计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(8):90−97 [doi: 10.15888/j.cnki.csa.007537]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于区块链的持久标识符系统①

孙善鹏[1,2], 田　野[3], 李　琢[1,2], 王毅蒙[1,2], 刘　佳[1], 王　姝[1], 郭志斌[1]

[1](中国科学院 计算机网络信息中心, 北京 100190)
[2](中国科学院大学, 北京 100049)
[3](中国工业互联网研究院, 北京 100102)
通讯作者: 孙善鹏, E-mail: sunshanpeng@cnic.cn

摘　要: 随着信息产业的发展, 数据生产者产生了大量价值数据. 为了进行数据共享, 赋予数据相应的标识符用于解析数据所在位置,同时为使数据可长期通过标识符访问, 还需保证标识符解析服务长期可用, 但现有标识符系统多数采用半去中心化结构, 由于过于依赖最终解析服务, 其中部分系统由于各种原因逐渐丧失解析能力. 本文基于区块链系统的分布式账本数据一致性, 提出了一种基于区块链的持久标识符系统, 在兼容现有标识符系统访问层的基础上, 提供存储层以保证标识符解析服务持久性及数据的长期正确保存. 基于 Handle 系统及 Hyperledger Fabric 的测试结果表明, 该系统能够在提供可接受的请求响应速度与存储占用率的前提下, 为持久标识符服务提供更好的数据完整 性与解析服务长期可用性.

关键词: 持久标识符; 区块链; Handle 系统; 标识解析

引用格式: 孙善鹏,田野,李琢,王毅蒙,刘佳,王姝,郭志斌.基于区块链的持久标识符系统.计算机系统应用,2020,29(8):90–97. http://www.c-s-a.org.cn/1003-3254/7537.html

Persistent Identifier System Based on Blockchain

SUN Shan-Peng[1,2], TIAN Ye[3], LI Zhuo[1,2], WANG Yi-Meng[1,2], LIU Jia[1], WANG Shu[1], GUO Zhi-Bin[1]

[1](Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)
[3](China Academy of Industrial Internet, Beijing 100102, China)

Abstract: As the information industry develops, data producers have generated great masses of valuable data. Data identifiers are assigned for data sharing to resolve where the data is located. In order to make the data accessible through the identifier for a long time, it is necessary to ensure that the identifier resolution service is available for a long time. Majority existing identifier systems use a semi-decentralized structure, while some of them have gradually lost their resolution capacity due to reliance on the final resolution service. Based on the consistency of distributed ledger data of Blockchain system, this paper proposes a persistent identifier system based on Blockchain. On the basis of compatibility with the access layer of the existing identifier system, a storage layer is provided to ensure the durability of the identifier resolution service and the long-term correct storage of data. Test results based on the Handle system and Hyperledger Fabric show that it can provide better data integrity and long-term availability of resolution service for persistent identifier service under the premise of providing acceptable request response speed and storage occupancy.

Key words: persistent identifier; Blockchain; Handle system; identifier resolution

① 基金项目: 工业和信息化部 2018 年工业互联网创新发展工程 (工业互联网标识解析整体架构技术标准制定与试验验证); 工业和信息化部 2019 年工业互联网创新发展工程 (工业互联网标识解析公共服务支撑平台)
Foundation item: Year 2018, Industrial Internet Innovation Development Project of Ministry of Industry and Information Technology (Technical Standards Formulation and Verification for Industrial Internet Identifier Resolution Overall Architecture); Year 2019, Industrial Internet Innovation Development Project of Ministry of Industry and Information Technology (Public Service Support Platform for Industrial Internet Identifier Resolution)
收稿时间: 2020-01-14; 修改时间: 2020-02-13; 采用时间: 2020-02-25; csa 在线出版时间: 2020-07-29

在包括科学数据、工业数据在内的若干领域中, 数据总量迅速增长, 据 IDC 预测[1], 2025 年全球数据存储总量将达到 175 ZB, 其中大量价值数据需要被长期保存并分享, 这就需要为数据本身赋予某种名称, 使用户可以通过名称对数据本身进行访问.

