ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.6, June 2008, pp.1473−1480
DOI: 10.3724/SP.J.1001.2008.01473

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

一种面向大规模 P2P 系统的快速搜索算法*

张一鸣+, 卢锡城, 郑倩冰, 李东升

(国防科学技术大学 并行与分布处理国家重点实验室,湖南 长沙 410073)

An Efficient Search Algorithm for Large-Scale P2P Systems

ZHANG Yi-Ming+, LU Xi-Cheng, ZHENG Qian-Bing, LI Dong-Sheng

(National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: ymzhang@nudt.edu.cn

Zhang YM, Lu XC, Zheng QB, Li DS. An efficient search algorithm for large-scale P2P systems. *Journal of Software*, 2008,19(6):1473−1480. http://www.jos.org.cn/1000-9825/19/1473.htm

Abstract: This paper presents a search algorithm called probabilistic search team (PST). In PST, all nodes advertise their resource sharing information, maintain and broadcast the information based on DDBF (distributed discarding bloom filter), which discards some information when transmitted to their neighbors. During the search process, PST extends the concept of walker in RW to search team. PST realizes collaborative and parallel search of multiple search teams by aggregating the resource information obtained in search process. Experimental results show that PST achieves a good tradeoff between performance and overhead.

Key words: probabilistic search team; distributed discarding Bloom filter; search direction; search intensity; iVCE

摘 要: 提出一种面向大规模 P2P 系统的概率搜索小组(probabilistic search team,简称 PST)算法.各节点首先发布本节点的资源共享信息,并基于分布式丢弃 Bloom Filter 技术(distributed discarding bloom filter,简称 DDBF)对从其他节点收到的信息进行保存和转发.PST 算法把 RW 算法中漫步者的概念扩充为搜索小组.通过聚合各小组在搜索过程中获得的资源信息,PST 算法实现了多个小组之间相互协同的并行搜索.分析模拟结果表明,PST 算法在保持低定位开销的同时取得了较好的定位性能.

关键词: 概率搜索小组;分布式丢弃 Bloom Filter;搜索方向;搜索强度;虚拟计算环境

中图法分类号: TP393 **文献标识码**: A

互联网资源的“成长性”、“自治性”和“多样性”等自然特性,给虚拟计算环境(iVCE)中的资源定位带来了巨大的挑战[1].非结构化 P2P 资源定位方法由于其简单性和易用性,目前在 Internet 上得到了大量应用.然而,Christos 等人的研究表明,受限泛洪算法[2]、RW 算法[3]、ARW 算法[4]等“盲搜索(blind search)”类方法在搜索过程中具有很大的盲目性,导致当资源请求节点距离资源共享节点较远时将产生大量的冗余消息,无法迅速定位资源[3].近年来,出现了很多关于“提示性搜索(informed search)”的研究,其主要思想是由各资源共享节点发布共

* Supported by the National Natural Science Foundation of China under Grant Nos.60673167, 60703072 (国家自然科学基金); the National Basic Research Program of China under Grant No.2005CB321801 (国家重点基础研究发展计划(973))

Received 2006-07-10; Accepted 2007-01-23

享信息,并在网络中传播和维护这些共享信息,用于指导资源定位消息的转发[5−7].

维护与更新索引信息需要消耗大量的存储空间和网络带宽.研究者们提出了 Bloom Filter(BF)技术[8]表示共享资源[5,9].为了进一步降低共享信息的维护开销和正向错误的概率,指数衰减 Bloom Filter(EDBF)[9]对传统 Bloom Filter 进行了改进,每个度数为 d 的节点维护了一个具有 d 个表项的一维概率路由表,表中每个表项是一个 Bloom Filter 向量,分别维护了通过各邻居节点可达的资源信息.在信息发布阶段,首先设定一个全局统一的固定衰减比例 d,在传播资源信息的每一步,每个中间节点仅保留所接收到资源信息的$(1–d)$.

上述方法的问题在于,为了满足系统的可扩展性要求,共享信息的传播和维护只能在一个相对较小的范围内进行,因此,对转发资源定位消息的指导作用较为有限.本文提出一种基于分布式丢弃 Bloom Filter(distributed discarding bloom filter,简称 DDBF)技术的概率搜索小组(probabilistic search team,简称 PST)算法.首先,各节点发布自己的资源共享信息,为了减小资源信息的维护和更新开销,在资源信息的传播过程中,各节点基于 DDBF 技术对收到的信息在丢弃一定比例(不同节点对不同邻居的丢弃比例可能不同)后进行转发.其次,PST 算法把传统漫步算法中漫步者(walker)的概念扩充为搜索小组(search team).资源请求节点发出 k 个搜索小组(资源定位消息),在资源搜索过程中,根据资源信息的分布情况动态调整各小组的搜索方向和搜索强度.通过聚合各小组在搜索过程中获得的资源信息,PST 算法实现了多个小组之间相互协同的并行搜索.分析模拟结果表明,与现有同类算法相比,PST 算法能够在保持低定位开销的同时取得较好的定位性能.

