IEICE TRANS. INF. & SYST., VOL.E92–D, NO.12 DECEMBER 2009

PAPER

Randomized Online File Allocation on Uniform Cactus Graphs*

Yasuyuki KAWAMURA†, *Nonmember and* **Akira MATSUBAYASHI**†a), *Member*

SUMMARY We study the online file allocation problem on ring networks. In this paper, we present a 7-competitive randomized algorithm against an adaptive online adversary on uniform cactus graphs. The algorithm is deterministic if the file size is 1. Moreover, we obtain lower bounds of 4.25 and 3.833 for a deterministic algorithm and a randomized algorithm against an adaptive online adversary, respectively, on ring networks.
key words: *online algorithm, file allocation, data management, cactus graph*

1. Introduction

Parallel and distributed systems, such as multiprocessor computer systems and the Internet, consist of nodes each having their own local memory module and communication links between the nodes. Data objects, such as files on distributed file servers and pages in a virtual shared memory system, are distributed among the nodes, and a node requiring access to a data object issues a request for the data. Because such requests are served using communication on the underlying network, it is important to allocate the data objects so that the communication load for the requests is minimized. In particular, dynamic reallocation of the data objects is effective in reducing the communication load in a situation where the requests are issued sequentially. This problem has been formulated as various types of online data management problems and studied extensively so far (e.g., [1]–[3]). In this study, we consider one of the variations, called the *file allocation problem* [4], in which read and write requests are served using unicast and multicast communication, respectively, and we are allowed to replicate copies of the data objects on the network. Serving a request costs the total distance of communication, and reallocating the data objects costs the total distance of replication multiplied by the data size. The objective of the file allocation problem is to minimize the total costs of services and reallocations.

Bartal, Fiat, and Rabani [4] presented a randomized $O(\log n)$-competitive algorithm against an adaptive online adversary on n-node general networks. Awerbuch, Bartal, and Fiat [5] improved the result by presenting a deterministic $O(\min\{\log n, \log(Diam)\})$-competitive algorithm, where $Diam$ is the diameter of a network. The algorithms are optimal in terms of order, i.e., there exists an n-node network on which any randomized algorithm against an oblivious adversary is $\Omega(\log n)$-competitive [4]. Better algorithms have been proposed for restricted networks. A randomized 3-competitive algorithm against an adaptive online adversary on trees and a deterministic 3-competitive algorithm on uniform complete networks were provided in [4]. Lund, Reingold, Westbrook, and Yan [6] improved the algorithm on trees by presenting a deterministic 3-competitive algorithm and a randomized $(2 + \frac{1}{D})$-competitive algorithm against an oblivious adversary, where D is a positive integer representing the data size. The algorithms are optimal because even on a single link, no randomized algorithm has a competitive ratio less than 3 [4], [7] and $2 + \frac{1}{D}$ [6] for adaptive and oblivious adversaries, respectively. It is mentioned that the $O(\log n)$-competitive algorithm on general networks is 7.464-competitive on ring networks [4]. This is because the algorithm is actually a $(2 + \sqrt{3})c$-competitive algorithm against an adaptive online adversary that uses a c-competitive online Steiner tree algorithm, and because a greedy Steiner tree algorithm is 2-competitive on ring networks.

In this paper, we present a 7-competitive randomized algorithm against an adaptive online adversary on uniform cactus graphs. The algorithm is deterministic if $D = 1$. Moreover, we obtain lower bounds of 4.254 and 3.833 for a deterministic algorithm and a randomized algorithm against an adaptive online adversary, respectively, on ring networks.

2. Preliminaries

A network can be represented by a graph G with edge weights. Let $V(G)$ and $E(G)$ denote the node set and edge set, respectively, of G. G is said to be *uniform* if every edge has a weight of 1. A *ring* is a graph consisting of a single cycle. A *cactus graph* is a connected graph in which any two cycles have at most one node in common. An example of a cactus graph is shown in Fig. 1.

The distance between two nodes u and v, denoted by $dist(u, v)$, is the minimum sum of weights of edges of a path connecting u and v. For $U \subseteq V(G)$, let $dist(u, U) = \min_{v \in U} dist(u, v)$. Let $T(U)$ denote a minimum Steiner tree containing U. Let $w(U)$ be the sum of weights of edges in $T(U)$. For $S, U \subseteq V(G)$, let $w(S, U)$ be the minimum sum

Manuscript received March 6, 2009.
Manuscript revised August 10, 2009.
†The authors are with the Division of Electrical Engineering and Computer Science, Kanazawa University, Kanazawa-shi, 920–1192 Japan.
*A preliminary version of this paper appeared in Proc. 7th International Symposium on Parallel and Distributed Computing, pp.449–453, 2008.
a) E-mail: mbayashi@t.kanazawa-u.ac.jp
DOI: 10.1587/transinf.E92.D.2416

of weights of edges of a forest that connects every node of $U \setminus S$ to a node of S.

