



(12)发明专利申请

(10)申请公布号 CN 111258913 A

(43)申请公布日 2020.06.09

(21)申请号 202010102333.0

(22)申请日 2020.02.19

(71)申请人 深圳壹账通智能科技有限公司

地址 518052 广东省深圳市前海深港合作区前湾一路1号A栋201室

(72)发明人 刘璐 臧磊 单以磊 匡原 彭涛

(74)专利代理机构 北京英特普罗知识产权代理有限公司 11015

代理人 程超

(51)Int.Cl.

G06F 11/36(2006.01)

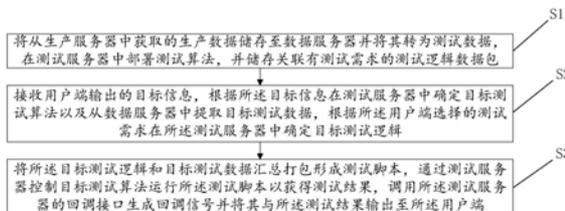
权利要求书2页 说明书12页 附图5页

(54)发明名称

算法自动测试方法、装置、计算机系统及可读存储介质

(57)摘要

本发明公开了算法自动测试方法、装置、计算机系统及可读存储介质,基于测试技术领域,包括:将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据,在测试服务器中部署测试算法,并储存关联有测试需求的测试逻辑数据包;接收用户端输出的目标信息,根据目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据,根据测试需求在测试服务器中确定目标测试逻辑;将目标测试逻辑和目标测试数据汇总打包形成测试脚本,控制目标测试算法运行测试脚本以获得测试结果,调用回调接口生成回调信号并将其与测试结果输出至用户端。本发明实现了自动对算法进行测试的效果,避免了当前测试的工作效率低、随意性大的问题。



1. 一种算法自动测试方法,其特征在于,包括:

将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据,在测试服务器中部署测试算法,并储存关联有测试需求的测试逻辑数据包;

接收用户端输出的目标信息,根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据,根据所述用户端选择的测试需求在所述测试服务器中确定目标测试逻辑;

将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本,通过测试服务器控制目标测试算法运行所述测试脚本以获得测试结果,调用所述测试服务器的回调接口生成回调信号并将其与所述测试结果输出至所述用户端。

2. 根据权利要求1所述的算法自动测试方法,其特征在于,所述在测试服务器中部署测试算法包括:

在所述测试服务器中储存具有测试信息的测试算法;

将所述测试算法部署在所述测试服务器中,并将所述测试算法在所述测试服务器中的部署路径载入所述测试信息中。

3. 根据权利要求1所述的算法自动测试方法,其特征在于,所述储存关联有测试需求的测试逻辑数据包包括:

将预生成的测试逻辑数据包储存至测试服务器,并获得所述测试逻辑数据包在测试服务器中的储存路径;其中,所述测试逻辑数据包是指记载有对测试数据进行测试的方法的数据包;

在测试服务器中设置逻辑表,将所述储存路径和测试需求写入逻辑表中使测试逻辑数据包与测试需求关联。

4. 根据权利要求1所述的算法自动测试方法,其特征在于,所述接收用户端输出的目标信息包括:

在所述测试服务器中创建前端节点和储存有测试算法及测试逻辑数据包的后端节点;

通过所述前端节点向用户端输出具有测试算法ID的算法对话框,并将所述用户端在所述算法对话框中选择并发送的测试算法ID设为目标算法ID;

通过所述后端节点根据前端节点发送的目标算法ID,向所述前端节点输出与所述目标算法ID匹配的测试版本ID;通过所述前端节点向所述用户端输出具有所述测试版本ID的版本对话框,并将所述用户端在所述版本对话框选择并发送的版本ID设为目标版本ID;

汇总所述目标算法ID和目标版本ID形成目标信息。

5. 根据权利要求4所述的算法自动测试方法,其特征在于,所述根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据包括:

通过所述前端节点根据所述目标信息在所述后端节点中确定目标测试算法;

通过所述后端节点从数据服务器中提取与所述目标信息匹配的测试数据,并将所述测试数据的编号通过前端节点以数据对话框的方式发送至所述用户端;

所述前端节点将所述用户端在数据对话框中选择并发送的编号转发至所述后端节点,通过所述后端节点提取该编号对应的测试数据并将其设为目标测试数据。

6. 根据权利要求4所述的算法自动测试方法,其特征在于,所述根据用户端选择的测试需求在所述测试服务器中确定目标测试逻辑包括:

通过所述前端节点向所述用户端输出具有测试需求的需求对话框,将所述用户端在需求对话框中选择并发送的测试需求设为目标测试需求;

通过所述前端节点在后端节点中将与所述目标测试需求关联的测试逻辑数据包设为目标测试逻辑。

7. 根据权利要求1所述的算法自动测试方法,其特征在于,所述通过所述测试服务器控制所述测试算法运行所述测试脚本以获得测试结果包括:

通过所述测试服务器提取所述测试脚本中的目标测试逻辑,判断获取所述目标测试逻辑的测试需求;

根据所述测试需求的种类通过测试服务器控制测试算法运行所述测试脚本以获得测试结果;其中,若所述测试需求的种类为冒烟测试,则通过测试算法运行所述测试脚本以获得冒烟测试结果;若所述测试需求的种类为算法响应时间测试,则通过测试算法运行所述测试脚本以获得响应测试结果;若所述测试需求的种类为性能测试,则通过测试算法运行测试脚本并启动采集程序,使所述采集程序采集所述测试服务器在运行测试脚本时的机器资源占用情况并生成性能测试结果。

8. 一种算法自动测试装置,其特征在于,包括:

测试预备模块,用于将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据,在测试服务器中部署测试算法,并储存关联有测试需求的测试逻辑数据包;

需求认定模块,用于接收用户端输出的目标信息,根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据,根据所述用户端选择的测试需求在所述测试服务器中确定目标测试逻辑;

测试执行模块,用于将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本,通过测试服务器控制目标测试算法运行所述测试脚本以获得测试结果,调用所述测试服务器的回调接口生成回调信号并将其与所述测试结果输出至所述用户端。

9. 一种计算机系统,其包括多个计算机设备,各计算机设备包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述多个计算机设备的处理器执行所述计算机程序时共同实现权利要求1至7任一项所述算法自动测试方法的步骤。

10. 一种计算机可读存储介质,其包括多个存储介质,各存储介质上存储有计算机程序,其特征在于,所述多个存储介质存储的所述计算机程序被处理器执行时共同实现权利要求1至7任一项所述算法自动测试方法的步骤。

算法自动测试方法、装置、计算机系统及可读存储介质

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种算法自动测试方法、装置、计算机系统及可读存储介质。

背景技术

[0002] 随着以深度学习为主流的人工智能应用越来越广泛,深度学习算法工程化后的规模和复杂度不断提高,如对现有深度学习算法接口进行功能扩展、性能优化、模型迭代等,需要大量的测试以及大量的测试数据来保证算法的性能;

[0003] 当前常用做法为采用手工测试的方式对算法进行测试,由于需要重复大量手动测试操作,因此造成工作效率低、随意性大、出错机会大、拉长开发周期的问题,进而无法保证需求、测试用例和测试结果一致性和准确性;又由于手动测试以及当前较为初级的自动化测试的测试资源监控不到位,并且无法对算法功能和性能要求实现覆盖性测试。

发明内容

[0004] 本发明的目的是提供一种算法自动测试方法、装置、计算机系统及可读存储介质,用于解决现有技术存在的因重复大量手动测试操作导致的工作效率低、随意性大、出错机会大、拉长开发周期的问题,以及无法保证需求、测试用例和测试结果一致性和准确性的问题。