对于在线数据而言, 往往通过 URL 对其进行命名并藉由 DNS 及相应 Web 服务向用户提供数据访问能力, 在这种方式下可以通过人为管理 URL 到数据的映射来保证用户通过特定 URL 访问数据的稳定性[2], 但当数据服务面临迁移、合并时, 通过原 URL 进行数据访问不可避免会遇到访问失败问题, 这就需要为其赋予持久化标识符将数据对象身份与其 Web 位置相分离以允许通过标识符明确指代对应数据对象, 从而保证标识符持久可解析, 为了达成这一目的, 出现了包括 Handle、ARK、PRUL 在内的超过 10 种标识符系统尝试为数据提供名称及解析服务, 但随着时间推移, 其中部分系统不再提供服务, 造成这种情况的主要原因一方面是其标识符系统结构过于依赖于中心化基础设施[3], 另一方面是标识符服务提供者大多未找到合适的盈利模式以至于不可持续发展. 因此需要加强现有标识符系统的解析服务长期可用性与标识符数据完整性, 最终保证标识符解析持久性, 同时由于现有标识符系统服务规模普遍较小, 无法像 DNS 系统一样提供广泛的缓存基础设施, 所以在标识符解析流程中还需通过与客户端物理位置接近的服务节点为客户端提供解析服务, 以加速标识符解析速度.

因此我们设计了基于区块链的持久标识符系统, 并兼容现有标识符解析系统进行了初步实现: 通过基于区块链的存储层使多节点间互相形成冗余, 允许提供多节点解析服务以避免单点服务失效以保持解析服务的持久性, 并通过多节点中的近地节点服务优化标识符解析速度, 同时通过区块链链式结构及其共识机制保证标识符数据安全性.

本文余下章节中, 第 1 部分将介绍持久性标识符和区块链的相关研究工作, 第 2 部分将介绍基于区块链的持久标识符, 第 3 部分将对所提方案进行系统设计并于第 4 部分中进行系统实验结果分析, 第 5 部分对基于区块链的持久性标识符的下一步发展以及挑战做了简要分析并总结全文.

1 相关工作

如何保证标识符解析服务持久性及数据正确性一直是域名、身份隐私及数据领域研究者长期关注的问题.

Namecoin[4] 基于区块链提供了类似 DNS 解析的功能, 通过向类比特币[5] 系统的交易事务数据中添加额外信息来将.bit 域名映射到指定位置, 借助于区块链系统的去中心化特性避免网络审查以保证域名解析服务持久性从而允许信息自由发布, 但由于 Namecoin 易于遭受 51% 攻击，Ali 等重新设计了可跨链的 Blockstack 系统[6], 以保证域名解析数据源可靠性, 虽然 Namecoin 及 Blockstack 对 DNS 解析服务持久性及数据源可靠性做出了初步尝试, 但其与现有域名系统相互独立, 需要用户使用另外的解析系统. HandShake[7] 基于区块链系统通过链上具有根区数据的完全节点代替现有 DNS 根区文件与根区服务器, 从而对 DNS 顶级域进行管理, 同时使用轻量级节点用于资源证明, 从而希望能与现有 DNS 体系兼容形成完整的 DNS 解析体系, 但该系统仅为代替根区服务.

此外, PPK 技术社区在 2015 年设计了基于比特币的奥丁号 (Open Data Index Name, ODIN)[8], 它使用区块号+块内序号作为标识符来唯一确定一个区块内事务, 如 600 000.100 表示将标识符信息写入第 600 000 块内第 100 条事务, 从而使解析可以直接定位到区块中具体的数据. W3C 工作组也在 2014 年开始进行去中心化标识符的讨论[9], 并在 2019 年成立了去中心化标识符工作组发布了首个公开工作草案[10], 以期设计一种符合 URI 规范的可验证的去中心化标识符以独立于任何中心化验证机构.

为了解决持久标识符系统单点故障问题, Bolikowski 等[11] 提出了将标识符及其对应数据与拥有者公钥信息保存在区块链中的持久标识符生成和管理方案, 试图通过区块链系统解决单点故障对标识符寿命及长期保存造成的影响, 但在该方案中每个区块仅存放单一用户的标识符, 这将导致每区块数据密度过低对存储资源造成了浪费. Golodoniuc 等[12] 设计了一种类似 P2P 文件共享网络基于 DHT 式结构以 Magnet Link 作为名称的持久标识符系统, 期望在所有级别上去中心化来增加系统容错能力从而保证其可靠性. Sicilia 等[13] 对持久标识符模型进行建模并描述了如何将标识符及其元数据存储于 IPFS[14] 及以太坊[15]为标识符提供持久化服务.

