論文94-31B-11-3

K개의 집합에 연결이 있는 네트에 K(K-1)/2의 비용을 주는 경우의 네트워크의 다중 분할

(A Multiple-Way Partitioning of a Network When the Cost of the Net Which Connects K Subsets is K(K-1)/2)

張 宇 徹*, 金 仁 基**, 金 敬 植***

(Woo-Choul Jang, In-Ki Kim and Kyung-Sik Kim)

要 約

본 논문에서는 네트워크의 노드들을 여러 집합으로 분할할 때 서로 다른 k개의 집합 사이를 연결하는 네트에 대해 비용을 k(k-1)/2 로 주는 경우, 즉 완전 그래프 형태의 비용을 주는 경우의 분할 알고리듬을 제시한다. 이 문제는 Sanchis의 다중 분할 문제[5]중의 하나로서 분산처리 시스템의 자원 할당에 적용될 수 있다. 제시된 알고리듬은 Fiduccia, Mattheyses 알고리듬[3]을 확장하여 일시에 다중 분할하는 기법을 사용하였으며 네트워크의 크기에 선형으로 비례하는 시간 복잡도와 공간 복잡도를 가진다. 제시된 알고리듬의 성능을 평가하기 위해 네트별로 노드들을 그룹화시키는 클러스터 성장 방식과 결과를 비교했다. 실험 결과 일시에 다중 분할하는 제시된 알고리듬이 우수한 결과를 내었다.

Abstract

In this paper, we propose an algorithm on partitioning a network into several subsets where the cost of a net which connects nodes in k subsets is given as k(k-1)/2 indicating the typical pattern of complete graphs. This problem is one of generalizations for multiple-way partitioning proposed by Sanchis.[5] Its solution can be applied to resource allocation problem in distributed systems. The proposed algorithm expanded the algorithm of Fiduccia and Mattheyses[3] to handle the multiple-way partitioning simultaneously. It has time and space complexity linear to the size of the network. To evaluate the performance of the proposed algorithm, we implemented also a traditional cluster growth method which groups connected nodes for nets, and compared experimental results with those of the proposed algorithm. The proposed algorithm shows some enhancement made.

* 正會員, 太平洋 SYSTEMS
(Pacific Systems)

** 正會員, 同和 ADVANTEST
(Dong-hwa Advantest)

*** 正會員, 湖西大學校 電子計算學科
(Dept. of Computer Science, Hoseo Univ.)

接受日字 : 1993年 12月 21日

Ⅰ. 서 론

네트워크(network)는 노드(node)들의 집합과 그 노드들을 연결하는 네트(net)들의 집합으로 구성된다. 하나의 네트는 2개 이상의 노드에 연결될 수 있으며, 그래프(graph)는 네트가 2개의 노드에만 연결된 에지(edge)인 네트워크의 특수한 경우이다.

네트워크의 분할 문제는 노드들을 몇개의 집합으로 나눌 때 서로 다른 집합의 노드들에 연결된 네트의 수가 최소가 되도록 하는 것이다. 이러한 네트워크의 분할 문제는 NP-complete이다.[1] 그러나 그 응용이 광범위하여 다항식 시간(polynomial time) 내에 최적해에 가까운 해를 얻기 위해 많은 휴리스틱(heuristic)알고리듬들이 연구되었다.[2,3,4,5,6,7]

