 DOI: https://doi.org/10.12345/gcjsygl.v5i17.8312

Design and Implementation of Log System Based on Docker and ELK

Xuehua Chen Jing Huang He Sun Yonggang Wang

Beijing Institute of Remote Sensing Information, Beijing, 100011, China

Abstract

According to the problems of low efficiency of log retrieval and analysis, difficult fault positioning and unclear abnormal rules in business applications, this paper designs a high available log system scheme based on Docker and ELK, constructs a platform of log collection, storage and display, and realizes the centralized, and rapid retrieval and analysis of application system log. The test results show that the log system designed in this paper has high availability and scalability, with excellent performance in both massive log loading and retrieval efficiency, and has reference significance for similar search platforms.

Keywords

Docker; ELK; log; retrieval

基于 Docker 和 ELK 的日志系统的设计与实现

陈雪华 黄静 孙赫 王永刚

北京市遥感信息研究所，中国·北京 100011

摘要

针对业务应用中日志检索分析效率低、故障定位难、异常规律掌握不清的问题，论文设计了一种基于Docker和ELK的高可用日志系统方案，构建了日志采集、存储和展示平台，实现了应用系统日志的集中快速检索分析。测试结果表明，论文设计的日志系统具有高可用性和可扩展性，在海量日志的载入和检索效率两个方面都具有出色的性能，对于类似的搜索平台有着借鉴意义。

关键词

Docker；ELK；日志；检索

1 引言

随着计算机信息系统的快速发展，各类应用层出不穷，系统由单机应用一步步演变为大型分布式应用。随着用户和访问量的逐渐增加，业务逻辑更加复杂，系统也会迎来各类问题。日志是系统故障诊断的主要信息来源，由于海量日志可能分散在系统中不同位置，传统的日志采集和分析模式弊端凸显，经常导致系统软件异常发现不及时、诊断效率低、定位难度大等问题，日志的检索、统计缺乏行之有效的手段，无法通过海量日志信息总结故障规律和跟踪软件生命周期。

针对上述问题，论文基于 Docker 和 ELK（Elasticsearch + Logstash+Kibana）构建分布式日志分析平台，自动汇集日志记录程序运行时的动态信息，根据用户需求定制日志展示格式和图形面板，为用户精准分析异常、探测异常规律和制定系统维护计划提供有效支撑，不断提升系统运行的稳定性和可靠性。

【作者简介】陈雪华（1982–），男，中国河南商丘人，硕士，高级工程师，从事计算机科学与技术研究。

2 概述

2.1 ELK

ELK 是 Elasticsearch（弹性搜索引擎）、Logstash（日志采集解析工具）、Kibana（可视化展现平台）三个开源软件的组合 [1]。在实时查询和大数据分析等场景中经常配合使用。ElasticSearch（称 ES）是一个支持分布式、多用户、Restful 设计的开源搜索工具 [2]，基于 Lucene 开发，具备稳定、可靠、实时搜索等优势，在诸如 GitHub 等大型应用中得到了充分的验证。Logstash[3] 是一个完全开源、基于 Ruby 的应用工具，具备日志汇集、分析、过滤，并将其存储至 Elasticsearch 中，其提供了多种管道工具，满足用户日志搜集、按用户需求过滤信息等功能。Kibana 具备分析、查询、统计和可视化功能，为用户提供友好的 Web 页面，满足大多数用户的实时搜索和展示功能。一般生产环境中采用 Logstash 进行管道过滤，利用更轻量级的 Filebeat 进行日志采集，Filebeat 用于在没有安装 java 的服务器上专门收集日志，并将日志转发到 logstash、

elasticsearch 或 redis 等场景中进行下一步处理。

2.2 Docker

Docker 是一个轻量级虚拟化技术，以容器为单位进行隔离和调度，具备安装和使用方式简易、服务集成自动化、启动速度快速和持续开发部署等特点，适合构建和运行分布式平台，提供了简单易用的跨平台、可移植的容器解决方案，这些优势是传统虚拟机无可比拟的[4]。Docker 包含 Docker Client、Docker Daemon、Docker register、Drive、libcontainer 和 Docker container 等模块，支撑整个容器的构建运行等全生命周期应用[5]。