1 算法设计

1.1 分布式丢弃Bloom Filter

EDBF[9]在一定程度上降低了共享信息维护开销和正向错误概率,提高了搜索性能.然而在大规模 P2P 系统中,通常各节点具有不同的节点度数、网络带宽和错误率要求等限制,进而对邻居节点的衰减比例有不同的要求.因此,EDBF 统一设定衰减比例的方式无法适应实际 P2P 系统的异构、自治的特点.针对上述不足,本文提出支持异构丢弃比例的“分布式丢弃 Bloom Filter(DDBF)”技术.

在 EDBF 中,衰减比例是一个统一设定的常量,从而保证资源信息的强度与各节点到信息发布节点的距离形成一种严格的单调递减关系.与 EDBF 不同,由于 DDBF 在传播资源信息的过程中,允许各中间节点针对不同邻居选择不同的丢弃比例,从而使 DDBF 无法保证这种严格的单调递减关系,进而无法简单地仅根据资源信息的强弱来判断各节点与信息发布节点的距离.因此,在 DDBF 中,每个度数为 d 的节点维护了一个 d 行 c 列的二维邻居信息表 T,表中每一个表项是一个 Bloom Filter 向量.表项 $T_{ij}(1\le i\le d,1\le j<c)$维护了通过第 i 个邻居且从信息发布节点经过 j 步到达本节点的资源信息;表项 $T_{ic}(1\le i\le d)$则维护了通过第 i 个邻居且从信息发布节点经过 c 步或 c 步以上到达本节点的资源信息.在传播资源信息的过程中,各中间节点根据信息的上一跳节点(确定行号)和已经过跳步数(确定列号)把结果保存在邻居信息表 T 的相应表项中,并分别按照各邻居节点相应的丢弃比例对信息进行丢弃并传播.为了便于叙述,本文将使用如下简化方法表示:

- $BF(x)$表示资源 x 的 Bloom Filter 向量;
- $BF(A)$表示节点 A 的共享资源集合的 Bloom Filter 向量,$Resource(A)$表示节点 A 的共享资源集合;
- $T^{A}_{B,j}(1\le j\le c)$ 表示节点 A 的邻居信息表中对应于邻居节点 B 的行第 j 列 Bloom Filter 向量;
- $DDBF(A,B,j)$表示节点 A 向其邻居节点 B 转发的资源更新消息,该消息在到达节点 A 时已经传播的跳步数为 j,j 为 0 表示节点 A 为该信息的原始发布节点;
- $DP_{A,N}$ 表示节点 A 对邻居 N 的丢弃比例,$p_{A,N}$ 表示节点 A 对邻居 N 的保留比例,$p_{A,N}=1–DP_{A,N}$.

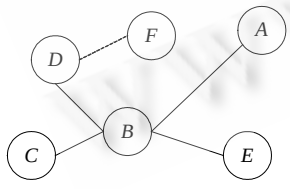

Fig.1 Example of unstructured P2P system

图 1 非结构化 P2P 系统举例

下面以图 1 所示的 P2P 系统为例,说明使用 DDBF 技术进行资源信息的传播和更新过程.

当节点 A 增加新的资源时,它首先检查本节点资源集合的

Bloom Filter 值 $BF(A)$.通过比较新旧 $BF(A)$值的差异(按位异或)计算$\delta(BF(A))$,进而有 $DDBF(A,B,0)=\delta(BF(A))$,然后把 $DDBF(A,B,0)$发送给邻居节点 B.节点 B 首先通过比较原 $T^B_{A,1}$ 和 $DDBF(A,B,0)$的差异(按位异或)得到新的向量值,然后把 $DDBF(A,B,0)$中比例为 $DP_{B,E}$ 的值为 1 的位重置为 0,得到 $DDBF(B,E,1)$,然后发送给节点 B 的邻居节点 E.节点 E 将根据此信息更新邻居信息表的对应表项,并通过与节点 B 类似的步骤把更新信息传播给自己的邻居节点.当有一个新的节点 F 加入并成为节点 D 的邻居时可以被看作上述过程的特例,此时的$\delta(BF(F))$就是节点 F 全部资源集合的 Bloom Filter 值 $BF(F)$,进而有 $DDBF(F,D,0)=BF(F)$,节点 F 把 $DDBF(F,D,0)$发送给邻居节点 D.

