



# 计算机科学

COMPUTER SCIENCE

## 自学习星型链空间自适应分配方法

杜立宽, 刘晨, 王俊陆, 宋宝燕

### 引用本文

杜立宽, 刘晨, 王俊陆, 宋宝燕. [自学习星型链空间自适应分配方法](#)[J]. 计算机科学, 2025, 52(3): 359-365.

DU Likuan, LIU Chen, WANG Junlu, SONG Baoyan. [Self-learning Star Chain Space Adaptive Allocation Method](#) [J]. Computer Science, 2025, 52(3): 359-365.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

### [基于深度强化学习的Windows域渗透攻击路径生成方法](#)

Windows Domain Penetration Testing Attack Path Generation Based on Deep Reinforcement Learning  
计算机科学, 2025, 52(3): 400-406. <https://doi.org/10.11896/jsjcx.231200074>

### [基于图强化学习的多边缘协同负载均衡方法](#)

Graph Reinforcement Learning Based Multi-edge Cooperative Load Balancing Method  
计算机科学, 2025, 52(3): 338-348. <https://doi.org/10.11896/jsjcx.240100091>

### [融合关系模式和类比迁移的知识图谱补全方法](#)

Joint Relational Patterns and Analogy Transfer Knowledge Graph Completion Method  
计算机科学, 2025, 52(3): 287-294. <https://doi.org/10.11896/jsjcx.240700156>

### [面向云数据中心基于改进A2C算法的任务调度策略](#)

Task Scheduling Strategy Based on Improved A2C Algorithm for Cloud Data Center  
计算机科学, 2025, 52(2): 310-322. <https://doi.org/10.11896/jsjcx.240500111>

### [面向数字孪生的混合业务确定性传输调度机制](#)

Deterministic Transmission Scheduling Mechanism for Mixed Traffic Flows Towards Digital Twin Networks  
计算机科学, 2024, 51(12): 37-45. <https://doi.org/10.11896/jsjcx.240200063>

# 自学习星型链空间自适应分配方法

杜立宽 刘晨 王俊陆 宋宝燕

辽宁大学信息学院 沈阳 110031

(dulikuan2022@outlook.com)

**摘要** 区块链分片技术是提高区块链系统吞吐量的有效方法。现有的区块链分片方法大多采用静态平行架构分片方案,未解决跨分片交易比例高的问题,导致吞吐量降低以及潜在的无限交易确认延迟。针对这些问题,提出一种基于自学习的星型链空间自适应分配架构。首先,针对区块链分片系统中跨分片交易比例高的问题,提出一种信标链-分片链架构吞吐量模型;其次,综合分片区块链的吞吐量和时延的关系,在星型链空间的动态决策过程中设计星型链空间奖励函数;最后,提出一种分布式多智能体强化学习动态聚类方法,将每个分片作为智能体共同学习合作策略。实验结果表明,所提方法在吞吐量、跨分片交易比率和交易确认延迟等方面,相比现有方法分别约提升 31.74%,35.96%和 37.13%。

**关键词:** 区块链;分片;深度强化学习;跨分片交易

**中图分类号** TP314

## Self-learning Star Chain Space Adaptive Allocation Method

DU Likuan, LIU Chen, WANG Junlu and SONG Baoyan

School of Information, Liaoning University, Shenyang 110031, China

**Abstract** Blockchain sharding technology is an effective method to improve the throughput of blockchain systems. Existing blockchain sharding methods mostly adopt parallel architecture sharding schemes, which have not solved the problem of high cross-shard transaction ratios, leading to reduced throughput and potential infinite transaction confirmation delays. To address these issues, a self-learning-based star chain space adaptive allocation structure is proposed. Firstly, to address the issue of high cross-shard transaction ratios in blockchain sharding systems, a beacon chain-shard chain architecture throughput model is proposed. Secondly, considering the relationship between the throughput and latency of sharded blockchain, a star chain space dynamic decision-making process is designed, with a reward function for star chain space. Finally, a distributed multi-agent reinforcement learning dynamic clustering method is proposed, treating each shard as an agent to collectively learn cooperative strategies. Experimental results show that the proposed method improves throughput, cross-shard transaction ratio, and transaction confirmation delay by approximately 31.74%, 35.96%, and 37.13% respectively, compared with existing methods.

**Keywords** Blockchain, Sharding, Deep reinforcement learning, Cross-shard transaction

## 1 引言

近年来,区块链技术已经从最初的数字货币发展到医疗<sup>[1]</sup>、数字资产交易<sup>[2]</sup>等多个领域<sup>[3]</sup>。随着链上数据规模的指数级增长,各领域对区块链的吞吐量提出了新的要求,区块链分片技术开始受到各领域专家学者的广泛关注<sup>[4]</sup>。与传统的数据库分片不同,区块链分片是对区块链网络进行分区,允许区块链网络各部分并行处理事务,同时减少区块链系统的

堵塞,降低交易的代价,具有较好的水平拓展性<sup>[5-6]</sup>。然而,现有的区块链分片方法大多采用平行架构静态分片,无法在保持高吞吐量的同时降低跨分片交易负载。因此,如何对区块链进行高效分片,成为各领域的研究热点和难点<sup>[7-12]</sup>。

针对上述问题,本文提出一种星型链空间自适应分配架构(Adaptive Algorithm of Atar Sharding structure, A3S)。通过结合价值函数学习和策略学习的多智能体强化学习技术,基于各个智能体之间的协作实现动态分配分片。本文的主要

到稿日期:2024-07-22 返修日期:2024-09-30

基金项目:国家重点研发计划(2021YFF0901004);辽宁省教育厅高校基本科研项目(理工类)面上项目(揭榜挂帅服务地方项目)(JYTMS20230761);辽宁大学青年科研基金项目(LDYBJB2301);辽宁省应用基础研究计划(2022JH2/101300250)

This work was supported by the National Key Research and Development Program of China(2021YFF0901004), General Project(Service Local Project of Unveiling and Recruiting Talents) of Basic Scientific Research Project of Colleges and Universities of Liaoning Provincial Department of Education(Science and Engineering)(JYTMS20230761), Liaoning University Youth Scientific Research Fund Project(LDYBJB2301) and Liaoning Provincial Applied Basic Research Program(2022JH2/101300250).

