



计算机应用  
*Journal of Computer Applications*  
ISSN 1001-9081, CN 51-1307/TP

## 《计算机应用》网络首发论文

题目: 基于 Kubernetes 的 Fabric 链码管理及高可用技术  
作者: 刘宏宇, 梁秀波, 吴俊涵  
收稿日期: 2020-12-16  
网络首发日期: 2021-02-05  
引用格式: 刘宏宇, 梁秀波, 吴俊涵. 基于 Kubernetes 的 Fabric 链码管理及高可用技术 [J/OL]. 计算机应用.  
<https://kns.cnki.net/kcms/detail/51.1307.TP.20210205.1332.005.html>



**网络首发:** 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式 (包括网络呈现版式) 排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

**出版确认:** 纸质期刊编辑部通过与《中国学术期刊 (光盘版)》电子杂志社有限公司签约, 在《中国学术期刊 (网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊 (网络版)》是国家新闻出版广电总局批准的网络连续型出版物 (ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

# 基于 Kubernetes 的 Fabric 链码管理及高可用技术

刘宏宇, 梁秀波\*, 吴俊涵

(浙江大学软件学院, 浙江 宁波 315048)

(\* 通信作者电子邮箱 xiubo@zju.edu.cn)

**摘要:** 区块链即服务(BaaS)平台的核心在于如何将区块链网络部署在云计算平台上。Fabric 部署可以按照组件启动时间分为静态组件和动态链码两部分,链码部署是 Fabric 云化最核心、最复杂的部分。Fabric 本身没有针对 Kubernetes 开发接口,所以业界当前的方案均是通过一系列辅助技术实现链码部署的,但这些方案并没有将链码随静态组件一起纳入到 Kubernetes 管理环境中。针对当前 BaaS 方案的问题,主要做了如下几项工作:1)比较全面地研究了底层基础设施,尤其是生产环境下 Kubernetes 平台的高可用;2)设计并实现了 Fabric 在 Kubernetes 上的云化部署,尤其是链码部分,通过一个全新的容器控制插件,实现了对 Kubernetes 在代码级别上的支持,完成了将链码纳入 Kubernetes 环境管理的目标;3)探索函数计算服务管理 Fabric 链码,实现了一个全新的链码执行模式:从“启动-等待-调用-等待”的模式改变为“启动-调用-退出”的模式。本文在 Fabric 云化部署尤其是链码部署管理方面的工作,对基于 Fabric 和 Kubernetes 的 BaaS 平台优化,有一定的参考价值。

**关键词:** 区块链; 区块链即服务; Hyperledger Fabric; Kubernetes; 链码; 无服务器函数计算

**中图分类号:** TP311.5 **文献标志码:** A

## Kubernetes-based Fabric chaincode management and high availability technology

LIU Hongyu, LIANG Xiubo\*, WU Junhan

(School of Software Technology, Zhejiang University, Ningbo Zhejiang 315048, China)

**Abstract:** The core of the Blockchain as a Service (BaaS) platform is how to deploy the blockchain network on the cloud computing platform. Fabric deployment can be divided into static components and dynamic chaincode according to component startup time. Chaincode deployment is the core and most complex part of Fabric cloudification. Fabric itself does not develop an interface for Kubernetes, so the current solutions in the industry implement chaincode deployment through a series of auxiliary technologies, but these solutions do not incorporate the chaincode into the Kubernetes management environment along with static components. In response to the current BaaS solution, this article had mainly done the following tasks: (1) A comprehensive study of the underlying infrastructure, especially the high availability of the Kubernetes platform in the production environment; (2) Design and implementation of Fabric on Kubernetes The cloud deployment of Kubernetes, especially the chaincode part, through a brand-new container control plug-in, realized the support for Kubernetes at the code level, and completed the goal of incorporating chaincode into Kubernetes environment management; (3) Explore functional computing service management Fabric chaincode realized a brand new chaincode execution mode: from the "start-wait-call-wait" mode to the "start-call-exit" mode. This article's work in Fabric cloud deployment, especially chaincode deployment management, has certain reference value for the optimization of the BaaS platform based on Fabric and Kubernetes.

**Key words:** Blockchain; Blockchain as a Service (BaaS); Hyperledger Fabric; Kubernetes; chaincode; serverless function computing

## 0 引言

区块链即服务(Blockchain as a Service, BaaS)是一种创建、管理和运维企业级区块链网络及应用的云服务平台。它可以为开发者提供便捷、高性能的区块链服务,并降低开发成本<sup>[1]</sup>。按照云计算服务的划分,BaaS 有 PaaS (Platform as a Service)和 SaaS (Software as a Service)两种形态,目前绝大多

数以 PaaS 为主,对外提供一种打包封装好的区块链技术平台<sup>[2]</sup>。BaaS 平台的发展需要云计算和区块链两个方向的技术能力,具有较高的技术门槛,头部的云计算公司提供了大多数的 BaaS 平台。

IBM 在 2016 年 2 月宣布推出区块链服务平台,支持用户通过 Bluemix 的区块链服务完成创建、部署、运行和监控区块链应用程序的任务;微软在 2016 年 8 月正式开放其 Azure 云

收稿日期:2020-12-16;修回日期:2021-01-14;录用日期:2021-01-15。

基金项目:浙江省教育厅一般科研项目(Y202044224);浙江省研究生教育学会项目(2020-002);浙江省文物保护科技项目(2019009)。

作者简介:刘宏宇(1994—),男,河北邯郸人,硕士,主要研究方向:容器云、区块链; 梁秀波(1983—),男,山东莱阳人,副研究员,博士,CCF 会员,主要研究方向:区块链、人工智能、自然人机交互、移动互联网; 吴俊涵(1998—),男,浙江丽水人,硕士研究生,CCF 会员,主要研究方向:区块链隐私保护和数据存储。

