设计研究与应用

基于微服务架构的日志监控系统的设计与实现

张 振，刘俊艳

（北京汇通金财信息科技有限公司，北京 100053）

摘 要： 由于软件系统的规模日趋扩大和由此带来的复杂性，会产生大量的日志信息，这些日志需要存储以备查询和分析，而传统的关系数据库对日志存储、查询、分析的能力有限，因此，需要考虑一种大容量复杂场景的日志解决方案。本文介绍了一种基于微服务化架构的日志系统，在日志的收集、处理、存储、展示各个流程都使用微服务方式部署，支持动态扩容缩容、支持大规模日志的处理和存储，满足了复杂使用场景的日志需[1]。

关键词： 微服务；日志系统

中图分类号： TP311 **文献标识码：** A **DOI：** 10.3969/j.issn.1003-6970.2017.11.037

本文著录格式： 张振，刘俊艳. 基于微服务架构的日志监控系统的设计与实现[J]. 软件，2017，38（11）：196–201

Design and Implementation of Log Monitoring System Based on Microservice Architecture

ZHANG Zhen, LIU Jun-yan

(*Beijing huitong jincai information technology Co., Ltd., Beijing* 100053, *China*)

【Abstract】: Due to the size of the software system and widening of the complexity of the resulting will produce large amounts of log information, these logs need to be stored for query and analysis, and the traditional relational database log storage, query, analysis ability is limited, therefore, need to consider a large log solution of complex scenes.In this paper, a logging system based on micro structure of service, in log collection, processing, storage and display each process using micro service deployment, support for dynamic expansion shrinkage mass log processing and storage capacity, support, to satisfy the complex usage scenarios of log.

【Key words】: Micro-service; Log system

0 引言

随着科学技术的发展以及云计算、P2P 等技术的普及，全球数据量呈现爆炸式的增长，尤其是大数据时代的到来，通过互联网，用户制造了海量的数据日志信息。2017 年 1 月 22 日中国互联网络信息中心发布的《第 39 次中国互联网络发展状况统计报告》中指出：我国 2016 年全年共计新增网民 4299 万人，增长率为 6.2%，其中手机网民规模达 6.95 亿，占比达 95.1%，增长率连续 3 年超过 10%[2]。手机网民最常使用即时通信 APP：2016 年，网民在手机端最经常使用的 APP 应用前三位分别是微信、QQ、淘宝，无论是微信、QQ、微博等社交通信软件还是淘宝、京东等电商软件，各系统每天产生的海量日志信息都达到了指数级。而传统的关系数据库对日志存储、查询、分析的能力有限，已无法满足呈爆炸性增长的海量数据日志需求，为解决信息存储容量、数据安全、日志搜索分析等问题，基于微服务架构的日志监控系统应运而生，如今已得到广泛应用。使用日志监控系统能够提前对潜在的风险进行发掘，分析、判断并形成定性或定量的描述，从而采取应对措施来降低风险。这对提高信息通信系统的安全性、稳定性及其服务能力具有重要的理论价值和实际意义。系统采用多任务分布式技术对海量日志进行分析挖掘，应用规则关联、统计关联等分析方法，可以建立科学的分析模型，使得对日志的分析深度与事件的识别准确度得到进一步的提升。

作者简介： 张振(1986–)，男，本科，北京汇通金财信息科技有限公司，主要研究方向：互联网技术；刘俊艳(1978–)，女，硕士，北京汇通金财信息科技有限公司，主要研究方向：计算机应用。

本文阐述了基于微服务架构的日志处理方案设计，并结合近几年来微服务架构在日志监控系统的应用情况，对日志监控系统的概念、特点、体系架构进行研究，以提升日志管理规模和管理效率。

