Dynamic Page Migration Under Brownian Motion⋆

Marcin Bienkowski and Miroslaw Korzeniowski

International Graduate School of Dynamic Intelligent Systems,
University of Paderborn, Germany
{young,rudy}@upb.de

Abstract. We consider *Dynamic Page Migration* (DPM) problem, one of the fundamental subproblems of data management in dynamically changing networks. We investigate a hybrid scenario, where access patterns to the shared object are dictated by an adversary, and each processor performs a random walk in $\mathcal{X}$. We extend the previous results of [4]: we develop algorithms for the case where $\mathcal{X}$ is a ring, and prove that with high probability they achieve a competitive ratio of $\tilde{\mathcal{O}}(\min\{\sqrt[4]{D}, n\})$, where D is the size of the shared object and n is the number of nodes in the network. These results hold also for any d-dimensional torus or mesh with diameter at least $\tilde{\Omega}(\sqrt{D})$.

1 Introduction

The *Dynamic Page Migration* problem [3, 4] arises in a network of processors which share some global data. Shared variables or memory pages are stored in the local memory of these processors. If a processor wants to access (read or write) a single unit of data from a page, and the page is not stored in its local memory, it has to send a request to the processor holding the page, and appropriate data is sent back. Such transactions incur a cost which is defined to be the distance between these two processors plus a constant overhead for communication. To avoid the problem of maintaining consistency among multiple copies of the page, the model allows only one copy of the page to be stored within the network. Additionally, nodes can move with a bounded speed, thus changing the communication costs. This is typical in mobile wireless networks but also attempts to capture the dynamics of wired ones.

To reduce the communication cost, the system can migrate the page between processors. The migration cost is proportional to the cost of sending one unit of data times the size of the memory page. The problem is to decide, online, when and where to move the page in order to minimize the total cost of communication for all possible sequences of requests and network changes. The performance of

⋆ Partially supported by DFG-Sonderforschungsbereich 376 "Massive Parallelität: Algorithmen Entwurfsmethoden Anwendungen" and by the Future and Emerging Technologies programme of the EU under EU Contract 001907 DELIS "Dynamically Evolving, Large Scale Information Systems".

J.C. Cunha and P.D. Medeiros (Eds.): Euro-Par 2005, LNCS 3648, pp. 962–971, 2005.

an online algorithm is measured by competitive analysis [6, 11], i.e. by comparing its total cost to the total cost of the optimal offline algorithm on the same input sequence.

Since the input consists of two independent sequences, one describing access patterns and one for mobility of the network, it is reasonable to assume that they are created by two adversarial entities; by a request and a network adversary, respectively. The thoroughly studied problem of *page migration* (PM) [1, 2, 5, 7, 8, 12] is a special case of DPM, in which the network is static, and only the request adversary is present[1].

Whereas there exist $\mathcal{O}(1)$-competitive algorithms for PM problem [1, 2, 12] in general networks, in [3, 4] much larger lower bounds for the DPM are given. They are $\Omega(\min\{\sqrt{D}\cdot n, D\})$ if both adversaries are adaptive, and $\Omega(\sqrt{D \cdot \log n}, D^{2/3})$ if they are oblivious. D denotes the size of the page and n is the number of processors in the network. The size of this ratio motivates us to search for a reasonable restriction on the adversary. Following the approach of [4] we consider the *Brownian motion* scenario. In this scenario the network adversary is replaced by a random process – each node performs a random walk on a 1-dimensional torus, i.e., on a ring.

The Model. Following [3, 4], we describe the DPM problem formally below. The network is modelled as a set V of n nodes (processors) labelled $v_1, v_2, \ldots, v_n$ placed in a metric space $(\mathcal{X}, d)$, where d is the distance function. In our case $\mathcal{X}$ is one-dimensional discrete torus (or alternatively speaking, discrete ring) of diameter (size) B. We assume discrete time steps $t = 1, 2, \ldots$. We denote the distance between two nodes v_x and v_y in time step t as $d_t(v_x, v_y)$. Since the nodes can move, this distance can change with time. A tuple C_t describing the positions of all nodes in a time step t is called a *configuration at time t*.

