第 34 卷 第 20 期 计 算 机 工 程 2008 年 10 月
Vol.34 No.20 Computer Engineering October 2008
·软件技术与数据库· 文章编号：1000—3428(2008)20—0029—02 文献标识码：A 中图分类号：TP391

基于 Kademlia 的 P2P 多维范围查询系统

侯祥松[1]，曹元大[2]，张 煜[1]
(1. 北京理工大学计算机科学技术学院，北京 100081；2. 北京理工大学软件学院，北京 100081)

摘 要：结构化 P2P 网络具有良好的可扩展性，但难以支持多关键词查询、范围查询等复杂查询。该文分析已有复杂查询方法，提出一种基于 Kademlia 的 P2P 多维范围查询系统 K-net。K-net 在进行多维范围查询时，完成高维范围数据降维和范围划分，减少查询所需带宽。模拟结果显示，该系统具有良好的查询准确度和可扩展性。

关键词：结构化 P2P；多维范围；空间填充曲线

P2P Multi-dimensional Range Query System Based on Kademlia

HOU Xiang-song [1], CAO Yuan-da[2], ZHANG Yu[1]
(1. School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081;
2. School of Software, Beijing Institute of Technology, Beijing 100081)

【Abstract】 Structured P2P networks has good expandability, but it is difficult to support multiple keywords query or range queries. This paper analyzes the existing complex query, and proposes a P2P multi-dimensional range query system based on Kademlia which named K-net. When processes multi-dimensional range query, it deals with dimension reduction and range partition simultaneously. This method can reduce the bandwidth required for query. The simulation results show that this system has good query accuracy and expansibility.

【Key words】 structured P2P; multi-dimensional range; space filling curve

1 概述

P2P是一项新技术，近年来逐渐被人们关注。早期P2P是集中式的，如Napster，由于其资源存放于一个目录服务器中，因此存在瓶颈问题。为了解决此问题，出现了分散化P2P，但因为其工作方式是洪泛的，所以容易造成网络拥塞。以上 2 种非结构化方式的优点是系统构造简单、支持多维(多关键词)和部分匹配等多种查询方式，但在查询准确率和查询时间上均不够理想。为了解决非结构化方式带来的问题，出现了结构化P2P。此方式采用分布式哈希表(Distributed Hash Table, DHT)，如Kademlia[1]。它在各节点间建立某些联系，并在搜索中利用这些信息提高搜索性能。

随着资源共享需求的增长，越来越多异构资源开始系统地利用P2P实现资源管理和查询。例如网格计算[2]中一个可能的查询请求可以是寻找内存大小在 1 GB~2 GB之间、Linux操作系统、提供Java环境的系统，由于该查询是一个多维范围查询，因此需要结构化P2P提供多维范围查询功能。传统结构化P2P只支持精确查询，对多维、模糊、范围等复杂查询缺乏有效支持。本文基于DHT结构中的Kademlia设计了K-net，使用空间填充曲线将数据映射到Kademlia结构上，从而很好地支持多维范围查找。

2 相关工作

Kademlia 协议是一种 DHT 技术，它以异或算法为距离度量基础，建立了一种全新 DHT 拓扑结构，极大提高了路由查询速度。Kademlia 协议包括 4 种远程 RPC 操作：PING, STORE, FIND_NODE, FIND_VALUE。

多维范围查询是在多维空间中寻找落在某个值域范围内的数据，结构化 P2P 不能处理多维数据，因此，需要把它转化为若干个一维数据。文献[3]首次使用空间填充曲线，将与数据相近的区间映射到 CAN 中相近的节点上。空间填充曲线是一种充满空间、连续不可导、非自交、自相似的简单曲线，它在维数大于 1 的欧几里德空间内有一个非空的内部，即唯一一次通过空间中的每个点。文献[4]利用空间填充曲线，把一个 d 维数据映射到一维空间，实现对多维查询的支持。

3 多维范围数据处理

本文使用一种空间填充曲线，即 Z_order 进行降维，Z_order 如图 1 所示。

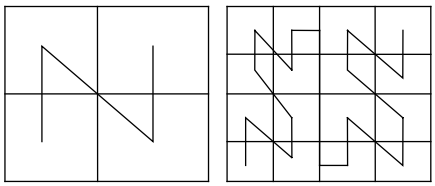

图 1 Z_order 示意图

假定初始值V有 2 个属性(x,y)，属性的值域为[0,1]。通过Z_order将其降为一维的位串$B(b_1b_2\cdots b_n)$。降维算法 $GetID(D)$ 如下：