[0005] 为实现上述目的,本发明提供一种算法自动测试方法,包括:

[0006] 将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据,在测试服务器中部署测试算法,并储存关联有测试需求的测试逻辑数据包;

[0007] 接收用户端输出的目标信息,根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据,根据所述用户端选择的测试需求在所述测试服务器中确定目标测试逻辑;

[0008] 将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本,通过测试服务器控制目标测试算法运行所述测试脚本以获得测试结果,调用所述测试服务器的回调接口生成回调信号并将其与所述测试结果输出至所述用户端。

[0009] 在上述方案中,所述在测试服务器中部署测试算法包括:

[0010] 在所述测试服务器中储存具有测试信息的测试算法;

[0011] 将所述测试算法部署在所述测试服务器中,并将所述测试算法在所述测试服务器中的部署路径载入所述测试信息中。

[0012] 在上述方案中,所述储存关联有测试需求的测试逻辑数据包包括:

[0013] 将预生成的测试逻辑数据包储存至测试服务器,并获得所述测试逻辑数据包在测试服务器中的储存路径;其中,所述测试逻辑数据包是指记载有对测试数据进行测试的方法的数据包;

[0014] 在测试服务器中设置逻辑表,将所述储存路径和测试需求写入逻辑表中使测试逻

辑数据包与测试需求关联。

[0015] 在上述方案中,所述接收用户端输出的目标信息包括:

[0016] 在所述测试服务器中创建前端节点和储存有测试算法及测试逻辑数据包的后端节点;

[0017] 通过所述前端节点向用户端输出具有测试算法ID的算法对话框,并将所述用户端在所述算法对话框中选择并发送的测试算法ID设为目标算法ID;

[0018] 通过所述后端节点根据前端节点发送的目标算法ID,向所述前端节点输出与所述目标算法ID匹配的测试版本ID;通过所述前端节点向所述用户端输出具有所述测试版本ID的版本对话框,并将所述用户端在所述版本对话框选择并发送的版本ID设为目标版本ID;

[0019] 汇总所述目标算法ID和目标版本ID形成目标信息。

[0020] 在上述方案中,所述根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据包括:

[0021] 通过所述前端节点根据所述目标信息在所述后端节点中确定目标测试算法;

[0022] 通过所述后端节点从数据服务器中提取与所述目标信息匹配的测试数据,并将所述测试数据的编号通过前端节点以数据对话框的方式发送至所述用户端;

[0023] 所述前端节点将所述用户端在数据对话框中选择并发送的编号转发至所述后端节点,通过所述后端节点提取该编号对应的测试数据并将其设为目标测试数据。

[0024] 在上述方案中,所述根据用户端的选择的测试需求在所述测试服务器中确定目标测试逻辑包括:

[0025] 通过所述前端节点向所述用户端输出具有测试需求的需求对话框,将所述用户端在需求对话框中选择并发送的测试需求设为目标测试需求;

[0026] 通过所述前端节点在后端节点中将与所述目标测试需求关联的测试逻辑数据包设为目标测试逻辑。

[0027] 在上述方案中,通过所述测试服务器控制所述测试算法运行所述测试脚本以获得测试结果包括:

[0028] 通过所述测试服务器提取所述测试脚本中的目标测试逻辑,判断获取所述目标测试逻辑的测试需求;

[0029] 根据所述测试需求的种类通过测试服务器控制测试算法运行所述测试脚本以获得测试结果;其中,若所述测试需求的种类为冒烟测试,则通过测试算法运行所述测试脚本以获得冒烟测试结果;若所述测试需求的种类为算法响应时间测试,则通过测试算法运行所述测试脚本以获得响应测试结果;若所述测试需求的种类为性能测试,则通过测试算法运行测试脚本并启动采集程序,使所述采集程序采集所述测试服务器在运行测试脚本时的机器资源占用情况并生成性能测试结果。

[0030] 为实现上述目的,本发明还提供一种算法自动测试装置,包括:

[0031] 测试预备模块,用于将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据,在测试服务器中部署测试算法,并储存关联有测试需求的测试逻辑数据包;

[0032] 需求认定模块,用于接收用户端输出的目标信息,根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据,根据所述用户端选择的测试需求在所述测试服务器中确定目标测试逻辑;

[0033] 测试执行模块,用于将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本,通过测试服务器控制目标测试算法运行所述测试脚本以获得测试结果,调用所述测试服务器的回调接口生成回调信号并将其与所述测试结果输出至所述用户端。

[0034] 为实现上述目的,本发明还提供一种计算机系统,其包括多个计算机设备,各计算机设备包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,所述多个计算机设备的处理器执行所述计算机程序时共同实现上述算法自动测试方法的步骤。

[0035] 为实现上述目的,本发明还提供一种计算机可读存储介质,其包括多个存储介质,各存储介质上存储有计算机程序,所述多个存储介质存储的所述计算机程序被处理器执行时共同实现上述算法自动测试方法的步骤。

[0036] 本发明提供的算法自动测试方法、装置、计算机系统及可读存储介质,通过自动从生产服务器中获取生产数据以及运行所述生产数据的生产算法,并将其储存至数据服务器后再相互关联形成测试数据,因此无需测试人员编制或手动获取测试数据,降低了测试人员的疲劳强度,提高了数据获取效率,保证了测试数据完全满足生产环境的要求,进而保证了测试的准确性和可靠性以及测试资源的监控覆盖率;通过将测试算法部署在测试服务器中用于运行所述测试数据,并将与目标信息匹配的测试算法设为目标测试算法,使用户仅需通过用户端的选择即可获得已提前部署好的目标算法,为用户选择并执行该测试算法提供了便利,并且避免了因加载测试算法而浪费测试人员时间的情况;通过在预储存有测试逻辑数据包的测试服务器中设置逻辑表,用于关联所述测试逻辑数据包和测试需求;用户端通过确定目标测试需求即可使测试服务器直接从数据服务器中提取与其关联的测试逻辑数据包,提高了测试脚本的生成效率;通过数据服务器中将与所述目标信息匹配的测试数据设为目标测试数据,保证了测试需求、测试数据及其制成的测试脚本与测试算法的一致性;通过获取所述目标测试算法的测试信息中的部署路径,控制所述测试服务器使所述目标测试算法运行测试脚本以获得测试结果,使用户端获得多种测试结果,不仅实现了对算法功能的覆盖性测试,还实现了自动对算法进行测试的技术效果,避免了当前因手动制作测试脚本并对算法进行测试所造成的工作效率低、随意性大、出错机会大、拉长开发周期的问题。

附图说明

[0037] 图1为本发明算法自动测试方法实施例一的环境应用示意图;

[0038] 图2为本发明算法自动测试方法实施例一的流程图;

[0039] 图3为本发明算法自动测试方法实施例一S1中获取的生产数据并将其转为测试数据的流程图;

[0040] 图4为本发明算法自动测试方法实施例一S1中在测试服务器中部署测试算法的流程图;

[0041] 图5为本发明算法自动测试方法实施例一S1中储存关联有测试需求的测试逻辑数据包的流程图;

[0042] 图6为本发明算法自动测试方法实施例一S2中接收用户端输出的目标信息的流程图;

