

文章编号:1001-1595(2009)03-0236-06

## 基于主动缓存的 P2P 海量地形漫游瓦片调度算法

潘少明<sup>1</sup>, 喻占武<sup>1</sup>, 李 锐<sup>1</sup>

1. 武汉大学 测绘遥感信息工程国家重点实验室, 湖北 武汉 430079

## Tile Scheduling Algorithm Based on Active Cache in P2P Massive Terrain Navigation

PAN Shaoming<sup>1</sup>, YU Zhanwu<sup>1</sup>, LI Rui<sup>1</sup>

1. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

**Abstract:** With continuous development of computer technology, 3D terrain navigation is a hot application in network Geographic Information System (GIS), and massive data transmission is highly demanded. As the number of clients increases, the server performance will drop too rapidly to satisfy the client real-time rendering requirement because the server loads are increasing linearly for the kind of client/server architecture. The method through Peer-to-Peer (P2P) to address the server bandwidth and performance bottleneck in large scale massive terrain navigation has attracted much effort based on P2P networks. Due to peers dynamics and heterogeneity, some peers become the hot-pots, while some others are idle. An active cache, a kind of terrain data scheduling algorithm based on grouped locality exiguity, is proposed. The scheduling algorithm, called GLRF, is proposed for the idle peers, which just helps others getting their data and does not play the tile data. When deciding which segments to get to improve the peer hit rate and peer matching rate, in addition to the rarity and uniformity considered in existing data algorithms. Compared test result in prototype system indicates that the proposed algorithm has high tile request hit ratio and match ratio, and the two metrics have high increase speed than traditional method.

**Key words:** active cache; group; LRF; P2P

**摘 要:** 针对利用 P2P 技术解决大规模海量地形漫游中服务器带宽和性能瓶颈问题产生的部分节点成为热点,而部分节点闲置的问题,提出一种基于分组局部稀少性的主动缓存地形数据调度算法 GLRF,闲置节点自身并不显示瓦片,只为其他节点提供服务。该算法充分利用节点的闲置资源,通过快速均匀分布瓦片以提高瓦片请求时的节点命中率和节点匹配率。通过实验原型系统的比较测试表明,该算法瓦片请求节点命中率和匹配率更高,且提升速度更快。

**关键词:** 主动缓存; 分组; LRF; P2P

**中图分类号:** P208      **文献标识码:** A

**基金项目:** 国家 973 计划(2004CB318206); 国家 863 计划(2007AA12Z214); 测绘遥感信息工程国家重点实验室专项科研经费资助项目

### 1 相关工作

利用 P2P 技术来解决海量地形漫游中的服务器性能瓶颈问题,成为研究热点<sup>[1-4]</sup>,主动缓存就是利用 P2P 节点的闲置资源(带宽、CPU),主动从其他节点下载使用价值较大的瓦片,这有 2 个目的:① 通过一定的策略,在相对空闲时(如请求服务队列较小),主动从中心节点下载普通节点没有的瓦片,提前为 P2P 共享准备数据,平滑和间接降低对中心节点的峰值负载;② 从普通邻居节点下载相对“稀少”的瓦片,为 P2P 共享提供更多的节点选择,由于有更多的节点选择,能有效分担下载服务,可以减少部分“好”节点成为“热点”

的可能。

有效的主动缓存策略实际上包含 2 个方面的问题:① 如何选择下载的目标对象,即要选择今后“有用性”高的瓦片进行下载;② 如何选择下载的节点源,即所选择下载的节点是最能提供服务的。

针对第一个问题,基于海量地形漫游方面研究不多,相关文献很少,孙立峰等的实时漫游与调度策略主要从本地数据预取调入内存运算<sup>[5]</sup>,并没有考虑主动从服务器获取数据缓存机制,在视频点播领域,国内外有许多文献加以研究,包括随机获取(Random)和局部稀少获取(Local Rare First,LRF)等。随机获取是指,节点首先统计自

