



(12)发明专利申请

(10)申请公布号 CN 112866319 A

(43)申请公布日 2021.05.28

(21)申请号 201911191616.0

G06F 16/18(2019.01)

(22)申请日 2019.11.28

(71)申请人 顺丰科技有限公司

地址 518000 广东省深圳市南山区学府路
软件产业基地1栋B座8楼

(72)发明人 戴婧 文勇 赵甜 黄强 李正新
张文英 曾媛媛

(74)专利代理机构 广州华进联合专利商标代理
有限公司 44224

代理人 杨欢

(51)Int.Cl.

H04L 29/08(2006.01)

H04L 29/06(2006.01)

H04L 12/851(2013.01)

H04L 12/801(2013.01)

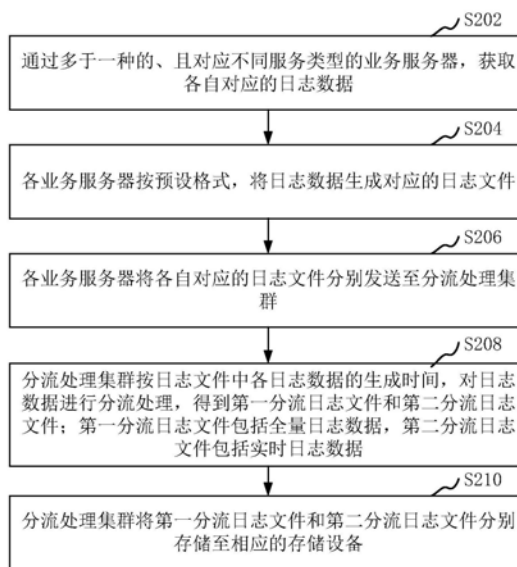
权利要求书2页 说明书11页 附图3页

(54)发明名称

日志数据处理方法、系统和存储介质

(57)摘要

本申请涉及一种日志数据处理方法、系统和存储介质。所述方法包括:通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据;各业务服务器按预设格式,将日志数据生成对应的日志文件;各业务服务器将各自对应的日志文件分别发送至分流处理集群;分流处理集群按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;第一分流日志文件包括全量日志数据,第二分流日志文件包括实时日志数据;分流处理集群将第一分流日志文件和第二分流日志文件分别存储至相应的存储设备。采用本方法能够提升日志数据处理效率。



1. 一种日志数据处理方法,其特征在于,所述方法包括:
通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据;
各所述业务服务器按预设格式,将所述日志数据生成对应的日志文件;
各所述业务服务器将各自对应的日志文件分别发送至分流处理集群;
所述分流处理集群按所述日志文件中各日志数据的生成时间,对所述日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;所述第一分流日志文件包括全量日志数据,所述第二分流日志文件包括实时日志数据;
所述分流处理集群将所述第一分流日志文件和所述第二分流日志文件分别存储至相应的存储设备。
2. 根据权利要求1所述的方法,其特征在于,所述不同服务类型的业务服务器包括通用服务器和微服务器,所述通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据,包括:
通过所述通用服务器获取用户终端和所述通用服务器本地所生成的日志数据;
通过所述微服务器获取所述微服务器本地所生成的日志数据。
3. 根据权利要求2所述的方法,其特征在于,所述通过所述通用服务器获取用户终端和所述通用服务器本地所生成的日志数据,包括:
所述通用服务器接收不同用户终端所发送的日志数据;
所述通用服务器获取所述通用服务器本地所生成的日志数据;
所述用户终端发送日志数据的步骤包括:
所述用户终端通过预先融合有日志数据拦截器的前端页面框架,获取所述用户终端产生的日志数据;
所述用户终端将所述用户终端产生的日志数据异步传输给所述通用服务器。
4. 根据权利要求1所述的方法,其特征在于,所述各所述业务服务器按预设格式,将所述日志数据生成对应的日志文件,包括:
各所述业务服务器根据预设业务需求,确定相同的预设格式;
各所述业务服务器按所述相同的预设格式中的预设字段和预设字段的排序,将所述日志数据生成对应的日志文件。
5. 根据权利要求1所述的方法,其特征在于,所述各所述业务服务器将各自对应的日志文件分别发送至分流处理集群,包括:
各所述业务服务器通过采集代码监测各自对应的日志文件;
当各所述业务服务器监测到各自对应的日志文件更新时,将更新后的各所述业务服务器各自对应的日志文件分别发送至分流处理集群。
6. 根据权利要求1所述的方法,其特征在于,所述分流处理集群按所述日志文件中各日志数据的生成时间,对所述日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件,包括:
所述分流处理集群根据所述预设格式中的时间字段,确定所述日志文件中各日志数据的生成时间;
所述分流处理集群将所述日志文件中各日志数据的生成时间进行降序排序,得到对应的时间排序结果;

所述分流处理集群根据所述时间排序结果和预设待存储字段,从所述日志文件的各日志数据中,筛选出符合第一预设条件的第一分流日志数据,并根据所述第一分流日志数据生成第一分流日志文件;

所述分流处理集群根据所述时间排序结果,从所述日志文件的各日志数据中,筛选出符合第二预设条件的第二分流日志数据,并根据所述第二分流日志数据生成第二分流日志文件。

7. 根据权利要求1至6中任一项所述的方法,其特征在于,所述存储设备包括全量存储设备和实时存储设备,所述分流处理集群将所述第一分流日志文件和所述第二分流日志文件分别存储至相应的存储设备,包括:

所述分流处理集群分别获取所述全量存储设备的地址和所述实时存储设备的地址;

所述分流处理集群根据所述全量存储设备的地址,将所述第一分流日志文件存储至所述全量存储设备的物理内存中;

所述分流处理集群根据所述实时存储设备的地址,将所述第二分流日志文件存储至所述实时存储设备。

8. 一种日志数据处理系统,其特征在于,所述系统包括多于一种的、且对应不同服务类型的业务服务器、分流处理集群和存储设备,

各业务服务器,用于获取各自对应的日志数据;