1　微服务架构

1.1　微服务介绍

微服务是构建分布式系统的架构风格，是指开发一个单个小型的但有业务功能的服务，每个服务都有自己的处理和轻量通讯机制，可以部署在单个或多个服务器上。微服务也指一种松耦合的、有一定的有界上下文的面向服务架构。也就是说，如果每个服务都要同时修改，那么它们就不是微服务，因为它们紧耦合在一起；如果你需要掌握一个服务太多的上下文场景使用条件，那么它就是一个有上下文边界的服务，这个定义来自 DDD 领域驱动设计[4-5]。

相对于单体架构和 SOA，它的主要特点是组件化、松耦合、自治、去中心化，体现在以下几个方面[4-5]：

1. 一组小的服务

服务粒度要小，而每个服务是针对一个单一职责的业务能力的封装，专注做好一件事情。

2. 独立部署运行和扩展

每个服务能够独立被部署并运行在一个进程内。这种运行和部署方式能够赋予系统灵活的代码组织方式和发布节奏，使得快速交付和应对变化成为可能。

3. 独立开发和演化

技术选型灵活，不受遗留系统技术约束。合适的业务问题选择合适的技术可以独立演化。服务与服务之间采取与语言无关的 API 进行集成。相对单体架构，微服务架构是更面向业务创新的一种架构模式。

4. 独立团队和自治

团队对服务的整个生命周期负责，工作在独立的上下文中，自己决策自己治理，而不需要统一的指挥中心。团队和团队之间通过松散的社区部落进行衔接。

1.2　ELK 介绍

ELK 由 Elasticsearch、Logstash 和 Kibana 三部分组件组成；

ELK 具有如下几个特点：

（1）处理方式灵活：Elasticsearch 是实时全文索引，不需要像 storm 那样预先编程才能使用[3]；

（2）集群线性扩展：不管是 Elasticsearch 集群还是 Logstash 集群都是可以线性扩展的[3]；

（3）前端操作炫丽：Kibana 界面上，只需要点击鼠标，就可以完成搜索、聚合功能，生成炫丽的仪表板[3]；

（4）检索性能高效：虽然每次查询都是实时计算，但是优秀的设计和实现基本可以达到全天数据查询的秒级响应[3]；

（5）配置简易上手：Elasticsearch 全部采用 JSON 接口，Logstash 是 Ruby DSL 设计，都是目前业界最通用的配置语法设计[3]；

在本日志系统中，将 Elasticsearch、Logstash、Kibana 开源套件作为日志系统架构组件，并且在此基础上进行了一系列的封装开发，使之适用于本日志系统的架构模式。

1.2.1　Elasticsearch

Elasticsearch 是一个基于 Lucene 的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于 RESTful web 接口。Elasticsearch 是用 Java 开发的，并作为 Apache 许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便，零配置，支持集群（Cluster，见图 1）自动发现，索引自动分片、索引副本机制（见图 2），多数据源，自动搜索负载等特点[7-10]。

