第 34 卷 第 11 期 计 算 机 工 程 2008 年 6 月
Vol.34 No.11 Computer Engineering June 2008
·博士论文· 文章编号：1000—3428(2008)11—0023—03 文献标识码：A 中图分类号：TP303.03

Tree-Ring:一种结构化的应用层组播模型

陈永刚[1]，贾春福[1]，吕述望[2]，徐 亮[3]
(1. 南开大学信息技术科学学院，天津 300071；2. 信息安全国家重点实验室，北京 100049；3. 捷开通讯有限公司上海分公司，上海 201203)

摘 要：在分析现有应用层组播协议基础上，提出 Tree-Ring 模型，该模型构建于 Pastry 之上，采用 Pastry 的路由与定位机制，构造一个树与环相结合的覆盖网络。实验显示，模型中 70%以上的节点出度为 1，80%以上的节点的相对延迟比控制在 2.7 以内。结果表明，Tree-Ring 能有效地平衡节点负载，满足大规模网络中大内容传播的需要。

关键词：应用层组播；对等网；组通信

Tree-Ring: A Structured Application Level Multicast Model

CHEN Yong-gang[1], JIA Chun-fu[1], LV Shu-wang[2], XU Liang[3]
(1. College of Information Technical Science, Nankai University, Tianjin 300071; 2. State Key Laboratory of Information Security, Beijing 100049; 3. Shanghai Branch, JRD Communication, Inc., Shanghai 201203)

【Abstract】 This paper presents the Tree-Ring, by analyzing traditional application level multicast protocol, which is built on top of Pastry. It combines tree concept with ring concept to build an overlay network with the location and routing mechanism of Pastry. Experiment shows that the out-degree of nodes over 70% is 1 and the RDP of nodes over 80% is within 2.7. The results show that Tree-Ring can balance the load on the nodes efficiently and meet the demands of the transmitting of large files on the large scale networks.

【Key words】 application level multicast; Peer-to-Peer; group communication

1 概述

虽然IP组播技术能较好地解决多方通信的需求，但是自身却存在着一些难以解决的问题[1]，如扩展性差、网络管理复杂、对高层应用支持性差等。因此，尽管IP组播技术已提出 10 多年，但是并没有得到很好发展。

近来应用层组播技术受到广泛关注。应用层组播就是在应用层为组播数据包进行路径选择，数据包的复制及转发都由终端完成，组成员的管理由终端动态完成并及时更新。

根据路由机制的不同，应用层组播协议可以分为 2 种：结构化和非结构化[1]。在非结构化协议(如ALMI, Yoid, NICE等)中，节点的命名标识采用IP地址，内容的存放与网络拓扑无关，此类协议突出特点是扩展性差；在结构化协议(如Scribe, Bayeux等)中，在IP地址之上添加了新的命名层nodeId，文件存放在指定位置，根据路由表，查询可以高效到达节点。

本文提出的 Tree-Ring 是一种结构化、大规模、分散式的应用层组播模型，具有低出度、低延迟、可扩展、自组织等特点。

2 相关研究

2.1 Pastry

Pastry[2]是一个自组织覆盖网络，为上层应用提供底层路由服务。Pastry为每个节点分配一个nodeID(0~2^{128}-1)。节点加入时，系统随机分配一个nodeID给它，2 个nodeID相邻的节点可能没有任何物理上的联系。

假设覆盖网络有N个节点，则任一节点的路由表有$\lceil \log_{2^b} N \rceil$行($b$是配置参数)，每行有 2^b-1 条记录，同行中的节点nodeID拥有相同前缀。除路由表外，节点还需维护叶节点集的IP地址。

消息路由过程中，节点把消息发送到下一个与目标节点拥有更长相同前缀的节点，若无此类节点，则把消息发送给与当前节点拥有相同长度前缀且数值上与目标更接近的节点。

2.2 Scribe

Scribe[3]建立在Pastry之上，每个组有唯一的groupID，nodeID与groupID最接近的节点担当该组的汇聚点(即组播树的根)。

Scribe 采用类似反向路径组播的机制建立组播树。树中的节点称为传递者，传递者可以不是该组成员，每个传递者为每个组单独维护一个孩子节点表。节点加入组时，由 Pastry 路由一条 key 值为 groupId 的 JOIN 消息，路径上的节点收到 JOIN 消息后，先判断自己是否为该组的传递者：若是，则把发送 JOIN 消息的源节点加入自己的孩子节点表，JOIN 消息到此结束；否则，自己先成为该组传递者，然后把 JOIN 消息源节点加入自己的孩子节点表，再继续路由 JOIN 消息到下一个节点。

