第 34 卷 第 7 期 计 算 机 工 程 2008 年 4 月
Vol.34 No.7 Computer Engineering April 2008
·多媒体技术及应用· 文章编号：1000—3428(2008)07—0232—03 文献标识码：A 中图分类号：TP301.6

CircleSearch：流媒体 VoD 环境下的 P2P 搜索算法

林青松，熊 焰，张伟超
(中国科技大学计算机系，合肥 230027)

摘 要：在基于 P2P 的流媒体 VoD 环境下，搜索机制是影响服务质量的关键因素之一。该文提出一种新的基于环状网络的分布式自适应概率搜索算法——CircleSearch。它依据节点本体内容生成结构化的环状拓扑网络，保证算法的可扩展性、健壮性和分布性。自发组织基于本体距离和连接可靠性的 small world 覆盖网络，减小消息的转发次数，采用自适应搜索算法提高文件定位的准确性。仿真实验表明，CircleSearch 以其高成功率、低服务器负载和很小的平均搜索长度实现媒体流传输的即时、稳定、可控和连续，为 VoD 服务提供 QoS 保证。
关键词：环状网络；small world 网络；最小距离

CircleSearch: P2P Search Algorithm in Streaming Video-on-Demand Environment

LIN Qing-song, XIONG Yan, ZHANG Wei-chao
(Computer Dept., University of Science and Technology of China, Hefei 230027)

【Abstract】 In the environment of streaming Video-on-Demand(VoD) over P2P, search scheme is one of the key factors affecting the Quality of Service(QoS). This article proposes a new self-adaptive probability based search algorithm over circle networks——CircleSearch, which creates circle networks based on the contents of nodes to guarantee the scalability, robustness and distribution of the algorithm. This algorithm spontaneously constructs the "small world" overlay network based on the distances and link reliabilities, which significantly decreases the number of transmission. It uses self-adaptive probability based search algorithm to increase the accuracy of file location. Simulation results show that CircleSearch can provide real-time, stable, controllable and continuous media streaming for system and guarantee the QoS of VoD services with small average path lengths, high success rates and low servers' loads.
【Key words】 circle network; small world network; minimum distance

1 概述

视频服务领域在流媒体技术中引入 P2P 思想，目的是通过利用普通节点的带宽、存储等资源为其他节点提供服务，将服务分散化，极大地降低服务器和网络的开销，同时实现系统的可扩展性。典型的应用就是目前互联网上广泛流行的流媒体直播服务。而基于 P2P 流媒体 VoD 技术与直播相比，能够发布更多数目的文件，可为用户提供异步服务。

基于 P2P 流媒体 VoD 技术在节点广泛分布、数量巨大、行为不可控、请求异步、计算能力和网络连接不均匀等复杂环境下面临如下挑战：(1)节点加入；(2)服务容错；(3)媒体流控制；(4)瞬间媒体内容切换。VoD 系统最大特点在于节点异步请求服务，系统要以较小的代价完成此功能并且迅速进入搜索算法稳定期。

本文提出一种流媒体VoD环境下基于环状拓扑网络的自适应搜索算法——CircleSearch，它的中心思想是采用BitTorrent[1]算法的分片策略，利用每个节点额外存储一段流媒体内容，并据此将网络拓扑优化成环状网，进而构建small world覆盖网络，使得查询请求能被转发给与目的节点存在最小距离的节点。该算法具有以下几个特点：(1)通过集中式控制协议，服务器保存每个节点的部分状态信息，节点可以在服务器的帮助下快速地加入子网；(2)数据合并和冗余请求策略可以提供高容错的服务和稳定、可控、连续的媒体流；(3)基于内容的环状网络保证了搜索的成功率和以较小的代价实现瞬间媒体内容切换；(4)基于small world覆盖网络的自适应概率搜索实现高效文件定位和节点负载均衡。

目前，已有的相关研究工作基本上都是基于应用层组播树来实现P2P机制的。P2Cast[2]借鉴了传统IP组播中Patching策略，每个Peer节点的Patching流从服务器或者Peer节点获取，但存在复杂性、拥塞控制、可靠性管理等方面的不足；DirectStream[3]采用节点数据缓存机制和中心控制协议，不足是服务器负载太大且系统容错性能差。总体来说，基于组播树系统最大的缺点是服务器维护组播树的代价很高，并且瞬间媒体内容切换和单点失效都易造成组播树的频繁重构。本文提出了借助BitTorrent算法分片思想，采用数据合并和冗余请求策略来克服以上不足，将基于内容的半结构化环状网络和改进的自适应概率搜索方法相结合，构建small world网络[4]，实现高效路由。

