

无线环境中多数据项广播调度算法综述

张卓瑶 孙未未 余平 覃泳睿 施伯乐

(复旦大学计算机与信息技术系 上海 200433)

摘要 数据广播是无线环境中信息传递的主要方式,其中多数据项广播是一个重要且具有挑战性的研究热点。对多数据项广播调度算法进行了综述。分类介绍了各种典型算法,并做了系统的比较和分析,最后讨论了该领域研究的未来发展方向。

关键词 无线网络,多数据项广播,调度算法

中图法分类号 TP311

Survey of Broadcast Scheduling for Multi-item Queries in Wireless Environment

ZHANG Zhuo-yao SUN Wei-wei YU Ping QIN Yong-rui SHI Bo-le

(Department of Computing and Information Technology, Fudan University, Shanghai 200433, China)

Abstract Broadcasting is widely used in wireless environment for data dissemination. Broadcasting for multi-item queries is one of the most important and challenging issues in the area. This paper gave an overview of the multi-item broadcasting. It introduced and compared some typical scheduling algorithms designed for multi-item broadcasting by category. Finally, it discussed the open issues and promising directions in this area.

Keywords Wireless network, Broadcasting for multi-item queries, Scheduling algorithm

1 引言

无线数据广播是移动数据管理的一个重要研究分支,通过向空中广播数据,可以支持大量移动计算设备并发访问服务器上的数据,发送代价与接收者个数基本无关;利用无线信道的天然广播特性,提高了系统的可伸缩性,其研究的核心是服务器广播信道的组织和调度^[1,2]。

无线数据广播体现无线通信的无线、低带宽、非对称、考虑节能、支持用户移动和广播特性等主要特点^[2,3],是一种新型数据访问方式,一直受到重点关注。数据库和移动计算这两个领域,最重要的会议和期刊均发表了多篇研究无线数据广播的论文。同时,无线数据广播技术有着很好的应用前景,在消费电子、公共信息的发布、军事等很多领域已有实际的应用。而无线局域网的高速推广和 3G 网络的部署,将为无线数据应用提供更好的基础网络平台。

多数据项广播是具有重要理论意义和广阔应用前景的一个研究方向,近年来得到了学术界的广泛关注。已有许多研究者提出了一批有效的解决方法,但目前还未见对该问题的系统分析和综述。

本文首先介绍了数据广播的相关背景知识;其次分析了多数据项广播的主要特点;然后重点分类介绍了各种典型的多数据项广播调度算法,并进行了比较和分析;最后讨论了该领域研究的现存问题和未来发展方向。

2 无线数据广播基础知识

无线数据广播是目前无线移动网络中广泛采用的数据传

递方法,广播服务器通过公共的信道发送信息,用户在广播信道上侦听并及时将自己感兴趣的数据下载到本地。从用户角度,无线信道可看成是存储数据的载体,和可随机访问的内存和硬盘不同,无线信道中的数据只能顺序访问^[2]。

2.1 无线数据广播的模式

按照调度模式不同,无线数据广播可分为以下 3 类^[4]。

周期广播模式(Broadcast Mode/Push-based Mode):服务器端将存储的数据以一种确定的调度方式在广播信道上循环广播,用户端只需要在广播信道上侦听,一旦发现自己感兴趣的数据则下载到本地。

On-demand 广播模式(On-demand Mode/Pull-based Mode):用户通过上行信道将自己的请求发送给服务器端,服务器根据所有用户的请求状况安排调度。

混合模式(Hybrid Mode):混合模式^[5]对于热门数据采用周期广播模式而对于不热门的数据采用 On-demand 模式。用户首先在广播信道上侦听,一旦发现自己需要的数据时,则将自己的请求显式地发送给服务器,然后在 on-demand 信道上等待需要的数据。

2.2 无线数据广播性能的评价标准

如何快速访问和节省能源,是无线数据广播中研究的两个主要问题^[2]。相应地,有两个评价广播性能的主要参数:访问时间(Access Time)和调谐时间(Tuning Time)。

访问时间:从用户提出请求到请求得到满足之间经过的时间。

调谐时间:用户在提出请求到请求得到满足之间需要保持侦听状态的时间。在广播信道中插入索引,是减少调谐时

间的主要方法。如何设计高效率的索引也得到了很多研究者的关注^[6,7]。

其他标准包括失效率、带宽利用率和服务器端的计算量等也是考察广播系统设计合理与否的依据。

2.3 广播调度

广播调度策略是影响无线数据广播系统性能的关键因素。广播模式中一个完整的数据广播序列称为一个广播周期 (Broadcast cycle), 如何合理地安排一个广播周期中每个数据出现的频率以及这些数据之间的顺序, 使平均访问时间最少, 是数据广播中的主要研究问题之一。对于单数据项广播 (每次用户只请求一个数据) 的调度方法包括: 平坦调度 (Flat Broadcast)^[4]、基于访问概率的调度 (Probabilistic-based Broadcast)^[8]、多盘调度 (Broadcast Disks)^[9] 等。S. Hameed 和 N. H. Vaidya 在 1999 年提出了平方根理论^[10], 并从理论上证明了满足该理论的调度可以达到最优的平均访问时间。