通信作者:宋宝燕(bysong@lnu.edu.cn)

贡献如下:

1) 针对区块链分片系统中跨分片交易比例高的问题, 构建星型区块链性能约束模型, 提出一种信标链-分片链架构吞吐量模型 (Beacon chain-Shard Chain Throughput Model, BSTM)。

2) 在此基础上, 综合分片区块链的吞吐量和时延的关系, 在星型链空间动态决策过程中设计星型链空间奖励函数。

3) 提出一种分布式多智能体强化学习动态聚类方法 (A Distributed-based Dynamic Clustering Method for Multi-Intelligent Deep Reinforcement Learning, DDCM), 将每个分片作为智能体共同学习合作策略, 并引入 MA-DRL 解决集中学习和分散执行问题, 以优化系统效用。

4) 在以太坊真实数据集上进行仿真实验。结果表明, 对比现有区块链分片方法, 本文提出的 A3S 算法能够提高 31.74% 的区块链吞吐量, 同时降低 35.96% 的跨分片交易比例以及 37.13% 的交易确认时长。

许多学者对区块链分片方法进行了深入研究, 并取得了一定的研究成果<sup>[13-15]</sup>。目前, 区块链分片方法主要采取平行架构静态分片。片内交易直接经过片内共识完成, 跨片交易在分片链之间直接交互, 各个分片独立维护自身状态。

Elastico<sup>[16]</sup> 使用随机数生成协议将节点随机分配到固定大小的分片, 每个分片独立运行共识协议并生成微区块, 最后由全局委员会将所有分片的微区块组合成全局区块。但 Elastico 的分片是静态的, 固定分片的节点会使分片内负载过高或产生过多跨分片通信, 从而导致交易确认延迟增加。DRLPB<sup>[17]</sup> 是一个基于分片的可扩展区块链系统, 其采用结构化 DAG 方法实现分片, 实现提议区块、交易区块和投票区块的并行处理。DAG 方法实质上是一种静态分片方法, 最终导致交易确认时延较高。OmniLedger<sup>[18]</sup> 是一种身份链以及多条子链构成的分片架构, 通过 RandHound 协议将所有节点分成不同的组。每个分片独立运行共识协议处理交易, 并通过 Atomix 协议处理跨分片交易。其平行分片结构导致复杂的跨分片协议, 增加网络的通信开销以及交易处理延迟。此外, 当某些分片的负载明显高于其他分片时, 高负载分片会成为系统瓶颈, 限制整体吞吐量。SkyChain<sup>[19]</sup> 是一种区块链动态分片框架, 通过 DRL 技术实现对分片策略的自适应调整; 采用自适应账本协议, 确保账本能够根据分片策略的动态变化高效且无冲突地合并或分割。其仍然是平行分片结构, 多个分片之间进行跨分片交易时, 跨分片交易需要在多个链之间进行确认和同步; 需要额外的通信来确保交易的原子性和一致性, 对交易确认延迟产生很大影响。为了解决这些问题, A3S 通过提供不依赖于可信第三方的分片机制, 在提高区块链系统性能的同时减小 CST 开销, 解决区块链分片系统内交易确认时间过长的问题。

本文第 2 章介绍信标链-分片链架构吞吐量模型、星型分片动态决策过程; 第 3 章描述自学习模型构建以及具体的星型链空间动态分配方法; 第 4 章进行实验评估, 并对实验结果进行分析; 最后总结全文并展望未来工作。

## 2 融合星型架构的系统性能模型的构建

针对现有方法存在的跨分片交易比例高的问题, 提出一种改进星型架构性能模型, 通过构建信标链-分片链架构, 使信标链承载交易最终确认, 分片链维护并同步信标链与分片内的数据。

首先, 建立星型架构性能模型约束。

1) 对交易进行分片存储, 每个分片负责特定交易, 减少节点的网络负载。

2) 分片内节点间交易, 此类交易属于片内交易。

3) 两个节点进行交易但分别在不同的分片中。此类交易属于跨片交易, 需要通过信标链进行中转。信标链需要负责交易的转发, 以及交易确认。

### 2.1 BSTM 构建

基于信标链-分片链结构, 本文提出一种信标链-分片链架构吞吐量模型 BSTM。定义每种分片内包含 3 种需要处理的交易信息。

1) 片内交易;

2) 跨分片交易, 交易的发起者在本分片上;

3) 跨分片交易, 交易的接收者在本分片上。

假设  $t_{\text{intra}}$ ,  $t_A$  和  $t_B$  分别是以上 3 种交易的处理时间,  $t_{\text{mainchain}}$  是信标链处理交易的时间。对区块链交易过程涉及的变量做如下定义。

情况 1 每产生一笔交易, 该交易属于分片  $i$  的概率是:

$$\theta(i) = P(\text{shardID} = i), i = 1, 2, \dots, N, \sum_{i=1}^N \theta(i) = 1 \quad (1)$$

情况 2 每产生一笔交易, 该交易属于分片  $i$ , 且交易的目的账户属于分片  $j$  的概率为:

$$\gamma(j, i) = P(\text{receiveID} = j | \text{shardID} = i), i = 1, 2, \dots, N; j = 1, 2, \dots, N, \sum_{j=1}^N \gamma(j, i) = 1 \quad (2)$$

