



(12) 发明专利申请

(10) 申请公布号 CN 113946499 A

(43) 申请公布日 2022. 01. 18

(21) 申请号 202111237797.3

(22) 申请日 2021.10.22

(71) 申请人 神州数码系统集成服务有限公司
地址 100085 北京市海淀区西北旺东路10
号院东区18号楼5层101-501

(72) 发明人 张煌 孔祥文 郭涛

(74) 专利代理机构 北京京万通知识产权代理有
限公司 11440
代理人 万学堂

(51) Int. Cl.
G06F 11/34 (2006.01)

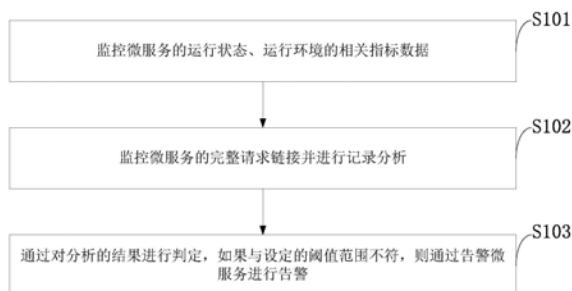
权利要求书2页 说明书8页 附图1页

(54) 发明名称

一种微服务链路跟踪及性能分析方法、系统、设备及应用

(57) 摘要

本发明属于网络监控服务技术领域,公开了一种微服务链路跟踪及性能分析方法、系统、设备及应用,所述微服务链路跟踪及性能分析方法基于微服务架构,面向微服务提供多种集成方式,微服务可根据实际的环境及硬件情况对本系统进行集成;对微服务无任何侵入,微服务可结合服务日志及探针,实时分析请求及响应的完整链路,并对链路进行综合评价。本发明支持三种集成方式,在集成时,对微服务无任何侵入;并且在监控链路信息的过程中,不影响用户体验,既能保证满足服务的链路信息分析,又不会过多占用系统资源;提高了系统的整体运维能力,为企业节省运维及运营成本。自动将性能较低的服务、端点、函数进行展示,全程无需人工干预。



1. 一种微服务链路跟踪及性能分析方法,其特征在于,所述微服务链路跟踪及性能分析方法基于微服务架构,面向微服务提供多种集成方式,微服务可根据实际的环境及硬件情况对本系统进行集成;对微服务无任何侵入,微服务可结合服务日志及探针,实时分析请求及响应的完整链路,并对链路进行综合评价。

2. 如权利要求1所述的微服务链路跟踪及性能分析方法,其特征在于,所述微服务可结合服务日志的实现方法包括:

通过对日志中输出的信息进行分析,在日志中可以输出如下信息:时间、当前的服务信息、内存使用情况、CPU使用情况等,此方式为面向切面的编程思想,根据日志中输出的执行时间或者顺序标记来记录出完整的链路信息,并且解析其他指标记录已完成性能的监控。

3. 如权利要求1所述的微服务链路跟踪及性能分析方法,其特征在于,所述微服务可结合探针的实现方法包括:

jdk1.5以后引入了javaAgent技术,javaAgent是运行方法之前的拦截器,利用javaAgent和ASM字节码技术,在JVM加载class二进制文件的时候,利用ASM动态的修改加载的class文件,在监控的方法前后添加计时器功能,用于计算监控方法耗时,同时将方法耗时及内部调用情况放入处理器,处理器利用栈先进后出的特点对方法调用先后顺序做处理,当一个请求处理结束后,将耗时方法轨迹和入参map输出到文件中,然后根据map中相应参数或耗时方法轨迹中的关键代码区分出本发明要抓取的耗时业务。最后将相应耗时轨迹文件取下来,转化为xml格式并进行解析,通过浏览器将代码分层结构展示出来,方便耗时分析。

4. 如权利要求1所述的微服务链路跟踪及性能分析方法,其特征在于,所述微服务链路跟踪及性能分析方法对服务进行性能分析,可通过探针对微服务的指标数据进行实时推送至服务端,从而实时分析链路的请求及响应,对微服务的状态进行实时跟踪,并对链路进行综合评价。

5. 如权利要求1所述的微服务链路跟踪及性能分析方法,其特征在于,所述微服务链路跟踪及性能分析方法具体包括:

第一步,监控微服务的运行状态、运行环境的相关指标数据;

第二步,监控微服务的完整请求链接并进行记录分析;

