

피보나치 트리에서 후위순회를 이용한 상호 연결망의 설계

유명기*, 김용석**

**서남대학교 정보통신학과 부교수 *서남대학교 교육대학원 전자계산교육전공

전화 : 063-625-3746 / 핸드폰 : 019-689-3746

The design of interconnection network using postorder traversal on Fibonacci tree

Myoung-Ki Yoo Yong-Seok Kim
Computer Education, Seonam University
E-mail : rtg2028@cein.or.kr

Abstract

In this paper, We propose the new interconnection network which is designed to edge numbering labeling using postorder traversal which can reduce diameter when it has same node number with Hypercube, which can reduce total node numbers considering node degree and diameter on Fibonacci trees and its jump sequence of circulant is Fibonacci numbers.

It has a simple (shortest path)routing algorithm, diameter, node degree. It has a spanning subtree which is Fibonacci tree and it is embedded to Fibonacci tree. It is compared with Hypercube.

We improve diameter compared with Hypercube.

I. 서론

현대는 과학기술의 발달로 인하여 정보화 물결이 급속하게 빨라지고 있으며, 컴퓨터를 주축으로 멀티미디어가 초고속 정보망과 결합하여 사회의 변화와 발전을 주도하고 있다. 이런 변화는 인터넷 인구가 기하급수적으로 증가하고 있음을 보여주는 것으로, 이는 보다 빠르게 정보를 검색할 수 있는 새로운 상호 연결망을 연구하고 제안하게 하는 필수적인 요인이다. 이런 새로운 상호 연결망을 구성하기 위해서는 하드웨어적인 방법과 소프트웨어적인 방법이 있다. 하드웨어적인 방법은 높은 성능을 요구하는 응용 분야의 폭발적인 증대로 성능을 만족시킬 수 있는 슈퍼컴퓨터가 요구된

다. 하지만 단일 프로세서의 성능 향상은 하드웨어적으로는 기술적인 한계가 존재한다. 그러나 소프트웨어적인 방법은 이러한 기술적인 한계를 상호 연결망과 알고리즘으로 해결할 수 있는데, 이는 반도체 제조 기술의 발달로 더 많은 논리 소자를 값싸게 이용할 수 있기 때문에 병렬 처리를 위한 컴퓨터 구조에 대한 많은 연구가 진행되고 있는 게 지금의 흐름이다. 그러므로 멀티 프로세싱을 위한 시스템 성능의 향상은 다중 컴퓨터의 노드 컴퓨터를 연결하는 상호 연결망에 의해 좌우될 수밖에 없다. 멀티 프로세싱을 좌우하는 것은 위상이며, 위상을 어떻게 구성하는가에 따라 병렬성과 멀티 프로세싱을 해결할 수 있기 때문에 상호 연결망은 병렬 처리와 멀티 프로세싱에서 중요한 의미를 갖는다. 이에 본 논문에서는 새로운 상호 연결망을 설계하게 되었다.

피보나치 트리에 모든 피보나치 수를 사용하여 $f_{n+2}, n \geq 0$ -에지 번호 매김을 하는 방법은 트리에서의 노드 방문 연산인 전위 순회, 중위 순회, 후위 순회 방법을 사용할 수 있다.

본 논문에서는 하이퍼 큐브와 노드 수가 같을 때 지름을 줄이고, 분지수와 지름을 동시에 고려하여 전체 노드 수를 줄일 수 있는 노드 방문 연산 중 후위 순회 방법으로 에지 번호 매김을 하여 만들어진 피보나치 수를 원형 군의 점프열로하는 새로운 상호 연결망을 설계하는데 그 목적을 두며, 이 상호 연결망은 최단 경로 알고리즘, 지름, 분지수가 포함되며, 피보나치 트

리 상의 스페닝 서브트리와 피보나치 트리 상의 임베딩 문제를 다룬다. 또한 하이퍼 큐브와 망치도면에서 비교하였다.

II. 피보나치 트리에서 후위순회를 이용한 상호연결망

2.1 피보나치 트리

정의 2.1 피보나치 트리

(1) 높이가 0 인 피보나치 트리 FT_0 는 하나의 노드를 갖고 높이가 1인 피보나치 트리 FT_1 은 두 개의 노드를 갖는다.

