

一种适用于 AETA 的日志系统的设计与实现

周康生¹, 张 兴¹, 王新安¹, 雍珊珊¹, 李柏杭¹, 张 丹²

(1. 北京大学深圳研究生院 地震监测预测技术研究中心, 广东 深圳 518055;
2. 武汉大学 计算机学院, 湖北 武汉 430072)

摘 要:多分量地震监测预测系统 AETA 已经在中国四川、云南、河北、广东、西藏、台湾等地区布设 250 余套。目前, 单台服务器平均每天需要接收并处理 100 万条以上的日志。海量的日志数据如果不加以规范化处理, 将对服务器的并发处理能力提出挑战, 并且会极大地增加系统运维工作的难度。基于多分量地震监测预测系统 AETA, 设计了一套日志系统, 从日志数据采集层和日志数据汇集层协同设计, 有效地降低了日志数据给服务器集群带来的并发请求, 并提高了服务器集群的高并发处理能力, 保证了服务的稳定运行。采用批处理技术对日志类型和日志内容进行统计分析, 使得运维人员可以快速了解各台站设备的运行情况。设计了一个网页对日志相关内容进行展示, 运维人员能够在界面友好的网页完成对日志的查询操作。

关键词:地震监测预测; 日志系统; 高并发处理; 批处理; 系统运维

中图分类号: TP302

文献标识码: A

文章编号: 1673-629X(2019)12-0008-06

doi: 10.3969/j.issn.1673-629X.2019.12.002

Design and Implementation of a Log System for AETA

ZHOU Kang-sheng¹, ZHANG Xing¹, WANG Xin-an¹, YONG Shan-shan¹, LI Bo-hang¹, ZHANG Dan²

(1. Earthquake Monitoring and Prediction Technology Research Center, Peking University

Shenzhen Graduate School, Shenzhen 518055, China;

2. School of Computer Science, Wuhan University, Wuhan 430072, China)

Abstract: Multi-component seismic monitoring and prediction system AETA has been deployed in more than 250 sets in Sichuan, Yunnan, Hebei, Guangdong, Tibet and Taiwan. Currently, a single server needs to receive and process more than 1 million logs per day on average. If the massive log data is not standardized, it will challenge the concurrent processing capability of the server and greatly increase the difficulty of operation and maintenance. Based on AETA, a log system is designed from the log data collection layer and the log data collection layer to effectively reduce the concurrent requests brought by the log data to the server cluster and improve the server cluster. Concurrent processing capability ensures stable operation of the service. The batch analysis technology is used to perform statistical analysis on the log types and log contents, so that O&M personnel can quickly understand the running status of each station device. A web page is designed to display the log related content, so that the operation and maintenance personnel can complete the query operation of the log on the friendly webpage.

Key words: seismic monitoring and prediction; log system; high concurrent processing; batch processing; system operation and maintenance

0 引 言

日志可以记录下系统运行过程中所产生的关键信息, 并且按照某种规范表达出来, 有助于运维人员和开发人员了解系统运行状态, 快速定位系统问题^[1-3]。

规范和充分的日志是良好代码质量的必要因素, 也是软件故障诊断的重要手段^[4]。随着计算机技术的发展, 大规模的软件系统层出不穷, 日志对于系统的重要性越来越明显。文献[5]对 Apache httpd, OpenSSH,

收稿日期: 2019-01-07

修回日期: 2019-05-09

网络出版时间: 2019-06-27

基金项目: 广东省省级科技计划项目(2014B090913001); 深圳市科技计划项目(JCYJ20170412151159461); 深圳市科技计划项目(KJYY20170721151955849)

作者简介: 周康生(1994-), 男, 硕士, 研究方向为地震监测预测系统; 张 兴, 教授, 研究方向为新结构器件、纳米级 MOS 器件, CMOS 集成电路设计及加工工艺, 嵌入式系统设计, 地震监测预测技术; 王新安, 教授, 研究方向为集成微系统、地震监测预测技术和生命健康监测康复技术。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190627.1115.082.html>

