

# 셀 분할 병렬 계층 상향 군집화 기반의 효과적인 데이터 분할 기법

서 병 석\*

## An Efficient Data Partitioning Method Based Cell Partitioning Parallel Hierarchical Agglomerative Clustering

Byung-suk Seo\*

요 약

오늘날 CPU 기반 병렬 및 분산처리 컴퓨팅 분야의 성장 속에서 범용 그래픽 처리 장치(GPGPU)는 계산 속도를 몇 배로 높이는 엄청난 성과를 보여주었다. 데이터 분할은 다양한 도메인을 갖는 데이터 마이닝 분야의 응용 프로그램에서 매우 중요한 작업이다. 또한, 계층 상향 군집화도 군집화 계층을 이용하여 클러스터의 수와 패턴을 식별할 수 있는 유용한 방법이다. 전통적인 계층 상향 군집화는 모든 클러스터가 단일 클러스터에 속할 때까지 가장 가까운 클러스터 쌍을 반복적으로 찾는 작업을 한다. 이 작업은 시간과 메모리 복잡도가 매우 높아지게 되는데, 이를 해결하기 위해 Shalom 등은 효율적인 셀 분할 부분 중복 계층 상향 군집화를 설명하고 구현하였다. 본 논문은 Shalom 등이 제안한 셀 분할 부분 중복 계층 상향 군집화 방법을 개선하여 셀 분할 병렬 계층 상향 군집화 기법을 제시하였다. 실험결과 전통적인 계층 상향 군집화와 비교하여 제안한 방법이 약 2~10배의 속도 향상을 얻었다.

**Key Words** : data partitioning, efficient partitioning, clustering, hierarchical agglomerative clustering, parallel clustering

### ABSTRACT

In today's growth in CPU-based parallel and distributed computing fields, general-purpose graphics processing unit(GPGPU) has shown tremendous success in computing speeds. Data partitioning is a very important task in applications in the field of data mining with various domains. In addition, hierarchical agglomerative clustering is a useful method of identifying the number and pattern of clusters using the clustering hierarchy. The traditional hierarchical agglomerative clustering repeatedly searches for the closest cluster pair until all clusters belong to a single cluster. This task increases complexity of time and memory, to solve this problem, Shalom et al. described and implemented an efficient cell partitioning partially overlapping hierarchical agglomerative clustering. In this paper, we propose a cell partitioning parallel hierarchical agglomerative method by improving the cell partitioning partially overlapping hierarchical agglomerative clustering method proposed by Shalom et al. Experimental results show that the proposed method improves about 2~10 times than traditional hierarchical agglomerative clustering.

\* First and Corresponding Author : Sangji Youngseo College, Department of National Defense Communication Engineering, seobs@sy.ac.kr, 정회원

논문번호 : 201909-199-D-RU, Received September 17, 2019; Revised October 4, 2019; Accepted October 4, 2019

## I. 서 론

오늘날 다양한 멀티코어 프로세서 고성능 컴퓨팅은 프로그래밍 분야에서 여전히 어려운 과제로 남아 있다. CPU 기반 병렬 및 분산처리 컴퓨팅 분야의 성장 속에서 범용 그래픽 처리 장치(GPGPU)도 계산 속도를 몇 배로 높이는 엄청난 성과를 보여주었다<sup>1)</sup>.

전통적인 데이터 분할 기법은 많은 양의 데이터에 대하여 사용하는 메모리와 처리하는 시간이 매우 크다. 일반적인 예로 두 데이터 사이의 거리(Euclidean distance, Euclidean metric)와 같은 반복적인 산술 계산이 많기 때문이다.