设一个时间段内总共产生  $M$  个交易, 第  $i$  个分片产生的交易数量为  $M_i = M \cdot \theta(i)$ , 那么 3 种交易的处理时间如下。

1) 片内交易。分片  $i$  中, 处理一个区块所需的时间为  $T_b = \frac{B_s}{N_c}$ , 其中  $B_s$  指分片中区块的大小,  $N_c$  指分片内节点的计算能力。信标链中, 处理一个区块所需的时间为  $T_b^* = \frac{B_s^*}{N_c^*}$ , 其中  $B_s^*$  指分片中区块头的大小,  $N_c^*$  指信标链中节点的计算能力。

2) 跨分片交易。交易发送方账户在分片  $i$  上, 每笔该种交易的处理时间  $T_c$  如式(3)所示, 其中  $N_c^r$  与  $N_c^s$  分别指发送节点与接收节点的计算能力。

$$T_c = \max\left(\frac{B_s}{N_c^r}, \frac{B_s}{N_c^s}\right) \quad (3)$$

3) 跨分片交易。交易接收方账户在分片  $i$  上, 每笔该种交易的处理时间同样为  $T_c$ 。随着分片数量  $N$  的增加, 星型分片架构下的系统的吞吐量变化会出现以下 2 种情况。

情况 1 当跨分片交易量远小于信标链的处理能力时,  $M$  笔交易处理完的时间等价于耗时最长的分片处理交易所需的时间。

对于分片内交易, 分片  $i$  内的交易时间开销为:

$$t_{\text{intra}}^i = M_i \cdot \eta(i, i) \cdot \frac{B_s^i}{N_c^i} \quad (4)$$

对于跨分片交易,若交易的发起者在分片  $i$  上,则系统中交易的开销为:

$$t_A^i = M_i \cdot [1 - \eta(i, i)] \max\left(\frac{B_s^i}{N_c^i}, \frac{B_s^j}{N_c^j}\right) \quad (5)$$

若交易的接收者在  $i$  分片上,则交易开销为:

$$t_B^i = \sum_{i \neq j} M_i \cdot \theta(i) \eta(i, j) \max\left(\frac{B_s^i}{N_c^i}, \frac{B_s^j}{N_c^j}\right) \quad (6)$$

每一个分片的处理时间可以表示为:

$$T_{\text{shard}} = T_{\text{intra-ShardA}} + T_{\text{inter-ShardA}} + T_{\text{inter-ShardB}} \quad (7)$$

则分片  $i$  的交易处理时间为:

$$t_i = t_{\text{intra}}^i + t_A^i + t_B^i \quad (8)$$

单位时间内处理交易的吞吐量为:

$$TPS_1 = \frac{M}{\max(t_i)} \quad (9)$$

情况 2 当跨分片交易量超过了信标链的处理能力时,区块链系统交易处理的总时间  $t_{\text{total}}$  为信标链处理交易的时间  $t_{\text{mainchain}}$ ,如式(10)所示:

$$t_{\text{total}} = t_{\text{mainchain}} \quad (10)$$

在信标链-分片链结构中,信标链上的交易串行处理。当所有的交易都是跨分片交易时,则  $t_{\text{mainchain}}$  为:

$$t_{\text{mainchain}} = M \left[ 1 - \sum_{i=1}^N \theta(i) \eta(i, i) \right] T_b^* \quad (11)$$

每一个交易都在不同的区块中,即每一个跨分片交易都需要转发一个区块头,则单位时间内系统处理交易的吞吐量为:

$$TPS_2 = \frac{M}{t_{\text{mainchain}}} = \frac{M}{M \left[ 1 - \sum_{i=1}^N \theta(i) \eta(i, i) \right] T_b^*} \quad (12)$$

结合式(9)和式(12)可以推导出信标链-分片链架构的吞吐量表达式:

$$TPS = \begin{cases} TPS_1 \\ TPS_2 \end{cases} \quad (13)$$

## 2.2 星型分片动态决策过程

基于 2.1 节的 BSTM,通过将吞吐量最大化问题转化为马尔可夫决策中的奖励最大化问题,实现星型分片动态决策过程。

**定义 1(四元动态决策结构)** 设  $\langle N, S, A, O \rangle$  表示马尔可夫博弈过程。集合  $N = \{1, 2, \dots, N\}$  表示智能体。每个分区都有 1 个节点负责分配资源,调整分片,该节点作为智能体。

**定义 2(离散状态空间)** 集合  $S = \{s_1, s_2, \dots, s_N\}$  表示  $N$  个智能体的状态空间。在离散时间  $t$  时刻,定义离散的状态空间  $S^t = \{M^t, T^t, C^t\}$ 。

**定义 3(信标链-分片链架构环境)** 集合  $O = \{o_1, o_2, \dots, o_N\}$  表示  $N$  个智能体所观测到的环境。

本文将信标链-分片链网络作为所有节点所共享的环境。将智能体的动作作为输入,并输出状态。

**定义 4(信标链-分片链架构行动空间)** 通过对区块大小、节点数量和分片数量进行选择,来实现对区块链网络中节点的分片。集合  $\mathcal{A} = \{a_1, a_2, \dots, a_N\}$  表示  $N$  个智能体的行动空间。本文定义离散时间  $t$  时刻的离散状态空间为:  $\mathcal{A}^t = \{N^t, B^t, G^t\}$ 。

**定义 5(奖励函数)** 通过定义最大化长期奖励来训练深度强化学习的智能体,一段时间  $T$  内整个系统的奖励是吞吐量与系统开销之差。星型分片区块链的性能指标主要指吞吐量和时延两方面。定义奖励函数为:

$$R(S(t), A(t)) = TPS(t) - (\Omega_{i,\epsilon}^{\text{PoW}} + \Omega_i^{\text{Comm}}) \quad (14)$$

