

基于 P2P 的 MMOG 兴趣域内多播改进算法

王 岩, 李照奎, 石祥滨

(沈阳航空工业学院计算机学院, 沈阳 110136)

摘要: 针对 SCRIBE 在解决基于 P2P 的巨量多玩家网络游戏兴趣域内多播时, 产生的多播效率低、游戏状态数据同步性差问题, 提出一种 SCRIBE 改进多播算法, 该算法考虑 P2P 网络中终端节点在带宽和计算能力上的异构性, 同时保证同一组中节点具有共同的兴趣。仿真结果表明, 该算法能够在较大程度上减小消息冗余和降低多播延迟, 提高游戏状态数据的同步性。

关键词: 对等网络; 兴趣域; SCRIBE 算法; 应用层多播

MMOG AOI Multicast Advanced Algorithm Based on P2P

WANG Yan, LI Zhao-kui, SHI Xiang-bin

(School of Computer, Shenyang Institute of Aeronautical Engineering, Shenyang 110136)

【Abstract】 Aiming at the question of low multicast efficiency and poor game states synchronization in the SCRIBE algorithm based on P2P for Massively Multiplayer Online Games(MMOG) Area Of Interest(AOI) multicast, an advanced multicast algorithm based on SCRIBE is presented. The algorithm mainly considers the P2P network terminal nodes in bandwidth and computing power on the heterogeneity and guarantees that all nodes have the same interest in the group. Simulation results show that the algorithm gains a greater improvement on reducing message redundancy and the latency of multicast and the synchronization of game states.

【Key words】 P2P network; Area Of Interest(AOI); SCRIBE algorithm; application-layer multicast

1 概述

传统的巨量多玩家网络游戏(Massively Multiplayer Online Games, MMOG)通常采用 C/S 模式, 全局游戏状态以集中的方式进行管理。该方式保证了系统的安全性和简单性, 但在玩家数量的伸缩性以及单点失效性上存在缺陷。通常可以通过集群技术或计算网格来解决服务器的伸缩性问题, 但必须提供大量的服务器来处理高峰的负载, 缺乏足够的灵活性。采用 P2P 覆盖网络技术, 通过多个玩家节点来解决 MMOG 中的全局游戏状态的管理是一种理想的选择。

采用 P2P 的 MMOG 设计的关键问题是如何进行兴趣域的划分和管理, 兴趣域是游戏中实体所关心的状态和事件的集合, 通过对兴趣域的划分可以避免某个玩家接收自己不关心的事件^[1], 从而有效地减少系统带宽消耗。所有兴趣相同的玩家可以看成一个多播组, 因此, 需要有效的多播策略来保证不同玩家之间游戏状态的一致性。

SCRIBE^[2]是建立在 Pastry 上的一种 P2P 覆盖网络多播算法。文献[3]提出了用 Pastry 和 SCRIBE 作为网络游戏中的多播算法, 但由于 SCRIBE 多播在设计时没有考虑到 P2P 网络中终端节点在带宽和计算能力上的异构性, 而这种异构性^[4]对多播的性能影响较大。另外, SCRIBE 对兴趣域不感兴趣节点(以下简称非兴趣节点)的接纳, 必然导致多播效率降低。

2 SCRIBE 算法的不足

2.1 非兴趣节点的加入

一般将每一个兴趣域作为一个多播组, 然后利用 SCRIBE 进行多播。当新的节点加入兴趣域时, SCRIBE 通过 Pastry 来为新节点寻找协调者, 由于 Pastry 的路由过程必然导致一些对该兴趣域不感兴趣的节点加入兴趣域, 因此这不但增加了多播的负担, 也降低了系统的性能。

2.2 节点异构性

SCRIBE 多播在设计的时候根本没有考虑到 P2P 网络中所有终端节点在带宽和计算能力上的异构性, 这必然会导致某些接入带宽小、计算能力弱的节点被动地去处理大量的连接和数据处理, 从而影响了游戏状态的同步, 导致游戏的可玩性降低。

3 SCRIBE 多播算法改进

3.1 Pastry 改进

为每个 Pastry 节点增加级别属性和孩子属性, 级别属性用以标志其强弱(0 最弱, 也称级别最低), 级别越高的节点, 在形成多播树时拥有的孩子越多。孩子属性存放形成多播树时最多拥有的孩子数。节点级别和孩子在节点加入时确定, 具体方法如下:

设新节点 X 为加入系统, 它首先连接系统中的某已有节点 S 。假定 S 的级别为 L_s , S 获取其带宽 B_s 和计算能力 C_s , 设 X 的带宽为 B_x , 计算能力为 C_x , 则可以估算 X 的级别为

$$L_x = \left\lceil L_s + \lg \frac{B_x C_x}{B_s C_s} \right\rceil$$

若 S 的孩子属性为 N_s , 则可以估算 X 的最大孩子数为 $N_x = \left\lceil N_s + \lg \frac{L_x}{L_s} \right\rceil$ 。

3.2 SCRIBE 改进

3.2.1 非兴趣节点的加入禁止

当某个节点要加入某个兴趣域时, 由 Pastry 路由由一条 CREAT 消息到协调者节点, 通过节点不断地加入从而形成一棵多播树。对于新加入节点, 如果路由到协调者的过程中

作者简介: 王 岩(1978 -), 女, 讲师, 主研方向: 分布式系统, 网络游戏; 李照奎, 讲师; 石祥滨, 教授

收稿日期: 2009-11-25 **E-mail:** lovelast@sina.com

都没有碰到树中节点,则直接把自己作为协调者的孩子。如果在路由过程中碰到某个节点是树中节点,则加入点作为该节点的孩子,路由过程终止。本算法最终目的是保证对该组不感兴趣的节点不落在树中。图1解释了如何禁止非兴趣节点的加入过程。

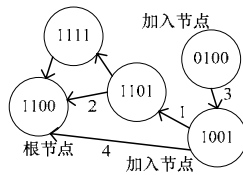


图1 非兴趣节点的加入禁止

假设图中圆代表兴趣域内的玩家节点,其中,节点1100是协调者节点。节点1001首先请求加入,箭头1表示1001初始的加入请求首先被路由到节点1101,1101不是树上的节点,它不用记录1001为孩子节点,箭头2表示被继续路由到该组的协调者节点1100,这时1100只把1001作为自己的孩子,并把自己的IP地址发送给1001,路由过程中的节点1101本身并不对该组感兴趣,故不加入该组。这时另一节点0100请求加入组1100,该消息首先被路由到1001,由于1001已经是树中的节点,因此1001把0100加入到自己的孩子节点,路由过程停止。当节点1001在组内多播消息时,它直接把消息发往协调者1100,如图中的箭头4。然后由节点1100将多播数据在组内散发,从而保证区域内游戏状态的一致性。由于这里并没有把路由过程中的点放入树中,因此减轻了多播的消息量。树中节点离开的处理方法等同于SCRIBE的处理。

3.2.2 节点异构性处理

为了处理P2P网络中所有终端节点在带宽和计算能力上的异构性带来的问题,本算法结合底层Pastry的改进来创建多播树。对于节点级别高的节点,所连接的孩子数就多,相反则少。

无论加入者直接把协调者节点作为父节点或是把某个路由过程中的节点作为父节点,都应先判断该父节点是否还能拥有更多的孩子(这里称还能拥有更多孩子的节点叫非满节点,已经不能拥有更多孩子的节点叫满节点)。如果该父节点为非满节点则直接把加入者作为其孩子节点,否则判断该父节点的原有孩子节点中有没有非满节点,有则加入,如果仍没有则继续判断孩子的孩子节点,最坏情况直到某个叶子节点终止。

图2解释了对节点异构性的处理过程。假定节点S1, S2, S3, S4, S6, S7, S8已经形成一颗树, S1为协调者节点, S5为加入节点,虚线1表示该节点S5的加入消息被路由到节点S1,如果S1是满节点,则通过消息2查找非满节点,如果仍没找到,则通过消息3,一直到消息4找到非满节点S8,实线5表示把S5作为S8的孩子节点。

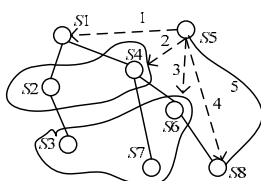


图2 节点异构性处理

图2只考虑了路由路径中遇到的第1个树中节点是协调者的情况。至于路由路径中某个非协调者节点已经是树中节点的情况与此类似,不做详解。节点主动退出时,该节点通知自己的孩子节点首先加入自己的双亲节点,具体加入过程等同于节点加入过程。

总之,通过对节点异构性的考虑,保证了每个多播树中的节点处理能力相对协调,提高了多播树的工作效率。

3.3 网络模型及问题定义

由于玩家有限的运动速度和感知能力,因此游戏中的玩家基于他们在虚拟世界中的位置形成了自组织的兴趣域。兴趣域是游戏中实体所关心的状态和事件的集合。通过划分兴趣域可以有效地减少整个系统的带宽消耗。兴趣域内的玩家通过多播机制将游戏状态的更新通知给同一兴趣域内的其他玩家以保持每个玩家局部拷贝的最新性。本文使用基于N-Tree的划分方法将游戏世界划分成多个子空间,每个子空间中的玩家节点具有相同的兴趣域。

