

# Hyperledger Fabric背书策略的提案分发改进方案

张龙静

北京信息科技大学, 网络文化与数字传播北京市重点实验室, 北京  
Email: 2380364429@qq.com

收稿日期: 2021年3月28日; 录用日期: 2021年4月21日; 发布日期: 2021年4月28日

---

## 摘要

Hyperledger Fabric是当前最热门的联盟链技术之一, 在大量的领域得到了应用。Fabric的背书节点的验证阶段通常都是系统的性能瓶颈。由于背书节点的数量固定, 提案请求往往得不到及时的处理, 影响系统的吞吐量。本文提出了一种根据当前物理资源动态调整背书节点的机制。根据实验分析, 优化后的Hyperledger fabric具有更快的交易处理速度, 在背书成功率仅下降1%的前提下, 数据上链的吞吐量提高了约19.6%, 数据上链的平均时延下降了26.7%。

## 关键词

Hyperledger Fabric, 联盟链, 背书策略, 背书节点, 负载均衡

---

# Hyperledger Fabric Endorsement Strategy Proposal Distribution Improvement Plan

Longjing Zhang

Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, School of Computer Science, Beijing Information Science and Technology University, Beijing  
Email: 2380364429@qq.com

Received: Mar. 28<sup>th</sup>, 2021; accepted: Apr. 21<sup>st</sup>, 2021; published: Apr. 28<sup>th</sup>, 2021

---

## Abstract

Hyperledger Fabric is currently one of the most popular consortium chain technologies and has

been applied in a large number of fields. The verification phase of the fabric endorsement node is usually the performance bottleneck of the system. The number of endorsement nodes is fixed, and proposal requests are often not processed in time to form a queue, which affects the throughput of the system. This paper proposes a mechanism to dynamically adjust endorsement nodes based on current physical resources. According to experimental analysis, the optimized fabric has a faster transaction processing speed, and the throughput is increased by about 19.6%, and the average delay of data uploading has dropped by 26.7% under the premise that the endorsement success rate is only reduced by 1%.

## Keywords

Hyperledger Fabric, Alliance Chain, Endorsement Strategy, Endorser, Load Balancing

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

2008年中本聪提出了比特币[1]的概念,在此基础之上发展出了区块链技术。区块链是在数学、密码学、计算机科学的发展成果之上发展起来的一个的领域。比特币是区块链技术的第一代应用。随着人们对区块链的深入研究,将区块链更广泛地运用在金融,交通,军事等领域。区块链很可能对当今互联网产生颠覆式的改变,实现信息互联网向价值互联网的转变[2]。区块链技术的一个核心问题是在无中心、弱信任的分布式系统中让各个节点达成共识[3],克服了中心化机构中权力高度集中,数据高度集中,安全成本高的缺点。

Hyperledger Fabric 是一个许可的区块链架构(permissioned blockchain infrastructure)。其由 IBM 和 Digital Asset 提交给 Linux 基金会,是一个成熟的商业区块链平台[4]。Fabric 将节点区分为客户节点(Client)、对等节点(Peer)和排序节点(Orderer)。客户创建交易,提交给背书节点背书。Hyperledger Fabric 采用“背书-排序-验证”这三步最后达到共识[5]。Hyperledger Fabric 将应用层的信任模型同底层的共识协议解耦,共识机制设计成“可插入式”的架构。背书提高了交易的成功率,验证保证交易内容的合法性,排序保证交易顺序的最终一致性。通过将这两部分进行解耦[6],开发者可以更灵活的使用超级账本。

目前,Hyperledger Fabric 相关研究都集中在对于共识机制的改进,对于背书过程的研究较少。Wang C.等[7]在 Fabric 平台上进行了一项性能研究和瓶颈分析。研究了交易生命周期各个阶段的表现,以及比较了不同的排序服务。实验结果表明背书节点的验证阶段是测试平台的性能瓶颈,因为验证节点的计算工作量比较大。本文为了突破 Fabric 验证阶段的性能瓶颈,对背书策略的分发方案进行了改进。