네트워크를 2개의 집합으로 분할할 때 분할 문제는 양쪽 집합의 노드를 연결하는 네트수를 최소화하는 것이었다.[2,3] Sanchis는 네트워크를 2개의 집합으로 분할하는 것에서 n개의 집합으로 분할하는 것으로 확장할 때 네트에 비용(cost)을 정하는 방식에 따라 이 최소화의 문제가 다음의 3가지 문제로 일반화된다고 제시하였다.[5] 첫째는 서로 다른 집합 내의 노드들을 연결하는 네트에 비용 1을 주어 네트의 총비용을 최소화하는 문제이다. 그 응용으로는 고정된 몇개의 프로세서(집합)에 병렬 프로그램(노드)을 할당할 때, 프로그램간의 데이타 공유(네트)를 위한 공용 메모리의 크기를 가능한한 줄이도록 프로그램을 할당하는데 이용될 수 있다. Sanchis는 네트워크의 크기에 선형으로 비례하는 시간 복잡도의 알고리듬을 제안하였다.[5] 둘째는 k개의 집합내의 노드에 연결이 있는 네트에 k-1의 비용을 주는 방식이다. 그 응용으로는 VLSI 레이아웃의 배치를 위한 분할이 될 수 있다. 이 문제에 대해서는 네트워크의 크기에 선형으로 비례하는 시간 복잡도의 알고리듬이 제안되었다.[6] 세째는 k개의 집합내의 노드에 연결이 있는 네트에 k(k-1)/2의 비용을 주는 방식이다. 그 응용으로는 분산 시스템에서 각 프로세서 간의 통신에 공통의 비용을 부가하는 것이 바람직할 때 적용하기에 적합하다.

본 연구에서 다루는 다중분할은 Sanchis의 세째 최소화 문제이다. 이 문제에 대한 기존 해결책은 클러스터 성장(cluster growth) 방식[10]이 있다. 클러스터 성장방식은 분할의 집합 크기 차이나 interface 제한 등 분할의 조건이 까다로울 때 적용이 용이한 기존 방식으로서 비용함수가 특이한 본 연구의 문제에도 적용이 용이한 알고리듬이다. 그러나 클러스터 성장 방식은 씨드(seed) 선정 및 같은 비용의 노드중 선택에 따라 결과가 크게 차이나는 단점이 있다. 본 연구에서는 Fiduccia, Mattheses의 2중분할 알고리듬[3]을 확장하여 일시에 n개의 집합으로 분할하는 일시 다중 분할 기법[6]을 활용하여 네트워크의 크기에 선형으로 비례하는 시간 복잡도와 공간 복잡도를 가진 알고리듬을 제안하였다.

II장에서는 분할 모형으로서 분할 모델과 본 연구의 착안 사항을 기술하고, III장에서는 본 연구에서 제시하는 일시다중 분할 방법을 기술한다. IV장에서는 제시하는 알고리듬을 흐름도로 보이며 시간 복잡도 및 공간 복잡도를 분석하고 본 알고리듬의 성능을 평가하기 위해 클러스터 성장 방식과 실험 결과를 비교하였다. V장에서 결론을 맺는다.

Ⅱ. 분할 모형

1. 분할 모델

분산 시스템의 예에서 볼 때 한 프로그램이 k개의 프로세서에 분산되어 수행된다면 k개의 프로세서는 서로 긴밀한 통신을 해야할 것이다. 즉 각각의 프로세서 끼리는 통신 경로를 확보해야 하므로 k개의 프로세서들은 노드 k개의 완전 그래프 형태로 통신 경로를 갖는다고 볼 수 있다. 본 연구에서는 이러한 경우의 경로비용을 완전 그래프의 연결선 수인 k(k-1)/2로 보았다.

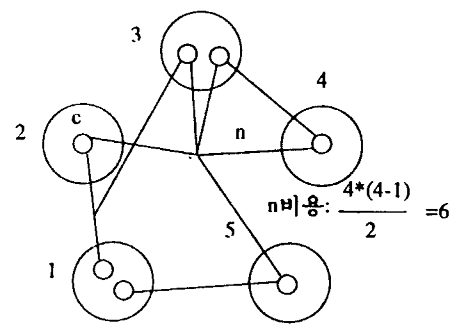

그림 1. 분할 모형(네트 n의 경우 : 4개 지역에 5개의 프로세스가 분산됨)

Fig. 1. Partitioning model.(case of net n: 5 processes are partitioned in 4 groups)