设无向完全图 $G=<V,E>$ 代表整个Pastry覆盖网络,其中, V 为所有Pastry节点的集合; E 代表任意2点间的单播路径集合; $deg(V)$ 表示节点 V 的当前度数; NV 为节点 V 能容纳的孩子数, $V.Child[i]$ 表示节点 V 的第 i 个孩子节点。在Pastry覆盖网络中多播消息由源节点 $T \in V$ 生成,并发送给所有的兴趣域内多播组成员 $M \subseteq V$ 。这样问题就变成了如何在 G 中构建一棵以 T 为根多播生成树,令 h 为树中从消息源点到节点的跳数, K 为兴趣域内的节点数。定义多播的平均消息跳数 $ah = \frac{1}{n} \sum_{i=1}^n h_i (n = K - 1)$ 。在构建多播树的过程中保证避免消息冗余,同时最小化平均消息跳数 ah 。

3.4 算法描述

首先讨论具体的节点加入算法,当某个节点 X 加入以节点 T 为根的多播树时,它首先向节点 T 发送请求消息,该消息被Pastry向节点 T 路由,在路由过程中,请求消息遇到的第1个树中节点可能是根节点,也可能不是根节点。这里假定第1个遇到的树中节点为节点 R , Q 为存放节点的队列, $deg(R)$ 表示节点 R 的当前度数, N_R 为节点 R 能容纳的孩子数, $R.Child[i]$ 表示节点 R 的第 i 个孩子节点。具体算法如下:

```

If (R≠null) then do EnQueue(Q,R);
do
DeQueue(Q, S);
If(deg(S)<NS) then do
    Insert node X, X as child of node S,
    deg(S) ← deg(S)+1
    return;
Else do
    For (i←0 to deg(S)-1)
        If(deg(S.Child[i])<NP, P=S.Child[i]) Then do
            Insert node X, X as child of nodeP
            deg(P) ← deg(P)+1
            Else do
                If (S.Child[i]≠null, P= S.Child[i])
                    Then do EnQueue(Q,P);
                    While(QueueEmpty(Q))

```

下面讨论在 G 中构建一棵以 T 为根的多播生成树的多播算法。主要讨论对于兴趣域内的一个节点在接收到多播消息时的节点转发过程。这里 S 为产生多播消息的源节点, R 为多播树中的节点,接收来自根节点 T 的多播消息并进行转发。下面给出兴趣域内的节点多播消息算法的伪代码:

```

If (R≠null) then do EnQueue(Q,R);
do
DeQueue(Q, S)
  For (i←0 to deg(S)-1)
    If(deg(S.Child[i]) is active ,P=S.Child[i]) Then do
      Send message from S to P;
      EnQueue(Q,P);
  While(QueueEmpty(Q))

```

4 仿真与结果分析

本文用一台 PC 机进行仿真。机器环境：CPU 为 AMD Athlon(tm) 64 X2 Dual Core Processor 3800+，2 GB 内存。操作系统为 Microsoft Windows 2000 Server，SDK 为 SUN Jave2 (版本号 jdk1.6.0_02)，仿真器为 Peersim^[5]。Peersim 是 BISON 项目发布的对等网络仿真框架，通过 Java 语言模拟相关的 P2P 协议，它具有高扩展性、动态性和可配置性，支持周期驱动模拟和事件驱动模拟。本实验采用事件驱动模拟实现算法的改进。实验采用的网络拓扑为 ER 随机图模型，在该模型中每个节点随机产生 10 条边。底层传输协议采用 Peersim 提供的可靠传输，该协议用随机延迟进行消息的可靠递送。相关实验参数通过配置文件提交。

本文将虚拟的 MMOG 世界划分成 $M \times N$ 的网格，具体的划分方法已经超出本文的讨论范围，属于应用层的实现问题。每个矩形域代表一个兴趣域，将 Pastry 节点随机地分配到这些矩形域中。只考虑玩家移动所引起的多播情况，多播的消息内容主要为位置的更新。由于暂时不考虑组间通信的问题，因此假设每个玩家在随机移动的同时，不会超出他们的兴趣域。玩家移动采用时间随机概率模型。在 MMOG 中玩家主要和邻接节点通信，这种局部通信通常集中在小范围内，这里令每个兴趣域内的节点数量低于 200 个。

4.1 多播树中感兴趣节点比率

SCRIBE 算法并没有考虑是否感兴趣的节点都加入到多播树中，这无疑增加了多播的负担。改进后的算法禁止了非兴趣节点的加入，从而保证多播树中的节点都是感兴趣的节点，避免产生多余的多播消息。本实验重点测算在多播树中感兴趣的节点所占树节点的比率。图 3 展示了不同大小多播树下感兴趣节点所占比率，从图中可以看出改进后的算法，保证加入的节点都是感兴趣的节点。

