈希值连接形成区块链(Blockchain)。另外,区块中还可以包括有区块生成时的时间戳等信息。区块链,本质上是一个去中心化的数据库,是一串使用密码学方法相

关联产生的数据块,每一个数据块中包含了相关的信息,用于验证其信息的有效性(防伪)和生成下一个区块。

[0045] 1) 应用,用于部署在区块链中,根据实际业务需求而实现特定业务,记录实现功能相关的数据形成记录数据,在记录数据中携带数字签名以表示任务数据的来源,将记录数据发送到区块链网络中的其他节点,供其他节点在验证记录数据来源以及完整性成功时,将记录数据添加到临时区块中。

[0046] 例如,应用实现的业务包括:

[0047] 1.1) 钱包,用于提供进行虚拟资源的交易的功能,包括发起交易即,将当前交易的交易记录发送给区块链网络中的其他节点,其他节点验证成功后,作为承认交易有效的响应,将交易的记录数据存入区块链的临时区块中。

[0048] 1.2) 共享账本,用于提供账目数据的存储、查询和修改等操作的功能,将对账目数据的操作的记录数据发送到区块链网络中的其他节点,其他节点验证有效后,作为承认账目数据有效的响应,将记录数据存入临时区块中,还可以向发起操作的节点发送确认。

[0049] 1.3) 智能合约,计算机化的协议,可以执行某个合约的条款,通过部署在共享账本上的用于在满足一定条件时而执行的代码实现,根据实际的业务需求代码用于完成自动化的交易,例如查询买家所购买商品的物流状态,在买家签收货物后将买家的虚拟资源转移到商户的地址;当然,智能合约不仅限于执行用于交易的合约,还可以执行对接收的信息进行处理的合约。

[0050] 2) 共识,用于解决并保证每一笔交易或者数据在所有记账节点上的一致性和正确性问题。区块链的共识机制就是确定达成某种共识和维护共识的方式。区块链的共识机制使其在不依靠中心化组织的情况下,依然大规模高效协作完成运转。

[0051] 区块链是一种多方共同维护,使用密码学保证传输和访问安全,能够实现数据一致存储、难以篡改、防止抵赖的记账技术,也就是分布式账本技术。区块链的生命周期如下:1、客户端发送交易给节点,节点通过P2P网络广播交易到区块链网络中的其他节点。2、利用共识机制随机选出记账节点。3、记账节点将若干交易列表打包组装成区块结构,生成区块提案,并基于区块提案生成区块提案消息,然后将区块提案消息并广播给其他节点;若交易列表为空,则打包成空块广播出去。4、其他节点收到区块提案消息后,验证其携带的区块所包括的交易,并基于验证结果广播对相应区块提案的投票信息。5、针对该区块达成BFT共识后,提交该区块到区块链存储层,开始下一轮选举和共识。

[0052] 为了达到多中心、不可篡改的目的,典型的区块链网络中的多节点满足一定的容错机制,该容错机制通常是BFT共识,即,在多节点构成的共识网络中,若存在某节点宕机、内存溢出或者作恶,整个区块链网络也是可以正常运行的,当然这些异常导致对区块链运行状态的影响必须得到可靠验证。在典型的节点异常场景中,分为资源异常和业务异常,该资源异常可以包括CPU(central processing unit,中央处理器)、内存、磁盘、网络等异常,这些资源异常在真实的生产环境或者测试环境都会发生,在区块链分布式系统验证异常场景的情况下,需要考虑多种资源交叉异常,涉及到组合资源异常和时序资源异常等。传统针对区块链网络的资源异常测试,往往采用手工+脚本的方式执行命令,具体地,通过事先规划好模拟路径,等到了预定时间点,由人工触发模拟路径的执行,从而完成对区块链网络的资源异常测试。这样的资源异常测试方式使得测试准确率低下,耗时且人工成本大,也很难覆

盖到组合资源异常场景和时序资源异常场景。其中,该组合资源异常场景可以指多种单一资源异常场景的组合,例如区块链网络中的多个节点运行不同的资源异常场景;该时序资源异常场景可以包括单节点多时序异常和多节点多时序异常,单节点多时序是指区块链网络中的一个节点运行不同时序的资源异常场景,多节点多时序异常是指区块链网络中的多个节点运行不同时序的资源异常场景。

[0053] 为了实现区块链网络资源异常测试的自动化,从而有效提高资源异常测试效率和准确率,本申请实施例提出了一种区块链网络测试系统,并基于该区块链网络测试系统提出了一种区块链网络测试方案。参见图2a,本申请实施例所提出的区块链网络测试系统可包括:区块链网络测试平台对应的服务器10(该服务器10可以理解为测试平台的后台服务器)、区块链网络测试平台对应的客户端11和区块链网络12,该区块链网络12中可包括多个节点设备(简称节点);此处的节点是指区块链网络中用于进行数据处理的基础设备,其可以包括但不限于:智能终端、平板电脑、台式计算机、服务器等等。在一种实施方式中,区块链网络测试平台对应的服务器10可以为区块链网络12外的服务器;此实施方式下的区块链网络测试系统可以参见图2a所示,区块链网络测试平台对应的服务器10可与区块链网络12内的节点进行通信。再一种实施方式中,区块链网络测试平台对应的服务器10可以作为区块链网络12中的节点;此实施方式下的区块链网络测试系统可以参见图2b所示。应理解的是,图2a和图2b只是示例性地表征数据处理系统的架构,其并不对该数据处理系统的具体架构进行限定。为便于阐述,除非特别指明,后续所提及的区块链网络测试系统均以图2a所示的区块链网络测试系统为例进行说明。

[0054] 在一个实施例,适用于上述区块链网络测试系统,本申请实施例所提出的区块链网络测试方案可由图2a或者图2b所示的区块链网络测试平台对应的服务器10和测试平台对应的客户端11执行。其中,测试平台可以是REQ平台。参见图3,该区块链网络测试方案的实现流程可以包括以下步骤:

[0055] S300:测试平台对应的客户端获取用于对区块链网络进行资源异常测试的工作流。具体实现中,测试平台可以对多个区块链网络进行管理,该管理包括测试、发布等等。

[0056] 在一个实施例中,当用户想要对某一区块链网络进行资源异常测试时,可以通过客户端登录测试平台,在测试平台中从多个区块链网络中选取待测的区块链网络,并编写用于对待测的区块链网络进行资源异常测试的工作流,客户端可以获取该用户编写的工作流。其中,该工作流包括至少一个目标资源异常场景的运行参数和运行各目标资源异常场景的目标节点的地址信息;任一目标资源异常场景的运行参数可以包括运行开始时间、运行结束时间,任一目标节点的地址信息包括IP(Internet Protocol,网际互连协议)地址、端口地址等等;目标节点为待测的区块链网络中的一个或者多个节点。

[0057] 或者,在另一个实施例中,开发人员可以预先配置多种资源异常场景(以下简称预设资源异常场景),以及各预设资源异常场景的工作流。后续,测试平台对应的客户端可以展示各预设资源异常场景对应的选项供用户选取,当用户选取任一预设资源异常场景后,客户端可以获取预先配置的该任一预设资源异常场景对应的工作流。

[0058] 其中,上述预设资源异常场景可以为单一的资源异常场景,例如CPU、网络、磁盘、内存、进程等单一资源异常场景,也可以为组合资源异常场景和时序资源异常场景。

[0059] S301:当测试平台对应客户端检测到对上述区块链网络的资源异常测试指令时,

向测试平台对应服务器发送资源异常测试请求,该资源异常测试请求中包括上述 workflow。

[0060] 具体实现中,用户可以通过客户端输入对区块链网络的资源异常测试指令,当客户端检测到资源异常测试指令后,可以向测试平台对应的服务器发送携带有上述用于对区块链网络进行资源异常测试的 workflow。

[0061] S302:测试平台对应服务器依照 workflow 触发区块链网络中的目标节点运行对应的目标资源异常场景。在一个实施例中,一个资源异常场景对应一个模拟执行脚本,该模拟执行脚本为预先配置的,节点运行任一模拟执行脚本,则相当于运行该任一模拟执行脚本对应的资源异常场景。或者,也可以理解为节点运行任一模拟执行脚本,可以使得该节点处于该任一模拟执行脚本对应的资源异常场景。示例性地,假设资源异常场景1为磁盘满载,该资源异常场景1对应模拟执行脚本1,这种情况下,若节点A运行模拟执行脚本1则可以使得节点A的磁盘满载,也使得节点处于资源异常场景1。这种情况下,上述步骤S302的具体实施方式可以为:服务器获取与 workflow 中目标资源异常场景关联的模拟执行脚本,依照目标节点的地址信息向目标节点下发模拟执行脚本,进一步地,可以基于目标资源异常场景的运行数据向目标节点发送资源异常运行指令,该资源异常运行指令用于指示运行模拟执行脚本,以及运行该模拟执行脚本的运行结束时间,从而实现目标节点对对应的目标资源异常场景的运行。