平台的 BaaS 服务,支持多种区块链网络(Hyperledger Fabric、Ethereum、Corda 等),提供快速构建、管理和扩展区块链网络的能力,Azure 用户可以专注于业务开发,实现高效上链;AWS 在 2018 年 4 月,发布了区块链模板服务,AWS 用户可以选择 Ethereum、Hyperledger Fabric 等常见的开源框架,快速创建和部署区块链网络<sup>[3-4]</sup>。

中国互联网公司先后推出 BaaS 平台:2017 年 4 月,腾讯发布了区块链白皮书和自主研发的 TrustSQL 区块链平台;2018 年 2 月,华为云推出 BCS(Blockchain Service)区块链解决方案,旨在通过与人工智能(Artificial Intelligence, AI)、物联网(Internet of Things, IoT)、5G 等前沿技术的结合构建一个坚实的技术底座,使能各行业数字化高效转型;2018 年 7 月,阿里云商业产品区块链服务(公测)发布,该服务可以帮助用户快速构建稳定、安全的生产级区块链网络,减少在区块链部署、运维、开发等方面的挑战,让用户专注于核心业务创新,实现快速上链;2018 年 8 月,京东区块链开放平台正式发布,该平台可以帮助客户快速构建、管理和使用区块链应用<sup>[5-6]</sup>。

几乎所有的云计算巨头及区块链创业公司都提供了 BaaS 平台,但当前的平台有一个显著的同质化的问题,产品功能甚至底层实现方案大同小异,没有明显的差异化。事实上,大多数的云计算公司只是简单地提供了一种开源区块链平台的一体化部署管理方案,没有深入到区块链技术的底层,而区块链公司关注更多的是区块链技术本身,这就导致目前的 BaaS 底层仍有许多改进和优化的地方。另外,BaaS 研发的门槛较高,只有大型公司和高收入的公司有能力负担,无论国内外,BaaS 几乎都是由商业巨头把控,行业的马太效应明显<sup>[7]</sup>。商业 BaaS 平台虽然提供了对开源区块链平台的支持,但 BaaS 本身是闭源的,对于 BaaS 发展和需要构建自己私有 BaaS 平台的企业来说都是不友好的。

针对上述问题,本文选择基于开源的 Hyperledger Fabric 联盟链和 Kubernetes 容器编排技术研究构建 BaaS 的底层核心技术,通过对 Fabric 源代码的解析和改造更好地适配 Kubernetes。此外学术界和产业界也在积极探索 Serverless Function 新技术和区块链的融合应用<sup>[8-9]</sup>,本文跟踪了相关内容,并提出一种基于函数计算思想改造 Fabric 的 BaaS 平台方案。

## 1 相关技术

### 1.1 BaaS 核心技术

BaaS 实质上是一种利用各种云计算技术打包封装指定区块链平台并对外提供一键部署和管理区块链能力的云服务。BaaS 有 PaaS 和 SaaS 两种类型,PaaS 提供部署指定区块链平台的能力,SaaS 则可以直接提供某种类型的区块链应用服务<sup>[10-11]</sup>。PaaS 类型的区块链服务是业界当前主流的区块链云化方案。在区块链服务层,本文调研了一些主流云计算公司的 BaaS 平台对区块链的支持情况,详细情况如表 1 所示。

### 1.2 Hyperledger Fabric

Hyperledger Fabric(简称 Fabric)是一个带有准入机制的企业级联盟链项目,它的前身是 IBM 的 OpenBlockchain 项目。目前 Fabric 已经成为企业区块链平台的典范,也是绝大多数厂商构建 BaaS 平台的首要选择。Fabric 是一种许可网络,而不是在没有身份的公共网络中建立分散的信任;Fabric 各参与方可以仅仅将需要共享的数据分享给他人,实现保密交易;

Fabric 采用可插拔的架构设计,各行各业可以针对特定需求量身定制各种插件<sup>[12]</sup>。

表 1 主要公有云的 BaaS 平台

Tab. 1 BaaS platform for major public clouds

BaaS 平台	区块链平台
AWS 区块链服务	Hyperledger Fabric, Ethereum
Azure 区块链服务	Ethereum
Google Cloud Platform	不支持
IBM 区块链即服务	Hyperledger Fabric
阿里云区块链服务	Hyperledger Fabric, 蚂蚁区块链, Ethereum
腾讯云区块链服务	Hyperledger Fabric, FISCO BC OS, Tencent
TBaaS	TrustSQL
华为云区块链服务 BCS	Hyperledger Fabric

### 1.3 Kubernetes 容器编排技术

Kubernetes 是一个用于自动部署、扩展和管理容器的开源平台,吸收了 Borg 系统的许多功能特性<sup>[13]</sup>。Kubernetes 字面意思为“舵手”,其图标对应为一个方向舵,另外 Kubernetes 单词中间有 8 个字母故也简称 K8s。Google 在大规模工作负载的运维管理上经历十几年先后构建了 Borg、Omega 以及 Kubernetes 等系统,并在 2014 年开源了 Kubernetes 系统。2017 年 Docker 公司宣布在其核心产品中内置 Kubernetes 服务,2018 年 Kubernetes 和容器成为所有云厂商的既定标准,以“云”为核心的软件研发思想逐步形成<sup>[14]</sup>。

