A Probabilistic Model for an Internet Search Engine

Angel Kuri Morales

Instituto Tecnológico Autónomo de México, México D.F., México,
akuri@itam.mx

Abstract. We discuss a probabilistic model of an information retrieval system in which large amounts of information are to be searched and processed which satisfy a boolean predicate. In particular, we are interested in the case where one wants to access all documents from a list of selected words (or terms) which are logically linked by a set of boolean operators. It is assumed that the system is servicing a large number of users and that it operates in real-time.
This is the typical Internet search engine mode of operation. Therafter, a time-sharing operating system assigns finite segments of time to each query. If the predicate is fully processed, the user gets the title of those documents satisfying the predicate; otherwise, the semi-processed query is queued and awaits for further processing time. We call each finite segment of alotted time a quantum of service. This information retrieval system is further assumed to consist of two processors: one for the selection of terms and one for the compliance of the predicates. This arrangement allows for a more efficient overall service time.
As a result of the normal operation of any such system two queues are formed: one in front of the term selecting CPU (which we shall denote as CPU_1) and another in front of the predicate selecting CPU (which we shall denote as CPU_2).
We assume that the interarrival time of the queries and the processing time of both CPUs are random variables. We also assume that these variables are Poisson distributed. Given these assumptions, we derive the equations for the expected number of tasks, the expected waiting time, the number of tasks in the system, the number of tasks being serviced in the system and the number of tasks waiting in the system.

Keywords: Search engine, information retrieval, queueing system, Poisson distribution.

1 Introduction

In this paper we introduce a probabilistic model of an information system of documents in an inverted file. In such system the file of indices contains an entry for each of the terms of the index determined as a descriptor of the data file. Each entry in the index file contains a pointer to an inverted list of pointers in the file of terms. The pointers in this list, in turn, point to all documents in the data file which contain the term of the corresponding index. A query to an information system of documents in an inverted file has the form of a boolean expression of terms in the index. As a response to the query, the system makes an access to the lists corresponding to the terms in the index; merges and selects, from the merged list, those documents which satisfy the boolean expression.

This organization of files has been broadly analyzed and is used in several well known systems [1]. Their behaviour, measured as a function of the required storage and the average response time, has been estimated for different levels of complexity of the queries [2].

Our aim is to model the part of the system which performs the access to the lists and their merged sets. For this purpose it is sufficient to model the system schematically as shown in figure 1. The inverted lists are stored in disk. In the course of processing a query, the lists corresponding to the terms of the index in the query are read to memory and merged. The resulting list is written back to disk or delivered to the user.

2 General Considerations

We shall assume that the amount of main memory assigned to a user during the processing of a query remains constant. Let L be the full length of all the lists corresponding to the desired index terms during the processing of a users query. If L is less than the maximum alotted memory space M, the user will receive a memory space of size L and its full list will be read into main memory in one step. If the lists corresponding to the terms of the index of a query require a space larger than M the user will need additional processing.

We shall also assume that the arrivals of the users are Poisson, statistically independent. Let N be the number of terms in the index of a particular query. For different users i, j we assume that the number of terms (N_i, N_j) is statistically independent. Further, we assume that for any given user the lengths of the lists corresponding to the N terms of the index are statistically independent, identically distributed random variables. It is obvious that such assumptions are only valid in systems where the number of terms in the index of any given query is small as compared to the total number of terms in the index and that the lists are written onto disk randomly.

A user will join the service queue of the disk as soon as the disk addresses of the lists in the query have been determined. After main memory is assigned to a user, these lists will be transferred from disk to main memory. At this time, the user will enter a queue in front of the processor. Its function will be to merge the lists.

In the case where $L \leq M$ the resulting list will be delivered for output and the user will abandon the system. If $L > M$ the list in memory will be merged with an intermediate list residing in disk and it will either leave the system or will be stored in disk.