[0062] 其中,目标资源异常场景的运行数据包括运行开始时间和运行结束时间,或者也可以为运行开始时间和运行维持时长。这种情况下,基于目标资源异常场景的运行数据向目标节点发送资源异常运行指令的具体实施方式可以为:检测当前的系统时间,当检测到系统时间到达目标资源异常场景的运行开始时间时,向目标节点发送资源异常运行指令,该资源异常运行指令用于指示运行模拟执行脚本以及运行该模拟执行脚本的运行结束时间(例如2020-9-2 15:25),或者运行该模拟执行脚本的维持时长(例如维持5分钟)。

[0063] 其中,在目标节点运行与目标资源场景关联的模拟执行脚本的过程中,目标节点可以将运行日志记录到目标节点的指定目录下,后续测试平台对应的服务器可以从各目标节点的指定目录下获取各自对应的运行日志。

[0064] 或者,在目标节点运行与目标资源场景关联的模拟执行脚本的过程中,也可以由测试平台对应的服务器监控目标节点的运行状态,生成针对该目标节点的运行日志,并将该运行日志写入目标节点对应的指定目录下。

[0065] S303:触发测试用例执行。具体实现中,测试平台对应服务器可以获取测试用例,并触发区块链网络中的各节点执行测试用例。该测试用例为一种正常用例,包括数据交易记录、区块生成、区块上链等。

[0066] 本申请实施例中,可以在测试平台对应服务器依照 workflow 触发区块链网络中的目标节点运行对应的目标资源异常场景的过程中,触发测试用例执行,使得区块链网络中的目标节点在各自对应的目标资源异常场景下执行测试用例,并向测试平台对应的服务器返回对测试用例的执行结果,从而自动化实现对区块链网络的资源异常测试。

[0067] S304:测试平台对应服务器获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。具体实现中,测试平台对应的服务器生成对区块链网络的资源异常测试结果之后,可以将该资源异常测试结果下发至客户端,由客户端输出该资源异常测试结果,测试人员可以通过客户端查看资源异常测试结果定位问题。

[0068] 采用上述方式,一方面,可以精准模拟多种资源交叉异常场景,包括组合资源异常场景、时序资源异常场景;另一方面,在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

[0069] 请参见图4,是本申请实施例提供的一种区块链网络测试方法的流程示意图。该区块链网络测试方法可以由上述所提及的测试平台对应的服务器执行,请参见图4,该区块链网络测试方法可包括以下步骤S401-S404:

[0070] S401:接收针对区块链网络的资源异常测试请求,该资源异常测试请求包括用于对区块链网络进行资源异常测试的工作流,该区块链网络包括至少一个节点。

[0071] S402:依照工作流触发区块链网络中的目标节点运行对应的目标资源异常场景,目标节点为至少一个节点中的一个或者多个。

[0072] 其中,工作流包括目标资源异常场景的运行数据和目标节点的地址信息。在一个实施例中,当目标节点的数量为1时,目标节点对应的目标资源异常场景包括多个不同时序运行的资源异常场景,这种情况下,工作流可以包括目标节点对应的多个不同时序运行的资源异常场景的运行数据和目标节点的地址信息,该运行数据包括不同时序的资源异常场景的运行开始时间和运行结束时间,服务器可以基于该工作流模拟单节点多时序的时序资源异常场景,示例性地,该工作流可以参见图5。示例性地,假设工作流用于模拟单节点多时序的时序资源异常场景,目标节点为节点A,目标节点对应目标资源异常场景包括3个不同时序运行的资源异常场景,该目标资源异常场景包括的多个不同时序运行的资源异常场景以及运行数据如表1-1所示。从表1-1可以看出,单节点多时序的时序资源异常场景下,目标节点运行各自对应的目标资源异常场景需要考虑时序问题,也即各目标资源异常场景的运行时间不可以重合,存在运行的间隔时间。

[0073] 表1-1

[0074]

目标资源异常场景	目标节点	运行开始时间	运行结束时间
磁盘满载	节点A	2020-5-29 11:25:00	2020-5-29 11:30:00
网络延迟	节点A	2020-5-29 11:35:00	2020-5-29 11:40:00
CPU负载50%	节点A	2020-5-29 11:45:00	2020-5-29 11:50:00

[0075] 或者,在另一个实施例中,当目标节点的数量为 n (n 为大于1的整数)时, n 个目标节点中任一目标节点对应的目标资源异常场景包括一个资源异常场景或者多个不同时序运行的资源异常场景(即单节点多时序的时序资源异常场景)。

[0076] 可以理解的是, n 个目标节点中各目标节点运行对应目标资源异常场景的时序可以相同或者不同。本申请实施例中,对于各目标节点运行对应目标资源异常场景时无需考虑时序的情况,可以称为组合资源异常场景。在组合资源异常场景下,各目标节点运行对应目标资源异常场景的运行时间可以重合,无需存在运行的间隔时间。这种情况下,服务器可以基于工作流模拟组合资源异常场景。示例性地,假设工作流用于模拟组合资源异常场景, n 为3,区块链网络中目标节点包括节点A、节点B和节点C,各目标节点各自对应的目标资源异常场景,以及各目标资源异常场景的运行数据如表1-2所示。从表1-2可以看出,组合资源异常场景下,各目标节点运行各自对应的目标资源异常场景无需考虑时序问题,也即各目

标资源异常场景的运行时间可以重合。

[0077] 表1-2

目标资源异常场景	目标节点	运行开始时间	运行结束时间
磁盘满载	节点A	2020-5-29 11:25:00	2020-5-29 11:30:00
网络延迟	节点B	2020-5-29 11:25:00	2020-5-29 11:40:00
CPU负载50%	节点C	2020-5-29 11:35:00	2020-5-29 11:40:00

[0079] 或者,本申请实施例中,对于各目标节点运行对应目标资源异常场景时考虑时序的情况,可以称为多节点多时序的时序资源异常场景。在多节点多时序的时序资源异常场景下,各目标节点运行各自对应目标资源异常场景的运行时间不可以重合,存在运行的间隔时间。这种情况下,服务器可以基于 workflow 模拟多节点多时序的时序资源异常场景。示例性地,假设 workflow 用于模拟多节点多时序的时序资源异常场景,n为3,区块链网络中目标节点包括节点A、节点B和节点C,各目标节点各自对应的目标资源异常场景,以及各目标资源异常场景的运行数据如表1-3所示。从表1-3可以看出,多节点多时序的时序资源异常场景下,各目标节点运行各自对应的目标资源异常场景需要考虑时序问题,也即各目标资源异常场景的运行时间不可以重合。

[0080] 表1-3

目标资源异常场景	目标节点	运行开始时间	运行结束时间
磁盘满载	节点A	2020-5-29 11:25:00	2020-5-29 11:30:00
网络延迟	节点B	2020-5-29 11:35:00	2020-5-29 11:40:00
CPU负载50%	节点C	2020-5-29 11:45:00	2020-5-29 11:50:00

[0082] 在一个实施例中,一个单一的资源异常场景预先配置有一个模拟执行脚本,例如磁盘满载的资源异常场景对应模拟执行脚本1、网络延迟的资源异常场景对应模拟执行脚本2、CPU负载50%的资源异常场景对应模拟执行脚本3等等。这种情况下,上述步骤S402的具体实施方式可以为:获取与目标资源异常场景关联的模拟执行脚本,依照目标节点的地址信息向目标节点发送模拟执行脚本,基于目标资源异常场景的运行数据向目标节点发送资源异常运行指令,该资源异常运行指令用于指示目标节点运行该模拟执行脚本,并且资源异常指令包括运行结束时间。

[0083] 其中,目标资源异常场景的运行数据包括目标资源异常场景的运行开始时间和运行结束时间,资源异常运行指令是在检测到系统时间到达运行开始时间时发送的。

[0084] 具体实现中,目标资源异常场景可以为多个,假设目标节点为节点A,workflow 用于模拟单节点多时序的时序资源异常场景,workflow 中包括的各目标资源异常场景和各目标资源异常场景的运行数据如表1-1所示,磁盘满载的资源异常场景对应模拟执行脚本1、网络延迟的资源异常场景对应模拟执行脚本2、CPU负载50%的资源异常场景对应模拟执行脚本3。这种情况下,服务器可以解析 workflow,依照节点A的地址信息向节点A发送模拟执行脚本1(即与磁盘满载的资源异常场景关联的模拟执行脚本)、模拟执行脚本2(即与网络延迟的资源异常场景关联的模拟执行脚本)和模拟执行脚本3(即与CPU负载50%的资源异常场景关联的模拟执行脚本)。进一步地,在服务器检测到系统时间到达2020-5-29 11:25:00(即磁盘满载的资源异常场景的运行开始时间)时,可以向节点A发送包括运行结束时间“2020-5-29 11:30:00”和模拟执行脚本1的标识的资源异常运行指令,该资源异常运行指令用于指

