多节点集群 P2P 系统研究

申新鹏 李战怀
(西北工业大学计算机科学学院 西安 710072)

摘 要 分布式结构化对等网络可以提供精确的资源发现,有着良好的可扩展性和自组织能力,成为对等计算研究的重点。但目前结构化对等计算的研究往往专注于如何提高路由效率,忽视了结构化算法存在的其他问题。提出了多节点集群 P2P 系统模型。系统中多个节点可以有相同的标识符,它们共同形成一个集群。集群中的节点管理相同的资源,从而可以有效避免节点突然失效造成的资源丢失。同时,资源的多个备份相互协作可以提高资源下载速度。本系统还采用自动搜集查询路径上的相关关键字的方法来实现模糊查询功能。模拟实验证明该方法是有效的。

关键词 对等计算,模糊查询,算法
中图法分类号 TP393 **文献标识码** A

Research on Multi-peers Cluster P2P System

SHEN Xin-peng LI Zhan-huai
(College of Computer Science & Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract Distributed structured peer-to-peer network can provide precise resource discovery and has a good scalability and self-organizing capabilities, so it has become the focus of the study. However, the research often focuses on how to improve routing efficiency and neglects of other serious problems. This paper proposed a multi-peers cluster P2P system. In the system, multiple peers could have same identifier. They form a cluster. All peers in a cluster save the same resource. So the resource was not lost even if one or more peers failed suddenly. The multiple copies of the resource could work together to effectively improve the download speed. Most of the structured peer-to-peer systems only provide the precise key query. This system could automatically collect related keywords on the query route. Using this method, the fuzzy query could be implemented according to related keywords. Experimental results show that this work is effective and efficient.

Keywords Peer-to-peer computering, Fuzzy query, Arithmetic

1 结构化对等计算存在的问题

2001年提出的基于分布式哈希表[1]的结构化发现算法极大地推动了对等计算的发展,从此对等计算进入了结构化的时代。结构化对等算法可以提供精确的资源发现,能够自适应节点的动态加入/退出,有着良好的可扩展性、鲁棒性、节点 ID 分配的均匀性和自组织能力。经典的案例包括 Tapestry[2],Chord[3]和 CAN[4]等。例如经典的 Chord 系统,通过使用 DHT 技术使得发现指定对象只需要维护 O(log N)长度的路由表。即使一个包含 10,000 个节点的 Chord 系统,发现指定对象的平均跳数(hops)也只有 6 左右,因此再进行路由算法的研究,试图继续提高路由效率,难度很大。但是经典的结构化算法存在其他一些难以克服的问题。

1)在结构化算法中,使用确定的算法将资源分配到合适的节点上,每个资源往往只有一个备份。如果保存资源的节点突然离开了系统,则可能会引起资源丢失。虽然可以使用路径缓存等方法增加备份数量,但结构化的精确定位方法降低了实际效果。

2)在结构化算法中,资源的下载速度很慢。每个资源只有一个备份,资源的下载完全是点对点传输。很多用户使用 ADSL 方式上网。ADSL 的下行速度很快,但上行速度很慢,再加上网络延迟,传输速度往往只能达到几 k 甚至更低。

3)在结构化算法中,查询算法是基于关键字的,只支持精确匹配。例如在系统中保存有名为"西北工业大学"的资源,如果使用关键字"大学"进行查询,则返回结果为空。必须使用"西北工业大学"进行查询,才能找到相应的资源。

为了综合解决这些问题,作者进行了一系列的探索,在分析现有的分布式结构化对等网研究成果的基础上,提出了一种全新的结构化 P2P 系统:多节点集群 P2P 系统(Multi-Peers Cluster P2P System,简称 MPCPS)。

2 多节点集群 P2P 系统简介

本节简单介绍一下多节点集群 P2P 系统的基本思想。本文提到的结构化 P2P 算法是一种泛指,可以是任何一种经

到稿日期:2009-03-17 返修日期:2009-05-26 本文受国家自然科学基金(60573096)资助。
申新鹏(1975—),男,博士生,讲师,主要研究方向为对等计算、存储区域网络、数据挖掘等,E-mail:shenxinpeng@nwpu.edu.cn;**李战怀**(1961—),男,教授,博士生导师,主要研究方向为存储区域网络、数据库系统等。

