



(12) 发明专利申请

(10) 申请公布号 CN 112612675 A

(43) 申请公布日 2021. 04. 06

(21) 申请号 202011566084.7

(22) 申请日 2020.12.25

(71) 申请人 山东经伟晟睿数据技术有限公司
地址 250101 山东省济南市高新区新泺大街2008号银荷大厦C座1002

(72) 发明人 刘宏亮 李铭 张丽丽 张辉
贾立平

(74) 专利代理机构 济南圣达知识产权代理有限公司 37221

代理人 张庆骞

(51) Int. Cl.

G06F 11/30 (2006.01)

G06F 11/32 (2006.01)

G06F 11/34 (2006.01)

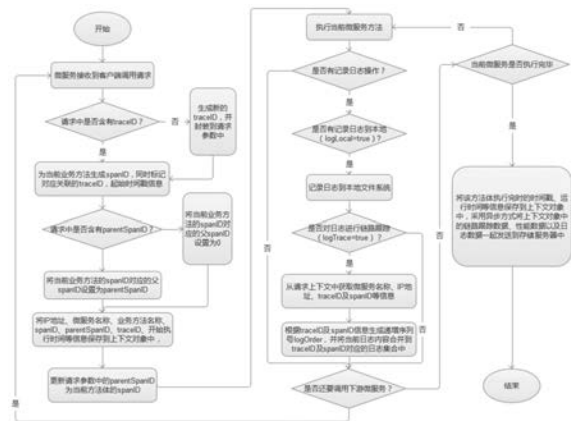
权利要求书2页 说明书9页 附图2页

(54) 发明名称

微服务架构下的分布式大数据日志链路跟踪方法及系统

(57) 摘要

本发明提供了微服务架构下的分布式大数据日志链路跟踪方法及系统。其中该方法包括微服务模块接收到调用请求后,判断当前微服务模块是否为第一次调用,得到traceID和spanID;判断当前业务方法是否为第一次调用的业务方法,更新调用链中的spanID关系;缓存微服务模块所在服务器IP地址、微服务模块名称、业务方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息;根据链路日志跟踪标记来执行日志跟踪记录,将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定,对日志信息按照调用链路信息进行排序处理;绑定微服务模块运行的性能数据与调用链路信息,将排序后的日志记录压缩后发送到日志存储服务器,然后删除调用链路对应的缓存数据,释放内存空间。



1. 一种微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,包括:

当微服务模块接收到调用请求后,判断当前微服务模块是否为第一次调用,若是,则生成链路唯一标识traceID,再生成业务方法唯一标识spanID;否则,直接生成spanID,同时标记对应关联的traceID及起始时间戳信息;

根据请求中是否存在父业务方法唯一标识parentSpanID,来判断当前业务方法是否为第一次调用的业务方法,并更新调用链中的spanID关系;

执行当前业务方法时缓存微服务模块所在服务器IP地址、微服务模块名称、业务方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息;

根据链路日志跟踪标记来执行日志跟踪记录,将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定,形成日志记录并将其按照调用链路信息进行排序处理;

业务方法执行完毕时,绑定微服务模块运行的性能数据与调用链路信息,将排序后的日志记录压缩后发送到日志存储服务器,然后删除调用链路对应的缓存数据,释放内存空间。

2. 如权利要求1所述的微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,根据请求中是否存在父业务方法唯一标识parentSpanID,来判断当前业务方法是否为第一次调用的业务方法。

3. 如权利要求1所述的微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,若当前业务方法为第一次调用的业务方法,则将当前业务方法对应的父业务方法parentSpanID绑定为null,否则将当前业务方法对应的父业务方法绑定为请求参数中的parentSpanID。

4. 如权利要求1所述的微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,在执行当前微服务业务方法的过程中,如果业务方法中有记录日志的操作,则先将日志内容记录到微服务模块所在服务器的本地文件系统中,然后检查是否有进行链路日志跟踪的标记标量。

5. 如权利要求4所述的微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,如果没有记录日志的操作,则直接检查是否有进行链路日志跟踪的标记标量。

6. 如权利要求4所述的微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,如果需要进行日志跟踪记录,则从缓存中获取traceID、spanID信息,然后根据traceID和spanID为当前日志记录生成自增序列logOrder,并将logOrder以及当前日志信息追加到缓存中对应的日志有序集合列表中。

7. 如权利要求1所述的微服务架构下的分布式大数据日志链路跟踪方法,其特征在于,所述分布式大数据日志链路跟踪方法还包括:当前微服务是否继续调用下游微服务,若是,则将调用请求中的parentSpanID设置为当前的spanID后,发起对下游微服务的调用;否则,检查当前微服务是否执行完毕。

