基于临近结点聚类构建层次化BitTorrent文件共享系统

薛广涛, 俞嘉地, 尤晋元
(上海交通大学计算机科学与工程系, 上海 200030)

摘 要: 为了提高BitTorrent对等网络系统的文件共享性能, 本文提出基于临近结点聚类方法, 将临近的结点聚合成结点簇, 同一结点簇中结点优先建立共享连接, 构建了层次化BitTorrent文件共享机制. 通过基于马尔可夫链的流体数学模型分析该系统性能, 证明了层次化结构的BitTorrent系统比原BitTorrent系统具有更好的文件共享性能. 模拟实验证实了理论分析结果, 并显示该系统有效地降低了中央服务器Tracker的负载, 提高了系统可扩展性和稳定性.

关键词: BT系统; 层次化结构; 超级结点
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2008) 02-0291-07

Building Hierarchical BitTorrent-like Peer-to-Peer File Sharing Systems Based on Proximity-Aware Peer Clustering

XUE Guang-tao, YU Jia-di, YOU Jin-yuan
(*Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai* 200030, *China*)

Abstract: In this paper, a hierarchical BitTorrent-like file sharing system is proposed to improve the performance of file sharing. Peers in such system are grouped into clusters according to their proximity in the underlying overlay network. A fluid model is developed to compare the proposed hierarchical BitTorrent-like system with the original BitTorrent system. With this model, we find that the hierarchical BitTorrent-like system improves the performance of file sharing quite effectively. Finally, the simulation results further prove that the hierarchical BitTorrent-like system achieves better scalability and efficiency while retaining the robustness and incentives of the original BitTorrent paradigm.

Key words: BitTorrent system; hierarchical architecture; super peer

1 引言

BitTorrent系统[1]是最流行的对等网络(P2P)文件共享系统之一[2], 已经引起人们广泛的兴趣[3,4]. 在BitTorrent系统中, 一个共享文件被分成多个等尺寸的文件块, 每个系统参与结点可以并行地从多个不同的结点下载同一个共享文件的不同文件块, 同时上载自己拥有的共享文件块给其他请求结点. 共享文件提供者首先创建一个被称为. torrent的元信息文件. 在这个文件中包含了文件的下载信息, 例如文件块大小、服务器IP地址等. 然后由文件提供者将. torrent文件发布到一个Web服务器上, 等待其他对此文件感兴趣的用户进行下载. 在BitTorrent系统中有三个组件: Tracker服务器、种子(Seed)和下载结点(Downloader). Tracker服务器是一个中央服务器, 它主要负责收集与统计系统中所有参与结点的状态, 并帮助参与结点互相发现并进行文件共享. 系统中所有结点(包括Tracker服务器、种子和下载结点)构成了一个被称为torrent的P2P网络[3].

然而现有的BitTorrent系统存在着一些缺点:

(1)结点连接没有优化. 在BitTorrent系统中, 由于每个参与结点是随机选择与其他结点建立连接进行文件共享, 使得结点有可能连接到与其拓扑距离较远、网络时延较长、连接带宽较小的结点, 增加了网络带宽的损耗和网络堵塞的几率, 影响文件共享效率.

(2)Tracker服务器存在瓶颈. BitTorrent系统中结点可以自由加入或离开系统, Tracker服务器需要不断地更新每个参与结点的状态. 当系统中参与结点数量较大时, 大量结点不断地离开和加入给Tracker服务器带来很大的负载.

为解决BitTorrent系统中存在的以上问题, 本文提

收稿日期: 2007-04-02; 修回日期: 2007-10-31
基金项目: 国家自然科学基金(No. 60673166, No. 60433040)

出了层次化 BitTorrent 文件共享系统. 通过将临近的结点聚合成结点簇, 每个结点簇中保持原有 BitTorrent 系统机制, 同一结点簇中结点优先建立共享连接, 实现文件本地共享最大化, 从而提高了系统文件共享性能. 为了构建此系统, 本文提出了结点临近性比较算法和基于结点临近性的分布式结点加入算法. 并通过使用基于马尔可夫链的流体数学模型(fluid model) 比较了层次化结构的 BitTorrent 系统与原 BitTorrent 系统的性能, 证明了层次化结构的 BitTorrent 系统能有效地提高文件共享性能. 最后通过模拟实验证实了理论分析结果, 并显示该系统有效地降低了中央服务器 Tracker 的负载, 提高了系统可扩展性和稳定性.