第三步,通过对分析的结果进行判定,如果与设定的阈值范围不符,则通过告警微服务进行告警。

6. 如权利要求5所述的微服务链路跟踪及性能分析方法,其特征在于,所述监控微服务的完整请求链接并进行记录分析有三种模式对微服务进行完整的链路监控:日志分析方式、引入jar包方式、Http方式;将链路信息发送给微服务监控模块端;微服务监控模块端根据信息对完整的链接进行实时的评价得分,并且对该条链路的指标进行实时的质量等级划分,对于响应较慢的Top5会进行界面展示,对不在阈值范围内的指标通知告警微服务进行告警;使用日志分析方式时,用户通过管理界面选择需要分析的服务的日志,从而复原完整的链路信息;使用Http方式时,通过探针对微服务的所有请求进行记录并恢复完整的链路信息;使用jar包方式,可提升扩展到对函数级别的性能或问题监控。

7. 如权利要求6所述的微服务链路跟踪及性能分析方法,其特征在于,在所述评价得分的算法为:需要获取某个微服务实例请求相关的指标信息;在响应状态为200的前提下,对

于响应时间在100毫秒以内的请求为满意请求;在响应时间在100毫秒以上3秒内的请求为可容忍请求,3秒以上为不满意请求,响应状态非200则统一归为不满意请求;

评价得分计算模型为:

评价得分 = (满意请求数量 + (不满意请求数量/2)) / 总请求数 * 100;

质量等级划分:以截至上一日存量数据为基础,将质量等级划分为A、B、C、D四个响应等级;

A等级为:一个端点在一分钟内所有满意响应总耗时/所有满意请求的个数的值最终的计算结果为一个响应时间;将小于等于该响应值的响应时间划分为A等级;在可视化界面上对该等级标记为绿色;

B等级为:一个端点在一分钟内可容忍的响应的Top30的总耗时/可容忍的请求的Top30的总个数的值最终的计算结果为一个响应时间;将小于等于该响应值的响应时间划分为B等级;在可视化界面上对该等级标记为蓝色;

C等级为:一个端点在一分钟内可容忍的响应的Top70的总耗时/可容忍的请求的Top70的总个数的值最终的计算结果为一个响应时间;将小于等于该响应值的响应时间划分为C等级;在可视化界面上对该等级标记为淡红色;

D等级:将大于C等级的响应耗时均记为D等级;在可视化界面上对该等级标记为红色。

8. 一种微服务处理设备,其特征在于,所述微服务处理设备包括存储器和处理器,所述存储器存储有计算机程序,所述计算机程序被所述处理器执行时,使得所述处理器执行如下步骤:基于微服务架构,面向微服务提供多种集成方式,微服务可根据实际的环境及硬件情况对本系统进行集成;对微服务无任何侵入,微服务可结合服务日志及探针,实时分析请求及响应的完整链路,并对链路进行综合评价。

9. 一种实施权利要求1~7任意一项所述微服务链路跟踪及性能分析方法的微服务链路跟踪及性能分析系统,其特征在于,所述微服务链路跟踪及性能分析他包括:

微服务运行模块,用于监控微服务的运行状态、运行环境的相关指标数据;

微服务监控模块,用于监控微服务的完整请求链接并进行记录分析;

微服务告警模块,用于通过对分析的结果进行判定,如果与设定的阈值范围不符,则通过告警微服务进行告警。

10. 一种如权利要求1~7任意一项所述微服务链路跟踪及性能分析方法在微服务架构中的应用。

一种微服务链路跟踪及性能分析方法、系统、设备及应用

技术领域

[0001] 本发明属于网络监控服务技术领域,尤其涉及一种微服务链路跟踪及性能分析方法、系统、设备及应用。

背景技术

[0002] 目前,传统的微服务架构在故障分析或者性能优化一般是通过运维人员对服务器的内存、网络、磁盘的各个性能进行分析从而达到预期的目标,这样的方式一是需要人工干预,增加了分析出错的概率,二是在分析期间会影响服务的业务请求,导致此期间的用户请求失败,影响业务进行。

[0003] 随着信息技术的不断发展,互联网在生活和工作方面成为不可或缺的工具。随着互联网服务越来越多,如何对各些复杂的网络服务进行有效合理的监控、优化一直是存在的问题,微服务架构是目前应用比较多并且广泛的软件服务架构。微服务良好的接口规范,方便的服务治理方法提高了软件开发效率及使用效率。

