

# A Pull Based Data Delivery Model for Maximizing User Satisfaction and Optimizing Resource Consumption

<sup>1</sup>V. Anushadevi, <sup>2</sup>Dr. R.V.Krishnaiah

<sup>1</sup>Dept. of CSE, DRK Institute of Science and Technology, Hyderabad, AP, India

<sup>2</sup>Dept. of CSE, DRK Group of Institutions, Hyderabad, AP, India

## ABSTRACT

With the advent of Internet technologies and innovative applications online data delivery is made possible. The data might be delivered to diverse destinations such as programs over Internet or human users. As the number of online users who need the data to be delivered increase rapidly, it is inevitable to study the possible means of optimizing resources in order to ensure user satisfaction. Ideally such online delivery system should provide high user satisfaction while minimizing resource utilization. In this paper we implement the framework proposed by Roitman et al. in order to achieve this dual goal. We

Implemented two approaches for optimization of resources. The former is to maximize user utility with prior constraints while the latter is to do the same by considering satisfaction of all users. We build a prototype application which demonstrates the proof of concept. The empirical results revealed that the application is effective in achieving high degree of user satisfaction and can be used in real world.

**Index Terms** –Online data delivery, resource optimization, information services

## I. INTRODUCTION

Data sources over Internet are diversified and distributed computing technologies make the usage of diversified data sources possible. The technologies included might be Web Services, Grid Computing and Cloud Computing. Data delivery in distributed environment causes infrastructure challenges. In this paper we implement a framework proposed by Roitman et al. [1] for optimal resource allocation while ensuring user satisfaction. The challenge here includes providing right kind of information to right people at right time with available resources by optimally utilizing resources. For content delivery in a distributed environment there are many mechanisms. They include auctions, stock prices, RSS news feeds and so on. We consider the usage of hybrid protocols and user profile management for achieving optimal resources allocation for online data delivery. The approach we followed in this paper is push based for online data delivery. There are push based technologies such as caching dynamic content [2], push based policies [3], JMS Messaging, and BlackBerry [4]. The Pull based technologies include Web caching [5], Web crawlers [6] and other techniques [7]. In this paper we focus on pull based

approach for ensuring user satisfaction and also resource monitoring. RSS feeds are the best example for pull based approach. User profile management concept can reduce the burden on data delivery systems as they can make certain decisions with ease with the help of user profiles that leads to user satisfaction. The pull-based data delivery models are explored in [8] and [9]. In this paper we consider the problem of data delivery with user satisfaction as an optimization problem. The hypothetical scenario we considered is to minimize resources consumption provided certain user profiles while ensuring satisfaction of all users involved. We implement algorithms for solving data delivery problems.

The solution we followed is stochastic in nature. It keeps on monitoring the process and make notice of updates when they exceed expected time limits. Moreover this should work in real time while data is streamed. Probes and feedbacks are used to improve dynamic scheduling decisions. The pull based solutions presented in the framework are used to solve the problem of online data delivery. The proposed work is compared with other such techniques like WIC algorithm [9], and ubiquitous TTL algorithm [5]. For experiments we used both

synthetic data and also real RSS feeds along with several user profiles. We implemented a prototype application to demonstrate the proof of concept. The empirical results revealed that our solution is capable of achieving highest user satisfaction while optimizing the resource utilization.

## II. PRIOR WORK

Dynamic content delivery can be made over Internet using different approaches. The online data delivery can be done using two models such as pull model and push model. When the client program is responsible to take the content by probing, it is known as pull model. When the client is automatically given when content updates are available, this model is known as push model. Web crawlers are examples for pull-based examples while freshness policies [5] are examples of pull – based approaches. Typically the models rely on the freshness of objects that are to be delivered. Various update models came into existence of content freshness. The update models of [10], [11] focused on representing update arrival in stochastic terms. In the context of synchronization, pull – based freshness models are also proposed. The best examples for this are web crawlers [6], [12]. They proposed policies for maximize objects that exist in cache memory. Refreshing the objects offline is the goal of the work. Offline index freshness is the goal rather than handling queries online. Efficient algorithms are inevitable for quality data delivery that results in user satisfaction. Implementation of such algorithms is difficult in pull based models when compared with push models. The reason for this is that only in stochastic terms the update model is known. The existing pull based policies and the dimensions are presented in fig. 1.

