软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2019,30(9):2655−2670 [doi: 10.13328/j.cnki.jos.005774]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 存储容量可扩展区块链系统的高效查询模型*

贾大宇[1], 信俊昌[1,2], 王之琼[3], 郭 薇[4], 王国仁[5]

[1](东北大学 计算机科学与工程学院,辽宁 沈阳 110819)

[2](辽宁省大数据管理与分析重点实验室,辽宁 沈阳 110819)

[3](东北大学 中荷生物医学与信息工程学院,辽宁 沈阳 110819)

[4](沈阳航空航天大学 计算机学院,辽宁 沈阳 110136)

[5](北京理工大学 计算机学院,北京 100081)

通讯作者: 信俊昌, E-mail: xinjunchang@mail.neu.edu.cn

**摘 要**: 区块链技术是目前计算机领域的研究热点,其实现了去中心化,并且能够安全地存储数字信息,有效降低现实经济的信任成本.提出一种区块链存储容量可扩展模型的高效查询方法——ElasticQM.此查询模型由用户层、查询层、存储层和数据层这 4 个模块组成.在用户层,模型将查询结果缓存,加快再次查询相同数据时的查询速度;在查询层,模型采用容量可扩展区块链模型的全局查询优化算法,增加了查询超级节点、查询验证节点和查询叶子节点这 3 种节点角色,提高了查询效率;在存储层,模型改进了区块链的容量可扩展模型 ElasticChain 的数据存储过程,实现了存储的可扩展性,并减少了占用的存储空间;在数据层,提出一种基于 B-M 树的区块链存储结构,并给出了 B-M 树的建立算法和基于 B-M 树的查找算法,基于 B-M 树的存储结构,区块链会在进行块内局部查找时提高区块链的查询速度.最后,通过在多节点不同数据量的区块链中查询的实验结果表明,ElasticQM 查询方法具有高效的查询效率.

**关键词**: 区块链;查询算法;容量可扩展;B-M 树;ElasticQM

**中图法分类号**: TP311

中文引用格式: 贾大宇,信俊昌,王之琼,郭薇,王国仁.存储容量可扩展区块链系统的高效查询模型.软件学报,2019,30(9):2655−2670. http://www.jos.org.cn/1000-9825/5774.htm

英文引用格式: Jia DY, Xin JC, Wang ZQ, Guo W, Wang GR. ElasticQM: A query model for storage capacity scalable blockchain system. Ruan Jian Xue Bao/Journal of Software, 2019,30(9):2655−2670 (in Chinese). http://www.jos.org.cn/1000-9825/5774.htm

## Efficient Query Model for Storage Capacity Scalable Blockchain System

JIA Da-Yu[1], XIN Jun-Chang[1,2], WANG Zhi-Qiong[3], GUO Wei[4], WANG Guo-Ren[5]

[1](School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China)

[2](Liaoning Provincial Key Laboratory of Big Data Management and Analytics, , Shenyang 110819, China)

[3](Sino-Dutch Biomedical and Information Engineering School, Northeastern University, Shenyang 110819, China)

[4](School of Computer, Shenyang Aerospace University, Shenyang 110136, China)

* 基金项目: 国家自然科学基金(61472069, 61402089, U1401256, 61732003, 61729201); 辽宁省自然基金(0170540702); 中国博士后科学基金(2018M641706); 中央高校基本科研业务费(N161602003)

Foundation item: National Natural Science Foundation of China (61472069, 61402089, U1401256, 61732003, 61729201); Natural Science Foundation of Liaoning Province (20170540702); China Postdoctoral Science Foundation (2018M641706); Fundamental Research Funds for the Central Universities (N161602003)

本文由“区块链数据管理”专题特约编辑于戈教授、牛保宁教授、金澈清教授推荐.

收稿时间: 2018-06-09; 修改时间: 2018-08-28; 采用时间: 2018-12-14; jos 在线出版时间: 2019-04-10

CNKI 网络优先出版: 2019-04-09 17:46:28, http://kns.cnki.net/kcms/detail/11.2560.TP.20190409.1746.011.html

[5](School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

**Abstract**: Blockchain technology is a research hotspot in the field of computers today. The decentralized and secured blockchain data effectively reduces the trust costs of the real economy. This study proposes an efficient query method for the scalable model of blockchain storage capacity—ElasticQM. The ElasticQM query model consists of four layers of modules: user layer, query layer, storage layer, and data layer. The user layer model puts the query results into the cache, which speeds up the query speed when querying the same data again. In the query level, this study proposes a global query optimization algorithm for the scalable blockchain model, which increases the roles of querying super nodes, query verification nodes and querying leaf nodes. It improves the efficiency of global queries. In the storage layer, the model improves the data storage process of the ElasticChain, which supports large scale blockchain. The storage layer achieves the scalability of the blockchain's capacity and reduces the storage space. In the data layer, this study proposes a blockchain storage structure based on B-M tree, and gives the establishment algorithm of B-M tree and search algorithm based on B-M tree. Blockchains based on B-M trees will increase the speed of queries in local search within a block. The experimental results on real datasets show that the ElasticQM model has efficient query efficiency.

**Key words**: blockchain; query algorithm; scalable capacity; B-M tree; ElasticQM

区块链技术起源于 2008 年化名为"中本聪"(Satoshi nakamoto)的学者在密码学邮件组发表的奠基性论文《比特币:一种点对点电子现金系统》[1].区块链技术具有去中心化、透明性、开放性、自治性、匿名性和信息不可篡改等特点[2],被认为是继大型机、个人电脑、互联网、移动社交网络之后,计算范式的第 5 次颠覆式创新,是人类信用进化史上继血亲信用、贵金属信用、央行纸币信用之后的第 4 个里程碑[3].区块链技术为解决中心化机构普遍存在的高成本、低效率和数据存储不安全等问题提供了解决方案[4].

但目前存在储存扩展性较差的问题,以迄今为止最为成功的区块链应用场景比特币为例,截至 2018 年 5 月 6 日,产生了521 534个区块[5]、17 019 个比特币,总容量为 174.34GB.截至 2017 年 5 月 7 日,链上已认证地址 9 892 723 个[6],并且经过一年多大家对比特币关注度的增加,已认证地址数量大幅提升.在区块链系统中,节点有效保证区块链数据安全的方法是作为完全节点保存完整的区块链数据.如果目前比特币系统中所有节点都作为完全节点,那么近 1 000 万个节点各自提供近 200GB 的磁盘空间来储存区块链数据,整个系统将占用约 2 000PB 的存储容量保存 200GB 左右的数据,这极大地浪费了存储空间.在未来,区块链技术将会被大规模地应用,预计到 2027 年,全球 10%的 GDP 将会通过区块链技术存储[7].随着区块链容量的不断增加,参与节点的存储容量将逐渐不能满足其存储空间要求,这些不能满足要求的节点就不能继续作为完全节点保留在系统中.随着系统中完全节点数量的减少,对区块链系统的安全性必将产生影响,如:基于工作量证明(proof of work,简称 POW)[8]机制的区块链系统的总算力就会下降;基于股权证明(proof of stake,简称 POS)[9]机制系统中的股权容易集中;基于委任权益证明(delegated proof of stake,简称 DPOS)[10]机制的区块链系统更容易被少数节点控制等.同时,如果不具备足够存储空间的节点不能加入到区块链系统中,区块链的可扩展性也会降低.因此,区块链具有良好的存储容量可扩展性是非常重要的.

目前,对于区块链存储容量可扩展性的研究不是很多,文献[11]提出了名为 ElasticChain 的区块链模型,该模型在有效保证数据安全的前提下,增加了区块链的存储容量.但是在原有的区块链模型中,全节点保存着完整的区块链数据,在查询数据时,全节点只需要在本地磁盘进行查找操作.而 ElasticChain 模型在增加存储容量之后,模型中的大部分节点没有保存全部的区块链数据,所以当一个节点发起查询操作后,它将访问系统中其他大量节点,遍历一条完整的区块数据.因此,ElasticChain 模型与原区块链模型相比,数据的查询效率明显降低.同时,由于 ElasticChain 模型在查询时数据来自不同节点,系统中也存在一些恶意节点返回的虚假数据的现象,这也给数据查询的准确度和数据的安全造成一定的影响.随着区块链技术的广泛应用,人们对区块链中数据查找速度和准确度的要求会越来越高,没有有效的数据查询方法,将会对未来区块链技术的广泛应用带来巨大限制.