[0004] 现有技术一:CN202010347906.6一种基于微服务架构的软件持续集成方法。本发明公开了一种基于微服务架构的软件持续集成方法,属于互联网软件开发领域,用于软件开发到软件测试的交付,解决了现有软件项目开发过程中,软件持续集成效率差,项目交付测试之前容易出现编译失败,以及没有自动化测试功能,导致影响项目发版部署的问题;包括以下步骤:首先搭建并配置Jenkins CI平台,然后在Jenkins CI平台上构建微服务工程编包Job1,对编包后的微服务安装包打上标识标签并上传至安装包仓库,再按照项目版本的维度进行编排和存放微服务安装包,并通过MD5算法进行数字校验,然后在Jenkins CI平台上带参数构建取包部署测试环境Job2,最后输出测试环境发版清单;使得项目交付测试时能够达到高度持续集成,从而提高软件开发和测试的工作效率。

[0005] 现有技术二:CN201911000437.4微服务架构下软件授权校验方法、终端及软件授权系统。本申请公开了一种微服务架构下软件授权校验方法、终端及软件授权系统,该方法应用于与软件授权系统连接的多个终端中的任意一个终端,所述多个终端分别用于实现基于微服务架构的软件的不同功能,所述方法包括:当接收到功能调用指令时,调用软件授权系统对存储的授权信息进行校验;如果软件授权系统对授权信息校验通过,则确认取得合法授权;执行功能调用指令所指示的操作。本申请可以解决当前单机授权校验方法难以实现的分布式软件授权校验的问题。

[0006] 现有技术三:基于微服务架构的服务发现与服务可靠性的研究_徐康明。基于微服务架构设计的系统由许多通过网络相互通信的小型服务组成,每个服务都有单独的软件开发生命周期,各个服务之间相互依赖。虽然目前对于微服务架构的研究有很多,但仍然存在一些问题。包括服务发现时延过大、准确性不够;对于单个服务的服务质量的描述不够全面;服务在动态环境运行过程中差错较多、无法实时调整等问题。在微服务架构的建模研究中,当前的依赖图生成方案存在不够便捷、准确性较低的问题;当前的服务故障树模型中未考虑差错容忍和系统实际执行情况的问题。在服务风险的研究中,现有服务风险计算模型

存在未考虑系统执行路径概率不同对于最后计算结果的影响。

[0007] 本文主要对微服务架构中服务可靠性、服务发现、系统可靠性建模、系统服务风险以及服务恢复等问题进行了研究。成果概述如下：

[0008] (1) 研究了微服务架构中服务运行可靠性的问题。本文提出了自定义参数的QoS计算模型用于服务质量的计算，可用于服务运行状态的评估。基于服务运行状态设计了自适应的服务模型，达到服务运行过程中的服务差错容忍和提升吞吐量的目标。在本实验环境中，本文提出的QoS计算模型和自适应服务模型使服务运行时的成功率提升了3.87%，吞吐量提升了18.7%。

[0009] (2) 研究了微服务架构中的服务发现问题，如何对应用程序之间的依赖关系进行建模，以及如何使用这些模型来分析微服务架构系统的可靠性。本文提出了依赖图自动化生成方案，并改进了服务故障树模型，同时将其用于系统的可靠性分析。实验结果表明服务故障树模型能够有效的分析微服务架构系统的可靠性。改进后的故障树模型用于分析系统故障率时，准确率上有一定的提升，在实验环境中准确率提高了4.53%。

[0010] (3) 微服务架构中服务有独立部署的特性，如何高效地监控、管理这些服务是一个重要问题。如果仅在部署或运行时解决服务故障往往会导致大量失败任务的产生，因此，服务故障的预测与恢复是同样重要的。本文改进了基于服务风险的差错恢复方案，通过对于改进方案的应用，可以在大量故障发生之前完成风险服务的移除，从而避免连续错误的发生。实验结果表明，改进后的服务风险计算模型和差错恢复方案可以提升系统的稳健性，并减少由于差错恢复方案所导致的系统资源浪费。

[0011] 但基于微服务架构的软件系统存在如下一下问题：

[0012] (1) 无法动态监控服务的链路，需要人工参与，容易引起系统不稳定，提高了系统运维工作复杂性。

[0013] (2) 可视化较差，往往需要通过日志来查看服务调用的信息。

[0014] (3) 无法确定各个微服务到底需要运行多少实例，导致有些服务负载过高影响最大用户并发请求数，有些服务负载很低仍然占用资源，提高了IT成本。

[0015] (4) 在服务出现较底层的故障或者性能问题时，需要人工参与进行分析，耗费的成本较高，且将服务恢复正常状态需要耗费较长时间，影响业务的进行，导致客户无法交易，造成经济损失。

