A Practical Constructive Scheme for Deterministic Shared-Memory Access*

A. Pietracaprina[†‡] and F.P. Preparata[‡]

Abstract

We present an explicit memory organization scheme for distributing M data items among N memory modules where $M \in \Theta(N^{1.5-O(1/\log N)})$. Each datum is replicated into a constant number of copies assigned to distinct modules. Assuming that N processors are connected to the memories through a complete graph, we provide an access protocol so that the processors can read/write any set of $N' \leq N$ distinct data in $O((N')^{1/3}\log^* N' + \log N)$ worst-case time. The address computation can be carried out efficiently without resort to a complete memory map and using $O(1)$ internal registers per processor.

1 Introduction

Consider a parallel system with N processors and N memory modules collectively storing $M \gg N$ *variables* that are available for access by the processors. A *memory organization scheme* is sought to distribute the data among the modules so that any set of $N' \leq N$ variables can be efficiently accessed by the processors in parallel. This problem, originally referred to as *granularity problem*, naturally arises in the design and implementation of parallel systems (such as PRAMs and parallel databases) and has received considerable attention in the literature. An early survey by [Kuc77] quotes fourteen works that deal with some special cases. More recently, it has become the main focus of the large body of work concerning the simulation of the PRAM on distributed memory machines [MV84, UW87, AHMP87, KU88, LPP88, Her89, LPP90, Her90, Ran91, Mey92, KLM92].

*This paper was partially supported by NFS Grant CCR-91-96152 and

ONR Contract N00014-91-J-4052, ARPA order 8225.

[†]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801

[‡]Department of Computer Science, Brown University, Providence, RI 02912

ACM-SPAA'93-6/93/Velen,Germany.
© 1993 ACM 0-89791-599-2/93/0006/0100...$1.50

It is convenient to study this problem on a synchronous system where processors and memories are ideally thought of as being connected by a complete bipartite graph and each memory module is able to fulfill at most one access request (read/write) per time unit (*Module Parallel Computer* (MPC)) [MV84]. Thus, the time needed to access a set of variables is proportional to the maximum number of access requests that a single module must fulfill. This modeling enables us to separate the request routing problem - to be dealt with when the bipartite graph is simulated by a bounded-degree network - from the more difficult memory organization problem.

For the latter, a number of randomized schemes have been successfully developed based on the use of universal classes of hash functions to distribute the variables among the modules [MV84, KU88, LPP88, Ran91, Mey92, KLM92]. Instead, the development of efficient deterministic schemes appears to be much harder. The pioneering work of Mehlhorn and Vishkin [MV84] introduced the idea of representing each variable by several copies so that a read operation needs access only one (the most convenient) copy. This is necessary to avoid the worst case when all the requests are addressed to the same module. For $M \in O(N^c)$, they present a memory organization scheme that uses c copies per variable and allows a set of N read requests to be satisfied in time $O(cN^{1-1/c})$. However, this use of the copies penalizes the execution of write operations where all the copies of the variables must be accessed, so requiring $O(cN)$ time in the worst case.

Later, Upfal and Widgerson [UW87] proposed a more balanced use of the multiple copies exploiting the majority concept previously adopted for databases [Tho79]. Each variable is represented by $2c-1$ copies. Each copy contains the value of the variable and a timestamp indicating the last time that particular copy has been accessed. Thus, a read/write operation must access only a majority c of the copies to assure that the most recent value of the variable is always retrieved. The assignment of the copies to the memory modules is

represented by a bipartite graph $G = (V, U; E)$, where V represents the set of variables, U the set of modules, and $2c-1$ edges connect each variable to the modules storing its copies. For M polynomial in N and $c \in \Theta(\log N)$, [UW87] show that there exist graphs G with suitable expansion properties so that N variables can be accessed in $O(\log N(\log\log N)^2)$ worst-case time on the MPC. They do not provide an explicit construction for G but show that a random graph exhibits the desired properties, with high probability.

Several authors followed the ideas in [UW87] improving the time complexity and using bounded degree networks instead of the complete network of the MPC [AHMP87, HB88, Her89, LPP90, Her90, AS90]. Such schemes, however, are all based on the *existence* of similar expanding random graphs, which represents the basic shortcoming (maybe fatal from the practical standpoint) of this class of approaches, for the following reasons.

(1) No efficient way is known of testing a random graph for such expansion properties. As [PP93] have recently pointed out, the only known technique, based on the second eigenvalue, cannot be applied when the sizes of the two sets V and U differ by more than a constant factor, which is the case of memory organizations.

(2) The representation of the memory map poses substantial implementation problems. How can a processor determine, for any variable, the modules storing its copies and the physical address of each copy within its module? The hypothesis of a complete memory map stored internally in each processor appears eminently impractical and even the approach of Herley [Her90], where the memory map is distributed among the processors with only polylogarithmic storage per processor, has a very involved implementation.

Recently, we presented explicit deterministic memory organization schemes for $M \in \Theta(N^2)$ and $M \in \Theta(N^3)$ variables where any set of N variables can be accessed in $O(\sqrt{N})$ and $O(N^{2/3})$ time on the MPC, respectively, using constant redundancy [PP93]. In both schemes the implementation is simple and each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ storage. In this paper, we deal with a smaller number of variables; this is an interesting case, as [Her90] already pointed out, since, in a typical parallel application, the size of the data memory required is not enormously larger than the number of processors. Our main result is summarized in the following theorem.

Theorem 1 *For $M \in \Theta(N^{1.5-O(1/\log N)})$ there exists an explicit memory organization scheme that uses $O(1)$ copies per variable and such that any set of $N' \leq N$ variables can be accessed in $O((N')^{1/3}\log^* N' + \log N)$ worst-case time on the MPC. Moreover, each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage per processor.*

The advantage of our scheme is twofold: (1) The memory organization is *constructive*, i.e., not based on existential arguments. (2) Its implementation is simple and involves only elementary algebra, so that the address computation can be carried out efficiently with a limited use of resources. The less attractive time complexity, however, is mitigated by the fact that with constant redundancy (desirable from a practical standpoint) a polylogarithmic complexity cannot be obtained, at least for $M \in \Omega(N^{1+\epsilon})$. In fact, [UW87] show that using r copies per variables on average, the time complexity of any memory organization scheme is at least $\Omega((M/N)^{1/2r})$. In the paper we show that when the copies per variable are exactly r the lower bound can be raised to $\Omega((M/N)^{1/r})$, thus showing that our time is not so far from being optimal.

