Capturing Approximated Data Delivery Tradeoffs

Haggai Roitman #1, Avigdor Gal #2, Louiqa Raschid *3

Technion–Israel Institute of Technology
Haifa 32000 Israel
1 haggair@tx.technion.ac.il
2 avigal@ie.technion.ac.il

* *University of Maryland*
College Park MD 20742
3 louiqa@umiacs.umd.edu

***Abstract*— This paper presents a middleware data delivery setting with a proxy that is required to maximize the completeness of captured updates, specified in its clients' profiles, while minimizing at the same time the delay in delivering the updates to clients. The two objectives may conflict when the monitoring budget is limited. Therefore, any solution should consider this tradeoff in satisfying both objectives. We term this problem the "proxy dilemma" and formalize it as a biobjective optimization problem. Such problem occurs in many contemporary applications, such as mobile and sensor networks, and poses scalability challenges in delivering up-to-date data from remote resources to meet client specifications. We present a Pareto set as a formal solution to the proxy dilemma. We discuss the complexity of generating a Pareto set for the proxy dilemma and suggest an approximation scheme to this problem.**

I. Introduction

Proxy middleware is a common practice nowadays and is designed for increasing scalability in the face of client sophistication. Typical applications that use such middle-ware are monitoring sensor or mobile networks, Web feeds aggregation services, stock prices and sponsored search auctions on the commercial Internet, and monitoring resources on the computational Grid.

In this work we consider a middleware setting managed by a proxy that delivers notifications to multiple clients according to their data delivery requirements, specified in the form of client profiles. A client profile identifies a set of resources of interest with which the client requires to be synchronized and a set of events (*e.g.,* updates to some resource) that identify when such synchronization should take place. Each event is associated with a deadline in getting notification about the event occurrence and we further assume that clients prefer to get notifications as soon as possible to minimize their observed delay in the system. The time frame associated with each resource and each such event is termed an *Execution Interval*, and requires from the proxy to deliver a *notification* to the client during that interval. A client profile may consist of multiple such intervals.

The proxy is responsible to monitor a set of resources of interest specified in its clients' profiles and deliver notifications to the clients. The proxy has two data delivery objectives. On the one hand, the proxy is interested in maximizing *completeness* by capturing as many execution intervals as possible. However, with limited monitoring budget and with thousands of clients and resources to monitor, data delivery may not be *current*, resulting in a delay in delivering updates. On the other hand, the proxy is interested in minimizing such delay in clients' notifications and aims at reducing the delay observed by clients, and to accomplish that it may sacrifice completeness. The proxy, therefore, has to consider two conflicting data delivery objectives, and face a *proxy dilemma*, where a data delivery *tradeoff* exists between completeness and currency.

In this paper, we propose a solution to a biobjective problem that optimally solves the proxy dilemma. This solution considers both conflicting objectives *at the same time*. The conventional definition of a solution to multiobjective problems is a Pareto set that identifies the set of all non-dominated alternative solutions. We show that generating a Pareto set for the proxy dilemma using full enumeration is of a high order polynomial complexity. We then present a fully polynomial time approximation scheme (FPTAS) solution to a Global Assignment Problem (GAP) that can be used to construct an approximated Pareto set. Finally, we compare the complexity of the approximated solution and its full enumeration counterpart.

The rest of the paper is organized as follows: In Section II, we survey the literature and classify solutions. We introduce the proxy dilemma in Section III. In Section IV, we present solutions to the proxy dilemma and we conclude and discuss future work in Section V.

II. Related Work

We classify related research as follows: We first determine if one or both objectives are considered. When both objectives are considered, we classify solutions based on whether an aggregate solution is used or if there is a precedence among the objectives.

Uni-objective: These solutions focus only on one objective while ignoring the other. For example, [1] considers *coherency*, the maximization of completeness. Also in [2], the optimization objective is *average freshness*. As another example, [3], [4] consider only currency.

Bi-objective aggregation: These solutions consider aggregations of both objectives. As an example, in [5] both objectives are aggregated to model the utility in capturing updates to Web resources using a monitoring schedule.

Bi-objective preference: These solutions consider lexicographic preferences over both objectives. Such preference is very common in broadcast system solutions (see for example the RxW algorithm [6]) and may consist of either a lexical ordering of the two objectives, or it can have a more flexible mechanism where a tuning parameter is used to determine the relative importance (or weights) of each objective.

