

中国矿业大学本科毕业设计中期报告

题目：SpringCloud微服务平台设计

姓 名：陈德枫

学 号：03190872

任课教师：刘迎春

中国矿业大学

2023年03月

中国矿业大学教师评阅书

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 |  | 学号 |  |
| 设计题目 |  | | |
| 评阅教师评语（①选题的意义；②基础理论及基本技能的掌握；③综合运用所学知识解决实际问题的能力；④工作量的大小；⑤中期检查表现；⑥取得的主要成果及创新点；⑦写作的规范程度；⑧总体评价及建议成绩；）：  成 绩：  评阅教师签字： | | | |

**目 录**

1. 选题背景与意义
2. 主要内容，拟要解决的问题
3. 研究进展和成果
4. 下一步研究计划
5. 参考文献

1.选题背景与意义

随着互联网和移动互联网的快速发展，软件系统规模越来越大，复杂度也越来越高。传统的单体式应用已经不能满足业务发展的需求，因为单体应用往往难以扩展、难以维护、难以部署，同时对于业务拆分和技术升级也很不友好。因此，为了更好地应对业务发展的需要，软件设计的趋势向着微服务架构的方向发展。微服务架构是一种将应用程序拆分为一系列小型、独立、可独立部署的服务的方法。每个服务都专注于单个业务功能，并可以使用多种编程语言和技术栈来实现。由于微服务架构的服务是松耦合的，因此每个服务可以独立进行开发、部署和维护，从而提高了开发效率和可维护性，并且可以更好地满足业务的变化和扩展需求。与传统的多模块分布式架构相比，尽管它们都涉及到将应用程序拆分为多个组件并分布在多个服务器上运行，但是它们之间有以下几个 区别：

1.领域模型的不同：分布式架构的领域模型通常是单一的，即所有组件都使用相同的数据模型，而微服务架构的领域模型则是多样化的，每个服务都拥有自己的领域模型。

2.通信方式的不同：在分布式架构中，不同组件之间的通信通常是基于远程调用，而在微服务架构中，组件之间的通信通常是基于轻量级协议（如REST API）。

3.部署和维护的不同：在分布式架构中，组件的部署和维护通常是集中式的，而在微服务架构中，每个服务都可以独立进行部署和维护。拆分程度的不同：在分布式架构中，组件之间的拆分程度通常比较低，而在微服务架构中，组件之间的拆分程度通常比较高。 4. 可伸缩性的不同：在分布式架构中，组件的可伸缩性通常比较低，而在微服务架构中，每个服务都可以独立进行伸缩，从而更好地支持高负载和高可用性。

总之，微服务架构和分布式架构都是基于将应用程序拆分为多个组件并分布在多个服务器上运行的思想，但它们的实现方式和目的略有不同。微服务架构更适合构建复杂的业务系统，而分布式架构更适合构建简单的系统或单个功能模块，主要区别可以概括为服务的拆分粒度不同。不过目前在国内外互联网的软件架构中更多的都是微服务与多模块分布式架构的结合，因为服务拆分的越细那么在服务治理与维护上也会更复杂，同时调用链路也会越长，访问的耗时也会越多。目前，微服务架构已经成为业界的标准之一，越来越多的企业开始采用微服务架构来开发应用程序。微服务架构的优势在于它可以支持大规模分布式系统，并且可以提供高可用性、弹性和灵活性如Spring Cloud Netflix，Spring Cloud Alibaba。其中本次设计原生Spring Cloud 和Spring Cloud Alibaba 都会使用。Spring Cloud框架的优势与特点：易于开发和部署：SpringCloud框架提供了一系列微服务组件的集成解决方案，开发人员可以使用SpringBoot快速开发微服务应用程序，并可以使用Docker等容器化技术快速部署和升级应用程序。服务注册与发：SpringCloud框架提供了服务注册与发现组件，可以实现服务的自动发现和注册，使得服务之间的调用更加方便和灵活。负载均衡：SpringCloud框架提供了多种负载均衡的策略，可以根据实际的业务需求进行选择和配置。断路器：SpringCloud框架提供了断路器组件，可以实现服务的容错和熔断，避免了服务之间的故障级联。分布式配置：SpringCloud-Config。