(2) $k \geq 2$ 인 경우 높이가 k 인 는 두 개의 피보나치 트리 FT_{k-1} 과 FT_{k-2} 로 이 FT_k 루어 졌고, 루트 노드에 의해 FT_{k-1} 은 왼쪽 부트리로 FT_{k-2} 는 오른쪽 부트리가 된다. 피보나치 트리과 피보나치 수는 다음의 성질을 갖는다

성질 1. FT_k 에서 노드의 개수는 $F_{k+3} - 1$

성질 2. FT_k 에서 단말 노드의 개수는 F_{k+1}

성질 3. FT_k 에서 단말 노드들을 제거하고 남은 트리는 FT_{k-1}

성질 4. 높이가 k 인 피보나치트리를 FT_k 피보나치 트리에 대한 예는 그림 2.1과 같다.

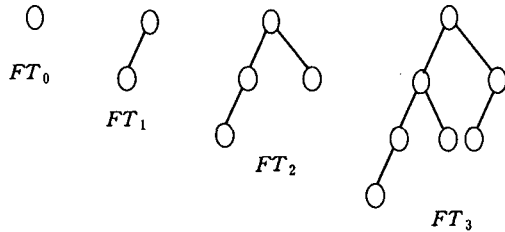


그림 2.1 피보나치 트리에 대한 예

2.2 후위 순회를 이용한 번호매김 방법

본 논문에서 제안한 피보나치 트리에서 후위순환방법을 이용한 에지 번호 매김은 피보나치 트리에서 노드 1부터 노드의 개수까지 서로 다른 노드 번호를 부여하고 에지 번호는 피보나치 수가 된다는 조건을 만족하는 번호매김 방법이다. 이러한 연구 결과는 피보나치 트리가 FP_n 의 스페닝 부트임을 쉽게 알 수가 있다.

FT_1, FT_2, FT_3, FT_4 에 대한 후위순회 방법을 이용한 에지 번호 매김의 예는 그림 2.2와 같다.

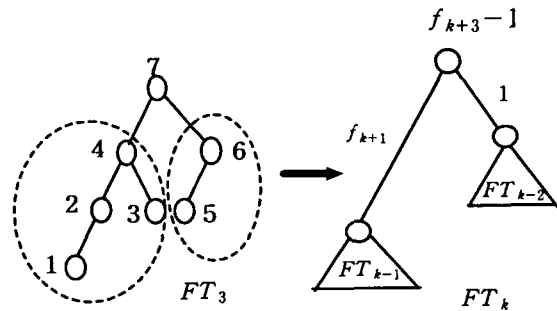


그림 2.2 후위순회 방법을 이용한 에지 번호매김의 예

2.3 후위순환법을 이용한 상호연결망

정의 2.3 FP_n 을 그래프 $G(V, E)$ 이라고 할 때 노드들의 집합은 $V = \{0, 1, \dots, f_{n+3}-1\}$ 이고 에지들의 집합은

$$E = \{(v, w) | v + f_i \equiv w \pmod{f_{n+3}}, 2 \leq i \leq n+1\} \text{이다.}$$

2.3.1 FP 의 분지수

정의 2.3.1 n 차원 FP_n 은

- ① 0부터 $f_{n+3}-1$ 인 f_{n+3} 개의 노드를 갖는다
- ② $v + f_i \equiv w \pmod{f_{n+3}}, 2 \leq i \leq n+1$ 이므로 n 개의 점프를 갖는다.

∴ 항상 각 노드는 정의2.3에 의해 $2n$ 개의 분지수를 갖는다

FP_3 에는 그림 2.3과 같다.

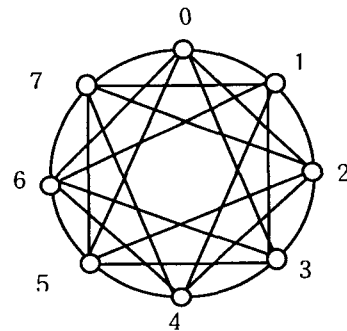


그림 2.3 FP_3 의 예

2.4 피보나치 트리의 임베딩

특정한 자료구조나 계산구조를 기반으로 하는 작업을 쉽게 처리하기 위해서는 그러한 자료구조나 계산구조

피보나치 트리에서 후위순회를 이용한 상호 연결망의 설계

를 사상하는 효율적인 방법이 필요하다. 이러한 사상을 임베딩(embedding)이라한다. 그래프 G 의 그래프 H 에 대한 임베딩 $f = (\varphi, \rho)$ 는 다음과 같이 정의한다. φ 는 G 의 노드 v 에서 H 의 노드 $\varphi(v)$ 로의 함수이고, ρ 는 G 의 에지 $e = (v, w)$ 에서 $\varphi(v)$ 와 $\varphi(w)$ 를 잇는 H 의 경로 $\rho(e)$ 로의 함수이다. 임베딩의 비용을 측정하는 척도로는 연장을(dilation) 밀집을(congestion), 확장율(expansion) 등이 있다. 후위순회 방법을 이용하여 번호매김을 한후 FP_3 의 임베딩의 예는 그림 2.4와 같다.

