

정이진트리를 피터슨-토러스(PT) 네트워크에 임베딩

서정현*, 이형욱, 장문석, 한순희

순천대학교 (컴퓨터과학과, 컴퓨터교육과, 컴퓨터과학과), 전남대학교 모바일 소프트웨어 전공
 {jhseo, oklee, jang}@suncheon.ac.kr, shhan@chonnam.ac.kr

Embedding Full Binary Trees into Petersen-Torus(PT) Networks

Junghyun Seo*, Hyeongok Lee, Moonsuk Jang, Soon-Hee Han

{Dept. Computer Sci., Dept. Computer Edu., Dept. Computer Sci.} Suncheon National Univ.,
 Dept. Mobile Software Chonnam National Univ.

요 약

본 논문은 정이진트리(full binary tree)를 PT 네트워크에 임베딩 가능성을 보인다. 메쉬 네트워크에 H-트리를 생성하는 방법을 응용하여 밀집을 1, 확장을 ≈ 5 그리고 연장을 $\frac{3(n+1)}{4} + 1$ 에 일대일 임베딩 하였다. 밀집을 1에 임베딩 함으로써 store-and-forward 라우팅 방식과 웜홀 라우팅 방식에 적합하며 일대일 임베딩 함으로써 프로세서의 부하를 줄였다.

1. 서론

멀티컴퓨팅을 위한 상호연결망은 크게 동적(dynamic)연결망과 정적(static)연결망으로 나눈다. 정적연결망은 두 프로세서 사이의 통신 링크가 수동적이므로 다른 프로세서와 다시 연결될 수 없으며, 동적연결망은 스위칭 요소에 의해 통신 링크가 다른 프로세서와 연결될 수 있다. 정적 연결망은 망을 구성하는 노드와 에지의 구성에 따라 트리 부류, 메쉬 부류, 하이퍼큐브 부류 그리고 스타그래프 부류로 나눌 수 있다 (1). 새로운 상호연결망이 설계되면 연결망의 구조에 적합한 병렬 알고리즘들이 설계되는데 그 연구비용은 적지 않다. 서로 다른 연결망간의 임베딩은 설계된 알고리즘들을 재사용할 수 있게 함으로써 알고리즘 설계 연구에 드는 비용을 절감할 수 있다. 임베딩에 관여되는 두 개의 연결망의 구조에 따라 임베딩 알고리즘 작성의 난이도가 결정된다. 멀티 컴퓨팅을 효과적으로 이용하기 위해서 링, 트리, 메쉬, 하이퍼큐브와 같은 공통적인 위상 구조를 토러스, 허니컴브(honeycomb) 메쉬, 하이퍼큐브, folded 하이퍼큐브, 스타 그래프, 매크로-스타(macro-star) 그래프 등의 여러 가지 연결망에 임베딩하는 문제가 연구 되었다.

이진트리는 분할-정복 알고리즘, backtrack searching, branch-and-bound computation, game-tree evaluation, functional and logical programming 그리고 다양한 수 연산에 사용되므로 다른 연결망으로 임베딩하

는 연구가 활발히 진행되었다. 링이나 이진트리를 허니컴브 메쉬, 하이퍼큐브, 스타그래프, 폴디드 피터슨 상호연결망에 임베딩하는 연구 [5,6,7,8]가 있다.

상호연결망은 멀티컴퓨팅 시스템에서 프로세서간의 연결 구조를 제공하고 그 위에 작성된 병렬알고리즘의 설계에 중요한 역할을 한다. 상호연결망은 그래프 $G=(V,E)$ 로 모델링 될 수 있고, 상호연결망의 프로세서는 노드집합 $V(G)$ 로 표현되고 프로세서간의 통신링크는 에지집합 $E(G)$ 로 표현된다. $|V(G)|$ 는 노드의 개수이다. 상호연결망 $H(host)$ 에서 $G(guest)$ 를 시뮬레이션 하기 위해, G 를 H 에 효과적으로 임베딩 f 한다는 것은 $V(G)$ 를 $V(H)$ 에 효과적으로 사상하는 것이다. 임베딩을 평가하는 척도는 확장을(expansion), 부하계수(load factor), 연장을(dilation), 밀집을(congestion)이 있다(3). 본 논문에서는 정이진트리(full binary tree)를 PT 네트워크로 일대일 임베딩 한다. 정이진 트리는 밀집을 1에 임베딩 하였고, 이것은 라우팅 거리에 영향을 받는 store-and-forward 라우팅 보다는 웜홀(worm hole) 라우팅 시스템에서 더 적합하다. 본 논문의 구성은 2장에서 PT 네트워크를 소개하고, H-트리를 메쉬에서 생성하는 알고리즘을 소개한다. 3장에서 정이진트리를 PT 네트워크에 임베딩 가능성을 보였다. 마지막으로 결론을 맺는다.