典的结构化 P2P 算法。下一节将使用 Chord 算法作为例子来实现一个具体的多节点集群 P2P 系统。使用其他结构化 P2P 算法的实现与此类似。

在经典的结构化 P2P 算法中，给每个节点分配一个唯一节点标识符(Node ID)，给每个资源对象分配一个唯一的资源标识符(Object ID)，并将资源存储在其节点 ID 与资源 ID 相等或者相近的节点上。需要查找该资源时，采用一定的路由算法将其定位到存储该资源的节点上。

多节点集群 P2P 系统的主要思想是将多个节点聚集到一起，形成一个集群。然后对集群按照一定的方式分配一个唯一的集群标识符(Cluster ID)，将资源存储在集群标识符 ID 与之相等或者相近的集群上。这样，资源同时存放在属于同一集群的多个节点上，提高了资源的安全性和下载速度。

一个集群选择一个节点，参与形成最终的结构化 P2P 系统，这个被选中的节点起到了类似网关的作用，被称为网关节点(Gateway Node)。这可以看作是所有的网关节点连接在一起形成一个结构化 P2P 系统，而集群的其他节点是附着在网关节点上的。同时，集群中其他节点与网关节点一样也保存系统的链表和邻居节点信息，所有节点保存的资源信息也是相同的。网关节点的特殊之处仅仅在于其他集群指向本集群的链表是指向网关节点而不是其他节点的，因此并没有超级节点的存在。当网关节点失效时，立即选择一个新的网关节点，并通知其他集群的节点更新信息。使用这种方法可以使得系统的开销不会因为使用集群代替节点而大量增加。

图 1 是一个示例。3 个集群的网关节点分别是节点 P_{11}，P_{20} 和 P_{31}，它们参与形成结构化 P2P 系统。其他节点不参与形成 P2P 系统，但也保存和网关节点相同的指针。注意所有的指针都是指向网关节点的。

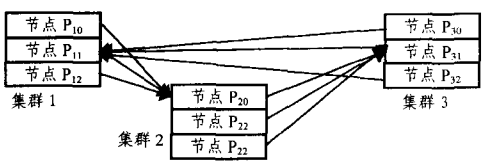

图 1　网关节点示意图

同一集群的节点简单地聚集在一起，形成一个小的系统。为了保证集群中的节点能够相互联系，而且系统开销最小，它们之间使用 Gnutella[5] 协议形成一个非结构化的 P2P 系统。集群中的节点定期相互通信，保证它们存储的资源和链表等信息的一致性。集群中的节点数量为几个到几十个之间，因此不会产生流量过大的情况。

当一个节点需要某个资源时，在结构化 P2P 系统中进行查找。找到资源所在的集群后，集群中所有的节点相互协作提供下载服务。集群中的节点数量较多并且都是随机分配的，因此虽然没有考虑节点地理位置的关系，但总能提供令人满意的下载速度(可以采用类似 BT 下载的分片下载方法进一步提高下载速度)。

经典的结构化 P2P 系统只支持基于关键字的精确匹配查询。在多节点集群 P2P 系统中实现了模糊查询的功能。每个节点划出一小块缓存，记录自己保存的资源的关键字。当某个节点要查询资源时，路由路径上的所有节点在转发查询请求的同时，将自己保存与查询的关键字部分匹配的关键字同时转发。最后这些部分匹配的关键字就聚集到了资源的后继节点上。后继节点在向请求发起节点返回资源的同时，将部分匹配的关键字也返回给请求发起节点。如果请求发起节点对得到的部分匹配的关键字感兴趣，就可以取得相应的资源。经过一段时间的运行之后，每个资源的后继节点都会记录很多相关的关键字，使用它们就可以部分实现模糊查询的功能。

这种方法不可能找到所有部分匹配的关键字，仅能部分实现模糊查询的功能。但是系统运行足够长的时间之后，可以找到大部分相关的资源。另外在 P2P 的环境下，要找到所有相关的资源本身也是不可能的。

图 2 是一个模糊查询的示意图。节点 P_0 要查询关键字为“大学”的资源，在路由的过程中经过了节点 P_1，P_2，P_3，到达资源的后继节点 P_4。其中 P_1 和 P_3 保存的资源的关键字与要查询的关键字“大学”部分匹配，因此 P_1 和 P_3 分别将它们的关键字随查询请求转发。P_2 保存的资源不能与查询关键字部分匹配，就不转发它的关键字。达到 P_4 后，P_4 将关键字为“大学”的资源返回给 P_0，同时将它已知的、能部分匹配的关键字记录到自己的缓存中，并转发给 P_0。P_0 收到相关的关键字后，判断如果需要相关的信息，就再次发出查询请求，获得对应的资源。