Our memory access mechanism incorporates most of the ideas and notations pioneered by [UW87] and adopted since by several authors. Each variable is represented by $q+1$ copies (i.e., $q+1 = 2c-1$), where q is an even prime power, and a read/write operation needs access only $q/2+1$ copies. The assignment of the copies to the modules is described, as before, by a bipartite graph $G = (V, U; E)$.

The paramount contribution of this paper is the construction and implementation of the graph G and the analysis of its expansion capabilities. Indeed, the scheme presented here and that of [PP93] are the first constructive approaches known to achieve sublinear, and eminently practical, *worst-case* access time with constant redundancy. Inspired by the construction presented by [Mor91] for bounded concentrators, we associate the sets V and U with two quotients of $PGL_2(q^n)$ (the group of non-singular 2×2 matrices over the field $\mathbb{F}_{q^n}$, modulo its center), and define the edges as intersections of cosets. This kind of graphs exhibit a rich algebraic structure and a remarkable 'isotropy' (the graph appears the same from any of its vertices), which make them very attractive for many computer science applications. Here, in particular, we prove that for any set $S \subset V$, we have $|\Gamma(S)| \in \Omega(|S|^{2/3}q)$, where $\Gamma(S) \subset U$ represents the image of S over U in G. This is accomplished without resort to the (here ineffective) standard second-eigenvalue bound, and solves one of the problems

left open in [Mor91]. This result allows us to achieve the time complexity claimed in Theorem 1 using an access protocol similar to [UW87].

The rest of the paper describes the memory organization scheme. In Section 2 we define the graph G, study its structure and give a tight bound to its expansion capability. Section 3 presents the protocol for satisfying a set of memory requests on the MPC and analyzes its time complexity. Finally, in Section 4 we discuss how to represent the graph so that each processor can efficiently determine the physical address of any copy.

2 The graph $G(V, U; E)$

Recall that V and U, with $|V| = M$ and $|U| = N$, represent the sets of variables and memory modules, respectively. Let $\mathbb{F}_k$ denote the finite field of order k, with k a prime power, and $\mathbb{F}_k^*$ its multiplicative subgroup. The *Projective Linear Group of degree 2 over* $\mathbb{F}_k$ ($PGL_2(k)$) is the group of 2×2 nonsingular matrices with entries in $\mathbb{F}_k$, modulo its center, the group of scalar matrices [Gor68]. For convenience a matrix of $PGL_2(k)$ will be written either as $\begin{pmatrix} \alpha & \beta \\ \gamma & 1 \end{pmatrix}$, with $\alpha, \beta, \gamma \in \mathbb{F}_k$, or $\begin{pmatrix} \alpha & \beta \\ 1 & 0 \end{pmatrix}$, and the equality between matrices will be modulo scalar multiples. Let q be a power of 2 and $n \geq 3$. Consider the following two subgroups of $PGL_2(q^n)$:

$$H_{n-1} \triangleq \left\{ \begin{pmatrix} a & \alpha \\ 0 & 1 \end{pmatrix}, \ a \in \mathbb{F}_q^*, \text{ and } \alpha \in \mathbb{F}_{q^n} \right\}$$

$$H_0 \triangleq PGL_2(q)$$

The graph G is defined as follows:

$$\begin{aligned} V &= PGL_2(q^n)/H_0 \\ U &= PGL_2(q^n)/H_{n-1} \end{aligned}$$

Thus, the variables are associated with the left cosets of H_0 and the modules with the left cosets of H_{n-1}. The edge set is defined as follows:

$$\begin{aligned} E = \{&(AH_0, BH_{n-1}) : A, B \in PGL_2(q^n) \\ &\text{and } AH_0 \cap BH_{n-1} \neq \emptyset\}, \end{aligned}$$

and it can be easily seen that E is in a one-to-one correspondence with the cosets of $PGL_2(q^n)/(H_0 \cap H_{n-1})$. The following properties derive from well known facts in group theory (see [Gor68]).

Fact 1 *We have*

1. $|V| = \frac{(q^n+1)q^n(q^n-1)}{(q+1)q(q-1)}$
2. $|U| = (q^n+1)\frac{q^n-1}{q-1}$
3. *The degree of each node in* V *is* $q+1$
4. *The degree of each node in* U *is* q^{n-1}

Therefore, $N \in \Theta(q^{2n-1})$ and $M \in \Theta(q^{3n-3}) = \Theta(N^{\frac{3}{2}-\frac{3}{4n-2}})$. Each variable is represented by $q+1$ copies and each module is assigned q^{n-1} copies of distinct variables.

2.1 Notations and Technical Facts

Let γ be a primitive element of $\mathbb{F}_{q^n}$. We have

$$\mathbb{F}_{q^n} = \left\{ \sum_{i=0}^{n-1} a_i \gamma^i, \quad a_i \in \mathbb{F}_q, 0 \leq i < n \right\}$$

We need to distinguish a particular subset of $\mathbb{F}_{q^n}$, consisting of all those elements that can be written as polynomials in γ with constant term a_0 equal to 0. More specifically, we define

$$P_\gamma \triangleq \left\{ \sum_{i=1}^{n-1} a_i \gamma^i, \quad a_i \in \mathbb{F}_q \right\}.$$

Note that $|P_\gamma| = q^{n-1}$.

For $v \in V$ let $\Gamma(v) \subset U$ denote the set of modules storing the copies of v (i.e., $\Gamma(v) = \{u \in U : (v,u) \in E\}$). Correspondingly, for $u \in U$ let $\Gamma(u)$ denote the set of variables storing copies in u (i.e., $\Gamma(u) = \{v \in V : (v,u) \in E\}$). Thus, Γ gives the set of neighbors of a node in the graph. Its definition is naturally extended to subsets. Also define

$$\Gamma^2(u) \triangleq \Gamma(\Gamma(u)) - u \quad \forall\, u \in U.$$

A set of representatives for the cosets associated with U can be chosen as follows:

$$\begin{aligned} U = &\left\{ \begin{pmatrix} \gamma^i & 0 \\ 0 & 1 \end{pmatrix} H_{n-1} : \ 0 \leq i < \tfrac{q^n-1}{q-1} \right\} \bigcup \\ &\bigcup \left\{ \begin{pmatrix} \alpha & \gamma^i \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \text{ and } 0 \leq i < \tfrac{q^n-1}{q-1} \right\} \end{aligned} \tag{1}$$

It is not difficult to see that all of the above cosets are distinct and, therefore, (since their number is $(q^n+1)\frac{q^n-1}{q-1}$) they form a partition of U.