上述工作初步阐述了如何通过去中心化系统增加系统可靠性从而保证标识符解析持久性，但这些工作都对标识符系统进行了重新设计，无法在兼容现有系统的情况下，使客户端直接迁移到去中心化的标识符解析系统.

2 基于区块链的持久标识符模型

传统标识符系统解析服务在解析流程中依赖于系统的分布式结构而具有一定的鲁棒性, 但最终解析行为往往依赖于单一数据源, 这使得解析服务面临着末端服务单点失效导致解析失败的可能, 现有系统主要依靠构建冗余服务来解决这一问题. 同时, 虽然标识符系统协议可能在数据报文中定义了消息凭据通过类似 DNSSEC 的机制提供签名验证能力来保证解析数据在其传输过程中的真实性与完整性, 但当数据源被攻击后, 无法保证数据本身的安全性从而解析错误的标识符数据.

而基于区块链作为标识存储层为成员间数据一致性做出了保证, 使系统参与节点间形成冗余, 当用户发起解析请求时, 所有系统参与节点皆可完成对请求的响应, 这样就避免了单节点崩溃或不再服务所导致的该节点命名空间内标识符解析失效情况. 同时当单节点崩溃事故发生后, 崩溃节点只需重新接入区块链网络, 待区块链数据同步完成后扫描全部数据重建标识符索引, 即可再次提供服务. 在数据安全性上, 区块链的链式结构也简化了数据正确性验证流程, 增加了攻击者的数据篡改成本, 同时结合具体系统所采用的共识机制保证正确数据的一致性同步, 保障了存储节点内标识符数据的安全性.

2.1 模型概述

为了兼容现有标识符系统, 图 1 中基于区块链的持久标识符模型总体上分为标识符系统访问层、区块链服务层两层结构. 现有标识符系统访问层继续保持对外提供标识符解析与管理服务的能力, 区块链服务层则基于智能合约为标识系统访问层提供区块链存储服务及对块内标识符数据进行操作的能力, 同时基于共识算法为节点间数据一致性提供保障.

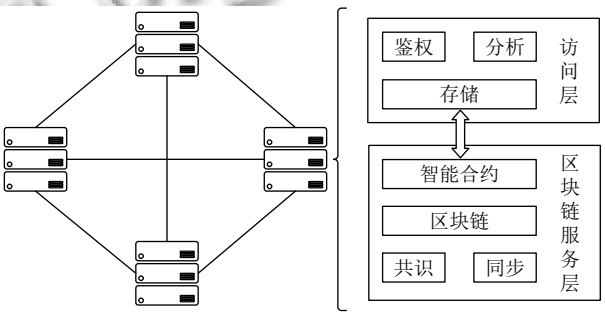

图 1 基于区块链的持久标识符模型

2.2 区块概述

由于对现有标识符系统访问层进行重用, 所以将不会直接基于区块链为用户提供标识符管理能力, 因此不会将管理者密钥等数据直接存储至区块结构中, 具体结构如图 2 所示.

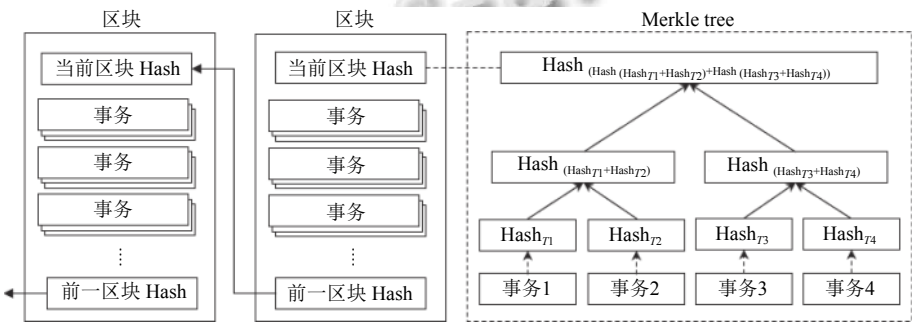

图 2 持久标识符的区块设计

图 2 中每个区块将存储多个操作符事务数据以增加区块利用率, 并存储基于事务数据形成的 Merkle Tree 根节点值作为当前区块 Hash 值, 该值是基于当前区块内所有事务数据的 Hash 值构造而成 (如图 2 右侧所示), 同时每一区块将存储前一区块 Hash 值以形成链式结构, 从而允许对数据完整性与正确性进行验证, 若某事务数据被攻击者修改后, 将影响包括被篡改数据所在区块 Hash 值及其后的所有区块中的前继 Hash

值, 若当前区块链链长度为 L, 被攻击区块号为 X, X 块内事务数据量为 M, 则攻击者篡改单节点内某一标识符数据的时间复杂度 T 为$\log_2 M+1+L-X$.