3 系统总体设计

3.1 应用系统日志规范设计

为了更好地对日志进行检索分析，系统必须建立一个普遍性的日志标准。论文结合应用系统实际，主要从日志文件名定义和内容格式两个方面进行约束。

3.1.1 日志文件名定义

日志文件名按照应用程序的名称（如 topic、算法名等）定义，扩展名为 log。当日的日志为 xxx.log，往日的日志按照 xxx.log_yyyymmdd（年、月、日）归类。应用程序日志分目录保存至指定位置，部分日志目录结构如表 1 所示。

表 1 部分日志目录结构

一级目录	二级目录	三级目录	四级目录下内容
子系统名称	调度名称	调度模块名称	XXpolicy.log_yyyymmdd
	算法	算法名称	XX 算法 .log_yyyymmdd
接口日志	接口 topic 名称	接口名称	接口名称 .log_yyyymmdd

3.1.2 日志内容格式规范

根据日志的重要性或严重程度划分等级，只有合理定义日志级别，才能避免日志混乱。日志要素有日志级别、日志时间和日志内容等，且日志要素根据需求适当裁剪。日志时间[必选]：规定的格式是yyyy-MM-dd HH:mm:ss.SSS；日志级别[必选]：根据日志的重要性或严重程度划分为DEBUG、INFO、WARN、ERROR；线程名称[可选]：在分布式应用或Web应用程序中，输出线程名称可以区分一次具体的请求上下文；业务标识[可选]：用来区分日志属于哪部分业务；记录器名称[可选]：声明日志记录器实例的类名；日志内容[必选]：根据不同的日志等级，在日志内容上会有不同的侧重点。异常堆栈[可选]：堆栈异常信息有助于程序异常的排查定位，但这部分信息的记录输出；产生行数[可选]：产生日志的源代码行，该记录对程序性能有比较大的影响。

除此之外，各业务应用软件还可以根据需要增加用户自定义附加信息，如主机 IP、主机名（可选）、应用名（可选）、服务名、请求 IP、请求来源、用户 ID、用户名（可选，用于方便查询）和请求 ID 等。

3.2 实时日志系统架构设计

在日志规范设计的基础上，ELK 日志系统由 Filebeat、Redis、Logstash、elasticsearch 和 kibana 组成，其中为了应对高可用和高并发需求，elasticsearch 采用集群部署，如图 1 所示。

Filebeat 通过一个或多个 prospectors（探测器）检测指定应用程序的日志目录，对探测到的每一个日志文件启动 Harvester（收割进程），每个收割进程读取日志文件的新内容，并发送至 Spooler（处理程序），由处理程序完成内容的汇集发送到 Redis 中间件。

Logstash 读取 Redis 中内容，通过既定的过滤过则格

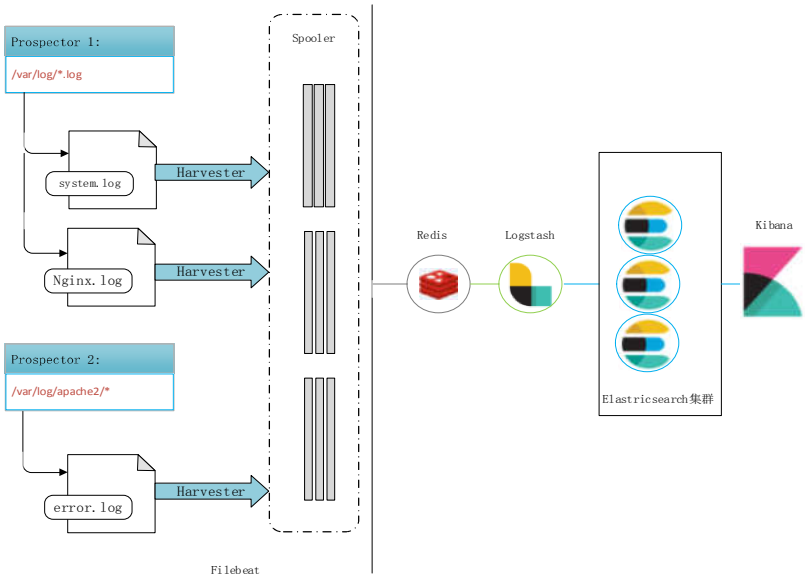

图 1 集群部署图

式化日志信息，发送至 elasticsearch 集群存储。此外，通过 Kibana 对接 elasticsearch 获取日志信息，以 Web 界面方式提供实时检索、分析和图形化展示等应用。