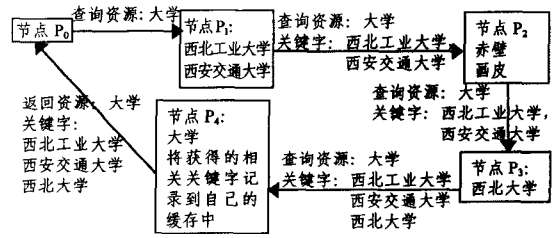

图 2　模糊查询示意图

这里仅仅转发相关的关键字而不转发 IP 地址，是因为在 P2P 系统中节点随时可能失效。如果节点 P_4 记录了资源关键字所在的节点，就要维护节点的正确性，从而增加了系统负担。如果不进行维护，记录节点也就没有意义。

从以上的描述可以看出，在多节点集群 P2P 系统中实现了模糊查询的功能，但没有额外增加系统开销。

3　多节点集群 P2P 系统的实现

本节在上一节的基础上，使用 Chord 算法作为例子来实现一个具体的多节点集群 P2P 系统。

3.1　节点集群

在 Chord 算法中，对节点的 IP 地址进行哈希运算，得到一个唯一的 m 位的节点标识符。每个节点标识符对应一个节点，任意两个节点的标识符都互不相同。节点标识符按顺时针方向从小到大依次排列，形成 Chord 标识符环。

在多节点集群 P2P 系统中，对节点的 IP 地址进行哈希运算，得到一个 m 位的节点标识符后，节点标识符相近的多个节点聚集在一起，形成一个集群，并取集群中第一个节点的标识符作为整个集群所有节点的标识符。即整个集群中所有节点的节点标识符相等，它们在 Chord 环中只占据一个位置。或者说，每个集群只选择一个节点参与形成 Chord 环。这个被选中的节点起到了将集群中其它节点联入 Chord 网的作用，称为网关节点，如图 3 所示。

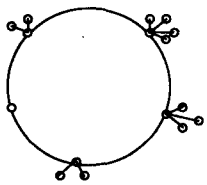

图 3　节点集群 Chord 环

集群中的多个节点具有相同的节点标识符，它们保存相同的路由表和邻居节点信息，并且保存的资源文件也是相同的。采用节点集群的目的是提高资源的安全性和下载速度。

集群中有几个节点，资源就会有几个备份。因此，即使某个节点突然失效，也不会引起资源的丢失。另一方面，多个备份同时提供下载服务，可以提高下载速度。影响下载速度的因素有很多，如网络条件、地理位置、系统工作负荷等。如果只有一个备份，很难保证提供稳定的下载服务。但如果同一个资源有多个备份，它们分布在不同的地理位置，处于不同的工作状态，并且协同工作，则可以提供稳定安全的下载服务。

使用过 BT[6]，eMule 等 P2P 下载工具的人都知道下载源越多，下载速度就越快、越稳定。因此集群中节点越多越好。但这也要考虑网络的规模和资源的数量。如果网络规模小，资源多，集群中的节点也很多，那么每个节点就必须缓存大量的资源。另一方面，要保证集群中节点数据的一致性，也需要一定的系统开销。如果集群中节点数量太多，增加的系统开销将抵消节点增加带来的好处。因此，在多节点集群 P2P 系统中，对集群中的节点数量规定了最大值（一般为 20 左右）。

由于 P2P 系统的动态性，不同的集群中节点的数量是不同的。但不论集群中有多少节点，必须保证它们保存的信息（资源和路由表）的一致性，需要使用一定的算法将它们结合在一起。又集群中的节点数量较少，而且各个节点地位相等，因此使用系统开销最小的 P2P 算法最为合适。本系统选用 Gnutella 算法来组织集群内部节点。Gnutella 算法不做具体介绍。节点定期向它的邻居节点发出查询请求、同步数据。