(1)如果x位于值域的左半部，则b_1为 0，否则b_1为 1。

(2)如果y位于值域的左半部，则b_2为 0，否则b_2为 1。

(3)x,y 的值域减少一半，递归以上步骤。

例如，将二维数据(0.8, 0.2)转换为 4 位位串的步骤如下：

(1)0.8 位于值域[0.1]右半部，值为 1；0.2 位于值域[0,1]

作者简介：侯祥松(1977－)，男，博士研究生，主研方向：分布式计算，移动计算；曹元大，教授、博士生导师；张 煜，博士研究生
收稿日期：2008-01-20 **E-mail：**houxsc@bit.edu.cn

左半部，值为 0。

(2)0.8 位于值域(0.5,1)右半部，值为 1；0.2 位于值域[0, 0.5]左半部，值为 0。

(3)最后可得结果为 1010。

当数据是多维范围时，可以根据 Z_order 的划分过程来扩展对多维范围数据的处理。对 n 维范围数据 $M[(l_1,h_1),(l_2,h_2),\cdots,(l_n,h_n)]$，利用算法 Split-Range($Q$) 将其转化为二进制串，算法步骤如下：

(1)如果 (l_1,h_1) 包含在[0,0.5]或(0.5,1)中，则直接令二进制串第 1 位为 1，否则将 (l_1,h_1) 分裂为 $(l_1,0.5]$ 和 $(0.5,h_1)$ 2 个部分，并产生对应的 2 个第 1 位分别为 0,1 的二进制串。

(2)对于步骤(1)产生的每个区间对应的二进制串，如果 (l_2,h_2) 包含在[0,0.5]或(0.5,1)中，则直接令二进制串的第 2 位为 1，否则将 (l_2,h_2) 分裂为 $(l_2,0.5]$ 和 $(0.5,h_2)$ 2 个部分，并产生 2 个第 1 位分别为 0,1 的二进制串。对 n 个属性重复分裂。

(3)对每个属性分裂后的值域空间重复步骤(1)、步骤(2)。

在每轮操作中，n 个属性的范围划分可能将 Q 分为 2^n 个分支，所有操作结束时最多可能得到的分支数为 $2^{k\times n}$ 个。

如图 2 所示，对二维范围数据 $[(0.6,0.7),(0.3,0.8)]$ 利用多维范围划分算法，第 1 次划分为 1，第 2 次为 11,10，第 3 次划分为 110,100，第 4 次划分为 1101,1100,1001，其中，第 1 次、第 3 次划分是针对第 1 个属性在不同值域范围上的分裂；第 2 次、第 4 次划分是针对第 2 个属性在不同值域范围上的分裂。

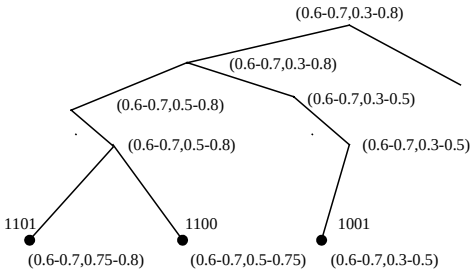

图 2　多维范围划分树

4　基于 Kademlia 的 P2P 多维范围查询算法

本文使用Kademlia作为K-net的overlay，$key(k_1k_2\cdots k_n)$高位是对多维数据的简单划分，位数越低对多维数据的划分越详细。对于$key1(k_1k_2\cdots k_n)$，$key2(k_1k_2\cdots k_n)$，两者的距离$d(x,y)=x\oplus y$，d越大多维数据的差别越大。

4.1　数据的发布

在 Kademlia 中，每个数据转换为一个 160 位的 *key*，存放<*key*,*value*>对数据的过程如下：发起者首先定位 k 个 ID 值最接近 *key* 的节点，然后对这 k 个节点发起 STORE 操作。执行 STORE 操作的 k 个节点每小时重发布自己所有的<*key*,*value*>对数据。为了保证数据发布、搜寻的一致性，规定在任何时刻，当节点 w 发现新节点 u 比 w 上的某些<*key*, *value*>对数据更接近时，w 把这些<*key*,*value*>对数据复制到 u 上，但不从 w 上删除。