2.4 多数据项广播

现实中, 很多情况下我们需要同时请求多个数据。例如用户想要知道最近几天的天气情况, 或者是要同时了解几只股票的行情。Web 页面请求是另一个典型的多数据项广播的例子, 通常一个 Web 页面上都嵌有许多其他数据信息, 如图片、视频等, 用户请求该页面的同时, 应当也把这些数据下载下来。此时, 相比于同时发送多个单数据项请求, 发送一个多数据项请求对用户来说是更自然和有效的方式。这就对服务器端提出了新的要求, 即如何设计一个合理的调度算法以尽快满足大多数用户的多数据项请求。

2.5 多数据项广播调度与单数据项广播调度的区别

在多数据项广播中, 主要的性能评价标准依然是访问时间, 它指的是从用户提出请求到用户所请求的所有数据全部获得之间经过的时间^[11]。绝大部分单数据项调度算法都基于一个基本的原则, 即数据项的访问频率越高, 广播的频率也越高。而多数据项请求具有的特点显示不适合直接采用之前的调度算法。首先, 请求与数据项不再是一对一的关系, 若直接沿用单数据项调度算法, 则很有可能只满足了一个请求的部分要求, 而剩下的部分在较长一段时间内都不能得到满足。其次, 不同请求需要的数据项之间可能有重叠, 重叠的程度反映了请求之间的联系。合理的调度应当能够利用这种请求之间的联系。而单数据项请求调度不会考虑这方面的情况。同时考虑数据与请求两方面的要求, 以达到一个较好的平均访问性能, 是多数据项广播中的关键问题。

2.6 多数据项广播调度算法

调度和索引是无线数据广播中最重要的两个方法。目前对于多数据项广播的研究基本上集中于调度方面。多数据项广播的复杂性对于调度算法的设计提出了更高的要求, 也使得在此领域的研究起步较晚。近年来有研究者提出了一些富有创新和有效的方法, 下文将分别对其中具有代表性的方法进行介绍。

3 面向请求的多数据项广播调度算法

早期的多数据项广播调度算法大多基于一个直观的想法, 即应当优先满足出现频率高的请求。后来的研究者将这一类算法归为面向请求的多数据项广播调度算法。

3.1 QEM 算法

Chung Yon Dohn 和 Kim Myoung Ho 在 1999 年提出了 QEM 算法^[12], 归纳了许多符合多数据项广播性质的规则, 是多数据项广播中具有相当影响力的算法。之后, 许多研究者在此基础上做了进一步改进。QEM 算法被大量其后的研究作为比较对象。

QEM 算法的思想基于同一个请求访问的数据尽可能地靠近可以缩短访问时间的发现。例如, 在图 1 所示的一个广播调度中, 用户在广播 d3 的时刻提出了需要数据 d2 和 d5 的请求。当用户的请求到达时, d2 已经被广播了, 用户需要等待下次广播周期的时候才能够获得 d2, 该请求的平均访问时间为 5。但如果广播序列为 {d1, d4, d3, d2, d5, d6}, 如图 2 所示, 用户可以在当前广播周期内获得所有的数据, 平均访问时间为 4.3。

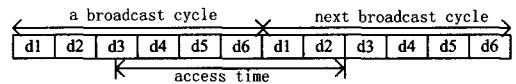


图 1 一个简单的广播调度

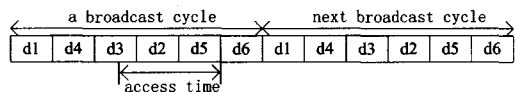


图 2 调整后的广播调度

基于这个发现, 文献[12]提出了请求距离 (Query Distance, QD) 的概念。QD 指的是一个请求的最短访问时间, 定义为

$$QD(q, \sigma) = BSize - \text{MAX}(\delta_1, \delta_2, \dots, \delta_n)$$

其中 $BSize$ 为一个广播周期的长度, δ_j 表示数据项 d_j 与 d_{j+1} 在调度序列 σ 下的间隔距离。

文献[12]证明 QD 在特定情况下可以代表请求的平均访问时间,

QEM 算法是在优先考虑请求的访问频率的基础上将请求的数据不断加入到调度序列, 最终扩展为一个完整的广播序列。在扩展时应满足以下 3 个基本原则:

