Time Lower Bounds for Sorting on Multi-Dimensional Mesh-Connected Processor Arrays *

Yijie Han

Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA

Yoshihide Igarashi

Department of Computer Science, Gunma University, Kiryu, Japan 376

Keywords: Sorting, parallel computation, time lower bound, mesh-connected processor array

1 Introduction

The problem of sorting on a mesh-connected processor array has been studied much [1-7, 9-11]. It is known that $(2d-1)n$ steps are optimal computing time within the leading term for sorting n^d items into d-dimensional snake-like order on the d-dimensional mesh-connected model [4,5,10]. However, upto now we do not know whether the snake-like order is the best for sorting. A question whether the distance bound $2n-2$ is ultimately achievable on the $n \times n$ mesh-connected model by using some super indexing scheme has been raised [7].

The authors of the present paper have shown that $2.2247n$ steps are a time lower bound independent of indexing schemes for sorting n^2 items on the $n \times n$ mesh-connected model [1]. This lower bound has recently been improved to 2.27n steps [3]. The existence of a poor indexing scheme with $4n - 2\sqrt{2n} - 3$ time lower bound have also been shown [1]. These results have been obtained using a new technique called the chain argument [1].

In this paper we develop the chain argument in order that we can apply its extended version to derive nontrivial lower bounds for sorting. We show a theorem that gives a relation between computing time for sorting n^d items and the number of processors in a certain region of the mesh-connected model. We can numerically calculate the best lower bound obtainable from the theorem. For each $d \geq 2$, our result is significantly better than the distance bound of dn.

2 Preliminary

We consider a general model of a synchronous d-dimensional mesh-connected processor array consisting of n^d identical processors. It is denoted by $M[(1..n)^d]$. Each processor at location $(i_1, ..., i_d)$ is denoted by $M[i_1, ..., i_d]$. The distance between $M[i_1, ..., i_d]$ and $M[j_1, ..., j_d]$ is defined to be $\sum_{k=1}^{d} |i_k - j_k|$ and denoted by $dis((i_1, ..., i_d), (j_1, ..., j_d))$. Processor $M[i_1, ..., i_d]$ is directly connected with processor $M[j_1, ..., j_d]$ if and only if $dis((i_1, ..., i_d), (j_1, ..., j_d)) = 1$.

*Work reported herein was partially supported by the University of Kentucky research initiation grant.

All n^d processors work in parallel with a single clock, but they may run different programs. As for sorting computation, the initial contents of $M[(1..n)^d]$ are assumed of n^d linearly ordered items, where each processor has exactly one item. The final contents of $M[(1..n)^d]$ are the sorted sequence of the items in a specific order. In one step each processor can communicate with all of its directly-connected neighbor processors. The interchange of items in a pair of directly-connected processors or the replacement of the item in a processor by the item in one of its neighbor processors can be done in one step. Our model is the same as one given by Schnorr and Shamir [10]. The computing time is defined as the number of parallel steps of the basic operations to reach the final configuration.

An index function on $M[(1..n)^d]$ is a one-to-one mapping from $\{1,...,n\}^d$ to $\{1,...,n^d\}$. For an index function I, the index of $M[i_1,...,i_d]$ is denoted by $I(i_1,...,i_d)$. A subset of $M[(1..n)^d]$ is called a region. If S is a region, $dis((i_1,...,i_d),S)$ denotes $\min\{dis((i_1,...,i_d),(j_1,...,j_d))|M[j_1,...,j_d]$ is in $S\}$. A sequence $((i_{11},...,i_{1d}),...,(i_{c1},...,i_{cd}))$ is called a chain under index function I if and only if $(I(i_{11},...,i_{1d}),..,I(i_{c1},...,i_{cd}))$ is a consecutive integer sequence. For the above chain its length is $c-1$. Processor $M[i_1,...,i_d]$ is called a corner if and only if for each t $(1 \le t \le d)$ i_t is 1 or n. For an integer i we denote $n-i+1$ by $\bar{i}$. If $M[i_1,...,i_d]$ is a corner and k is a positive integer, $\{M[j_1,...,j_d]|dis((i_1,...,i_d),(j_1,...,j_d)) < k\}$ is called a corner region and denoted by $CREG((i_1,...,i_d),k)$. If S is a region, the cardinality of S is defined as the number of processors in S and denoted by $|S|$. An ordered pair of corner regions $CREG((i_1,...,i_d),k_1)$ and $CREG((\overline{i_1},...,\overline{i_d}),k_2)$ is called a sweep. The first corner region and the second corner region of the sweep are called the residing region and the stretching region, respectively. The length of the sweep is to be defined $d(n-1)+k_1-k_2$.