한 프로그램은 여러 프로세스(process)로 나뉠 수 있다. 여러 프로그램의 프로세스들을 k개의 프로세서(집합)에 분산 시켜야 할 때 프로세스들을 어떻게 분할하여 각 프로세서에 할당할 것인가 하는 문제는 한 연결선(프로그램)이 여러개의 노드(프로세스)를 연결

하는 다단자 네트를 가진 네트워크를 k개의 집합으로 나눌 때 서로 다른 집합을 연결하는 네트의 연결 비용을 최소화하도록 분할하는 문제로 해석할 수 있다.

2. 착안사항

본 일시 다중 분할을 구상하게 된 것은 Fiduccia. Mattheyses의 선형 시간에 2중 분할을 수행하는 기법[3]에서 착안하였다. 그 기법은 단일 노드 이동의 반복을 통해 분할을 개선한 것과. 비용 함수의 관리를 위한 자료 구조로서 버킷(bucket)을 사용한 것이다. 버킷은 일종의 어레이(array)로서. 분류(sorting)의 시간을 줄이기 위하여 일련의 기억 장소를 잡아 놓은 것이다.

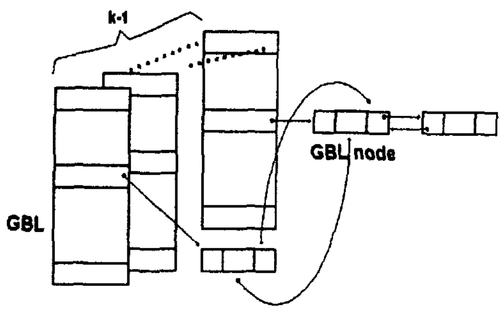

그림 2. 각 집합의 이득버킷리스트(GBL) 구조

Fig. 2. Gain Bucket List (GBL) structure for each subset.

단계 1. 네트워크의 입력
단계 2. 초기 분할
단계 3. 각 노드의 가능한 이동에 대한 이득 계산. 이득 버킷 리스트의 구축
단계 4. 이득 버킷 리스트를 이용하여 크기 균형을 만족하는 최대 이득의 이동 선택. 선택 대상 없으면 종료
단계 5. 선택된 노드의 이동
단계 6. 이득 버킷 리스트의 수정.
　· 단계 4 로 이동

그림 3. 일시 다중 분할의 주요 단계들

Fig. 3. Primary steps of simultaneous multiple-way partitioning.

본 연구에서는 이러한 Fiduccia. Mattheyses의 기법을 확장하여 한 노드의 이동을 k개의 집합 중 임의의 방향으로 가능하게 하였다. 한 노드가 이동해서 들어갈 수 있는 집합은 k-1개가 있으므로. 이들 각 이동 방향에 대해 버킷을 하나씩 두었다 (그림 2 참고). 이것을 이득 버킷 리스트(Gain Bucket List. 이하에서 GBL로 표기함)라 한다. 각 이동에 따라 이득을 계산하고 그 이득의 크기 대로 해당 이동 방향의 버킷에 이동에 대한 정보를 담은 노드를 연결시킨다. 이 노드를 GBL노드라 한다. 그림 2에 각 집합마다 가지는 GBL 구조를 보인다. 그림 2에서 같은 노드로부터 만들어지는 GBL 노드들은 link로 연결되어 있어 update시 수정이 용이하다.