- 在扩展调度序列时, 访问频率较高的请求比访问频率较低请求有较高优先权。
- 在将一个请求需要的数据项加入到调度序列中时, 需要保证加入的数据不会影响已经加入的请求的 QD。
- 在满足第二个条件的前提下, 调整调度序列使新加入的请求的 QD 最小。

具体的方法可以通过下面的例子说明:

假设目前共有 3 个请求 Q1, Q2, Q3, 访问频率分别为 $freq(Q1) = 100, freq(Q2) = 80, freq(Q3) = 50$ 。

请求的数据如图 3 所示。

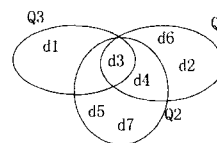


图 3 3 个请求的数据关系

首先将访问频率最高的 Q1 加入到广播序列中, 得到 $\sigma = [d_2, d_3, d_4, d_6]$

改变 σ 中的数据顺序不会影响请求的 QD。然后将 Q2 加入并调整相交数据顺序, 得到

$$\sigma = [d_5, d_7][d_3, d_4][d_2, d_6]$$

$[\]$ 与 $[\]$ 之间不能交换顺序, 注意到 Q1 与 Q2 相交的数据

$\{d_3, d_4\}$ 向 Q_2 靠近了。最后将 Q_3 加入,得到

$$\sigma = [d_5, d_7][d_3, d_4][d_2, d_6][d_1]$$

3.2 改进的 QEM 算法

在 QEM 算法提出之后, Lee Guanling 等人对该算法提出了改进^[13]。QEM 算法要求在扩展调度序列时不能影响之前已经加入的请求的 QD。如果放松这一条件,则系统的平均访问时间有可能进一步减小。

图 4(a)、(b)中灰色的部分表示新请求需要的数据,图 4(c)是满足 QEM 原则的调度序列,图 4(d)是采用改进的 QEM 算法的结果。可以看出后者将新请求需要的数据都连续放置。这在算法中称为一次移动(move)。这样做的后果会影响之前请求的 QD,但是新请求的 QD 却变为最小。综合两者的结果可以使整个系统的性能得到改进。由于改变之前请求的 QD 是有风险的,因此改进的 QEM 算法在加入新请求时需要首先判断一下是否应该执行移动动作。判断的依据包括两个方面。首先,执行移动后整体的性能必须得到提高,方法是计算移动后的总平均访问是否更小。其次,执行移动后会对之后加入的请求产生影响,这些影响可能是有益的也可能是有害的。必须保证正面的影响大于负面的影响。方法是判断下面的不等式,仅当不等式成立时才执行移动操作。

$$freq(Q_i) + \sum_{\substack{QDS(Q_i) \cap QDS(Q_j) \neq \emptyset \\ QDS(Q_i) \cap (QDS(\sigma) - QDS(Q_j)) = \emptyset}} freq(Q_j) > \sum_{\substack{QDS(Q_j) \cap QDS(Q_i) \neq \emptyset \\ QDS(Q_j) \cap (QDS(\sigma) - QDS(Q_i)) = \emptyset}} freq(Q_j)$$

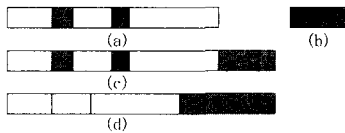


图 4 改进的 QEM 算法

3.3 GCM 算法

GCM 算法^[14,15]的思想是把数据库中的每一个数据映射为一个二进制字符串,其中每一位对应一个请求。如果这一位为 1,表示相应的请求需要这个数据。对于访问频率高的请求,应当对应二进制字符串中较高位的字符。再将这些二进制字符串使用 Grey 码重新编码并计算相应的 Grey 码值。然后按照 Grey 码值的大小从大到小广播全部数据。

由于 Grey 码相邻数据最多只改变一个字符的性质,同一个请求的数据会比较靠近。图 5 是一个 GCM 算法的例子,其中共有 8 个数据项和 3 个请求。

Gray Value	Gray Code	Binary Code
0	0 0 0	0 0 0
1	0 0 1	0 0 1
2	0 1 1	0 1 0
3	0 1 0	0 1 1
4	1 1 0	1 0 0
5	1 1 1	1 0 1
6	1 0 1	1 1 0
7	1 0 0	1 1 1

图 5 GCD 算法

以上算法都是从请求角度考虑调度,共同的原则是在优先考虑请求的访问频率的基础上将同一请求的数据尽量靠拢。从请求角度考虑,调度的缺点在于算法的性能会随着请求数量的增加而降低。当所处的系统中请求的数量很多,同时每个请求的访问频率又比较接近的时候,从请求角度考虑的算法不能得到较好的结果。因此之后的研究者更多地把注