## 2. Hyperledger Fabric 共识机制分析

Fabric 由各个节点以及各个组织架构共同完成整个上链、查询等功能。节点只是一个逻辑上起到通信功能的实体。在一台物理服务器上可能运行着多个不同类型的节点。节点主要包括客户端,Peer 节点,排序节点这三种:

客户端(Client): 客户与系统交互的入口,代表着客户实体。主要是创建交易并将交易提交给选中的背书节点,以及背书成功后将广播交易提交到排序节点。

Peer 节点: 主要是提交交易、维持世界态以及将账本的复制到所有节点。Peer 中的一部分节点还可

以附加背书节点(Endorser)的功能, 以及提交节点(committer peer)的功能。排序节点主要是对排序后的区块验证有效性, 并提交有效交易到账本。背书节点会对客户端发来的交易进行证书检查等验证, 执行链码模拟交易, 确保交易不会造成状态数据库错误。交易背书成功后会生成读写集, 该背书节点会对其签名, 并附上自己的身份, 达到为其“背书”的目的[8]。

排序节点(Orderer): 交易背书成功后, 由排序节点进行广播。多个排序节点通过共识算法达成一致, 提供排序服务[9]。排序节点并不能获得交易的详细信息, 保护了数据的安全性。排序节点在对交排序后, 打包成块。

Hyperledger fabric 建立共识主要由 Ordering of Transactions (交易排序)和 Validating Transactions (交易验证)这两个过程组成。在逻辑上将这两个过程分离, 让 Fabric 可以运用各种共识模块。客户端将背书成功的交易发送给排序节点。排序节点根据共识算法和配置策略对交易进行排序。为了提高效率, 排序服务会将多个交易打包成一个块再输出, 排序服务中需要保证块内交易的顺序。Hyperledger Fabric 中常用的 Kafka [10]共识机制, 用 orderer 集群(排序节点集群)对交易进行排序。整个排序服务是将全局交易作为输入, 在集群内形成统一的、唯一的、确定的、经过排序的交易分发给提交节点。为了校验交易的正确性, 共识的建立依赖于智能合约层, 智能合约层定义了商业逻辑来确认如何验证交易有效。智能合约层根据特定的策略与约定来确认每一笔交易都是有效的。无效的交易会被拒绝, 并在块中剔除。

### 3. Hyperledger Fabric 共识机制改进

#### 3.1. 提案分发数量的确定

超级账本的默认的背书策略是每份提案由所有背书节点进行装载链码, 模拟交易过程。背书节点的数量有限, 当交易量增大时, 往往得不到及时处理而成为系统的性能瓶颈。假如背书节点的总数为  $c$ , 背书节点处理后将模拟交易结果返回提交节点, 提交节点根据背书策略收集  $l$  个一致的提案响应, 就等于此次交易背书成功, 进行下一步处理。区块链为了达到信息安全, 不可篡改的目的, 通常是多个节点对同一个任务进行处理多次(区块链的共识机制也是如此, 交易要在所有的排序节点中达成一致)。正是由于区块链的这种特性, 造成了区块链处理交易的压力比较大。为了减轻背书压力, 提高系统效率, 提出了动态选取的背书策略提案分发方案。将提案分发方式由分发给所有背书节点改进为动态选取当前负载低的部分背书节点处理提案。

在本文中, 对默认的背书策略进行改进的方法如下:

将提案请求不再分发给所有背书节点处理, 在尽量保证背书成功率的情况下, 尽可能提高背书性能。可以选取负载低的  $s$  个背书节点处理提案。分发提案数量  $s$  的选取, 当  $s$  越小, 背书性能越高, 但背书成功率越低。需要找到一个合理的值。