PostgreSQL 和 Squid 等常用开源软件进行了分析,结果表明,平均每 30 行代码中就有一行是日志;文献[5-6]通过对部分大规模开源软件的失效报告进行随机采样分析,发现 77% 的系统失效可以归结为几类常见的错误诊断模式,而其中 57% 的错误没有记录日志信息,导致运维人员需要花费大量的精力来定位错误。文献[7]对微软内部的 54 个经验丰富的开发人员进行了调查研究,并得出结论:适量的日志对于故障诊断起着非常重要的作用。

在一个计算机应用系统中可能会包含很多的信息设备,所有设备产生的日志信息总量是巨大的,如果将所有信息都进行记录,将会影响整个系统的性能。因此,尽量减少计算机资源占用率并且高效地收集日志数据是日志系统设计的重点。此外,日志系统若将这些数据收集起来,却不加以处理,如此海量的日志数据将会极大地增加系统运维人员的工作量。北京大学深圳地震监测预测技术研究中心研发的多分量地震监测预测系统 AETA 正在全国地震高发区域大规模布设,经过统计和分析发现,目前单台服务器每天需要处理 100 万条以上的日志,而大量的日志信息和数据并发量过多地挤占了服务器的处理能力和处理时间。因此,为了改变这种现状,需要设计一套日志系统,规范化日志的描述、采集和存储,并且对收集到的日志进行统计分析,提高服务器的资源利用率,降低系统维护人员的运维成本。

1 多分量地震监测预测系统 AETA

多分量地震监测预测系统 AETA 由地声传感探头、电磁传感探头、数据处理终端,及监测数据云平台和分析系统组成,如图 1 所示。

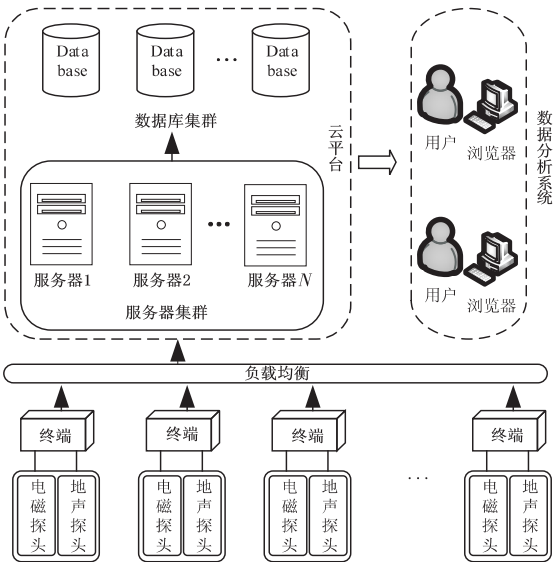


图 1 多分量地震监测预测系统 AETA 系统框架

传感探头感知来自地下的电磁扰动和地声信号,

数据处理终端实时采集数据并通过互联网(有线或无线)将数据传输到云平台进行特征提取、持久化存储和异常分析等。

截止目前,在中国地震局的支持下,AETA 系统布设范围已覆盖了河北、四川、云南、西藏、广东和台湾等地区,其中在四川布设密度最大,布设数量已达 100 余套,基本覆盖四川全境重点区域^[8-14]。AETA 系统具有跨区域布设、布设数量大等特点,通过现场分析来定位问题和了解设备运行状况成本高、效率低,而日志技术可以记录软件的运行轨迹,进而回溯软件运行过程,快速定位问题所在;如果能够将设备记录的日志数据传送到数据中心,并进行持久化存储,就可以在本地对日志数据进行分析来达到定位问题的目的,从而降低运维成本。因此,设计一个日志系统对 AETA 系统的运维管理具有重要的作用。

