Towards an Improved Hoarding Procedure in a Mobile Environment

Oluwaranti A.I. (Corresponding Author)
Department of Computer Science and Engineering,
Obafemi Awolowo University, Ile-Ife.
E-mail: aranti@oauife.edu.ng

and Affiah I. U.
Department of Computer Science and Engineering,
Obafemi Awolowo University, Ile-Ife.
E-mail: idywhyte@yahoo.com

Abstract

Frequent disconnection has been a critical issue in wireless network communication therefore causing excessive delay in data delivery. In this paper, we formulated a management mechanism based on computational optimization to achieve efficient and fast computation in order to reduce inherent delay during the hoarding process. The simulated result obtained is evaluated based on hoard size and delivery time.

Keywords: Hoarding Procedure, Mobile computing Environment and Computational Optimization.

1. Introduction

The evolutions of wireless technology as well as the growing capacity of mobile devices have led to an increased in the number of mobile users who access information in a mobile computing environment. However, due to the unreliable nature of connection in mobile environment, disconnected operation in an area of very expensive bandwidth has to be efficiently supported by a mechanism called hoarding (Helal *et al.*, 2002).

Hoarding is the process of caching important and relevant user data onto mobile devices prior to disconnection (Helal *et al.*, 2002). This is done in order to ensure availability of data and to reduce client side latency. However, the main reason for hoarding mobile data is to improve the efficiency of first-time access to information by predicting the information items that will be needed in the near future (Mazen and Frederic, 2006). Despite this benefit, users still encounter problems of predicting in advance their likely needed information/data so as to conserve bandwidth resources, storage capacity and time of data delivery during hoarding process.

In the existing works (Eldosoky *et al.*,2008; Kubach and Kurt, 2001), efforts have been made to formulate a management mechanism for hoarding but not based on computational optimization, thus rendering these approaches inefficient and slow to minimizing inherent delay in data delivery. Also, the major limitation was choosing hoard contents given its limited size in order to minimize future cache misses.

Having examined these problems, this paper addresses the inefficiency of the existing approaches such as Adaptive Hoarding technique (AHT) by introducing a computational optimization function to select the maximum relevant document that can be used to populate the hoard. The proposed method is aimed at increasing the speed of execution of the algorithm as well as knowing the total size of hoard utilized for hoarded data. The proposed computation formulation also takes into account the architecture (Affiah, 2011) that uses a surrogate server in between the content server and the file selector. This architecture incorporates intelligent service portal that takes the storage capacity of the users' mobile device before rendering data into it.

The result of the proposed work was evaluated with AHT with hoard size and delivery time as the performance metrics. The rest of this paper is organized as follows. In Section 2, we presented reviews of related works. We described the model formulation in Section 3. In Section 4, we analyzed the data set. In Section 5, we present the result obtained during simulation and Section 6 offers concluding thoughts and future works.

2. Related works

Quite a number of literatures on hoarding have been consulted but only a few related to this study were considered. This was carried out with a view to identifying the drawbacks of wireless communication such as frequent disconnection which leads to loss of relevant data. This loss of data is often checked by hoarding.

The Coda approach (Kistler andSatyanarayanan, 1992; Mummert, 1995; and Satyanarayanan *et al.*, 1993) was one of the first approaches that used the concept of hoarding to allow disconnected operation in a network file systems. This relies on users' interaction for its operation; it requires a list of preferred information items for each user. Coda used remote procedural call to callback missing data when there is a disconnection. The authors in Tait *et al.*, (1995) proposed an intelligent file hoarding for mobile file systems, called transparent analytical spying. In this technique, the contents of the hoard were determined based on past usage pattern.

Another method is Incremental Hoarding and Integration (*Helal et al.,* 2003).The main idea of this work was to carry out an incremental hoarding and integration by exploiting the fact that users make minimal changes while users are mobile. In this approach, differential transfer of data objects (files, databases etc.) instead of hoarding and reintegrating full data object was achieved. The incremental approach implemented was based on the transfer of file changes in either direction (i.e. from server to client and vice versa). To this end, this method used a version control system to compute and maintain object differentials. The main drawback of this approach was that it largely depends on the similarity of the two versions of a particular file. According to Kuenning and Popek (1997), Automated hoarding method used a semantic distance between file to feed a clustering algorithm that selects the files to be hoarded.