背书节点总数为  $c$ , 将所有背书节点看作一个服务系统, 假设每个节点正常运行(不出现宕机情况)的概率为  $p$ 。改进前, 每个提案分发给  $c$  个背书节点, 提交节点收集  $l$  个一致的提案响应。此时背书成功率为  $c$  个背书节点中至少  $l$  个正常工作, 设  $P_{def}$  表示改进前的背书成功率[11], 计算公式为:

$$P_{def} = 1 - \sum_{i=0}^{l-1} p^i (1-p)^{c-i}$$

改进后, 从  $c$  个背书节点中选取  $s$  个背书节点处理提案, 提交节点收集  $l$  个一致的背书响应。此时背书成功率为  $s$  个背书节点中至少  $l$  个正常工作, 设  $P_{imp}$  表示改进后的背书成功率, 计算公式为:

$$P_{imp} = 1 - \sum_{i=0}^{l-1} p^i (1-p)^{s-i}$$

令  $kP_{imp} = P_{def}$  简化变形可得:

$$s = \log_{1-p} \frac{k-1+a}{ak} + c$$

### 3.2. 背书节点的负载率

背书节点的负载率代表当前背书节点的计算资源的高低, 负载率越高, 该节点所能够承受验证交易的能力越差。设  $C$  为当前节点 CPU 的占用率,  $M$  为内存的占用率。  $w_1$  为 CPU 常量系数,  $w_2$  为内存常量的系数。那么背书节点当前的负载率  $L$  的值可表示为:

$$L = w_1 * C + w_2 * M$$

其中  $w_1$ ,  $w_2$  的大小的设置是由背书节点的所处理交易的性质决定的。背书节点主要的任务还是对节点状态和权限的验证, 所以对 CPU 的资源消耗比较大, 内存的占用较小。在本文中只是选取了 CPU 和内存作为参考依据, 但是在超级账本的运行过程中, 网络拥堵程度, 磁盘的读写能力等都会影响整个系统的性能。在本文中选取了最能影响背书节点性能的两个方面, 作为动态调度依据。

### 3.3. 背书节点的选取规则

从  $c$  个背书节点中选取  $s$  个节点处理提案, 最简单的方法为随机选取, 但随机选取无法保证提案分发的均衡性, 可能导致大量请求集中分配给少部分背书节点。根据木桶效应, 背书效率将取决于少部分节点的处理效率。当尽可能实现提案均衡分发时, 才能达到背书效率最大化的目标。区块链的负载均衡相较于分布式系统有所不同。分布式系统的负载均衡是将一个任务分成多个步骤给不同的机器处理。而区块链为了实现一致性, 不可篡改等特性需要将同一笔交易分配给所有节点重复去处理(对交易的背书是默认分发给所有的背书节点, 共识机制是所有的共识节点参与排序, 保持最终一致)。文献[11]中采用的是基于虚拟节点的哈希环的方式, 在负载均衡中属于静态算法。该方法在哈希环上轮流选取虚拟节点, 没有考虑到节点的实时负载。只有当背书节点的处理性能大致相同, 并且只能在稍微短的一段时间内保持负载均衡。本文中动态的选取当前负载低, 性能高的高的背书节点, 能大大减少被少数低性能节点拖累的概率。背书节点选取算法的具体写法见表 1。

**Table 1.** Dynamic selection algorithm

**表 1.** 动态选取算法

算法 1 动态选取算法
<b>输入:</b> $c$ 为背书节点的 CPU 占用率; $m$ 为背书节点的内存占用率; $w_1, w_2$ 为常量系数; $ip$ 为背书接的 $ip$ 地址;
<b>输出:</b> 所选取的背书节点以及相应 $tls$ 证书的集合;
1: 根据背书节点的 $ip$ 获取背书节点 $x$ ,
2: $L = w_1 * c + w_2 * m$ , 计算节点的综合负载率 $L$ ;
3: $sort(x_1, x, \dots, x_n)$ 根据节点的综合负载率对节点进行递增排序;
4: <b>for</b> $i = 1; i < s; i++$ <b>do</b> 获取负载较低的前 $s$ 个背书节点
5: <b>end for</b>
6: 对 $s$ 个背书节点的 $ip$ 和根据配置获取的所有的背书节点的 $ip$ 作对比, 匹配相应的 $tls$ 证书文件;
7: 返回 $s$ 个背书节点;