3 The chain theorem

The next theorem gives a relation between the computing time, the length of a sweep and the length of a chain.

Theorem 3.1 (Chain Theorem). *For an index function I on the d-dimensional mesh-connected model and a sweep of length T, if there is no chain in its residing region such that its length is equal to the cardinality of the stretching region, then there is no algorithm of time complexity less than T for sorting n^d items on $M[(1..n)^d]$ into the order specified by I.*

Proof: Let $(CREG((i_1,...,i_d),k_1), CREG((\overline{i_1},...,\overline{i_d}),k_2))$ be a sweep. Then its length is $T = d(n-1)+k_1-k_2$. Let S be the cardinality of the stretching region. Suppose that an algorithm of time complexity $T-1$ is executed on $M[(1..n)^d]$. The effect of the initial contents of the stretching region of $M[i_1,...,i_d]$ does not appear before $((n-1)d-k_2+1)$st step. Let a be the item in $M[i_1,...,i_d]$ immediately after the $((n-1)d-k_2)$th step. The destination of a depends on the initial contents of the stretching region. By assigning different initial values to the processors in the stretching region, we can force item a into $S+1$ different positions. These different positions form a chain of length S, and should be within the residing region since the computing time is $T-1$. □

4 A poor indexing scheme

Kunde [4,5] has shown that within the leading term, $(2d-1)n$ steps are the asymptotically optimal computing time for sorting n^d items into snake-like order. We show the existence of a poorer indexing scheme than the snake-like indexing scheme.

Lemma 4.1. *Let $M[i_1, ..., i_d]$ be a corner, If $1 \leq k \leq n$, then*

$$k^d/d! \leq |CREG((i_1, ..., i_d), k)| < (k+d-1)^d/d!.$$

Proof. Follows immediately from $|CREG((i_1, ..., i_d), k)| = \begin{pmatrix} k+d-1 \\ d \end{pmatrix}$. □

Theorem 4.2 *There exists an indexing scheme such that any algorithm for sorting n^d items by the indexing scheme on $M[(1, ..., n)^d]$ takes at least $2dn - 2\lceil (d!)^{1/d} n^{1/2} \rceil - 2d + 1$ steps.*

Proof. Let $k = \lceil (d!)^{1/d} n^{1/2} \rceil$. We assume that $\lceil (d!)^{1/d} n^{1/2} \rceil < n$. Consider the sweep $(CREG((1, .., 1), d(n-1) - k + 1), CREG((n, ..., n), k))$. The length of the sweep is $2dn - 2\lceil (d!)^{1/d} n^{1/2} \rceil - 2d + 1$. From Lemma 4.1 the cardinality of the stretching region is not smaller than $\lceil n^{d/2} \rceil$. We define an indexing scheme as follows: The first $\lceil n^{d/2} \rceil$ sorted positions are in the residing region, the $(\lceil n^{d/2} \rceil + 1)$st sorted position is in the stretching region, the next $\lceil n^{d/2} \rceil$ sorted positions are in the residing region, the $(2\lceil n^{d/2} \rceil + 2)$nd sorted position is in the stretching region, and so on. Then the length of the longest chain in the residing region is $\lceil n^{d/2} \rceil - 1$. This length is smaller than the cardinality of the stretching region. Therefore, from the Chain Theorem this theorem holds. □

5 Cardinality of various regions

The union of k-corner regions is defined by

$$\bigcup_{(i_1, ..., i_d) \in \{1, n\}^d} CREG((i_1, .., i_d), k)$$

and denoted by $UCREG(k, d)$.