8. 一种微服务架构下的分布式大数据日志链路跟踪系统,其特征在于,包括:

服务模块调用判断模块,其用于当微服务模块接收到调用请求后,判断当前微服务模块是否为第一次调用,若是,则生成链路唯一标识traceID,再生成业务方法唯一标识spanID;否则,直接生成spanID,同时标记对应关联的traceID及起始时间戳信息;

业务方法调用判断模块,其用于根据请求中是否存在父业务方法唯一标识

parentSpanID,来判断当前业务方法是否为第一次调用的业务方法,并更新调用链中的spanID关系;

缓存模块,其用于执行当前业务方法时缓存微服务模块所在服务器IP地址、微服务模块名称、业务方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息;

信息绑定模块,其用于根据链路日志跟踪标记来执行日志跟踪记录,将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定,形成日志记录并将其按照调用链路信息进行排序处理;

日志信息发送模块,其用于业务方法执行完毕时,绑定微服务模块运行的性能数据与调用链路信息,将排序后的日志记录压缩后发送到日志存储服务器,然后删除调用链路对应的缓存数据,释放内存空间。

9. 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,该程序被处理器执行时实现如权利要求1-7中任一项所述的微服务架构下的分布式大数据日志链路跟踪方法中的步骤。

10. 一种计算机设备,包括存储器、处理器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,所述处理器执行所述程序时实现如权利要求1-7中任一项所述的微服务架构下的分布式大数据日志链路跟踪方法中的步骤。

微服务架构下的分布式大数据日志链路跟踪方法及系统

技术领域

[0001] 本发明属于计算机微服务分布式系统技术领域,尤其涉及一种微服务架构下的分布式大数据日志链路跟踪方法及系统。

背景技术

[0002] 本部分的陈述仅仅是提供了与本发明相关的背景技术信息,不必然构成在先技术。

[0003] 随着系统业务需求越来越复杂,大型的业务系统不得不按照业务功能将系统进行各种模块拆分,然后通过模块的动态组合,最终提供丰富的业务功能,同时提高系统灵活性和扩展性,特别是随着计算机软件技术中微服务架构和容器技术的兴起,看似简单的一个应用,后台可能有几十个甚至几百个微服务在支撑,一个简单的前端调用请求可能需要透传多次的微服务调用最后才能完成,被调用的微服务可能还要继续调用其它微服务。当系统拆分的微服务越多时,这种分布式服务架构下服务之间的调用关系也就自然而然的更加复杂化,见附图1。

[0004] 由于微服务模块数量众多,并且分布式部署,各个微服务的运行日志分散到不同的存储位置,因此发明人发现,在微服务架构下的计算机系统面临以下几个技术问题:1) 无法实现微服务之间复杂调用关系的链路跟踪,使得开发人员无法对微服务模块的边界进行准确地规划。2) 运维人员无法从数量众多的微服务中快速识别定位到系统性能瓶颈模块。3) 现有技术可以对微服务模块的运行日志文件数据进行查询,但是存在运行日志的查询过程较为繁琐,日志的查询效率和准确率均不高的问题。

发明内容

[0005] 为了解决上述背景技术中存在的至少一项技术问题,本发明提供一种微服务架构下的分布式大数据日志链路跟踪方法及系统,其能够实现对微服务架构系统的服务链路展示和跟踪、服务性能的自动检测、详细运行日志的跟踪排查等全系列操作,可显著提高系统的开发和运维效率,降低企业硬件成本,为微服务架构下分布式系统的稳定运行提供了技术保障。

[0006] 为了实现上述目的,本发明采用如下技术方案:

[0007] 本发明的第一个方面提供一种微服务架构下的分布式大数据日志链路跟踪方法。

[0008] 一种微服务架构下的分布式大数据日志链路跟踪方法,包括:

[0009] 当微服务模块接收到调用请求后,判断当前微服务模块是否为第一次调用,若是,则生成链路唯一标识traceID,再生成业务方法唯一标识spanID;否则,直接生成spanID,同时标记对应关联的traceID及起始时间戳信息;

[0010] 根据请求中是否存在父业务方法唯一标识parentSpanID,来判断当前业务方法是否为第一次调用的业务方法,并更新调用链中的spanID关系;

[0011] 执行当前业务方法时缓存微服务模块所在服务器IP地址、微服务模块名称、业务

方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息；

[0012] 根据链路日志跟踪标记来执行日志跟踪记录，将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定，形成日志记录并将其按照调用链路信息进行排序处理；