方案的时间开销分析: 改进前后的背书时延进行对比分析。设背书节点总数为  $c$ , 提交节点等待收集  $l$  个一致的背书响应。将所有背书节点看作一个服务系统, 在时间  $t$  内, 用户请求率为  $q$ , 背书系统的平均服务率为。设改进前的背书时延为  $t_e$ , 计算公式为:

$$t_e = \frac{qtc}{\mu}$$

设改进方案的提案分发数量为  $s$ ，背书节点的搜索时间为  $t_h$ ，改进后的背书时延为  $t'_e$ ，计算公式为：

$$t'_e = \frac{qts}{\mu} + t_h + t_s$$

设改进前后的时延差为  $t_d$ ，计算公式为：

$$t_d = qt \frac{c-s}{\mu} - t_h - t_s$$

其中， $t_h$  为快速排序法对  $c$  个背书节点的按负载度排序的时延，排序算法的时间复杂度为  $O(n \log_2 n)$ ，其中  $n$  等于  $c$ 。 $t_s$  为遍历前  $s$  个节点的时间复杂度，为  $s$ 。相对于对于整个背书的花费的时间，算法的耗时可以忽略不计。理论上背书过程花费的时间为  $qt \frac{c-s}{\mu}$ 。

方案的存储开销分析：方案的额外存储开销为存储背书节点负载的文件。若背书节点数为  $c$ ，单个节点的 IP 地址和负载度为  $y$  bit。则改进的方案为  $c * y$ ，对于节点的存储开销也较小。

#### 4. 实验

实验采用 Caliper 工具完成测试(Hyperledger Caliper 是一款由华为开发的标准化的区块链性能测试工具[12])。实验是在一台笔记本电脑上搭建的测试网络。该电脑的 CPU 为 6 个核心，型号 Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz。内存为 16 GB RAM，操作系统为 Ubuntu 16.04。实验中设置一个排序节点(采用的是测试模式下，用 solo 作为排序服务)，设置 8 个背书节点。

原始方案中背书策略为：默认情况下提交节点将提案分发给所有的背书节点处理，等待收集 3 个背书节点的响应。根据上文中背书成功率公式计算可得：优化方案中，背书成功率为改进前的 0.99，提案分发数量  $s$  为 5，即选取背书节点中的 5 个来处理提案，等待 3 个背书节点的背书响应。

