

基于以太网区块链的边缘计算资源分布式编排系统

张雪芹

(国网山西省电力公司,山西太原 030021)

摘要:随着物联网的快速发展,大量边缘设备产生的海量数据为雾计算应用带来了挑战。雾计算节点通常分布广泛,资源有限,对其进行集中管理存在困难,利用区块链技术和智能合约提出了一个去中心化的雾计算资源管理架构 HIDRA,该架构实现了雾节点之间的自治协作,通过在雾节点之间共享对等的区块链账本和执行智能合约,对雾计算环境进行应用编排和资源监控,无需中心化控制器的参与,在单板计算机组成的测试平台上对 HIDRA 进行了原型实现和评估,结果表明系统资源开销很小,能够在保证安全性和审计性的同时,有效地管理雾计算环境下的资源和应用,克服传统集中式方法的限制。

关键词:区块链;分布式系统;雾计算;物联网;资源编排;智能合约

中图分类号: TP311.13;TP393.08

文献标志码: A

文章编号: 1003-7241(2025)12-0130-04

Distributed orchestration system for edge computing resources
based on Ethernet block-chain

ZHANG Xueqin

(State Grid Shanxi Electric Power Co., Ltd., Taiyuan 030021, China)

Abstract: With the rapid development of the Internet of Things (IoT), the massive amount of data generated by a large number of edge devices poses a challenge for fog computing applications. Fog computing nodes are usually widely distributed with limited resources, and it is difficult to manage them centrally. A decentralized fog computing resource management architecture, HIDRA, is proposed using blockchain technology and smart contracts, which achieves autonomous collaboration among fog nodes, and performs application scheduling and resource monitoring of the fog computing environment through the sharing of a peer-to-peer blockchain ledger and the execution of smart contracts among fog nodes without the need for the involvement of a centralized controller, HIDRA is prototypically implemented and evaluated on a testbed consisting of a single-board computer, and the results show that the system has a small resource overhead and is able to efficiently manage resources and applications in a fog computing environment while ensuring security and auditability, overcoming the limitations of traditional centralized approaches.

Keywords: Blockchain; distributed system; fog computing; Internet of Things; resource orchestration; smart contracts

0 引言

随着物联网的快速发展,大量设备和传感器产生的海量数据为许多应用带来了挑战^[1-4]。国外方面,文献[5]最早提出了雾计算的概念,用于延伸云计算并解决其带宽与延迟的问题,文献[6]设计了基于雾计算的架构并应用于智能运输系统。国内方面,文献[7]研究了智能物联网在雾计算环境下的应用场景,文献[8]设计了一种基于雾的软件定义网络架构,文献[9]提出了一个雾驱动的智能运输系统。另外基于区块链技术的研究也取得了一定的进展,总体上国内外研究主要集中在雾计算框架与典型应用方面,当前仍缺乏涵盖面广、适用性强的雾计算资源管理方法与平台。基于此,本文提出 HIDRA 架构,一个基于区块链的去中心化雾计算环境编排架构,能够对雾节点资源进行自治管理,使用智能合约进行容器化应用编

排,并使用信誉机制识别问题节点。

1 提出的架构

1.1 系统概述

图1描绘了一个由三层组成的通用物联网架构:设备层托管物联网设备,分布式雾层 HIDRA 所关注的层,迷你数据中心和云提供商所在的云层。

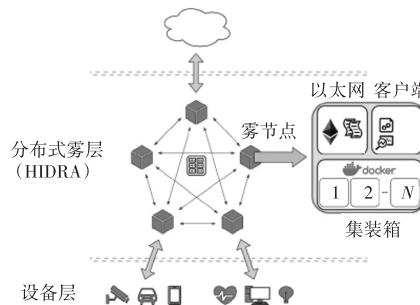


图1 系统概述

分布式雾层是一种紧邻物联网设备的云层扩展,位于

*基金项目:国网山西省电力公司科技项目(520530220007)