意力放在从数据角度考虑的调度方法。

4 面向数据的多数据项广播调度算法

从数据角度考虑的调度关注于数据之间联系的紧密程度,其原则是尽量将联系紧密的数据连续广播,算法的性能不会受无线系统中请求数量变化的影响。

4.1 访问图算法

文献[16,17]首先提出了从数据角度考虑调度的算法。该算法首先根据用户的请求情况构造一个无向带权图 $G(V, E)$,称为访问图(access graph)。其中每一个结点表示数据库中的一个数据项。如果至少有一个请求同时访问了数据 d_i 和 d_j ,则在 d_i 和 d_j 结点之间建立一条边,如图 6 所示。边上的权值为所有同时访问这两个数据的请求的访问频率之和。

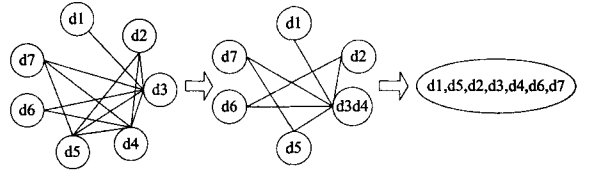


图 6 访问图算法

在建立调度序列时,首先将权值最大的边连接的两个结点合并为一个复合结点,同时更新访问图。若合并之后有一条以上的边与新结点相连,将它们合并为一条边,边上的权值为它们权值的总和。反复执行合并,直到所有的数据项都合并为一个结点,该结点中的数据顺序就是最终的调度序列。

在合并两个复合结点时,需要考虑两者的前后顺序,使形成的结点序列性能最优。在该算法中使用 `weighted_distance` 作为判断标准。它从最初的访问图中结点的关系出发考察两个复合结点中所有关联数据项对之间距离与权值的情况。定义如下

$$weighted_distance(u, v) = \sum_{\substack{\text{all edges } (i, j) \\ \text{between } u \text{ and } v}} \max \left(\frac{w(i, j)}{\text{length}(u) - \text{order}(i) + \text{order}(j)}, \frac{w(i, j)}{BL - (\text{length}(u) - \text{order}(i) + \text{order}(j))} \right)$$

其中 $\text{length}(u)$ 指复合结点中的数据项个数, $\text{order}(i)$ 指数据项 i 在复合结点中所处的位置。边 (i, j) 指在最初的访问图中存在的边。比较 $weighted_distance(u, v)$ 和 $weighted_distance(v, u)$ 的大小,选择结果值大的最后合并时的顺序。

文献[17]提出的从数据角度考虑的调度解决了从请求角度考虑的调度会受请求数量影响的不利因素,但是在判断数据间的联系程度时选择的标准比较简单。之后又有一些研究者提出了新的算法,主要在判断数据间联系程度的时候提出了更为合理的判断标准。

4.2 从数据联系角度考虑的调度算法

文献[18]提出了一种从数据联系角度考虑的调度算法,其基本思想是:计算每两个数据项之间的关系密切度(data affinity),根据数据密切度进一步求出段密切度(Segment affinity,其中“段”指的是一个或多个数据项组成的序列),通过不断将密切度最高的段合并,最终形成一个调度序列。

数据密切度定义为

$$aff(d_i, d_j) = \begin{cases} \sum_{q_k \in Q} \frac{has(q_k, d_i, d_j) \times freq(q_k)}{N_{data}(q_k)^2}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

$$has(q_k, d_i, d_j) = \begin{cases} 1, & \text{if } d_i \in q_k \wedge d_j \in q_k \\ 0, & \text{otherwise} \end{cases}$$

其中 $N_{data}(q_k)^{C_2}$ 表示请求 q_k 中所有两个数据的可能组合。可以看出,较长的数据请求(请求的数据较多)对于数据间密切度的影响较小。这个定义同时考虑到了请求的访问频率和请求的长度两个因素。

段密切度定义为

$$SegAff(S_i, S_j) = \begin{cases} \sum_{d_k \in S_i} \sum_{d_l \in S_j} aff(d_k, d_l) \\ \quad \times \frac{MaxDist(d_k, d_l) - dist(d_k, d_l)}{MaxDist(d_k, d_l)}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

定义将两个段中的每个数据对的密切度乘以它们间的距离因子(这里的距离指的是在当前调度序列中的距离,不是段中的距离)。距离因子定义为

$$dist(d_k, d_l) = \begin{cases} |d_k| + \delta + |d_l|, & \text{if } \delta < \frac{BSize - |d_k| - |d_l|}{2} \\ BSize - \delta, & \text{otherwise} \end{cases}$$

