DOI:10.13718/j.cnki.xsxb.2022.02.017

基于资源路由环的对等云区间搜索技术①

贺道德, 胡如会

贵州工程应用技术学院 信息工程学院, 贵州 毕节 551700

摘要: 运用结构化的对等技术来构建云系统, 针对传统 P2P 技术仅适合精确搜索的问题, 以及用户的隐私保护需求, 文章提出一种在密文状态下实现的对等云区间搜索技术. 在 Pastry 的基础上架构对等云, 在资源发布前, 运用同态加密技术对资源属性值进行加密; 在资源发布时, 将存储相同类型资源的节点链接起来形成资源路由环; 在资源搜索时, 采用密文结合路由环的方式进行区间搜索. 本文提出的算法弥补了结构化的对等云在区间搜索方面的不足, 且密文搜索实现了对用户隐私的保护.

关 键 词: 同态加密; 分布式哈希表; 对等云; 区间搜索

中图分类号: TP393 **文献标志码**: A **文章编号**: 1000-5471(2022)02-0109-09

A P2P Cloud Searching in a District Technology Based on Resource Routing Ring

HE Daode, HU Ruhui

School of Information Engineering, Guizhou University of Engineering Science, Bijie Guizhou 551700, China

Abstract: In this paper, structured P2P technology has been used to build cloud system. Aiming at the problem that traditional P2P technology is only suitable for precise search and the demand of user privacy protection, a P2P cloud interval searching technology in ciphertext state has been proposed. The P2P cloud is built on the basis of pastry. Before the resources are released, homomorphic encryption technology has been used to encrypt the resource attribute value. When the resources are published, the nodes that store the same type of resource are linked together to form a resource routing ring. In the process of resource searching, ciphertext and routing ring are used for interval searching. The algorithm proposed in this paper makes up for the shortcomings of that structured P2P cloud couldn't do interval searching, and ciphertext searching realizes the protection of user privacy.

Key words: homomorphic encryption; distributed hash table; P2P cloud; searching in a district

云计算是基于 Internet 来共享软硬件资源与数据的一种计算方式[1-2], 它运用虚拟的方式来整合资源, 以实现用户便捷地使用共享资源. 目前, 云计算中的服务与资源都由服务提供商控制, 具有较好的可靠性与可控性, 但由于云资源由少数服务提供商垄断, 使得云的可扩展性不好, 且成本偏高. 对等计算整合互联网中用户提供的资源来实现资源的共享, 各用户的地位对等, 资源的利用率高, 且网络的可扩展性好[3];

① 收稿日期: 2021-02-24

基金项目: 国家自然科学基金资助项目(61966005); 贵州省教育厅青年科技人才成长项目(黔教合 KY 字[2018]391, 黔教合 KY 字[2018]397); 贵州工程应用技术学院本科教学质量提升工程项目(2019SZ007, JK202023, 2018JG169).

作者简介: 贺道德, 硕士, 副教授, 主要从事对等云计算、无线传感器网络技术等方面的研究.

但由于对等网络中的用户具有会话异构等特征，使得网络稳定性和可控性不够好. 由上述描述可知，对等计算技术和云计算技术相互补；运用对等计算技术架构底层网络，可充分利用资源且可扩展性好；然后利用云计算的虚拟技术以确保服务的可靠性与可用性，从而形成了对等云技术[4-6].

运用结构化的对等计算系统构成的对等云采用分布式哈希表[7](DHT，Distributed Hash Table)来进行资源的发布与定位. 在资源发布时，先将资源关键字运用哈希算法(例如 SHA-1)计算出对象标识 objId，然后将资源发布到与节点标识 nodeId 相近的节点上；在资源搜索时，亦依据哈希值来搜索. 由于哈希算法往往将属性值相近的资源映射成完全不相关的 objId，然后将其发布到不相关的节点上，从而使其难以支持资源关键字区间搜索. 为实现在结构化对等云系统中进行区间搜索，本文提出如下思想：在资源发布时，运用同态加密[8-10]具有在密文状态下可进行操作的特征，首先将属性值 VALUE 使用具有同态特性的加密算法计算出 FHVALUE，并运用此值计算出一个资源标记，拥有相似属性值的资源具有相同的资源标记；然后，再哈希 FHVALUE 得到 objId，并将资源发布到对应 nodeId 的节点上；节点除存储资源外，还需依据资源标记将相同标记的资源节点链接成资源路由环. 在区间搜索时，运用同态加密过的属性值计算出资源标记，从而进行区间搜索. 为实现上述思想，本文采用典型的结构化对等系统 Pastry[11]来构建对等云，运用同态加密算法 GSW[12]来计算资源标记从而形成资源路由环.

