



(12) 发明专利申请

(10) 申请公布号 CN 114745295 A

(43) 申请公布日 2022. 07. 12

(21) 申请号 202210408515.X

(22) 申请日 2022.04.19

(71) 申请人 京东科技控股股份有限公司
地址 100176 北京市北京经济技术开发区
科创十一街18号C座2层221室

(72) 发明人 刘启翔

(74) 专利代理机构 北京同立钧成知识产权代理
有限公司 11205
专利代理师 黄健 刘芳

(51) Int. Cl.
H04L 43/04 (2022.01)
H04L 41/0677 (2022.01)
H04L 41/069 (2022.01)

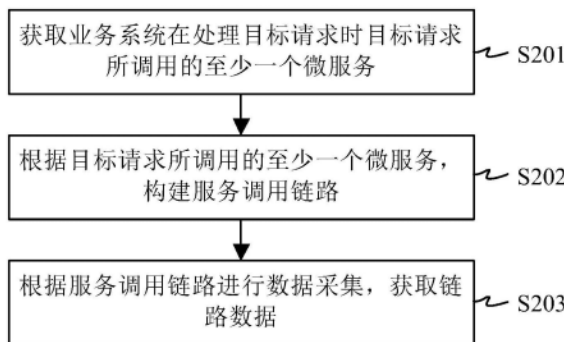
权利要求书2页 说明书11页 附图3页

(54) 发明名称

数据采集方法、装置、设备和可读存储介质

(57) 摘要

本申请提供一种数据采集方法、装置、设备和可读存储介质,涉及金融科技技术领域,该方法包括:获取业务系统在处理目标请求时目标请求所调用的微服务,根据目标请求所调用的至少一个微服务,构建服务调用链路,根据服务调用链路进行数据采集,获取链路数据。该技术方案中,通过对目标请求所调用的微服务进行跟踪,确定目标请求的服务调用链路以获取链路数据,能够根据链路数据来实现故障定位,避免使用日志的方式进行故障定位,提高故障定位效率以快速解决故障。



1. 一种数据采集方法,其特征在于,应用于基于微服务架构的业务系统,所述业务系统包括有至少一个微服务,所述方法包括:

获取所述业务系统在处理目标请求时所述目标请求所调用的至少一个微服务;

根据所述目标请求所调用的所述至少一个微服务,构建服务调用链路;

根据所述服务调用链路进行数据采集,获取链路数据,所述链路数据用于所述业务系统的故障定位,所述链路数据至少包括有所述服务调用链路的上下文信息、链路标识、调用开始时间和耗时、调用方网络协议地址和端口中的至少一种。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述服务调用链路进行数据采集,获取链路数据,包括:

根据各个微服务中的埋点组件获取所述链路数据,每个微服务包括有至少一个埋点组件,所述埋点组件用于进行链路数据的采集。

3. 根据权利要求1所述的方法,其特征在于,若所述目标请求调用至少两个微服务,所述根据所述目标请求所调用的至少一个微服务,构建服务调用链路,包括:

获取所述目标请求所调用的各个微服务之间的调用先后顺序;

根据所述目标请求所调用的各个微服务和所述调用先后顺序,构建所述服务调用链路。

4. 根据权利要求3所述的方法,其特征在于,所述获取所述目标请求所调用的各个微服务之间的调用先后顺序,包括:

在所述目标请求调用当前微服务时,获取所述当前微服务的标识信息;

在所述目标请求完成所述当前微服务的调用并继续调用下一微服务时,将所述当前微服务的标识信息传输至下一微服务以进行记录;

在目标请求完成各个微服务的调用时,根据各个微服务记录的标识信息,确定各个微服务之间的调用先后顺序。

5. 根据权利要求4所述的方法,其特征在于,所述方法还包括:

获取全局标识,在所述目标请求调用当前微服务时,将所述全局标识传输至当前微服务以进行记录,所述全局标识用于对所述目标请求进行标识;

在所述目标请求调用下一微服务时,将所述全局标识传输至所述下一微服务以进行记录;

根据各个微服务中记录的全局标识,确定所述目标请求所调用的微服务。

6. 根据权利要求5所述的方法,其特征在于,所述获取全局标识,包括:

获取所述目标请求的请求时间戳和所述当前微服务的网络协议地址;

根据所述请求时间戳和所述当前微服务的网络协议地址,生成所述全局标识。

7. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述业务系统发出故障提示时,根据所述链路数据进行故障排查,得到排查结果,所述排查结果用于指示所述业务系统发生故障的位置。

8. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

响应于所述业务系统中预设开关的开启或关闭,控制各个微服务中的埋点组件的启动或停止,所述埋点组件用于进行链路数据的收集。

9. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

获取所述业务系统中预设的白名单路径,所述白名单路径用于指示埋点组件进行数据采集,所述微服务中包括有埋点组件,所述埋点组件用于进行数据采集;

根据所述埋点组件在所述白名单路径中采集得到的数据,获取所述链路数据。

10. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

根据所述服务调用链路,获取各个微服务之间的拓扑关系;

根据所述拓扑关系,构建所述业务系统的服务框架,所述服务框架用于指示所述业务系统所包含的微服务以及各个微服务的关联关系。

11. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

根据消息队列机制,将所述服务调用链路异步发送存储至预设数据库中。

12. 一种数据采集装置,其特征在于,包括:

服务获取模块,用于获取业务系统在处理目标请求时所述目标请求所调用的微服务,所述目标请求可调用至少一个微服务;

链路构建模块,用于根据所述目标请求所调用的所述至少一个微服务,构建服务调用链路;

数据获取模块,用于根据所述服务调用链路进行数据采集,获取链路数据,所述链路数据至少包括有所述服务调用链路的上下文信息、链路标识、调用开始时间和耗时、调用方网络协议地址和端口中的至少一种。

13. 一种电子设备,其特征在于,包括:处理器,以及与所述处理器通信连接的存储器;

所述存储器存储计算机执行指令;

所述处理器执行所述存储器存储的计算机执行指令,以实现如权利要求11中任一项所述的方法。

14. 一种可读存储介质,其特征在于,所述可读存储介质中存储有计算机指令,所述计算机指令被处理器执行时用于实现如权利要求1-11任一项所述的方法。

15. 一种程序产品,包括计算机指令,其特征在于,该计算机指令被处理器执行时实现权利要求1-11任一项所述的方法。

数据采集方法、装置、设备和可读存储介质

技术领域

[0001] 本申请涉及互联网技术领域,尤其涉及一种数据采集方法、装置、设备和可读存储介质。

背景技术

[0002] 随着互联网技术的发展,基于微服务架构设计的系统得到广泛运用,微服务架构是把一个大型的单个应用程序和服务拆分为数个或数十个的微服务,旨在将系统功能分离到各个离散的服务当中,降低系统的耦合性并提供更加灵活的服务支持。

