数理解析研究所講究録
第 871 巻 1994 年 182-189

Tight Bound on the Competitive Ratio for the Page Replication Problem

Susanne Albers Max-Planck-Institut Saarbrücken, Germany
スザンヌ アルベルス マックスプランクインスティテュート

Hisashi Koga The University of Tokyo Tokyo, Japan
古賀 久志 東京大学理学部情報科学科

Abstract

In a distributed shared memory system, each read-only page needs to be located at appropriate processors by replication to make the total access cost lower. The purpose of the page replication problem is to implement this low-cost locating. In this paper, on-line algorithms for the page replication problem are considered in terms of competitiveness, the ratio of the cost of the on-line algorithms to that of the off-line optimal algorithms. As results, we present a randomized on-line algorithm for trees that is $\frac{e}{e-1} \simeq 1.58$-competitive. Furthermore, we prove that this algorithm achieves the best competitive ratio for trees. In other words, no randomized on-line algorithm is better than $\frac{e}{e-1} \simeq 1.58$-competitive.

1 Introduction

A common design for a shared memory multiprocessor system is a network of processors, each of which has its own local memory. In such a design, a programming abstraction of a simple global memory is supported by a virtual memory system that distributes the physical pages among the local memories.

In a distributed shared memory system, when processor q wishes to access memory address a of page b, first q examines whether the page is contained in its local memory. If so, the page access is done locally at 0 cost. If not, q searches the processor p having the page b and sends a access request to p. Then the processor p responses to the request and the value of the location a is transmitted back to q. The cost of this action is proportional to the distance between p and q. However, if q requires the page access to b frequently, the migration/replication of a full page of b to q may result in spending lower cost in total, because once q has the page copy, q accesses the page with no cost after that. On the other hand, moving a full page incurs a large amount of communication cost proportional to the distance.

For writable pages, it is reasonable to store only one copy in the entire system in order to avoid the difficulty of maintaining consistency among multiple copies. In such a situation, it is important to consider the *page migration problem* whose purpose is to devise residency strategies that decide which local memory should have the only copy of a writable page to reduce the cost in processing a sequence of page-access requests. On the other hand, for read-only pages, many copies may exist at the same time, because the consistency cannot be broken. Therefore, to find residency strategies that decide which subset of the local memories should contain the page copy is essential. This problem is called the *page replication problem.*

This paper focuses on on-line algorithms for the page replication problem. An algorithm is said to be on-line if it processes a request based only on that request and past requests. To evaluate on-line algorithms, we use competitiveness, the ratio of the cost of the on-line algorithms to that of the off-line optimal algorithms, that was introduced by Sleator and Tarjan [6].

The page replication problem is an important problem that has been studied in many papers recently. Black and Sleator [3] who have initiated the study of the page replication problem proved that no deterministic on-line algorithm can be better than 2-competitive for any networks even when the graph consists of only 2 nodes connected by a single edge. They actually devised a 2-competitive deterministic on-line algorithm when the network topology is a tree.

As for randomized cases, Koga [4] developed a 1.71-competitive randomized on-line algorithm for trees, thereby beating the deterministic lower bound. He also presented a 4-competitive randomized on-line algorithm for rings. For general graphs, Bartal et. al. [1] gave a $O(\log n)$-competitive randomized on-line algorithm, where n is the number of processors in the entire system. Furthermore they proved that no on-line algorithm is better than $O(\log n)$-competitive for general graphs. However, this bound is not very expressive for many realistic topologies like trees.

Therefore in this paper, we investigate the page replication problem for trees in detail. As a result we present a $\frac{e}{e-1} \simeq 1.58$-competitive randomized algorithm for trees. Our algorithm has the advantage that it uses only one random number only during an initialization phase. We also show that our algorithm is optimal. Specifically we prove that no randomized on-line algorithm is not better than $\frac{e}{e-1}$-competitive. We say in other words that the *randomized lower bound of the competitive ratio* is $\frac{e}{e-1}$. Uncommonly, to obtain the randomized lower bound, we do not make use of Yao's minimax principle [7] which was almost necessarily used in obtaining the randomized lower bound in previous researches on on-line problems.