1 相关研究

1.1 Pastry

莱斯大学与微软研究院共同提出的 Pastry 是采用 DHT 构造的结构化对等系统. 该系统中的每个节点都拥有一个 128 b 的 nodeId，且为自组织重叠网络. 节点的 nodeId 在节点空间中$(0\sim 2^{128}-1)$标识节点的位置，由于散列值的随机性，使得节点 nodeId 在节点空间中能均匀分布. 当需要发布资源时，对资源属性值运用散列算法计算出资源 objId，然后运用 Pastry 的路由算法将资源发布到节点 nodeId 与资源 objId 在数值上最接近的那个节点；在资源定位时，按此路由算法查找资源. 由于 Pastry 具有完全分布式特性，且自组织、扩展性好，因此，用其作为云系统的底层架构，可克服普通云系统扩展性不好、成本高的不足.

在 Pastry 系统中，每个节点维护 3 个状态表，包括一个路由表(R，Routing Table)，一个邻居节点集(N，Neighborhood Set)和一个叶子节点集(L，Leaf Set). 系统运用上述状态表来维护网络的拓扑信息，文献[11]列举了节点标识为 10233102 的状态表，如图 1 所示.

nodeId 10233102

Leaf Set

SMALLER		LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing Table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Neighborhood Set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

图 1 节点标识为 10233102 的状态表

图 1 中描述的节点标识为 10233102，标识的构成采用 2^b 进制，b 取值为 2. 路由表 R 中的每行包含 2^b-1 个表项，R 中的第 n 行的节点标识和当前节点标识的前 n 位相同（n 从 0 开始）. 叶子节点集 L 存储的节点标识与当前节点标识值相近，其中包含两部分，前半部分的值略大于当前节点标识，后半部分的值则略小于当前节点标识. 邻居节点集 N 存放与当前节点物理位置相近的节点的 nodeId，它主要用于维护路由的本地性[13]，在正常路由过程中并不被使用.

在 Pastry 系统中，若当前节点 V 收到一条路由信息，则首先从叶子节点集 L 中查找与路由信息中目标节点 D 的节点标识更接近的 nodeId，若查找到，则路由到此节点. 第二步，在叶子节点中查找失败的情况下，转去查路由表 R，计算出目标节点 D 与当前节点 V 的相同前缀长度 j. 第三步，如果 R 中第 j 行的第 D_j 表项（D_j 为目标节点标识中的第 j 个数值）不为空，则路由到此节点去. 第四步，若查找路由表失败，则从 3 个状态集中找一个和目标节点 D 标识最接近的节点，并路由到此节点. Pastry 的路由算法（PR，Pastry Routing）如下所示.

$R_{(i,j)}$：表示路由表 R 中第 j 行第 i 项

L_i：表示叶子节点集 L 中第 i 项存储的节点标识（若为负数，则表示该标识小于当前节点标识）

D_j：目标节点 D 的第 j 个数值

shl(D，V)：节点 D 与节点 V 具有相同标识的前缀长度

Function PR(nodeId D)

1) {/ * 该算法为对等云覆盖网络 Pastry 的主路由算法 * /

2) if($L_{-|L|/2}$<=D<=$L_{|L|/2}$){/ * 如果目标节点 D 在叶子节点集中 * /

3) 路由到节点 L_i，其中|D-L_i|为最小；}

4) else{/ * 叶子节点集查找失败，则转查路由表 R * /

5) j=shl(D，V)；/ * 计算目标节点 D 与当前节点 V 的相同标识前缀长度 * /

6) if(R(D_j，j)!=NULL){/ * 若路由表 R 的第 j 行的第 D_j 项不为空 * /

7) 路由到 R(D_j，j)所存储的节点标识对应的节点；}

8) else{/ * 路由表查找失败 * /

9) 路由到节点 T，T∈L∪R∪M，shl(T，D)>=j，|T−D|<|V−D|；}}

10) / * 算法结束 * /}

1.2　GSW

GSW 是文献[12]中提出的一种基于容错学习（Learining With Error，LWE）的同态加密方案. GSW 是运用矩阵与近似特征向量构造出的基于身份的全同态系统，它与传统同态加密方案不同的是该方案无需同态操作密钥亦可实现同态加密. 其基本方案是一个基于公钥的密码体系，其组成包括 Setup，SecretKeyGen，PublicKeyGen，Enc，Dec 和 MPDec 这 6 部分.