2.2 Structure of G

We first list a number of facts concerning the functions Γ and Γ^2. Let $A \in PGL_2(q^n)$. We have

Lemma 1
$\Gamma(AH_0) = \{AH_{n-1}\} \cup \left\{ A \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$

Proof: We first prove that

$$\Gamma(H_0) = \{H_{n-1}\} \cup \left\{ \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$$

Note that $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in H_0 \cap H_{n-1}$ and $\begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} \in H_0$, for $a \in \mathbb{F}_q$, so that $\begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} \in H_0 \cap \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1}$ Therefore, $H_0 \cap H_{n-1} \neq \emptyset$ and $H_0 \cap \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} \neq \emptyset$, for $a \in \mathbb{F}_q$. This implies

$$\Gamma(H_0) \supseteq \{H_{n-1}\} \cup \left\{ \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$$

However, since $|\Gamma(H_0)| = q + 1$ (by Fact 1), and the set $\{H_{n-1}\} \cup \left\{ \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$, consists of $q + 1$ distinct cosets, the above containment relation holds with equality.

Now, it is easy to see that for any $A, B \in PGL_2(q^n)$,

$$H_0 \cap BH_{n-1} \neq \emptyset \iff AH_0 \cap ABH_{n-1} \neq \emptyset$$

This implies

$$\Gamma(AH_0) = \{AH_{n-1}\} \cup \left\{ A \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$$

□

Lemma 2
$\Gamma(AH_{n-1}) = \left\{ A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$

Proof: Reasoning as before, we only need to prove

$$\Gamma(H_{n-1}) = \left\{ \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$$

Since $\begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} \in H_{n-1}$, we have $H_{n-1} \cap \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 \neq \emptyset$ and, therefore,

$$\Gamma(H_{n-1}) \supseteq \left\{ \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$$

By Fact 1, $|\Gamma(H_{n-1})| = q^{n-1} = |P_\gamma|$. So, all we have to show is that for any $p_i, p_j \in P_\gamma$, $p_i \neq p_j$, $\begin{pmatrix} 1 & p_i \\ 0 & 1 \end{pmatrix} H_0 \neq \begin{pmatrix} 1 & p_j \\ 0 & 1 \end{pmatrix} H_0$. Suppose, for a contradiction, that there exists $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in H_0$ such that $\begin{pmatrix} 1 & p_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & p_j \\ 0 & 1 \end{pmatrix}$, that is, $\begin{pmatrix} cp_i + a & dp_i + b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & p_j \\ 0 & 1 \end{pmatrix}$. This implies $c = 0, d = 1, a = 1$ and $p_i + b = p_j$, which violates the definition of P_γ. □

Combining the two lemmas we can prove that, given any pair of variables, their copies share at most one memory module. This is formally stated in the following theorem.

Theorem 2 *Let $A, B \in PGL_2(q^n)$ with $AH_0 \neq BH_0$. Then $|\Gamma(AH_0) \cap \Gamma(BH_0)| \leq 1$.*

Proof: For a contradiction suppose that $|\Gamma(AH_0) \cap \Gamma(BH_0)| \geq 2$. Then, there exist $C, D \in PGL_2(q^n)$ with $CH_{n-1} \neq DH_{n-1}$, such that $CH_{n-1}, DH_{n-1} \in \Gamma(AH_0) \cap \Gamma(BH_0)$ (see Fig. 1). Without loss of generality, we assume that $CH_{n-1} = H_{n-1}$, otherwise we could premultiply A, B, C and D by C^{-1} and obtain a similar situation. With this choice of C, using Lemma 2 we can choose $A = \begin{pmatrix} 1 & p_1 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & p_2 \\ 0 & 1 \end{pmatrix}$, with $p_1, p_2 \in P_\gamma$, $p_1 \neq p_2$. Note that $AH_{n-1} = BH_{n-1} = H_{n-1}$, since, for $p \in P_\gamma$, $\begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_{n-1} = H_{n-1}$. Then by Lemma 1

$$\Gamma(AH_0) - H_{n-1} = \Gamma(AH_0) - AH_{n-1} =$$

$$= \left\{ \begin{pmatrix} 1 & p_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\} =$$

$$= \left\{ \begin{pmatrix} p_1 + a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$$

and

$$\Gamma(BH_0) - H_{n-1} = \Gamma(BH_0) - BH_{n-1} =$$

$$= \left\{ \begin{pmatrix} 1 & p_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\} =$$

$$= \left\{ \begin{pmatrix} p_2 + a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$$

This implies $|(\Gamma(AH_0) - H_{n-1}) \cap (\Gamma(BH_0) - H_{n-1})| = 0$, which contradicts the assumption that $\Gamma(AH_0)$ and $\Gamma(BH_0)$ share more than one element. □

Next we consider the function Γ^2.

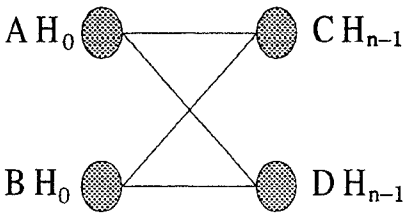

Figure 1: AH_0, BH_0, CH_{n-1}, and DH_{n-1}

Lemma 3
$\Gamma^2(AH_{n-1}) = \left\{ A \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \right\}$.