[0016] 通过上述分析，现有技术存在的问题及缺陷为：

[0017] (1) 现有基于微服务架构的软件系统无法动态监控服务的链路，需要人工参与，容易引起系统不稳定，提高了系统运维工作复杂性。

[0018] (2) 现有基于微服务架构的软件系统可视化较差。

[0019] (3) 现有基于微服务架构的软件系统无法确定各个微服务到底需要运行多少实例，导致有些服务负载过高影响最大用户并发请求数，有些服务负载很低仍然占用资源，提高了IT成本。

[0020] (4) 现有基于微服务架构的软件系统在服务出现较底层的故障或者性能问题时，需要人工参与进行分析，耗费的成本较高，且将服务恢复正常状态需要耗费较长时间，影响业务的进行，导致客户无法交易，造成经济损失。

[0021] 解决以上问题及缺陷的难度为：较高。

[0022] 解决以上问题及缺陷的意义为:在不影响正常的业务交易应用的前提下,能够及时展示完整的链路信息,使得一条请求的完整链路进行了一个可视化的展示,对于链路中出现的故障、问题可以做到及时的发现,并且可以帮助客户估算出应用服务到底需要多少硬件资源或在资源紧缺时对客户进行告警,可以大大的降低业务在正常的运行中因为资源而出现故障,导致业务应用出现暂停的问题,保障业务服务可以长时间的平稳运行。

发明内容

[0023] 针对现有技术存在的问题,本发明提供了一种微服务链路跟踪及性能分析方法、系统、设备及应用。

[0024] 本发明是这样实现的,一种微服务链路跟踪及性能分析方法,所述微服务链路跟踪及性能分析方法基于微服务架构,面向微服务提供多种集成方式,微服务可根据实际的环境及硬件情况对本系统进行集成;对微服务无任何侵入,微服务可结合服务日志及探针,实时分析请求及响应的完整链路,并对链路进行综合评价。

[0025] 进一步,所述微服务可结合服务日志的实现方法包括:

[0026] 通过对日志中输出的信息进行分析,在日志中可以输出如下信息:时间、当前的服务信息、内存使用情况、CPU使用情况等,此方式为面向切面的编程思想,根据日志中输出的执行时间或者顺序标记来记录出完整的链路信息,并且解析其他指标记录已完成性能的监控。

[0027] 进一步,所述微服务可结合探针的实现方法包括:

[0028] jdk1.5以后引入了javaAgent技术,javaAgent是运行方法之前的拦截器,利用javaAgent和ASM字节码技术,在JVM加载class二进制文件的时候,利用ASM动态的修改加载的class文件,在监控的方法前后添加计时器功能,用于计算监控方法耗时,同时将方法耗时及内部调用情况放入处理器,处理器利用栈先进后出的特点对方法调用先后顺序做处理,当一个请求处理结束后,将耗时方法轨迹和入参map输出到文件中,然后根据map中相应参数或耗时方法轨迹中的关键代码区分出本发明要抓取的耗时业务。最后将相应耗时轨迹文件取下来,转化为xml格式并进行解析,通过浏览器将代码分层结构展示出来,方便耗时分析。

[0029] 进一步,所述微服务链路跟踪及性能分析方法对服务进行性能分析,可通过探针,对微服务的指标数据进行实时推送至服务端,从而实时分析链路的请求及响应,对微服务的状态进行实时跟踪,并对链路进行综合评价。

[0030] 进一步,所述微服务链路跟踪及性能分析方法具体包括:

[0031] 第一步,监控微服务的运行状态、运行环境的相关指标数据;

[0032] 第二步,监控微服务的完整请求链接并进行记录分析;

[0033] 第三步,通过对分析的结果进行判定,如果与设定的阈值范围不符,则通过告警微服务进行告警。

[0034] 进一步,所述监控微服务的完整请求链接并进行记录分析有三种模式对微服务进行完整的链路监控:日志分析方式、引入jar包方式、Http方式;将链路信息发送给微服务监控模块端;微服务监控模块端根据信息对完整的链接进行实时的评价得分,并且对该条链路的指标进行实时的质量等级划分,对于响应较慢的Top5会进行界面展示,对不在阈值范