收稿日期:2024-04-30

网络边缘,旨在为各类终端设备提供更可靠、可扩展、快速的服务。

分布式雾层中的节点,通过区块链技术和底层点对点网络相连,共享并请求关于节点群集状态的信息。通过智能合约,这些节点之间建立了一系列必须遵守的规则。

区块链技术在这里扮演关键角色,维护一个分布式容器注册表,记录应用程序/容器的部署历史和实际运行状态。管理器处理区块链交互、雾节点上的轻量级虚拟化服务控制以及与节点群集状态相关的事件管理。监视器负责监控节点状态,当节点违反某些规则或策略时,它会向管理器发出警报。

1.2 关键组件和功能模块

1.2.1 区块链服务

HIDRA 利用以太网平台的区块链技术和底层 P2P 网络来管理和分发群集内的资源和服务。每个雾节点运行一个以太网客户端,负责处理数据和控制事件的发送、接收和存储。此外,以太网客户端利用 EVM 来执行智能合约,使得雾节点能够共享状态、同步全局状态、达成共识,以及共同决定资源和应用管理。考虑到设备的异构性,系统支持设备根据其能力执行完整或轻量级区块链服务,模拟以太网的完整和轻量级节点。完整服务存储所有区块链数据并参与共识事件,而轻量级服务只存储部分数据。

1.2.2 功能模块

智能合约在控制平面中实现了四个功能模块(图 2 所示),用于管理资源和应用编排。这些模块包括:分布式设备注册表(DDR)、分布式事件记录器(DEL)、分布式容器注册表(DCR)和分布式信誉系统(DRS)。它们存储在区块链上,确保所有节点都能访问和执行。

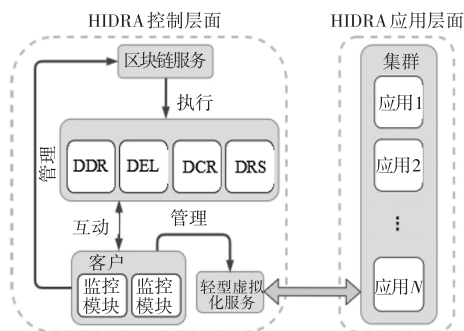


图 2 HIDRA 控制和应用层面

1.2.3 雾节点客户端

客户端负责运行区块链服务和连接控制平面的功能模块。它包括管理器和监视器模块,用于控制群集、执行规则和资源监控。客户端还负责管理雾节点和代理受限 IoT 设备的系统通信。管理器模块与区块链服务、智能合约和虚拟化服务交互,以实现协调。客户端在添加新节点时生成唯一的以太网地址,并可以与功能模块和智能合约互动。监视器模块监控资源。

1.2.4 轻型虚拟化服务

雾节点通过 Docker 服务和容器来管理软件应用,根据集群的决策进行编排。客户端通过 Docker 引擎 HTTP API 控制 Docker 服务,实现容器的创建、编辑和删除。在部署新容器时,客户端会检查本地是否存在相应镜像。客户端还可以配置镜像存储库来下载缺失的镜像。通过 Docker 与 DCR 模块结合,实现了集群中的全局容器状态共享。客户端作为各功能模块间的中间层,允许根据 DCR 状态调整本地容器状态。DEL 模块则支持向其他雾节点发送与容器相关的事件和元数据。

2 编排细节

2.1 主要工作流程协议

在 HIDRA 系统中,雾节点的互联和同步主要是通过一个精心设计的工作流程协议来实现(如图 3 所示)。该协议主要在雾节点不再处于稳定状态或遇到问题时触发。具体来说,雾节点可以根据系统需求处于不同的模式:常规模式、申请者模式和解决者模式。常规模式与节点的稳定状态相对应,而在监控模块发现不符合特定规则的问题时,节点会切换到申请者模式。此外,当某个节点被选为最适合为申请者节点提供解决方案的节点时,它则进入解决者模式。集群状态的更新在雾节点切换到申请者模式后自动开始,从而使申请者节点和常规节点共同努力解决问题。