己缺少的即可以下载的数据块数目  $n$ , 然后每块数据按照  $1/n$  的概率获取, 这种方法没有考虑数据的均匀性, 不如 LRF 好<sup>[6]</sup>。LRF 是一种启发式的算法, 首次由 Byers 等提出并进行系统分析<sup>[7]</sup>, Tian 等进一步对其进行了改进以提高带宽利用率<sup>[6]</sup>, 其主要思想是依据邻居节点信息获取数据块的分布情况, 然后根据局部稀少的原则获取那些自己看起来很稀少的数据。根据极大似然法则, 每个节点看来比较稀少的数据块, 也应该是全局稀少的, 这样能减少有带宽没数据传输的情况, 提高数据传输效率。但这些算法都没有考虑地形漫游的特点, 地形漫游中最稀少的并不一定是最“有用”的, 比如全球影像数据中的海洋区域瓦片和城市区域瓦片, 其“热”度并不一样。

针对第二个问题, 主要是从服务质量(如带宽等)方面选择, 包括依据节点历史服务质量选择服务节点<sup>[1]</sup>, 以及根据 IP 地址分类, 依据 IP 地址同类的原则选择服务节点等<sup>[8]</sup>。由于网络节点的差异性, 这很容易导致某些“好”的节点拥塞而出现新的性能瓶颈。

本文针对已有调度算法存在的问题, 提出一种新的基于分组的局部稀少性瓦片选择策略, 同时考虑节点服务质量和负载的节点选择机制, 并给出相关分析和算法模型, 最后对算法开发了原型系统进行检验。

## 2 基于主动缓冲的瓦片数据调度算法

### 2.1 主要思想

如图 1 所示是一种混合式的 P2P 海量地形漫游共享模型<sup>[1]</sup>。每个节点依据其当前漫游的区域加入组(中心服务节点属于特殊组成员, 属于所有组), 节点都有自己组内的邻居节点列表, 每个节点都会在每隔一个调度周期(Scheduling Period)向组内其他节点(不包含中心节点 0)发送本地瓦片信息。这种信息称为缓存映射(Buffer Map, BM), 通过交换缓存映射(为了降低数据量, 采取更新交换的原则, 即只交换变化的部分, 无变化则不交换), 组内所有节点都可以通过邻居列表了解邻居的已有数据的分布情况。每个节点根据这些邻居节点的信息, 从瓦片与节点选择 2 个方面加以考虑: ① 依据组内邻居节点信息, 在节点空闲时, 选择兼顾局部稀少性、全局域属性、且本节点不拥有的瓦片主动下载; ② 考虑节点带宽和负载, 选择合适的节点下载瓦片, 条件相同时, 随

机选择。区别于 LRF, 称该算法为 GLRF(Group & Local Rare First)。

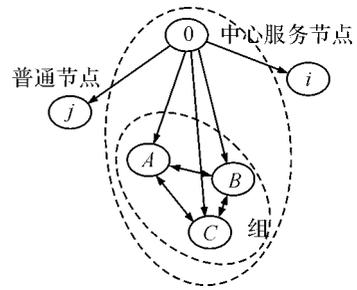


图 1 一种混合式的 P2P 海量地形共享模型

Fig. 1 A hybrid P2P massive terrain sharing model

首先给出算法中缓存映射 BM 的结构和定义。

不失一般性, 设区域  $G$  内某组的节点总个数为  $M$ , 该域内地形数据瓦片总块数为  $S$ , 节点  $i$  称为  $m_i$ , 瓦片  $i$  称为  $p_i$ , 节点  $i$  由一个三元组  $\langle b_i, q_i, u_i \rangle$  描述, 其中  $b_i$  是带宽, 由各个节点自行估计<sup>[9-10]</sup>,  $q_i$  是服务请求队列长度(表明负载大小),  $u_i$  是数组(指针), 用于存放是否拥有某块瓦片, 即  $u_i^k$  表示节点  $i$  是否拥有瓦片  $k$ ,  $u_i^k = 1$  表示拥有瓦片, 为 0 则不拥有; 缓存映射 BM 则是由不同节点的三元组组成的内存块, 按照  $\langle i, \langle b_i, q_i, u_i \rangle, j, \langle b_j, q_j, u_j \rangle, \dots \rangle$  的形式组织。

### 2.2 节点加入组策略

海量地形漫游中, 瓦片数量巨大, 以全球 SRTM90 地形数据为例, 共有 3 538 890 块瓦片。简单地利用局部稀少的原则进行主动缓存, 将可能导致下载的是完全不需要的瓦片, 不但对组内节点数据共享没有贡献, 反而容易增加节点负载。因此, 需要在下载瓦片时, 限制选择范围, 使所选择的瓦片对组内共享是“有用”的。