데이터 분할 기법은 다양한 도메인을 갖는 데이터 마이닝 분야의 응용 프로그램에서 매우 중요한 작업이다<sup>2)</sup>. 계층 상향 군집화(Hierarchical Agglomerative Clustering, HAC)도 중요하고 유용한 기술로 군집화 계층(clustering hierarchy)을 이용하여 클러스터의 수와 패턴을 식별할 수 있는 유연성을 가진 클러스터들을 생성한다. DNA 마이크로어레이 및 염기 서열, 생물정보학에 관한 연구는 광대한 생물학적 데이터 세트를 처리하고 조작하는 이러한 연구 중에서 가장 중요한 영역이라고 말할 수 있다<sup>3)</sup>.

전통적인 계층 상향 군집화는 각 데이터를 별도의 클러스터로 간주하여 모든 클러스터가 단일 클러스터에 속할 때까지 가장 가까운 클러스터 쌍을 반복적으로 찾는 작업을 한다<sup>4)</sup>.

BIRCH 및 Cure 논문에서는 샘플링 및 요약 기반으로 대규모 데이터 세트를 처리하는 효율적인 기술이 제안되었다<sup>5,6)</sup>. X. Li 등은 계층 및 병렬 군집화에 대해 효율적인 방법을 제안하였다<sup>7)8)</sup>. K-means, Fuzzy C-Means(FCM) 및 전통적인 계층 상향 군집화와 같은 군집화 기술이 GPU 상에서 구현되어 CPU 상에서의 구현에 비해 현저한 속도 향상도 얻었다<sup>9-13)</sup>. Shalom 등은 효율적인 셀 분할 부분 중복 계층 상향 군집화 데이터 분할 방법을 설명하고 구현하였다<sup>14,15)</sup>.

본 논문은 Shalom 등이 제안한 효율적인 셀 분할 부분 중복 계층 상향 군집화 방법의 문제점을 개선하여 셀 분할 병렬 계층 상향 군집화 기법을 제시하고, 전통적인 계층 상향 군집화와 성능을 비교하여 논의하고자 한다.

## II. 본 론

### 2.1 셀 분할 기반 데이터 분할 방법

#### 2.1.1 전통적인 계층 상향 군집화

전통적인 계층 상향 군집화의 시작 전제 조건은 2차원 공간에 하나의 데이터가 하나의 클러스터가 된다.(데이터의 개수가  $n$ 이라면 초기 클러스터의 개수도  $n$ ) 모든 클러스터는 자신을 제외한 다른 클러스터들과의 거리를 계산하여  $D(\text{istance})$  matrix를 만들고, 자기와 가장 가까운 클러스터가 어떤 것인지 찾아  $N(\text{earest})$   $N(\text{eighbor})$  matrix와 그 클러스터와의 거리를 나타내는  $N(\text{earest})$   $N(\text{eighbor})$   $D(\text{istance})$  matrix를 만든다. 이 결과 모든 클러스터 중 가장 가까운 한 쌍의 클러스터만 병합한다. 병합이 일어난 후, 병합된 클러스터는 다른 모든 클러스터와의 거리( $D$  matrix)를 계산하고, 자기와 가장 가까운 클러스터( $NN$  matrix)와 그 거리( $NND$  matrix)를 업데이트한다. 업데이트 후, 다시 전체에서 가장 가까운 한 쌍의 클러스터만 병합한다. 이러한 과정을 모든 클러스터가 하나의 클러스터에 속할 때까지 반복하여 수행한다.

#### 2.1.2 셀 분할 계층 상향 군집화

##### 1) 셀 분할 계층 상향 군집화의 기본 아이디어

셀 분할 계층 상향 군집화의 기본 아이디어는  $n$ 개의 클러스터가 있는 2차원 공간을 여러 개의 셀로 나누고, 각 셀은 독립적으로 셀 내부에서 거리가 가장 가까운 두 클러스터를 선정한다. 각 셀에서 발견한 거리가 가장 가까운 한 쌍의 클러스터 중 가장 최소거리인 한 쌍의 클러스터만 병합한다.(그림 1에서