各业务服务器,还用于按预设格式,将所述日志数据生成对应的日志文件;

各业务服务器,还用于将各自对应的日志文件分别发送至分流处理集群;

分流处理集群,用于按所述日志文件中各日志数据的生成时间,对所述日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;所述第一分流日志文件包括全量日志数据,所述第二分流日志文件包括实时日志数据;

分流处理集群,还用于将所述第一分流日志文件和所述第二分流日志文件分别存储至相应的存储设备。

9. 根据权利要求8所述的系统,其特征在于,所述各业务服务器还用于获取用户终端和所述通用服务器本地所生成的日志数据;获取所述微服务器本地所生成的日志数据。

10. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的方法的步骤。

日志数据处理方法、系统和存储介质

技术领域

[0001] 本申请涉及计算机技术领域,特别是涉及一种日志数据处理方法、系统和存储介质。

背景技术

[0002] 随着计算机技术的发展,各公司提供的服务功能越来越复杂,对应的系统服务器也越来越多,日志数据的有效处理,对统一管理和维护各系统各模块起着至关重要的作用。在传统技术中,针对不同的系统和不同的服务功能,对应的日志数据也存储在不同的服务器中。在需要查看日志数据时,可依次遍历所有的服务器进行相关日志数据的查询。

[0003] 然而,目前的日志数据处理方法,在日志数据查询、问题快速定位等方面操作复杂,从而导致了日志数据处理效率低。

发明内容

[0004] 基于此,有必要针对上述技术问题,提供一种能够提升日志数据处理效率的日志数据处理方法、系统和存储介质。

[0005] 一种日志数据处理方法,所述方法包括:

[0006] 通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据;

[0007] 各所述业务服务器按预设格式,将所述日志数据生成对应的日志文件;

[0008] 各所述业务服务器将各自对应的日志文件分别发送至分流处理集群;

[0009] 所述分流处理集群按所述日志文件中各日志数据的生成时间,对所述日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;所述第一分流日志文件包括全量日志数据,所述第二分流日志文件包括实时日志数据;

[0010] 所述分流处理集群将所述第一分流日志文件和所述第二分流日志文件分别存储至相应的存储设备。

[0011] 一种日志数据处理系统,所述系统包括多于一种的、且对应不同服务类型的业务服务器、分流处理集群和存储设备,

[0012] 各业务服务器,用于获取各自对应的日志数据;

[0013] 各业务服务器,还用于按预设格式,将所述日志数据生成对应的日志文件;

[0014] 各业务服务器,还用于将各自对应的日志文件分别发送至分流处理集群;

[0015] 分流处理集群,用于按所述日志文件中各日志数据的生成时间,对所述日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;所述第一分流日志文件包括全量日志数据,所述第二分流日志文件包括实时日志数据;

[0016] 分流处理集群,还用于将所述第一分流日志文件和所述第二分流日志文件分别存储至相应的存储设备。

[0017] 一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现以下步骤:

- [0018] 通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据;
- [0019] 各所述业务服务器按预设格式,将所述日志数据生成对应的日志文件;
- [0020] 各所述业务服务器将各自对应的日志文件分别发送至分流处理集群;
- [0021] 所述分流处理集群按所述日志文件中各日志数据的生成时间,对所述日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;所述第一分流日志文件包括全量日志数据,所述第二分流日志文件包括实时日志数据;
- [0022] 所述分流处理集群将所述第一分流日志文件和所述第二分流日志文件分别存储至相应的存储设备。
- [0023] 上述日志数据处理方法、系统和存储介质,通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据,由于各服务类型的业务服务器各自对应的日志数据有不同的日志格式,进而各业务服务器可按预设格式,将日志数据生成对应的日志文件,使得日志文件中的各日志数据具有相同的日志格式。分流处理集群可按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件,且分别进行存储,使得日志查询操作更便捷,问题定位更快速,进而提升日志数据处理效率。

附图说明

- [0024] 图1为一个实施例中日志数据处理方法的应用场景图;
- [0025] 图2为一个实施例中日志数据处理方法的流程示意图;
- [0026] 图3为一个实施例中用户终端的日志数据获取流程示意图;
- [0027] 图4为一个实施例中日志数据分流处理的流程示意图;
- [0028] 图5为一个实施例中日志数据处理方法的系统关联流程示意图;
- [0029] 图6为一个实施例中计算机设备的内部结构图。

具体实施方式

[0030] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0031] 本申请提供的日志数据处理方法,可以应用于如图1所示的应用环境中。该应用环境包括多于一种的、且对应不同服务类型的业务服务器102、分流处理集群104和存储设备106。各业务服务器102与分流处理集群104通过网络进行通信,分流处理集群104与各存储设备106通过网络进行通信。其中,各业务服务器102可以用独立的服务器或者是多个服务器组成的服务器集群来实现。各存储设备106可以是终端,也可以是服务器。终端具体可以是台式终端或移动终端,移动终端具体可以手机、平板电脑、笔记本电脑等中的至少一种。

[0032] 各业务服务器102获取各自对应的日志数据;各业务服务器102按预设格式,将所述日志数据生成对应的日志文件;各业务服务器102将各自对应的日志文件分别发送至分流处理集群;分流处理集群104按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;分流处理集群104将第一分流日志文件和第二分流日志文件分别存储至相应的存储设备106。本领域技术人员可以理解,图1中示

出的应用环境,仅仅是与本申请方案相关的部分场景,并不构成对本申请方案应用环境的限定。

[0033] 在一个实施例中,如图2所示,提供了一种日志数据处理方法,以该方法应用于图1中的各业务服务器102、分流处理集群104和存储设备106为例进行说明,包括以下步骤:

[0034] S202,通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据。

[0035] 其中,业务服务器是拥有处理对应业务逻辑功能的服务器,不同服务类型的业务服务器具体可以是通用服务器和微服务器。通用服务器是拥有处理用户终端和通用服务器本地业务逻辑功能的服务器。微服务器是拥有处理微服务所对应业务逻辑功能的服务器。日志数据是记录系统中硬件、软件和系统问题的信息,同时还可以监视系统中发生的事件,用于检查错误发生的原因,或者寻找受到攻击时攻击者留下的痕迹。