[0043] 图7为本发明算法自动测试方法实施例一S2中确定目标测试算法以及提取目标测

试数据的流程图；

[0044] 图8为本发明算法自动测试方法实施例一S2中确定目标测试逻辑的流程图；

[0045] 图9为本发明算法自动测试方法实施例一S3中获得测试结果的流程图；

[0046] 图10为本发明算法自动测试方法实施例一S3中根据测试需求的种类通过测试算法获得测试结果的流程图；

[0047] 图11为本发明算法自动测试装置实施例二的程序模块示意图；

[0048] 图12为本发明计算机系统实施例三中计算机设备的硬件结构示意图。

[0049] 附图标记：

[0050] 1、算法自动测试装置 2、测试服务器 3、生产服务器

[0051] 4、数字服务器 5、用户端 6、网络 7、计算机设备

[0052] 11、测试预备模块 12、需求认定模块 13、测试执行模块

[0053] 14、结果汇总模块 61、网络a 62、网络b 63、网络c

[0054] 71、存储器 72、处理器

具体实施方式

[0055] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本发明，并不用于限定本发明。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0056] 本发明提供的算法自动测试方法、装置、计算机系统及可读存储介质，适用于计算机领域，为提供一种基于测试预备模块、需求认定模块、测试执行模块、结果汇总模块的算法自动测试方法。本发明通过从生产服务器中获取生产数据以及运行所述生产数据的生产算法，并将其储存至数据服务器后再相互关联形成测试数据；将测试算法部署在测试服务器中用于运行所述测试数据，通过用户端发送的目标信息，并在测试服务器中将与所述目标信息匹配的测试算法设为目标测试算法；通过在预储存有测试逻辑数据包的测试服务器中设置逻辑表，用于关联所述测试逻辑数据包和测试需求；用户端通过确定目标测试需求即可使测试服务器直接从数据服务器中提取与其关联的测试逻辑数据包；通过数据服务器中将与所述目标信息匹配的测试数据设为目标测试数据，并将该目标测试数据与所述测试需求所关联的测试逻辑数据包打包形成测试脚本；通过获取所述目标测试算法的测试信息中的部署路径，控制所述测试服务器使所述目标测试算法运行测试脚本以获得测试结果，并通过调用回调接口所生成的回调信号，将测试结果和回调信号输出至用户端。

[0057] 图1示意性示出了根据本申请实施例一的算法自动测试方法的环境应用示意图。

[0058] 在示例性的实施例中，算法自动测试方法运行在测试服务器2中，测试服务器2通过网络6与数字服务器4和多个用户端5连接，数字服务器4通过网络6与生产服务器3连接；其中，所述网络6包括网络a61、网络b62、网络c63，所述测试服务器2通过网络a61与数字服务器4连接，所述测试服务器2通过网络b62与用户端5连接，所述数字服务器4与生产服务器3通过网络c63连接；测试服务器2通过数字服务器4从生产服务器3中获取生产数据及算法信息并将其储存至数字服务器4后关联，以获得测试数据；

[0059] 用户通过用户端5输入目标信息，测试服务器2根据所述目标信息在测试服务器2

中确定目标测试算法以及从数据服务器中提取目标测试数据；测试服务器2将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本，并控制目标测试算法运行所述测试脚本以获得测试结果，调用所述测试服务器2的回调接口生成回调信号并将其与所述测试结果输出至所述用户端5。

[0060] 测试服务器2可以通过一个或多个网络6提供服务，网络6可以包括各种网络设备，例如路由器，交换机，多路复用器，集线器，调制解调器，网桥，中继器，防火墙，代理设备和/或等等。网络6可以包括物理链路，例如同轴电缆链路，双绞线电缆链路，光纤链路，它们的组合和/或类似物。网络6可以包括无线链路，例如蜂窝链路，卫星链路，Wi-Fi链路和/或类似物。

[0061] 测试服务器2可以由单个或多个计算机设备(如，服务器)组成。该单个或多个计算设备可以包括虚拟化计算实例。虚拟化计算实例可以包括虚拟机，诸如计算机系统的仿真，操作系统，服务器等。计算设备可以基于定义用于仿真的特定软件(例如，操作系统，专用应用程序，服务器)的虚拟映像和/或其他数据来加载虚拟机。随着对不同类型的处理服务的需求改变，可以在一个或多个计算设备上加载和/或终止不同的虚拟机。可以实现管理程序以管理同一计算设备上的不同虚拟机的使用。

[0062] 实施例一

[0063] 请参阅图2，本实施例的一种算法自动测试方法，包括：

[0064] S1：将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据，在测试服务器中部署测试算法，并储存关联有测试需求的测试逻辑数据包；

[0065] S2：接收用户端输出的目标信息，根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据，根据所述用户端选择的测试需求在所述测试服务器中确定目标测试逻辑；

[0066] S3：将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本，通过测试服务器控制目标测试算法运行所述测试脚本以获得测试结果，调用所述测试服务器的回调接口生成回调信号并将其与所述测试结果输出至所述用户端。

[0067] 本实施例提供的算法自动测试方法用于自动从生产服务器中获取生产数据以及运行所述生产数据的生产算法，并将其储存至数据服务器后再相互关联形成测试数据，因此无需测试人员编制或手动获取测试数据，降低了测试人员的疲劳强度，提高了数据获取效率，同时由于测试数据是基于生产环境获得的，因此保证了测试数据完全满足生产环境的要求，进而保证了测试的准确性和可靠性；又由于本申请是自动从生产服务器中获取的生产数据，因此保证了测试资源的监控覆盖率。

[0068] 将测试算法部署在测试服务器中用于运行所述测试数据，通过用户端发送的目标信息，即在对话框中选择或输入测试算法ID和测试版本ID，并在测试服务器中将与所述目标信息匹配的测试算法设为目标测试算法，实现自动确定目标算法以及从数据服务器中提取目标测试数据，因此用户仅需通过用户端的选择即可获得已提前部署好的目标算法，为用户选择并执行该测试算法提供了便利，并且避免了因加载测试算法而浪费测试人员时间的情况；

[0069] 通过在预储存有测试逻辑数据包的测试服务器中设置逻辑表，用于关联所述测试逻辑数据包和测试需求；用户端通过确定目标测试需求即可使测试服务器直接从数据服务

器中提取与其关联的测试逻辑数据包,提高了测试脚本的生成效率;于本申请中,所述测试需求至少包括冒烟测试选项、算法响应时间测试选项或性能测试选项;通过数据服务器中将与所述目标信息匹配的测试数据设为目标测试数据,并将该目标测试数据与所述测试需求所关联的测试逻辑数据包打包形成测试脚本,保证了测试需求、测试数据及其制成的测试脚本与测试算法的一致性。

[0070] 通过获取所述目标测试算法的测试信息中的部署路径,控制所述测试服务器使所述目标测试算法运行测试脚本以获得测试结果,并通过调用回调接口所生成的回调信号,将测试结果和回调信号输出至用户端;通过自动生成测试脚本并利用目标测试算法对其进行运行,使用户端获得冒烟测试结果、或响应测试结果、或性能测试结果,不仅实现了对算法功能(冒烟测试)、响应(响应测试)和性能(性能测试)的覆盖性测试,还实现了自动对算法进行测试的技术效果,避免了当前因手动制作测试脚本并对算法进行测试所造成的工作效率低、随意性大、出错机会大、拉长开发周期的问题。

[0071] 需要说明的是,所述测试脚本为基于测试逻辑数据包和测试数据所构建的用于模拟多用户访问测试算法的计算机程序,因生成测试脚本的方法属于本领域公知常识,而本申请主要解决的技术问题是如何准确获得测试逻辑数据包及测试数据以获得测试脚本,因此关于测试脚本的生成原理在本申请中不再赘述;所述回调接口为测试服务器预存的,根据调用规则向用户端输出回调信号的计算机程序;例如:通过测试服务器中知著系统后台调用回调接口生成回调信号,通过该测试服务器将测试结果及回调信号输出至用户端以实现通知使用者并向其展示测试结果的技术效果,保证使用者能够及时获得测试结果。