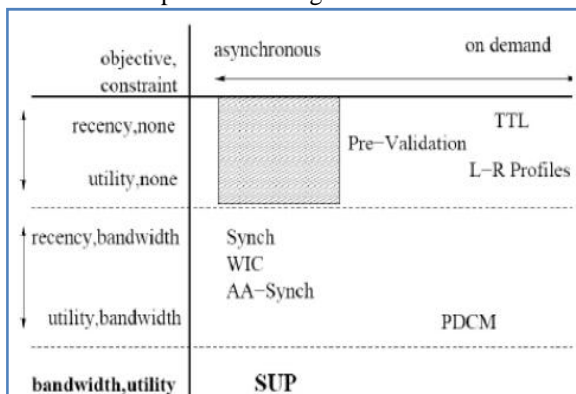


Fig. 1 – Pull based policies (excerpt from [1])

As can be seen in fig. 1, the classification of pull based policies is illustrated. The TTL and PDCM are on demand in nature while the Synch, WIC, AA-synch and SUP are asynchronous in nature. They are illustrated in terms of objective and constraint such as utility, recency, and bandwidth. The pure asynchronous works are presented in [6], [8] and [9] that keep on refreshing data irrespective of client

requests. They are push models by nature. On demand works are presented in [5] and [13] that refresh objects only when client makes request. Other approaches which are in the middle of both the extremes include [14], [15] and [16] as they support both asynchronous and on-demand data access models.

## III. FRAMEWORK FOR ONLINE DATA DELIVERY

This section describes the framework which is implemented in the prototype application of this paper. The framework is based on two problems. The first one is “Maximize user satisfaction under given system constraints”. The second problem is “Minimize the usage of resources under user satisfaction constraints”. Based on the predefined constraints that can maximize utility of every user, the proposed framework is built. For instance in [9] and [17] there is involvement of resource constraints and number of probes. The constraints in [17] represent the count of crawling tasks that ensure freshness of web resources. The proposed model is based on the notion of execution intervals. The execution intervals are obtained from user profiles. We take a case study using RSS as it is popular form of content sharing approach over World Wide Web.

With regard to problem1, the proposed framework assumes that system constraints are hard constraints. And their assignment has to be done with pre-defined constraints. For optimal resource allocation the approach presented in [24] is used. In order to solve the problem 2, we formulate a dual approach in which user satisfaction constraints are kept as hard rules while the optimization of resource constraints is performed.

## IV. EXPERIMENTAL RESULTS

The prototype application is built as a custom simulator in Java platform. The application has provisions to solve the two optimization problems discussed earlier. The tradeoffs can be fixed by end users. The environment used for the application development is PC with 4 GM RAM, Core 2 Dual processor running Windows 7 OS. The IDE (Integrated Development Environment) used is Net Beans. The experimental results with RSS feeds example are presented as a series of graphs.

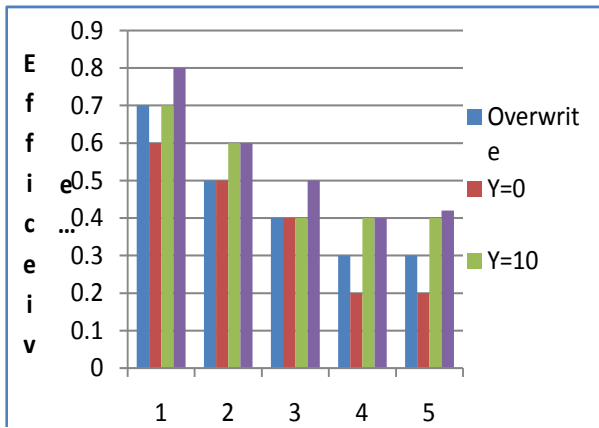


Fig. 2 - Satisfy User Profiles Performance(synthetic dataset)

As can be seen in fig. 2, the horizontal axis represents the maximum number of updates a client can tolerate while the vertical axis represents effective utility. The maximum number of updates a client can tolerate is from 1 to 5 only. The graph shows different life parameters and their performance.

