

# 3D GIS를 위한 병렬 컴퓨터 시스템의 설계

김진석                      조정우<sup>0</sup>

서울시립대학교 컴퓨터·통계학과  
{jskim, jwjo95<sup>0</sup>}@venus.uos.ac.kr

## Design of Parallel Computer System for 3D Geographic Information

Jin Suk Kim, and Jeongwoo Jo<sup>0</sup>

Dept. of Computer Science & Statistics, University of Seoul

### 요 약

3D 지도를 이용하여 3D 영상을 처리하는 시스템이 많이 상용화되어 있다. 기존에 3D 지도를 처리하기 위한 방법으로 고성능의 시스템을 이용하였다. 하지만 고성능의 시스템을 사용하여 GIS 시스템을 구현할 경우 가격의 부담이 크다는 문제점이 있다. 또한 일반 시스템에서 3D 지도를 처리하려면 3D 지도의 파일의 크기가 크기 때문에 공간영상을 처리하는데 시간이 오래 걸린다는 단점이 있다. 따라서 본 논문에서는 Linux를 기반으로 3D GIS를 위한 병렬 컴퓨터 시스템을 설계한다.

### 1. 서 론

3D GIS를 이용하여 3D 영상을 처리하는 시스템이 많이 상용화되어 있다. 기존에 3D 영상을 처리하기 위한 방법으로 고성능의 시스템을 이용하였지만 고성능의 시스템을 사용하여 GIS 시스템을 구현할 경우 가격의 부담이 크다는 문제점이 있다. 또한 일반 시스템에서 3D 영상을 처리하려면 3D 영상의 파일의 크기가 크기 때문에 공간영상을 처리하는데 시간이 오래 걸린다는 단점이 있다. 따라서 본 논문에서는 가격 대비 성능에서 큰 효과를 볼 수 있는 3D GIS를 위한 병렬화 방법을 제안한다. GIS 시스템을 병렬화하면 3D 영상을 처리하는 시간이 단축되며 원래의 3D 영상을 손실 없이 처리할 수 있게 된다.

본 논문에서 구현하고자 하는 시스템의 구조는 <그림 1>과 같다. 각 node는 하나의 processor와 memory를 가지고 있으며, 각 node끼리는 network으로 연결되어있어서 서로의 정보를 공유한다.

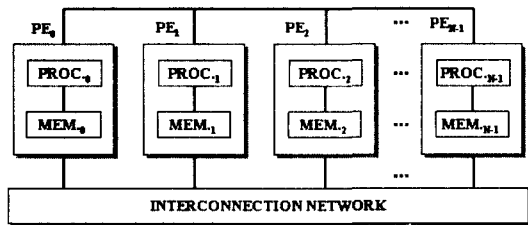


그림 1. 병렬 컴퓨터 시스템 구조

### 2. 관련연구

#### 2.1 기존 GIS시스템의 3D영상처리

기존 GIS시스템에서는 3D영상을 처리할 때, 하나의 processor에서 영상을 입력받아 전 과정을 처리하였다. 이렇게 하나의 processor에서 모든 과정을 처리할 경우에 3D영상의 파일 크기가 너무 크기 때문에 속도가 현저하게 느려지는 결과를 보인다.

#### 2.2 병렬 GIS시스템의 3D영상처리

병렬 GIS시스템에서 3D영상을 처리할 경우를 살펴보면 <그림 2>와 같다. <그림 2>에서 보듯이 병렬 GIS에서는 원 3D영상을 분할하여 여러 processor에서 실행을 시킨다. 이렇게 여러 processor에서 3D영상을 처리하면 하나의 processor를 이용하는 기존의 GIS시스템보다 빠른 시간에 3D영상을 처리할 수 있다.

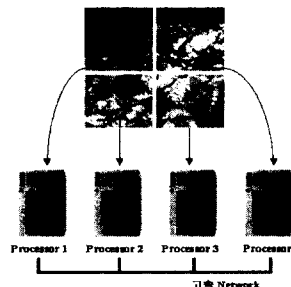


그림 2. 병렬 GIS 시스템

### 3. 영상분할 방법

본 논문에서 제안하고자하는 architecture는 다수의 node를 가지는 3D GIS를 위한 분산 MIMD architecture를 설계하는 것이다. GIS에는 3D 영상을 표시하기 위한 여러 가지의 display format이 있다. 예를 들어 raster, bitmap, tiff, vector등의 format을 지원한다. 본 논문에서는 이런 display format중에서 raster format을 사용하기로 한다. raster는 화면공간상에 x, y, z 좌표정보를 가진 영상이다.

raster영상을 3D GIS를 사용하여 display할 경우, 3D GIS에서는 raster 영상을 한 줄씩 왼쪽에서 오른쪽으로 line을 그리면서 display한다. 이때 그려지는 방법은 두 점의 좌표값을 연결한 line을 그리는 것이다.

예를들면,

`draw_line(x1, y1, x2, y2)`

과 같다.

