Evolution Strategy Based Optimization of On-Demand Dependent Data Broadcast Scheduling

Rinku Dewri, Darrell Whitley, Indrakshi Ray and Indrajit Ray
Department of Computer Science
Colorado State University
Fort Collins, CO 80523, USA
{rinku,whitley,iray,indrajit}@cs.colostate.edu

ABSTRACT

Data broadcasting makes effective use of low bandwidth and is commonly used in applications involving mobile devices. We consider the case where data must be broadcast in a particular order and within a specified response time. However, communication bottlenecks prohibit the timely serving of all requests; although, missing the deadline does not make the data utility zero. In this work, we consider the problem of real-time data broadcast scheduling in the presence of soft deadlines together with constraints on the order in which data-items should be broadcast to be useful. We explore the method of evolution strategy to solve the problem, keeping in view that the real-time scheduler has to effectively trade-off between its running time and the quality of schedules generated.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: [Scheduling]

General Terms

Performance

1. INTRODUCTION

Many application domains involve clients interested in similar data objects. A data object in this context refers to a group of data items that can be processed one at a time following some order. A scheduling problem is imminent in such application platforms when the number of data access requests gets larger than what can be handled by the existing bandwidth capacity. Another aspect of the problem is the actual utility that remains in the data items when received by a client. Various soft deadlines may be imposed on the requested data items, which if not served within a specific time window may result in near zero utility residual in the data item when finally received. A broadcast schedule is sought which can serve clients with as much utility as possible.

We propose an utility accrual method for data requests involving constraints on the order in which the data items are to be served. The utility function is used by an evolution strategy based schedule optimizer to evaluate the effectiveness of the encoded schedules. We pay particular attention to the run time constraint of the scheduler and argue that simple variants of the method of evolution strategy can be employed to satisfy this requirement. Based on our initial observations on the quality of the schedules so generated, we propose an adaptive sampling rate for the evolution strategy which results in a balanced trade off between the running time and the quality of the scheduler.

2. RELATED WORK

Chehadeh et al. propose object graphs as the method to impose ordering constraints [1]. Hurson et al. propose an extension of their work for multiple broadcast channels [4]. Several theoretical properties for the average access time of ordered queries in multi-channel environments is studied by Huang et al. [3]. The authors further argue that several special cases of the problem of broadcasting dependent data is NP-hard and propose a genetic algorithm to address the problem [2]. Our approach to the problem differs in the utilization of an utility metric to evaluate the potency of schedules on the client end and introduce evolution strategy as a fast and effective method to obtain high utility schedules.

3. BROADCAST SCHEDULING

Clients use an uplink channel to a data provider to request various data items served by the provider. Each request Q_j takes the form of a tuple $\langle D_j, R_j \rangle$, where R_j is the response time within which the requesting client expects the first data item from the ordered set D_j, hereafter called a *data group*. A scheduler is invoked every time a new request is received. At each instance, the scheduler first determines the requests that will be fully served by the current broadcast, and removes them from the request queue. For the remaining requests, the data items required to serve them are determined and a schedule is generated.

3.1 Utility Metric

For a request Q_j arriving at time T_j and involving the data group $D_j = \{d_{1j}, d_{2j}, \ldots, d_{N_j j}\}$, let $t_{1j}, t_{2j}, \ldots, t_{N_j j}$ be the time when the respective data items are retrieved by the client. The utility generated by serving the first data item is given as,

$$u_j[t_{1j}] = \begin{cases} 1 & , t_{1j} - T_j \leq R_j \\ e^{-\alpha(t_{1j} - T_j - R_j)} & , t_{1j} - T_j > R_j \end{cases} \tag{1}$$

The utility generated from the subsequent items is then given as, for $i = 2, \ldots, N_j$,

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

$$u_j[t_{ij}] = \begin{cases} u_j[t_{(i-1)j}] & , t_{ij} - t_{(i-1)j} \leq R_T \\ u_j[t_{(i-1)j}]e^{-\alpha(t_{ij} - t_{(i-1)j} - R_T)} & , t_{ij} - t_{(i-1)j} > R_T \end{cases} \tag{2}$$

In this work, we assume that the utility of a data item for a client decays by half for every factor of increase in the response time, i.e. $\alpha = \ln 0.5/R$, where $R = R_j$ for the first data item in the requested group and $R = R_T$ for any subsequent data item. We then say that the utility generated by serving the request is given by the utility generated at the last item of the data group, i.e. $U_j = u_j[t_{N_j j}]$.