示节点A运行模拟执行脚本1,以及运行结束时间为2020-5-29 11:30:00。进一步地,节点可以基于该资源异常运行指令的指示运行模拟执行脚本1,并在系统时间到达运行结束时间“2020-5-29 11:30:00”时,停止运行模拟执行脚本1,且将磁盘恢复至运行模拟执行脚本1之前的状态。例如,运行模拟执行脚本1之前,节点A的磁盘占用率为50%,运行模拟执行脚本1后磁盘满载,那么,当节点A检测到系统时间到达运行结束时间“2020-5-29 11:30:00”后,停止运行模拟执行脚本1,且将磁盘的占用率恢复至之前的50%。

[0085] 与运行模拟执行脚本1相似地,当服务器检测到系统时间到达2020-5-29 11:35:00时,可以向节点A发送包括运行结束时间“2020-5-29 11:40:00”和模拟执行脚本2的标识的资源异常运行指令,该资源异常运行指令用于指示节点A运行模拟执行脚本2,以及运行结束时间为2020-5-29 11:40:00。进一步地,节点可以基于该资源异常运行指令的指示运行模拟执行脚本2,并在系统时间到达运行结束时间“2020-5-29 11:40:00”时,停止运行模拟执行脚本1,且将网络恢复至运行模拟执行脚本2之前的状态。

[0086] 依次类推,当服务器检测到系统时间到达2020-5-29 11:45:00时,可以向节点A发送包括运行结束时间“2020-5-29 11:50:00”和模拟执行脚本3的标识的资源异常运行指令,该资源异常运行指令用于指示节点A运行模拟执行脚本3,以及运行结束时间为2020-5-29 11:50:00。进一步地,节点可以基于该资源异常运行指令的指示运行模拟执行脚本3,并在系统时间到达运行结束时间“2020-5-29 11:50:00”时,停止运行模拟执行脚本1,且将CPU负载恢复至运行模拟执行脚本3之前的状态。

[0087] 或者,在另一个实施例中,上述与目标资源异常场景关联的模拟执行脚本可以携带在资源异常运行指令中,服务器可以在检测到系统时间到达目标资源场景的运行开始时,依照目标节点的地址信息向目标节点发送携带有模拟执行脚本的资源异常运行指令,该资源异常运行指令用于指示目标节点运行自身携带的模拟执行脚本,并且资源异常指令包括运行结束时间。

[0088] S403:获取测试用例,并触发区块链网络中的各节点执行该测试用例。

[0089] S404:获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。

[0090] 在一个实施例中,测试平台预先部署有多个用于对区块链网络进行正常测试的测试用例,服务器可以从预先部署的多个测试用例中获取至少一个测试用例,并基于获取的至少一个测试用例触发区块链网络中的各节点执行测试用例。其中,该测试用例可以包括执行类(例如交易处理)、查询类(例如目标数据查询)、部署类(例如智能合约部署)、上链类(例如区块上链)等类型的测试用例。

[0091] 在一个实施例中,测试平台对应的服务器可以为区块链网络中的一个节点,如图2b所示,服务器获取到测试用例之后,可以将测试用例在区块链网络中进行广播,区块链网络中的各个节点接收到测试用例之后,可以执行测试用例,并返回针对测试用例的执行结果。服务器可以获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

[0092] 其中,上述步骤S404的具体实施方式可以为:根据各节点对测试用例的执行结果确定区块链网络中运行异常的节点数量,依照区块链网络中运行异常的节点数量对区块链

网络进行拜占庭共识分析,得到分析结果,根据分析结果生成对区块链网络的资源异常测试结果。

[0093] 具体实现中,若区块链网络中异常的节点数量大于拜占庭共识机制指示的数量阈值,则确定区块链网络不满足拜占庭共识,测试用例执行失败,服务器可以生成测试用例执行失败的分析结果。或者,若区块链网络中异常的节点数量小于或者等于拜占庭共识机制指示的数量阈值,则确定区块链网络满足拜占庭共识,测试用例执行成功,服务器可以生成测试用例执行成功的分析结果。

[0094] 示例性地,假设区块链网络包括节点A、节点B、节点C和节点D,如果工作流用于指示模拟单节点A进程退出,导致A节点服务不可用,但是BFT共识机制允许在4个节点构成的区块链网络中有一个容错节点(也即数量阈值为1),所以整个区块链服务依然正常运行,自动化用例执行成功。但是如果模拟节点A和B同时进程退出,其节点上的区块链服务同时不可用,这样就不满足BFT共识,最终导致整个区块链网络不可用,自动化用例执行失败。

[0095] 本申请实施例中,服务器依照工作流模拟的资源交叉异常场景(组合资源异常场景和时序资源异常场景)可以包括满足BFT共识的资源交叉异常场景,和不满足BFT共识的资源交叉异常场景。

[0096] 在一个实施例中,资源异常测试结果是根据分析结果和区块链网络中目标节点运行对应的目标资源异常场景的运行日志确定的,该资源异常测试结果可以为资源异常测试报告,服务器可以通过测试平台对应的客户端展示资源异常测试报告。示例性地,该资源异常测试报告可以如图6所示。

[0097] 其中,上述运行日志的获取方式为:服务器可以在依照工作流触发区块链网络中的目标节点运行对应的目标资源异常场景之后,在目标节点运行对应的目标资源异常场景的过程中,监控区块链网络中的目标节点运行对应的目标资源异常场景的运行状态,基于监控结果生成目标节点运行对应的目标资源异常场景的运行日志。

[0098] 在一个实施例中,在目标节点运行对应的目标资源异常场景的过程中,测试平台对应的服务器还可以检测目标节点的资源,实时或者周期性获取目标节点的资源动态变化信息,基于该资源动态变化信息生成目标节点的节点资源监控视图,并通过测试平台对应的客户端展示目标节点的节点资源监控视图,测试人员可以结合该节点资源监控视图查看目标节点运行目标资源异常场景的结果。例如通过节点资源监控视图查看目标节点的CPU是否负载50%,网络是否延迟、磁盘是否满载等等。示例性地,假设目标节点为节点A,该目标节点的资源监控视图可以如图7a所示。

[0099] 在一个实施例中,触发区块链网络中的各节点执行测试用例之后,还可以在测试用例执行过程中,对区块链网络中的各节点进行业务监控,实时或者周期性采集各节点与监控指标关联的数据,该监控指标包括区块高度、交易数量、最大响应时间和最小响应时间等等。进一步地,服务器可以基于该与监控指标关联的数据生成区块链网络中各节点的业务监控视图,并通过测试平台对应的客户端展示业务监控视图。测试人员可以结合该业务监控视图查看区块链网络中各节点实施目标资源异常场景后,执行测试用例过程中运行状态。示例性地,该业务监控视图可以如图7b所示。

[0100] 本申请实施例中,测试平台对应的服务器可以接收携带有工作流的资源异常测试请求,依照工作流触发区块链网络中的目标节点运行对应的目标资源异常场景,获取测试

用例,并触发区块链网络中的各节点执行该测试用例。进一步地,可以获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。采用这样的方式,可以模拟运行资源异常场景,并在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

[0101] 为更好的理解上述测试方法,下面结合实际测试过程所涉及的一些具体步骤以及实现方式进行举例说明。

[0102] 可以理解的是,一个区块链网络可以视为一个区块链产品,在区块链产品迭代周期内,研发工程师将开发好的区块链产品提测到REQ平台(即软件研发提测流程管理平台),之后触发测试执行。测试人员在完成基本的功能测试之后,会进一步进行资源异常测试。本申请实施例中,资源异常测试主要用于模拟资源交叉异常场景(包括组合资源异常场景和时序资源异常场景),并在对区块链网络中目标节点实施对应的资源异常场景后,触发测试用例执行,使得区块链网络中的目标节点在各自对应的目标资源异常场景下执行测试用例,并向测试平台对应的服务器返回对测试用例的执行结果,从而自动化实现对区块链网络的资源异常测试。

[0103] 参见图8,区块链产品的生成至发布的具体流程可以分为区块链产品提测阶段、资源异常测试阶段和发布阶段,其中,区块链产品提测阶段具体可以包括:(1)开发区块链产品。(2)将开发好的区块链产品提测到REQ平台。(3)对区块链产品进行审核。(4)审核通过,进入资源异常测试阶段。资源异常测试阶段具体可以包括:(5)提交用于对区块链产品进行资源异常测试的工作流。(6)登记区块链产品中各节点的节点信息。(7)依照工作流触发资源异常场景运行。(8)触发测试用例执行,并基于测试用例对区块链网络进行测试。发布阶段具体可以包括:(9)分析BFT共识结果。(10)生成测试报告。(11)发布上线。其中,上述节点信息可以包括节点的IP、端口、用户名和用户密码等等。

