Quantitative Models for Performance Enhancement of Information Retrieval from Relational Databases

Jenna Estep
Corvis Corporation,
Columbia, MD 21046

Natarajan Gautam
Harold and Inge Marcus Department of Industrial and Manufacturing Engineering,
Pennsylvania State University,
University Park, PA 16802

Abstract

We consider a collection of relational databases that are accessed by users with different profiles. We develop optimal retrieval strategies and system designs to present to the users their required information from relevant databases. We use queuing-theory-based analytical models as well as simulations to obtain system performance measures.

Keywords: Information retrieval, customer profile, queuing models, simulation, response time, caching, pipelining.

1. Introduction

Consider a system where a user makes a query and also lists his or her profile. Based on the query and the profile, the system retrieves an appropriate set of databases and presents to the user. There are several such systems that are both currently deployed and also in consideration for future deployment. One such instance is while making a query to a web search engine or a library catalog wherein if the user also specifies his/her profile, only the most relevant documents could be retrieved. For example if a student enters “Conference” and “Quality” in the search field and in the profile field he/she enters, “student”, “Industrial Engineering” and “Eastern US”, then he/she could expect a list of conferences in the Easter United States that encourage student papers on the topic “quality” (Industrial Engineering related, not other disciplines). Another such example is where a customer is interested in a map of a certain region that is customized so that besides the street information, the restaurants catering to his/her profile are also marked on the map. There are also several Military applications where especially during combat, instead of overloading the user with information only the most appropriate ones based on the user profile can be presented. In fact this research study stemmed out of one such application that Raytheon Systems Inc. was developing.

One key to succeeding in this endeavor is to provide greater accuracy in retrieving the most relevant information for users based on their profiles. For example, displaying information that is truly of interest to users in an e-business system can increase sales by directing customers to products more suited to them, while creating a larger customer base because customers will be attracted to a website that is “customized” to match their profile. Instead of overloading the user with information, most of which is of no consequence, it would be beneficial to customize a web page with information relevant to the user. One such system is available in Yahoo [7]. Whenever a user logs into the Yahoo site, the system looks at the profile on record and displays information or data that is relevant to that user, including advertisements, local attractions, etc.

These systems currently operate as follows: the user enters the main search terms and then his or her profile. Then the user submits the query and the system calculates using Bayesian networks (see [2] and [6]) the most appropriate set of documents to retrieve. This is done using a standard Bayesian Network software package such as Hugin [4]. The system rank-orders the documents, retrieves them and presents them to the user. There are two main concerns in these systems, (i) how to select the most relevant documents based on the user profile, and (ii) how to speed up the retrieval process? In this paper we will address the latter concern keeping in mind that the user would spend a lot of time entering the profile information and therefore would be quite impatient to get the query results. To improve the response time to user requests, we consider a few techniques such as starting to retrieve documents simultaneously as the user types up his/her profile, storing some popular documents in a cache, and, pipelining the retrieval process. The main contribution of this research can be summarized as follows:

- A quantitative modeling tool to evaluate the performance benefits for the various enhancement schemes. Thereby several what-if analyses can be easily undertaken to determine the enhancement in response times by considering several factors.
- Bringing together the enhancement techniques (caching, pipelining and intermediate retrievals) under one framework that have been traditionally considered in isolation. This way the combined benefits of all the schemes can be simultaneously considered.

This paper is organized as follows: the first enhancement of retrieving the databases simultaneously while the user is entering his/her profile as opposed to waiting until the user completes entering the entire profile is modeled in Section 2. In Section 3 an investigation of how the response time can be reduced by using a cache and pipelining is undertaken. We present our concluding remarks and make suggestions for future work in Section 4.

2. Enhancement: Intermediate Retrievals

The current implementation of the system (that we will call "current system" henceforth) operates as follows: the user enters his/her entire profile, and then the processor computes (computing time) the best databases using the profile information, retrieves the most relevant databases (retrieval time), and finally transmits and displays the most relevant databases. The total time to complete these activities defines the time in system, or response time. However, this existing system has several demerits. One of the major problems is its long response time, i.e. the time from the point when the user enters the profile until the most relevant databases are displayed.