其中,  $\beta = [1 - \eta(i, i)] + \theta(i) \cdot \eta(i, j)$  是跨分片事务所占比例,  $N$  表示分片的数量,  $N_{\text{wait}}$  表示一个时间段内没有挖矿活动的分片数,  $\hat{\Omega}_i^{\text{Comm}}$  表示挖矿一轮的单位通信开销,  $(N - N_{\text{wait}}) K \hat{\Omega}_i^{\text{Comm}}$  表示分片内通信开销,  $(N - N_{\text{wait}})^2 \beta K \hat{\Omega}_i^{\text{Comm}}$  表示跨分片通信开销。星型分片结构中达成共识所消耗的算力可以表示为:

$$\Omega_{i,\epsilon}^{\text{PoW}} = K \hat{\Omega}_{i,\epsilon}^{\text{PoW}} \quad (15)$$

其中,  $\hat{\Omega}_{i,\epsilon}^{\text{PoW}}$  是可信节点  $i$  在一个 episode 内的算力开销;  $K$  是分片中的区块被共识次数;  $\hat{\Omega}_{i,\epsilon}^{\text{PoW}}$  是分片  $i$  在一个 episode 内的共识计算开销,且  $\hat{\Omega}_{i,\epsilon}^{\text{PoW}} = (1 - \delta_i^t) \Pi_i^{\text{PoW}}$ , 其中  $\delta_i^t$  是可信节点  $i$  在一个 episode 所留用的算力的比例,  $\Pi_i^{\text{PoW}}$  是挖矿节点  $i$  的总算力。因此,定义奖励函数为:

$$R(S(t), A(t)) = TPS(t) - (\Omega_{i,\epsilon}^{\text{PoW}} + \Omega_i^{\text{Comm}}) \quad (16)$$

## 3 链空间节点动态聚类方法

基于星型分片动态决策过程,本文综合吞吐量和确认时延性能指标,提出链空间动态分配方法。如图 1 所示,  $N$  个智能体负责维护全部的交易池。  $N$  个智能共同通过 DDCM 算法将系统中的挖矿节点进行分片。在每一轮共识结束之后,重新应用 DDCM 算法动态调整分片。

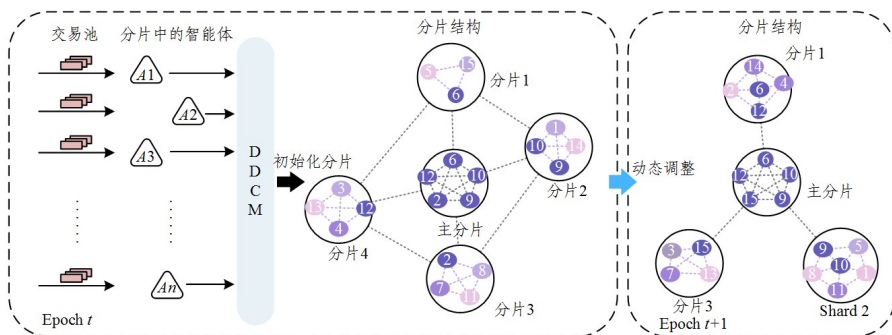


图 1 星型链空间自适应分配架构

Fig. 1 Star chain space adaptive fragmentation architecture

### 3.1 自学习模型构建

为解决星型区块链分片的动态调整问题,提出一种分布式多智能体强化学习动态聚类方法 DDCM。如图 2 所示,  $N$  个智能体、信标链的中心服务器共同构成一个共享的学习环境。首先,各个分片中智能体  $\{1, 2, \dots, N\}$  将局