[0072] 本实施例提供的算法自动测试方法由服务端计算机设备执行。

[0073] 在一个优选的实施例中,请参阅图3,所述将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据包括:

[0074] S101:从生产服务器中获取生产数据及所述生产服务器中生产算法的算法信息;

[0075] 本步骤中,通过测试服务器控制数字服务器定时的从生产服务器中,获取生产数据及所述生产服务器中生产算法的算法信息。

[0076] S102:将所述生产数据和算法信息储存至数据服务器并将其相互关联形成测试数据。

[0077] 本步骤中,通过测试服务器控制数字服务器将所述生产数据和算法信息储存至数据服务器并将其相互关联形成测试数据;同时,本步骤中的生产算法的算法信息包括:生产算法ID,生产算法名称,生产版本ID;所述数据服务器中具有算法清单,通过在所述算法清单中记录生产算法的生产算法ID,生产算法名称,生产版本ID,以及用于表达所述生产数据在所述数据服务器中储存位置的储存路径,使所述生产数据与算法信息关联形成测试数据,例如:生产信息的算法信息如下:

	生产算法 ID	生产算法名称	生产版本 ID	储存路径
[0078]	e8600ced4f83458abb1 c406d591a5778	monalisa	9b748a337ddd40a78a d514c9af	***/shengchan/m onalisa/3367

[0079] 因此,通过该算法清单使得生产算法与生产数据相互关联;通过这种方式保证了生产算法与生产数据能够相互对应,若测试服务器需要测试多种生产算法时,不会因生产

算法与生产数据匹配错误而导致测试失败。

[0080] 需要说明的是,所述生产服务器具有生产环境,所述生产环境是指正式提供对外服务的,所述生产环境是指为了完成对外服务工作所必需的计算机硬件、软件、网络设备、历史数据的总称;所述生产数据是指用户基于生产环境所生成或提出的需求指令的信息,其可以数据集的方式体现,如申请指令数据集、图片数据集等;生产算法是部署在生产环境中用于对生产数据进行处理计算机程序。

[0081] 在一个优选的实施例中,请参阅图4,所述在测试服务器中部署测试算法包括:

[0082] S111:在所述测试服务器中储存具有测试信息的测试算法;

[0083] 本步骤中,所述测试信息包括测试算法名称、测试算法ID、测试版本ID、版本号;将所述测试算法的在测试服务器中的储存路径载入所述测试信息中,并将其设为版本路径;其中,某一测试算法ID下至少具有一个版本ID及其版本号。

[0084] S112:将所述测试算法部署在所述测试服务器中,将所述测试算法在所述测试服务器中的部署路径载入所述测试信息中;

[0085] 本步骤中,根据所述测试信息的版本路径从测试服务器中提取测试算法并将其部署在所述测试服务器中;获取所述测试算法部署在测试服务器的储存路径,将该储存路径载入所述测试信息中并将其设为部署路径,通过这种方式以提高拉取部署的测试算法的效率。

[0086] 于本实施例中,所述测试算法名称为测试算法的通用名称;所述算法ID为自动生成的用于标识测试算法的唯一标识符;所述测试版本ID为自动生成的用于表示测试算法版本的唯一标识符;所述版本号为测试算法的版本信息;所述版本路径为测试算法的储存路径,用于算法自动化交付,同时,所述版本号与版本路径匹配,用于描述该路径下储存的测试算法的版本信息;其中,所述版本路径为SVN,即Subversion,是一个开放源代码的版本控制系统,版本路径则为该版本的测试算法在测试服务器中的储存路径,所述部署路径为测试算法部署在测试服务器中的储存路径,因此可通过部署路径从测试服务器中拉取测试算法。

[0087] 例如:测试算法ID:e8600ced4f83458abb1c406d591a5778;测试算法名称:monalisa;测试版本ID:9b748a337ddd40a78ad514c9af;算法版本号:3367;版本路径:***/monalisa/3367;部署路径:***/test/monalisa/。

[0088] 通过建立测试信息以便于使用者在测试服务器中快速准确的获取需要的测试信息,例如:通过测试算法名称和算法版本号获取需要的测试算法,通过版本路径获取该测试算法的数据包并将其部署在测试服务器中,当需要使用到该测试算法时,则通过部署路径访问该测试算法并运行。

[0089] 于本实施例中,所述测试服务器内为测试环境,所述测试环境(Testingenvironment)是指测试运行其上的软件和硬件环境的描述,以及任何其它与被测软件交互的软件,包括驱动和桩;测试环境是指为了完成软件测试工作所必需的计算机硬件、软件、网络设备、历史数据的总称,通常通过复制生产环境的配置获得;本实施例中的测试环境为Python的虚拟环境,其为深度学习算法创建互相独立的运行环境,所述虚拟环境可通过virtualenv或pyenv工具创建虚拟环境;进一步地,本实施例中的测试数据是通过从生产服务器的生产环境中所获得的生产数据所制成的,因此,不仅不会影响生产环境的业

务相应,还保证了测试数据的真实有效,本实施例中可采用goreplay工具从生产服务器中获得生产数据;所述测试算法是部署在测试环境中的算法,用于通过测试脚本对其进行测试以获得测试结果,如,该算法的可靠性、效率和稳定性;若测试算法通过了测试,该算法将会部署到生产环境中使用。

[0090] 在一个优选的实施例中,请参阅图5,所述储存关联有测试需求的测试逻辑数据包包括:

[0091] S121:将预生成的测试逻辑数据包储存至测试服务器,并获得所述测试逻辑数据包在测试服务器中的储存路径;其中,所述测试逻辑数据包是指记载有对测试数据进行测试的方法的数据包。

[0092] S122:在测试服务器中设置逻辑表,将所述储存路径和测试需求写入逻辑表中使测试逻辑数据包与测试需求关联;

[0093] 本步骤中的测试需求是指用户向测试服务器中发送的测试项目,其可为可为冒烟测试、算法响应时间测试或性能测试。

[0094] 需要说明的是,所述测试逻辑数据包由测试人员预先编写的测试逻辑并将其打包所形成。其中,冒烟测试为一种用于判断输入的测试数据是否通过测试的方法;算法响应时间测试为测试不同大小的数据算法推理时间,例如,输入大小不同的数据,如以图片为测试数据,则输入50kb,100kb,200kb等不同大小的图片,记录算法响应时间;性能测试为模拟多用户访问算法服务,监控机器资源(如,cpu使用率,cpu占用率、gpu使用率、显存占用率、内存占用率等)的占用信息,监控占用过高等异常事件,并且在并发数量增长过程中,找到每秒事务处理量(tps)最优的并发数。

[0095] 在一个优选的实施例中,请参阅图6,所述接收用户端输出的目标信息包括:

[0096] S201:在所述测试服务器中创建前端节点和储存有测试算法及测试逻辑数据包的后端节点;

[0097] 本步骤中,所述前端节点用于实现所述用户端与后端节点的通信;通过将服务器分为前端节点和后端节点,并通过前端节点向用户端发送对话框以获得用户端的需求,及根据该需求从后端节点提取信息;使得前端节点负责用户端的需求要约(对话框)发送,以及需求判断(用户端在对话框上的点选或输入的信息),因此,储存有大量测试算法的后端节点无需负担大量的数据处理及发送工作,而仅需根据前端节点发送的请求发送相应的信息即可,避免了该后端节点因大量数据处理工作而导致死机的情况发生。

[0098] S202:通过所述前端节点向用户端输出具有测试算法ID的算法对话框,并将所述用户端在所述算法对话框中选择并发送的测试算法ID设为目标算法ID;