因此,本文在ElasticChain模型基础上提出一种新的容量可扩展区块链系统的高效查询方法——ElasticQM (elastic query model).ElasticQM 的框架共有 4 层,分别是用户层、查询层、存储层和数据层.

- 在用户层中,包括了数据缓存、数据验证和数据同步等模块:数据同步模块保证了数据的时效性;数据缓存模块将查询过的数据缓存在本地磁盘中,增加再次查询该数据的查询速度;

- 在查询层中,ElasticQM 结合了 P2P 网络超级节点查找技术,提出了容量可扩展区块链模型的全局查询优化算法.通过建立具有高可靠性和高稳定性的查询超级节点,在模型响应数据查询请求时,优先访问查询超级节点,在保证数据安全的前提下,提高了数据查询效率;
- 在存储层中,模型采用基于 ElasticChain 的区块链模型,保证了区块链数据的容量可扩展性;
- 在数据层中,我们提出了 B-M tree 的数据存储结构.该存储结构结合了平衡二叉树(balanced binary tree)[12]和梅克尔树(Merkle tree)[13]的各种特点,在保证区块数据验证速度的同时,提高了每个区块内的局部查询速度,并使区块链支持数据范围查询.

本文的主要贡献如下.

(1) 提出了一种区块链存储容量可扩展模型的高效查询方法——ElasticQM.此查询模型由用户层、查询层、存储层和数据层共 4 层模块组成;

(2) 在 ElasticQM 查询层,提出了一种容量可扩展区块链模型的全局查询优化算法,增加了查询超级节点、查询验证节点和查询叶子节点这 3 种模型中的角色,提高了查询效率;

(3) 在 ElasticQM 数据层,提出了一种基于 B-M 树的区块链存储结构,并给出了 B-M 树的建立算法和基于 B-M 树的查找算法.区块链基于 B-M 树的存储结构,会在区块链进行的块内局部查找时,提高区块链的查询速度;

(4) 通过在不同节点可扩展区块链中对不同数据量区块进行的查询实验结果表明,ElasticQM 查询方法具有高效的查询效率.

## 1 相关工作

近年来,对于区块链技术的研究越来越受到人们关注.首先,在提高区块链中数据查询效率方面,文献[14]开发了基于 Ethereum 的高效查询层 EtherQL.EtherQL 会自动同步区块链系统中新的区块数据,并将其存储在专用数据库中,以确保查询准确性和高效性.同时,EtherQL 提供了比其他区块链应用系统更高效、更灵活的数据查询接口,并且支持对区块链数据的范围查询和 top-$k$ 查询.在区块链系统性能评价方面,文献[15]首次提出了区块链应用的评价框架——Blockbench.文献中, Blockbench 在 Hyperledger Fabric 和以太坊的两个客户端(Parity 和 Geth)这 3 个区块链私有链平台上,评价了每个平台在吞吐量、延迟、可扩展性和容错方面的性能,并做了详细的比较和分析.并且,区块链技术在未来也是计算机、数据库领域的研究热点.文献[16]通过查阅大量区块链系统的框架和应用情况,分析出未来在区块链数据管理和分析方面的主要研究方向:区块链充分利用现有成熟的数据和信息系统的方法、增强区块链数据安全性和隐私性的有效方法、在区块链上和区块链下的数据分析方法和如何使基于区块链的系统更加活跃和智能.

其次,在保证区块链数据安全性方面,文献[17]提出了基于股权证明(POS)的区块链协议 Ouroboros Praos.该协议通过设置安全的数字签名和一种新型可验证随机函数,来保证在半同步状态下区块链的安全性.文献开发了一个通用的基于新协议的区块链模型,并通过建立随机预言机模型进行模拟实验,验证了新协议的高安全性.文献[18]提出 ForkBase 数据库存储引擎,ForkBase 解决了在数据容易产生分叉或分歧的系统中出现的多版本、交叉语义和恶意篡改攻击等问题.同时,文献提出了名为 POS-Tree 的管理大型数据对象的索引结构,减少了系统的存储开销.文献评估了 ForkBase 与区块链平台、wiki 服务和一个协作分析应用软件 OrpheusDB 等 3 个具有代表性的复杂应用各自的性能,展示了 Fork-Base 的可用性和高效性.文献[19]分析了比特币以外的区块链的安全性和网络可靠性,分析发现,Namecoin 区块链系统中拥有单个矿工在数月里的计算能力超过全网的 51%,这是区块链系统中非常严重的安全隐患.因此,文献提出了开源软件 Blockstack,在 Blockstack 框架中设计了 4 层架构——区块链层、虚拟层、路由层和数据存储层,避免了 Namecoin 区块链系统检测出来的安全问题.文献[20]提出了量化框架来分析基于 PoW 共识机制的区块链系统在不同网络参数下的安全性.框架分析了区块链系统在实际应用中存在的网络传播、不同区块大小、区块生成时间间隔、信息传播机制以及日食攻击(eclipse attacks)等影响系统安全性的约束.设计了应对双重支付(double-spending)和自私挖掘(selfish mining)的最优对抗策略.

## 2 预备知识和问题定义

### 2.1 区块链的区块结构和查询方法

目前,区块链应用的主要模型架构基于中本聪的论文[1],如图 1 所示,在每个区块中,存储了版本号、上一个区块的哈希值、本区块产生的随机数、时间戳、难度值和由交易信息生成的 Merkle 树的根.

- 版本号也是该区块的区块号;
- 上一个区块的哈希值是上一个刚刚生成区块的区块头通过 SHA256 算法生成的哈希值,并填入到当前区块中;
- 本区块产生的随机数是在区块链工作量证明机制下,区块链各个节点根据上一个区块的哈希值,并通过反复尝试来找到的符合设定哈希函数的随机数;
- 时间戳记录了该区块产生的时间,标识了该区块的唯一性;
- 难度值会根据之前一段时间区块的平均生成时间进行调整,以应对整个网络不断变化的整体计算总量,如果计算总量增长了,则系统会调高数学题的难度值,使得预期完成下一个区块的时间依然在一定时间内;
- Merkle 树的根是由区块主体中所有交易的哈希值再逐级两两哈希计算出来的数值,主要用于快速检验一笔交易是否在这个区块中存在[13].当系统验证区块链中是否包含某笔交易时,只需要一个交易哈希,一个 Merkle 树根哈希和一个 Merkle 路径,实现了区块链的简易支付验证(simplified payment verification,简称 SPV)[21].

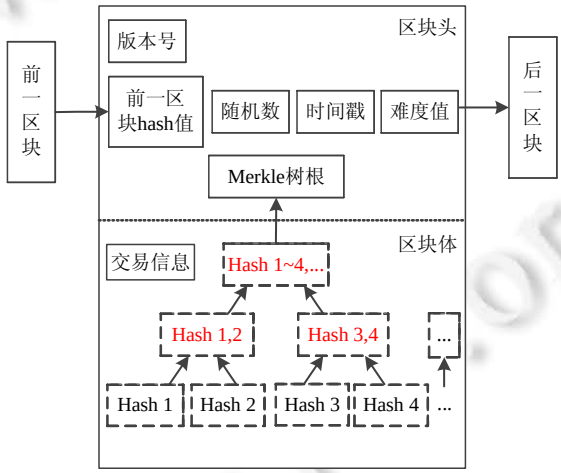



Fig.1 Structure of blocks

图 1 区块结构

因为在区块链的机制中,要求每个完全节点必须存储一个区块链的完整副本.当查询其中一条交易信息时,会在本地完整的区块链副本中进行遍历查询.但随着越来越多的人使用区块链系统,不太可能每个人都去运行全节点,一部分区块链的用户会选择使用轻量级钱包.轻量级钱包不会保存区块链的数据,当其发起查询请求时,需将查询请求发送到相邻全节点,在全节点中进行查询操作,并将查询结果返回给轻量级节点.