Proof: By definition,

$$\Gamma^2(AH_{n-1}) = \Gamma(\Gamma(AH_{n-1})) - AH_{n-1}$$

By Lemma 2,

$$\Gamma(AH_{n-1}) = \left\{ A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$$

By Lemma 1,

$$\Gamma\left(A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 \right) =$$

$$= \left\{ A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_{n-1} \right\} \cup$$
$$\cup \left\{ A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\} =$$

$$= \{AH_{n-1}\} \cup \left\{ A \begin{pmatrix} p+a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$$

since $\begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_{n-1} = H_{n-1}$. Hence,

$$\Gamma^2(AH_{n-1}) =$$
$$= \left\{ A \begin{pmatrix} p+a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \text{ and } p \in P_\gamma \right\}$$

The lemma is proved by observing that $\{p + a : \ p \in P_\gamma \text{ and } a \in \mathbb{F}_q\} = \mathbb{F}_{q^n}$. □

Theorem 3 *Let $A, B \in PGL_2(q^n)$ with $AH_{n-1} \neq BH_{n-1}$. Then $|\Gamma^2(AH_{n-1}) \cap \Gamma^2(BH_{n-1})| \leq q - 1$.*

Proof: Without loss of generality, assume $AH_{n-1} = H_{n-1}$ (same argument as in the proof of Theorem 2). By Lemma 3, $\Gamma^2(H_{n-1}) = \left\{ \begin{pmatrix} \delta & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \delta \in \mathbb{F}_{q^n} \right\}$. Then, we have three cases, depending upon the form of BH_{n-1}.

CASE 1: $BH_{n-1} = \begin{pmatrix} \gamma^i & 0 \\ 0 & 1 \end{pmatrix} H_{n-1}$, for some i, $1 \leq i < (q^n - 1)/(q - 1)$. Applying Lemma 3 we get

$$\Gamma^2(BH_{n-1}) =$$

$$= \left\{ \begin{pmatrix} \gamma^i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \right\} =$$

$$= \left\{ \begin{pmatrix} \gamma^i \alpha & \gamma^i \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \right\}$$

Since $\gamma^i \neq 1$, we have $|\Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})| = 0$.

CASE 2: $BH_{n-1} = \begin{pmatrix} \beta & 1 \\ 1 & 0 \end{pmatrix} H_{n-1}$, for some $\beta \in \mathbb{F}_{q^n}$. Applying Lemma 3 we get

$$\Gamma^2(BH_{n-1}) =$$

$$= \left\{ \begin{pmatrix} \beta & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \right\} =$$

$$= \left\{ \begin{pmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \right\}$$

Now, if $\alpha = 0$ then $\begin{pmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} = H_{n-1} \notin \Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})$. Otherwise, let $-\alpha^{-2} = b\gamma^k$, for some k, $0 \leq k < (q^n - 1)/(q - 1)$, and $b \in \mathbb{F}_q^*$. Simple calculations show that

$$\begin{pmatrix} \beta + \alpha^{-1} & \gamma^k \\ 1 & 0 \end{pmatrix} \begin{pmatrix} b^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \alpha^{-1} \\ 0 & 1 \end{pmatrix} H_{n-1} =$$

$$= b^{-1}\alpha^{-1} \begin{pmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1}$$

$$= \begin{pmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1}.$$

Since both $\begin{pmatrix} b^{-1} & 0 \\ 0 & 1 \end{pmatrix} \in H_{n-1}$ and $\begin{pmatrix} 1 & \alpha^{-1} \\ 0 & 1 \end{pmatrix} \in H_{n-1}$, we have $\begin{pmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} = \begin{pmatrix} \beta + \alpha^{-1} & \gamma^k \\ 1 & 0 \end{pmatrix} H_{n-1}$. Notice that $\{-\alpha^{-2} : \ \alpha \in \mathbb{F}_{q^n}^*\} = \{\gamma^{2i} : \ 0 \leq i \leq q^n - 2\}$. Since q is even, there are exactly $q - 1$ values of α such that $-\alpha^{-2} \in \mathbb{F}_q^*$, i.e., $k = 0$. These are the values of α for which $\begin{pmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} \in \Gamma^2(H_{n-1})$. Thus $|\Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})| = q - 1$.

CASE 3: $BH_{n-1} = \begin{pmatrix} \beta & \gamma^i \\ 1 & 0 \end{pmatrix} H_{n-1}$, for some $\beta \in \mathbb{F}_{q^n}$ and i, $1 \leq i < (q^n - 1)/(q - 1)$. The argument is similar to the one developed for CASE 2.

By recalling the set of representatives for U given in (1), one can easily conclude that the above three cases cover all possible cosets of U. □

Theorems 2 and 3 have two important consequences, stated below as corollaries, which will play a key role in proving the expansion capability of G.

Corollary 1 *Let $u \in U$ and consider a set of t distinct variables $v_1, \ldots, v_t$ such that $u \in \Gamma(v_i)$, for $1 \leq i \leq t$. Then*

$$|\Gamma(\{v_1, \ldots, v_t\}) - u| = qt.$$

Proof: Immediate from Theorem 2. □

Corollary 2 *Let $S \subset V$ be a set of variables and consider a set of k distinct modules $u_1, \ldots, u_k$. Let $t_i = |\Gamma(u_i) \cap S|$. Then*

$$|\Gamma(S)| \geq \sum_{i=1}^{k} qt_i - \binom{k}{2}(q-1).$$

Proof: Let $\mathcal{A}_i$ denote the set $\Gamma(\Gamma(u_i) \cap S) - u_i$. We have

$$\begin{aligned} |\Gamma(S)| &\geq \sum_{i=1}^{k} |\mathcal{A}_i| - \sum_{1 \leq i < j \leq k} |\mathcal{A}_i \cap \mathcal{A}_j| \\ &\overset{\text{by Cor. 1}}{\geq} \sum_{i=1}^{k} qt_i - \sum_{1 \leq i < j \leq k} |\Gamma^2(u_i) \cap \Gamma^2(u_j)| \\ &\overset{\text{by Th. 3}}{\geq} \sum_{i=1}^{k} qt_i - \binom{k}{2}(q-1) \end{aligned}$$

□

We are now ready to analyze the expansion properties of graph G. In particular, given a set of variables $S \subset V$ we want to bound from below the size of the set of modules $\Gamma(S)$ containing the copies of the variables in S. This problem was left open in [Mor91], where the node sets V and U were two stages of a multistage diagram, with each stage consisting of a quotient of $PGL_2(q^n)$ with respect to a different subgroup.