[0104] 在一个实施例中,为了更好的理解本申请实施例提出的测试平台,可以参见图9所示的测试平台的架构图,该测试平台可以包括展示层、通信层、应用层、工具层和存储层。其中,展示层为用户交互层,是测试平台的前端入口,用户可以在展示层可触发故障(即资源异常)执行,展示层的功能包括集群管理、节点管理、故障管理(即资源异常场景管理)、任务管理、流水日志、故障定义(即资源异常场景定义)和报告展示等。示例性地,集群管理可以为管理测试环境下的集群或者生产环境下的集群,一个集群可以理解为一个区块链网络;节点管理可以为管理各区块链网络中的节点,具体可以用于登记区块链网络中各节点的节点信息;故障管理可以为配置各种资源异常场景,该资源异常场景可以为单一的资源异常场景,例如CPU、磁盘、进程、网络异常等等,也可以为资源交叉异常场景(包括组合资源异常和时序资源异常);任务管理可以为管理执行任务的一次执行流;流水日志可以为展示区块链网络中各目标节点模拟运行各种资源异常场景的运行日志;故障定义可以为编写用于模拟多种资源异常场景的工作流;报告展示可以为展示资源异常测试报告。

[0105] 通信层是承上启下的中间件,它提供调用服务的API接口,负责将前端发送的指令传给后台,并回调上报处理的结果。应用层包括故障服务(Server)和故障代理(Agent),故障Server接收来自上层的请求,下发与目标资源异常场景关联的模拟执行脚本做初始化配

置,并向故障Agent发送执行指令(即资源异常运行指令);故障Agent负责直接运行模拟执行脚本,记录实施目标资源异常场景后的运行日志,并上报处理结果。工具层包含各种执行脚本和辅助工具(例如定时恢复、循环执行等等。存储层主要包含文件存储和数据库存储,它负责存储运行日志、资源异常测试报告等等。

[0106] 示例性地,通过图9所示的测试平台,对区块链网络进行资源异常测试的执行框架可以如图10所示。测试平台可以读取用于对区块链网络进行资源异常测试的工作流,依照运行的先后顺序依序从工作流中读取一个任务(即待运行的目标资源异常场景),并将读取到的任务依次放入任务队列中,调度器依序调度任务队列中的各任务,通过故障服务向各任务对应的目标节点下发用于执行任务的模拟执行脚本,各目标节点通过故障代理直接运行模拟执行脚本,从而实现资源异常场景的模拟。在对区块链网络中的目标节点实施资源异常场景后,可以触发区块链网络中的各节点执行测试用例,使得区块链网络中的目标节点在各自对应的目标资源异常场景下执行测试用例,并向测试平台返回对测试用例的执行结果,从而自动化实现对区块链网络的资源异常测试。

[0107] 本申请实施例还提供了一种计算机存储介质,该计算机存储介质中存储有程序指令,该程序指令被执行时,用于实现上述实施例中描述的相应方法。

[0108] 再请参见图11,是本申请实施例的一种区块链网络测试装置的结构示意图,本申请实施例的区块链网络测试装置可以设置在上述服务器(例如测试平台对应的服务器)中,也可以为运行于服务器中的一个计算机程序(包括程序代码)。

[0109] 本申请实施例的装置的一个实现方式中,装置包括如下结构。

[0110] 通信模块100,用于接收针对区块链网络的资源异常测试请求,所述资源异常测试请求包括用于对所述区块链网络进行资源异常测试的工作流,所述区块链网络包括至少一个节点;

[0111] 处理模块101,用于依照所述工作流触发所述区块链网络中的目标节点运行对应的目标资源异常场景,所述目标节点为所述至少一个节点中的一个或者多个;

[0112] 所述处理模块101,还用于获取测试用例,并触发所述区块链网络中的各节点执行所述测试用例;

[0113] 所述处理模块101,还用于获取所述各节点对所述测试用例的执行结果,并基于所述执行结果生成对所述区块链网络的资源异常测试结果。

[0114] 在一个实施例中,当所述目标节点的数量为1时,所述目标节点对应的目标资源异常场景包括多个不同时序运行的资源异常场景;

[0115] 当所述目标节点的数量为n时,所述n个目标节点中任一目标节点对应的目标资源异常场景包括一个资源异常场景或者多个不同时序运行的资源异常场景,所述n为大于1的整数。

[0116] 在一个实施例中,所述工作流包括所述目标资源异常场景的运行数据和所述目标节点的地址信息;处理模块101,具体用于:

[0117] 获取与所述目标资源异常场景关联的模拟执行脚本;

[0118] 依照所述目标节点的地址信息向所述目标节点发送所述模拟执行脚本;

[0119] 基于所述目标资源异常场景的运行数据向所述目标节点发送资源异常运行指令,所述资源异常运行指令用于指示所述目标节点运行所述模拟执行脚本,并且所述资源异常

指令包括运行结束时间。

[0120] 在一个实施例中,所述目标资源异常场景的运行数据包括所述目标资源异常场景的运行开始时间和运行结束时间,所述资源异常运行指令是在检测到系统时间到达所述运行开始时间时发送的。

[0121] 在一个实施例中,处理模块101,还具体用于:

[0122] 根据所述各节点对所述测试用例的执行结果确定所述区块链网络中运行异常的节点数量;

[0123] 依照所述区块链网络中运行异常的节点数量对所述区块链网络进行拜占庭共识分析,得到分析结果;

[0124] 根据所述分析结果生成对所述区块链网络的资源异常测试结果。

[0125] 在一个实施例中,所述资源异常测试结果是根据所述分析结果和所述区块链网络中目标节点运行对应的目标资源异常场景的运行日志确定的。

[0126] 在一个实施例中,所述依照所述 workflow 触发所述区块链网络中的目标节点运行对应的目标资源异常场景之后,所述处理模块101,还用于:

[0127] 监控所述区块链网络中的目标节点运行对应的目标资源异常场景的运行状态;

[0128] 基于监控结果生成目标节点运行对应的目标资源异常场景的运行日志。

[0129] 在一个实施例中,所述触发所述区块链网络中的各节点执行测试用例,所述处理模块101,还具体用于:

[0130] 将所述测试用例在所述区块链网络中进行广播,以使得所述区块链网络中的各个节点接收所述测试用例,并返回针对所述测试用例的执行结果。

[0131] 在本申请实施例中,上述各个模块的具体实现可参考前述各个附图所对应的实施例中相关内容的描述。

[0132] 本申请实施例中的区块链网络测试装置可接收携带有工作流的资源异常测试请求,依照 workflow 触发区块链网络中的目标节点运行对应的目标资源异常场景,获取测试用例,并触发区块链网络中的各节点执行该测试用例。进一步地,可以获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。采用这样的方式,可以针对区块链网络模拟运行资源异常场景,并在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

[0133] 再请参见图12,是本申请实施例的一种服务器的结构示意图,该服务器可以指上述测试平台对应的服务器。本申请实施例的服务器包括供电模块等结构,并包括处理器80、存储装置81和通信接口82。处理器80、存储装置81和通信接口82之间可以交互数据,由处理器80实现相应的区块链网络测试功能。

[0134] 存储装置81可以包括易失性存储器(volatile memory),例如随机存取存储器(random-access memory, RAM);存储装置81也可以包括非易失性存储器(non-volatile memory),例如快闪存储器(flash memory),固态硬盘(solid-state drive, SSD)等;存储装置81还可以包括上述种类的存储器的组合。

[0135] 处理器80可以是中央处理器80(central processing unit, CPU)。在一个实施例

中,处理器80还可以是图形处理器80(Graphics Processing Unit,GPU)。处理器80也可以是由CPU和GPU的组合。在服务器中,可以根据需要包括多个CPU和GPU进行相应的区块链网络测试。

[0136] 在一个实施例中,存储装置81用于存储程序指令。处理器80可以调用程序指令,实现如本申请实施例中上述涉及的各种方法。

[0137] 在第一个可能的实施方式中,服务器的处理器80,调用存储装置81中存储的程序指令,用于:

[0138] 通过通信接口82接收针对区块链网络的资源异常测试请求,所述资源异常测试请求包括用于对所述区块链网络进行资源异常测试的工作流,所述区块链网络包括至少一个节点;

[0139] 依照所述工作流触发所述区块链网络中的目标节点运行对应的目标资源异常场景,所述目标节点为所述至少一个节点中的一个或者多个;

[0140] 获取测试用例,并触发所述区块链网络中的各节点执行所述测试用例;