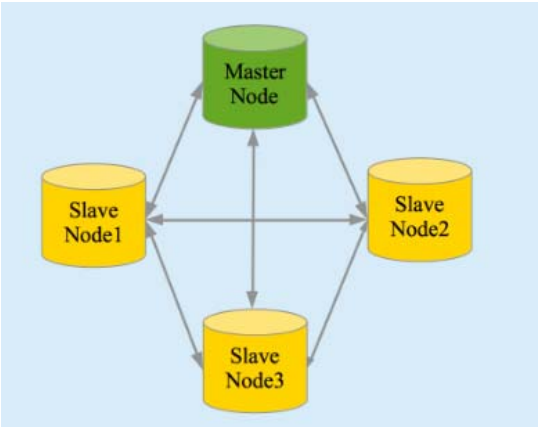

图 1　集群
Fig.1　Cluster

（1）节点（Node）和集群（Cluster）

节点是集群中的一个 Elasticeach 实例。集群是一组拥有共同的 cluster name 的节点。其中一个

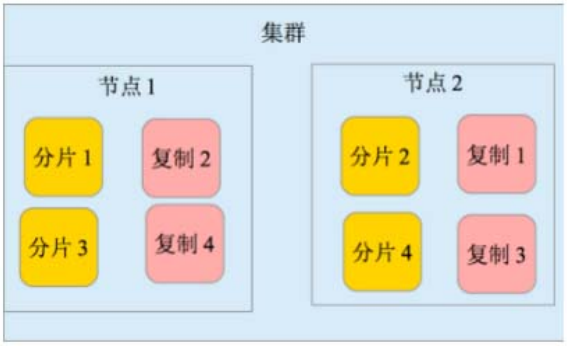

图 2　分片、复制
Fig.2　Fragmentation, replication

节点就是一个 ES 进程，多个节点组成一个集群。一般每个节点都运行在不同的操作系统上，配置好集群相关参数后 ES 会自动组成集群。集群内部通过 ES 的选主算法选出主节点，而集群外部则是可以通过任何节点进行操作，无主从节点之分，对外表现对等，去中心化，有利于客户端编程。

（2）索引 Index

索引在 ES 中索引有两层含义，作为动词，它指的是把一个文档保存到 ES 中的过程，索引一个文档后，我们就可以使用 ES 搜索到这个文档；作为名词，它是指保存文档的地方，相当于关系数据库中的 database 概念，一个集群中可以包含多个索引。

（3）分片 shard

ES 是一个分布式系统，它保存索引时会选择适合的“主分片”（Primary Shard），把索引保存到其中（我们可以把分片理解为一块物理存储区域）。分片的分法是固定的，而且是安装时候就必须要决定好的（默认是 5），后面就不能改变了。

既然有主分片，那肯定是有“从”分片的，在 ES 里称之为“副本分片”（Replica Shard）。副本分片主要有两个作用：高可用：某分片节点挂了的话可走其他副本分片节点，节点恢复后上面的分片数据可通过其他节点恢复负载均衡：ES 会自动根据负载情况控制搜索路由，副本分片可以将负载均摊。

（4）RESTful 支持

ES 支持 RESTful 访问，并且 ES 的 HTTP 接口不只是可以进行业务操作（索引/搜索），还可以进行一些配置等。

（5）document 文档

一个文档就是一个保存在 es 中的 JSON 文本，可以把它理解为关系型数据库表中的一行。每个文档都是保存在索引中的，拥有一种类型和 id。一个文档是一个 JSON 对象（一些语言中的 hash / hashmap / associative array）包含了 0 或多个字段（键值对）。原始的 JSON 文本在索引后将被保存在 _source 字段里，搜索完成后返回值中默认是包含该字段的。

（6）id

Id 是用于标识文档的，一个文档的索引/类型/id 必须是唯一的。文档 id 是自动生成的（如果不指定）。

（7）field 字段

一个文档包含了若干字段，或称之为键值对。字段的值可以是简单（标量）值（例如字符串、整型、日期），也可以是嵌套结构，例如数组或对象。一个字段类似于关系型数据库表中的一列。每个字段的映射都有一个字段类型（不要和文档类型搞混了），它描述了这个字段可以保存的值类型，例如整型、字符串、对象。映射还可以让我们定义一个字段的值如何进行分析。

（8）mapping 映射

一个映射类似于关系型数据库中的模式定义。每个索引都存在一个映射，它定义了该索引中的每一种类型，以及索引相关的配置。映射可以显示定义，或者在文档被索引时自动创建。

1.2.2　Logstash

Logstash 使用 Jruby 语言编写，对于使用者来讲，Logstash 本身是基于命令行界面，面向任务处理的。Logstash 的软件架构是一种带有“管道-过滤器”风格的插件式架构，作为一个开源软件，Logstash 遵循 Apache 2.0 进行开源，第三方社区为其贡献了大量插件，它可以实现数据传输，格式处理，格式化输出，还有强大的插件功能它可以对日志进行收集、分析，并将其存储供以后使用。主要特点：几乎可以访问任何数据、可以和多种外部应用结合、支持弹性扩展[7-10]。