The page replication problem is a fundamental on-line problem. For example, the simplest case corresponds to the ski rental problem [5]. The results and techniques developed in this paper would have more applications for on-line problems.

2 Competitive Analysis

An on-line algorithm is an algorithm which must satisfy a request without knowing the future requests. In this paper, we focus on the competitiveness of on-line algorithms introduced by Sleator and Tarjan [6].

The definition of competitiveness is as follows. The cost of an algorithm A on request sequence σ is denoted by $C_{\mathrm{A}}(\sigma)$. Generally, a deterministic on-line algorithm A is c-competitive if, for all request sequences σ, there is a constant b such that $C_{\mathrm{A}}(\sigma) \leq$

$c \cdot C_{\mathrm{OPT}}(\sigma) + b$. Here OPT is the off-line algorithm which achieves the optimal minimum cost on σ knowing the entire sequence in advance.

However, in the page replication problem, the trivial algorithm that initially copies the page to all nodes becomes 0-competitive by setting b to the sum of the costs of the initial page replications. To give meaningful results, we must redefine the competitiveness according to [3] as follows. A deterministic on-line algorithm A is c-competitive if, for all request sequences σ, $C_{\mathrm{A}}(\sigma) \leq c \cdot C_{\mathrm{OPT}}(\sigma)$.

A randomized on-line algorithm B is c-competitive against an oblivious adversary (see [2] for detail) if, for all request sequences σ, $\mathrm{E}[C_{\mathrm{B}}(\sigma)] \leq c \cdot C_{\mathrm{OPT}}(\sigma)$. The expectation is taken over the random choice made by the on-line algorithm.

3 Formal Definition of the Page Replication Problem

The component of this problem is an undirected graph G which represents the network. The vertices correspond to processors. The edges represent the links between two adjacent processors, and their length denotes the distance between them. Let δ_{ij} be the length of the shortest path between node i and node j.

In the page replication problem, we concentrate on a particular page b. A request from node q is a reference by processor q to some address of page b. Satisfying a request from q costs the distance from q to the nearest node p with the page copy (i.e. δ_{pq}) if q does not hold the copy of b yet. Else if q holds the copy already, the request is satisfied locally at 0 cost. When q does not have the page copy, p can replicate the page to q after satisfying the request at the cost of $r\delta_{pq}$ if necessary, where r is a constant bigger than 1 proportional to the page size. Note the fact that $r > 1$ implies that the replication of the page requires more cost than simply satisfying a request.

The replication problem is to decide (in an on-line fashion) which nodes should have the page to process the request sequence at low cost provided initially only one particular node, (which we call s in this paper) has the page.

In this paper three assumptions, which are originally used for the deterministic page replication problem in [3], are made to simplify the problem.

1. Once a node has the copy of the page, it never drops it.
2. A node can replicate the page copy only to its adjacent nodes.
3. Every local memory has infinite capacity.

From these assumptions the set of nodes with a page copy shall always be a connected component of the graph.

4 An Optimal On-line Algorithms for Trees

In the remainder of this paper, we deal with the page replication problem for trees. Let s be the only node which holds the page at the beginning. We first show that on-line page replication algorithms for trees can be analyzed by partitioning the total costs into parts incurred by each edge. That is, an edge incurs a cost equal to the length of the edge for a page access operation, if the path from the requesting node to the closest node with the page contains the edge. Otherwise the edge incurs no cost. The edge also incurs the replication cost if a replication is made across it.

More formally, let σ be a request sequence for the given tree. We denote the cost incurred by edge e when A serves σ as $C_{\mathrm{A}}(\sigma, e)$. In case A is a randomized one, $C_{\mathrm{A}}(\sigma, e)$ represents the expected cost incurred by e, where the expectation is taken over the random choice made by A. We generally evaluate the performance of an on-line algorithm A by comparing $C_{\mathrm{A}}(\sigma, e)$ to $C_{\mathrm{OPT}}(\sigma, e)$ for all edges e of the tree.