消息多播时，消息的源节点首先发送 MULTICAST 消息到汇聚点，汇聚点返回一条包含自己 IP 地址的消息，此后源节点把数据发送给汇聚点，由汇聚点沿着组播树把数据发送到整个组播组中。

基金项目：国家自然科学基金资助项目(60577039)；天津市科技发展计划基金资助项目(05YFGZGX24200)

作者简介：陈永刚(1980－)，男，博士研究生，主研方向：信息安全，分布式计算；贾春福、吕述望，教授、博士生导师；徐 亮，工程师、硕士

收稿日期：2007-06-12 **E-mail：**yongang@mail.nankai.edu.cn

3 Tree-Ring 模型

3.1 Tree-Ring 模型分析

Tree-Ring 构建于 Pastry 之上，利用 Pastry 的路由功能，完成组播组创建、成员管理、数据传播及节点失效的处理。

Tree-Ring充分利用全体节点的带宽资源，减小最大上传带宽至一个合理的上限。与Scribe, Bayeux[4]等采用的树型结构相比，Tree-Ring节点的出度始终在 {0,1,2} 中取值。模型的网络拓扑结合了树型结构和环状结构的特点(见图 1)，拓扑的层次数为 $\log_m n$ (m 是环内节点数，n 是网络节点数)，在网络节点数量较大的情况下，通过设定 m 值，使模型保持较少的层次。

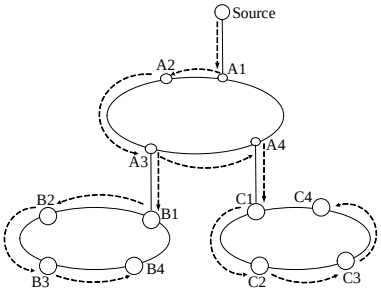

图 1　Tree-Ring 模型的数据转发路径

在模型中，同环的节点称为兄弟节点，与上层环相连的节点称为首节点。父节点是处于上层环的节点，孩子节点处于下层环，父节点与孩子节点之间有边相连。

数据包从源点到叶节点，最坏情况下要途经 $\log_m n$ 层及每层环上的所有节点。设环内节点平均延迟为 d，环间节点平均延迟为 D，则从源点到叶节点的最大延迟为 $[d\times(m-1)+D]\times\log_m n$。构造 TR 时，尽量使临近的节点处于同一个环上，相距较远的节点处于不同的环，以保证 d 远小于 D 的性质。因此，与树型结构相比，TR 模型中数据包要走更长路径的特性不会显著增加延迟。

3.2 Tree-Ring 算法描述

3.2.1　数据传播

除源节点外，所有节点都有入向边。通过入向边与节点相连的邻接节点称为前导节点；通过出向边与节点相连的邻接节点称为后续节点，节点的后续节点包括孩子节点和位于它之后的第一个兄弟节点(如 A3 的后续节点包括 B1 和 A4)，环上的兄弟节点按照加入时间排序。

数据发送就是为后续节点提供数据。数据从数据源节点发送开始，途中经过节点转发直至最底层的节点，整个过程如图 1 中虚线所示。环是一组节点的逻辑特征，如 A1, A2, A3, A4 组成一个环，但在数据发布的过程中，A1 和 A4 之间并没有数据交互。

3.2.2　组创建

源节点将自己的 nodeId 作为组 groupId，这样源节点即成为组的根节点，其他节点加入组时，以 groupId 为目的地址发送 JOIN 消息，JOIN 消息通过底层网络传递，直至到达属于此组播组的某一节点。

3.2.3　成员管理

节点 j 加入组播组时，向源节点发送 JOIN 消息，JOIN 消息到达属于此组的节点 i，i 收到 JOIN 消息后，首先测度和 j 之间的延迟，如果延迟小于设定的阈值，则把 j 加为兄弟节点，否则加为孩子节点；若延迟小于设定的阈值，但 i 的兄弟节点已满，则把 j 加为孩子节点；若 i 已有孩子节点 k，则把 JOIN 消息转发至 k，重复上述步骤，直至 j 加入组中某个环。