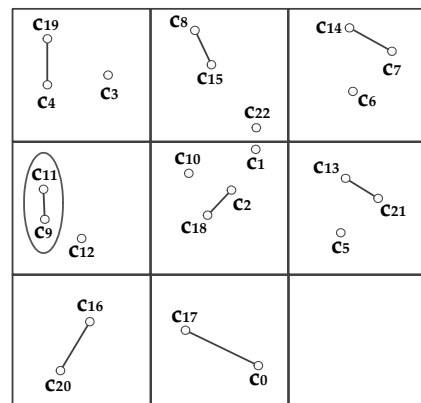


그림 1. 셀 분할 계층 상향 군집화의 기본 아이디어  
Fig. 1. The basic idea of cell partitioning hierarchical agglomerative clustering

{C9,C11}만 병합) 병합이 일어난 해당 셀만 D, NN, NND matrix를 업데이트한다. 다시 각 셀로부터 선정된 거리가 가장 가까운 클러스터의 쌍 중 가장 최소거리인 한 쌍의 클러스터만 병합한다. 앞에서와같이 병합이 일어난 셀만 D, NN, NND matrix를 업데이트한다. 이러한 과정을 초기 클러스터의 개수 n이 1이 될 때까지 반복한다.

2) 셀 분할 계층 상향 군집화의 문제점

전통적인 계층 상향 군집화와 셀 분할 계층 상향 군집화에서의 병합 조건은 각 셀에서 선정된 거리가 가장 가까운 클러스터의 쌍 중 가장 최소거리인 한 쌍의 클러스터만 병합하는 것이다.

가장 최소거리인 한 쌍의 클러스터는 반드시 각 셀에서 선정된 가장 가까운 한 쌍의 클러스터 중 가장 최소거리여야만 한다. 이러한 조건은 셀을 분할하기 전 전체 클러스터 중 거리가 가장 최소거리인 한 쌍의 클러스터만 병합해야 한다는 조건과 다르지 않다. 하지만, 그림 2에서 보듯이 두 클러스터가 다른 셀에 소속된 가장 가까운 거리의 클러스터 쌍은 선정되지 않는다. {C9,C11} 클러스터 쌍보다 더 가까운 최소거리의 클러스터 쌍인 {C1,C22}가 있는데 이것을 먼저 병합하지 않은 것이다. 이렇게 되면 군집화의 결과물인 군집화 계층(clustering hierarchy)이 달라지게 된다.

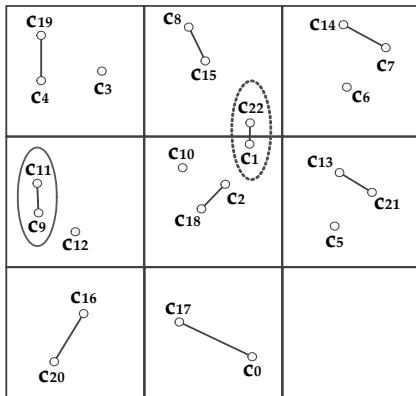


그림 2. 셀 분할 계층 상향 군집화의 문제점  
Fig. 2. Problem of cell partitioning hierarchical agglomerative clustering

2.1.3 셀 분할 부분 중복 계층 상향 군집화

1) 셀 분할 부분 중복 계층 상향 군집화의 기본 아이디어

셀 분할 계층 상향 군집화의 문제점을 해결하기 위해 각 셀에서 해당 셀을 확장하여 인접 셀의 확장된

영역에 대해서도 두 클러스터 간의 거리를 찾아내도록 한 방법이 있다<sup>14,15</sup>. 그림 3은 셀 분할 부분 중복 계층 상향 군집화의 기본 아이디어를 보여주고 있다.

