

---

각 정점별 이웃 정보로부터 유향 사이클 그래프의  
정점 순서를 찾는 상수 시간 병렬 알고리즘

김수환\* · 최진오\*

\*부산외국어대학교

A Constant Time Parallel Algorithm for Finding a Vertex Sequence of  
the Directed Cycle Graph from the Individual Neighborhood Information

Soo-Hwan Kim\* · Jinoh Choi\*

\*Busan University of Foreign Studies

E-mail : {shkim, jochoi}@bufs.ac.kr

요 약

본 논문에서는 재구성가능한 메쉬에서 유향 사이클 그래프의 각 정점별 이웃 정보만 가지고 유향 사이클을 구성하는 정점들의 순서를 찾는 문제를 고려한다. 이 문제는 순차 알고리즘으로는 선형 시간에 해결되는 문제이지만 선형 시간보다 낮은 차수의 병렬 알고리즘을 고안하는 것은 어려운 문제이다. 모든 종류의 다각형은 유향 사이클 그래프에 해당하므로, 이 문제에 대한 해는 다각형의 정점별 이웃 정보로부터 다각형을 구성해야 하는 문제의 해결에 활용될 수 있다. 본 논문에서는 정점의 수가  $n$ 인 유향 사이클 그래프의 정점 순서를 구하는 문제를  $n \times n^2$  크기의 재구성가능한 메쉬에서 상수 시간에 해결하는 병렬 알고리즘을 제시한다.

ABSTRACT

In this paper, we consider the problem for finding a vertex sequence of the directed cycle graph from the individual neighborhood information on a reconfigurable mesh(in short, RMESH). This problem can be solved in linear time using a sequential algorithm. However, it is difficult to develop a sublinear time parallel algorithm for the problem because of its sequential nature. All kinds of polygons can be represented by directed cycles, hence a solution of the problem may be used to solving problems in which a polygon should be constructed from the adjacency information for each vertex. In this paper, we present a constant time  $n \times n^2$  RMESH algorithm for the problem with  $n$  vertices.

키워드

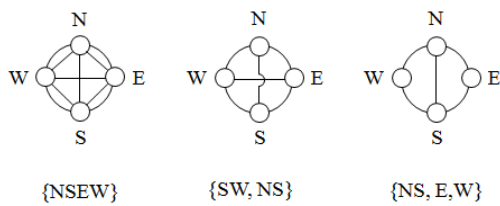
재구성가능한 메쉬, 사이클 그래프, 병렬 알고리즘

1. 서 론

재구성가능한 메쉬(Reconfigurable Mesh; 줄여서 RMESH)는 1988년 Miller 등[1]에 의해 처

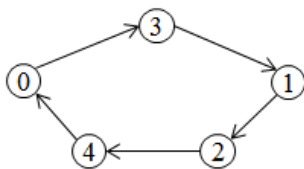
음 소개된 병렬처리 모델로서 최근까지 RMESH 모델을 기반으로 한 병렬 알고리즘 설계에 대한 논문이 많이 발표되고 있다[2]. RMESH의 기본 구조는 프로세서들을 재구성가능한 버스 시스템에 의해 메쉬 형태로 연결된 것이다. 각 프로세서

는 동(E), 서(W), 남(S), 북(N)의 4개 포트를 가지며, 알고리즘의 실행 중에 버스 스위치에 의해 각 포트 사이를 연결하거나 또는 차단하는 것이 가능하다(그림 1 참조). 프로세서의 포트 연결을 적절히 조절하여 프로세서들을 여러 버스 조각(subbus)으로 분할할 수 있다. 한 순간에 하나의 프로세서만이 버스 조각에 대한 방송(broadcast)을 할 수 있고, 같은 버스 조각에 연결된 모든 프로세서들은 방송된 자료를 상수 시간에 읽을 수 있다.  $n \times n$  RMESH의 각 프로세서는  $O(\log n)$  비트 크기의 기억공간을 상수 개 저장할 수 있고, 사칙연산을 비롯한 기본 연산을 상수 시간에 수행할 수 있다. 또한, 각 프로세서는 자신이 속한 행과 열을 인지할 수 있다.