Theorem 4 *Let $S \subset V$. We have*

$$|\Gamma(S)| \geq \frac{1}{2^{1/3}} |S|^{2/3} q$$

Proof: We first prove that $|\Gamma(S)| > \sqrt{|S|}q$. There are $|S|(q+1)$ edges incident to members of S. Suppose $|\Gamma(S)| \leq \sqrt{|S|}q$. Then there exists a node in U receiving at least $\sqrt{|S|}(q+1)/q$ edges from S. By Corollary 1 this implies $|\Gamma(S)| > q\sqrt{|S|}(q+1)/q > \sqrt{|S|}q$, which contradicts the assumption $|\Gamma(S)| \leq \sqrt{|S|}q$.

Now, let $|\Gamma(S)| = \sqrt{|S|}\epsilon$, with $\epsilon > q$. We claim that each node of $\Gamma(S)$ receives fewer than $\sqrt{|S|}\epsilon/q$ edges from S. Indeed, suppose there is some $u \in \Gamma(S)$ receiving $s \geq \sqrt{|S|}\epsilon/q$ edges; clearly, $\Gamma(\Gamma(u) \cap S) \subset \Gamma(S)$ and, by Corollary 1, $|\Gamma(\Gamma(u) \cap S)| = sq + 1$, yielding $|\Gamma(S)| \geq |\Gamma(\Gamma(u) \cap S)| \geq \sqrt{|S|}\epsilon + 1$, a contradiction. Let t be the number of nodes in $\Gamma(S)$ receiving at least $\sqrt{|S|}q/\epsilon$ edges from S. We have

$$t\sqrt{|S|}\epsilon/q + (\sqrt{|S|}\epsilon - t)\sqrt{|S|}q/\epsilon \geq |S|(q+1)$$

This implies $t \geq \sqrt{|S|}\frac{q\epsilon}{\epsilon^2 - q^2}$, and the fact $\epsilon > q$ implies $t \geq \sqrt{|S|}q/\epsilon$. Now, let u_i, for $1 \leq i \leq \sqrt{|S|}q/\epsilon$, be distinct nodes of $\Gamma(S)$ receiving each at least $\sqrt{|S|}q/\epsilon$ edges from S. From Corollary 2 we get

$$\begin{aligned} |\Gamma(S)| &\geq \sum_{i=1}^{\sqrt{|S|}q/\epsilon} q|\Gamma(u_i) \cap S| - \binom{\sqrt{|S|}q/\epsilon}{2}(q-1) \geq \\ &\geq q(\sqrt{|S|}q/\epsilon)^2 - \frac{1}{2}(\sqrt{|S|}q/\epsilon)^2(q-1) \geq \\ &\geq |S|\frac{q^3}{2\epsilon^2} \end{aligned}$$

Combining the hypothesis $|\Gamma(S)| = \sqrt{|S|}\epsilon$ with the above inequality we obtain

$$|\Gamma(S)| \geq \frac{1}{2^{1/3}} |S|^{2/3} q$$

□

Unfortunately, it can be proved that, at least when n is composite, there exist sets S for which the bound established in Theorem 4 is tight. However, it is not known whether for n prime a stronger lower bound exists.

3 Access Protocol and Analysis

Consider an MPC with N processors, N memory modules and M variables with their copies distributed among the modules according to the graph G defined in the previous section. Suppose each processor issues a read/write request for a distinct variable. Recall that, because of the majority rule, for each variable it is sufficient to access only $q/2 + 1$ copies.

The access protocol we propose is similar to the ones by [UW87, PP93]. The processors are subdivided into $N/(q+1)$ clusters, with $q+1$ processors per cluster. Let $P(i,j)$ denote the jth processor in cluster i, for $1 \leq i \leq N/(q+1)$ and $1 \leq j \leq q+1$. Let also $v(i,j)$ denote the variable requested by processor $P(i,j)$. The protocol consists of $q+1$ phases. In the kth phase the processors of each cluster cooperate to access the variable requested by their kth companion. More specifically, processor $P(i,j)$ is in charge of the jth copy of

$v(i,k)$, for any i and j. A number of iterations are executed. In each iteration each processor tries to access its assigned variable unless it previously succeeded or other $q/2+1$ copies of the same variable have already been accessed. A memory module can satisfy at most one request per iteration, so the number of copies accessed in an iteration is equal to the number of modules receiving requests in that iteration.

Let Φ be the maximum number of iterations executed in any phase. As we will see in the next section, each processor is able to determine the physical address of any copy in $O(\log N)$ time. Therefore, it can be seen that the entire access protocol is completed in $O(q(\Phi \log q + \log N))$ steps on the MPC. We will now give an upper bound to the value of Φ. Consider a phase. At some point during the execution of the phase, a copy is said to be *alive* if it has not been accessed yet; a variable is said to be *alive* if at least $q/2+1$ of its copies are still alive. (This terminology is used only for variables and copies requested in the phase under consideration.) We need to slightly modify the result in Theorem 4. The proof is analogous.

Theorem 5 *Let $S \subset V$ be a set of live variables and let $\Gamma'(S)$ denote the memory modules storing the live copies of the variables in S. Then*

$$|\Gamma'(S)| \geq \frac{1}{4}|S|^{2/3}q$$

For a given phase, let R_k denote the number of live variables remaining after the kth iteration, where $R_0 = N$. Using Theorem 5 and an argument similar to the one used to prove [UW87, Lemma 3.3], we can show that for any $k \geq 0$

$$R_{k+1} \leq R_k \left(1 - c\left(\frac{q}{R_k}\right)^{1/3}\right) \qquad (2)$$

with $c \simeq 0.397$.