[0003] 现有技术中,基于微服务架构的系统在出现故障问题时,通常是通过调用日志方式来实现问题定位和解决。

[0004] 但是,由于各个服务按照不同的维度和定位进行拆分且多节点多机房部署,一次请求往往涉及到调用多个部署在不同机房的服务,现有技术的这种通过日志来来故障定位方式,日志获取困难,不能实现快速问题定位和故障影响范围的判断,导致故障排查效率低。

发明内容

[0005] 本申请提供一种数据采集方法、装置、设备和可读存储介质,用于解决现有的通过日志数据定位基于微服务架构的业务系统的故障,故障排查效率低的问题。

[0006] 第一方面,本申请实施例提供一种数据采集方法,应用于基于微服务架构的业务系统,所述业务系统包括有至少一个微服务,所述方法包括:

[0007] 获取所述业务系统在处理目标请求时所述目标请求所调用的微服务,所述目标请求可调用至少一个微服务;

[0008] 根据所述目标请求所调用的所述至少一个微服务,构建服务调用链路;

[0009] 根据所述服务调用链路进行数据采集,获取链路数据,所述链路数据用于所述业务系统的故障定位,所述链路数据至少包括有所述服务调用链路的上下文信息、链路标识、调用开始时间和耗时、调用方网络协议地址和端口中的至少一种。

[0010] 在第一方面的一种可能设计中,所述根据所述服务调用链路进行数据采集,获取链路数据,包括:

[0011] 根据各个微服务中的埋点组件获取所述链路数据,每个微服务包括有至少一个埋点组件,所述埋点组件用于进行链路数据的采集。

[0012] 在第一方面的另一种可能设计中,若所述目标请求调用至少两个微服务,所述根据所述目标请求所调用的至少一个微服务,构建服务调用链路,包括:

[0013] 获取所述目标请求所调用的各个微服务之间的调用先后顺序;

[0014] 根据所述目标请求所调用的各个微服务和所述调用先后顺序,构建所述服务调用链路。

[0015] 在第一方面的再一种可能设计中,所述获取所述目标请求所调用的各个微服务之

间的调用先后顺序,包括:

- [0016] 在所述目标请求调用当前微服务时,获取所述当前微服务的标识信息;
- [0017] 在所述目标请求完成所述当前微服务的调用并继续调用下一微服务时,将所述当前微服务的标识信息传输至下一微服务以进行记录;
- [0018] 在目标请求完成各个微服务的调用时,根据各个微服务记录的标识信息,确定各个微服务之间的调用先后顺序。
- [0019] 在第一方面的又一种可能设计中,所述方法还包括:
- [0020] 获取全局标识,在所述目标请求调用当前微服务时,将所述全局标识传输至当前微服务以进行记录,所述全局标识用于对所述目标请求进行标识;
- [0021] 在所述目标请求调用下一微服务时,将所述全局标识传输至所述下一微服务以进行记录;
- [0022] 根据各个微服务中记录的全局标识,确定所述目标请求所调用的微服务。
- [0023] 在第一方面的又一种可能设计中,所述获取全局标识,包括:
- [0024] 获取所述目标请求的请求时间戳和所述当前微服务的网络协议地址;
- [0025] 根据所述请求时间戳和所述当前微服务的网络协议地址,生成所述全局标识。
- [0026] 在第一方面的又一种可能设计中,所述方法还包括:
- [0027] 当所述业务系统发出故障提示时,根据所述链路数据进行故障排查,得到排查结果,所述排查结果用于指示所述业务系统发生故障的位置。
- [0028] 在第一方面的又一种可能设计中,所述方法还包括:
- [0029] 响应于所述业务系统中预设开关的开启或关闭,控制各个微服务中的埋点组件的启动或停止,所述埋点组件用于进行链路数据的收集。
- [0030] 在第一方面的又一种可能设计中,所述方法还包括:
- [0031] 获取所述业务系统中预设的白名单路径,所述白名单路径用于指示埋点组件进行数据采集,所述微服务中包括有埋点组件,所述埋点组件用于进行数据采集;
- [0032] 根据所述埋点组件在所述白名单路径中采集得到的数据,获取所述链路数据。
- [0033] 在第一方面的又一种可能设计中,所述方法还包括:
- [0034] 根据所述服务调用链路,获取各个微服务之间的拓扑关系;
- [0035] 根据所述拓扑关系,构建所述业务系统的服务框架,所述服务框架用于指示所述业务系统所包含的微服务以及各个微服务的关联关系。
- [0036] 在第一方面的又一种可能设计中,所述方法还包括:
- [0037] 根据消息队列机制,将所述服务调用链路异步发送存储至预设数据库中。
- [0038] 第二方面,本申请实施例提供一种数据采集装置,包括:
- [0039] 服务获取模块,用于获取业务系统在处理目标请求时所述目标请求所调用的微服务,所述目标请求可调用至少一个微服务;
- [0040] 链路构建模块,用于根据所述目标请求所调用的所述至少一个微服务,构建服务调用链路;
- [0041] 数据获取模块,用于根据所述服务调用链路进行数据采集,获取链路数据,所述链路数据至少包括有所述服务调用链路的上下文信息、链路标识、调用开始时间和耗时、调用方网络协议地址和端口中的至少一种。

[0042] 第三方面,本申请实施例提供一种电子设备,包括:处理器,以及与所述处理器通信连接的存储器;

[0043] 所述存储器存储计算机执行指令;

[0044] 所述处理器执行所述存储器存储的计算机执行指令,以实现上述的方法。

[0045] 第四方面,本申请实施例提供一种可读存储介质,所述可读存储介质中存储有计算机指令,所述计算机指令被处理器执行时用于实现上述的方法。

[0046] 第五方面,本申请实施例提供一种程序产品,包括计算机指令,该计算机指令被处理器执行时实现上述的方法。

[0047] 本申请实施例提供的数据采集方法、装置、设备和可读存储介质,通过对目标请求所调用的微服务进行跟踪,确定目标请求的服务调用链路以获取链路数据,能够根据链路数据来实现故障定位,避免使用日志的方式进行故障定位,提高故障定位效率以快速解决故障。