The goal in this section is to investigate methods of reducing the retrieval time. The current system has user profile that can be segregated into different profile characteristics. Consider a system where the computing and retrieval based on the first profile characteristic can begin as soon as the user enters that first characteristic. That way, while the second characteristic is being entered, the computing and retrieving based on the first characteristic can occur. The idea behind this enhanced system is that the most relevant databases based on the first characteristic will not differ that much from those based on a combination of the first and second characteristics. For example, suppose that the five most relevant databases based on characteristic one are databases A, B, D, E, and G. Now suppose that the most relevant databases based on characteristics one and two are A, C, D, F, and H. Assuming that the user takes a long time to enter characteristic 2, databases A and D would have already been retrieved, so the system must discard the other three and replace them by retrieving databases C, F, and H. All of this is occurring while the user is still entering his/her profile. In the experiments we consider it is possible for the next user characteristic to be typed before all the databases are retrieved for the previous characteristic. By the time the last characteristic is entered, perhaps only one or two new databases will have to be retrieved.

Note that in the current system, all databases are retrieved only after the last profile characteristic is entered. The worst this intermediate retrieval enhancement can perform (i.e. when the last characteristic requires retrieving all databases afresh) is as good as the current system. For most practical applications, at least 2 or 3 documents can be retrieved while the user is typing up characteristics. However note that this enhancement can result in excessive computational time as a computation phase is undertaken every time the user enters a characteristic.

2.1 Analysis: Current System

The database retrieval process has four stages: keyboard entry, computing, retrieving, and transmit & display. From now on, for convenience, we assume that the user has to enter five characteristics. Once the user enters the 5 characteristics, the computing and retrieving the five most relevant databases process starts. Finally, the databases selected are transmitted and displayed. Note that there are five profile characteristics and 5 databases only for ease of description. However, this can easily be relaxed so that the results can be generalized to cases with *n* profile characteristics. The response time for this model can be represented by the random variable T_1, which is a function of the random times required for each of the four stages in this process:

- K = time to enter entire profile (i.e. all five profile characteristics)
- C = time to compute the five most relevant databases
- R = time to retrieve the five most relevant databases
- D = time to transmit and display the most relevant databases.

Therefore, the total response time, T_I, can be expressed as follows:

$$T_I = K + C + R + D.$$

This response time will be compared with the response time for the intermediate retrieval system in order to illustrate that the intermediate retrieval system will reduce the time in system. This enhanced system is described in the following section.

2.2 Analysis: Enhanced System with Intermediate Retrievals

Define the following random variables:

- K_t = time to input profile characteristic t, where t = 1, 2, 3, 4, 5
- C_t = time to compute the most relevant databases based on characteristics 1 to t, where t = 1, 2, 3, 4, 5
- R_t = time to retrieve remaining relevant databases based on characteristics 1 to t, where t = 1, 2, 3, 4, 5
- D = time to transmit and display the most relevant databases.

The random variables K_1 through K_5 and C_1 through C_5 are iid. The time parameters for using the keyboard (inputting a characteristic) are the same for each characteristic. However, R_1 through R_5 are not independent. Using these random variables, the expression for T_{II} is

$$T_{II} = K_1 + K_2 + K_3 + K_4 + K_5 + C_5 + R_5 + D.$$

The assumption here is that two different processors are being used such that simultaneously doing computing and retrieving does not slow the processing rate. To compare the above formula with that for T_I, note that the time parameters for using the keyboard (K_1 through K_5) are the same for each characteristic. Therefore, the time to input all five characteristics $K_1 + K_2 + K_3 + K_4 + K_5$ will be equivalent to the time to enter the entire profile, defined as K in T_I. In addition, C_5 is equivalent to C because this model assumes that the computing time does not depend on the number of characteristics entered. The only difference in response time is due to the difference between R and R_5. Clearly, R_5 will be less than R, and T_{II} is stochastically smaller than T_I. Simulation is used to show that this is indeed true and to quantify the reduction in response time achieved by using the intermediate retrieval system.

2.3 Results

Several input models in terms of random variables, relevant database computation and parameter values have been used to illustrate the enhancement in performance of the intermediate retrieval scheme. However due to space restrictions only one example is provided here. In a forthcoming paper we will illustrate other examples as well.