일시 다중 분할의 방법은 어느 순간에 분할의 크기 균형을 만족하면서 최대의 이득을 내는 노드의 이동을 수행해 나가는 것이다. 이 방법은 각 버킷을 이용하여 한 이동 노드를 선택해서 그 이동을 수행하고. 이동에 따른 다른 노드의 변경되는 이득을 갱신하여 다시 버킷에 반영시키는 일련의 작업을 반복하는 것이다. 그림 3에 일시 다중 분할의 주요 단계들을 보인다.[6]

Ⅲ. 일시 다중 분할 방법

1. 비용함수

그림 3의 단계 4-6의 과정을 분할 개선(partition improvement)의 과정이라 볼 수 있다. 분할 개선시 어느 노드를 어느 집합으로 이동할 것인가를 결정하기 위해 비용 함수(cost function)를 설정한다. 비용함수는 각 네트에 대해 정의되는데 그 네트가 연결하는 노드들이 k개의 집합에 분산되어 있을 때 완전 그래프의 에지수와 같은 $k(k-1)/2$ 값을 준다. 그리고 한 노드를 이동할 때 비용 함수가 줄어드는 정도를 이득(gain)이라 부른다.

노드 C가 집합 i에 있을 때 노드 C에 연결된 한 네트 n에 대한 비용을 cost(n. C. i)라 하고 노드 C를 집합 j로 이동했을 때 네트 n에 대한 비용을 cost(n. C. j)라 하면. 그 때의 이득 gain(C. i. j. n)은

$$\begin{aligned} \text{gain}(C,\ i,\ j,\ n) &= \text{이동전 cost} - \text{이동후 cost} \\ &= \text{cost}(n,\ C,\ i) - \text{cost}(n,\ C,\ j) \end{aligned}$$

그림 1의 예에서 노드 C를 집합 2에서 5로 옮기는 경우 :

$\text{cost}(n,\ C,\ 2) = 4(4-1)/2 = 6$

$\text{cost}(n,\ C,\ 5) = 3(3-1)/2 = 3$

$\text{gain}(C,\ 2,\ 5,\ n) = \text{cost}(n,\ C,\ 2) - \text{cost}(n,\ C,\ 5) = 3$

즉 이득은 3이 발생한다.

정리 1. 이득 G는 한 노드에 최대로 연결된 네트 수를 m 이라 할 때

$$-m(k-1) \le G \le m(k-1)$$

의 범위를 가진다.

[증명] 임의의 노드 C를 i에서 j로 이동할 때 한 네

트 n 에 대해 연결하고 있는 노드의 집 합 수가 k_1개에서 k_2개로 변경된다면

$$gain(C,i,j,n) = \frac{k_1(k_1-1)}{2} - \frac{k_2(k_2-1)}{2}$$
$$= \frac{1}{2}(k_1-k_2)(k_1+k_2-1)$$

이때 $0 \le k_1,\ k_2 \le k$이고, $|k_1-k_2| = 1$ 이므로

$$-(k-1) \le gain(C,i,j,n) \le k-1$$

따라서 최대 m개의 네트에 대한 이득은

$$-m(k-1) \le G \le m(k-1)$$ [증명끝]

정리 1로부터 한 버킷의 크기는 $O(mk)$임을 알 수 있다.

2. 크기 균형 조건과 이동할 노드의 선택

크기 균형 조건이란 분할된 집합들이 서로 비슷한 크기를 가져야 하는 조건을 말한다. 네트워크를 입력할 때 각 노드의 크기를 함께 입력하여 모든 노드의 크기의 합 W를 계산하면, 분할되는 집합의 평균크기는 W/k이다. 그러나 노드 크기는 서로 차이가 있어서 (최대크기를 maxW, 최소크기를 minW로 표시) 분할된 집합의 크기가 꼭 W/k가 되지는 않는다. 서로 같은 크기로 맞추어졌다 하더라도 임의의 노드 한 개를 이동할 수 있도록 하기위해 본 논문에서는 다음과 같은 조건이면 "집합 Si가 균형 상태에 있다"라고 간주한다.[3]

$$W/k - maxW \le S_i \le W/k + maxW$$
(여기서 maxW는 노드중 최대 크기)

어느 노드를 어디로 이동할 것인가는 다음과 같은 절차에 의해 결정된다.