考虑地形漫游的特点, 根据如图 2 所示的 GIS 地形数据瓦片分层规则, 设某层  $l$  中某瓦片  $p_i$  的坐标值为  $(x_i, y_i)$ , 按照金字塔模型对应的最低分辨率(定义为 0 层)瓦片的坐标为  $(x_0, y_0)$ , 则有

$$\left. \begin{aligned} x_0 &= \frac{x_i}{2^l} \\ y_0 &= \frac{y_i}{2^l} \end{aligned} \right\} \quad (1)$$

定义地形漫游中当前漫游区域满足式(1)的节点属于同一区域, 并赋予相同域号  $G$ , 域内的所有节点依据随机加入的原则组成多个组, 该区域

内不同的组在相同的区域漫游,有共同的全局域属性  $G$ 。

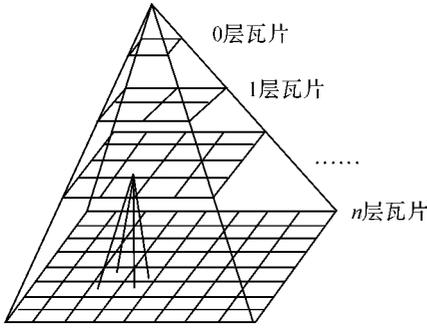


图2 GIS地形数据金字塔模型

Fig. 2 Pyramid model for GIS terrain data

### 2.3 瓦片选择算法

一个组内,与标准 LRF 类似,考虑局部稀少的数据,同时兼顾全局域属性以限制瓦片选择,对某瓦片  $p_i$  定义 TSC(Tile Spreading Capability) 作为选择依据,TSC 值越大,瓦片越被优先主动缓存。分别用  $LRF(p_i)$  表示瓦片  $p_i$  的局部稀少性, $G(p_i)$  表示瓦片  $p_i$  的全局域属性,将地形服务器看成特殊节点 0,对任意瓦片  $k$ ,设为  $u_0^k=1$ ,则有

$$LRF(p_i) = \frac{\sum_{k=0}^M (1 - u_k^i)}{\sum_{k=0}^M u_k^i} \quad (2)$$

$$G(p_i) = \begin{cases} 1 & p_i \in G \\ 0 & p_i \notin G \end{cases} \quad (3)$$

在式(2)中,分子表示所有邻居节点中(包含自己)不拥有瓦片  $p_i$  的节点数量,分母表示拥有瓦片  $p_i$  的节点数量,则  $LRF(p_i) \in [0, M-1]$ 。对 TSC 作如下定义:

$$TSC(p_i) = LRF(p_i) + M \times G(p_i) \quad (4)$$

式(4)表明,① 组内瓦片的 TSC 值总是大于组外瓦片的 TSC 值(至少大 1); ② 对于组内任意一块瓦片数据,拥有瓦片的节点越少,其 TSC 值越大,被选择进行主动缓存的机会越大。对组内存在 TSC 相同的瓦片时,随机选择其中一个。因此式(4)是对 LRF 的改进,它不但考虑稀少性,同时兼顾组内共享的“有用”性,避免下载对本组无共享性的瓦片,更加符合地形漫游的特点,在海量地形漫游中性能更优。

### 2.4 节点选择算法

由于网络中节点的动态性和差异性,选择一个“合适”的节点进行瓦片数据的下载,必须在充

分考虑下载速度的同时,尽量降低对“好”节点的负载,这主要基于如下 3 点:① 拥有更大带宽的节点拥有更好的服务质量,能提供更多的下载服务和下载速度,但同时更多的服务请求反过来又降低了节点的服务质量,所以,② 节点的服务请求队列越短,等待服务的时间延迟将越小,能更快的响应服务请求,同时,③ 节点拥有的瓦片数量越多,被请求提供服务的“次数”将越“频繁”,成为潜在“热点”的可能性越大。

对某瓦片  $p_i$  和节点  $m_k$ ,定义 PPC (Peer Providing Capability) 作为节点选择依据,PPC 最大的节点被选择提供下载服务,如果存在相同 PPC 的节点,则从中随机选择。对此,节点  $k$  对瓦片  $p_i$  的 PPC 值为

$$PPC(p_i, m_k) = \frac{b_k \times u_k^i}{S \times q_k + \sum_{j=1}^S u_k^j} \quad (5)$$

