

基于微服务和 TCC 分布式事务的在线教育平台设计方法研究

李经纬¹, 韩树河²

(1. 安艾艾迪信息技术(上海)有限公司 技术部, 上海 200050;
2. 江苏航运职业技术学院 智能制造与信息学院, 江苏 南通 226010)

摘要:传统的单体应用架构开发的在线教育平台存在可扩展性差、部署效率低等问题。通过研究提出一种基于微服务架构和 TCC 分布式事务的设计方法,对平台按照功能进行解耦垂直拆分成多个独立的微服务,使用 Spring Cloud 构建微服务来应对高并发和实现快速自动化部署。基于 TCC 分布式事务框架,执行服务间的调用事务,确保数据一致性。平台在线运营已经超过一年,为超过 8 万多名学习者提供稳定的学习管理服务,证明该设计方法有效。

关键词:在线教育平台;微服务架构;TCC 分布式事务

中图分类号:G434

文献标志码:A

文章编号:1671-9891(2020)2-0048-04

0 引言

随着互联网的高速发展以及教育行业数字化转型升级加快,互联网和新兴信息技术已经开始引领教育未来,甚至重塑教育。为了适应这些变化,越来越多的教育者和学习者开始参与在线课程以突破时间、空间、设备的限制,获取更多知识。在线教育平台作为在线教育的重要支撑,需要为教育者和学习者提供便捷、全面的课程资源和符合网络教育环境特征的学习空间^[1]。同时,从互联网技术角度而言,在线教育平台属于互联网在线平台,同样拥有着互联网平台的高并发、业务数据复杂和迭代频繁等特点。

现阶段的在线教育平台多采用传统的单体应用架构,较少采用微服务架构。在传统的单体应用架构中,对于事务处理一般局限于资源层面和单机事务,在业务升级和功能扩展过程中往往容易出现牵一发动全身的局面,导致系统升级和扩展成本高昂。

本文将对微服务和 TCC(Try、Confirm、Cancel)分布式事务优势进行介绍,基于微服务和 TCC 分布式事务设计在线教育平台的架构,为在线教育平台的研究开发提供支撑。

1 微服务与 TCC 分布式事务

1.1 微服务

随着业务不断增加,传统的单体应用架构在项目的复杂性、可靠性、扩展性和部署效率等方面暴露出越来越多的问题。因为业务的升级,项目的代码量会急剧增加,项目模块也会随之增加,同时模块与模块之间的关系将会变得越来越复杂,从而导致整个项目变得非常复杂。当项目中需要新增或修改业务的时候,工程师需要在现有业务逻辑上编写代码和重复多次测试,同时也增加了因代码改动而导致影响之前业务的风险。在项目部署时,可能面临因为改动简单的缺陷而不得不重新部署整个项目,增加了部署时间。随着以云计算、容器虚拟化、敏捷开发和 Devops 等为代表的互联网技术的发展,面向服务架构的微服务架构因能解决单体应用问题而成为开发者和企业的首选。微服务架构主张将复杂的单体应用拆解为细粒度且功能专一、松耦合和高度自治的分布式服务,这样可以降低开发难度、按需伸缩,增强扩展性、便于敏捷开发。此外,微服务与容器虚拟化、Devops 技术天然契合,从而被越来越多的开发者和公司运用。

在线教育平台的功能模块众多,需要覆盖学习者从注册学习到考试认证的整个过程。尤其是在考证冲刺期间,在线教育平台的短时并发量将急剧上升,需要根据学习者并发量合理调配软硬件资源,按需伸缩。因此,采用基于微服务架构的在线教育平台,按照单一职责原则将模块划分为服务,结合 Docker 容器技术自

收稿日期:2020-05-20

作者简介:李经纬(1984—),男,江苏南通人,安艾艾迪信息技术(上海)有限公司技术部助理工程师,硕士。

动化独立部署服务,减少部署时间,从而实现按需伸缩,提高平台整体的可用性和扩展性。

1.2 分布式事务与 TCC 分布式事务

在传统单体应用架构中,业务数据通常存储在一个数据库中,应用中的各个模块对数据库直接进行操作。在这种场景中,事务是由数据库的 ACID(Atomicity、Consistency、Isolation、Durability)特性来保证的。^[2]

