

文章编号:1005-0523(2014)01-0119-05

基于流式计算的Web实时故障诊断分析与设计

刘子英,唐宏建,肖嘉耀,张 骞

(华东交通大学电气与电子工程学院,江西 南昌 330013)

摘要:随着Web服务技术在分布式系统的广泛应用,Web服务正逐步成为Internet环境中资源封装的标准形式,Web服务粗粒度、松耦合等特性在增加业务应用的灵活性和分布式系统的自适应能力同时,也增加了应用系统的复杂性。为了提高Web服务分布式系统的可用性,提出了采用Storm流式计算架构的Web故障诊断新方式,为实时发现故障问题以及迅速排除故障提供了方案。

关键词:Web服务;Storm;流式计算;故障诊断

中图分类号:TP391

文献标志码:A

故障诊断虽然在控制领域应用广泛,但在Web服务领域,还比较原始,形象地说就是人工故障诊断,诊断效率非常低。基于上述原因提出了一个Web故障^[1-3]的自动化解决方案,它可以实时发现问题、自动定位问题(或者辅助定位问题),还能预测问题,大大提高工作效率、提升系统监控状态^[4]。自动化有两个主要的问题需要考虑,一是根据系统的依赖关系和部署状况,来分析问题;二是上述分析过程需要实时的一个解决方法,如果故障诊断在故障发生1 min后才出结果,就不利于故障问题的快速解决。Web服务请求常常是由一系列RPC(Remote Procedure Call Protocol:远程过程协议调用)调用^[5-6]过程组成的,图1给出了其服务依赖树的举例。

Web服务,建立通过任务间依赖规范实施合成的方法,具有柔性和自适应能力实现Web服务的自动合成,而Web服务中族的概念阐述了与服务树的关系,文中服务族相对于服务树来说是更多Web服务依赖树的集合。

图1中用户(User)请求通常在经过一系列处理之后得到最终结果。如果系统任何一个节点(RPC拓扑中的服务)出现问题,都会影响最终用户的使用,如果出现问题常常很难知道问题出在哪个节点。导致的原因主要有

- 1) Web服务变化太快,新节点随时会增加、旧节点随时会修改,很难知道系统依赖了哪些节点;
- 2) 一些节点及节点所在的机器很可能是被多个系统共用的。

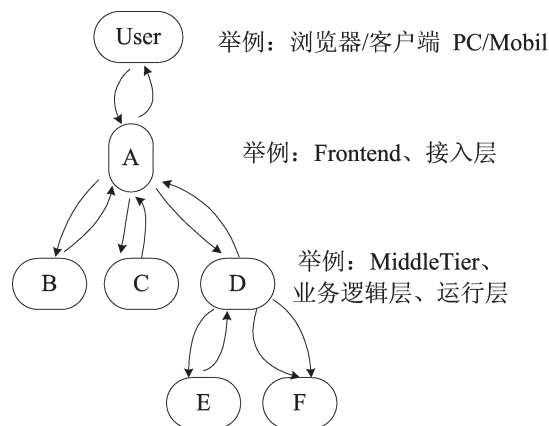


图1 Web系统RPC拓扑图

Fig.1 RPC topological graph of the Web system

收稿日期:2013-08-27

作者简介:刘子英(1964—),女,教授,硕士,研究方向为信息管理系统以及数据库技术。

系统的依赖关系和部署情况之所以成为比较困难的问题,是因为很难以人工方式生成,因此必须实现自动化生成。要实现自动化生成可以借鉴 Web 服务中人工故障诊断分析日志的方法,找到规律,并通过程序来实现。

因此文中提出的解决方案就是先制定一套日志规范,然后基于日志产出系统的依赖关系和部署状况及其他数据并应用实时流式计算框架来实现这套计算逻辑,以保证实时性。

1 日志规范

Web 服务日志^[7]能够记录 Web 服务器捕获的所有请求,因此 Web 服务日志对故障诊断、数据分析有非常重要的意义,因此根据系统 RPC 拓扑结构可以制定一套可用的日志规范,使用这套日志规范的目标是能够自动分析出系统的 RPC 拓扑结构,同时包含其他必要的项目。

日志规范的约定条件:

- 1) RPC 拓扑节点,每个节点表示一个软件组件,通常部署在由多台服务器组成的集群上;
- 2) 每次请求响应流经任何节点,必须由日志打印,其中正常日志 Notice ≤ 1 条,异常日志 Warning ≥ 0 条,Notice 和 Warn 日志都必须包含完整的字段信息;
- 3) 一次用户请求响应,无论流经多少节点,都采用唯一的 traceId 标记、贯穿始终;
- 4) 在一个系统内,节点名(node)要求必须唯一;
- 5) 如果一次请求可能流经同一节点多次(无论是否在同一台服务器),则需要用唯一的 nodeId 记录节点的每次处理;
- 6) 每次跟 RPC 交互时,都需要有唯一的 rpcId 标记;
- 7) 给出其他故障诊断的字段:logTime(打印日志的时间 ms)、flag(是否异常标记)、cost(响应时间 ms)、ip(本机 IP)。

2 基于日志规范的 RPC 拓扑生成算法及数据分析

2.1 RPC 工作模式^[8]

通过 RPC 我们可以从网络上的计算机请求服务,而不需要了解底层网络协议。RPC 采用客户机/服务器模式。请求程序就是一个客户机,而服务提供程序就是一个服务器。首先,客户机调用进程发送一个有进程参数的调用信息到服务进程,然后等待应答信息。在服务器端,进程保持睡眠状态直到调用信息的到达为止。当一个调用信息到达,服务器获得进程参数,计算结果,发送答复信息,然后等待下一个调用信息,最后,客户端调用进程接收答复信息,获得进程结果,然后调用执行继续进行。

2.2 RPC 拓扑生成算法

根据上述日志规范给出 RPC 拓扑生成算法的实现过程,算法的主要步骤是将探测源模拟成一个网管站与 SNMP Agent 通信,先取得探测源的默认网关,存入待探测队列(ToDoList)。依次取出 ToDoList 中的 IP 地址,获得 IP 的 MIB 库中的路由表(ipRouteTable)中的数据。当 ipRouteType=indirect 时,可以获得路由的下跳地址(ipRouteNextHop),由此获得路由器-路由器的连接,并将 ipRouteDest 存入 ToDoList;当 ipRouteType = direct 时,利用 ipRouteDest 可以获得当前路由器连接的子网。然后 Ping 子网内的每个 IP 地址,析取 ICMP 回应报文的 IP 地址,确定出子网内活动主机的信息。当 ToDoList 所有 IP 均被处理后,生成网络拓扑结构图。下面给出用户的一次请求响应过程。如图 2 所示,假设一次请求的 traceId=T1,node 分为 U,A,B,⋯,F,总共 6 次 RPC 交互的 rpcId 分别为 R1,R2,⋯,R6。分析 RPC 拓扑时不需要 logTime,flag,cost 和 ip,图 2 中将其省略了。

生成请求拓扑日志描述:rpcId 该列的含义为,生成请求拓扑节点作为下游的 rpcId 以及节点作为上游的 rpcId。对应到树中,就是 RPC 节点入边以及节点出边。