〈 이동할 노드와 노드의 이동 방향 선택 절차 〉

단계 1 : 각 이득 버킷 리스트에서 이득이 가장 큰 첫 GBL 노드를 뽑아 크기 균형 조건을 만 족하는지 검사한다. 크기 균형이 만족되지 않으면 그 GBL 노드는 대상에서 제외시킨다.

단계 2 : 단계 1에서 뽑힌 GBL 노드들 중에서 최대의 이득인 GBL 노드를 선택한다.

3. 이동에 따른 갱신

한 노드가 이동함에 따라 그 노드에 직접 연결된 다른 노드들의 이득이 변경된다. 본 절에서는 이동한 노드에 연결된 다른 노드들의 이득 갱신 및 이득 버킷 리스트 상에서의 그들의 GBL 노드 위치 갱신에 대해 기술한다. 이러한 갱신을 위해 다음과 같은 절차를 따른다.

〈 이동할 노드에 연결된 다른 노드들의 이득 갱신 절차 〉

단계 1 : 갱신하고자 하는 노드의 GBL 노드들을 이득 버킷 리스트 상에서 제거한다.

단계 2 : 노드의 이득을 다시 계산하여 GBL 노드를 생성해서 이득 버킷 리스트 상에 덧붙인다.

Ⅳ. 실험 및 고찰

1. 제시된 알고리듬

Ⅱ장의 일시 다중 분할의 주요 단계에 대해 Ⅲ장에서 기술했던 방법들을 이용하여 각 네트의 비용함수의 총합을 최소화하는 알고리듬을 설계하였다. 그림 4에 본 알고리듬의 흐름도를 보인다. 그림 4에서 단계 3부터 단계 10까지의 루우프(loop)는 모든 노드들이 한번씩 이동하는 과정이다.[2] 이 과정을 분할 개선 1 패스라 한다. 이 과정의 시간 복잡도와 공간 복잡도를 정리 2, 3에 보이며 제시된 알고리듬이 네트워크의 크기라 할 수 있는 총핀수인 p에 선형으로 비례하는 것을 밝힌다.

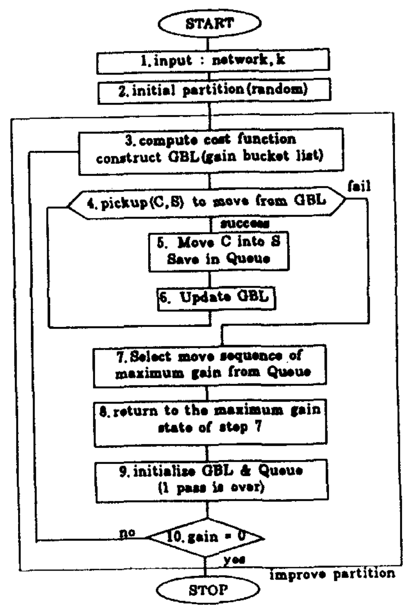

그림 4. 제시된 알고리듬의 흐름도

Fig. 4. The flowchart of the proposed algorithm.

정리 2. 제시된 분할 알고리듬의 분할 개선 1 패스의 시간 복잡도는 $O(pk^2m^2t)$ 이다. (여

기서 p는 네트워크의 총핀수, k는 집합수, m은 한 노드에 최대로 연결된 네트수, t는 한 네트에 최대로 연결된 터미널(노드)수임).

[증명] 그림 4의 각 단계에 대해 기술한다.

단계 3 : 매 이동마다 이득을 계산하고 이득 버킷 리스트를 구성하는 부분으로서 노드는 O(p)개 존재하고 각 노드에 대해서 이득을 계산하여 GBL노드를 생성하는 데에 $O(k^2m)$의 시간복잡도를 가지므로 제시된 알고리듬의 이득 버킷 리스트를 구축하는 과정은 $O(pk^2m)$의 시간 복잡도를 갖는다.

