·安全技术· 文章编号：1000-3428(2018)01-0160-05 文献标志码：A 中图分类号：TP302

具有状态合法性验证的区块链一致性算法研究

吴 腾[1,2,3]，黄 锴[1,2,3]，周琳琳[2]，孔 宁[2]

（1. 中国科学院计算机网络信息中心，北京 100190；2. 中国互联网络信息中心，北京 100190；3. 中国科学院大学，北京 100190）

摘 要： 传统拜占庭一致性中常见的中心化和去中心化算法在解决合法性验证的过程中存在容错率低、消息复杂度高等问题。为此，提出新的区块链一致性算法。引入两阶段提交和法定人数投票的过程，利用区块链协议的分布式总账特点，解决去中心化环境中的合法性验证问题，随后对其最终一致性进行理论证明。实验结果表明，与传统拜占庭一致性协议相比，该算法减少了消息传递次数，提高了系统容错率。

关键词： 区块链；中心节点；状态合法性；两阶段提交；去中心化

中文引用格式： 吴 腾，黄 锴，周琳琳，等. 具有状态合法性验证的区块链一致性算法研究[J]. 计算机工程，2018，44(1)：160-164.

英文引用格式： WU Teng，HUANG Kai，ZHOU Linlin，et al. Research on Blockchain Consistency Algorithm with State Legality Verification[J]. Computer Engineering，2018，44(1)：160-164.

Research on Blockchain Consistency Algorithm with State Legality Verification

WU Teng[1,2,3]，HUANG Kai[1,2,3]，ZHOU Linlin[2]，KONG Ning[2]

（1. Computer Network Information Center，Chinese Academy of Sciences，Beijing 100190，China；
2. China Internet Network Information Center，Beijing 100190，China；
3. University of Chinese Academy of Sciences，Beijing 100190，China）

【Abstract】 Traditional leader based and decentralized algorithm for solving Byzantine consensus exists the problem of low fault tolerance and high message complexity during the process of solving legitimacy verification. In order to solve these problems，this paper proposes a new blockchain consistency algorithm. It introduces two-phrase commit and quorum voting，using the characteristics of distributed ledge in Blockchain protocol to solve legitimacy verification and ultimate consistency is proved later on. Experimental result shows that compared with traditional Byzantine consensus protocol，this algorithm reduces the complexity of message passing and improves the system fault tolerance rate.

【Key words】 blockchain；center node；state legality；two-phase commit；decentralization

DOI： 10.3969/j.issn.1000-3428.2018.01.027

0 概述

区块链在不可靠的分布式环境中维护了一个公共的总账[1]，这个总账由一系列的匿名参与者来维护和扩展。近年来，区块链网络吸引了越来越多的工程人员、学者和投资者的注意。伴随着大量的资本投入[2]，区块链得到了快速部署，已经成为了一项公共基础设施。

由于区块链没有可信任的中心节点，这使得设计和实现去中心化的域名系统（Domain Name System，DNS）[3]、公钥基础设施（Public Key Infrastructure，PKI）和去中心化的文件存储成为现实。

现有金融体系中，为了确保安全，当涉及大额交易和可疑交易时通常需要特殊的监管和审查[4]。而对于去中心化的比特币来说，不受监管是其能够流行的部分原因之一，但同时也会给大额交易的交易方带来潜在的风险。研究者提出让国际货币基金组织（International Monetary Fund，IMF）参与到比特币的运营中[5]，但如何保证监管者 IMF 不成为新的中心，还尚未解决。

对于去中心化的域名系统，当域名注册存在冲突，即权威机构和个人同时申请一个权威域名，权威域名的归属问题是由获得区块写入机会的单个节点决定，单个节点一旦出错，将导致此权威域名被个人注册而无法更改。

去中心化的文件存储系统中，有版权问题或者不合法规的资源一旦上传成功，除了上传者本身，其

基金项目： 国家自然科学基金青年基金“物联网异构标识解析关键技术研究”（61402436）。

作者简介： 吴 腾（1992—），男，硕士研究生，主研方向为网络安全；黄 锴，硕士研究生；周琳琳，工程师、硕士；孔 宁，研究员、博士。

收稿日期： 2016-12-02 **修回日期：** 2017-01-11 **E-mail：** wutengucas@163.com

他人很难删除。对于广泛运行的点对点(Peer to peer,P2P)网络而言[6],通常是在其部署爬虫网络来检测非法资源的路径,并将其加入黑名单,从而达到保护网络的目的。爬虫软件只能检测哪些资源是非法,而无法阻止资源的上传,并且引入第三方的检测方法,会带来更多的问题。