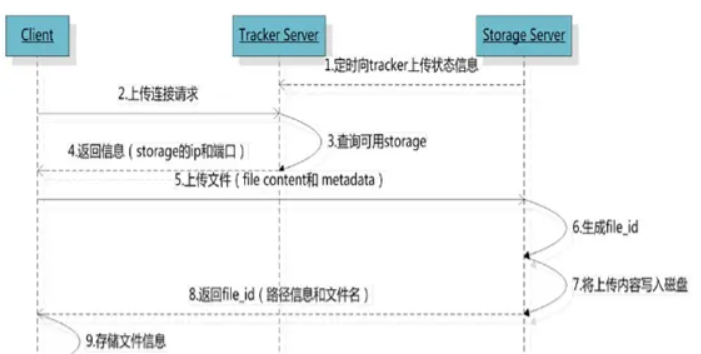
2．主要内容、拟要解决的问题

本次毕业设计的主要目的是设计一个分布式的文档博客管理系统，该系统基于Spring Cloud微服务架构，旨在解决文档和博客的在线查看、检索、权限管理和分布式存储等问题，并且实现整个系统的高可用，同时本次也是一次对于分布式系统的技术架构以及共识算法的学习。以下是技术栈与实现思路，首先，系统的在线查看和检索是本次设计的核心功能之一。①为了实现这一功能，整体采用Spring Boot提供的内嵌式Web服务器，并使用VUE搭建一个简单的前端界面，用于展示文档和博客的内容。②为了实现高可用，本次设计采用Spring Cloud Alibaba提供的Nacos作为服务注册中心与配置中心，将文档博客管理系统中的各个微服务注册到Nacos服务器中，Nacos服务器可以通过心跳检测等机制来监控微服务的状态，并将可用的服务地址信息返回给客户端，从而实现负载均衡和故障转移。③同时，使用Spring Cloud Netflix提供的Hystrix断路器来防止微服务出现故障时的雪崩效应，当某个微服务不可用时，Hystrix可以通过快速失败和服务降级等机制来避免整个系统的崩溃。⑤为了避免直接暴露后端接口，使用Spring Cloud Gateway提供的API网关，实现请求的转发和路由，从而实现负载均衡和动态路由的功能。在实际应用中，可以根据用户的地理位置、网络带宽和负载情况等因素，自动选择最优的服务器节点，从而实现高效的内容分发和访问。⑥系统的权限管理是设计中不可或缺的一环。为了保证文档和博客的安全性，可以采用Shiro（相比于Spring Security，Shiro比较轻量）安全框架来实现用户认证和授权。通过安全框架，可以对用户进行身份验证和授权，以确保文档和博客的安全访问。⑦文档和博客应存储在分布式文件系统中，本次选用的是FastDFS。使用分布式文件系统可以将文档和博客的存储和访问分散到多个节点上，避免单点故障，提高文件的可用性和安全性。⑧为了实现文档的快速检索，Mysql数据库并不能支持，所以将选用 Elasticsearch作为搜索引擎，Elasticsearch支持快速的文本搜索和分析，能够快速地索引和搜索大量的文本数据，从而提高检索效率和准确度；⑨最后，为了保证系统的高可用和稳定性与易于维护，采用SkyWalking提供的分布式跟踪和日志收集工具来监控系统的运行状态和调用链。通过分布式跟踪和日志收集，可以实时监控系统的运行状况和性能指标，及时发现和解决问题，从而保证系统的可靠性和稳定性。

3.研究进展和成果

目前博客文档系统已完成文档上传、下载、检索功能。文档管理服务是该微服务系统提供的核心功能，该服务主要有三部分组成，分别是文档存储服务，文档元数据服务，文档检索服务。文档存储服务是博客文档系统中非常重要的一部分，用于存储博客系统中的各种文件，包括图片、视频、音频、文档等。与传统的文件存储方式不同，博客文档系统采用了FastDfs技术来存储文件。FastDfs是一个开源的轻量级分布式文件系统，由淘宝开发并贡献给开源社区，它具有高性能、高可靠性和高可扩展性的特点，被广泛应用于各种互联网应用中。FastDfs的设计原理是将文件切分成若干个小块，每个小块的大小是固定的，然后将这些小块分别存储到不同的存储节点上。FastDfs由两个主要的组件组成，一个是Tracker Server，另一个是Storage Server。Tracker Server用来管理所有的Storage Server，它可以动态的进行负载均衡、故障转移和容错处理。Storage Server用来存储文件，每个Storage Server上可以存储多个文件，一个文件可以被存储到多个Storage Server上，以实现数据冗余和高可用性。