그림 3의 점선 사각형에서 해당 셀이 아닌 인접 셀이 부분 중복 분할(Partially Overlapping Partitioning, PoP) 영역이다. 부분 중복 분할영역의 클러스터들은 셀 내부의 모든 클러스터와 거리를 고려한다.(C1,C2,C10,C18,C22) 부분 중복 분할영역 내의 두 클러스터 간의 거리는 고려하지 않는다. 클러스터의 해당 셀 내부에서 가장 가까운 클러스터인지 이미 확인했기 때문이다.

두 클러스터 중 한 클러스터는 셀 내부에 또 한 클러스터는 부분 중복 분할영역에 존재하고 그 두 클러스터의 거리가 가장 가깝다면({C1,C22}) 셀 내부의 가장 가까운 두 클러스터는 보고하지 않는다.({C2,C18},{C8,C15}) 각 클러스터의 쌍은 소속된 셀이 달라서 D, NN, NND matrix의 정보가 소속된 셀에 존재하여, 두 클러스터가 아무리 가장 최소거리의 클러스터 쌍이어도 병합할 수 없다.

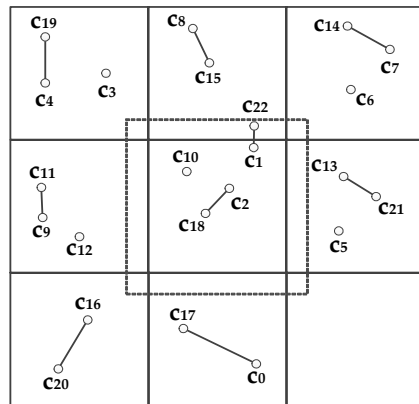


그림 3. 셀 분할 부분 중복 계층 상향 군집화의 기본 아이디어  
Fig. 3. The basic idea of cell partitioning partially overlapping hierarchical agglomerative clustering

2) 셀 분할 부분 중복 계층 상향 군집화의 문제점  
셀 분할 군집화 방법을 사용하지 않을 경우, 모든 클러스터에 대하여 거리 계산이 여러 번 필요하다. 여기서 모든 클러스터는 매우 큰 수이다.(10만 개 이상) 셀 분할 군집화 방법을 사용하면 해당 셀에 소속된 클러스터들의 거리 계산만 하면 된다. 이 해당 셀에 소속된 클러스터들은 모든 클러스터의 수보다 상대적으로 아주 작은 수이다.(수천 개 이하)

하지만, 셀 분할 부분 중복 계층 상향 군집화의 문

제점은 각 셀이 독립적으로 셀 내부의 가장 가까운 한 쌍의 클러스터를 찾고, 이들 중 가장 최소거리의 한 쌍의 클러스터만 병합하기 때문에 나머지 셀들은 항상 대기하게 된다. 셀 분할 군집화 방법을 사용한 이유는 병렬성(parallelism)을 높이기 위함이었는데, 이렇게 되면 병렬성이 낮아지게 된다.

## 2.2 개선된 셀 분할 기반 데이터 분할 기법

### 2.2.1 병렬 계층 상향 군집화의 기본 아이디어

그림 4는 병렬 계층 상향 군집화의 기본 아이디어를 보여주고 있다. 전체 클러스터 중 가장 가까운 한 쌍의 클러스터 C0와 C1이 존재하더라도, C3와 C4가  $NN(C3)=NN(C4)$ ,  $NN(C4)=NN(C3)$ 를 만족하면(서로-NN) 병렬 계층 상향 군집화에서는 가장 최소거리의 클러스터 쌍이 아니어도 병합한다. 그러나, 전통적인 계층 상향 군집화에서는 전체 클러스터 중 가장 최소거리의 쌍인 클러스터가 존재한다면, 서로-NN을 만족하는 클러스터가 존재하여도 병합하면 안 된다. 가장 최소거리의 한 쌍의 클러스터만 병합할 수 있기 때문이다. 다시 말해, 서로-NN 조건을 만족한다고 무조건 병합하면 안 된다.