### 1.4 Serverless Function 技术

Serverless 本意是指一种不需要专门服务器的技术,例如 P2P 网络无需设置服务器,因为每个节点既是服务器又是客户端;但在云环境下 Serverless 已经转变为一种开发人员不必关心服务器的设计思想、架构模式<sup>[15]</sup>。Serverless 有 BaaS (Backend as a Service)和 FaaS (Function as a Service)两种类型:BaaS 主要是指承载数据存储的服务,例如消息队列、内容分发网络(Content Delivery Network, CDN)、云对象存储等;而 FaaS 是指一种将用户自定义代码运行在无状态的计算容器中并由事件触发的计算服务<sup>[16]</sup>。

## 2 核心系统概要

### 2.1 系统功能

基于相关技术需求,设计了核心系统的相关功能,如表 2 所示。系统功能主要分为底层云计算平台、Fabric 静态组件部署管理、动态组件链码部署管理以及基于函数计算思想改造链码管理几个项目,每个项目下又划分了若干个功能点。

底层云计算平台是区块链网络的实际运行环境,它提供了计算、存储等资源。计算资源主要由 Kubernetes 平台管理,存储资源由分布式文件系统提供。作为底层支撑平台,它必须满足高可用的特性,保证在部分节点失效的情况下仍可有效提供服务,并能通过一定的运维手段恢复失效节点。上层区块链平台数据不断增长的存储需求,要求底层的存储资源能够动态扩展,本文使用分布式文件系统管理存储资源,通过增加存储设备实现存储资源的在线扩展。存储资源管理系统独立于容器计算平台,这使得底层平台还要为 Kubernetes 提供一个存储接入的功能,从而方便地为上层业务提供持久化存储。

BaaS 平台的一个基本的要求就是将区块链平台部署在云计算平台上。Fabric 组件根据不同的启动时间,可以分为

静态组件和动态组件两部分。静态组件指的是证书颁发机构 (Certificate Authority, CA)、Peer、Orderer 以及共识算法所依赖的外部组件,这些组件随区块链网络的启动而启动的。Fabric 动态组件链码,和其他组件的不同点在于它是通过实例化命令启动的。Fabric 项目提供了 Docker-Compose 的部署样例,只需做适当修改即可将静态组件部署到 Kubernetes 上。在实验环境中,链码和 Peer 服务部署在同一个宿主机上,链码镜像的编译、存储在同一个 Docker 服务中,可以方便地安装与实例化链码。但是在云计算环境下,链码可能会被调度到不同的计算节点上,所以跨节点实例化链码需要构建链码镜像的分发能力,实现链码镜像从 Peer 节点到部署节点的传输。最后一个项目仍是针对链码的,链码的函数计算改造,主要涉及对 Fabric 链码原有生命周期的改造,使其在调用前启动容器,调用完成即退出,让出计算资源。

表 2 核心系统功能  
Tab. 2 Core system functions

项目名称	功能细节
底层云计算平台	高可用 Kubernetes 平台 Kubernetes 接入网络存储
Fabric 静态组件部署管理	Peer、Orderer、CA 组件部署、共识算法依赖组件
动态组件链码部署管理	Kubernetes 部署管理链码 链码镜像存储、分发功能
函数计算思想改造	链码生命周期改造 链码容器编译功能 容器镜像分发功能

2.2 系统架构

BaaS 底层的区块链平台云化管理,可以分为云计算和区块链两层结构。本文所涉及的核心系统同样可以分为两层,底层为高可用 Kubernetes 平台和分布式文件系统,上层为 Fabric 区块链平台,系统架构如图 1 所示。

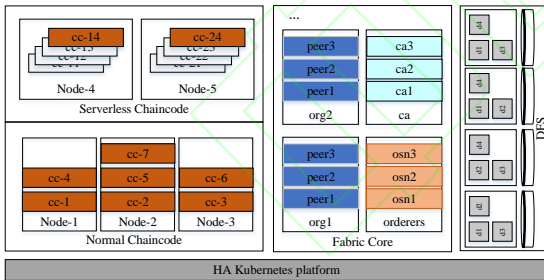


图 1 BaaS 底层架构

Fig. 1 BaaS Underlying Architecture

底层高可用 Kubernetes 系统是上层业务的托管平台,承载了上层所有的服务压力。作为一个容器管理平台, Kubernetes 要为上层 Fabric 提供高效的容器编排管理能力,将应用组件合理地分散到计算集群中健康的节点上。Kubernetes 具有高效的服务治理能力,利用系统提供的应用容器健康检测功能,可以选择相关策略自动恢复失效组件。凭借这些能力,开发者可以在 Kubernetes 平台上很方便地实施 Fabric 相关组件的高可用管理。同时, Kubernetes 的高可用也保证了在部分控制节点失效的情况下,仍能提供容器编排能力,确保上层服务的可靠运行。

区块链网络层位于高可用 Kubernetes 平台之上,这一层完成区块链所有的业务逻辑。Fabric 有静态组件和动态链码

两部分,这里的静态是指相关组件在部署网络时启动,动态主要是指链码在区块链业务运行阶段实例化时启动。动态和静态的划分,关系到 Fabric 网络在 Kubernetes 上的部署设计。在链码管理上,该架构增加了函数计算管理模块,改造后的链码容器共享计算节点,调用任务结束立即退出容器,让出计算资源。