An input consists of a *configuration sequence* (C_t) and a *request sequence* (σ_t), where σ_t denotes the node issuing the request at time t. These sequences are chosen as follows. First, the adversary picks the request sequence (σ_t) and the initial configuration C_0 of n points on the ring. The rest of the configuration sequence (C_t) is generated randomly, i.e. for each t, C_{t+1} is generated from C_t in the following way. For each node v its new position is chosen. Let $x_v(t)$ denote the position (coordinate) of v at step t. For each node v we define a random variable $Z_v(t)$.

$$Z_v(t) = \begin{cases} -1 \text{ with probability } 1/3 \\ 0 \text{ with probability } 1/3 \\ 1 \text{ with probability } 1/3 \end{cases} \tag{1}$$

The position of v in step $t+1$ is defined as $x_v(t+1) = x_v(t) + Z_v(t)$. This is further referred to as the *movement rule.*

Any two nodes are able to communicate directly with each other. The cost of sending a unit of data from node v_x to node v_y at time step t is defined in the following way by a cost function $c_t(v_x, v_y)$. If v_x and v_y are the same node (which

[1] In fact, in the PM model we do not have a constant overhead, but this may change the competitive ratio in this model only by a constant factor.

we denote by $v_x \equiv v_y$), then $c_t(v_x, v_y) = 0$. Otherwise, $c_t(v_x, v_y) = d_t(v_x, v_y) + 1$. We have one shared, indivisible memory page of size D, initially stored at the node v_1. The cost of moving the whole page from v_x to v_y in time step t is equal to $D \cdot c_t(v_x, v_y)$.

In time step $t \geq 1$, the positions of the nodes are set according to C_t, and then a request is issued at the node σ_t. Let $P_{\mathrm{ALG}}(t)$ denote a node keeping the algorithm ALG's page. First, ALG has to pay $c_t(P_{\mathrm{ALG}}(t), \sigma_t)$ for serving the request. Then it can optionally move the page to a new position $P'_{\mathrm{ALG}}(t)$ paying the cost $D \cdot c_t(P_{\mathrm{ALG}}(t), P'_{\mathrm{ALG}}(t))$. Sometimes, we will abuse the notation by writing that an algorithm is at v_i or moves to v_j, meaning that the algorithm's page is at v_i or the algorithm moves its page to v_j.

We consider only online algorithms, i.e. the ones which make decision in step t solely on the basis of the initial part of the input up to step t, i.e. on the sequence $C_1, \sigma_1, C_2, \sigma_2, \ldots, C_t, \sigma_t$.

In order to analyze the performance in Brownian motion scenario, we follow [4] and adapt classical competitive analysis [6, 11] for the model, where the input sequence is created both by the adversary and the stochastic process. We say that an algorithm ALG achieves competitive ratio $\mathcal{R}$ (or is $\mathcal{R}$-competitive) with probability p, if there exists a constant A, s.t. for all request sequences (σ_t) holds

$$\mathbf{Pr}_{(C_t)}\left[C_{\mathrm{ALG}}(C_t, \sigma_t) \leq \mathcal{R} \cdot C_{\mathrm{OPT}}(C_t, \sigma_t) + A\right] \geq p \ ,$$

where $C_{\mathrm{ALG}}(C_t, \sigma_t)$ and $C_{\mathrm{OPT}}(C_t, \sigma_t)$ are costs of ALG and the optimal algorithm, respectively. The probability is taken over all possible configuration sequences generated by the random movement (1).

Contribution of the Paper. We present three deterministic online algorithms: MAJL, MAJM, and MAJS, for long, middle and short diameters, respectively. Let $\gamma = \sqrt{2 \cdot \ln(n \cdot B^4)}$ and $Q = \min\{\sqrt{B}, \sqrt{D/B}, n\}$. We prove that these algorithms, with high probability, attain the competitive ratio $\mathcal{O}(\mathcal{R}) = \tilde{\mathcal{O}}(\min\{\sqrt[4]{D}, n\})$.

Algorithm	diameter	$\mathcal{R}$
MAJL	long: $B \geq 64 \cdot \gamma\sqrt{D}$	$\gamma^2 \cdot \max\{1, Q\}$
MAJM	middle: $\sqrt[3]{D} \leq B \leq 64 \cdot \gamma\sqrt{D}$	$\gamma^2 \cdot \max\{1, Q\} \cdot \log B$
MAJS	short: $B \leq \sqrt[3]{D}$	γ^2

This extends the result of [4], where an $\mathcal{O}(\log^2 D)$-competitive algorithm, working only for $B = \Theta(\sqrt{D})$ and for a constant n was presented. Furthermore, similarly to [4], it is possible to extend the result for long diameters to any d-dimensional torus or mesh of diameter $B = \Omega(\gamma \cdot \sqrt{D})$, losing only a factor of d in the competitive ratio[2].

[2] In this case, each node performs a random walk which is a superposition of independent random walks (1) in each dimension.

2 The Algorithms

In this section we present MajL, MajM and MajS. Although they differ in details, their framework is essentially the same. For the sake of this presentation, we denote the algorithm by Maj; the three algorithms will be just refinements of the Maj framework.