[0099] 本步骤中的算法对话框包括算法ID输入项,所述算法ID输入项为输入框、单选框、复选框中任一种形式;用户端可通过在算法对话框的算法ID输入项中点选或输入以实现测试算法ID的选择。

[0100] S203:通过所述后端节点根据前端节点发送的目标算法ID,向所述前端节点输出与所述目标算法ID匹配的测试版本ID;通过所述前端节点向所述用户端输出具有所述测试版本ID的版本对话框,并将所述用户端在所述版本对话框选择并发送的版本ID设为目标版本ID;

[0101] 本步骤中的版本对话框包括版本ID输入项,所述版本ID输入项为输入框、单选框、

复选框中任一种形式；用户端可通过在版本对话框的版本ID输入项中点选或输入以实现测试版本ID的选择。

[0102] S204:汇总所述目标算法ID和目标版本ID形成目标信息。

[0103] 本步骤中,所述目标信息储存在所述后端节点中。

[0104] 在一个优选的实施例中,请参阅图7,所述根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据包括:

[0105] S211:通过所述前端节点根据所述目标信息在所述后端节点中确定目标测试算法;

[0106] 本步骤中,通过所述前端节点从所述后端节点中提取与所述目标信息匹配的测试信息,即:与所述目标算法ID和目标版本ID均匹配的测试信息;提取所述测试信息中的部署路径;根据所述部署路径在所述测试服务器中获得测试算法并将其设为目标测试算法;例如,用户需要测试3367版本的monalisa测试算法,其中,目标算法ID:e8600ced4f83458abb1c406d591a5778,目标版本ID:9b748a337ddd40a78ad514c9af;则通过前端节点在后端节点中获取测试算法ID:e8600ced4f83458abb1c406d591a5778,测试版本ID:9b748a337ddd40a78ad514c9af,测试算法名称:monalisa,版本号:3367的测试算法并将其设为目标测试算法。

[0107] S212:通过所述后端节点从数据服务器中提取与所述目标信息匹配的测试数据,并将所述测试数据的编号通过前端节点以数据对话框的方式发送至所述用户端;

[0108] 示例性地,将所述目标算法ID和目标版本ID依次与各测试数据的算法信息比对,若算法信息中的生产算法ID和生产版本ID分别与所述目标算法ID和目标版本ID一致,则判定该测试数据与所述目标算法ID和目标版本ID匹配,因此通过这种方式保证了提取的测试数据与使用者需要测试的算法相匹配的效果。所述数据对话框包括测试需求输入项,所述测试需求输入项为输入框、单选框、复选框中任一种形式;用户端可通过在数据对话框的测试需求输入项中点选或输入以实现测试数据的选择,以便于使用者选择测试数据。

[0109] S213:所述前端节点将所述用户端在数据对话框中选择并发送的编号转发至所述后端节点,通过所述后端节点提取该编号对应的测试数据并将其设为目标测试数据。

[0110] 示例性地,所述测试数据具有唯一的编号,通过将该编号以对话框的形式发送至用户端,以便于用户端对测试数据进行选择。

[0111] 在一个优选的实施例中,请参阅图8,所述根据用户端的选择的测试需求在所述测试服务器中确定目标测试逻辑包括:

[0112] S221:通过所述前端节点向所述用户端输出具有测试需求的需求对话框,将所述用户端在需求对话框中选择并发送的测试需求设为目标测试需求;

[0113] 本步骤中的版本对话框包括测试需求输入项,所述测试需求输入项为输入框、单选框、复选框中任一种形式;用户端可通过在版本对话框的测试需求输入项中点选或输入以实现测试需求的选择;示例性地,所述测试需求输入项至少包括:冒烟测试输入项、算法响应时间测试输入项和性能测试输入项。

[0114] S222:通过所述前端节点在后端节点中将与所述目标测试需求关联的测试逻辑数据包设为目标测试逻辑;

[0115] 本步骤中,从所述测试服务器中提取逻辑表,从逻辑表中获得与所述目标需求匹

配的测试逻辑数据包,并将其设为目标测试逻辑。

[0116] 在一个优选的实施例中,请参阅图9,通过所述测试服务器控制所述测试算法运行所述测试脚本以获得测试结果包括:

[0117] S301:通过所述测试服务器提取所述测试脚本中的目标测试逻辑,并获取所述目标测试逻辑的测试需求;

[0118] S302:根据所述测试需求的种类通过测试服务器控制测试算法运行所述测试脚本以获得测试结果。

[0119] 具体的,请参阅图10,所述S302包括以下步骤:

[0120] S302-1:判断测试需求的种类;若该种类为冒烟测试则进入S302-2;若该种类为算法响应时间测试,则进入S302-3;若该种类为性能测试,则进入S303-4;

[0121] S302-2:若所述测试需求的种类为冒烟测试,通过所述测试服务器控制所述测试算法运行所述测试脚本以获得冒烟测试结果;

[0122] S302-3:若所述测试需求的种类为算法响应时间测试,通过所述测试服务器控制所述测试算法运行所述测试脚本以获得响应测试结果;

[0123] S302-4:若所述测试需求的种类为性能测试,通过所述测试服务器控制所述测试算法运行测试脚本并启动采集程序,使所述采集程序采集所述测试服务器在运行测试脚本时的机器资源占用情况并生成性能测试结果。

[0124] 于本实施例中,所述冒烟测试结果为记载通过冒烟测试或未通过冒烟测试的信息;所述响应测试结果为记载运行所述测试脚本所使用的时间的信息;例如,以图片为所述测试脚本的目标测试数据,使所述目标测试数据为图片数据集,其包括有50kb,100kb,200kb等不同大小的图片,记录所述目标测试算法运行所述测试脚本的响应时间;所述性能测试结果为记载所述机器资源占用情况的信息,于本实施例中,所述机器资源占用情况至少包括cpu占用率、内存使用率、显存使用率gpu占用率。

[0125] 需要说明的是,所述采集程序为测试服务器中预存的,根据算法规则采集测试服务器的机器资源占用情况的计算机程序;于本实施例中,所述采集程序的时间间隔设置为30秒,所述时间间隔用于表达采集被测试算法实例进程占用资源的时间间隔,因此将时间间隔设置为30秒即为每30秒取得一次目标测试算法进程的机器资源占用情况。

[0126] 在示例性的实施例中,在获取所述目标测试逻辑的测试需求之后还可包括:

[0127] S301-1:将测试脚本与测试服务器中预设的服务阈值进行比对;

[0128] S301-2:若所述测试脚本中的测试数据大于服务阈值,则复制所述测试服务器中的测试算法并将其部署在预设的备用服务器中,分解所述测试脚本形成若干个子测试脚本,将所述若干个子测试脚本平均分配至所述测试服务器和备用服务器中,使其分别运行所述子测试脚本。

[0129] 通过这种方式以避免因测试脚本过大而导致测试服务器负载过高,使其出现死机或系统崩溃的情况。

[0130] 在示例性的实施例中,在获得性能测试结果之后还可包括:

[0131] S302-1:在所述测试服务器中设置GPU阈值、CPU阈值、显存阈值和内存阈值;

[0132] S302-2:将所述机器资源的cpu占用率与CPU阈值比对,若cpu占用率大于CPU阈值,则生成CPU警告信息并将其载入所述测试结果;

[0133] 将所述机器资源的内存使用率与内存阈值比对,若内存使用率大于内存阈值比对,则生成内存警告信息并将其载入所述测试结果;

[0134] 将所述机器资源的显存使用率与显存阈值比对,显存使用率大于显存阈值,则生成显存警告信息并将其载入所述测试结果;