为了降低资源信息传播过程中的通信开销,实际系统中的节点并不是从某邻居节点收到更新消息时立刻传播给其他邻居节点,而是以批处理方式进行周期性传播.例如,为了更新邻居节点 E 的 $T^E_{B,2}$ 表项,节点 B 对一个周期内从其他邻居节点收到的所有更新消息 $DDBF(A,B,0)$,$DDBF(D,B,0)$和 $DDBF(C,B,0)$进行按位或操作,然后把或操作结果中比例为 $DP_{B,E}$ 的值为 1 的位重置为 0,得到 $DDBF(B,E,1)$并发送给节点 E.

各节点周期性地处理接收到的更新消息并产生新的更新消息的算法见算法 1.

算法 1.

Procedure ReceiveAndCreateUpdate (Node A) /*更新邻居信息表并产生针对各邻居的更新信息*/

1. for each $U\in neighbors(A)$ { //节点 A 的每一个邻居节点 U
2. for (j=0; j<c; j++) /*更新本节点信息表的对应表项*/
3. $T^A_{U,j+1}\leftarrow T^A_{U,j+1}\wedge DDBF(U,A,j)$; $T^A_{U,c}\leftarrow T^A_{U,c}\wedge DDBF(U,A,c)$;
4. for (j=1; j<=c; j++) { /*产生对节点 U 的更新信息*/
5. $DDBF(A,U,j)\leftarrow 0$; //首先初始化对节点 U 的更新消息
6. for each $V\in neighbors(A)$, $V\neq U$ { //节点 A 的每一个邻居节点 V,$V\neq U$
7. $DDBF(A,U,j)\leftarrow DDBF(A,U,j)|DDBF(V,A,j-1)$; //对所有更新消息进行或操作
8. if (j==c) $DDBF(A,U,j)\leftarrow DDBF(A,U,j)|DDBF(V,A,j)$}
9. $DDBF(A,U,j)\leftarrow DDBF(A,U,j)\times(1-DP_{A,U})$;} //丢弃比例为 $DP_{A,U}$ 的更新信息
10. $DDBF(A,U,j)\leftarrow\delta(BF(A))$; *SendAllUpdates*($A$,$U$);} /*把所有更新发送到节点 U*/

1.2 概率搜索小组(PST)算法

本文提出一种新的可扩展快速资源定位算法:概率搜索小组算法,其基本思想是聚合搜索过程中在各中间节点获得的资源信息,实现相互协同的并行搜索.

与传统 Bloom Filter 不同,DDBF 在中间节点 A 丢弃了部分资源信息,因此,在判断通过某邻居节点 N 第 j 步是否可以访问到资源 x 的时候,不能简单地回答“是”或“否”,而是返回一个表示匹配度的值.假设资源 x 对应的位向量为 U,$T^A_{N,j}$ 表项为 V,资源 x 与 $T^A_{N,j}$ 表项的匹配度 $Sim(x,T^A_{N,j})$ 可以通过式(1)进行计算.

$$Sim(x,T^A_{N,j})=\sum_{i=1}^{m}(U[i]\times V[i])\Big/\sum_{i=1}^{m}U[i] \tag{1}$$

在 PST 算法中,我们称资源定位消息为搜索小组 ST.如果一个定位消息在节点 A 仅转发给一个邻居节点 B,则称 ST 从节点 A 跳至节点 B;否则,如果一个定位消息在节点 A 转发给多个邻居节点,则称 ST 在节点 A 分派出多个子搜索小组,此时,ST 将驻留在节点 A 并且以 ST_A 表示,分派至某邻居节点 N 的子搜索小组以 $ST_{A,N}$ 表示.特别地,如果资源请求节点 $N_{original}$ 最初发出 k 个资源定位消息,那么,我们认为初始搜索小组 $ST_{original}$ 发出了 k 个子搜索小组,并且 $ST_{original}$ 驻留在节点 $N_{original}$.假设 ST 为 $ST_{original}$ 发出的某搜索小组,经过 m 步到达中间节点 A,我们称其为 ST_A.ST_A 的搜索过程如下:

(1) 在中间节点 A,ST_A 根据式(1)计算当前节点邻居信息表中各 Bloom Filter 向量与目标资源的匹配度并得到局部最大匹配度 Max$Sim_{l,A}$.如果 Max$Sim_{l,A}$ 大于 ST_A 已知的全局最大匹配度 MaxSim_g,则把 Max$Sim_{l,A}$ 通告给 $ST_{original}$,$ST_{original}$ 通过比较各局部最大匹配度,获得当前的全局最大匹配度 MaxSim_g.