Maj works in phases of fixed length K. $K = D$ for MajL, $K = B^2 \cdot \log B$ for MajM, and $K = \Theta(\frac{D}{B} \cdot \log(B \cdot n))$ for MajS. In a phase P, Maj remains in one node denoted $P_{\mathrm{MAJ}}(P)$. For any time interval I and a node v, *weight* of v in I, denoted by $w_I(v)$, is defined as the number of requests issued by v during I. The name Maj is an abbreviation of *Majority*. Namely, if there exists a node $v^* \not\equiv P_{\mathrm{MAJ}}(P)$ s.t. $w_P(v^*) \geq K/2$, then Maj decides to move to v^*. For long diameters it moves immediately in the last step of P; for middle and short ones it waits for a good opportunity for the next $6 \cdot B^2 \cdot \log B$ steps. These steps are called *migration sequence*. Good opportunity means that $P_{\mathrm{MAJ}}(P)$ and v^* come to each other at the distance of at most 1. If this occurs, Maj moves in this case to v^*, otherwise it moves at the end of the migration sequence. The next phase begins right after the migration sequence.

Theorem 1. Maj *achieves competitive ratio $\mathcal{O}(\mathcal{R})$ in the Brownian motion scenario of the DPM, with high probability (w.h.p.).*

We show that there exists a constant $c_{B,D,n}$ (depending on B, D, and n), s.t. for any α, any input sequence (σ_t) and any starting configuration C_0, if (C_t) is generated according to (1), then for $\mathcal{S} = ((C_t), (\sigma_t))$

$$\mathbf{Pr}[C_{\mathrm{MAJ}}(\mathcal{S}) \leq \mathcal{O}(\mathcal{R}) \cdot C_{\mathrm{OPT}}(\mathcal{S}) + \alpha \cdot \mathcal{O}(c_{B,D,n})] \geq 1 - 2 \cdot D^{-\alpha} \quad . \qquad (2)$$

Let $c_{B,D,n} = D \cdot B + \mathcal{R} \cdot B^3 \cdot \log(B \cdot D \cdot n)$. We group phases (and corresponding migration sequences) in epochs. For MajL an epoch consists of $\lceil B^2/D \rceil$ phases; for MajM and MajS an epoch consists of just one phase, optionally with its migration sequence. This guarantees that each epoch's length is at least B^2 and at most $L_{\mathrm{p}} = \mathcal{O}((D/B + B^2) \cdot \log(Bn))$. An important property of such division into phases and epochs is the independence of the configuration sequence or the algorithm, i.e. the division can be determined entirely on the basis of the request sequence (σ_t).

We fix any input sequence $\mathcal{S}$ and divide it into epochs $M_1, M_2, \ldots$. We note that the cost of communication on the ring is bounded by $\lceil B/2 \rceil + 1 \leq B$ (we may assume that $B \geq 2$), and thus the cost of moving the page is at most $D \cdot B$. For any epoch, MajL moves the page at most $\lceil B^2/D \rceil$ times, whereas MajM and MajS move the page at most once. Since the total cost of serving requests is at most $L_{\mathrm{p}} \cdot B$, the total cost of Maj in the first two epochs is bounded by $\mathcal{O}(D \cdot B + B^3 \cdot \log(Bn)) = \mathcal{O}(c_{B,D,n})$. We may also safely assume that the input sequence consists of finished epochs only, because we can hide the Maj's cost in the last (unfinished) epoch in the additive term of (2), too.

Thus, it is sufficient to relate the cost of Maj to the cost of optimal offline algorithm Opt in any epoch, but the first and the second one.

Lemma 1 (Crucial Lemma). *For any $j \geq 3$ holds* $\mathbf{E}[C_{\text{MAJ}}(M_j)] \leq \mathcal{O}(R) \cdot \mathbf{E}[C_{\text{OPT}}(M_{j-1} \uplus M_j)]$. *The expected value is taken over all random movements in M_{j-2}, M_{j-1}, and M_j.*

We defer the proof of the Crucial Lemma to the next subsection, and we sketch the proof how Theorem 1 follows from this lemma. The complete proof will be presented in the full version of the paper.

Proof (of Theorem 1). Fix any constant α. We already bound the cost in the two first epochs. We divide the remaining ones into three disjoint sets $\mathcal{M}_0$, $\mathcal{M}_1$, and $\mathcal{M}_2$. $\mathcal{M}_k := \{M_j : j \equiv k \mod 3\}$. From the average argument, there exists $\chi \in \{0, 1, 2\}$, s.t. $C_{\text{MAJ}}(\mathcal{M}_\chi) \geq \frac{1}{3} \cdot C_{\text{MAJ}}(\mathcal{M}_0 \uplus \mathcal{M}_1 \uplus \mathcal{M}_2)$. By Lemma 1 we have $\mathbf{E}[C_{\text{MAJ}}(\mathcal{M}_\chi)] \leq \mathcal{O}(\mathcal{R}) \cdot \mathbf{E}[C_{\text{OPT}}(\mathcal{S})]$.