附图说明

[0048] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本申请的实施例,并与说明书一起用于解释本申请的原理;

[0049] 图1为本申请实施例提供的数据采集方法的场景示意图;

[0050] 图2为本申请实施例提供的数据采集方法的流程示意图;

[0051] 图3为本申请实施例通过的业务系统的服务调用链路的示意图;

[0052] 图4为本申请实施例提供的数据采集的过程示意图;

[0053] 图5为本申请实施例提供的数据收集装置的结构示意图;

[0054] 图6为本申请实施例提供的电子设备的结构示意图。

[0055] 通过上述附图,已示出本申请明确的实施例,后文中将有更详细的描述。这些附图和文字描述并不是为了通过任何方式限制本申请构思的范围,而是通过参考特定实施例为本领域技术人员说明本申请的概念。

具体实施方式

[0056] 为使本申请实施例的目的、技术方案和优点更加清楚,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。

[0057] 首先对本申请所涉及的名词进行解释:

[0058] 微服务:是指一种软件开发技术-面向服务的体系结构架构样式的一种变体,它提倡将单一应用程序划分成一组小的服务,服务之间互相协调、互相配合,为用户提供最终价值。每个服务运行在其独立的进程中,服务与服务间采用轻量级的通信机制互相沟通。每个服务都围绕着具体业务进行构建,并且能够独立地部署到生产环境、类生产环境等。

[0059] 图1为本申请实施例提供的数据采集方法的场景示意图,基于微服务架构的业务系统包括有多个微服务,微服务按照不同的维度和定位进行拆分,互相组合和依赖,如图1所示,各个微服务可以由不同团队开发并部署在不同区域(例如不同城市)的机房服务器

内,微服务与微服务之间采用通信机制互相沟通(例如超文本协议(HTTP)的表示性状态转移(Representation State Transfer,REST)应用程序接口(Application Programming Interface,API))。

[0060] 在实际生活应用中,在用户从终端设备(例如计算机设备或移动终端)的客户端发起请求时,该请求可能会调用多个微服务例如微服务A、微服务B和微服务C,通常业务系统往往有数量众多的微服务并且这些微服务可能部署在不同区域的机房中。当业务系统发生故障时,由于一个请求可能需要调用很多个服务,而服务的调用复杂性导致了故障很难去定位,现有技术通常是通过查看日志来进行故障定位和解决。这种方式由于微服务由不同团队开发并部署在不同区域的机房中,通常日志获取较为困难,即使获取到了日志,由于服务之间存在的复杂调用关系,也没有办法快速发现问题,不能判断系统发生故障产生的影响范围,不支持梳理微服务调用关系以及多模块间依赖的合理性。

[0061] 针对上述问题,本申请实施例提供了一种数据采集方法、装置、设备和可读存储介质,通过跟踪微服务调用链路,将所有跨应用的所有调用链进行信息服务端埋点手机,可以得到调用链路的上下文信息、链路ID、调用开始时间、调用方IP、端口、调用耗时等,可适配任何系统度量整体和依赖关系,根据收集的信息进行统计分析,方便找到故障产生的源头,生产上可极大缩短故障排除时间,也可进一步对系统的升级优化进行全方位的检查等。

[0062] 下面,通过具体实施例对本申请的技术方案进行详细说明。需要说明的是,下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。

[0063] 图2为本申请实施例提供的数据采集方法的流程示意图,该方法具体可以应用于基于微服务架构的业务系统,该业务系统包括有至少一个微服务。

[0064] 如图2所示,该方法具体可以包括如下步骤:

[0065] S201、获取业务系统在处理目标请求时目标请求所调用的至少一个微服务。

[0066] 示例性的,业务系统可以是资产证券化(Asset-backed Securities,ABS)系统,其包括的微服务可以有数据处理服务、ABS资产服务、业务流程服务、基础配置服务、报表服务和调度服务等。各个微服务之间可能存在有关联依赖关系,例如ABS资产服务与数据处理服务之间存在依赖关系。其中,用户可以通过客户端发起目标请求,例如资产查询请求,业务系统基于用户发起的目标请求,确定所需调用的微服务有哪些,然后通过各个微服务对目标请求进行处理,得到最终数据反馈给用户。业务系统作为后台服务系统,可以部署在各个机房的服务器中以获取不同用户发起的请求并进行处理。

[0067] 在本实施例中,当目标请求调用的微服务的数量为两个以上时,则存在有调用先后顺序,示例性的,目标请求会先发送至微服务A以进行处理,在微服务A完成请求处理之后,再将请求传送给微服务B以继续进行处理,在微服务B完成请求处理之后,可以将请求继续传送给微服务C,由此类推,直到完成整个目标请求的处理。例如一个用户发起的请求可能会先达到前端A,然后通过远程调用达到业务系统的中间件B和C(例如负载均衡、网关),最后达到后端服务D和E,后端服务通过一系列逻辑计算最终将数据返回给用户,这个过程中用户的请求就会涉及到非常多的微服务的调用。

[0068] S202、根据目标请求所调用的至少一个微服务,构建服务调用链路。

[0069] 示例性的,不同的请求所调用的微服务可能不同,例如当目标请求为查阅资产数

据请求时,则其需要调用的微服务有ABS资产服务和数据处理服务。其中,ABS资产服务与数据处理服务之间可以通过API建立通信关系以实现目标请求的处理。

[0070] 在本实施例中,服务调用链路指示出了目标请求所调用的微服务以及所调用的各个微服务之间的先后顺序,通过所调用的微服务以及各个服务之间的先后顺序,即可以构建得到服务调用链路。示例性的,例如该目标请求先通过微服务A处理,在微服务A处理完之后再流转 to 微服务B处理,微服务B处理完之后又流转 to 微服务C,直到整个目标请求处理完成,得到最终的数据返回给用户,则构建得到的服务调用链路可以为A-B-C。示例性的,图3为本申请实施例通过的业务系统的服务调用链路的示意图,如图3所示,以资产证券化系统作为业务系统为例,其可以包括有数据处理服务smart、资产服务smart、业务流程服务smart、基础配置服务smart、报表服务smart和调度服务smart。其中,数据处理服务smart可以与资产服务smart之间可以进行通信以进行数据处理,资产服务smart、业务流程服务smart、调度服务smart和报表服务smart均可以与基础配置服务smart通信以获取基础配置,调度服务smart可以异步调用资产服务smart和业务流程服务smart。进一步的,ABS系统之外还可以包括视图服务,视图服务可以从资产服务smart和基础配置服务smart获取展示数据。