2. 관련연구

2.1 피터슨-토러스 $PT(m,n)$ 네트워크 정의

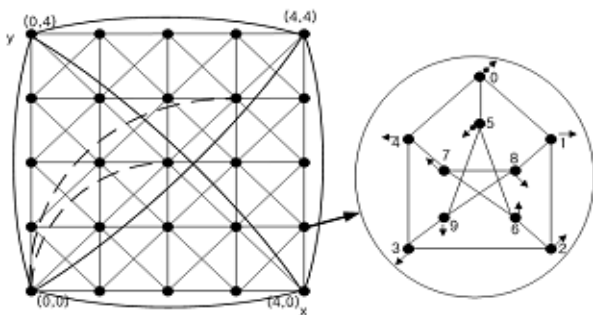
$PT(m,n)(m,n \geq 2)$ 은 피터슨 그래프(그림 1의 (b))를 기

* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-C1090-0801-0001)

본모듈로 하고, 기본모듈 $m(x\text{축}) \times n(y\text{축})$ 개를 격자점에 배치하고 에지 정의에 의해 연결한다. 피터슨-토러스 네트워크 $FT(m,n) = (V_{pt}, E_{pt})$ 이다. 본 논문에서는 $FT(m,n)$ 를 그림 1의 (a)와 같이 2차원 그래프에 사상시켜 설명한다. 단위 피터슨 그래프를 기본모듈이라고 하며, 기본모듈은 x 와 y 의 교차점에 있다. 기본모듈의 주소는 (x,y) 로 나타내고 노드 주소는 (x,y,p) 로 나타낸다. "노드 (x,y,p) 는 기본모듈 (x,y) 에 속한다"라고 말한다. x 는 기본모듈의 x 축의 좌표, y 는 기본모듈의 y 축의 좌표, p 는 기본모듈인 피터슨 그래프에 있는 노드의 주소이다. 피터슨-토러스 $FT(m,n)$ 의 노드정의는 다음과 같다.

$$V_{pt} = \{(x,y,p), 0 \leq x < m, 0 \leq y < n, 0 \leq p \leq 9\}$$

$FT(m,n)$ 의 에지는 다음과 같이 내부에지와 외부에지로 나눈다. 같은 기본모듈에 속한 노드들을 연결하는 에지를 내부에지라고 하고, 내부에지는 피터슨 그래프의 에지를 그대로 사용한다. 서로 다른 기본모듈에 있는 노드를 연결하는 에지를 외부에지라고 하고 다음과 같이 정의한다. 아래 에지를 나타내는 수식에서 삼별 '/'는 나머지 연산자이다. ① 세로에지는 $((x,y,6), (x,(y+1)/n,9))$ 과 $((x,y,9), (x,(y-1+n)/n,6))$ 이다. ② 가로에지는 $((x,y,1), ((x+1)/m,y,4))$ 과 $((x,y,4), ((x-1+m)/m,y,1))$ 이다. ③ 사선에지는 $((x,y,2), ((x+1)/m,(y+1)/n,3))$ 과 $((x,y,3), ((x-1+m)/m,(y-1+n)/n,2))$ 이다. ④ 역 사선에지는 $((x,y,7), ((x-1+m)/m,(y+1)/n,8))$ 과 $((x,y,8), ((x+1)/m,(y-1+n)/n,7))$ 이다. ⑤ 지름에지는 $((x + \lfloor \frac{m}{2} \rfloor)/m, (y + \lfloor \frac{n}{2} \rfloor)/n, 5)$ 과 $((x - \lfloor \frac{m}{2} \rfloor + m)/m, (y - \lfloor \frac{n}{2} \rfloor + n)/n, 5)$ 이다.