1) Setup(1^λ，1^L)，这是一个初始化操作. 选择一个 k 位的模数 q，其中，$k=k(\lambda, L)$，λ 为安全参数，L 为方案的层数；格的维度值 $n=n(\lambda, L)$；误差分布 $\chi=\chi(\lambda, L)$，以确保容错学习方案的安全强度在攻击情况已知条件下达到 2^λ. 再次，选取参数 $m=m(\lambda, L)=O(n \log q)$，令 $params=(n, q, \chi, m)$，$\ell=\lfloor \log q \rfloor+1$，$N=(n+1)\cdot\ell$.

2) SecretKeyGen($params$)，该部分用于生成私钥，其输入为参数 $params$，样本 $\boldsymbol{t}$ 向量随机均匀分布在 n 维的 $\mathbb{Z}_q$ 上，输出的私钥 sk 为向量 $\boldsymbol{s}=(1, -\boldsymbol{t}_1, \cdots, -\boldsymbol{t}_n)\in\mathbb{Z}_q$（$n+1$ 维），并令向量 $\boldsymbol{v}=Powersof2(\boldsymbol{s})$，$Powersof2$ 函数的输入为私钥向量 $\boldsymbol{s}$，输出向量 $\boldsymbol{v}$ 用于相关同态计算.

3) PublicKeyGen($params$，sk)，该部分用于生成公钥，其输入为参数 $params$ 与私钥 sk，生成一个 $m*n$ 矩阵 $\boldsymbol{B}$，且随机均匀分布在 $\mathbb{Z}_q$ 上，并依据误差分布 χ 选取 m 维误差向量 $\boldsymbol{e}\leftarrow\chi^m$，计算出 $b=\boldsymbol{B}\cdot t+\boldsymbol{e}$，然后输出公钥 $pk=\boldsymbol{A}=[b|\boldsymbol{B}]$（该 $\boldsymbol{A}$ 是在 $\mathbb{Z}_q$ 上的 $m*(n+1)$ 维矩阵）. 由于矩阵 $\boldsymbol{A}$ 与私钥向量 $\boldsymbol{s}$ 的乘积为误差向量 $\boldsymbol{e}$，从而确保了密钥的安全性.

4) Enc($params$，pk，μ)，该部分为加密函数，μ 为明文，其在空间 $\mathbb{Z}_q$ 之内，pk 为公钥，$params$ 为参数，其输出为密文矩阵 $\boldsymbol{C}$.

5) Dec($params$, sk, $\boldsymbol{C}$)，该部分为解密函数，$\boldsymbol{C}$ 为密文，sk 为私钥，$params$ 为参数，它可以在足够小的空间内恢复出明文 μ.

6) MPDec($params$, sk, $\boldsymbol{C}$)，该解密函数由 Micciancio 和 Peikert 在文献[14]中提出，它可以恢复出明文 μ 二进制表示的全部有效位.

此外，GSW 提供了一系列同态操作函数，包括同态数乘 MultConst、同态加法 Add、同态乘法 Mult 以及同态 NAND 门操作等.

2 基于资源路由环的对等云区间搜索拓扑模型

2.1 网络架构基础

为了在结构化的对等云中进行区间搜索，本文以如下网络架构为基础：

1) 为了克服传统云存储系统因中心化而存在对云服务提供者信任依赖等问题，本文所提网络架构中的节点地位对等.

2) 网络中的节点为稳定节点，以适应云存储的需求；为了描述区间搜索技术，本文对节点失效、会话异构等问题不做讨论.

3) 本文以 Pastry 系统为基础构建网络，运用分布式哈希表来发布资源与定位，本文运用的哈希函数具有抗原像性、抗第二原像性以及强抗碰撞性等特征.

4) 因考虑到目前流行的全同态算法存在加密速度慢，以及受搜索算法的搜索效率影响等问题，本文所提算法仅对资源属性值进行同态加密.