<그림 1> 프로세서 스위치 상태의 예

본 논문에서는 유향 사이클 그래프의 에지 집합이 주어질 때, 유향 사이클의 정점 순서를 찾는 문제를 고려한다. 예를 들면, 크기가 5인 유향 사이클 그래프의 에지 집합 E가 다음과 같다고 하자.  $E = \{ \langle 2, 4 \rangle, \langle 0, 3 \rangle, \langle 3, 1 \rangle, \langle 1, 2 \rangle, \langle 4, 0 \rangle \}$ . 이 에지 집합으로부터 유향 사이클의 정점들을 순서대로 나열하면 (0, 3, 1, 2, 4)가 된다(그림 2). 이 문제를 순차 알고리즘으로 푸는 것은 다음과 같이 간단하다. 주어진 유향 사이클 그래프의 정점의 수를  $n$ 이라고 하자. 먼저, 에지 집합  $\{ \langle s_i, t_i \rangle | i = 1, \dots, n \}$ 를 소스  $s_i$ 를 기준으로 정렬한다. 정점 집합은 1부터  $n$ 까지의 정수이므로 이 정렬은 선형 시간에 수행된다. 그 다음,  $\langle s_1, t_1 \rangle$ 부터 시작하여  $s_j = t_1$ 인  $\langle s_j, t_j \rangle$ 를 상수 시간에 찾고, 같은 방식으로 모든 정점을 방문할 때까지 순차적으로 정점들을 찾아나간다. 따라서 선형 시간에 이 문제를 해결할 수 있다. 이 문제를 선형시간보다 낮은 차수의 병렬 알고리즘으로 해결하는 것은 이 문제가 가지고 있는 순차적인 성질로 인해 쉽지 않다.



<그림 2> 유향사이클 그래프의 예

본 논문에서는 현실적인 병렬 계산 모델로 주목

을 받고 있는 재구성가능한 메쉬에서 이 문제를 해결하는 병렬 알고리즘을 제시한다. 이 알고리즘은  $n \times n^2$  크기의 RMESH에서 상수시간에 수행된다.

## II. 상수 시간 병렬 알고리즘

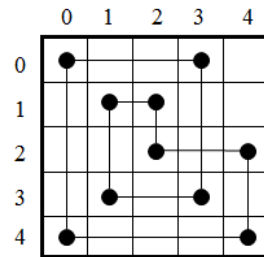
$n \times m$  RMESH에서  $i$ 번째 행과  $j$ 번째 열에 위치한 프로세서를  $R(i, j)$ 로 나타낸다, 여기서  $0 \leq i < n$ 이고  $0 \leq j < m$ 이다.  $R[i : j, a : b]$ 는  $R(i, a)$ 와  $R(j, b)$ 를 대각선 끝점으로 하는 직사각형 영역의 부메쉬를 나타낸다. 다음은 RMESH에서 수행되는 주요 연산과 수행 시간을 보여준다[3].

- ①  $n$ 개의 숫자가  $n \times n$  RMESH의 한 열에 놓여 있을 때, 이 숫자들에 대한 어떠한 순열도 상수 시간에 구할 수 있다.
- ②  $n$ 개의  $n \times n$  RMESH의 한 열에 놓여 있을 때, 상수 시간에 이 숫자들을 정렬하여 배치할 수 있다.
- ③  $n$ 개의 이진값에 대한 전위합(prefix sum)은  $n \times n$  RMESH에서 상수시간에 구할 수 있다.