### 2.2 区块链存储容量可扩展模型的结构和查询方法

目前,对于增加区块链可扩展性的研究还不是很多.文献[11]提出了区块链存储容量可扩展模型 ElasticChain,将一条完整的区块链副本进行分片处理,并将分片数据保存在一定比例的节点中,提升了区块链的存储容量.ElasticChain 模型首先采用了区块链数据副本分片策略,计算出了安全、合适的每个分片大小和分片

被存储的副本数.然后,ElasticChain 模型增加了验证节点对存储数据的节点进行基于 POR(数据可检索性证明)方法的实时检测,并记录更新存储节点稳定性值,依此选择高稳定性节点来储存新产生的数据副本,提高了数据存储的稳定性.

如图 2 所示,ElasticChain 模型包括了用户节点、储存节点和验证节点.用户节点是原始数据拥有者,储存节点是副本的保存者,而验证节点是储存节点稳定性的验证者.一个节点可以同时具备两种或者 3 种角色.当用户节点查询区块链数据时,需要首先访问保存本地的 P 链(P 链存储区块链数据分片存储的位置),查找到分片数据保存的相应存储节点.然后访问存储节点,将每个分片数据逐一返回到用户节点.最后才能在返回数据中进行查询操作.

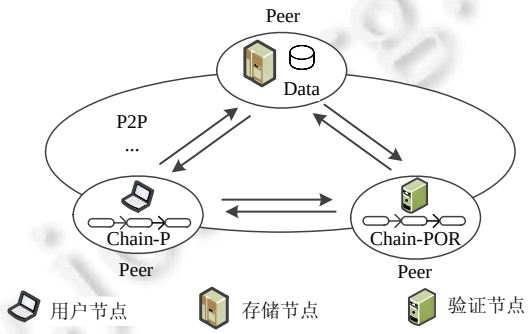



Fig.2 Structure of ElasticChain model

图 2 ElasticChain 模型架构

### 2.3 问题定义

令 $S$ 表示包含 $n$ 个区块的一条完整的区块链数据集合,即 $S=\{B_1,B_2,...,B_n\}$.并且每一个区块 $B$ 被表示为元组 $B=\langle H,K\rangle$,其中,$H$ 表示区块 $B$ 的区块头信息;$K$ 表示区块 $B$ 中的交易数据,交易数据 $K$ 是由 $m$ 个交易组成的集合,即 $K=\{T_1,T_2,...,T_m\}$.每一个交易 $T$ 被表示为一个元组 $T=\langle V,O,N,R,E\rangle$,其中,$V$ 表示此时交易 $T$ 的版本号, $O$ 表示交易 $T$ 发起者的地址,$N$ 表示交易的数额,$R$ 表示交易 $T$ 接收方公钥的哈希,$E$ 表示交易的其他信息.一个区块中第 $i$ 个交易可以表示为 $T_i=\langle V_i,O_i,N_i,R_i,E_i\rangle$.

本文要解决的问题是,当区块链 $S$ 存储在区块链存储容量可扩展模型中,进行基于交易发起者的地址 $O$、交易的数额 $N$、交易接收方地址 $R$ 的条件查询查询时,提高其查询速度.在本文中,以对地址为 $O_i$ 的节点所发起的所有交易进行查询操作为例进行阐述说明.

## 3 ElasticQM 查询模型总览

基于 ElasticChain 区块链存储容量可扩展模型,我们提出了在扩展模型上的高效查询方法——ElasticQM. ElasticQM 查询模型一共由 4 层模块组成,分别是用户层、查询层、存储层和数据层.ElasticQM 模型架构如图 3 所示.

- 用户在发起查询请求后首先访问用户层,在用户层的缓存数据中进行查找:如果找到了相应数据,则停止查找,返回查询结果;如果没有在用户层缓存中找到查询结果,则访问模型查询层;
- ElasticQM 查询层会根据容量可扩展区块链模型的全局查询优化算法找到查找到数据所在区块;
- ElasticQM 的存储层则是可扩展区块链的基于 ElasticChain 的区块存储方式,实现了区块链数据的存储可扩展性;
- 最后,ElasticQM 在数据层提出了基于 B-M tree 的区块数据存储方式,提高了区块链块内查找效率.

在 ElasticQM 模型中,区块链系统既实现了存储容量的可扩展性,又在不影响区块链去中心化特性和安全性的前提下,实现了区块链数据的全局和局部的快速查询.

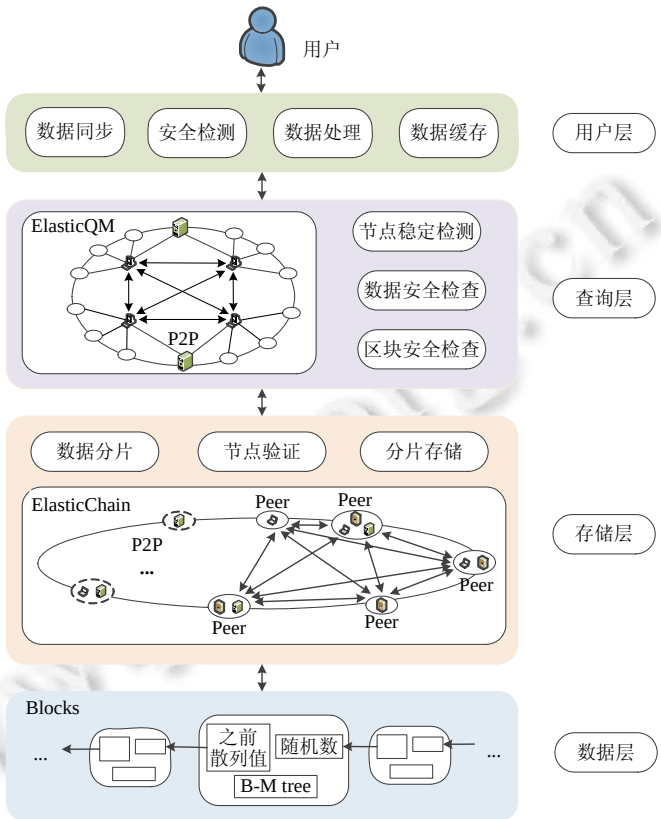



Fig.3 Structure of ElasticQM modle

图 3 ElasticQM 模型架构

## 4 ElasticQM 查询模型设计细节

用户通过 ElasticQM 查询模型 4 层模块的处理,快速得到查询结果.

- 首先,在 ElasticQM 查询层中,用户在执行查询操作后,ElasticQM 会与区块链数据实时同步,并经过数据验证缓存在支持 SQL 查询的数据库中,第 4.1 节将具体描述数据查询过程;
- 然后,在 ElasticQM 查询层中,结合 P2P 网络超级节点查找技术,提出了容量可扩展区块链模型的全局查询优化算法.在算法中,将区块链节点分为了查询超级节点、查询叶子节点和验证节点,叶子节点在查询数据时会优先访问查询超级节点,并将返回数据进行区块安全检验和数据未被篡改检验.模型中的验证节点实时记录节点的稳定性,来确定是否可以作为超级查询节点.查询操作详情可以在第 4.2 节看到;
- 其次,在 ElasticQM 存储层中,基本采用了文献[11]中 ElasticChain 模型的数据存储方法,实现了区块链的容量可扩展存储.但在 ElasticChain 模型的节点可靠性验证部分进行了改进.模型中,用户节点不再存储保存着分片存储路径的 P 链,减少了 ElasticChain 模型的存储空间,存储过程在第 4.3 节详细描述;
- 最后,在 ElasticQM 数据层中,为了能高效查询区块链数据中一个区块内的某一条数据,我们将区块里的数据保存在了 B-M tree 中,提高了每个区块内的局部查询速度.相关存储过程在第 4.4 节描述.