We shall call the time needed to complete the tasks of loading a users memory, merging the lists and, in the case where $L > M$, storing in disk, a *quantum of*

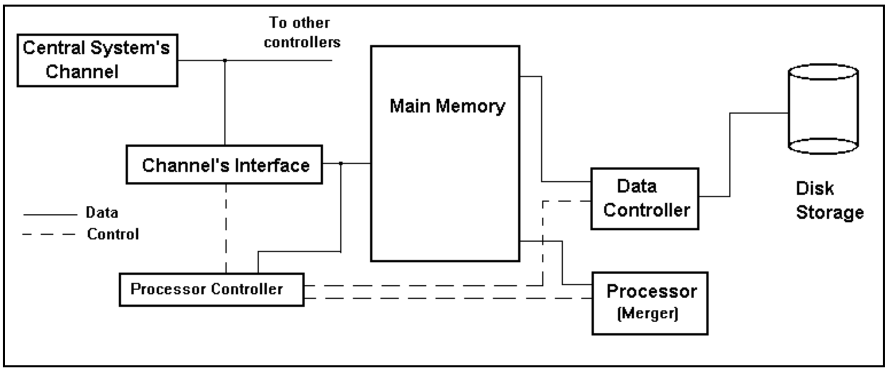

Fig. 1. Information Retrieval System

service or, simply, a *quantum.* If $L > M$, the user will return to the disk queue after his first quantum.

The length of a quantum is a random variable. Its probability density function depends on the number and length of the lists in memory, as well as on the length of the intermediate resulting list (for $L > M$) produced in previous quanta. Since every memory release increases the length of the resulting list, the length of the quantum will also increase as a function of previously received quanta.

Let $\nu(m)$ be the probability density function (pdf) of the number of received quanta m.Also, let $\varphi(n) = Pr(m \geq n)$. Theoretically, it is possible to derive $\nu(m)$ from the pdfs of the number of terms of the index N in the query, the lengths of the lists and the maximum space in memory M. When $N >> 0$ and the average length of the lists is small as compared to M then $E(m) = \frac{L}{M}$ and $Var(m)$ is much smaller than $E(m)$ (See [3]).

In the system to be modeled we allow a service queue before the disks to form. The process of the disk consists of the search of all the index terms in the query. When all these terms have been located, the resulting list will be downloaded to memory and a second queue will form in front of the processor which will search for those documents which satisfy the boolean relation in the query.

When the processor ends execution, if $L \leq M$ the resulting list will be delivered to the user. When $L > M$ the intermediate list will be written in disk and it will re-enter the first queue to receive additional quanta. For the second and next quanta the logical comparison of the processor will be between the new index elements and the intermediate list until all index terms have been processed.

The system under study may be schematically represented as shown in figure 2. Analogous systems have been studied before, but always assuming that the length of the quanta is independent of the previous quanta in is independent of the previous quanta in processes where $L > M$ [4].

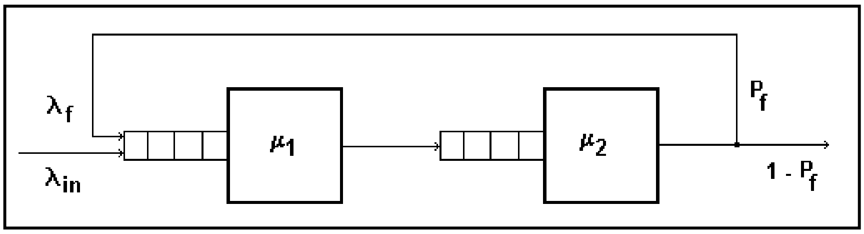

Fig. 2. System 2

Let

$$b_{ij}(t) = \mu_{ij} e^{-\mu_{ij} t}$$

denote the pdf of the service time of a user, where the first subindex denotes the processor and the second denotes the number of received quanta, plus one. We shall call the disk unit a "processor" in the broadest sense of the term.

We denote the probability that a user is about to receive his i-th quantum with ν_i

Let

$$a(t) = \lambda_{in} e^{-\lambda_{in} t}$$

be the pdf of the interarrival time and let

$$f(t) = \lambda_f e^{-\lambda_f t}$$

the pdf of the feedback time, such that

$$f(t) = f[a(t), b_{1j}(t), b_{2j}(t)]$$

3 The model

We shall call the system of figure 2 "System 2". System 2 cannot be described by the models developed before (See, for instance, [5], [7], [8], [9]). We shall model system 2, in turn, with the model shown in figure 3, which we call "System 3".