可见,当数据 d_k 与 d_l 邻近时,距离因子接近于1;当两者相距较远时,接近于0。

算法还考虑了逆序段的情况。如果一个段 S_i 包括数据 $\{d_1, d_2, \dots, d_n\}$, 则它的逆序段为 $S_i^{-1} = \{d_n, d_{n-1}, \dots, d_1\}$ 。本文作者发现一个请求在一个调度下的访问时间与它在该调度的逆序情况下访问时间是相同的。因此在两个段合并时需要考虑以下8种合并情况:

$$SegAff(S_i, S_j), SegAff(S_j, S_i), SegAff(S_i^{-1}, S_j), SegAff(S_j^{-1}, S_i), SegAff(S_i, S_j^{-1}), SegAff(S_j^{-1}, S_i), SegAff(S_i^{-1}, S_j^{-1}), SegAff(S_j^{-1}, S_i^{-1})$$

4.3 QDQ 算法

文献[19]提出了一种新的从数据角度考虑的调度算法。不同于之前以降低 QD 为目标的数据聚集算法,它提出应当考虑请求的间隔空间(Quadratic Query Distance):

$$QQD(q, \sigma) = \sum_{j=1}^{|q|} \delta_j^2$$

QQD 从二维的角度描述了同一请求数据间的关系。QQD 越大,平均访问时间越小。基于这一原则定义的数据密切度定义为

$$aff(d_i, d_j) = \begin{cases} \sum_k \frac{m_k^2}{|q_k|} QryHas(q_k, d_i) QryHas(q_k, d_j) & \text{for } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

$$QryHas(q_k, d_i) = \begin{cases} 1, & \text{if } d_i \in q_k \\ 0, & \text{otherwise} \end{cases}$$

其中 $m_k = N - |q_k|$, 表示 q_k 未访问到的数据的个数。

段密切度定义为

$$SegAff(S_i, S_j) = \begin{cases} \sum_k \frac{m_k - (R(S_i, k) + L(S_j, k))^2}{|q_k|} QryHas(q_k, d_i) \\ \quad \times QryHas(q_k, d_j) \times freq(q_k) & \text{for } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

其中 $R(S_i, k)$ 表示从段 S_i 的最右边开始连续未被 q_k 请求的数据的个数, $L(S_i, k)$ 则是从段 S_i 的最左边开始计数。

5 混合考虑的多数据项广播调度算法

此外还有研究者提出的一些算法综合了以上两类算法的

思想。

5.1 数据挖掘+QEM 算法

文献[20]中沿用了 QEM 的算法思想,但是与 QEM 不同的是,在进行广播序列的扩展时,不是以请求为单位逐步加入广播序列,而是采用了数据挖掘的关联规则,根据请求的出现频率及其所需要的数据项计算出所有数据组合的支持度(support)。若请求 T 需要的数据包含数据组合 X 中的所有数据项,则称请求 T 支持 X 。一个数据组合的支持度就是支持该组合的请求在所有请求中的比例。将支持度大于0的数据组合按照支持度的大小排列,得到一系列的大项集(large item-sets)。表1所列是这样的一个例子。然后以这些数据集合为单位,运用 QEM 扩展规则进行扩展,得到最终的广播序列。

表1 大项集列表

大项集	支持度(%)
{b, c}	40.0
{c, e}	40.0
{c, g}	40.0
{a, b, c}	33.3
{c, d, e, f}	26.7
{c, e, g}	13.3
{b, c, g}	6.7

5.2 平方根规则+QEM 算法

之前介绍的方法都设定一个广播周期中每个数据项只出现一次,这个限定使得这些方法不适用于非平坦调度的情况(即对每个数据项的访问频率各不相同的情况)。文献[21]提出将平方根规则与 QEM 算法综合起来的想法。算法的思想是,先把一个多数据项请求看作一个整体,根据访问频率的高低按照平方根规则将请求分布在广播周期上,然后以访问频率最高的请求的出现为标志,将整个广播周期分为若干个段,对每个段中出现的数据,采用 QEM 算法进行重新排列。这种方法综合了平方根规则和 QEM 算法的优点,对于非平坦的多数据广播调度有较好的性能。