当有新的节点加入集群时，集群中已有的节点相互协作，将已有的资源和路由表传送到新的节点。当有新的资源加入集群时，就将该资源同时分散到各个节点中。

3.2　网关节点

一个集群中有多个节点，但只有一个节点直接参与 Chord 环的形成，这个节点起到了将集群中其它节点联入 Chord 网的作用，类似传统网络中的网关，故称为网关节点。其他节点不直接参与 Chord 环的形成，但它们也都保存与网关节点相同的路由表和邻居信息，随时可以替代网关节点，因此在多节点集群 P2P 系统中并没有超级节点的存在。

当原来的网关节点失效时，系统将从集群节点中选择一个节点作为新的网关节点。那么如何进行选择呢？

网关节点直接参与 Chord 环的形成，比其他节点承担更多的管理工作，例如修改路由表维持系统的稳定，参与资源的查找等。这些都要消耗节点的带宽和计算能力。因此，选择网关节点的依据主要是带宽和计算能力。本文使用节点能力值作为定量的工具来进行节点的选择。节点的能力值是上传带宽和计算能力的综合反映，使用如下公式进行计算：

$$V = A_b * B/B_s + A_c * C/C_s$$

式中，V 是节点的能力值，A_b 是带宽所占的权重，B 是节点提供的最大上传带宽，B_s 是系统设定的标准上传带宽，A_c 是计算能力所占的权重，C 是节点的计算能力，C_s 是系统设定的标准计算能力。

经过反复实验和选择，在具体实现中，带宽所占的权重 A_b 取 0.8，标准上传带宽 B_s 取 10kB/s，计算能力所占的权重 A_c 取 0.2，标准计算能力 C_s 取 1.0GHz，根据上式确定节点的能力值。当网关节点失效时，选择同一集群中能力值最大的节点作为新的网关节点。

3.3　新节点的加入

Chord 算法中新节点的加入以一个称为向导的已知节点协助，根据标识符找到后继节点，然后复制后继节点的路由表，并执行 Stabilize 操作来修改相关节点的路由信息。

在多节点集群 P2P 系统中，由于集群的引入，新节点的加入算法发生了变化。新节点 P_n 在向导的帮助下找到后继节点 P_s。如果 P_s 所在的集群中节点的数量小于规定的最大值，则修改新节点 P_n 的节点标识符，使之等于 P_s 的节点标识符，并将 P_n 加入该集群。如果集群中的节点数量已经大于或等于规定的最大值，则按照经典的 Chord 算法加入系统，建立一个新的集群，使节点 P_n 成为网关节点。

如果完全按照这种方法，则在系统的形成阶段（例如前 20 个节点）会造成节点不是形成 Chord 环，而是聚集在一起。为了防止这种情况的发生，规定当新节点加入后继节点的集群时，要先检查后继节点的 Finger 表。如果 Finger 表的最后两项是同一个节点，则新节点不加入后继节点的集群，而是按照经典的 Chord 算法加入系统，建立一个新的集群。Finger 的最后两项分别是 $(\mathrm{Id}+2^{m-1}) \bmod 2^m$ 和 $(\mathrm{Id}+2^{m-2}) \bmod 2^m$，这两个节点的标识符的差最大为 2^{m-2}。如果两者相等，说明系统中的节点（集群）数很少，应该先尽量增加集群数。

根据以上描述的算法，新节点加入系统有两种可能：(1) 加入后继节点的集群；(2) 直接加入 Chord 环，形成新的集群。两种情况下，新节点都要在向导的帮助下找到后继节点，算法复杂度为 $O(\log N)$。对于 Stabilize 操作，第一种情况没有建立新的集群，也没有改变集群的网关节点，不会引起 Stabilize 操作。第二种情况建立了新的集群，触发 Stabilize 操作，算法复杂度为 $O(\log^2 N)$。如果集群中节点的最大数为 K，则第二种情况发生的概率为 $1/K$。因此 Stabilize 操作总的算法复杂度为 $O(\log^2 N)/K$，比经典 Chord 算法的 $O(\log^2 N)$ 略有下降。

3.4　节点失效和退出

节点的失效是节点没有通知其他节点而突然离开网络，此时失效节点的前继保存的后继信息变得不可用，从而造成 Chord 环的断裂。

在多节点集群 P2P 系统中，每个集群中有多个节点，这些节点分为网关节点和普通节点。失效的节点如果不是网关节点，则不会对 P2P 系统产生任何影响，不需要进行特殊处理。如果失效的是网关节点，那么将造成 Chord 环的断裂。处理过程如下。

