

基于 P2P 网络高效广播传递算法的研究

林晓帆, 李 超

(成都信息工程学院软件工程系, 成都 610225)

摘要:提出了一种基于 P2P 网络的高效广播传递算法, 算法综合了两种覆盖网络的广播传递算法的优点: 一种是规则网络, 另一种是采用感染算法通信的非结构化网络。形成的元结构算法比原来的算法具有更快的广播传递速度、更低的消息复杂度和更高的鲁棒性。实验表明该方法具有可行性。

关键词: P2P 网络; 规则广播算法; 感染广播算法; 元算法

Research on Efficient Broadcast Algorithm in P2P Grids

LIN Xiaofan, LI Chao

(Department of Software Engineering, Chengdu University of Information Engineering, Chengdu 610225)

【Abstract】 This paper presents an efficient algorithm for performing a broadcast operation in P2P grids. The approach combines the advantages of two overlay architectures: regular networks and unstructured networks with epidemic communication. The resulting meta architecture provides higher robustness as well as higher speed and lower message complexity than appropriate base methods. Preliminary experiments show the viability of the approach.

【Key words】 P2P grids; Regular broadcast algorithm; Epidemic broadcast algorithm; Meta algorithm

P2P 网络由于充分利用广域网络和 Internet 上的巨大资源(例如: 空闲的 CPU 时间), 因此成为目前分布式计算研究中最有前途的方法。在一个松耦合的分散网络中, 广播是一种重要的通信机制, 它被应用于资源定位、计算任务分发、紧急告示等很多应用中。理想状态下, 广播应该以非常快的速度和很低的消息复杂度到达网络中的所有节点, 但是 P2P 覆盖网络的动态变化会严重干扰广播的传递。P2P 网络为了对此进行补偿, 经常采用高冗余的广播传递方法。

正是基于以上问题, 本文提出了一种基于 P2P 网络的高效广播算法, 方法综合了规则(非冗余)和感染(冗余)广播传递算法的优点, 形成的元算法具有高度的容错性和快速的广播传递性能。这种方法有效地降低了非结构分布算法(例如: 泛洪法和感染法)的消息复杂度。

1 相关工作

近年来在 P2P 网络为背景的前提下, 网络计算得到了广泛研究, 涉及从网络设计^[8]、分布资源利用^[4]到负载均衡^[5]等诸多方面。与客户/服务器模式系统相比, 完全非集中化的系统提供更高的容错能力和更好的扩展性。几乎所有提出的系统都采用广播传递作为通信机制, 例如 Gnutella 的泛洪法^[1], 这种方法的缺点是广播消息的高度冗余, 导致网络开销的急剧加大。

对于像 Chord^[8]、CAN^[6]、Pastry^[7]这样的结构化网络, 它们充分利用规则网络设计的特点, 使用了一种特殊的广播算法, 例如 Chord 采用的广播算法^[2]具有最小的消息复杂度和非常快的传播速度, 为了加速广播传递, 网络地址空间分割成等尺寸片段。对规则化网络的广播算法和泛洪法的系统比较, 表明规则广播方法比泛洪法性能要好。但是在实际的 P2P 网络中, 由于会出现同时有多个节点并发离开或加入网络的情况, 对于规则网络广播会出现某些区域不能被广播所覆盖的

情况。这个问题经常在规则广播传递算法中被忽略。

另一方面, 感染算法^[6]采用随机挑选邻居的方法实现非结构化 P2P 网络通信, 它比泛洪法发送的消息数量要少得多^[3]。虽然这种方法已证明可以保证使广播到达所有节点, 但是由于具有比较高的消息复杂度, 因此它比规则广播网络算法速度要慢。

本文的方法综合了规则和感染广播算法的优点, 规则算法通过规范、非冗余的传递到达尽可能多的节点, 感染算法完成剩下不能覆盖的节点。

2 算法结构

2.1 自组织网络模型

本文的网络模型由一些分散的节点构成, 每个节点都有一个关于网络结构的邻居表, 表中包含邻居地址、最后与邻居联系的时间戳等实体。每个节点 S_i 的地址表示为 hash 函数 $adr(S_i)$, 即 $adr(S_i) \in [0..adrMax]$ 。节点 S_x 和 S_y 间的距离定义如下:

$$dist(s_x, s_y) = \begin{cases} adr(s_y) - adr(s_x), & \text{当 } adr(s_x) < adr(s_y) \\ adr(s_y) + (adrMax - adr(s_x)), & \text{当 } adr(s_x) > adr(s_y) \\ 0 & \end{cases}$$