이제  $n \times n^2$  RMESH  $R$ 의 첫 번째 열에 유향 사이클 그래프의 각 에지가 배치되어 있다고 하자. 즉,  $R(i, 0)$ 에 에지  $\langle s_i, t_i \rangle$ 가 저장되어 있다. 알고리즘의 개요는 다음과 같다.

### 알고리즘 1:

1.  $R$ 의 첫 번째 열에 저장된 각 에지  $\langle s, t \rangle$ 를 소스  $s$ 를 기준으로 정렬한다. 정렬 결과로서  $R(i, 0)$ 에는 에지  $\langle i, t_i \rangle$ 가 배치된다.
2.  $R$ 을  $n \times n$  크기의 부메쉬  $R_0, R_1, \dots, R_{n-1}$ 로 분할하고, 각 부메쉬  $R_k(i, 0)$ 에  $R(i, 0)$ 가 가지고 있는 정점  $t_i$ 를 저장한다.
3. 각 부메쉬  $R_k$ 는 유향사이클 그래프를 그림 3과 같은 구조로 임베딩한다.
4. 각 부메쉬  $R_k$ 는 정점  $k$ 와 정점 0 사이의 거리  $d_k$ 를 구한다. 이 거리(즉, 정점 순서에서의 위치)를  $R_k(k, 0)$ 에 저장한다.



<그림 3> 유향사이클 그래프의 배치

5.  $R_k(k,0)$ 에 저장된  $d_k$ 를  $R(k,0)$ 에 저장한 다음,  $R(k,0)$ 는 순서쌍  $\langle k, d_k \rangle$ 를 생성한다.
6.  $R$ 은 첫 번째 열에 저장된 각 순서쌍  $\langle c, d \rangle$ 를  $d$ 를 기준으로 정렬한다. 정렬 결과로서  $R(i,0)$ 에는 순서쌍  $\langle c_i, d_i \rangle$ 가 배치된다, 여기서  $d_i = i$ 이다.

알고리즘의 마지막 단계 5에서  $R$ 의 첫 번째 열에 배치된 값  $c_i$ 들은 입력으로 주어진 유향사이클 그래프의 정점 순서에 해당한다.

**정리 1.** 유향사이클 그래프의 에지 집합으로부터 유향사이클 정점 순서를  $n \times n^2$  RMESH에서 상수 시간에 구할 수 있다.

(증명) 알고리즘 1이 유향사이클 그래프의 정점 순서를 올바르게 구한다는 것은 자명하다. 이제 알고리즘 1이 상수 시간에 수행된다는 것을 알고리즘의 상세 설계와 분석을 통해 증명한다.

단계 1은 알고리즘 주요 연산 ②에 의해 상수 시간에 수행된다는 것을 알 수 있다.

단계 2에서  $R$ 을  $n \times n$  크기의 부메쉬  $R_0, R_1, \dots, R_{n-1}$ 로 분할하는 것은, 부메쉬  $R_k$ 를  $R[k \cdot n : (k+1) \cdot n - 1, 0 : n - 1]$ 로 설정하는 것을 말한다,  $k = 0, 1, \dots, n - 1$ .  $R(i,0)$ 가 가지고 있는 정점  $t_i$ 를  $R_k(i,0)$ 에 전달하는 것은 각 프로세서의 스위치 상태를 {EW, N, S}로 설정하여  $R(i,0)$ 가 방송한  $t_i$ 를 각  $R_k(i,0)$ 가 읽으면 되므로 상수 시간에 수행된다.

단계 3은 다음과 같이 수행한다.