网关节点失效后，所在的集群首先根据节点的能力值选择新的网关节点，然后将新的网关节点通知它的前继和后继集群。集群中所有的节点都保存有相同的 Finger 表和邻居节点信息，因此新的网关节点也必然知道前继和后继集群的网关节点信息。根据这些信息可以方便地将新的网关节点的

IP地址等信息通知给前继和后继集群。对于其他节点的Finger表项中保存的失效节点的记录，同样需要多次Stabilize，把失效信息扩散到Chord网络中。

从以上描述可以看出，在多节点集群P2P系统中，节点失效的算法比经典的Chord算法更加简单。如果是失效的网关节点，可以非常方便地通知前继和后继集群。但因为通知其他节点的算法无法进行优化，因此算法复杂度和经典的Chord算法一样，均为$O(\log^2 N)$。如果失效的是普通节点，则不需要进行任何处理。

另一方面，经典的Chord算法中，资源只保存在资源标识符的后继节点上，后继节点的失效将导致资源的丢失。但是在多节点集群P2P系统中，资源同时保存在集群的多个节点中，单个节点的失效不会产生资源丢失的现象。

节点的退出是指节点在通知其邻居节点并等待其作出反应之后离开网络的行为。节点退出可以看作是节点失效的特例，因此可以采取相同的方法进行。

3.5 新资源的加入

经典的Chord算法中，新资源的加入比较简单。加入资源的节点使用哈希算法计算新资源的资源标识符，然后调用路由算法找到该资源的后继节点，最后将资源传送到后继节点即可。这里没有考虑后继节点的缓冲区已满的情况。

在多节点集群P2P系统中，新资源的加入同样首先要确定资源的标识符，然后调用路由算法找到该资源的后继集群(图4中集群2)。如果集群2的网关节点的缓冲区已满(缓冲区利用率达到80%以上)，并且该集群中有两个以上的节点，就将该集群一分为二。新的集群的标识符ID3取集群2标识符ID2和它的前继集群(集群1)的标识符ID1的中间值，并将一半的节点分配给新的集群。集群分裂后，两个集群保存的资源完全相同。但实际上，新集群只需要保存标识符大于ID1、小于ID3的资源，集群2只需要保存标识符大于ID3、小于ID2的资源。因此两个集群分别删除多余的资源，降低缓冲区的利用率。然后重新选择新资源的后继集群(如果新资源的标识符大于ID3，后继集群是集群2，否则是新集群)，并将新资源传送到后继集群中。最后系统在适当的时机调用Stabilize操作发现新的集群。

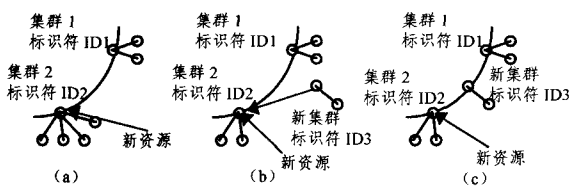

图4 新资源的加入引起集群分裂

如果新资源加入时后继集群的缓冲区没有满(缓冲区利用率小于80%)，则不进行集群的分裂，直接将新资源传送到后继集群即可。这样没有新集群的产生，因此也就不需要调用Stabilize操作。

根据以上描述的算法，新资源加入系统有两种可能：(1)后继集群缓冲区已满，引起集群分裂；(2)后继集群缓冲区没有满，直接插入资源。第一种情况查找后继集群的算法复杂度为$O(\log N)$。调用Stabilize操作，算法复杂度为$O(\log^2 N)$，但这种情况发生概率较小。第二种情况只需查找后继集群，算法复杂度为$O(\log N)$。相比经典Chord算法，复杂度略有增加，但考虑了缓冲区已满的情况。如果不考虑这种特殊情况，则与经典Chord算法的复杂度完全相同。

3.6 查询资源

经典的Chord算法是基于关键字查询的，无法实现模糊查询。即使对每个资源使用多个关键字进行保存，也还是只能根据指定的多个关键字进行查询。

在多节点集群P2P系统中，使用自动搜集关键字的方法实现了模糊查询的功能。在多节点集群P2P系统中，资源的存放和Chord系统一样是基于关键字的。首先使用分布式哈希算法根据资源的关键字得到资源的标识符，然后将资源存放到后继节点(集群)中。同时对每个节点增加一个缓冲区，用来记录相关的关键字。开始时，缓冲区里只记录了当前节点保存的资源的关键字。