- 输入阶段：接受不同来源的数据流入，可以配置 codec 插件进行简单的处理。
- 过滤阶段：对流入数据进行过滤等操作，传递给 output，其中“输入”与“输出”是必须有的，“过滤”阶段是可选的。
- 输出阶段：将数据传递到消息队列，文件系统等进行进一步处理，在 ELK 的日志系统中，输出到 ElasticSearch 索引中。

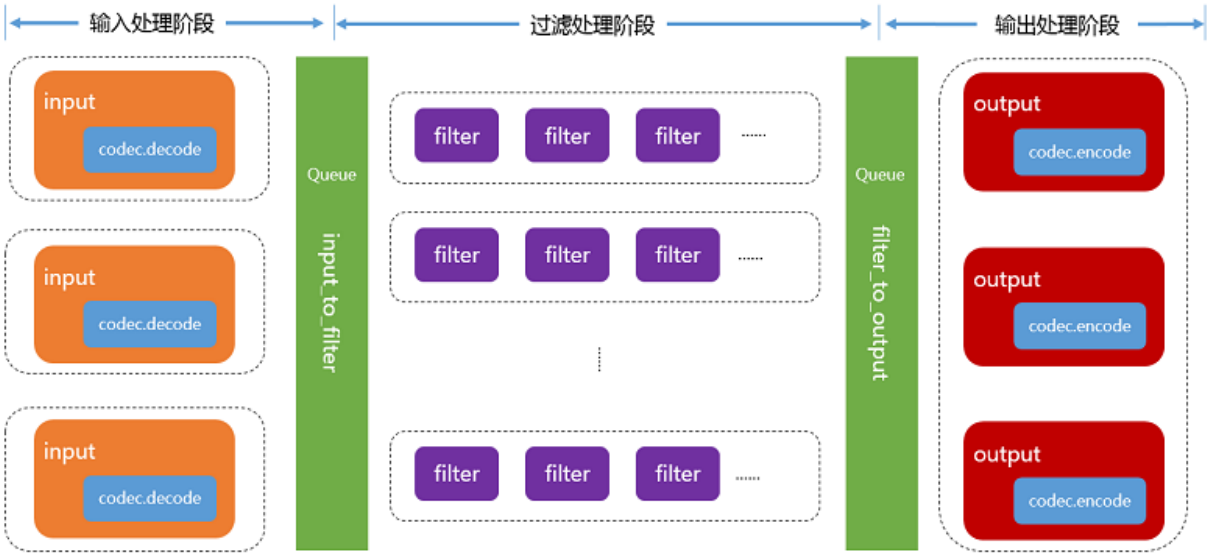

图 3　Logstash 业务流程图
Fig.3　Logstash business flowchart

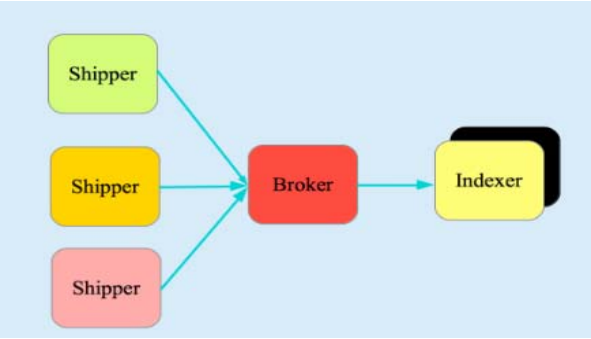

图 4　Logstash 构成图
Fig.4　Logstash constitute a figure

三个阶段的处理任务是异步的，不存在跨阶段任务执行与同一个线程中的情况。

logstash 由三个主要部分组成（见图 4）:

（1）Shipper－发送日志数据；

（2）Broker－收集数据，缺省内置 Redis；

（3）Indexer－数据写入；

1.2.3　Kibana

Kibana 是一款基于 Apache 开源协议，使用 JavaScript 语言编写，为 Elasticsearch 提供分析和可视化的 Web 平台。通过 Kibana 来查询、浏览并且可以与存储在 Elasticsearch 索引中的数据交互。还可以很容易的以各种各样的图表，表格和地图样式来针对数据执行高级的数据分析以及可视化。Kibana 使得我们可以很容易的了解海量数据。它非常简单，基于浏览器的界面使得您可以快速的创建并且实时显示修改的 Elasticsearch 查询的仪表盘，而且 Kibana 的配置非常简单，在几分钟内就可以成功安装 Kibana 并查询 Elasticsearch，不需要任何额外的基础设施[7-10]。

2　日志监控系统的设计与实现

2.1　requestID 生成及传递

各个应用统一配置请求拦截器，当拦截到 http 请求时，拦截器首先生成 RequestID，接着收集应用的调用信息及一些用户信息并加密输出到日志中。

RequestID 的生成是为各个应用日志间形成调用链，每次请求都会生成唯一的 RequestID，在日志量非常庞大的情况下，带有 RequestID 的日志在排查故障，查询用户每次请求所产生的日志非常方便、快捷。

2.2　日志采集

日志采集阶段会为每个应用服务器上都部署一个日志采集模块，这个日志采集模块的作用就是采集各服务器上的应用日志，logstash shipper 根据配置将各应用日志采集到 redis 集群中，logstash shipper 会为各个应用定义唯一 type 用来以后生成索引用。