In Saygin *et al.* (2000), the authors proposed another method based on data mining techniques. This latter used Association Rules for determining the information that should be hoarded. The major drawback on semantic distance lies in the fact that the time spent to calculate the closeness of files was enormous. Furthermore, in Mazen, and Frederic, (2006) an infostation-based information hoarding technique was introduced to overcome the difficulties encountered by previous methods. A distributed data storage model was described that proposed many classes of services to give more flexibility to users. This storage model also used file system model to easily manage and exchange data between different entities. An enhanced hoarding approach based on Graph Analysis was investigated by (Susanne *et al.*, 2000). This approach took into account the structured nature of files into consideration. The paper also presented an enhanced hoarding approach for semi structured information that relies in the analysis of graphs to determine information that needs to be hoarded. The drawback is that a huge amount of data is needed to achieve adequate results. The work in Eldosoky *et al.* (2003) observed the user's subject of interest as selection criterion to carry out the hoarding process. The objective of this technique was to reduce future cache misses and increase the network bandwidth utilization.

In summary, within the ambit of the reviewed literature, there have been no efforts made to formulate a management approach for hoarding based on computational optimization. This paper incorporates a Greedy Search algorithm and a linear programming function into an Adaptive Hoarding Technique (AHT) to achieve reduced delivery time of mobile data as well as having a huge amount of hoarded data in the hoard size.

3.0 Model formulation

The underlying principle of the proposed model formulation is based on the optimization function that translated into a linear programming function as shown in equation (1). The objective function is aimed at obtaining the maximum document to populate the hoard. The maximum relevant document was selected as shown in equation (1).

$$\mathrm{Max}\, Z_i = (f_1 * \frac{x_1}{y} + f_2 * \frac{x_2}{y} + \cdots + f_B * \frac{x_B}{y}) \qquad (1)$$

This can also be summarized in the form represented in equation (2).

$$Max\ \ Z_i = \sum_{i=1}^{B}(f_i * \frac{x_i}{y}) \qquad (2)$$

$$Max\ Z_i = n\ (\sum_{i=1}^{B}(f_i * x_i/y) \qquad (3)$$

Subject to
$f_i \geq 0$;represent the i^{th} frequency of the *x* document;
$y > 0$; represent the total document of interest in the server.

Where "Z" is the relevant document score, "n" is an enlarge factor that used to allocate a higher priority to most popular document. "x_i" is the *x* document in i^{th} document server, "f_i" is the corresponding frequency of document x_i, "y" is the total number of documents in the server and B is the total document of interest.

The objective function of equation (3) is to select the maximum relevant documents based on popularity of the web documents. The selected document will be used to populate the hoard for easy access by the client during disconnection.

3.1 The Greedy Search Hoarding Technique (GSHT)

The proposed GSHT incorporated a linear programming function in equation (3) into the existing AHT and a Greedy Search Algorithm.

3.2 Principles and procedure of the proposed technique.

The principle behind the proposed technique is described in Figure (1) However, the procedure involved in the proposed algorithm further described in steps. The hoarding flowchart as described in Figure (1) depicts a graphical representation of step-by-step approach to addressing a delivery delay problem that reduces the time of transfer of document from the servers, as well as acquiring huge number of document in the hoard.

Step 0:
The input phase: In this step, interested items (data) were inputted by the user in order to carry out the hoarding process.

Step 1:
Data searching and extraction of keywords: Relevant domain names (keywords) were extracted from the URLs in the log files. After which non relevant keywords were discarded given the document of interest.

Step 2
Training the data: The URLs were train to obtain the probability (P) for each keyword represented in equation (1) given the document of interest from respective category. This was repeated till all probabilities of each domain names are complete.

Step 3
Testing the data: They data were tested by performing optimization using equation (3). This was with a view to determine the maximum keywords.

4. Data Set Analysis

The Obafemi Awolowo University (O.A.U) web log files as shown in Table (1) were used for the experiment. The web log files were analyzed using a Squid log Analyzer to obtain the relevant URL from the overall web access logs.

The web log files obtained from general computing environment of Obafemi Awolowo University Network (OAUNet) for the experiment was analyzed based on the following steps:

i. **Preprocessing:** This is the process used in extracting useful information from the collectedweb log files. Squid Log Analyzer was used to accomplished this task. The output of preprocessing phase that made up the URL was stored in WordPad and was later compiled into Ms-Excel workbook for easy categorization. The pattern of this access log after preprocessing is displayed in Table 2.

ii. **Feature selection:** Extracted URL from preprocessing process was subjected to screening in order to disregard unwanted data. The feature selection was meant to select only domain names from the corresponding URL of the web access log.

iii. **URL-categorization:** The resulted data from feature selection was categorized for training and testing respectively.

iv. **Bayesian Rule:** This was used to calculate the probabilities of web log file obtained from URL-categorization to be used in the model evaluation.

v. **Model evaluation**: The result of model evaluation yields the maximum relevant document which can be used to populate the hoard.