[0036] 具体地,日志数据处理系统中部署有多于一种的、且对应不同服务类型的业务服务器,通过各业务服务器可获取各业务服务器各自对应的日志数据。

[0037] 在一个实施例中,不同服务类型的业务服务器包括通用服务器和微服务器,步骤S202,也就是通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据的步骤,具体包括:通过通用服务器获取用户终端和通用服务器本地所生成的日志数据;通过微服务器获取微服务器本地所生成的日志数据。

[0038] 具体地,用户操作用户终端的前端页面,用户终端可产生对应的前端日志数据。通用服务器中的应用层服务和管理后台服务,在各自提供对应的服务时可产生日志数据。通过通用服务器可获取用户终端和通用服务器本地所生成的日志数据。微服务器在提供微服务的过程中可产生日志数据,通过微服务器获取微服务器本地所生成的日志数据。这样,可快速获取到日志数据。

[0039] 在一个实施例中,通过通用服务器获取用户终端和通用服务器本地所生成的日志数据的步骤,具体包括:通用服务器接收不同用户终端所发送的日志数据;通用服务器获取通用服务器本地所生成的日志数据;用户终端发送日志数据的步骤包括:用户终端通过预先融合有日志数据拦截器的前端页面框架,获取用户终端产生的日志数据;用户终端将用户终端产生的日志数据异步传输给通用服务器。

[0040] 其中,日志数据拦截器是具有拦截所有正常和异常日志数据的拦截器,用于获取用户在操作用户终端的前端页面时所产生的日志数据。前端页面框架是开发人员在开发网站页面使用的框架,前端页面框架具体可以是Bootstrap网页框架)、AmazeUI (Amaze User Interface,跨屏前端框架)和LayUI (Lay User Interface,前端框架)等,本申请实施例在此对前端页面框架不做限定。

[0041] 具体地,前端页面框架中预先融合有日志数据拦截器,用户在操作用户终端时会产生对应的日志数据,用户终端通过预先融合有日志数据拦截器的前端页面框架,可获取用户终端产生的日志数据。用户终端可将用户在用户终端操作产生的日志数据,通过异步传输给对应的通用服务器,通用服务器可接受来自不同用户终端发送过来的日志数据。通用服务器在提供对应的服务时可产生对应的日志数据,通用服务器可直接获取本地所生成的日志数据。

[0042] 在一个实施例中,如图3所示,通过用户终端的日志数据获取流程示意图可知,用

户通过用户终端可访问页面,页面可从互联网中加载对应于用户操作的资源,当页面将资源加载完成时,用户终端通过预先融合至前端框架中的监听浏览器事件,可获取用户终端产生的所有正常和异常的日志数据。用户终端可通过异步传输,将用户终端产生的日志数据传输得对应的服务器。

[0043] 在一个实施例中,前端页面框架中可添加一个公共的异常拦截器,通过预先融合至前端框架中的异常拦截器,可获取用户在操作用户终端时产生的异常的日志数据。异常拦截器可调用公共代码,访问对应的服务器保存异常的日志数据。这样,可更快速获取到来自不同业务服务器各自对应的所有正常和异常的日志数据。

[0044] S204,各业务服务器按预设格式,将日志数据生成对应的日志文件。

[0045] 具体地,来自不同源的日志数据具有不同的日志配置文件,各业务服务器可按预设格式,将来自不同源的日志数据,生成具有统一格式的日志数据所对应的日志文件。

[0046] 在一个实施例中,日志文件可通过Log4j(Log4j--log for java,JAVA的日志)来配置。Log4j是一个可以配置日志格式的日志组件,Log4j包括三个重要的组件:日志记录器,输出端和日志格式化器。日志记录器用于控制要启用或禁用哪些日志记录语句,并对日志信息进行级别限制。输出端用于指定日志将打印到控制台还是文件中。日志格式化器用于控制日志信息的显示格式。

[0047] 在一个实施例中,日志格式可包括不同的日志字段,日志字段具体可包括日期、时间、日志级别、代码位置、日志内容和错误码等。各业务服务器可根据业务需要,对日志字段的顺序进行自定义设置。本申请在此对日志字段不做限定。

[0048] S206,各业务服务器将各自对应的日志文件分别发送至分流处理集群。

[0049] 其中,分流处理集群是具有对日志数据进行分流处理功能的集群。具体地,各业务服务器存储有各自对应的日志文件,各业务服务器可将各自对应的日志文件分别发送至分流处理集群。

[0050] 在一个实施例中,步骤S206,也就是各业务服务器将各自对应的日志文件分别发送至分流处理集群的步骤,具体包括:各业务服务器通过采集代码监测各自对应的日志文件;当各业务服务器监测到各自对应的日志文件更新时,将更新后的各业务服务器各自对应的日志文件分别发送至分流处理集群。

[0051] 其中,采集代码是具有从对应的日志文件中监测和采集日志数据功能的代码。具体地,各业务服务器中部署有对应的采集代码,日志文件可随着时间的增长更新日志文件中的日志数据。各业务服务器可通过采集代码监测各自对应的日志文件。当各业务服务器监测到各自对应的日志文件更新时,各业务服务器可将更新后的各业务服务器各自对应的日志文件分别发送至分流处理集群。

[0052] 在一个实施例中,采集代码具体可以是通过JAVA(面向对象编程语言)编写的代码,可将采集代码部署在各业务服务器上,并指定该采集代码所监测的路径。可以理解,各业务服务器上所有运行的功能代码,都可记录对应各业务服务器的日志数据,并将记录的日志数据输出至对应的日志文件中。采集代码可监测所有的日志文件,当监测到对应的日志文件后,进行日志数据读取。采集代码可记录当前读取的位置。当监测到下一次日志文件更新时,从上次记录的位置继续读取日志数据。这样,各业务服务器可将最新的日志数据发送至分流处理集群,进一步提升了日志处理效率。

