**日志分析系统需求分析**

项目背景

在早期的服务器应用故障分析的时候，可能我们直接进入服务器查询相关的日志，定位异常。但是随着服务化与业务的不断增长，部署的机器越来越多。甚至是管理权限中规定不得登陆生产服务器进行任意操作。又或者是由于数据滚动的问题，磁盘容量问题导致日志删除或压缩。所以，我们需要做一个搜集到服务器上的所有日志，并且能够进行统一检索的系统。甚至能够直接基于可视化的方式对数据进行展现。对系统的整体运行情况进行报表类的分析。

所以我们需要一套日志系统对相关数据进行采集，存储，索引，可视化等。

日志是解决软件在运行过程中出现问题的重要依据,在企业日常的业务平台上会产生大量的日志数据,管理人员如果以传统的方式在多节点上采集、传输、分析和存储相关日志数据,那么将会是繁琐耗时,降低工作效率的方式,如何解决自动采集、传输、分析和存储这些日志数据,以管理TB级的日志数据为目的,基于Docker技术,设计并实现了具备高可用性,高可靠性和可扩展性的日志平台系统。针对分布式系统上的多节点日志数据自动采集、采集过程中产生与消费不同步所造成的系统瓶颈压力、分布式环境下应用服务器集群自动部署和扩展、日志数据查询、统计和分析等技术问题,对日志平台系统进行了需求分析,根据需求分析,设计了客户端-服务器模型的日志平台的总体结构、工作流程。其软件框架采用分层结构。提出了解决日志平台系统的日志采集、聚合、解析索引、检索与存储功能模块的设计方案。针对日志平台系统的分布式环境下的部署和扩展,提出了以Docker容器技术的解决方法。基于日志平台整体架构,对日志平台进行了详细设计,确定了各功能模块的工作流程,并且通过关键技术编程,满足了各功能模块的设计需求。实现了日志数据的采集、聚合、解析索引、检索与存储等功能。体现了企业提高日志管理水平的业务思想。对Docker容器部署策略和其扩展性进行了阐述。完成了对日志平台系统瓶颈进行传输日志数据压力测试,验证了系统的高可用性、高可靠性。本平台设计并实现了复杂业务和部署环境下的日志采集、聚合,解析索引、存储与检索功能,提供了外部第三方分析平台的接口,帮助用户进行线上业务的准实时监控、业务异常定位、业务趋势分析、安全与合规审计,通过Docker的部署方式有效的支撑了后续的水平扩展,为深度挖掘日志的大数据价值提供了基础平台。

近年来，随着计算机技术的快速发展，迎来互联网企业的一个高速发展期。互联网

的应用场景和实际应用的功能要求越来越多，为了快速满足用户的功能需求和性能与容

量的发展要求，各个 Web 网站服务无论在应用模块的数量还是整个系统的复杂度都在

不断增加，各个组成部分之间的耦合性也越来越强。现代 Web 网站的大型分布式系统对

于运维人员来说，意味着需要维护大规模的服务器、存储和实时监控海量的业务日志，

处理大量的系统监控数据，同时需要有快捷的分析能力和故障响应能力[

1]。

Web 网站的日志管理面向的是各个层级和集群的各类日志[

2-3]。在互联网领域的企

业，由于大量采用了开源软件和分布式组件，一个大型网站由数百个甚至数以万计的分

布式服务模块组成，而每个模块又由几台到几百台以上的服务器形成一个个集群提供服

务。当这样的分布式系统有异常产生时，错误的来源可能来自于某几个应用模块，也可

能来自于某个模块的个别服务器，如果仅从分散存储的日志跟踪问题会变得十分困难。

另外，在大型互联网的架构设计中一般是采用分层的结构，例如用户的一个请求将先通

过网站的外层负载均衡性，按功能要求分发给后端的服务器，然后由后端的服务器再向

各个服务模块或数据库请求数据，各个模块在内层的相互访问将经过内层的负载均衡或

网关进行，这样一个用户的请求需要经过多层才到达目标服务器，需要从各个功能模块

的整体来看待日志。因此，如何将这些负载均衡日志和应用日志等统一采集，统一解析

和处理，把日志实时地可视化地呈现出来并协助运维人员快速地识别到系统问题，是现

代互联网企业的运维人员面临的巨大的挑战。

本课题拟通过对互联网企业的日志管理需求，探索 Web 网站可视化实时日志系统

的构建，搭建一套实时分布式日志分析系统，并以 Web 网站运维中作为流量入口的负载

均衡和应用日志为目标分析对象，提供一套完整的日志收集、存储、分析和监控服务的

日志分析系统。

商云服务需要跟踪设备请求和用户请求,这些请求日志往往分布在不同服务中,需要在大量日志中根据设备mac地址和用户ID来查找服务调用链路,便于找到问题.