针对区块事务内的标识符数据而言, 在其生命周期内将经历如图 3 所示的状态转换, 主要会涉及 3 种状态, 包括标识符发布、标识符修改、标识符删除.

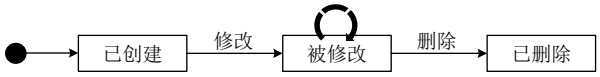

图 3 标识符生命周期状态转换图

具体地讲, 3 种状态总共涉及 5 种操作: 创建标识符、新增值、更改值、删除值、删除标识符. 为了缓解区块链存储压力, 不应在链上存储每次标识符操作后完整标识符数据, 而是存储当前操作造成的数据变化,即进行增量式的存储, 因此当标识符被创建时, 存储其初始数据, 而当进行后续操作时, 并不完全重新存储全部标识符数据, 而仅需存储变更数据同时附带指向上一版本标识符的指针即可.

上述标识符操作中, 标识符创建和删除仅会被执行一次或 0 次, 其余标识符值相关操作次数虽然与值数据变更频度有关, 但对于单个标识符而言, 其不同类型的操作次数有限且较少的, 这为增量存储但快速组合成完整标识符数据提供了有利条件, 因此一条事务数据具体包括:

(1) 标识符;

(2) 前继标识符数据位置;

(3) 标识符值.

基于增量式标识符存储使对标识符修改记录进行内容追溯成为可能, 当数据标识符使用者将元数据存储在标识符系统中时, 即可根据历史元数据生成相应的数据描述变化信息, 允许数据科学家从数据关系等角度对其进行进一步的分析.

但通常情况下单个标识符数据量较少, 若将每次操作都封装成单个事务写入区块则会面临事务结构数据 (事务头、签名等) 比事务内有效负载数据更大的情况, 因此应设计某种策略将符合策略的标识符数据批量封装至同一事务内以节省频繁存储事务结构信息带来的存储空间浪费, 本文中事务封装策略通过事务结构大小 ST 与事务负载大小 SP 确定, 当 $SP/ST>1$ 时将会封装当前所有等待数据至同一事务中写入区块.

3 基于区块链的标识符系统实现

现有持久标识符系统大多基于 Handle 系统[16–18]提供服务, 虽然该系统被设计为半中心化结构, 使系统中某单节点临时失效或完全不可用的情况下不会影响到系统整体解析流程的平稳、不间断运行,但这种单点失效情况依然影响了该节点所负责的命名空间内标识符的可解析性, 本节将在接下来阐述如何将 Handle 系统移植到基于 Hyperledger Fabric[19] 的区块链之上以使其成为在存储层面可去中心化的系统.

3.1 系统结构

为了兼容现有 Handle 系统, 首先需要使其保持现有的 GHR、LHS 式层级结构, 以保证用户进行标识符解析时解析流程不被改变. 其次对于传统 Handle 服务提供者而言, 依赖于 Key-Value 数据库或关系型数据库提供对 Handle 及其值的单节点存储服务, 这种单节点存储及访问服务存在的不可靠性带来了使解析服务崩溃风险, 而当数据存储层迁移至区块链之上后, 区块链参与节点将得益于全局一致性账本互为冗余, 同时可以提供多节点全量标识符解析服务. 从系统结构上看, 考虑到现有 Handle 节点间服务的互相独立, 应允许原有的 Primary 节点、Multi-Primary 节点、Mirror 节点都可以与基于区块链的 Handle 节点同时共存于整个体系之中, 最终系统整体结构如图 4 所示.

以 Hyperledger Fabric 作为区块链服务为例, 持久化标识符联盟中的每个参与成员都将在各自的 Handle 服务中集成 Fabric Peer 节点以代替原有数据存储系统, 将各自身份注册至 MSP 之中, 以确保对区块链系统的操作权限, 并基于 Raft 协议使参与成员提供排序服务, 其区块链网络结构如图 5 所示.