2 AETA 日志系统整体框架

AETA 日志系统是基于 AETA 数据采集系统设计的,由日志数据采集层、日志数据汇集层、持久化存储层和日志数据展示层组成,如图 2 所示。

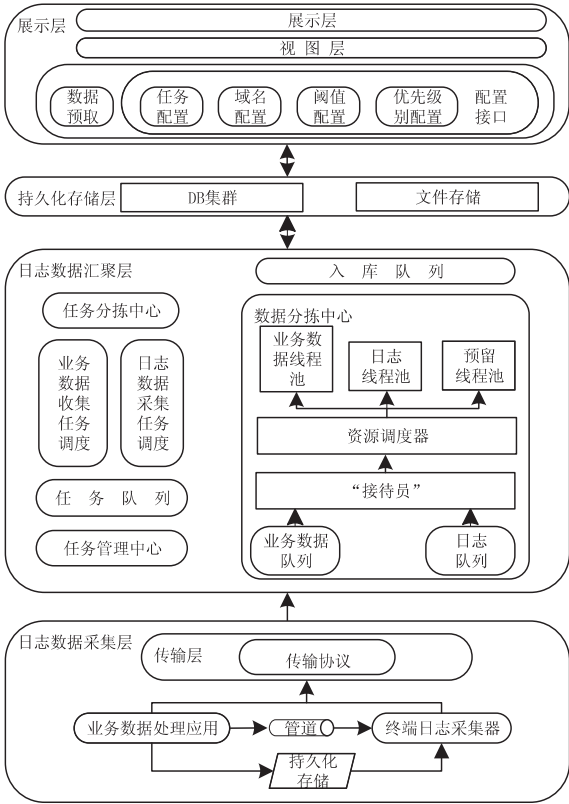


图 2 AETA 日志系统框架

日志数据采集层负责对设备的日志信息和状态信息进行采集,并将数据规范化处理,然后按照约定的应用层传输协议将数据传输至日志数据汇集层;其中,日志数据源是业务数据处理应用程序,而终端日志采集器完成日志数据的采集、规范化处理和传输。

汇聚层主要分为两个部分:任务管理部分和数据分拣中心。任务管理部分包括任务分拣中心、各类任务调度中心和任务管理中心,负责管理采集层的设备(终端和探头),检测设备的运行状态,实现设备任务分解、协同等调度工作;数据分拣中心负责接收采集层上传的电磁数据、地声数据、日志数据和状态数据,将数据分类处理,然后将处理好的数据送入入库队列,由入库队列统一入库;持久化存储层采用 MySQL 数据库和 Linux 文件系统存储数据,其中日志数据全部存储在 MySQL 数据库中。持久化存储层通过对数据库中的数据进行分析,生成事件、告警、视图等元素的基本信息。展示层通过 B/S 方式展现用户界面,通过数据预取技术将用户可能关心的数据预先缓存到用户本地,提高展示的效率,方便用户使用。此外,还可为用户提供配置任务、优先级别、阈值和域名的接口,为用户提供个性化服务。

3 AETA 日志系统的设计与实现

3.1 日志数据采集层

日志数据采集层部署在数据处理终端,主要完成数据处理终端日志的收集、处理和传输功能。由图 3 所示,日志数据采集层主要由数据处理应用和终端日志采集器组成,其中数据处理应用是 AETA 系统数据采集的关键应用,也是日志数据的来源,而终端日志采集器则通过管道(一种进程间通信方式)或者持久化存储设备获取数据处理应用产生的日志数据。终端日志采集器通过 HTTP 协议与云服务器通信,将日志内推发送到服务器。每一次的数据发送都是一个发向服务器的 HTTP 请求,服务器在接收到 HTTP 请求后,将其放入待处理请求队列,并为每一个请求设置超时时间,对于超时的请求则直接丢弃。

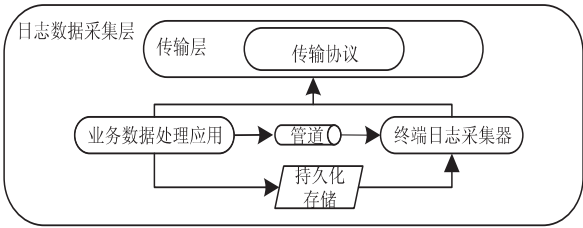


图 3 日志数据采集层

数据处理应用总是在实时处理来自传感探头的数据,为了比较完备地记录软件运行过程中的关键信息,会产生大量的日志。如果每产生一条日志就向服务器发送,则服务器会同时接收到大量的 HTTP 请求。由于服务器的处理能力有限,将会导致待处理请求队列中大量的请求超时失效。因此,为了降低终端日志采集器向服务器发送 HTTP 请求的频率,在日志数据采集层对收集到的日志作了合并处理。

