

文章编号: 2095-2163(2023)06-0193-04

中图分类号: TP311.11

文献标志码: A

基于云和微服务的区块链平台部署方法研究

朱冠烨, 裴伦浩, 郭彦辉

(中国联合网络通信有限公司 济南软件研究院, 济南 250000)

摘要: 天宫区块链服务系统(BCS)是基于开源框架 Hyperledger Fabric 开发的一款面向联通业务应用的区块链服务平台,用于支撑联通部分业务数据上链、存证等需求。传统的区块链平台部署方式主要依赖于物理主机,通过繁琐的配置流程搭建区块链的运行环境,运行过程面临节点扩缩、资源扩展等问题,系统整体稳定性、灵活性不高。基于此,本文提出了一种基于微服务拆分的区块链平台云环境快速部署方法,通过构建、部署服务镜像,可以在 k8s 集群中快速实现区块链平台部署。

关键词: 区块链平台; 云部署; 微服务

Research on the deployment method of blockchain platform based on cloud and microservice

ZHU Guanye, PEI Lunhao, GUO Yanhui

(Jinan Software Research Institute of China Unicom, Jinan 250000, China)

【Abstract】 The Tiangong blockchain service system (BCS) is a platform for business applications of China Unicom developed based on the open-source framework Hyperledger Fabric. It is used to support some demands of business data on-chain and certificate deposit. The traditional blockchain platform deployment method mainly relies on the physical host to build the operating environment of the blockchain through the cumbersome configuration process. It faces some problems such as node expansion and resource expansion at any time, resulting in low stability and flexibility of the system. To solve these problems, a rapid deployment method for the blockchain platform cloud environment based on microservice splitting is proposed. Through the building and deployment of service images, the blockchain platform deployment can be quickly implemented in the k8s cluster.

【Key words】 blockchain platform; cloud deployment; micro service

1 研究背景

1.1 区块链平台

区块链是按照一定的共识规则将数据存储于不同区块,并以一定的哈希规则组合相连的数据结构,通过密码学的方式保证其不可篡改的分布式账本^[1]。凭借“不可篡改”、“可追溯”等优良特性,近年来区块链技术发展如火如荼,众多大型企业纷纷将区块链技术作为重点的突破方向。比较经典的区块链应用,如比特币、以太坊、超级账本等,在各行业已经投入使用。联通链 BCS 系统是由中国联通软件研究院自研的区块链产品,BCS 是集区块链部署、运维、节点管控、应用等能力于一身的可视化操作平台,支持以租户的方式在联通云上快速构建稳定、安全的生产级区块链网络,降低用户对区块链底层技

术的获取成本,实现业务数据快速、稳定、安全上链。

传统的区块链部署方法较为冗杂,部署时需要进行各主机的适配和区块链调度的联调,使用过程中主机资源不足时,进行资源扩展的过程较为复杂,常常影响业务数据的使用。随着云技术越来越成熟,腾讯云、阿里云、联通云等提供了各类资源池和环境沙箱,资源支撑弹性调度。基于此,本文从云环境出发,进行了区块链平台搭建的尝试。

1.2 容器和 kubernetes

容器是指与系统其他部分隔离开的一系列进程,类似于一个封闭的箱子,该箱子提供了某些应用所需的全部依赖^[2]。由于容器的封闭性及其完备性,使得容器化的程序具有较高的可移植性。容器技术的出现,使得复杂的业务过程可以进行解耦,按照功能模块进行微服务拆分,各模块各司其职,当某个模块运行中

作者简介: 朱冠烨(1993-),男,硕士,中级工程师,主要研究方向:通信软件和区块链应用软件研究;裴伦浩(1994-),男,学士,中级工程师,主要研究方向:软件应用研发、区块链应用软件研究;郭彦辉(1997-),男,硕士,主要研究方向:通信软件、区块链应用软件研究。

通讯作者: 朱冠烨 Email: zhugy28@chinaunicom.cn

收稿日期: 2022-06-27

断时也不会使其他模块受到影响。因此,如果对区块链平台进行微服务拆分,当某个子模块出现资源不足或者故障时,也不会影响区块链平台的业务使用。

kubernetes(k8s)是一个开源,以集群方式部署调度容器应用,弹性伸缩以及运维容器集群的系统^[3]。区块链服务底层基于 k8s 引擎调度,因各节点之间调用复杂,各节点又可能面临资源扩缩的问题,传统的本地集群部署方式运维复杂,因此,区块链云化部署变得十分必要。