We consider the value of $C_{\text{MAJ}}(\mathcal{M}_\chi)$. If it is smaller than $\alpha \cdot \mathcal{O}(\mathcal{R} \cdot B^3 \cdot \ln D)$, then the whole cost of MAJ might be hidden in the additive constant of (2). Otherwise, for all each epoch $M \in \mathcal{M}_\chi$, the random variables $C_{\text{MAJ}}(M_j)$ and $C_{\text{OPT}}(M_j)$ are independent. Formally speaking, they are not independent, but the bounds on the costs of OPT and MAJ, which we use there, depend only on the randomness of the random walks in M_j and two preceding epochs (i.e. depend on disjoint events). Thus, we may apply Hoeffding bound [9] to get that both costs of $C_{\text{OPT}}(\mathcal{S})$ and $C_{\text{MAJ}}(M_\chi)$ deviate by more than a constant factor from their expected values with probability at most $D^{-\alpha}$. The calculations are similar to the calculations presented in [4]. Thus, MAJ is $\mathcal{O}(\mathcal{R})$-competitive with probability $1 - 2 \cdot D^{-\alpha}$. □

3 Proof of the Crucial Lemma

Fix any time interval I. We introduce a simple but useful notion of *auxiliary weight*. As we see later, we can use this notion to obtain a lower bound for $C_{\text{OPT}}(I)$, and an upper bound for $C_{\text{MAJ}}(I)$. Let $v_{\max}$ be the node which has the maximal weight in interval I, with ties broken arbitrarily. We define *auxiliary weight* of I as $W_{\text{A}}(I) = |I| - w_I(v_{\max})$. We note that $W_{\text{A}}(I)$ is a measure of requests' discrepancy in I. If it is low, then there exists a node $v_{\max}$, s.t. the algorithm, which remains in this node within I, pays relatively few. On the other hand, if it is high, there is no good single position for the page. Naturally, it can happen, that even if $W_{\text{A}}(I)$ is high, all nodes are very close to each other, which means that the cost of the algorithm in I could be very low. However, such a configuration sequence is very unlikely to occur.

We keep this rationale in mind, while describing a rough idea of the Crucial Lemma's proof. For a set of intervals $\mathcal{I}$, let $\mathsf{span}(\mathcal{I})$ denote the shortest time interval containing all $I \in \mathcal{I}$. For any epoch M_j (for $j \geq 3$) we prove the existence of a so-called *critical set* of disjoint intervals $\mathcal{I}(M_j)$, s.t.

1. $\mathsf{span}(\mathcal{I}(M_j)) \subseteq M_{j-1} \uplus M_j$,
2. $\mathbf{E}[C_{\text{MAJ}}(M_j)] = \mathcal{O}(\mathcal{R}_1) \cdot B \cdot \sum_{I_i \in \mathcal{I}(M_j)} W_{\text{A}}(I_i)$,
3. $\mathbf{E}[C_{\text{OPT}}(M_j)] = \Omega(1/\mathcal{R}_2) \cdot B \cdot \sum_{I_i \in \mathcal{I}(M_j)} W_{\text{A}}(I_i)$,

where the expected values are taken over the random walks in M_{j-2}, M_{j-1}, and M_j. $\mathcal{R}_1$ and $\mathcal{R}_2$ are defined as follows.

Algorithm	$\mathcal{R}_1$	$\mathcal{R}_2$
MajL	$\gamma^2 \cdot \max\{1, Q\}$	1
MajM	$\gamma^2 \cdot \max\{1, Q\} \cdot \log B$	1
MajS	1	γ^2

Clearly, if the conditions above are fulfilled, then the Crucial Lemma holds, since $\mathcal{R} = \mathcal{R}_1 \cdot \mathcal{R}_2$.

In this paper, we focus on proving the Crucial Lemma for long and middle diameters. The proof for short ones can be found in the full version of the paper. Due to space limitations we also moved the proofs of technical lemmas to the full version of the paper.

For any fixed epoch M_j ($j \geq 3$), we construct the critical set $\mathcal{I}(M_j)$, with the properties described above. Additionally, we will have

1. $|\mathsf{span}(\mathcal{I}(M_j))| \leq \frac{B^2}{(32\gamma)^2}$, and
2. for all $I_i \in \mathcal{I}(M_j)$, holds $|I_i| \leq \frac{B^2}{(16\gamma)^2 \cdot Q}$ and $|I_i| = \mathcal{O}(D)$.