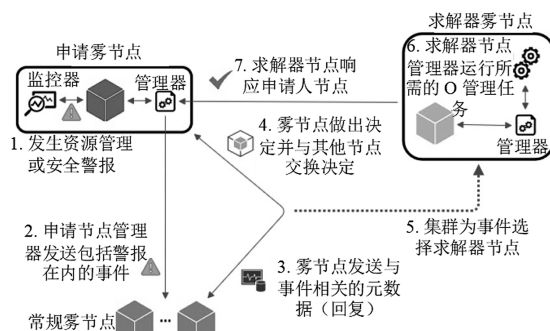


图 3 主要工作流程协议

轻量级虚拟化技术为应用提供隔离,简化资源编排,容器实例信息在区块链上进行记录,各模块协同工作实现集群范围内的容器管理。

2.2 功能模块详情

容器实例的信息被记录在区块链上,实现集群范围内的容器管理。分布式设备注册模块在系统中扮演重要角色,它不仅负责注册新设备,还存储用于监控和编排的静态元数据,如设备规格和集群配置。注册过程包括生成以太网地址、获取元数据,然后通过该地址进行注册。与此同时,分布式事件记录器模块执行主工作流程协议,监控动态元数据,并与预设规则进行比对。一旦触发规则,便会向管理模块发送警报。

管理智能合约是实现 DEL 工作流程逻辑的关键,包

含了发送 DEL 对象、选择解决者节点以及解决事件的过程。此外,所有与节点容器相关的注册信息都允许集群共享容器状态,管理器通过 Docker API 管理容器,并通过区块链事务通知状态变化。

2.3 客户端设计要点

在客户端设计方面,软件客户端主要包括监控器模块和管理器模块。监控器模块负责根据预设规则检测资源利用的异常情况,而管理器模块则基于监控信息执行容器编排任务。客户端还涵盖了以太网地址管理、交易模块、监控系统、规则配置、广播选择算法以及异步日志处理等关键功能。

3 评估和结果

3.1 原型规格

HIDRA 不受雾层资源类型限制,常见的雾节点实现方式是使用树莓派等单板计算机(SBCs)。原型由 $N = 3$ 个相同的 SBC 设备组成,即树莓派 4 型 B 型,配备 1.5 GHz 四核 CPU、4 GB 内存和 64 GB 磁盘存储,运行区块链服务、监控、软件客户端和轻量级虚拟化服务。每个雾节点运行以太网平台,充当完整的区块链服务,存储整个区块链,可在本地维护集群状态并执行智能合约。区块链服务部署了 Geth 1.9.16,用 Go 语言编写的以太网实现。私有区块链启动需要生成创世块,其中包含配置参数如默认封存节点、网络 ID 和块生成周期。软件客户端使用 Go 编程语言实现,与区块链服务兼容。智能合约使用稳固性编写,针对 EVM 进行编程。客户端监听稳固性事件,响应事件触发所需的管理任务。

3.2 提案验证

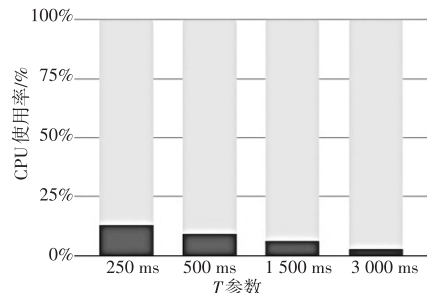
3.2.1 原型资源成本

首先将分析系统在稳定模式下运行时的资源使用情况,在监视模式下,监视客户端获取节点动态元数据,并且每个 T 时间间隔检查配置的规则,无需更改集群状态,因此不会生成或发送以太网交易。在监视模式中考虑的其他参数包括要检查的规则数 (R) 和以前的监视状态数 (S)。在默认值 $T = 1\,000$ 、 $S = 10$ 和 $R = 5$ 下获得的结果显示,控制平面存在低资源利用率现象(CPU 不到 7%,内存 20%,磁盘 10%),因此大多数雾节点资源可以分配给负载的执行。