[0071] S203、根据服务调用链路进行数据采集,获取链路数据。

[0072] 其中,链路数据用于业务系统的故障定位。链路数据至少包括有服务调用链路的上下文信息、链路标识、调用开始时间和耗时、调用方网络协议地址和端口中的至少一种。其中,上下文信息是调用过程中附带的环境信息,如方法名、参数类型、真实参数、本端/对端地址等。链路标识可以是用于标识服务调用链路,调用方网络协议地址即调用方IP。

[0073] 在本实施例中,通过链路数据可以度量业务系统整体和依赖关系,当业务系统发生故障时,通过统计分析链路数据即可以找到故障产生的源头,可极大缩短故障排除时间,也可进一步对系统的升级优化进行全方位的检查等。

[0074] 示例性的,可以通过前端界面功能根据链路标识ID,可实时查询发生错误的服务调用链路,然后根据发生错误的服务调用链路中所包含的微服务,定位出故障处于哪一个节点。

[0075] 其中,链路数据可以通过埋点的方式采集得到,即在业务系统的各个微服务中植入埋点组件以实现数据的采集。其中,埋点又称为事件追踪,指的是针对特定用户行为或事件进行捕获,处理和发送的相关技术及其实施过程。示例性的,继续参考图3,可以在各个数据处理服务中植入数据收集装置服务smart作为埋点组件以实现数据的采集。

[0076] 本申请实施例通过进行服务链路跟踪,将所有跨应用的所有调用链进行信息服务端埋点实现链路数据的收集,得到调用链路的上下文信息、链路ID、调用开始时间、调用方IP、端口、调用耗时等,可适配任何系统度量整体和依赖关系,根据收集的信息进行统计分析,在业务系统发生故障时方便找到故障产生的源头,生产上可极大缩短故障排除时间,也可进一步对业务系统的升级优化进行全方位的检查等。

[0077] 在一些实施例中,上述步骤S201具体可以通过如下步骤实现:

[0078] 根据各个微服务中的埋点组件获取链路数据。

[0079] 其中,每个微服务包括有至少一个埋点组件,埋点组件用于进行链路数据的采集。埋点组件可以简单理解为一段程序代码,业务系统的各个微服务中均可以植入埋点组件以

实现链路数据的获取。

[0080] 在本实施例中,收集链路数据的埋点组件可以基于底层框架实现无侵入式部署于微服务中,并将微服务所涉及的业务逻辑和性能测量完全分离,植入的埋点组件可以收集到任意类和任意方法的执行时间、调用过程等。示例性的,在植入埋点组件至微服务时,实现的程序通过底层代码自动捕捉需要埋点的代码,无需修改业务系统其它代码即可以实现链路数据的收集。

[0081] 本申请实施例通过在微服务中埋点以进行链路数据的埋点采集,能够基于底层框架实现无侵入式部署,将数据收集与微服务的业务逻辑完全分离,能够准确高效的获取到链路数据并进行上报,避免使用日志来进行信息记录,降低了数据的获取难度和所花费的时间,当业务系统出现故障时,可以直接根据上报的链路数据实现故障排查,提高故障排查的效率。

[0082] 示例性的,在一些实施例中,目标请求可以调用一个或两个以上的微服务,其中,当目标请求调用的微服务为一个时,则服务调用链路中只包含有该被调用的微服务,而不包含微服务之间的调用关系。

[0083] 进一步的,在一些实施例中,当目标请求调用的微服务为两个以上时,上述步骤S202具体可以通过如下步骤实现:

[0084] 获取目标请求所调用的各个微服务之间的调用先后顺序;

[0085] 根据目标请求所调用的各个微服务和调用先后顺序,构建服务调用链路。

[0086] 在本实施例中,当目标请求调用的微服务为两个以上时,各个微服务之间存在有参与调用的先后顺序(即调用先后顺序),例如微服务A先参与对目标请求的处理,当微服务A完成处理之后再传送该目标请求给微服务B,由微服务B继续进行处理。此时,微服务A先被调用,而微服务B后被调用。

[0087] 示例性的,在目标请求的处理过程中,可以通过服务链路追踪确定所调用的微服务的先后顺序。

[0088] 本申请实施例通过进行服务链路跟踪,构建服务调用链路,可以从系统整体维度到局部维度展示各项指标,收集出应用服务的拓扑关系,方便快捷搞清楚应用架构,在业务系统出现故障时能够结合服务调用链路快速的实现问题的定位,提高故障排查效率。

[0089] 进一步的,在一些实施例中,上述“获取目标请求所调用的各个微服务之间的调用先后顺序”具体可以通过如下步骤实现:

[0090] 在目标请求调用当前微服务时,获取当前微服务的标识信息;

[0091] 在目标请求完成当前微服务的调用并继续调用下一微服务时,将当前微服务的标识信息传输至下一微服务以进行记录;

[0092] 在目标请求完成各个微服务的调用时,根据各个微服务记录的标识信息,确定各个微服务之间的调用先后顺序。

[0093] 在本实施例中,标识信息可以用span_id表示,span_id可以是一个64位ID唯一标识,即不同的微服务的标识信息不相同,span_id用于记录调用父子关系,当一次请求在各个微服务之间流转时,上一个微服务的span_id会透传至下一个微服务,每一个微服务会记录上一个微服务的span_id和自己的span_id,在目标请求完成整个处理流程之后,可以直接根据各个微服务记录下的span_pid确定出目标请求所调用的微服务和各个微服务的调

用先后顺序。

[0094] 本申请实施例通过记录标识信息,当目标请求在各个微服务之间流转时,能够基于标识信息确定出目标请求调用的微服务以及各个微服务之间的调用父子关系,得到完整的服务调用链路,方便后续使用服务调用链路来进行故障的排查。

[0095] 进一步的,在一些实施例中,在确定目标请求所调用的微服务时,上述方法还可以包括如下步骤:

[0096] 获取全局标识,在目标请求调用当前微服务时,将全局标识传输至当前微服务以进行记录;

[0097] 在目标请求调用下一微服务时,将全局标识传输至下一微服务以进行记录;