2 层次化结构的 BitTorrent 系统结构

2.1 系统结构

图 1 显示了层次化结构的 BitTorrent 系统基本结构, 所有系统参与结点依据临近关系被分为多个结点簇, 每个结点簇由一个超级结点管理和维护, 中央服务器 Tracker 管理着所有超级结点. 系统初始时仅包含一个结点簇, 该结点簇被称为基础结点簇. 基础结点簇中的超级结点即为 BitTorrent 系统的 Tracker 服务器, 它监控系统中所有参与结点的状态. 随着系统参与结点的增加, 基础结点簇将逐渐分裂为多个结点簇, 形成一个层次化结构的 BitTorrent 系统. 层次化结构的 BitTorrent 系统包括一个系统 Tracker 服务器和三类系统结点: 种子、下载结点和超级结点(Super-peer) . 系统 Tracker 服务器负责收集基础结点簇中所有结点的状态, 管理系统中所有超级结点; 种子结点是那些拥有共享文件所有文件块部分的系统参与结点; 下载结点是既可以提供文件块给其他结点去下载, 又要从其他结点下载所需文件块的结点; 超级结点的主要任务是作为其所在结点簇的本地 Tracker 服务器, 收集本地结点簇中所有参与结点的状态, 并为新加入本地结点簇中的结点分配一个本结点簇中的随机下载结点列表. 所有的超级结点(包括系统 Tracker 服务器) 互相连接形成了此覆盖网的主干网.

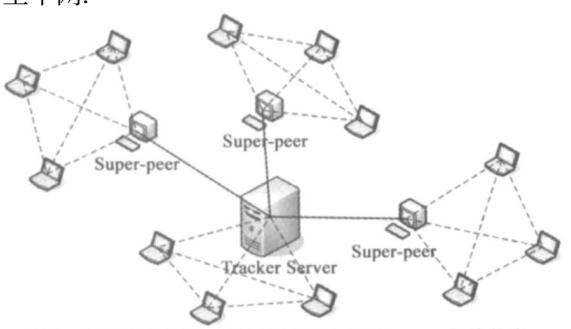

图 1 基于临近结点聚类的层次化 BitTorrent 文件共享系统的体系结构

将相互间网络时延较低, 连接带宽较大的结点聚合在同一个结点簇中, 结点与同一个结点簇中的结点建立上传/下载连接. 由于每个结点簇中的种子结点、下载结点和超级结点相互间具有较好的网络连接, 减少网络拥塞, 提高系统可扩展性.

系统优先从种子中选择结点作为超级结点, 因为种子有较多的网络服务资源和较长的在线时间, 并愿意提供服务给其他系统结点. 如果一个结点簇中没有种子, 也可以选择愿意充当超级结点的下载结点去作为超级结点, 这个下载结点可以优先地从其他的超级结点下载共享文件的文件块, 从而使其能快速地拥有共享文件的所有文件块, 成为种子去服务其结点簇中的其他下载结点. 因此保证了每一个结点簇都有完整的共享文件块.

此外系统为每个结点簇引入了一个备份结点, 周期地复制本地结点簇中超级结点的所有信息. 当超级结点突然离开, 备份结点将代替原超级结点成为本地结点簇中新的超级结点.

2.2 结点临近性比较

新加入系统的结点首先根据结点的临近性, 即结点间路由跳数、网络时延, 连接带宽等特征, 决定加入系统中哪个结点簇. 由于 P2P 覆盖网中有大量的结点, 结点在 P2P 覆盖网中要找寻与其最临近的结点是一个复杂的过程, 对每个结点遍历一遍显然是不实际也不经济的. 因此找到距其最近的超级结点, 并加入由此超级结点管理的结点簇, 是一个比较合理的、经济的方法.

在 P2P 覆盖网中, 两个结点的距离可以通过一个探测包在两主机结点间的网络时延和路由跳数来测量[5]. RTT(Round-Trip-Time) 和 TTL(Time-To-Live) 可以用来描述真实的 Internet 中两结点之间的距离. RTT 值指的是两主机之间的网络时延, TTL 值指的是两主机间的跳数. 通过在客户端运行 traceroute 命令可以跟踪发射包在 Internet 中的路由过程和时延, 并可从返回信息中得到两个主机之间的 RTT 值和 TTL 值.

由于网络中突发的网络堵塞, 两结点之间的 RTT 值在短时间内可能会发生很大的波动. 在层次化结构的 BitTorrent 系统中, 为了精确找到与结点最临近的超级结点, 我们以 RTT 值为主, TTL 值为辅, 将两者相结合作为结点间临近性测量标准. 我们设 $dist(p, q)$ 是结点 p 和结点 q 之间的距离; $dist_{RTT}(p; q)$ 代表结点 p 和结点 q 之间的 RTT 值; $dist_{TTL}(p, q)$ 代表结点 p 和结点 q 之间的TTL 值. 给定结点 p、q、s、t, 通过用基于 traceroute 的方法, 可以得到 $dist_{RTT}(p, q)$、$dist_{TTL}(p, q)$、$dist_{RTT}(s, t)$ 和 $dist_{TTL}(s, t)$ 值. 通过表 1 的网络结点临近性比较算法, 判定结点间的临近性.