2 基于环状拓扑网络的分布式自适应搜索算法

基于 P2P 的流媒体技术充分利用节点的内容缓存为其他节点服务，直播系统中各个节点缓存内容相似度较大，所以很容易找到服务节点；而点播系统则相反，需要一个高效可靠的搜索方法。CircleSearch 通过对拓扑网络和转发机制的优

基金项目：国家自然科学基金资助项目(60673111)；安徽省自然科学基金资助项目(050420211)
作者简介：林青松(1981－)，男，硕士研究生，主研方向：分布式计算；熊 焰，教授、博士、博士生导师；张伟超，博士研究生
收稿日期：2007-04-23 **E-mail：**qslin@mail.ustc.edu.cn

化来减小转发次数，缩短平均最短搜索路径长度。

本文提出的算法中，服务器中的每个流媒体文件将被分为若干个段(piece)，每个段又被分为若干个块(block)，每个节点专门存储文件固定的一段于内存为其他节点服务，数据传输采用冗余请求、分块传送和数据合并合并策略。

已知ID为f的文件，路由服务器维护着一张名为f的路由表，描述了点播该文件的节点集的信息。假设f依照流媒体协议分为大小相当的Num_b块，每段由 2^r个块组成，则段数$Num_p=\lceil Num_b / 2^r \rceil$。

2.1 构造环状网络和覆盖网络

定义 1 假设流媒体文件共分为Num_p段，任意 2 个节点a, b，存储的数据段号分别为p_a, p_b，则a和b的距离是指如图 1 中p_a和p_b在Num_p个点构成的圆上的距离：

$$\Phi=\min(Num_p+p_a-p_b,\ Num_p+p_b-p_a,\ |p_a-p_b|) \quad (1)$$

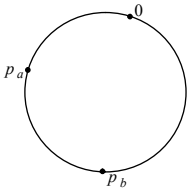

图 1　2 点在圆上的距离

定义 2 任意 2 个节点a, b，存储的数据段号分别为p_a, p_b。a和b距离为 0，称b和a互为同段节点；若同时b与a保持连接，则称b和a互为兄弟节点；a和b距离为 1 且$p_a<p_b$，称b为a的前段节点；若同时b与a保持连接，则称b为a的前驱节点；a和b距离为 1 且$p_a>p_b$，称b为a的后段节点；若同时b与a保持连接，则称b为a的后继节点；a和b距离为 1 且b与a保持连接，则称b为a的邻居节点。

定义 3 一个资源请求可以描述为一个六元组 $Que=(f,\ \lambda,\ b, s, n, y)$。其中，$f$ 指请求的流媒体文件的 ID；λ 指请求的段号；b 指请求的块在段中的序号；s 指发出请求的节点网络地址；n 是一个关于请求的全局唯一的标记符；y 是标志了是否有保存了 λ 的节点收到该消息的开关量。

定义 4 一个节点在服务器中保存的路由信息可以描述为一个五元组$Rec=(s, p, \delta, t,\ \theta_i)$。其中，$s$指节点网络地址；$p$是$s$保存数据的段号；$\delta\in\theta_i$是$s$到服务器的延迟，这在一定意义上反映了$s$与服务器在物理上的相对位置；实际的延迟时间空间是一个有限集$\theta=\bigcup_{i=1}^{m}\theta_i$，$\theta_i$是$\theta$的一个划分；$t$是$s$登录的时间。

定义5 任一节点 a 中保存的节点路由信息可描述为一个四元组 $Rou=(s,\ p,\ q,\ t)$。其中，s 指节点网络地址；p 是 s 保存数据的段号；q 是 s 的转发概率，它量化了节点间连通度，衡量节点间连接的可靠性和时间延迟；t 表示首次建立连接的时间。