3.1 Relating C_{MAJ} to Auxiliary Weight

First, we present a useful characterization of the Maj's phases. Fix any phase P of length K. By P_{migr} we denote the corresponding migration sequence (if there is none, then $P_{\text{migr}} = \emptyset$). We distinguish between three cases.

1. *Wait phase* occurs, if $w_P(P_{\text{MAJ}}(P)) > K/2$. Maj does not move and pays only for requests issued not at $P_{\text{MAJ}}(P)$. Since each request incurs a cost of at most B, $C_{\text{MAJ}}(P) \leq (K - w_P(P_{\text{MAJ}}(P)) \cdot B$.
2. *Mixed phase* occurs, if for all nodes v, $w_P(v) \leq K/2$. We have a trivial upper bound, $C_{\text{MAJ}}(P) \leq B \cdot K$.
3. *Change phase* occurs, if there exists a node $v^* \not\equiv P_{\text{MAJ}}(P)$ s.t. $w_P(v^*) > K/2$. $C_{\text{MAJ}}(P) \leq B \cdot K$. For long diameters Maj pays at most $B \cdot D = B \cdot K$ for moving the page. For middle ones it pays $B \cdot 6 \cdot B^2 \cdot \log B = \mathcal{O}(B \cdot K)$ for requests in P_{migr} and the expected cost for moving the page is $\frac{1}{B} \cdot B \cdot D + (1 - \frac{1}{B}) \cdot D = \mathcal{O}(D) = \mathcal{O}(B \cdot K / \log B)$ as follows from the technical lemma below.

Lemma 2. *Consider any two nodes v_a and v_b. If both move according to the movement rule, then with probability at least $1 - 1/B$ there exists a time step t within next $6 \cdot B^2 \cdot \log B$ steps s.t. $d_t(v_a, v_b) \leq 1$.*

For any phase P, we denote the phase preceding it by P_{prev}. As a conclusion from the phase characterization we get the following lemma.

Lemma 3. *For any phase P, $\mathbf{E}[C_{\text{MAJ}}(P \uplus P_{\text{migr}})] = \mathcal{O}(1) \cdot B \cdot W_A(P_{\text{prev}} \uplus P)$.*

Proof. If P is a wait phase, then $C_{\mathrm{MAJ}}(P) \leq B \cdot W_A(P)$. If P is a mixed phase, then $w_P(v) \leq K/2$ for all v, and hence $W_A(P) \geq K/2$. It follows that $\mathbf{E}[C_{\mathrm{MAJ}}(P)] \leq 2 \cdot B \cdot W_A(P)$. Since $P_{\mathrm{migr}} = \emptyset$ and $W_A(\cdot)$ is monotonic, the lemma holds in these cases.

If P is a change phase, then there exists a node $v^* \not\equiv P_{\mathrm{MAJ}}(P)$, to which Maj moved at the end of P. However, $w_{P_{\mathrm{prev}}}(v^*) \leq K/2$, because otherwise Maj would have moved to v^* after phase P_{prev}, and would have been in v^* in the whole phase P. Therefore, $w_{P_{\mathrm{prev}} \uplus P}(v^*) \leq \frac{3}{2} \cdot K$. This inequality holds also for any v_i. Indeed, since $w_P(v^*) > K/2$, for any node $v_i \not\equiv v^*$ holds $w_P(v_i) < K/2$, and hence $w_{P_{\mathrm{prev}} \uplus P}(v_i) < \frac{3}{2} \cdot K$. Therefore, $W_A(P_{\mathrm{prev}} \uplus P) \geq K/2$, and thus the lemma holds. □

Constructing $\mathcal{I}(M_j)$ for middle diameters. For middle diameters, an epoch M_j consists of only one phase. Therefore, $\mathbf{E}[C_{\mathrm{MAJ}}(M_j)] = \mathcal{O}(1) \cdot B \cdot W_{\mathrm{A}}(M_{j-1} \uplus M_j)$. The set $\{M_{j-1} \uplus M_j\}$ could be our critical set, consisting of one interval, but this interval is too long, i.e. has length $\Theta(B^2 \cdot \log B)$. We may shorten it to the desired length $\min\{\frac{B^2}{(32\gamma)^2}, \frac{B^2}{(16\gamma)^2 \cdot Q}\}$ losing at most a factor of $\mathcal{O}(\gamma^2 \cdot \max\{1, Q\} \cdot \log B) = \mathcal{O}(R_1)$ in auxiliary weight, using the following technical lemma.