[0013] 业务方法执行完毕时，绑定微服务模块运行的性能数据与调用链路信息，将排序后的日志记录压缩后发送到日志存储服务器，然后删除调用链路对应的缓存数据，释放内存空间。

[0014] 作为一种实施方式，根据请求中是否存在父业务方法唯一标识parentSpanID，来判断当前业务方法是否为第一次调用的业务方法。

[0015] 作为一种实施方式，若当前业务方法为第一次调用的业务方法，则将当前业务方法对应的父业务方法parentSpanID绑定为null，否则将当前业务方法对应的父业务方法绑定为请求参数中的parentSpanID。

[0016] 作为一种实施方式，在执行当前微服务业务方法的过程中，如果业务方法中有记录日志的操作，则先将日志内容记录到微服务模块所在服务器的本地文件系统中，然后检查是否有进行链路日志跟踪的标记标量。

[0017] 作为一种实施方式，如果没有记录日志的操作，则直接检查是否有进行链路日志跟踪的标记标量。

[0018] 作为一种实施方式，如果需要进行日志跟踪记录，则从缓存中获取traceID、spanID信息，然后根据traceID和spanID为当前日志记录生成自增序列logOrder，并将logOrder以及当前日志信息追加到缓存中对应的日志有序集合列表中。

[0019] 作为一种实施方式，所述分布式大数据日志链路跟踪方法还包括：当前微服务是否继续调用下游微服务，若是，则将调用请求中的parentSpanID设置为当前的spanID后，发起对下游微服务的调用；否则，检查当前微服务是否执行完毕。

[0020] 本发明的第二个方面提供一种微服务架构下的分布式大数据日志链路跟踪系统。

[0021] 一种微服务架构下的分布式大数据日志链路跟踪系统，包括：

[0022] 服务模块调用判断模块，其用于当微服务模块接收到调用请求后，判断当前微服务模块是否为第一次调用，若是，则生成链路唯一标识traceID，再生成业务方法唯一标识spanID；否则，直接生成spanID，同时标记对应关联的traceID及起始时间戳信息；

[0023] 业务方法调用判断模块，其用于根据请求中是否存在父业务方法唯一标识parentSpanID，来判断当前业务方法是否为第一次调用的业务方法，并更新调用链中的spanID关系；

[0024] 缓存模块，其用于执行当前业务方法时缓存微服务模块所在服务器IP地址、微服务模块名称、业务方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息；

[0025] 信息绑定模块，其用于根据链路日志跟踪标记来执行日志跟踪记录，将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定，形成日志记录并将其按照调用链路信息进行排序处理；

[0026] 日志信息发送模块，其用于业务方法执行完毕时，绑定微服务模块运行的性能数据与调用链路信息，将排序后的日志记录压缩后发送到日志存储服务器，然后删除调用链

路对应的缓存数据,释放内存空间。

[0027] 本发明的第三个方面提供一种计算机可读存储介质。

[0028] 一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如上述所述的微服务架构下的分布式大数据日志链路跟踪方法中的步骤。

[0029] 本发明的第四个方面提供一种计算机设备。

[0030] 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现如上述所述的微服务架构下的分布式大数据日志链路跟踪方法中的步骤。

[0031] 与现有技术相比,本发明的有益效果是:

[0032] 本发明创新性的将调用链路信息与微服务运行日志数据进行绑定,并将运行日志数据按照微服务模块进行压缩存储,不但可显著提高系统运行日志数据的存储和检索查询效率,节省服务器硬件成本投入,而且还可以保证日志序列的准确性,可使开发人员和运维人员方便快捷地,通过图形化方式点击微服务链路跟踪图上的问题节点,就可以直接查看对应节点的详细的运行日志信息,大大提高了相关人员问题排查的便捷性和准确性。

[0033] 本发明实现了微服务之间复杂调用关系的链路跟踪,便于开发人员对微服务模块的边界进行更加准确地规划设计,实现了微服务模块的性能检测,便于运维人员从数量众多的微服务中快速识别定位到系统性能瓶颈模块,还实现了微服务模块运行日志与微服务跟踪链路数据以及性能数据的有效结合,让开发人员和运维人员能够通过图形化方式、在统一的平台就可以十分便捷、快速、准确高效地检索出问题模块的系统运行日志,极大提高系统整体开发和运维效率,节省开发成本。

附图说明

[0034] 构成本发明的一部分的说明书附图用来提供对本发明的进一步理解,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。

[0035] 图1是本发明所述发微服务之间复杂调用关系的说明示意图;

[0036] 图2是本发明实施例提供的服务链路跟踪的流程示意图;