In order to analyze $C_{\mathrm{A}}(\sigma, e)$, we use some notation. Let $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(m)$ be a request sequence of length m and let $\sigma(t)$, $1 \leq t \leq m$, be the request at time t. Suppose $\sigma(t)$ is a request at node v. We set

$$a_\sigma(e, \sigma(t)) = 1$$

if e belongs to the path from v to s. Otherwise we set

$$a_\sigma(e, \sigma(t)) = 0.$$

If $a_\sigma(e, \sigma(t)) = 1$, we say that $\sigma(t)$ *causes an access at edge e*. Let

$$a_\sigma(e) = \sum_{t=1}^{m} a_\sigma(e, \sigma(t)),$$

i.e. $a_\sigma(e)$ is the number of requests that cause an access at edge e.

The following lemma is crucial in our analysis. Let $l(e)$ denote the length of the edge e hereafter.

Lemma 1 *Let A be an on-line replication algorithm for trees. If, for an arbitrary tree and any request sequence σ for that tree, A satisfies,*

$$C_A(\sigma, e) \leq c \cdot \min\{a_\sigma(e), r\} \cdot l(e)$$

for every edge e. Then the algorithm A is c-competitive. (Again, if A is a randomized algorithm, then $C_A(\sigma, e)$ is the expected cost incurred by e.)

Proof : We prove that for any edge e, $C_{\mathrm{A}}(\sigma, e) \leq c \cdot C_{\mathrm{OPT}}(\sigma, e)$. This implies the lemma, because $C_{\mathrm{A}}(\sigma) = \sum_e C_{\mathrm{A}}(\sigma, e) \leq \sum_e c \cdot C_{\mathrm{OPT}}(\sigma, e) = c \cdot C_{\mathrm{OPT}}(\sigma)$.

If $a_\sigma(e) < r$, then OPT does not replicate the page across e and hence e incurs a cost of $a_\sigma(e) l(e)$. Hence

$$C_{\mathrm{A}}(\sigma, e) \leq c \cdot \min\{a_\sigma, r\} l(e) = c \cdot a_\sigma(e) \cdot l(e) = c \cdot C_{\mathrm{OPT}}(\sigma, e).$$

On the other hand, if $a_\sigma(e) \geq r$, then OPT replicates the page across e before serving any request, and e incurs a cost of $rl(e)$. Thus

$$C_{\mathrm{A}}(\sigma, e) \leq c \cdot \min\{a_\sigma, r\} l(e) = c \cdot r \cdot l(e) = c \cdot C_{\mathrm{OPT}}(\sigma, e). \quad \blacksquare$$

From now on, we present our new randomized algorithm called GEOMETRIC and prove that the competitive factor of our algorithm is optimal for any value of r.

Algorithm GEOMETRIC for trees: : In this algorithm, each node v has counter c_v. All counters are initialized to 0. At the beginning, the algorithm chooses a random number from the set $\{1, 2, \ldots, r\}$. Specifically, the number i is chosen with probability $p_i = \alpha \cdot \rho^{i-1}$, where $\alpha = \frac{\rho-1}{\rho^r-1}$. When a node d which does not have the page requires the access to the page, the counter of every node along the path from d to the closest node with the page is incremented. When a counter reaches the randomly chose number, the page is replicated to the corresponding node.

Note the value of α is chosen so that

$$\sum_{i=1}^{r} p_i = \alpha \sum_{i=1}^{r} \rho^{i-1} = \alpha \cdot \frac{\rho^r - 1}{\rho - 1} = \alpha \cdot \frac{1}{\alpha} = 1.$$

The description of GEOMETRIC implies that the longer the distance between some node and s is, the smaller the counter value for the node becomes. Thus, the nodes holding the page form a connected component of the underlying tree. GEOMETRIC has the advantage that it only uses one random number only during the initialization phase.