Lemma 4. *For any interval I and any length $3 \leq \ell \leq |I|$, there exists an interval $J \subseteq I$ of length ℓ, s.t. $W_A(J) \geq \Omega(1) \cdot \frac{\ell}{|I|} \cdot W_A(I)$.*

Constructing $\mathcal{I}(M_j)$ for long diameters. In case of long diameters finding critical set is more complicated, because each epoch consists of multiple phases, and we cannot apply Lemma 3 directly. M_j consists of $\kappa := \lceil \frac{B^2}{D} \rceil$ phases $P_1, P_2, \ldots, P_\kappa$, each of length D. Let P_0 be the last phase of M_{j-1}. Let $L = B^2/(32\gamma)^2 \geq 4 \cdot D$ be the desired $\mathsf{span}(\mathcal{I}(M_j))$ length. First, we find a contiguous sequence A of phases, such that the cost of Maj in A is large. Precisely, there is a sequence A of $\lfloor L/D \rfloor - 1$ phases from M_j such that $C_{\mathrm{MAJ}}(A) \geq \Omega(\frac{\lfloor L/D \rfloor - 1}{\kappa}) \cdot C_{\mathrm{MAJ}}(M_j) = \Omega(\frac{1}{\gamma^2}) \cdot C_{\mathrm{MAJ}}(M_j)$.

Let A' be a subset of A created by taking each second phase from A, in such way that $C_{\mathrm{MAJ}}(A') \geq \frac{1}{2} \cdot C_{\mathrm{MAJ}}(A)$. As a consequence, each two change phases from A are separated by at least one phase not belonging to A'. Let $\mathcal{I}_0 := \{(P_{\mathrm{prev}} \uplus P) : P \in A'\}$. All intervals from $\mathcal{I}_0$ are disjoint and their union contains whole A. Moreover, $\mathcal{I}_0$ is contained in $\lfloor L/D \rfloor$ consecutive phases, and hence $|\mathsf{span}(\mathcal{I}_0)| \leq L$. By Lemma 3 we get $\sum_{I_i \in \mathcal{I}_0} W_A(I_i) \geq \sum_{I_i \in \mathcal{I}_0} \Omega(\frac{1}{B}) \cdot C_{\mathrm{MAJ}}(I_i) = \Omega(\frac{1}{B}) \cdot C_{\mathrm{MAJ}}(A) = \Omega(\frac{1}{\gamma^2}) \cdot \frac{1}{B} \cdot C_{\mathrm{MAJ}}(M_j)$.

Since each interval from $\mathcal{I}_0$ has length at most $\frac{B^2}{(32\gamma)^2}$, we can use Lemma 4 to shorten each $I_i \in \mathcal{I}_0$ to length $\frac{B^2}{(16\gamma)^2 \cdot Q}$ losing additional factor of $\max\{1, Q\}$. Let $\mathcal{I}(M_j)$ be the set of shortened intervals from $\mathcal{I}_0$. Then, $C_{\mathrm{MAJ}}(M_j) = B \cdot \mathcal{O}(\gamma^2 \cdot \max\{1, Q\}) \cdot \sum_{I_i \in \mathcal{I}(M_j)} W_A(I_i)$.

3.2 Relating C_{OPT} to Auxiliary Weight

Let $\mathcal{I}(M_i)$ be the critical set chosen as described in the previous subsection. Consider any single interval $I_i \in \mathcal{I}(M_j)$. Informally, a condition for incurring a

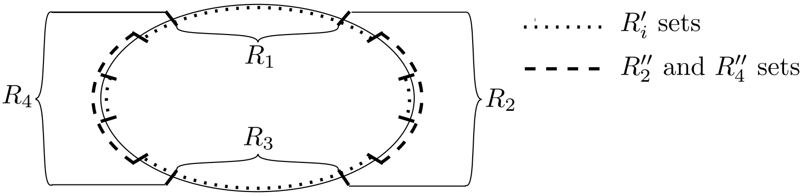

Fig. 1. Ring partitioning

high cost on any algorithm (in particular, on the optimal offline) in I_i, is that the nodes have to be distributed on the ring, so that the requests are issued from the different parts of the ring. This would assure, that an algorithm which remains at any node v in I_i pays $\Omega(B)$ for any request not at v, which amounts to at least $\Omega(B) \cdot W_{\mathrm{A}}(I_i)$.

We show that it is sufficient that nodes are well distributed at the beginning of $\mathsf{span}(\mathcal{I}(M_j))$, and that they behave nicely, i.e. they never run away quickly from their starting positions. First, we formally define these two properties. Then we prove that for a single fixed interval $I_i \in \mathcal{I}(M_j)$, these properties are fulfilled with a constant probability. Finally, we show that if they hold, then $C_{\mathrm{OPT}}(I_i) = \Omega(1) \cdot B \cdot W_{\mathrm{A}}(I_i)$. From this immediately follows that $\mathbf{E}[C_{\mathrm{OPT}}(I_i)] = \Omega(1) \cdot B \cdot W_{\mathrm{A}}(I_i)$, and by linearity of expectation we get $\mathbf{E}[C_{\mathrm{OPT}}(M_{j-1} \uplus M_j)] = \Omega(1) \cdot B \cdot \sum_{I_i \in \mathcal{I}(M_j)} W_{\mathrm{A}}(I_i)$. This would finish the proof of the Crucial Lemma.