2 基于微服务的区块链平台部署

2.1 区块链微服务架构

BCS 系统架构如图 1 所示。通常一个完整的区

块链平台系统调用逻辑如下:前端业务层用于和不同的业务系统对接,提供区块链对外服务的接口,例如提供数据上链、查询等操作支撑,当用户发送前端的区块链操作请求之后,请求会转发到区块链 SDK 调度层,即区块链平台后台服务,后台服务会对前端的请求进行判断和转发,如果是对平台本身的操作,则进行数据库交互;如果请求到了智能合约,则转发到智能合约层^[4];如果请求到了区块链引擎或者 k8s 引擎,则转发至相应的服务接口处。当请求涉及到区块链操作时,区块链底层服务会对交易进行排序、打包、记账等一系列操作,生成新的交易区块并广播到所有区块,该过程在区块链底层服务层完成,完成一次完整的前台请求调用区块链服务。

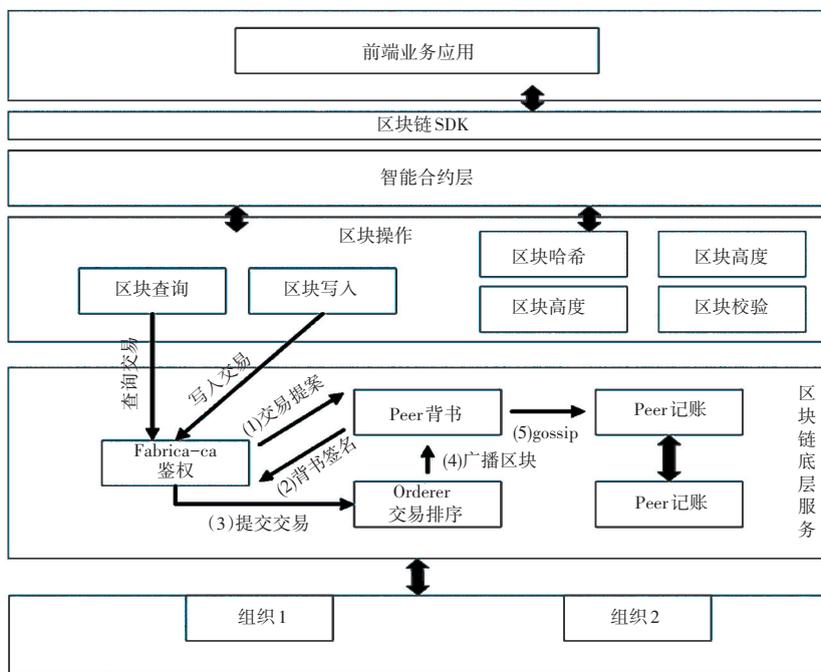


图 1 BCS 系统架构图

Fig. 1 BCS system architecture diagram

2.2 基于微服务的云部署方案

按照区块链平台的工作流程,本文提出了基于微服务拆分的区块链平台部署方法,如图 2 所示。首先需要进行云集群的搭建,目前市面上流行的云开发环境都提供了直接可用的 k8s 集群环境;之后申请云环境配套的数据库、共享存储等资源;根据区块链各个模块的功能特点,进行区块链平台微服务拆分、服务镜像构建、部署;最后,对部署的平台进行可用性验证。

2.3 微服务构建与部署

每个模块需要构建其服务镜像,以区块链引擎模块为例,对镜像构建以及应用部署过程介绍如下:在编写镜像构建的 Dockerfile 时,需要先将启动

fabric 的配置文件,根据实际环境进行修改;在构建镜像时,需要将 fabric 的引擎程序和区块链节点模板放在同一目录下,并将启动区块链节点的配置共享文件放在共享存储的磁盘上,镜像构建完毕,推送至镜像仓库。

部署时需要编写该服务模块启动使用的 yaml 文件,使用 deployment 的方式将该服务启动起来。首先需要为该 pod 提供挂载的 PV 和 PVC,配置好镜像拉取策略进行镜像拉取,指定好服务的 pod 对应的资源量和指定容器的服务端口。需要注意的是,因为区块链引擎需要调用智能合约部分,需要指定好智能合约的上传路径,在环境变量中进行配置,将 GOPATH 环境变量配置在启动文件中。