Theorem 1 *The algorithm GEOMETRIC is $\frac{\rho^r}{\rho^r-1}$-competitive for an arbitrary tree.*

Note that $\frac{\rho^r}{\rho^r-1}$ goes to $\frac{e}{e-1} \approx 1.58$ as r tends to infinity. The proof of this theorem follows directly from Lemma 1 and Lemma 2 below.

Lemma 2 *Let $E[C_{GE}(\sigma, e)]$ denote the expected cost incurred by edge e when GEOMETRIC for trees serves a request sequence σ. For any tree T and any request sequence σ for T,*

$$E[C_{GE}(\sigma, e)] \leq \frac{\rho^r}{\rho^r - 1} \cdot \min\{a_\sigma(e), r\} \cdot l(e).$$

Proof : Consider an arbitrary tree T and a request sequence σ for the tree. We concentrate on a single edge e of the tree. Let v be the node on the end of the edge e. Furthermore, let $k = a_\sigma(e)$ and $\sigma(t_1), \sigma(t_2), \ldots, \sigma(t_k)$ be the requests that cause an access at the edge e. Note, whenever a request causes an access at the edge e, the counter c_v of the end node v is incremented by 1, until the page is replicated to v. There are two cases to consider with respect to the value of k.

1. When $1 \leq k \leq r$.

Let i be the random number chosen by GEOMETRIC at the beginning. If $i < k$ then the edge e incurs a cost of $(i + r) \cdot l(e)$. Otherwise e incurs a cost of $k \cdot l(e)$. Therefore,

$$\begin{aligned}
\mathrm{E}[C_{\mathrm{GE}}(\sigma, e)] &= l(e) \cdot (\sum_{i=1}^{k} (i + r) p_i + k(1 - \sum_{i=1}^{k} p_i)) \\
&= l(e) \cdot (k + (r - k) \sum_{i=1}^{k} p_i + \sum_{i=1}^{k} i p_i) \\
&= l(e) \cdot (k + (r - k) \sum_{i=1}^{k} \alpha \rho^{i-1} + \sum_{i=1}^{k} i \alpha \rho^{i-1}) \\
&= l(e) \cdot (k + (r - k) \frac{\rho^k - 1}{\rho - 1} \alpha - \frac{\rho^k - 1}{(\rho - 1)^2} \alpha + \frac{k \rho^k}{\rho - 1} \alpha).
\end{aligned}$$

We have $\rho - 1 = \frac{1}{r}$, thus

$$\begin{aligned} \mathrm{E}[C_{\mathrm{GE}}(\sigma, e)] &= l(e) \cdot (k + \alpha((r^2 - kr - r^2)(\rho^k - 1) + kr\rho^k)) \\ &= l(e) \cdot (k + kr\alpha) \\ &= l(e) \cdot k \cdot \frac{\rho^r}{\rho^r - 1} = \frac{\rho^r}{\rho^r - 1} \cdot \min\{a_\sigma(e), r\} \cdot l(e) \end{aligned}$$

Thus, the theorem holds for this case.
2. When $k > r$.
Since $\sum_{i=1}^{r} p_i = 1$, the page has been replicated across e before $\sigma(t_{r+1})$ is generated. Therefore the cost incurred by GEOMETRIC becomes the same as if we had $k = r$. Moreover, $\min\{a_\sigma(e), r\} = r$. Hence we can reduce this case to the case when $1 \leq k \leq r$. Thus, we complete the proof. ∎

Next we show that GEOMETRIC's competitive factor is optimal by proving the following theorem.

Theorem 2 *Any randomized on-line replication algorithm.A cannot be better than $\frac{\rho^r}{\rho^r-1}$-competitive for trees.*

Proof : To obtain the lower bound of the competitive ratio, it suffices to consider the case when the graph consists of 2 nodes, because in any tree, there are request sequences such that all requests are generated at only one node adjacent to s. Let s and u be 2 nodes connected by a single edge. Without loss of generality, we can assume that $\delta_{su} = 1$. We will construct a request sequence σ consisting of requests at node u such that the expected cost of A is at least $\frac{\rho^r}{\rho^r-1}$ times the optimal off-line cost.