方案改进前后，查询交易吞吐量的对比实验结果如图 1 所示，查询交易平均延迟的对比实验结果如图 2 所示。

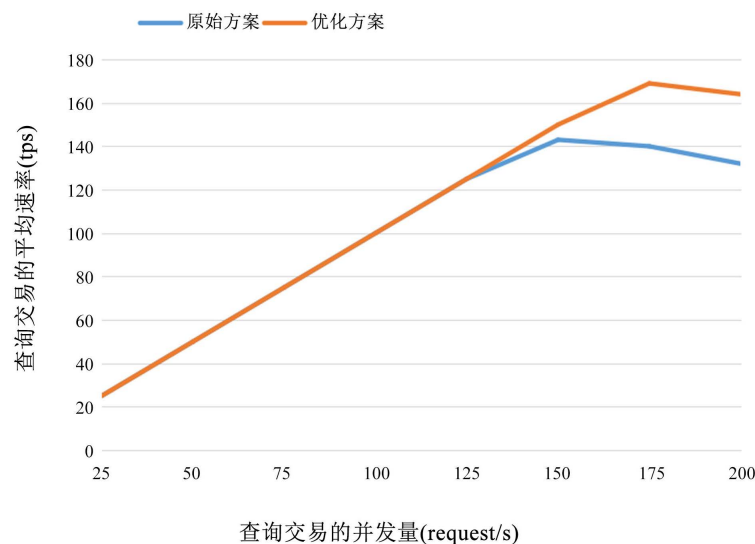


Figure 1. Query transaction throughput comparison

图 1. 查询交易吞吐量对比

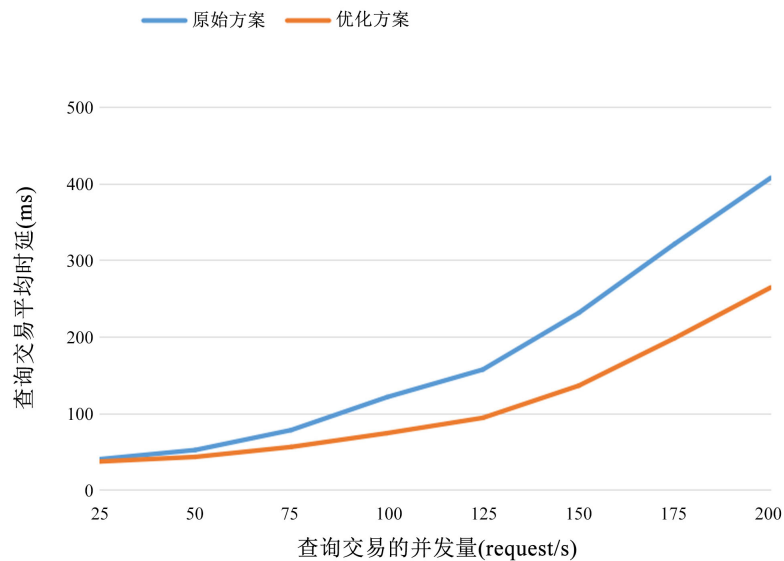


Figure 2. Query transaction average latency comparison

图 2. 查询交易平均延迟对比

如图 1 所示, 两种方案的查询数据吞吐量都随着并发量的增加而线性增加, 随后原始方案和优化方案都先后各自达到最大吞吐量分别为 143 tps 和 169 tps, 优化方案的最大吞吐量相比优化前提高了约 18%。图 2 中, 两种方案在各自吞吐量饱和点的延迟分别为 231 ms 和 198 ms, 优化方案查询数据的延迟相比优化前下降了约 16.7%。随着并发量的继续增大, 系统都出现了拥堵现象, 吞吐量都有小幅的下降趋势。

方案改进前后, 数据上链的吞吐量及平均延迟的对比实验结果分别如图 3 和图 4。

通过图 3 可以看出, 原始方案和优化方案的数据上链的最大吞吐量分别为 46 tps 和 55 tps, 优化方案的最大吞吐量较优化前提高了约 19.6%。图 4 中, 各自饱和点的数据上链平均延迟分别为 19 s 和 15 s, 优化方案的数据上链延迟相比优化前下降了约 26.7%。