该架构上还有一个独立于 Kubernetes 计算集群的分布式文件系统 (Distributed File System, DFS) 集群。以 HDFS (Hadoop Distributed File System) 为例介绍外部存储。HDFS 是一个分布式文件系统,它提供一种高可靠的、可动态扩展的数据存储能力。HDFS 接入到 Kubernetes 系统,可以为上层的区块链服务提供可动态扩展存储空间的持久化存储服务。Fabric 的账本数据最终存储在了可靠的分布式文件系统上。

2.3 系统流程

本文在链码部署管理上引入了函数计算思想,改变了链码容器原有的生命周期,部署流程如图 2 所示。引入函数计算的系统在链码实例化消息发出后, Fabric 静态组件 Peer 会完成链码容器镜像的编译,并将镜像上传到存储中心,但不会启动链码容器。函数计算下的链码容器是按需启动的,在区块链系统发起一笔交易产生链码调用时,才会触发链码容器的启动。链码容器在完成计算返回结果后,会立即结束其生命周期让出计算资源。

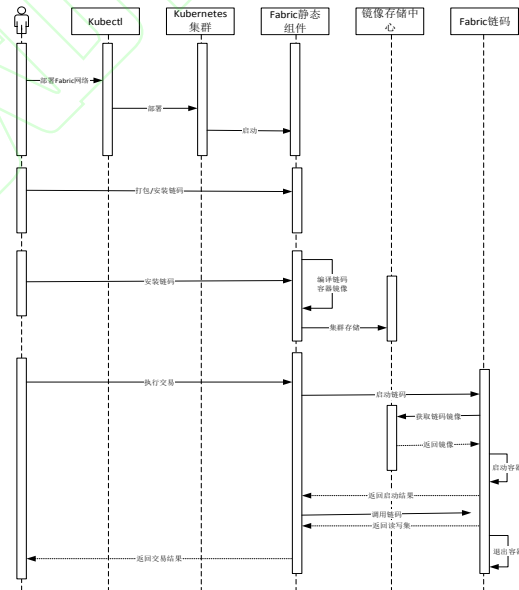


图 2 引入函数计算的 Fabric 链码部署流程

Fig. 2 Fabric Chaincode Deployment Process for Function Calculation

3 核心系统具体实现

3.1 高可用架构设计

Kubernetes 分为 Master 和 Node (Slave) 两个部分,其中 Node 的高可用由系统本身完成,保证将 Pod 调度到健康的工作节点上。Kubernetes 高可用主要是指 Master 组件的高可用。本文所设计的高可用架构如图 3 所示,可以发现集群外还增加了一个高可用负载均衡器,这个均衡器主要承担 Apiserver 的高可用任务。整个架构按照集群内外分为两部分:集群内多 Master 节点的高可用和集群外基于主备切换的负载均衡器高可用。Kubernetes 上 Kubelet 是通过 Systemctl 服务管理的,除它之外所有组件都可以直接部署在 Kubernetes 上,本文借

用这种能力来设计 Kubernetes 的高可用方案。

Master 节点上的 Etcd 组件本身就是一个基于 Raft 协议的高可用分布式键值数据库,只需为它配置多个实例即可。Master 上的其他组件,根据有无状态分为两种部署方案:无状态组件 Apiserver 采用“多实例+负载均衡器”的高可用设计;有状态组件 Scheduler 和 ControllerManager 采用“多实例+竞争锁”的高可用设计。集群外的负载均衡器采用“虚拟 IP+双节点”的主备设计,保证主代理服务失效后备份服务器能够立刻提供服务。同时负载均衡器还应具备主动健康检测功能,保证将流量分配到健康的服务器上。

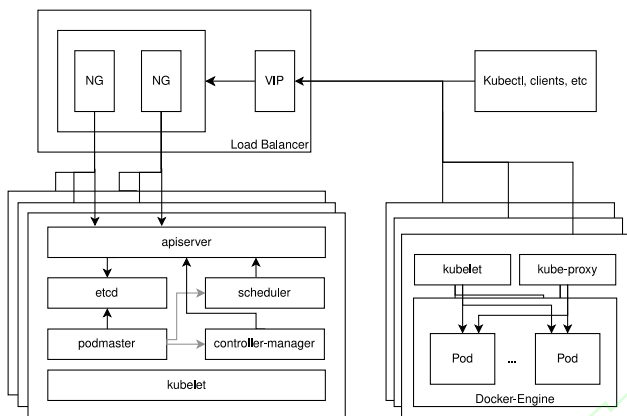


图3 Kubernetes高可用架构

Fig. 3 Kubernetes High Availability Architecture

图3中的 Apiserver 高可用依赖一个外部高可用负载均衡器,本节基于虚拟 IP、Keepalived、Nginx 等技术设计了一种双节点主备自动切换高可用负载均衡器,系统架构如图4所示。每台物理服务器上都需要安装 Keepalived 和 Nginx 服务。主备服务器上 Keepalived 需要配置物理网卡、虚拟 IP、state、认证信息及优先级,其中主服务器的 state 设置为 MASTER,备份服务器设置为 BACKUP。Keepalived 和虚拟 IP 完成 Nginx 主备切换的高可用,Keepalived 基于虚拟路由冗余协议实现了虚拟 IP 和服务器的自动绑定,初始状态主节点绑定虚拟 IP 提供服务。主节点失效,虚拟 IP 会自动漂移到备份服务器,此时由备份节点提供服务。负载均衡器还需要主动探测代理的 Apiserver 健康状态,保证将流量转发到可用的服务上。官方版本的 Nginx 提供被动健康检测方法,即目标服务异常,下次将流量分配到其他服务上,后端 Apiserver 异常,仍有机会被 Nginx 转发过来流量。本文基于 ngx\_healthcheck\_module 插件为 Nginx 提供了主动健康检测能力,保证将流量分配到健康的 Apiserver 上。