For $i = 1, 2, \ldots$, let q_i be the probability that A replicates the pages from s to u after exactly i requests, given a request sequence that consists only of requests at node u. In the following we compare the algorithm A to our algorithm GEOMETRIC. We consider two cases.
Case 1: There exists an l, where $1 \leq l \leq r$, such that $\sum_{i=1}^{l} q_i \geq \sum_{i=1}^{l} p_i$.
Let k be the smallest index satisfying the above inequality, i.e. $\sum_{i=1}^{k} q_i \geq \sum_{i=1}^{k} p_i$ and $\sum_{i=1}^{j} q_i < \sum_{i=1}^{j} p_i$ for all j with $1 \leq j < k$. Let σ be the request sequence that consists of k requests at node u. We show that

$$\mathrm{E}[C_{\mathrm{A}}(\sigma)] \geq \mathrm{E}[C_{\mathrm{GE}}(\sigma)] = \frac{\rho^r}{\rho^r - 1} \cdot k. \tag{1}$$

This implies that A cannot be better than $(\frac{\rho^r}{\rho^r-1})$-competitive because the optimal off-line cost on σ is k.

We have

$$\mathrm{E}[C_{\mathrm{A}}(\sigma)] = \sum_{i=1}^{k}(i + r)q_i + k(1 - \sum_{i=1}^{k} q_i)$$

$$\mathrm{E}[C_{\mathrm{GE}}(\sigma)] = \sum_{i=1}^{k}(i + r)p_i + k(1 - \sum_{i=1}^{k} p_i)$$

Subtracting $\mathrm{E}[C_{\mathrm{GE}}(\sigma)]$ from $\mathrm{E}[C_{\mathrm{A}}(\sigma)]$, we obtain

$$\begin{aligned}\mathrm{E}[C_{\mathrm{A}}(\sigma)] - \mathrm{E}[C_{\mathrm{GE}}(\sigma)] &= \sum_{i=1}^{k}(i+r)(q_i - p_i) + k\sum_{i=1}^{k}(p_i - q_i) \\ &= (r+1-k)\sum_{i=1}^{k}(q_i - p_i) + \sum_{i=2}^{k}(i-1)(q_i - p_i)\end{aligned}$$

Since $r \geq k$, we obtain

$$\mathrm{E}[C_{\mathrm{A}}(\sigma)] - \mathrm{E}[C_{\mathrm{GE}}(\sigma)] \geq \sum_{i=2}^{k}(i-1)(q_i - p_i) = \sum_{i=2}^{k}(\sum_{j=i}^{k} q_j - \sum_{j=i}^{k} p_j) \tag{2}$$

Since $\sum_{j=1}^{k} q_j \geq \sum_{j=1}^{k} p_j$, we have $\sum_{j=1}^{i-1} q_j + \sum_{j=i}^{k} q_j \geq \sum_{j=1}^{i-1} p_j + \sum_{j=i}^{k} p_j$ for $i = 2, 3, \ldots, k$. Hence for $i = 2, 3, \ldots, k$,

$$\sum_{j=i}^{k} q_j - \sum_{j=i}^{k} p_j \geq \sum_{j=1}^{i-1} p_j - \sum_{j=1}^{i-1} q_j. \tag{3}$$