表 1 是系统日志合并协议,其中 TerminalId 是该数据处理终端的编号,表示日志内容来自于哪台终端设备;LogNums 表示合并的日志条数,以便于后续对日志内容进行解析处理;LogFlagN 作为分隔多条日志的标志;LogN 则代表单条日志的内容,如表 2 所示。其中 Time 记录该条日志产生的 Unix 时间戳。数据处理终端除了能够记录数据处理应用产生的日志,还能够将电源和传感探头的运行状况以日志的形式记录下来。

表 1 多分量地震监测预测系统 AETA 日志合并协议

参数名	参数值
TerminalId	终端号
LogNums	日志条数
LogFlag1	日志 1 分界标志
Log	日志 1 的内容
...	...
LogFlagN	日志 N 分界标志
LogN	日志 N 内容

表 2 多分量地震监测预测系统 AETA 单条日志内容格式

参数名	参数值
Time	Unix 时间戳,代表日志产生时间
DeviceType	Dev01(终端)
	Dev02(电源)
	Dev03(探头)
DeviceId	设备编号
LogType	日志类型
LogLength	日志长度
LogContent	日志内容

因此用 DeviceType 来表示该条日志记录的是哪个设备的信息,Dev01、Dev02 和 Dev03 为约定的标识,分别代表终端、电源和探头;Loglength 和 LogContent 分别代表日志长度和日志内容;LogType 代表日志的类型或者等级。

虽然日志合并能够很大程度上减少服务器由于日志请求带来的高并发量,但是,日志合并需要等待多条日志才能进行合并处理,降低了日志的实时性。为了保证部分关键的日志能够实时发送到服务器,对日志内容按照事件的严重程度进行了分级。管道是一种进程间通信机制,对于严重级别高的日志通过管道机制直接发往终端日志采集器,由终端日志采集器对日志进行处理后直接发往服务器;对于实时性要求不高的日志内容,存储在数据处理终端的持久化存储设备中。终端日志采集器每隔一定的时间从持久化存储设备中读取一定量的日志进行合并处理后再发往服务器。国外成熟的系统软件,都对日志和告警进行了分级定义,

以 syslog 为例,将日志分为八种安全级别,分别为紧急、告警、严重、错误、警告、通知、信息和调试^[15];借鉴于 syslog 的日志分级方式和对多分量地震监测预测系统 AETA 的业务需求分析,将日志内容分为 5 个级别,如表 3 所示。

表 3 多分量地震监测预测系统 AETA 五种日志级别的含义

日志级别	日志级别描述	系统状态
0	紧急	系统不可用
1	错误	错误消息
2	警告	警告消息
3	通知	普通日志信息
4	调试	调试信息

对于日志级别为 0-1 的日志内容,日志一旦产生,数据处理应用程序将通过管道机制将日志信息直接传递给终端日志采集器。终端日志采集器接收到级别为 0-1 的日志,将日志按照约定的规范处理好后直

接发往服务器。系统在接收到此类日志后,通过短信、邮件等方式将信息传递给运维人员,同时用红色大字传递到运维人员办公电脑,并发出警告声音提醒运维人员进行处理;对于 2-3 级的日志数据则不保证日志的实时性。数据处理应用程序在产生 2-3 级的日志后,先将日志数据记录到持久化存储设备中,然后由终端日志采集器周期性地从持久化存储设备中读取日志数据进行合并后再发往服务器,系统在接收到此类日志后直接录入数据库,不做报警提示。图 4 是持久化存储设备中的日志文件目录结构,按天创建日志文件夹,文件夹内文件名按序列号从 001 开始命名,每小时加 1,每天最多 24,每个小时内的日志写入同一个文件内。对于每一个日志文件都有相对应的 index 文件,详细记录了每条日志数据的地址、长度和读取标志,用来描述日志文件中日志数据的读取位置,是否已读等信息;对于第 4 级日志,只有在软件调试、问题定位的时候才会打开,设备正常运行时关闭,软件将不会产生级别为 4 的日志,从而节省存储空间和处理时间。