随着互联网数据规模的急剧膨胀，以及服务类型的复杂多样，各类企业的运营业务

逐渐由单机的系统转为分布式应用软件系统来承载。与此同时，系统运维的复杂度也呈

指数级上升，大量分散的日志不易管理，人工检索日志中的关键异常信息效率太低，并

且遗漏的可能性较大，导致大量有价值的日志信息没有被充分利用。本文正是针对当前

分布式应用场景下的运维痛点，设计并实现了一个适用于分布式应用软件的日志分析系

统。从分散日志的统一收集、存储和日志分析以及分析结果的可视化展示，将日志分析

系统分为日志聚合和日志分析两大模块。

功能性需求

采集

存储

分析

展示

基于 ELK 的实时大数据日志搜索平台的需求分析大致可分为功能性需求分析与

非功能性需求分析。功能性需求分析在本文可体现在本文构建的大数据日志分析平台

是否能满足各个子模块的功能需求，在尽量满足性能需求的同时达到相应子模块的预

期效果。通过前两章关于传统大数据搜索平台的经典案例介绍以及相应的国内外现状

的分析。本文可将基于 ELK 的实时大数据日志搜索与分析平台分为如下 4 个子模块

进行具体分析，系统的整体功能需求图如图 3.1 所示。并且重点关注以下问题：（1）

日志采集需求。万事开头难，日志采集作为第一步，如何能够在尽量不消耗大量硬件

资源的条件下采集到较为规范的日志，且在持续集成的系统中能保证日志按照一定的

日志规范稳步增长，且保证日志数据的正确性。（2）日志索引与存储需求。面对海量

的日志信息，如何高效地利用大数据框架实现实时云计算并准确无误的存储到基于倒

排索引的 Elasticsearch 中，并支持用户在离线后对各种类型的日志诸如 mysql 日志、

Java 日志、Docker 日志等日志进行查阅与分析。（3）数据可视化需求，官方 kibana 网

页界面给出了许多解决方案以便分析大数据日志，但因日志格式与应用场景的不同，

如何通过自定义的数据可视化实现在不同日志信息的个性化可视化解决方案。以上即

为当前大数据日志分析系统中亟待解决的问题，也是本文所提出的日志系统中重点关

注的问题。由图 3.1 可知，本系统从功能性角度来分，可分为如下四个子模块，分别

为：日志采集模块、Kafka 消息发布与订阅模块、日志索引与存储模块以及日志分析

与可视化模块，下面分别介绍上述 4 个模块的功能

日志处理和分析需要解决日志的采集、存储和分析，从方式上可以分为离线日志分

析和实时日志分析两种，离线分析是把 Web 网站各类的日志统一收集起来，先放到中央

存储或者分布式存储中，然后再进行分析。而实时日志分析的要求则高很多，需要把不

断产生的增量日志进行收集并实时统计分析并反馈系统的现状和发出预警。

在收集日志的方法上，由于考虑到日志量巨大，如果从中央存储服务器集中对日志

进行拉取，对该中央存储服务器的网络带宽、IO 性能以及 CPU 都造成巨大的压力，当

成千上万的日志主机不断增加时，拉取的方式无法实现实时的处理。因此，目前主流的

日志收集都是在各个主机时安装一个代理(Agent)或客户端应用(Client)，通过在各个业务

主机上向消息队列或中央存储进行推送的方式进行收集，遇到某些不能安装客户端的设

备才使用拉取的方式。

随着业务量的增长，数据量也越来越大。如何处理这些日志成为运维人员关注的焦

点，因为这些日志对于互联网服务的安全、监控预警、故障处理和业务分析都是至关重

要的。我们希望通过采集各个应用层和服务层服务器的应用日志，为企业技术人员提供

一个实时的日志查询和分析系统，另外能够根据查询条件对日志进行可视化的展示，直

观地展示日志的关键信息。

在设计该日志分析系统时，本文将充分考虑互联网架构的需求，并根据业务需求去

选择合适的工具，核心问题是技术架构选型问题。我们需要面对的问题有:

 实时性。数据摄入的实时性指的是当业务日志数据发生变化时候，系统能接受

多少的延迟能看到这个数据。理想的情况是希望系统越实时越好，但是现实中

受到成本和技术的限制，我们一般分为实时系统（毫秒延迟）、近实时系统（秒

级延迟）、准实时系统（分钟级延迟）和离线系统（小时级或者天延迟）。一般

延迟时间和吞吐能力，和计算能力都是反比的，吞吐越强，计算越精确，延迟

时间会更长。另一方面，实时性也体现在查询的延迟上面，这个延迟计算的是，

用户发出查询请求之后，要等待多长时间，服务端能够返回计算结果。日志分

析对于实时性有一定的要求，至少是需要准实时性，能够在分钟级别上反映系

统的实际情况。

 海量的日志数据处理。量级是影响日志系统的一个重要因素，其本身也是一种

业务场景的衡量。对于小型的企业日志是 GB 级别，随着应用数量的急剧增长

到 TB 或者 PB 级别，对日志分析系统的实时性与性能也会带来挑战。面对海量

的数据，采用分布式的日志存储架构是互联网企业的第一选择。

 维护性与低侵入性。在满足业务需求和数据量需求的前提下，最好的选择是技

术架构最简单的，因为往往这种是最容易使用和维护的。使用成本与组件的复

杂度成正比，一般来说组件复杂度越高，组件数量越多，多种组件配合使用成

本会越高。对于日志分析会有各类需求，需要考虑不断加入的新增的主机和应

用、不同类型和格式的日志、规模变大时系统的扩容，以及对团队成员的技术

要求等。另外要求日志分析系统要对现有的应用侵入性小，不影响原来应用及

主机的性能和配置等。

 便利的日志的检索与可视化。当各类日志在收集和分析之后，需要满足各类人

员的检索和监控需求，包括测试人员和开发人员对各类应用日志的实时检索和

查询、运维人员对整个系统的各类指标的巡查以及把各类关键的指标实时地展

示在监控屏幕上以准确反映 Web 系统的整体状态。日志的检索与可视化需要可

灵活地制定，例如能够以指定字段信息自动分类统计，以不同时间粒度显示和

多种不同的形式进行展示。

 在日志收集的基础上实现智能监控。传统的监控着重以单机的运行，使用固定

的阈值进行监控和异常判断，过度依赖人工分析和诊断，已不能满足现代复杂

系统的需求。因此如何利用日志系统采用智能化的手段监控 Web 网站的整体运

行情况，使用统计的方法或机器学习的方法进行异常检测和故障判断是日志分

析系统建设后的主要探索方向。

本文的主要工作就是根据上面的需求并结合互联网企业本身的业务需求，设计能

满足业务需要和性能需求的软件架构。根据软件设计中需要实现不同的功能选择适当的

组件和实施方案，并阐述组件选择的原因和原理。根据整个日志分析系统的架构和实施

方案，给出具体的实现方式、配置和部分代码的实现，展示实现的效果。

非功能性需求

本文设计和实现的日志分析系统的优点在于，文本收集和抓包信息收集的性能都

很高，对应用主机本身具有很低的侵入性；日志分析的配置比较灵活，能够快速为新的

日志类型进行解析；具有较高的实时性和可扩展性，能够满足企业在不同规模上的日志

分析需求。该系统的缺点，在处理更大型的 Web 网站时存在不足，例如在架构上没有考

虑把数天或数周以前的数据存放在机械硬盘上，而较新的数据存储在 SSD 磁盘上，实现

冷热分离；在索引的处理上，对索引进行适当的合并优化，减少 Elasticsearch 的 segment

碎片也是规模变大时必须考虑的

非功能性需求是在正常功能能运行的情况的前提下，能够保证系统具有易

用性、可扩展性、可靠性、安全性、运行性能佳等方面的需求。使得系统更稳

定、更快速的运行。

可扩展性需求

性能需求

总结

ELK 具有以下优点：一是 ELK 具有分布式搜索的快速检索

功能，在海量的数据中检索速度能达到毫秒级，而且支持 DSL 的语法来进行搜索，

通过类似配置的语言，快速筛选数据；二是 ELK 拥有完美的展示功能，可以展示非

常详细的图表信息，而且可以定制展示内容，将数据可视化发挥的淋漓尽致；三是具

有分布式管理功能，能够解决大型集群运维工作很多问题，包括监控、预警、日志收

集解析等。四是具有半结构化数据库的功能。将 ELK 应用于大数据平台运维管理既

可以整合资源又可以快速、详尽的查看调用角色信息进行日志分析。在高速发展的大

数据时代背景下，研究具有一定的应用价值。

集中式日志系统

日志，对于任何系统来说都是及其重要的组成部分。在计算机系统里面，更是如此。

但是由于现在的计算机系统大多比较复杂，很多系统都不是在一个地方，甚至都是

跨国界的；即使是在一个地方的系统，也有不同的来源，比如，操作系统，应用服

务，业务逻辑等等。他们都在不停产生各种各样的日志数据。根据不完全统计，我