表 1　结点临近性比较算法

Require: $dist_{RTT}(p,q)$、$dist_{TTL}(p,q)$、$dist_{RTT}(s,t)$、and $dist_{TTL}(s,t)$
1. If $|dist_{RTT}(p,q)-dist_{RTT}(s,t)|>d$ then
2.　If $dist_{RTT}(p,q)\geqslant dist_{RTT}(s,t)$ then
3.　　$dist(p,q)>dist(s,t)$
4.　Else
5.　　$dist(p,q)<dist(s,t)$
6. Elself $dist_{TTL}(p,q)\geqslant dist_{TTL}(s,t)$ then
7.　　$dist(p,q)>dist(s,t)$
8. Else
9.　　$dist(p,q)<dist(s,t)$

2.3　基于结点临近性的分布式结点加入算法

在层次化结构 BitTorrent 系统中, 由于系统中超级结点的数量远小于系统参与结点, 我们用超级结点代表结点簇的中心, 通过为新加入结点找到最临近的超级结点, 并加入该超级结点所在的结点簇, 保持了同一结点簇内的结点临近性, 有效降低了结点加入系统过程中的临近性测量开销. 由此我们提出了一个基于结点临近性的分布式结点加入算法, 使得结点加入临近的系统结点簇.

系统参数设置如下: 设 $H\{p_0, p_1, \cdots, p_{k-1}\}$ 是一个层次化结构的 BitTorrent 系统网络所有参与结点的集合, 其中 k 是集合 H 中结点的数量; 设 $P\{sp_0, sp_1, \cdots, sp_{m-1}\}$ 是层次化结构的 BitTorrent 系统网络中所有超级结点的集合, 其中 m 是集合 P 中超级结点的数量, 并且 $P\subseteq H$; 设 $SP_i\{sp_i, b_0, b_1, \cdots, b_{n-1}\}(i=0,1,\cdots,m-1)$ 是层次化结构的 BitTorrent 系统第 i 个结点簇中所有结点的集合, 其中 n 是结点簇 SP_i 中结点的数量, sp_i 是结点簇 SP_i 的超级结点, 我们有 $SP_i\subseteq H$ 并且 $H=SP_0\cup SP_1\cup\cdots\cup SP_{m-1}$; 设结点 u 是一个新请求加入系统网络的结点. 结点加入算法将返回一个超级结点 sp_j, 使得 sp_j 是与新请求加入结点 u 最为临近的超级结点, 即 s.t. $\forall q\in P, dist(u,sp_j)\leqslant dist(u,q)$. 然后结点 u 加入结点簇 SP_j.

表 2　基于结点临近性的分布式结点加入算法

Instance: 超级结点集合 P 和 $|P|=m$, 结点簇 $SP_i(i=0,1,\cdots,m-1)$, 新请求加入结点 u
1. 新请求加入结点 u 测量自己与集合 P 中每个超级结点的距离, 根据 RTT 值($dist_{RTT}(u,sp_i)$) 和 TTL 值($dist_{TTL}(u,sp_i)$) 得到距离信息 $dist(u,sp_i)(i=0,1,\cdots,m-1)$
2. 从集合 P 中找到一个超级结点 sp_j, 使得 sp_j 是结点 u 最为临近的超级结点, 即 s.t.
3. 结点 u 加入超级结点 sp_j 维护的结点簇 SP_j, 即 $SP_j=SP_j\cup\{u\}$

3　模型与分析

Internet 是由各种不同的路由域(routing domains) 相互连接而形成的[6]. 在 Internet 中每个路由域可以被分为主干域(transit domain) 和子域(stub domain)[7]. 子域是由局域网(local area networks, LANs) 或自治网络(autonomous systems, ASes) 构成. 主干域连接子域后形成主干网络(network backbone). 主干域连接了一系列主干结点, 称之为核心路由器(core routers), 每个路由器作为一个或多个子域的网关(gateway). 图 2 显示了以上的 Internet 网络描述.

在原 BitTorrent 系统中, 每个系统参与结点向系统 Tracker 服务器请求一个参与结点列表, 用于构建上载/下载结点集合. 系统 Tracker 服务器将返回一个随机结点列表(一般包括 50 个系统参与结点) 给请求结点. 由于这组结点是被 Tracker 服务器随机选择的, 因此这些结点可能分散在不同的子域中.