节点退出分 2 种情况：正常退出和意外退出。节点正常退出时，发送退出消息给所有相邻节点，相邻节点收到退出消息后调用相应的拓扑修复机制；意外退出可能是由链路断裂或节点失效引起的。图 2 中虚线部分描述了节点意外退出的情形。

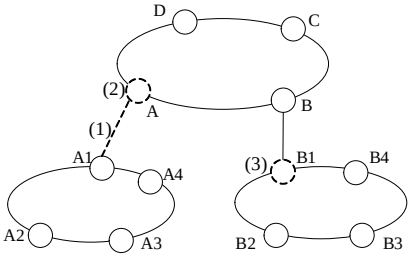

图 2　节点或链路失效时的拓扑维护

当链路不通时，如图 2 中的虚线(1)所示，A 和 A1 失去通信，则 A 把孩子节点域标记为空，同时，A1 检测到自己与父节点通信中断，则代表环 {A1, A2, A3, A4} 以及之下的所有环重新申请加入组播网络。

当节点 A 失效时，如图 2 中的虚线(2)所示，不仅引发以 A1 为代表的节点重新加入，还将引发环 {A, B, C, D} 的拓扑信息更新，产生新的环结构 {B, C, D}。

如果是环内首节点失效，那么还需要先推举一个新的首节点。如图 2 中的虚线(3)所示，B1 失效后，B2 成为首节点(推举按照加入时间顺序确定，最先加入的节点即是首节点)。

上述涉及节点生存状态的维护，后续节点如果在一定时间内没有收到前导节点的数据，则发送探测消息，如果设定时间内得不到答复就认为前导节点已经失效；同时，后续节点定期向前导节点发送心跳消息，如果前导节点在一定时间内没有收到心跳消息，则认为后续节点失效。

4 Tree-Ring 仿真测试

4.1 度量参数

4.1.1　相对延迟比

本文比较 Tree-Ring 和 IP 组播下的数据传输延迟。与所有应用层组播(Application Level Multicast, ALM)协议一样，Tree-Ring 中数据包由端系统复制转发，相对 IP 组播而言传输延迟有所增加。本文使用相对延迟比 *RDP* 来评测 2 种情况下的延迟：

$$RDP = Delay_{[alm]} / Delay_{[ipmulticast]}$$

4.1.2　节点出度

节点出度表示该节点需要向多少节点转发数据。理想模型中，各节点的出度应该比较均衡，且维持在较低水平，这样既可以充分利用节点的带宽资源，又能避免节点过载。

4.2 仿真环境

笔者设计了一个数据分组层的离散事件仿真器来评价 Tree-Ring 性能。仿真器运行在有 100 个路由器的网络上，网络拓扑由 GT-ITM 根据 Transit-Stub 模型生成。Transit-Stub 模型是分层的，本文所用的模型有一个 Transit 域，Transit 域的平均节点数为 4，Stub 域的平均节点数为 8，每个 Transit 节点平均有 3 个 Stub 域，每个路由器带 10 台端系统，通过局域网互联。使用不同的随机数生成了 10 个拓扑，然后在此拓扑基础上重复了 10 次实验，文中最后的实验数据是这

10 次实验所得数据的平均值。

4.3 实验数据及分析

为了验证 Tree-Ring 的效果，把 Tree-Ring 和同是构建在 Pastry 上的 Scribe 进行了对比，测试结果见图 3 和图 4。图 3 中横坐标值 2+表示节点出度大于等于 2，但此处 Tree-Ring 节点出度值等于 2。