(a) $PT(5,5)$ (b) 피터슨 그래프
그림 1. 피터슨-토러스 $PT(5,5)$

그림 1의 (a)는 피터슨-토러스 $PT(5,5)$ 에서 기본모듈을 격자점으로 표현하였다. 기본모듈은 이웃한 기본모듈들과 완

전 그래프 형태로 연결되어 있다. 가장자리를 제외한 모든 기본모듈은 지름에지를 제외한 에지들이 그려져 있으며, 기본모듈 (0,0)의 지름에지만 굵은 파선(굵은 선이 일정한 규칙으로 반복되는)으로 그려져 있다. 가장자리의 기본모듈들은 랩 어라운드(wraparound) 에지가 생략되어 있으나 4개의 꼭지점에 있는 기본모듈은 몇 개의 랩어라운드 에지가 굵은 실선으로 그려져 있다. 피터슨-토러스 $FT(m,n)$ 의 기본모듈인 피터슨 그래프(그림 1의 (b))를 보면, 노드 1, 4는 가로에지, 노드 6, 9는 세로에지, 노드 2, 3은 사선에지, 노드 7, 8은 역 사선에지, 노드 0, 5는 지름에지와 결합된 노드들이다. $FT(m,n)$ 은 노드수 $10mn$, 에지 수 $20mn$, 연결도 4 그리고 분지수 4를 가지는 정규 그래프이다. FT에서 단순 라우팅 알고리즘, 최적 라우팅 알고리즘, 해밀톤 사이클 알고리즘이 제안되었고, 지름, bisection width가 분석되었다. $FT(m,n)$ 는 $m=n$ 일 경우 지름이 $1.5n+2$ 이고, bisection width가 n^2+5n 이다(9).

2.2 H-트리

출수 깊이의 경이진트리를 2차원 메쉬에 밀집을 1에 임베딩 하는 것은 잘 알려진 H-트리 생성을 이용해서 할 수 있다. H-트리를 만드는 자세한 설명은 [4]에 있고, H-트리를 만드는 방법을 간단하게 설명하면 다음과 같다. 깊이가 k 인 경이진트리를 T_k 라고 하고, n 행 m 열 2차원 메쉬를 $n \times m$ 메쉬라고 한다. k 가 4보다 큰 출수일 때, H-트리 T_k 는 2개의 T_{k-1} 트리의 루트노드를 자식노드로 하고 4개의 T_{k-2} 트리의 루트 노드를 단말 노드로 한다. 4개의 T_{k-2} 트리는 재귀적인 방법으로 남은 노드들을 연결하고, 루트 노드는 메쉬의 중앙 노드에 사상한다. 7×7 메쉬에 H-트리 T_5 를 생성하는 것을 그림 2에 보였고, 이것은 T_5 를 7×7 메쉬에 임베딩 한 것과 같다. T_k 는 2^k-1 개의 노드를 가지고 있으며, $(2^{\frac{k+1}{2}} - 1) \times (2^{\frac{k+1}{2}} - 1)$ 메쉬에 임베딩 된다.