The *file allocation problem* is as follows: given a graph G with edge weights, a positive integer D, $S_0 \subseteq V(G)$, and a sequence $R_1, \dots, R_k \in V(G) \times \{\text{read}, \text{write}\}$, to compute $S_1, \dots, S_k \subseteq V(G)$ so that $\sum_{\substack{R_i=(u,\text{read})\\ 1\le i\le k}} dist(u, S_{i-1}) + \sum_{\substack{R_i=(u,\text{write})\\ 1\le i\le k}} w(\{u\} \cup S_{i-1}) + D \sum_{i=1}^{k} w(S_{i-1}, S_i)$ is minimized. The file allocation problem is a formulation of the following scenario: Initially, each node of $S_0 \subseteq V(G)$ holds a copy of data, which is also called a file. A file allocation algorithm receives a sequence of requests generated at nodes in $V(G)$. Each request is either a read request or a write request. After each request is served, the algorithm can reallocate the copies by replicating and/or deleting copies. The algorithm serves a read request at node u using unicast communication between u and the closest node p in the set S of the nodes holding a copy at that time. The cost to serve this read request is $dist(u, p)$. If a write request is generated at u, then all the copies of the file on the nodes in S must be updated. The algorithm serves this write request by multicast communication from u to all the nodes in S, and it pays a cost equal to $w(\{u\} \cup S)$. The algorithm can delete a copy unless it is the last copy in the network, at no cost. The copies on the nodes in S can also be replicated to another set $S' \subseteq V(G)$. The cost of this replication is $D \cdot w(S, S')$, where D denotes the size of the file.

The following is a basic notion of online algorithms. See, e.g., [8] for further details. An algorithm to compute S_i after having known the entire sequence of requests is called an *offline algorithm*. By contrast, an *online algorithm* computes S_i using only information of $R_1, \dots, R_i$. An algorithm that provides an input to an online file allocation algorithm and also computes its own output is called an *adversary*. The adversary has the knowledge of the online algorithm and constructs the worst possible input. There are three types of adversaries. The *oblivious adversary* must construct the request sequence in advance and serves it optimally. By contrast, the *adaptive adversary* constructs the request one by one from the information of the current output of the online algorithm. There are two types of adaptive adversaries. The *adaptive online adversary* serves the current request online, and then chooses the next request based on the online algorithm's action thus far. The *adaptive offline adversary* chooses the next request based on the online algorithm's action thus far, but pays the optimal cost to the resulting request sequence. Let $\textsc{Alg}(\sigma)$ be the cost of a file allocation algorithm ALG for an input σ. For any adversary type ADV and any σ, if $\mathbf{E}[\textsc{Alg}(\sigma)] + \alpha \le \mathbf{E}[c \cdot \textsc{Adv}(\sigma)]$, then ALG is *c-competitive against ADV*, where α is a value independent of the number of requests. The competitiveness is typically analyzed by using a *potential function*. Let $e_1, e_2, \dots, e_m$ be any event sequence, i.e., a sequence of fragments of operations of ALG and ADV for σ. Suppose $\Phi : S_{\text{ALG}} \times S_{\text{ADV}} \to \mathbf{R}$, where S_{ALG} and S_{ADV} are the sets of nodes on which ALG and ADV hold copies, respectively. Let Φ_i be the value of Φ just after the ith event and Φ_0 be the value of Φ before e_1. To prove that ALG is c-competitive against ADV, it is sufficient to find Φ that satisfies $\Phi_i - \Phi_{i-1} \le c \cdot \mathbf{E}[(\text{cost of ADV for } e_i)] - \mathbf{E}[(\text{cost of ALG for } e_i)]$ for $1 \le i \le m$. This is because by summing up this inequality, we can obtain $\mathbf{E}[\textsc{Alg}(\sigma)] + \Phi_m - \Phi_0 \le \mathbf{E}[c \cdot \textsc{Adv}(\sigma)]$, and because $\Phi_m - \Phi_0$ is independent of the number of requests.

3. Randomized Algorithm on Uniform Cactus Graphs

In this section we present a randomized file allocation algorithm on uniform cactus graphs, called RUCG.

Let G be a uniform cactus graph. Let $\mathcal{C} = \{C_0, \dots, C_n\}$ be the set of cycles and 2-node paths consisting of edges not contained in a cycle of G. By the definition of a cactus graph, any elements C_i and C_j ($i \ne j$) of $\mathcal{C}$ share at most one node, and there is a unique sequence of elements of $\mathcal{C}$ such that any path from a node of C_i to a node of C_j contains an edge of each element in the order of the sequence. Suppose that $C \in \mathcal{C}$ and that $S \subseteq V(C)$ is the set of nodes of a path on C. For $u \in V(C) \setminus S$, let s be a closest node in S to u. Since $S = V(T(S))$, s is an end-node of $T(S)$. Let $\overline{s}$ be the other end-node of $T(S)$ if $|S| \ge 2$ and s otherwise. Let $P(S, u)$ be a shortest path connecting s and u, and $\overline{P}(S, u)$ be the path of length $dist(s, u)$ that starts from $\overline{s}$ and passes along nodes in $V(C) \setminus ((S \cup V(P(S, u))) \setminus \{\overline{s}, u\})$ (Fig. 2).