然而,随着微服务架构的引入,在平台被微服务化之后,服务间的调用关系将变得繁杂,事务问题也将变得突出。根据 CAP(Consistency、Availability、Partition tolerance)理论^[3],强一致性、可用性和分区容错性最多只能同时满足两个。因此,分布式事务成了微服务架构落地的技术难题。基于 CAP 理论和 BASE(Basically Available、Soft state、Eventually consistent)理论^[4],微服务架构下,我们可以通过弱化一致性和采用基本可用的原则来帮助平台最终达到一致性。在分布式系统中,尤其是业务层,主要关注业务拆分,在按照业务横向扩展资源时,解决微服务间调用的一致性,保证资源访问的事务属性。相比较于传统单体应用架构下一般采用的分布式事务处理模型 DTP(Distributed Transaction Processing)方案,TCC 柔性事务补偿在业务层针对业务需求灵活控制事务,并发性好并且保证了隔离性。^{[5][157]}

TCC 为跨服务场景下提供 Try 接口、Confirm 接口和 Cancel 接口解决分布式事务问题。TCC 要求对于分布式系统中的主业务服务 Z 和特定业务服务 T,当特定业务服务 T 向其他业务提供服务时,T 必须接受不确定性。因为主业务服务 Z 对特定业务服务 T 的调用只是一个临时 Try 操作,Z 享有后续的取消权。如果主业务服务 Z 需要回滚全局事务,则将会取消执行临时 Try 操作,对应执行 Cancel 操作。当主业务服务 Z 需要提交全局事务时,将会放弃 Cancel 操作权限,对应执行 Confirm 操作。最终,临时 Try 操作都将执行 Confirm 操作或者 Cancel 操作。因此,TCC 分布式事务模型要求业务系统需要提供三个业务逻辑段,即 Try 操作、Confirm 操作和 Cancel 操作。

在线教育平台的分布式事务框架上,选用基于 TCC 柔性事务补偿方案,能够满足在线教育平台高并发量的特点,同时也能在较短的时间内执行业务,确保数据一致性。^{[5][157]}

2 基于微服务和 TCC 分布式事务的在线教育平台

2.1 在线教育平台微服务架构

在线教育平台采用基于 Java 微服务技术开发。在 Java 微服务开发领域,往往采用基于 Dubbo 或 Spring Cloud 框架来实现微服务。在线教育平台采用的是基于 Spring Cloud 微服务框架, Spring Cloud 通过 HTTP 协议调用 REST 接口的方式进行服务间通信,比 Dubbo 更为灵活。服务的提供方和调用方之间不存在代码级别的强依赖,更加适合强调快速演进的微服务环境。如图 1 所示,整个在线教育平台自下而上划分为数据层、业务层、接入服务层和前端 UI 层。