### 4.1 用户层

在 ElasticQM 用户层中,当一个节点发起查询请求后,会首先访问 ElasticQM 用户层缓存数据:如果节点找到

相应数据,则停止查找,返回查询结果;如果节点没有在用户层中找到查询结果,则访问模型 ElasticQM 查询层进行查找操作.

因此,ElasticQM 在用户层中设计了数据同步、安全检测、数据处理和数据缓存这 4 个模块.数据缓存模块的作用是:当用户层在每个节点成功完成一次查询之后,将查询结果缓存在用户层缓存模块中.用户层缓存模块是一个持 SQL 查询的数据库,因此在下一次查询相同数据时,ElasticQM 就可以在 SQL 数据库中进行查找,增加了查询速度.但是,区块链数据每时每刻都在不断地更新和变化,这就需要缓存模块的数据也随这不断更新.用户层中的数据同步模块就会在每经过一个相同的时间段,对数据缓存模块中的数据与区块链中数据进行实时的同步和更新.在数据更新过程中,用户层的安全检查模块会对新增数据进行安全检验,保证数据的真实性.通过安全检测模块的数据,才能被保存在数据缓存模块中.而数据处理模块则是在用户层经过数据同步和数据安全检测后,将区块链数据处理成可以保存在 SQL 数据库中的形式.例如在比特币交易系统中,将进行比特币交易的买方地址、卖方地址和交易数量以表格的形式分别存入数据缓存模块,加快下次查找速度.

### 4.2 查询层

ElasticQM 的查询层模块接收模型用户层发送的查询请求,再根据查询层中的查询算法,在模型的存储层快速找到相应的数据返回给用户层.在 ElasticQM 查询层中,我们提出了基于容量可扩展区块链模型的全局查询优化算法.该算法结合了 P2P 网络中基于 Super-peer 的分布式拓扑结构,将可扩展区块链模型中的节点分为了查询超级节点、查询叶子节点和查询验证节点,查询层结构如图 4 所示.

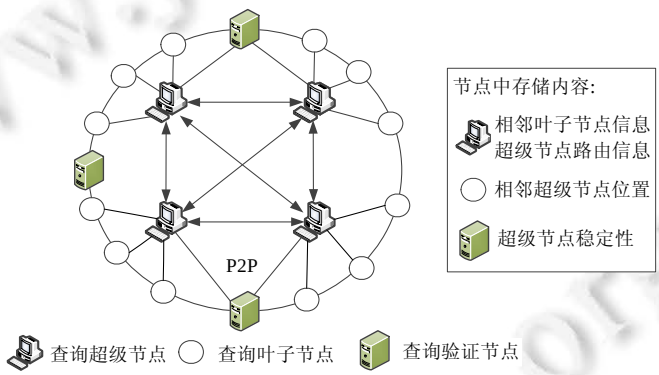



Fig.4 Architecture of query layer in ElasticQM

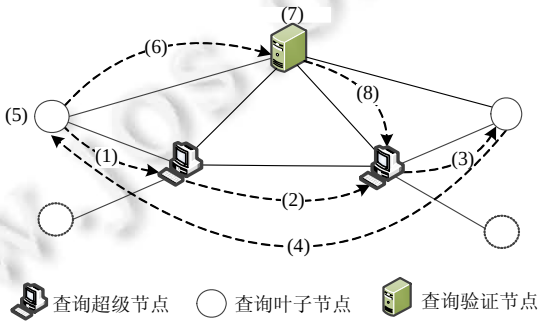
图 4 ElasticQM 查询层架构

模型中,查询超级节点上存储了系统中相邻叶子结点的信息和超级节点间的路由信息.当进行查询操作时,发现算法仅在查询超级结点之间转发,超级结点再将查询请求转发给适当的叶子结点.模型中,查询叶子节点至少保存着区块链每个区块的区块头数据,来验证收到的查询数据是否被篡改,这个与比特币钱包中的轻量级钱包相似.同时,查询叶子节点保存着相邻查询超级节点位置,在查询数据时,如果本地没有完整的区块链数据,则会优先访问网络中查询超级节点.每一个新加入区块链系统的节点,都会先被视为查询叶子节点.查询验证节点和区块中每个查询超级节点和活跃的查询叶子节点相连,实时记录相连节点的可靠性.

查询验证节点会根据记录的查询超级节点是否出现恶意篡改区块链数据来判断超级节点的安全性,根据记录的查询超级节点在线时间和工作量得出超级节点的稳定性,根据记录的查询超级节点工作速度得到超级节点的处理能力.超级节点的工作量和工作速度通过查询叶子节点实时向验证节点反馈,查询验证节点会根据查询超级节点的安全性、稳定性和处理能力决定其是否继续作为查询超级节点.同时,查询验证节点还会实时检查整个区块链数据,对于在区块链上操作较多的活跃的查询叶子节点,验证节点也会记录其节点的可靠性,作为查询超级的候补节点.

ElasticQM 查询层采用了容量可扩展区块链模型的全局查询优化算法,如图 5 所示.当一个节点收到查询请

求时,节点首先判断自身是否为查询超级节点:如果该节点不为查询超级节点,则会访问距离最近的查询超级节点,然后,该超级节点根据本地保存路由信息,找到接近查找目标的超级节点;如果该节点为查询超级节点,则直接访问本地路由信息,找到接近查找目标的超级节点.接着,由这个接近查找目标的超级节点找到与它相连的保存着需要查找的数据的叶子节点,将查询结果同查询结果所在的区块和与之相连的其他区块的区块头作为最终查询结果.最后,将最终查询结果按查找时的相同路径返回给发送查询请求的节点.当最初送查询请求的节点收到了查询数据后,首先要通过本地保存的区块链的区块头数据与返回数据做对比,检验查询结果所在的区块链没有在查询过程中被篡改过.然后,查询发起节点通过对查询结果的哈希值与查询结果所在区块的区块头哈希值进行校验,检验查询结果是在其区块中的真实数据,未被恶意篡改.查询发起节点会将检验结果返回给验证节点,使验证节点能够实时地监督查询路径上的超级节点和叶子节点的可靠性程度.如果出现某一节点恶意篡改数据的情况,验证节点则会减少该节点的可靠性值;如果数据查询结果正常,则增加路径上所有节点的可靠性值.可靠值低的节点将不能继续作为查找超级节点,而可靠性较高的叶子节点可以成为新的查询超级节点.




(1) 发起查询请求,发送给相邻超级节点; (5) 验证查询结果的真实性;
(2) 找到接近查找目标的超级节点; (6) 将查询结果真实性发送给验证节点;
(3) 找出查询目标节点; (7) 验证节点记录查询路径节点可靠性值;
(4) 将查询结果按查询路径返回; (8) 验证节点排除可靠性较低的超级节点.

Fig.5 Global query optimization algorithm for query layer in ElasticQM

图 5 ElasticQM 查询层的全局查询优化算法

容量可扩展区块链模型的全局查询优化算法的具体过程见算法 1.

**算法 1**. 容量可扩展区块链模型的全局查询优化算法.

输入:发起查询节点 $A$,查询条件:$O_i$,$N_i$ 或 $R_i$;

输出:查询结果,交易集合 $T$.