[0053] S208,分流处理集群按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;第一分流日志文件包括全量日志数据,第二分流日志文件包括实时日志数据。

[0054] 其中,第一分流日志文件是历史日志文件,用于记录各业务服务器各自对应的所有日志数据,比如,通过第一分流日志文件,可查询到6个月之前的日志数据。第一分流日志文件包括全量日志数据,全量日志数据是根据预设的日志字段,从对应的日志文件的所有日志数据中筛选出来的所有的日志数据。第二分流日志文件是预设时间段内的实时日志文件,用于记录各业务服务器各自对应的预设时间段内的日志数据,比如,预设时间段为15天,通过第二分流日志文件,只可查询到15天之前的日志数据。第二分流日志文件包括实时日志数据,实时日志数据是根据预设时间段,从日志文件的所有日志数据中筛选出来的日志数据。其中,全量日志数据包括实时日志数据。

[0055] 具体地,日志数据处理系统中部署有分流处理集群,日志文件中各日志数据都包括有各日志数据的生成时间。分流处理集群可按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件。

[0056] 在一个实施例中,分流处理集群对日志数据进行分流处理,具体可以是根据日志文件中各日志数据的生成时间,设定分流时间阈值,进而根据该分流时间阈值,对日志数据进行分流处理。举例说明,分流时间阈值为15天,分流处理集群可根据15天之内和15天之外的全部日志数据,生成第一分流日志文件。分流处理集群可根据近15天之内的日志数据,生成第二分流日志文件。

[0057] 在一个实施例中,分流处理集群可通过KAFKA(分布式发布订阅消息系统)接收来自各业务服务器发送的日志数据,进而可通过FLINK(分布式流数据流引擎)对日志数据进行分流处理。如图4所示,FLINK会自动消费KAFKA中的日志数据,当从KAFKA中读取日志数据后,FLINK通过ACSP_LOG_SOURCE(数据源)组件、TRIM_JSON(计算)组件、NULL_FILTER(计算)组件、ACSP_LOG_BDP_FORMAT(计算)组件、ACSP_LOG_HIVE(SINK)组件和ACSP_LOG_ES(SINK)组件对日志数据进行清洗过滤。可以理解,ACSP_LOG_SOURCE组件可从KAFKA的《ESG_ACSP_CORE_BUS_LOG》主题中消费数据。TRIM_JSON组件可把从KAFKA中消费的每一行日志进行格式化处理,获取其中JSON格式的数据内容,通过JSON工具类转换成一个JSON字符串。NULL_FILTER组件是对格式化处理过程中产生的、数据为空的内容进行拦截,不传递给下游。ACSP_LOG_BDP_FORMAT组件是对JSON数据内容进行拆分,获取预设字段数,比如五个字段,进行保存到HIVE(数据仓库分析系统)数据库中。ACSP_LOG_ES组件是一个第三方的ES(ElasticSearch,分布式全文检索)搜索平台,用于存储日志数据并支持快速搜索。

[0058] S210,分流处理集群将第一分流日志文件和第二分流日志文件分别存储至相应的存储设备。

[0059] 具体地,流处理集群可分别获取第一分流日志文件和第二分流日志文件对应的存储设备的地址。根据各自的存储设备的地址,分流处理集群可将第一分流日志文件和第二分流日志文件分别存储至相应的存储设备。

[0060] 在一个实施例中,存储设备具体可以是HIVE和ES检索引擎。具体地,分流处理集群可将第一分流日志文件存储在HIVE物理存储中,便于后续的全量日志数据统计。比如,可在管理后台服务中做一个页面,通过该页面可查询到任何时间的日志数据,例如,可查询3个

月之前的日志数据。分流处理集群可将第二分流日志文件存储在ES检索引擎中,分流处理集群可设定第二分流日志文件存储在ES检索引擎中的时间周期。比如,第二分流日志文件只能在ES检索引擎中存储15天,15天之后的日志数据将会被自动删除。可在管理后台服务中做一个页面,通过ES检索引擎可查询当前15天的实时日志数据。

[0061] 在一个实施例中,如图5所示,客户通过互联网访问外网应用中的用户终端,用户终端可展示前端页面,客户操作用户终端的前端页面可产生日志数据。用户终端可将该日志数据异步传输至JETTY(轻量级容器)应用层。同时,用户终端与应用层之间还可正常的业务调用。应用层可通过PUBLIC(公共)网关与DUBBOX(开源微服务架构)微服务进行通信,应用层可通过域名地址与管理后台服务进行通信,并将用户终端和应用层生成的日志数据异步传输给管理后台服务。管理后台服务可将本服务生成的日志数据、接收到的用户终端和应用层的日志数据,存储至服务器的Log4j日志文件中。微服务生成的日志可记录在微服务本身的Log4j日志文件中。管理后台服务和微服务上分别部署有采集程序,采集程序可定时监控并采集管理后台服务和微服务上的所有日志文件,将采集到的日志文件,即BEE(存档)文件分别发送给KAFKA集群。FLINK可自动消费KAFKA中日志文件的日志数据,并对所有日志数据进行分流处理。FLINK可将所有日志数据全量存储至HIVE物理存储中,便于后续全量统计。FLINK可将近15天的日志数据存储于ES检索引擎中。在管理后台中可做一个查询展示页面,查询ES中的实时日志数据和查询HIVE中的历史日志数据。

[0062] 在一个实施例中,用户终端生成的前端日志数据、应用层生成的日志数据、后台管理服务生成的日志数据和微服务生成的日志数据具有不同的日志格式。对应的各业务服务器可按预设格式生成具有相同日志格式的日志文件。对应的各业务服务器可将具有相同日志格式的所有日志文件,分别发送给KAFKA集群。KAFKA可对接收到的所有日志文件中的日志数据进行采集抽样,查看日志数据对应的日志格式是否符合业务需求。当KAFKA日志采集抽样不符合业务需求时,可重新设定预设格式,生成对应的否符合业务需求的日志文件。