(1)数据层。主要负责在线教育平台关系数据持久化存储、缓存、消息队列和文件存储。包括为关系数据

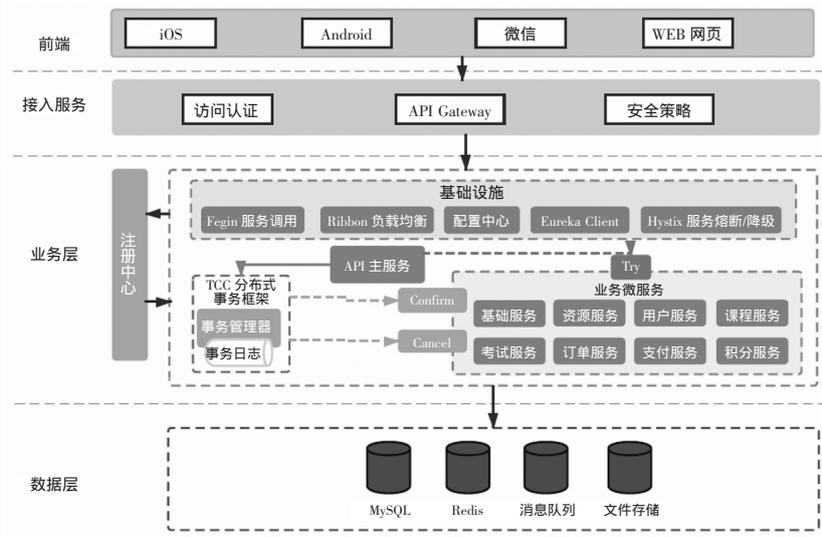


图 1 在线教育平台系统架构图

提供存储的 MySQL 数据库、负责平台分布式 Session 和重要缓存的 Redis 数据库、为平台消息异步提供高可用的 Rabbit MQ 消息队列和为静态资源和视频文件提供存储的文件系统。

(2)业务层。主要负责从数据层获取数据,为前端展示提供数据支持。该层是在线教育平台的核心层。由注册中心、微服务基础设施、TCC 分布式事务框架、API 主服务和业务微服务组成。

1)注册中心使用基于 Spring Cloud Eureka 服务治理组件,为平台各服务提供服务注册和发现。

2)微服务基础设施为服务间调用提供配置管理、负载均衡和熔断/降级。使用 Fegin 声明性服务调用组件为平台前端展示层和业务微服务间的调用提供良好的封装,简化服务间的调用。使用基于 Spring Cloud Config 作为配置中心,实现平台各服务配置数据能够实现外部化存储,避免因为配置改变而频繁重启服务。使用 Ribbon 为平台客户端与服务间调用提供负载均衡支持。使用 Hystrix 及时阻断服务调用的故障传播,防止平台雪崩,提升平台的稳定性。

3)业务微服务对在线教育平台按照功能进行解耦合垂直拆分成多个微服务。服务内部进行封装,以 Rest API 的方式对外暴露。

4)TCC 分布式事务框架为服务间调用提供分布式事务支持,保证数据一致性。

(3)接入服务层。作为前端展示层和业务层的中间桥梁,使用 Zuul 网关组件,通过配置安全策略,为前端访问业务层微服务提供智能路由、访问认证等。

(4)前端展示层。负责为平台使用者提供多端设备的访问支持。

2.2 在线教育平台 TCC 分布式事务架构

业务层的 TCC 分布式事务框架主要由事务管理器、API 主服务(主业务服务)和多个业务微服务(从业务服务)组成。

(1)业务微服务作为从业务服务,是整个业务活动的操作方,为 API 主服务和事务管理器的调用提供 Try、Confirm、Cancel 接口。在 Try 接口中,负责检查所需业务资源是否足够并预留该资源,从而满足事务的一致性和隔离性。在 Confirm 接口中,不需要做任何业务检查,只需要使用在 Try 接口中保留的业务资源来执行业务逻辑。在 Cancel 接口中,负责释放 Try 接口中预留的业务资源。

(2)API 主服务作为主业务服务为整个业务活动的发起方,负责操作多个从业务服务上的资源。

(3)事务管理器管理和控制整个业务活动。结合事务日志,记录和维护在线教育平台 TCC 分布式框架中全局事务的事务状态和每个业务微服务业务活动的子事务状态。在提交业务活动时调用业务微服务的 Confirm 接口,在业务活动被取消时调用所有业务微服务的 Cancel 接口。

在实现 TCC 分布式事务框架后,还需要注意由于网络超时、宕机等原因引起的分布式事务执行异常,常见异常有幂等、空回滚和悬挂等。

幂等是分布式和高并发领域重要的技术指标,要求 TCC 的 Confirm 接口和 Cancel 接口保证幂等。若幂等控制不好,很可能会导致一系列严重问题,如资损。因此,在设计 TCC 分布式事务框架中,需要在事务管理器中加入事务日志来记录每个调用分支事务的执行状态。在 Try 接口执行“是”,记录状态为“INITIAL”。在 Confirm 接口执行后修改状态为“UPDATED”或者在 Cancel 接口执行后修改状态为“ROLLED”。当重复调用 Confirm 接口或者 Cancel 接口时,先从事务日志中检查该事务状态,如果是“UPDATED”或者“ROLLED”则不再继续执行,直接返回成功,否则,正常执行。