[0135] 将所述机器资源的gpu占用率与GPU阈值比对,若gpu占用率大于GPU阈值,则生成GPU警告信息并将其载入所述性能测试结果。通过这种方式为用户提供可视化数据,以提高使用者的数据获取效率和便利度。

[0136] 于本实施例中,所述测试结果中的机器资源占用情况的cpu占用率、内存使用率、显存使用率、gpu占用率,以及GPU阈值、CPU阈值、显存阈值和内存阈值在所述测试结果中可以数字、饼图、折线图或柱状图的方式展示,以便于使用者直观获知测试结果中所记载的数据。

[0137] 在示例性的实施例中,所述步骤S30之后还可包括:

[0138] S40:按照所述步骤S20和步骤S30依次获得测试服务器中任一测试算法的所有版本ID的测试结果,汇总所述测试算法中各版本ID的测试结果形成算法结果集,并通过回调接口将所述算法结果集输出至用户端。

[0139] 其中,通过所述步骤S40汇总各版本ID的测试算法运行测试脚本后所生成的测试结果,以便于使用者对各版本ID的测试算法进行比较,有助于使用者判断各版本测试算法的优劣。

[0140] 实施例二

[0141] 请参阅图11,本实施例的一种算法自动测试装置1,包括:

[0142] 测试预备模块11,用于将从生产服务器中获取的生产数据储存至数据服务器并将其转为测试数据,在测试服务器中部署测试算法,并储存关联有测试需求的测试逻辑数据包;

[0143] 需求认定模块12,用于接收用户端输出的目标信息,根据所述目标信息在测试服务器中确定目标测试算法以及从数据服务器中提取目标测试数据,根据所述用户端选择的测试需求在所述测试服务器中确定目标测试逻辑;

[0144] 测试执行模块13,用于将所述目标测试逻辑和目标测试数据汇总打包形成测试脚本,通过测试服务器控制目标测试算法运行所述测试脚本以获得测试结果,调用所述测试服务器的回调接口生成回调信号并将其与所述测试结果输出至所述用户端。

[0145] 在示例性的实施例中,所述算法自动测试装置还可包括:

[0146] 结果汇总模块14,用于调用需求认定模块12和测试执行模块13依次获得测试服务器中任一测试算法的所有版本ID的测试结果,汇总所述测试算法中各版本ID的测试结果形成算法结果集,并通过回调接口将所述算法结果集输出至用户端。

[0147] 本技术方案基于研发管理领域的测试技术,即提供一种自动化测试工具从生产服务器中获取生产数据以及运行所述生产数据的生产算法,并将其相互关联形成测试数据;将测试算法部署在测试服务器中用于运行所述测试数据,并在测试服务器中将与所述目标信息匹配的测试算法设为目标测试算法;通过在预储存有测试逻辑数据包的测试服务器中设置逻辑表,用于关联所述测试逻辑数据包和测试需求;通过数据服务器中将与所述目标信息匹配的测试数据设为目标测试数据,并将该目标测试数据与所述测试需求所关联的测

试逻辑数据包打包形成测试脚本;通过获取所述目标测试算法的测试信息中的部署路径,控制所述测试服务器使所述目标测试算法运行测试脚本以获得测试结果,并通过调用回调接口所生成的回调信号,将测试结果和回调信号输出至用户端。

[0148] 实施例三:

[0149] 为实现上述目的,本发明还提供一种计算机系统,该计算机系统包括多个计算机设备7,实施例二的算法自动测试装置1的组成部分可分散于不同的计算机设备7中,计算机设备7可以是执行程序的智能手机、平板电脑、笔记本电脑、台式计算机、机架式服务器、刀片式服务器、塔式服务器或机柜式服务器(包括独立的服务器,或者多个服务器所组成的服务器集群)等。本实施例的计算机设备7至少包括但不限于:可通过系统总线相互通信连接的存储器71、处理器72,如图12所示。需要指出的是,图12仅示出了具有组件-的计算机设备7,但是应理解的是,并不要求实施所有示出的组件,可以替代的实施更多或者更少的组件。

[0150] 本实施例中,存储器71(即可读存储介质)包括闪存、硬盘、多媒体卡、卡型存储器(例如,SD或DX存储器等)、随机访问存储器(RAM)、静态随机访问存储器(SRAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、可编程只读存储器(PROM)、磁性存储器、磁盘、光盘等。在一些实施例中,存储器71可以是计算机设备7的内部存储单元,例如该计算机设备7的硬盘或内存。在另一些实施例中,存储器71也可以是计算机设备7的外部存储设备,例如该计算机设备7上配备的插接式硬盘,智能存储卡(SmartMedia Card, SMC),安全数字(Secure Digital, SD)卡,闪存卡(Flash Card)等。当然,存储器71还可以既包括计算机设备7的内部存储单元也包括其外部存储设备。本实施例中,存储器71通常用于存储安装于计算机设备7的操作系统和各类应用软件,例如实施例一的算法自动测试装置的程序代码等。此外,存储器71还可以用于暂时地存储已经输出或者将要输出的各类数据。

[0151] 处理器72在一些实施例中可以是中央处理器(Central Processing Unit, CPU)、控制器、微控制器、微处理器、或其他数据处理芯片。该处理器72通常用于控制计算机设备7的总体操作。本实施例中,处理器72用于运行存储器71中存储的程序代码或者处理数据,例如运行算法自动测试装置,以实现实施例一的算法自动测试方法。

[0152] 实施例四:

[0153] 为实现上述目的,本发明还提供一种计算机可读存储系统,其包括多个存储介质,如闪存、硬盘、多媒体卡、卡型存储器(例如,SD或DX存储器等)、随机访问存储器(RAM)、静态随机访问存储器(SRAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、可编程只读存储器(PROM)、磁性存储器、磁盘、光盘、服务器、App应用商城等等,其上存储有计算机程序,程序被处理器72执行时实现相应功能。本实施例的计算机可读存储介质用于存储算法自动测试装置,被处理器72执行时实现实施例一的算法自动测试方法。

[0154] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0155] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。