上述基于区块链的 3 种应用,都需要在不安全的分布式环境中保持全局一致性。转账交易的记录、域名的管理、文件的上传和更改,对外都需要保持一致性。每一个具体的操作,都涉及了状态的转换。上述问题的根源在于,区块链在写入时,只需要提供正确工作量的证明,而不对其状态转换进行合法性验证。

不安全的分布式环境中的一致性问题称为拜占庭一致性[7]。最初拜占庭一致性问题需要指数级的算法才能解决,随后人们提出了多项式级别的拜占庭协议[8],极大降低了拜占庭系统的开销,使其在分布式系统中的应用成为可能。解决拜占庭一致性的算法大致可分为 2 类:中心化算法和去中心化的算法。

去中心化算法能够解决中心化算法中存在的单点故障,并且能够容忍更多的故障节点。区块链技术作为去中心化算法的代表,其有效性经历了广泛的实践验证。但去中心化一致性算法存在未考虑到状态转换的合法性由谁决定的问题。区块链的处理方法是由获得区块写入权力的单个节点决定,这就违背了去中心化的初衷。因此,本文在区块链协议上引入了合法性验证的法定人数投票,由法定人数对系统的状态转换有效性进行验证。

1　相关研究

拜占庭将军问题是点对点通信中的基本问题,其含义是一些拜占庭将军率领他们的部队要攻占敌人的一个城池,每个将军只能控制自己的部队,并且通过信使传递消息给其他将军,将军之中存在叛徒,如何设计一种策略使忠诚的将军达成一个不算坏的行动协议而不受叛徒的挑拨。

随后研究者提出了中心化的拜占庭一致性协议[9],协调者的引入降低了消息传递的复杂度,同时提高了最好情况下的性能。中心化算法在最好情况下,需要 α 轮消息交换(α 是常数),但在最坏情况下,则需要 $\alpha(t+1)$ 轮消息交换[10](t 是最大出错节点个数)。中心化一致性协议中协调者的更换通常由超时条件触发,因此,协调者能够在不被检测到的情况下,故意推迟消息的发送,降低系统的吞吐[11]。这促使了去中心化一致性算法的发展。因此,本文研究的重点在于如何使去中心化的区块链一致性算法具有状态合法性验证,并提高系统容错率。

1.1　单节点合法性验证

两阶段提交协议[12]是在计算机网络、数据库领域内,为了在进行事务处理过程中能够保持原子性和一致性而设计的一种算法。两阶段提交将一个事务的提交处理过程分成了投票和执行 2 个阶段,协调者向所有的参与者广播投票请求,当协调者收到了所有参与者的同意消息后,协调者广播提交请求,当协调者收到了所有参与者的提交完成消息后,这个事务才算是提交成功。两阶段提交的优点在于消息传递次数少,实现方便。其缺点是容错率低,只要有一个参与者拒绝投票或者提交失败,整个事务无法完成。此外,两阶段协议要应用在拜占庭环境中,还需要解决选举协调者和更换协调者的问题。

本文在两阶段提交基础上,设计了选举和更换协调者的策略,使其能够在区块链中完成法定人数的投票确认。当正常节点获得区块写入机会进行统计投票时,执行了类似的两阶段提交协议,区别在于传统两阶段的法定集合是所有节点,而本文的法定集合是半数节点,这提高了系统的容错率。

1.2　多节点合法性验证

文献[13]提出了实用拜占庭容错协议(Practical Byzantine Fault Tolerance,PBFT),使得拜占庭协议在分布式系统中应用成为可能。PBFT 采用了主从模式(Primary and backups)和法定人数(Quorum),法定人数为 $2f+1$,其中,f 为出错节点个数,总结点个数为 $3f+1$。正常情况下 PBFT 协议的运行如图 1 所示。当主节点收到客户的请求后,向其他的从属节点发送 PRE-PREPARE 消息,当从属节点收到 PRE-PREPARE 消息后,如果其同意,那么向网络中的其他从属节点广播 PREPARE 消息,并收集来自其他节点的 PREPARE 消息,如果收到的 PREPARE 消息数目超过 $2f$,那么这个节点进入了准备就绪的状态,然后发送再次向其他节点广播 COMMIT 消息,并收集来自其他节点的 COMMIT 消息,如果 COMMIT 消息的数目超过 $2f$,那么这个节点进入了提交状态,开始执行客户发起的请求,操作完成后,向客户回复 REPLY 消息。