由于 K-net 支持复杂查询，因此需要处理多维数据的发布和查询。当节点 s 要把数据 D 存放到 P2P 网络中时，先判断数据是否是多维范围的，如果是，则需要利用 $GetID(D)$ 将 D 转换为一维 *key*，然后通过路由机制找到 k 个 ID 值最接近 *key* 的节点，将数据 D 发布到这些节点上。转换后的数据可能只有前 $k\times n$ 位，剩余的 j 位作为冗余，可以任意取值，因此，最多可以将数据 D 发布到 2^j 个节点上，提高了查询成功率。

4.2　查询算法

对每个 $0\le i\le 160$，Kademlia中每个节点的K-buckets都保存有一些和自己距离范围在区间$[2^i,2^{i+1}]$内的一些节点信息，这些信息由一些(IP address, UDP port, Node ID)数据列表构成。节点 x 要查找多维范围数据 $Q=[(l_1,h_1),(l_2,h_2),\cdots,(l_n,h_n)]$，Kademlia按如下递归操作步骤进行路由查找：

(1)节点 X 对 Q 利用 Split-Query(Q) 进行一次递归划分，获得 n 位二进制串 t。

(2)计算到 t 的距离 $d(x,t)$，将值域减少一半并利用 Split-Range(Q) 对 Q 进行一次递归划分，得 n 位二进制串 t'，通过 Kademlia 的路由算法找到和 t 距离最近的α个节点。

(3)对每个距离最近的节点 y 计算 $d(y,tt')$ (tt' 表示 t 为二进制串的最高若干位，t' 为 t 后续的 n 位)，从自己的 K-bucket 第$\lfloor \lg(d)\rfloor$个 k 桶中取出α个节点的信息，继续寻找和 tt' 距离最近的节点。

(4)对每个距离最近的节点令 $t=tt'$，重复步骤(2)、步骤(3)，直到 Q 划分完毕。

(5)最后的节点返回自己存储资源的 IP 地址给 x，x 直接从该地址获得资源。

ID 值为 t 的节点不一定存在于网络中，即 t 可能没有被分配给任何一台电脑。本文的α是为系统优化而设立的一个参数，表示同时进行查询的并发数目。图 3 给出了查询二维范围数据 $[(0.6,0.7),(0.3,0.8)]$ 的过程。

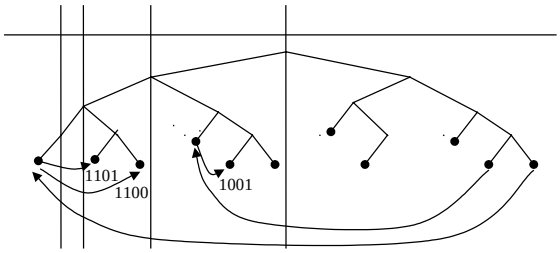

图 3　查询过程示例

5　实验结果及分析

笔者利用MIT的P2PSIM作为模拟工具，采用PVC[5]作为评价的理论基础。实验拓扑通过King[6]获得。对于给定维数，查询范围在$[0,1]$间自动生成。

使用 1 024 个节点的拓扑结构来测试数据维度对多维范围查询系统的影响。在相同实验环境中比较二维数据、三维数据在系统中的不同性能，并在该实验环境中对 Kademlia 进行模拟，可以把 Kademlia 作为一维优化结果。

实验结果如图 4、图 5 所示。由图 4 可以看出，根据 PVC 理论，二维数据、三维数据和 Kademlia 之间的面积很小，说明多维范围查询和原始 Kademlia 的系统性能相差不大。在相同延迟时间下，二维范围查找、三维范围查找需要发送的信息多于 Kademlia，这是由于多维范围数据可能的聚类数增加，使得所需流量增加。当流量超过 36 Byte/(node·s)时，系统可以进入稳定状态，此时增加流量对系统性能的提高作用很小。当系统流量达到 32 Byte/(node·s)时，系统处于最佳状态。由图 5 可以看出，维度对查询失败率的影响很小。多维范围查找的失败率略小于 Kademlia，这是因为多维范围查找是查找给定范围内一定数目的节点，因此，符合条件的节点数大于 Kademlia 精确查找，失败率较小。

(下转第 33 页)