[0063] 上述日志数据处理方法中,通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据,由于各服务类型的业务服务器各自对应的日志数据有不同的日志格式,进而各业务服务器可按预设格式,将日志数据生成对应的日志文件,使得日志文件中的各日志数据具有相同的日志格式。分流处理集群可按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件,且分别进行存储,使得日志查询操作更便捷,问题定位更快速,进而提升日志数据处理效率。

[0064] 在一个实施例中,步骤S204,也就是各业务服务器按预设格式,将日志数据生成对应的日志文件的步骤,具体包括:各业务服务器根据预设业务需求,确定相同的预设格式;各业务服务器按相同的预设格式中的预设字段和预设字段的排序,将日志数据生成对应的日志文件。

[0065] 具体地,各业务服务器获取到的日志数据具有不同的日志格式,各业务服务器可根据预设业务需求,确定相同的预设格式。预设格式可包括预设字段和预设字段的排序。各业务服务器可按相同的预设格式中的预设字段和预设字段的排序,将日志数据生成对应的日志文件。

[0066] 在一个实施例中,预设字段中包括日志级别,Log4j中将要输出的日志数据信息定义了6种日志级别,依次为TRACE、DEBUG、INFO、WARN、ERROR和FATAL。当输出日志时,只有级

别高过配置中规定的级别的日志数据信息才能真正的输出。其中,TRACE是很低的日志级别,TRACE跟踪的是函数的调用,并且TRACE不能含变量参数,而仅能提示函数的调用关系。DEBUG是细粒度级别,主要用于开发过程中打印一些运行信息。INFO是粗粒度级别,用于突出强调应用程序的运行过程,打印一些用户感兴趣的或者重要的信息。WARN表示会出现潜在错误的情形,有些信息不是错误信息,但是也要给开发人员一些提示。ERROR指出虽然发生错误事件,但仍然不影响系统的继续运行,用于打印错误和异常信息。FATAL是高日志级别,用于指出每个严重的错误事件将会导致应用程序的退出。

[0067] 上述实施例中,通过预设字段设定统一的日志格式,使得根据业务需求管理日志数据更加方便,便于在同一个页面中统一查询日志数据。

[0068] 在一个实施例中,步骤S208,也就是分流处理集群按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件的步骤,具体包括:分流处理集群根据预设格式中的时间字段,确定日志文件中各日志数据的生成时间;分流处理集群将日志文件中各日志数据的生成时间进行降序排序,得到对应的时间排序结果;分流处理集群根据时间排序结果和预设待存储字段,从日志文件的各日志数据中,筛选出符合第一预设条件的第一分流日志数据,并根据第一分流日志数据生成第一分流日志文件;分流处理集群根据时间排序结果,从日志文件的各日志数据中,筛选出符合第二预设条件的第二分流日志数据,并根据第二分流日志数据生成第二分流日志文件。

[0069] 具体地,预设格式包括时间字段,每一条日志数据都对应有该日志数据的生成时间。分流处理集群可根据预设格式中的时间字段,确定日志文件中各日志数据的生成时间。分流处理集群可将日志文件中各日志数据的生成时间进行降序排序,得到对应的时间排序结果。分流处理集群可根据业务需求,确定日志数据对应的预设待存储字段、第一预设条件和第二预设条件。分流处理集群可根据时间排序结果和预设待存储字段,从日志文件的各日志数据中,筛选出符合第一预设条件的第一分流日志数据,并根据第一分流日志数据生成第一分流日志文件。分流处理集群可根据时间排序结果,从日志文件的各日志数据中,筛选出符合第二预设条件的第二分流日志数据,并根据第二分流日志数据生成第二分流日志文件。

[0070] 在一个实施例中,第一预设条件具体可以是从时间排序结果中对应的所有日志数据中,筛选5个字段对应的日志数据进行全量存储,比如,5个字段具体可以是timeMillis(日志时间)、thread(代码位置)、level(日志级别)、loggerName(日志名称)和message(日志内容)。第二预设条件具体可以是从时间排序结果中对应的所有日志数据中,筛选最近15天对应的日志数据进行实时存储。

[0071] 上述实施例中,根据第一预设条件和第二预设条件,对日志数据进行分流处理,使得日志数据可以根据业务需求进行分类管理,进一步提升日志数据处理效率。

[0072] 在一个实施例中,存储设备包括全量存储设备和实时存储设备,步骤S210,也就是分流处理集群将第一分流日志文件和第二分流日志文件分别存储至相应的存储设备的步骤,具体包括:分流处理集群分别获取全量存储设备的地址和实时存储设备的地址;分流处理集群根据全量存储设备的地址,将第一分流日志文件存储至全量存储设备的物理内存中;分流处理集群根据实时存储设备的地址,将第二分流日志文件存储至实时存储设备。

[0073] 其中,全量存储设备是全量存储第一分流日志文件的设备,实时存储设备是实时

存储第二分流日志文件的设备。具体地,日志数据处理系统中部署有全量存储设备和实时存储设备,分流处理集群可分别获取全量存储设备的地址和实时存储设备的地址。分流处理集群可根据全量存储设备的地址,查找到该全量存储设备,进而将第一分流日志文件存储至全量存储设备的物理内存中。分流处理集群可根据实时存储设备的地址,查找到该实时存储设备,将第二分流日志文件存储至实时存储设备。

[0074] 在一个实施例中,开发人员可在用户终端对应的功能代码中,添加用户终端请求的追踪标识,进而根据追踪标识可将用户终端生成的日志数据与各业务服务器生成的日志数据进行关联。在管理后台服务中生成一个日志数据查询页面,通过该查询页面可链路查询全量存储设备中的全量日志数据,通过该查询页面可链路查询实时存储设备中的实时日志数据。

[0075] 上述实施例中,根据全量存储设备的地址和实时存储设备的地址,分流处理集群可快速查询到全量存储设备和实时存储设备,进而可将第一分流日志文件和第二分流日志文件实时更新存储至对应的存储设备中,提升日志数据的存储效率。

[0076] 在一个具体的实施例中,日志数据处理方法包括以下步骤:

[0077] 通用服务器接收不同用户终端所发送的日志数据。

[0078] 用户终端通过预先融合有日志数据拦截器的前端页面框架,获取用户终端产生的日志数据。

[0079] 用户终端将用户终端产生的日志数据异步传输给通用服务器。

[0080] 通用服务器获取通用服务器本地所生成的日志数据。

[0081] 通过微服务器获取微服务器本地所生成的日志数据。

[0082] 各业务服务器根据预设业务需求,确定相同的预设格式。

[0083] 各业务服务器按相同的预设格式中的预设字段和预设字段的排序,将日志数据生成对应的日志文件。

[0084] 各业务服务器通过采集代码监测各自对应的日志文件。

[0085] 当各业务服务器监测到各自对应的日志文件更新时,将更新后的各业务服务器各自对应的日志文件分别发送至分流处理集群。

[0086] 分流处理集群根据预设格式中的时间字段,确定日志文件中各日志数据的生成时间。

[0087] 分流处理集群将日志文件中各日志数据的生成时间进行降序排序,得到对应的时间排序结果。

[0088] 分流处理集群根据时间排序结果和预设待存储字段,从日志文件的各日志数据中,筛选出符合第一预设条件的第一分流日志数据,并根据第一分流日志数据生成第一分流日志文件。

[0089] 分流处理集群根据时间排序结果,从日志文件的各日志数据中,筛选出符合第二预设条件的第二分流日志数据,并根据第二分流日志数据生成第二分流日志文件。

[0090] 分流处理集群分别获取全量存储设备的地址和实时存储设备的地址。

[0091] 分流处理集群根据全量存储设备的地址,将第一分流日志文件存储至全量存储设备的物理内存中。

[0092] 分流处理集群根据实时存储设备的地址,将第二分流日志文件存储至实时存储设备。

备。

[0093] 上述日志数据处理方法,通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据,由于各服务类型的业务服务器各自对应的日志数据有不同的日志格式,进而各业务服务器可按预设格式,将日志数据生成对应的日志文件,使得日志文件中的各日志数据具有相同的日志格式。分流处理集群可按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件,且分别进行存储,使得日志查询操作更便捷,问题定位更快速,进而提升日志数据处理效率。

[0094] 应该理解的是,虽然上述具体的实施例中的各个步骤按照顺序依次显示,但是这些步骤并不是必然按照顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,上述具体的实施例中的至少一部分步骤可以包括多个子步骤或者多个阶段,这些子步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些子步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

[0095] 在一个实施例中,参考图1,提供了一种日志数据处理系统,包括:多于一种的、且对应不同服务类型的业务服务器102、分流处理集群104和存储设备106,其中:

[0096] 各业务服务器102,用于获取各自对应的日志数据。

[0097] 各业务服务器102,还用于按预设格式,将日志数据生成对应的日志文件。

[0098] 各业务服务器102,还用于将各自对应的日志文件分别发送至分流处理集群104。

[0099] 分流处理集群104,用于按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件;第一分流日志文件包括全量日志数据,第二分流日志文件包括实时日志数据。

[0100] 分流处理集群104,还用于将第一分流日志文件和第二分流日志文件分别存储至相应的存储设备106。

[0101] 在一个实施例中,各业务服务器102,还用于获取用户终端和通用服务器本地所生成的日志数据;获取微服务器本地所生成的日志数据。

[0102] 在一个实施例中,各业务服务器102,还用于接收不同用户终端所发送的日志数据;获取通用服务器本地所生成的日志数据;用户终端发送日志数据的步骤包括:用户终端通过预先融合有日志数据拦截器的前端页面框架,获取用户终端产生的日志数据;用户终端将用户终端产生的日志数据异步传输给通用服务器。

[0103] 在一个实施例中,各业务服务器102,还用于根据预设业务需求,确定相同的预设格式;按相同的预设格式中的预设字段和预设字段的排序,将日志数据生成对应的日志文件。

[0104] 在一个实施例中,各业务服务器102,还用于通过采集代码监测各自对应的日志文件;当各业务服务器102监测到各自对应的日志文件更新时,将更新后的各业务服务器各自对应的日志文件分别发送至分流处理集群。

[0105] 在一个实施例中,分流处理集群104,还用于根据预设格式中的时间字段,确定日志文件中各日志数据的生成时间;将日志文件中各日志数据的生成时间进行降序排序,得到对应的时间排序结果;根据时间排序结果和预设待存储字段,从日志文件的各日志数据

中,筛选出符合第一预设条件的第一分流日志数据,并根据第一分流日志数据生成第一分流日志文件;根据时间排序结果,从日志文件的各日志数据中,筛选出符合第二预设条件的第二分流日志数据,并根据第二分流日志数据生成第二分流日志文件。

[0106] 在一个实施例中,分流处理集群104,还用于分别获取全量存储设备的地址和实时存储设备的地址;根据全量存储设备的地址,将第一分流日志文件存储至全量存储设备的物理内存中;根据实时存储设备的地址,将第二分流日志文件存储至实时存储设备。

[0107] 上述日志数据处理系统,通过多于一种的、且对应不同服务类型的业务服务器,获取各自对应的日志数据,由于各服务类型的业务服务器各自对应的日志数据有不同的日志格式,进而各业务服务器可按预设格式,将日志数据生成对应的日志文件,使得日志文件中的各日志数据具有相同的日志格式。分流处理集群可按日志文件中各日志数据的生成时间,对日志数据进行分流处理,得到第一分流日志文件和第二分流日志文件,且分别进行存储,使得日志查询操作更便捷,问题定位更快速,进而提升日志数据处理效率。

[0108] 在一个实施例中,提供了一种计算机设备,该计算机设备可以是上述图1中的各业务服务器102,其内部结构图可以如图6所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储日志数据处理数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种日志数据处理方法。