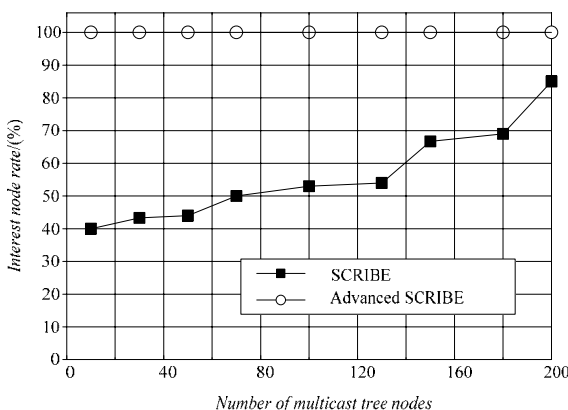


图 3 多播树中感兴趣节点所占比率

4.2 平均延时时间

由于禁止了非兴趣节点的加入，因此减少了多播的消息总数，同时又考虑了对节点异构性的处理，保证了多播树中的节点处理能力相对协调，提高了多播树的工作效率。图 4

展示了改进前后算法的平均延时时间的不同。这里假定消息在底层传输时的最大和最小延迟分别是 900 ms 和 500 ms。

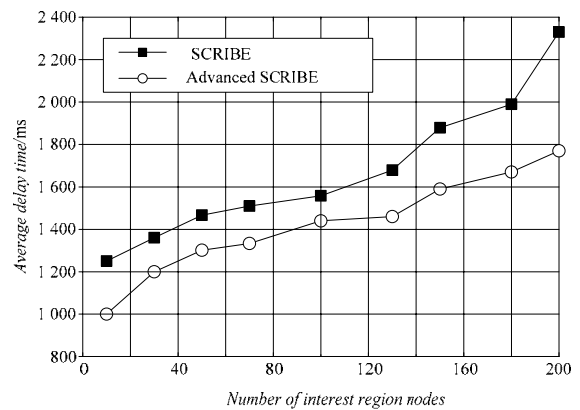


图 4 平均延时时间

4.3 平均消息跳数

平均消息跳数也从某种程度上反映了多播算法的效率。图 5 展示了改进前后算法的平均消息跳数的不同。从图中可以看出，改进后的算法消息跳数相对减少。

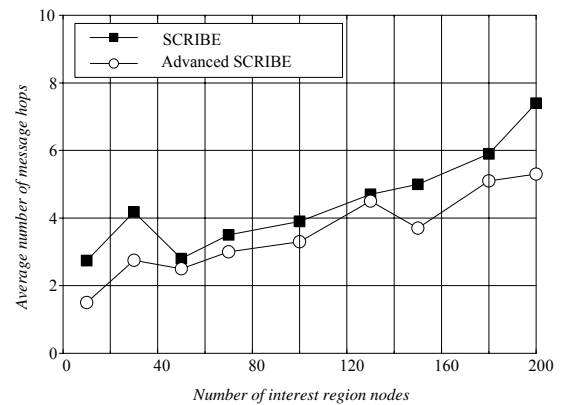


图 5 平均消息跳数

5 结束语

本文提出一种改进的 SCRIBE 算法来解决基于 P2P 的 MMOG 的多播问题，该算法减轻了多播的消息总量，保证了每个多播树中的节点处理能力相对协调，提高了多播的工作效率。由于 P2P 网络的动态性，因此如何保证兴趣域内节点的可靠性将是下一步研究的重点。

参考文献

- [1] Benford S, Greenhalgh C, Rodden T, et al. Collaborative Virtual Environments[J]. Communications of the ACM, 2001, 44(7): 79-85.
- [2] Castro M, Druschel P A. Scribe: A Large-scale and Decentralized Application-level Multicast Infrastructure[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(8): 1489-1499.
- [3] Knutsson B, Hopkins B. Peer-to-Peer Support for Massively Multiplayer Games[C]//Proc. of INFOCOM'04. Hong Kong, China: [s. n.], 2004: 96-107.
- [4] 杨峰, 郑纬民, 余宏亮, 等. 基于 P2P 的应用层组播结构研究[J]. 计算机工程, 2007, 33(15): 115-117.
- [5] Jelastiy M, Montresor A, Jesi G P. PeerSim: A Peer-to-Peer Simulator[EB/OL]. (2008-05-20). <http://peersim.sourceforge.net/>.

编辑 索书志