2.2 相关定义

为准确描述区间搜索模型及技术，本文对所涉及的相关定义描述如下：

定义 1：节点标识，用以标识对等云网络中不同的节点，记为 nodeId.

定义 2：资源属性值，能够代表某资源相关特征的值，主要包括资源关键字、资源名以及所有者身份标识等，记为 VALUE；同态加密后的资源属性值记为 HFVALUE.

定义 3：对象标识，用以唯一标识对等云网络中存储的资源对象，记为 objId.

定义 4：资源标记，又称为资源类型，具有相同类型的资源拥有相同的资源标记，记为 TYPE.

定义 5：路由环节点信息表(RRNT，Routing Ring Node Table)，用于构建资源路由环的数据结构，其中包括下一个存储同类型资源节点的节点标识、对象标识，具体定义如表 1 所示.

表 1 路由环节点信息表

名称	定　义
nodeId	同资源类型的下一个路由节点的标识
objId	同资源类型的下一个路由节点存储资源的对象标识

定义 6：节点的资源信息表(SNT，Source Node Table)，用于对等云节点存储资源相关信息的数据结构，其中包括资源对象标识、资源密文属性值、资源标记，具体定义如表 2 所示.

表 2 节点的资源信息表

名称	定　义
objId	唯一标识资源的对象标识
HFVALUE	同态加密后的资源密文属性值
TYPE	同类型资源的标记

2.3 网络拓扑模型

在结构化的对等系统中，由于采用散列函数的散列属性值来生成对象 Id 或节点 Id，因此 Id 间没有关联性，仅适用于精确搜索.

为实现区间搜索，本对等云系统在 Pastry 系统的基础上，首先采用同态加密算法 GSW 对属性值进行加密，并且以密文的形式计算出资源标记；然后将相同资源标记的节点链在一起形成路由环以便区间搜索. 具体举例如下：现有相似资源(A,B,C,D)，为保证其机密性，运用 GSW 算法计算出密文属性(A',B',C',D')，再通过哈希函数计算得到其对象 objId 为(126,359,87,98)，并确定其资源标记为 S；然后，将这

些资源及其资源标记 S 发布到节点 nodeId 为(127，400，87，100)的节点上，并将这些节点构造成一个路由环，以便实现区间搜索，具体如图 2 所示.

图 2 给出了基于资源路由环的对等云区间搜索拓扑图，从图中可以看出，相似属性的资源被发布到 nodeId 没有关联性的节点上，因此，不适合进行区间搜索. 为实现区间搜索，将存储相似资源的节点存储一个相同的资源标记 S，然后通过链接形成一个路由环，本模型的具体构建方法如下所示.

1) 运用 Pastry 系统的网络架构方案来构建对等网络.

2) 结合同态加密算法计算出基于密文的资源标记.

3) 在资源发布时，依据资源标记，将资源标记相同的同类型资源采用链式环的方法链接在一起，形成资源路由环.