6 算法性能比较

表2对以上介绍的各种多数据项广播调度的算法进行了比较。

表2 多数据项广播调度算法比较

算法	考虑角度	性能
QEM 算法	请求角度	最早提出的详细讨论多数据项广播的算法,相比随机调度平均访问时间减少了25%左右
改进的 QEM 算法	请求角度	在任何情况下平均访问时间都较 QEM 短,但复杂度较高
GCM 算法	请求角度	各种情况下平均访问时间较 QEM 长,复杂度低
访问图算法	数据角度	当请求的 selective(请求中重叠数据的比例)小的时候性能较 QEM 差,selective 大的时候性能好
数据密切度算法	数据角度	各种情况下平均访问时间都比 QEM 及改进的 QEM 短
QQD 算法	数据角度	比数据密切度算法性能稍好,且复杂度较低
数据挖掘+QEM 算法	混合角度	平均访问时间比 QEM 及改进的 QEM 短,复杂度高
平方根规则+QEM 算法	混合角度	对于典型非平坦调度,平均访问时间比 QEM 减少20%左右

7 其他多数据项广播调度方法

7.1 采用遗传算法的多数据广播调度

文献[22]提出使用遗传算法[23]来解决多数据项广播调

度的问题。先根据每个数据的访问频率遵循平方根规则得到最初的调度序列,然后利用遗传算法不断产生和筛选更加合适的调度序列(一次产生和筛选新的调度序列的过程称为进化了一代)。经过几十代的“进化”后得到一个平均访问时间较短的调度序列。算法的关键在于评价调度序列的优劣,使得在结果的“进化”过程中不断优胜劣汰,最终得到符合我们要求的结果。文中定义的调度序列的适应性评判公式为

$$Fitness(P) = \frac{1}{T_{Access}(Q)}$$

这种方法虽然可以得到一个较好的结果,但需要消耗服务器端大量的计算能力。

7.2 On-demand 模式下的多数据项广播

目前 On-demand 模式下的多数据项广播的研究较少,采用的模式借鉴了周期广播的方式。其思想为:每次广播一个长度确定的广播周期,并且对于一个周期内的数据,根据多数据项请求的特点进行调度优化。由于 On-demand 模式的特点,每次广播周期的内容通常是不同的。基于以上模式,On-demand 模式下的多数据项广播调度方法采用的大都是两步走的方法。首先确定下一次广播周期应当广播的数据项(请求选择问题),其次针对多数据项请求的特点合理安排该周期内的数据顺序(广播调度问题)。两者结合以达到使用户请求的平均访问时间最少的目的。其中后者采用的方法参照广播模式下的多数据广播的调度算法。

文献[13]对请求选择问题提出了 3 种方法。

- 仅考虑请求频率的方法:每次选择将访问频率最高的请求需要的数据项加入到广播序列中。

- 频率/长度比方法:每次选择将访问频率与长度的比值 $freq(q_i)/size(q_i)$ 最大的请求需要的数据项加入广播周期。

- 考虑了交集的频率/长度比方法:每次选择 $freq(q_i)/|size(q_i)|$ 值最大的请求需要的数据项加入直到达到广播周期长度上限为止。其中 $|size(q_i)|$ 指的是请求 q_i 还未加入到当前广播周期的数据项长度。

文献[24]提出了根据请求的“温度”作为选择依据的方法。温度定义为:

$$Temp_i = \sum R_{d_i} / num_i, \text{ for all } d_i \in QDS_i$$

其中 R_{d_i} 指数据项 d_i 的访问频率, num_i 指请求 i 需要的数据项个数, QDS_i 表示请求需要的数据项集合。然后选择 $Temp_i \times W_i$ 值最高的请求加入下一次广播周期。这里 W_i 指请求 q_i 的等待时间。

On-demand 广播模式下的多数据项广播研究目前尚处于探索阶段。虽然以上研究者提出了一些调度算法,但这些工作尚未突破周期广播模式下的思想。

结束语 相比简单的单数据项请求,多数据项请求是一种更一般化的请求方式,研究多数据项广播技术具有重要的理论和实际意义。本文首次对调度算法进行综述,按“面向访问”和“面向数据”这两个基本出发点对有算法进行分类,介绍其代表性算法,并进行分析和比较。

由于多数据项广播的复杂性,仍存在许多需要进一步研究的问题。我们认为这一领域目前最具潜力的发展方向包括:

- On-demand 模式下的多数据项广播。由于多数据项必须全局考虑的特点,多数据项广播在 On-demand 模式下对如

何安排调度问题的研究仍然处于探索阶段。然而,鉴于 On-demand 模式广播能够动态适应请求情况变化的优点,多数据项广播在 On-demand 模式下的实现将是发展的趋势。

- 多信道下多数据项广播的尝试。多信道广播是近年来讨论热烈的问题^[25,26],采用多信道广播来广播多数据项请求是一个值得考虑的方法。

- 多数据项广播中的索引问题。在广播序列中插入数据索引是目前广泛采用的减少调谐时间的有效方法^[27,28],但目前还很少有研究资料讨论如何在多数据项广播中设计和分布索引的问题。