[0141] 获取所述各节点对所述测试用例的执行结果,并基于所述执行结果生成对所述区块链网络的资源异常测试结果。

[0142] 在一个实施例中,当所述目标节点的数量为1时,所述目标节点对应的目标资源异常场景包括多个不同时序运行的资源异常场景;

[0143] 当所述目标节点的数量为n时,所述n个目标节点中任一目标节点对应的目标资源异常场景包括一个资源异常场景或者多个不同时序运行的资源异常场景,所述n为大于1的整数。

[0144] 在一个实施例中,所述工作流包括所述目标资源异常场景的运行数据和所述目标节点的地址信息;处理器80,具体用于:

[0145] 获取与所述目标资源异常场景关联的模拟执行脚本;

[0146] 依照所述目标节点的地址信息向所述目标节点发送所述模拟执行脚本;

[0147] 基于所述目标资源异常场景的运行数据向所述目标节点发送资源异常运行指令,所述资源异常运行指令用于指示所述目标节点运行所述模拟执行脚本,并且所述资源异常指令包括运行结束时间。

[0148] 在一个实施例中,所述目标资源异常场景的运行数据包括所述目标资源异常场景的运行开始时间和运行结束时间,所述资源异常运行指令是在检测到系统时间到达所述运行开始时间时发送的。

[0149] 在一个实施例中,处理器80,还具体用于:

[0150] 根据所述各节点对所述测试用例的执行结果确定所述区块链网络中运行异常的节点数量;

[0151] 依照所述区块链网络中运行异常的节点数量对所述区块链网络进行拜占庭共识分析,得到分析结果;

[0152] 根据所述分析结果生成对所述区块链网络的资源异常测试结果。

[0153] 在一个实施例中,所述资源异常测试结果是依据所述分析结果和所述区块链网络中目标节点运行对应的目标资源异常场景的运行日志确定的。

[0154] 在一个实施例中,所述依照所述工作流触发所述区块链网络中的目标节点运行对

应的目标资源异常场景之后,所述处理器80,还用于:

[0155] 监控所述区块链网络中的目标节点运行对应的目标资源异常场景的运行状态;

[0156] 基于监控结果生成目标节点运行对应的目标资源异常场景的运行日志。

[0157] 在一个实施例中,所述触发所述区块链网络中的各节点执行测试用例,所述处理器80,还具体用于:

[0158] 将所述测试用例在所述区块链网络中进行广播,以使得所述区块链网络中的各个节点接收所述测试用例,并返回针对所述测试用例的执行结果。

[0159] 在本申请实施例中,上述处理器80的具体实现可参考前述各个附图所对应的实施例中相关内容的描述。

[0160] 本申请实施例中的服务器可接收携带有工作流的资源异常测试请求,依照工作流触发区块链网络中的目标节点运行对应的目标资源异常场景,获取测试用例,并触发区块链网络中的各节点执行该测试用例。进一步地,可以获取各节点对测试用例的执行结果,并基于执行结果生成对区块链网络的资源异常测试结果。采用这样的方式,可以针对区块链网络模拟运行资源异常场景,并在资源异常场景实施后,配合测试用例,基于区块链网络中各节点对测试用例的执行结果生成对区块链网络的资源异常测试结果,从而自动化实现对区块链网络的资源异常测试,有利于提高对区块链网络进行资源异常测试的测试效率和测试准确度。

[0161] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所描述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random Access Memory,RAM)等。

[0162] 以上所揭露的仅为本申请的部分实施例而已,当然不能以此来限定本申请之权利范围,本领域普通技术人员可以理解实现上述实施例的全部或部分流程,并依本申请权利要求所作的等同变化,仍属于发明所涵盖的范围。

