IJDCST @December Issue- V-1, I-8, SW-4
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Targeted Online Data Delivery Using Push based Mechanisms Based on SUP

P. Sai Krishna
NRI Institute of Technology, Agiripalli, Krishna Dist.

Abstract: Online data delivery is the main aspect in present days according to client request presented in network data delivery in client server architecture. This scheme maximizes user utility by satisfying all users present with an alternative and more flexible approach. It diminishes the system resource usages. We introduce develop an adaptive monitoring solution Satisfy User Profiles (SUPs) after deliberate the advantages of the previous approach. We empathize adequate optimality conditions for SUP through the formal evaluation. We examine the behavior of SUP under various condition by using the real (RSS feeds) and synthetic traces. There is a problem in delivery of data with sequential propagation delay of each client request. In this paper we propose to develop a dual frame process for data delivery in each client with appropriate request. In this application, we can get a high degree of satisfaction of user utility when the measure of sup closely estimate the real event and has the potential to save a significant amount of system resources through our analysis. These relations are accepted in push based servers for efficient data delivery. User utility with only a moderate increase in resource utilization can be maximized by SUP in this schema.

Index Terms: Distributed databases, online information services, client/server multitier systems, online data delivery.

I. INTRODUCTION

We introduce develop an adaptive monitoring solution Satisfy User Profiles (SUPs) after deliberate the advantages of the previous approach. We empathize adequate optimality conditions for SUP through the formal evaluation. We examine the behavior of SUP under various condition by using the real (RSS feeds) and synthetic traces. There is a problem in delivery of data with sequential propagation delay of each client request. In this paper we propose to develop a dual frame process for data delivery in each client with appropriate request. In this application, we can get a high degree of satisfaction of user utility when the measure of sup closely estimate the real event and has the potential to save a significant amount of system resources through our analysis.

These relations are accepted in push based servers for efficient data delivery is a large distributed system of servers deployed in multiple data centers across the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance. CDNs serve a large fraction of the Internet content today, including web objects (text, graphics and scripts), downloadable objects (media files, software, and documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks.

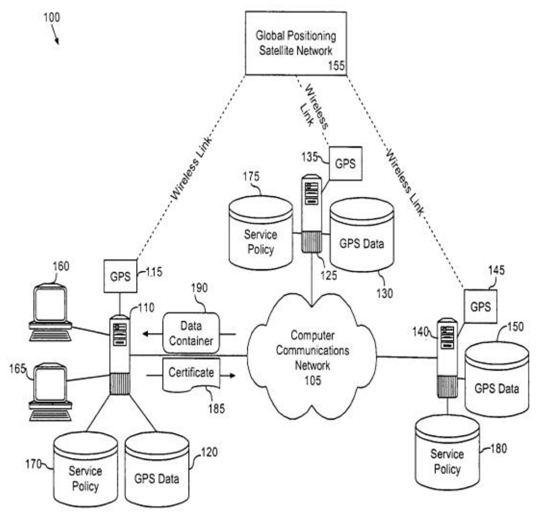


Figure 1: Managing data delivery in a data communications network.

A CDN operator gets paid by content providers such as media companies and e-commerce vendors for delivering their content to their audience of end-users. In turn, a CDN pays ISPs, carriers, and network operators for hosting its servers in their data centers. Besides better performance and availability, CDNs also offload the traffic served directly from the content provider's origin infrastructure, resulting in cost savings for the content provider. In addition, CDNs provide the content provider a degree of protection from DoS attacks by using their large distributed server infrastructure to absorb the attack traffic. While most early CDNs served content using dedicated servers owned and operated by the CDN, there is a recent trend to use a hybrid model that uses P2P technology. In the hybrid model, content is served using both dedicated servers and other peer-user-owned computers as applicable.

The Web is becoming a universal medium for disseminating information of all kinds, including highly dynamic information. Significant amount of valuable dynamic information is being posted to the Web and people want to access it. Direct manual viewing of dynamic Web pages is not an adequate mode of access for one or both of the following two reasons:

- Most information posted on the Web is not made available forever and may disappear or be replaced by new information at any time
- Many applications require automated synthesis of information from multiple dynamic Web sources

Push, pull, and hybrid protocols have been used to solve a variety of data delivery problems. Push-based consistency in the context of caching dynamic Web content. Push is typically not scalable and reaching a large number of potentially transient clients is expensive. Pushing information may overwhelm the client with unsolicited information. Several hybrid push-pulls solutions have also been presented. We focus on pull-based resource monitoring and satisfying user profiles. The burden of when to probe an RSS resource lies with the client. Much of the existing research in pull-based data delivery casts the problem of data delivery as follows: Given some set of limited system resources. We refer to this problem as OptMon1.