Theorem 6 *If q is a constant, $\Phi \in O(N^{1/3}\log^* N)$.*

Proof: Let $\{k_i\}_{i\geq 0}$ be a sequence of indices such that $0 = k_0 < k_1 < k_2 < \ldots < k_i < \ldots$, and define $\epsilon_i = k_i - k_{i-1}$, for $i \geq 1$. By applying (2) it follows immediately that

$$R_{k_i} \leq R_{k_{i-1}} \left(1 - c\left(\frac{q}{R_{k_{i-1}}}\right)^{1/3}\right)^{\epsilon_i} \qquad (3)$$

for any $i \geq 1$. Let $b > 1$ be a suitable constant and define the sequence $\{\delta_i\}_{i\geq 0}$ as follows

$$\begin{cases} \delta_0 &= 1 \\ \delta_{i+1} &= b^{\delta_i^{1/3}} \quad \text{for } i \geq 0 \end{cases}$$

For $i \geq 1$ we have

$$\delta_i = b^{w^{\cdot^{\cdot^{w}}}}$$

where the exponent of b is a tower of $i-1$ w's and $w = b^{1/3}$. We choose the k_i's so that

$$R_{k_i} \leq \frac{N(q+1)}{\delta_i}$$

Thus, for some smallest $j \in O(\log^* N)$, $R_{k_j} \leq 1$.

Claim *For any $i \geq 1$, $\epsilon_i \in O(N^{1/3})$.*
Proof: We need to estimate $\epsilon_i = k_i - k_{i-1}$ knowing that $R_{k_i} \leq \frac{N(q+1)}{\delta_i}$ and $R_{k_{i-1}} \leq \frac{N(q+1)}{\delta_{i-1}}$. Since the RHS of (3) is an increasing function of $R_{k_{i-1}}$, we can substitute $\frac{N(q+1)}{\delta_{i-1}}$ for $R_{k_{i-1}}$ and obtain

$$R_{k_i} \leq \frac{N(q+1)}{\delta_{i-1}} \left(1 - c\left(\frac{q}{q+1}\frac{\delta_{i-1}}{N}\right)^{1/3}\right)^{\epsilon_i}$$

Imposing

$$\frac{N(q+1)}{\delta_{i-1}} \left(1 - c\left(\frac{q}{q+1}\frac{\delta_{i-1}}{N}\right)^{1/3}\right)^{\epsilon_i} = \frac{N(q+1)}{\delta_i}$$

and solving for ϵ_i, we get

$$\epsilon_i = \frac{\log_b \delta_i - \log_b \delta_{i-1}}{-\log_b \left(1 - c\left(\frac{q}{q+1}\frac{\delta_{i-1}}{N}\right)^{1/3}\right)}$$

Using the well known fact $\log_b(1+x) \leq \log_b ex$, for $x > -1$, we get

$$\begin{aligned} \epsilon_i &\leq N^{1/3} \frac{1}{c\log_b e \left(\frac{q}{q+1}\right)^{1/3}} \frac{\log_b \delta_i - \log_b \delta_{i-1}}{\delta_i^{1/3}} \\ &\leq N^{1/3} \frac{1}{c\log_b e \left(\frac{q}{q+1}\right)^{1/3}} \frac{\log_b \delta_i}{\delta_{i-1}^{1/3}} \end{aligned}$$

By definition of δ_i, $\log_b \delta_i = \delta_{i-1}^{1/3}$, therefore,

$$\epsilon_i \leq N^{1/3} \frac{1}{c\log_b e \left(\frac{q}{q+1}\right)^{1/3}} \in O(N^{1/3})$$

that proves the claim. □

Combining the result in the claim with the fact that for some $j \in O(\log^* N)$ $R_{k_j} \leq 1$, we conclude that after $\Phi \in O(N^{1/3}\log^* N)$ iterations there are no copies alive, i.e., $R_\Phi \leq 1$. □

Therefore, if $q \in O(1)$ the entire access protocol for satisfying a set of N requests is executed in $O(N^{1/3}\log^* N)$ steps.

In general, if $N' < N$ requests are issued by the processors a similar protocol is executed with $\Phi \in O((N')^{1/3}\log^* N')$ iterations per phase. Then the total time complexity becomes $O((N')^{1/3}\log^* N' + \log N)$, as claimed in Theorem 1.

Before closing this section, we want to give the reader an idea on how far from optimal the performance of our scheme is. The following theorem, which is patterned after [UW87, Theorem 4.1] with appropriate modifications, provides a lower bound to the performance of any memory organization scheme.

Theorem 7 *In any memory organization scheme where M variables, each represented by exactly r copies, are distributed among N memory modules, the worst-case time needed to access a set of N variables is*

$$\Omega((M/N)^{1/r}$$

Note that the result is independent of any strategy adopted to access the variables (e.g., majority rule) and, any interconnection pattern between processors and memories. In our case, choosing $q = 2$ (i.e., $r = 3$), the lower bound becomes $\Omega(N^{((1/2)-O(1/\log N))(1/3)}) = \Omega(N^{(1/6)-O(1/\log N)})$.

4 Implementation

A crucial aspect of the development of a memory organization scheme concerns its implementation, an issue that has been often ignored in the past. In this section, we illustrate how our memory organization can be implemented.

Let $v_0, \ldots, v_{M-1}$ denote the variables, and $u_0, \ldots, u_{N-1}$ the memory modules. Recall that each module stores copies of $M(q+1)/N = q^{n-1}$ distinct variables. Let v_k^j denote the variable such that the kth item stored in module u_j is one of its copies. Recall that the variables are associated with the cosets of $PGL_2(q^n)/H_0$ and the modules with those of $PGL_2(q^n)/H_{n-1}$. We first need to establish the following bijections.

1. For $0 \le i < M$, $v_i \longleftrightarrow A_i H_0$, for some $A_i \in PGL_2(q^n)$.
2. For $0 \le j < N$, $u_j \longleftrightarrow B_j H_{n-1}$, for some $B_j \in PGL_2(q^n)$.
3. For $0 \le j < N$ and $0 \le k < q^{n-1}$, $v_k^j \longleftrightarrow C_k^j H_0$, for some $C_k^j \in PGL_2(q^n)$.

The goal is to define the A_i's, B_j's and C_k^j's so that a processor that wants to access a specific copy of a variable is able to efficiently determine both the module storing that copy and the physical address of the copy within the module.