[0037] 图3是本发明实施例提供的微服务架构下的分布式大数据日志链路跟踪方法流程图。

具体实施方式

[0038] 下面结合附图与实施例对本发明作进一步说明。

[0039] 应该指出,以下详细说明都是例示性的,旨在对本发明提供进一步的说明。除非另有指明,本文使用的所有技术和科学术语具有与本发明所属技术领域的普通技术人员通常理解的含义。

[0040] 需要注意的是,这里所使用的术语仅是为了描述具体实施方式,而非意图限制根据本发明的示例性实施方式。如在这里所使用的,除非上下文另外明确指出,否则单数形式也意图包括复数形式,此外,还应当理解的是,当在本说明书中使用术语“包含”和/或“包括”时,其指明存在特征、步骤、操作、器件、组件和/或它们的组合。

[0041] 实施例一

[0042] 微服务链路跟踪技术,一般是通过请求源生成本次调用请求的traceID标识并发起请求,收到请求的目标微服务解析traceID,并生成调用方法对应的spanID以及附加信息(如被调用方法名称、上一级SpanID、节点IP地址等),通过这些链路ID信息就可以构建出整个微服务调用依赖关系,如Spring Cloud Sleuth框架就是采用该技术方案实现,如图2所示。

[0043] 1) 该技术方案依赖于抽象的traceID来生成服务间的跟踪链路信息,当系统出现性能或功能故障时,运维或开发人员往往先通过查找日志文件,找出异常日志中的traceID,然后再通过traceID去检索完整的运行日志,效率低、准确度差,尤其当服务模块众多、系统运行日志数量巨大时此技术方案的弊端更加明显;

[0044] 2) 该技术方案无法方便快捷的统计各个模块的运行时间等性能参数信息,无法进行性能检测,不便于系统出现性能故障时的问题排查;

[0045] 3) 该技术方案由于设计上的缺陷无法提供方便、准确、快速的微服务运行日志查询方案,不利于运维人员和开发人员快速定位系统问题根源。

[0046] 微服务链路跟踪技术+微服务计时跟踪技术是在微服务链路跟踪技术的基础上引入服务计时跟踪技术,服务计时跟踪技术是通过记录每一个服务节点的开始执行时间、持续时间、结束执行时间等信息,可统计出整个链路中每个服务节点的平均总执行时间,方便对服务节点进行性能统计。

[0047] 通过微服务链路跟踪技术+微服务计时跟踪技术两种技术的结合,该技术方案可实现微服务链路跟踪以及性能监控,可跟踪出微服务的调用链路,并且可根据每个微服务节点的运行时间统计数据快速识别造成系统性能瓶颈的微服务节点,但是该方案无法帮助开发人员和运维人员进一步准确定位性能或功能问题根源:服务运行时的日志输出到底是怎样的,到底是什么原因导致性能问题,即该技术方案只能解决技术背景中的问题1和2,无法解决问题3。

[0048] 微服务链路跟踪技术+微服务计时跟踪技术+日志数据收集和可视化技术,在微服务链路跟踪技术+微服务计时跟踪技术的基础上引入日志数据收集和可视化技术,将微服务调用的数据链路信息、以及微服务性能跟踪数据发送到日志文件中,然后通过Logstash将日志数据逐条发送到Elasticsearch(数据存储服务器),然后通过Kibana图形化界面进行查询展示。微服务链路跟踪技术+微服务计时跟踪技术+日志数据收集和可视化技术可以对微服务模块的运行日志文件数据进行查询,但是存在以下缺点:

[0049] 运行日志的查询过程较为繁琐:先通过zipkin和sleuth查询到对应的traceID等信息,然后再到Kibana中查询展示对应的日志内容,开发人员和运维人员需要在不同的管理模块中来回切换,当需要查询的微服务调用关系复杂时,时间成本会非常高,另外,该技术方案也增加了运维部门和开发部门之间的沟通成本,同时由于该方案涉及的工具软件较多,也对相关人员提出了更高的技术要求。

[0050] 日志的查询效率不高:在Kibana中用图形化方式查询和展示日志结果,是从Elasticsearch的海量日志数据中进行的直接查询,由于该方案中系统运行的日志未做任何压缩优化处理直接记录存储到Elasticsearch中,查询效率会随着日志数量上升而显著下降,也对硬件成本提出了更高的要求。

[0051] 日志查询的准确率不高:该方案通过zipkin和sleuth查询到traceID,然后根据

traceID去Elasticsearch中查询本次链路的有关日志信息时,一般是基于时间戳进行排序来展示系统运行日志情况,但是由于微服务是分布式部署,实际生产环境中各个微服务所部署的服务器的时钟时间频率存在一定的偏差,这往往导致查询展示出来的日志序列与实际的运行日志序列不一致,而这非常不利于技术人员对问题根源的快速、准确定位,甚至有时候会产生误导。