式(5)中,  $S \times q_k$  代表服务排队队列大小(放大  $S$  倍,目的是和请求瓦片数量平衡大小),  $\sum_{j=1}^S u_k^j$  代表可以被请求的瓦片数量,服务排队队列越长,节点负载越大。可以被请求的瓦片数量越多,成为“热点”的可能性越大。算式表明,在相同条件下,优先选择负载小、成为热点可能性小,且服务带宽越大的节点提供下载服务(在拥有该瓦片的前提下)。

### 2.5 算法调度流程

图 3 所示为一个调度的例子,  $M=4, S=10$ , 若  $b_k=10$  kbps,  $q_k=0$ , 其中 ( $k=0, 1, 2, 3$ ), 阴影部分表示节点拥有该瓦片,中心服务节点拥有所有瓦片。任意节点将优先选择组内局部稀少的瓦片。根据计算 TSC 值和 PPC 值,节点  $m_2$  将从  $m_0$  下载瓦片 3,  $m_3$  将从  $m_0$  下载瓦片 7,  $m_1$  将从  $m_2$  下载瓦片 8,  $m_3$  将从  $m_2$  下载瓦片 5,  $m_2$  将从  $m_3$  下载瓦片 9,  $m_1$  因为拥有较多的瓦片,被选择作为下载源的可能性降低,有利于降低其成为潜在“热点”的可能性。具体算法调度流程如下:

//节点  $m_k$  的调度流程

1. OnSchedule() { // 在一个调度周期内处理
2. Broadcast(BM); // 广播自己的缓存映射
3. // 1. 先计算 TSC 值
4. for each  $p_i$  in  $S$  do // 轮询所有瓦片
5. if ( $p_i \notin m_k$ ) // 只处理没有的瓦片
6. list[i][0] = TSC( $p_i$ ) // 计算 TSC 值并存入列表

```

    list[i][1]=i //对应保存瓦片索引
7.   end if
8.   end for
9.   sort(list) //列表元素按从大到小排序
10.   $p_{select} = list[0][1]$  //TSC 最大的瓦片被
选择
11. //2. 再计算 PPC 值
12. for each  $m_j$  in  $M$  do
13.   if( $m_j \neq m_k$ )
14.    list[j][0]=PPC( $p_{select}, m_j$ )
15.    list[j][1]=j //计算 PPC 值并
保存
16.   end if
17. end for
18. sort(list)
19.  $m_{select} = list[0][1]$  //PPC 最大的节点
被选择
20. Download( $p_{select}, m_{select}$ ) //从选择的节
点下载瓦片
21. }
```

该算法首先通过 2 次计算后得到需要下载的瓦片  $p_{select}$  和选择的下载节点  $m_{select}$ , 然后通过 P2P 的方式进行主动下载缓存。

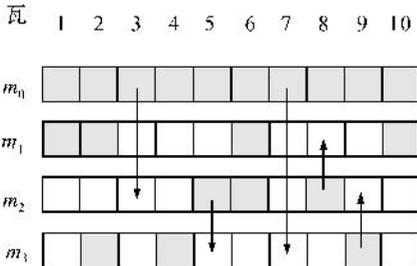


图 3 一个主动缓存调度的例子

Fig. 3 An active cache scheduling example

## 2.6 缓存映射对性能的影响分析

全球 SRTM90 地形数据共 3 538 890 块瓦片, 每块瓦片数据大小约为 44 KB。三元组  $\langle b_i, q_i, u_i \rangle$  中, 按照更新交换的原则, 以平均 512 Kbps (普通电信 ADSL 用户) 的终端带宽作为计算依据, 每秒最多可下载的瓦片数为 1.5 块, 每块瓦片更新占用的字节数为 4 字节+1 位(分别表示地址和数据位, 数据位为 1 表示新拥有该瓦片, 否则为新删除该瓦片), 则最多产生 49.5 bps 的  $u_i$  数据。按照本地缓冲区满状态考虑(一般定义为 1 GB), 新下载一块瓦片, 则对应的应该删除一块

瓦片, 则最大产生的流量为 99 bps。另外一次更新中  $b_i$  和  $q_i$  分别固定占用 4 字节, 由于  $b_i$  变化小, 不需要经常更新, 故基本可以忽略。如此, 一个节点产生的更新数据量最大约为 131 bps, 按照每组节点成员 50 个计, 组内产生的网络流量也只有 6.4 Kbps, 这相对于海量地形漫游时产生的网络流量, 对性能的影响是有限的。考虑到有部分节点处于空闲状态, 不需要发送 BM 信息, 这个估计还是趋于保守的。