1. **if** (节点 $A$ 是查询叶子节点)
2. {访问距离节点 $A$ 最近的查询超级节点 $B$;
3. 根据节点 $B$ 保存的路由信息,按照查询条件 $O_i$,$N_i$ 或 $R_i$,找到接近查找目标的超级节点 $C$;}
4. **else**
5. {根据节点 $A$ 保存的路由信息,按照查询条件 $O_i$,$N_i$ 或 $R_i$,找到接近查找目标的超级节点 $C$;}
6. 超级节点 $C$ 找到和它相连的保存着需要查找的数据的叶子节点 $D$;
7. 叶子节点 $D$ 将查询结果、查询结果所在的区块和与之相连的其他区块的区块头作为最终查询结果,并将最终查询结果按查询路径返回给节点 $A$;
8. 节点 $A$ 检验查询结果所在区块链的安全性;
9. 节点 $A$ 检验查询结果为区块中真实数据;
10. **if** (节点 $A$ 校验发现查询结果被恶意篡改)
11. {
12. **for** 查询路径上每个节点;

```
13.        {
14.          if (节点为恶意节点)
15.            {节点 A 将信息反馈给查询验证节点;
16.            在验证节点中,将该恶意节点可靠性降低;
17.            }
18.          if (该恶意节点为查询超级节点)
19.            if (该恶意节点可靠值不能够胜任作为查询超级节点)
20.              {将该超级节点转变为叶子节点;
21.                if (恶意节点附近的叶子节点可靠性值足够高)
22.                  {该叶子节点转变为查询超级节点;}
23.              }
24.        }
25.    }
26. else
27.    {节点 A 查询成功,得到查询结果,交易集合 T;
28.    节点 A 将查询成功信息发送给验证节点;
29.    验证节点增加查询路径上的所以节点可靠性值;}
```

ElasticQM 的查询层模块中,查询超级节点只有在区块链系统中发起查询操作请求时,才会被优先访问.查询层的超级节点在区块链数据存储时没有任何优先级,与网络中其他节点相同.因此,查询层中的超级节点不会影响整个可扩展区块链系统的安全性和去中心化的特征.

在算法 1 的步骤 8 中,由于 ElasticQM 模型中一些节点没有保存着一条完整的区块链数据,因此节点需要在访问完整的区块链数据后,才能验证查询结果的安全性.不在节点内存储的区块链数据,将由模型中其他节点(查询超级节点、查询叶子节点和查询验证节点)共同恢复并验证.ElasticQM 的查询层采用全局查询优化算法后,避免了目前 ElasticChain 模型查询区块数据时采用泛洪方法在系统中盲目查找的过程,减少了数据查找对区块链网络带来的巨大处理压力,并有效地减少了查询所需时间,增加了查询效率.

ElasticQM 在查询层采用全局查询优化算法后,同样可以保证区块链系统对数据的一致性要求.当模型中的超级节点将本地的路由信息恶意修改或丢失,附近的叶子节点将不能访问这个超级节点或超级节点返回错误路径.在这种情况下,叶子节点会访问附近其他超级节点进行数据查询.如果系统中存储节点将本地的数据丢失或篡改,在查询发起节点收到查询结果后,会根据本地存储的区块链哈希头的数据,对查询结果的哈希值进行验证.如果发起节点发现数据错误或不完整的情况,将会把验证信息发个查询超级节点,超级节点确认后,将恶意的存储节点从路由表中删除.因此,节点在 ElasticQM 在查询层中可以达到共识.

### 4.3 存储层

ElasticQM 的存储层响应查询层的查询请求,并发送请求到数据层进行数据查询.在 ElasticQM 存储层中,基本采用了文献[11]中 ElasticChain 模型的数据存储方法,实现了区块链的容量可扩展存储.ElasticQM 的存储层主要包括了数据分片、节点验证和分片存储这 3 个部分.在数据数据分片和分片存储部分,ElasticQM 的存储层采用了和 ElasticChain 模型相同的算法,得到分片的方法、每个分片的大小和分片的副本数.但存储层在节点可靠性验证部分对 ElasticChain 模型进行了改进:在 ElasticQM 存储层中的用户节点完成每次数据分片存储后,不再存储 P 链来记录分片的存储位置.其他节点的可靠性验证过程都与 ElasticChain 模型相同.

ElasticChain 模型中,用户节点保存 P 链数据是为了快速查询数据,但是 P 链占据了大量的存储空间.经过 ElasticQM 模型的改进,在区块链上的查询操作交给了模型查询层完成,因此减少了区块链模型的存储空间,进一步增加了区块链的存储容量可扩展性.

### 4.4 数据层

目前,区块链数据在每个块中以梅克尔树的形式存储.梅克尔树的特点是一个数据利用其生成的哈希值,可以快速地验证是否存在于区块链中.当用户想要访问区块链中的一条具体数据信息时,对于一个完全节点就需要遍历区块内利用梅克尔树存储的全部数据.但是随着区块链应用的广泛普及,区块链中保存的数据量也会急剧增加,在一条完整的区块链上进行数据查询,效率随之越来越慢.因此,本节提出了基于 B-M 树的区块链存储结构,既实现了梅克尔树的特点(一个节点可以在不下载整个块的情况下,验证区块中是否包含某笔交易),又提高了在一条完整区块链上的数据查询效率,并使区块链支持数据范围查询.

基于 B-M 树的区块链存储结构结合了平衡二叉树和梅克尔树的的各自特点,其节点的数据结构如图 6 所示.一个 B-M 树的节点包括了数据序列化后的哈希值或合并后的哈希值(hash)、其包含所有叶子节点记录发起者地址的最大值(max)和最小值(min)、该位置在平衡二叉树映射节点地址(*K*)和指向叶子节点的左指针(*L*1)与右指针(*R*1).

| *L*1<br>左指针 | *K*<br>平衡二叉树<br>映射节点地址 | min<br>叶子节点<br>地址最小值 | hash<br>哈希值/合并哈希值 | max<br>叶子节点<br>地址最大值 | *R*1<br>右指针 |
|---|---|---|---|---|---|

Fig.6 Structure of B-M tree

图 6 B-M 树数据结构

在 B-M 树建立过程中,首先,系统确认在一个区块产生的固定时间里写入区块的数据;然后,根据每个用户地址数值的大小,建立起平衡二叉树,保证树中每个节点左右两个子树的高度差的绝对值不超过 1;最后,从树的底部开始,逐层地将这个平衡二叉树的叶子节点的哈希值两两进行合并,组成新的哈希值,并同时保存着所有合并的记录发起者用户地址的最大值和最小值、该位置在平衡二叉树映射节点地址和指向叶子节点的左指针与右指针.模型会不断重复这个过程,直到合并成一个哈希值,并将其保存在区块头.在哈希值两两合并过程中,从平衡二叉树的底部开始,左叶子节点和其父节点先做一次合并,然后将得到的合并哈希值与右叶子节点再次合并,得到父节点和左右两个叶子节点的合并在一起的哈希值,直到合并成一个哈希值.这样,B-M 树的存储结构就被建立起来,B-M 树详细的建立算法见算法 2.

**算法 2**. B-M 树建立算法.

输入:一串交易数据的数组;

输出:B-M 树存储模型.

1. 确认写入区块的数据;
2. 根据每个记录发起者的用户地址数值大小,建立起一个平衡二叉树(假设共 *N* 层,第 *N* 层一共有 *n* 个叶子节点);
3. **if** (*n* 为奇数)
4. {最后一个叶子节点复制一份;}
5. **for** (*N*>1) //建立 B-M 树
6. {
7. **for** (*n*>2)
8. {
9. 第 *N* 层左叶子节点和其父节点先做一次合并;
10. 第 *N* 层合并结果再与右叶子节点再次合并;
11. 记录此处节点的 max 值、min 值、*K* 值;
12. *L*1,*R*1 分别指向左右叶子节点;
13. *n*=*n*−2;

14. }

15. $N=N-1$;

16. }

17. 将最终合并的结果作为 B-M 树根,保存在区块头中;

同时,我们通过举例具体阐述 B-M 树的建立过程.当记录发起者的用户地址为 4,7,8,10,14,20,25,30,40 时,B-M 树的建立过程如图 7 所示.