단계 4 : 이동할 노드와 집합을 선택하는 부분으로서 각 이동 방향별 대상 GBL 노드 선택은 대상 GBL 노드 수가 k(k-1)이 있으므로 $O(k^2)$시간에 수행되고 최대의 이득을 내 는 이동의 선택은 $O(k^2)$개의 대상에 대해 일어나므로 $O(k^2)$시간에 가능하다.

단계 5 : 노드를 이동시키고 이동 정보를 큐에 저장하는 부분으로서 GBL로부터 이동한 노드의 GBL 노드를 제거하는데는 한 노드의 GBL 노드수가 k-1 개이므로 O(k)시간에 가능하다. 이동한 노드는 고정되므로 제시된 알고리듬의 노드 이동 반복 과정은 전체 노드수 만큼 반복될 것이다. 따라서 단계 4,5의 전체적인 시간 복잡도는 $O(pk^2)$이다.

단계 6 : 갱신하는 노드의 갯수는 O(mt)이다. 새로운 이득을 가진 노드에 대해 새로운 GBL 노드를 만드는데는 $O(k^2m)$의 시간 복잡도를 갖는다. 따라서 단계 6은 전체적 인 노드 이동 반복 과정에서 $O(pk^2m^2t)$의 시간 복잡도를 갖는다.

단계 7,8 : 이동된 노드의 복잡도를 따르므로 시간 복잡도가 O(p)이다.

단계 9 : 이득 버킷 리스트에 남아 있는 노드들을 제거하고 큐를 초기화 하는 부분으로서 GBL의 최대 노드수가 O(pk)이므로 시간 복잡도는 O(pk)이다.

그러므로 제시된 알고리듬의 분할 개선 부분의 1 패스는 $O(pk^2m^2t)$의 시간 복잡도를 갖는다.[증명끝]

정리 3. 제시된 알고리듬의 분할 개선 1 패스의 공간 복잡도는 $O(pk+mk^3)$ 이다.

[증명]

한 이득 버킷 리스트는 정리 1에 의해 -m(k-1)로부터 m(k-1)까지의 크기를 가지며 모두 k(k-1)개의 이득 버킷 리스트가 있으므로 이득버킷리스트의 공간 복잡도는 $O(mk^3)$이다. 이동 노드는 GBL 노드수가 최대 O(pk)개 생성되므로 그 공간 복잡도는 O (pk)이다. 큐의 공간 복잡도는 O(p)이다. 그러므로 제시된 분할 알고리듬의 분할 개선 1 패스의 공간 복잡도는 $O(pk+mk^3)$ 이다. [증명끝]

2. 실험결과

제시된 알고리듬은 C 언어로 구현되었고 MIPS시스템(EXL7340)에서 실행되었다. 표 1에 실험에 사용한 네트워크의 노드수, 네트수, p, m, t를 보인다.[8] 이 네트워크들은 VLSI 표준셀과 게이트 어레이의 벤치마크(benchmark) 데이터로서 분할 문제에 대한 데이터로도 많이 사용되고 있다.[7]

표 1. 다중 분할 실험에 사용한 네트워크

Table 1. Networks tested in experiments of multiple-way partitioning.

네트워크	노드수	네트수	p	m	t	maxW	minW
decin	186	150	558	5	37	100	30
regfile	356	196	676	3	11	140	100
primary1	752	831	2738	9	18	200	30
primary2	2907	2961	10971	9	37	200	30
test1	63	83	126	3	6	100	100

(m : 노드당 최대 네트수, t : 네트당 최대 노드수, p : 네트워크의 총핀수, maxW : 노드의 최대 크기, minW : 노드의 최소 크기)