FastDfs的设计原则是简单、快速和可靠。相比其他分布式文件系统，FastDfs具有快速和可靠的优势。它采用了基于内存的数据结构，避免了随机写磁盘寻址的开销，从而大大提高了存储性能。此外，FastDfs采用了数据冗余和故障转移的技术，可以保证数据的可靠性和高可用性。对于开发者来说，FastDfs的API非常简单易用，无需关心具体的存储细节。尽管FastDfs有很多优点，但也存在一些劣势。首先，FastDfs只是一个轻量级的分布式文件系统，对于大规模的数据存储和管理来说可能不够强大。相比之下，像Hadoop Distributed File System（HDFS）这样的分布式文件系统可以处理大量的数据，并提供了更多的高级功能，例如数据备份和数据恢复。其次，FastDfs的文件上传和下载依赖于网络，如果网络出现问题，就可能会导致文件传输失败或速度变慢。这对于一些需要高可靠性和高速度的应用场景来说可能是不可接受的。第三，FastDfs的元数据管理需要额外的配置和管理，如果不好地进行配置和管理，可能会导致文件的访问性能下降或元数据的丢失。第四，FastDFS没有提供文件访问控制和安全机制，如果需要对文件进行权限控制和加密保护，就需要在应用层进行实现。这对于一些对文件安全性要求较高的应用场景来说可能是一个不足之处。总的来说，FastDFS是一款非常适合用于单文件大小较小（64MB以下）大容量存储和负载均衡的分布式文件系统，它具有高可用性、高性能、简单易用等优点，比较适合于本次纯文本的文档存储服务，如以后需要存储大容量文件如视频、安装包等，则需要升级使用HDFS，如以后需要更稳定的存储服务以及更简单的配置和使用CDN服务来实现全国各地都能快速下载文件则可以更换为云服务厂商提供的对象存储服务（如阿里云oss，腾讯云cos）。

  
 图3.4 FastDfs上传流程

文件上传至FastDfs流程如图3.4所示，在启动FastDfs服务时，Tracker节点担任着集群管理员的角色，其他Storage存储节点会定期向Tracker节点发送心跳包，心跳包中包含了一些关键的信息，能够让 Tracker Server 知道 Storage Server的健康状况、负载状况和存储情况，从而进行相应的管理和调度，用户上传文件的时首先会请求Tracker Server节点，Tracker Server收到文件上传请求后，会查询自身维护的Storage节点信息表，从中进行负载均衡选出最优的Storage节点，并将其信息（即IP和端口）返回给客户端，客户端受收到返回信息后，就会去请求对应的Storage节点上传文件本身内容以及一些附带的元信息例如文件名、用户信息等。Strorage收到文件上传请求后，会根据文件名生成对应的“file\_id”,并将文件写入磁盘，最后将文件的file\_id返回给客户端。FastDFS的fileid由两部分组成，分别是group\_name和filename，用一个“/”符号隔开。其中，group\_name是存储组的名称，由管理员在配置文件中指定，可以根据实际需求进行配置，例如可以将同一类型的文件放在同一个存储组中。而filename是由FastDFS自动生成的，它包含两个部分：

1. timestamp：当前时间的时间戳，单位为秒，可以保证生成的fileid在同一存储组内唯一。
2. 一个自增的序列号：每个存储节点内部维护一个计数器，用于生成唯一的序列号，确保同一存储节点内生成的fileid唯一。

因此，FastDFS的fileid可以保证在同一存储组内唯一，并且可以快速定位到具体的存储节点。例如，一个fileid为“group1/M00/00/01/wKgBzFhTwP2AV7-dAAAABRhZ-8w.jpg”的文件，表示它属于group1存储组，“M00”表示是group1中0号节点，“/00”表示该节点的一级目录为0，“/01”表示二级目录为2最后的的“wKgBzFhTwP2AV7-dAAAABRhZ-8w.jpg”有上传的文件名加上时间戳和版本号生成而来。文件中。通过这个fileid，可以快速找到这个文件所在的存储节点，并进行读写操作；FastDfs文件下载过程与上传文件相似，首先客户端向Tracker节点发送包含file\_id的下载请求，Tracker节点收到请求后，会根据file\_id的前两个字段找到对应的存储节点信息并返回给客户端，最好客户端直接与对应的Storage节点进行通讯完成下载。