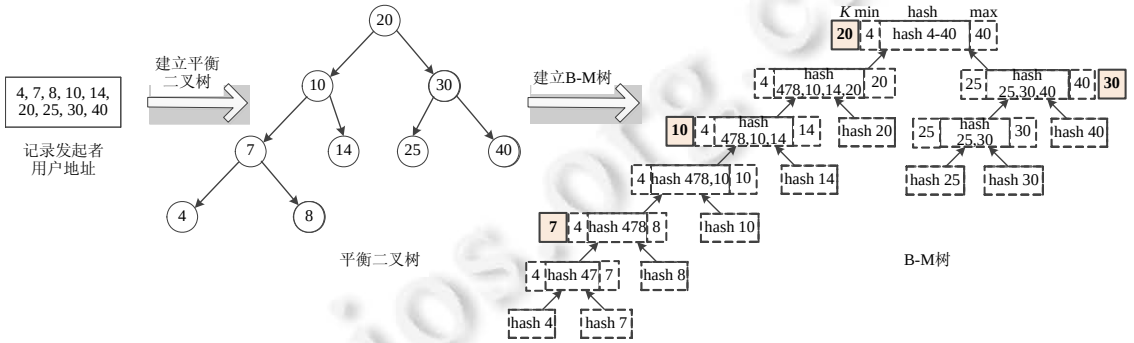



Fig.7 Establishment process of B-M tree

图 7 B-M 树建立过程

当某节点发起一个范围查询时,首先,如果这个节点是完全节点,则在本地查询;如果不是完全节点,则连接一个完全节点,在这个完全节点中进行查询.在完全节点中查询数据时,首先从新区块到旧区块的顺序遍历每个区块的区块头中 B-M 树的树根,根据 B-M 树根中所有叶子节点的记录发起者地址最大值和最小值,判断要查询数据是否存在于这个区块中:如果不在树根的最大值和最小值范围内,则校验下一个区块;如果在范围内,则根据 B-M 树中保存的平衡二叉树映射节点地址 $K$ 进行基于平衡二叉树查找算法的搜索,直至找到相关数据,将结果返回给发起查找节点.如果搜索完整个 B-M 树还没有找到所查找数据,则以相同方法搜索下一个区块.B-M 树查找算法见算法 3.例如在图 7 的 B-M 树中,如果查找记录发起者地址为 8 的数据,首先从 B-M 树根节点处判断其叶子节点包括记录的发起者地址范围为 4~40,地址 8 在其范围内,对这个 B-M 树进行查找操作.在 B-M 树内查找时,首先访问根节点保存的 $K$ 值为 20,大于 8,所以向这个 B-M 树的左叶子节点进行搜索.当访问到 $K$ 值为 10 的节点,地址值仍然大于 8,所以继续范围左叶子节点,到 $K$ 值为 7 的节点.因为该节点 $K$ 值小于所搜索地址,因此 B-M 树继续查找该节点的右叶子节点,并访问到了记录发起者地址为 8 的数据.最后,将查找结果返回给查找节点.当一个节点发起交易验证时,验证过程同目前区块链系统相似.当一个节点验证区块链中是否包含某笔交易时,节点利用 Merkle 树特点,只需要一个交易哈希,一个 Merkle 树根哈希和一个 Merkle 路径,在不下载整个块的情况下进行验证.

**算法 3**. B-M 树查找算法.

输入:基于 B-M 树的区块链数据,查询条件:$O_i$;

输出:查询结果,交易集合 $T$.

```
1.  if (发起查询节点不是完全节点)
2.      {在网络中找到一个可靠完全节点;}
3.  for (访问从新到旧每个区块)
4.      {
5.          if (访问区块不是第 1 个区块)
6.              {
7.              访问区块头中 B-M 树根中 max 值、min 值和 K 值;  //(min≤K≤max)
```

```
8.          if (min≤Oi≤max)
9.            {
10.             if (Oi=K)
11.               {返回 K 值地址所生成的数据;}
12.             else if (此节点还有叶子节点)
13.               {
14.                 if (Oi<K)
15.                   {max,min,K=左叶子节点 max 值、min 值和 K 值;
16.                   跳转到步骤 10;}
17.               else
18.                 {max,min,K=右叶子节点 max 值、min 值和 K 值;
19.                 跳转到步骤 10;}
20.           }
21.         }
22.       访问下一个区块;
23.     }
24.   }
```

基于 B-M 树存储结构的区块链模型,既保证了数据在区块中可快速验证,又充分利用二叉树特点保证了查询的高效性.基于 B-M 树模型与使用 B+树建立区块内数据索引的方法相比,省略了根据索引再查找数据的过程,因此具有更快的查询速度.

### 4.5 理论分析与讨论

本节总结 ElasticQM 查询模型中查询层所提出的容量可扩展区块链模型的全局查询优化算法、数据层所提出的 B-M 树建立算法和 B-M 树查找算法的时间代价和空间代价.对于区块链容量可扩展模型 ElasticChain,进行查询操作时,最坏的情况是不能根据 P 链中信息找到存储分片数据的节点(节点故障),这样,模型需要采用泛洪搜索算法,在区块链网络中进行搜索,这样,搜索的时间复杂度为 $O(n_p)$,$n_p$ 为系统中的节点数.而在 ElasticQM 容量可扩展区块链模型的全局查询优化算法中,查询叶子节点通过向查询超级节点请求,再由超级节点间进行查询,这样,在查询层 ElasticQM 模型的查询时间复杂度为 $O(n_{sp})$,$n_{sp}$ 为系统中的查询超级节点数.在 ElasticQM 数据层,基于 B-M 树的查找算法时间复杂度接近于平衡二叉树为 $O(\log n_t)$,$n_t$ 为一个区块中的交易总数.而在 ElasticChain 模型中,在区块内进行查询需要遍历区块完整数据,查询时间复杂度为 $O(n_t)$.因此,ElasticQM 查询模型整体查询时间复杂度为 $O(n_{sp}\times\log n_t)$,ElasticChain 模型整体查询时间复杂度为 $O(n_p\times n_t)$.在空间代价方面,ElasticQM 模型在查询层的全局优化算法与 ElasticChain 模型相同,都是基于副本分片策略进行存储.而在 ElasticQM 模型数据层,与区块链和 ElasticChain 模型相比增加了指针和叶子节点范围等数据,但在一个区块中.两个模型空间复杂度同为 $O(n_t)$,$n_t$ 为一个区块中的交易总数.

## 5 实验与分析

### 5.1 实验环境

实验的开发环境为 Intel Core i5-6500 3.20GHz CPU 和 16GB 内存的 PC 机上.利用 VMware Workstation 12.5.2 建立了 4,8,12 和 16 个节点.每个节点为内存 1GB 硬盘大小为 60GB 的 ubuntu16.04 系统.借助 IBM 开发的开源的 Hyperledge fabric v0.6 版本,构建起 ElasticChain 区块链的容量可扩展模型.

在实验中,我们在 ElasticQM 模型、基于 ElasticChain 容量可扩展区块链模型和基于 fabric 的区块链系统上分别进行查询操作.实验循环运行调用 chaincode_example02.go 交易代码,每完成一次交易,会生成大小为