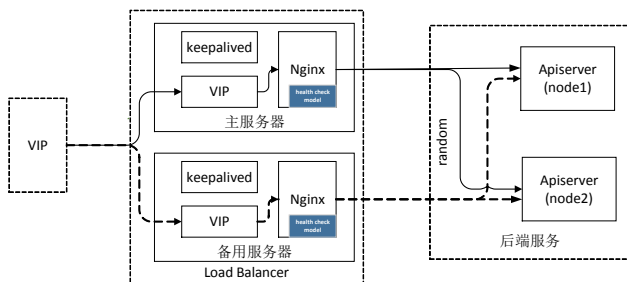


图4 高可用负载均衡器架构

Fig. 4 High Availability Load Balancer Architecture

### 3.2 Fabric 静态组件部署设计

Fabric 区块链平台是一个复杂的分布式系统,云化部署复杂度较高。本文将 Fabric 的组件分为了静态和动态两种,静态组件如 CA、Orderer、Peer 等构成了分布式系统的主要的基本框架。静态组件在 Fabric 网络部署时就可以启动完毕(不包括运行中动态加入节点)。图5描述了一个具备两个组织的部署架构,该架构中 Fabric 所有组件部署在 Kubernetes 上,所有资源通过 Kubernetes 进行管理。

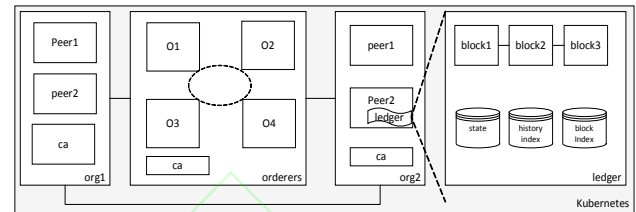


图5 Fabric云化部署架构

Fig. 5 Fabric Cloud Deployment Architecture

CA 是 Fabric 提供信任的基础,主要提供签发、管理证书以及身份识别等功能。CA 是一个 C/S 架构的服务,Server 端可以实现高可用扩展,但是必须共享存储以统一管理身份证书信息。Orderer 是 Fabric 的核心,也是最为复杂的组件之一。Fabric 的插件化设计支持多种算法,Orderer 是实现插件化共识的基本框架。Peer 节点是 Fabric 执行交易和账本管理的组件。交易执行意味着链码管理,Fabric 系统存在系统链码和用户链码两种类型,系统链码有 CSCC (Configuration System Chaincode)、LSCC (Lifestyle System Chaincode)、QSCC (Querier System Chaincode)、ESCC (Endorser System Chaincode)、VSCC (Validator System Chaincode) 五种分别完成通道配置、链码生命周期管理、交易查询、交易背书、交易验证等功能。

### 3.3 Fabric 链码部署设计

链码是区块链的核心,也是整个部署方案中最具挑战的地方。将链码纳入到 Kubernetes 的环境中是最理想的部署方案,但社区和业界的方案均没有实现这一点,这主要是因为 Fabric 的代码本身没有对这块内容的支持。本节所要介绍的部署方案就是要通过解构、二次开发 Fabric 代码将链码纳入 Kubernetes 环境。

Peer 服务在链码管理上完成了链码安装、Docker 镜像编译、创建容器、启动容器以及健康检测等工作。原生 Peer 服务,启动时要指定好 Docker Daemon 服务地址,随后链码实例化过程的镜像编译和创建均在同一个宿主主机上进行,但是 Kubernetes 是一个分布式的计算集群,链码容器会被调度到不同的节点上,也就是获得部署链码容器任务的宿主机可能没有链码镜像。通过链码镜像编译和容器创建分离,镜像编译仍在 Peer 服务绑定的 Docker Daemon 中进行,然后再将镜像分发到执行创建任务的节点上。

集群内镜像分发可以构建一个镜像仓库实现,高可用镜像仓库采用的 Harbor 的主从复制模式构建,同时为了降低集群内的网络带宽占用和提高镜像分发速度,基于 Dragonfly 构建 P2P 镜像下载加速器,相比原生方式,可将容器分发速度提高 57 倍,网络出口流量降低 99.5%<sup>[17]</sup>。引入镜像仓库后,Peer 和部署节点之间增加了向仓库 Push 和 Pull 镜像的两步操作,完整的工作流如图6所示。原生 Peer 通过 Docker SDK 向 Docker Daemon 发起镜像编译和创建容器的任务,现在链码实

例化分为镜像编译和创建两个过程:镜像编译仍通过Peer完成,但是要将镜像Push到集群内的镜像仓库;容器创建,则需要在Peer中集成Kubernetes的SDK完成容器的部署,获得部