Biobjective dilemma: Here, both objectives are simultaneously considered, with no internal preference between them. Applications that face the proxy dilemma can use solutions in this class to deliver as many updates while minimizing delay, both requirements are (a-priori) equally important. To the best of our knowledge, our work is the first to consider this class of problems in the domain of data delivery.

III. Model and Problem Definition

We now follow the data delivery model of [7] and consider a simplified profile model with *non-complex* execution intervals (defined by $rank(\mathcal{P}) = 1$ in [7]). We first present a summary of the three building blocks used by this model, namely *client profiles*, *execution intervals*, and *schedules*, and two performance metrics, namely *completeness* and *currency*. We then present the *Proxy Dilemma* formally.

A. Profiles and execution intervals

Let $\mathcal{R} = \{r_1, r_2, ..., r_n\}$ be a set of n resources and let $\mathcal{T} = (T_1, T_2, ..., T_K)$ be an epoch with K chronons. A client profile p is a set of execution intervals $p = \{I_1, I_2, \ldots, I_k\}$, where each execution interval $I \in p$ is a pair $I = \langle r, [T_s, T_f] \rangle; T_s, T_f \in \mathcal{T}; T_s \leq T_f$ that associates an interval of time in $\mathcal{T}$ with some resource $r \in \mathcal{R}$ on which the client requires to be synchronized with the state of resource r. The interval start and finish times can further represent the occurrence time of some events of interest that initiate and terminate this synchronization period. We assume that only execution intervals that are associated with different resources may overlap, while more general overlap relationships, which are further discussed in [7], are considered as future work.

B. Schedules and data delivery metrics

A data delivery schedule $S = \{s_{i,j}\}_{i=1,\ldots,n; j=1,\ldots,K}$ assigns $s_{i,j} = 1$ if resource $r_i \in \mathcal{R}$ should be monitored (probed) by the proxy at chronon $T_j \in \mathcal{T}$, else $s_{i,j} = 0$. We denote by $\mathbb{S}$ the set of all possible schedules. Given a client profile p and an execution interval $I \in p$, we say that I is *captured* by schedule $S \in \mathbb{S}$ if $\exists T_j \in I : s_{i,j} = 1$ and $r = r_i$ with $r \in I$. Given an execution interval I captured by S at chronon $T_j \in I$, the delay $d(I, T_j)$ is as follows:

$$d(I, T_j) = T_j - I.T_s \tag{1}$$

Given a set of client profiles $\mathcal{P} = \{p_1, p_2, ..., p_m\}$, we denote by $N(\mathcal{P}, S)$ the total number of execution intervals captured by schedule S and further denote $D(\mathcal{P}, S)$ the total obtained delay. Given a schedule S, we measure its performance with regard to the *completeness* objective (denoted N_{ratio}) by calculating the ratio of total captured intervals out of all possible intervals in $\mathcal{P}$.

We then define $D_{avg}(\mathcal{P}, S)$ the *average (total) delay* per profile (*i.e.*, the average of profiles' total delay) that measures the *currency* in terms of clients' average observed delay in capturing intervals specified in $\mathcal{P}$.

C. The proxy dilemma

We assume that the proxy has a limited budget given as an upper bound on the number of resources that the proxy can monitor in parallel at each chronon $T_j \in \mathcal{T}$. This constraint is represented by the budget vector $\vec{C} = (C_1, C_2, \ldots, C_K)$. A similar constraint type is also used in other settings including Web Monitoring [5] and Web Crawlers [4].

Problem 1 (Proxy Dilemma): Given a set of client profiles $\mathcal{P} = \{p_1, p_2, ..., p_m\}$, *maximize* completeness N_{ratio} while *minimizing* currency $D_{avg}(\mathcal{P}, S)$ at the same time, subject to the budget $\vec{C}$. Formally, the proxy dilemma can be represented as the following biobjective optimization problem:

$$\begin{aligned} &\textit{maximize } N_{ratio}(\mathcal{P}, S) \\ &\textit{minimize } D_{avg}(\mathcal{P}, S) \\ &\textit{s.t. } \textstyle\sum_{i=1}^{n} s_{i,j} \leq C_j, \ \forall j = 1, 2, \ldots, K \end{aligned} \tag{2}$$

IV. Solutions to the proxy dilemma

We present Pareto sets as a solution to the biobjective problem that models the proxy dilemma and then present a description of an approximation to the problem.

