부분합 문제의 하이퍼큐브 매핑에 관한 고찰

이정문 김형중

강원대학교 제어계측공학과

Note on the Partial Sum Problem Mapping into a Hypercube

Jung Moon LEE and Hyoung Joong KIM

Department of Control and Instrumentation Engineering

Kangwon National University

I. Introduction

A method for mapping rings, grids, trees and multidimensional meshes into hypercubes has been proposed [1,2,3,8]. Recently, a partial sum problem is mapped into a hypercube [5]. Since conventional mapping methods based on the ordinary binary number system are inefficient due to the irregular and excessive interprocess communication (IPC), a mapping method based on the reflected Gray code (RGC) has been utilized [1,2,3,5,7]. In this paper, a unique RGC is utilized to facilitate a mapping for partial sum problems. It is shown that the mapping provides the minimum number of IPC. A routing algorithm is provided to eliminate contentions during the IPC.

II. Partial sum problem

A general first-order linear recurrence system has the form

$$x_{k+1} = a_k x_k + b_k.$$

A partial sum x_i is defined to be $\sum_{k=1}^{i} b_k$ under the assumption that all the a_k's are 1. Computing all the partial sums is a special case of a first-order linear recurrence. For simplicity of the mapping and the routing algorithms for regular and reduced number of IPC, a special case is considered. However, the result for the partial sum problem can be easily extended to general first-order recurrence systems. The general first-order recurrence system is very important in various engineering problems, including partial differential equations, spline approximations and Kalman filters.

An n-cube is an n-dimensional hypercube consisting of 2^n nodes. Let $H(p,q)$ denote the Hamming distance between nodes p and q. The nodes are labeled by n-bit binary numbers, from 0 to 2^n-1; in general, two nodes a and b are adjacent if the Hamming distance $H(a,b)$ is 1. The Hamming distance between two nodes is the number of bits in which their node labels differ. For example, the Hamming distance between [1110] and [1010] is 1.

An algorithm to find all the partial sums has been presented [4,6]. Notice that the node [010] computes x_2 since the binary number is 2 due to the definition of ordinary binary number system. A partial sum x_i is computed in a node i, where i is mapped into a binary number [i]. During the first IPC stage, each node is to communicate with a next node in the sequence. For example, the node 0 [000] is to communicate with the node 1 [001]. In this case, the Hamming distance is 1. However, for example, the Hamming distance between the nodes 1 [001] and

2 [010] is 2. It states that the node 1 [001] should communicate with the node 2 [010] via the nodes 0 [000] or 3 [011]. Moreover, the Hamming distance between the nodes 3 [011] and 4 [100] is 3. Until the communication is completed, the relevant nodes are left to be idle.

Proposition 1: The total Hamming distance for the partial sum problem mapping into an n-dimensional hypercube based on the ordinary binary number system is $\sum_{k=1}^{n} k$.

Proof: See [5].

The above proposition states that the total Hamming distance is excessive. It results in an excessive IPC. The Hamming distance varies from 1 to n, which is very irregular. It requires different IPC mechanism for each node. The side effect can be resolved with a proper mapping method.

III. Hypercube mapping

A Gray code ia a binary code in which sequential numbers differ in one bit position only. To provides a mapping of a ring and a mesh into an n-dimensional hypercube, an RGC has been introduced [1,2,5,7]. The code has the property that the Hamming distance between the first and the last node in the sequence is also 1. A binary number $[p]_c$ denotes an RGC of an ordinary number p. Now, a partial sum x_i is computed in the node i, where i is mapped into an unique RGC. Then notice that the node $[011]_c$ computes x_2 rather than x_3.

At the first stage in the new hypercube mapping, the Hamming distance is all 1. At the second and the third stage, the Hamming distance is all 2. The total Hamming distance is 5, which is smaller than the conventional load assignment scheme by 1 when n is 3. Moreover, the routing algorithm for the IPC to be presented in the next section is regular at each stage.

Now it would be interesting to consider the lower bound on the total Hamming distance. Assume that $H(p,r) = u$ and $H(r,q) = v$ where u and v are positive integers such that $u \geq v$. Then, $H(p,q)$ has the following properties.

Property 1: $H(p,q)$ is a positive integer which belongs to the set $\{u-v, u-v+2, u-v+4, \ldots, u+v\}$.