[0098] 根据各个微服务中记录的全局标识,确定目标请求所调用的微服务。

[0099] 其中,全局标识用于对目标请求进行标识。

[0100] 在本实施例中,全局标识可以通过trace_id表示,每一个请求在开始调用微服务时会生成一个全局的trace_id,当请求在各个微服务之间流转时,该trace_id也会被微服务中的埋点组件采集到,最后通过trace_id串联起整个服务调用链路。其中,每一个请求都必须透传trace_id和span_id进行上下文传输。在查看某一次完整的调用时,只需要根据trace_id即可查出所有调用记录,包括该请求所调用的微服务,然后通过span_id组织起调用父子关系即可。

[0101] 其中,不同的请求所生成的trace_id不相同。示例性的,trace_id可以为64位bit。

[0102] 本申请实施例通过设置唯一全局标识对服务调用链路进行标识,当业务系统出现故障时,只需要根据唯一全局标识即可实时查询到发生故障的服务调用链路,实现故障的快速定位。

[0103] 进一步的,在一些实施例中,上述步骤“获取全局标识”具体可以通过如下步骤实现:

[0104] 获取目标请求的请求时间戳和当前微服务的网络协议地址;

[0105] 根据请求时间戳和当前微服务的网络协议地址,生成全局标识。

[0106] 在本实施例中,可以采用雪花算法64位bit作为trace_id.trace_id可以通过当前的请求时间以及当前微服务的IP地址拼装得到,示例性的,在一些实施方式中,在同一个机房同一毫秒内,若存在有多个请求,则在生成每一个请求的唯一trace_id时,可以根据每个请求的序号、请求时间戳以及当前微服务的IP进行拼装得到唯一trace_id。其中,当trace_id为64位bit时,请求的序号作为最后12个bit。

[0107] 本申请实施例通过利用时间戳以及IP来拼装得到全局标识,保证全局标识的唯一性,即使是在同一毫秒内存在多个请求,也可以基于请求的序号拼装得到唯一全局标识,保证不同请求的服务调用链路的全局标识不相同,避免不同服务调用链路之间发生混淆,当业务系统出现故障时,能够快速找到发生故障的服务调用链路。

[0108] 在一些实施例中,上述方法还包括如下步骤:

[0109] 当业务系统发出故障提示时,根据链路数据进行故障排查,得到排查结果。

[0110] 其中,排查结果用于指示业务系统发生故障的位置。在本实施例中,不同的目标请求的trace_id(即全局标识)是不同的,而不同的目标请求有不同的服务调用链路,当业务系统出现故障时,可以直接根据trace_id来找到发生错误的服务调用链路,从服务调用链

路中定位出发生异常的节点,实现对故障的排查。

[0111] 本申请实施例通过使用链路数据进行故障排查,在业务系统出现故障时,可以实时查询发生故障的服务调用链路,定位服务调用链路中的异常节点,实现故障的快速排查定位。

[0112] 在一些实施例中,上述方法还可以包括如下步骤:

[0113] 响应于业务系统中预设开关的开启或关闭,控制各个微服务中的埋点组件的启动或停止。

[0114] 其中,埋点组件用于进行链路数据的收集。在本实施例中,可以通过微服务程序的配置开关来控制埋点组件的启动或关闭,当开关开启时,则埋点组件启动,实现链路数据的采集,当开关关闭时,则埋点组件关闭,不进行链路数据的采集。通过设置动态开关,可以实现链路数据采集服务的可插拔式控制,实现链路数据的灵活采集。

[0115] 示例性的,在一些实施例中,可以通过守护线程进行监听服务链路调用的日志收集并进行异步消息的发送,其中守护线程是指在程序运行的时候在后台提供一种通用服务的线程。

[0116] 在一些实施例中,上述方法还可以包括如下步骤:

[0117] 获取业务系统中预设的白名单路径;

[0118] 根据埋点组件在白名单路径中采集得到的数据,获取链路数据。

[0119] 其中,白名单路径用于指示埋点组件进行数据采集,微服务中包括有埋点组件,埋点组件用于进行数据采集。

[0120] 在本实施例中,埋点组件在进行链路数据的采集时,只采集白名单路径目录中的数据,比如com.aaa.xxx。示例性的,可以在业务系统中预先自定义配置白名单路径和黑名单路径,实现黑名单路径和白名单路径的过滤,对白名单路径进行分类拦截,具体可以包括远程调用、消息队列和异步任务等。

[0121] 示例性的,对于远程调用的白名单路径,可以实现生产者与消费者的区分,对于生产者如有多层循环链路进行深层次的追踪和记录的保存。其中,深层即多层的链路调用,每一次超文本传输协议(Hyper Text Transfer Protocol,HTTP)或远程过程调用(Remote Procedure Call,RPC)接口的调用,异步消息队列都可以进行链路数据的收集。其中,生产者为微服务的被调用方,即为其他微服务提供服务的微服务,消费者为微服务的调用方,即依赖其他微服务的微服务。

[0122] 本申请实施例通过设置白名单路径,可以支持客户端自定义选择想要收集哪些目录的埋点信息,即只采集白名单路径的目录下的埋点信息,实现数据针对性采集。

[0123] 在一些实施例中,上述方法还可以包括如下步骤:

[0124] 根据服务调用链路,获取各个微服务之间的拓扑关系;

[0125] 根据拓扑关系,构建业务系统的服务框架。

[0126] 其中,服务框架用于指示业务系统所包含的微服务以及各个微服务的关联关系。

[0127] 在本实施例中,在请求调用微服务完成后,可以获取到每一个请求的服务调用链路,通过每一个请求的服务调用链路,即可以梳理得到各个微服务之间的拓扑关系。示例性的,可以参见上述图3,以ABS业务系统包含有数据处理服务smart、资产服务smart、业务流程服务smart、基础配置服务smart、调度服务smart和报表服务smart为例,通过梳理各个微

服务之间的拓扑关系,即可以得到整个ABS业务系统的服务架构。

[0128] 本申请实施例通过服务调用链路获取各个微服务之间的拓扑关系和服务框架,在排查业务系统的故障时,可以通过各个微服务之间的拓扑关系和服务框架,快速梳理服务调用关系,判断出业务系统故障发生的影响范围。同时也方便统计业务系统的吞吐量、相应时间和错误记录等。

[0129] 在一些实施例中,上述方法还可以包括如下步骤:

[0130] 根据消息队列机制,将服务调用链路异步发送存储至预设数据库中。

[0131] 在本实施例中,预设数据库可以是MySQL关系型数据库,其可以通过功能实现提供接口。示例性的,可以通过接口来获取预设数据库中存储的服务调用链路,并在可视化的界面上进行显示。本申请实施例通过消息队列机制异步存储服务调用链路,能够达到不影响业务系统解耦操作,避免影响业务系统的正常运行。