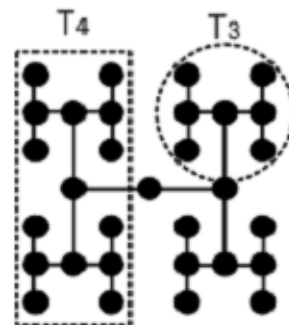


그림 2. H-트리 T_5

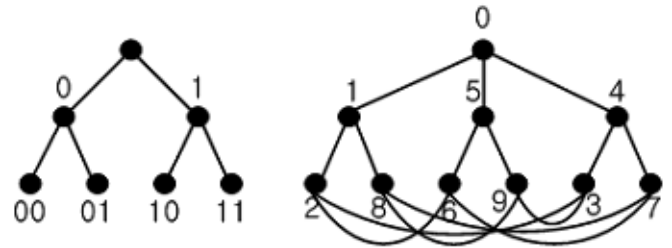
3. 정이진트리의 임베딩

그래프 G 의 그래프 H 에 대한 임베딩 f 는 다음과 같이 정의되는 함수의 쌍 (ϕ, ρ) 을 말한다. ϕ 는 G 의 정점 집합 $V(G)$ 를 H 의 정점 집합 $V(H)$ 에 대응시키는 함수이고, ρ 는 G 의 에지 $e=(v, w)$ 를 $\phi(v)$ 와 $\phi(w)$ 를 잇는 H 상의 경로와 대응시키는 함수이다. 확장율은 G 의 정점의 개수에 대한 H 의 정점의 개수의 비를 말한다. 확장율이 1보다 작은 경우 다대일 임베딩이라고 하고 G 에서 설계된 알고리즘을 H 에서 효율적으로 사용하기 어렵다. 확장율이 1과 같거나 큰 경우 일대일 임베딩이라 하고 확장율이 1보다 크면 G 에서 설계된 알고리즘이 H 에서 실행될 때 H 의 모든 노드를 이용하지 못하는 비효율이 발생한다. 확장율은 1에 근접할수록 좋다. 그래프 G 의 에지 e 의 연장은 H 상에서의 경로 $\rho(e)$ 의 길이를 말하고, 임베딩 f 의 연장은 G 의 모든 에지의 연장을 중 최대값이다. 연장의 최소값은 1이며, 연장이 크면 G 에서 설계된 알고리즘이 H 에서 적용될 때 연장을 만큼 메시지 전송 시간이 길어진다. 그래프 H 의 에지 e' 의 밀집율은 e' 에 포함되는 $\rho(e)$ 의 개수를 말하고, 임베딩 f 의 밀집율은 H 의 모든 에지의 밀집율 중 최대값이다. 밀집율의 최적은 1이며, 그 연장이 크면 전송 트래픽이 많이 발생한다[3].

정이진트리를 PT에 임베딩 하는 것은 PT에 정이진트리를 만드는 것으로 대체한다. PT에 정이진트리를 만드는 방법은 간단하다. 그림 1의 (a)는 PT의 기본모듈을 하나의 점으로 표현하였고, 이것은 PT가 토러스(메쉬)를 서브 그래프로 가진다는 것을 보인다. 그림 1의 (a)에 H-트리를 생성하고 H-트리의 단말노드에서 H-트리의 단말노드를 루트노드로 하는 깊이 3 정이진트리를 생성한다.

깊이 k 인 정이진트리 T_k 는 2^k-1 개의 노드와 2^k-2 개의 에지로 구성된다. 루트 노드의 깊이는 1이고 단말 노드의 깊이는 k 이다. 노드의 주소는 최대 $k-1$ 길이의 2진 스트링으로 구성되어있고, 루트 노드는 비어있다. 에지는 j 길이의 비트 스트링으로 이루어진 노드 x 와 $j+1$ 길이의 비트 스트링으로 이루어진 노드 xa 사이에 있다($a \in \{0, 1\}$, $0 \leq j < k-1$). 루트 노드는 깊이 1이고 노드의 주소 스트링의 길이가 j 인 노드의 깊이는 j 이다. 깊이 j 인 노드 x 가 있을 때, $x0$ 은 x 의 왼쪽 자식노드라고 하고 $x1$ 은 오른쪽 자식노드라고 하고, x 는 $x0$ 과 $x1$ 의 부모 노드라고 한다.

보조정리 1 깊이 3인 정이진트리(full binary tree) T_3 를 피터슨 그래프에 밀집율 1, 확장율 10/7 그리고 연장을 1에 임베딩 가능하다.