다음 단계로 클러스터 {C3,C4}와 C5가 서로-NN 즉,  $NN(\{C3,C4\})=NN(C5)$ ,  $NN(C5)=NN(\{C3,C4\})$ 의 조건을 만족하기 때문에 가장 최소거리의 클러스터 쌍({C0,C1})이 존재하고, 아직 병합하지 않았더라도 클러스터 {C3,C4}와 C5를 먼저 병합해도 문제되지 않는다.

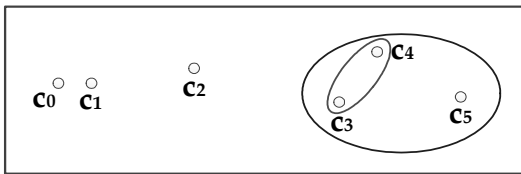


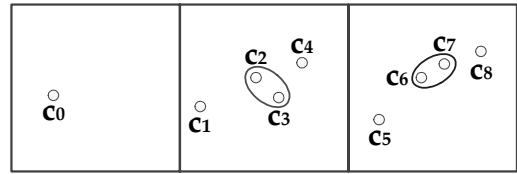
그림 4. 병렬 계층 상향 군집화의 기본 아이디어  
Fig. 4. The basic idea of parallel hierarchical agglomerative clustering

### 2.2.2 제안하는 셀 분할 병렬 계층 상향 군집화

#### 1) 셀 분할 병렬 계층 상향 군집화 과정

그림 5는 제안하는 셀 분할 병렬 계층 상향 군집화 과정을 보여주고 있다. 먼저 각 셀에서 가장 가까운 두 클러스터를 찾고, 서로-NN 조건을 만족하는지 확인한다.  $NN(C2)=NN(C3)$ ,  $NN(C3)=NN(C2)$ ,  $NN(C6)=NN(C7)$ ,  $NN(C7)=NN(C6)$ 으로 서로-NN 조건을 만족하기 때문에, 각 셀의 두 클러스터 쌍을

[Step 1]



[Step 2]

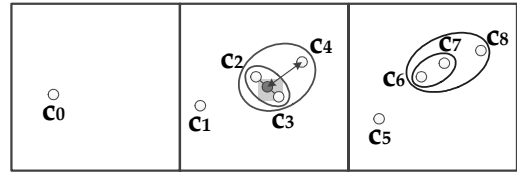


그림 5. 셀 분할 병렬 계층 상향 군집화 과정  
Fig. 5. Cell partitioning parallel hierarchical agglomerative clustering process

동시에 병합한다.([Step 1])

그다음 병합이 일어난 각 셀에서만 가장 가까운 두 클러스터를 찾고, 앞에서와 같이 서로-NN 조건을 만족하는지 확인한다.  $NN(\{C2,C3\})=NN(C4)$ ,  $NN(C4)=NN(\{C2,C3\})$ ,  $NN(\{C6,C7\})=NN(C8)$ ,  $NN(C8)=NN(\{C6,C7\})$ 으로 서로-NN 조건을 만족하기 때문에 각 셀의 두 클러스터 쌍을 동시에 병합한다.([Step 2]) 이런 방식으로 각 셀에 두 개의 클러스터가 남을 때까지 반복해서 수행한다.

앞에서 말한 셀 분할 계층 상향 군집화와 셀 분할 부분 중복 계층 상향 군집화의 공통적인 문제점은 인접 셀 또는 PoP 영역에 가장 최소거리의 클러스터가 존재할 수도 있다는 것이다. 그림 6에서 보면, 해당 셀

