Efficient Indirect All-to-All Personalized Communication on Rings and 2-D Tori

GU Naijie (顾乃杰)
Department of Computer Science & Technology, University of Science and Technology of China
Hefei 230026, P.R. China
E-mail: gunjie@ustc.edu.cn
Received July 3, 2000; revised December 18, 2000.

Abstract All-to-All personalized communication is a basic communication operation in a parallel computing environment. There are a lot of results appearing in literature. All these communication algorithms can be divided into two kinds: direct communication algorithm and indirect communication algorithm. The optimal direct all-to-all communication algorithm on rings and 2-D tori does exist. But, for indirect all-to-all communication algorithms, there is a gap between the time complexity of the already existing algorithm and the lower bound. In this paper an efficient indirect algorithm for all-to-all communication on rings and 2-D square tori with bidirection channels is presented. The algorithms is faster than any previous indirect algorithms. The main items of the time complexity of the algorithm is $p^2/8$ and $p^3/2/8$ on rings and 2-D tori respectively, both reaching the theoretical lower bound, where $p$ is the number of processors.

Keywords algorithm, all-to-all communication, parallel computing

1 Introduction

All-to-all personalized communication is a basic communication operation where each processor sends a distinct message of the same length to each other. This operation has wide applications, such as parallel sorting, fast Fourier transform, and some matrix operations\cite{1}. There are a lot of results about all-to-all communication algorithm in 2-D mesh and 2-D torus\cite{2-6}. Tseng\cite{2} and Lam\cite{4} proved the lower bounds for all-to-all communication on rings and 2-D tori. They also showed their direct algorithms using exactly as many phases as lower bounds. In their scheme, no message combination is allowed. Let $m$ be the length of message, $p$ be the number of processors, $t_s$ be the start-up time for sending a message, and $t_w$ be the per-word transmission time. Then to transmit a message of length $m$ between any pair of processors costs $t_s + mt_w$ time. Using this analysis model, the time complexities of direct algorithms\cite{2,4} are $\frac{1}{8}p^2(t_s + mt_w)$ and $\frac{1}{8}p^{3/2}(t_s + mt_w)$ on rings and 2-D tori respectively. Tseng\cite{5,6} also proposed an indirect all-to-all communication algorithm on 2-D torus. The time complexity of the algorithm is $\frac{1}{2}((\sqrt{p} + 4)t_s + \frac{1}{4}p(\sqrt{p} + 4)mt_w)$, which is larger than the lower bound.

In this paper, we present two indirect algorithms for all-to-all communication on rings and 2-D tori respectively. The time complexity of the algorithm on rings is $\frac{1}{2}pt_s + (p^2/8 + p/2)mt_w$, and the time complexity of the algorithm on 2-D tori is $\frac{1}{2}((\sqrt{p} + 4)t_s + \frac{1}{4}p(\sqrt{p} + 16)mt_w$. Our algorithm is faster than other indirect algorithms proposed in the literature when $p \geq 64$. Since the start-up time is smaller than the direct algorithm, and the main item of the transmission time is the same as the direct algorithm, when the message size is not too large, our algorithm is more efficient than the direct algorithm.

This work is supported by the Ph.D. Research Foundation The Ministry of Education, China (No.9703825).
2 All-to-All Communication on Rings

For all-to-all communication on a ring with wormhole routing, we consider each bidirectional channel as two unidirectional channels, and consider a ring of \( p \) processors as two rings of \( p/2 \) processors with unidirectional channels (assuming \( p \) is an even number). We first perform a message exchanging between two neighbor processors. Each processor sends out a message that consists of all messages with even (or odd) index. Then, we perform an all-to-all communication on each of two rings of \( p/2 \) processors with the message size \( 2m \). Denote the \( j \)-th message of processor \( PE_i \) as \( M_j^i \), and while \( i < 0 \) denote \((i + p) \mod p = i \). We can describe the algorithm as follows.

Algorithm 1. (All-to-all communication on a ring with wormhole routing)

1. For \( i = 0 \) to \( p/2 - 1 \) Para-Do
   - \( PE_{2i} \) sends messages \( M_{1,3,5,...,p-1} \) to \( PE_{2i+1} \);
   - \( PE_{2i+1} \) sends messages \( M_{2,4,6,...,p-2} \) to \( PE_{2i} \);
2. For \( k = 0 \) to \( [(p/2 - 1)/2] - 1 \) Do
   - \( PE_{2i} \) sends even indexed messages \( M_{2i+2,...,2i+[(p/2-1)/2]} \) to \( PE_{2i+2} \);
   - \( PE_{2i+1} \) sends odd indexed messages \( M_{2i+1,2i+3,...,2i+[(p/2-1)/2]-2i+1} \) to \( PE_{2i-1} \);
3. For \( k = 0 \) to \( [(p/2 - 1)/2] - 1 \) Do
   - \( PE_{2i} \) sends even indexed messages \( M_{2i+2,...,2i+[(p/2-1)/2]} \) to \( PE_{2i+2} \);
   - \( PE_{2i+1} \) sends odd indexed messages \( M_{2i+1,2i+3,...,2i+[(p/2-1)/2]-2i+1} \) to \( PE_{2i+3} \);

Assume \( p \) is an even number. The running time of Step (1) is \( t_s + twmp/2 \); the time for Step (2) is \( t_s \cdot [(p/2 - 1)/2] + twm \cdot \sum_{k=1}^{[(p/2-1)/2]} k \), and the time for Step (3) is \( t_s \cdot [(p/2 - 1)/2] + twm \cdot \sum_{k=1}^{[(p/2-1)/2]} k \). The total running time is

\[
T_{\text{ring}} = p/2 \cdot t_s + [(p^2/8) + p/2]mt_w
\]

3 All-to-All Communication on 2-D Tori

For the all-to-all communication on 2-D tori with wormhole routing, we use the “Red and Black” method. Assume \( \sqrt{p} \) is a multiple of four, we denote the processor of the \( i \)-th row and \( j \)-th column as \( PE_{i,j} \). If \( 2(i+j) \) then \( PE_{i,j} \) is called a “Red” node, otherwise it is called a “Black” node. In each pair of row and column, there are \( \sqrt{p}/2 \) “Red” nodes and \( \sqrt{p}/2 \) “Black” nodes respectively. We can consider \( \sqrt{p}/2 \) “Red” nodes in each row (or each column) as a ring, and \( \sqrt{p}/2 \) “Black” nodes as a ring. In this way, the original torus can be partitioned into four sub-tori each with \( \sqrt{p}/2 \times \sqrt{p}/2 \) processors. Our all-to-all communication algorithm can be described as follows.

Algorithm 2. (All-to-all communication on 2-D tori)

Phase 1: /*each sends \( p/2 \) messages */
   1. \( PE_{2i,2j} \leftrightarrow PE_{2i+1,2j}, PE_{2i,2j+1} \leftrightarrow PE_{2i+1,2j+1} \);
   2. \( PE_{2i,2j} \leftrightarrow PE_{2i+1,2j+1}, PE_{2i+1,2j} \leftrightarrow PE_{2i+2,2j+1} \);
Phase 2: /* Do Steps (1) and (2) of this phase in parallel*/
   1. consider processor row as a ring, all “Red” nodes running Algorithm 1, to perform all-to-all communication among processor rows;
   2. consider processor column as a ring, all “Black” nodes running Algorithm 1, to perform all-to-all communication among processor columns;
Phase 3: /* Do Steps (1) and (2) of this phase in parallel*/