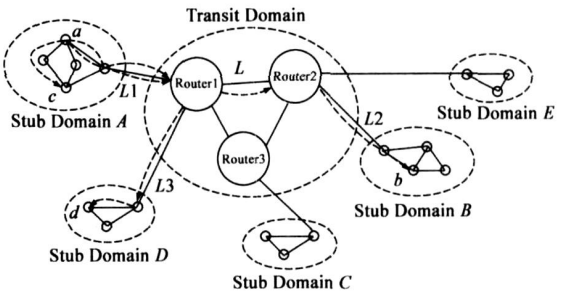

图 2　Internet 网络域的结构

图 2 中, 如果在子域 A 中一个结点 a 从子域 B 中的结点 b 下载共享文件, 那么整个下载的路由是 a-Router1-Router2-b, 结点 a 的下载带宽 D_{ab} 将由链路 $L1$, L 和 $L2$ 分配给结点 a、b 之间带宽所决定. 假设 transmit-transmit 连接分配给 a-b 的带宽为 W_L; stub-transmit 连接 $L1$ 分配给 a-b 的带宽为 W_{L1}; stub-transmit 连接 $L2$ 分配给 a-b 的带宽为 W_{L2}, 则有 $D_{ab}=\min\{W_L, W_{L1}, W_{L2}\}$. 实际的网络环境中, 由于主干网连接太多的子域, 大量结点间连接要通过主干网, 所以分配到一个具体的网络连接带宽就变得非常小, 通常有 $W_L<W_{L1}$ 和 $W_L<W_{L2}$. 因此结点 a、b 间的下载带宽 D_{ab} 为 W_L. 然而, 如果子域 A 中的结点 c 有与结点 b 相同的共享文件(此文件是结点 a 希望得到的), 结点 a 可以与结点 c 建立直接连接下载共享文件. 结点 a 的下载带宽 D_{ac} 将由结点 a 的下载带宽或结点 c 的上载带宽所决定. 在实际的网络环境中, 同一子域中结点 a、c 间的下载带宽 D_{ac} 要大于不同子域中结点 a、b 间的下载带宽 D_{ab}, 即结点 a 从相同子域结点 c 下载文件一般要快于从跨域结点 b 下载文件.

图 2 中, 如果结点 a 下载共享文件从子域 D 中的结点 d(子域 D 和子域 A 是相临子域), 那么下载路由为 a-Router1-c. 结点 a 的下载带宽 D_{ad} 将由 stub-transmit 连接 $L1$ 和 $L3$ 决定, 即 $D_{ad}=\min\{W_{L1}, W_{L3}\}$. 在实际的网络环境中, 有 $D_{ab}\leqslant D_{ad}\leqslant D_{ac}$, 即结点 a 从结点 d 下载文

件一般要快于从结点 b 下载文件, 慢于从结点 c 下载文件. 从以上分析可知同一子域或相近子域之间的结点相互建立连接去交换文件, 能有效地提高 BitTorrent 系统的性能. 在上述分析中, 我们忽略了结点在路由器上的等待延时和连接传播延时, 主要是因为在整个文件下载时延中, 它们通常占的比率相对较小.

我们使用一个基于马尔可夫链的流体数学模型去分析层次化结构的 BitTorrent 系统性能. 在此模型中, 系统参与结点被分为两个独立的结点簇: $C1$ 和 $C2$. 假设系统中所有的结点都是同构的, 即具有相同的服务能力, 并假设同一子域结点被组织在同一个结点簇中, 设同一结点簇中任意两结点之间的连接由结点本身的服务能力决定, 即有相同的上载/下载带宽(设为 μ). 而在不同结点簇中的两个结点的连接带宽将由子域间的连接带宽(设为 c) 决定. 为了简化模型, 假设在系统稳定状态下, 下载结点完成其下载工作就离开系统, 即在系统中没有种子结点. 因此系统中有两个状态: $C1$ 中结点下载状态和 $C2$ 中结点下载状态. 我们可以分别得到 BitTorrent 系统和层次化结构的 BitTorrent 系统有关这两个状态的马尔可夫描述. 模型中设定的参数如下: $x_1(t)$ 是在 t 时刻结点簇 $C1$ 中的结点数量; $x_2(t)$ 是在 t 时刻结点簇 $C2$ 中的结点数量; λ 是新结点的到达率; n 是一个结点并行上载连接数; η 是指结点的文件共享效力, 它已经被证明是非常接近于 1[3].