署任务的工作节点从远程仓库拉取链码镜像并创建链码实例。

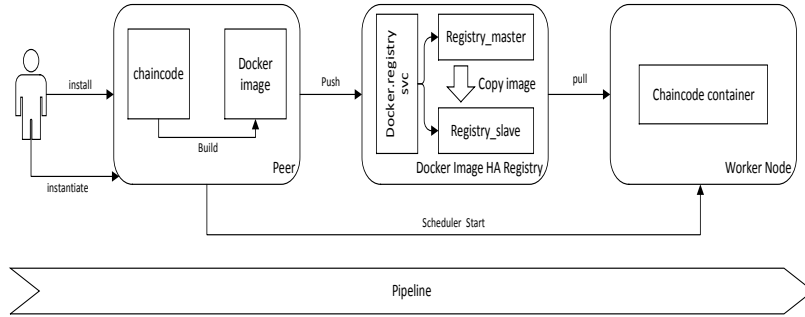


图6 Kubernetes环境中链码部署流

Fig. 6 Chaincode Deployment Streams in Kubernetes Environments

链码部署管理的改造涉及客户端部分:Peer服务端实现了多种链码容器管理插件,部署链码时需要客户端指明使用哪一个插件。Peer客户端提供解析配置文件的形式确定链码部署环境,具体是客户端配置文件的vm.type参数。客户端将链码使用的虚拟执行环境类型存储在了ChaincodeDeploymentSpec(CDS)结构中,而CDS是在链码安装

时确定的,所以还需要在链码安装部分加上对Kubernetes的支持。链码部分的改造实现拟全部封装在k8sontrroller包中,除了K8sDockerVM接口的实现外,还有如表3所示的一些Kubernetes客户端接口的封装实现。这些函数完成链码部署时的相关资源的创建。

表3 K8sDockerVM中的K8s接口  
Tab. 3 K8s interface in K8sDockerVM

主要功能	函数名
创建、删除	func (vm*K8sDockerVM)createConfigMap(clients*k8sClient.Clientset,name,ns string,data map[string]string)error
ConfigMap资源对象	func (vm*K8sDockerVM)deleteConfigMap(clients*k8sClient.Clientset,name,ns string,opts*metav1.DeleteOptions)error
创建、删除	func (vm*K8sDockerVM)createDeployment(clients*k8sClient.Clientset,deployment*appsV1.Deployment)error
Deployment资源对象	func (vm*K8sDockerVM)deleteDeployment(clients*k8sClient.Clientset,name,ns string,opts*metav1.DeleteOptions)error
创建、删除	func (vm *K8sDockerVM)createSVC(clients *k8sClient.Clientset,svc *apiv1.Service)error
Service资源对象	func (vm *K8sDockerVM)deleteSVC(clients *k8sClient.Clientset,name,ns string,opts*metav1.DeleteOptions)error

### 3.4 函数计算管理链码设计

根据区块链和函数计算服务的相似性,链码是Fabric上基于函数计算改造最具潜力的组件。Fabric中一笔交易的完成需要多数节点达成共识,并落到区块中,部分链码容器挂掉并不影响区块链整体网络。当有新的交易需要执行计算时,Fabric会再次拉起容器。Fabric的这项能力为实现链码容器函数计算化管理提供了可能。Fabric链码是一个只提供计算服务的组件,它的所有数据均来自于Peer服务,这种特性满足函数计算服务对代码无状态性的要求。Fabric链码的生命周期如图7所示,完成链码调用任务需要依次经历链码安装、实例化的过程。Peer服务端的链码容器在实例化后一直处于Running状态等待调用,只有在链码升级或者链码本身内部错误才会结束链码容器。而函数计算服务管理链码,首先要做的就是改变链码的生命周期,要求链码容器在调用时再实例化启动,调用结束即退出。

算节点从仓库中拉取镜像并启动实例,完成任务后退出。Fabric Peer本身具备链码镜像的编译能力,本文设计的Kubernetes部署方案也完成了集群内链码镜像的分发,所以只需改造链码实例的调度任务即可。

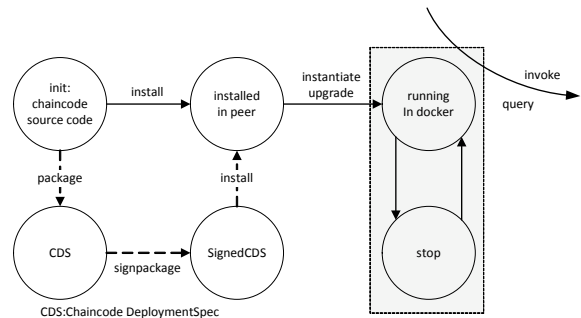


图7 Fabric Chaincode生命周期  
Fig. 7 Fabric Chaincode Lifecycle

无论函数计算服务还是区块链服务,部署用户代码都需要解决代码的编译问题。Fabric链码的编译是一个复杂的过程不能像函数计算服务一样提前准备好代码和依赖包直接上传,目前业界或者社区的函数计算服务无法完成Fabric链码的编译。鉴于实际情况,本文采用模拟函数计算服务的方法进行实验。函数计算服务的用户代码部署可以分为镜像编译、分发、部署等过程,以Knative函数计算服务为例,用户代码被编译成Docker镜像存储在远程镜像仓库中,事件触发计

通过在CDS结构中增加了Serverless字段和在Chaincode Install子命令下增加serverless bool全局参数,实现改变链码容器的运行模式和完成调度任务。引入Serverless参数的链码启动流程图如图8所示。引入Serverless的链码调用,会增加单次链码调用的处理时间,但它在空闲时让出了计算资源,可以极大地提高资源的利用率,多租户网络共享计算池可以显著降低区块链的部署运维费用,降低用户上链的费用