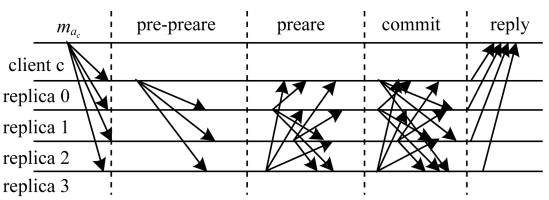

图 1　正常情况下实用拜占庭容错协议运行情况

可以看到 PBFT 协议最多只能容忍总结点数的 1/3 的出错节点,并且在一次同步过程中需要两轮两两交互,才能使得非出错节点达到一致,数据处理量较大。

文献[14]提出了混合法定人数(Hybrid

Quorum,HQ)来改进 PBFT,其运行情况如图 2 所示。HQ 将写入过程分成了 2 个阶段,首先客户端发起请求的同时收集服务器的状态信息,如果有 $2f+1$ 个节点处于相同状态,并且同意执行请求,则客户端第2 次发起请求,服务器开始执行请求;否则,执行 PBFT 协议,由主节点重新安排请求序号。

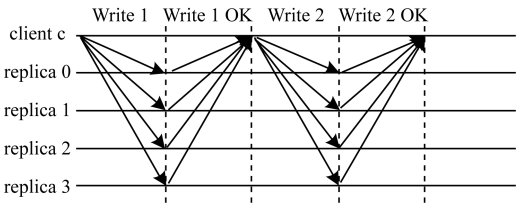

图 2 混合法定人数协议运行情况

HQ 协议并没有从根本上改变 PBFT 的结构,由于其对客户端发送请求没有竞争,服务器正常运行的情况进行了优化,使得其性能得到了提升。

PBFT 协议需要两轮的两两交互信息,才能容忍 1/3 的出错节点,在保证状态转换有效的前提下,如何快速将分布式系统从一个一致性状态同步到另一个一致性状态,是本文待解决的问题。本文提出与 PBFT 不同的法定人数(Quorum),即数目为 $n/2+1$,n 为系统中节点个数,在只需要一轮两两交互的情况下,容忍 38% 的出错节点。

2 系统模型

一个蒙特卡洛一致性协议[15]需要满足如下3 个条件:

1)可终结性,所有的过程需要在一个有限时间内做出决定。

2)一致性,所有的过程最终能够达到一致性。

3)输入的有效性,最终决定的值一定是过程产生的值。

2.1 消息传递过程

整个网络被看做 n 个独立配置的过程(*process*),$\{p_1,p_2,\cdots,p_n\}$。一个执行过程可以看做是一系列的状态和转换,$\pi_0,s_0,\pi_1,s_1,\cdots,\pi_i,s_i,\cdots$,从状态到可行的转换之间由以下规则决定:

每一个转换由一个单调递增的时间戳,一个时间戳称为一轮(*round*)。每一个过程在每一轮中只能执行一次 compute 指令。

匿名的过程之间通过 *broadcast*(m)指令来传递消息。每条消息的最大延迟为 Δ,如果消息在时间 r 广播,那么每一个过程在不超过 $r+\Delta$ 的时间里执行 *receive*(m)指令。

2.2 合法状态转换

计算模型引入了状态合法性验证后,原有的规则发生了变化。

2.2.1 状态转换

一个状态转换需要经过两阶段,从 s_0 转换到 s_1 需要经过中间状态 π_0,当一个节点发布了一个状态转换的消息后,收到这个消息的节点开始对这个转换广播自己的投票结果,同时节点开始收集消息,如果存在一个法定人数(Quorum)集合,里面所有的节点都同意,则网络能够被转换到状态 s_1。

2.2.2 不确定性

区块链协议使用了一种非交互式的公开可验证的谜题。每一个过程执行了一个 *coinflip* 指令产生一个随机数$[0,1)\subset\mathbb{R}$,$H:[0,1)\rightarrow[0,1)$是随机单向哈希函数。过程 p_i 对某一个中间状态的投票结果也是随机的,赞成的概率为 p',那么一个中间状态的表决结果为 $Binom(k,n,p')$,转换能够完成概率为 $vote=\sum_{k=n/2}^{n}\binom{n}{k}p'^{k}(1-p')^{n-k}$。