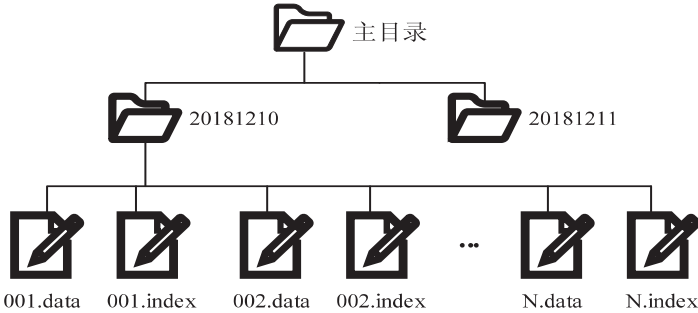


图 4 持久化存储目录结构

3.2 日志数据汇集层

日志数据汇聚层部署在阿里云服务器,既要完成系统业务数据的汇集,也要完成日志数据的汇集,其中系统业务数据又包括电磁数据、地声数据、数据处理终端状态数据。如图 2 所示,日志数据汇聚层主要有任务管理功能和数据分拣功能。

任务管理功能又分三个子功能:任务分拣功能、任务调度功能和任务管理功能。任务分拣功能定期访问持久化存储获取新任务,解析任务为子任务或者命令,然后将任务或者命令分类(可按照业务分类或者其他)。任务调度功能接收任务分拣中心解析分类后的任务或者命令,根据系统的资源或者设备状态调度任务,并将调度后的任务输入任务队列。任务管理功能响应数据处理终端的任务请求,为数据处理终端返回需要完成的任务。

数据分拣功能主要完成系统业务数据和日志数据的接收、分类和入库。图 5 为数据分拣功能逻辑架构。数据分拣功能按照数据类型的不同分为不同的业

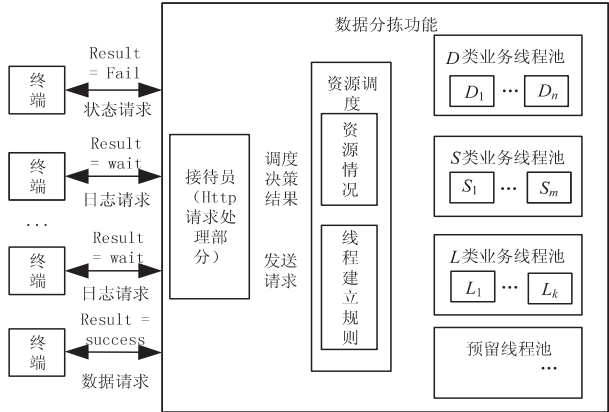


图 5 数据分拣功能逻辑架构

务类别,其中电磁和地声数据简称为 D 类业务,数据处理终端状态数据简称 S 类业务,日志数据简称 L 类业务。如图 5 所示,“接待员”模块用于将各种请求分类和判断请求是否符合约定。例如,数据请求中包含有 logfile 的请求被归类为日志数据,即 L 类业务;数据请求中包含 datapost 的请求被归类为电磁或者地声数据,即 D 类业务,对于格式和参数不符合约定的请

求直接返回“result = fail”。

数据分拣功能拟采用四种线程池来完成三种业务的处理,预留线程池是在各类业务线程池已经超过最大线程数的情况下,处理优先级别高的请求,其中各类线程池的线程数阈值以及业务类别的优先级可由运维人员配置。资源调度模块负责为“接待员”发送来的请求分配线程资源进行处理。以日志类业务为例,资源调度器会根据日志类线程总数 x_1 、日志类线程数阈值 n_1 判断是否在日志类线程池中建立线程处理此请求中的数据。如果 $x_1 < n_1$,则在日志类线程池中建立线程处理数据;如果 $x_1 \geq n_1$,则继续判断此请求的优先级别 a_1 是否可以进去预留线程池,如果 a_1 级别较低不能进去预留线程池,则为终端返回“result = wait”;如果预留线程池中的线程总数 x_4 小于预留线程池线程数阈值 n_4 ,则在预留线程池中建立线程处理数据;如果 $x_4 \geq n_4$,则表示资源已不足以支持处理新请求,则为数据处理终端返回“result = wait”。