Applying (3) to (2), we conclude

$$\mathrm{E}[C_{\mathrm{A}}(\sigma)] - \mathrm{E}[C_{\mathrm{GE}}(\sigma)] \geq \sum_{i=2}^{k}(\sum_{j=i}^{k} q_i - \sum_{j=i}^{k} p_i) \geq \sum_{i=2}^{k}(\sum_{j=1}^{i-1} p_i - \sum_{j=1}^{i-1} q_i) > 0,$$

since $\sum_{j=1}^{m} q_i < \sum_{j=1}^{m} p_i$ for all m with $1 \leq m \leq k-1$. Thus, inequality (1) is proved.
Case 2: For all $k = 1, 2, \ldots, r$, the inequality $\sum_{i=1}^{k} q_i < \sum_{i=1}^{k} p_i$ is satisfied.
Let σ be the request sequence that consists of $2r$ requests at node t. Let A' be the on-line algorithm with $q'_i = q_i$, for $i = 1, 2, \ldots, r-1$, and $q'_r = \sum_{i \geq r} q_i$. Then

$$\mathrm{E}[C_{\mathrm{A}}(\sigma)] = \sum_{i=1}^{2r}(r+i)q_i + \sum_{i>2r} 2rq_i \geq \sum_{i=1}^{r-1}(r+i)q_i + 2rq'_r = \sum_{i=1}^{r-1}(r+i)q'_i + 2rq'_r = \mathrm{E}[C_{\mathrm{A'}}(\sigma)].$$

Since $\sum_{i=1}^{r} q'_i = \sum_{i=1}^{r} p_i = 1$ and $\sum_{i=1}^{j} q'_i < \sum_{i=1}^{j} p_i$ for all j with $1 \leq j < r$, Case 1 immediately implies

$$\mathrm{E}[C_{\mathrm{A}}(\sigma)] \geq \mathrm{E}[C_{\mathrm{A'}}(\sigma)] \geq \mathrm{E}[C_{\mathrm{GE}}(\sigma)] = \frac{\rho^r}{\rho^r - 1} r,$$

and A cannot be better than $(\frac{\rho^r}{\rho^r-1})$-competitive because the optimal off-line cost equals r. This concludes the proof of the theorem. ■

Note that the proof of Theorem 2 does not use Yao's minimax principle [7] which appeared so often in order to obtain the randomized lower bound in previous researches on on-line problems.

5 Summary

This paper has presented a new on-line algorithm for the page replication problem that arises in the management of a distributed shared memory. Especially we examined this problem when the network topology is a tree. Our algorithm improved the previously best competitive factor for trees and had the advantage of being simple. There are several interesting open problems for our work. One natural open problem is to find an optimal randomized on-line algorithm for rings. Another interesting open problem is to solve many page replication problems, each for an individual page, simultaneously in a single distributed shared memory. In this version, we need to consider the capacity of each local memory and deal with conflicts of pages. The model of this version itself has been proposed as the constrained file allocation problem in [1]. However, competitive on-line algorithms are found only for uniform networks, the simplest case.

References

[1] Y. Bartal, A. Fiat and Y. Rabani. "Competitive algorithms for distributed data management". In *Proceedings of the 24th ACM Symposium on Theory of Computing*, pages 39–50, 1992.

[2] S. Ben-David, A. Borodin, R.M. Karp, G. Tardos and A. Wigderson. "On the power of randomization in on-line algorithms". In *Proceedings of the 22th ACM Symposium on Theory of Computing*, pages 379–386, 1990.

[3] D.L. Black and D.D. Sleator. "Competitive algorithms for replication and migration problem". Technical Report CMU-CS-89-201, Department of Computer Science, Carnegie Mellon University, 1989.

[4] H. Koga. "Randomized on-line algorithms for page replication problem". In *Proceedings of 4th International Symposium on Algorithm and Computation*, pages 436–445, 1993.

[5] P. Raghavan. "Lecture notes on randomized algorithms". Research Report RC 15340, IBM T.J. Watson Research Center, 1989.

[6] R.E. Tarjan and D.D. Sleator. "Amortized efficiency of list update and paging rules". *Communications of the ACM*, 28:202–208, 1985.

[7] A.C.C. Yao. "Probabilistic computations: Toward a unified measure of complexity". In *Proceedings of Symposium on 17th Foundation of Computer Science*, pages 222–227, 1977.