门槛,同时计算资源池对用户是透明的,共享资源的同时又保证了数据隐私性。

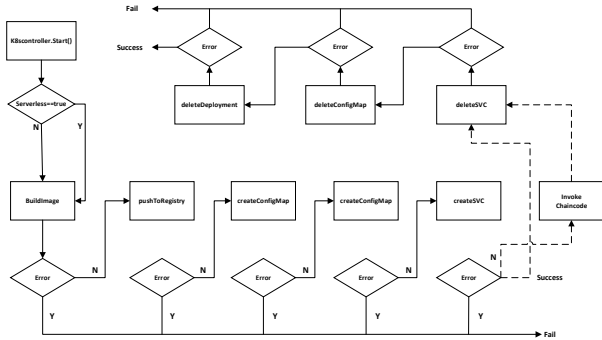


图8 链码启动简化流程

Fig. 8 Chaincode Activation Simplified Flowchart

3.5 链码性能测试

Kubernetes 环境和普通 Docker 环境中链码在镜像管理和运行环境方面存在区别。本节通过相同任务量下任务完成时间的快慢来评估改造后的链码调用性能,测试基准为基于 Docker 运行环境的链码完成时间。测试环境为双组织单节点部署环境,共识采用 Solo 算法。

测试方案,通过脚本封装链码调用,通过参数指定调用次数,并输出调用开始和结束的时间,每项测试 5 次,并求取平均值,实验结果如表 4 和表 5 所示。根据两个表中的数据,可以看出 Docker 和 Kubernetes 环境下,Kubernetes 部署的链码调用性能跟 Docker 链码相近。此外,统计了 Docker 和 Kubernetes 链码在 100 次调用下成功的次数分别为 18.4 和 19,二者调用的成功率也相近。为了不影响记时,测试脚本并未进行 Sleep 操作等待区块链网络数据的同步,所以链码调用的成功率普遍低一些。

表4 Docker 链码调用性能测试

Tab. 4 Docker Chaincode Call Performance Testing

测试次数	调用次数							
	1	2	3	4	5	10	50	100
1	0	1	1	2	2	4	21	44
2	1	1	1	2	2	4	20	42
3	0	1	2	3	3	4	23	49
4	1	1	1	2	3	5	21	44
5	0	1	1	2	2	4	21	43
平均值	0.4	1	1.2	2.2	2.4	4.2	21.4	44.4

表5 Kubernetes 链码调用性能测试

Tab. 5 Kubernetes chaincode call performance testing

测试次数	调用次数							
	1	2	3	4	5	10	50	100
1	0	1	1	2	2	4	21	44
2	0	0	1	1	2	4	22	43
3	0	1	1	2	3	4	23	49
4	0	0	1	1	2	4	21	42
5	1	1	3	3	4	6	23	45
平均值	0.2	0.6	1.4	1.8	2.6	4.4	22	44.6

此外,也对 Serverless 模式下的链码进行了测试,测试结果如表 6 所示。函数计算有一个明显的特点是冷启动时间较长,在这种模式下,链码在有调用需求时才会启动容器

的创建、启动花费了较长的时间。Serverless 链码调用时间较长,为区块链数据同步提供了比较充足的时间,实验测得链码调用成功率要比 Docker 和 Kubernetes 高,100 次调用下成功调用的平均值为 68.8。

表6 Serverless 链码调用性能测试

Tab. 6 Serverless chaincode call performance testing

测试次数	调用次数							
	1	2	3	4	5	10	50	100
1	16	32	48	65	73	158	813	1530
2	7	22	38	46	61	125	785	1600
3	8	16	31	47	62	125	747	1596
4	8	24	39	47	62	155	835	1574
5	7	23	38	69	76	156	766	1611
平均值	9	23	39	55	67	144	789	1582

4 结语

基于业界典型的 BaaS 方案,本文选择了 Fabric 和 Kubernetes 两项基本技术研究 BaaS 的底层实现。在 Kubernetes 方面,主要研究底层技术设施的高可用性;在 Fabric 方面,主要研究基于高可用 Kubernetes 的云化部署技术。

Fabric 链码的部署管理是本文研究的核心之一,也是 Fabric 云化部署方案中最复杂的一部分。现有的链码管理技术均没有将链码纳入 Kubernetes 环境中。本文通过对 Fabric Peer 的二次开发,通过新增一个 K8sDockerVM 控制器插件,实现了对 Kubernetes 在代码级别上的支持。此外,本文也通过函数计算思想对链码运行机制进行了一个创新尝试,实现了链码实例执行模式的改变,将实例化后链码容器常驻等待调用改变为有调用需求时启动。

函数计算引入区块链网络,可以为它带来几个能力:链码高度弹性部署,链码按需部署,用完即释放资源,可以降低大大降低成本。但函数计算服务的冷启动时间较长,由 Serverless 链码调用性能测试结果也可以看到,调用时间比常规部署方式的链码调用要久。对区块链网络来说,混合部署模式可能更加合适:高频应用,链码常驻提供高效的服务;低频应用,Serverless 模式部署,按需调用降低成本。混合部署及高频、低频应用区分方案是接下来区块链云化部署的研究方向之一。

参考文献 (References)

- [1] 可信区块链推进计划. 区块链即服务平台 BaaS 白皮书(1.0 版) [R/OL]. (2019-01-08) [2019-11-17]. <http://www.trustedblockchain.cn/schedule/detail/2989>. (Trusted Blockchain Advancement Program. Blockchain-as-a-Service Platform BaaS White Paper (Version 1.0) [R/OL]. (2019-01-08)[2019-11-17]. <http://www.trustedblockchain.cn/schedule/detail/2989>.)
- [2] Onik M M H, Miraz M H. Performance analytical comparison of Blockchain-as-a-Service (BaaS) platforms [C]// Proceedings of the 2019 International Conference for Emerging Technologies in Computing. Cham: Springer, 2019:3-18.
- [3] 刘楠,刘露. 区块链与云计算融合发展 BaaS 成大势所趋[J]. 通信世界,2017(17): 61-62. (LIU N, LIU L. Blockchain and cloud computing integration development BaaS becomes a major trend [J]. COMMUNICATIONS WORLD, 2017(17):61-62.)
- [4] 才丽. 面向 BaaS 平台的资源调度算法研究与实现[D]. 杭州:浙