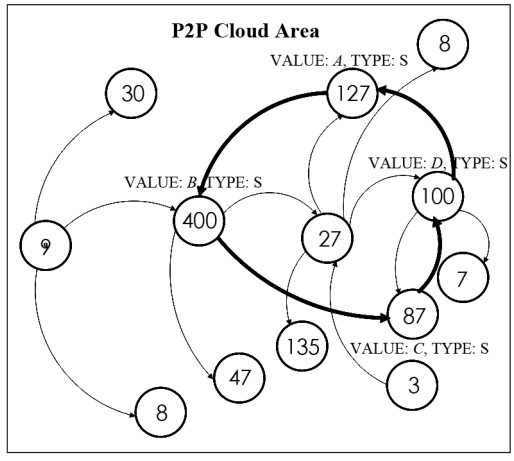

图 2　基于资源路由环的对等云区间搜索拓扑图

3　基于资源路由环的对等云资源发布算法

为实现在对等云中进行区间搜索，资源应依据基于资源路由环的对等云区间搜索拓扑模型来进行发布. 第一步，首先依据资源的属性值 VALUE 按同态加密算法 GSW 计算出密文属性 FHVALUE，然后对密文运用同态算法计算出资源标记，同种类型的资源拥有相同的资源标记 S. 这样处理既确保属性值的机密性，又便于链接相同类型的资源形成资源路由环. 第二步，将资源密文属性 FHVALUE 按 SHA-1 算法哈希出对象标识 objId；然后运用 Pastry 的路由算法 PR 发布路由消息，查找到与 objId 值相近的 nodeId 的网络节点 N，然后将资源、资源标记以及资源其他信息发布到该节点上. 第三步，依据资源标记值，在对等云系统中查找一个拥有相同资源标记值的节点 K；如果找到这样的节点 K，则把 K 的路由环节点信息表(RRNT，Routing Ring Node Table)中记载的节点 M 的路由信息发送给节点 N；节点 N 据此信息生成自己的 RRNT，并将自己的路由信息发送给节点 K，节点 K 依此信息更新 RRNT 表，从而实现节点 N 加入资源路由环. 如果在查找时，没有找到拥有相同资源标记的节点，说明该类型资源是第一次加入系统，则将自己的路由信息放入 RRNT 中. 基于资源路由环的对等云资源发布算法(RPARRR，Resource Publishing Algorithm based on Resource Routing Ring)如下所示.

Function RPARRR (VALUE V)

1) {/* 该算法用于对等云系统资源发布，输入为资源属性值 V */

2) /* 运用同态加密算法 GSW 计算出密文属性值 FHV */

3) FHV=GSW(V)；

4) /* 调用密文计算函数 getSouTy 计算资源标记 S，并计算该类型数据区间最小值 MIN 与最大值 MAX */

5) S=getSouTy(FHV)；

6) /* 运用 SHA-1 算法计算出对象标识 objId */

7) objId= SHA-1(FHV)；

8) /* 调用 Pastry 的路由算法 PR，查找到资源存储节点 N */

9) N=PR(objId)；

10) 将资源发布到节点 N 中；

11) /* 调用资源定位算法 RLART，查找具有相似标记 S 的资源节点 */

12) /* MIN 至 MAX 为资源的属性区间 */

13) K=RLART(S，MIN，MAX)；

```
14) if(K!=NULL)
15) {/* 如果查找到的节点 K 不为空 */
16) 将 K 节点的路由环节点信息表 RRNT 中记载的路由信息发送给节点 N;
17) 节点 N 据此信息生成自己的 RRNT;
18) 将节点 N 的路由信息发送给节点 K,节点 K 依据此信息更新其 RRNT 表;}
19) else{/* 如果没有找到这样的节点,表示该节点是第一个节点 */
20) 将节点 N 自己的路由信息存入 RRNT;}
21) /*算法结束*/    }
```

4 基于资源路由环的对等云区间搜索技术

在基于资源路由环的对等云资源发布算法中,我们可以在不改变原有对等云结构的基础上,将存储相同类型资源的节点链接成一个资源路由环.完成资源路由环设计后,下面的任务就是如何在基于资源路由环的对等云系统中进行资源的区间搜索.

4.1 依据资源类型进行资源定位

在资源发布算法中,节点在插入到某资源路由环之前,需按资源类型定位到该资源路由环.因此,在给定一种资源类型后,如何在系统中查找到这种资源是本区间搜索技术的主要算法之一.为实现该算法,本文提出了如下思想:由于本文所提的同类型资源定位算法采用基于密文搜索的机制,因此,若用户已知资源类型为明文 M,则在其明文区间[MMIN, MMAX]中随机选择一个值 V,运用同态加密机制运算出其资源标记 S,及其属性值区间[MIN, MAX].第一步,用户在资源属性值区间中运用随机函数计算得到一个密文属性值 FHV,然后,依据 SHA-1 算法计算出该属性值的对象标识 objId.第二步,运用对等云系统的路由算法 PR 路由到节点 K,查看 K 节点是否存在资源标记为 S 的资源,如果存在,则查找结束并返回成功.第三步,如果没有找到,则重新在属性区间中随机产生一个新的密文属性值 FHV′,重新搜索.第四步,直到搜索到此种类型的资源,返回成功算法结束;或者搜索次数超限返回失败算法结束.依据资源类型进行资源定位的算法(RLART, Resource Location Algorithm based on Resource Type)如下所示.

```
Function RLART(TYPE S, FHVALUE MIN , FHVALUE MAX)
1) {/* 该算法在给定资源类型标记的情况下进行资源定位,算法返回值为资源存储所在节点的标识 */
2) count=0; /* count 变量用来记载搜索次数,最大值为 MaxCount */
3) flag=0; /* flag 变量为是否搜索成功标记,查找成功时,该值为 1 */
4) while(count<=MaxCount)
5) {/* 在 S 类资源的属性值区间内随机产生一个属性值 FHV */
6) FHV=random(S, MIN, MAX);
7) objId=SHA-1(FHV); /* 运用 SHA-1 算法计算出对象标识 objId */
8) /* 调用 Pastry 的路由算法 PR,将资源发布到路由到的节点 K */
9) K=PR(objId);
10) forEach(id in 节点 K 的 SNT)
11) {/* 遍历 K 节点的资源信息表 */
12) if(id= =objId) /* 如果查找的资源对象标识找到 */
13) {flag=1;
14)    break; /* 在查找成功时,中止循环 */}}
15) if(flag= =1) break;
16) count++; /* 计数器自加 */}
17) if(flag= =0){/* 没有查找到对应资源的节点时,返回空值 */
18) return NULL; }
19) else{/* 若找到对应资源的节点时,返回节点的标识 */
```