A. Pareto sets

The proxy dilemma identifies a tradeoff between completeness and currency when its two objective functions are dependent. Maximizing completeness may decrease currency and vice versa. Such a tradeoff between objectives of general multiobjective optimization problems is well studied (*e.g.*, [8] and many references therein). The conventional definition of a solution to multiobjective problems include a set of *non-dominated feasible* solutions, also known as the *Pareto Set* (or its geometric representation in the form of *a Pareto curve*). Non dominance between two schedules means that each schedule has better performance than other schedules with regard to *at least one* of the objectives. Thus, the Pareto set contains schedules that dominate all other schedules that do not belong to the set. Within the set there is no preference between the schedules. The Pareto set identifies the data delivery optimal tradeoff.

A feasible solution to the data delivery proxy dilemma is a schedule that satisfies the problem constraints. Formally, a schedule $S \in \mathbb{S}$ is *feasible* if for each $1 \leq j \leq K$, $\sum_{i=1}^{n} s_{i,j} \leq C_j$. We denote by $\mathcal{S}(\vec{C}) \subseteq \mathbb{S}$ the set of all feasible schedules in the problem domain. Dominance between two feasible schedules is defined next.

Definition 1: Given a set of profiles $\mathcal{P}$, upper bounds $\vec{C}$, and schedules $S_1, S_2 \in \mathcal{S}(\vec{C})$, schedule S_1 is said to *dominate* schedule S_2 (denoted as $S_1 \succ S_2$) if $N(\mathcal{P}, S_1) > N(\mathcal{P}, S_2)$

and $D(\mathcal{P}, S_1) \leq D(\mathcal{P}, S_2)$, or $N(\mathcal{P}, S_1) \geq N(\mathcal{P}, S_2)$ and $D(\mathcal{P}, S_1) < D(\mathcal{P}, S_2)$.
We denote by $\bar{\mathcal{S}}(\mathcal{P})$ the set of *all* non-dominated feasible schedules that construct the Pareto set.

Discrete multiobjective optimization problems with even two linear objectives tend to be NP-hard [9]. In some cases, maximizing one objective while keeping the other fixed (known as the Global Assignment Problem – GAP) is an NP-hard problem by itself. The proxy dilemma, however, is not a hard problem, but rather polynomial in the number of resources. Given the budget vector $\vec{C}$, it is easy to show that the complexity in generating the Pareto set has a worst case of $\mathrm{O}\left(n^{2KC_{\max}}\right)$, with $C_{\max} = \max_{j=1,2,\dots,K}(C_j)$ and assuming that $C_{\max} \neq \mathrm{O}(n)$ (otherwise, the solution is trivial). Finding the exact Pareto set can be very costly for large values of K or n, and therefore, we now present a cheaper solution that finds an approximated Pareto set instead.

B. Approximating Pareto sets

We now discuss how an ε-approximated solution to the proxy dilemma is achieved. For this purpose we first relax the definition of dominance between schedules.

Definition 2: Given a set of profiles $\mathcal{P}$, constraints $\vec{C}$, and a pair of schedules $S_1, S_2 \in \mathcal{S}(\vec{C})$, schedule S_1 is said to *ε-dominate* schedule S_2 ($S_1 \succ_\varepsilon S_2$) if $N(\mathcal{P}, S_1)(1+\varepsilon) \geq N(\mathcal{P}, S_2)$ and $D(\mathcal{P}, S_1) \leq D(\mathcal{P}, S_2)(1-\varepsilon)$.

That is, $S_1 \succ_\varepsilon S_2$ if S_1 has at least better or equal performance than S_2 for both objectives up to a difference factor of ε. We further denote by $\bar{\mathcal{S}}_\varepsilon(\mathcal{P})$ the data delivery tradeoff approximated Pareto set.

Sufficient conditions for the existence of an efficient algorithm that provides ε-approximated Pareto sets for general multiobjective problems were suggested in [9]. [9] shows that in order to find an efficient approximation, polynomial in the problem size and $\frac{1}{\varepsilon}$, one needs to provide a Fully Polynomial Time Approximation Scheme (FPTAS) to the *General Assignment Problem* (*GAP*) [10]. Using GAP, one can further utilize the zigzag technique described in [11] to construct the approximated Pareto set (curve). The GAP formulation of the proxy dilemma is given next.

Problem 2 (Data Delivery GAP): Given $\mathcal{P}$, $\vec{C}$, $N \in \mathbb{N}$, $D \in \mathbb{N}$, and ε, determine if there is a schedule $S \in \mathcal{S}(\vec{C})$ such that $N(\mathcal{P}, S) \geq N$ and $D(\mathcal{P}, S) \leq D$ or there is no S' such that $N(\mathcal{P}, S') \geq N(1+\varepsilon)$ and $D(\mathcal{P}, S') \leq D(1-\varepsilon)$.