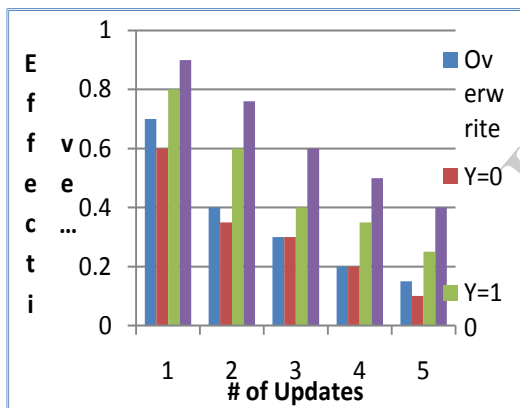


Fig. 3 - Satisfy User Profiles Performance (RSSfeeds)

As can be seen in fig. 3, the horizontal axis represents the maximum number of updates a client can tolerate while the vertical axis represents effective utility. The maximum number of updates a client can tolerate is from 1 to 5 only. The graph shows different life parameters and their performance.

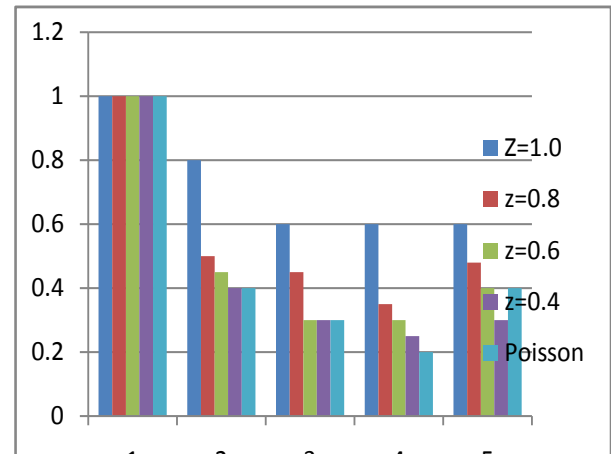


Fig. 4 - Satisfy User Profiles Performance of Various Update Models(synthetic data)

As can be seen in fig. 4, the horizontal axis represents the maximum number of updates a client can tolerate while the vertical axis represents effective utility. The maximum number of updates a client can tolerate is from 1 to 5 only. The graph shows different life parameters and their performance.

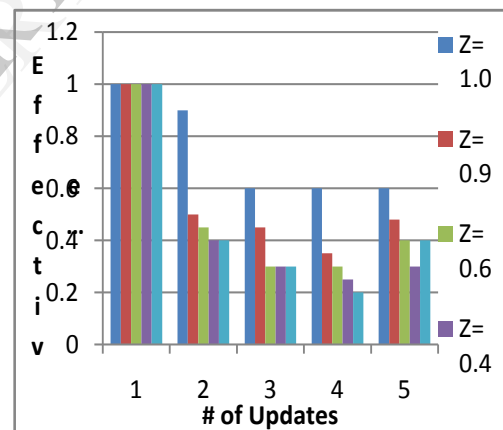


Fig. 5 - Satisfy User Profiles Performance of Various Update Models (RSS feeds)

As can be seen in fig. 5, the horizontal axis represents the maximum number of updates a client can tolerate while the vertical axis represents effective utility. The maximum number of updates a client can tolerate is from 1 to 5 only. The graph shows different life parameters and their performance.