Lemma 5.1. *Let $tn \leq k \leq (t+1)n$, $0 \leq t \leq d-1$. Then for each $(c_1, ..., c_d)$ in $\{1, n\}^d$, the following inequalities hold:*

$$\begin{aligned} & \frac{k^d}{d!} - \frac{(k-n)^d}{(d-1)!} + \frac{(k-2n)^d}{2!(d-2)!} - \cdots + (-1)^t \frac{(k-tn)^d}{t!(d-t)!} \\ & \quad < |CREG((c_1, ..., c_d), k)| \\ & \quad < \frac{(k+d-1)^d}{d!} - \frac{(k-n+d-1)^d}{(d-1)!} + \frac{(k-2n+d-1)^d}{2!(d-2)!} - \cdots + (-1)^t \frac{(k-tn+d-1)^d}{t!(d-t)!}. \end{aligned}$$

Proof. Let us first evaluate the volume of a region on the d-dimensional real space. Let $R(k)$ be $\{(i_1, ..., i_d) |$ for each j $(1 \leq j \leq d)$ i_j is a positive real number, and $\sum_{j=1}^{d} i_j < k\}$, and let $V(k)$ be $\{(i_1, ..., i_d) |$ for each j $(1 \leq j \leq d)$ i_j is a positive real number not greater than n, and $\sum_{j=1}^{d} i_j < k\}$. If S is a region on the d-dimensional real space, the volume of S is denoted by $||S||$. Then

$$||R(k)|| = \int_0^k dx_1 \int_0^{k-x_1} dx_2 \cdots \int_0^{k-\sum_{i=1}^{d-1} x_i} dx_d = \frac{k^d}{d!}.$$

Since $CREG((c_1, ..., c_d), k)$ includes $V(k)$ and is included in $V(k+d-1)$, $||V(k)|| < |CREG((c_1, ..., c_d), k)| < ||V(k+d-1)||$. Let $\{a_1, ..., a_d\}$ be the set of properties on elements in $R(k)$, where a_i is the property that the value of the ith coordinate is greater than n. Let $R(a_i, k)$ be the subregion of $R(k)$ having property a_i, and let $N(a_i)$ be the volume of $R(a_i, k)$. We also define $R(a_i', k)$ as the subregion of $R(k)$ not having property a_i and $N(a_i')$ as the volume of $R(a_i', k)$. Since an element in $R(k)$ can have more than one property, we also use the following notations: $N(a_{i_1}, ..., a_{i_t})$ is the volume of the subregion of $R(k)$ having properties $a_{i_1}, ..., a_{i_t}$ and $N(a_{i_1}', ..., a_{i_t}')$ is the volume of the subregion of $R(k)$ not having properties $a_{i_1}, ..., a_{i_t}$. Then $||V(k)|| = N(a_1', ..., a_d')$. From the principle of inclusion and exclusion [8], $||V(k)|| = ||R(k)|| - \sum N(a_{i_1}) + \sum N(a_{i_1}, a_{i_2}) - \sum N(a_{i_1}, a_{i_2}, a_{i_3}) + ... + (-1)^d N(a_1, a_2, ..., a_d)$, where the sum of $\sum N(a_{i_1}, ..., a_{i_t})$ is taken over all combinations of t properties. If $k \le tn$, then all the terms after the tth term in the right-hand side of the above formula are 0. By a simple integration we have $N(a_{i_1}, ..., a_{i_t}) = (k-tn)^d/d!$ if $k \ge tn$. Hence

$$||V(k)|| = \frac{k^d}{d!} - \begin{pmatrix} d \\ 1 \end{pmatrix} \frac{(k-n)^d}{d!} + \begin{pmatrix} d \\ 2 \end{pmatrix} \frac{(k-2n)^d}{d!} - \cdots + (-1)^t \begin{pmatrix} d \\ t \end{pmatrix} \frac{(k-tn)^d}{d!}$$

$$= \frac{k^d}{d!} - \frac{(k-n)^d}{(d-1)!} + \frac{(k-2n)^d}{2!(d-2)!} - \cdots + (-1)^t \frac{(k-tn)^d}{t!(d-t)!}$$

if $k \ge tn$. □

Lemma 5.2. *If* $t\lfloor \frac{1}{2}n \rfloor \le k \le (t+1)\lfloor \frac{1}{2}n \rfloor$ *and* $0 \le t \le d-1$, *then*

$$\frac{(2k)^d}{d!} - \frac{(2k-n)^d}{(d-1)!} + \frac{(2k-2n)^d}{2!(d-2)!} - \cdots + (-1)^t \frac{(2k-tn)^d}{t!(d-t)!} \le |UCREG(k, d)|$$

$$\le \frac{(2(k+d-1))^d}{d!} - \frac{(2(k+d-1)-n)^d}{(d-1)!} + \frac{(2(k+d-1)-2n)^d}{2!(d-2)!}$$