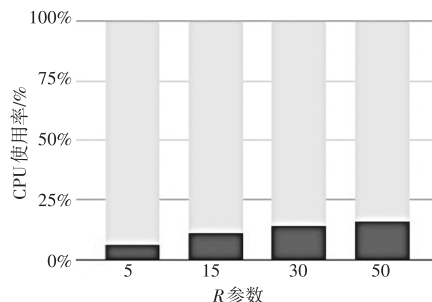
3.2.2 监测计算成本

雾节点在监测模式下支持不同的配置参数。图 4 显示了节点上的计算成本,具体取决于配置参数 T 和 R 的值。结果显示,监测间隔 T 越小,每次测量检查的规则数越多,雾节点的 CPU 使用率就越高。尽管可以观察到 CPU 使用率与参数 T 和 R 之间的线性关系,但结果显示了即使在不寻常的参数化值下,提案的可行性。在所有情况下,最大 CPU 使用率都不超过 20%。根据雾环境和用例的不同,可能需要不同的参数设置,以影响监视模块的操作,即使在同

一个用例中,每个雾节点的参数也可以根据其需求进行不同配置。参数 S 不影响雾节点的资源利用情况,尽管它会受影响发现节点中的异常利用值所需的时间。



(a) T 参数

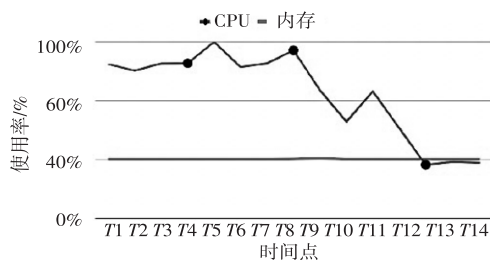


(b) R 参数

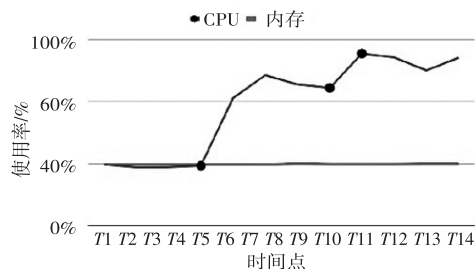
图 4 CPU 使用率对比

3.2.3 DEL 资源成本

本节分析了 DEL 容器迁移事件期间资源使用的成本,图 5(a) 和图 5(b) 分别显示了申请者 and 求解者雾节点在容器迁移事件期间的 CPU 和内存使用情况。结果显示了迁移期间内存使用稳定,但 CPU 使用率变化。



(a) 申请者节点



(b) 求解者节点

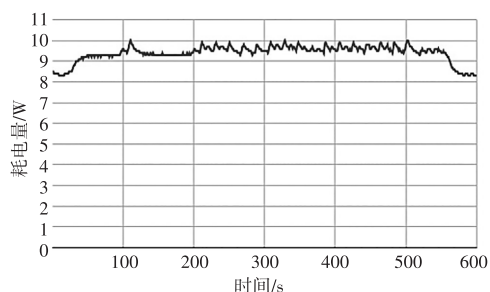
图 5 DEL 迁移过程中的资源使用情况

在申请者节点中,与 CPU 使用率相关的三个关键时间点可观察到, T_3 : 容器迁移事件生成; T_7 : 事件处理结束; T_{12} : 节点返回监测模式。申请者节点在事件解决前

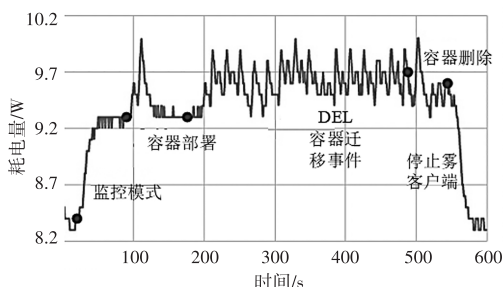
不会删除容器实例。在求解者节点,容器迁移期间 CPU 使用率也有关键时间点, T3:新容器开始部署;T7:部署完成,准备就绪;T8:容器处理请求, CPU 使用率增加。DEL 工作流程的总延迟约为 1.95 秒,主要是容器部署时间约 1.66 秒。最后,测量了容器迁移事件期间区块链服务资源的利用情况。