20) return K. nodeId; }

21)　/ * 算法结束 * /　}

该算法在不改变对等云覆盖网络结构的基础上构建，具有较强的自适应性，既支持密文资源定位，也支持明文资源定位；用户在拥有明文的情况下进行定位时，为确保操作过程的机密性，只需在调用该算法前进行一次同态密码运算即可完成.

4.2　基于资源路由环的区间定位

在以资源路由环的方式发布资源后，在对等云系统中，拥有相同资源的节点通过存储相同资源标记的路由信息后，形成资源路由环；本小节将描述在资源路由环中如何进行区间搜索. 首先，当用户搜索关键字区间为$[V_1, V_2]$时，若关键字为明文，则运用同态加密机制计算出密文区间$[FHV_1, FHV_2]$，并计算出资源标记 S. 第二步，依据资源标记值 S，通过资源定位算法 RLART 搜索到存储该类型资源的节点 N. 第三步，以节点 N 为起始节点，在资源路由环中比对搜索区间关键字；若在此区间，则将存储资源节点的节点标识返回给用户，直至遍历资源路由环结束. 基于资源路由环的区间定位算法(ILARRR，Interval Location Algorithm based on Resource Routing Ring)如下所示.

Function ILARRR(FHVALUE FHV_1, FHVALUE FHV_2)

1) {/ * 该算法实现在对等云系统中进行区间搜索，区间为$[FHV_1, FHV_2]$，FHVALUE 为同态密文属性类型 * /

2) / * 调用密文计算函数 getSouTy 计算出资源标记 S，并计算出该类型数据区间最值 MIN 与 MAX * /

3) S=getSouTy(FHV_1);

4) / * 调用依据资源标记定位资源算法 RLART 查找第一个存储 S 类资源的节点 N * /

5) N=RLART(S, MIN, MAX);

6) if(N!=NULL){/ * 如 N 节点不为空，以 N 为第一个节点遍历资源路由环 * /

7) I=N; / * 用临时变量 I 存储拥有 S 类资源的节点的路由信息 * /

8) 定义集合 SourNode[]存储资源节点的路由信息;

9) j=0;

10) do{

11) forEach(id in 节点 I 的 SNT)

12) {/ * 遍历 I 节点的资源信息表 * /

13) if(id>=FHV_1 &&id<=FHV_2) / * 如果查找的资源属性值在搜索区间之内 * /

14) {SourNode[j]. nodeId =I. nodeId; / * 将 I 节点的路由信息存入资源节点集合 SourNode * /

15) j++;

16) break; }}

17) I=I. RRNT. nodeId; / * 取出 I 节点的路由环节点信息表中的路由节点作为下一查找的节点 * /

18) }while(I. nodeId !=N. nodeId); / * 循环到路由起点结束 * /

19) }else return NULL; / * 搜索失败，返回空值 * /

20) if(j>0){

21) / * 搜索成功，返回资源节点集合 * /

22) return SourNode;

23) }else{/ * 搜索失败，返回空值 * /

24) return NULL; }

25) / * 算法结束 * / }

上述算法的输入为密文属性区间，若用户在已知明文区间的情况下进行区间搜索，则需运用同态加密机制计算出密文区间，然后调用此算法来完成基于资源路由环的区间定位.