[0156] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

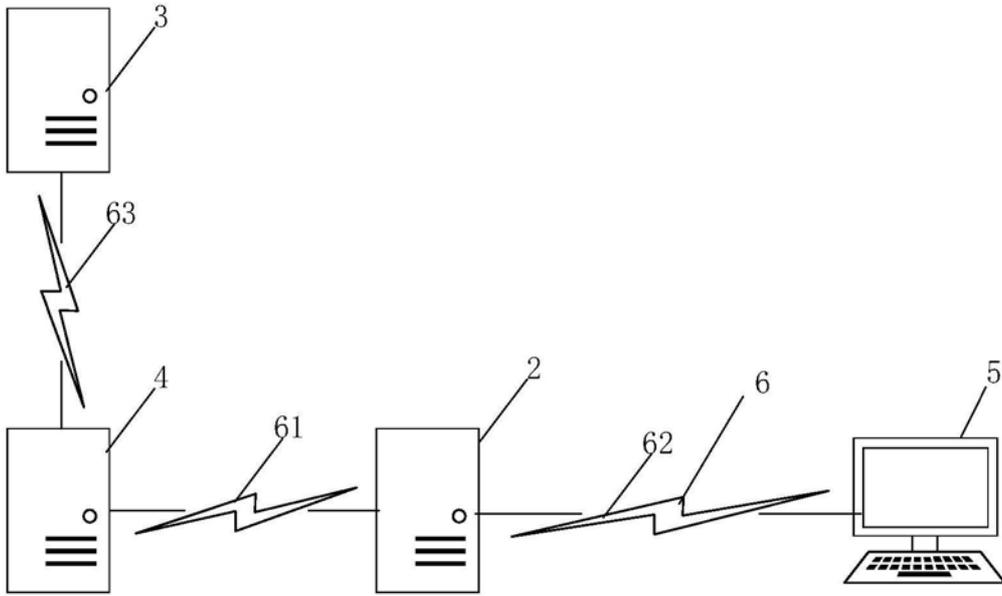


图1

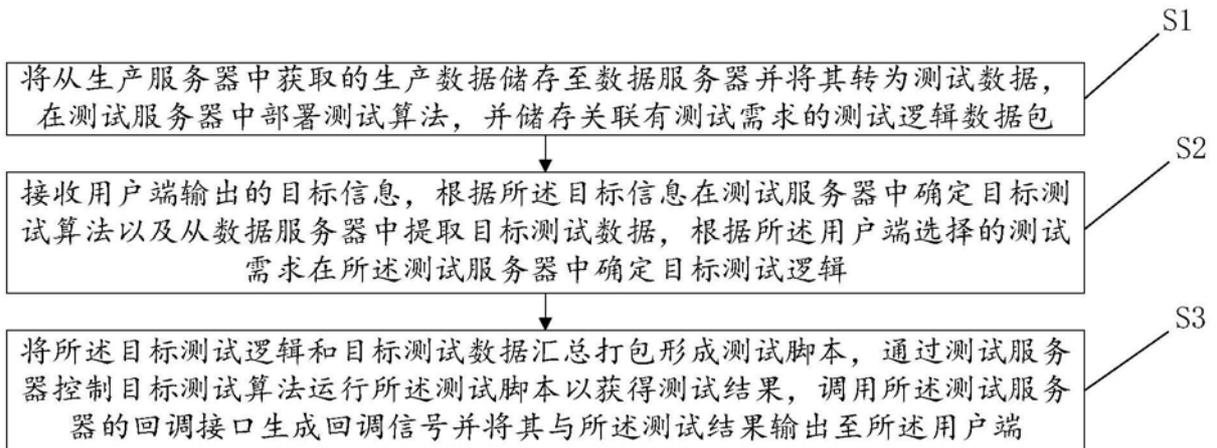


图2

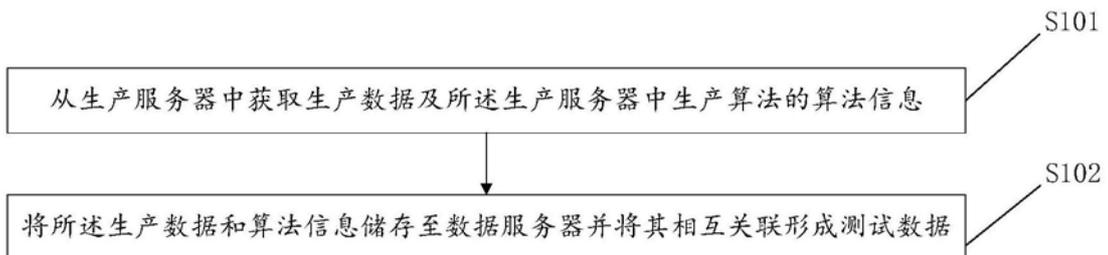


图3

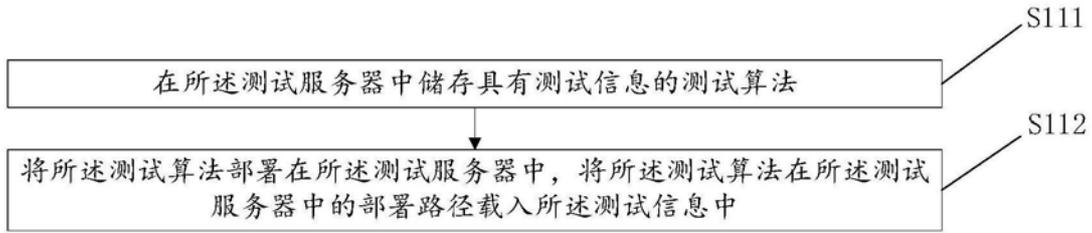


图4

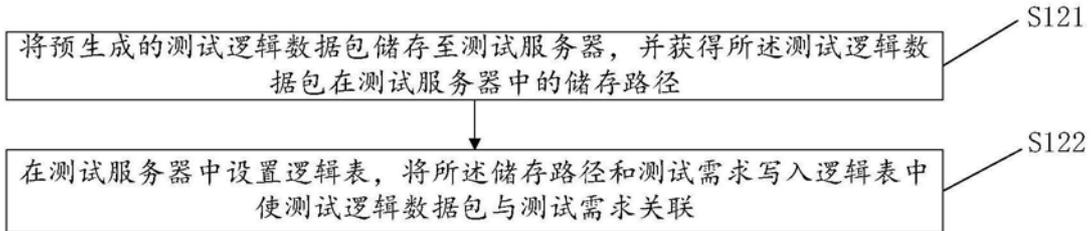


图5