3.1 BitTorrent 系统模型

假设在 BitTorrent 系统中每个结点选择 n 个请求下载结点为其提供上载服务, 对方结点选择是随机的, 与其所在位置无关. 设任意两个位于同一结点簇中结点的连接带宽为, 任意两个位于不同结点簇中的结点连接带宽为 c. 在时间 t, 结点簇 $C1$ 中一个结点的上载率期望是:

$$E[Dc_1(t)] = \sum_{k=0}^{n} C_n^k \left(\frac{x_1(t)}{x_1(t)+x_2(t)}\right)^k \left(1-\frac{x_1(t)}{x_1(t)+x_2(t)}\right)^{n-k} [k\mu+(n-k)c] = n\mu\frac{x_1(t)}{x_1(t)+x_2(t)} + nc\frac{x_2(t)}{x_1(t)+x_2(t)} \quad (1)$$

在时间 t, 结点簇 $C2$ 中一个结点的上载率期望是:

$$E[Dc_2(t)] = n\mu\frac{x_2(t)}{x_1(t)+x_2(t)} + nc\frac{x_1(t)}{x_1(t)+x_2(t)} \quad (2)$$

模型中新结点到达过程符合泊松分布[3], 假设新结点以速率 λ 分别流入 $C1$ 状态和 $C2$ 状态. 在时间 t, 结点簇 $C1$ 中所有结点的上载率期望是 $E[Dc_1(t)]\cdot\eta x_1(t)$, 结点簇 $C2$ 中所有结点的上载率期望是 $E[Dc_2(t)]\eta x_2(t)$. 因此, BitTorrent 系统中结点簇 $C1$、$C2$ 中结点数量变化率可以用以下的方程组表示:

$$\begin{aligned}\frac{dx_1(t)}{dt} &= \lambda - E[Dc_1(t)]\eta x_1(t)\\ \frac{dx_2(t)}{dt} &= \lambda - E[Dc_2(t)]\eta x_2(t)\end{aligned} \quad (3)$$

方程组(3) 描述了 BitTorrent 系统两个状态的动态进程.

为了研究稳定状态下系统的性能, 假设: $\lim_{t\to\infty} x_1(t)$ 和 $\lim_{t\to\infty} x_2(t)$ 存在, 即 $\lim_{t\to\infty} x_1(t)=\bar{x}_1$, $\lim_{t\to\infty} x_2(t)=\bar{x}_2$, 其中 $\bar{x}_1$ 和 $\bar{x}_2$, 分别是 $x_1(t)$ 和 $x_2(t)$ 的平衡值. 在稳定状态下(即 $t\to\infty$), 我们有 $\frac{dx_1(t)}{dt}=\frac{dx_2(t)}{dt}=0$. 因此, 可以得到稳定状态下, 系统稳态的方程组:

$$\begin{aligned}0 &= \lambda - \left[n\mu\frac{\bar{x}_1}{\bar{x}_1+\bar{x}_2} + nc\frac{\bar{x}_2}{\bar{x}_1+\bar{x}_2}\right]\eta\bar{x}_1\\ 0 &= \lambda - \left[n\mu\frac{\bar{x}_2}{\bar{x}_1+\bar{x}_2} + nc\frac{\bar{x}_1}{\bar{x}_1+\bar{x}_2}\right]\eta\bar{x}_2\end{aligned} \quad (4)$$

解方程组(4), 得:

$$\bar{x}_1=\bar{x}_2=\frac{2\lambda}{n\eta(\mu+c)} \quad (5)$$

在式(5)中, 我们知道方程组(4) 有唯一的解, 系统存在一个平衡点 $(\bar{x}_1,\bar{x}_2)$.

Little's law[3,8] 被用于估计每个系统参与结点在系统稳定状态时的平均下载时间. 相似地, 在该模型中, 结点簇 $C1$、$C2$ 中结点的平均下载时间分别是 $T_1=\frac{\bar{x}_1}{\lambda}$ 和 $T_2=\frac{\bar{x}_2}{\lambda}$. 如果一个系统参与结点刚完成自己的下载工作, 该结点是结点簇 $C1$ 中结点的概率为 $\frac{\bar{x}_1}{\bar{x}_1+\bar{x}_2}$, 是结点簇 $C2$ 中结点的概率为 $\frac{\bar{x}_2}{\bar{x}_1+\bar{x}_2}$. 因此, BitTorrent 系统中所有结点平均下载时间是 $T_{BT}=\left(\frac{\bar{x}_1}{\bar{x}_1+\bar{x}_2}\right)T_1+\left(\frac{\bar{x}_2}{\bar{x}_1+\bar{x}_2}\right)T_2$, 基于表达式(5), 有:

$$T_1=T_2=\frac{2}{n\eta(\mu+c)},\ T_{BT}=\frac{2}{n\eta(\mu+c)} \quad (6)$$

3.2 层次化结构的 BitTorrent 系统模型

在层次化结构的 BitTorrent 系统中, 每个结点在其所在的结点簇中选择固定数量 n 的请求下载结点, 并与这些结点建立连接交换共享文件. 在时间 t, 结点簇 $C1$ 中所有结点的上载率是 $n\mu x_1(t)$; 结点簇 $C2$ 中所有结点的上载率是 $n\mu x_2(t)$. 因此, 结点簇 $C1$、$C2$ 中结点数量的变化率可以用以下的方程组表示:

$$\begin{aligned}\frac{dx_1(t)}{dt} &= \lambda - n\mu x_1(t)\\ \frac{dx_2(t)}{dt} &= \lambda - n\mu x_2(t)\end{aligned} \quad (7)$$

在稳定状态下(即 $t \to \infty$), 我们有 $\frac{dx_1(t)}{dt} = \frac{dx_2(t)}{dt} = 1$. 因此, 可以得到稳定状态下, 系统稳态的方程组:

$$\begin{aligned} 0 &= \lambda - n\mu\bar{x}_1 \\ 0 &= \lambda - n\mu\bar{x}_2 \end{aligned} \quad (8)$$

其中 $\bar{x}_1$ 和 $\bar{x}_2$ 分别是 $x_1(t)$ 和 $x_2(t)$ 的平衡值. 解方程组(8), 得:

$$\bar{x}_1(t) = \bar{x}_2(t) = \frac{\lambda}{n\mu} \quad (9)$$

根据等式(9), 我们知道式(3)有一个唯一得解, 系统存在一个平衡点 $(\bar{x}_1, \bar{x}_2)$.

用 Little's law, 我们可以计算在稳定状态下结点的平均下载时间:

$$T_1 = T_2 = \frac{1}{n\mu}, \; T_{HBT} = \frac{1}{n\mu} \quad (10)$$

其中 T_1 和 T_2 分别是结点簇 $C1$ 和结点簇 $C2$ 中结点的平均下载时间, T_{HBT} 是层次化结构的 BitTorrent 系统中所有结点的平均下载时间.

在层次化结构的 BitTorrent 系统中相同结点簇中的结点连接带宽总是大于在不同子域中的两个结点的连接带宽, 即 $\mu > c$. 比较式(6)和式(10), 我们发现 BitTorrent 系统中结点的平均下载时间大于层次化结构的 BitTorrent 系统中结点的平均下载时间, 即 $T_{BT} > T_{HBT}$. 模型分析结果显示层次化结构的 BitTorrent 系统结构减少了系统参与结点平均下载时间, 提高系统性能. 同时在层次化结构的 BitTorrent 系统中, 结点平均下载时间与结点到达率 λ 无关, 因此层次化结构的 BitTorrent 系统与原 BitTorrent 系统一样, 系统具有很好的健壮性.

4　模拟实验

为了模拟 Internet 的网络时延, 我们使用 GT-ITM 生成的 Transit-Stub 网络拓扑, 并采用了一个类似 BitTorrent 系统协议的事件驱动模型. 为了保证实验结果与底层网络拓扑无关, 模拟实验分别使用了 5 种不同的网络拓扑结构. 设被共享的文件大小是 8M, 此文件被分为 16 个文件块, 每个文件块大小为 512K. 每个结点的并行上载连接数为 5. 系统初始时有 8 个种子结点随机分布在系统中.

图 3(a)~(d)显示了不同网络规模下结点完成时间的累积分布(CDF). 我们分别选择 100、500、1000、1500 个结点随机分布到 Transit-Stub 覆盖网上, 设系统中有 4 个结点簇被 4 个超级结点所管理. 在层次化结构 BitTorrent 和原 BitTorrent 两个系统中, 随着时间的增长, 结点完成时间的 CDF 也随之增加. 实验结果显示, 层次化结构 BitTorrent 系统中结点完成时间的 CDF 总是大于原 BitTorrent 的结点完成时间的 CDF. 在图 3(c)中, 我们注意到在层次化结构的 BitTorrent 系统中所有系统参与结点完成下载平均时间约为 20min, 而在原 BitTorrent 系统中所有结点完成下载平均时间约为 80 分钟, 较层次化结构 BitTorrent 系统慢了 4 倍, 说明层次化结构的 BitTorrent 系统能显著地提高系统下载效率.