A solution to OptMon1 is accompanied by the need to meet rigid a priori bounds on system resource constraints. A rigid a priori setting may also have the unintended consequence of forcing excessive resource consumption even when there is no additional utility to the user. While resource consumption is dynamic and changes with needs with this class of problems user needs are set as the constraining factor of the problem. We present an optimal algorithm in the OptMon2 class. SUP is simple yet powerful in its ability to generate optimal scheduling of pull requests. SUP is an online algorithm; at each time point. SUP depends on an accurate model of when updates occur to perform correctly.

Most work on continuous query processing assumes that data is “pushed” into the query engine in the form of *data streams*. Only heuristics with no formal guarantees on effectiveness have been proposed for converting pull- oriented Web sources into push-oriented data streams. Periodic polling breaks down in the presence of a large number of frequently updated Web sources, when resources become inadequate for polling all Web pages at a fast rate.

II. Monitoring the Web & Dual Framework

Our models for the Web monitoring scheduling problem and the way in which Web pages change extend the framework. Let P be the set of Web pages under consideration for monitoring. Time is divided into discrete time instants and monitoring is performed in epochs of N consecutive time instants. We focus on the problem of scheduling monitoring of the pages in P during a single epoch. The monitoring a page includes the duties of fetching the page from its remote source and determining whether it has undergone one or more changes of interest. This simplification is based on the assumption that the fixed overhead for the operations required is the dominant factor. Let C denote the maximum number of pages that can be monitored in a single time instant. Value of C depends on the availability of resources for monitoring. If C equals or exceeds the number of pages then the scheduling problem is trivial: simply monitor each page at every instant. A legal *monitoring schedule* for an epoch is one that performs at most C monitoring of pages during each time instant $T_1, T_2, \ldots, T_N$.

Let $R = \{r_1; r_{2;} \ldots ; r_n\}$ be a set of resources and T be an epoch and let $\{T_1; T_2; \ldots ; T_N\}$ be a set of equidistant chronons1 in T. schedule S = $\{si,j\}_{i=1\ldots n;j=1\ldots N}$ is a set of binary decision variables.

The OptMon1 formulation assumes that system constraints are hard constraints where their assignment is a priori independent of specific user utility maximization task. OptMon1 involves a system resource constraint of the maximum number of probes per chronon for all resources. This constraint represents the number of monitoring tasks that a Web monitoring system can allocate per chronon for the task of maximizing the utility gained from capturing updates to Web pages. The benefits of OptMon1 are apparent whenever there are hard system constraints on resources. OptMon1 formulation has two main limitations:

- We expect that there will be periods of varying intensity with respect to the intensity of updates at the server(s) as well as the intensity of probes needed to satisfy client profiles
- The rigidity of OptMon1 algorithms with respect to system resource allocation

Solutions to OptMon1 have not dynamically attempted to reduce resource consumption. Once the upper bound on bandwidth has been set, bandwidth can no longer be adjusted and user needs may not be fully met. We propose a dual formulation OptMon2 setting the fulfillment of user needs as the hard constraint. It assumes that the system resources that will be consumed to satisfy user profiles should be determined by the specific profiles and the environment.

III. TARGETED DATA DELIVERY MODEL

The centerpiece of our model is the notion of execution intervals and a simple modeling tool for representing dynamically changing client needs. We then turn our attention to the formal definition of a schedule and the utility of probing. We present a case study using RSS a popular format for publishing information summaries on the Web. The diverse data types are nowadays available as publications in RSS. The use of RSS feeds is continuously growing. A user of such a reader can customize her profile by specifying the rate of monitoring each RSS feed and is supported by a pull-based protocol. The RSS protocol was extended with special metatags such as server side TTL that hint when new updates are expected. Despite these, feature a client who is only interested in being alerted of updates for a particular item in some news category. A profile should be easy to specify and sufficiently rich to capture client requirements. Specified in the trigger part of the notification rule, the trigger condition is immediately evaluated and if it is true.