3.1 Definition

Initially, RUCG replicates a copy to each node of $V(T(S_0))$ before R_1 is generated. By this operation, we denote $V(T(S_0))$ by S_0 for simplicity. RUCG keeps the property that $S_i = V(T(S_i))$ for each $i > 0$. Suppose that R_i is generated at u_i ($1 \le i \le k$) and that s_i is a closest node in S_{i-1} to u_i. For convenience, we denote the unique sequence of elements of $\mathcal{C}$ along a path from s_i to u_i by $C_0, \dots, C_t$ so that $s_i \in C_0$ and $u_i \in C_t$. For $1 \le j \le t$, we denote the unique node shared by C_{j-1} and C_j by v_j. Let $v_0 = s_i$ and $v_{t+1} = u_i$. After serving R_i, RUCG reallocates the copies as follows:

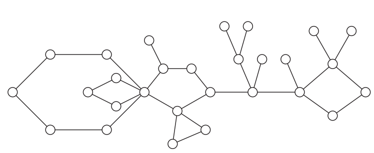

Fig. 1 An example of a cactus graph.

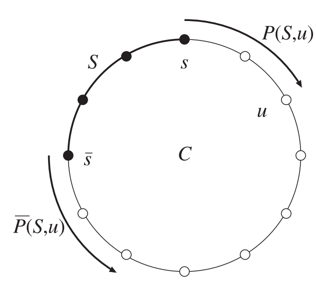

Fig. 2 $P(S, u)$ and $\overline{P}(S, u)$.

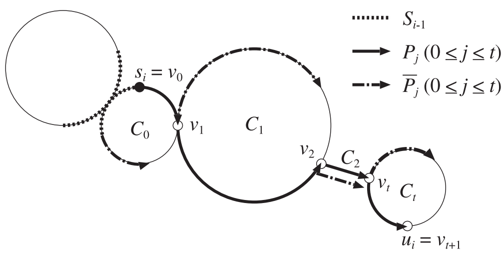

Fig. 3 Replication of RUCG for (u_i, read) with $u_i \notin S_{i-1}$.

1. If $R_i = (u_i, \text{read})$, then:
   a. If $u_i \in S_{i-1}$, then $S_i = S_{i-1}$.
   b. Otherwise,
      i. $P_0 = P(S_{i-1} \cap V(C_0), v_1)$ and $\overline{P}_0 = \overline{P}(S_{i-1} \cap V(C_0), v_1)$,
      ii. $P_j = P(\{v_j\}, v_{j+1})$ and $\overline{P}_j = \overline{P}(\{v_j\}, v_{j+1})$ for $1 \le j \le t$,
      iii. $S_i = S_{i-1} \cup \bigcup_{j=0}^{t}(V(P_j) \cup V(\overline{P}_j))$ with probability $\frac{1}{D}$ (Fig. 3) and $S_i = S_{i-1}$ with probability $1 - \frac{1}{D}$.
2. If $R_i = (u_i, \text{write})$, then set $S_i = \{u_i\}$ with probability $\frac{1}{D}$ and $S_i = S_{i-1}$ with probability $1 - \frac{1}{D}$.

It should be noted that $S_i = V(T(S_i))$ inductively. Moreover, for every C_j $(0 \le j \le t)$, the average cost for 1b is at most $2dist(v_j, v_{j+1}) - 1$ if $\overline{P}_j$ contains v_{j+1} and $2dist(v_j, v_{j+1})$ otherwise. This is because if both P_j and $\overline{P}_j$ contain v_{j+1}, then it is sufficient for RUCG to replicate copies along $V(P_j)$ and $V(\overline{P}_j) \setminus \{v_{j+1}\}$.

3.2 Competitiveness

We present the following theorem:

Theorem 1: RUCG is 7-competitive on uniform cactus graphs against an adaptive online adversary.

Proof We prove the theorem using a potential function. Let ADON be an adaptive online adversary. The potential function for RUCG just before R_i is generated is defined as $\Phi = D\cdot(5w(A_{i-1} \cup S_{i-1}) - 4w(S_{i-1}))$, where A_{i-1} is the set of nodes on which ADON holds the copies just before R_i is generated. For each request, we show $\mathbf{E}[\Delta\Phi] \le 7\Delta\text{Adon} - \mathbf{E}[\Delta\text{Rucg}]$ in each event of (i) ADON's reallocation, (ii) RUCG's service and reallocation, and ADON's service for a read request, and (iii) RUCG's service and reallocation, and ADON's service for a write request, where $\Delta\Phi$ is the increased amount of Φ due to the event, and ΔAdon and ΔRucg are costs paid by ADON and RUCG, respectively, in the event.