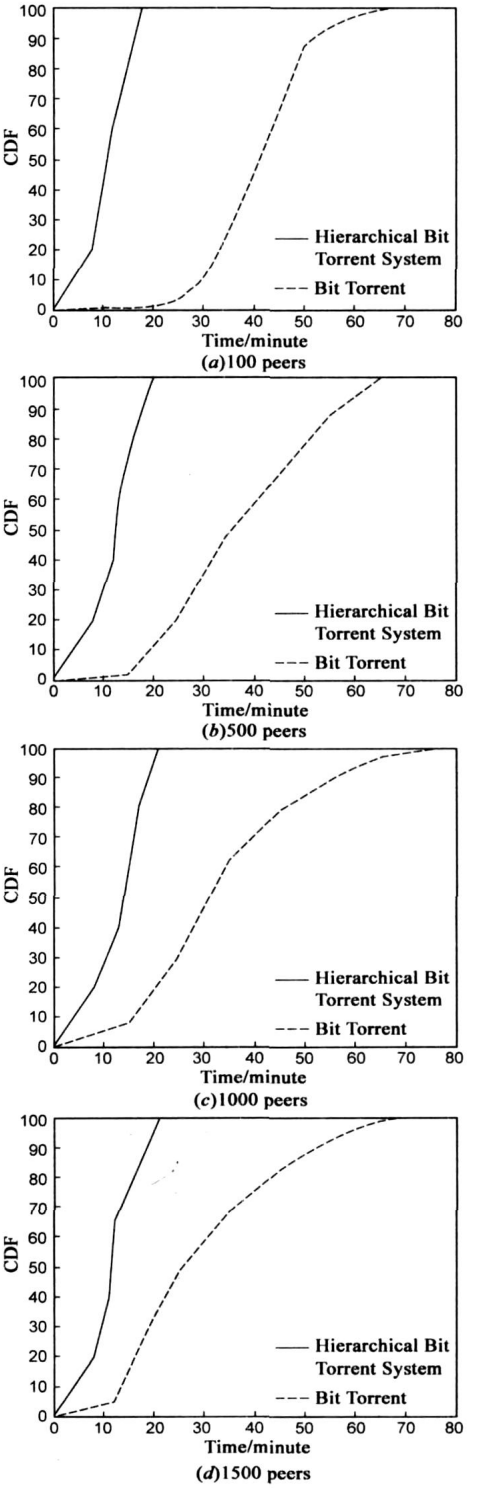

图 3　不同网络规模下结点完成时间的 CDF

图 4 显示了在层次化结构的 BitTorrent 与原 BitTorrent 两个系统中, 不同网络规模下结点平均下载完成时间. 在不同的网络规模下, 系统中所有结点平均下载完成时间总是小于原 BitTorrent 系统中结点平均下载完成时间. 例如, 在具有 100 个系统参与结点的网络中, 层次化结构的 BitTorrent 系统中所有结点平均下载完成时间只有原 BitTorrent 系统中结点平均下载完成时间的 30%; 在具有 1500 个系统参与结点的网络中, 层次化结构的 BitTorrent 系统中所有结点的平均下载完成时间只有原 BitTorrent 系统中所有结点的平均下载完成时间的 35%.

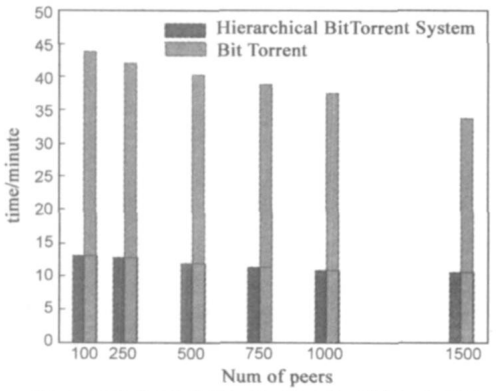

图 4 不同网络规模下结点平均下载完成时间

性能的改善主要由于系统将相互间网络时延低、连接带宽高的结点聚合在同一个结点簇中, 结点簇内的结点优先建立共享连接, 从而优化系统中结点连接, 实现文件共享本地最大化.

5 相关工作

对等网文件共享系统的覆盖网拓扑结构近年来已经引起了广泛的研究, 例如 CoopNet[9]、Splitstream[10]、Slurpie[11]、SODON[12]. BitTorrent 作为目前最为流行的对等网文件共享系统的应用, 已经引起人们广泛的兴趣[3,4]. 然而, BitTorrent 系统覆盖网拓扑结构上的缺陷并没有得到完全充分地研究.

分层的网络结构方法已经被应用到对等网的网络应用中, 它有利于提高对等网络性能, 例如 FastTrack[13]、SODON[12]、ECSP[14]. 然而, 几乎大部分分层结构的 P2P 系统都关注在共享文件的查找, 很少被引入到大规模文件分布与下载的 P2P 网络系统中. 在文献[15]中, 邻近结点之间的文件交换的特性已经被介绍, 并且说明了此方法能有效地提高 P2P 系统的文件共享效率. 该研究建立了一个基于结点邻近性的 P2P 覆盖网络, 目标是在网络中找到最邻近的结点相互交换共享文件, 但是严格的邻近结点间的文件交换将导致系统缺乏健壮性. 本文提出的系统既能保证邻近结点之间共享文件的相互交互从而提高文件下载速度, 又能保证结点的随机选择使得系统具有较好的健壮性和可扩展性.