每个节点占据其邻居表中最靠近“理想位置”的地址, 所谓“理想位置”是指节点与其邻居节点的距离为 2^i 。节点从邻居表中随机选择邻居并采用规则算法进行联系, 同时更新时间戳, 然后寻找与“理想位置”更近的邻居进行联系。对邻居表进行更新的“推-拉”算法总结如下:

(1) 每个节点 S_i 根据一定的时间间隔, 每次随机地从其邻

作者简介: 林晓帆(1968 -), 男, 讲师、硕士, 主研方向: 计算机网络, 软件工程; 李 超, 教授、博导

收稿日期: 2006-12-26

E-mail: xiaoflin68@163.com

居表中选择一邻居 S_b 的地址；

(2)节点 S_a 和节点 S_b 交换它们邻居表的信息，把获得的新实体增加到各自的表中。当 S_a 与 S_b 共同拥有同样的邻居 S_c 时， S_c 的时间戳在 S_a 与 S_b 的邻居表中设置为最年轻值；

(3)比给定 TTL 门限值老的实体从邻居表中移出；

(4) S_a 计算它的“理想位置”集合 x_i ，定义为

$$dist(s_a, x_i) = 2^i, i = 0..log_2(adrMax)$$

(5)对每个理想位置 $x_k \in \{x_i\}$ ， S_a 在其邻居表寻找最近的节点 $nb(i)$ ，定义为

$$\arg \min(i, \text{Min}\{dist(x_k, nb(i)), dist(nb(i), x_k)\})$$

(6) S_a 移出所有不符合理想位置的邻居。

一个同伴只需要一个生存地址便能加入网络成为新节点。新节点 S_n 寻找网络中的优化位置，并且直接发送查询请求给它的直接邻居 S_c ， S_c 处理请求过程如下：

(1)计算请求节点与自己的距离 $dist(s_n, s_c)$ ，以及邻居表中所有邻居与 S_n 的距离；

(2)当邻居表中包含有距离新节点 S_n 更近的实体，即： $dist(s_n, neighbor(k)) < dist(s_n, s_c)$ ，请求转发到更近的邻居节点 S_k ；

(3)否则 S_c 发出响应消息给 S_n ，响应消息包括它的地址和邻居表信息；

(4) S_c 更新它的邻居表（如果它们距离最近， S_n 加入 S_c 的邻居表中）。

从网络中移出一个同伴不需要任何管理操作，在一个 TTL 周期后，它的地址将会从所有邻居表中删除。

2.2 广播算法

网络中的每个节点都可以发起广播。广播消息包含需要广播的信息和一个limit限制参数，这个参数用来局限广播传递的空间。当节点 S_a 接收到广播消息时，它从邻居表中挑选最接近它与limit之间中间距离位置的邻居 S_b ，中间距离定义为： $middle = adr(s_a) + 0.5 \times dist(s_a, limit)$ ，然后 S_a 向 S_b 发送广播消息和limit参数。节点 S_a 继续在它的地址与 $adr(s_b) - 1$ 间按照上述方法传递广播。广播传递如图 1 所示。

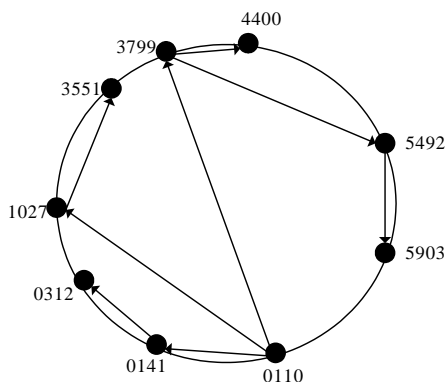


图 1 广播传递示意图

涉及广播传递的每个节点在每次与其它节点建立广播联系后，广播传递空间折半。在具有稳定邻居表的 N 个节点网络中，经过 \log^N 跳和发送 N-1 条消息后，广播可以到达所有节点。这种算法称为规则广播传递算法。

网络结构的动态变化会给广播传递带来问题，由于规则的非冗余广播传递会形成一个覆盖所有节点的扩展树，因此一旦某个特定的节点出现故障，都会造成这个节点下的整个子树广播无法达到。为了解决这一问题，本文的算法综合了

规则和感染传递算法的特点。所有接收广播信息的节点，除了进行规则的广播传递外，还周期性地随机选择邻居进行感染通信。与规则传递算法比较，感染广播的传递空间不受约束。从随机图论的结果可以表明：感染广播能够以非常高的概率到达网络中的所有节点。