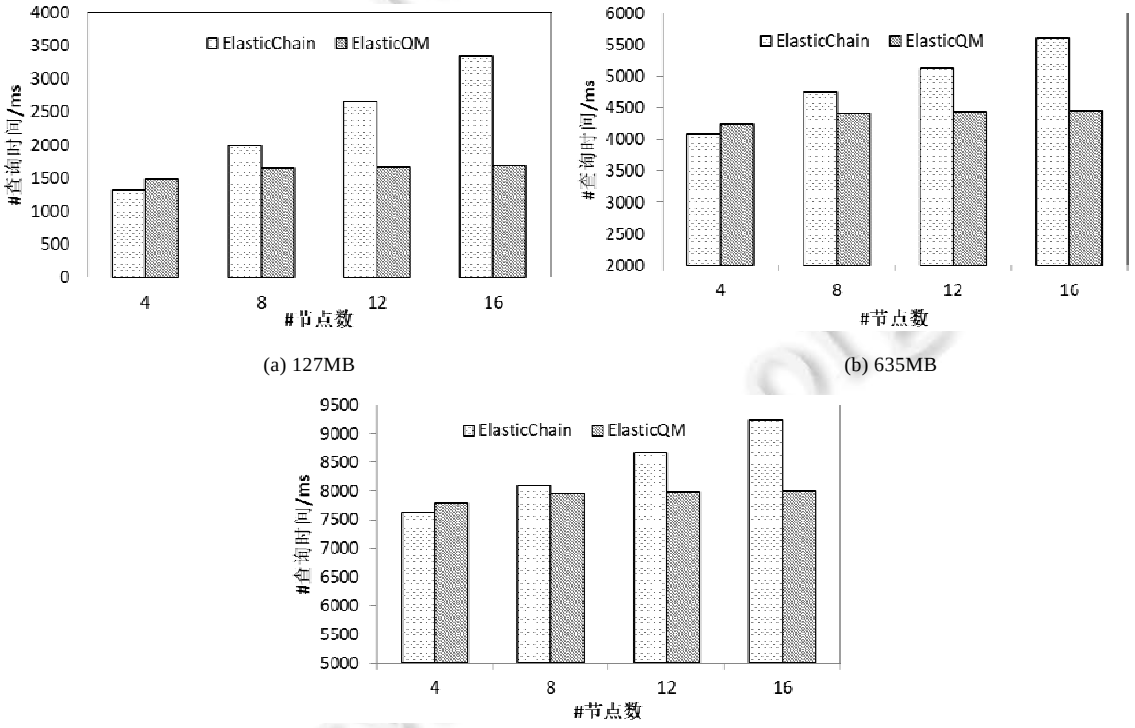
5.39KB 的广播消息和具有唯一标识的哈希值.在查询一条交易或验证交易安全性时,都可以根据生成的唯一的哈希值进行查询验证操作.当 3 个模型分别产生 127MB,635MB 和 1270MB 数据时,停止调用交易代码.

在基于 ElasticChain 区块链模型中,我们将数据进行基于副本分配策略分片处理,在分片时,我们设置了每 64MB 数据作为一个分片,每个分片保存的副本数根据分配策略计算得出,每个分片的最小副本数为 2 份.并且在 ElasticQM 的存储层,也采用相同的副本分配方法.同时,实验分析了 3 种区块链模型在不同节点数下(当在网络中共有 4 个节点、8 个节点、12 个节点和 16 个节点时)查询速度的快慢.当网络中节点数为 4,8,12 和 16 时,在 ElasticQM 查询层实验设置的查询超级节点数分别为 1 个~4 个,其余节点为查询叶子节点,每个查询超级节点分别链接着 3 个不重复的叶子节点,并且实验在查询叶子节点中随机选取了 2 个作为查询验证节点.

### 5.2 评估与结果分析

首先,实验分析在可扩展模型上,基于 ElasticQM 查询方法的查询效率.

实验的查询目标是在容量大小为 127MB,635MB 和 1270MB 的区块链数据中,随机抽取 90 条交易记录和 10 条恶意编造的交易记录;然后,在基于 ElasticQM 和基于 ElasticChain 区块链模型中,对这 100 条记录进行查找操作.两种模型在 4 个、8 个、12 个和 16 个节点时在总容量为 127MB 区块链数据中,平均查找一条交易记录所需时间如图 8(a)所示.基于 ElasticQM 和基于 ElasticChain 模型在总容量为 635MB 和 1270MB 区块链数据中,平均查找一条交易记录所需时间如图 8(b)和图 8(c)所示.




(a) 127MB

(b) 635MB

(c) 1270MB

Fig.8 Queries speed for ElasticQM and ElasticChain model

图 8 ElasticQM 和 ElasticChain 模型的查询速度

通过分析图 8 的实验结果, 可以得到以下结论.

(1) 在相同的区块链数据中进行查询操作时,ElasticQM 模型与 ElasticChain 模型相比,当网络中节点数较少时,ElasticChain 模型的平均查询一条记录的速度比 ElasticQM 模型略快;而当网络中节点数较增加时,ElasticChain 模型的平均查询一条记录的所需时间明显增加,而 ElasticQM 模型所用的查询时间增

量较小;并且当节点数量越多时,ElasticQM 的查询速度优于 ElasticChain 模型越明显.这是由于 ElasticChain 模型将区块链数据分片存储后,数据查询算法会首先遍历 P 链数据,而当 P 链中存储的节点存在故障,模型就会采用基于 P2P 网路中的泛洪算法,因此查询效率较低.而 ElasticQM 模型在查询层采用了基于超级节点的查询算法,网络中当节点数增加后,ElasticQM 在查询时仍然可以通过超级节点中保存的路由信息快速找到相应的数据;

(2) 通过对图 8(a)~图 8(c)进行对比可以看出:如果网络中节点数相同,当区块链数据增加后,ElasticQM 模型和 ElasticChain 模型在区块链上数据查询时间随着查找范围的扩大成比例地增加.但由于两种模型采用副本分配策略将数据分片处理后再存储,查询时间开销主要包括了在区块链中查找时间和访问存储节点和验证节点的时间.随着区块链数据增加,ElasticQM 模型和 ElasticChain 模型节点间的通讯次数增加缓慢.因此,两个模型与基于 fabric 的区块链模型相比,查询时间增速较慢;

(3) 随着区块链数据的增加,网络中相同节点数时,ElasticQM 模型在区块链数据上进行查找的时间增长的速率比 ElasticChain 模型的查找时间增长较慢.因为在 ElasticQM 模型的每个区块中,数据以 B-M 树的结构进行存储,其在块内的查询效率接近平和二叉树查找效率.

然后,实验分析 ElasticQM 模型所占用存储空间的大小.

在实验中,在 ElasticQM 模型、基于 ElasticChain 容量可扩展区块链模型和基于 fabric 的区块链系统上分别存储比特币钱包中前一个、五个和十个区块数据(数据大小分别为 127MB,635MB 和 1270MB).ElasticQM 模型的存储层和 ElasticChain 模型的副本分片策略与查询实验相同.当网络中节点数为 4,8,12 和 16 时(ElasticQM 查询层查询超级节点、查询叶子节点和查询验证节点的设置也与上述查询实验相同),ElasticQM 模型、ElasticChain 模型和基于 fabric 的区块链系统中所有节点存储数据的总量如图 9 所示.

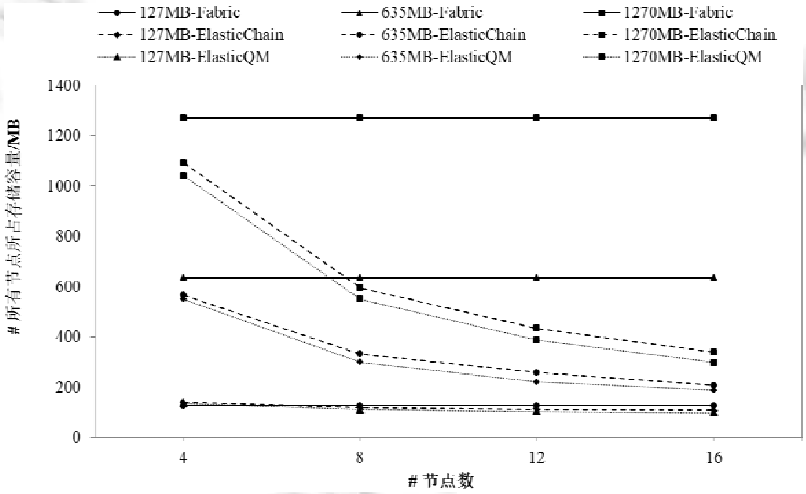



Fig.9 Storage space occupied by ElasticQM, ElasticChain and fabric-based blockchain model

图 9 ElasticQM,ElasticChain 和基于 fabric 区块链模型占用的存储空间总量

通过分析图 9 的实验结果,可以得到以下结论.