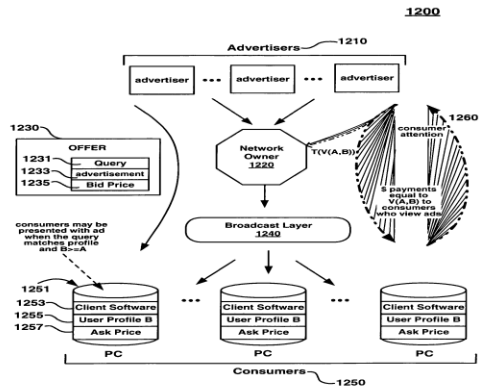


FIG. 12

Figure 2: delivering targeted data in a system for targeted data delivery

The period in which a notification rule is executable was referred to in the literature as life. We emphasize here the difference between the executable period of a notification (life) and the period in which rules. An execution interval starts with an event and its length is determined by the relevant life policy. With pull-based monitoring, content is delivered upon request with limited effectiveness in estimating object freshness. Hybrid approach combines push and pull, based either on resource constraints or on role definition. The mediator can monitor servers by periodically pulling their content and determine when to push data to clients based on their content delivery profiles.

It was argued that the use of an update model based on Poisson processes suits well updates in a Web environment. The Poisson processes are suitable for modeling a world where data updates are independent from one another such as updates to auction Web sites. Such a model reflects well scenarios in which e-mails arrive more rapidly during work hours and more bids arrive toward the end of an auction.

IV. SUP Algorithm

Let R = {r_1; r_2; . . . ; r_n} be a set of resources and T be an epoch and let {T_1; T_2; . . . ; T_N} be a set of equidistant chronons1 in T. schedule S = {$s_{i,j}$}∈ S be a schedule. P = {p_1; p_2; . . . ; p_m} be a set of user profiles. Given a notification rule η and the set of its execution intervals EI(η). SUP identifies the set of resources Q_I η Domain (Q, η) that must be probed in an execution interval I ∈ EI(η).

The main intuition behind the SUP algorithm is to identify the best candidate chronon in which the assignment of probes to resources maximizes the number of execution intervals that can benefit from each probe. We identify the best candidate chronons by delaying the probes of execution intervals to the last possible chronon in which the utility is still positive. We now provide a description of the algorithm. Pseudo code of the SUP algorithm and the two routines:

a. *AdaptiveEIsUpdate*
b. *UpdateNotificationEIs*

The algorithm builds a schedule iteratively. It starts with an empty schedule and repeatedly adds probes. Then determines the earliest chronon in which to probe and the notification rule associated with this monitoring task. SUP depends on an accurate set of execution intervals to perform correctly. Determining a set of execution intervals suffers from two main problems:

- The underlying update model that is used to compute the execution intervals is stochastic in nature
- It is possible that the underlying update model is incorrect and the real data stream behaves differently than expected

We propose to exploit feedback from probes to revise the probing schedule in a dynamic manner after each monitoring task. We first introduce the general scheme of SUP that addresses the first problem and does not require changes to any parameters.

SUP uses the AdaptiveEIsUpdate routine to apply the adaptive modifications. Routine first applies adaptive modification to notification rule η by recalculating a new execution interval I^* to be scheduled. The routine determines a set of notification rules that may be associated with execution intervals that need to be modified by identifying those notification rules that reference resource r_i in their trigger part. The UpdateNotificationEIs routine is called to ensure that resources that belong to overlapping intervals are only probed once. Let l = η be the assignment of SUP where η is the notification rule whose execution interval I is processed at time T_j.