显然, 当组内成员扩大到 500 个时, 组内产生的网络流量将达到 64 Kbps, 这将对地形漫游产生影响, 因此, 组内成员的个数也不能无限制扩大。

## 3 实验与结果分析

考察 2 个指标: ① 节点命中率, 是指某个节点在地形漫游时, 请求的瓦片能够从组内普通节点获取的次数与请求的总次数的比率, 节点命中率越高, 能够通过 P2P 共享下载的几率越大, 越容易降低对中心服务器的负载; ② 节点匹配率, 是指节点请求瓦片时, 组内拥有该瓦片的节点个数与组内总节点个数的比率, 显然节点匹配率越高, 请求下载的源节点越多, 越容易降低对部分“好”节点的负载, 减少“好”节点成为新的性能瓶颈的风险。

利用所开发的基于 GlobeSight 的 P2P 原型系统 PeerVOLT<sup>[1,11]</sup>, 对客户端进行主动缓存改造, 终端普通节点实现在空闲时的主动瓦片数据备份, 提高瓦片的缓存率。以最高分辨率为 90 m 的全球 SRTM90 地形数据为实验数据(数据块大小为  $128 \times 128$  像素, 每块传输瓦片数据大小约 44 KB, 共 3 538 890 块瓦片)。在一个局域网内, 组织 10 台 P4 计算机, 分别运行普通节点终端和中心服务节点, 组成一个组(Group),  $M$  为总节点数,  $S$  为总瓦片数已知, 设置普通节点带宽相同, 为 256 Kbps, 中心服务节点带宽为 10 Mbps, 分别利用原型及其改进进行地形数据下载。

### 3.1 算法性能对比实验

分别对采用 GLRF 算法和不采用 GLRF 算法模型进行实验, 记录对比实验结果。

1. 考察某块区域, 如湖北省, 域内的瓦片数量为 21 845 块, 收集实验数据前, 各个终端先随机对该区域进行“浏览”, 形成本地初始缓存映射信息 BM。

2. 各个节点按照随机规律进入“浏览”和“空闲”状态, 以下载 20 个数据块为一个调度周期进

行状态选择,在“浏览”时,终端在“湖北”区域内随机选择一个瓦片地址开始漫游,行走路线(即下一瓦片地址)也按照随机规律生成,当“下一个瓦片”超出“湖北”区域范围时,重新随机选择一个起点。

3. 节点在“空闲”状态下,分别按照 GLRF 算法进行主动下载备份或不进行处理。

4. 节点在“浏览”状态下,每次依据 BM 信息与所随机选择的瓦片进行比较,记录请求时组内节点包含该瓦片的次数  $N'$  和请求总次数  $N$ , 计算节点命中率  $f_{hitpeer} = N'/N \times 100\%$ , 结果取重复运行 10 次的平均值,图 4 给出了不同时间点的节点命中率对比测试结果及其拟合曲线。拟合方程分别为

$$\left. \begin{aligned} F_{hit无} &= 0.45672 \times t + 1.2490 \\ F_{hit有} &= 0.63579 \times t + 1.2917 \end{aligned} \right\} \quad (6)$$

5. 节点在“浏览”状态下,记录请求瓦片时包含该瓦片的节点个数  $M'$  和总节点个数  $M$ , 计算节点匹配率  $f_{matchpeer} = M'/M \times 100\%$ 。图 5 给出了不同时间点节点匹配率对比测试结果及其拟合曲线。拟合方程分别为

$$\left. \begin{aligned} F_{match无} &= 0.05251 \times t + 9.9898 \\ F_{match有} &= 0.07932 \times t + 9.9410 \end{aligned} \right\} \quad (7)$$