2.2.3 拜占庭错误

有 f 个出错的过程,这些拜占庭错误节点能够共谋对其他的正常节点发送错误消息。拜占庭出错过程反对正常过程,正常过程可能会赞成拜占庭过程。正常过程提出的方案通过的概率为:

$$u=\sum_{k=n/2}^{n-f}\binom{n-f}{k}p'^{k}(1-p')^{n-f-k}$$

拜占庭错误过程提出的方案能通过的概率为:

$$v=\sum_{k=n/2-f}^{n-f}\binom{n-f}{k}p'^{k}(1-p')^{n-f-k}$$

2.3 具有状态合法性验证的区块链算法

每一个过程在每一轮中执行一次 compute 过程,oracle 过程将任意的输入映射到 0 ~ 1 的区间,verify 过程验证事先给定的 *CRS*,挖矿节点提供的信息 *nonce*,区块的哈希值 *Prefer* 三者组合是否满足规则。一致性算法如下所示。

```
procedure oracle(x)
return H(x)
precedure verify(nonce,Prefer)
hash = oracle(CRS || Prefer || nonce)
    If hash/{0,1}^k ≤ p then return True
Initially do
Votes = {}
    Prefer = proposed_i
on receive(Votes',Prefer') do
    If verify(v,Prefer') and on receive_op(u,op) for all v
in Votes',all u in v and |Votes'| > |Votes| then
Votes = Votes'
        Prefer = Prefer'
    S = Prefer' - Prefer
    for u in S
        broadcast(u,op),op = {0,1}
on receive_op(u,op)
If count(op) >= n/2
        return True
```

```
    else return False
on compute(r) do
nonce = coinflip
    for u in Prefer
        If on receive_op(u,op)
set u.status = 1
    if verify(nonce,Prefer) then
        broadcast(Votes || nonce,Prefer)
if r >= r̂ then decide Prefer
```

算法过程如下:

每一个节点 P_i 同时执行 on receive、on receive_op 和 compute 过程。

on receive 过程当收到新的区块时,执行 verify 过程验证每一个区块是否满足工作量,对于区块中的每一个待完成的状态转换,执行 on receive 过程,验证其是否满足合法性验证。如果上述条件均满足,并且区块长度大于本地长度,则将这个区块作为新的区块。

对于新增区块中与原有区块链的差集中的中间状态,对其状态转换进行投票广播。

on receive_op 过程对收到的投票进行统计,如果超过半数,就返回成功,否则返回失败。

on compute 过程执行挖矿过程,首先调用 coinflip 过程得到 nonce,对于收到的待完成状态转换请求,调用 on receive_op 过程,修改其状态。随后调用 verify 过程,如果满足了正确的工作量证明,则广播这个区块。

2.4　一致性证明

在 n 个 process 中有 $f < n/2$ 个出错节点,假设一个过程能够解决一个难题的概率为 p,一个过程赞成某一个状态转换的概率为 p',那么正常过程能够完成状态转换的概率为 $p_c = (1-(1-p)^{n-f}) \times u$。出错的过程可以联合起来解决难题,一个出错的过程能够完成的概率转换为 $p_f = fpv$。

经过 $\hat{r}$ 轮后,正常过程完成的状态转换次数服从二项分布 $Block_c = Binom(p_c, \hat{r})$,出错过程次数服从另外的二项分布 $Block_f = Binom(p_f, \hat{r})$。

当 $np > 5, n(1-p) > 5$,使用 $N(np, np(1-p))$ 拟合 $Binom(n,p)$ 得到:

$$Block_c = N(\hat{r}p_c, \hat{r}p_c(1-p_c))$$

$$Block_f = N(\hat{r}p_f, \hat{r}p_f(1-p_f))$$

区块链协议中每个过程只会选择最长的链,协议需要保证 $Block_c - Block_f > 0$,由正态分布性质得到:

$$Block_c - Block_f = N(\hat{r}(p_c - p_f), \hat{r}p_c(1-p_c) + \hat{r}p_f(1-p_f))$$

而对于标准正态分布有如下性质:

$$prob = P\{u - k\sigma < X < u + k\sigma\}$$

当 $k=3$ 时,$prob = 99.73\%$,当 $k=4$ 时,$prob = 99.99\%$。

为了使 $\dfrac{\hat{r}(p_c - p_f)}{\sqrt{\hat{r}p_c(1-p_c) + \hat{r}p_f(1-p_f)}} > k$ 成立,需要满足:

$$\hat{r} > k\frac{p_c(1-p_c) + p_f(1-p_f)}{(p_c - p_f)^2}$$

当 $k=4$ 时,且 $\hat{r}$ 取上述值时,有 99.99% 的概率能够达到最终一致性。算法的可终结性由 $r > \hat{r}$ 决定。因此,蒙特卡洛最终一致性得证。