$$- \cdots + (-1)^t \frac{(2(k+d-1)-tn)^d}{t!(d-t)!}.$$

Lemma 5.3. *If* $(d-t-1)\lfloor \frac{1}{2}n \rfloor \le k \le (d-t)\lfloor \frac{1}{2}n \rfloor$ *and* $0 \le t \le d-1$, *then*

$$n^d - \frac{(dn-2k)^d}{d!} + \frac{((d-1)n-2k)^d}{(d-1)!} - \frac{((d-2)n-2k)^d}{2!(d-2)!}$$

$$+ \cdots + (-1)^{t+1} \frac{((d-t)n-2k)^d}{t!(d-t)!}$$

$$\le |UCREG(k, d)| \le n^d - \frac{(dn-2(k+d-1))^d}{d!} + \frac{((d-1)n-2(k+d-1))^d}{(d-1)!}$$

$$- \frac{((d-2)n-2(k+d-1))^d}{2!(d-2)!} + \cdots + (-1)^{t+1} \frac{((d-t)n-2(k+d-1))^d}{t!(d-t)!}.$$

The proofs of Lemmas 5.2 and 5.3 are similar to that of Lemma 5.1 and are given in [2]. The values of the formulae bounding $|UCREG(k,d)|$ in Lemma 5.2 are exactly the same as the values of the formulae in Lemma 5.3. If $k \le \frac{1}{4}dn$, then the evaluation of $|UCREG(k,d)|$ by Lemma 5.2 is easier than by Lemma 5.3, and otherwise the evaluation by Lemma 5.3 is easier than by Lemma 5.4.

6 A lower bound for an arbitrary sorting order

Theorem 6.1. *Let $V = |UCREG(k_1,d)|$ and $|CREG((1,...,1),k_2)| \ge n^d - \lceil \frac{1}{2}V \rceil$, where $1 \le k_1, k_2 \le (n-1)d+1$. Then a lower bound for sorting n^d items by any indexing scheme on the d-dimensional mesh-connected model is $2d(n-1) - k_1 - k_2 + 1$ steps.*

Proof. We consider an arbitrary indexing function I on $M[(1..n)^d]$. There exists a position (i.e., a processor) in $UCREG(k_1,d)$ such that $\lceil \frac{1}{2}V \rceil \le I(b) \le n^d - \lfloor \frac{1}{2}V \rfloor + 1$ or $\lfloor \frac{1}{2}V \rfloor \le I(b) \le n^d - \lceil \frac{1}{2}V \rceil + 1$. Such a position b is in at least one corner region $CREG((i_1,...,i_d),k_1)$. Consider the sweep, $(CREG((\overline{i_1},...,\overline{i_d}), (n-1)d-k_1+1), CREG((i_1,...,i_d),k_2))$. Since b is outside the residing region, the length of the longest chain in the residing region is at most $n^d - \lceil \frac{1}{2}V \rceil - 1$. Since the cardinality of the stretching region is not less than $n^d - \lceil \frac{1}{2}V \rceil$, from the Chain Theorem a lower bound for sorting n^d items by indexing function I on the model is $2d(n-1) - k_1 - k_2 + 1$ steps. □

Good time lower bounds for arbitrary sorting order can be obtained from Theorem 6.1 by minimizing the value of $k_1 + k_2$. Although it is difficult to give a general formula of the maximized lower bound as a function of n and d, we can numerically derive it for an arbitrary d by using Lemmas 5.1-5.3. Since we are mainly interested in asymptotic lower bounds, we hereafter omit minor terms, ceilings and floors in formulae.

From Lemma 5.2, $V = |UCREG(n,d)| \ge ((2n)^d - dn^d)/d!$. If the cardinality of $CREG((i_1,...,i_d),r)$ is not less than $n^d - \frac{1}{2}V$, from Theorem 5.2 a lower bound for sorting n^d items on the d-dimensional model is $(2d-1)n - r$ steps. In this case there exists such r in the range between $(d-1)n$ and $(d-2)n$. Let $r = (d-1-t)n$, where $0 < t < 1$. Then $|CREG((i_1,...,i_d),r)| \ge n^d - \frac{1}{2}V$ if and only if $(1+t)^d - dt^d \le \frac{1}{2}(2^d - d)$. Let $t+1 = (\frac{1}{2}(2^d-d))^{1/d}$. Then $|CREG(i_1,...,i_d),r)| \ge n^d - \frac{1}{2}V$. Therefore, $((\frac{1}{2}(2^d-d))^{1/d} + d - 1)n$ is a time lower bound for sorting n^d items on the model. Hence we have the next proposition.