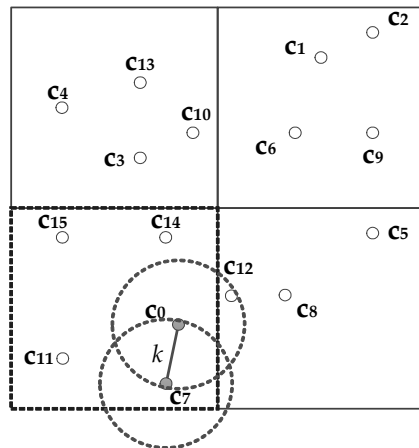


그림 6. 인접 셀에 최소거리의 다른 클러스터 존재 여부 확인  
Fig. 6. Check for the existence of another cluster of minimum distance in adjacent cells

에 가장 가까운 두 클러스터 C0, C7이 존재하고 서로 -NN 조건을 만족한다. 하지만 위의 이유로 이 두 클러스터는 바로 병합할 수 없다.

셀 내부의 가장 가까운 이 두 클러스터의 최소거리를  $k$  라하고, 클러스터 C0에서  $k$ 를 반지름으로 하는 주변을 살폈을 때, 인접 셀의 경계를 넘어서게 되면 경계를 넘어서 인접 셀의 모든 클러스터와의 거리를 비교해서  $k$ 보다 거리가 가까운 클러스터가 존재하는지 찾아보아야 한다. 이것을 클러스터 C7에서도 같은 방법으로 수행해 본다.

이렇게 수행하여 인접 셀에서 클러스터 C0와 C7의 최소거리  $k$ 보다 가까운 클러스터가 나타나지 않았을 때(전체-NN), 해당 셀의 두 클러스터 C0와 C7을 병합한다. 하지만, 그림 6의 경우에는 두 클러스터 C0와 C7이  $NN(C0)=NN(C7)$ ,  $NN(C7)=NN(C0)$ 으로 서로 -NN을 만족하지만, 클러스터 C0에서 인접 셀의 클러스터 C12가 클러스터 C0와 C7의 최소거리  $k$ 보다 더 최소거리이기 때문에 클러스터 C0와 C7은 전체-NN 조건을 만족하지 못하여 병합할 수 없다.

2) 셀 분할 병렬 계층 상향 군집화의 병합 조건

제안하는 셀 분할 병렬 계층 상향 군집화 방법은 각 셀 내부에서 두 클러스터가 서로-NN 조건을 만족하고, 인접한 외부 셀에 그 두 클러스터의 최소거리보다 더 가까운 클러스터가 존재하지 않는 전체-NN 조건을 동시에 만족해야 병합할 수 있다. 이러한 조건이 반드시 갖추어져야만 각 셀에서 독립적으로 병합이 일어날 수 있으며, 병렬성 또한 높아지게 된다. 인접한 외부 셀에 가장 가까운 거리의 두 클러스터가 모두 존재하는 경우에는 두 클러스터의 해당 소속 셀에서 최소거리의 여부가 이미 처리가 됐을 뿐만 아니라, 병합하더라도 병합 뒤 새로운 클러스터의 소속 결정에 있어서 병합 방법(single-link, complete-link, group-average, centroid 등)에 따라 다르기 때문에 전체-NN 조건을 만족하더라도 병합하면 안 된다. 반드시 동일한 셀에 두 클러스터가 존재하고, 서로-NN과 전체-NN 조건을 만족하는 경우에만 병합할 수 있다.

III. 실험 및 결과

3.1 실험환경

실험한 데스크탑 컴퓨터의 사양은 Intel Core i5-2400 (3.1GHz), RAM 8GB(DDR2), nVidia GeForce GT-520 이다. 그래픽카드의 세부 사양은 그림 7과 같다.

그림 8은 병합 전 클러스터의 초기 세팅 sk모습을 보여준다. 셀의 개수는 64개(8\*8)로 고정하였으며, 클러스터의 수는 0개에서 20,000개까지 1000개씩 증가시키면서, 하나의 단일 클러스터가 되기까지의 시간을 측정하였다.