由此,生成RPC拓扑的问题可以抽象成:已知树的节点及节点的入边和所有出边,求生成树。算法过程如图3所示(先假设已把同一traceId的日志汇聚在一起)。

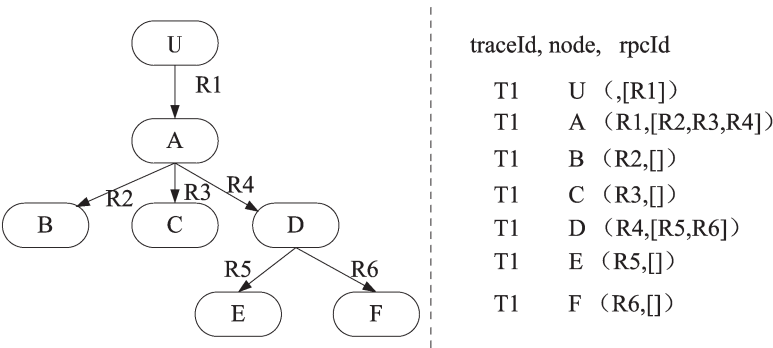


图2 生成请求拓扑的日志过程

Fig.2 Log process of survival request topology

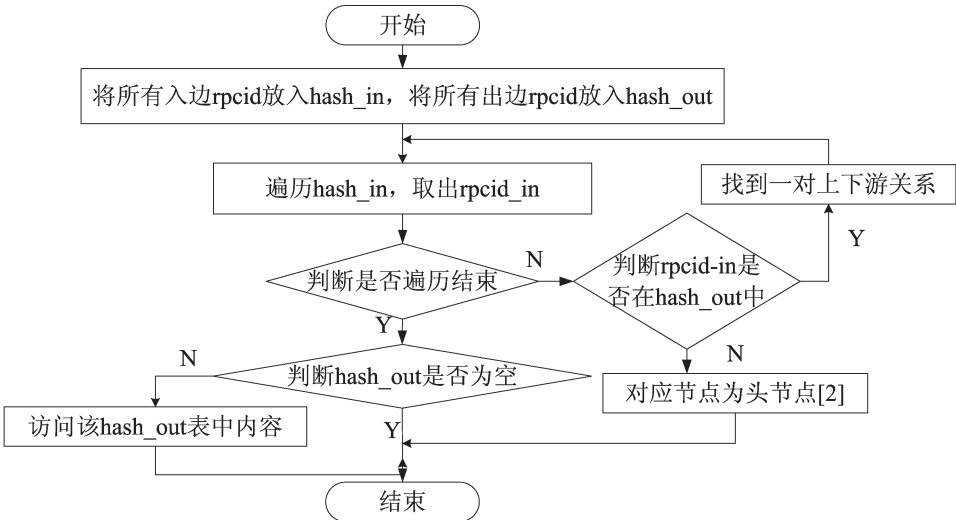


图3 算法的生成请求拓扑

Fig.3 Survival request typology of algorithm

上述算法的时间复杂度为: $t = n + m$ (n :入边个数, m :出边个数)。

因此根据生成的RPC请求拓扑,就能分析出web服务的依赖关系。系统出现的问题常常会反映到用户使用的终端上,也就是树的根节点上,根据服务的依赖关系就能找到故障问题所在。因此,根据RPC拓扑可以实现自动化定位故障问题,或者辅助定位故障问题。

2.3 统计数据分析

上述日志规范中字段logTime有两个用途:

- 1) 对同一节点发起多次交互时,对这些交互进行排序;
- 2) 在计算统计数据时需要按区间汇聚。比如汇聚[22:10:10,22:10:20]这10 s的数据就要参照log-Time选择区间。

这里提到系统数据,就是拓扑中的每个节点,以及每个节点所在机器,都可以按一定时间区间计算统计数据,包括计算次数(如异常次数)、平均值(如响应时间平均值)。

数据统计与分析拓扑图的关系,统计数据的生成是依赖于请求拓扑图的。对于服务依赖或者说系统

整体拓扑图,需要根据单次请求的拓扑图,才能分析出来。这里需要注意的是

- 1) 任何节点都可以被不同的系统复用;
- 2) 一个系统内部上下游的依赖是有区别的,统计数据可以用于发现问题,如果采用足够小的时间区间,如10 s,那么统计数据就能做到10 s实时,也就是实时发现问题;
- 3) 统计数据也能用于分析问题,假设某节点表现出问题了,根据服务的依赖拓扑,分析对比该节点依赖的节点的统计数据,就能发现问题。

3 系统总体实现框架

3.1 流式计算的拓扑结构

故障诊断系统的核心在流式处理^[9]的计算逻辑中,前面描述的所有计算都在这里实现。流式处理的拓扑结构,如图4所示。

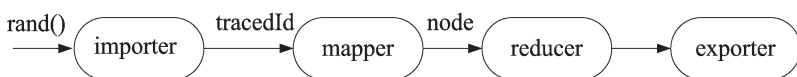


图4 流式计算拓扑结构

Fig.4 Typological structure of stream computing

可套用 storm 的定义^[10-11],对各节点进行描述。

- 1) importer是消息源(spout),它从外部源读取数据,做日志切分,产出提取数据。
- 2) mape/reducer是核心部分,从命名就能看出 mapper 会汇聚部分数据,而 reducer 完成最终数据的汇聚;mapper 包含生成树的逻辑。另外,mapper 和 reducer 内部都需要维护状态数据(比如10 s统计数据),因此在处理上分为 insert 和 nSecProc。
- 3) exporter,用于输出。