[0052] 如图3所示,本实施例提供了一种微服务架构下的分布式大数据日志链路跟踪方法,具体包括如下步骤:

[0053] S101:当微服务模块接收到调用请求后,判断当前微服务模块是否为第一次调用,若是,则生成链路唯一标识traceID,再生成业务方法唯一标识spanID;否则,直接生成spanID,同时标记对应关联的traceID及起始时间戳信息。

[0054] 在具体实施中,微服务模块接收到调用请求,如果请求中没有traceID则说明当前微服务是第一个被调用的微服务,生成代表当前调用链的唯一标识traceID;如果请求中已经有traceID则说明当前微服务模块是后续请求的微服务,此时无需进一步处理直接当前调用的业务方法生成spanID。

[0055] 在为当前调用的业务方法生成spanID,同时标记该spanID对应的traceID,以及当前业务方法的开始执行时间戳、IP地址、当前登录用户等信息。

[0056] S102:根据请求中是否存在父业务方法唯一标识parentSpanID,来判断当前业务方法是否为第一次调用的业务方法,并更新调用链中的spanID关系。

[0057] 在具体实施中,根据请求中是否存在父业务方法唯一标识parentSpanID,来判断当前业务方法是否为第一次调用的业务方法。若当前业务方法为第一次调用的业务方法,则将当前业务方法对应的父业务方法parentSpanID绑定为null,否则将当前业务方法对应的父业务方法绑定为请求参数中的parentSpanID。

[0058] S103:执行当前业务方法时缓存微服务模块所在服务器IP地址、微服务模块名称、业务方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息。

[0059] 在具体实施中,将当前微服务所在服务器IP地址、微服务名称、业务方法名称、链路唯一标识traceID、业务方法唯一标识spanID、父业务方法唯一标识parentSpanID、开始执行时间startTime、当前用户标识userID等信息保存到Redis上下文缓存对象中。

[0060] 此处需要说明的是,实施举例中提及的Redis缓存还可以用mysql等数据库产品、ActiveMQ等消息中间件来代替,甚至可以直接用机器缓存来实现,都可以用来完成本发明提出的技术方案,都属于本发明的保护范围。

[0061] S104:根据链路日志跟踪标记来执行日志跟踪记录,将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定,形成日志记录并将其按照调用链路信息进行排序处理。

[0062] 在执行当前微服务业务方法的过程中,如果业务方法中有记录日志的操作,则先将日志内容记录到微服务模块所在服务器的本地文件系统中,然后检查是否有进行链路日志跟踪的标记标量logTrace=true。如果没有记录日志的操作,则直接检查是否有进行链路日志跟踪的标记标量。

[0063] 如果需要进行日志跟踪记录,则从缓存(比如:Redis上下文缓存对象)中获取traceID、spanID信息,然后根据traceID和spanID为当前日志记录生成自增序列logOrder,

并将logOrder以及当前日志信息(比如:当前日志内容、日志级别、日志时间戳等)追加到缓存中对应的日志有序集合列表中。

[0064] 如果不进行日志链路跟踪(logTrace=null或true),则判断当前微服务是否继续调用下游微服务。

[0065] 当前微服务是否继续调用下游微服务,若是,则将调用请求中的parentSpanID设置为当前的spanID后,发起对下游微服务的调用;否则,检查当前微服务是否执行完毕。

[0066] 检查当前微服务是否执行完毕,如果没有执行完毕,则继续执行当前微服务业务方法,如果业务方法中有记录日志的操作,则先将日志内容记录到微服务所在服务器的本地文件系统中,然后检查是否有进行链路日志跟踪的标记标量logTrace=true;如果没有记录日志的操作,则直接检查是否有进行链路日志跟踪的标记标量logTrace=true。

[0067] S105:业务方法执行完毕时,绑定微服务模块运行的性能数据与调用链路信息,将排序后的日志记录压缩后发送到日志存储服务器,然后删除调用链路对应的缓存数据,释放内存空间。

[0068] 业务方法执行完毕,将执行完时的时间戳信息、运行时间段信息保存到上下文缓存对象中,采用异步方式将上下文对象中的链路跟踪数据、性能数据以及当前微服务业务方法中所有有序日志数据一起打包发送到Elasticsearch存储服务器中,并将Redis上下文缓存对象中删除对应的数据,释放内存。

