内容寻址网络中几种负载均衡优化方法

熊继平 齐庆虎 洪佩琳 李津生
(中国科学技术大学电子工程与信息科学系 合肥 230027)

摘 要 内容寻址网络(Content Addressable Network，CAN)是 P2P 的一种，它利用分布式散列(hash)表(DHT)实现了文件信息和存放位置的有效映射，具有完全自组织和分布式的结构，并且有良好的可扩展性和容错性。但对 CAN 在负载均衡方面存在的问题并未提出有效的解决方法。该文首先介绍了内容寻址网络的基本工作原理，然后提出了几种有效的负载均衡优化方法：空间均衡划分、文件密度划分。最后通过仿真验证了这些方法的有效性。

关键词 P2P，内容寻址网络，分布式哈希表，负载均衡，文件密度分布
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2006)08-1488-04

A Few Optimized Load Balancing Methods of Content Addressable Network

Xiong Ji-ping Qi Qing-hu Hong Pei-lin Li Jin-sheng
(*Dept. of Electronic Engineering and Information Science, USTC, Hefei* 230027, *China*)

Abstract The Content Addressable Network (CAN) is a sort of P2P overlay network. CAN realizes the efficient mapping of the file information and its storage location by using Distributed Hash Table(DHT). CAN is scalable, fault-tolerant and completely self-organizing. In this paper an introduction to the basic architecture and the principle of CAN is given first. And then some methods of CAN's load balancing are proposed: such as largest area based uniform partitioning, uniform distributing of the keys. Finally, these methods are proved that they are effective by simulation.

Key words Peer-to-Peer, Content addressable network, Distributed hash table, Load balancing, File density distribution

1 引言

目前互联网主要的通信模式是客户/服务器方式(简称 C/S 方式)。但是随着网络规模的扩大，引发了一些问题：服务器的负担太重，难以管理大量的客户机，系统的性能容易变坏；当流量增加时，容易在服务器处产生瓶颈；在服务器端对应用程序的微小修改，都有可能导致所有客户端程序的重新安装；并且只有少数服务器得到有效利用，而大多数机器上的资源都被浪费了。

于是，出现了一种不同于 C/S 的模式，称之为 Peer-to-Peer 模式(简称 P2P)，也就是对等模式。P2P 中各个节点是逻辑对等的，也就是在 P2P 计算模型中不再区别服务器以及客户端，系统中的各个节点之间可以直接进行数据通信而不需要通过中间服务器。P2P 模式的对等性弱化了服务器甚至取消了服务器，使网络用户很容易加入系统中。每一个对等体可以充分利用其它对等体的信息资源、处理器周期、高速缓存和磁盘空间；用户直接输入要存取的信息而不是地址；信息的存储和发布随意，不需要集中管理。

P2P最基本的问题是如何高效地查找到存放文件的节点，为此加州大学伯克利分校的研究人员提出了内容寻址网络(CAN)[1-3]。CAN利用分布式哈希表(DHT)[1,4,5]实现了文件信息和存放位置的有效映射，具有完全自组织和分布式的结构，并且有良好的可扩展性和容错性。

负载均衡是 P2P 系统设计时需要考虑的公平性问题之一。在 CAN 中，节点的负载表现为其维护的空间大小。本文首先对 CAN 的空间划分方法提出了一种优化措施：最大面积空间均衡划分，即节点的加入请求由离目的节点较近的邻居中选择拥有区域面积最大的节点进行空间划分，而不是简单地对目的节点进行划分。更进一步，在实际应用中，节点空间中实际维护的是文件信息，因此，引入了文件密度的概念，并在此基础上给出了一种负载均衡的划分方法。最后，本文给出的仿真结果证明了这些优化措施的有效性。

2 CAN的基本原理[1]

CAN 的设计运用虚拟的 d 维笛卡尔坐标空间。在任何时候，在整个坐标空间对所有节点间动态划分，每个节点拥有整个空间中属于自己的区域。图 1 示出了一个 2 维的 X，Y 轴坐标范围为[0，1]区间的坐标空间，记为([0，1],[0，1])(使用归一化算法可使表征节点内容的关键词的散列值(hashed value)落在该区域)，该坐标空间被 5 个节点所划分。这些节点自组织成一个代表这个虚拟坐标空间的重叠网络(overlay network)。每个节点把坐标空间中与自己的区域相邻的节点作为邻居，并把这些邻居的信息存入自身的路由表。通过这种方法，CAN 中的任意两点间可以利用坐标进行寻路。下面简要介绍 CAN 的基本工作原理。