围内的指标通知告警微服务进行告警。

[0035] 进一步,使用日志分析方式时,用户通过管理界面选择需要分析的服务的日志,从而复原完整的链路信息。

[0036] 进一步,使用Http方式时,通过探针对微服务的所有请求进行记录并恢复完整的链路信息。

[0037] 进一步,使用jar包方式,可提升扩展到对函数级别的性能或问题监控。

[0038] 进一步,在所述评价得分的算法为:需要获取某个微服务实例请求相关的指标信息;在响应状态为200的前提下,对于响应时间在100毫秒以内的请求为满意请求;在响应时间在100毫秒以上3秒内的请求为可容忍请求,3秒以上为不满意请求,响应状态非200则统一归为不满意请求;

[0039] 评价得分计算模型为:

[0040] 评价得分 = (满意请求数量 + (不满意请求数量/2)) / 总请求数 * 100;

[0041] 质量等级划分:以截至上一日存量数据为基础,将质量等级划分为A、B、C、D四个响应等级;

[0042] A等级为:一个端点在一分钟内所有满意响应总耗时/所有满意请求的个数的值最终的计算结果为一个响应时间;将小于等于该响应值的响应时间划分为A等级。在可视化界面上对该等级标记为绿色;

[0043] B等级为:一个端点在一分钟内可容忍的响应的Top30的总耗时/可容忍的请求的Top30的总个数的值最终的计算结果为一个响应时间。将小于等于该响应值的响应时间划分为B等级。在可视化界面上对该等级标记为蓝色;

[0044] C等级为:一个端点在一分钟内可容忍的响应的Top70的总耗时/可容忍的请求的Top70的总个数的值最终的计算结果为一个响应时间;将小于等于该响应值的响应时间划分为C等级。在可视化界面上对该等级标记为淡红色;

[0045] D等级:将大于C等级的响应耗时均记为D等级;在可视化界面上对该等级标记为红色。

[0046] 本发明的另一目的在于提供一种微服务处理设备,所述微服务处理设备包括存储器和处理器,所述存储器存储有计算机程序,所述计算机程序被所述处理器执行时,使得所述处理器执行如下步骤:基于微服务架构,面向微服务提供多种集成方式,微服务可根据实际的环境及硬件情况对本系统进行集成;对微服务无任何侵入,微服务可结合服务日志及探针,实时分析请求及响应的完整链路,并对链路进行综合评价。

[0047] 本发明的另一目的在于提供一种实施所述微服务链路跟踪及性能分析方法的微服务链路跟踪及性能分析系统,所述微服务链路跟踪及性能分析他包括:

[0048] 微服务运行模块,用于监控微服务的运行状态、运行环境的相关指标数据;

[0049] 微服务监控模块,用于监控微服务的完整请求链接并进行记录分析;

[0050] 微服务告警模块,用于通过对分析的结果进行判定,如果与设定的阈值范围不符,则通过告警微服务进行告警。

[0051] 本发明的另一目的在于提供一种所述微服务链路跟踪及性能分析方法在微服务架构中的应用。

[0052] 结合上述的所有技术方案,本发明所具备的优点及积极效果为:本发明基于微服

务架构,面向微服务提供多种集成方式;微服务可根据实际的环境及硬件情况对本系统进行集成,对微服务无任何侵入,微服务可结合服务日志及探针,实时分析请求及响应的完整链路,并对链路进行综合评价。本发明提供给开发、运维人员进行参考及优化微服务,提供函数级别的执行性能分析,可以使微服务在生产环境下对业务零影响情况下进行快速的性能分析及故障排除。

[0053] 本系统提供了一个完整的线上环境服务链路追踪及性能分析的方法,该系统支持三种集成方式,在集成时,对微服务无任何侵入,并且在监控链路信息的过程中,不影响用户体验,既能保证满足服务的链路信息分析,又不会过多占用系统资源,提高了系统的整体运维能力,为企业节省IT成本。

[0054] 同时该系统支持线上环境服务的性能分析功能,自动将性能较低的服务、端点、函数进行展示,全程无需人工干预,而且整个升级过程对系统性能无明显影响。

[0055] 该系统的微服务扩缩容和微服务升级都对微服务业务开发零侵入,微服务业务开发者不需要关心服务扩缩容和服务升级,所有环节不需要为这些功能增加新的代码或是增加新的配置参数。对软件开发者友好。

[0056] 与现有技术相比,本发明具有以下优势:

[0057] (1) 服务零失败:可以监控到所有的服务的请求及响应。

[0058] (2) 零侵入:所有环节对开发者零侵入,开发者无需关心微服务扩缩容和微服务升级实现细节,无需在代码开发中引入任何和这些功能相关的代码逻辑。

[0059] (3) 实时性高:该系统所有环节均考虑到实时性,几秒内在微服务管理模块页面看到完整的链路信息、及性能分析结果、评分等信息。

[0060] (4) 高并发支持:该系统所有环节考虑实际应用中可能产生的性能瓶颈,采取时许数据库、微服务、容器化等技术对大并发产生的大量数据进行平滑弹性扩容,实现十万级并发规模的TPS业务性能支持。