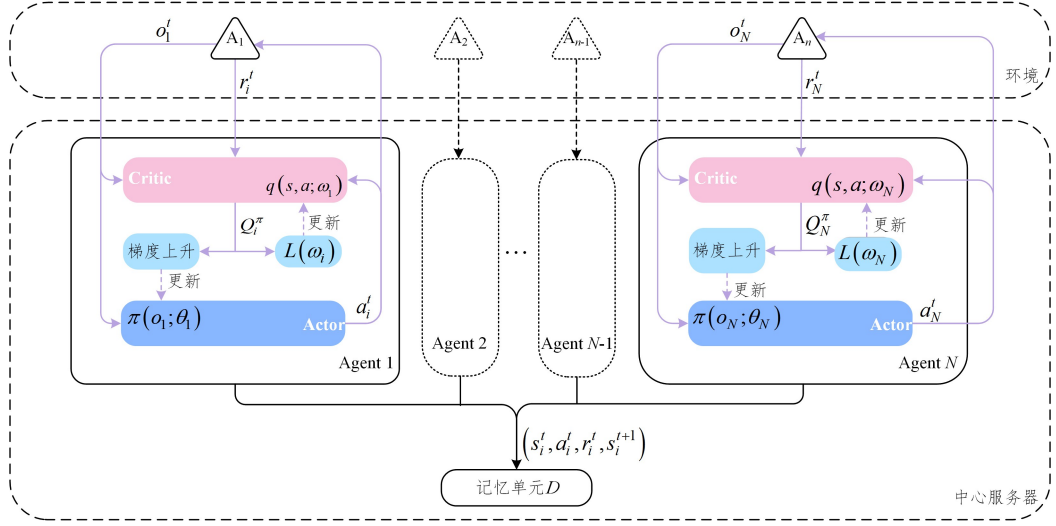


图 2 DDCM 模型示意图

Fig. 2 DDCM model diagram

### 3.2 DDCM 算法

DDCM 在多智能体环境中的连续动作空间上运行,每个智能体对应一个策略网络和价值网络。对于智能体  $i$ , actor 通过策略网络  $\pi(o_i; \theta_i)$  生成一个确定的动作  $a$ , critic 通过价值网络  $q(s, a; \omega_i)$  评估 actor 生成的行为。此外,信标链上的中央控制器还设有目标策略网络与目标价值网络,分别记为  $\pi(a_i | o_i; \theta_i^-)$  与  $q(s, a; \omega_i^-)$ 。第  $i$  号智能体当前的参数为  $\theta_i^{\text{now}}, \omega_i^{\text{now}}, \theta_i^{\text{old}}, \omega_i^{\text{old}}$ 。  $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$  表示各智能体代理的集合,  $\theta = \{\theta_1, \theta_2, \dots, \theta_N\}$  是各个策略网络的参数集合。在集中式的训练结束之后,每个智能体更新其本地模型的参数  $\theta_i$  来训练最优策略  $\pi_{\theta_i}^* = \arg \max_{\theta_i} J(\theta_i)$ , 其中  $J(\theta_i)$  作为智能体  $i$  的目标函数。

在训练策略网络中, Critic 通过  $q(s|a; \omega_i^-)$  拟合动作价值函数  $Q_i^\pi(o, a_1, a_2, \dots, a_N)$ , 通过最小化 Loss 函数的方式来更新  $Q$  函数。定义损失函数为:

$$L(\omega_i) = \mathbb{E}_{o_i^t, a_i^t, r_i^t, o_i^{t+1}} [(\hat{y}_i - Q_i^\pi(o_i^t, a_i^t, a_2^t, \dots, a_N^t))^2]$$

$$\hat{y}_i = r_i + \gamma Q_i^\pi(o_i^{t+1}, a_1^{t+1}, a_2^{t+1}, \dots, a_N^{t+1}) |_{a_i^{t+1} = \pi_i^{t+1}(o_i)} \quad (17)$$

其中,  $\hat{y}_i$  是 TD 目标,  $\pi_i^{t+1}(o_i)$  指带有延迟参数  $\theta_i$  的目标策略。 Actor 通过计算目标网络生成的目标函数  $J(\theta_i)$  的梯度来更新行为策略网络, 其梯度如下:

$$\nabla_{\theta_i} J(\pi_i) = \mathbb{E}_{o_i, a_i \sim D} [\nabla_{\theta_i} Q_i^\pi(o_i^t, a_1^t, a_2^t, \dots, a_N^t) |_{a_i = \pi_i(o_i)} \cdot \nabla_{\theta_i} \pi_i(a_i | o_i)] \quad (18)$$

其中,  $o = \{o_1, o_2, \dots, o_N\}$  为观测集;  $Q_i^\pi(o_i^t, a_1^t, a_2^t, \dots, a_N^t) |_{a_i = \pi_i(o_i)}$  是智能体  $i$  集中式训练价值网络得到的动作价值函数,  $i \in N$ ;  $a_1, a_2, \dots, a_N$  分别是  $N$  个智能体的动作, 这些动作是独立训练的;  $D$  是训练经验回放记忆单元, 包含多个事件样本  $(o_i^t, a_i^t, r_i^t, o_i^{t+1})$ ; 策略网络中使用梯度上升算法来更新  $\theta_i$ , 使得智能体  $i$  可以学习得到最佳策略  $\pi_i^*(o_i)$ 。

部信息上传到信标链, 经过集中训练后将策略下载到本地执行。

如图 2 所示, 为了训练第  $i$  号策略网络和第  $i$  号价值网络, 需要用到如下信息: 从经验回放单元  $D$  中取出的  $(s_i^t, a_i^t, r_i^t, s_i^{t+1})$ 、 $N$  个策略网络, 以及第  $i$  号价值网络。

$$\theta_i^{\text{new}} \leftarrow \theta_i^{\text{now}} - \beta \cdot \nabla_{\theta_i} \mu(o_i^t; \theta_i^{\text{now}}) \cdot \nabla_{a_i} q(s^t, \hat{a}_i^t; \omega_i^{\text{now}}) \quad (19)$$

DDCM 的详细流程见算法 1, 主要包括 2 个阶段: 初始化阶段和更新阶段。在初始化阶段, 使用  $\epsilon$ -greedy 策略更新动作价值函数  $Q$ , 每个智能体将执行动作所得的奖励存储在缓冲池  $D$  中(第 8—10 行)。在当前动作执行结束之后, 智能体将会执行下一个动作, 同时更新 Actor 和 Critic 网络以及各自对应的目标网络(第 11—17 行)。该过程不断迭代, 直到达到期望的系统奖励性能。

训练完成后, 智能体从信标链的中央控制器获取学习策略网络参数  $\theta_i$ 。智能体  $i$  基于本地观测的  $o_i$  在本地进行独立决策,  $a_i = \pi(o_i; \theta_i)$ 。 actor 和 critic 的目标网络参数分别更新为:

$$\theta_i^{\text{new}} \leftarrow \tau \cdot \theta_i^{\text{now}} + (1 - \tau) \cdot \theta_i^{\text{old}} \quad (20)$$

$$\omega_i^{\text{new}} \leftarrow \tau \cdot \omega_i^{\text{now}} + (1 - \tau) \cdot \omega_i^{\text{old}} \quad (21)$$