(2) ST_A 向非 0 局部最大匹配度 Max$Sim_{l,A}$ 所对应的邻居节点 N 转发定位消息(我们称其为第 1 类子小组,以 $ST^1_{A,N}$ 表示), $ST^1_{A,N}$ 取 Max$Sim_{l,A}$ 对应 Bloom Filter 向量的列号作为该子小组的 *TTL* 值.

(3) 在中间节点 A,除了向具有 Max$Sim_{l,A}$ 的邻居节点转发定位消息以外,ST_A 还向所有匹配度大于或等于某阈值的邻居节点转发定位消息,该阈值等于全局最大匹配度 MaxSim_g 乘以容错因子 $\alpha(\alpha\le 1)$,即 ST_A 将向所有满足 $P(x,T^A_{N,j})\ge\alpha\times\text{Max}Sim_g, 1\le j\le c$,且与第 1 类子小组不重复的邻居 N 转发定位消息(我们称其为第 2 类子小组,以 $ST^2_{A,N}$ 表示). $ST^2_{A,N}$ 取满足 $P(x,T^A_{N,j})\ge\alpha\times\text{Max}Sim_g, 1\le j\le c$ 的最大 j 值作为 *TTL* 值.

由于 DDBF 在资源信息发布过程中允许各中间节点选择不同的丢弃比例,因此,在选择定位消息的路由方向时,除了匹配度大小,各中间节点还将比较各非 0 匹配度所对应 Bloom Filter 向量所在的列号,列号越小,则表示资源信息到达本节点前所经过的跳步数越小,进而信息发布节点距离本节点就可能越近.

(4) ST_A 首先计算出局部最大匹配度 Max$Sim_{l,A}$ 对应的向量所在的列号 $j_{\max}$,然后,向所有满足 $Sim(x,T^A_{N,j})>0$, $1\le j<j_{\max}$ 且与前两类子小组的方向不重复的邻居节点 N 转发定位消息(我们称其为第 3 类子小组,以 $ST^3_{A,N}$ 表示).每个 $ST^3_{A,N}$ 取满足 $Sim(x,T^A_{N,j})>0,(1\le j<j_{\max})$ 的最大 j 值作为该小组的 *TTL* 值.其任务是确定关于"通过节点 N 经过 j 步即可到达信息发布节点"的判断是否正确.

在下一跳节点 N,子小组 $ST^i_{A,N}, i=1,2,3$ 将重复父小组 ST_A 的操作(上述 1~4 步),这里不再赘述.系统中的某节点(例如节点 B)转发资源定位消息的算法见算法 2.

算法 2.

```
Procedure ForwardQuery (Node B, Team ST, Resource x)                    /*转发资源定位消息*/
1    if (HasFound(B,x)) Return B;                                       //在节点 B 发现了目标资源
2    if (HasSeenBefore(B,ST)) {SelectAnotherMaxSimNeighbor(ST); Return 0;} //是重复消息
3    ST_parent←ST→parent; A←ST_parent→reside_node; ST→MaxSim←0;         //设置父搜索小组驻留的节点
4    for each U∈neighbors(B)∧U≠A
5        for (j=1; j++; j<=c)
6            if  (Sim(x,T^B_{U,j}) > ST → MaxSim)  {
7                ST → MaxSim ← Sim(x,T^B_{U,j}) ; MaxSim_Node←U; MaxSim_TTL←j;}
8    if ((ST→MaxSim<A→MaxSim) && (ST→type==1||ST→type==2))
9        {GoBackToParent(ST); return 0;}
10   if (ST→MaxSim>ST_parent→MaxSim)    ST_parent→MaxSim←ST→MaxSim;
11   CandidateSet←{Null};                                               //初始化待转发的节点集合
12   if (ST→type==1||ST→type==2)
13       if (ST→MaxSim>0) {
14           GenerateSubST(B,MaxSim_Node,ST,MaxSim_TTL,1);
15           CandidateSet←CandidateSet∪{MaxSim_Node};}
16   if (ST→type==3)
17       if (∃j'<ST→TTL,∃V∈neighbors(B),使得 Sim(x,T^B_{V,j'}) ≥ Sim(x,T^A_{B,ST→TTL0}) ){
18           if (V∉CandidateSet) {
19               GenerateSubST(B,V,ST,j',3); CandidateSet←CandidateSet∪{V};}
20       } else {Terminate(ST); Return 0;}
21   for each U∈neighbors(B)∧U≠A∧U∉CandidateSet
22       for (j=1; j++; j<=c)
23           if  (Sim(x,T^B_{U,j}) ≥ α×(ST_parent → MaxSim))  {//α≤1
24               GenerateSubST(B,U,ST,j,2); CandidateSet←CandidateSet∪{U};}
25   if (CandidateSet==∅) {
26       ST→TTL←ST→TTL−1;if (ST→TTL==0) {Terminate(T); Return 0;}
27       SelectRandomNeighbor(V); V→ST←ST; CandidateSet←{V};}
28   for each V∈CandidateSet ForwardQuery(V,V→ST,x);                    //在下一跳节点进行相同处理
29   ContactRequestingNode(B,ST);
30   Return 0;                                                          //在节点 B 还没有发现目标资源
```