附图说明

[0061] 图1是本发明实施例提供的微服务链路跟踪及性能分析方法流程图。

[0062] 图2是本发明实施例提供的微服务链路跟踪及性能分析系统的结构示意图;

[0063] 图2中:1、微服务运行模块;2、微服务监控模块;3、微服务告警模块。

具体实施方式

[0064] 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仅用以解释本发明,并不用于限定本发明。

[0065] 针对现有技术存在的问题,本发明提供了一种微服务链路跟踪及性能分析方法和系统,下面结合附图对本发明作详细的描述。

[0066] 如图1所示,本发明提供的微服务链路跟踪及性能分析方法包括以下步骤:

[0067] S101:监控微服务的运行状态、运行环境的相关指标数据;

[0068] S102:监控微服务的完整请求链接并进行记录分析;

[0069] S103:通过对分析的结果进行判定,如果与设定的阈值范围不符,则通过告警微服

务进行告警。

[0070] 基于日志:可通过对日志中输出的信息进行分析,在日志中可以输出如下信息:时间、当前的服务信息、内存使用情况、CPU使用情况等,此方式为面向切面的编程思想,根据日志中输出的执行时间或者顺序标记来记录出完整的链路信息,并且解析其他指标记录已完成性能的监控。

[0071] 基于探针:jdk1.5以后引入了javaAgent技术,javaAgent是运行方法之前的拦截器。本发明利用javaAgent和ASM字节码技术,在JVM加载class二进制文件的时候,利用ASM动态的修改加载的class文件,在监控的方法前后添加计时器功能,用于计算监控方法耗时,同时将方法耗时及内部调用情况放入处理器,处理器利用栈先进后出的特点对方法调用先后顺序做处理,当一个请求处理结束后,将耗时方法轨迹和入参map输出到文件中,然后根据map中相应参数或耗时方法轨迹中的关键代码区分出本发明要抓取的耗时业务。最后将相应耗时轨迹文件取下来,转化为xml格式并进行解析,通过浏览器将代码分层结构展示出来,方便耗时分析

[0072] Java探针工具功能点:

[0073] ①支持方法执行耗时范围抓取设置,根据耗时范围抓取系统运行时出现在设置耗时范围的代码运行轨迹。

[0074] ②支持抓取特定的代码配置,方便对配置的特定方法进行抓取,过滤出关系的代码执行耗时情况。

[0075] ③支持APP层入口方法过滤,配置入口运行前的方法进行监控,相当于监控特有的方法耗时,进行方法专题分析。

[0076] ④支持入口方法参数输出功能,方便跟踪耗时高的时候对应的入参数。

[0077] ⑤提供WEB页面展示接口耗时展示、代码调用关系图展示、方法耗时百分比展示、可疑方法凸显功能。

[0078] 对服务进行性能分析,可通过探针对微服务的指标数据进行实时推送至服务端,从而实时分析链路的请求及响应,对微服务的状态进行实时跟踪,如JVM信息、链路耗时、链路异常状态、同服务下其他节点的状态等,并对链路进行综合评价。

[0079] 本发明提供的微服务链路跟踪及性能分析方法业内的普通技术人员还可以采用其他的步骤实施,图1的本发明提供的微服务链路跟踪及性能分析方法仅仅是一个具体实施例而已。

[0080] 如图2所示,本发明提供的微服务链路跟踪及性能分析他包括:

[0081] 微服务运行模块1,用于监控微服务的运行状态、运行环境的相关指标数据。

[0082] 微服务监控模块2,用于监控微服务的完整请求链接并进行记录分析。

[0083] 微服务告警模块3,用于通过对分析的结果进行判定,如果与设定的阈值范围不符,则通过告警微服务进行告警。

[0084] 下面结合附图对本发明的技术方案作进一步的描述。

[0085] 本发明的微服务监控模块有三种模式对微服务进行完整的链路监控:日志分析方式、引入jar包方式、Http方式。使用日志分析方式时,用户通过管理界面选择需要分析的服务的日志,从而复原完整的链路信息。使用Http方式时,通过探针对微服务的所有请求进行记录并恢复完整的链路信息。若使用jar包方式,可提升扩展到对函数级别的性能或问题监