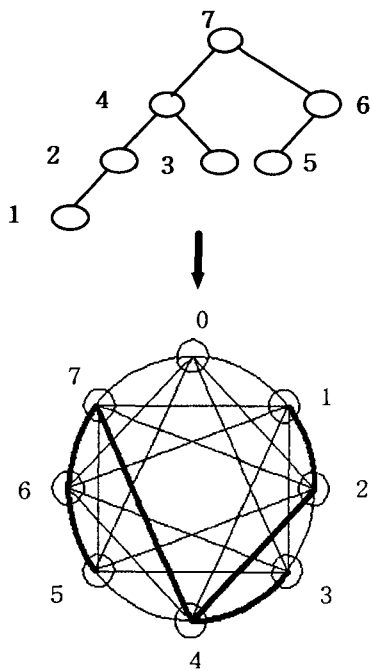


그림2.4 후위순회 번호매김을 한후의 FP_3 임베딩의 예 FP_3 그림에서 연장은 1가되고, 밀집율은 1이다. 확장율은 1이다. 임베딩 중에서 트리구조를 임베딩하는 문제는 특별한 중요성을 띄고 있다. 트리는 자료구조 또는 분할정복 알고리즘의 기반이 되거나 NP-complete 문제의 해 공간(solution space)을 제공하는 구조로서 널리 이용되고 있기 때문이다. 즉, 트리를 효율적으로 임베딩할 수 있다는 것은 트리 형태의 자료 구조를 쉽게 시뮬레이션 할 수 있으며 그 트리를 기반으로 하는 분할정복 알고리즘을 효율적으로 수행할 수 있음을 의미한다.

III. 라우팅 알고리즘과 지름

3.1 라우팅 알고리즘

라우팅 알고리즘에서 메시지가 전달되는 경로 상에 있는 각 노드들을 전달과정에서 새로운 원시노드로 간주되고 목적노드에 한 단계 가까운 이웃노드로 메시지를 전달하는데 탐욕기법을 사용하여 가장 큰 점프값을 가진 에지로 메시지를 전달함으로써 최단 거리 라우팅이 이루어진다.

알고리즘

step 1

임의의 두 노드 v, w 사이의 최단 경로 $dist(v, w)$ 를 구한다

step 2

만약 $dist(v, w) > \frac{1}{2}(f_{n+3})$ 이면

$$dist(v, w) = f_{n+3} - dist(v, w)$$

step 3

만약 임의의 노드가 같지 않으면

$dist(v, w) \geq MAX(f_2, f_3, \dots, f_n)$ 을 만족하는 f_i 를 구한다

step 4

v 를 $(v + f_i)$ 로 치환하고 step 3을 반복 실행

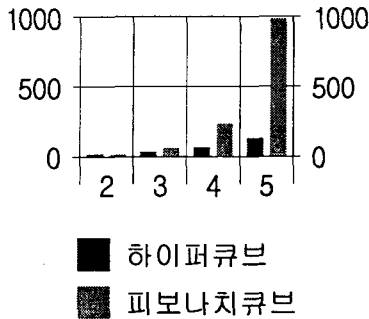
3.2 지름

$$\text{정리3.2.1 } dia(FP_n) = \lfloor \frac{k}{3} \rfloor, k = n + 3$$

3.3 하이퍼 큐브와 비교

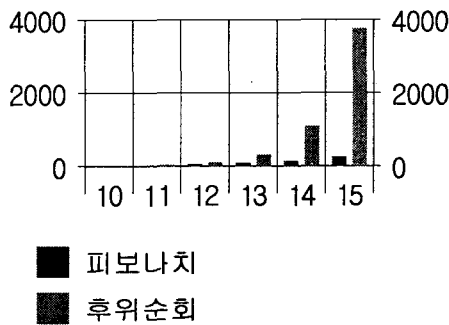
3.3.1 하이퍼 큐브와 후위순회의 노드수와 지름 비교 [표3.1]