본 논문에서 구현한 architecture는 이 raster 영상을 분할하여 display할 수 있다. 즉 하나의 raster 영상을 여러 node에서 분할하여 각 조각진 영상을 display하는 것이다. 위의 방법을 사용하려면 raster image 분할 방법이 필요하게 된다. raster image를 분할하는 방법을 살펴보면 크게 3가지로 나뉜다. 상·하로 나누는 방법과, 좌·우로 나누는 방법, 그리고 마지막으로 수직으로 나누는 방법이 그것이다.

<그림 3>, <그림 4>, <그림 5>가 raster 영상을 분할하는 방법을 보여주고 있다.



그림 3.  
상·하 분할



그림 4.  
좌·우 분할



그림 5.  
수직 분할

하지만 <그림 3>에서 보듯이 영상을 상·하로 분할 할 경우에는 3D GIS에서 문제가 있다. 예를 들어, 2개로 분할 한 경우를 살펴보면, 문제는 영상을 다수의 node가 동시에 그리게 되는데 거의 영상이 완료되었을 경우에 위의 영상이 아래의 영상을 덮어쓰면서 display하게 된다는 것이다. 즉 뒤에 있어야 할 영상이 앞으로 나오는 경우가 발생한다. 이렇게 되면 원 raster 영상과 다른 결과를 display하게 되는 것이다.

<그림 6>에 보면 화면을 상하로 분할하여 display 할 경우에 문제가 되는 것을 보여주고 있다. 그림 가운데 파란 부분이 상하로 분할하여 display 할 때 위의 그림이 아래의 그림을 덮어쓰게 된다. 이렇게 되면 원래의 raster 영상에 손실을 가져오는 결과를 초래한다.

이러한 문제점은 <그림 4>와 <그림 5> 같은 분할 방법을 사용하면 제거할 수 있다. <그림 4>와 <그림 5> 둘다 뒤의 영상이 앞의 영상에 영향을 주지 않는다. 하지만 그리는 방법을 보면 <그림 5>의 영상이 분할할 때 더 많은 연산이 필요하며, 분할 하기가 <그림 4>보다 어렵다는 것을 알 수 있다. 따라서 본 논문에서 구현한 3D GIS를 위한 병렬 컴퓨터에서는 <그림 4>의 방법을 이용한다.

<그림 4>은 raster 영상이 두점의 line을 연결하기 때문에 한 줄에서는 겹쳐지지 않는다는 특징을 이용한 것이다.

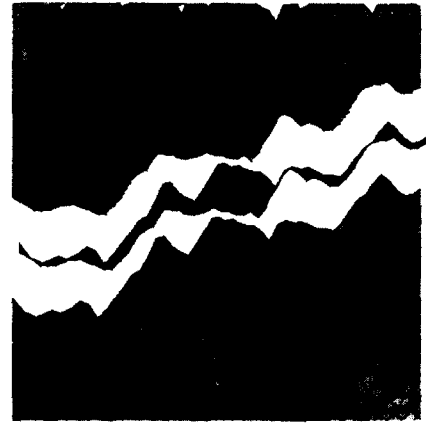


그림 6. 상하 분할 시 겹쳐지는 영상

### 4. 실험 및 결과

본 논문에서 제안하고자하는 architecture는 6개의 node를 가지고 각 node는 하나의 cpu와 memory, disk를 가진다. 우선 6개의 node 중 하나를 master node로 사용하고, 나머지는 slave node로 구성한다. master node는 raster file을 입력받아 file을 partition 하는 역할과 partition된 정보를 slave node로 보내는 역할을 담당한다. slave node는 master node에서 보내온 정보를 이용하여 각 partition된 file의 영상을 화면에 display 하는 역할을 담당한다. 본 논문에서는 master node와 slave node가 같은 정보와 file을 가지고 있다는 가정 하에 구현되었다. 만약 master node에만 file이 있고, 이 파일을 slave node에게 보내게 되면 network traffic이 많아지게 되어 display 속도가 감소하는 작용을 하게 된다. 본 논문에서의 시뮬레이션을 위해 6개의 node를 갖는 병렬 architecture를 설계한다. 각 node는 linux를 기반으로 하고 100mbps Ethernet으로 연결된다. 또한 GIS 프로그램은 linux에서 동작하는 Grass라는 프로그램을 사용하였다. Grass는 source가 공개되어 있기 때문에 본 논문에서 시뮬레이션하기에 적당하다.

본 시뮬레이션에 사용된 raster image는 <그림 7>에 나와있는 KDEM이라는 한반도 영상이다. 시뮬레이션 결과는 아래 <표 1>에 나타나 있다.



그림 7. 시뮬레이션에 사용된 raster영상

아래의 <그림 8>은 총수행시간을 각 node 수 별로 그래프를 이용하여 보여주고 있다. 이때, 각 node수마다 시간의 차이가 있으므로 시간이 가장 많이 걸리는 것을 기준으로 <그림 8>의 그래프를 작성하였다. 그래프를 보면 좀더 확실히 node의 수가 증가할수록 총 수행 시간이 줄어드는 것을 볼 수 있다.