[0069] 此处需要说明的是,本实施举例中提及的数据存储服务器Elasticsearch也可以用mysql、neo4j等关系型数据库或nosql数据库产品来代替,都可以用来完成本发明提出的技术方案,都属于发明专利的保护范围。

[0070] 在具体实施中,将所有日志信息按照调用的相对先后顺序先进行排序合并,最后将当前业务方法的所有日志与链路信息绑定,并且这个过程随着业务方法中日志记录的过程而动态增长,直至业务方法执行完毕。

[0071] 在一些实施例中,除了绑定日志内容外,还将当前业务方法的日志最高级别与链路信息进行了绑定,便于图形化展示模块根据日志级别进行自动标记;

[0072] 在其他实施例中,除了绑定日志内容外,还将当前用户信息与链路信息进行了绑定,便于图形化展示模块根据用户信息进行数据展示;

[0073] 在一些实施例中,还可以对微服务的性能数据进行标记和监控,当微服务性能数据超出设定阈值时,该图形化展示模块会在链路图中自动高亮标记相应的微服务模块;

[0074] 在其他实施例中,还可以对微服务的运行日志健康状态进行标记和监控,当微服务运行日志超过设置的监控日志级别时,该图形化展示模块会在链路图中自动高亮标记相应的微服务模块;

[0075] 在一些实施例中,还可以按需逐级展示微服务调用链路中每个微服务模块的详细运行日志信息,包括但不限于日志级别、操作人、日志时间、日志所在服务器IP、日志描述等信息;

[0076] 需要说明的是,按需逐级展示是指可以很方便的在微服务链路图中直接点击任意的微服务模块名字来查看对应的运行日志数据,不需要相关人员切换到其它工具软件或管理页面中;

[0077] 运行日志数据的排序不是按照日志所在服务器时间戳进行排序,而是日志在存储

时已经按照实际运行的相对顺序排序处理,该图形展示模块中展示的日志数据无需再进行排序处理。

[0078] 本实施例通过spanID、parentSpanID等信息建立起本次微服务调用中的链路依赖信息;将微服务方法的运行时间、所有运行日志、以及日志级别信息与链路信息进行了绑定;本实施例还将当前业务方法的所有有序日志数据与链路信息数据打包压缩后异步方式上传到存储服务器,这样可大大减少对存储服务器的性能压力,大大降低了存储服务器中存储数据的记录数量,显著提高日志检索时的查询效率,并且存储时已经对日志内容进行了排序处理,这样查询出来的业务方法日志内容就是整体且有序的,不需要再根据时间戳排序,所以可以保证查询检索后日志序列的准确性。

[0079] 实施例二

[0080] 本实施例提供了一种微服务架构下的分布式大数据日志链路跟踪系统,其包括:

[0081] 服务模块调用判断模块,其用于当微服务模块接收到调用请求后,判断当前微服务模块是否为第一次调用,若是,则生成链路唯一标识traceID,再生成业务方法唯一标识spanID;否则,直接生成spanID,同时标记对应关联的traceID及起始时间戳信息;

[0082] 业务方法调用判断模块,其用于根据请求中是否存在父业务方法唯一标识parentSpanID,来判断当前业务方法是否为第一次调用的业务方法,并更新调用链中的spanID关系;

[0083] 缓存模块,其用于执行当前业务方法时缓存微服务模块所在服务器IP地址、微服务模块名称、业务方法名称、traceID、spanID、parentSpanID及业务方法开始执行时间信息;

[0084] 信息绑定模块,其用于根据链路日志跟踪标记来执行日志跟踪记录,将日志信息与调用链路信息和微服务模块运行性能信息分别进行绑定,形成日志记录并将其按照调用链路信息进行排序处理;

[0085] 日志信息发送模块,其用于业务方法执行完毕时,绑定微服务模块运行的性能数据与调用链路信息,将排序后的日志记录压缩后发送到日志存储服务器,然后删除调用链路对应的缓存数据,释放内存空间。

[0086] 此处需要说明的是,本实施例的微服务架构下的分布式大数据日志链路跟踪系统中的各个模块,与实施例一中的各个步骤一一对应,其具体实施过程相同,此处不再累述。