Lemma 1: In an event of ADON's replication and deletion, $\Delta\Phi \le 5\Delta\text{Adon} - \mathbf{E}[\Delta\text{Rucg}]$.

Proof Clearly, if ADON replicates or deletes copies, then $w(S_{i-1})$ does not change, and hence, the term of Φ that changes is only $5D \cdot w(A_{i-1} \cup S_{i-1})$. A deletion made by ADON only decreases the term. If ADON replicates a copy resulting with A_i,

$$\begin{aligned}\Delta\Phi &= D \cdot \{5(w(A_i \cup S_{i-1}) - w(A_{i-1} \cup S_{i-1}))\}\\ &\le 5D \cdot w(A_{i-1}, A_i)\\ &= 5\Delta\text{Adon}.\end{aligned}$$

The inequality holds because $w(A_{i-1} \cup S_{i-1}) + w(A_{i-1}, A_i)$ is the sum of weights of edges of a connected subgraph containing A_{i-1}, S_{i-1}, and A_i, and because this sum of weights is at least $w(A_i \cup S_{i-1})$. Because $\Delta\text{Rucg} = 0$ in the event, the lemma holds. □

We present claims to prove the subsequent lemma. Let a_i be a closest node in A_{i-1} to u_i and b_i be the closest node in $\bigcup_{j=0}^{t} V(C_j)$ to a_i. Suppose that $a_i \in C_x$ and $b_i \in C_q$ $(0 \le q \le t)$.

Claim 1: $dist(a_i, u_i) \ge dist(b_i, u_i)$.

Proof By the definition of a cactus graph, there exists a unique sequence of elements of C between C_x and C_q such that any path connecting a_i to a node of $\bigcup_{j=0}^{t} V(C_j)$ passes along the sequence and that two consecutive elements in the sequence share exactly one node. If $a_i \in \bigcup_{j=0}^{t} V(C_j)$, then $b_i = a_i$. Otherwise, b_i is the node shared by C_q and another element in the sequence. Therefore, any shortest path connecting a_i and u_i contains b_i, which proves the claim. □

Claim 2: If $R_i = (u_i, \text{read})$ with $u_i \notin S_{i-1}$, then for $0 \le j < q$, at least one of P_j or $\overline{P}_j$ is a subgraph of $T(A_{i-1} \cup S_{i-1})$.

Proof Because $T(A_{i-1} \cup S_{i-1})$ contains a node of C_0, it follows from a similar argument of the proof of Claim 1 that $T(A_{i-1} \cup S_{i-1})$ contains b_i. Therefore, $T(A_{i-1} \cup S_{i-1})$ passes through $C_0, \ldots, C_q$, and hence, contains v_j and v_{j+1} for $0 \le j < q$. Because any path connecting v_j and v_{j+1} in C_j contains P_j or $\overline{P}_j$ as a subgraph by definition, at least one of P_j or $\overline{P}_j$ is a subgraph of $T(A_{i-1} \cup S_{i-1})$. □

Claim 3: If $R_i = (u_i, \text{read})$ with $u_i \notin S_{i-1}$ and $dist(s_i, u_i) > dist(b_i, u_i)$, then at least one of P_q and $\overline{P}_q$ contains b_i or is a subgraph of $T(A_{i-1} \cup S_{i-1})$.

Proof Suppose that neither P_q nor $\overline{P}_q$ contains b_i. It should be noted that the nodes of C_q contained in neither P_q nor $\overline{P}_q$ induces at most two paths.

If $q > 0$, then P_q and $\overline{P}_q$ share v_q by definition, and hence, there exists exactly one such induced path. If $q = 0$, then one of the induced paths is $T(S_{i-1} \cap V(C_0))$. It follows that $b_i \notin S_{i-1} \cap V(C_0)$, for otherwise, $dist(s_i, u_i) \le dist(b_i, u_i)$, contradicting the assumption of the claim.

Therefore, because neither P_q nor $\overline{P}_q$ contains b_i, each path connecting v_q and b_i contains P_q or $\overline{P}_q$ as a subgraph. Because $T(A_{i-1} \cup S_{i-1})$ contains v_q and b_i as shown in the proof of Claim 2, the claim holds. □

Lemma 2: In the event of (ii) for $R_i = (u_i, \text{read})$, $\mathbf{E}[\Delta\Phi] \le 5\Delta\text{Adon} - \mathbf{E}[\Delta\text{Rucg}]$.