#### 算法 1 DDCM

输入: 回放记忆单元  $D$ , 记忆单元最大容量  $D_n$ , 折扣率  $\gamma$ , 学习率  $\alpha$

输出: 优化后的策略  $\pi_{\theta_i}^*$  与最大化的奖励值  $r^*$

1. For episode = 1, ..., m do
2. 初始环境状态  $s_0 \leftarrow \{M^t, T^t, C^t\} |_{t=0}$
3. For t = 1, 2, ..., T do
4. 每一个星型分片中的可信节点  $TN_i, i \in \mathcal{N}$ , 选取一个随机的动作  $a_i^t$ , 其概率为  $\epsilon$ , 且  $a_i^t = \pi_{\theta_i}(s_i^t)$
5. 执行动作空间  $a^t = \{a_1^t, a_2^t, \dots, a_N^t\}$
6. 系统反馈奖励  $r^t$  以及新状态  $s^{t+1}$
7. 将样本  $(s^t, a^t, r^t, s^{t+1})$  存储到回放记忆单元  $D$  中
8. For agent  $i = 1$  to  $N$  do
9. 从回放记忆单元  $D$  中随机抽取一个四元组  $(s_i^t, a_i^t, r_i^t, s_i^{t+1})$
10. 对  $N$  个目标策略网络做预测, 并将这些预测汇总成集合  $\hat{a}_{i-}^{t+1} = [\hat{a}_{1-}^{t+1}, \dots, \hat{a}_{N-}^{t+1}]$ ,  $\hat{a}_{i-}^{t+1} = \pi(o_i^{t+1}; \theta_i^{\text{now}})$
11. 对第  $i$  个目标价值网络做预测  $Q_i^{\pi^-}(o_i^{t+1}, a_1^{t+1}, a_2^{t+1}, \dots, a_N^{t+1}) =$

$$q(s^{t+1}, \hat{a}_{-}^{t+1}; \omega_{-}^{\text{now}})$$

12. 设置 TD 网络的目标函数  $\hat{y}_i = r_i + \gamma Q_i^{\pi}(o^{t+1}, a_1^{t+1}, a_2^{t+1}, \dots, a_N^{t+1})$

$$|_{a_i = a_i^{t+1}(o_t)}$$

13. 对第  $i$  个价值网络作出预测,  $Q_i^{\pi}(o^t, a_1^t, a_2^t, \dots, a_N^t) = q(s^t, a^t; \omega_i^{\text{now}})$

14. 计算 TD 误差  $\delta_i^t = \hat{y}_i - Q_i^{\pi}(o^t, a_1^t, a_2^t, \dots, a_N^t)$ ;

15. 通过最小化 Loss 函数更新 Critic 价值网络, 更新价值网络  $\omega_{\text{new}}^i \leftarrow \omega_{\text{now}}^i - \alpha \cdot \delta_i^t \cdot \nabla_{\omega_i} q(s^t, a^t; \omega_i^{\text{now}})$

16. 对所有  $M$  个策略网络作出预测  $\hat{a}^t = [\hat{a}_1^t, \dots, \hat{a}_N^t]$ ,  $\hat{a}_i^t = \pi(o_i^t; \theta_i^{\text{now}})$ , 更新所有  $M$  个策略网络  $\theta_i^{\text{new}} \leftarrow \theta_i^{\text{now}} - \beta \cdot \nabla_{\theta_i} \mu(o_i^t; \theta_i^{\text{now}}) \cdot \nabla_{\theta_i} q(s^t, \hat{a}^t; \omega_i^{\text{now}})$

17. end For

18. 由式(19)、式(20)更新所有  $2n$  个目标网络

19. end For

20. end For

## 4 实验与分析

### 4.1 实验设计

本文实验环境为 10 台 1TB 存储空间, 16GB RAM, 8 核 CPU 的服务器, 服务器之间通过高速网络通信。每台服务器使用 ubuntu20 系统, 采用 Docker 虚拟化技术承载分片。算法采用 Python 3.6 语言实现, 对比分片算法为 SkyChain, DRLPG, DQN。

### 4.2 实验参数

实验中使用的参数如表 1 所列, 本文设置系统中每个节点的计算能力相同。

表 1 实验参数

Table 1 Experimental parameters

Parameter	Value
分片的最大数量	64
平均区块的大小/MB	6
平均区块的生成间隔/s	16
平均交易大小/B	64
节点之间的信息传输速率/Mbps	10
消息验证时间/s	0.5
随机交易生成时间/s	1
状态区块创建时间/s	0.5
初始节点的数量	200
初始 TXs arrival rate	500
节点的计算能力/GHz	1

本文基于以太坊真实数据集, 设计了 3 类实验, 如表 2 所列。

表 2 实验设计

Table 2 Experimental design

实验	分片数量	每秒交易数量/ (TX/s)	节点数量
1	4~64	1 500	2 000
2	16	300~1 900	2 000
3	16	1 500	2 000~10 000

### 4.3 分片数量分析

当前存在的区块链分片方案存在低吞吐量和高延迟的问题。增加分片数量可以获得更高的吞吐量, 但会增加 CST 的比例<sup>[20]</sup>。分别测试分片数量对吞吐量、跨分片交易的比例以及交易确认延迟的影响。如图 3(a) 所示, 当分片数量少于

32 时, 所有的 baseline 的吞吐量都在增加, 之后所有的算法均保持收敛。可以观察到, 在分片数量为 4 时, A3S 效率最佳。如图 3(b) 所示, 随着分片数量的增加, DQN, DRLPG 与 SkyChain 的平均 CST 比例持续增长。当分片数量为 32 时, 其平均 CST 比例比其他 baseline 要低 19.53%。由图 3(c) 中可以观察到, 交易确认时延随着分片数量从 4 增加到 64, 呈下降趋势。尽管所有分片算法都可以降低系统内的交易确认延迟, 但 A3S 比其他 baseline 的延迟要低 38.76%。综上, 当分片数量增加时, 对比其他方法, A3S 在提高吞吐量的同时减少了系统内的交易确认延迟, 并且具有较低的平均 CST 比例。