3　实验结果与分析

容易看到,当出错节点比例 f 越大时,$\hat{r}$ 越大,即算法需要运行更长时间才能保证一致性。

实验验证了上述算法能够在较少的运行时间内(即 $\hat{r} \leqslant 300$)达到最终一致性,并且通过实验数据计算出了相对确切的出错节点数目上限。

3.1　出错节点比例上限

假设正确节点和出错节点的提案有很大概率通过,令 $p \approx 1/n$,根据公式:

$$\hat{r} > k\frac{p_c(1-p_c) + p_f(1-p_f)}{(p_c - p_f)^2}$$

当 $n = 1\,000$ 时,通过调整 f 从 $n/4 \sim n/2 - 1$,观察图 3 中 $\hat{r}$ 随出错节点比例 f 变化的情况。当 f 小于 $0.39n$ 时,算法在较少时间内能够达到一致性。当 f 大于 $0.43n$ 时,算法同样在较少时间内能够达到一致性,但是此时系统的一致性是由出错节点达成的。而在 f 介于 $0.38n$ 和 $0.43n$ 时,算法需要运行很长的时间。

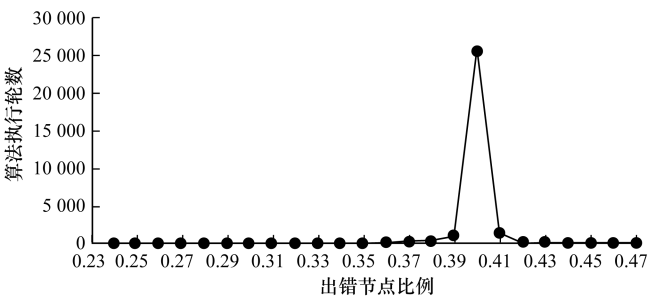

图 3　执行轮数随出错节点比例变化曲线

算法需要保证正常节点的转换率大于出错节点的转换率,这是系统正确达成最终一致性的必要条件。p_c 和 p_f 随出错节点比例 f 变化曲线如图 4 所示,可以看出,要使得 $p_c > p_f$,则 $f < 0.41n$。

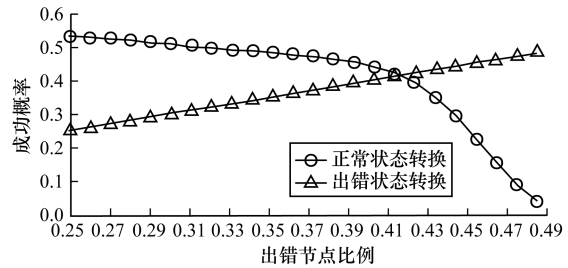

图 4 正确和出错状态转换概率随出错节点比例变化曲线

由于实验假定了 $\hat{r} \leqslant 300$,因此图 3 只有 $\hat{r} \leqslant 300$ 并且 $f < 0.41n$ 的部分是有效的,将其重新绘制如图 5所示。当 $f = 0.38n$ 时,算法能够在相对少的时间内达到一致性。

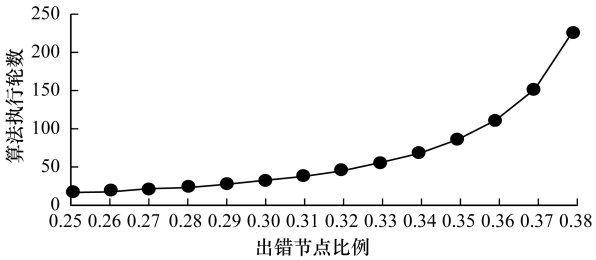

图 5 执行轮数随出错节点比例 $f(f < 0.38n)$ 变化曲线

3.2 赞成概率的选择

根据之前的实验,选取 $f = 0.38n$ 作为合理的出错节点个数。根据之前的公式:

$$u = \sum_{k=n/2}^{n-f} \binom{n-f}{k} p'^{k} (1-p')^{n-f-k}$$

$$v = \sum_{k=n/2-f}^{n-f} \binom{n-f}{k} p'^{k} (1-p')^{n-f-k}$$

将 p'作为自变量,绘制 u 和 v 随其变化的曲线,如图 6 所示。当 $p' > 0.75$ 的时候,u 和 v 差不多都趋近于 1。所以,当节点对于一个状态转换通过的概率至少为 0.75 时,上述算法才能够有效运行。