4.3　区间搜索效率分析

本文提出的区间搜索算法通过改进 Pastry 对等系统，运用资源路由环实现了密文资源区间的搜索，本搜索

算法的路由开销包括在Pastry网络中的路由开销以及在路由环中的路由开销，现就其搜索效率分析如下：

1)在区间搜索前，需运用Pastry的搜索算法搜索到同类型资源的首个资源节点，即资源路由环的入口；RLART算法需多次调用Pastry的路由算法来定位这个入口. 由于Pastry路由算法的平均路由开销为$\lceil \log_{2^b} N \rceil$，其中$2^b$为标识构成采用的进制，$N$为节点总数[11]，因此，RLART算法的平均路由开销则为$m\lceil \log_{2^b} N \rceil$,其中$m$为完成算法调用Pastry路由算法的平均次数.

2)找到资源路由环入口后，ILARRR算法运用遍历环中资源节点的方法搜索资源，因此，其路由开销与环的大小成正比；在资源规模较小的情况下，环内路由开销远远小于该算法调用RLART算法搜索环入口的路由开销. 但若资源规模增大，环内搜索势必成为主要的搜索开销之一，因此，为提高搜索效率，本文提出如下改进措施：

① 在资源发布时，以资源属性值为关键字来构建有序链表的资源路由环；

② 在资源定位时，运用资源路由环的有序性，优化查找算法，以达到在资源路由环内的搜索效率最优.

5 仿真与性能分析

为了验证运用结构化对等系统Pastry作为对等云覆盖网络的有效性，我们选择FreePastry为原型仿真器[15]来进行仿真测试，网络规模确定其最大值为10 000个网络节点；节点标识的构成采用16进制，N的大小为$|N|=32$，L的大小为$|L|=16$. 为了实现密文下的区间搜索，运用资源发布算法发布资源，设定算法RLART的最大搜索次数为10次，资源类型数量为200种；搜索区间的确定方法为从不同类型资源的属性区间中随机抽取. 实验主要测试了网络规模与路由开销之间的关系，以及路由开销与搜索区间长度之间的关系.

在测试网络规模与路由开销之间的关系时，从200种资源中随机抽取资源，并确定搜索区间的长度为20个节点，网络节点数量从1 000个增加至最大值10 000个，增值长度为500；最终在不同网络规模下进行了100次实验，取其均值，获得的平均路由开销与网络规模的关系如图3所示.

图3为网络节点个数与平均路由跳数之间的关系，其中横坐标为网络节点数(个)，纵坐标为平均路由开销，单位为步跳数(hops). 从图3可以看出，网络规模从1 000增至10 000个节点的过程中，区间搜索的平均路由开销的增长接近线性增长趋势. 得到上述实验结果的原因是：本文所提的对等云系统采用资源路由环进行构造，即将存储同种类型资源的节点运用链接的方式形成一个资源环，在路由查找过程中，主要开销为基本覆盖网络的路由开销，因此，平均路由开销与网络规模成正比关系.

区间搜索相比于精确搜索，其搜索的数据量大；常规情况下，搜索区间包含的节点个数越多，则路由开销也会越大. 因此，搜索区间长度与路由开销间的关系是区间搜索的重要评估指标. 在测试时，我们确定网络规模为2 000个节点，搜索区间长度从10个节点增至100个节点，增值长度为10；从200种资源中随机抽取资源，在不同的搜索区间长度下完成了100次实验，取其平均路由开销. 平均路由开销与搜索区间长度之间的关系如图4所示.