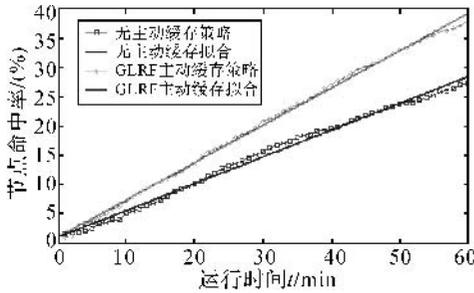


图 4 不同策略下节点命中率对比测试结果  
Fig. 4 Peer hit rate in different polices

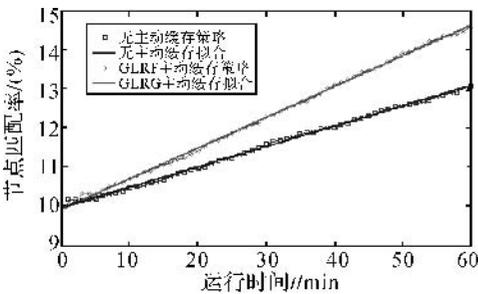


图 5 不同策略下的节点匹配率对比测试结果  
Fig. 5 Peer matching rate in different polices

对比测试表明,在基于 GLRF 的主动缓存调度下,由于部分空闲的节点进行瓦片主动下载缓

存,组内能够更快地将稀少和有用的数据提前获取,随时间变化,空闲节点获取的数据更多,节点的命中率和匹配率提升得越快。从式(6)和式(7)可以看出,节点命中率的斜率比为 1 : 1.39,节点匹配率的斜率比为 1 : 1.51,这表明,采用 GLRF 算法,节点命中率和节点匹配率的提升速度要比原方法分别快 39%和 51%左右。

### 3.2 节点规模 M 对算法性能影响实验

分别将总节点个数增加为 20 个、30 个和 40 个,分别记录不同节点规模下的节点匹配率和节点命中率,图 6 和图 7 分别给出了不同节点规模下的对比测试结果。

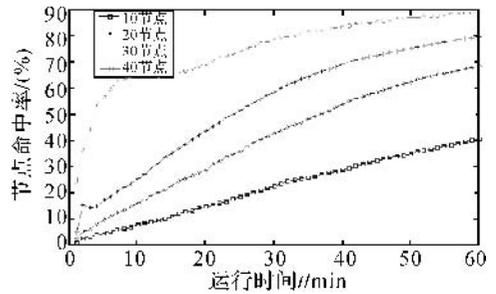


图 6 不同节点规模下的节点命中率对比测试结果  
Fig. 6 Peer hit rate in different node scale

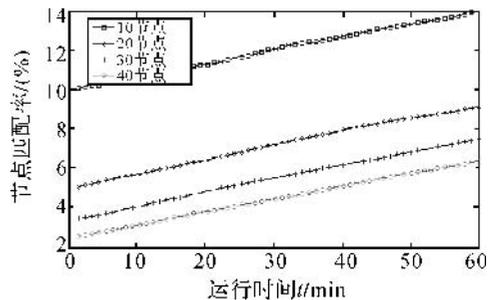


图 7 不同节点规模下的节点匹配率对比测试结果  
Fig. 7 Peer matching rate in different node scale

不同节点规模的对比测试表明,随着节点规模的扩大,节点命中率提升更快,但相对来说,总体的节点匹配率却基本上按比例下降。显然,节点规模越大,瓦片数据的均匀化更快,命中率的提升速度也就越快,但瓦片数据的均匀性的提升,并不需要增加大量的副本,在瓦片数据均匀的前提下,相同运行时间内不同节点规模的数据副本数量基本上是一致的,而相同副本数和不同的节点规模,导致对应的匹配率的下降。测试表明,GLRF 算法能够快速的保证瓦片数据域内的均匀性,更加有利于地形漫游时降低对中心服务器的

负载,同时,在满足 BM 信息更新对性能影响限制条件下,可以尽量扩大组内节点规模。

### 3.3 节点动态性对算法性能影响实验

考虑 P2P 网络的动态性,设置节点规模为 30,以 120 s 为一个调度周期,按照 20% 的比率随机选择节点上线和下线进行对比测试。图 8 和图 9 分别给出了 20% 动态性下的对比测试结果。