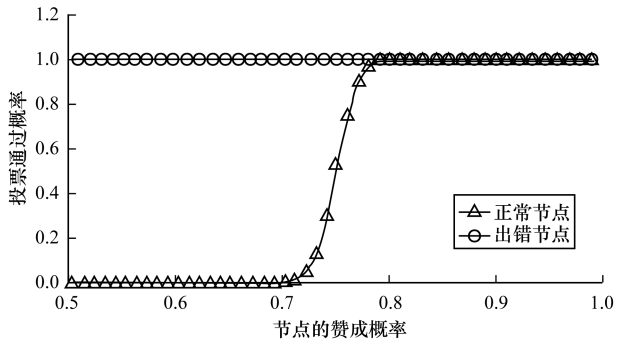

图 6 正常和出错节点投票通过概率随赞成概率变化曲线

4 结束语

本文分析具有状态合法性验证的拜占庭一致性协议的研究现状,介绍多节点合法性验证的一致性协议。针对其存在的总节点数 1/3 的容错上限和 2 次两两信息交换的问题,本文提出一种具有状态合法性验证的区块链一致性算法,设计新的法定人数集合,改进现有的区块链协议。实验结果表明,该算法使得区块链一致性算法能够具备合法性验证,并能解决拜占庭一致性,只使用了一次两两信息交换,使容错上限达到 38% 。

本文在网络规模、工作量难度和通信延迟给定的假设下,使用概率模型证明了算法的正确性。下一步工作将研究如何避免这些较强的假设条件,以提高算法的健壮性。

参考文献

[1] 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.

[2] DWYER G P. The Economics of Bitcoin and Similar Private Digital Currencies[J]. Journal of Financial Stability,2015,17(1):81-91.

[3] ALI M, NELSON J, SHEA R. Blockstack: A Global Naming and Storage System Secured by Blockchains[C]// Proceedings of USENIX Annual Technical Conference. [S.l.]:USENIX Association,2016.

[4] 杨胜刚,何 靖. 反洗钱领域大额与可疑信息报告制度的经济学分析[J]. 金融研究,2004,10(1):113-119.

[5] PLASSARAS N A. Regulating Digital Currencies: Bringing Bitcoin Within the Reach of IMF[J]. Chicago Journal of International Law,2013,14(1):377-407.

[6] 陈 晗,施 勇,薛 质. P2P 网络资源传播模型分析及监测[J]. 信息安全与通信保密,2011,9(5):56-58.

[7] LAMPORT L,SHOSTAK R,PEASE M. The Byzantine Generals Problem[J]. ACM Transactions on Programming Languages and Systems, 1982, 4(3): 382-401.

[8] CLEMENT A,KAPRITSOS M,LEE S. Upright Cluster Services[C]//Proceedings of the 22nd ACM SIGOPS Symposium on Operating Systems Principles. New York: ACM Press,2009:277-290.

[9] LAMPORT L. The Part-time Parliament[J]. ACM Transactions on Computer Systems,1998,16(2):133-169.

[10] BORRAN F,HUTLE M,SCHIPER A. Timing Analysis of Leader-based and Decentralized Byzantine Consensus Algorithms[J]. Journal of the Brazilian Computer Society,2012,18(1):29-42.

[11] 范 捷,易乐天,舒继武. 拜占庭系统技术研究综述[J]. 软件学报,2013,24(6):1346-1360.

[12] LAMPORT L. Time,Clocks,and the Ordering of Events in a Distributed System[J]. Communications of the ACM,1978,21(7):558-565.

[13] CASTRO M, LISKOV B. Practical Byzantine Fault Tolerance and Proactive Recovery[J]. ACM Transactions on Computer Systems, 2002, 20(4): 398-461.

[14] COWLING J,MYERS D,LISKOV B. HQ Replication: A Hybrid Quorum Protocol for Byzantine Fault Tolerance[C]//Proceedings of the 7th Symposium on Operating Systems Design and Implementation. [S.l.]: USENIX Association,2006:177-190.

[15] ATTIYA H. Lower Bounds for Randomized Consensus Under a Weak Adversary[J]. SIAM Journal on Computing,2010,39(8):3995-3904.

编辑 刘 冰