A是点播文件f的节点网络，可以定义为$A=\{A_0,\ A_1,\cdots,\ A_{Nump}\}$，$A_i$是由保存了第$i$段数据的节点构成的子网，它们拓扑成如图 2 所示的一个环状网络。

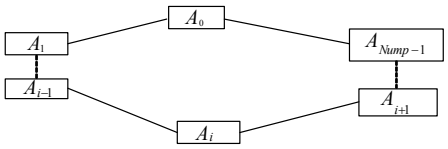

图 2　子网拓扑结构

A_i可以定义为$A_i=\{A_{i0},\ A_{i1},\cdots,\ A_{iq}\}$，$A_{ij}$分别与服务器和不超过$k$个前趋节点、兄弟节点、后继节点保持连接，并且与在搜索过程中通过学习建立的快捷节点建立虚拟连接，这样整个网络上就叠加了一层聚集度很高的small world网络。

当一节点a点播文件f时，路由服务器在表f中插入一项节点信息，延迟$\delta\in\theta_i$，依据以下算法确定要加入的子网号d：

```
d=-1;λ=1;    Succeed=0; signal=0;
While ((λ<= Num_p)or ! succeed) do { //以最高优先级加入到段号
//小的子网
if (|A_i|<q){ d=λ; succeed =1; break; }//以次优先级加入到大小不
//够的子网
if(! signal and (A_i的不存在节点延迟属于θ_i)){d=λ; signal=1;}
//以最低优先级加入到不存在 θ_i 区间的子网
λ++;}
if(d!=-1) d=random(Num_p);//随机插入到某一个子网
```

然后服务器返回给 a 其前段节点集、同段节点集和后段节点集中与 a 延迟空间相同节点。a 向这些节点发送 Hello 消息，收到问候的节点返回一个 Hello 消息，a 根据这些节点返回消息的延迟伟它们建立转发权重 $q=1/\sqrt{\delta}$，然后选取权重最大的节点分别生成 m 个作为前驱节点、后继节点和兄弟节点，最后向兄弟节点请求第 d 段数据并存储为其他节点服务。

2.2 自适应搜索算法

任意节点a都保存部分同段节点、前段节点、后段节点和子网$A_i(i\neq d\text{-}1,d,d\text{+}1)$中$m$个快捷节点(连通度较好的历史请求节点)的信息，并且保证与足够数量的兄弟节点、前趋节点和后继节点保持连接。如果a发现与自己连接的节点失效则告知服务器，服务器将从路由表中删除关于该节点的项，这样新加入的节点可以获得准确的路由信息。

a 每次总是将请求转发到与目的节点距离最小的节点直到找到一个目的节点，然后由该第 1 个目的节点转发 m-1 条消息给它的兄弟节点，期望由多路回复获得最好的时间延迟和容错性能。以下算法描述的 a 请求资源的过程里，选取转发节点总是依据权重 q 的大小。

(1)a 从路由表中离目标节点最近的节点集中选取一个节点作为转发节点，优先选择邻居节点，若转发成功，将该节点的转发权值增加：$q=(1+\alpha)\times q$, $(0<\alpha<1)$，转至(2)，否则将该节点的转发权值减小：$q=(1-\alpha)\times q$，转至(1)。

(2)收到请求的节点 c 执行以下动作：

1)判断该消息的 y 位是否为开，如果是则转至 3)，否则转至 2)。

2)判断请求的段号是否与自己存储的段号相同，如相同则将消息的 y 位置开，然后将数据传给 a，转至(3)；否则 c 从邻居节点中选取一个离目标最近的转发节点，将该节点的转发权值增加，转至(2)，若没有邻居节点直接转发到服务器。

3)直接将所请求的数据传给 a，转至(3)。

(3)a 收到来自 c 的数据后执行以下操作：

1)若 c 是 a 的前段节点，同段节点或后段节点则转至 2)，否则检查路由表，如果没有 c 的记录，则根据节点路由更新算法更新路由表，否则增加 c 的转发权重，转至 2)。