4 日志系统应用设计

4.1 elasticsearch 集群部署

采用 Docker 技术对 elasticsearch 进行集群部署，主要有以下优势。一是消除单点故障，提高系统可用度；二是采用分布式架构，提高系统吞吐性能，提升用户体验；三是快速部署持续滚动更新，采用 Docker 方式能够根据压力负载情况，一键式扩容服务实例副本，便于应对服务高峰等情况。

论文采用 docker-compose 部署由 3 节点组成的 elasticsearch 集群，通过在容器中注入环境变量方式，实现 elasticsearch 各节点的配置。包括节点名称、集群名称、节点发现地址、端口号以及最小集群数量等。分别在三个服务器中启动容器，即完成集群部署，运行状态如图 2 所示。其中，es01、es02、es03 为 elasticsearch 节点名称，es02 为该时刻 master 节点，其余两个为从节点，当 master 节点异常时，通过投票方式由其中一个从节点接替 master 节点职能。

4.2 日志搜集和过滤入库

4.2.1 日志搜集

论文采用Filebeat对分散在各服务器中的日志进行采集，为了降低与过滤、入库环节的耦合度，采用异步方式将采集到的信息录入 redis。根据业务系统日志特点，对 Filebeat 进行相应配置，输入为该服务器部署的应用程序日志路径，输出为 redis 队列。有时收集的信息可能包含跨越多行，如 Java 堆栈跟踪日志等，堆栈跟踪的每一行在 elasticsearch 被当作一个独立的文档，从而与上下文脱离了共同的事件，这使得后续在堆栈跟踪中搜索和理解异常变得很困难，因此需要通过 Filebeat 的 multiline 功能进行处理以避免此问题。

4.2.2 日志过滤入库

论文采用 Logstash 对日志进行过滤并格式化，根据设计 Logstash 输入为 redis 队列，从上图 redis 缓存 db0 队列的 key=redis-fb 里消费日志记录，输出至 elasticsearch。论文根据需求做了三个方面的设计优化。

一是移除“_id”“_type”“_version”“_score”等系统默认且与应用日志无关字段，减少冗余信息，避免干扰后续检索。

二是根据日志信息的 tag，以日志名 + 日期建立不同的索引，便于后续分类检索分析。

三是抽取日志本身的时间戳替换 @timestamp 字段，该字段的值与日志生成时间不符，可能导致通过 Kibana 检索时，出现错误的结果。

4.3 Web 应用展示

日志采集进入 elasticsearch 后，Kibana 可通过特定条件对其进行搜索和分析展示。其中，日志统计展示样例如图 3 所示。

Kibana 提供了实时日志检索、统计分析和面板展示等功

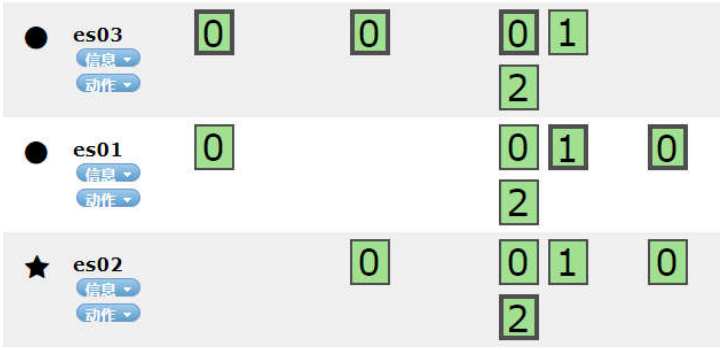

图 2 节点配置运行状态分布

图 3 日志统计展示样例

能。首先根据索引名称创建“索引模式”，然后进行后续操作。其 Discover 默认显示 elasticsearch 近期记录，通过搜索栏可通过 KQL 语句对索引中的字段进行查询和显示，可支持“AND”“OR”等精确查询条件，也可进行模糊查询，匹配到的关键词高亮显示；其 Dashboard 面板为用户提供了可视化定制服务，包括地图、折线图、饼图、柱状图、标签云图、热力图等多种图形，搜索到的数据能够可以灵活多维度的可视化展示。

通过对日志的长期跟踪，可对应用系统的健康状态进行统计，如图 4 所示，展示了一周时间内“error”日志数量的分布情况。以论文作者所维护的应用系统为例，发生大量错误日志的阶段均为系统升级的时间段，说明研发人员在代码上线前测试不充分，需要更加注重测试环节的管理。