1. (Bijection: $v_i \longleftrightarrow A_i H_0$). We will consider only the case of $q = 2$ and n odd, leaving the general case for an extended version of the paper [1]. Recall that

$$M = |PGL_2(2^n)/H_0| = 2^n \frac{2^{2n}-1}{6}$$

We need to find a set of M matrices belonging to distinct cosets of $PGL_2(2^n)/H_0$, ordered in such a way that given an index i, $0 \le i < M$, the ith matrix can be efficiently computed. A crucial fact we exploit in choosing the matrices is that each row, consisting of two elements from the field $\mathbb{F}_{2^n}$, can be uniquely associated with an element of the extension field $\mathbb{F}_{2^{2n}}$. More specifically, let λ be a generator of the multiplicative group $\mathbb{F}^*_{2^{2n}}$. Consider the following integers

$$\begin{aligned} \rho &\triangleq \frac{2^{2n}-1}{3} \\ \sigma &\triangleq 2^n + 1 \\ \tau &\triangleq \frac{2^n+1}{3} \end{aligned}$$

and observe that

$$\begin{aligned} \mathbb{F}^*_{2^2} &= \{\lambda^{i\rho} : 0 \le i < 3\} \\ \mathbb{F}^*_{2^n} &= \{\lambda^{i\sigma} : 0 \le i < 2^n - 1\} \end{aligned}$$

Let $w \triangleq \lambda^\rho$, so w is a generator of $\mathbb{F}^*_{2^2}$; note that $\mathbb{F}_{2^2} \subset \mathbb{F}_{2^{2n}}$ but, since n is odd, $\mathbb{F}_{2^2} \not\subset \mathbb{F}_{2^n}$. Therefore, $w \in \mathbb{F}_{2^{2n}} - \mathbb{F}_{2^n}$ and $(w, 1)$ forms a basis for $\mathbb{F}_{2^{2n}}$ over $\mathbb{F}_{2^n}$, i.e., each element of $\mathbb{F}_{2^{2n}}$ can be uniquely written as $aw + b$, for some $a, b \in \mathbb{F}_{2^n}$. Thus, we shall conveniently denote a matrix $\begin{pmatrix} x & y \\ z & v \end{pmatrix}$, with $x, y, z, v \in \mathbb{F}_{2^n}$, by the pair $\langle \alpha, \beta \rangle$, where α and β are the two elements of $\mathbb{F}_{2^{2n}}$ defined as $\alpha = xw + y$ and $\beta = zw + v$.

For $1 \le s \le (2^{n-1}-1)/3$ and $0 \le t < 2^n - 1$ let

$$k(s,t) \triangleq (s + t\sigma) \bmod \rho$$

We are now ready to define the matrices representatives for $PGL_2(2^n)/H_0$. For convenience, we partition these matrices into four sets, $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$. In the

[1] Note that when $q = 2$ there are three copies per variable, which, as [Mey92] pointed out, appears to be one of the interesting cases for practical PRAM simulations.

definition of the four sets, the indices s, t and j will take values in the following ranges:

$$1 \le s \le \frac{2^{n-1}-1}{3}$$

$$0 \le t < 2^n - 1$$

$$0 \le j < 3$$

We set

$$\begin{aligned}
\mathcal{S}_1 &\triangleq \{\langle 1, \lambda^{i\sigma} w\rangle \ : \ 0 \le i < 2^n - 1\} \\
\mathcal{S}_2 &\triangleq \{\langle 1, \lambda^{k(s,t)} w^j\rangle\} \\
\mathcal{S}_3 &\triangleq \{\langle \lambda^{k(s,t)} w^j, 1\rangle\} \\
\mathcal{S}_4 &\triangleq \{\langle \lambda^{k(s,0)}, \lambda^i w^j\rangle \ : \ 1 \le i < \rho, \ \tau \not| \ i \ \text{ and } \\
&\qquad \lambda^{k(s,0)}(w^j\lambda^i)^{-1} \notin \mathbb{F}_{2^n}^*\}
\end{aligned}$$

It is easily seen that

$$\begin{aligned}
|\mathcal{S}_1| &= 2^n - 1 \\
|\mathcal{S}_2| = |\mathcal{S}_3| &= (2^n - 1)(2^{n-1} - 1)
\end{aligned}$$

As for the cardinality of $\mathcal{S}_4$, note that there are $\frac{(2^n-1)(2^n-2)}{3}$ values of i between 1 and ρ not multiples of τ. Furthermore, it is not difficult to show that for each s there are exactly $2^n - 1$ pairs of indices i and j, with $1 \le i < \rho$, $\tau \not| \ i$ and $0 \le j < 3$, such that $\lambda^{k(s,0)}(w^j\lambda^i)^{-1} \in \mathbb{F}_{2^n}^*$. Therefore,

$$\begin{aligned}
|\mathcal{S}_4| &= \frac{2^{n-1}-1}{3}[3\frac{(2^n-1)(2^n-2)}{3} - (2^n - 1)] \\
&= \frac{2^{n-1}-1}{3}[(2^n-1)(2^n-2) - (2^n-1)]
\end{aligned}$$

Simple algebra shows that $\sum_{l=1}^{4} |\mathcal{S}_l| = M$. Thus, the A_i's defining the bijection between the variables and the cosets of $PGL_2(2^n)/H_0$, can be identified with the matrices in the above four sets, based on the following theorem whose proof is omitted due to space limitations.

Theorem 8 *The matrices in $\mathcal{S}_1 \bigcup \mathcal{S}_2 \bigcup \mathcal{S}_3 \bigcup \mathcal{S}_4$ belong to distinct cosets of $PGL_2(2^n)/H_0$, thus forming a complete set of representatives. Furthermore, the matrices can be ordered so that, given an index i, $0 \le i < M$, the ith matrix can be computed in $O(\log N)$ time using $O(1)$ storage.*

2. (Bijection: $u_j \longleftrightarrow B_j H_{n-1}$). The choice of the matrices B_j's can be naturally done using the set of representatives of $PGL_2(q^n)/H_{n-1}$ given in (1). For $0 \le s < \frac{q^n-1}{q-1}$ and $-1 \le t < q^n$ let

$$f(s,t) \triangleq s(q^n+1) + t + 1.$$

Note that $0 \le f(s,t) < N$. Also let $\mathbb{F}_{q^n} = \{\alpha_0, \ldots, \alpha_{q^n-1}\}$. Define

$$B_{f(s,t)} \triangleq \begin{cases} \begin{pmatrix} \gamma^s & 0 \\ 0 & 1 \end{pmatrix} & \text{if } t = -1 \\ \begin{pmatrix} \alpha_t & \gamma^s \\ 1 & 0 \end{pmatrix} & \text{otherwise} \end{cases}$$