[0132] 图4为本申请实施例提供的数据采集的过程示意图,如图4所示,微服务通常被部署在服务器中并通过容器实现隔离。微服务包括有应用程序、服务端配置和网关等,应用程序进行包的依赖引入以实现埋点,通过服务端配置中的开关配置可以启动埋点数据采集服务,收集白名单路径的目录下的相关埋点指标数据得到链路数据,异步发送至数据库存储,在链路数据上报时可以通过链路标识span_id来确定微服务的父子调用关系,并通过唯一全局标识trace_id来对整个服务调用链路进行标识,然后通过日志服务将链路数据转发至追踪控制台,追踪控制台通过代码实现展示出来链路关系和指标并进行拓扑分析,界面产品化大屏可以实现链路关系的可视化,实现对业务系统的故障排查定位。其中,唯一全局标识trace_id可以通过雪花算法生成得到。

[0133] 下述为本申请装置实施例,可以用于执行本申请方法实施例。对于本申请装置实施例中未披露的细节,请参照本申请方法实施例。

[0134] 图5为本申请实施例提供的数据收集装置的结构示意图,该数据采集装置可以集成在服务器中的微服务中,如图5所示,该数据采集装置50可以包括服务获取模块51、链路构建模块52、数据获取模块53。

[0135] 其中,服务获取模块51用于获取业务系统在处理目标请求时目标请求所调用的微服务。链路构建模块52用于根据目标请求所调用的至少一个微服务,构建服务调用链路。数据获取模块53用于根据服务调用链路进行数据采集,获取链路数据。

[0136] 其中,目标请求可调用至少一个微服务,链路数据至少包括有服务调用链路的上下文信息、链路标识、调用开始时间和耗时、调用方网络协议地址和端口中的至少一种。

[0137] 在一些实施例中,上述数据获取模块具体可以用于根据各个微服务中的埋点组件获取链路数据。

[0138] 其中,每个微服务包括有至少一个埋点组件,埋点组件用于进行链路数据的采集。

[0139] 在一些实施例中,若目标请求调用至少两个微服务,则上述链路构建模块具体可以用于:

[0140] 获取目标请求所调用的各个微服务之间的调用先后顺序;

[0141] 根据目标请求所调用的各个微服务和调用先后顺序,构建服务调用链路。

[0142] 可选的,在一些实施例中,上述链路构建模块具体可以用于:

[0143] 在目标请求调用当前微服务时,获取当前微服务的标识信息;

- [0144] 在目标请求完成当前微服务的调用并继续调用下一微服务时,将当前微服务的标识信息传输至下一微服务以进行记录;
- [0145] 在目标请求完成各个微服务的调用时,根据各个微服务记录的标识信息,确定各个微服务之间的调用先后顺序。
- [0146] 可选的,在一些实施例中,上述数据采集装置还可以包括全局标识模块,用于:
- [0147] 获取全局标识,在目标请求调用当前微服务时,将全局标识传输至当前微服务以进行记录,全局标识用于对目标请求进行标识;
- [0148] 在目标请求调用下一微服务时,将全局标识传输至下一微服务以进行记录;
- [0149] 根据各个微服务中记录的全局标识,确定目标请求所调用的微服务。
- [0150] 在一些实施例中,上述全局标识模块具体还可以用于:
- [0151] 获取目标请求的请求时间戳和当前微服务的网络协议地址;
- [0152] 根据请求时间戳和当前微服务的网络协议地址,生成全局标识。
- [0153] 在一些实施例中,上述数据采集装置还包括故障排查模块,用于:
- [0154] 当业务系统发出故障提示时,根据链路数据进行故障排查,得到排查结果,排查结果用于指示业务系统发生故障的位置。
- [0155] 在一些实施例中,上述数据采集装置还可以包括开关控制模块,用于响应于业务系统中预设开关的开启或关闭,控制各个微服务中的埋点组件的启动或停止。
- [0156] 其中,埋点组件用于进行链路数据的收集。
- [0157] 在一些实施例中,上述数据采集装置还可以包括名单获取模块,用于:
- [0158] 获取业务系统中预设的白名单路径;
- [0159] 根据埋点组件在白名单路径中采集得到的数据,获取链路数据。
- [0160] 其中,白名单路径用于指示埋点组件进行数据采集,微服务中包括有埋点组件,埋点组件用于进行数据采集。
- [0161] 在一些实施例中,上述数据采集装置还包括关系获取模块,用于:
- [0162] 根据服务调用链路,获取各个微服务之间的拓扑关系;
- [0163] 根据拓扑关系,构建业务系统的服务框架。
- [0164] 其中,服务框架用于指示业务系统所包含的微服务以及各个微服务的关联关系。
- [0165] 在一些实施例中,上述数据采集装置还包括存储模块,用于根据消息队列机制,将服务调用链路异步发送存储至预设数据库中。
- [0166] 本申请实施例提供的装置,可用于执行上述实施例中的方法,其实现原理和技术效果类似,在此不再赘述。
- [0167] 需要说明的是,应理解以上装置的各个模块的划分仅仅是一种逻辑功能的划分,实际实现时可以全部或部分集成到一个物理实体上,也可以物理上分开。且这些模块可以全部以软件通过处理元件调用的形式实现;也可以全部以硬件的形式实现;还可以部分模块通过处理元件调用软件的形式实现,部分模块通过硬件的形式实现。例如,服务获取模块可以以程序代码的形式存储于上述装置的存储器中,由上述装置的某一个处理元件调用并执行以上服务获取模块的功能。其它模块的实现与之类似。
- [0168] 图6为本申请实施例提供的电子设备的结构示意图。如图6所示,该电子设备60包括:至少一个处理器61和与处理器61连接的存储器62。示例性的,该电子设备可以是一个服

务器或多个服务器组成。其中处理器61、通信接口64以及存储器62通过总线63完成相互间的通信。通信接口64用于与其它设备进行通信,该通信接口包括用于进行数据传输的通信接口。

[0169] 处理器61,用于执行存储器62中存储的计算机指令,具体可以执行上述实施例中所述的方法中的相关步骤。处理器可能是中央处理器。电子设备60包括的一个或多个处理器。

[0170] 存储器62,用于存放计算机指令。存储器可能包含高速RAM存储器,也可能还包括非易失性存储器,例如至少一个磁盘存储器。

[0171] 本实施例还提供一种可读存储介质,可读存储介质中存储有计算机指令,当电子设备的至少一个处理器执行该计算机指令时,电子设备执行上述的各种实施方式提供的数据采集方法。