2.1 信息的插入和获取

坐标空间中存储的是(Key,Value)对，其中 Key 为表征文

2004-12-20收到, 2005-04-18改回
国家自然科学基金(90104011)资助课题

件特征的关键词(例如，文件名)，Value 为与存储文件的主机相关的值(例如，IP 地址)。插入(Key，Value)对的方法如下：先把 Key 用散列函数(hash funchion)映射到空间中的某一点 P，(Key,Value)对就被存储在拥有 P 所在区域的节点上；要根据 Key 获取相应的 Value 时，节点按特定的规则用相同的散列函数通过 Key 找到对应的 P，然后从拥有 P 的节点即可获得 Value。如果查询请求经过的节点不拥有 P，请求将被转发，直至到达 P 所在的节点。减少寻路跳数，提高路由的有效性是 CAN 中另一个重要研究方向。

如图 2 所示，节点 A 发出对 $k1$ 的查找请求，用散列函数对 $k1$ 进行散列运算，得到存储 $k1$ 的点 $P(x, y)$。节点 A 所在的区域不包含点 P，节点 A 的所有的邻居节点也不包含点 P，则传递请求，直到 B 点，B 所在的区间包含点 P，也就是 P 点存储了关于关键字 $K1$ 的信息。然后，A 节点获得 B 节点的 IP 地址后，两者可以直接通信，A 就可获得想要的关于 $K1$ 的信息。

一般而言，在两个节点之间存在不同的路径。如图 2 中，A 节点向 B 节点寻路，既可以沿着实线所示的路径，也可以沿着点划线所示的路径，也可以沿着其他的路径。由于有不同的路径存在，如果一个节点的一个或者更多的邻居节点失效后，它可以自动选择其它的路径传递；但是如果一个节点失去了所有的邻居节点，则传送可能会暂时失败。

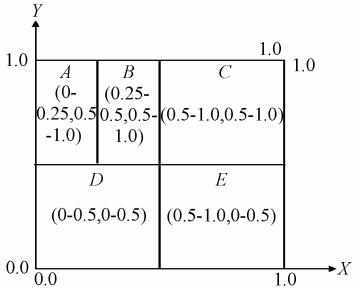

图 1 由二维坐标空间构成的 CAN
Fig.1 2-d coordinate space of CAN

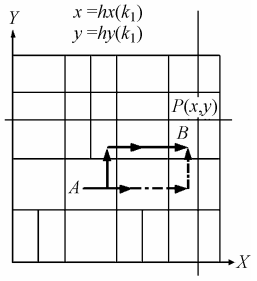

图 2　CAN 中的寻路过程
Fig.2 Routing process in CAN

2.2 节点的加入

在 CAN 中，节点的加入过程分为 3 个步骤：(1) 新加入的节点在 CAN 中查找已经存在的节点，查找的算法同于信息的获取算法，只是此时所用的关键字为节点信息的散列值；(2) 找到可以划分空间的节点，将此节点的空间的一半划分为新加入的节点；(3) 通知该节点的邻居节点进行路由表更新。

从仿真结果可以观察到，CAN 的节点加入策略并不能达到较好的空间均衡划分，因此我们提出了相应的改进优化方案，使得负载能够得到较为有效的均衡。

3　几种有效的负载均衡方法

3.1 空间均衡划分

现有文献[1–3]指出 d 维坐标空间中每个节点平均维护 $2d$ 个邻居节点。在我们提出的空间均衡划分方法中，扩展了邻居节点的定义，将对角线节点纳入节点的邻居节点定义中。

对于二维坐标空间，假设节点 A 拥有的区域为([$x1$, $x2$], [$y1$, $y2$])，([$x1$, $x2$]表示区域在 X 轴上的范围，[$y1$, $y2$]表示区域在 Y 轴上的范围)，节点 B 拥有的区域为([$x3$, $x4$],[$y3$, $y4$])。原来的邻居定义为如果满足以下条件之一则 B 是 A 的邻居节点：(1)[$x3$, $x4$] ⊆ [$x1$, $x2$], $y3$＝$y2$ 或[$x3$, $x4$] ⊆ [$x1$, $x2$], $y4$＝$y1$ ； (2)[$y3$, $y4$] ⊆ [$y1$, $y2$], $x3$=$x2$ 或[$y3$, $y4$] ⊆ [$y1$, $y2$], $x4$=$x1$ ； (3)[$x1$, $x2$] ⊆ [$x3$, $x4$] , $y3$＝$y2$ 或[$x1$, $x2$] ⊆ [$x3$, $x4$] , $y4$＝$y1$ ； (4)[$y1$, $y2$] ⊆ [$y3$, $y4$] , $x3$=$x2$。或[$y1$, $y2$] ⊆ [$y3$, $y4$] , $x4$=$x1$。由此可见，在空间完全均衡划分的情况下，一个节点最多可以有 8 个邻居节点。我们认为与一个节点逻辑上相连的节点都可以作为该节点的邻居，即除上述邻居节点外满足以下条件之一的节点 B 也是节点 A 的邻居节点：(1) $x4$= $x1$, $y3$=$y2$; (2) $x3$=$x2$, $y3$=$y2$; (3)$x3$= $x2$, $y4$= $y1$; (4)$x4$=$x1$, $y4$=$y1$。因此，在完全均衡划分的情况下二维坐标空间中一个节点最多可以有 12 个邻居节点。