지름	하이퍼큐브 노드 수	FP노드 수
2	4	21
3	8	89
4	16	377
5	32	1,597
6	64	6,765
7	128	28,657
8	256	121,393
9	512	514,229
10	1,024	2,178,309
11	2,048	9,227,465
12	4,096	39,088,169
13	8,192	165,580,141
14	16,384	701,408,733
15	32,768	2,971,215,073
16	65,536	12,586,269,025
17	131,072	53,316,291,173



3.3.2 전체노드와 분지 수 비교[표3.2]

상호 연결망 지름	하이퍼 큐브			FP_n		
	하이퍼큐브 의 분지수	분지수 /지름	전체노드수 /(분지수*지름)	FP의 분지수	분지수 /지름	전체노드수 /(분지수*지름)
2	2	1	1	1	5	1
3	3	1	1	16	5	2
4	4	1	1	22	5	4
5	5	1	1	28	5	11
6	6	1	2	34	5	33
7	7	1	3	40	5	102
8	8	1	4	46	5	330
9	9	1	6	52	5	1,099
10	10	1	10	58	5	3,756
11	11	1	17	64	5	13,107
12	12	1	28	70	5	46,534
13	13	1	48	76	5	167,591
14	14	1	84	82	5	610,983
15	15	1	146	88	5	2,250,921
16	16	1	256	94	5	8,368,530
17	17	1	454	100	5	31,362,524



IV. 결론

본 논문에서 다중 컴퓨터의 전체 성능과 시스템 확장성에 영향을 미치는 요소 중 하나인 프로세서를 연결하기 위해 지금까지 알려진 노드 방문 연산중 하나인

후위 순회방법으로 에지 번호 매김을 하여 만들어진 피보나치 수를 원형군의 점프열로 하는 새로운 상호 연결망을 설계하여 하이퍼 큐브와 노드 수가 같을 때 지름을 줄이고, 분지수와 지름을 동시에 고려하여 전체 노드 수를 줄이는 상호 연결망에 대해 연구하였다. 하이퍼 큐브의 전체노드 개수가 524,288개 일 때 하이퍼 큐브의 지름은 19가 되고, 본 논문에서 제안한 후위 순회 방법을 이용한 상호 연결망에서는 지름이 9일 때 514,229개의 노드 수를 가지고 있다. 이것은 하이퍼 큐브에 비하여 지름은 최대 2.1배정도 개선되었음을 보여주며, 또한 분지수와 지름을 다 고려했을 때에도 하이퍼 큐브는 각 노드가 가지는 분지수에 비해 짧은 지름을 가지지 못한다는 단점이 있어 모든 에지를 효율적으로 사용하지 못한 반면 FP_n 는 분지수에 비해 상대적으로 작은 지름을 갖고 있어 에지를 효율적으로 사용할 수 있다는 장점이 있다. 지름이 같은 조건하에서 하이퍼 큐브와 FP_n 의 분지수를 비교하였을 때 분지수 면에서 하이퍼 큐브가 $\theta(\log n)$ 이라면 FP_n 는 $\theta(5\log n)$ 이었다. 이는 거의 FP_n 이 $O(\log n)$ 과 같다는 것을 볼 수 있으며, 이는 분지수가 많아짐은 별 문제가 되지 않아 분지수와 지름을 동시에 고려했을 때 전체 노드 개수 면에서 비교해 본다면, 하이퍼 큐브보다 더 훌륭한 새로운 상호 연결망임을 증명할 수 있다.

참고문헌

- [1] 박정훈, "Circulant 그래프," 한국정보과학회지, 10권 6호, pp.28-36, 1992.
- [2] 한국과학기술원, 고성능 컴퓨터 설계 및 개발에 관한 연구, 1차년도 보고서, 1993.
- [3] B. W. Arden and H.Lee, Analysis of chordal ring network, IEEE Trans. Computer C-30(1981)291-295.
- [4] C. L. Seitz, "The cosmic cube," common. ACM. vol.28, pp. 22-23, 1985
- [5] F. Harary, "The maximum connectivity of a graph," Proc. Nat. Acad. Sci. U.S.A., Vol. 48, pp. 11142-1146, 1962.
- [6] Y.saad and M.H Schultz, "Topological properties of hypercubes," IEEE Trans. on computers, Vol 37, pp867-872, 1988.
- [7] Lim, Hyeong-Seok. "On the labeling of graphs and their applications," Ph.D.thesis, KAIST, 1993.
- [8] S.B.Akers and B.Krishnamurthy. On group graphs and their fault tolerance, IEEE Trans. Computer C-36 (1997)885-888.