本文利用了文献[3]提出的数学模型理论上比较了基于临近结点聚类构建层次化 BT 系统和 BT 系统. 在本文的模型中, 利用了数学期望的方法来确定稳定状态下节点的上载带宽, 并且考虑了优化不阻塞和阻塞算法对带宽选择的作用. 这是文献[3]没有做到的.

6 小结

针对传统 BitTorrent 系统中结点连接没有优化以及 Tracker 服务器存在瓶颈等问题, 为了提高 P2P 系统的性能, 本文提出结合 RTT 与 TTL 的结点间临近性比较算法和基于结点临近性的分布式结点加入算法, 将临近的结点聚合成结点簇, 同一结点簇中结点优先建立共享连接, 从而构建了层次化 BitTorrent 文件共享机制. 通过流体数学模型分析和模拟实验证实了层次化 BitTorrent 系统具有更好的文件共享性能.

参考文献:

[1] B Cohen, Incentives build robustness in BitTorrent[A]. Proc P2P Economics Workshop[C]. Berkeley: ACM Press, 2003. 43 – 48.

[2] T Karagiannis. Is p2p dying or just hiding? [A]. Proc Globecom[C]. Dallas, TX, USA, 2004. 1532– 1538.

[3] D Qiu, et al. Modeling and performance analysis of bitTorrent-like peer-to-peer Networks[A]. Proc ACM Sigcomm2004[C]. Portland: ACM Press, 2004. 367– 377.

[4] L Guo, et al. Measurements, analysis, and modeling of BitTorrent-like systems[A]. Proc Internet Measurement Conference [C]. Berkeley, CA: ACM Press, 2005. 213– 221.

[5] P Francis, et al. IDMaps: A global internet host distance estimation service[J]. IEEE/ACM Transaction on Network, 2001, 9 (5): 525– 540.

[6] RFC 1102, Policy Routing in Internet Protocols[S].

[7] E W Zegura, et al. How to model an Internetwork[A]. Proc IEEE INFOCOM[C]. San Francisco, CA, USA: IEEE Computer Society, 1996. 135– 144.

[8] D Bertsekas, et al. Data Networks[M]. NJ, USA: Prentice Hall, Englewood Cliffs, NJ, 1987.

[9] V N, et al. The case for cooperative networking[A]. Proc 1st International Workshop on Peer-to-Peer Systems[C]. Berlin: Springer Verlag, 2002. 178– 190.

[10] M Castro, et al. Splitstream: highbandwidth content distribution in cooperative environments[A]. Proc 2nd International Workshop on Peer-to-Peer Systems[C]. London: Springer Verlag, 2003. 292– 303.

[11] R Sherwood, et al. Slurpie: a cooperative bulk data transfer protocol[A]. Proc IEEE INFOCOM[C]. HongKong: IEEE

Computer Society, 2004. 941 - 951.

[12] P Zheng, et al. SODON: a high availability multi-source content distribution overlay[A]. Proc 13th International Conference on Computer Communications and Networks[C]. Chicago, USA: IEEE Computer Society, 2004. 87 - 92.

[13] FastTrack Website[EB/OL]. http://www.fasttrack.nu/, 2002-02-13.

[14] J Li, et al. An efficient clustered architecture for P2P Networks [A]. Proc the 18th International Conference on Advanced Information Networking and Applications[C]. Fukuoka, Japan: IEEE Computer Society, 2004. 278 - 283.

[15] A Qureshi. Exploring Proximity Based Peer Selection in BitTorrent-like Protocol[EB/OL]. http://pdos.csail.mit.edu/6.824-2004/reports/asfandyar.pdf, 2004-06-08.

作者简介:

薛广涛　男, 1976年生于江苏徐州, 上海交通大学计算机科学与工程系, 博士, 讲师. 主要研究方向为对等网络、无线传感器网络、移动计算和分布式系统. E-mail: xuegt@cs.sjtu.edu.cn

俞嘉地　男, 1975 年生于陕西西安, 上海交通大学计算机科学与工程系博士后, 讲师. 主要研究方向为对等网络和分布式系统.

尤晋元　男, 1939 年生, 上海交通大学计算机科学与工程系教授, 博导. 主要研究方向为分布式系统、计算、面向对象技术和软件工程.