System	Average Computing Time	Average Time in System
Current	0.49891	52.531
Intermediate retrieval	2.5007	43.483

Table 1: Simulation results for current and enhanced systems

With 95% confidence, there is a significant difference in the mean time in system between the current system and the enhanced system with intermediate retrievals. From Table 1, the average response time for the enhanced model is 9.04 seconds lower than the that of the current model. This is a fairly significant difference. However the computing time for the intermediate retrieval system is more than 5 times higher.

3. Enhancement: Caching and Pipelining

In this section we investigate the use of caching as well as pipelining mechanisms to enhance the performance of the current system. The current system requires that all users obtain all their information from disks (main memory). One way to decrease the response time is to store and retrieve information from a cache, which is typically much faster than disk retrieval. The constraint is that the amount of information that can be cached is limited. The caching policy considered in this paper is that when each user exits the system, the five documents that were retrieved are stored in the cache for possible use by the next user. We compare these techniques based on caching against the current system. Note that the intermediate retrieval of databases discussed in Section 2 is not included in the comparisons.

Two situations are considered. The first, called the *sequential service*, allows only one user in the system and any new users arriving should have to wait until the databases are displayed for the current user. In the second case, called *pipelined service*, several users can be in the system at various stages of the retrieval process (such as keyboard entry, computing/retrieving and transmission/display). We will first analyze the sequential service system followed by the pipelined service system. Note that both systems use caching strategies. They will be compared against the current system that is essentially the sequential service system with no caching.

3.1 Sequential Service

The time in system or the response time for a user who is about to start entering his/her profile is a random variable T. Clearly T is the sum of the profile entry time, computation time, retrieval time, and transmission/display time. Define the following variables:

- T_1 = time to enter profile
- T_2 = computing time to determine 5 most relevant databases
- T_3 = time to retrieve 5 most relevant databases
- T_4 = transmission and display time.

Since the above times T_1, T_2, T_3, and T_4 are independent,

$$T = T\_1 + T\_2 + T\_3 + T\_4$$

$$E(T) = E(T\_1) + E(T\_2) + E(T\_3) + E(T\_4)$$

$$Var(T) = Var(T\_1) + Var(T\_2) + Var(T\_3) + Var(T\_4)$$

The random variables T_1, T_2 and T_4 have a general distribution, with a given mean and standard deviation. In order to obtain T_3, it is required to know the number of databases to retrieve from the cache and the number to retrieve from the disk. Therefore, we define:

- D = time to retrieve one database from the disk, with mean β and variance S_β.
- C = time to retrieve one database from the cache, with mean α and variance S_α.

For the current system, E(T_3) = 5β and Var(T_3) = 25 S_β. These values can be used to find E(T) and Var(T). However for the sequential service system with cache, we need to first know the probability distribution of the number of documents to retrieve from the disk in order to compute E(T_3) and Var(T_3). Let p_i be the probability that i of the five databases must be retrieved from the disk (based on the user profile). We assume that the cache size is five and therefore all the cache contains are the five documents retrieved by the previous user.

We consider several models for p_i values but present only one here due to space restrictions (others will be included in a forthcoming paper). Researchers [1] have shown that the popularity of web documents follows a Zipf-like distribution. If there are M documents, then the probability that document i will be accessed is K/(i^n), where r is a parameter which Zipf used as one, but other values can be used, too. K is a normalizing constant such that ∑K/i^n = 1. Therefore, document one is most popular and document M is the least popular. We use this distribution to get p_i values. Based on the p_i values we compute E(T_3) and Var(T_3) for the sequential service system via conditioning on the number of documents to be retrieved from the disk (detailed results will appear in a forthcoming paper).