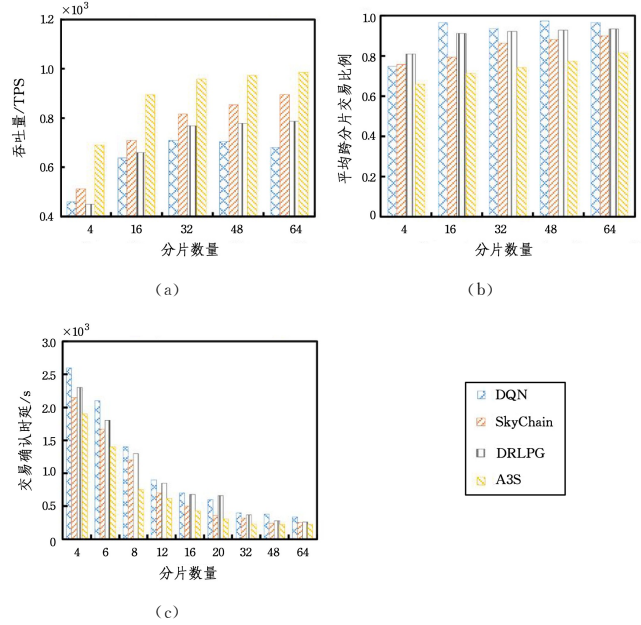


图 3 当事务到达率为 1500 TXs 每秒且节点数量为 2000 时的 3 种指标

Fig. 3 Three indicators when the transaction arrival rate is 1500TXs/sec and the number of nodes is 2000

### 4.4 交易到达率分析

为测试交易到达率对吞吐量、跨分片交易的比例以及交易确认延迟的影响, 将分片数量固定为 16, 节点的数量固定为 2000, 测试不同分片数量下系统吞吐量、平均 CST 比例以及交易确认延迟的情况。观察图 4(b) 可知, 随着系统内交易到达率的增加, 4 种分片方案的平均 CST 比例均持续增长。其中, DQN 快速增长, 当交易到达率从 200 增加到 1150 时, 平均 CST 比例增加了 26.73%; DRLPG 与 SkyChain 对应的增长幅度较小; 尽管采用 A3S 的平均 CST 比例也在增长, 但其跨分片交易比例相对最低。当交易到达率为 1400 时, 采用 A3S 的平均 CST 比例要比 DDPG 低 31.02%。

### 4.5 节点数量分析

节点数量的增加可能会使得通信成本和共识延迟增加, 从而影响系统的整体性能。为测试节点数量对吞吐量、跨分片交易的比例以及交易确认延迟的影响, 将分片时固定为 16, 交易到达率设置为 1500, 来测试不同分片数量时系统吞吐量、平均 CST 比例以及交易确认延迟的情况。

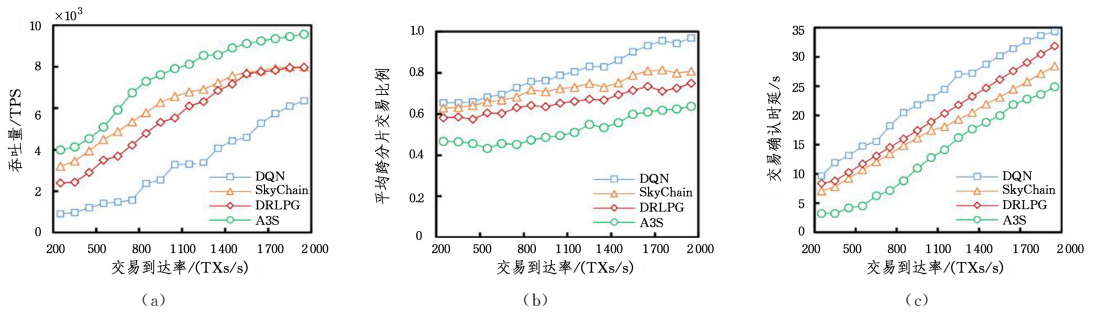


图4 当分片数量为16且节点数量为2000时的3种指标

Fig. 4 Three indicators when the number of shards and nodes is 16 and 2000 respectively

如图5(a)所示,系统吞吐量随着节点数量的增加而增加。在系统内节点数量达到2000之前,所有4种算法分片对应的吞吐量均快速降低,DRLPG甚至降低了58.39%。当节点数量超过2000时,系统吞吐量趋于收敛。在分片数量、交易数量以及节点数量一致的情况下,采用A3S算法分片的吞吐量比DQN,DRLPG和SkyChain分别高43.09%,17.99%,25.75%。从图5(b)可以观察到,

随着节点数量从0增加到10000,系统内的平均CST比例持续增长。采用A3S算法分片相比3种baseline中性能最优的DDPG平均CST比例降低了17.99%。如图5(c)所示,随着节点数量的增加,延迟逐渐增大,这是因为更多的节点需要更多的通信次数和同步时间。采用A3S算法分片的交易确认时延比DQN,DRLPG和SkyChain分别降低了30.43%,14.99%,20.41%。

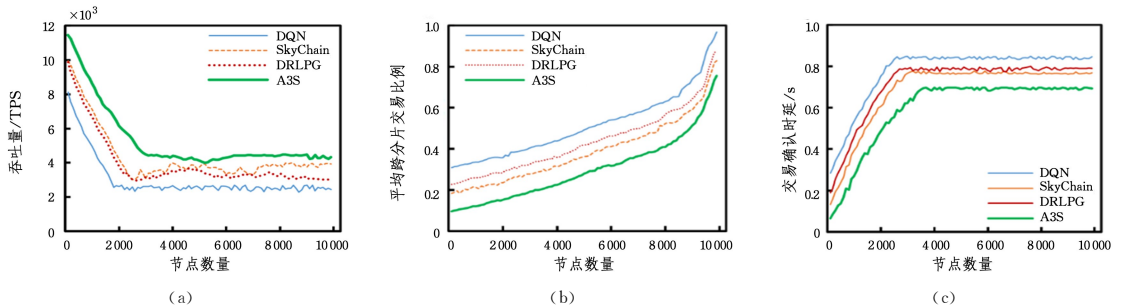


图5 当事务到达率为1500 TXs每秒且分片数量为16时的3种指标

Fig. 5 Three indicators when the transaction arrival rate is 1500 TXs/sec and the number of shards is 16