(a) 정이진트리 T_3 (b) 피터슨 그래프의 서브 그래프
그림 3. 정이진트리 T_3 를 피터슨 그래프에 임베딩

증명 그림 3의 (b)는 피터슨 그래프의 서브 그래프이고 T_3 인 그림 3의 (a)를 포함하고 있으므로 연장과 밀집율은 1이고 확장율은 10/7이다. T_3 의 루트노드를 피터슨 그래프의 노드 0에 사상하고, T_3 의 노드 0은 피터슨 그래프의 노드 1, 노드 00은 노드 2, 노드 01은 노드 8, 노드 1은 노드 4, 노드 10은 노드 3, 노드 11은 노드 7에 각각 사상한다. T_3 의 루트노드를 피터슨 그래프의 임의의 노드에 사상하더라도 피터슨 그래프는 노드-에지 대칭이므로 피터슨 그래프에 임베딩 가능하다. □

정리 1 정이진트리(full binary tree) $T_{2\log_2(n+1)+1}$ 를 $FT(n, n)$ 에 밀집율 1, 확장율 $=5$ 그리고 연장을 $\frac{3(n+1)}{4} + 1$ 에 임베딩 가능하다($n=2^a-1$, a 는 1보다 큰 자연수).

증명 PT에 H-트리를 만드는 과정은 2.2절에 소개된 방법과 같고 자세히 설명하면 다음과 같다. $r = \frac{n-1}{2}$ 라고 하고, PT의 노드 $(r, r, 0)$ 을 루트노드로 한다. 왼쪽 자식노드는 $((r-1)/2, r, 5)$ 이고 오른쪽 자식노드는 $(n-(r-1)/2, r, 5)$ 이다. 루트노드의 왼쪽 자식노드의 왼쪽 자식노드는 $((r-1)/2, n-(r-1)/2, 0)$ 이고 오른쪽 자식노드는 $((r-1)/2, (r-1)/2, 0)$ 이다. 루트노드의 오른쪽 자식노드의 왼쪽 자식노드는 $(n-(r-1)/2, (r-1)/2, 0)$ 이고 오른쪽 자식노드는 $(n-(r-1)/2, n-(r-1)/2, 0)$ 이다. 이것은 정이진트리 T_3 을 만든다. 4개의 단말 노드에서 T_3 을 만드는 방법과 동일하게 여분의 노드가 없어질 때 까지 재귀적으로 정이진트리를 만든다. 마지막으로 모든 H-트리의 단말노드가 속한 그림 5의 (b)와 같은 PT의 기본모듈에서 노드 0을 루트노드로 하는 그림 5의 (a)와 같은 T_3 을 만든다. 그림 6은 $PT(3, 3)$ 에 노드 $(1, 1, 0)$ 을 루트노드로 한 정이진트리 T_3 를 생성하였다. 크게 표시된 노드가 정이진트리의 노드들이고 큰 이진 비트 스트링이 노드 주소이다.

PT에 밀집율 1에 H-트리가 생성되고, 보조정리 1에 의해

H-트리의 단말노드에서 밑집을 1에 T_3 가 생성되므로 밑집을 1이다. 보조정리 1에 의해 H-트리의 단말노드에서 연장을 1에 T_3 가 생성되고, FT에 H-트리를 생성할 때 노드의 깊이에 따라 부모노드와 자식노드사이의 경로길이는 다르다. H-트리의 루트노드와 루트노드의 자식노드 사이의 경로가 가장 길고 그 값이 연장율이다. 루트노드는 $(r, r, 0)$ 에 사상되고 왼쪽 자식노드는 $((r-1)/2, r, 5)$, 오른쪽 자식노드는 $(n-(r-1)/2, r, 5)$ 에 사상된다. 루트노드와 2개의 자식노드 사이의 외부경로길이는 $r - ((r-1)/2) (= \frac{n+1}{4})$ 이고, 내부 경로길이는 루트노드가 있는 기본모듈에서 1이고, 자식노드가 있는 기본모듈에서 2이고, 루트노드와 자식노드의 사이에 있는 $\frac{n+1}{4} - 1$ 개의 기본모듈에서 각각 2이다. 루트노드와 자식노드 사이의 경로길이인 연장율은 $\frac{n+1}{4} + (\frac{n+1}{4} - 1) \times 2 + 3$ 이다. 이 식을 정리하면 $\frac{3(n+1)}{4} + 1$ 이다. 정이진트리 $T_{2\log_2(n+1)+1}$ 의 노드수는 $2(n+1)^2 - 1$ 이고 $FT(n, n)$ 의 노드수는 $10n^2$ 이므로 확장율은 $10n^2 / (2n^2 + 4n + 1)$ 이다. □