3.2 存储层次结构

对于一个具体的标识符服务节点而言, 其标识符注册数量通常处于较大的数量级, 所以对于每个标识符服务都需要具有类似标识符目录的机制对当前服务节点管理的标识符进行索引从而允许解析服务快速的定位标识符数据所在区块以读取数据, 因此将索引数据存储于 Hyperledger Fabric 世界状态数据库中, 使其与具体区块链数据文件分离, 如图 6 所示.

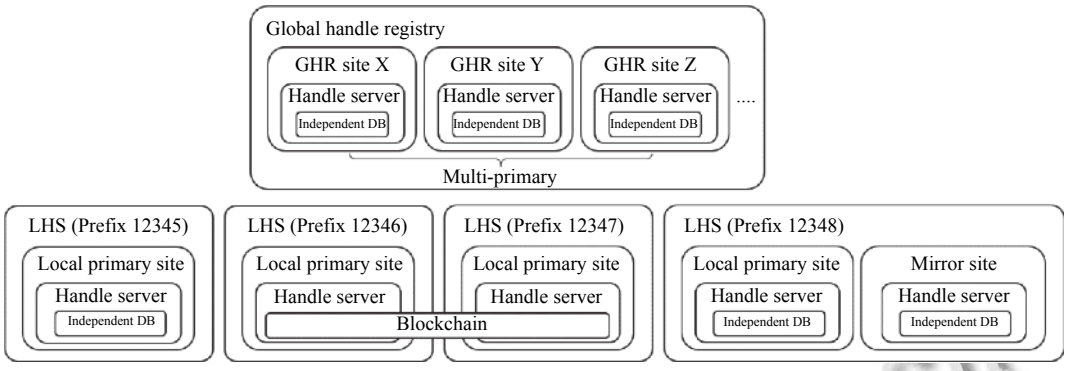

图 4 包含基于区块链服务的 Handle 系统结构

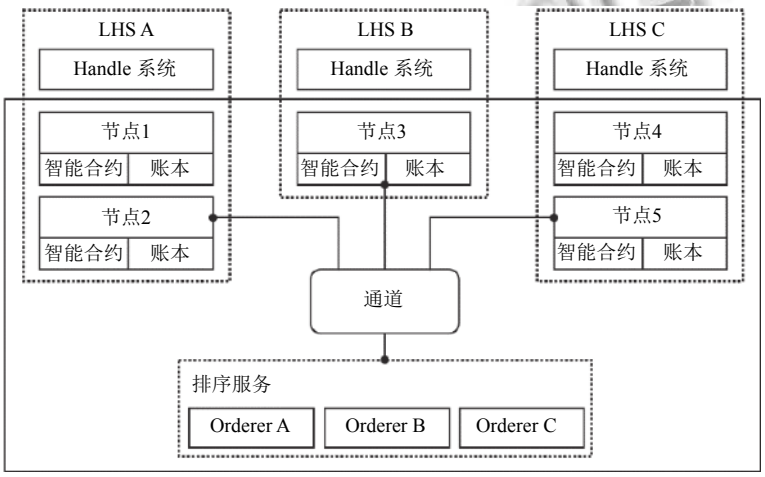

图 5 区块链网络结构

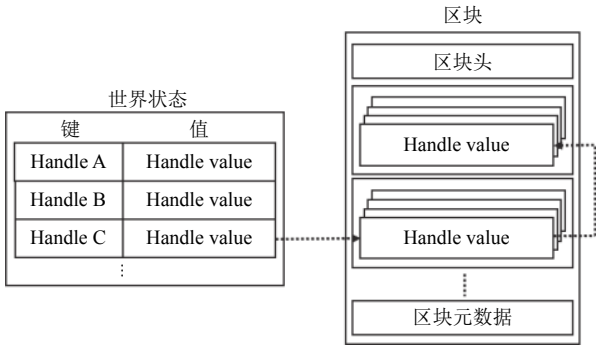

图 6 存储层次结构

为了不影响 Handle 系统用户的现有信息系统, 基于区块链的标识符系统将继续遵循 Handle 协议规范规定的 Handle 标识符数据结构以提供完整的 Handle 数据存储, 基于 2.2 节中所描述的方案进行标识符数据存储, 其中 Handle 号及最新数据存储在世界状态数据库之中, 因此事务内有效数据仅包括:

(1) 前继标识符信息位置 (除创建标识符操作外), 包括事务 ID 与在该事务中的写集下标;

(2) Handle 值列表, 包括 Index、Type、Value、TTL、Timestamp、Reference、admin_read、admin_write、pub_read、pub_write.