们全球每天大约要产生2EB 的数据。

面对如此海量的数据，又是分布在各个不同地方，如果我们需要去查找一些重要的

信息，难道还是使用传统的方法，去登陆到一台台机器上查看？看来传统的工具和

方法已经显得非常笨拙和低效了。于是，一些聪明人就提出了建立一套集中式的方

法，把不同来源的数据集中整合到一个地方。

一个完整的集中式日志系统，是离不开以下几个主要特点的。

• 收集－能够采集多种来源的日志数据

• 传输－能够稳定的把日志数据传输到中央系统

• 存储－如何存储日志数据

• 分析－可以支持UI 分析

• 警告－能够提供错误报告，监控机制

1　引言

1．1编写目的

　　说明编写这份软件需求说明书的目的，指出预期的读者。

1．2背景

　　说明：

　　a．待开发的软件系统的名称；

　　b．本项目的任务提出者、开发者、用户及实现该软件的计算中心或计算机网络；

　　C．该软件系统同其他系统或其他机构的基本的相互来往关系。

1．3定义

　　列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

1．4参考资料

　　列出用得着的参考资料，如：

　　a．本项目的经核准的计划任务书或合同、上级机关的批文；

　　b．属于本项目的其他已发表的文件；

　　c．本文件中各处引用的文件、资料、包括所要用到的软件开发标准。 列出这些文件资料的标题、文件编号、发表日期和出版单位，说明能够得到这些文件资料的来源。

2　任务概述

2．1目标

　　叙述该项软件开发的意图、应用目标、作用范围以及其他应向读者说明的有关该软件开发的背景材料。解释被开发软件与其他有关软件之间的关系。如果本软件产品是一项独立的软件，而且全部内容自含，则说明这一点。如果所定义的产品是一个更大的系统的一个组成部分，则应说明本产品与该系统中其他各组成部分之间的关系，为此可使用一张方框图来说明该系统的组成和本产品同其他各部分的联系和接口。|

2．2用户的特点

　　列出本软件的最终用户的特点，充分说明操作人员、维护人员的教育水平和技术专长，以及本软件的预期使甩频度。这些是软件设计工作的重要约束

2．3假定和约束

　　列出进行本软件开发工作的假定和约束，例如经费限制、开发期限等。

3　需求规定

3．1对功能的规定

　　用列表的方式（例如IPO表即输入、处理、输出表的形式），逐项定量和定性地叙述对软件所提出的功能要求，说明输入什么量、经怎样的处理、得到什么输出，说明软件应支持的终端数和应支持的并行操作的用户数。

3．2对性能的规定

3．2．1精度

　　说明对该软件的输入、输出数据精度的要求，可能包括传输过程中的精度。

3．2．2时间特性要求

　　说明对于该软件的时间特性要求，如对：

　　a．响应时间；

　　b．更新处理时间；

　　c．数据的转换和传送时间；

　　d．解题时间； 等的要求。

3．2．3灵活性

　　说明对该软件的灵活性的要求，即当需求发生某些变化时，该软件对这些变化的适应能力，如：

　　a．操作方式上的变化；

　　b．运行环境的变化；

　　c．同其他软件的接口的变化；

　　d．精度和有效时限的变化；

　　e．计划的变化或改进。

　　对于为了提供这些灵活性而进行的专门设计的部分应该加以标明。

3．3输人输出要求

　　解释各输入输出数据类型，并逐项说明其媒体、格式、数值范围、精度等。对软件的数据输出及必须标明的控制输出量进行解释并举例，包括对硬拷贝报告（正常结果输出、状态输出及异常输出）以及图形或显示报告的描述。

3．4数据管理能力要求

　　说明需要管理的文卷和记录的个数、表和文卷的大小规模，要按可预见的增长对数据及其分量的存储要求作出估算。

3．5故障处理要求

　　列出可能的软件、硬件故障以及对各项性能而言所产生的后果和对故障处理的要求。

3．6其他专门要求

　　如用户单位对安全保密的要求，对使用方便的要求，对可维护性、可补充性、易读性、可靠性、运行环境可转换性的特殊要求等。

4　运行环境规定

4．1设备

　　列出运行该软件所需要的硬设备。说明其中的新型设备及其专门功能，包括：

　　a．处理器型号及内存容量；

　　b．外存容量、联机或脱机、媒体及其存储格式，设备的型号及数量；

　　c．输入及输出设备的型号和数量，联机或脱机；

　　d．数据通信设备的型号和数量；

　　e．功能键及其他专用硬件

4．2支持软件

　　列出支持软件,包括要用到的操作系统、编译（或汇编）程序、测试支持软件等。

4．3 接口

　　说明该软件同其他软件之间的接口、数据通信协议等。

4．4控制

　　说明控制该软件的运行的方法和控制信号，并说明这些控制信号的来源。