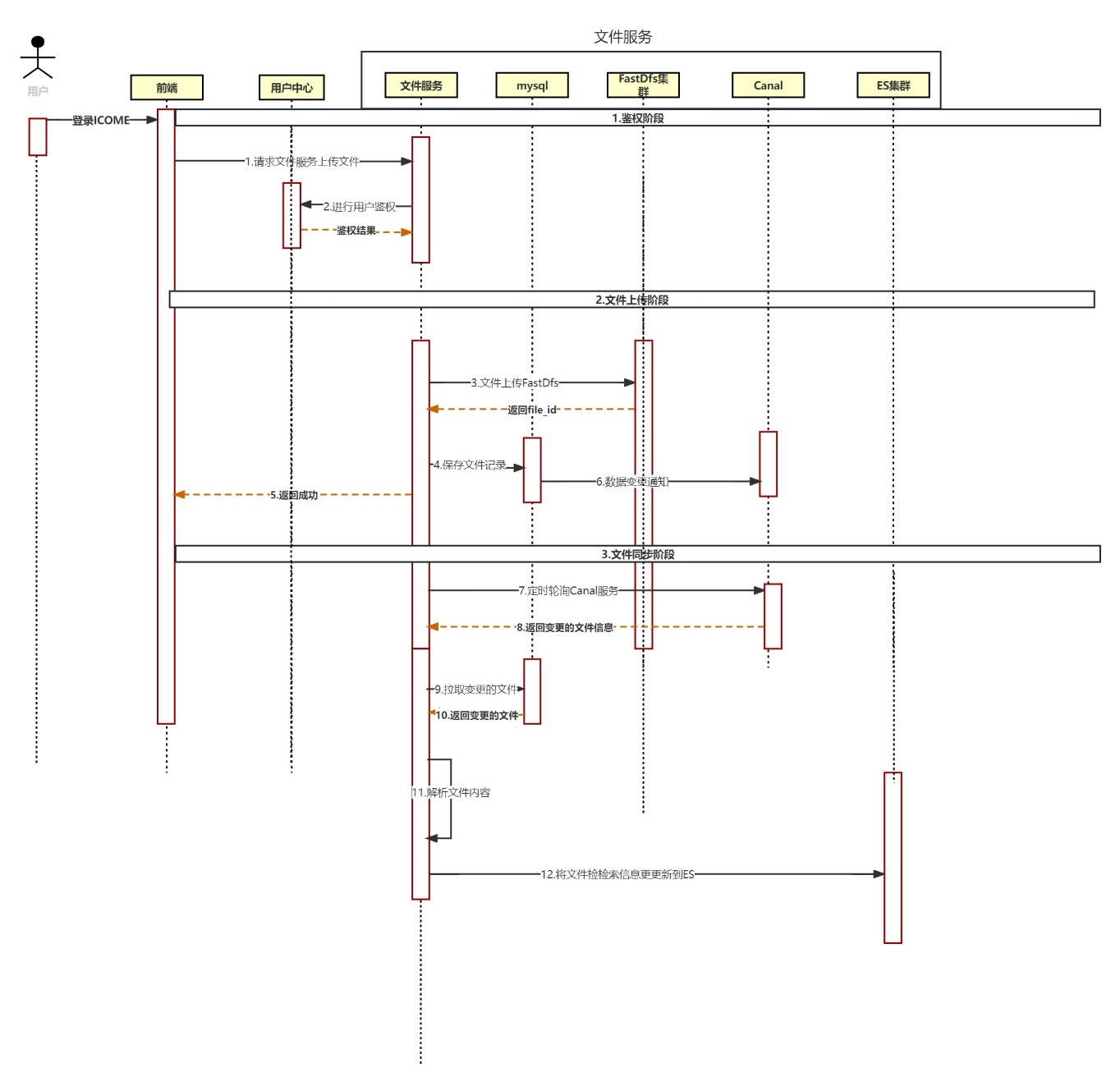


图3.5文件上传时序图

文件上传流程如图3.5所示，

1. 在文件上传流程中，权限校验是保障文件安全的重要一环。拦截器会首先拦截文件上传请求，并将请求发送给用户中心进行权限校验。用户中心会根据用户的身份信息和权限等级判断该用户是否具有上传该文件的权限。如果用户没有权限，系统会直接返回错误提示，避免非法文件上传造成安全隐患。只有通过权限校验的请求才能进入下一阶段，即文件上传阶段。在实际应用中，文件上传流程还可以进一步完善，例如可以对上传的文件进行格式和大小的限制，以及检测上传的文件是否包含恶意代码等安全问题。这些措施可以有效保障文件的安全性和可靠性，避免给系统带来潜在风险。