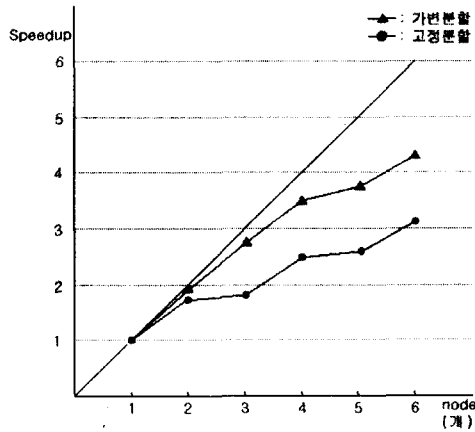


그림 8. 분할 영상 display결과

그런데 node를 여러개로 나누었을 때, 각 node마다 수행시간의 차이를 보이는 것을 알 수 있다. 이것은 각 node마다 할당된 영상에 차이가 있기 때문이다. 그래서 전체적인 수행시간을 줄이려면 각 node에 영상을 분배할 때, 영상을 가변적으로 분할하여야 한다. 가변길이 분할은 raster영상의 좌표가 (0,0)인 것은 그림을 그리지 않는다는 특성을 이용하여 분할하였다.

영상 분할을 고정으로 하지 않고 가변으로 했을 경우를 살펴보면 다음 <표 1>과 같다. 테스트 영상을 살펴보면 좌·우의 display할 영상이 가운데보다 적다는 것을 알 수 있다. <표 1>에서 살펴보면 4개 이상의 node일 경우 고정분할인 경우 처음과 끝쪽의 node에서 수행시간이 빠르다는 것을 알 수 있다. 이 테스트는 한 행을 정확히 n등분 (n : node의 수) 했을 때의 결과이다. 이 방법을 수정하여 가변길이를 나누었을 경우 속도는 더욱 향상되었다. 아래의 <표 1>는 고정길이와 가변길이를 영상 분할 했을 때의 결과이다. 결과적으로 4개의 node일 경우 고정길이는 최대 21.8282의 속도로서 약 2.47배의 속도향상을 보이지만 가변길이는 최대 15.7516의 속도로서 약 3.42배의 속도향상을 보인다. 6개의 node에서는 3.16배에서 4.27배의 성능향상을 보인다.

5. 결론

본 논문에서는 3D GIS영상을 처리하기 위한 병렬화 시스템을 설계 및 구현하였다. 실험결과에서도 알 수 있듯이 GIS영상을 병렬화했을 때 성능이 향상되었다. 향후연구로는 병렬화를 통해 display 한 영상을 하나의 영상으로 병합하는 것과 본 시스템을 영상뿐만 아니라 GIS의 다른 기능에도 적용하는 것이다.

| Node | 고정분할  |         | 가변분할   |         |
|------|---|---------|--|---------|
|      | (수행시간 : 초)  | Speedup | (수행시간 : 초)   | Speedup |
| 1    | 1/1 : 53.8739   | 1       |  |         |
| 2    | 1/2 : 24.5555<br>2/2 : 31.5995  | 1.70    | 1/2 : 27.4617<br>2/2 : 25.0651   | 1.96    |
| 3    | 1/3 : 9.0354<br>2/3 : 31.0731<br>3/3 : 8.3634   | 1.73    | 1/3 : 19.5512<br>2/3 : 19.4247<br>3/3 : 17.6154  | 2.76    |
| 4    | 1/4 : 4.3113<br>2/4 : 21.8282<br>3/4 : 20.0196<br>4/4 : 4.0950                                  | 2.47    | 1/4 : 15.7516<br>2/4 : 15.2315<br>3/4 : 15.3939<br>4/4 : 14.0777                                   | 3.42    |
| 5    | 1/5 : 2.7088<br>2/5 : 12.6981<br>3/5 : 20.3799<br>4/5 : 12.0330<br>5/5 : 2.9319                 | 2.64    | 1/5 : 14.3565<br>2/5 : 12.6885<br>3/5 : 12.8686<br>4/5 : 13.1528<br>5/5 : 11.5329                  | 3.75    |
| 6    | 1/6 : 3.6276<br>2/6 : 11.6028<br>3/6 : 17.0304<br>4/6 : 12.0375<br>5/6 : 4.6230<br>6/6 : 3.4866 | 3.16    | 1/6 : 12.6068<br>2/6 : 11.2270<br>3/6 : 11.2211<br>4/6 : 11.2960<br>5/6 : 11.6522<br>6/6 : 10.1926 | 4.27    |

참고 문헌

- 1) M. Mirmehdi, and M. Petrou, "Segmentation of color textures," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, pp. 142-159, Feb, 2000.
- 2) W. Li, D. Zhang, Z. Liu, and X. Qiao, "A parallel algorithm for image information restoration," Proc. of High Performance Computing in the Asia-Pacific Region, vol. 2, pp. 790-793, 2000.
- 3) M. Ishii, "Cluster technologies for high performance computing," Proc. of Parallel Architectures, Algorithms, and Networks, pp. 168-170, 1999.
- 4) 성운재, 허성준, 손호준, 원광연, "병렬 디스플레이 시스템과 이를 이용한 분산 이미지 모자이크 기법," 졸업논문.