실험은 초기 분할로부터 제시한 알고리듬을 따라 다중 분할 결과를 얻는 것으로서 초기분할의 영향을 피하기 위해 랜덤(random) 초기 분할로 100회의 결과를 얻어, 그 결과의 최소치, 최대치, 평균과 표준편차를 구했다.[9] 제시된 알고리듬의 성능을 비교하기 위해 기존 방식으로서 클러스터 성장 방식[10]을 구현하였다. 구현한 클러스터 성장 방식은 네트별로 연결된 모든 노드들을 가능한한 한 집합에 넣는 것을 목표로 하였다. 그 이유는 분산된 집합수 k를 최소화해야 네트비용 k(k-1)/2를 최소화할 수 있기 때문이

다. 씨드(seed)의 영향을 피하기 위해 씨드 네트를 랜덤으로 선택하여 100회의 결과를 얻어, 결과의 최소치, 최대치, 평균과 표준편차를 구했다.

표 2의 실험 결과에 제안한 알고리듬에 의한 결과 100개의 분포와 기존 클러스터 성장방식에 의한 결과 100개의 결과의 분포를 비교하여 보이고 있다. 100회 중에 가장 최소의 총비용이 최선의 결과로서 제안한 알고리듬이 결과가 우수함을 볼 수 있다. 100회에 대한 결과의 분포도 제안한 알고리듬이 우수한 평균과 표준편차를 보이고 있다.

표 2. 실험 결과

Table 2. Experimental results.

네트워크	분할수	총비용(완전그래프의 형태)							
		최소		최대		평균		표준편차	
		Proposed	Clustering	Proposed	Clustering	Proposed	Clustering	Proposed	Clustering
decin	2	16	53	33	84	25.4	68.6	3.4	7.2
	3	37	98	65	143	51.9	121.9	6.1	11.7
	4	60	112	101	197	79.6	161.3	9.0	21.8
	5	72	136	163	212	114.5	186.2	14.2	24.8
regfile	2	29	40	36	56	32.9	48.1	1.4	4.1
	3	51	116	65	121	58.5	120.4	2.8	1.2
	4	62	130	93	156	80.3	131.2	5.7	4.8
	5	69	186	121	191	103.9	188.5	6.4	1.3
primary1	2	75	163	271	388	118.7	233.4	21.7	37.3
	3	176	256	327	553	240.5	406.8	29.5	63.6
	4	256	400	489	674	382.1	530.3	41.7	78.8
	5	402	500	713	778	542.2	683.1	60.6	72.1
primary2	2	348	516	606	1117	406.2	710.0	38.2	140.9
	3	774	936	1191	1826	970.9	1286.7	80.8	213.5
	4	1210	1456	1763	2206	1522.8	1823.5	112.6	187.7
	5	1786	1879	2411	2586	2072.9	2204.9	132.4	150.2
test1	2	4	7	15	24	7.6	16.6	3.6	4.0
	3	11	17	24	38	17.1	30.3	3.3	6.2
	4	15	26	33	60	22.5	44.0	3.2	7.9
	5	19	37	34	70	28.4	54.0	3.4	8.6

본 실험에서 일시에 다중 분할하는 제안한 알고리듬이 본 문제에 직접적인 해결 방안이 되며 기존 클러스터 성장방식보다 결과도 우수함을 알 수 있다. 한편 본 실험 결과로부터 랜덤 초기 분할에 따라 산출되는 총 비용이 평균에서 큰 폭으로 변화하고 있음[4]을 확인할 수 있다.

V. 결 론

본 논문에서는 네트워크의 노드들을 n개의 집합으로 분할할 때 서로다른 k개의 집합내의 노드들을 연결하는 네트에 대해 k(k-1)/2의 비용을 주는 경우 즉, 네트비용이 완전 그래프 형태일 때 네트워크의 다중 분할 알고리듬을 제안하여 기술하였다. 이 문제는 Sanchis가 제한했던 세번째 다중 분할 문제[5]로서 제안된 알고리듬은 Fiduccia와 Mattheyses의 알고리듬[3]을 확장하여 일시에 다중 분할하는 것으로서 네트워크의 크기에 선형으로 비례하는 시간 복잡도와 공간 복잡도를 가진다.