- 3.1 각 프로세서  $R_k(i, j)$ 에  $R_k(i,0)$ 에 저장된  $t_i$ 를 전송한 다음, 각 프로세스의 스위치 상태를 {N, S, E, W}로 설정한다.
- 3.2  $i < t_i$ 이면  $R_k(i, i)$ 와  $R_k(i, t_i)$  사이의 각 프로세서는 자신의 스위치 상태를 {EW, N, S}로 설정하고,  $R_k(i, i)$ 는 자신의 W-스위치를 끊고,  $R_k(i, t_i)$ 는 자신의 E-스위치를 끊는다. 그 다음  $R_k(i, t_i)$ 와  $R_k(t_i, t_i)$  사이의 각 프로세서는 자신의 스위치 상태를 {NS, E, W}로 설정하고,  $R_k(i, t_i)$ 는 자신의 N-스위치를 끊고,  $R_k(t_i, t_i)$ 는 자신의 S-스위치를 끊는다.
- 3.3  $i > t_i$ 이면  $R_k(i, t_i)$ 와  $R_k(i, i)$  사이의 각 프로세서는 자신의 스위치 상태를 {EW, N, S}로 설정하고,  $R_k(i, t_i)$ 는 자신의 W-스위치를 끊고,  $R_k(i, i)$ 는 자신의 E-스위치를 끊는다. 그 다음  $R_k(t_i, t_i)$ 와  $R_k(i, t_i)$  사이의 각 프로세서는 자신의 스위치 상태를 {NS, E, W}로 설정하고,  $R_k(i, t_i)$ 는 자신의 S-스위치를 끊고,  $R_k(t_i, t_i)$ 는 자신의 N-스위치를 끊

는다.

위 과정은 RMESH의 기본 연산만을 사용하는 것으로 상수 시간에 수행됨을 알 수 있다.

단계 4는 다음과 같이 수행한다.

- 4.1  $R_k(0,0)$ 은 자신의 S-스위치를 끊고,  $R_k(i, i)$ 는 자신의 E, W-스위치를 끊는다.
- 4.5 모든 프로세서는 변수  $d$ 를 생성하여 0으로 초기화한다. 그 다음,  $R_k(0,0)$ 는 특별한 기호를 방송한다.  $R_k(i, i)$  중에서 이 기호를 읽은 프로세서는  $d$ 에 1을 저장한다.
- 4.6 각 프로세서의  $d$  값의 합을 구하여  $R_k(k,0)$ 에 배치한다.

위 과정은 RMESH의 기본 연산과 알고리즘 주요 연산 ③을 이용하면 상수 시간에 수행된다.

단계 5는 단계 2와 유사한 방식을 사용하여 상수 시간에 수행할 수 있다.

단계 6은 알고리즘 주요 연산 ②에 의해 상수 시간에 수행된다.

따라서 알고리즘 1은 상수 시간에 수행된다.

□

### III. 결 론

본 논문에서는 현실적인 병렬 계산 모델로 주목을 받고 있는 재구성가능한 메쉬에서 유향사이클 그래프의 에지 집합으로부터 정점 순서를 구하는 상수 시간 병렬 알고리즘을 제시했다. 이 알고리즘은 직사각형 또는 다각형들의 교차 및 합 영역, 외부 윤곽선 등을 구할 때 이용할 수 있다. 이 논문은 상수 시간에 수행되지만, 프로세서의 수를  $n^3$ 개 사용하기 때문에 프로세서-시간 곱의 측면에서 효율적이지 못하다. 차후 연구과제로서 수행 시간을 최대 로그 시간으로 제한하면서 RMESH의 프로세서의 수를  $O(n^2)$ 로 축소할 수 있는 알고리즘을 설계하는 것이다.

### 참고문헌

- [1] R. Miller, V. K. Prasanna Kumar, D. Reisis, and Q. Stout, Meshes with Reconfigurable Buses, Proc. 5th MIT Conf. on Adv. Res. in VLSI, 163-178, 1988.
- [2] R. Wankar and R. Akerkar, "Reconfigurable architectures and algorithms: a resear survey", I.J. of Computer Science and Applications, vol. 6, no. 1, 108-123, 2009.
- [3] Jang, J. and Prasanna, V. K., "An optimal sorting algorithm on reconfigurable mesh", Proc. of International Parallel Processing Symposium, pp. 130-137, 1992.