(1) ElasticQM 和 ElasticChain 模型的所有节点储存总量与 fabric 区块链相比,在节点数较少的情况下相差不大;但当节点数增多时,ElasticQM 和 ElasticChain 模型所占用的存储空间与 fabric 区块链系统相比明显减少.这是由于在实验中 fabric 区块链系统节点都是完全节点,随节点数的增加,系统存储开销也正比例地增加;而 ElasticQM 和 ElasticChain 模型采用了副本分片策略,在保证安全的情况下将数据进行了分片处理并保存,实现了区块链数据的存储容量的可扩展性;

(2) 当网络中的区块链数据量较小时,ElasticQM 和 ElasticChain 模型储存总量与 fabric 区块链相比相差不

大.这是由于 ElasticChain 模型在存储交易数据的同时,在 P 链中保存了存储节点位置信息,在 POR 链中保存了储存节点的可靠性评价信息.而 ElasticQM 模型在查询层查询超级节点中保存了超级节点间的路由信息,查询验证节点存储了各个节点的查询稳定性,在数据层每个区块中数据以 B-M 树的结构进行存储,B-M 树中记录着发起者地址的最大值和最小值、该位置在平衡二叉树映射节点地址和指向叶子节点的左指针与右指针,并且在用户层增加了数据缓存模块.这些 ElasticQM 和 ElasticChain 模型比目前区块链系统增加的数据占据着一定比例的存储空间.但随着区块链中区块的不断增加,基于实验中的副本分配策略就会大量减少 ElasticQM 和 ElasticChain 模型中的存储总量.

(3) ElasticQM 模型占用的存储空间略少于 ElasticChain 模型,但两个模型占用的存储空间的差距不大;并且当网络中的区块链数据不断增加时,ElasticQM 和 ElasticChain 模型储存总量的增量趋于平缓.

## 6 总结与展望

区块链技术是目前计算机领域的研究热点,随着其广泛应用,对于区块链存储容量的可扩展有着越来越高的要求.基于 ElasticChain 区块链存储容量可扩展模型,将一条完整的区块链副本进行分片处理,并将分片数据保存在一定比例的节点中,提升了区块链的存储容量.本文提出一种区块链容量可扩展模型的高效查询方法——ElasticQM,将数据保存在用户层、查询层、存储层和数据层模块中:在用户层,模型将查询结果缓存,加快再次查询相同数据时的查询速度;模型在查询层采用容量可扩展区块链模型的全局查询优化算法,增加了查询超级节点、查询验证节点和查询叶子节点这 3 种模型中的角色,提高了查询效率;在存储层,模型基于 ElasticChain 区块链存储方法实现了区块链的容量可扩展存储;在数据层,ElasticQM 采用基于 B-M 树的区块链存储结构,并给出了 B-M 树的建立算法和基于 B-M 树的查找算法.通过在多节点不同数据量的区块链中查询的实验表明, ElasticQM 所有节点占用存储空间略小于 ElasticChain 模型,但明显小于目前区块链模型,并且 ElasticQM 的查询效率明显优于 ElasticChain 模型.

在未来区块链技术的应用中,多链共存将是一个普遍现象.为了解决不同体系的区块链中的代币互转的问题,就产生了对跨链操作的需求.目前,主流的跨链技术包括公证人机制、侧链技术和中继技术等.但在本文的模型中,尚未对可跨链的区块链模型的高效查询方法进行研究.因此,在可以实现跨链操作的区块链系统中进行高效的数据查询,将是未来区块链技术的重要研究方向之一.

**References**:

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2009. https://bitcoin.org/bitcoin.pdf

[2] He P, Yu G, Zhang YF, Bao YB. Survey on blockchain technology and its application prospect. Computer Science, 2017,44(4):1−7 (in Chinese with English abstract).

[3] Swan M. Blockchain: Blueprint for a New Economy. USA: O'Reilly Media Inc., 2015.

[4] Yuan Y, Wang FY. Blockchain: The state of the art and future trends. Acta Automatic Sinica, 2016,42(4):481−494 (in Chinese with English abstract).

[5] Blockcypher, recent blocks. 2018. https://live.blockcypher.com/btc/

[6] Blockmeta, the blockchain data of Bitcion. 2017. https://blockmeta.com//btc-stat

[7] World economic forum survey. 2016. http://www.coinfox.info/news/3184-world-economic-forum-survey-10-of-global-gdp-may-be-stored-with-blockchain-technology-by-2027

[8] Gervais A, Karame GO, Glykantzis V, Ritzdorf H, Capkun S. On the security and performance of proof of work blockchains. In: Proc. of the ACM Conf. on Computer and Communications Security. 2016. 3−16.

[9] Spasovski J, Eklund P. Proof of stake blockchain: Performance and scalability for groupware communications. In: Proc. of the MEDES. 2017. 251−258.

[10] Chen ZX, Zhu YX. Personal archive service system using blockchain technology: Case study, promising and challenging. In: Proc. of the IEEE Int'l Conf. on Ai & Mobile Services. 2017. 93−99.

[11] Jia DY, Xin JC, Wang ZQ, Guo W, Wang GR. ElasticChain: Support very large blockchain by reducing data redundancy. In: Proc. of the APWeb-WAIM 2018. 2018.

[12] Zhang ZL, Chong EK, Pezeshki A, Moran W, Howard SD. Near-Optimal distributed detection in balanced binary relay trees. IEEE Trans. on Control of Network Systems, 2017,4(4):826−837.

[13] Xu J, Wei LW, Zhang Y, Wang A, Zhou FC, Gao CZ. Dynamic fully homomorphic encryption-based Merkle tree for lightweight streaming authenticated data structures. Journal of Network and Computer Applications, 2018,107:113−124.

[14] Li Y, Zheng K, YanY, Liu Q, Zhou XF. EtherQL: A query layer for blockchain system. In: Proc. of the DASFAA (2). 2017. 556−567.

[15] Dinh TTA, Wang J, Chen G, Liu R, Ooi BC, Yan KL. BLOCKBENCH: A framework for analyzing private blockchains. In: Proc. of the SIGMOD Conf. 2017. 1085−1100.

[16] Vo HT, Kundu A, Mohania M. Research directions in blockchain data management and analytics. In: Proc. of the EDBT. 2018. 445−448.

[17] David B, Gazi P, Kiayias A, Russell A. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In: Proc. of the Eurocrypt 2018. 2018. 66−98.

[18] Wang S, Dinh TTA, Lin Q, Xie ZL, Zhang MH, Cai QC, Chen G, Fu WZ, Ooi BC, Ruan PC. ForkBase: An efficient storage engine for blockchain and forkable applications. CoRR abs/1802.04949, 2018.

[19] Ali M, Nelson J, Shea R, Freedman MJ. Blockstack: A global naming and storage system secured by blockchains. In: Proc. of the USENIX Annual Technical Conf. 2016. 181−194.

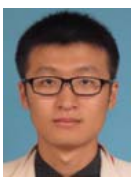
[20] Gervais A, Capkun S, Karame GO, Gruber D. On the privacy provisions of Bloom filters in lightweight bitcoin clients. In: Proc. of the ACSAC. 2014. 326−335.

[21] Gervais A, Karame GO, Glykantzis V, Ritzdorf H, Capkun S. On the security and performance of proof of work blockchains. In: Proc. of the ACM Conf. on Computer and Communications Security. 2016. 3−16.

**附中文参考文献:**

[2] 袁勇,王飞跃.区块链技术发展现状与展望.自动化学报,2016,42(4):481−494.

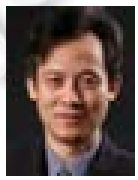
[4] 何蒲,于戈,张岩峰,鲍玉斌.区块链技术与应用前瞻综述.计算机科学,2017,44(4):1−7.



**贾大宇**(1990－),男,辽宁沈阳人,博士生,主要研究领域为区块链存储与查询,大数据管理与分析.

**郭薇**(1983－),女,博士,副教授,主要研究领域为数字图像处理,模式识别,人工智能.

**信俊昌**(1977－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为大数据管理与分析,感知数据管理,计算机辅助诊断,医学信息学.



**王国仁**(1966－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库,大数据管理与分析,生物信息学.

**王之琼**(1980－),女,博士,副教授,CCF 专业会员,主要研究领域为计算机辅助诊断,医学信息学,健康大数据分析技术.