为了进一步提高资源定位性能,在查找资源 x 的最初阶段,如果在经过连续 NH 步搜索之后所有小组都没有

发现任何目标资源的共享信息,那么,每个小组都将在第NH+1步派出NT个子小组以加强全局搜索.当全局最大匹配度MaxSim_g大于停止大规模搜索的匹配度阈值MaxSim_0时,将终止所有未发现任何信息的小组的搜索.

2 分 析

在分析过程中,我们假设系统中共有n个节点,每个节点的邻居信息表有d行c列(d为节点度数).每个节点上有l个资源,每个Bloom Filter向量有m位,有k个哈希函数,且$m>>k\times l$.我们进一步假设系统拓扑为随机图结构,节点度数在$[a,b]$范围内均匀分布,平均节点度数为$d_a=(a+b)/2$,根据Kumar的分析[9],当i较小时,可以近似认为系统拓扑类似于树形结构(无环),进而与任意节点A的距离为i的节点数$n_i\approx d_a\times(d_a-1)^{i-1}$.

2.1 错误率

下面仅对PST算法的分析结果进行简要介绍,详细的分析过程请参考文献[10].在中间节点A选择资源定位消息的下一跳转发节点集合S时,PST算法有可能发生错误.如果集合S中包括了错误路由方向的邻居节点,则认为发生了正向错误;否则,如果集合S中没有包括正确路由方向的邻居节点,则认为发生了负向错误.

我们以(负向)错误率$P_{err,i}$(或正确率$P_{eff,i}=1-P_{err,i}$)来表示PST算法的路由准确性.所谓错误率,是指在距离信息发布节点实际距离为i的中间节点A对定位消息进行路由时,所选择的转发节点子集中没有包括正确方向(资源信息发布方向的反方向)上的邻居节点的概率.假设系统中各节点对不同邻居节点的保留比例p为随机变量,并且近似认为任给p_i和n,近似有$\prod_{i=1}^{n}p_i\approx p_0^n$,其中,$p_i$为任意选取的任意节点对任意邻居的保留比例,$n$为任意正整数,$p_0$为小于1的常数.当错误方向匹配度的最大值(噪声)与容错因子α的乘积大于正确方向匹配度的最大值时,路由将发生错误.

用于表示自身所共享资源的向量$BF(U)$中1的个数的期望为$w\approx kl$.令$q=(d_a-1)\times p_0$,令所有与节点A距离为i的节点资源信息中值为1的位传播到A的个数期望为S_i,任意节点与节点A的最大距离为h_0,有

$$S_i=n_i\times w\times\prod_{j=1}^{i-1}p_j\approx d_a\times(d_a-1)^{i-1}\times w\times p_0^{i-1}=d_awq^{i-1},\ 1\le i<c \tag{2}$$

$$S_c\approx\sum_{i=c}^{h_0-1}S_i\approx d_awq^{c-1}(1-q^{h_0-c})/(1-q) \tag{3}$$

第i列的每个Bloom Filter中任意一位为1的概率$p_r(i)$可以通过式(4)和式(5)计算

$$p_r(i)\approx1-(1-1/m)^{S_i/d_a}\approx1-\mathrm{e}^{-wq^{i-1}/m}\approx1-\mathrm{e}^{-klq^{i-1}/m}\approx klq^{i-1}/m,\ 1\le i<c \tag{4}$$

$$p_r(c)\approx1-(1-1/m)^{S_c/d_a}\approx klq^{c-1}(1-q^{h_0-c})/m(1-q) \tag{5}$$