[0087] 在实际开发设计中,微服务架构下的分布式大数据日志链路跟踪系统可采用分布式链路跟踪及性能数据收集模块、链路日志信息收集模块、图形化展示模块及数据存储模块来实现。其中,所述分布式链路跟踪及性能数据收集模块,用来生成和收集微服务之间复杂调用链路信息以及运行的性能数据信息;所述链路日志信息收集模块,用来将服务的运行日志同调用链路信进行绑定,并将绑定信息发送到所述的数据存储器模块;所述的图形化展示模块,用可视化方式展示微服务之间的调用链路信息、服务性能数据以及对应的运行日志信息,并且可以对性能数据超出设定阈值或有异常日志的微服务自动进行区别标记,便于运维和开发人员便捷、快速地定位出问题的微服务,在出问题的微服务节点还可以进一步查看该微服务本次调用中的所有运行日志信息,帮助相关人员快速定位问题根源;数据存储模块用来存储前述的微服务调用链路数据、服务性能数据、运行日志数据等。

[0088] 本实施例可通过图形化方式展示微服务之间的调用链路跟踪图信息;可通过图形

化方式对系统中运行性能时间明显低于设置阈值的服务模块自动进行高亮标记,实现对整个系统中所有微服务模块的自动性能检测,便于相关人员快速主动发现系统性能问题;可通过图形化方式对系统运行日志中出现告警或异常级别的服务模块自动进行高亮标记,实现对整个系统中所有微服务模块的自动健康检查,便于相关人员快速主动发现运行问题。

[0089] 本实施例创新性的将调用链路信息与微服务运行日志数据进行绑定,并将运行日志数据按照微服务模块进行压缩存储,不但可显著提高系统运行日志数据的存储和检索查询效率,节省服务器硬件成本投入,而且还可以保证日志序列的准确性,可使开发人员和运维人员方便快捷地,通过图形化方式点击微服务链路跟踪图上的问题节点,就可以直接查看对应节点的详细的运行日志信息,大大提高了相关人员问题排查的便捷性和准确性。

[0090] 本发明提出的技术方案可让开发人员或运维人员通过统一的图形化入口,实现对微服务架构系统的服务链路展示和跟踪、服务性能的自动检测、详细运行日志的跟踪排查等全系列操作,可显著提高系统的开发和运维效率,降低企业硬件成本,为微服务架构下分布式系统的稳定运行提供了技术保障,具有非常大的价值。

[0091] 实施例三

[0092] 本实施例提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如上述所述的微服务架构下的分布式大数据日志链路跟踪方法中的步骤。

[0093] 实施例四

[0094] 本实施例提供了一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现如上述所述的微服务架构下的分布式大数据日志链路跟踪方法中的步骤。

[0095] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用硬件实施例、软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器和光学存储器等)上实施的计算机程序产品的形式。

[0096] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0097] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0098] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0099] 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,所述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 AccessMemory,RAM)等。