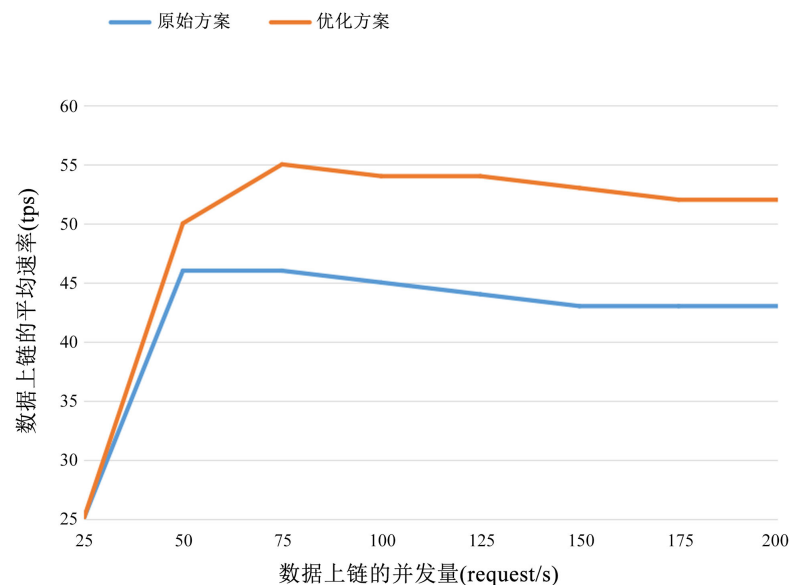


Figure 3. Throughput comparison of data uploading

图 3. 数据上链的吞吐量对比

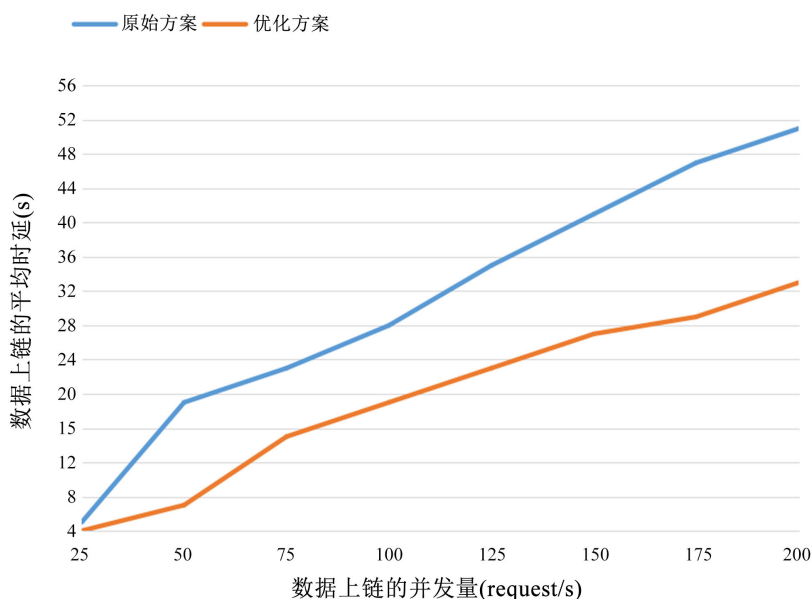


Figure 4. Comparison of the average latency of data uploading  
图 4. 数据上链的平均延迟对比

## 5. 结论

本文针对 Hyperledger Fabric 原有共识机制存在的问题, 提出了一种基于动态选取的背书策略的提案分发方法, 并对优化后的共识机制进行了对比实验和分析。实验分别用数据查询和数据上链(query 和 invoke)两种方式调用链码, 实验及分析结果表明, 优化后的 fabric 在背书成功率下降很小的情况下, 都获得了更快的交易处理速度, 更低的交易平均时延。

## 参考文献

- [1] Nakamoto, S. (2019) Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>
- [2] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494.
- [3] Liu, A.D., Du, X.H., Wang, N. and Li, S.Z. (2018) Research Progress of Blockchain Technology and Its Application in Information Security. *Journal of Software*, **29**, 2092-2115.
- [4] Cachin, C. (2016) Architecture of the Hyperledger Blockchain Fabric. *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, Chicago, 25 July 2016, 310.
- [5] Elli, A., Artem, B., Vita, B., Christian, C., Konstantinos, C. and Angelo, D. (2018) Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceedings of the Thirteenth EuroSys Conference*. Porto, April 2018, Article No. 30, 1-15.
- [6] Vukolić, M. (2017) Rethinking Permissioned Blockchains. *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, Abu Dhabi, April 2017, 3-7. <https://doi.org/10.1145/3055518.3055526>
- [7] Wang, C. and Chu, X. (2020) Performance Characterization and Bottleneck Analysis of Hyperledger Fabric. 2020 *IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, Singapore, 29 November-1 December 2020, 1281-1286. <https://doi.org/10.1109/ICDCS47774.2020.00165>
- [8] Manevich, Y., Barger, A. and Tock, Y. (2019) Endorsement in Hyperledger Fabric via Service Discovery. *IBM Journal of Research and Development*, **63**, 1-9. <https://doi.org/10.1147/JRD.2019.2900647>
- [9] Brandenburger, M., Cachin, C., Kapitza, R. and Sorniotti, A. (2018) Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric. arXiv: 1805.08541.
- [10] Sukhwani, H., Wang, N., Trivedi, K.S. and Rindos, A. (2018) Performance Modeling of Hyperledger Fabric (Permissioned Blockchain Network). *Proceedings of 2018 IEEE 17th International Symposium on Network Computing and*

*Applications (NCA)*. Cambridge, 1-3 November 2018, 1-8. <https://doi.org/10.1109/NCA.2018.8548070>

- [11] 赵国锋, 何英, 周继华. 基于区块链的数字版权登记技术[J]. 信息技术与网络安全, 2019, 38(4): 79-83.
- [12] HuaWei. Hyperleger Fabric Test Tool: Project Caliper. <https://www.hyperledger.org/projects/caliper>, 2018-03-15.