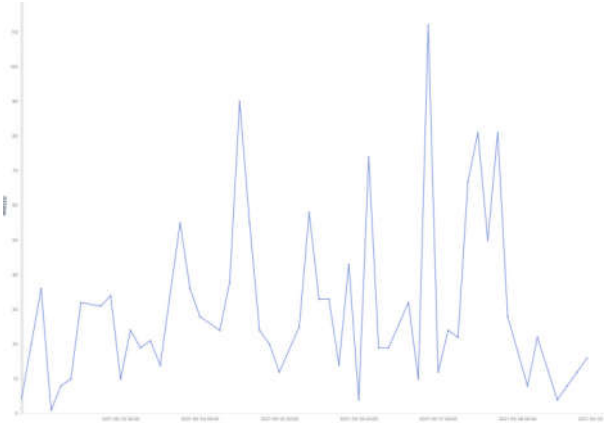

图 4 一周内“error”日志数量分布统计

图 5 为一年时间，应用系统各模块错误日志的占比，从图中可以看出 ftp 模块占比最高，达到 30% 以上，均为 ftp 服务不稳定所致，迫切需要对 ftp 服务的负载情况进行排查，以提高应用系统可用性。

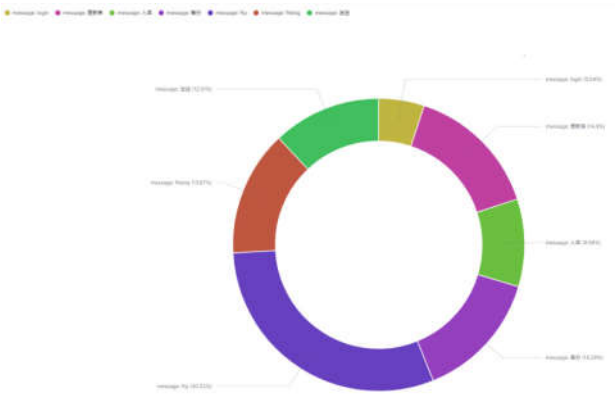

图 5 一年内应用系统各模块错误日志的占比

4.4 系统性能测试

论文采用了三台 Linux 服务器部署 ELK 平台，操作系统为 Centos7.5，内存为 256GB、CPU 为双路 Intel（R）Xeon（R）CPU E5-2667，硬盘为 SSD。

ELK 版本为 7.9.3，elasticsearch 采用了系统默认的 1 副本和 1 分片的模式，测试数据为应用系统 3 年共 28719363 条日志记录。采用 jmeter 压力测试工具分别从入库速率和检索时间（模糊查询）等指标对该日志系统进行测试，日志入库性能为 38732 条 /s，检索返回性能为 90.8ms，如表 2 所示。后续当日志记录海量增加时，可从集群、副本和分片数量上进一步优化系统性能。

表 2 jmeter 压力测试记录表

日志入库性能（条 /s)	38732
检索响应时间（ms）	90.8

5 结语

论文基于 ELK 对海量日志平台系统进行了设计与实现。同时，结合业务系统实际情况，规范了日志输出设计。利用 Docker 方式实现了 elasticsearch 集群的高可用性和可扩展性，能够充分应对未来系统日志急剧增加带来的挑战，解决了当前应用系统面临的日志检索分析效率低、故障定位难、异常规律掌握不清的问题，最后通过对日志入库性能和检索响应时间两个方面对该日志平台系统进行了测试。测试结果表明，elasticsearch 的载入和检索效率非常高，能够满足实时查询要求。

参考文献

[1] 饶琛琳.ELKstack权威指南[M].北京:北京机械工业出版社,2015.

[2] 姜康,冯钧,唐志贤,等.基于Elastic Search的元数据搜索与共享平台[J].计算机与现代化,2015(2):117-121+126.

[3] 高凯.实战Elasticsearch、Logstash、Kibana:分布式大数据搜索与日志挖掘及可视化解决方案[M].北京:清华大学出版社,2015.

[4] 浙江大学SEL实验室.Docker容器与容器云第2版[M].北京:人民邮电出版社,2016.

[5] 孙洪亮.Docker源码分析[EB/OL].http://open.daocloud.io/tag/yuan-ma-fen-xi/,2014-09.