2)如果是第 1 次收到应答，则接收数据，转至 3)，否则丢弃。

3)如果 a 随机请求资源转至(1)，否则 a 向 c 请求下一个块数据，如果请求成功转至(2)，如果请求不成功则将 c 的转发权值减小，转至(1)。

如果 a 收到来自 c 的第 λ 段数据，时延为 δ，a 则利用以下算法更新路由表：

(1)若 c 是 a 的前段节点、同段节点或后段节点，转至(4)，否则转至(2)。

(2)检查路由表，若已存在 c 的路由信息则增加其转发权重，否则转至(3)。

(3)计算c的转发权重$q=1/\sqrt{\delta}$，若a的路由表中A_λ子网的快捷节点个数少于m，则直接插入c的记录，转至(4)，否则比较这m个快捷节点与c的权重，保留最大的m个作为快捷节点，转至(4)。

(4)若$x=\lambda-Num_p/2>0$，删除路由表中A_d子网的快捷节点的路由信息。

3 算法性能分析

3.1 系统模型理论分析

节点进入系统需先向服务器请求加入系统，再向各个子网A_i中的节点分阶段请求服务。假设进入系统的节点流是参数为μ的泊松流，那么可以将整个系统看成是多阶段循环排队系统。考虑到第 1 阶段(请求加入系统)服务时间极短，并且子网A_i中任意节点都可同时为多个客户异步提供服务，由上述假设知，请求服务的节点到达间隔时间和服务时间均服从负指数分布，所以可以近似认为第 1 阶段是等待队长为零的$M/M/1/\infty$排队系统，以后的各个阶段无限源的$M/M/\infty$排队系统。假设流媒体文件时长为T，共分成D段，则T时间后系统中将总是有μT个节点。网络状态稳定时，服务器只提供协助加入系统服务，负载非常小，此时需满足下式：

$$\Delta=\mu T-D>0$$

由上式可以得出，T 一定时，D 越小，μ 越大，则 Δ 越大。也就是说段长越大，流行度越高，服务器的负载越小。显然段长太大会给节点造成很大的负载压力，因此实际的系统则需在节点负载和服务器负载之间取得一个较好的妥协；值得庆幸的是，μ 越大，D 对 Δ 影响越小。

3.2 平均最短路径分析

实际的系统中，节点可能会执行随机搜索或随机地退出系统，因此该系统并非严格的循环排队系统。下面分析考虑系统中节点退出情况下的平均最短路径。

假设节点 a 保存的段号为 d，节点历史记录中的节点任何时候以平均概率 ρ 退出网络，即该节点失效。考虑首次请求资源的情况，此时路由历史记录为空，所以只能选择邻居节点转发。若请求的段号为 λ，则期望最短路径长度就是 d 与 λ 的距离，由式(1)得

$$\Phi_0(d,\lambda)=\min(D+d-\lambda, D+\lambda-d, |d-\lambda|) \quad (2)$$

通常情况下，$\lambda=0$，那么$\Phi_0(d,0)=\mathrm{Min}(D-d,d)$与$d$值密切相关。

接下来算法进入稳定期，a 顺序请求数据。若 a 请求第 λ 段μ块时，期望最短路径长度与μ值相关。

(1)$\mu\neq0$，a的历史记录中有m个A_λ中的节点，如果这m个节点全部失效则从路由表中选取最近节点转发。

1)$\lambda<d$。d 与 λ 的距离 $\varphi=\min(D+\lambda-d, d-\lambda)$。当 $\lambda>\varphi$ 时，

$$\Phi'_1(d,\lambda)=1\times(1-\rho^m)+2\times(1-\rho^m)\times\rho^m+\cdots+(\varphi-1)\times(1-\rho^m)\times\rho^{m\times(\varphi-2)}+\varphi\times\rho^{m\times(\varphi-1)}=1+\rho^m+\cdots+\rho^{m\times(\varphi-1)}=(1-\rho^{m\times\varphi})/(1-\rho^m) \quad (3)$$

当 $\lambda<\varphi$ 时，

$$\Phi''_1(d,\lambda)=1\times(1-\rho^m)+2\times(1-\rho^m)\times\rho^m+\cdots+\lambda\times(1-\rho^m)\times\rho^{m(\lambda-1)}+\varphi\times\rho^{m\times\lambda} \quad (4)$$

2) $\lambda<d$。d 与 λ 的距离 $\varphi=\min(D+d-\lambda, \lambda-d)$。

$$\Phi'''_1(d,\lambda)=1\times(1-\rho^m)+2\times(1-\rho^m)\times\rho^m+\cdots+(\varphi-1)\times(1-\rho^m)\times\rho^{m\times(\varphi-2)}+\varphi\times\rho^{m\times(\varphi-1)}=(1-\rho^{m\times\varphi})/(1-\rho^m) \quad (5)$$