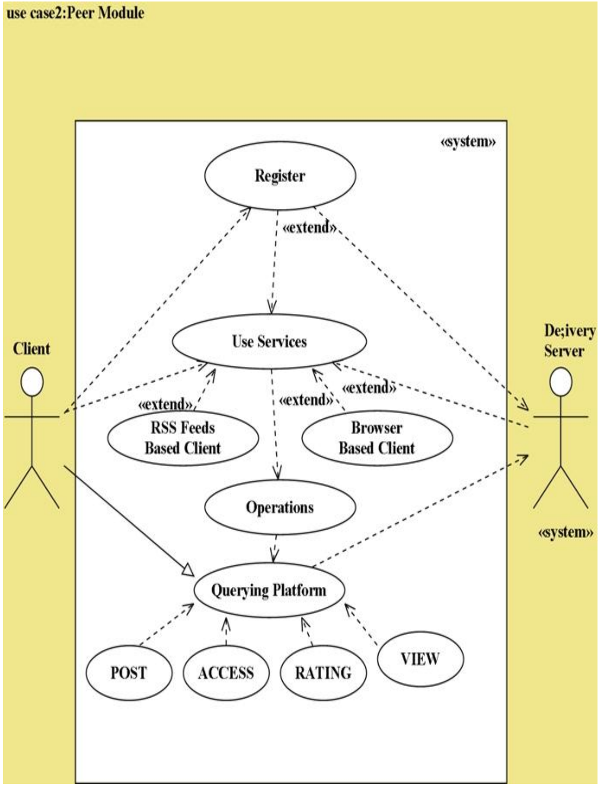
IV. EXPERIMENTAL ANALAYSIS

We implemented SUP experimented with it on various trace data sets, life parameters and profiles. Traces of update events include real RSS feed traces and synthetic traces. We also implemented WIC to determine a schedule for OptMon1 and TTL as another OptMon2 solution. Recall that while

OptMon1 problems set hard constraints on system resources and OptMon2 aims at minimizing system resource utilization. OptMon2 secures the full satisfaction of user specification, while OptMon1 can only aim at maximizing it. We make the following indirect comparison:

a. We compare the system resource utilization of the different solutions
b. Given some level of system resource utilization when we compare the effective utility of the different solutions

The TTL solution will use the server provided TTL to determine when the next probe to a resource should be to satisfy a profile. WIC is a solution to the OptMon1 provides the system resource utilization and corresponding utility of the three algorithms. We add a parameter denoted by M to represent a system constraint on the total number of probes allowed per chronon.



The optimal number of probes for SUP is 2462 for this data set. We also varied the M level for WIC. Given that, TTL is allowed to probe the same total number of probes as WIC (N.M) and assuming that there are n resources. We now focus on the data set and the Poisson update model of (b). The effective utility for SUP is about 0.62, the effective utility is represented by a single point.

V. CONCLUSION

We focused on pull-based data delivery that supports user profile diversity. The minimizing the number of probes to sources is important for pull-based applications to conserve resources and improve scalability. We introduce develop an adaptive monitoring solution Satisfy User Profiles (SUPs) after deliberate the advantages of the previous approach. We empathize adequate optimality conditions for SUP through the formal evaluation. We examine the behavior of SUP under various condition by using the real (RSS feeds) and synthetic traces. There is a problem in delivery of data with sequential propagation delay of each client request. In this paper we propose to develop a dual frame process for data delivery in each client with appropriate request. In this application, we can get a high degree of satisfaction of user utility when the measure of sup closely estimate the real event and has the potential to save a significant amount of system resources through our analysis. These relations are accepted in push based servers for efficient data delivery.

VI. REFERENCES

[1] Haggai Roitman, Avigdor Gal and Louiqa Raschid, "A Dual Framework and Algorithms for Targeted Online Data Delivery," IEEE Transactions on Knowledge and Data Engineering, VOL. 23, NO. 1,pp. 5-21, JANUARY 2011.

[2] Sandeep Pandey, Kedar Dhamdhere_, Christopher Olston, "WIC: A General-Purpose Algorithm for Monitoring Web Information Sources," NSF ITR grants CCR-0085982 and CCR-0122581, Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

[3] C. Liu and P. Cao, "Maintaining Strong Cache Consistency on the World Wide Web," Proc. Int'l Conf. Distributed Computing Systems (ICDCS), 1997.

[4] J. Yin, L. Alvisi, M. Dahlin, and A. Iyengar, "Engineering Server- Driven Consistency for Large Scale Dynamic Web Services," Proc. Int'l World Wide Web Conf. (WWW), pp. 45-57, May 2001.

[5] H. Liu, V. Ramasubramanian, and E.G. Sirer, "Client and Feed Characteristics of rss, a Publish-Subscribe System for Web Micronews," Proc. Internet Measurement Conf. (IMC), Oct. 2005.

[6] A. Adi and O. Etzion, "Amit—The Situation Manager," Int'l J. Very Large Data Bases, vol. 13, no. 2, pp. 177-203, May 2004

[7] J. Cho and H. Garcia-Molina. Synchronizing a database to improve freshness. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 2000.