3.2 故障诊断系统的架构

故障诊断系统的输入符合规范日志数据,系统基于对日志数据的实时分析,实现所有功能。故障诊断系统的架构如图5所示。

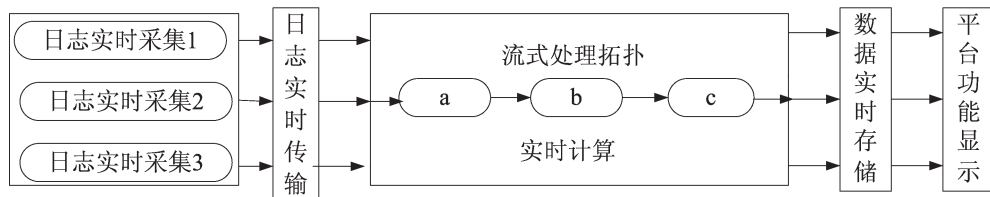


图5 故障诊断系统框架图

Fig.5 Frame diagram of fault diagnosis system

系统整体分为实时采集、实时传输、实时计算、实时存储、平台功能显示等部分。

- 1) 实时采集、传输:可选的方案有 Facebook 开源的 scribe、LinkedIn 开源的 Kafka、Cloudera 开源的 Flume、Hadoop 开源的 Chukwa。
- 2) 实时计算:采用流式计算框架,可选的方案 Twitter 开源的 Storm、Yahoo、开源的 S4 以及还有 Esper、Streambase、HSStreaming。
- 3) 数据实时存储:本系统的特点是写多读少,可以做相应的优化。
- 4) 平台功能显示:故障信息能通过平台功能友好的展现。

4 结束语

首先分析了故障诊断的重要性以及实现思路,然后分析了故障诊断的实现方法,列出了日志规范和生

成拓扑的算法,介绍了统计数据的产生与作业;最后给出具体的技术架构,尤其是流式计算的拓扑结构。将该方法应用于实际系统之后,可以更好地发现问题,分析问题的效率也有很大的提高,从而使系统整体的健康状况大大提升。

参考文献:

- [1] 蹇崇军,洪欣. 基于Web服务的FMS远程故障诊断系统研究[J]. 机械设计与制造,2011,12(12):258-260.
- [2] 马笑潇,黄席樾,柴毅. 基于SVM的二叉树朵类分类算法及其在故障诊断中的应用[J]. 控制与决策,2004,18(3):272-284.
- [3] 刘丽,况晓辉,方兰,等. Web服务故障分类方法[J]. 计算机系统应用,2010,19(8):258-263.
- [4] 任洪敏,刘晋. 基于链接件的Web服务自愈框架研究[J]. 计算机应用研究,2012,29(8):2999-3007.
- [5] 许卓明,栗明,董逸生. 基于RPC和基于REST的Web服务交互模型比较分析[J]. 2003,29(20):6-8.
- [6] 谭海波,马宗萼,李晓风. Web service 分布式网络测量的研究和实现[J]. 计算机工程与设计,2012,33(6):2241-2245.
- [7] 袁佳琳,高建华. 基于工作量和错误日志的Web应用可靠性度量方法[J]. 计算机应用与软件,2012,29(10):44-49.
- [8] 文威. 网络应用层数据分类采集器的设计与实现[D]. 武汉:华中科技大学,2007.
- [9] 顾昕. 分布式流式计算框架关键技术的研究与实现[D]. 北京:北京邮电大学,2012.
- [10] 马如悦. 分布式流式计算平台-S4[J]. 程序员,2011,103(1):103-105.
- [11] 陈涛. 一种流式计算平台的设计与实现[D]. 合肥:中国科学技术大学,2012.

Analysis and Design of Web Real-time Fault Diagnosis Based on Stream Computing

Liu Ziyang, Tang Hongjian, Xiao Jiayao, Zhang Qian

(School of Electrical and Electronic Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: With wider applications of Web service technology in distributed application system, the Web service is gradually becoming the standard form of resource packages in internet environment, though its features of coarse-grainedness, loosely-coupledness increase the complexity of the application system in addition to the flexibility and self-distributed systems adaptation. In order to improve the availability of distributed application system based on the Web service, this paper put forward a new mode of Web fault diagnosis, proposing the storm stream computing architecture in order to provide rapid troubleshooting scheme for the real-time faults.

Key words: the Web service; Storm; stream computing; fault diagnosis