Proof If $u_i \in S_{i-1}$, then $\Delta\Phi = 0$ and $\Delta\text{RUCG} = 0$, which proves the lemma. Therefore, we assume $u_i \notin S_{i-1}$. ADON's cost to serve R_i is at least $dist(a_i, u_i) \geq dist(b_i, u_i)$ by Claim 1. RUCG's cost to serve R_i is $dist(s_i, u_i)$. If RUCG replicates, then the replication cost is $D \cdot (2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j)$, where λ_j is 1 if $\overline{P}_j$ contains v_{j+1} and 0 otherwise. Thus,

$$\begin{aligned}\mathbf{E}[\Delta\text{RUCG}] &= dist(s_i, u_i) + \frac{1}{D} \cdot D \cdot \left(2dist(s_i, u_i)\right.\\ &\qquad \left. - \sum_{j=0}^{t} \lambda_j\right) + \left(1 - \frac{1}{D}\right) \cdot 0 \\ &= 3dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j,\end{aligned}$$

Φ changes with probability $\frac{1}{D}$, only when RUCG reallocates the copies. Moreover, if RUCG reallocates the copies on S_{i-1} to S_i, then $w(S_i) = w(S_{i-1}) + 2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j$. Therefore, it follows that

$$\begin{aligned}\mathbf{E}[\Delta\Phi] &= \frac{1}{D} \cdot D \cdot \{5w(A_{i-1} \cup S_i) - 4w(S_i) \\ &\quad - (5w(A_{i-1} \cup S_{i-1}) - 4w(S_{i-1}))\} \\ &= 5(w(A_{i-1} \cup S_i) - w(A_{i-1} \cup S_{i-1})) \\ &\quad - 4 \cdot \left(2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j\right).\end{aligned}$$

We first consider the case that $dist(s_i, u_i) \leq dist(b_i, u_i)$. RUCG's replication increases $w(A_{i-1} \cup S_{i-1})$ by at most $2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j$. Therefore,

$$\begin{aligned}\mathbf{E}[\Delta\Phi] &\leq 5\left(2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j\right) \\ &\quad -4 \cdot \left(2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j\right) \\ &\leq 5\left(dist(b_i, u_i) + dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j\right) \\ &\quad -4 \cdot \left(2dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j\right) \\ &= 5dist(b_i, u_i) - 3dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j \\ &\leq 5\Delta\text{ADON} - \mathbf{E}[\Delta\text{RUCG}].\end{aligned}$$

We then consider the other case that $dist(s_i, u_i) > dist(b_i, u_i)$. If RUCG reallocates the copies to S_i, then it replicates the copies along both P_j and $\overline{P}_j$ for $0 \leq j \leq t$. By Claim 2, for $0 \leq j < q$, $T(A_{j-1} \cup S_{j-1})$ contains at least one path Q_j of P_j and $\overline{P}_j$. Therefore, the replication along Q_j never increases $w(A_{i-1} \cup S_{i-1})$. By Claim 3, for $j = q$, at least one path Q_q of P_q and $\overline{P}_q$ contains b_i or is a subgraph of $T(A_{i-1} \cup S_{i-1})$. If Q_q contains b_i, then the replication along Q_q increases $w(A_{i-1} \cup S_{i-1})$ by at most $dist(b_i, v_{q+1})$. If Q_q is a subgraph of $T(A_{i-1} \cup S_{i-1})$, then the replication along Q_q never increases $w(A_{i-1} \cup S_{i-1})$. For $j > q$, because P_j is a shortest path connecting v_j and v_{j+1}, any shortest path connecting b_i and u_i contains a path in C_j of the same length as that of P_j. Thus,

$$\begin{aligned}&w(A_{i-1} \cup S_i) - w(A_{i-1} \cup S_{i-1}) \\ &\leq \sum_{j=0}^{q-1} dist(v_j, v_{j+1}) + dist(v_q, v_{q+1}) + dist(b_i, v_{q+1}) \\ &\qquad + 2\sum_{j=q+1}^{t} dist(v_j, v_{j+1}) - \sum_{j=0}^{t} \lambda_j \\ &= \sum_{j=0}^{t} dist(v_j, v_{j+1}) + dist(b_i, v_{q+1}) \\ &\qquad + \sum_{j=q+1}^{t} dist(v_j, v_{j+1}) - \sum_{j=0}^{t} \lambda_j \\ &= dist(v_0, v_{t+1}) + dist(b_i, v_{t+1}) - \sum_{j=0}^{t} \lambda_j \\ &= dist(s_i, u_i) + dist(b_i, u_i) - \sum_{j=0}^{t} \lambda_j.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbf{E}[\Delta\Phi] &\leq 5dist(b_i, u_i) - 3dist(s_i, u_i) - \sum_{j=0}^{t} \lambda_j \\ &\leq 5\Delta\text{ADON} - \mathbf{E}[\Delta\text{RUCG}].\end{aligned}$$

□

Lemma 3: In the event of (iii) for $R_i = (u_i, \text{write})$, $\Delta\Phi \leq 7\Delta\text{ADON} - \mathbf{E}[\Delta\text{RUCG}]$.