参 考 文 献

- [1] Barbara D. Mobile Computing and Database: a Survey. IEEE Transactions on Knowledge and Data Engineering, 1999, 108-117
- [2] Imielinski T, Viswanathan S, Badrinath B R. Data on Air: Organization and Access[J]. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(3): 353-372
- [3] Imielinski T. Mobile Wireless Computing: Challenges in Data Management. Communications of the ACM, 1994
- [4] Xu J, Lee D. Data Broadcast. Handbook of Wireless Networks and Mobile Computing, 2002
- [5] Wong J W, Dykeman H D. Architecture and Performance of Large Scale Information Delivery Networks[C]// The 12th International Tele-traffic Congress, Italy, 1988
- [6] Xu J, Lee W C, Tang X, et al. An Error-resilient and Tunable Distributed Indexing Scheme for Wireless Data Broadcast[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(3): 392-404
- [7] Yao Y, Tang X, Lim E, et al. An Energy-efficient and Access Latency Optimized Indexing Scheme for Wireless Data Broadcast [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(8): 1111-1124
- [8] Wong J W. Broadcast Delivery. Proceedings of the IEEE 1988, 76(12): 1566-1577
- [9] Acharya S, Alonso R, Franklin M, et al. Broadcast Disks: Data Management for Asymmetric Communications Environments [C]// ACM SIGMOD Conference on Management of Data, 1995
- [10] Hameed S, Vaidya N H. Efficient Algorithms for Scheduling Data Broadcast[J]. ACM/Baltzer Journal of Wireless Networks, 1999, 5(3): 183-193
- [11] Chehadeh Y C, Hurson A R, Kavehrad M. Object Organization on a Single Broadcast Channel in the Mobile Computing Environment[J]. Multimedia Tools and Applications, 1999, 9(1): 69-92
- [12] Chung Y D, Kim M H. QEM: A Scheduling Method for Wireless Broadcast[C]// International Conference on Database Systems for Advanced Applications, 1999
- [13] Lee G, Lo S C, Chen A L P. Data Allocation on the Wireless Broadcast Channel for Efficient Query Processing. IEEE Transactions on Computers, 2002
- [14] Lee J Y, Chung Y D, Lee Y J, et al. Grey Code Clustering of Wireless Data for Partial Match Queries[J]. Journal of Systems Architecture, 2001(47): 445-458
- [15] Chung Y D, Kim M H. A Wireless Data Clustering Method for Multipoint Queries[J]. Decision Support Systems, 2001, 30 (4): 469-482

(下转第 38 页)

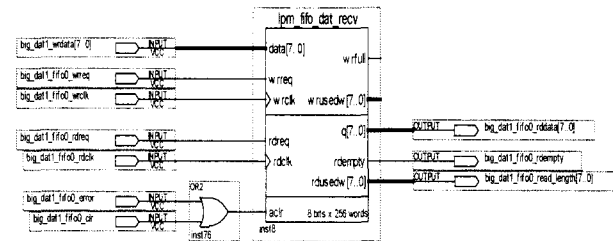
高速数据流,即前 8ms 数据流速度为 48Mb/s,后 8ms 空闲,总的平均速度仍为 24Mbps。这种情况下,至少要将上述的 16 帧扩大到 2 倍,即 32 帧,才能满足峰值流量的缓存需求。

考虑到系统的性能余量和 FPGA 片上存储器可开出 FIFO 容量的大小,每个 2 级 FIFO 缓存设为 4k 字节(19 个帧),两个共 8k 字节(38 个帧)。

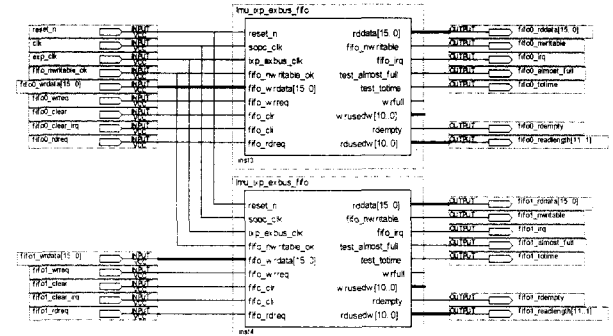
4 方案的实现

选用 intel 的 IXP460 网络处理器作为基站的主控 CPU, altera 的 CycloneII 系列 FPGA EP2C35F484 做逻辑处理和数据缓存,采用 50MHz 的外部总线时钟驱动,该二级缓存结构设计用 Quartus II 的实现如图 4 所示。