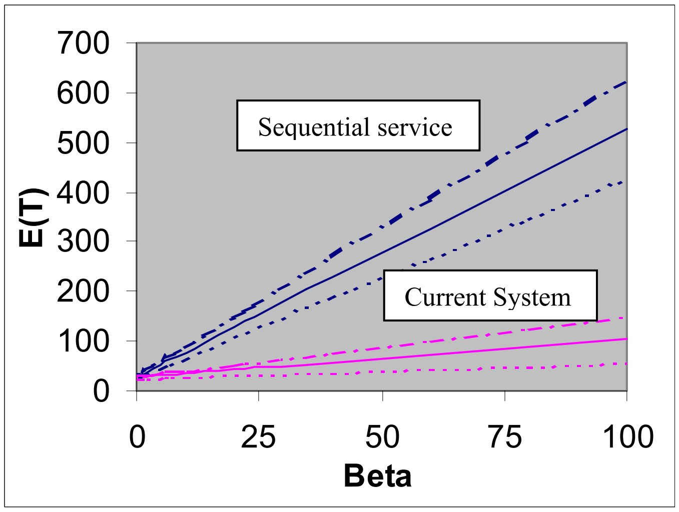


Figure 1: Comparing the sequential service system with the current system

In order to compare the sequential service system with the current system, E(T) and one standard deviation from E(T) are plotted for both systems over a range of β values, where β is the mean time to retrieve one database from the disk. The comparison is illustrated in Figure 1. The decrease in time in system is very large when using a cache, with a reduction of about 40 percent for mid-range β-values and 80 percent for high β-values.

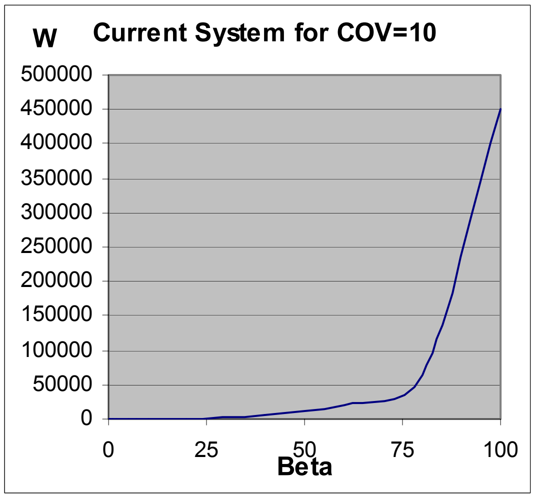
3.2 Pipelined Service

In Section 3.1 we only compared the time between entering the profile and displaying the databases. In this section we are going to consider a queuing model where users arrive according to a Poisson process with mean rate λ into the systems and wait for any users in service before beginning their service. All queues will be served according to a first-come-first-serve basis. Another way to think about Section 3.1 is that the mean rate λ is so small that there is never a user waiting for another to complete.

We first describe the performance of the sequential and the current systems and then move on to the pipelined service case. For the sequential and the current systems the waiting room for users is outside the system (i.e., before entering the profile). Both systems can be analyzed using M/G/1 queuing models. The service times are indeed the T values described in Section 3.1. We assume that the service times are iid. Therefore we use the Pollaczek-Khintchine (PK) formula and obtain the mean queue length L (see [5]). The expected wait in the system, W, (which is the expected time in system including time in the queue) can be obtained via Little's formula: $L = \lambda W$.

For the pipelined service system, the model is similar to the sequential service system in terms of caching, with the exception that it involves some simultaneous processing, which reduces the time in system. Multiple users enter profiles on their keyboards (using their own browsers). When the profiles are submitted, they are processed (involving computing and retrieving of databases) one user at a time. The relevant databases can then be transmitted and displayed simultaneously for the various users. A tandem queuing network with 3 stages back to back is used to model the system.

The total time in the system is equal to the sum of the times spent in the three stages: keyboard, computing/retrieving and transmission/display. Since all the users can potentially enter their profiles in parallel, the first stage of the queuing network is modeled as an M/G/∞ queue. The keyboard time is equal to the time to enter all five user characteristics and has the same general distribution for all servers. Since the output process from an M/G/∞ queue is a Poisson process with parameter λ [3], the second stage that involves computing and retrieving the five most relevant databases can be modeled as an M/G/1 queue. The mean of the time in this stage can be computed using a conditional argument (via p_i values described in Section 3.1) and the PK formula [3]. Finally the mean of the time spent in the third stage (i.e. transmission/display) can be directly computed since information can be simultaneously transmitted and displayed on all the user machines.