Property 2: If $H(p,q)$ is to be odd, it is required that either u or v should be even and the rest odd. Also, if $H(p,q)$ is to be even, it is required that both u and v should be even or odd.

Optimality of the mapping for the partial sum problem based on the unique RGC to the conventional mapping based on any binary number system is shown in the following propositions.

Proposition 2: The total Hamming distance for the partial sum problem mapping into an n-dimensional hypercube based on the unique RGC is $2n-1$.

Proof: See [5].

It is shown that the total Hamming distance for the partial sum problem mapping into an n-dimensional hypercube based on the unique RGC is smaller than that of based on the ordinary binary number system. Now the total Hamming distance $2n-1$ is shown to be minimum.

Proposition 3: The total Hamming distance for the partial sum problem mapping into an n-dimensional hypercube based on any binary number system is no smaller than $2n-1$.

Proof: Property 1 implies that if the Hamming distances between communicating nodes at some stage are all 1, then those at higher stages cannot be smaller than 2. Furthermore, the maximum Hamming distance for any lower stage should be greater than 1 because all of the Hamming distances for any lower stage can be neither even nor odd according to Property 2. Thus the total Hamming distance with n stages is no smaller than $2n-1$. Q.E.D.

IV. Routing algorithm

At the first stage, each node should communicate with the next node in the sequence. At the kth stage where k is an integer over 1, each node $(p)_c$ should communicate with a node $(p+2^{k-1})_c$. However, there are two shortest routes, which can cause contention. For example, $(0)_c$ $[0000]_c$ should communicate with $(2)_c$ $[0011]_c$ via $(1)_c$ $[0001]_c$ or $(3)_c$ $[0010]_c$ at the third stage. Notice that the node $(1)_c$ is selected as an intermediate node for the IPC since the node label is between the source and the destination node labels, respectively. Other intermediate nodes are selected in this way.

Now, the problem is how to determine intermediate candidate nodes. First, compute $(p)_c * (p+2^{k-1})_c$ where $(p)_c$ is a source node, where "*" denotes the bitwise exclusive OR operation. Then, we can obtain $(2^k)_c$. The intermediate candidate nodes differ from the source and the destination nodes in the bit positions that are nonzero in (2^k). For example, with the source and the destination nodes $[1101]_c$ and $[1011]_c$, respectively, the intermediate candidate nodes are obtained by $[1111]_c$ and $[1001]_c$. We can obtain $[0110]_c$ from $[1101]_c * [1011]_c$. There are four nodes $[1001]_c$, $[1011]_c$, $[1101]_c$ and $[1111]_c$, which are all differ in the second and the third bit positions from $[0110]_c$. However, the second and the third nodes in the four nodes are the source and the destination nodes. The above-mentioned routing algorithm is summarized as follows:

```
begin
Route from (s)_c to (t)_c := (s+1)_c.
for i := 1 to n-1 do
Set (s)_c and (t)_c := (s+2^i)_c to be the source and the destination nodes.
Find the intermediate candidate nodes (p)_c and (q)_c.
Choose one node (p)_c such that (s)_c < (p)_c < (t)_c.
Route from (s)_c to (t)_c via (p)_c.
endfor
end
```

V. Conclusion

It is evident that, due to the mapping method based on the unique RGC, the number of IPC is minimized considerably for partial sum problems. Moreover, regular communication is possible. No contention has occurred during the IPC. Moreover, no node is left to be idle.

References

1. D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods* (Prentice-Hall, 1989).

2. T. F. Chan and Y. Saad, "Multigrid Algorithms on the Hypercube Multiprocessor," *IEEE Tr. Comput.* C-35 (11) (1986) 969-977.

3. G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon and D. Walker, *Solving Problems on Concurrent Processors: General Techniques and Regular Problems* (Prentice-Hall, 1988).

4. R. W. Hockney and C. R. Jesshope, *Parallel Computers 2: Architecture, Programming, and Algorithms* (Adam Hilger, 1988).

5. H. J. Kim and J. G. Lee, "Partial Sum Problem Mapping into a Hypercube," *Information Processing Letters*, (*to be pressed*), 1990.

6. M. J. Quinn, *Designing Efficient Algorithms for Parallel Computers* (McGraw-Hill, 1987).

7. E. M. Reingold, J. Nievergelt and N. Deo, *Combinatorial Algorithms: Theory and Practice* (Prentice-Hall, 1977).

8. Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Tr. Comput*. C-37 (7) (1988) 867-872.