All the parameters, as pointed out in section 2, are applied to system 3. In system 3 there are N queues with two processors and N pdfs for interarrival times.

Clearly, the original parameters have to be used with a particular methodology to achieve that, in fact, system 2 and system 3 be equivalent. To this end, we shall handle the parameters of system 2 such that the parameters of system 3 may be gotten from the former and that both systems works similarly, as will be shown.

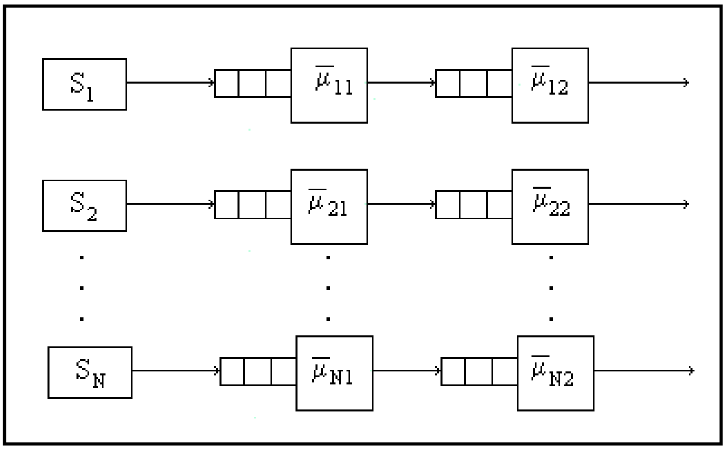

Fig. 3. System 3

3.1 The Sources of System 3

In figure 3 we show several "sources" s_i which we want to behave generating arrivals to the system in such a way that their characteristics in number and interarrival times simulate exactly the characteristics of number and interarrival times of the tasks arriving to the queue of the first processor in system 2.

Let

$$a_j(t) = \lambda_j e^{-\lambda_j t}$$

be the pdf of the time of interarrival of those tasks arriving to system 3 when generated by source s_i. λ_j is defined as

$$\lambda_i = \begin{cases} \lambda_{in} & i = 1 \\ \nu_i \lambda_f & i > 1 \\ & i = 1, 2, ..., N \end{cases}$$

for the i-th tandem of system 3. This is true because the first tandem will only service new arrivals (i.e. those tasks receiving their first quantum) whereas the $N - 1$ remaining tandems will receive ν_i parts of the tasks being fedback to the system.

For convenience we shall make $\phi(i) = \phi_i$. From system 3 and the fact that interdeparture times are exponentially distributed with the same parameter as the interarrival times (Burke's theorem [6]) we can see that

$$\lambda_f = \lambda_{in}\phi_1 + \lambda_f \nu_2 \phi_2 + ... + \lambda_f \nu_{N-1} \phi_{N-1}$$

and

$$\lambda_f = \lambda_{in}\phi_1 + \lambda_f \sum_{i=2}^{N-1} \nu_i \phi_i$$

so that

$$\lambda_f - \lambda_f \sum_{i=2}^{N-1} \nu_i \phi_i = \lambda_{in} \phi_1$$

$$\lambda_f = \frac{\phi_1}{1 - \sum_{i=2}^{N-1} \nu_i \phi_i} \lambda_{in}$$

The last equation may be put as a function of ϕ_m exclusively since $\nu_i = \phi_{i-1} - \phi_i$ [1]. We can then write

$$a_t(t) = \lambda_i e^{-\lambda_j t} \quad (1)$$

where

$$\lambda_j = \begin{cases} \lambda_{in} & i = 1 \\ \frac{\phi_1(\phi_{i-1} - \phi_i)}{1 - \sum_{j=2}^{N-1} (\phi_j \phi_{j-1} - \phi_j^2)} & i > 1 \end{cases} \quad (2)$$

Therefore, we can see system 3 as a system of N tandems where each tandem has an associated source which generates arrivals as per (1). Thus, we have eliminated feedback and system 3 is nothing but a set of tandems in parallel, each with an associated pdf.

It should be clear that, since system 3 consists of $2N$ processors coexisting in time, its processing capacity will be highly increased relative to system 2. We must introduce further modifications to system 3 so that both systems are equivalent. In order to do this, we shall define a new access discipline to the tandems of system 3.