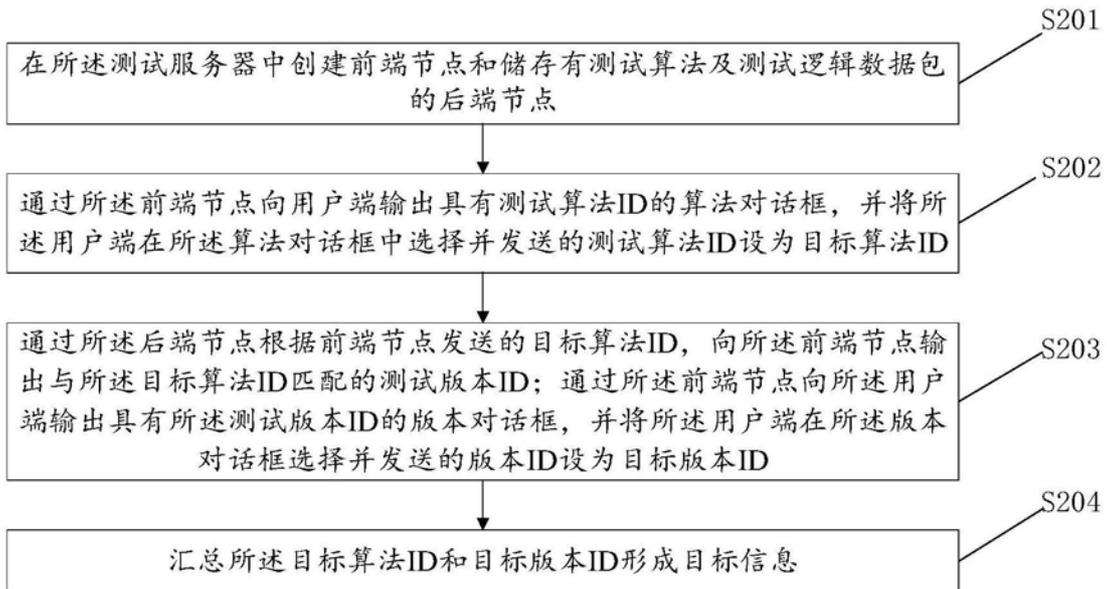


图6

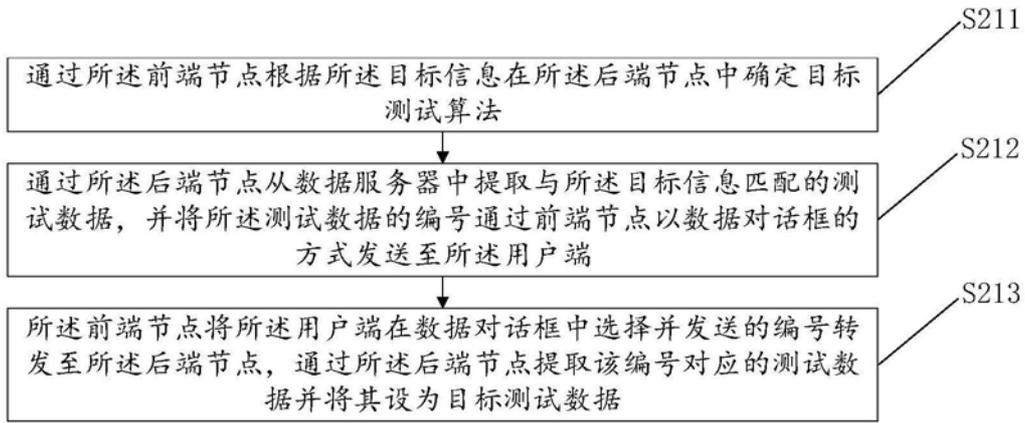


图7

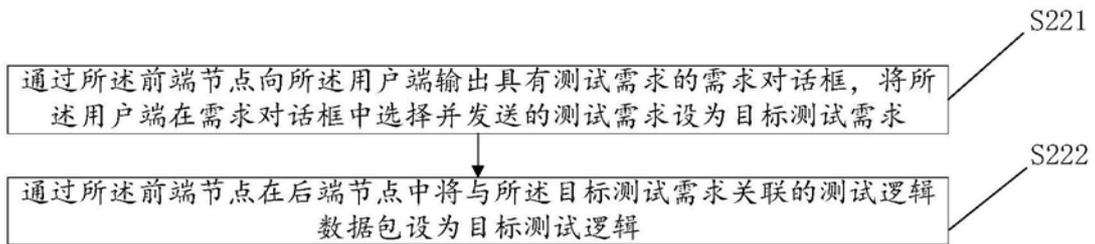


图8

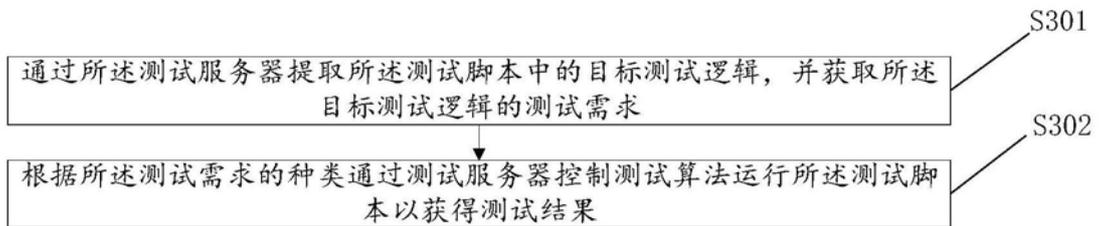


图9

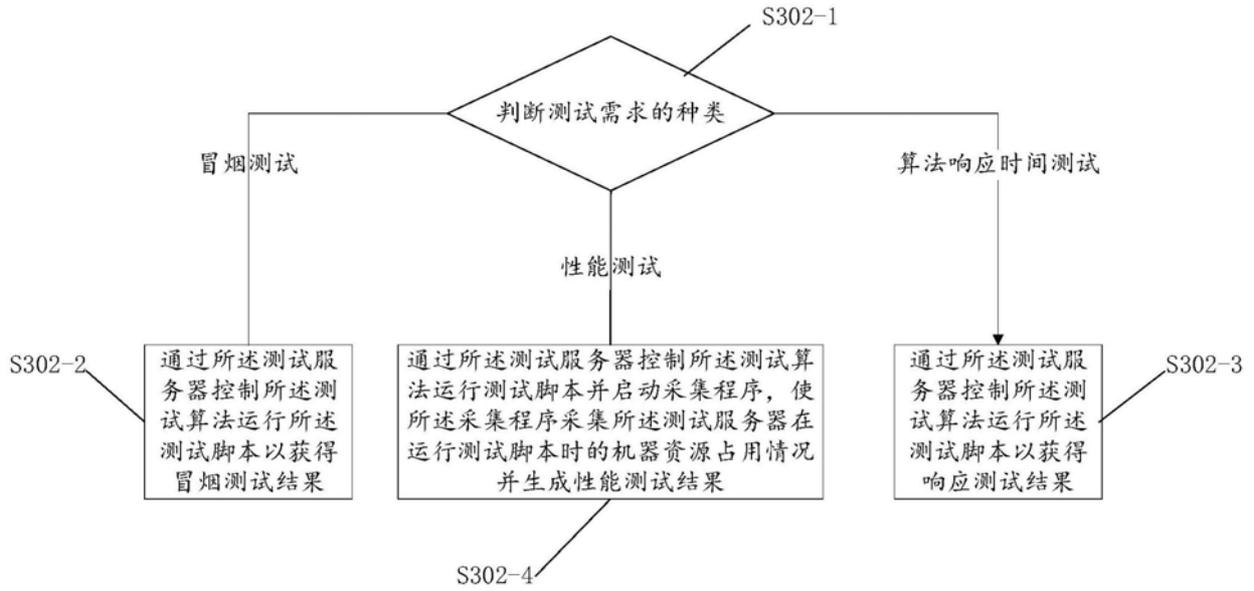


图10

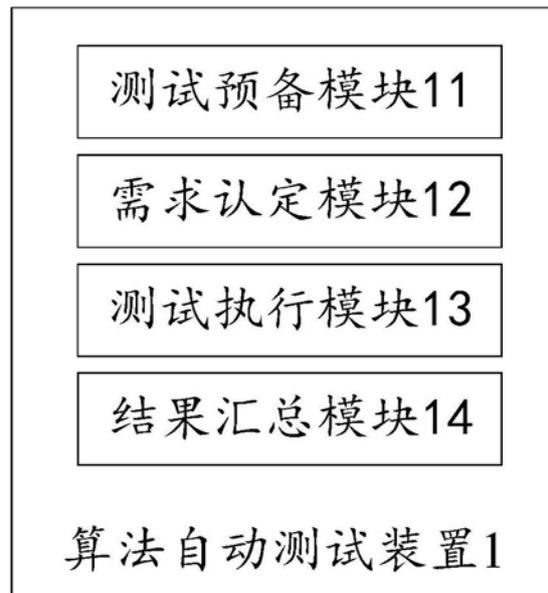


图11

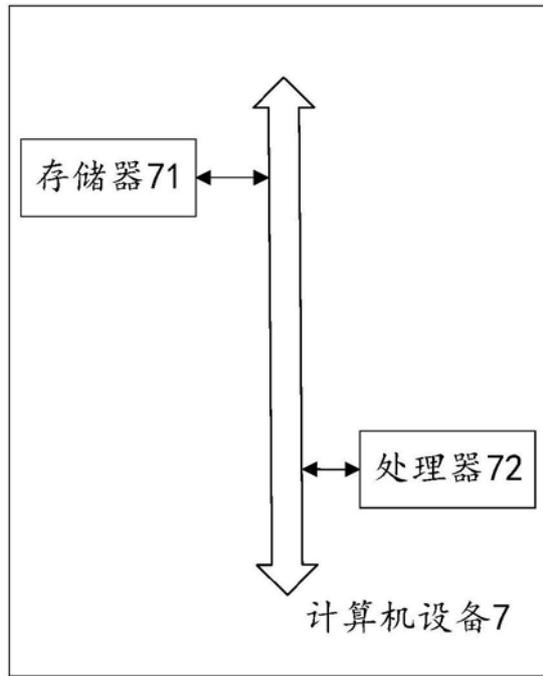


图12