在将对角线节点引入节点的邻居节点定义中后，下面给出按最大面积划分空间区域的负载均衡方法。当一个新节点加入系统时，加入系统的消息发给一些随机的没有失效的节点。已经存在的节点，不仅知道自己的空间坐标，还知道其邻居节点的空间坐标。这样，新加入的节点找到一个节点后，这个节点并不是直接划分自己的空间，而是把自己的空间与各个邻居节点的空间进行比较，然后选择一个面积最大(对二维空间而言)的节点的空间进行划分。这样就使划分获得更好的均衡性。<key，value>是使用均衡的散列函数分布在整个的坐标空间中，一个节点的区域面积也就表示了这个节点所要存储的资料<key，value>的大小，也就表示了这个节点的负载。因此，空间均衡的划分可以得到负载平衡的效果。

3.2 文件密度划分

CAN 中没有考虑 key 的实际存储情况，只假设每个节点上存储有管辖区域中任何值对应散列(key)值。在实际应用中，文件的分布情况往往会决定负载的多少。本文考虑 CAN 实际存储 key 集合及 key 分布的差异，每个文件有一个一一对应的 key，在虚平面上用坐标点来表示，点的坐标是由 key 计算出的散列值，如图 3 所示。

为了描述在一个节点的实际存储 key 负载情况，很自然

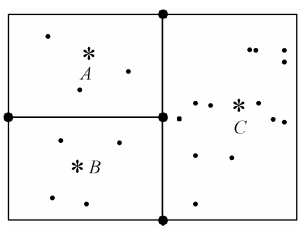

图 3　CAN 中 key 的分布
Fig.3　Distribution of key in CAN（·key，*文件重心）

地引入存储密度的概[1)]。如果文件的重要程度相当，则每个关键字的权重为 1，则有节点i的文件存储密度是：

$$\mathrm{FD}_i=\frac{\sum_j \mathrm{FW}_j^{\mathrm{key}}\delta(i,j)}{S_i}=\frac{N_i^{\mathrm{key}}}{S_i} \tag{1}$$

其中 $\delta(i,j)=\begin{cases}1, & \text{文件}j\text{存蓄在P上;}\\ 0, & \text{其他;}\end{cases}$ $\mathrm{FW}_j^{\mathrm{key}}$：由文件重要程度、优先权决定的文件$j$的权重；$N_i^{\mathrm{key}}$：$i$节点上存储的文件总数；$S_i$：节点$i$管辖的区域的面积。

引入了节点文件存储密度，就可以衡量节点存储负载。当新节点加入时，将综合考虑选目标节点的邻居节点中的空间面积大小、文件存储密度以及存储的文件总数划分区域，从而分担该节点的负载。这样达到的效果是节点的文件密度较为均衡，从而使系统中节点的负载达到均衡。实际上，如果认为文件 key 均匀分布在所有的节点上，即节点的文件密度都相同，这里提的负载均衡方法就与空间均衡划分的方法相同了。因此文件密度划分是在实际应用中对空间均衡划分的一种改进。

4 性能仿真和分析

本文的仿真是在用 Visual C++6.0 搭建的仿真平台上进行的。该仿真平台主要由 3 个类组成(如图 4)：P2P Simulator 是总的控制器，提供了用户与系统交互的界面；CAN Simulator 将 P2P Simulator 发出的命令传递给 CAN 网络，同时也执行部分的操作；CANNode 定义了 CAN 网络中基本参数，执行大部分具体操作。仿真的前提条件是：坐标空间为二维；所有节点随机产生；用归一化算法使表征节点内容的关键词的散列值落在([0，1]，[0，1])区域。

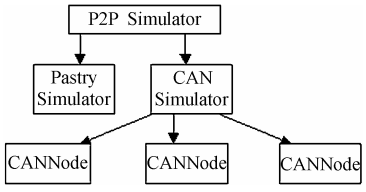

图 4 仿真平台基本结构

Fig.4 Basic structure of simulation plane

4.1 空间均衡划分的仿真

本文对二维坐标空间的情况进行了仿真。假设整个坐标空间的面积是 VT，整个空间中共有 n 个节点，那么完全均衡划分的结果应该是每个节点的面积都是 VT/n，这里用 V 表示 VT/n。随机产生 210 个节点，对于采用和不采用均衡划分两种情况，我们分别仿真 20 次，每次计算每个节点所获得的面积，然后统计其平均值。结果如图 5 所示。X 轴表示不同的面积，如：V，2V，Y 轴表示具有某一面积(如 V)的节点所占的百分比，从图中我们可以看出，在没有均衡划分的情况下，达到面积为 V 的节点的百分比约为 43%，而使用了均衡划分，达到面积为 V 的节点的百分比增加到了约 79%，得到了较好的改善。