By solving a GAP problem we can determine if there *exists at least one* schedule S that is guaranteed to dominate any schedule that has performance of $\langle N, D\rangle$, and therefore, we can guarantee non-dominance in a square environment of $\langle N, D\rangle \times \langle N(1+\varepsilon), D(1-\varepsilon)\rangle$.

We now shortly discuss how an FPTAS for the data delivery GAP problem is achieved using a dynamic programming solution; Given the pair of values $\langle N, D\rangle$, we first fix the upper bound on total delay to be D, and then relax that bound by a factor of $(1-\varepsilon)$ and try to maximize completeness by capturing as many intervals as possible, subject to the relaxed upper bound on total delay $D(1-\varepsilon)$. The dynamic programming solution iterates over schedules in $\mathcal{S}(\vec{C})$ and finds an *optimal* schedule $\tilde{S}$ that is then used to determine whether we can guarantee to captured *at least* N intervals. Suppose that the schedule $\tilde{S}$ manages to guarantee the required level of completeness, then that schedule is considered as a candidate for the approximated Pareto set. Given the suggested FPTAS to the data delivery GAP, we further utilize the method of [11] to construct the Pareto set.

The runtime complexity of this solution is only $\mathrm{O}(\frac{n^2K^2}{\varepsilon})$ compared to the full enumeration solution complexity that requires $\mathrm{O}\left(n^{KC_{\max}}\right)$ to find a *single* non-dominated schedule.

V. Conclusions

In this paper we defined the proxy dilemma data delivery problem as a biobjective optimization problem. We discussed the optimal solution to this problem given in the form of a Pareto set. We further presented a GAP algorithm to construct an approximate Pareto set as a cheaper alternative solution.

However, GAP may still not scale well as n (resources) and K (epoch length) increase. We are now exploring online policies as scalable alternatives that can provide efficient solutions that are much cheaper then GAP. Additionally, we now look at problem instances with general overlap relationships between execution intervals that include also intra-resource overlaps ([7]). Identifying such overlaps efficiently may reduce remarkably the monitoring budget and increase scalability (*e.g.,* by further caching the captured execution intervals). Finally, we plan to use GAP as a feedback mechanism for online data delivery policies, where GAP can be used to select dominant policies or tune policies to improve their overall performance.

References

[1] S. Shah, K. Ramamritham, and P. Shenoy, "Maintaining coherency of dynamic data in cooperating repositories." in *VLDB*, 2002.

[2] J. Cho and H. Garcia-Molina, "Synchronizing a database to improve freshness," in *Proceedings of the ACM-SIGMOD conference on Management of Data (SIGMOD)*, Dallas, Texas, May 2000, pp. 117–128.

[3] K. C. Sia, J. Cho, and H.-K. Cho, "Efficient monitoring algorithm for fast news alerts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 7, pp. 950–961, 2007.

[4] J. L. Wolf, M. S. Squillante, P. S. Yu, J. Sethuraman, and L. Ozsen, "Optimal crawling strategies for web search engines," in *Proceedings International WWW Conference*. Honolulu, Hawaii, USA: ACM Press, 2002, pp. 136–147.

[5] S. Pandey, K. Dhamdhere, and C. Olston, "WIC: A general-purpose algorithm for monitoring web information sources," in *Proceedings of the International conference on very Large Data Bases (VLDB)*, Toronto, ON, Canada, Sept. 2004, pp. 360–371.

[6] D. Aksoy and M. Franklin, "RxW: a scheduling approach for large-scale on-demand data broadcast," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 846–860, 1999.

[7] H. Roitman, A. Gal, and L. Raschid, "Satisfying complex data needs using pull-based online monitoring of volatile data sources," in *International Conference on Data Engineering (ICDE), Cancun, Mexico*, 2008.

[8] M. Ehrgott, *Multicriteria Optimization*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999.

[9] C. Papadimitriou and M. Yannakakis, "On the approximability of trade-offs and optimal access of web sources." in *FOCS*, 2000, pp. 86–92.

[10] D. B. Shmoys and É. Tardos, "An approximation algorithm for the generalized assignment problem." *Math. Program.*, vol. 62, pp. 461–474, 1993.

[11] S. Vassilvitskii and M. Yannakakis, "Efficiently computing succinct trade-off curves." in *ICALP*, 2004, pp. 1201–1213.