控。以上三种方式,均会将链路信息发送给微服务监控模块端。微服务监控模块端会根据这些信息对完整的链接进行实时的评价得分,并且对该条链路的指标进行实时的质量等级划分,对于响应较慢的Top5会进行界面展示,对不在阈值范围内的指标通知告警微服务进行告警。

[0086] 在本发明中评价得分算法为:需要获取某个微服务实例请求相关的指标信息,比如响应时间,响应状态,TPS数量。在响应状态为200的前提下,对于响应时间在100毫秒以内的请求为满意请求,在响应时间在100毫秒以上3秒内的请求为可容忍请求,3秒以上为不满意请求。响应状态非200则统一归为不满意请求。

[0087] 评价得分计算模型为:

[0088]
$$\text{评价得分} = (\text{满意请求数量} + (\text{不满意请求数量}/2)) / \text{总请求数} * 100$$

[0089] 质量等级划分:以截至上一日存量数据为基础,将质量等级划分为A、B、C、D四个响应等级。

[0090] A等级为:一个端点在一分钟内所有满意响应总耗时/所有满意请求的个数的值最终的计算结果为一个响应时间。将小于等于该响应值的响应时间划分为A等级。在可视化界面上对该等级标记为绿色。

[0091] B等级为:一个端点在一分钟内可容忍的响应的Top30的总耗时/可容忍的请求的Top30的总个数的值最终的计算结果为一个响应时间。将小于等于该响应值的响应时间划分为B等级。在可视化界面上对该等级标记为蓝色。

[0092] C等级为:一个端点在一分钟内可容忍的响应的Top70的总耗时/可容忍的请求的Top70的总个数的值最终的计算结果为一个响应时间。将小于等于该响应值的响应时间划分为C等级。在可视化界面上对该等级标记为淡红色。

[0093] D等级:将大于C等级的响应耗时均记为D等级。在可视化界面上对该等级标记为红色。

[0094] 每日需对质量等级进行重新计算,不断更正等级的值,使得评分在海量数据下更为精准。

[0095] 性能分析:对于触发告警阈值的指标,以卡片的形式在可视化界面进行展示,并将卡片的背景色显示为淡红色,点击卡片对链路中的每一个节点的请求信息进行界面展示并对,如:服务名称、端点名称、响应时间,并展示相同的指标下为A等级标准的指标数据。对于更为精细的函数级别性能分析,对函数中耗时最长的一行代码进行标记。

[0096] 对全链路信息及服务性能进行实时分析,并不断对评分等级进行自动优化。对运维及开发人员提供高质量的优化参考方案。

[0097] 微服务:微服务是一种架构风格,一个大型复杂软件应用由一个或多个微服务组成。系统中的各个微服务可被独立部署,各个微服务之间是松耦合的。每个微服务仅关注于完成一件任务并很好地完成该任务。在所有情况下,每个任务代表着一个小的业务能力。

[0098] 微服务实例:每个微服务可以由一个或多个应用合到一起对外提供,每个应用称为微服务的一个实例,多个实例一般通过负载均衡器通过轮询的方式对外提供该微服务。

[0099] TPS:Queries Per Second意思是“每秒查询率”,是一台服务器每秒能够相应的查询次数,是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准。TPS是TransactionsPerSecond的缩写,也就是事务数/秒。它是软件测试结果的测量单位。

[0100] 应当注意,本发明的实施方式可以通过软件来实现。硬件部分可以利用专用逻辑来实现;软件部分可以存储在存储器中,由适当的指令执行系统,例如微处理器或者专用设计硬件来执行。本领域的普通技术人员可以理解上述的设备和方法可以使用计算机可执行指令和/或包含在处理器控制代码中来实现,例如在诸如磁盘、CD或DVD-ROM的载体介质、诸如只读存储器(固件)的可编程的存储器或者诸如光学或电子信号载体的数据载体上提供了这样的代码。本发明的设备及其模块可以由诸如超大规模集成电路或门阵列、诸如逻辑芯片、晶体管等的半导体、或者诸如现场可编程门阵列、可编程逻辑设备等的可编程硬件设备的硬件电路实现,也可以用由各种类型的处理器执行的软件实现,也可以由上述硬件电路和软件的结合例如固件来实现。