2.3 消息的复杂度

使用感染通信进行广播传递的一个主要缺点是消息的高度冗余性。如果不进行优化，每个节点 K 会收到邻居表中包含 K 的所有节点发来的广播。由于节点间交换和更新邻居表同步于广播的传递，因此 K 在最坏情况下会收到 N-1 个冗余广播消息。当广播信息的长度很小时，它的影响可以忽略，否则它会造成很大的网络开销。

为了减少感染广播传递可能会增大网络开销的情况，本文对感染广播算法进行了如下改进：每个节点的邻居表中增加一个“广播状态”标志。所有过去被当前节点感染的邻居或者发送广播消息的当前节点的邻居，都会在未来的广播传递中被排除掉。当两个节点交换它们的邻居表时，它们同时更新所有共享邻居的“广播状态”标志。节点进行感染联系的方法与原来一样，只是感染广播传递的节点仅包括“广播状态”标志没有被设置的节点。这种方法的合理性在于：“广播状态”标志能够以一种非常紧凑的方式进行编码，因此利用感染广播对它进行传输所带来的网络开销要比传输感染消息小得多。

3 实验

本文对提出的广播传递元算法与现存的感染、规则广播传递算法进行了实际比较。实验的重点是不同算法间的容错性、传递速度及消息的复杂度。系统地对 P2P 网络的动态行为进行了多方面仿真，包括：

(1)恒定网络：节点在仿真开始时加入网络，仿真结束时节点仍然在网络中保持激活状态；

(2)稳态网络：节点随机加入和离开网络，加入和离开的速度相同，平均的网络节点数保持一致。

(3)动态变化网络：短时间内大量节点加入或离开网络。

另外，我们还比较了基于上述几种不同算法在邻居表中有“广播状态”标志（见 2.3 节描述）和无此标志的消息复杂性。

3.1 测试

在本文的实验中，仿真网络具有以下特性：

- (1)参加节点数：100；
- (2)感染接触的时间间隔：200ms；
- (3)邻居表长度：10 个邻居；
- (4)平均消息传递延迟：10ms；

根据几种广播传递算法的节点特性，组成了不同结构的网络：

- (1)具有感染节点和采用感染广播传递的随机网络；
- (2)规则节点组成的规则网络；
- (3)感染节点和规则节点相混合的元模型网络；

为了描述动态变化的覆盖网络的广播传递行为，每种网络模型在多个固定的时间点触发广播。每种实验重复 10 次，同时附加随机因子。随机因子包括：特别节点加入或离开网络，感染接触的顺序，邻居表的更新等。

3.2 实验结果

图 2 显示了在短时间（10 000ms）50%的节点离开网络的广播传递情况。从图中可以看出，随着离开节点的数量

增加,规则广播到达节点的数量减少,而元算法通过感染传递到达的节点数仍然比较多。由于感染算法缺乏从邻居表中排除故障节点的机制,它将会影响广播的传递速度。从图中也可以清楚地看出,元算法要比纯粹的感染算法在广播传递上要快得多。

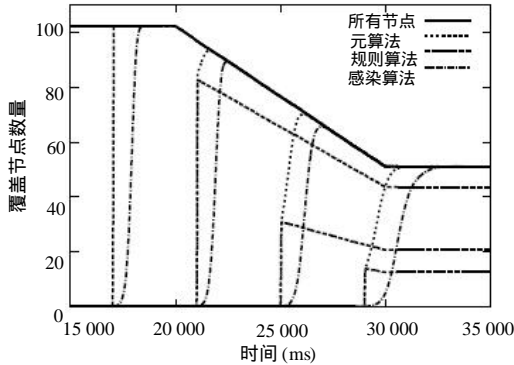


图2 广播传递图 (50%节点离开网络)

图3比较了当50%的节点在短时间内(10000ms)随机加入网络的广播传递情况。广播传递算法的性能与节点离开试验结果相似,但是可以清楚地看到,规则广播算法可以达到大部分节点,这主要得益于规则广播算法采用了高效的位置查询过程,它有助于新节点在很短的时间内集成进网络。

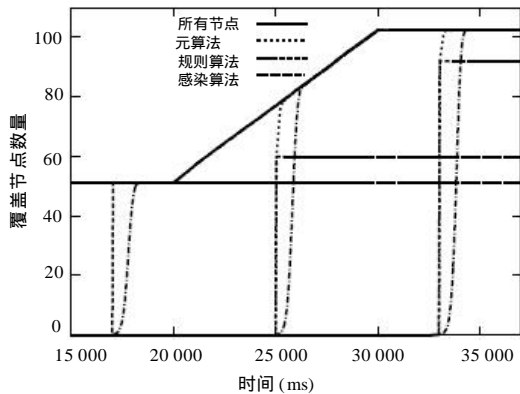


图3 广播传递图 (50%节点加入网络)