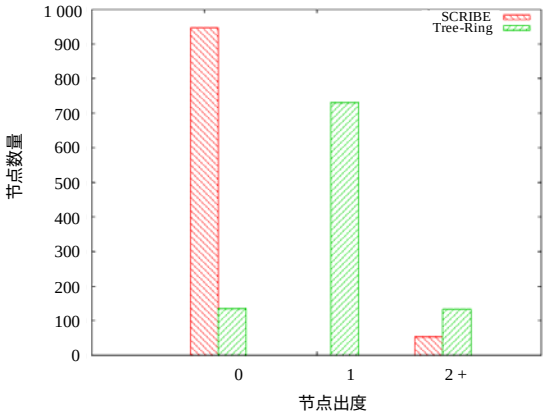

图 3　Tree-Ring 和 Scribe 的节点出度对比实验

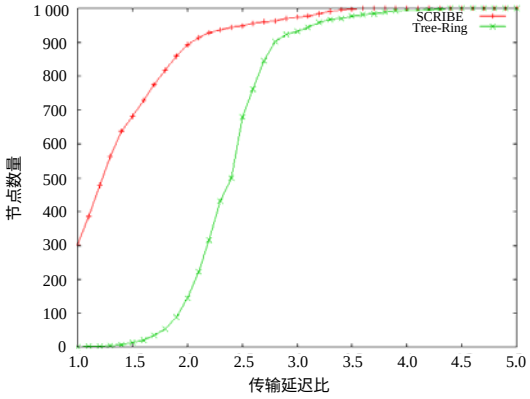

图 4　RDP 的累计分布

如图 3 所示，Tree-Ring 中 70%以上的节点出度为 1；其余节点，出度为 0 和 2 的各占一半。与之对应，Scribe 中 90%以上的节点出度为 0，其余的出度均大于 1。由此可知，Tree-Ring 的节点上载平衡能力要优于 Scribe。

在图 4 中，Tree-Ring 超过 80%的节点的 RDP 在 2.7 以内，99%的节点的 RDP 在 4.0 以内。虽然相比 Scribe，Tree-Ring 的 RDP 在 2 以内的节点数较少，但是大部分都控制在 3 以内，传输延迟维持在可以接受的较低水平。

5　结束语

本文提出了 Tree-Ring——基于结构化的应用层组播模型。模型构架在 Pastry 之上，为上层应用提供组加入、组退出、数据组播和任意播的服务。

Tree-Ring 中，节点加入和网络失效时的管理都是分布式的；树型结构和环形结构的结合应用避免了高连通度的关键节点；环间节点延迟低、拓扑层次少的特征使延迟维持在较低水平；利用键值路由层的自组织特性，拓扑维护机制确保组播网适应节点和链路失效引起的变化。

与 Scribe, Bayeux, CAN multicast 等基于树状结构的应用层组播模型相比，Tree-Ring 能够满足大规模网络中大内容传播的需要，在实时流媒体的发布中具有较好的应用前景。

参考文献

[1] Li Lao, Cui Junhong, Gerla M, et al. A Comparative Study of Multicast Protocols: Top, Bottom, or In the Middle?[R]. Computer Science Department of UCLA, Technical Report: TR040054, 2005.

[2] Rowstron A, Druschel P. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems[C]// Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms. Heidelberg, Germany: [s. n.], 2001.

[3] Castro M, Druschel P, Kermarrec A M, et al. Scribe: A Large-scale and Decentralized Application-level Multicast Infrastructure[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(8): 1489-1499.

[4] Zhuang S Q, Zhao B Y, Joseph A D, et al. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination[C]// Proc. of the 11th International Workshop on Network and Operating System Support for Digital Audio and Video. Port Jefferson, New York, USA: [s. n.], 2001.

（上接第 22 页）

使用本文方法重建的 Lena 图像、使用 Barnsley 提出的经典分形编码方法重建的图像分别见图 3、图 4，实验结果列于表 1 中。

图 3　本文方法重建图像

图 4　经典方法重建图像

表 1　重建图像的一些重要结果

图像	压缩比	峰值信噪比/dB
图 3	30.6	36.2
图 4	19.7	26.4

从实验结果可知，本文算法与经典分形压缩算法相比，在压缩比和重建图像的峰值信噪比上有了较大的改进。解码后的图像效果好，压缩比得到了提高。

参考文献

[1] Jacquin A E. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation[J]. IEEE Transformation on Image Processing, 1992, 1(1):18-30.

[2] Jacobs E W, Fisher Y. Image Compression: A study of the Iterated Transformation Method[J]. Signal Processing, 1992, 29(2): 127-142.

[3] Coper-Gordon R J. Julia Set of the Complex Carotid-Kundalinifunction[J]. Computer & Graphics, 2001, 25(1): 153-158.

[4] Popeseu D C, Dinca A. A Nonlinear Model for Fractal Image Coding[J]. IEEE Transformation on Image Processing, 1997, 6(3): 373-382.