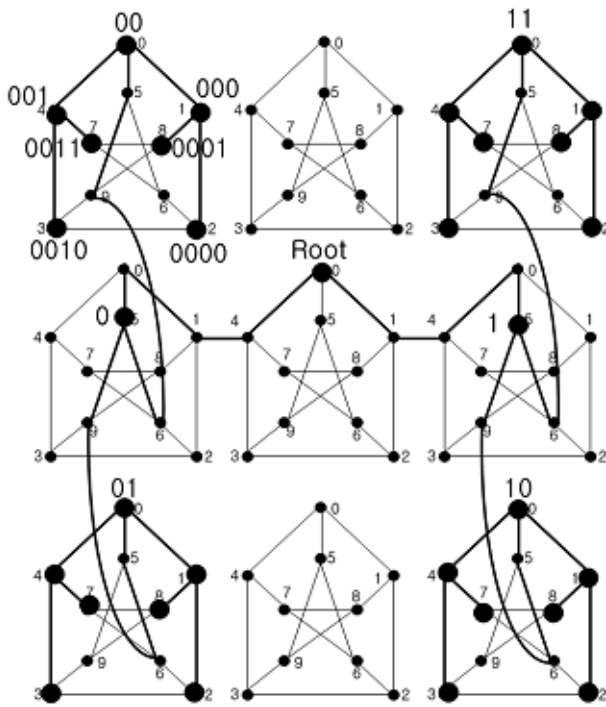


그림 4. $FT(3,3)$ 에 생성된 T_3

4. 결론

상호연결망 사이의 임베딩은 설계된 병렬알고리즘을 재사용할 수 있게 하는 의미 있는 작업이다. 본 논문에서 정이진트리 $T_{(2\log_2(n+1)+1)}$ 를 $FT(n, n)$ 에 밑집을 1, 확장율

$= 5$ 그리고 연장율 $\frac{3(n+1)}{4} + 1$ 에 임베딩 하였다. 역으로 FT 네트워크에서 개발된 알고리즘을 다른 연결망에서 사용할 수 있도록 FT 네트워크를 다른 연결망에 임베딩 하는 것은 더 연구해볼만한 가치가 있다.

5. 참고 문헌

- [1] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," IEEE Trans. Comput., Vol. 37, No. 7, pp. 867-872, July 1988.
- [2] S.K. Das and A.K. Banerjee. Hyper Petersen network: Yet another hypercube-like topology. In Proceedings of the 4th Symposium on the Frontiers of Massively Parallel Computation (Frontiers' 92), pages 270-277, McLean, Virginia, USA, October 1992. Computer Society Press.
- [3] S. Bettayeb and B. Cong and M. Girou and I. H. Sudborough, "Embedding Star Networks into Hypercubes", IEEE trans. comput., VOL. 45, No. 2, pp. 186-194, Feb. 1996.
- [4] J.D. Ullman, Computational Aspects of VLSI. Rockville, md.: Computer Science Press, 1984
- [5] S. Y. Hsieh, "Embedding of Cycles in the Faulty Hypercube", ACSAC2005, LNCS 3740, pp229-235, 2005.
- [6] P. F. Cortese and G. D. Battista, "On Embedding a Cycle in a Plane Graph", GD2005, LNCS 3843, pp. 49-60, 2005
- [7] S. Bezrukov, "Embedding complete trees into the Hypercube", Discrete Applied Mathematics, Vol. 100, No. 2-3, pp.101-119, 2001.
- [8] J. H. Park, H. S. Lim and H. C. Kim, "Embedding Starlike Trees into Hypercube-Like Interconnection Networks", ISPA 2006 Ws, LNCS 4331, pp 301-310, 2006.
- [9] 서경현, 이형욱, 장문석, "멀티컴퓨팅 시스템을 위한 피터슨-토러스(PT) 네트워크", 정보과학회 논문지 게재예정