[0101] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,都应涵盖在本发明的保护范围之内。

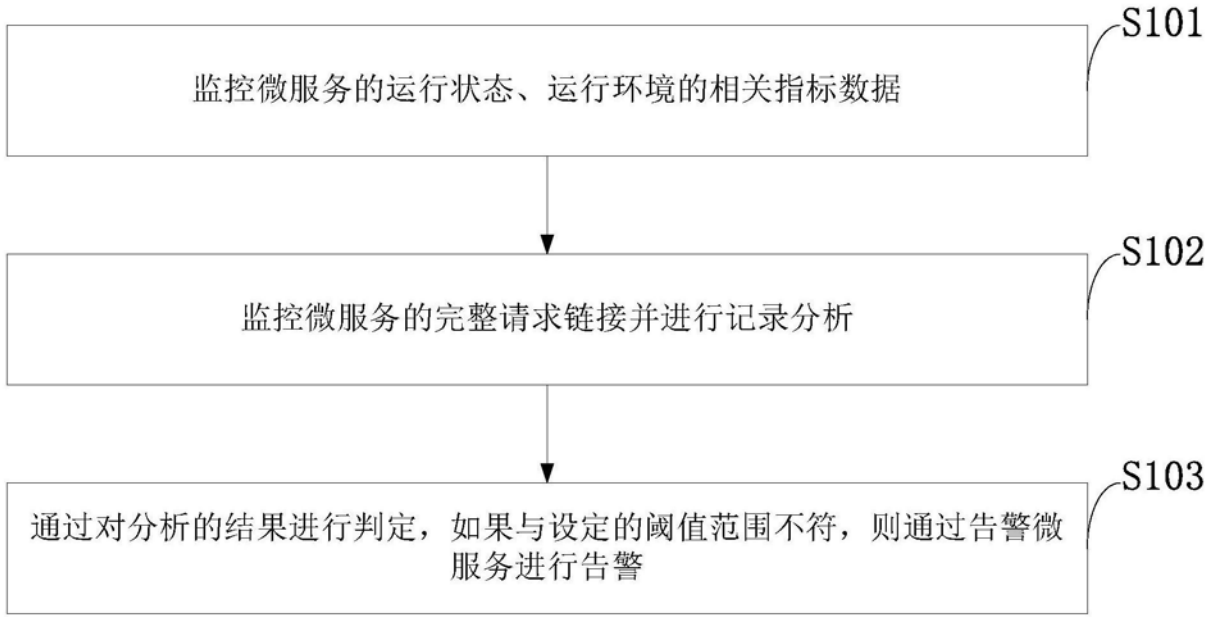


图1

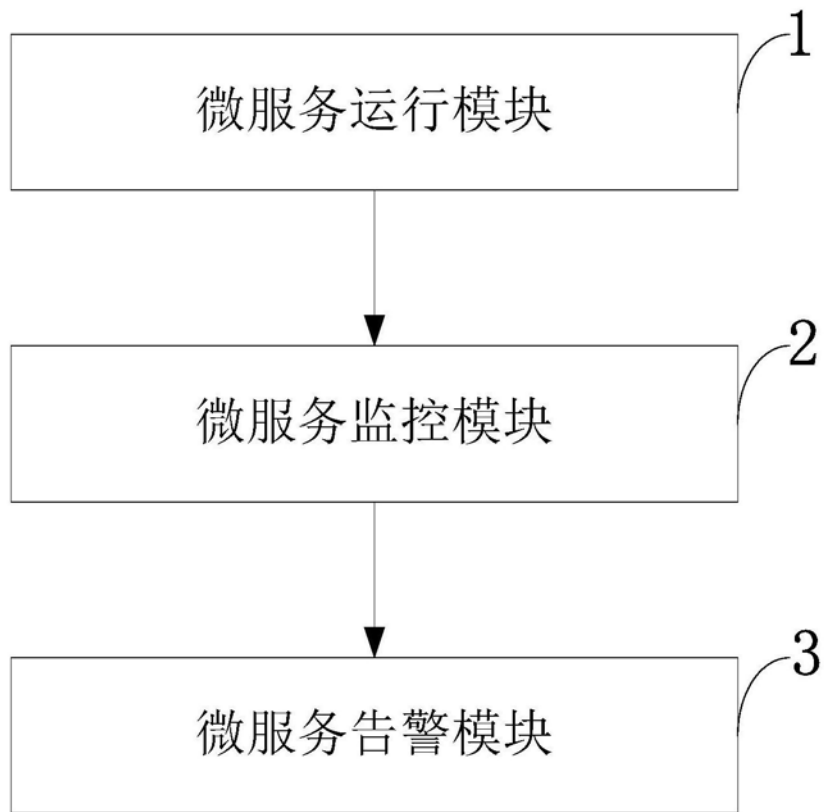


图2