제시된 알고리듬의 성능 평가를 위하여 비교한 기존 방식으로 네트별로 노드들을 그룹화시키는 클러스터 성장 방식을 사용하였다. 초기 분할의 문제점을 극복하기 위해 제시된 알고리듬을 100회 반복하였고 클러스터 성장 방식의 씨드 설정 문제를 극복하기 위해 임의의 씨드로 100회 반복하여 그 결과를 비교했다. 실험 결과 일시에 다중 분할하는 제시된 알고리듬이 문제에 직접적인 해결 방안이 되며 결과도 우수했다.

제안된 알고리듬은 분산 처리 시스템의 여러 자원들을 여러 시스템에서 활용코자 할 때 자원들간의 통신을 극소화 하여 효율적으로 자원들을 활용할 수 있도록 자원들을 시스템에 할당하는데 적용될 수 있어 앞으로의 연구에서 시도할 예정이다.

參考文獻

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability*. San Francisco, CA: Freeman, pp. 209-210, 1979.

[2] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", *BSTJ.*, pp. 291-307, Feb. 1970.

[3] C. M. Fiduccia and R. M. Matteyses, "A Linear-time Heuristic for improving Network Partitions", *Proc. 19th. DAC.*, pp.175-181, 1982.

[4] B. Krishnamurthy, "An Improved Min-Cut Algorithm For Partitioning VLSI Networks", *IEEE Trans. Comput.*, Vol.C-33, No.5, pp.438-446, May 1984.

[5] L. A. Sanchis, "Multiple-Way Network Partitioning", *IEEE Trans. on Computers*, Vol.38, No.1, pp.62-81, Jan. 1989.

[6] 김 경식, 초기 배치를 위한 VLSI 네트워크의 일시 다중 분할, 박사학위 논문, *서울대학교 컴퓨터 공학과*, 1990.2

[7] Chan-Ik Park, and Yun-Bo Park, "An Efficient Algorithm for VLSI Network Partitioning Problem Using a Cost Function with Balancing Factor", *IEEE Trans.* CAD, Vol.12, No.11, pp. 1686-1694, Nov. 1993.

[8] Benchmark data, IEEE/ACM SIGDA Physical Design Workshop: Placement and Floorplanning (Standard Cell &

Gate Array Benchmark data). Hilton Head. SC. Apr. 27-29. 1987.

[9] T.W. Ng , J. Oldfield and V. Pitchumani. "Improvements of a Mincut Partition Algorithm". *ICCAD '87*, pp. 470-473. 1987.

[10] M. Hanan, and J. M. Kurtzberg. :Placement Techiques", in *Design Automation of Digital Systems. Vol. 1 : Theory and Techniques.* M. A. Breuer. ed., Englewood Cliffs, New Jersey : Prentice-Hall. Inc., Chapter 5, pp. 213-282. 1972.

※ 이 논문은 1992년도 교육부 지원 한국학술진흥재단의 자유공모(지방대육성) 과제 학술 연구조성비에 의해 연구되었음.

著 者 紹 介

張 宇 徹(正會員)
1994年 호서대학교 전자계산학과 졸업. 1994年 ~ 현재 태평양 시스템 근무

金 仁 基(正會員)
1990年 호서대학교 전자계산학과 졸업. 1993年 호서대학교 대학원 전자계산학과 졸업. 1990年 ~ 현재 동화어드벤트 근무

金 敬 植(正會員)
1982年 서울대학교 전자계산기공학과 졸업(공학사). 1984年 서울대학교 대학원 전자계산기공학과 졸업(공학석사). 1990年 서울대학교 대학원 컴퓨터공학과 졸업(공학박사). 1984年 ~ 1991年 한국전자통신연구소 선임연구원. 1991年 ~ 현재 호서대학교 전자계산학과 조교수. 연구분야는 VLSI/CAD, 논리분할, 배치, VHDL 등임.