Proof Because ADON pays cost only for serving a write request in this event, $\Delta\text{ADON} = w(A_{i-1} \cup \{u_i\}) \geq dist(a_i, u_i)$. RUCG pays cost for serving the write request, and with probability $\frac{1}{D}$, for moving a copy to u_i. Thus,

$$\begin{aligned}\mathbf{E}[\Delta\text{RUCG}] &= w(S_{i-1} \cup \{u_i\}) + \frac{1}{D} \cdot D \cdot dist(s_i, u_i) \\ &\leq w(S_{i-1}) + 2dist(s_i, u_i).\end{aligned}$$

RUCG's reallocation makes $S_i = \{u_i\}$ and $w(S_i) = 0$. Thus,

$$\begin{aligned}\mathbf{E}[\Delta\Phi] &= \frac{1}{D} \cdot D \cdot \{5w(A_{i-1} \cup \{u_i\}) - 5w(A_{i-1} \cup S_{i-1}) \\ &\quad - 4(0 - w(S_{i-1}))\} \\ &= 5\Delta\text{ADON} - 5(w(A_{i-1} \cup S_{i-1}) - w(S_{i-1})) \\ &\quad -w(S_{i-1}) \\ &= 5\Delta\text{ADON} + 2dist(a_i, u_i) - 2dist(a_i, u_i) \\ &\quad - 5(w(A_{i-1} \cup S_{i-1}) - w(S_{i-1})) - w(S_{i-1}) \\ &\leq 7\Delta\text{ADON} - 2dist(a_i, u_i) \\ &\quad - 5(w(A_{i-1} \cup S_{i-1}) - w(S_{i-1})) - w(S_{i-1}).\end{aligned}$$

By the definition of s_i and the triangle inequality,

$$\begin{aligned} dist(s_i, u_i) &\le dist(a_i, u_i) + dist(a_i, S_{i-1}) \\ &\le dist(a_i, u_i) + w(A_{i-1} \cup S_{i-1}) - w(S_{i-1}) \\ &\le -\frac{y}{2}, \end{aligned}$$

where $y = -2dist(a_i, u_i) - 5(w(A_{i-1} \cup S_{i-1}) - w(S_{i-1}))$. Since $y \le -2dist(s_i, u_i)$,

$$\begin{aligned} \mathbf{E}[\Delta\Phi] &\le 7\Delta\textsc{Adon} + y - w(S_{i-1}) \\ &\le 7\Delta\textsc{Adon} - 2dist(s_i, u_i) - w(S_{i-1}) \\ &\le 7\Delta\textsc{Adon} - \mathbf{E}[\Delta\textsc{Rucg}]. \end{aligned}$$

□

The cost for RUCG's initial replication is independent of the number of requests. Therefore, the proof of Theorem 1 is completed from Lemmas 1-3. □

We can obtain the following corollary from Theorem 1 and the definition of RUCG.

Corollary 1: RUCG is a 7-competitive deterministic algorithm if $D = 1$.

4. Lower Bound on Rings

In this section, we present the following theorem:

Theorem 2: There is no randomized c-competitive file allocation algorithm against an adaptive online adversary on rings if $c < 3.833$.

Proof We prove the theorem by presenting an adaptive online adversary ADV defined as follows: Let $n \ge 4$ be an even integer, G be an n-node uniform ring, $D = 1$, and $S_0 = \{s_1\}$. Let $(u, \text{read})^+$ denote a sequence of read requests generated by ADV at a node u until a file allocation algorithm ALG replicates a copy to u. Let $(u, \text{write})^+$ denote a sequence of write requests generated by ADV at a node u until ALG has a single copy only on u.

ADV generates a sequence of requests consisting of l phases, each of which forces ALG to reallocate a single copy only on a node s_i at the beginning of the ith phase. Let $\overline{s_i}$ be the node at distance $\frac{n}{2}$ from s_i on G. Let P_1 and P_2 be the two paths connecting s_i and $\overline{s_i}$. ADV generates requests in the ith phase as follows:

Step 1: Until ALG holds copies at both s_i and $\overline{s_i}$, ADV generates $(s_i, \text{read})^+$ and $(\overline{s_i}, \text{read})^+$. Let Q_1 and Q_2 denote the sets of nodes of the longest subpaths of P_1 and P_2, respectively, such that ALG holds no copy on each internal node of the subpaths. Assume without loss of generality that the probability ϱ_i for $w(Q_1) \le w(Q_2)$ obeys $\frac{1}{2} \le \varrho_i \le 1$. Let $\rho = \frac{n\varrho_i}{2+2\varrho_i}$. Let s_{i+1} be the middle-node of $T(Q_2)$ (i.e., a node in Q_2 at distance $\lfloor \frac{w(Q_2)}{2} \rfloor$ from an end-node of $T(Q_2)$). Figure 4 shows $T(Q_1)$ and $T(Q_2)$ for an allocation of ALG.