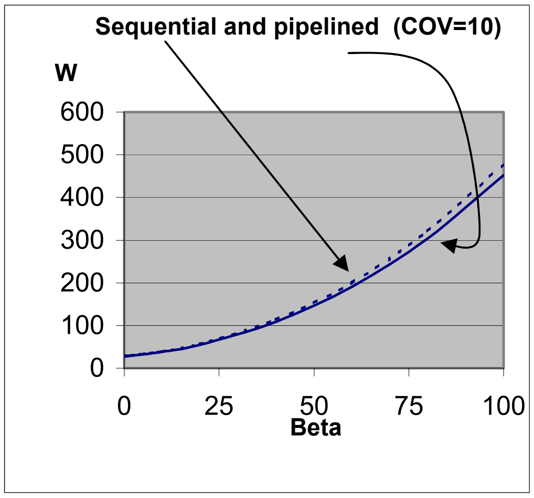


Figure 2: Comparing the pipelined, sequential and the current systems

We compare the pipelined service system with the sequential (Section 3.1) and the current systems. The expected time in system, W, is plotted against the range of β-values. W is plotted for different coefficients of variation (COV). The retrieval time is proportional to the document size. It is said that the size of web documents can have a very high COV [8]. Therefore we use COV values of 10 and show that the caching schemes (both pipelined and sequential) perform far better than the current system. Also, the pipelined system performs slightly better than the sequential case in our example. See Figure 2 for an illustration of the results.

4. Conclusions and Future Research

There are two directions for improving the current implementation of retrieving relevant databases based on a user profile. One is changing the structure of the Bayesian networks that are used to compute and retrieve the most relevant databases (we will present this in a forthcoming paper), and the other involves ways to reduce the user's time in system. We showed two different ways to reduce the user's time in the system: by intermediate retrievals and by caching/pipelining. Using simulation, it is found that the use of intermediate retrieval of databases can significantly reduce the user's time in system compared to the current implementation, although it does require more computing time. The reduction is greatest when the mean retrieval time per database is higher (5 seconds), resulting in a response time that is 9 seconds lower (17% reduction).

Finally, we investigate various models to reduce the response time through the use of a cache and pipelining for one or multiple users in the system at a time. When a maximum of one user is in the system at a time, the model using a cache (sequential service) has about 40 to 80 percent lower time in system than the original model while using a Zipf-like distribution for popularity of databases. When there can be more than one user in the system at a time, a system with parallel service in addition to using a cache is introduced. Users can simultaneously enter their profiles, and the most relevant databases can be transmitted and displayed simultaneously on the multiple users' machines. The models using cache always have a lower time in system when compared with the current system, and pipelined service is even slightly lower than sequential.

There are several areas where further research can be done. The first is to vary the cache size to find the optimal cache size while considering time in system and use of the computer's resources. One design issue is how to optimally allocate capacity to each stage in the process if only one CPU is used. Another issue is how to determine the steady state cache contents. All of these techniques can be used in Internet applications, which are expanding at a high rate in today's economy.

Acknowledgements

The authors thank the anonymous reviewers for suggestions that led to considerable improvements in the content and presentation of this paper. The authors are also grateful to Raytheon Systems Inc. for partially supporting this research.

References

1. Breslau, L., Cao, P., Fan, L., Phillips, G., , Shenker., 1999, “Web Caching and Zipf-like Distributions: Evidence and Implications”, *INFOCOM-99*, 126-134.
2. Charniak, E. 1991, “Bayesian networks without tears”, *AI Magazine*, Winter, 50-63.
3. Gross, D., and Harris, C.,M. 1998, “Fundamentals of Queuing Theory” Third Edition. John Wiley & Sons, Inc.
4. Hugin Lite 1999, <http://www.hugin.dk/>.
5. Kulkarni, V.G., 1995, “Modeling and Analysis of Stochastic Systems”, Chapman & Hall, New York.
6. Pearl, J., 1988, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference” Morgan Kaufmann, San Mateo, CA.
7. Yahoo, 1999 <http://www.yahoo.com>.
8. Zhang, Izmailov, Reininger, Ott, 1999, “Web Caching Framework: Analytical Models and Beyond”, *IEEE Workshop on Internet Applications*, 132-141.