GPU Technology	40nm
Cuda Cores	48
Memory Size	1024 MB
Memory Interface	64 BIT
Memory Type	DDR3
Core Clock	810Mhz
Memory Clock	1333Mhz
Processor Clock	1620Mhz
DX Support	DX11
Bus Interface	PCI Express 2.0

그림 7. nVidia GeForce GT-520의 사양  
Fig. 7. Specification of nVidia GeForce GT-520

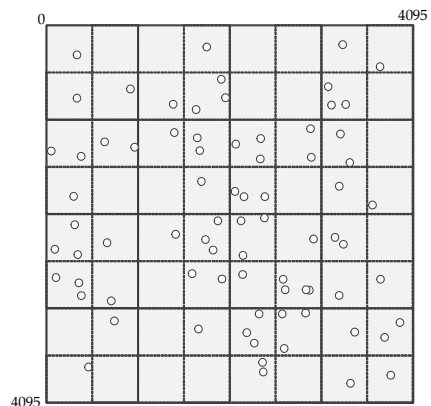


그림 8. 병합 전 클러스터의 초기 세팅  
Fig. 8. Initial setting of the cluster before merging

3.2 실험결과

클러스터의 수가 1,000개에서 3,000개까지는 성능 향상이 거의 없었는데, 이것은 제안하는 방법이 전통적인 계층 상향 군집화보다 수행시간이 더 오래 걸렸다는 것이다. 클러스터의 수가 상대적으로 적은 경우 셀 분할 기반의 군집화 방법이 해당 셀에서 확인해야 할 조건(서로-NN, 전체-NN) 때문에 오히려 순차적으로 처리하는 것보다 수행시간이 오래 걸린 것이라고 판단된다. 다시 말하면 클러스터의 수가 많아야 병렬

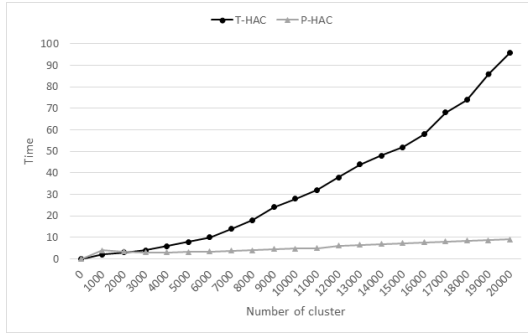


그림 9. 전통적인 계층 상향 군집화와 셀 분할 병렬 계층 상향 군집화의 성능 비교  
 Fig. 9. Performance comparison of Traditional HAC(T-HAC) and cell partitioning Parallel HAC(P-HAC)

성과 동시성의 의미가 있게 된다.

제안하는 셀 분할 병렬 계층 상향 군집화(P-HAC)와 전통적인 계층 상향 군집화(T-HAC)를 고정된 셀 환경(64개)에서 클러스터의 수를 증가시키며 실험한 결과 약 2~10배까지 속도 향상이 있었다.

#### IV. 결 론

본 논문은 주어진 클러스터를 셀 분할 부분 중복 계층 상향 군집화 방법으로 데이터 독립적인 계산을 병렬화하도록 처리한 Shalom 등의 방법<sup>[14][15]</sup>을 개선 하였다. 전통적인 계층 상향 군집화 방법은 순차적으로 처리하기 때문에 많은 양의 데이터를 한 번에 처리하기에 컴퓨터의 메모리 부족, 많은 계산량으로 인한 시간 지연, 낮은 병렬성 등의 문제점이 있다. 이것을 개선하여 동시성(concurrency)과 병렬성(parallelism)의 극대화에 초점을 맞추어 셀 분할 병렬 계층 상향 군집화 기법을 제안하였다.