图4比较了两种情况下广播传递的情况:一种是“理想状态”,即没有节点离开或加入网络;另一种是“动态状况”,即节点随机离开和加入网络的速率相同,节点的平均数目保持固定。在“理想状态”,广播传递几乎能够完全被规则传递算法完成;在“动态状况”,规则广播覆盖了大部分节点,剩下的节点由感染通信完成。在两种状态下,纯粹的感染广播传递算法没有显现出太大差异。

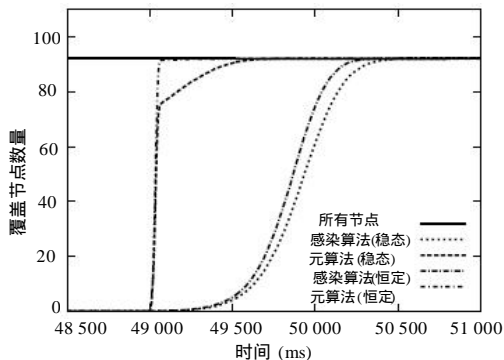


图4 稳定和变化网络的广播传递图

图5比较了当50%的节点离开网络时,感染算法与元算法的消息的复杂度。在没有节点故障时,元算法的传递的广播消息几乎全是规则广播消息;故障发生后,感染消息的数量变大,但是元算法的消息的复杂度始终比感染通信的消息复杂度低。

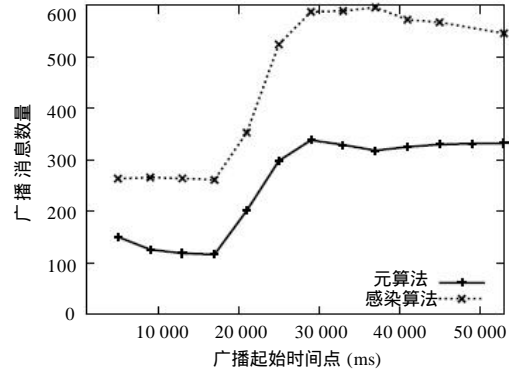


图5 广播消息复杂度图 (50%节点离开网络)

图6比较了元算法与感染算法的广播传递时间。可以清楚地看到元算法始终比感染通信传递广播的速度快。

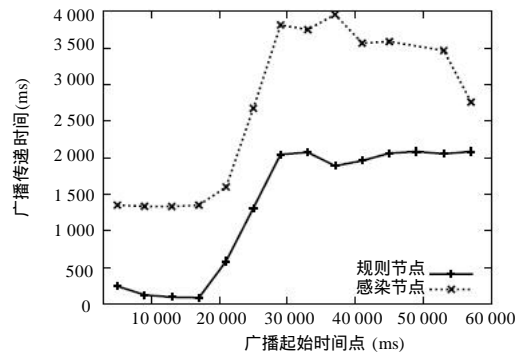


图6 广播传递时间图 (50%节点离开网络)

图7比较了在元算法中引入不引入“广播标志”对广播消息数量的影响。可以清楚地看到,当没有节点故障时,两种方法差异较小,随着节点故障数量增加,感染通信数量开始增加,引入了“广播标志”的广播明显减少了消息的数量,即减少了消息的复杂度。

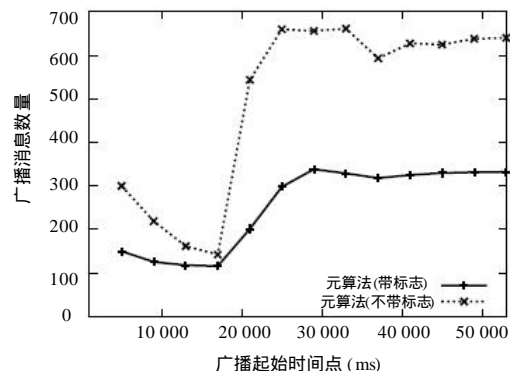


图7 标记信息对广播消息复杂度影响图 (50%节点离开网络)

4 结论

本文对异构的P2P网络提出了一种新的广播传递算法。本文的元算法综合了两种结构的优点:一种是规则网络,另一种是随机感染网络。通过实验仿真,可以清楚地看到我们提出算法的优点:更高的故障容错性,更快的广播传递速度,同时在邻居表信息中引入“广播标志”,极大地减少了消息的复杂度。

(下转第106页)