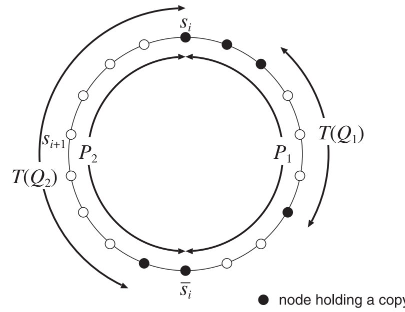

Fig. 4 $T(Q_1)$ and $T(Q_2)$ for an allocation of ALG.

Step 2: If $w(Q_2) \le \rho$, then ADV generates $(s_{i+1}, \text{write})^+$ and proceeds to the next phase.

Step 3: If $w(Q_2) > \rho$, then ADV generates (s_{i+1}, read). Then, until ALG holds copies on $V(P_2)$, ADV generates $(u, \text{read})^+$ at each node u on which ALG does not hold a copy.

Step 4: ADV generates $(s_{i+1}, \text{write})^+$ and proceeds to the next phase.

ADV replicates the copies to $V(P_2)$ before Step 1, and deletes all the copies on nodes except s_{i+1} before ADV generates $(s_{i+1}, \text{write})^+$.

Let $\Delta\textsc{Alg}$ and $\Delta\textsc{Adv}$ denote the total costs paid by ALG and ADV, respectively, in the ith phase. Let $\Delta\textsc{Alg}_j$ denote the cost paid by ALG in Step j of the ith phase. ADV pays only the cost for a replication in the ith phase. Thus,

$$\Delta\textsc{Adv} = \frac{n}{2}.$$

In Step 1, ALG pays at least the cost for serving $(\overline{s_i}, \text{read})$, for the replication to $\overline{s_i}$, for the replication to the nodes in P_2 with probability ϱ_i, and for the replication to the nodes in P_1 with probability $1 - \varrho_i$. Thus,

$$\begin{aligned} \mathbf{E}[\Delta\textsc{Alg}_1] &\ge 2 \cdot \frac{n}{2} + \varrho_i \left(\frac{n}{2} - w(Q_2)\right) \\ &\quad + (1 - \varrho_i)\left(\frac{n}{2} - w(Q_1)\right) \\ &= 3 \cdot \frac{n}{2} - (1 - \varrho_i)w(Q_1) - \varrho_i w(Q_2). \end{aligned}$$

In Step 2, ALG pays at least the cost for serving (s_{i+1}, write) and for moving the copy to s_{i+1}. Thus,

$$\begin{aligned} \mathbf{E}[\Delta\textsc{Alg}_2] &\ge \varrho_i \left(n - w(Q_2) + \left\lfloor \frac{w(Q_2)}{2} \right\rfloor\right) \\ &\quad + (1 - \varrho_i)\left(n - w(Q_1) + \left\lfloor \frac{w(Q_2)}{2} \right\rfloor\right) \\ &= n - w(Q_1) + \left\lfloor \frac{w(Q_2)}{2} \right\rfloor \\ &\quad + \varrho_i w(Q_1) - \varrho_i w(Q_2). \end{aligned}$$

In Steps 3 and 4, ALG pays at least the cost for

(s_{i+1}, read), for the replication to $V(P_2)$, and for serving (s_{i+1}, write). Thus,

$$\begin{aligned}\mathbf{E}[\Delta\textsc{Alg}_3 + \Delta\textsc{Alg}_4] \geq& \left\lfloor \frac{w(Q_2)}{2} \right\rfloor + w(Q_2) \\ &- 1 + n - w(Q_1) \\ =&\ n - w(Q_1) + \left\lfloor \frac{3w(Q_2)}{2} \right\rfloor - 1.\end{aligned}$$

If the ith phase ends via Step 2, then $w(Q_2) \leq \rho$. Thus,

$$\begin{aligned}\mathbf{E}[\Delta\textsc{Alg}] \geq&\ 3 \cdot \frac{n}{2} - (1 - \varrho_i)w(Q_1) - \varrho_i w(Q_2) + n \\ &- w(Q_1) + \left\lfloor \frac{w(Q_2)}{2} \right\rfloor + \varrho_i w(Q_1) - \varrho_i w(Q_2) \\ \geq&\ 5 \cdot \frac{n}{2} - 2(1 - \varrho_i)w(Q_1) \\ &- \left(2\varrho_i - \frac{1}{2}\right) w(Q_2) - 1 \\ \geq&\ 5 \cdot \frac{n}{2} - 2(1 - \varrho_i)\frac{n}{2} - \left(2\varrho_i - \frac{1}{2}\right)\rho - 1 \\ =&\ 3 \cdot \frac{n}{2} + \varrho_i n - \left(2\varrho_i - \frac{1}{2}\right) \frac{n\varrho_i}{2 + 2\varrho_i} - 1 \\ =&\ \frac{n}{2}\left(3 + \frac{5\varrho_i}{2 + 2\varrho_i}\right) - 1.\end{aligned}$$