以随机变量Y_i表示由于噪声影响任意中间节点A邻居信息表的$T_{N,i}^A(1\le i\le c)$表项中,目标资源的k个哈希函数值的对应位中1的个数,以$P_r(Y_i=y)$表示Y_i的值为y的概率(y为不大于k的非负整数),有

$$P_r(Y_i=y)=C_k^y p_r(i)^y(1-p_r(i))^{k-y},\ 1\le i\le c \tag{6}$$

假设目标资源信息经过h跳步,通过节点B到达中间节点A.当$h<c$时,$T_{B,h}^A$表项中目标资源的k个哈希函数值的任意对应位为1的概率$r(h)$可以通过式(7)计算:

$$r(h)=p_0^{h-1}+p_r(h)-p_0^{h-1}\times p_r(h),\ 1\le h<c \tag{7}$$

当$h\ge c$时,$T_{B,c}^A$表项中目标资源的k个哈希函数值的任意对应位为1的概率$r(h)$可以通过式(8)计算:

$$r(h)=p_0^{h-1}+p_r(c)-p_0^{h-1}\times p_r(c),\ h\ge c \tag{8}$$

以随机变量X表示$T_{B,h}^A$表项($h<c$)或$T_{B,c}^A$表项($h\ge c$)中目标资源的k个哈希函数值的对应位中1的个数,以$P_r(X=x)$表示X的值为x的概率(x为不大于k的非负整数),有

$$P_r(X=x)=C_k^x r(h)^x(1-r(h))^{k-x} \tag{9}$$

在资源信息发布的过程中,假设目标资源信息x经过h跳步,通过节点B到达中间节点A.当节点B对应行

的 c 个表项中至少有 1 项 $T_{B,j}^A$ 满足 $Sim(x,T_{B,j}^A)\geq\alpha\times \mathrm{Max}Sim(j\in[1,c],\alpha\leq 1)$ 时,即可使该次路由成为一次有效路由(转发节点子集中包括节点 B).首先,由于表项 $T_{B,h'}^A$ 的影响完成有效路由的概率 $P_{eff,h'}$ 为

$$P_{eff,h'}=\left(\prod_{i=1}^{c}P_r(X\geq\alpha Y_i)\right)^{d-1}=\prod_{i=1}^{c}{}^{d-1}\left(\sum_{s=1}^{k}(P_r(X=s)\cdot\sum_{j=0}^{\min(\lfloor s/\alpha\rfloor,k)}P_r(Y_i=j))\right) \tag{10}$$

然后考虑噪声的影响.在没有目标资源信息时,路由选择完全等价于随机转发,因此有 $P_{eff,h}=1/d$.

令 $f(d)$表示节点度数为 d 的概率,由于假设节点度数在$[a,b]$范围内均匀分布,因此,$f(d)$为常数 $1/(b-a+1)$,进而在距离信息发布节点距离为 h 的中间节点 A 的路由错误率 $P_{err,h}$ 和路由正确率 $P_{eff,h}$ 为

$$P_{err,h}\approx 1/(b-a+1)\cdot\sum_{d=a}^{b}(1-P_{eff,h'})(1-P_{eff,noise}) \tag{11}$$

$$P_{eff,h}=1-P_{err,h} \tag{12}$$

取 n=10000(最大距离 h_0=8),d_a=4,k=64,l=4,m=32k bit,p_0=0.3,α=0.5,c=5,a=3,b=5,MaxSim_0=4/64(停止大规模搜索的匹配度阈值).由式(11)得到 $P_{err,h}$ 的值见表 1.

Table 1 Computing the error rate

表 1 错误率数值计算

Distance (h)	1	2	3	4	5	6	7
Error rate $P_{err,h}$	1.15×10^{-14}	1.54×10^{-9}	2.03×10^{-2}	6.29×10^{-1}	7.22×10^{-1}	7.36×10^{-1}	7.37×10^{-1}

2.2 定位性能

随机选择路由方向的错误率 $P_{err,random}$ 可由式(13)计算.在下面的分析中,认为当 $h>h_1$ 时,资源信息的影响远小于噪声的影响,h_1 满足 $P_{err,h_1}\approx P_{err,random}$;认为当 $h\leq h_2$ 时,资源信息的影响远大于噪声的影响,h_2 满足 $P_{err,h_2}\approx 0$;认为当 $h_2<h\leq h_1$ 时,资源信息的影响与噪声的影响相当.

$$P_{err,random}=1/(b-a+1)\cdot\sum_{d=a}^{b}(1-P_{eff,noise}) \tag{13}$$

(1) 在距离信息发布节点很近的区域($h\leq h_2$)发起的搜索需经过的跳步数为

$$hops_1(h)\approx h \tag{14}$$

(2) 在上述两个区域之间的区域($h_2<h\leq h_1$)发起的搜索需要经过的跳步数为

$$hops_2(h)\approx\sum_{i=h_2+1}^{h}(P_{eff,i}\times hops_{eff,i}+P_{err,i}\times hops_{err,i})+h_2\approx h+(d_a-1)\times\sum_{i=h_2+1}^{h}P_{err,i} \tag{15}$$