3.3 标识符事务

以 12346/abc 为例, 其具体事务存储示例如图 7.

图 7(a) 代表 handle 号为 12346/abc 的 handle 被创建, 其 handle 值包括一条 Index 为 1 的 URL 类型的值 http://hdl.handle.net, 其 TTL 为 24 小时, 创建时间为 927314334000, 引用为空, 管理员可读且可写, 公众可读但不可写.

图 7(b) 代表对标识符 12346/abc 添加一条新值, 它的前继操作存储在事务 ID 为 100 的事务内写集下标为 1 的位置, 值内容是 Index 为 2 的 DES 类型的值 “Handle resolver”, 其 TTL 为 24 小时, 创建时间为 927314335000, 引用为空, 管理员可读且可写, 公众可

读但不可写.

图 7(c) 代表是对标识符 12346/abc 中 Index 为 2 的值进行修改, 其前继操作存储在事务 ID 为 101 的事务内写集下标为 1 的位置, 具体值被修改为“A Handle resolver”.

图 7(d) 为删除 12346/abc 中 Index 为 2 的值, 其前继操作存储在事务 ID 为 102 的事务内写集下标为 1 的位置.

(a)

1	URL	http://hdl.handle.net	24 h	927314334000	NULL	1	1	1	0

(b)

100	1	2	DES	Handle resolver	24 h	927314335000	NULL	1	1	1	0

(c)

101	1	2	DES	A Handle resolver	24 h	927314336000	NULL	1	1	1	0

(d)

102	1	2

图 7 标识符操作事务

3.4 标识符解析与操作

Handle 系统具有 GHR/LHS 两层结构, 并且基于 Handle 协议规范具有详细的数据交互流程设计, 因此不论在解析还是操作流程上都应遵循基本的 Handle 交互模式.

(1) 标识符解析流程

解析基本流程如图 8 所示.

当用户通过解析器向系统发起标识符解析请求时: ① 首先会通过解析器向 GHR 请求 LHS 服务地址; ② 接着 GHR 将返回负责用户欲解析的标识符前缀信息, 其中包含了所有参与区块链的成员节点信息, ③ 接下来解析器将向获取到的 LHS 发送具体解析请求; ④ LHS 接收到解析请求后首先检索缓存, 若命中则直接返回标识符数据; ⑤ 若不命中则调用存储层; ⑥ 触发当前节点集成的 Fabric 服务的标识符检索智能合约; ⑦ 合约将检索世界状态数据库得到最新标识符状态数据, 若该数据标志位非新建标志或删除标志则说明检索到的仅为最新增量数据; ⑧ 于是需要依据最新增量数据的前继数据索引字段去区块链存储中相应的区块检索历史数据继而组合出完整标识符数据; ⑨ 之后将其写入缓存并返回给访问层, 其后访问层基于用户是否具有相应值条目的读取权限筛选出最终数据, 以响应解析器的请求.

(2) 标识符操作流程

当解析器发起标识符操作请求时, 整体流程与解析请求流程类似, 但首先需经历操作权限验证, 其中标识符创建与标识符值修改/删除流程略有不同:

① 标识符创建: 当请求到达 Handle 系统后, 若请求为标识符创建请求则首先根据 Handle 规范验证其权限.

② 标识符值修改/删除: 首先进行标识符检索, 其后根据标识符数据进行权限验证.

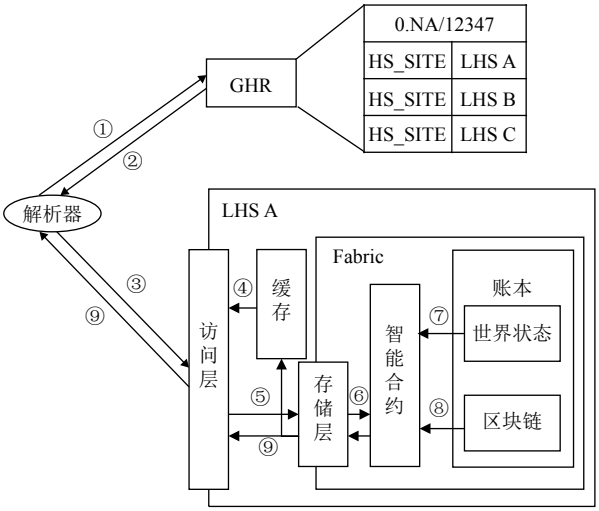

图 8 解析请求流程示意图

权限验证通过后将触发当前节点智能合约, 节点成功背书后, 即可将数据写入缓存提供服务, 以保证服务时效性. 其后当前节点 Orderer 服务将数据发送至当前的 Raft Leader 处, Leader 将该请求包含的 Handle 数据依照前文第 3 节所述格式打包成事务, 同步至其它 LHS 的对等节点中.