In summary, the flow of the hoarding process is as shown in Figure 2.

4.1 Hoarding procedure simulation

The simulation was carried out under the following phases:

i. Training phase

In this phase, training is used for the parameter estimation of the model in equation (3). The parameters estimated were maximum probabilities of domain names in each category; this was achieved using the Bayesian rule (Sang-Bum*et al.*, 2006) as shown in equation (4).

$$p\left(\frac{H_i}{A}\right) = \frac{P(A/H_i)P(H_i)}{P(A)} \tag{4}$$

Where the domain of interest "*A*" happens under each category H_i (i = 1 to 7) with a known probability. It is assumed that the probabilities of category H_i (i =1 to 7) are known. Then the conditional probability of categories H_i given document of interest is as stated in equation (4).
However, a total of 2000 web log files were used as dataset for the experiment, out of which 70% of log file were used for training to estimate the probabilities. The available domain name of interest (DNI) for parameters estimation were; .com, .net, .org, .edu, co.uk, .gov, ng.

ii. Testing phase

This phase aimed at producing results of the mathematical model from the remaining 30% of web log files. Categorization of the log files was performed based on the same criteria as was on the training phase. The available domain names of interest as used in the training phase were also repeated. The implementation of this phase was carried out with a suitable simulation tool.

5. Result and discussion

Simulation was performed to evaluate the performance of GSHT against AHT based on delivery

time and hoard size. Furthermore, the simulation result has shown that the GSHT outperforms AHT and achieved a reduced time with a huge amount of files during computation.

5.1 Delivery time

In Figure 3, the evaluation results of GSHT and the AHT is presented. The graph illustrates that as the number of files obtained during testing increases from 32 to 63 the delivery time of the proposed mechanism (GSHT) decreases from 7.2 to 6.7ms while delivery time of the existing mechanism (AHT) increases from 7.5 to 9.7ms given corresponding number of web log files. It was also observed that simulation result obtained showed that the proposed mechanism, Greedy Search Hoarding Technique (GSHT) gave a decrease of about 15.0% over the existing mechanism, Adaptive Hoarding Mechanism (AHT) on delivery time. This implies that there is an improvement on delivery time of the proposed technique over the existing technique.

5.2 Hoard size

Hoard size refers to the available cache space used during the hoarding process. The simulation result shows that the improvement on hoard size of GSHT is about 12.99% over Adaptive hoarding technique. This implies that the hoard size utilized in the proposed mechanism was more than the existing mechanism resulting in an improved hoarding process. The graph in Figure 4 illustrates hoard size versus categories of web log files which showed that the utilized hoard size was more in GSHT than in AHT.

6. Conclusion and future work

As a sequel to the result presented, which shows that GSHT outperforms AHT in terms of delivery time and hoard size, this paper concludes by recommending the GSHT as a better choice for an improved hoarding procedure in a mobile environment. The future work will base on evaluating the algorithm with other relevant performance metrics to further ascertain the efficiency of this approach.

References

Affiah, I. U. (2011). Development of an Improved Management Mechanism for Hoarding in a Mobile Computing Environment. M.Sc Thesis in the Department of Computer Science and Engineering Obafemi Awolowo University Ile-Ife, Nigeria.

Eldosoky, A.E., Mustafa, S., Ahmed, I.S. and Shereen, H, (2008). An adaptive Hoarding Technique (AHT) based on Naïve Bayes Classifier. *Technical Report: Faculty of Computer Science and Information Technology. Cairo University, Egypt.* Availableat http://www.fci.cu.edu.eg/fci-maganine/final accepted paper. php.

Helal, S. A., Zhang,J. and Khushraj, A.(2002). Incremental Hoarding and Reintegration in
Kistler, J.J and Satyanarayanan, M. (1992).Disconnected Operation in the Coda File System. ACM Transaction on Computer Systems, Vol.10, (1):3-25.

Kubach, U. and Kurt, R. (2001).Exploiting Location Information for Infostation Based
Hoarding. *In Proceedings of 7th Annual International Conference on Mobile Computing and Networking. Hongong, China. Pp. 145-157.*

Kuenning and Popek (1997). Automated Hoarding for Mobile Computer. *In Proceedings of the 16th ACM Symposium on Operating Systems Principles,* St. Malo, France. Pp. 264-275.
Mazen, T. and Frederic, W. (2006).Data Exchanging In Hierarchical Infostation Systems.*In Proceedings of Mobile and Wireless Communication.*, pp. 361-365.

