J. Parallel Distrib. Comput. 74 (2014) 2795–2807

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Data broadcasting for dependent information using multiple channels in wireless broadcast environments

Chuan-Ming Liu *, Ta-Chih Su, Jenq-Haur Wang, Yen-Lin Chen

Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan

HIGHLIGHTS

- Showing that the considered AFBML Problem is NP-complete.
- A linear-time algorithm to generate an optimal broadcast schedule for a conditioned weighted DAG.
- Three heuristics with analysis to arrange vertices to multiple channels for general weighted DAGs.
- Discussing the three proposed heuristics with simulation experiments.

ARTICLE INFO

Article history:
Received 2 April 2013
Received in revised form
13 March 2014
Accepted 7 May 2014
Available online 20 May 2014

Keywords:
Data broadcasting
Latency
Multiple channels
DAGs
NP-Completeness

ABSTRACT

Data broadcasting is an effective approach to disseminating information to mobile clients and has attracted much research attention in recent years. In many applications, the access pattern among the data can be represented by a weighted DAG. In this paper, we consider the problem of efficiently generating the broadcast schedules on multiple channels when the data set has a DAG access pattern. We show that it is NP-hard to find an optimal broadcast schedule which not only minimizes the latency but also satisfies the ancestor property that retains the data dependency. We further derive a condition for the input DAGs under which one can generate an optimal broadcast schedule in linear time and propose an algorithm to generate the schedule. Due to the NP-completeness, we provide three heuristics for general DAGs based on the level of a vertex in the input DAGs and each heuristic uses a different policy to place vertices into the broadcast channels. There are two categories for the policies. The first category mainly considers the probability for a process to stop at a considered vertex. The second category takes the vertices which are affected most when assigning a vertex into consideration. We analyze and discuss these heuristics. A short experimental simulation is given for supporting and validating the discussion. In particular, the experimental results indicate that roughly considering the whole posterior vertices of each vertex is not precise and may not lead to good results and considering the vertices affected most when assigning a vertex will help reducing the latency.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Modern technologies in communication, networking, information management, and positioning systems enable a wireless mobile environment where the clients can ubiquitously access public information, such as electronic news service, traffic information, and stock-price information. In such an environment, the bandwidth between a server and a client is asymmetric [1,16], that is, the downlink bandwidth is much greater than the uplink bandwidth. The traditional *pull-based approach* (client–server model) for accessing services, where each client sends the query to the server and the server processes the query and replies to the client with the result, will meet the bottleneck problem due to the limited uplink bandwidth. Besides, in wireless mobile environments the number of mobile clients is growing constantly and the information service system is expected to serve an increasingly large number of users. In recent years, the *push-based approach* (data broadcasting) has been considered as an attractive solution for the bottleneck problem and provides an efficient way to disseminate the information to a large pool of clients.

In data broadcasting environments, the server broadcasts the data periodically and the clients access the data by listening to the broadcast channel and derive the results by themselves. One instance of the broadcast data consists of a broadcast *cycle*. The

* Corresponding author.
E-mail addresses: cmliu@csie.ntut.edu.tw, cmliu@ntut.edu.tw (C.-M. Liu), t7598023@ntut.edu.tw (T.-C. Su), jhwang@ntut.edu.tw (J.-H. Wang), ylchen@ntut.edu.tw (Y.-L. Chen).

http://dx.doi.org/10.1016/j.jpdc.2014.05.002
0743-7315/© 2014 Elsevier Inc. All rights reserved.

related topics on data broadcasting have attracted much research attention in recent years [6,8,9,11,14,16,18–20,26,23,27,29,31,32,34]. One of the important topics on data broadcasting is how the server schedules the data in the broadcast channel in order that the mobile clients can access the information efficiently.

There are two important measurements for evaluation in data broadcasting. The *latency* presents the time elapsed from requesting to receiving data and the *tuning time* indicates the amount of time spent on listening to the broadcast. Short latency and tuning time can ensure the quality of service (QoS) provided by the server. Minimizing the latency is one of the objectives in many previous papers. Those papers usually considered the data access frequency [2,10,13,19,33] or the data access pattern [3,7,11,12,14,27,31]. We follow this trend and one of our main objectives is to minimize the latency. Furthermore, using multiple channels can significantly reduce the latency [9,13,22,29,32,33]. However, a poorly designed broadcast schedule can lead the *data conflicts* [9,14,25] in the broadcast. Because the client can tune into only one of the broadcast channels at a time instance, the data conflicts occur when either (1) two or more requested data items are broadcast at the same time on the broadcast channels or (2) the desired data items arrive during channel switching. The conflicts will result in an increment on the latency. In this work, we assume that the data broadcast can be synchronized among channels as well as the channel switching time is negligible.