4 实验与分析

4.1 实验环境与方案

实验环境主要采用一台 CPU 规格为 AMD R5 3500U, 内存规格为 DDR4 2133 MHz, 硬盘规格为 WDC SN520 的主机作为宿主机, 使用 VirtualBox 6.0 进行虚拟化提供运行环境, 基于 Handle Software v9.2 与 Hyperledger Fabric v1.4 进行基于区块链的标识符系统开发, 对照组

为基于 BerkeleyDB 及 MySQL 的 Handle 系统.

上述基于 Hyperledger Fabric 的系统中包含由 5 个 LHS 节点以组成持久标识符区块链系统, 在进行测试时, 使用单客户端 10 线程, 每个线程发送基于 TCP 发送 2000 次请求, 每个请求之间有 10 毫秒的延迟以避免客户端主机端口耗尽, 同时停用标识符系统缓存. 测试项目主要包括标识符解析性能测试、标识符操作性能测试、标识符服务单点失效测试、标识符服务内存消耗测试、标识符存储资源消耗测试.

4.2 实验分析

4.2.1 标识符解析性能测试

在标识符解析性能实验中, 测试脚本通过向系统解析预先创建的标识符来进行解析性能测试, 测试结果如表 1 所示.

表 1 解析性能测试

系统类型	解析吞吐量(解析量/s)	平均请求响应时间(ms)
基于BerkeleyDB	714	3.3
基于MySQL	322	20
基于Hyperledger Fabric	93	97

从表 1 中可见基于 Hyperledger Fabric 的 Handle 系统标识符解析吞吐量略低, 其平均解析响应时间慢于基于其他存储层的 Handle 系统, 这意味着对于需要高性能的标识符解析场景而言, 仍需进一步优化标识符解析流程同时需提供高效的缓存服务以减少标识符解析时的区块链区块文件的查询次数.

4.2.2 标识符操作性能测试

在标识符操作性能实验中, 测试脚本通过向系统批量创建标识符来进行操作性能测试, 测试结果如表 2 所示.

表 2 操作性能测试

系统类型	操作吞吐量(操作量/秒)	平均请求响应时间(ms)
基于BerkeleyDB	294	23
基于MySQL	80	112
基于Hyperledger Fabric	36	276

可见基于 Hyperledger Fabric 的 Handle 系统对的标识符操作响应时间慢于基于其他存储层的 Handle 系统, 需要进一步优化标识符存储流程以提供多级数据写入策略来保证系统对标识符操作的及时响应.

4.2.3 标识符单点失效测试

为了进行单点失效测试, 在实验中将设置 5 个持久标识符服务参与节点, 分别编号 A~E, 在节点运行过程中, 对标识符服务进行持续解析请求, 在此过程中随机反复停止并恢复节点的服务, 行为示例如表 3 所示, 解析客户端持续统计解析成功率, 结果如表 4 所示. 测试结果表明基于区块链的标识符解析系统通过互冗余为任一参与方 LHS 所管理的前缀命名空间内标识符提供多节点的解析服务可以有效避免单一 LHS 的解析服务单点失效所带来的解析失败问题.

表 3 单点失效测试行为示例

序号	行为/节点	节点A	节点B	节点C	节点D	节点E
1	启动全部	正常	正常	正常	正常	正常
2	停止节点A	正常	正常	正常	正常	正常
3	恢复节点A	正常	正常	正常	正常	正常
4	停止节点C	正常	正常	正常	正常	正常
5	恢复节点C	正常	正常	正常	正常	正常

表 4 单点失效解析测试

系统类型	结果
请求次数	20000
成功次数	20000
失败次数	0
成功率	100%

4.2.4 标识符存储资源消耗测试

基于区块链的标识符系统存储了所有标识符历史数据, 同时每条数据都被打包成事务以组成区块, 这将存储更多的事务及区块结构信息, 所以会比传统 Handle 系统消耗更多的存储资源, 存储资源消耗对比如图 9 所示, 标识符存储受事务及区块结构数据影响较大, 可按需进一步考量存储策略及数据压缩策略以节省存储资源.