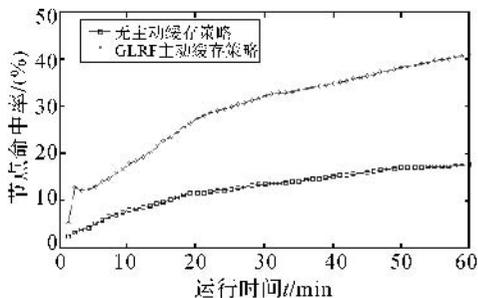


图 8 动态性下的节点命中率对比测试结果

Fig. 8 Peer hit rate in dynamic network

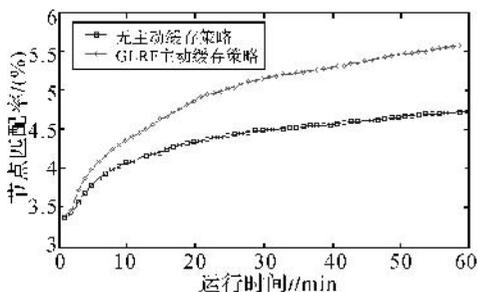


图 9 动态性下的节点匹配率对比测试结果

Fig. 9 Peer matching rate in dynamic network

测试结果表明,动态性下的 GLRF 算法仍然能够保证比较好的节点匹配率和节点命中率,而且,相对于静态节点模式,节点匹配率和命中率提高更多,性能更优,特别是节点命中率,优势明显。这表明,GLRF 算法能够迅速下载节点下线造成的瓦片数据的空缺,保持数据的均匀性,更加适合 P2P 网络。

### 3.4 LRF 和 GLRF 算法性能对比分析

LRF 和 GLRF 都是采用的局部稀少获取策略,但对于地形漫游系统,由于瓦片数量巨大,并且不同于视频点播等应用,还存在一些“无效”瓦片(如海洋、湖泊等)。采用传统 LRF 算法时,由于“无效”瓦片在组内相对来说是最稀少的,所以获取的往往是这些“无效”瓦片。同时,通过 LRF 算法获取时还要产生额外的消耗(带宽消耗、计算消耗),因此,这种获取对地形漫游是无益的。

同样可知,在“湖北”进行地形漫游时,也不需要“纽约”的瓦片,因此,“纽约”的瓦片相对于这次“漫游”来说,也是“无效”的,从这一方面来说,GLRF 也能避免这些“无效”瓦片的下载,从而提高效率。

基于以上原因,简单的利用 LRF 对于地形漫游系统是不利的,因此,本文并没有给出 LRF 算法的性能实验结果。

## 4 结 语

利用 P2P 进行海量地形数据传输时,最大限度地降低对中心服务节点的峰值负载是关键。如何充分利用节点闲置资源,采取主动备份策略,提前下载“有用”的瓦片进行储备,是解决普通节点有带宽、没瓦片和中心服务节点有瓦片、没带宽的桥梁。利用 P2P 进行海量地形漫游共享服务作为 GIS 领域的一个新兴课题,与此相关的文献还相对较少。本文借鉴视频点播领域的 LRF 算法,提出的基于稀少性和全局域属性的 GLRF 算法,能较好地解决当前通过 P2P 进行海量地形下载漫游的带宽和瓦片数据矛盾问题。实验原型测试结果表明,算法对提高瓦片请求节点命中率有效,性能更优。下一步工作包括研究如何利用节点闲置资源,通过“代理”的方式提高节点在地形漫游中的服务质量。实际的 GIS 系统中,由于存在缓存大小限制,对于节点缓存满状态,如何根据瓦片流行度和使用率选择被置换的瓦片数据也是下一步需要考虑的问题。

## 参考文献:

- [1] YU Zhanwu, ZHENG Sheng, LI Zhongmin. A Large-scale Terrain Transmission Mechanism Based on Hybrid P2P[J]. Acta Geodaetica et Cartographica Sinica, 2008, 37(1): 243-250. (喻占武, 郑胜, 李忠民. 一种混合式 P2P 下的大规模地形数据传输机制[J]. 测绘学报, 2008, 37(1): 243-250.)
- [2] MA Xiujun, LI Chenyu, XIE Kunqing, et al. A Peer-to-Peer Based Data Manipulating and Distributing Technology [J]. Geography and Geo-information Science, 2006, 22(4): 16-19. (马修军, 李晨煜, 谢昆青, 等. 基于 P2P 的空间数据操作分布技术研究[J]. 地理与地理信息科学, 2006, 22(4): 16-19.)
- [3] XU Jing, LI Chunhua, CHEN Tianzi. GIS Data Transportation Based on P2P[J]. Computer Engineering and Design. 2007, 28(12): 2851-2853. (徐静, 李春华, 陈天滋.

(下转第 249 页)