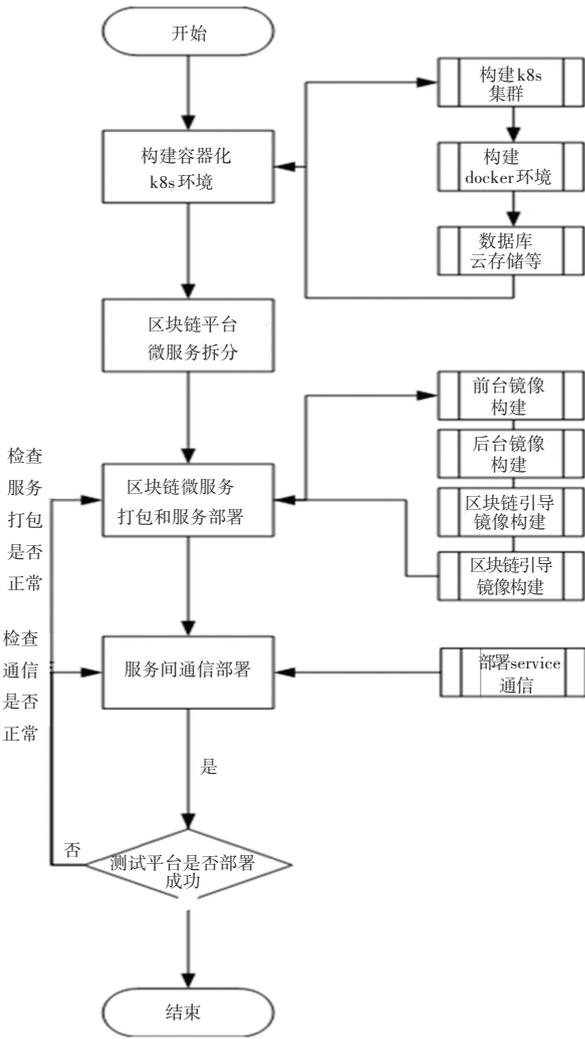


图 2 基于微服务拆分的区块链平台部署方法

Fig. 2 Blockchain platform deployment method based on microservices

最后通过 kubernetes 中 service 的部署方式,将容器的服务端口暴露出去,实现不同服务模块之间互访,完成配置。所有的服务部署完毕,通过尝试调用智能合约是否成功即可查看平台是否部署成功。

3 平台可用性验证

3.1 智能合约验证

区块链的应用场景很多,依托于区块链的加密特性以及智能合约的业务场景结合,在运营商领域,区块链的核心应用场景还是存证相关的业务,常见的场景如关联交易业务、供应链黑名单、供应链协同存证等等。为了支撑以上业务数据存证上链需求,需要开发相应的智能合约函数,通过对智能合约的请求频次统计发现,以下两个智能合约接口调用最为频繁,分别为: stub.PutState(args[0], [] byte(args[1])) , stub.GetState(args[0]) ,其中前者实现了

将数据以 key-value 的形式存入状态数据库;后者可以通过 key 值对数据进行查询,因此,通过对智能合约的读、写可以测试平台的稳定性,本文主要使用智能合约中最常用的存储键值(putstate)和查询键值(querystate)两个函数进行测试。

3.2 区块落地策略配置

Fabric 的落地策略会对系统的性能产生一定的影响,落地策略主要包括 4 个参数:BatchTimeout(出块响应时间),MaxMessageCount(区块最大交易数),AbsoluteMaxBytes(区块最大存储量),PreferredMaxBytes(单笔最大交易数据量),本文在实验时,为了简化模型,尽量避免使用时间超时出块策略的同时,多次调整参数后,统一采用响应时间为 2 s,最大交易数为 100 笔,最大存储量为 99 MB,单笔最大交易 512 KB 来进行测试。

3.3 测试结果

为了模拟区块链平台实际使用时的情形,本文对不同的状态数据库进行了压力测试,通过不同数据量的情形分别测试,得到的结果如图 3、图 4 所示:随着数据量的增加,请求时延有增加,但是并没有出现超时的情形,CPU 的利用率也保持在合理的范围内,没有出现异常情形,说明平台稳定运行,满足生产需求。

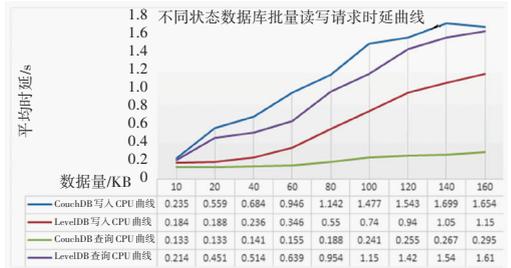


图 3 不同状态数据库批量读写平均请求时延曲线

Fig. 3 The average request latency for batch reads and writes of two state databases during a single request

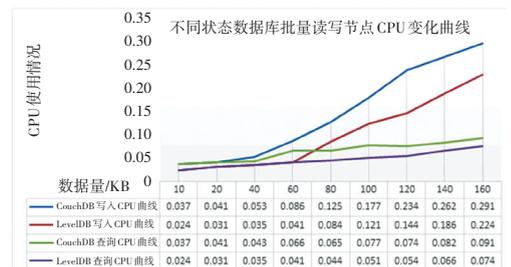


图 4 不同状态数据库批量读写-Node Peer CPU 使用量变化曲线
Fig. 4 CPU batch read-writing usage of Node Peer in different state databases