Definition 1. *Fix any nodes configuration C and an interval I. We say that the nodes are I-*distributed*, if it is possible to partition the ring into 4 disjoint contiguous parts R_1,R_2,R_3,R_4, each containing $B/4$ points from $\mathcal{X}$, s.t. both $w_I(R_1)$ and $w_I(R_3)$ are at least $\frac{1}{16} \cdot W_A(I)$. $w_I(R_i)$ denotes the total weight accumulated in the part R_i, i.e., $w_I(R_i) = \sum_{v \in R_i} w_I(v)$ [3].*

Definition 2. *We call a configuration sequence of length ℓ* convergent*, if for any $1 \le i < j \le \ell$ and any node v, positions of node v in time step i and j differ by at most $\gamma \cdot \sqrt{j-i}$. For any time interval I, we denote the event that a configuration sequence is convergent in I by $\mathsf{conv}(I)$.*

Clearly, between the beginning of M_{j-2} (called a *starting point*) and the beginning of $\mathsf{span}(\mathcal{I}(M_j)) \subseteq M_{j-1} \uplus M_j$, there are at least B^2 steps. We call them a *mixing sequence*. We observe, that for any configuration at the starting point, at the beginning of $\mathsf{span}(\mathcal{I}(M_i))$ the position of any node is a random variable with an almost uniform distribution. Formally, let $\mathcal{D}$ be the set of all probability distributions over our space $\mathcal{X}$, whose variation distance[4] to the uniform distribution on $\mathcal{X}$ is at most $1/64$. Any $\nu \in \mathcal{D}$ we call an *almost uniform distribution*. If for each node $v \in V$, its position is a random variable with distribution $\nu \in \mathcal{D}$, and all these variables are independent, then we denote it by

[3] By $v \in A$ we mean that the position of v is in the set A.

[4] In our discrete case, the variation distance between two probability distributions ν_1 and ν_2 is defined as $\|\nu_1 - \nu_2\| := \max_{A \subseteq \mathcal{X}} |\nu_1(A) - \nu_2(A)|$.

$V \sim \mathcal{D}$. The technical lemma below is a reformulation of [4, Observation 10], and follows from the convergence rate of Markov chain [10] induced by the random walk (1).

Lemma 5. *If a node v starts from any position $x_v(t) \in \mathcal{X}$ at some step t, then its position after $k \geq B^2$ steps is a random variable with an almost uniform probability distribution.*

Thus, we may safely assume that $V \sim \mathcal{D}$ at the beginning of $\mathsf{span}(\mathcal{I}(M_j))$. The next two lemmas are proven in the full version of the paper.

Lemma 6. *Fix any interval I. If $V \sim \mathcal{D}$, then the probability that nodes are I-distributed is at least $1/7$.*

Lemma 7. *For any time interval I starting with any configuration C, if $|I| \leq B^2$, then $\mathbf{Pr}[\mathsf{conv}(I)] \geq 1/2$.*

We are interested in the events $\mathsf{conv}(\mathcal{I}(M_j))$ and that nodes are I_i-distributed at the beginning of $\mathsf{span}(\mathcal{I}(M_j))$. These events are independent, as they rely on disjoint random experiments (random walk inside and before $\mathsf{span}(\mathcal{I}(M_j))$, respectively). Thus, their intersection occurs with probability $1/14$. It remains to show that, if they both occur, then $C_{\mathrm{OPT}}(I_i) = \Omega(1) \cdot B \cdot W_{\mathrm{A}}(I_i)$

We observe, that if $\mathsf{conv}(\mathcal{I}(M_j))$, then the speed restriction imposed on the nodes' movement creates a tradeoff: an algorithm either moves its page from one point of $\mathcal{X}$ to another slowly, or it has to pay much. To formalize this observation we need the following definition.