(3) 在距离信息发布节点很远的区域($h>h_1$)发起的搜索需要经过的跳步数为

$$hops_3(h)\approx(NH+1)\times h-NH\times h_1+(d_a-1)\times\sum_{i=h_2+1}^{h_1}P_{err,i} \tag{16}$$

其中,NH 的含义参见第 1.2 节.与信息发布节点距离为 h 的节点个数 $n_h\approx d_a\times(d_a-1)^{h-1}$.令 $P_r(h)$表示资源请求节点 $N_{original}$ 与信息发布节点的距离为 h 的概率,有 $P_r(h)=n_h/n$.平均搜索延迟 $E(hops)$为

$$E(hops)=\sum_{h=1}^{h_2}(P_r(h)\times hops_1(h))+\sum_{h=h_2+1}^{h_1}(P_r(h)\times hops_2(h))+\sum_{h=h_1+1}^{h_0}(P_r(h)\times hops_3(h)) \tag{17}$$

取 NH=2,h_1=5,h_2=2.当 n=10000(或 2500)时,根据式(17),平均搜索延迟见表 2.

Table 2 Computing average search latency

表 2 平均搜索延迟计算

No. of nodes (n)	Maximum distance (h_0)	Average search latency ($E(hops)$)
10 000	8	16.152
2 500	7	12.529

3 模拟验证

我们在 Neurogrid 模拟器[11]中实现了不同配置下的概率搜索小组(PST)算法,并且与 RW 算法(2 路随机转发)、受限泛洪算法以及扩展 SQR 算法[9]等进行了比较.各算法的公共模拟参数见表 3.

Table 3 Common simulation parameters

表 3 公共模拟参数

Parameter	Value	Parameter	Value
No. of nodes (n)	2 500,2 000	No. of available keys	3 000
Average node degree (d_a)	4	No. of keys per resource	1
Node degree distribution	Random	Resource and key distribution	Random
No. of available resources	2 500	Search distribution	Random
No. of resources per node (l)	2	No. of experiments	10 000

各算法的私有模拟参数见表 4.为了突出 DDBF 与 EDBF 的区别,保证比较的公平性,我们对 SQR 算法进行了扩展,在搜索的最初阶段采用了与 PST 算法相同的加强全局搜索方法,并且一旦发现关于目标资源的任何信息,则停止所有其他定位消息的进一步转发.在扩展 SQR 算法的模拟中,我们取衰减比例为 75%(常数).

Table 4 Private simulation parameters

表 4 私有模拟参数

Parameter	Value	Parameter	Value
PST (With DDBF)		SQR (With EDBF)	
Bloom filter width (m)	64k bit	Bloom filter width (m)	64k bit
No. of hash functions (k)	64	No. of hash functions (k)	64
Average discarding proportion (DP)	75% (3/4)	Decay factor (constant)	75% (3/4)
Initial no. of search team	3	Gnutella	
No. of hops between 2 enhances (NH)	2	Forwarding manner	Flooding
No. of sub-teams for each enhances (NT)	3	Random walk	
DP distribution: Uniform distributed from 62.5% to 87.5%		Forwarding manner	2-way

在 PST 算法中,如果发现全局最大匹配度 MaxSim_g 大于 MaxSim_0,那么将停止大规模搜索;而在 PST′算法中,没有采取任何控制定位开销的措施.模拟实验结果见表 5 以及图 2 所示.下面基于表 5 对结果进行分析.

Table 5 Latency and overhead (number of nodes: 2 500)

表 5 定位延迟和定位开销(节点数:2 500)

	Latency	Cost		Latency	Cost		Latency	Cost
PST	8.326	160.8	PST′	8.324	234.3	Gnutella	5.331	7 491.1
PST-E (theory)	12.496	Null	SQR-E (extended)	9.096	138.3	RW	14.473	4 318.4

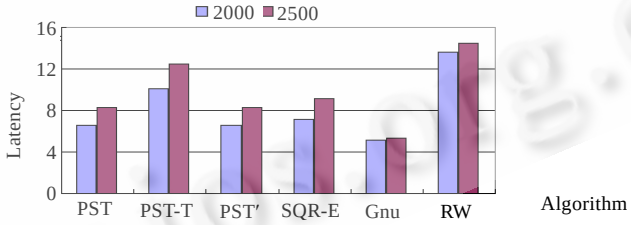

Fig.2 Latency (number of nodes: 2000,2500)

图 2 延迟(节点数:2000,2500)

与基于式(17)得到的平均定位延迟的理论值相比,模拟结果与理论推导值相差 33.4%,这是因为理论推导中假设与任意节点 A 的距离为 i 的节点数在 i 较大时与模拟过程中所采用的随机拓扑相差较远.通过比较 PST 算法和 PST′算法,我们可以看出,它们的平均延迟基本相同;同时,PST 算法的平均开销比 PST′算法下降了约 31.4%,证明 PST 算法的开销控制策略有效地降低了总定位开销.PST 算法的平均延迟与泛洪算法类似,只有约 RW 算法的一半,同时,PST 算法的平均开销比两者低 1 个数量级.