数据分拣功能对业务进行分类并划分优先级,通过资源调度器分配线程池资源来处理请求,可以在很大程度上避免在大流量情况下服务器资源耗尽,服务宕机的现象。

3.3 日志数据存储层

日志数据汇集层对合并后的数据进行解析并放入入库队列。由于日志数据不会被高频访问,对实时性要求也不高,因此入库队列中的日志数据会存储到关系型数据库 MySQL 集群中。表 4 即为关系型数据库 MySQL 的数据库表设计。

表 4 MySQL 数据库表

字段名	数据类型	主键	可空	描述
TerminalID	smallint	是	否	终端编号
Time	int		否	日志时间戳
LogType	smallint		是	日志类型
Length	int		是	日志长度
Event	varchar		是	日志内容
TimeSave	timestamp		是	入库时间

在软件出现故障时,可以通过查询故障终端在出现故障时间范围内的日志数据来定位问题,对于运维人员来说工作量较小。但是,由于数据处理终端和传感探头通常部署在干扰较小的野外或山洞,电力和网络等环境是设备正常运行必要条件。因此运维工作除了排查故障外,还需要能够掌握布设在外的各台设备的运行情况。

对于经常断电重启、网络中断的区域,需要及时安排当地值班人员去现场检修电力和网络。面对大量的日志,运维人员如果一条条查看日志来判断电力或者

网络环境是否稳定,显然工作量巨大。因此,采用批处理技术,按日志类型、日志内容按天进行统计。运维人员可以通过查询日志数据统计表,快速了解到某台设备在某一天或某一段时间内的重启次数、断网次数以及各级别日志的数据量。对于地震高风险区域,日志数据统计表可以高效地了解该区域设备的运行环境,对于电力、网络或者设备不稳定的区域可以及时进行检修,保证震前数据采集的完整性。

3.4 日志数据展示层

日志数据展示层用于向用户直接展示日志数据、告警信息和日志统计信息等。由于日志数据存储层是分布在多个数据库上的,存储在不同的物理主机上,因此,如何实现对底层多数据源访问的支持,同时达到日志数据展示层用户对底层多数据源无感知的效果,是日志数据展示层设计的重要问题。

因此,将日志数据展示层划分为数据服务层和数据应用层。数据服务层将数据应用层和日志数据存储层隔离开来,借助数据服务层的代理功能解决上述多数据源访问和透明化的问题。数据应用层即日志数据展示网页。该网页的主要功能是使得运维人员在界面友好的网页端即可完成对日志数据、告警信息和日志统计信息的查询操作。

数据服务层的主体是 AETA 数据访问中间件。数据访问中间件是平台应用层和持久化存储层之间的中间代理,所有数据应用层数据请求均经过该中间件代理。

数据访问中间件主要包括配置信息服务 aeta_niddleware_AR 和数据接口服务 aeta_midleware_DS。配置信息服务提供用户验证、权限查询等接口,保证数据应用层应用的用户登陆、权限控制等基础功能;数据接口服务是日志数据存储层的对外数据接口,向用户提供统一的数据访问接口。该服务响应数据应用层的数据请求,将对日志数据存储层的操作结果返回给接口调用者。数据服务接口 aeta_midleware_DS 对外提供基于 HTTP 的访问接口,代理外部应用向数据库的所有操作请求。

该项目基于 Spring 框架开发,其带来的依赖注入机制实现了模块解耦,简化了开发流程;持久化框架选择了 MyBatis,该框架依靠 XML 或注解配置 SQL 语句,将 SQL 与程序代码剥离开来,能很大程度上简化应用对持久化存储层的访问过程。配置信息服务 aeta_niddleware_AR 的接口技术方案与数据接口服务 aeta_midleware_DS 类似。

数据应用层即日志数据展示网页。日志数据展示网页的总体逻辑结构自底向上,可以分为五层,分别是 DTO 层、DAO 层、Service 层和 Action 层以及表现层。

网站的 MVC 框架选择了稳定的 Struts2,未选用数据库持久化框架。DTO 层,指 Data Transfer Object,即数据传输对象层;DAO 层,指 Data Access Object,即数据获取对象层。