综合分析以上 2 种情况，$\mu\neq0$ 时，d 与 λ 的距离 $\varphi=\min(D+d-\lambda, |\lambda-d|, D+\lambda-d)$，比较式(3)~式(5)可得期望最短路径长度：

$$\Phi_1(d,\lambda)\approx(1-\rho^{m\times\varphi})/(1-\rho^m) \quad (6)$$

$\mu\neq0$ 指请求的数据不是某段的首块，因此绝大多数的请求都属于此种情形，通常情况下约有$D\times(2^k-1)$个，假设节点任何时候皆以 1/2 的概率失效，也就是说$\rho=1/2$，$\Phi_1(d,\lambda)\in(1,2)$，当$m$较大时，$\Phi_1(d,\lambda)\to1$。

(2)$\mu=0$ 时，a的历史记录中没有A^λ中的节点，需要$A^{\lambda-1}$中的节点转发，由如(1)的类似分析可得期望最短路径长度：

$$\Phi_2(d,\lambda)=2\times(1-\rho^m)+3\times(1-\rho^m)\times\rho^m+\cdots+(\varphi-1)\times(1-\rho^m)\times\rho^{m\times(\varphi-3)}+\varphi\times\rho^{m\times(\varphi-2)}=1+(1-\rho^{m\times(\varphi-1)})/(1-\rho^m)<\Phi_1(d,\lambda)+1 \quad (7)$$

$\mu=0$ 对应非首段的首块请求，因此通常情况下有D-1 个请求属于此种情形，同样假设$\rho=1/2$ 并且m较大时，$\Phi_2(d,\lambda)\to2$。

由以上分析，a 的平均最短路径长度：

$$\overline{\Phi}(d,\lambda)=(\Phi_0(d,\lambda)+\Phi_1(d,\lambda)\times D\times(2^k-1)+(D-1)\times\Phi_2(d,\lambda))/(D\times2^k)$$

当 k 较大时，$\Phi(d,\lambda)\to1$，其大小与 d 无关。

分析路由算法可以发现，查询消息平均个数也非常少：

$$N=\overline{\Phi}(d,\lambda)+m-1$$

4 仿真实验

网络仿真环境是 NS2。实验中假设流媒体文件共 D 段，每段 d 片，时长 100 min。系统首先生成长度为 D 的环状网络和服务器节点，其他主机节点以强度为 σ(个/min)的泊松分布进入系统均匀分配到各个子网。节点每 6 s 请求 1 片数据，而其历史记录中的节点以 ρ=1/2 的概率失效，若找到存储该段数据的对应节点则视为查询成功，否则为失败，只能从服务器获取数据，所以说成功率 Succ 越大在一定程度上也体现了服务器的负载越小。

σ 越大也就是流行度越大，点播节目的人就越多，整个网络就越大；整个网络大小一定时，段长越小，子网也就越大；子网越大和 m 越大，则每个节点就可以拥有更多的邻居节点和有效快捷节点，从而可以提高查找成功率和减小平均最短路径长度。表 1 的测试数据给出了 σ, D 和 d 对成功率 *Succ* 和平均最短路径长度 *Ave Path Length* 的影响，并且可以看出 σ 越大，D 影响就越小，算法的平均性能越好。

表 1 不同条件下查询性能比较

σ(个/min)	D/段	d/片	m/个	*Succ*/(%)	*Ave Path Length*/步
2	50	20	2	90.2	1.28
2	100	10	3	86.4	1.24
4	50	20	3	99.2	1.16
4	100	10	5	98.2	1.09

5 结束语

本文给出了一种基于环状拓扑网络的自适应概率搜索算法，CircleSearch 通过拓扑网络优化和转化机制的改进实现了总是将请求消息转发给与目的节点距离最短的节点，并且很好地适应了 VoD 系统异步服务的特点和高度交互性的需求。理论分析和仿真试验表明，CircleSearch 平均最短路径接近 1，并以较低的服务器和网络代价为系统提供了即时、稳定、可

（下转第 237 页）