When a task arrives to the system it is "tagged" in such a way that, when joining the first queue of the i-th tandem, it will not receive service from the first processor until all the tasks present in any of the queues of the first processors of the N tandems and which would have been tagged before, have been serviced. A similar discipline will be observed for the queues of the second processor ot the i-th tandem. Thereafter, the behavior of the system of N tandems, over a large enough period of time, will be analogous to that of system 2.

However, this discipline disallows the application of the results of queueing theory regarding queues in series, since these have been obtained assuming constant flow though the system. To avoid this restriction, we shall now model the tandems of system 3 as follows.

3.2 The Tandems of System 3

Let

$$s_{ij} = \mu_{ij} e^{-\mu_{ij} t} \quad \begin{matrix} i = 1, ..., N \\ j = 1, 2 \end{matrix} \quad (3)$$

[1] For instance, $\nu_4 = Pr(\text{being in the } 4^{th}\text{quantum})$ is equivalent to $\varphi_3 - \varphi_4 = Pr(\text{needing more than 3 quanta}) - Pr(\text{needing more than 4 quanta})$.

be the pdf of the service time of the j-th processor of the i-th tandem. We denote with ω_{ij} the probability that the j-th processor of the i-th tandem is busy. If we observe system 3 for a long enough time, we can speak of an average parameter of service $\overline{\mu_{ij}}$ such that

$$\overline{\mu_{ij}} = \omega_{ij}\mu_{ij} \tag{4}$$

Let

$$\overline{\rho_{ij}} = \frac{\lambda_i}{\mu_{ij}} \begin{array}{l} i = 1, ..., N \\ j = 1, 2 \end{array}$$

be the utilization factor of processor ij of system 3. For the set of $2N$ processors we may obtain the expected value of the utilization factor as follows:

$$\overline{\rho_j} = \sum_{i=1}^{N} \frac{\lambda_i}{\mu_{ij}} \nu_i \quad j = 1, 2 \tag{5}$$

Thus, the probability that processor i of system 2 is busy is equal to $\overline{\rho_i}$. From this we can find ω_{ij} since

$$\omega_{ij} = \nu_i \overline{\rho_j} \begin{array}{l} i = 1, ..., N \\ j = 1, 2 \end{array} \tag{6}$$

and, therefore,

$$\overline{\mu_{ij}} = \nu_i \overline{\rho_j} \mu_{ij} \tag{7}$$

Replacing (7) in (5) we get

$$\begin{array}{l} \overline{\rho_j} = \sum_{i=1}^{N} \frac{\lambda_i}{\nu_i \overline{\rho_j} \mu_{ij}} \nu_i \; j = 1, 2 \\ \overline{\rho_j} = \frac{1}{\overline{\rho_j}} \sum_{i=1}^{N} \frac{\lambda_i}{\mu_{ij}} \quad j = 1, 2 \end{array}$$

and

$$\overline{\rho_i} = \sqrt{\sum_{i=1}^{N} \frac{\lambda_i}{\mu_{ij}}} \; j = 1, 2 \tag{8}$$

from which

$$\omega_{ij} = \nu_i \sqrt{\sum_{k=1}^{N} \frac{\lambda_k}{\mu_{kj}}} \begin{array}{l} i = 1, ..., N \\ j = 1, 2 \end{array} \tag{9}$$

and, finally,

$$\overline{\mu_{ij}} = \nu_i \sqrt{\sum_{k=1}^{N} \frac{\lambda_k}{\mu_{kj}} \mu_{ij}} \begin{array}{l} i = 1, ..., N \\ j = 1, 2 \end{array} \tag{10}$$

In this fashion we can work with a pdf of the service time given by

$$\overline{s_{ij}} = \overline{\mu_{ij}} e^{-\overline{\mu_{ij}} t} \tag{11}$$

such that a constant flow discipline is applied.

Using equations (1) and (9) for system 3, we find a model which behaves, on the average, as system 2 and we may apply known results from queueing theory.