由于数据服务层已经将所有对日志数据存储层的操作封装,日志数据展示网站的数据操作均可通过数据服务层完成。因此 DAO 层不再负责与数据库的交互,仅负责与数据服务层交互的细节。Service 层,基于 DAO 层提供的数据库操作能力,为 Action 层提供必要的业务方法支撑。Action 层担任着 MVC 结构中的控制器角色,负责处理用户特定请求并将结果转发到不同的表现层组件。表现层,即 MVC 框架中的 View 视图,由服务器端的 Action 层生成,用户端接收并解析后即可在浏览器中展示。表现层的基本页面使用 JSP 技术编写,页面渲染基于 CSS 文件,页面交互采用了 JavaScript 技术。此外,页面上的部分组件使用到了 jQuery 和 Bootstrap 库。

4 结束语

文中设计并实现了一套应用于多分量地震监测预测系统 AETA 的日志系统,已稳定运行 1 年左右,能够高效地处理来自 250 个台站的日志数据,运维人员定位故障平均耗时 1 小时。实践表明,该系统能够有效地收集日志数据并进行展示,提高了运维人员的工作效率。

在未来的设计中,可以进一步优化数据库的设计,采用读写分离技术,降低数据库的压力,并使用非关系型数据库 redis 或者 Hbase 等,将关键日志存储在非关系型数据库中,提高日志数据查询的效率。

参考文献:

- [1] 黄 侠.“日志系统”在网络维护中的重要性[J]. 无线互联科技,2012(1):44-46.
- [2] PECCHIA A, RUSSO S. Detection of software failures through event logs; an experimental study[C]//IEEE 23rd international symposium on software reliability engineering. Dallas, TX, USA: IEEE, 2012: 31-40.
- [3] CINQUE M, COTRONEO D, PECCHIA A. Event logs for the analysis of software failures; a rule-based approach[J]. IEEE Transactions on Software Engineering, 2013, 39(6): 806-821.
- [4] 廖湘科, 李姗姗, 董 威, 等. 大规模软件系统日志研究综述[J]. 软件学报, 2016, 27(8): 1934-1947.
- [5] YUAN Ding, PARK S, ZHOU Yuanyuan. Characterizing logging practices in open-source software[C]//Proceedings of the 34th international conference on software engineering. Zurich, Switzerland: IEEE, 2012: 102-112.
- [6] YUAN Ding, PARK S, HUANG Peng, et al. Be conservative: enhancing failure diagnosis with proactive logging [C]//Proceedings of the 10th USENIX conference on operating systems design and implementation. Hollywood, CA, USA: USENIX Association, 2012: 293-306.
- [7] FU Qiang, ZHU Jieming, HU Wenlun, et al. Where do developers log? An empirical study on logging practices in industry[C]//Proceedings of the 36th international conference on software engineering. Hyderabad, India: ACM, 2014: 24-33.
- [8] 林 科, 王新安, 张 兴, 等. 一种适用于大地震临震预测的地声监测系统[J]. 华南地震, 2013, 33(4): 54-62.
- [9] 曾敬武, 雍珊珊, 郑文先, 等. 适用于大地震临震预测的地声传感单元[J]. 计算机技术与发展, 2015, 25(12): 133-137.
- [10] 曾敬武. 适用于地震临震监测的磁传感器设计及实现[D]. 北京: 北京大学, 2016.
- [11] 金秀如, 雍珊珊, 王新安, 等. 地震监测系统 AETA 的数据处理设计与实现[J]. 计算机技术与发展, 2018, 28(1): 45-50.
- [12] 庞瑞涛, 雍珊珊, 王新安, 等. 地震监测系统的电磁信号的采集设计与实现[J]. 计算机技术与发展, 2018, 28(2): 27-30.
- [13] 刘晨光, 王新安, 雍珊珊, 等. AETA 多分量地震监测系统的数据存储与安全系统[J]. 计算机技术与发展, 2018, 28(12): 7-12.
- [14] 王新安, 雍珊珊, 徐伯星, 等. 多分量地震监测系统 AETA 的研究与实现[J]. 北京大学学报: 自然科学版, 2018, 54(3): 487-494.
- [15] 徐萌飞, 王军玲. 控制系统软件的日志功能设计[J]. 船电技术, 2011, 31(7): 8-12.