제안한 셀 분할 병렬 계층 상향 군집화의 성능을 비교하기 위해 전통적인 계층 상향 군집화와 고정된 셀 환경(64개)에서 클러스터의 수를 증가시키며 실험 하였다. 그 결과 전통적인 계층 상향 군집화 알고리즘(T-HAC)과 비교했을 때 약 2~10배까지 속도 향상이 있었다. 클러스터의 개수가 1,000개에서 3,000개까지는 성능향상이 거의 없었으며, 전체적으로는 두 방법 모두 수행시간이 증가하는 추세를 보였다. 특히 전통적인 계층 상향 군집화 방법은 클러스터의 수가 증가할수록 기하급수적으로 수행시간이 증가하였다.

본 논문에서 제안하는 셀 분할 병렬 계층 상향 군집화와 Shalom 등이 제안한 셀 분할 부분 중복 계층 상향 군집화와의 성능 비교는 자세히 언급하지 못했

다. 또한, 추후 연구해야 할 과제로는 어느 정도의 셀 크기가 성능향상에 가장 좋은지, 생체정보와 같은 다차원 다량 데이터의 병렬 군집화 방법 등이 있다.

#### References

- [1] NVIDIA Corporation, *CUDA C Programming Guide Version 10.0* (2018), Retrieved Aug., 30, 2019, from <http://developer.nvidia.com>.
- [2] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*, Morgan Kaufmann Pub, 2011.
- [3] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*, Prentice-Hall, Inc., 1988.
- [4] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
- [5] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proc. 1996 ACM SIGMOD Int. Conf. Management of Data*, vol. 25, no. 2, pp. 103-114, Montreal, Quebec, Canada, Jun. 1996.
- [6] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," *Inf. Systems*, vol. 26, no. 1, pp. 35-38, Mar. 2001.
- [7] X. Li, "Parallel algorithms for hierarchical clustering and cluster validity," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 12, no. 11, pp. 1088-1092, Nov. 1990.
- [8] X. Li and Z. Fang, "Parallel clustering algorithms," *Parallel Computing*, vol. 11, no. 3, pp. 275-290, Aug. 1989.
- [9] F. Cao and A. Y. Zhou, "Fast clustering of data streams using graphics processors," *J. Software*, vol. 18, no. 2, Feb. 2007.
- [10] D. Chang, M. Kantardzic, and M. Ouyang, "Hierarchical clustering with CUDA/GPU," *Symp. Computer Animation*, pp. 7-12, New Orleans, USA, 2009.
- [11] F. Cao, A. Tung, and A. Zhou, "Scalable clustering using graphics processors," *Int. Conf. WAIM 2006: Advances in Web-Age Inf. Management*, pp. 372-384, Jun. 2006.

- [12] S. A. Arul Shalom, M. Dash, and M. Tue, "Efficient K-Means clustering using accelerated graphics processors," *Int. Conf. DaWaK 2008: Data Warehousing and Knowledge Discovery*, pp. 166-175, Italy, 2008.
- [13] S. A. A. Shalom, M. Dash, and M. Tue, "Graphics hardware based efficient and scalable fuzzy C-Means clustering," in *Proc. 7th AusDM 2008, ACS*, vol. 87, pp. 179-186, Glenelg, South Australia, Nov. 2008.
- [14] S. A. Arul Shalom and M. Dash, "Efficient hierarchical agglomerative clustering algorithms on GPU using data partitioning," *12th PDCAT*, pp. 134-139, 2011.
- [15] S. A. Arul Shalom and M. Dash, "Efficient partitioning based hierarchical agglomerative clustering using graphics accelerators with CUDA," *IJAIA*, vol. 4, no. 2, Mar. 2013.

서 병 석 (Byung-suk Seo)



2001년 2월 : 연세대학교 문리  
대학 전산학과 학사  
2008년 2월 : 연세대학교 일반  
대학원 전산학과 석사  
2011년 2월 : 연세대학교 일반  
대학원 전산학과 박사 수료  
2012년 3월~현재 : 상지영서대

학교 국방정보통신과 조교수  
<관심분야> 병렬처리, 정보보호시스템, 네트워크보안  
[ORCID:0000-0002-2771-0112]