与扩展 SQR 算法相比,PST 算法的定位开销和计算复杂度都有一定的增加.然而,PST 算法的定位延迟与扩

展 SQR 算法相比降低了大约 8.47%.更为重要的是,PST 算法支持在传播资源信息的过程中异构地选择丢弃比例.我们进一步模拟了不同 *DP* 均值和 *NH* 参数下的 PST 算法,结果见表 6.

Table 6 Latency and overhead of PST under different combination of *DP* and *NH*

表 6 不同 *DP* 和 *NH* 组合下的 PST 算法的延迟和开销

	Search latency	Search cost		Search latency	Search cost
PST (50%,2)	7.780	103.5	PST (66.7%,4)	8.497	76.4
PST (50%,4)	8.181	52.3	PST (75%,2)	8.324	160.8
PST (66.7%,2)	7.892	142.0	PST (75%,4)	9.602	112.2

4 结束语

针对传统随机漫步者算法及其改进算法的不足,本文提出了基于 DDBF 技术的概率搜索小组(PST)算法.在 PST 算法中,各节点首先发布本节点的资源共享信息,并基于 DDBF 技术对从其他节点收到的信息进行保存和转发.PST 算法根据资源信息的分布情况动态调整各小组的搜索方向和搜索强度.通过聚合各小组(及其子小组)在搜索过程中获得的资源信息,PST 算法实现了多个小组之间相互协同的并行搜索.

References:

[1] Lu XC, Wang HM, Wang J. Virtual computing environment (IVCE): Concept and architecture. Science in China (Series E), 2006,36(10):1081−1099 (in Chinese with English abstract).

[2] Matei R, Ian F, Adriana I. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Computing Journal, 2002,6(1):50−57.

[3] Christos G, Milena M, Amin S. Random walks in peer-to-peer networks. In: Proc. of the IEEE INFOCOM 2004. New York: IEEE Press, 2004. 120−130.

[4] Zheng QB, Lu XC, Zhu PD, Peng W. An efficient random walks based approach to reducing file locating delay in unstructured P2P network. In: Proc. of the IEEE GLOBECOM 2005, Vol.2. St. Louis: IEEE Press, 2005. 980−984.

[5] Francisco MCA, Christopher P, Richard PM, Thu DN. PlanetP: Using gossiping to build content addressable peer-to-peer information sharing communities. Technical Report, DCS-TR-487, Piscataway: Rutgers University, 2002.

[6] Yatin C, Sylvia R, Lee B, Nick L, Scott S. Making Gnutella-like P2P systems scalable. In: Proc. of the ACM SIGCOMM 2003. New York: ACM Press, 2003. 407−418.

[7] Beverly Y, Hector GM. Efficient search in peer-to-peer networks. In: Proc. of the ICDCS 2002. Vienna: IEEE Computer Society, 2002. 5−14.

[8] Burton HB. Space/Time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970,13(7):422−426.

[9] Abhishek K, Jun (Jim) X, Ellen WZ. Efficient and scalable query routing for unstructured peer-to-peer networks. In: Proc. of the IEEE INFOCOM. 2005. 1162−1173.

[10] http://vce.org.cn/ymzhang/PST_TR.pdf

[11] Sam J. NeuroGrid: Semantically routing queries in peer-to-peer networks. In: Proc. of the Int'l Workshop on Peer-to-Peer Computing. Pisa: IEEE Computer Society, 2002. 202−214.

附中文参考文献:

[1] 卢锡城,王怀民,王戟.虚拟计算环境 iVCE:概念与体系结构.中国科学(E 辑),2006,36(10):1081−1099.

张一鸣(1978－),男,山东济南人,博士生,主要研究领域为 P2P 资源定位技术,高性能并行分布处理技术.

郑倩冰(1977－),女,博士,主要研究领域为 P2P 资源定位技术.

卢锡城(1946－),男,教授,博士生导师,中国工程院院士,CCF 高级会员,主要研究领域为分布处理技术,网络技术.

李东升(1978－),男,博士,CCF 会员,主要研究领域为计算机网络,P2P 资源定位技术.