3.2.4 原型电力消耗

本节监测和分析了 HIDRA 原型部署的测试平台的电力消耗。在所有的实验中,监测频率被设置为每秒一次。实验包括在 10 分钟内测量电力消耗,其中每个雾节点最初部署了三个基于 Nginx 的 Web 服务器容器。在实验过程中,通过 DEL 容器迁移事件将部署的容器迁移到不同的集群节点之间。图 6(a)展示了实验期间测试平台的电力消耗情况。可以观察到电力消耗在整个实验过程中在每秒 8.3 W 和 10 W 之间变化,即相对于最低电力消耗最多增加了 20.5%。另一方面,图 6(b)放大了图 6(a)垂直轴的上部,以便更清楚地看到一些关键时刻和时间间隔。标有“监控模式”的时间间隔是指所有雾节点初始化其客户端的时期。当此时段结束时,雾节点正在运行区块链服务、轻量级虚拟化服务和监控其自身资源的软件客户端,但它们不部署容器或发送交易。在容器部署时间间隔开始时,每个节点上部署了三个 Nginx 容器。在此间隔期间,观察到了在部署过程中电力消耗的增加,但在容器部署后电力消耗保持稳定,在 DEL 容器迁移事件间隔期间,集群进行容器迁移,这涉及不断发送交易和部署/删除容器。为结束实验,每个雾节点上运行的容器被删除,客户端停止运行。总结结果,实验中每秒平均消耗 9.39 W 的电力,执行了 45 个 DEL 容器迁移事件。



(a) 整体视图



(b) 关键时刻和时间间隔

图 6 测试功耗

4 结束语

本文目标是研究基于区块链技术的分布式雾计算环境资源编排方法。依托分布式理论框架,设计了 HIDRA 架构,实现了基于区块链的去中心化网络,用于连接异构雾计算节点,并通过智能合约实现对节点的自治管理,包括应用编排、设备监测、信誉计算等,克服了传统中心化方法的缺陷。在原型实现中,设计了容器化应用的动态迁移机制,降低了移植成本,初步验证了方法的有效性。下一步工作将在网络规模效应的基础上,研究系统的可扩展性,同时从资源优化角度设计任务调度算法。

参考文献

- [1] 刘奇旭,刘嘉熹,靳泽,等. 基于人工智能的物联网恶意代码检测综述[J]. 计算机研究与发展, 2023, 60(10):2234-2254.
- [2] 李天明,严翔,张增年,等. 区块链+物联网在农产品溯源中的应用研究[J]. 计算机工程与应用, 2021, 57(23):50-60.
- [3] 耿明菲. 用于智能医疗监控的医疗大数据架构[J]. 信息技术, 2019(3):91-95, 101.
- [4] 臧宝志,郭亮,张志,等. 基于雾计算的电能计量箱物联网电子控制锁设计[J]. 自动化技术与应用, 2022, 41(2):187-191, 194.
- [5] BONOMI F, MILITO R, ZHU J, et al. Fog computing and its role in the internet of things[C]//Proceedings of the 1st MCC Workshop on Mobile Cloud Computing. Firenze, Italy: ACM, 2012.
- [6] AAZAM M, ZEADALLY S, HARRAS K A. Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities[J]. Future Generation Computer Systems, 2018(87):278-289.
- [7] 高微,陈新元,王榕国. 智能物联网 AIoT 的概念及应用场景的研究[J]. 信息通信技术, 2023, 17(3):80-84.
- [8] 杨守义,成昊泽,党亚萍. 基于集群协作的云雾混合计算资源分配和负载均衡策略[J]. 电子与信息学报, 2023, 45(7):2423-2431.
- [9] 杨斌. 煤矿带式输送机多电机驱动智能控制系统的设计与应用[J]. 机械管理开发, 2023, 38(5):182-184.

作者简介:张雪芹(1975—),女,硕士,高级工程师,研究方向:电网数字化建设与管理工作。