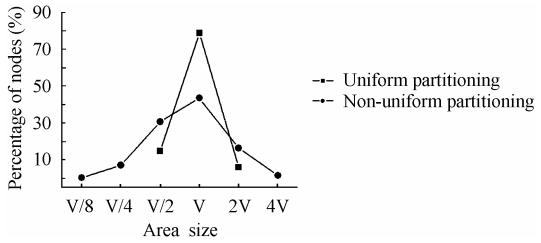

图 5 均衡划分与非均衡划分的比较

Fig.5 Comparison uniform partitioning and non-uniform partitioning

4.2 文件存储密度均衡的仿真

为了简化问题，不失一般性，假定整个坐标空间被 4 个节点A，B，C，D等分成 4 个区域，如图 6 所示，各区域中分别均匀的分布着 2^6,2^7,2^8,2^9个权重为 1 的文件。在仿真中，随机产生 26 个节点，进行非均衡划分和文件存储密度均衡划分的仿真。

图 7 给出了仿真结果。在理想的完全负载均衡的情况下，从区域 A 一直到 D 的节点个数分布比应该正比于区域内的文件总数比，亦即分别是 1:2:3:4，对应着图中的理想节点分布曲线。在采用非均衡划分的情况下，由于节点的加入是随机的，因此，每个区域的节点个数基本一致，不能反映各区域负载情况。而基于文件密度分布的划分使得节点分布靠近负载重的区域，如图中文件密度分布曲线所示，从而达到均衡负载高区域节点的效果。

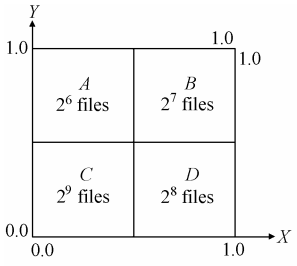

图 6 初始空间划分和文件分布

Fig.6 Initial space partitioning and file distribution

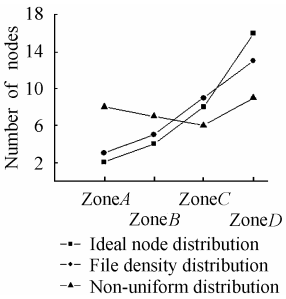

图 7 非均衡划分和文件存储密度均衡的对比

Fig.7 Comparison non-uniform partitioning and file density partitioning

5 结束语

本文是对 CAN 的负载均衡方面提出一些优化改进措施，提出了空间均衡划分、文件密度划分这两种方法，并用我们自己搭建的仿真平台进行了仿真。从讨论和仿真的结果来看，这些方法使系统的负载较为均衡，证实了方案的有效性。均衡划分的目的是在逻辑网络层面上进行节点的负载均衡，但实际网络往往是异构的，这就存在逻辑网络和物理网络不匹配的情况。因此，如何将负载平衡与节点异构性以及

[1)]实际上文件在CAN上是以文件的关键字（key）形式存储的，所以文件的存储密度就是key的存储密度。

网络结构异构性相结合来考虑是我们下一步的研究方向。

参 考 文 献

[1] Rathasamy S, Francis P, Handley M, *et al*.. A scalable content-addressable network. In ACM SIGCOMM’01, San Diego, CA, 2001, 31(4): 161−172.

[2] Rathasamy S. A scalable content-addressable network. A dissertation submitted in partial satisfaction of the requirements. for the degree of Doctor of Philosophy in Computer Science in the Graduate Division of the University of California at Berkeley, Fall 2002.

[3] Rathasamy S, Francis P, Handley M, *et al*.. A scalable content-addressable network. In ICSI Technical Report, Jan 2001.

[4] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large scale peer-to-peer systems. In Proceedings of the 18th IFIP/ACM International Conference on Distributed System Platforms, 2001: 329−350.

[5] Stoica I, Morris R, Karger D, Kaashoek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In ACM SIGCOMM’01, San Diego, CA, 2001, 31(4): 149−160.

[6] Tang Chunqiang, Xu Zhichen, Dwarkadas S. Peer-to-Peer information retrieval using self-organizing semantic overlay networks. In ACM SIGCOMM’03, Karlsruhe, Germany, 2003: 175−186.

熊继平：　男，1982年生，博士生，研究方向为对等网络技术、网络安全等.

齐庆虎：　男，1974年生，硕士生，研究方向为对等网络搜索、自组织网络等.

洪佩琳：　女，1961年生，博士生导师，研究方向为移动IPv6、QoS控制等.

李津生：　男，1967年生，博士生导师，研究方向为全光网络、OBS网络等.