Mobile Environments.*In Proceedings of the Symposium on Applications and Internet (SAINT).*
Mummert, L.E, Ebling, M.R, Satyanarayanan, M. (1995). Exploiting Weak Connectivity For Mobile File Access. *In Proceedings of the 5th Symposium on Operating System Principles, Copper Mountain Resort, Colorado*. Pp. 143-155.
Pp. 8-12.

Sang-Bum, K, Kyoung-Soo, H., Hae-Chang, R. and Sung Hyon, M. (2006). Some Effective Techniques for Naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18(11):1457-1466.

Satyanarayanan, M, Kistler L, Mummert, B. and Ebling, R. (1993). Experience with Disconnected Operation in a Mobile Environment. *Proc. of the Usenix Mobile and Location Independent Computing Symposium: USA*, pp. 11-28.

Saygin, Y., Ulusoy, O. and Elmagarmid, O. (2000).Association Rules for Supporting Hoarding in Mobile Computing Environments.*In Proceedings of the 10th International Workshop on Research Issues in Data Engineering (RIDE)*, pp. 71-78.

Susanne, B., Pedro, J.M and Kurt, R (2003).An Enhanced Hoarding Approach Based on Graph Analysis.University of Stuttgart Institute of Parallel and Distributed Systems, *Symposium on Reliability in Distributed Software and Database Systems*, pp. 12-21.

Tait, C., Lui, H., Acharya, S. and Chang, H. (1995). Intelligent File Hoarding for Mobile Computers.*In Proceedings 1st international conf. on mobile computing and network, Berkeley, CA, USA.pp.* 119-125.

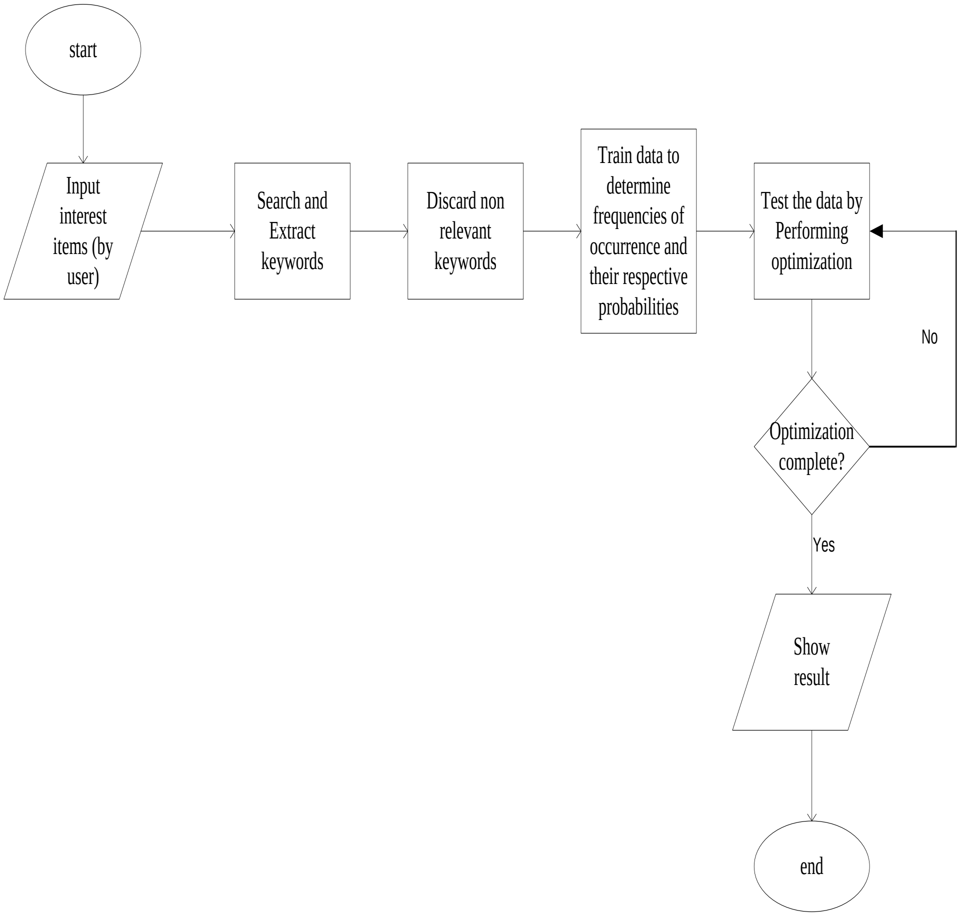
start
Input interest items (by user)
Search and Extract keywords
Discard non relevant keywords
Train data to determine frequencies of occurrence and their respective probabilities
Test the data by Performing optimization
No
Optimization complete?
Yes
Show result
end