[0172] 本实施例还提供一种程序产品,该程序产品包括计算机指令,该计算机指令存储在可读存储介质中。电子设备的至少一个处理器可以从可读存储介质读取该计算机指令,至少一个处理器执行该计算机指令使得电子设备实施上述的各种实施方式提供的数据采集方法。

[0173] 本申请中,“至少一个”是指一个或者多个,“多个”是指两个或两个以上。“和/或”,描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B的情况,其中A,B可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系;在公式中,字符“/”,表示前后关联对象是一种“相除”的关系。“以下至少一项(个)”或其类似表达,是指的这些项中的任意组合,包括单项(个)或复数项(个)的任意组合。例如,a,b,或c中的至少一项(个),可以表示:a,b,c,a-b,a-c,b-c,或a-b-c,其中,a,b,c可以是单个,也可以是多个。

[0174] 可以理解的是,在本申请实施例中涉及的各种数字编号仅为描述方便进行的区分,并不用来限制本申请的实施例的范围。在本申请的实施例中,上述各过程的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不应对本申请的实施例的实施过程构成任何限定。

[0175] 最后应说明的是:以上各实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述各实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

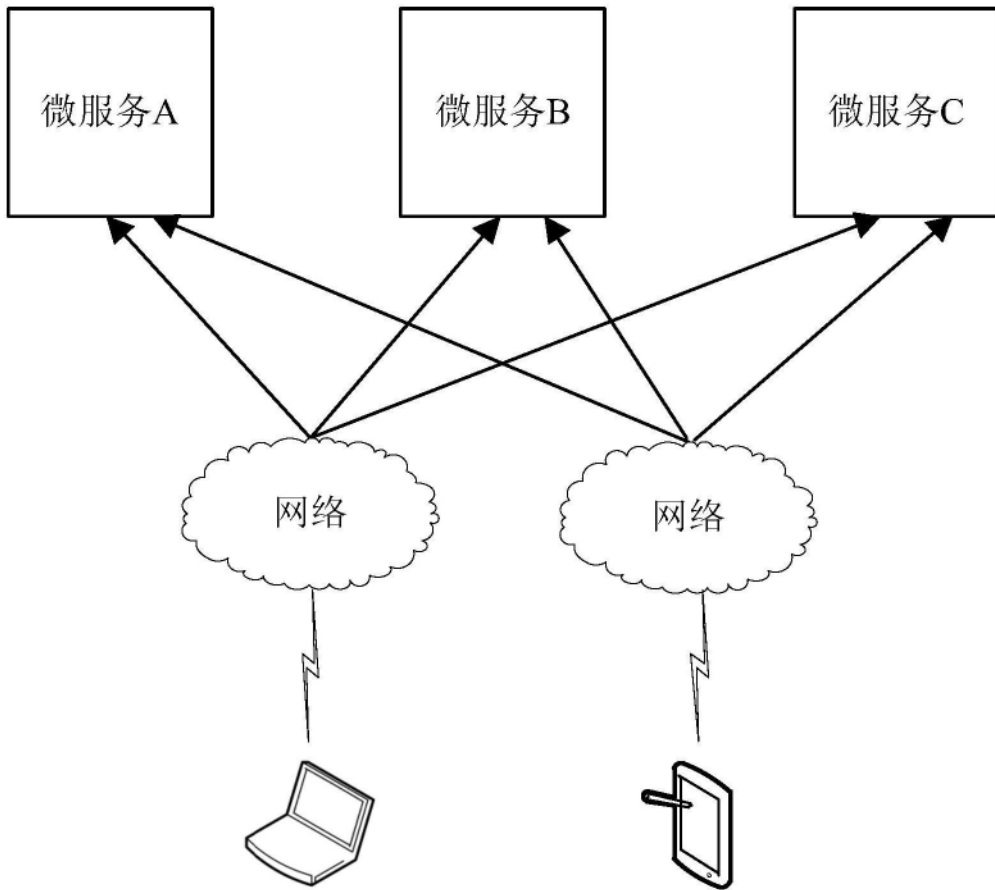


图1

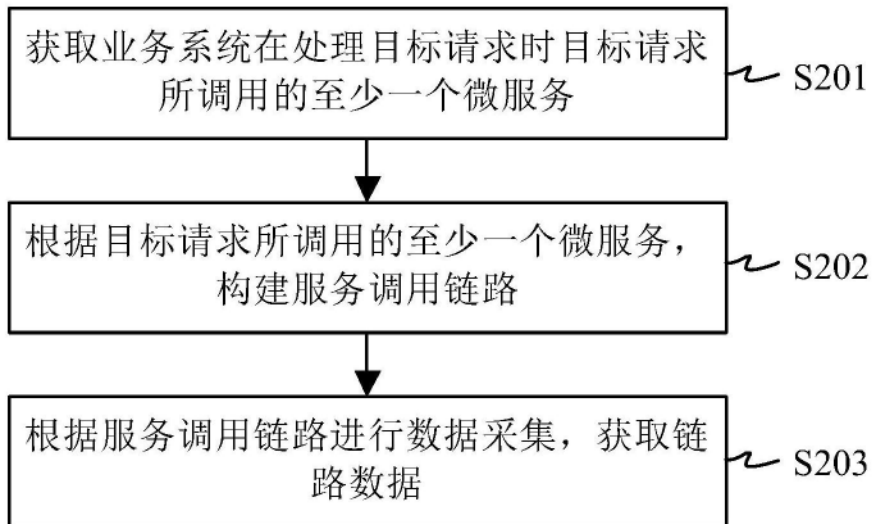


图2

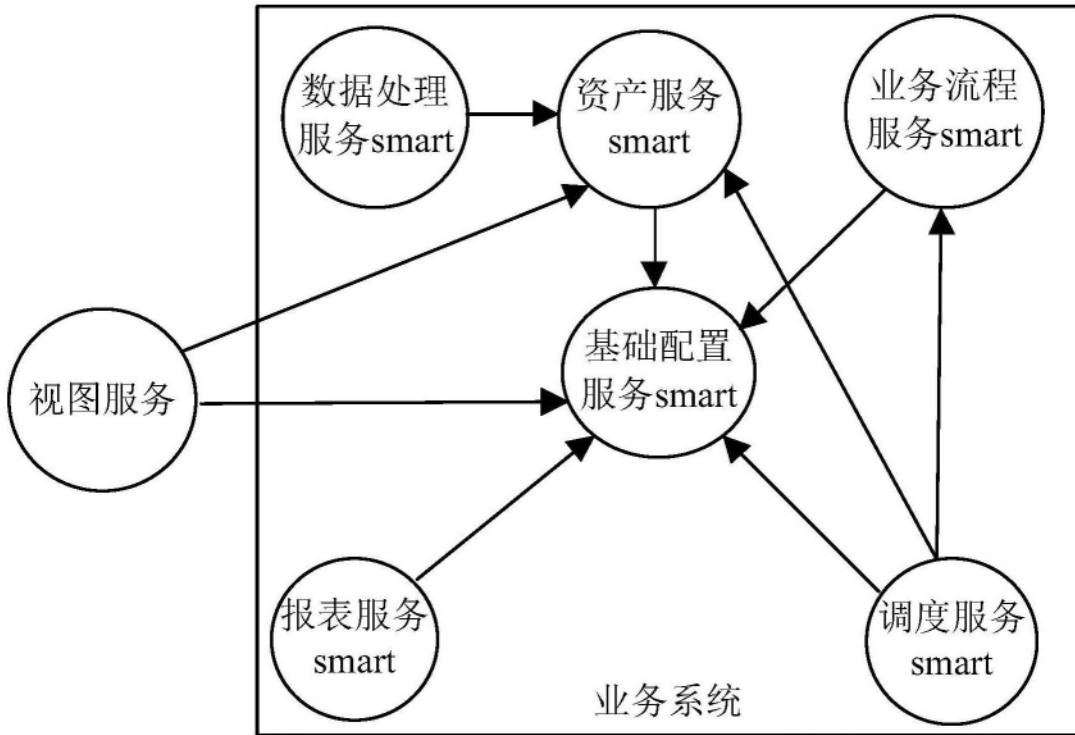


图3

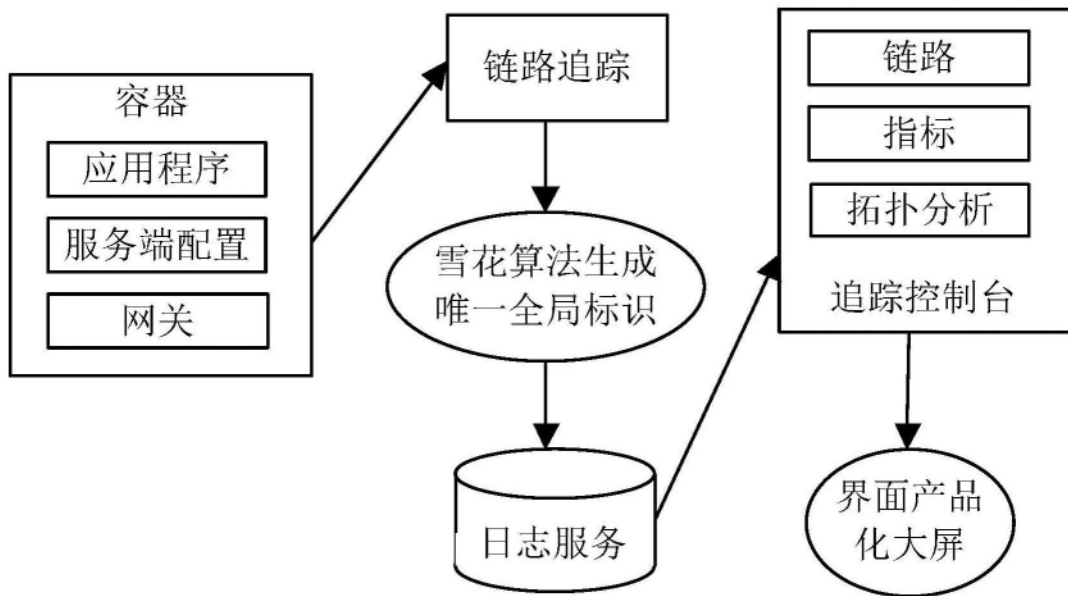


图4

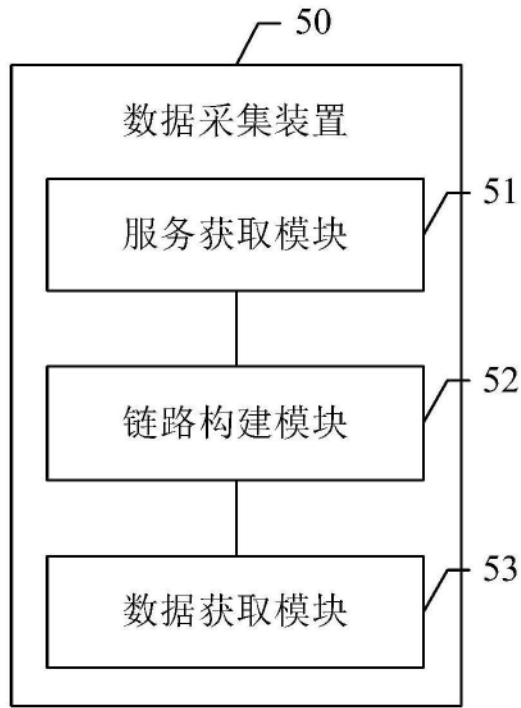


图5

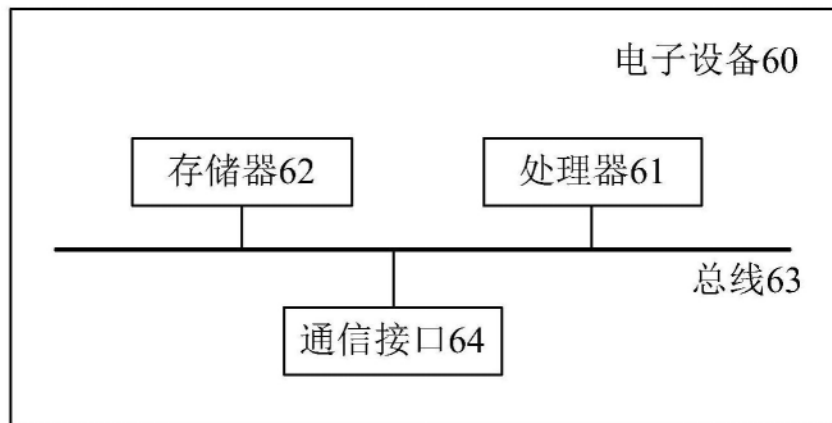


图6