2在第二个阶段中，文件上传的核心部分是将文件存储到FastDfs中。FastDfs是一种高性能、轻量级的分布式文件系统，它可以快速存储和访问海量的文件数据。在文件上传过程中，FastDfs可以提供高效的文件处理能力，以便快速响应用户上传请求。当文件服务将文件上传到FastDfs后，FastDfs服务会根据规则生成对应的file\_id。file\_id是FastDfs系统中的文件标识符，每个文件都有唯一的file\_id。上传完成后，文件服务会将记录保存到mysql数据库，以便后续的文件解析和检索。此阶段的成功上传后，文件服务会向用户返回文件上传成功的响应。用户现在可以根据file\_id来检索、查看和下载文件。需要注意的是，由于文件上传过程可能会遇到各种异常情况，例如网络故障或FastDfs服务异常等，因此文件服务需要实现错误处理机制来确保文件上传的可靠性。在上传过程中，文件服务应该记录并跟踪所有错误，以便对问题进行调试和排查。

3. 在第三个阶段，文件服务会将已上传的文件进行解析和同步，以便实现文件的模糊检索。由于该阶段耗时较长，因此与第二个阶段的文件上传是独立异步的。为了实现数据同步，文件服务使用了Canal服务，Canal是一个数据同步工具，它可以通过监听mysql的binlog来感知数据的变更。文件服务会定期去Canal服务请求数据同步，Canal会返回近期还没有同步的变更数据。一旦文件服务收到文件的变更信息，它会根据信息中的“file\_id”去FastDfs下载刚刚上传的文件并进行读取解析。解析的结果将被提交到ES集群，以便ES服务根据文件信息内容来生成用于检索的索引。这样就可以通过文件的内容和属性实现文件的模糊检索。ES是一种分布式搜索和分析引擎，它能够高效地存储和搜索大量数据。通过将解析结果提交到ES集群，文件服务可以使文件内容和属性能够被快速地索引和搜索。此外，ES还支持复杂的查询和分析，可以为用户提供更精准的搜索结果。总的来说，该文件上传流程通过权限校验、文件上传和文件解析同步三个阶段来实现文件的上传和搜索功能。它的设计使得系统具有较好的可扩展性和可维护性，同时也能够提供快速和准确的文件搜索服务。

4．下一步研究计划

截止到目前，该微服务平台已经基本实现了核心功能。下一步的研究计划主要包括以下几个方面：首先，需要进一步深入研究微服务的拆分和优化问题，通过对现有微服务进行分析和测试，确定拆分方案并进行优化，使得微服务更加健壮和通用，更符合商用标准。同时，需要对微服务的整体架构进行完善，以提高平台的稳定性和可靠性。

其次，需要对平台的性能进行全面测试，以了解平台的性能瓶颈和瓶颈部分，确定优化方案。需要选出潜在的性能瓶颈部分并对其进行优化，以提高平台的性能和可扩展性。

最后，需要合理规划应用的部署资源。为了保证应用在高并发的情况下能够快速响应，需要对资源进行合理规划和配置，确保应用能够平稳运行。需要对服务器的硬件配置、网络带宽、存储资源等进行评估和优化，以提高应用的性能和可靠性。

综上所述，下一步的研究计划主要集中在微服务的优化和完善、平台性能测试和优化、以及部署资源的规划和配置等方面。通过这些工作的实施，将能够进一步提高微服务平台的性能和可靠性，满足商用需求，并为今后的研究提供有益的经验和借鉴。

5.参考文献

[1] Fowler, M. (2014). Microservices: a definition of this new architectural term. Retrieved from <https://martinfowler.com/articles/microservices.html>

[2]Zhao, Y., & Li, S. (2017). Research on the Design and Implementation of Spring Cloud Microservice Archi tecture.． In 2017 2nd International Conference on Computer Science and Technologies in Education (CSTE), 98-101. doi: 10.1109/CSTE.2017.23

[3]Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7), 558-565. doi: 10.1145/359545.359563

[4]Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7), 558-565. doi: 10.1145/359545.359563

[5]郑响萍, 伏金娣,and 蔡海军."基于微服务和区块链的数字签章系统的设计与实现." *计算机时代* .04(2023):91-95+100. doi:10.16644/j.cnki.cn33-1094/tp.2023.04.018.

[6]术洪祥,陈成栋,吴晓园,周晓光.基于微服务架构的循环包装箱公共信息服务平台设计与实现[J].物流技术,2023,42(03):135-142.

[7]胡一凡,李兆峰,李少辉,高焕芝,王小英,黄猛.基于微服务架构的选矿工艺数字化计算系统设计与实现[J].矿业研究与开发,2023,43(03):185-190.DOI:10.13827/j.cnki.kyyk.2023.03.027.

[8]李俊,江海.基于微服务架构的新零售系统设计与应用[J].计算机时代,2023(03):106-110+115.DOI:10.16644/j.cnki.cn33-1094/tp.2023.03.025.

[9]刘庆伟.基于微服务架构的视频图像服务总线实现[J].警察技术,2023(02):77-80.