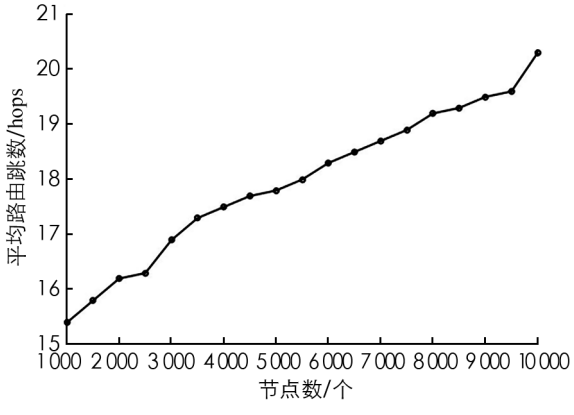

图3 平均路由开销与网络规模关系图

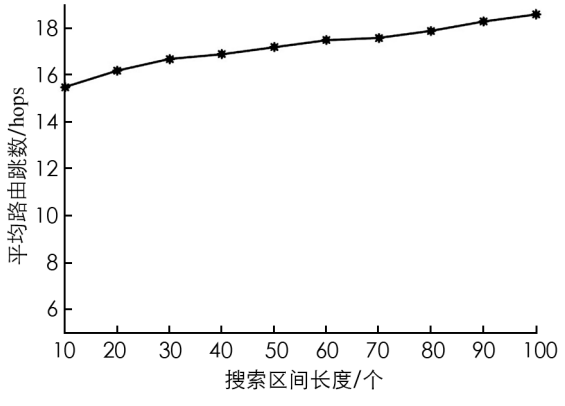

图4 平均路由开销与搜索区间长度关系图

图 4 为搜索区间长度与平均路由跳数之间的关系，横坐标为搜索区间长度，单位为搜索区间内资源数量(个)，纵坐标为平均路由跳数，单位为 hops. 从图中可以看出，随着搜索区间的增大，其平均路由跳数的增长趋于平缓. 出现上述实验结果的原因是：本系统运用资源路由环构造，在路由过程中，主要开销在于查找第一个资源，并且资源路由环中的路由开销不大.

6　结束语

本文运用结构化的对等系统来构建云系统的覆盖网络，采用将存储相同类型资源的节点链接成资源路由环，从而解决结构化的对等云系统不适合区间搜索的问题. 另外，为确保数据的机密性，运用同态加密机制来实现密文下的资源搜索. 同态加密时间复杂性较大，本系统仅对资源属性值进行了同态加密. 下一步的工作将优化同态加密算法，以使其在对等云系统中进行密文运算时不受条件限制.

参考文献：

[1] 蔡昌许. 基于重复异构最早完成时间的云计算任务调度算法 [J]. 西南师范大学学报(自然科学版)，2020，45(5)：141-147.
[2] 郑　瑛. 云计算数据中心节能调度算法改进研究 [J]. 西南大学学报(自然科学版)，2019，41(12)：135-142.
[3] 马满福，何春玲. 面向选择推荐节点的 P2P 网络信任模型 [J]. 计算机工程与科学，2018，40(6)：977-983.
[4] 王文丰，韩龙哲，李沛武，等. 一种基于语义的分布式云服务发现方法 [J]. 中山大学学报(自然科学版)，2019，58(3)：145-152.
[5] 刘成山，张秀君. 基于对等云的数字图书馆动态服务聚合 [J]. 四川大学学报(哲学社会科学版)，2019(2)：112-117.
[6] 王军正. 基于对等结构云存储的副本管理研究 [D]. 成都：电子科技大学，2016.
[7] 苏　扬，张　琦，唐善成. 改进 Kademlia 协议的 P2P 网络资源发现算法 [J]. 西安科技大学学报，2020，40(3)：464-469.
[8] 谭德林，谭　良. 具有全同态属性的密码方案的研究及应用 [J]. 计算机工程与设计，2019，40(5)：1205-1209.
[9] 谭作文，张连福. 机器学习隐私保护研究综述 [J]. 软件学报，2020，31(7)：2127-2156.
[10] 宋秀丽，周道洋，曹耘凡. d 维量子同态加密算法的设计与仿真 [J]. 计算机工程与应用，2020，56(7)：109-115.
[11] ROWSTRON A，DRUSCHEL P. Pastry：Scalable，Decentralized Object Location，and Routing for Large-Scale Peer-to-Peer Systems [C]//IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing. Berlin，Heidelberg：Springer，2001：329-350.
[12] GENTRY C，SAHAI A，WATERS B. Homomorphic Encryption from Learning with Errors：Conceptually-Simpler，Asymptotically-Faster，Attribute-Based [C]//Proceedings of the 33rd Annual Cryptology Conference on Advances in Cryptology. CA，USA：Springer，2013：75-92.
[13] 贺道德，邓晓衡. 基于物理位置与访问局部性的 P2P 路由算法 [J]. 计算机工程，2009，35(8)：146-149.
[14] MICCIANCIO D，PEIKERT C. Trapdoors for Lattices：Simpler，Tighter，Faster，Smaller [C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cambridge，UK：Springer，2012：700-718.
[15] Rice University. FreePastry [EB/OL]. (2009-03-13) [2020-05-01]. https：//freepastry. org/FreePastry.

责任编辑　崔玉洁