If the ith phase ends via Step 3, then $w(Q_2) > \rho$. Thus,

$$\begin{aligned}\mathbf{E}[\Delta\textsc{Alg}] \geq&\ 3 \cdot \frac{n}{2} - (1 - \varrho_i)w(Q_1) - \varrho_i w(Q_2) \\ &+ n - w(Q_1) + \left\lfloor \frac{3w(Q_2)}{2} \right\rfloor - 1 \\ \geq&\ 5 \cdot \frac{n}{2} - (2 - \varrho_i)w(Q_1) + \left(\frac{3}{2} - \varrho_i\right) w(Q_2) - 2 \\ \geq&\ 5 \cdot \frac{n}{2} - (2 - \varrho_i)\frac{n}{2} + \left(\frac{3}{2} - \varrho_i\right)\rho - 2 \\ =&\ 3 \cdot \frac{n}{2} + \frac{n}{2}\varrho_i + \left(\frac{3}{2} - \varrho_i\right) \frac{n\varrho_i}{2 + 2\varrho_i} - 2 \\ =&\ \frac{n}{2}\left(3 + \frac{5\varrho_i}{2 + 2\varrho_i}\right) - 2.\end{aligned}$$

Therefore, it follows that

$$\begin{aligned}\frac{\mathbf{E}[\sum_{i=1}^{l} \Delta\textsc{Alg}]}{\sum_{i=1}^{l} \Delta\textsc{Adv}} \geq&\ \frac{\sum_{i=1}^{l} \left\{\frac{n}{2}\left(3 + \frac{5\varrho_i}{2+2\varrho_i}\right) - 2\right\}}{\frac{n}{2} \cdot l} \\ \xrightarrow[n\to\infty]{}&\ \frac{\sum_{i=1}^{l} \left(3 + \frac{5\varrho_i}{2+2\varrho_i}\right)}{l} \\ \geq&\ \frac{\frac{23}{6} \cdot l}{l} \\ =&\ \frac{23}{6} \simeq 3.833,\end{aligned}$$

which completes the proof of Theorem 2. □

We can obtain the following theorem from the proof of Theorem 2 by setting $\varrho_1 = \varrho_2 = \cdots = \varrho_l = 1$.

Theorem 3: There is no deterministic c-competitive file allocation algorithm on rings if $c < 4.25$.

References

[1] Y. Bartal, M. Charikar, and P. Indyk, "On page migration and other relaxed task systems," Theor. Comput. Sci., vol.268, no.1, pp.43–66, 2001.

[2] R. Fleischer, W. Głazek, and S. Seiden, "New results for online page replication," Theor. Comput. Sci., vol.324, no.2-3, pp.219–251, 2004.

[3] B.M. Maggs, F.M. auf der Heide, B. Vöcking, and M. Westermann, "Exploiting locality for data management in systems of limited bandwidth," Proc. IEEE Symposium on Foundations of Computer Science, pp.284–293, 1997.

[4] Y. Bartal, A. Fiat, and Y. Rabani, "Competitive algorithms for distributed data management," J. Comput. Syst. Sci., vol.51, no.3, pp.341–358, 1995.

[5] B. Awerbuch, Y. Bartal, and A. Fiat, "Competitive distributed file allocation," Inf. Comput., vol.185, no.1, pp.1–40, 2003.

[6] C. Lund, N. Reingold, J. Westbrook, and D. Yan, "Competitive online algorithms for distributed data management," SIAM J. Comput., vol.28, no.3, pp.1086–1111, 1999.

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

[8] A. Borodin and R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge Univ. Press, 1998.

Yasuyuki Kawamura received the B.E. degree in information engineering in 2007 and M.E. degree in electrical engineering and computer science in 2009 both from Kanazawa University, Kanazawa, Japan. Currently, he is with INTEC Inc.

Akira Matsubayashi received the B.E. degree in electrical and electronic engineering in 1991, M.E. degree in intelligence science in 1993, and D.E. degree in electrical and electronic engineering in 1996 all from Tokyo Institute of Technology, Tokyo, Japan. From 1996 to 2000, he was a research associate in the Department of Information Science at Utsunomiya University, Utsunomiya, Japan. Currently, he is an assistant professor in the Department of Information and Systems Engineering at Kanazawa University, Kanazawa, Japan. His research interests are in parallel and VLSI computation. He is a member of the ACM, SIAM, and the Information Processing Society of Japan.