当查询资源时，根据Chord路由算法，查询的过程中会经过一系列的中间节点。当经过这些节点时，检查它们的缓冲区，看是否有与要查询的关键字相关的关键字。如果有，则将相关的关键字随查询请求转发。如果经过的节点缓冲区中的关键字与要查询的关键字无关，则不转发该关键字(如图2所示)。当最后达到后继节点时，先检查并传递需要的资源。同时将查询路径上得到的相关关键字发送给查询请求节点，并把它们都保存到后继节点的缓冲区中。查询请求节点收到相关的关键字后，判断是否有需要的资源。如果有，则根据获得的新关键字取得需要的资源。

使用这种方法，每进行一次基于关键字的查询，都会将查询路径上的相关关键字搜集到资源的后继节点的缓冲区中。经过多次查询操作后，资源后继节点的缓冲区中将会记录很多相关的关键字，使用它们就可以部分实现模糊查询的功能。而且在多节点集群P2P系统中，每个节点位置都有多个节点，它们都同时保存了这些关键字信息。因此节点的退出和失效不会引起搜集到的关键字丢失。

同时，这种方法还能解决查询关键字错误的问题。例如电影“画皮”在系统中是以关键字“赵薇电影画皮”保存的。在Chord算法中，使用关键字“电影画皮”无法找到需要的资源。当在多节点集群P2P系统中，在“电影画皮”这个关键字的后继节点上，虽然没有相关的资源，但在它的缓冲区中会有关键字“赵薇电影画皮”，系统会把这个关键字返回给查询请求者。查询请求者可以使用这个关键字取得需要的资源。

从以上的描述可以看出，在多节点集群P2P系统中实现了模糊查询的功能，但没有额外增加系统开销。进行一次查询的算法复杂度仍然是$O(\log N)$。不过为了得到需要的资源，可能需要进行多次查询。

4 多节点集群P2P系统测试

通过以上对多节点集群P2P系统(MPCPS)的分析，发现该系统具有诸多优点，并且系统的算法复杂度还略有下降。为了对系统性能有更好的了解，采用了模拟实验的方法来进行验证。这里选用OverSim模拟器进行模拟实验。先在计算机(Pentium4 2.8G，1G内存)上安装Ubuntu操作系统，在其上安装OMNeT++和INET框架，最后安装OverSim，经过编译形成可执行文件，然后就可以进行模拟测试了。

OverSim自带有Chord的实现，在此基础上进行修改以

(下转第93页)

速度，精度更高。

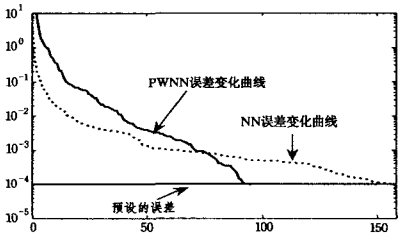

图 4　PWNN 训练与 WNN 训练对比

结束语　本文提出了一种基于改进的小波神经网络的信息安全风险评估方法，针对信息系统安全风险评估实践，构造出符合信息系统安全风险评估的模糊系统。针对信息系统安全风险评估的不确定性和当前评估手段的主观性，提出一种小波神经网络训练方法。针对小波神经网络中梯度下降法存在收敛速度慢和可能造成局部最优等问题，应用粒子群优化算法对小波神经网络进行训练。仿真结果表明，算法性能优良，为信息安全风险评估提供了可行的算法支持。

参 考 文 献

[1] David R, George G. Risk: A Practical Guide for Deciding What's Really Safe and What's Dangerous in the World Around You [D]. New York: Houghton Mifflin Company, 2002

[2] Maiwald E. Network Security: A Beginner's Guide [D]. The McGraw-Hill Companies, Inc, 2001

[3] Whitman M E, Herbert J. Principles of Information Security [M]. Canada: GEX Publishing Services, 2003

[4] ISO/IEC 17799. Information Technology-Code of practic for information security management[S]. 2000

[5] MnSCU. Security Risk Assessment - Applied Risk Management [R]. Minnesota State Colleges & Universities, 2002, 7