4 Application of the Model

4.1 Expected Number of Tasks

The expected number of tasks in the i-th tandem is given by:

$$E(n) = \frac{\rho_{i1}}{1-\rho_{i1}} + \frac{\rho_{i2}}{1-\rho_{i2}} \tag{12}$$

where $\rho_{ij} = \frac{\lambda_i}{\overline{\mu_{ij}}}$; with λ_i and $\overline{\mu_{ij}}$ defined as above.

For the whole system, the expected number of tasks is given by

$$E(n) = \sum_{1=1}^{N} \left(\frac{\rho_{i1}}{1-\rho_{i1}} + \frac{\rho_{i2}}{1-\rho_{i2}} \right) \tag{13}$$

4.2 Expected Waiting Time

The expected waiting time of a task that arrives to the i-th tandem is given by:

$$E(w)_i = \frac{\lambda_i(1-\rho_{i1})}{(\overline{\mu_{i1}}-\lambda_i)^2} + \frac{\lambda_i(1-\rho_{i2})}{(\overline{\mu_{i2}}-\lambda_i)^2} \tag{14}$$

The expected waiting time of a task is given by

$$E(w)_i = \sum_{i=1}^{N} \frac{\lambda_i(1-\rho_{i1})}{(\overline{\mu_{i1}}-\lambda_i)^2} + \frac{\lambda_i(1-\rho_{i2})}{(\overline{\mu_{i2}}-\lambda_i)^2} \tag{15}$$

4.3 Tasks in the System

The probability of having n_1 tasks in the first stage and n_2 tasks in the second stage of the i-th tandem is given by:

$$[P(n_1, n_2)]_i = \rho_{i1}^{n_{i1}} \rho_{i2}^{n_{i2}} (1-\rho_{i1})(1-\rho_{i2}) \tag{16}$$

And the probability of having n_1 tasks in the first stage and n_2 tasks in the second stage for the whole system is given by:

$$P(n_1, n_2) = \sum_{i=1}^{N} (\rho_{i1}^{n_{i1}} \rho_{i2}^{n_{i2}} (1-\rho_{i1})(1-\rho_{i2})) \tag{17}$$

4.4 Tasks being Serviced in the System

The expected number of tasks being serviced in the i-th tandem is given by

$$[E_s(n)]_i = \rho_{i1} + \rho_{i2} \tag{18}$$

The expected number of tasks being serviced in the system is

$$[E_s(n)]_i = \sum_{i=1}^{N} [\rho_{i1} + \rho_{i2}] \tag{19}$$

4.5 Tasks Waiting in the System

The expected number of tasks waiting in the queues of the i-th tandem in the system is given by

$$[E_w(n)]_i = \frac{\rho_{i1}^2}{1-\rho_{i1}} + \frac{\rho_{i2}^2}{1-\rho_{i2}} \tag{20}$$

$$E_w(n) = \sum_{1=1}^{N} \left(\frac{\rho_{i1}^2}{1-\rho_{i1}} + \frac{\rho_{i2}^2}{1-\rho_{i2}} \right) \tag{21}$$

References

1. Liu, Jane W. S.: Performance Analysis of Multiprocessor Systems Containing Functionally Dedicated Processors, Acta Informatica 10: 95-104 (1978)
2. Hsiao, D. and Harary, F.: A Formal System for Information Retrieval from Files, Comm. ACM, Vol. 13, No. 2, Feb. 1970.
3. Feller, W.: An Introduction to Probability Theory and its Applications, John Wiley & Sons, 1970.
4. Kuri, A.: A Probabilistic Model for a Specialized Information Retrieval System, M.S. Thesis, University of Illinois, May 1976.
5. Kleinrock, L.: Queueing Systems, Vol. I, John Wiley & Sons, 1975.
6. Kleinrock, L.: Queueing Systems, Vol. I, pp. 148-150, John Wiley & Sons, 1975.
7. Saaty, T.: Elements of Queueing Theory, McGraw Hill, 1961.
8. Kleinrock, L., Nilsson, A.: On Optimal Scheduling Algorithms for Time Shared Systems, JACM 15(4): 549-576, 1981.
9. Coffman, E., Flatto, l., Poonen, B., Wright, P.: The Processor Minimization Problem with Independent Waiting Time Constraints, Theoretical Comp. Sci., 125:3-16, 1994.