Proposition 6.2. *A time lower bound for sorting n^d items into an arbitrary order on the d-dimensional mesh-connected model is $((\frac{1}{2}(2^d-d))^{1/d} + d - 1)n$ steps.*

Better lower bounds can be obtained directly from Theorem 6.3. For $d = 2$ we choose $k_1 = (1 - \frac{1}{6}\sqrt{6})n$ and $k_2 = (2 - \frac{1}{3}\sqrt{6})n$. Then $|CREG(i_1,...,i_d),k_2)| \ge n^2 - \frac{1}{2}|UCREG(k_1,d)|$. Therefore, from Theorem 6.3, $4n - k_1 - k_2 \approx 2.2247n$ is a time lower bound [1].

Theorem 6.3. *An asymptotic time lower bound for sorting n^2 items into any sorting order on the 2-dimensional mesh-connected model is $(1 + \frac{1}{2}\sqrt{6})n \approx 2.2247n$ steps.*

For $d = 3$, let $k_1 = 0.87n$ and $k_2 = 1.7294n$; for $d = 4$, let $k_1 = 1.12n$ and $k_2 = 2.2667n$; for $d = 5$, let $k_1 = 1.395n$ and $k_2 = 2.7893n$. From Theorem 6.3 we have the following theorems.

Theorem 6.4. *An asymptotic time lower bound for sorting n^3 items into any sorting order on the 3-dimensional mesh-connected model is $3.4086n$ steps.*

Theorem 6.5. *An asymptotic time lower bound for sorting n^4 items into any sorting order on the 4-dimensional mesh-connected model is $4.6133n$ steps.*

Theorem 6.6 *An asymptotic time lower bound for sorting n^5 items into any sorting order on the 5-dimensional mesh-connected model is* $5.8207n$ *steps.*

Time lower bounds listed in Theorems 6.4-6.6 are the best ones obtainable from Theorem 6.1. These are better than lower bounds obtained from Proposition 6.2.

References

[1] Y. Han and Y. Igarashi, Time lower bounds for sorting on a mesh-connected processor array, in: *Proc. 3rd Aegean Workshop on Computing*, Corfu, Greece, Lecture Notes in Computer Science **319** (Springer, Berlin, 1988) 434-443.

[2] Y. Han and Y. Igarashi, Time lower bounds for parallel sorting on multi-dimensional mesh-connected processor arrays, Techn. Report TR No. 107-88. Dept. Comput. Sci., Univ. of Kentucky-Lexington, 1988.

[3] Y. Han, Y. Igarashi and M. Truszczynski, Indexing functions and time lower bounds for sorting on a mesh-connected computer, Techn. Report TR 114-88, Dept. of Comput. Sci., Univ. of Kentucky, 1988.

[4] M. Kunde, Lower bounds for sorting on mesh-connected architectures, *Acta Inform.*, **24** (1987). 121-130.

[5] M. Kunde, Optimal sorting on multi-dimensionally mesh-connected computers, in: *Proc. 4th Symp. on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science **247** (Springer, Berlin, 1987) 408-419.

[6] H.-W. Lang, M. Schimmler, H. Schmech and H. Schröder, Systolic sorting on a mesh-connected network, *IEEE Trans. Comput.*, **34** (1985) 652-658.

[7] Y. Ma, S. Sen and I.D. Scherson, The distance bound for sorting on mesh-connected processor arrays is tight, in: *Proc. 27th IEEE Symp. on Foundations of Comput. Sci.* (1986) 255-263.

[8] F.S. Roberts, *Applied Combinatorics* (Prentice-Hall, Englewood Cliffs, NJ, 1984).

[9] K. Sado and Y. Igarashi, A divide-and-conquer method of the parallel sort, IECE of Japan, Techn. Commit. of Automata and Languages, AL84-68, 1985, 41-50.

[10] C.P. Schnorr and A. Shamir, An optimal sorting algorithm for mesh-connected computers, in: *Proc. 18-th ACM Symp. on Theory of Computing*, (1986) 255-263.

[11] C.D. Thompson and H.T. Kung, Sorting on a mesh-connected parallel computer, *Comm. ACM* **20** (1977) 263-271.