[6] GB/T 20984-2007《信息安全技术 信息系统的风险评估规范》[S]. 中华人民共和国国家标准, 2007

[7] 赵冬梅，马建峰，王跃生. 信息系统的模糊风险评估模型[J]. 通信学报, 2007, 28 (4): 51-56

[8] Zhao Dongmei, Wang Changguang, Ma Jianfeng. A risk assessment method of the wireless network security[J]. Journal of Electronics, 2007, 24(3): 428-432

[9] Zhang Q, Benvenise A. Wavelet network [J]. Proc IEEE Trans on Neural Network, 1992, 3: 889-898

[10] Delyon B, Juditsky A, Benveniste A. Accurancy analysis for wavelet approximations[J]. IEEE Trans. on Neural Network, 1995, 6(2): 332-348

[11] Kennedy J, Eberhart R C. Particle swam optimization [C] // Proc. IEEE Int'l Conf. on Neural Networks. Perth, Australia, 1995: 1942-1948

[12] Eberhart R C, Shi Y. Particle swarm optinization: developments, applications and resources[C]// Proc. Congress on Evolutionary Computation 2001. Piscataway, NJ: IEEEE Press, 2001: 81-86

(上接第 74 页)

实现多节点集群 P2P 系统，然后将两者进行对比实验。在实现的过程中，规定集群中的节点数最大为 20。同时为了简化实现，集群中的节点仅仅通过链表连接在一起，没有进行集群内的 Gnutella 实现。

在进行模拟测试时主要基于查询跳数和查询延迟进行查询开销的评估。查询跳数是进行一次基于关键字的查询时经过的节点的个数。而每经过一个节点时，都会产生一定的系统延迟，一次查询操作中每一跳查询延迟的总和就是总的查询延迟。

模拟测试时，从节点数量 256 开始依次增加，对比两个系统的查询跳数和查询延迟，结果如图 5 和图 6 所示。

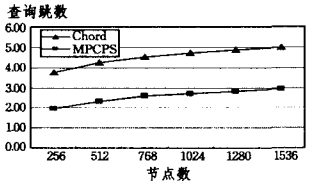

图 5　查询跳数模拟结果

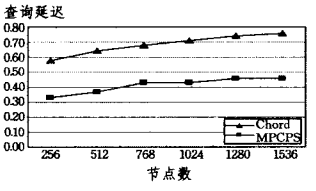

图 6　查询延迟模拟结果

模拟结果显示多节点集群 P2P 系统(MPCPS)的性能相比 Chord 系统有较大的提高。这主要是因为在节点相同的条件下，由于节点聚集，使得多节点集群 P2P 系统 Chord 环上的有效节点数大量减少。例如，当有 256 个节点时，Chord 系统的环上有 256 个节点，而多节点集群 P2P 系统的环上的有效节点数(集群数)为 13 左右。

5　多节点集群 P2P 系统应用

多节点集群 P2P 系统通过对经典的结构化 P2P 算法的改造，解决了结构化 P2P 算法中普遍存在的问题。主要体现在几个方面：(1)解决了节点失效造成资源丢失的问题；(2)提高了资源下载速度；(3)实现了模糊查询的功能，并且系统的性能还略有提高。因此多节点集群 P2P 系统是一个好的结构化 P2P 系统。

使用多节点集群的方法将资源同时存储在多个节点中，提高了系统资源的安全性和下载速度。再加上模糊查询功能的实现，使得多节点集群 P2P 系统非常适合作为 P2P 文件存储系统的底层实现。

参 考 文 献

[1] Karger D, Lehman E, Leighton F, et al. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web[C]// Proceedings of the 29th Annual ACM Symposium on Theory of Computing. 1997: 654-663

[2] Zhao Y, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing[R]. UCB/CSD-01-1141. Berkeley: University of California, 2000

[3] Stoica I, Morris R, Liben-Nowell D, et al. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications [J]. IEEE/ACM Transactions on Networking, 2003, 11(1): 17-32

[4] Ratnasamy S, Francis P, Handley M, et al. A scalable content addressable network[C]// Proc. ACM SIGCOMM. 2001

[5] Ripeanu M, Foster I, Iamnitchi A. Mapping the gnutella network: properties of large-scale peer-to-peer systems and implications for system design[J]. IEEE Internet Computing Journal Special Issue on Peer-to-Peer Networking, 2002, 6(1): 1-2

[6] Cohen B. Incentives Build Robustness in Bit Torrent[C]// Proceedings of the 1st Workshop on the Economics of Peer-to-Peer Systems. Berkeley, CA, USA, 2003