2.3　日志缓存

redis 作为日志转储组件可以有效提高系统可用性，使用集群或者主备结构代替单实例，可以有效提高组件的可用性。在 redis 中，数据统一缓存在

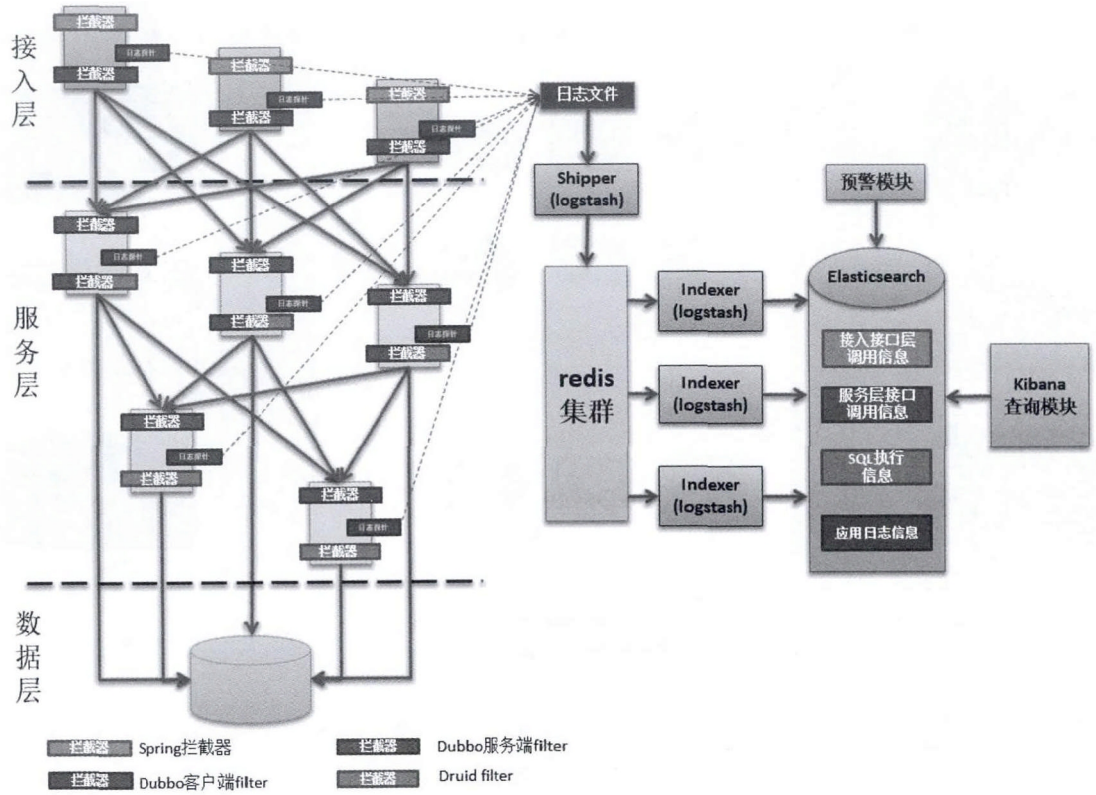

图 5 日志系统架构图

Fig.5 Log system architecture diagrams

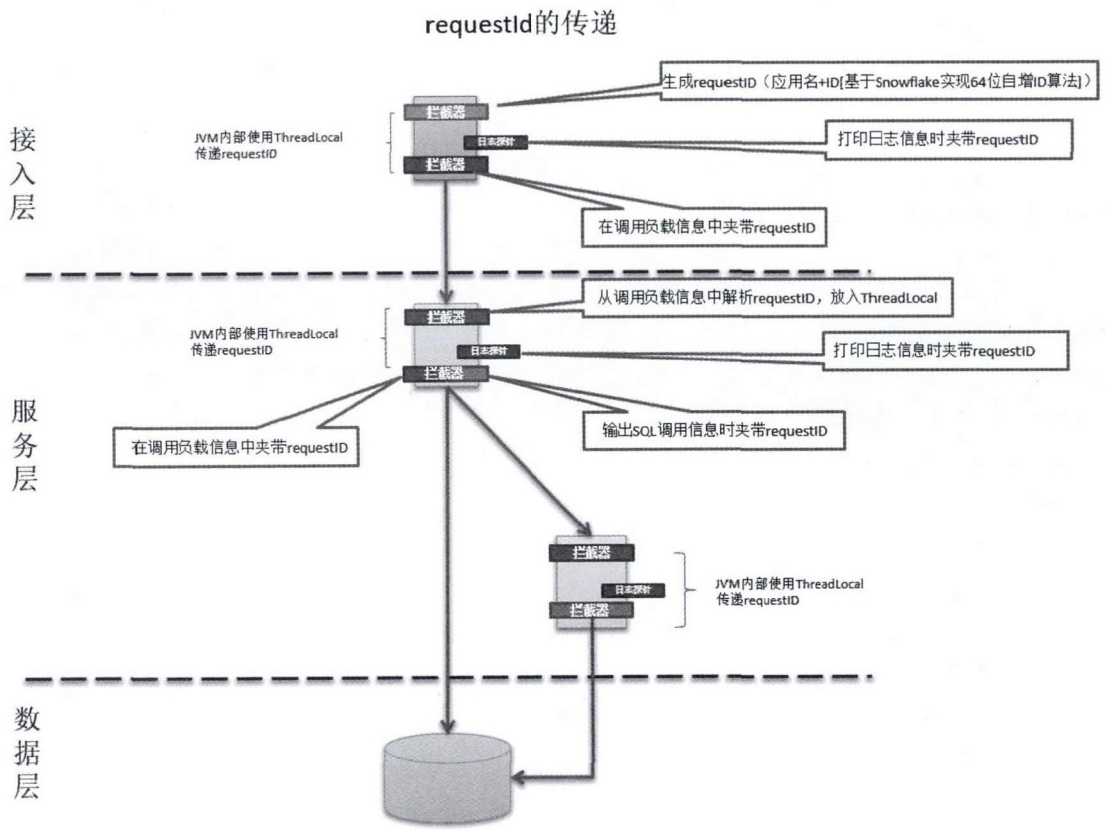

图 6 RequestId 传递图

Fig.6 RequestId passing figure

key 为 logstash（key 可自定义）的名字中，采用 list 数据类型作为日志采集的一个缓存区队列，list 类型是按照插入顺序排序的字符串链表，这意味着无论数据量多么巨大，头尾插入或删除数据的速度非常快，利用 List 的 Push/Pop 操作即实现了消息队列。