[0109] 本领域技术人员可以理解,图6中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0110] 在一个实施例中,提供了一种计算机设备,包括存储器和处理器,存储器存储有计算机程序,计算机程序被处理器执行时,使得处理器执行上述日志数据处理方法的步骤。此处日志数据处理方法的步骤可以是上述各个实施例的日志数据处理方法中的步骤。

[0111] 在一个实施例中,提供了一种计算机可读存储介质,存储有计算机程序,计算机程序被处理器执行时,使得处理器执行上述日志数据处理方法的步骤。此处日志数据处理方法的步骤可以是上述各个实施例的日志数据处理方法中的步骤。

[0112] 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,所述的计算机程序可存储于一非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限,RAM以多种形式可得,诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink)DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0113] 以上实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0114] 以上所述实施例仅表达了本申请的几种实施方式,其描述较为具体和详细,但不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本申请构思的前提下,还可以做出若干变形和改进,这些都属于本申请的保护范围。因此,本申请专利的保护范围应以所附权利要求为准。

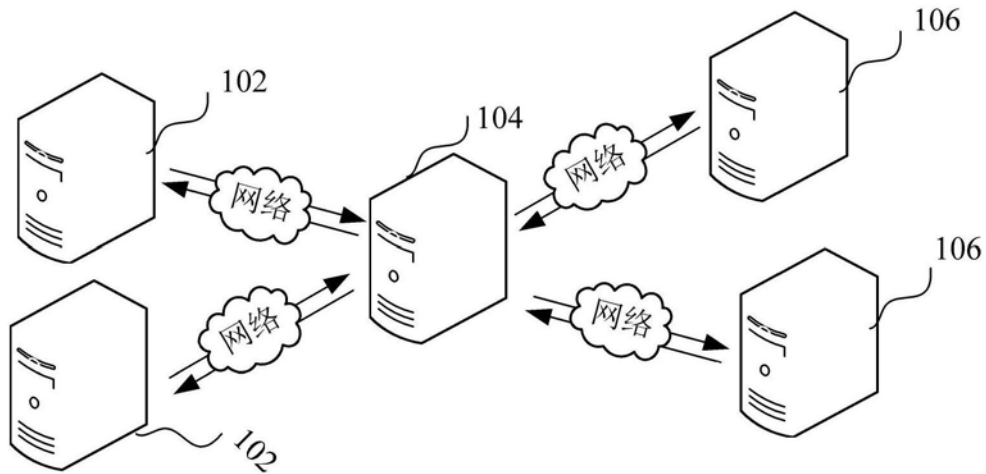


图1

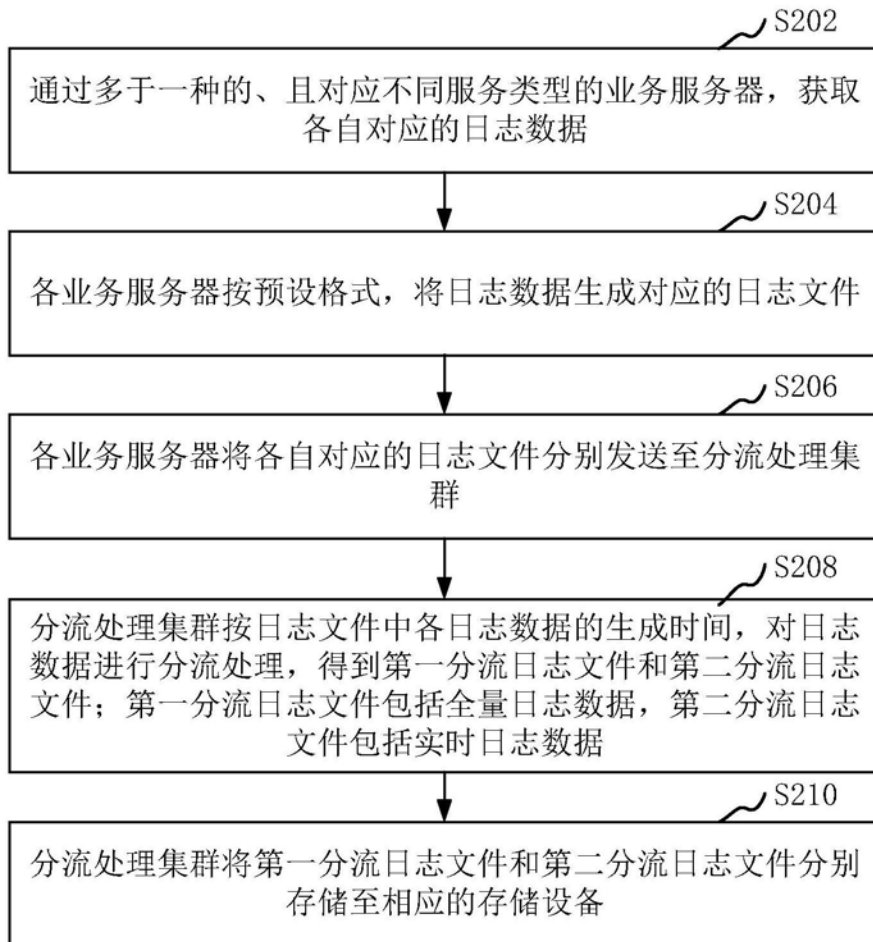


图2

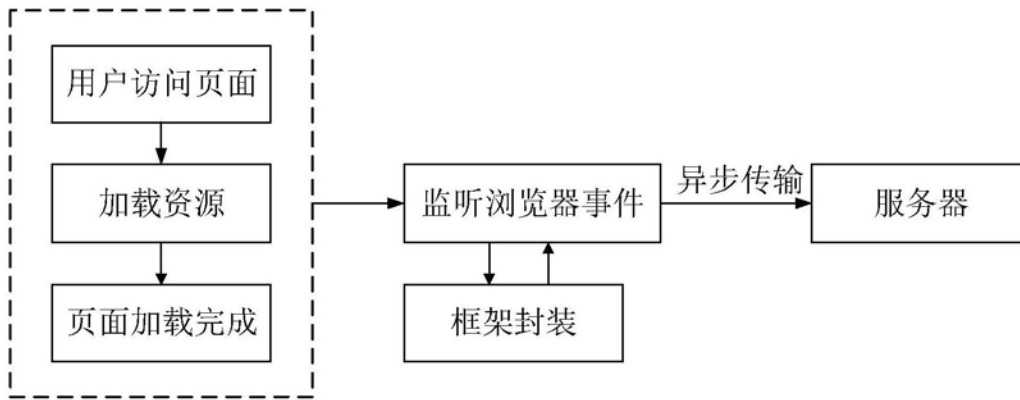


图3

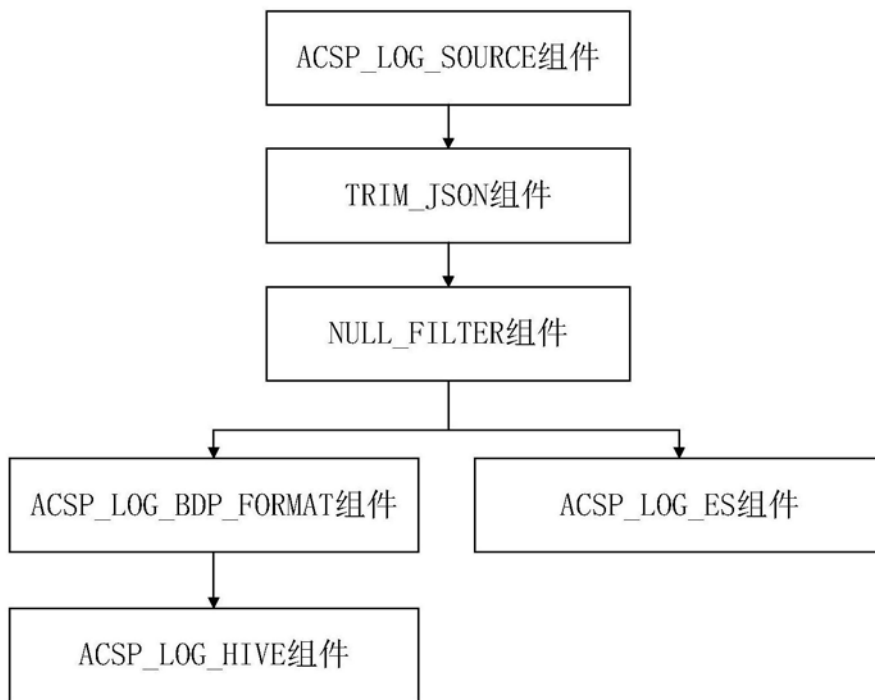


图4

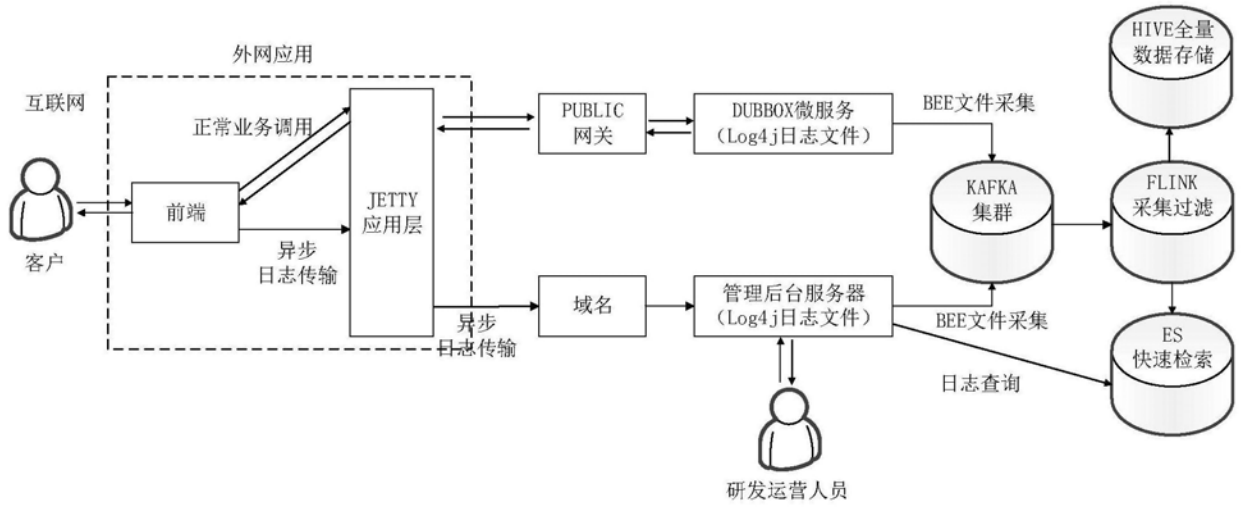


图5

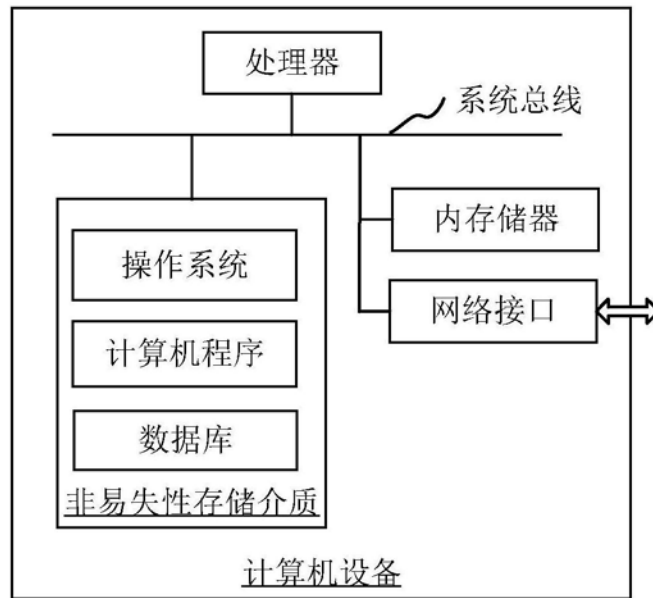


图6