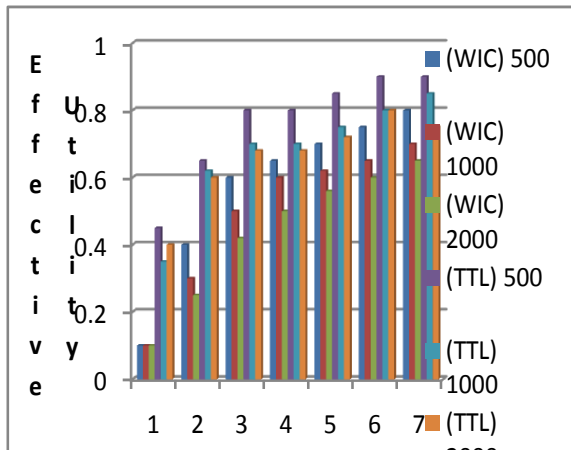


Fig. 6–Number of probes vs. Effective utility (FPN)

As can be seen in fig. 6, the horizontal axis represents number of probes while the vertical axis represents effective utility. For different TTL and WIC experiments, the results are presented.

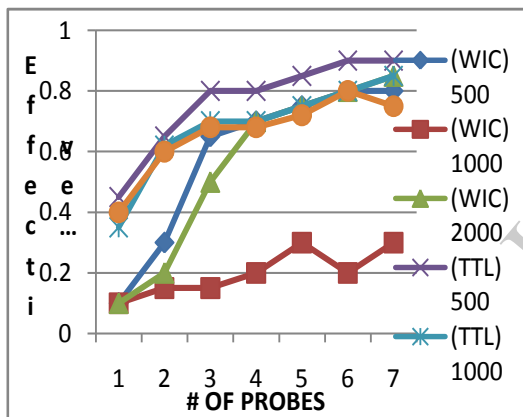


Fig. 7–Number of probes vs. Effective utility (Poisson)

In Fig. 7, the horizontal axis represents number of probes while the vertical axis represents effective utility. For different TTL and WIC experiments, the results are presented.

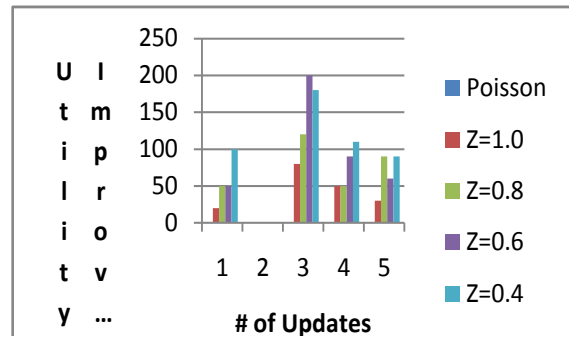


Fig. 8 – Impact of Feedback of RSS Data(Effective Utility Improvement)

As can be seen in fig. 8, the horizontal axis represents the maximum number of updates a client can tolerate while the vertical axis represents effective utility improvement. The maximum number of updates a client can tolerate is from 1 to 5 only. The graph shows different life parameters and their performance.

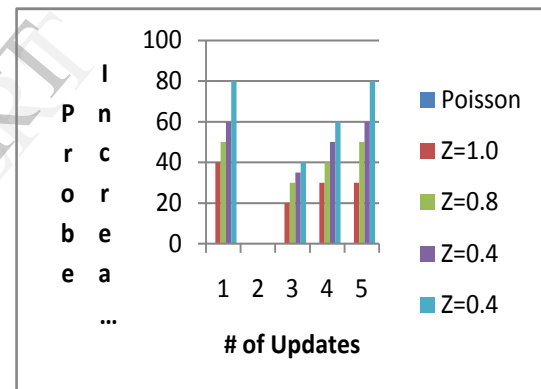


Fig. 9 – Impact of Feedback of RSS Data (Probe Increase)

As can be seen in fig. 9, the horizontal axis represents the maximum number of updates a client can tolerate while the vertical axis represents probe increase value. The maximum number of updates a client can tolerate is from 1 to 5 only. The graph shows different life parameters and their performance.

**V. CONCLUSIONS**

In this paper we implemented the content delivery model presented by Roitman et al. [1] which is a pull-based data delivery model. It used user profile diversity and algorithms for achieving two goals namely ensuring user satisfaction and optimal resource utilization. We considered the problem of fresh content delivery online as an optimization problem. We used RSS feeds live and also synthetic

data for experiments. We developed a prototype application to demonstrate the proof of concept. The application makes use of feedbacks in order to reduce number of probes. The user profile management is given importance in order to ensure that all users are satisfied with the content delivery. The experimental results revealed that our application achieves high user satisfaction while consuming optimal resources.