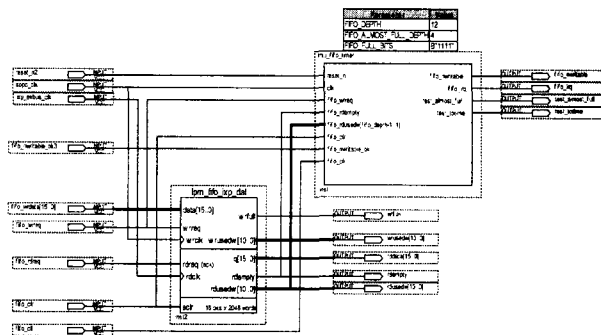
基站两侧的 X 线同时加上 24Mbps 的数据流,数据经过两级缓存后,最终基站通过 IXP460 外部总线读取的数据最高可达 40Mbps,远大于预设的 24Mbps,同时还实现了数据的实时处理和发送,完全达到了系统的设计要求。



(a) 一级缓存 (8 个, 每个 256 字节)



(b) 二级缓存 (2 个, 每个 4096 字节)



(c) 二级缓存的内部结构: FIFO 及其控制

图 4 二级缓存设计用 Quartus II 的实现

结束语 大型传感器网络是一种特殊的传感器网络,它由规则的大规模阵列传感器组成,网络中传感器数目多达上万个,产生的采集数据总流量可高达几百 Mbps,同时要实现数据采集的同步、采集数据的实时传输和处理等,是一个十分复杂的工程。

本实验中,运行在采集站和基站之间的是 RS-485 协议,该协议与缓存设计没有关系,只要保证 FPGA 内的接收器和相应的物理、链路层协议一致,该缓存策略适用于所有基站的缓存设计,有极大的扩展性。

参考文献

- [1] 王纲毅,王振华,田金文,等. 读写数据宽度不同的异步 FIFO 设计[J]. 计算机与数字工程,2005,33(6):107-110
- [2] 房海东,潘长勇,杨知行. 处理整帧数据的 FIFO 的巧妙控制设计[J]. 电讯技术,2003(4):63-67
- [3] Thor J, Akos D M. A direct RF sampling multifrequency GPS receiver [J]. Position Location and Navigation Symposium. IEEE,2002;44-51
- [4] Zinner C, Kubinger W. ROS-DMA: A DMA Double Buffering Method for Embedded Image Processing with Resource Optimized Slicing[C]// Real-Time and Embedded Technology and Applications Symposium. 2006;361-372

(上接第 20 页)

- [16] Lee G, Yeh M S, Lo S C, et al. A Strategy for Efficient Access of Multiple Data items in Mobile Environments[C]// International Conference on Mobile Data Management. 2001
- [17] Lee G, Lo S C. Broadcast Data Allocation for Efficient Access of Multiple Data Items in Mobile Environments[J]. Mobile Networks and Applications, 2003, 8: 365-375
- [18] Chung Y D, Bang S H, Kim M H. An Efficient Broadcast Data Clustering Method for Multipoint Queries in Wireless Information Systems[J]. The Journal of Systems and Software, 2002, 64: 173-181
- [19] Tsaih D, Wu G M, Wang C B, et al. An Efficient Broadcast Scheme for Wireless Data Schedule Under a New Data Affinity Model
- [20] Chang Y I, Hsieh W H. An Efficient Scheduling Method for Query-Set-based Broadcasting in Mobile Environments[C]// International Conference on Distributed Computing Systems Workshops, 2004
- [21] Sun W W, Zhang Z Y, Yu P, et al. Skewed Wireless Broadcast Scheduling for Multi-item Queries[C]// The 3rd IEEE International Conference on Wireless Communications, Networking and Mobile Computing. 2007

- [22] Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989
- [23] Huang J L, Chen M S. Broadcast Program Generation for Unordered Queries with Data Replication[C]// ACM Symposium on Applied Computing. 2003
- [24] Prabhu N, Kumar V. Data Scheduling for Multi-item and Transactional Requests in On-demand Broadcast[C]// International Conference on Mobile Data Management. 2005
- [25] Huang J L, Chen M S. Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1143-1156
- [26] Huang J L, Peng W C, Chen M S. SOM: Dynamic Push-pull Channel Allocation Framework for Mobile Data Broadcasting [J]. IEEE Transactions on Mobile Computing, 2006, 5(8): 974-990
- [27] Xu J, Lee W C, Tang X. Exponential Index: A Parameterized Distributed Indexing Scheme for Data on Air [C]// Second ACM/USENIX International Conference on Mobile Systems, Applications, and Services. 2004
- [28] Lo S C, Chen A L P. Efficient Index and Data Allocation for Wireless Broadcast Services[J]. Data Knowledge, 2007, 60(1): 235-255