空回滚往往是由于 API 主服务在调用业务微服务的 TCC 时,API 主服务宕机导致调用失败而未执行 Try 接口,根据分布式事务回滚机制,在回调业务微服务的 Cancel 接口时所造成的。在解决空回滚时,需要在事务日志中为分布式事务和分支事务生成唯一的 ID,用以识别出该事务回滚。在执行 Try 接口时候插入一条表示 Try 接口已经执行的记录。在 Cancel 接口执行的时候,读取该记录,如果记录存在,则可以正常回滚,否则是空回滚。

悬挂是由空回滚造成的一种长期占用资源无法释放的现象。对于 TCC 分布式事务框架来说,因为空回滚的出现,Cancel 接口将会比 Try 接口先执行。但是因为 Cancel 接口已经执行完,TCC 分布式事务框架将会认为该事务已经结束。然而在 Cancel 接口执行之后,Try 接口才真正执行,它将会根据自己的职能通过加锁

的方式检查所需业务资源是否足够并预留该资源。对于该预留资源,只有该分布式事务才能够使用,进而造成该资源被长期锁定形成悬挂。为了防止悬挂产生,需要确认 Cancel 接口或者 Confirm 接口执行完后,Try 接口就不能再继续执行。所以在 Cancel 接口或者 Confirm 接口执行时,在事务日志中插入对应的执行状态,例如:“ROLLED”或者“UPDATED”。这样,在 Try 接口执行前,先从事务日志中读取该执行状态记录,如果记录已经存在,则 Try 接口不再执行,否则正常进行。

3 结束语

本文基于 Spring Cloud 框架为在线教育平台设计了微服务架构,真正意义上达到了松耦合的架构设计原则,降低了开发和测试难度,提升了平台的可维护性。配合容器技术,实现业务模块快速自动化部署。选用 TCC 分布式事务框架,为在线教育平台的分布式事务提供了一种柔性事务补偿方案,在满足在线教育平台高并发需求的同时,确保了平台数据一致性。基于微服务架构的在线教育平台在线运营已经超过一年,为超过 8 万多名学习者提供稳定的学习管理服务,实践证明该设计方法有效。微服务架构和分布式事务是分布式领域的关键技术,但是随着平台的商业化宣传,学员的并发将越来越大,对未来 TCC 性能优化提出了更高要求,需要进一步深入研究才能在平台后续升级中满足更高的业务需求。

参考文献:

- [1]王永明,徐继存.论在线课程教学系统的建构[J].中国电化教育,2018(3):66-73.
- [2]Jim Gray,Andreas Reuter.事务处理:概念与技术[M].北京:机械工业出版社,2004:2-4.
- [3]陈明.分布系统设计的 CAP 理论[J].计算机教育,2013(15):109-112.
- [4]Pritchett,Dan.BASE: AN ACID ALTERNATIVE[J].Queue,2008(3):48-55.
- [5]方意,朱永强,宫学庆.微服务架构下的分布式事务处理[J].计算机应用与软件,2019(1):152-158.

(责任编辑 张 利)

Research on Design Methods of Online Education Platform Based on Micro-services and TCC Distributed Transaction

LI Jing-wei¹, HAN Shu-he²

(1. Technology Department, NIIT (Shanghai) Co., Ltd., Shanghai 200050, China;

2. School of Intelligent Manufacturing and Information, Jiangsu Shipping College, Nantong 226010, China)

Abstract: The online education platform developed by traditional single application architecture exists problems such as poor scalability and low deployment efficiency. Through research, a design method based on micro-service architecture and TCC distributed transactions is proposed. The platform is decoupled according to its functions and vertically split into multiple independent micro-services, and Spring Cloud is used to build micro-services to cope with high concurrency and realize rapid automatic deployment. Based on the TCC distributed transaction framework, recalling transactions between services are executed to ensure data consistency. The platform has been operating online for more than a year, providing stable learning management services for more than 80,000 learners, which proves the effectiveness of this designed method.

Key words: online education platform; micro-service architecture; TCC distributed transaction