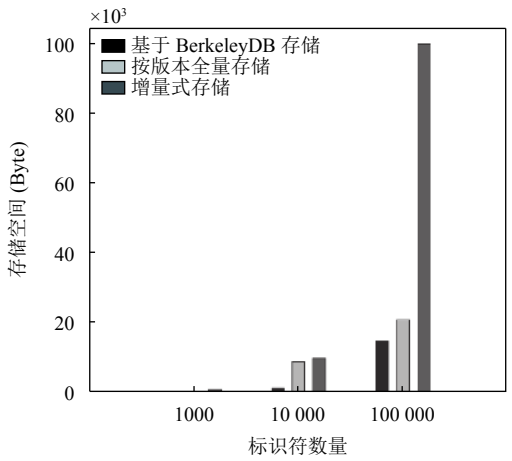

图 9 存储资源消耗对比图

5 结论与展望

本文阐述了一种基于区块链的持久标识符系统并基于 Handle 系统对其进行了初步实现. 通过对已有标识符系统访问层的兼容, 使现有系统从技术上获得了延续性, 同时通过区块链系统的分布式一致性账本优势使原有系统获得更多的数据完整性与解析服务长期可用性, 从而保障标识符解析持久性. 然而, 基于区块链的持久标识符系统具有大量的标识符服务参与成员, 且参与成员的标识符注册量较多, 这将对其成员服务节点造成过多的存储压力, 在接下来研究中我们将继续探究如何对压缩标识符数据同时改善事务打包策略或基于链外存储方式, 以期节省存储资源, 同时优化解析流程从而保持高性能的解析能力.

参考文献

1 Reinsel D, Gantz J, Rydning J. IDC's "Data Age 2025" whitepaper. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.

2 Berners-Lee T. Cool URIs don't change. Cambridge, MA: World Wide Web Consortium (W3C). http://www.w3.org/Provider/Style/URI.

3 Huber R, Klump J. How dead is dead in the PID Zombie Zoo? https://www.rd-alliance.org/sites/default/files/attachment/20160902-RDA_EU_View_on_PID_Systems_Garching-Robert_Huber-Jens_Klump-How_dead_is_dead_in_the_PID_Zombie_zoo.pdf.

4 Namecoin. https://namecoin.info.

5 Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf. 2019.

6 Ali M, Nelson J, Shea R, *et al*. Blockstack: A global naming and storage system secured by blockchains. Proceedings of 2016 Annual Technical Conference. Denver, CO, USA. 2016. 181–194.

7 Handshake project. https://www.handshake.org.

8 PPk Public Group. ODIN (Open Data Index Name). A peer trusted name system based blockchain. http://ppkpub.org/.

9 Longley D, Sporny M, Lehn DI. Web payments community grouptelecon.https://web-payments.org/minutes/2014-05-07/. (2014-05-07).

10 Burnett D, Zundel B. W3C decentralized identifier working group charter. https://www.w3.org/2019/09/did-wg-charter.html.

11 Bolikowski Ł, Nowiński A, Sylwestrzak W. A system for distributed minting and management of persistent identifiers. International Journal of Digital Curation, 2015, 10(1): 280–286. [doi: 10.2218/ijdc.v10i1.368]

12 Golodoniuc P, Car NJ, Klump J. Distributed persistent identifiers system design. Data Science Journal, 2017, 16: 34. [doi: 10.5334/dsj-2017-034]

13 Sicilia MA, García-Barriocanal E, Sánchez-Alonso S, *et al*. Decentralized Persistent Identifiers: A basic model for immutable handlers. Procedia Computer Science, 2019, 146: 123–130. [doi: 10.1016/j.procs.2019.01.087]

14 Benet J. IPFS-content addressed, versioned, P2P file system. arXiv: 1407.3561, 2014.

15 Wood G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014. 1–32.

16 Sun S, Lannom L, Boesch B. Handle system overview. RFC 3650, 2003.

17 Sun S, Reilly S, Lannom L. Handle system namespace and service definition. RFC 3651, 2003.

18 Sun S, Reilly S, Lannom L, *et al*. Handle system protocol (ver 2.1) specification. RFC 3652, 2003.

19 Androulaki E, Barger A, Bortnikov V, *et al*. Hyperledger fabric: A distributed operating system for permissioned blockchains. Proceedings of the 13th EuroSys Conference. Porto, Portugal. 2018. 30.