[0100] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

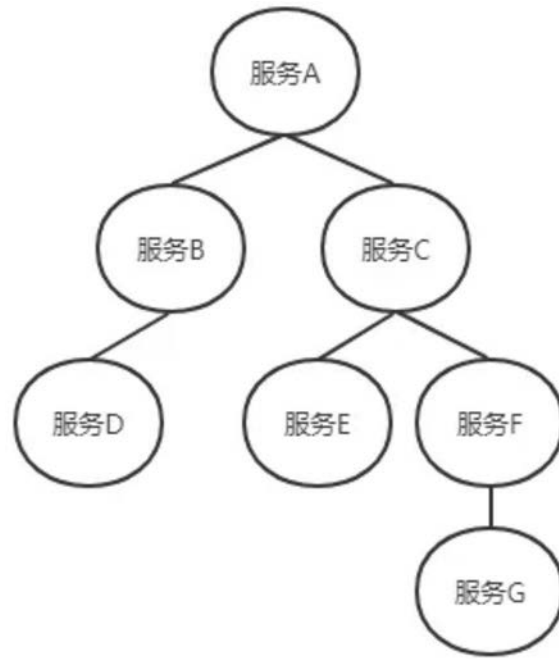


图1

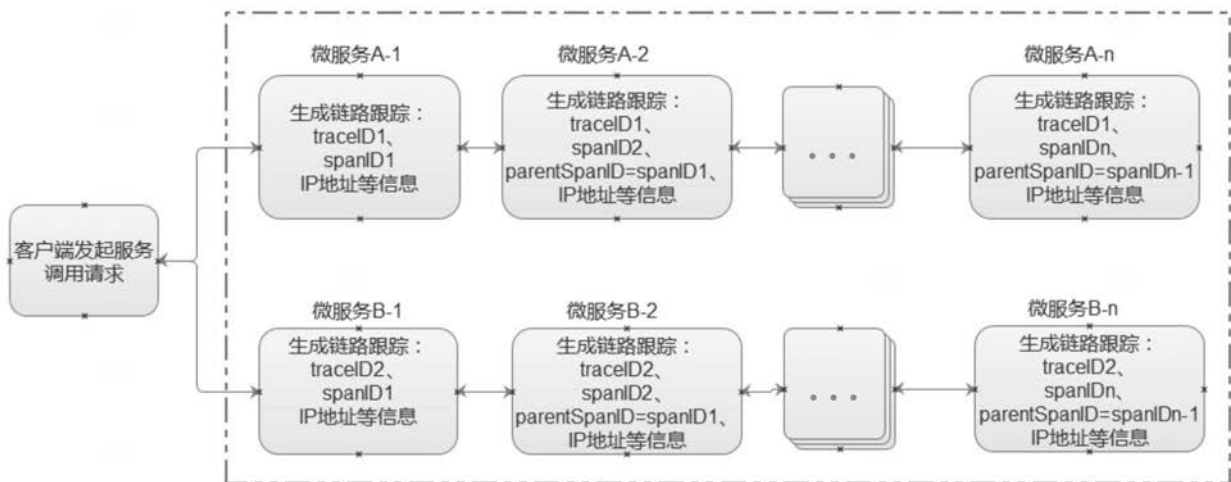


图2

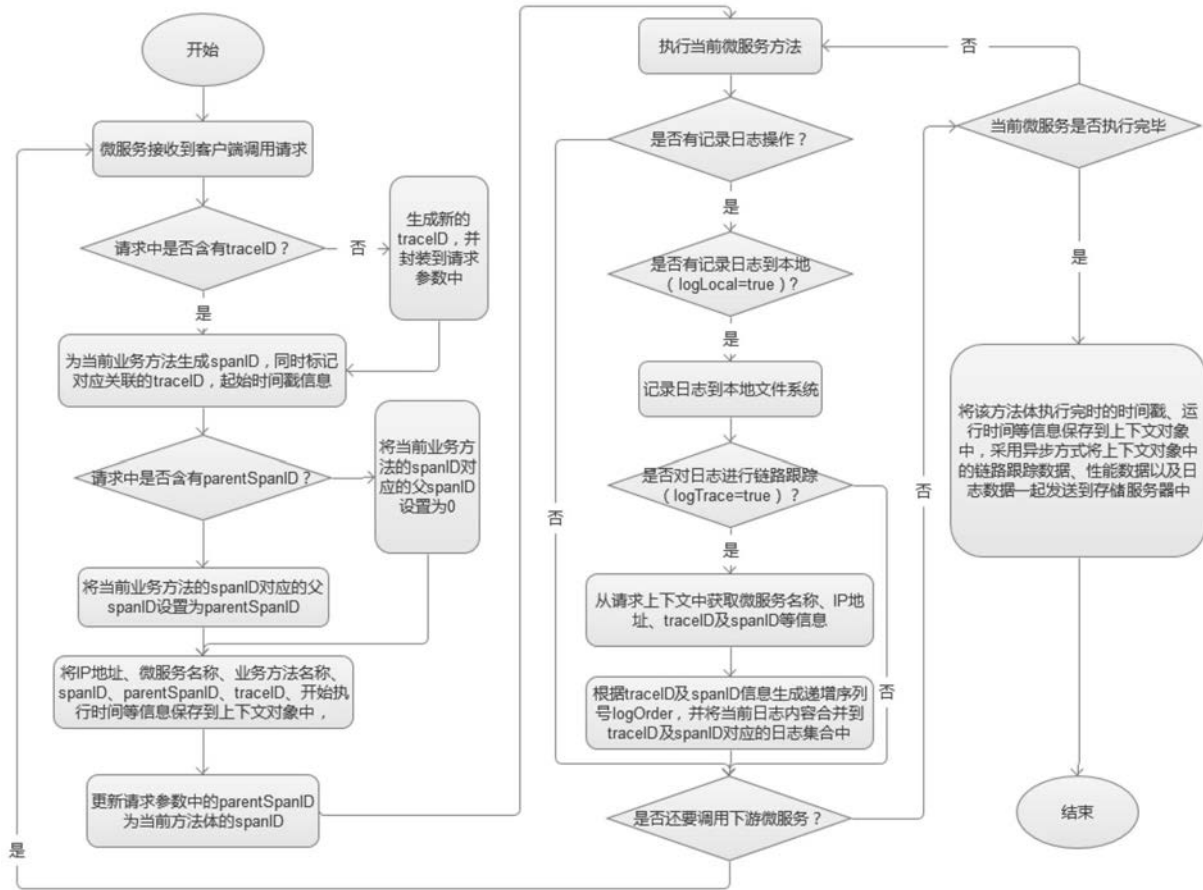


图3