Definition 3 (Trails). *Fix any interval I. By a trail $\mathcal{T}(I)$ we denote the sequence of points of $\mathcal{X}$, in which* OPT *had its page in interval I. The trail in one step t, $\mathcal{T}(t)$ is defined as $(P_{\mathrm{OPT}}(t))$ if* OPT *does not move, and as the sequence of points on the shortest path between $P_{\mathrm{OPT}}(t)$ and $P'_{\mathrm{OPT}}(t)$ if* OPT *moves.*

Lemma 8. *Fix any time interval I of length $\ell \leq \frac{B^2}{(16\gamma)^2 \cdot Q}$. If $\mathsf{conv}(I)$ and* OPT*'s trail $\mathcal{T}(I)$ contains two points from $\mathcal{X}$ lying at the distance of at least $B/8$, then $C_{\mathrm{OPT}}(I) = \Omega(\min\{1, 1/Q\}) \cdot D \cdot B$.*

At the beginning of $\mathsf{span}(\mathcal{I}(M_j))$, it is possible to partition the ring into four parts R_1,R_2,R_3,R_4 (see Fig. 1), s.t. $w_{I_i}(R_1), w_{I_i}(R_3) \geq W_A(I_i)/16$. Intuitively, since the configuration sequence in $\mathsf{span}(\mathcal{I}(M_j))$ is convergent, this partition is approximately preserved within whole $\mathsf{span}(\mathcal{I}(M_j))$, and thus in I_i. Formally, we define sets R'_1,R'_2,R'_3 and R'_4 as shown in Fig. 1. R'_1 (or respectively R'_3) has R_1 (R_3) in its center and contains $B/4 + 2 \cdot B/32$ points. R'_2 (or R'_4) is located in the center of R_2 (or R_4) and contains $B/8$ points. It follows that each pair of points from different R'_i sets is separated by a distance at least $B/32$. We define R''_1 (or respectively R''_3) as the part of length $3/8 \cdot B$ having R'_1 (R'_3) in its center. Thus, R''_1, R'_2, R''_3 and R'_4 create a partition of the whole ring. We make two key observations.

First, each node initially placed in R_1 (or respectively in R_3) can move by at most $\gamma \cdot \sqrt{|\mathsf{span}(\mathcal{I}(M_j))|} \leq B/32$, and thus remains within the set R_1' (or R_3') during the whole $\mathsf{span}(\mathcal{I}(M_j))$. This means that the number of requests issued in I_i at points from R_1' (or R_3') is at least $w_{I_i}(R_1) \geq W_A(I_i)/16$.

Second, OPT can either remain in $R_1'' \uplus R_2' \uplus R_4'$, remain in $R_3'' \uplus R_2' \uplus R_4'$ or its trail has to contain either all the points from R_2' or all the points from R_4'. We consider two cases, the other two are symmetric.

1. OPT remains in $R_1'' \uplus R_2' \uplus R_4'$ for the whole I_i. Then it has to pay at least $B/32$ for each of the requests issued at the points from R_3', i.e. for at least $W_A(I_i)/16$ requests. Thus, $C_{\mathrm{OPT}}(I_i) = \Omega(B \cdot W_A(I_i))$.
2. The trail of OPT's page contain whole C_2'. Since $|I_i| \leq \frac{1}{(16\gamma)^2 \cdot Q} \cdot B^2$, we can apply Lemma 8 to get $C_{\mathrm{OPT}}(I_i) = \Omega(\min\{1, 1/Q\}) \cdot D \cdot B$.

Thus, $C_{\mathrm{OPT}}(I_i) \geq \Omega(B \cdot \min\{D, D/Q, W_A(I_i)\}) = \Omega(B \cdot W_A(I_i))$, which finishes the proof.

References

1. B. Awerbuch, Y. Bartal, and A. Fiat. Competitive distributed file allocation. In *Proc. of the 25th ACM Symp. on Theory of Computing (STOC)*, pages 164–173, 1993.
2. Y. Bartal, M. Charikar, and P. Indyk. On page migration and other relaxed task systems. *Theoretical Computer Science*, 268(1):43–66, 2001.
3. M. Bienkowski, M. Dynia, and M. Korzeniowski. Improved algorithms for dynamic page migration. In *Proc. of the 22nd Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 365–376, 2005.
4. M. Bienkowski, M. Korzeniowski, and F. Meyer auf der Heide. Fighting against two adversaries: Page migration in dynamic networks. In *Proc. of the 16th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 64–73, 2004.
5. 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.
6. A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.
7. M. Chrobak, L. L. Larmore, N. Reingold, and J. Westbrook. Page migration algorithms using work functions. In *Proc. of the 4th Int. Symp. on Algorithms and Computation (ISAAC)*, pages 406–415, 1993.
8. C. Lund, N. Reingold, J. Westbrook, and D. C. K. Yan. Competitive online algorithms for distributed data management. *SIAM Journal on Computing*, 28(3):1086–1111, 1999.
9. S. Rajesekaran, P. M. Pardalos, J. H. Reif, and J. Rolim. *Handbook of Randomized Computing*, volume II. Kluwer Academic Publishers, 2001.
10. J. S. Rosenthal. Convergence rates for Markov chains. *SIAM Review*, 37(3):387–405, 1995.
11. D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.
12. J. Westbrook. Randomized algorithms for multiprocessor page migration. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 7:135–150, 1992.