**结束语** 本文研究了针对区块链分片系统中跨分片交易比例高的问题,提出了一种基于强化学习的星型链空间自适应分配方法A3S。该方法根据星型架构分片,采用深度强化学习动态调整分片内节点。实验结果表明,该方法可以有效地处理分片内和分片间的事务,在降低跨分片交易比例的同时提升整个分片系统的吞吐量,稳定交易确认延迟。由于区块链分片系统的复杂性,本文提出的方案并不完善,需要进行进一步的研究。具体工作包括:跨分片交易协议、分片间负载均衡,以及结合实际的应用场景实现技术落地等。

## 参考文献

[1] LI X, DONG X, XU X, et al. A Blockchain-Based Scheme for Efficient Medical Data Sharing with Attribute-Based Hierarchical Encryption [C] // WISA. Cham: Springer International Publishing, 2022: 661-673.  
 [2] WEI W, ZHOU Y, LI D, et al. BEAIV: Blockchain Empowered Accountable Integrity Verification Scheme for Cross-Chain Data [C] // Web Information Systems and Applications. Singapore: Springer Nature Singapore, 2023: 488-500.  
 [3] LI X H, LI C, ZHANG G G, et al. A Survey on the Integration of Blockchain and Database Technology [J]. Computer Science and Exploration, 2023, 17(4): 761-770.

[4] WU Y, WANG Z, MA Y, et al. Deep reinforcement learning for blockchain in industrial IoT: A survey [J]. Computer Networks, 2021, 191: 108004.  
 [5] HUANG H W, KONG W, PENG X W, et al. Survey on blockchain sharding technology [J]. Computer Engineering, 2022, 48(6): 1-10.  
 [6] LIU X, XIE H, YAN Z, et al. A survey on blockchain sharding [J]. ISA Transactions, 2023, 141: 30-43.  
 [7] WANG J, MA J W, LUO J X. A Blockchain Sharding Scheme for Edge Computing [J]. Journal of Internet of Things, 2023 (7): 88-100.  
 [8] JIA L, LIU Y, WANG K, et al. Estuary: A Low Cross-Shard Blockchain Sharding Protocol Based on State Splitting [J]. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(3): 405-420.  
 [9] ZHANG J, CHEN W, HONG Z, et al. Efficient Execution of Arbitrarily Complex Cross-Shard Contracts for Blockchain Sharding [J]. IEEE Transactions on Computers, 2024, 73(5): 1190-1205.  
 [10] XU Y, SHAO J, SLAATS T, et al. MWPoW+: A Strong Consensus Protocol for Intra-Shard Consensus in Blockchain Sharding [J]. ACM Transactions on Internet Technology, 2023, 23(2): 34:1-34:27.

- [11] LI M, LIN Y, ZHANG J, et al. CoChain: High Concurrency Blockchain Sharding via Consensus on Consensus[C] // IEEE INFOCOM 2023—IEEE Conference on Computer Communications, 2023; 1-10.
- [12] YU B, ZHAO H, ZHOU T, et al. OverShard: Scaling blockchain by full sharding with overlapping network and virtual accounts [J]. Journal of Network and Computer Applications, 2023, 220: 103748.
- [13] LIU Y, LIU J, VAZ SALLES M A, et al. Building blocks of sharding blockchain systems: Concepts, approaches, and open problems[J]. Computer Science Review, 2022, 46: 100513.
- [14] MU K, WEI X. EfShard: Toward Efficient State Sharding Blockchain via Flexible and Timely State Allocation[J]. IEEE Transactions on Network and Service Management, 2023, 20(3): 2817-2829.
- [15] LI M, LUO X, XUE K, et al. A Secure and Efficient Blockchain Sharding Scheme via Hybrid Consensus and Dynamic Management[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 5911-5924.
- [16] LUU L, NARAYANAN V, ZHENG C, et al. A Secure Sharding Protocol For Open Blockchains[C] // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: Association for Computing Machinery, 2016; 17-30.
- [17] GADIRAJU D S, LALITHA V, AGGARWAL V. An Optimization Framework Based on Deep Reinforcement Learning Approaches for Prism Blockchain[J]. IEEE Transactions on Services Computing, 2023, 16(4): 2451-2461.
- [18] KOKORIS-KOGIAS E, JOVANOVIĆ P, GASSER L, et al. OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding[C] // 2018 IEEE Symposium on Security and Privacy (SP), 2018; 583-598.
- [19] ZHANG J, HONG Z, QIU X, et al. SkyChain: A Deep Reinforcement Learning-Empowered Dynamic Blockchain Sharding System[C] // Proc of the 49th International Conference on Parallel Processing. New York, NY, USA: Association for Computing Machinery, 2020; 1-11.
- [20] ZHOU K, ZHANG X, WANG C, et al. Accelerating Cross-Shard Blockchain Consensus via Decentralized Coordinators Service With Verifiable Global States[J]. IEEE Transactions on Services Computing, 2024, 17(4): 1340-1353.



**DU Likuan**, born in 1999, postgraduate. His main research interests include blockchain and reinforcement learning.



**SONG Baoyan**, born in 1965, Ph.D., professor, Ph.D supervisor, is a member of CCF(No. 08767S). Her main research interests include large-scale graph processing technology, big data processing technology and stream data processing technology.

(责任编辑:柯颖)