3. (Bijection: $v_k^j \longleftrightarrow C_k^j H_0$). We define the C_k^j's as follows. Let $P_\gamma = \{p_0, \ldots, p_{q^{n-1}-1}\}$ (P_γ has been defined at the beginning of Section 2). For $0 \le s < \frac{q^n-1}{q-1}$, $-1 \le t < q^n$ and $0 \le k < q^{n-1}$ define

$$C_k^{f(s,t)} = B_{f(s,t)} \begin{pmatrix} 1 & p_k \\ 0 & 1 \end{pmatrix}$$

Lemma 4 *The $C_k^{f(s,t)}$'s are well defined, i.e., in graph G the node $B_{f(s,t)}H_{n-1}$ is adjacent to the nodes $C_k^{f(s,t)}H_0$, for $0 \le k < q^{n-1}$. Furthermore, if $t = -1$ then*

$$B_{f(s,t)}H_{n-1} \cap C_k^{f(s,t)}H_0 =$$

$$= \left\{ \begin{pmatrix} a\gamma^s & (p_k+b)\gamma^s \\ 0 & 1 \end{pmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}$$

Otherwise ($t \ge 0$),

$$B_{f(s,t)}H_{n-1} \cap C_k^{f(s,t)}H_0 =$$

$$= \left\{ \begin{pmatrix} a\alpha_t & (p_k+b)\alpha_t + \gamma^s \\ a & p_k + b \end{pmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}$$

Proof: By Lemma 2 and the fact that the edges of the graph are associated with the cosets of $H_0 \cap H_{n-1}$, where

$$H_0 \cap H_{n-1} = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}.$$

□

Now, suppose a processor wants to access a copy of the variable v_i. It first computes the matrix A_i such that $v_i = A_i H_0$. By Lemma 1, the copies of v_i are stored into the modules

$$\{A_i H_{n-1}\} \cup \left\{ A_i \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$$

Suppose the processor requests the copy stored into BH_{n-1}, where B is either A_i or $A_i \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix}$, for some $a \in \mathbb{F}_q$. Let $B = \begin{pmatrix} x & y \\ z & v \end{pmatrix}$, where $v = 1$ or $z = 1$ and $v = 0$, according to the usual notation. Given B

and using Lemma 4, it is not hard to derive the indices s, t and k such that the requested copy is the kth item in module $u_{f(s,t)}$. For example, if $z = 0$ we know that $t = -1$ and proceed as follows. Let $x = \gamma^r$ for some r, $0 \leq r < q^n - 1$. Then, $s = r \bmod (q^n - 1)/(q - 1)$ and $p_k + b = y\gamma^{-s}$, which allows to determine k. The other cases are handled similarly. The whole computation involves $O(n) = O(\log N)$ operations (arithmetic operations and operations in $\mathbb{F}_{q^n}$), and may be carried out using $O(1)$ internal registers.

References

[AHMP87] H. Alt, T. Hagerup, K. Mehlhorn, and F.P. Preparata. Deterministic simulation of idealized parallel computers on more realistic ones. *SIAM J. on Computing*, 16(5):808–835, 1987.

[AS90] Y. Aumann and A. Schuster. Improved memory utilization in deterministic PRAM simulations. Manuscript, 1990.

[Gor68] D. Gorenstein. *Finite Groups.* Harper and Row, New York NY, 1968.

[HB88] K.T. Herley and G. Bilardi. Deterministic simulations of PRAMs on bounded degree networks. *Proc. of the 26th Annual Allerton Conference on Communication, Control and Computation*, pages 1084–1093, 1988.

[Her89] K.T. Herley. Efficient simulations of small shared memories on bounded degree networks. *Proc. of the 30th IEEE Symp. on Foundations of Comp. Sc.*, pages 390–395, 1989.

[Her90] K.T. Herley. Space-efficient representations of shared data for parallel computers. *Proc. of the 2nd ACM Symp. on Parallel Algorithms and Architectures*, pages 407–416, 1990.

[KLM92] R. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on distributed machines. *Proc. of the 24th ACM Symp. on Theory of Comp.*, pages 318–326, 1992.

[KU88] A.R. Karlin and E. Upfal. Parallel hashing: An efficient implementation of shared memory. *J. ACM*, 35(4):876–892, 1988.

[Kuc77] D.J. Kuck. A survey of parallel machine organization and programming. *ACM Computing Surveys*, 21:339–374, 1977.

[LPP88] F. Luccio, A. Pietracaprina, and G. Pucci. A probabilistic simulation of PRAMs in VLSI. *Information Processing Lett.*, 28(3):141–147, 1988.

[LPP90] F. Luccio, A. Pietracaprina, and G. Pucci. A new scheme for the deterministic simulation of PRAMs in VLSI. *Algorithmica*, 5:529–544, 1990.

[Mey92] F. Meyer auf der Heide. Hashing strategies for simulating shared memory on distributed memory machines. *Proc. 1st Heinz Nixdorf Symp. on Parallel Architectures and their Efficient Use*, 1992. to appear in LNCS.

[Mor91] M. Morgenstern. Explicit construction of natural bounded concentrators. *Proc. of the 32th IEEE Symp. on Foundations of Comp. Sc.*, pages 392–397, 1991.

[MV84] K. Mehlhorn and U. Vishkin. Randomized and deterministic simulations of PRAMs by parallel machines with restricted granularity of parallel memories. *Acta Informatica*, 9(1):29–59, 1984.

[PP93] A. Pietracaprina and F.P. Preparata. An $O(\sqrt{n})$-worst-case-time solution to the granularity problem. *Proc. of the 10th Symp. on Theoretical Aspects of Comp. Sc.*, LNCS 665:110–119, 1993.

[Ran91] A.G. Ranade. How to emulate shared memory. *J. on Computers and System Sci.*, 42:307–326, 1991.

[Tho79] R.H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Databases Systems*, 4(2):180–209, 1979.

[UW87] E. Upfal and A. Widgerson. How to share memory in a distributed system. *J. ACM*, 34(1):116–127, 1987.