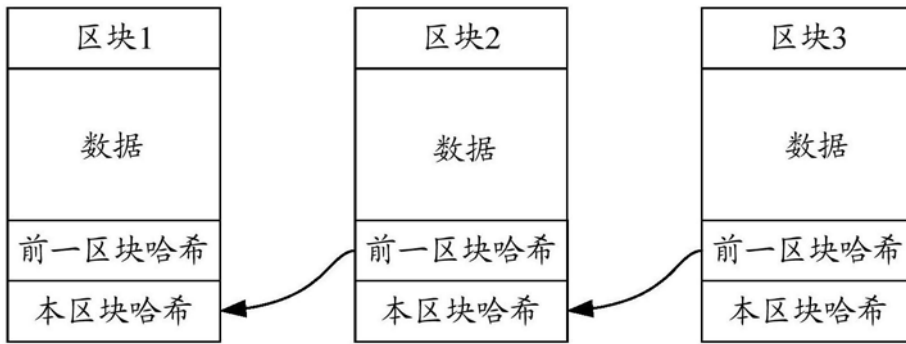


图1

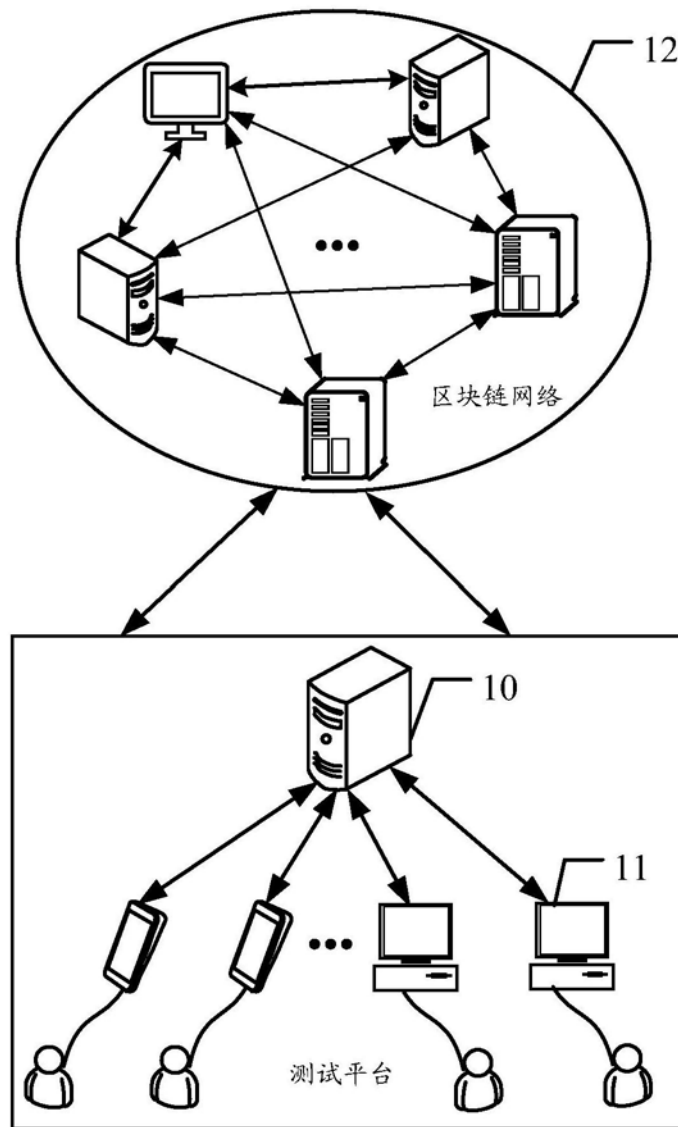


图2a

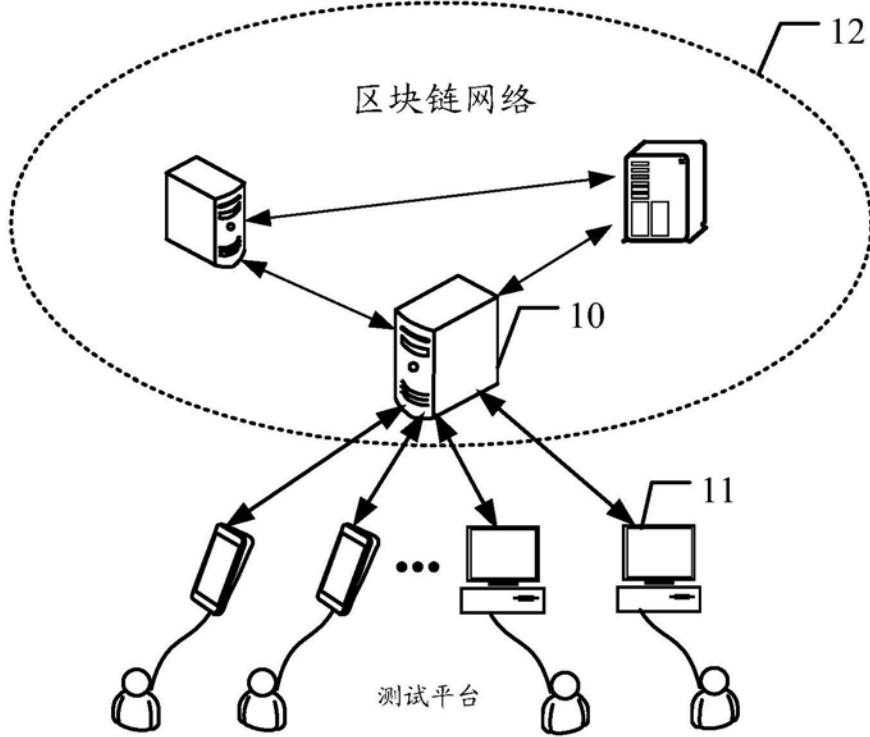


图2b

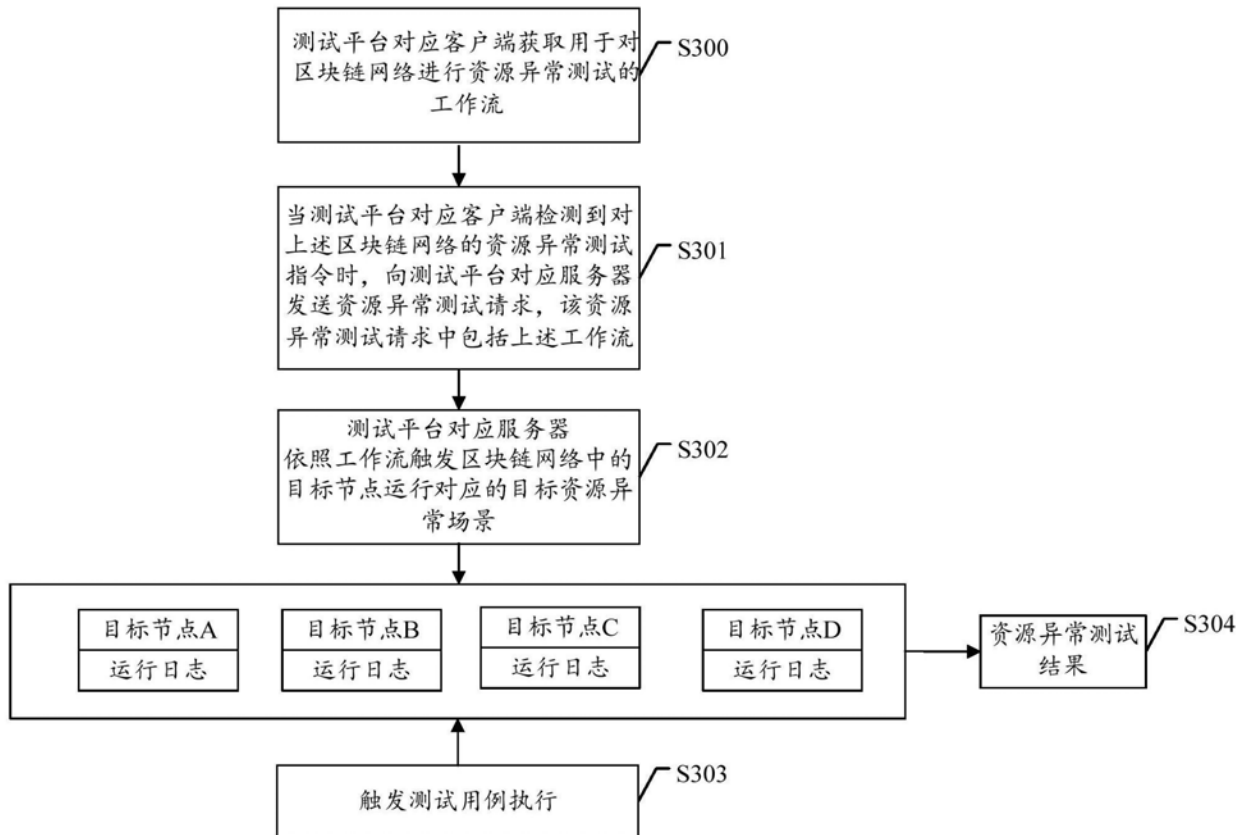


图3

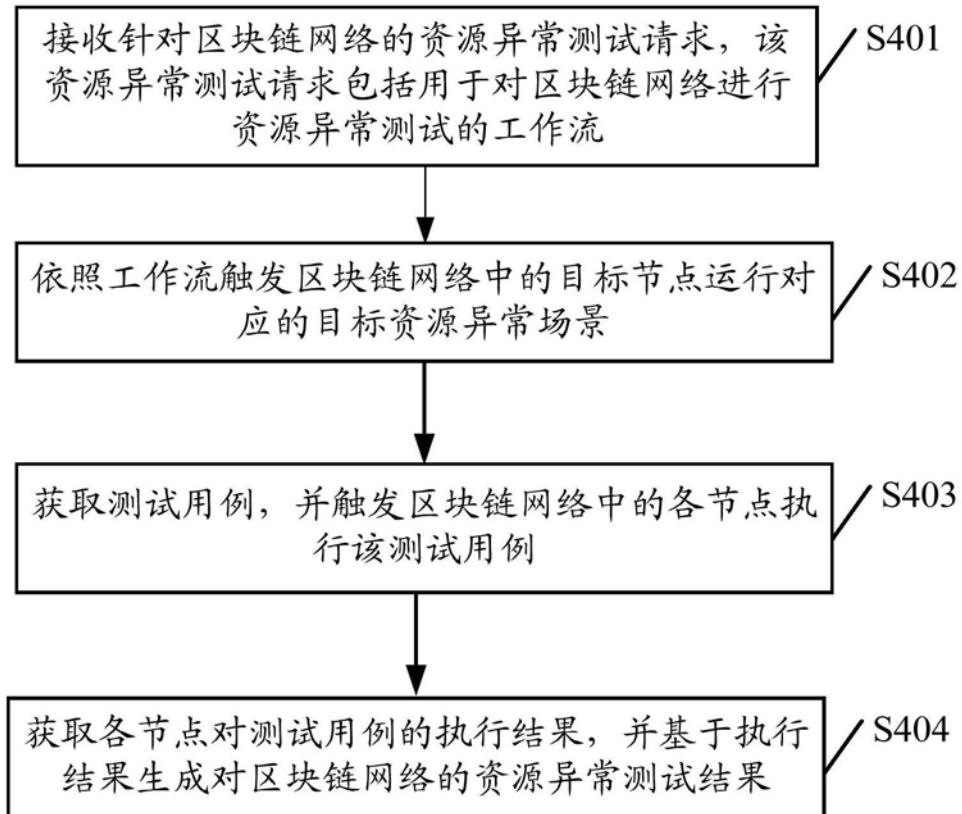


图4

```
{
  "workflows" :{
  {
    "target" : "network" ,
    "scope" : "****6:8080" ,
    "matcher" :{
      "time" : "300" ,
      "offset" : "100" ,
      "interface" : "eth1" ,
      "local-port" : "20020" ,
      "timeout" : "60"
    },
    "action" : "delay" ,
    "runtime" : "2020-5-29 11:45:00"
  },

  {
    "target" : "disk" ,
    "scope" : "****6:8080" ,
    "matcher" :{
      "path" : "/date/gopath/src/*****" ,
      "size" : "20" ,
      "timeout" : "60" ,
    },
    "action" : "fill" ,
    "runtime" : "2020-5-29 10:56:00"
  },
}
```