Figure 1: Flowchart of proposed procedure

Table 1. Obafemi Awolowo University web access log.

```
1290990308    37608  10.105.35.99 TCP_MISS/200 15145 GET
http://answers.yahoo.com/question/index?wewasola DIRECT/69.147.100.227 text/html.
1290990308    13246  10.105.35.40 TCP_MISS/200 4580 CONNECT urs.microsoft.com: 443
wewasola DIRECT/213.199.177.157
1290990308    739      10.105.35.99 TCP_HIT/200 2232 GET
http://www.ncbi.nlm.nih.gov/core/ext.ncbi/1/css/themes/xtheme-oldentrez.css wewasola
NONE/- text/css.
```

Table 2. Sample of O.A.U preprocessed web access log.

http://answers.yahoo.com
http://netque.oauife.edu.ng
http://www.ncbi.nlm.nih.gov
www.facebook.com

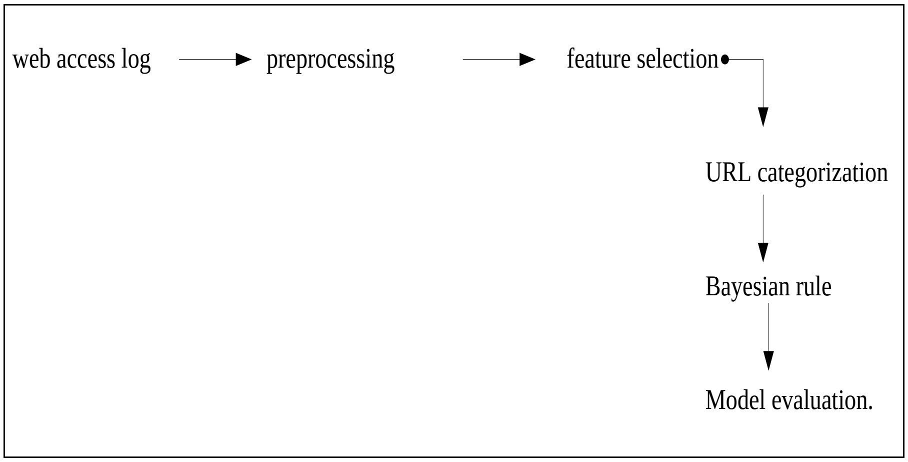

Figure 2: Schematic representation showing the flow of the hoarding process.

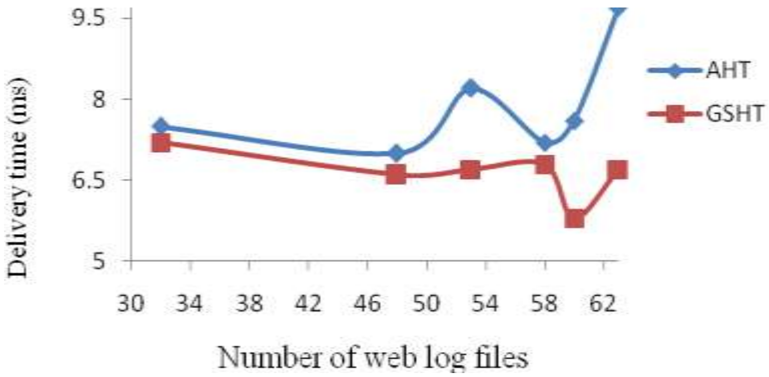

Figure 3: Graph of delivery time against categories of web log files

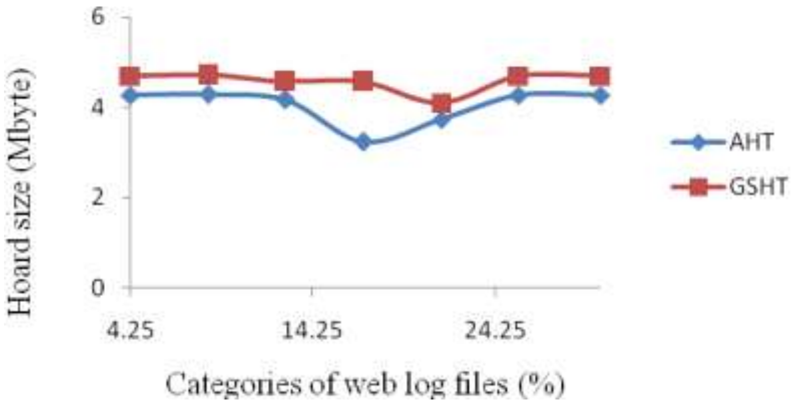

Figure 4: Graph of hoard size against categories of web log files