For a given schedule S, generated to serve the requests $Q_1, Q_2, \ldots, Q_M$ in the queue, the utility generated by the schedule is given as,

$$U_S = \sum_{k=1}^{M} U_k \tag{3}$$

3.2 Problem Statement

A data source $D = \{D_1, D_2, \ldots, D_N\}$ is a set of N ordered sets (or data groups), where $D_j = \{d_{1j}, d_{2j}, \ldots, d_{N_j j}\}$ with N_j being the cardinality of D_j and $j = 1, \ldots, N$. All data items d_{ij} are assumed to be unique and are of equal size d_{size}. A request queue at any instance is a dynamic queue Q with entries Q_j of the form $\langle D_j, R_j \rangle$, $D_j \in D$ and $R_j \geq 0$. At any instance, let $Q_1, Q_2, \ldots, Q_M$ be the entries in Q. We define $Rem[Q_j]$ as the ordered subset of data items that has been requested in Q_j but not yet received, i.e. $Rem[Q_j] \subseteq D_j$. A schedule is a total ordering of the elements in the multi-set $\bigcup_{j=1,\ldots,M} Rem[Q_j]$. At each scheduling instance, the scheduler needs to find a schedule with maximum utility as given by Eq. (3). In this study, we use simple stochastic local search variants using $(1 + \lambda)$-ES and $(1, \lambda)$-ES, for 1000 generations, to find such schedules. Further, we fix the maximum number of function evaluations allowed to 15000 for fast performance. The sampling of the neighborhood is performed using the "shift" mutation operator. We also experiment with a third variant involving an adaptive sampling rate λ. The sampling rate in this variant is changed as the search progresses.

4. EMPIRICAL RESULTS

The data set used in our experiments is generated using various well known distributions that are known to capture the dynamics of a public data access system quite well. Number of data items in a group is assigned using two different distributions: INC – most requested data group is the largest and DEC – most requested data group is the smallest.

For the INC type assignment, a simple local search involving $\lambda = 1$ yields an acceptably high ($> 90\%$) utility level. Both $(1 + 1)$-ES and $(1, 1)$-ES show similar levels of performance for this problem.

For the DEC type assignment, the bandwidth poses a hard bottleneck. The $(1 + 1)$-ES fails to provide the same level of performance as it does for the INC assignment. Increasing the sampling rate λ to $3, 5$ and 10 show improvements up to 80% utility levels. Although increasing the number of generations to 2000 improves the utility level up to 86%, the number of function evaluations ($2000 \times 10 = 20000$) exceeded the maximum set limit of 15000. These observations are similar in both the *comma* and the *plus* variant of the ES.

For the experimental data set, a simple $(1 + 1)$-ES for 1000 generations is sufficient when the QoS requirement is not too high ($< 75\%$). For the case when the utility requirement is higher, a higher sampling rate is desired. However, results obtained from running the different variants for 2000 generations (more function evaluations) do not always yield a high difference in the utility levels, as compared to those from running the same variants for 1000 generations. We believe that an increase in utility is more easily obtainable by changing the sampling rate, rather than the number of generations.

We experiment with another variant where the sampling rate is progressively increased from 1 to 10 as the generations progress. We consider a maximum of 1000 generations for the $(1 + \lambda)$-ES where λ is incremented by 1 in every 100 generations. Global utility reaches levels similar to the $(1 + 10)$-ES with 2000 generations. Noticeably, the number of function evaluations used is much less – 5500 compared to 20000.

5. CONCLUSIONS

In this paper, we consider the problem of on-demand data broadcast scheduling under the presence of dependency between the data items requested. We define an utility function to evaluate the effectiveness of a schedule and use an evolution strategy to maximize the utility of broadcasts. Our initial experiments with different sampling rates suggest that solving the problem when the most requested data group is the smallest in size is relatively much easier than the case when the most requested data group has the highest number of data items. The results obtained from the variant strategy suggest that better schedules can be generated without engaging too many function evaluations.

6. ACKNOWLEDGMENTS

This work was partially supported by the U.S. AFOSR under contract FA9550-07-1-0042. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies of the U.S. Air Force or other federal government agencies.

7. REFERENCES

[1] Chehadeh, Y., Hurson, A., and Kavehrad, M. Object Organization on a Single Broadcast Channel in the Mobile Computing Environment. *Multimedia Tools and Applications 9*, 1 (1999), 69–94.

[2] Huang, J.-L., and Chen, M.-S. Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment. *IEEE Transactions on Knowledge and Data Engineering 16*, 9 (2004), 1143–1156.

[3] Huang, J.-L., Chen, M.-S., and Peng, W.-C. Broadcasting Dependent Data for Ordered Queries without Replication in a Multi-Channel Mobile Environment. In *19th IEEE ICDE* (2003), pp. 692–694.

[4] Hurson, A., Chehadeh, Y., and Hannan, J. Object Organization on Parallel Broadcast Channels in a Global Information Sharing Environment. In *19th IEEE IPCCC* (2000), pp. 347–353.