图5

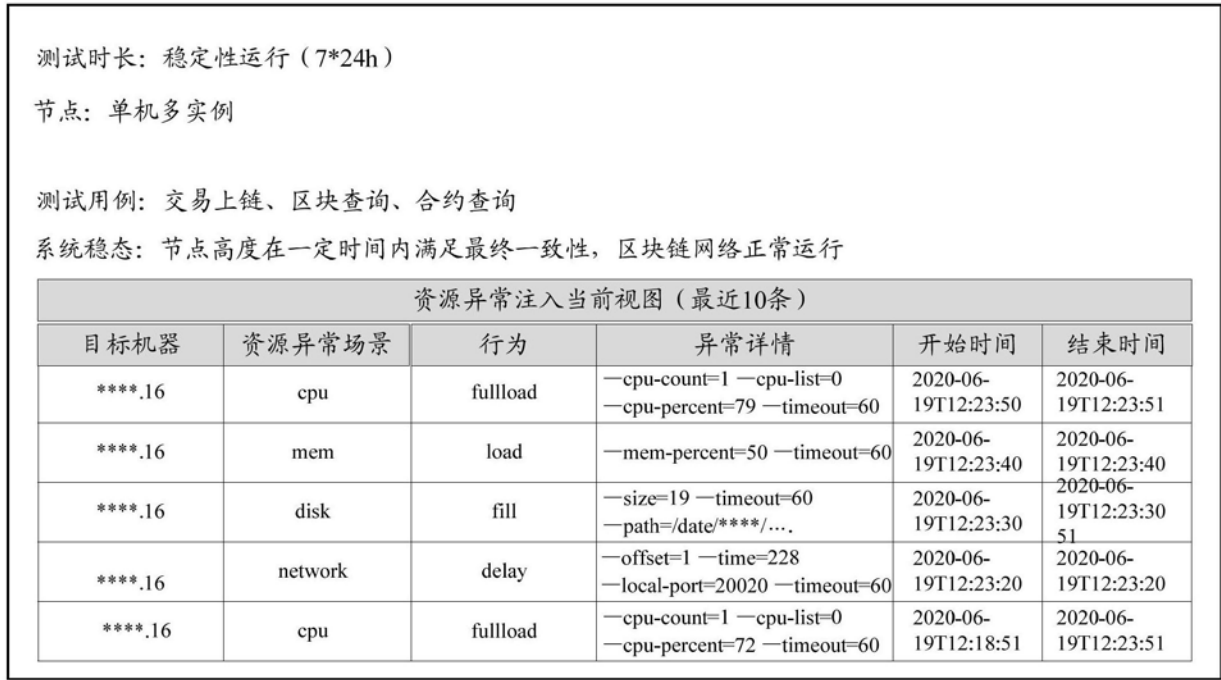


图6

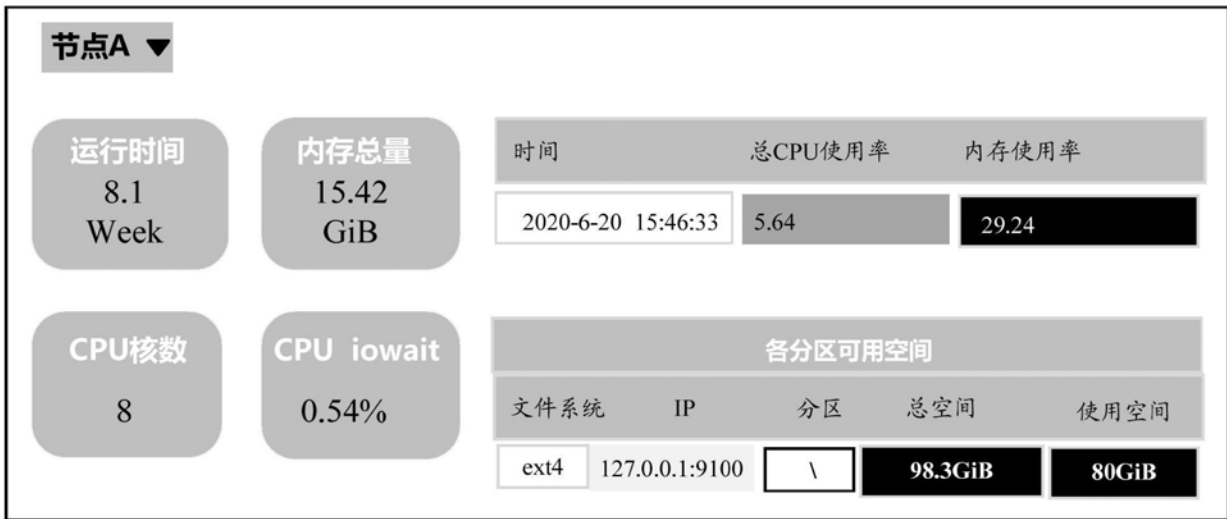


图7a

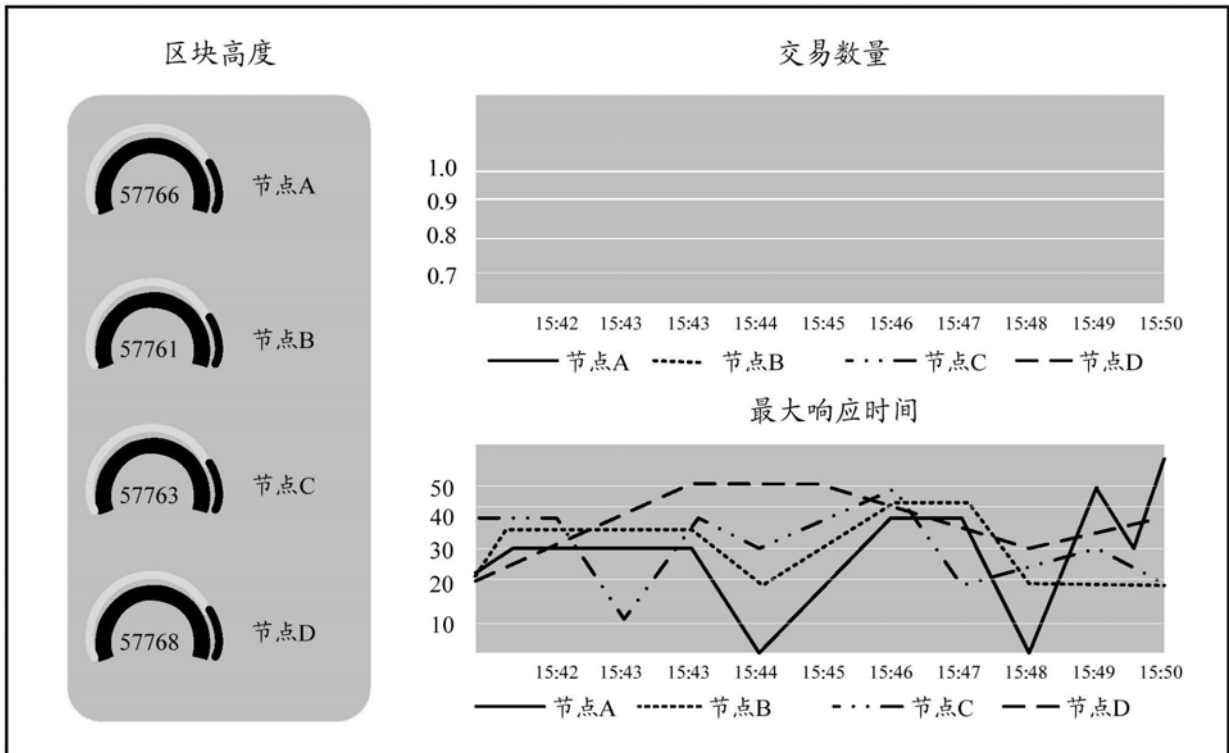


图7b

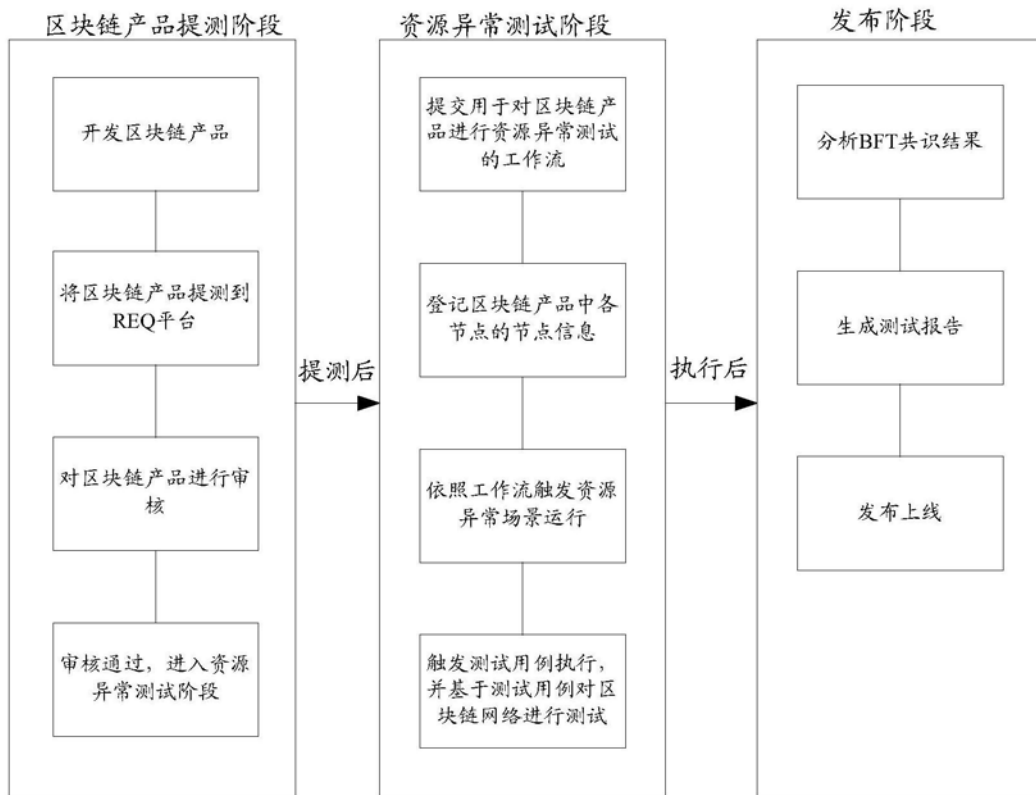