In many applications, the data to be broadcast are dependent. Consider that a client requests a Web page which contains some components, such as audio clips or images. When a client uses a browser to access the Web page, the browser will request all the components contained in the Web page after receiving the Web page. Therefore, a request leads to an access pattern where the data are dependent [3,12,14,27]. The other application can be found when a mobile client has an inquiry about the stock price of some company. It is highly possible that the mobile client would like to know the stock information of some other related companies. In this case, the information among all the related companies is dependently accessed. In addition, consider the on-demand broadcasting model [7,11,24,26,23,29] where the server collects a batch of requests from the mobile clients and then sends the results to that group of clients via broadcast. After receiving a signal from the server, each mobile client starts to receive the result from the beginning of the broadcast cycle. The broadcast data in this model are correlated and dependent. For the shortest-route service using data broadcasting [24], different clients may share parts of the route information of each other as Fig. 1 shows. Clients m_1 and m_2 need the road information (r_0, r_3, r_5, and r_6) and (r_1, r_3, r_5, and r_7), respectively. Hence, the data are dependent according to the need of the clients and after accessing r_5, r_6 or r_7 may be accessed. The data dependency of the aforementioned applications can be expressed using directed graphs. The authors in [19] discussed the broadcast generation for the data having dependency which can be modeled as a directed graph. The provided approach schedules the broadcast by decomposing the input directed graph and then place the vertices into the channel.

For the applications in the areas such as multimedia databases and knowledge bases, the data to be broadcast is of a multimedia or hypertext format with rich semantics [14]. These applications can use the object-oriented paradigm to model them. The objects(data) in an object-oriented paradigm are normally associated with one another via some semantic links, such as inheritance, aggregation, or association. In many cases, the relationships among the objects are not random but can be expressed as a hierarchy or edge-weighted *directed acyclic graph* (DAG), where the objects are vertices and the weighted edges denote the semantic strength between two vertices [14]. A DAG is a special case of the directed graphs. In this paper, by referring to [14], we consider the access pattern formed by all the possible requests where the data dependency and access frequency can be represented as an edge-weighted DAG. The weight on a directed edge in the considered DAG presents the conditional probability of accessing a vertex. Our study can be applied to the aforementioned applications if the considered case can be expressed as an edge-weighted DAG. We investigate how to generate multi-channel broadcast schedules in order to minimize the average latency for the broadcast data having an edge-weighted DAG access pattern.

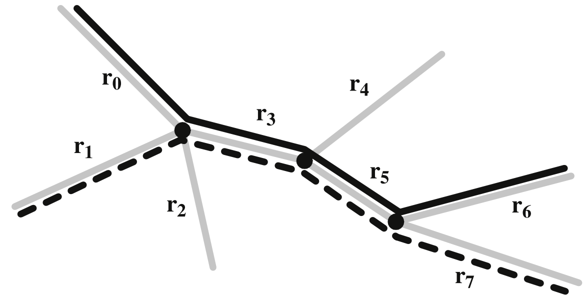

Fig. 1. Road segments r_0, r_3, r_5, and r_6 are for client m_1 (solid-line) and road segments r_1, r_3, r_5, and r_7 are for client m_2 (dashed-line); therefore, road segments r_3 and r_5 are shared by m_1 and m_2.

Allowing a mobile client to start receiving the data in the middle of a broadcast cycle can reduce the latency by skipping the waiting time for the beginning of a broadcast cycle. However, for the applications mentioned in the previous paragraph, because of the DAG access pattern [14], we assume that the client will start the query process from the beginning of the broadcast cycle as [16,17,28]. Such a consideration distinguishes our proposed algorithms from the others [3,12]. To simplify the discussion, we assume that each data item corresponds to one packet (or broadcast slot) in the broadcast. Actually, if the length of the data item is longer than the packet size, the data item can be partitioned into smaller data items which fit the packet and the dependency among these data items can be modeled as a chain in the DAG access pattern with probability 1 on each edge.

Due to data dependency, we consider that the generated broadcast schedule satisfies the *ancestor property* of the vertices in the input DAG G. For an edge (u, v) in G, the ancestor property indicates that vertex u should be broadcast before vertex v in the same cycle. Since a mobile client can only select one channel to tune into on multiple channels, dependent data cannot be broadcast at the same time on different channels. Otherwise, data conflicts will occur and an extension on the latency can be expected. Hence, the broadcast schedule generated without considering the weights (access frequencies) and without the ancestor property may result in a longer latency.

In this paper, we discuss the problem of generating an optimal broadcast schedule which satisfies the ancestor property and results in a minimum latency in a multi-channel broadcast environment. A discussion on the related work and some preliminaries are given in Section 2. Section 3 presents the formal definition of the problem. The latency in the considered problem is re-defined and is a refinement for the definition in the preliminary version of this paper appeared in [21]. We will show in Section 4 that this problem is NP-complete. Due to the NP-completeness of the problem, we explore a condition for the input DAG under which one can generate an optimal broadcast schedule in linear time and provide such a scheduling algorithm in Section 5. Note that the condition is derived by observations and not common in practice due to the limited broadcast bandwidth. In the design of the algorithm, since the minimized latency can be achieved, we also consider the number of switches between channels during the client query process, which may cause the time delay [14]. For a general case, we provide three

Download English Version:

https://daneshyari.com/en/article/432706

Download Persian Version:

https://daneshyari.com/article/432706

Daneshyari.com