- 江大学, 2018: 2-4. (CAI L. The BaaS Platform Orientated Research and Implementation for Scheduling Algorithm [D]. Hangzhou: Zhejiang University, 2018:2-4.)
- [5] 工业和信息化部信息中心. 2018年中国区块链白皮书[R/OL]. (2018-05-20) [2019-11-17]. <http://www.miit.gov.cn/n1146290/n1146402/n1146445/c6180238/part/6180297.pdf>. (Information Center of Ministry of Industry and Information Technology. 2018 China Blockchain White Paper [R/OL]. (2018-05-20) [2019-11-17]. <http://www.miit.gov.cn/n1146290/n1146402/n1146445/c6180238/part/6180297.pdf>.)
- [6] 赛迪区块链研究院. 2019年中国区块链发展形势展望[N]. 中国计算机报, 2019-03-04 (12). (Saidi blockchain Research Institute. China's blockchain development situation in 2019 [N]. China Information World, 2019-03-04(12))
- [7] 郝方舟, 李雪婷, 孔繁星. 2018年BaaS(区块链即服务)平台研究报告[R/OL]. (2019-03-26) [2019-11-17]. <https://www.odaily.com/post/5135837>. (HAO F Z, LI X T, KONG F X. BaaS (Blockchain as a Service) Platform Research Report 2018[R/OL]. (2019-03-26) [2019-11-17]. <https://www.odaily.com/post/5135837>.)
- [8] Chen H, Zhang L J. FBaaS: Functional Blockchain as a Service [C]// Proceedings of the 2018 Lecture Notes in Computer Science. Cham: Springer, 2018:243-250.
- [9] 段红. 一种基于区块链的高性能FAAS系统: 中国, CN109508432A[P/OL]. (2019-03-22) [2019-11-21]. <http://epub.sipo.gov.cn/pam.action>. (Duan H. A high-performance FAAS system based on blockchain: China, CN109508432A [P/OL]. (2019-03-22) [2019-11-21]. <http://epub.sipo.gov.cn/pam.action>.)
- [10] Singh J, Michels J D. Blockchain as a service (BaaS): Providers and trust [C]// Proceedings of the 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). Piscataway: IEEE, 2018: 67-74.
- [11] Wan Z, Cai M, Yang J, et al. A Novel Blockchain as a Service Paradigm [C]// Proceedings of the 2018 International Conference on Blockchain. Cham: Springer, 2018: 267-273.
- [12] CACHIN C. Architecture of the hyperledger blockchain fabric [C/OL]. Workshop on Distributed Cryptocurrencies and Consensus Ledgers. 2016 [2020-01-16]. <https://pdfs.semanticscholar.org/f852/c5f3fe649f8a17ded391df0796677a59927f.pdf>.
- [13] Oliveira C, Lung L C, Netto H, et al. Evaluating Raft in Docker on Kubernetes [C]// Proceedings of the 2016 International Conference on System Science. Cham: Springer, 2017:123-130.
- [14] 林琳. 为什么说2019年是云原生的关键节点[J]. 计算机与网络, 2019, 45(19): 38-40. (LIN L. Why 2019 is a critical point for cloud native[J]. 2019, 45(19): 38-40.)
- [15] Fox G C, Ishaklan V, Muthuamy V, et al. Status of Serverless Computing and Function-as-a-Service (FaaS) in Industry and Research [R/OL]. (2017-08-27) [2019-09-10]. <https://arxiv.org/abs/1708.08028>. DOI: 10.13140/RG.2.2.15007.87206.
- [16] 刘畅, 毋涛, 徐雷. 基于无服务器架构的边缘AI计算平台[J]. 信息通信技术, 2018, 12(5): 47-51. (LIU C, WU T, XU L. Edge AI computing platform based on serverless architecture [J]. 2018, 12(5): 47-51.)
- [17] 阿里云. 什么是 Dragonfly [P/OL]. (2016-04-20) [2020-12-19]. [https://d7y.io/zh-cn/docs/overview/what\\_is\\_dragonfly.html](https://d7y.io/zh-cn/docs/overview/what_is_dragonfly.html). (Alibaba cloud. What is Dragonfly [P/OL]. (2016-04-20) [2020-12-19]. [https://d7y.io/zh-cn/docs/overview/what\\_is\\_dragonfly.html](https://d7y.io/zh-cn/docs/overview/what_is_dragonfly.html).)

This work is partially supported by General Research Project of the Education Department of Zhejiang Province (Y202044224), Zhejiang Graduate Education Association Project (2020-002), Zhejiang Cultural Relic Protection Science and Technology Project (2019009).

**LIU Hongyu**, born in 1994, M. S.. His research interests include container cloud, blockchain.

**LIANG Xiubo**, born in 1983, Ph. D., associate professor. His research interests include blockchain, artificial intelligence, natural human-computer interaction, mobile Internet.

**WU Junhan**, born in 1998, M. S. candidate. His research interests include privacy protection and data storage of blockchain.