图8

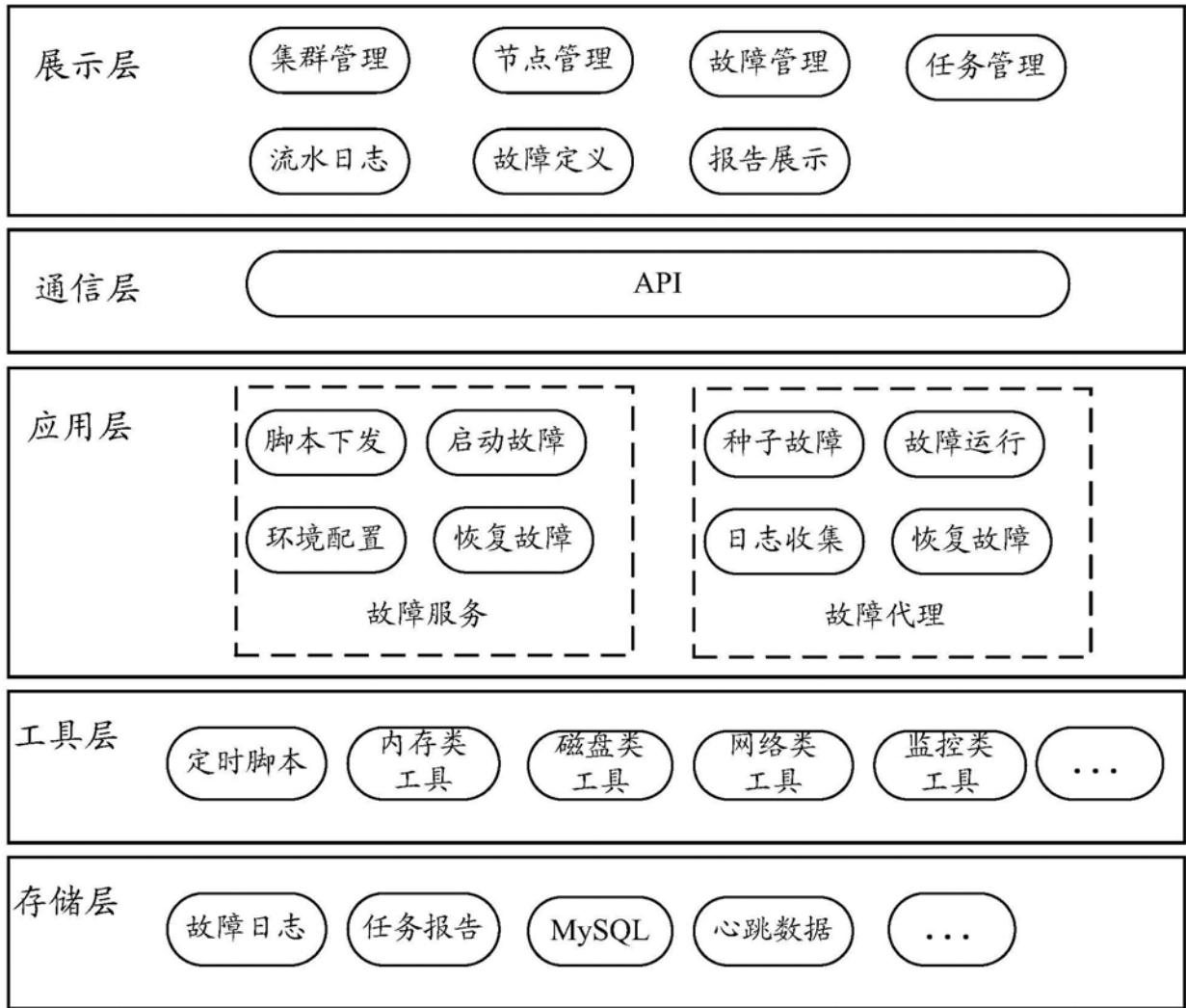


图9

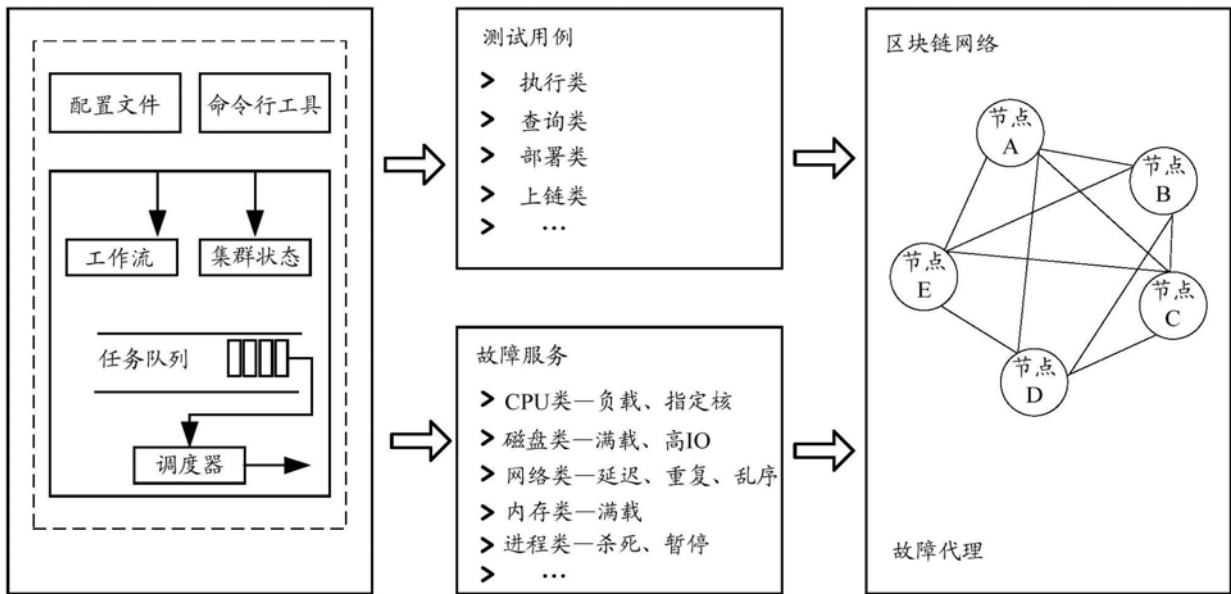


图10

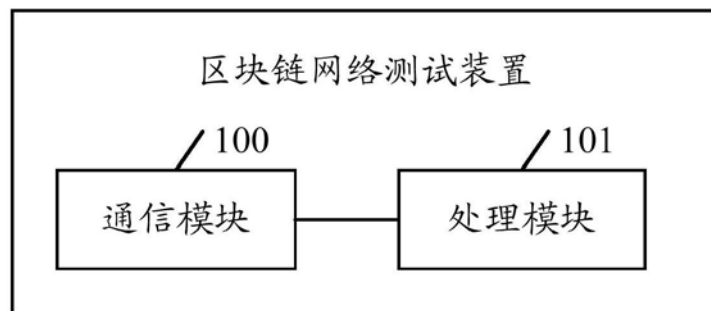


图11

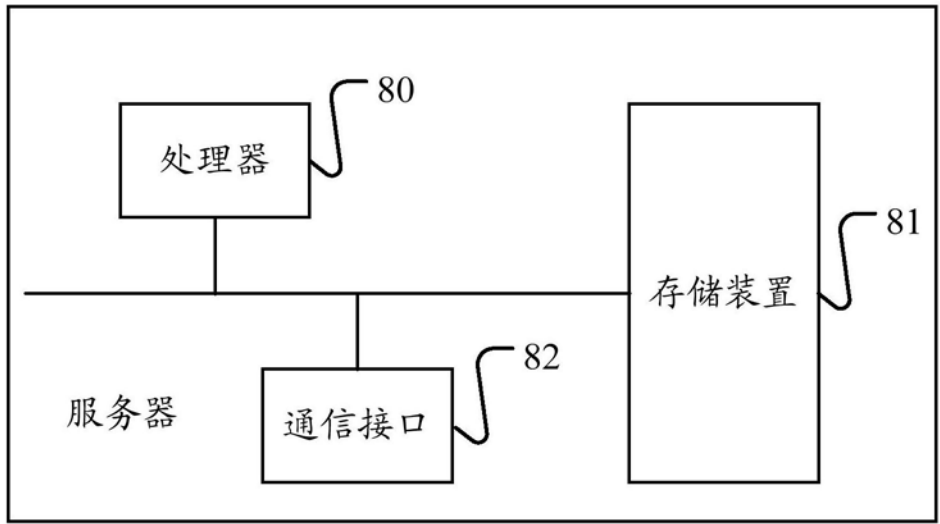


图12