2.4　日志处理

日志处理阶段是 logstash indexer 端从 redis 中读取日志数据，并进行一系列的过滤处理然后存储到到 ElasticSearch 中。其处理过程中要进行数据格式的转换，比如抽取日期然后转换为预定义的格式、抽取日志级别、RequestId 等字段，抽取 message 字段信息，过滤字段信息根据条件进行删除或合并等。还有通过 GeoIP 获取 ip 所在地，根据系统定制过滤条件，正则解析、Json 解析等；从日志信息中抓取关键字，根据 type 生成 ElasticSearch 的索引，并将日志信息写入到 ElasticSearch 的 index 中。

2.5　日志存储

日志存储在 ElasticSearch 中，并且 ElasticSearch 也是使用集群方式来进行部署，通过设置 elasticsearch.yml 的 cluster.name 来定义集群名称，并且多个 elasticsearch 集群名要完全一样，然后设置 node.name 来区别每个 elasticsearch。ElasticSearch 集群节点分为两种类型：node.master、node.data。[6][11][12]

node.master：当设置为 true 时，当前节点就成为了集群的管理节点（即主节点），主要功能是维护元数据，管理集群各个节点的状态[6,11-12]。

node.data：当设置为 true 时，当前节点就成了数据节点，主要负责数据的存储、查询和导入[6,11-12]。

2.6　日志展示

本日志系统的界面使用了 kibana 开源框架，为 Elasticsearch 提供分析和可视化的 Web 平台。它可以在 Elasticsearch 的索引中查找，交互数据，并生成各种维度的表图。展示出了日志的查询结果、日志的数量变化趋势图、日志生成的各种图表视图、分析仪表盘等。

3　结论

本日志系统与常规 elk 部署的日志系统不同，设计中为各个应用统一配置了请求拦截器，用来生成唯一的 RequestID 和收集请求信息等，通过 RequestID 能查询出整个请求的调用链日志详情，对于排查异常、快速定位非常方便快捷。并且在日志收集、缓冲、处理、存储等各个阶段均采用了分布式、微服务化的部署方式，使日志处理的全流程均可根据实际使用情况，进行动态弹缩，有效地利用了物理资源，并能够应对大规模的日志情况，由于采用了基于 lucene 的倒排索引方式的日志存储，提高了查询和统计效率，在实际项目中使用效果良好[1]。

参考文献

[1] 参考网FX361.COM. 基于微服务架构的日志系统. http://www.fx361.com/page/2017/0315/1185754.shtml.

[2] 杜振南，朱崇军. 分布式文件系统综述[J]. 软件工程与应用, 2017, 6(2): 21-27.

[3] 扣bubuki.com. ELK-实用日志分析系统. http://www.bubuko.com/infodetail-2025984.html.

[4] PetterLiu. 博客园. 微服务架构设计. http://www.cnblogs.com/wintersun/p/6219259.html.

[5] 纽曼(Sam Newman). 微服务设计. 软件工程及软件方法学, 2016-04-01.

[6] 饶琛琳. ELK Stack权威指南. 2017.

[7] Saurabh Chhajed(苏库拉·塞哈特). Learning ELK Stack. 2016.

[8] 徐刚. IBMdeveloperWords. 集中式日志系统ELK协议栈详解. https://www.ibm.com/developerworks/cn/opensource/os-cn-elk/.

[9] CSDN博客. 漫谈ELK在大数据运维中的应用. http://www.cnblogs.com/wintersun/p/6219259.html.

[10] 郑彦生. 51CTO博客. ELK日志分析系统. http://467754239.blog.51cto.com/4878013/1700828/.

[11] elastic. 官网. https://www.elastic.co/guide/index.html.

[12] 饶琛琳. GitBook. ELKstack 中文指南. https://www.gitbook.com/book/chenryn/elk-stack-guide-cn/details.