SEARCH IN THE WOSNet

Thomas Böhme

Department of Mathematics, Technical University of Ilmenau, D-98684 Ilmenau, Germany, phone: +49 3677 69 3630, E-mail: tboehme@theoinf.tu-ilmenau.de

Herwig Unger

Department of Computer Science, University of Rostock, D-18051 Rostock, Germany, phone: +49 381 498 3403, fax: +49 381 498 3366, E-mail: tboehme@theoinf.tu-ilmenau.de

INTRODUCTION

In [2] the concept of a virtual operating system is discussed that supports the use of the resources of the web. As an essential part of such a system mechanism are required to find the necessary software and a suitable computer to run this software (see [1]). Due to the highly distributed and dynamic nature of the web a lot of communication is needed to find these resources. Thus efficient strategies for communication and searching are needed. The present paper is concerned with a strategy for the communication during the search process.

SEARCH STRATEGIES

Suppose a certain service is needed on some machine. If the machine does not provide this service, the WOS is invoked to locate a suitable machine. In a first step warehouses nearby are searched for a list of machines that probably can perform the service. Then requests are submitted to the machines from this list until a suitable machine is found. This process can be carried out in serveral different ways. The two principal ways are the broadcast strategy and the serial strategy [1].

1. The broadcast strategy
   The requesting machine submits the request to each machine in the list. Each of these machines then sends a messages back to the requesting machine. Since these machines can almost work in parallel the answer will be quickly available on the requesting machine. If the list contains $n$ machines, $2n$ messages will be generated. Thus the network load is high. Furthermore, the broadcast implementation must be realized as multiple unicasts, hence delaying data transmission.

2. The serial request strategy
   In this case the requesting machine sends one message containing the list of the remaining machines to one of the machines from the list. If the service is available on this machine, a positive answer is directly sent back to the requesting machine.
Otherwise, the request will automatically be directed to one of the machines in the list included in the message. This procedure will be repeated until a suitable machine is found or there are no machines left in the list. If the initial list contains \( b \) machines, the total number of messages needed is at most \( b + 1 \). So the generated network load is much less than in the first case. On the other hand, the respond time is much higher than in the first case and any communication problems or long transfer times directly influence the respond time.

In [1] it is shown that by combining these two strategies the relations can be improved. Here we outline a stochastic model of the search process that slightly improves the one considered in [1].

Let \( m_0 \) denote the requesting machine, and denote by \( L \) the list of machines to be checked. The search strategy can be represented by a directed rooted tree \( T \) with vertex set \( L \cup \{m_0\} \) and root \( m_0 \). A vertex \( y \) is called a successor of a vertex \( x \) if there is a directed path from \( x \) to \( y \). \( y \) is a directed successor of \( x \) if there is a directed edge from \( x \) to \( y \). The search is carried out in the following way. At first \( m_0 \) broadcasts the request to its direct successors. When a machine \( m' \) receives the request, it searches whether the service is available. If yes, it immediately sends a positive answer back to \( m_0 \). Otherwise, the request is directed to its direct successors if there are any. If \( m' \) has no successors (i.e. \( m' \) is an end vertex of \( T \)) and the service is not available, it sends a negative answer back to \( m_0 \). Thus the final result of the search process is obtained when either a positive answer from some machine is received by \( m_0 \) or \( m_0 \) has received negative answers from all end vertices.

In the talk a method is explained that allows to compute the expectation of the time that it takes until the final result of the search is obtained by \( m_0 \). For this purpose we make use of the assumption that the transmission time from one machine to another may be considered as a normal random variable. This assumption is supported by the following experiment:

During a period of 20 days every 10 minutes the following procedure was carried out: From a machine in Rostock (Germany) a series of 20 packages each of length 256 byte was successively sent to a machine in Quebec (Canada). On the receiving machine each package was immediately sent back to the sending machine. The total transmission time was recorded on the sending machine. A similar experiment was conducted using a machine in Rostock and one in Ilmenau (Germany). The statistical analysis of the obtained data showed that the transmission time mainly depends on the day of the week and the time at which the transmission is carried out, and that if a certain day and time (e.g. Thursday 4pm) is fixed the transmission time can be considered as a normal random variable.

REFERENCES