## 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, JANUARY 2011.
- [2] 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.
- [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] "BlackBerry Wireless Handhelds," <http://www.blackberry.com>,2010.[17] Z. Jiang and L. Kleinrock, "Prefetching Links on the WWW," Proc.IEEE Int'l Conf. Comm., 1997.
- [5] J. Gwertzman and M. Seltzer, "World Wide Web Cache Consistency," Proc. USENIX Ann. Technical Conf., pp. 141-152,Jan. 1996.
- [6] D. Carney, S. Lee, and S. Zdonik, "Scalable Application-Aware Data Freshening," Proc. IEEE CS Int'l Conf. Data Eng., pp. 481-492,Mar. 2003.
- [7] P. Deolasee, A. Katkar, P. Panchbudhe, K. Ramamritham, and P. Shenoy, "Adaptive Push-Pull: Disseminating Dynamic Web Data," Proc. Int'l World Wide Web Conf. (WWW), pp. 265-274, May 2001.

- [8] J. Cho and H. Garcia-Molina, "Synchronizing a Database to Improve Freshness," Proc. ACM SIGMOD, pp. 117-128, May 2000.
- [9] S. Pandey, K. Dhamdhere, and C. Olston, "WIC: A General-Purpose Algorithm for Monitoring Web Information Sources," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 360-371, Sept.2004.
- [10] J.-J. Lee, K.-Y. Whang, B.S. Lee, and J.-W. Chang, "An Update-RiskBased Approach to TTL Estimation in Web Caching," Proc. Conf.Web Information Systems Eng. (WISE), pp. 21-29, Dec. 2002.
- [11] A. Gal and J. Eckstein, "Managing Periodically Updated Data in Relational Databases: A Stochastic Modeling Approach," J. ACM, vol. 48, no. 6, pp. 1141-1183, 2001.
- [12] J. Cho and A. Ntoulas, "Effective Change Detection Using Sampling," Proc. Int'l Conf. Very Large Data Bases (VLDB), 2002.[9] "CNN Top Stories RSS Feed," [http://rss.cnn.com/services/rss/cnn\\_topstories.rss](http://rss.cnn.com/services/rss/cnn_topstories.rss), 2010.
- [13] L. Bright and L. Raschid, "Using Latency-Recency Profiles for Data Delivery on the Web," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 550-561, Aug. 2002.
- [14] E. Cohen and H. Kaplan, "Refreshment Policies for Web Content Caches," Proc. IEEE INFOCOM, pp. 1398-1406, Apr. 2001.
- [15] V. Padmanabhan and J. Mogul, "Using Predictive Prefetching to Improve World Wide Web Latency," ACM SIGCOMM Computer Comm. Rev., vol. 26, no. 3, pp. 22-36, July 1996.
- [16] Z. Jiang and L. Kleinrock, "Prefetching Links on the WWW," Proc. IEEE Int'l Conf. Comm., 1997.
- [17] J.L. Wolf, M.S. Squillante, P.S. Yu, J. Sethuraman, and L. Ozsen, "Optimal Crawling Strategies for Web Search Engines," Proc. Int'l World Wide Web Conf. (WWW), pp. 136-147, 2002.

## Authors



**V. Anushadevi** has completed B.Tech (C.S.E) from Jayamukhi Institute of technological Sciences and pursuing M.Tech (C.S.E) in DRK institute of science and technology, JNTUH, Hyderabad, Andhra Pradesh,

India. Her main research interest includes Data Mining & Computer Networks.



**Dr. R. V. Krishnaiah**, did M.Tech (EIE) from NIT Waranagal, MTech(CSE) form JNTU, Ph.D, from JNTU Ananthapur, He has memberships in professional bodies MIE, MIETE, MISTE. His main

research interests include Image Processing, Security systems, Sensors, Intelligent Systems, Computer networks, Data mining, Software Engineering, network protection and security control. He has published many papers and Editorial Member and Reviewer for some national and international journals