

ISBN 978-89-5884-996-4 98560



Tiled Display 구축 방법

구 기 범 (Gee Bum Koo)

voxel@kisti.re.kr

Visualization Team, Supercomputing Center

한국과학기술정보연구원

Korea Institute of Science & Technology Information

제목 차례

1. 서론	1
2. Tiled Display의 하드웨어 구성	4
가. 컴퓨팅 시스템(클러스터)	4
(1) 메인보드	4
(2) GPU / 그래픽카드	5
(3) 디스크 및 파일시스템	6
(4) 네트워크	6
나. 출력장치	8
(1) LCD 모니터	8
(2) 프로젝터와 스크린	8
(3) 출력장치를 위한 구조물	9
(4) 입체영상 지원	10
다. 기타 장비	10
(1) 트래킹 장비	10
(2) 음향출력 장치	10
3. 소프트웨어	11
가. OS / X-Windows	11
(1) 리눅스 배포판	11
(2) X-Windows의 설정	11
(3) Netfilter 설정	12
나. Tiled display를 위한 미들웨어	12
(1) DMX	13
(2) Chromium	15
4. 활용사례	17
가. COVISE (일반 가시화 애플리케이션의 실행)	17
나. 유체 데이터의 가시화 (애플리케이션 개발)	17
다. 고해상도 화상회의	18
5. 결론	19

표 차례

[표 2-1] 네트워크 사양 비교	7
--------------------------	---

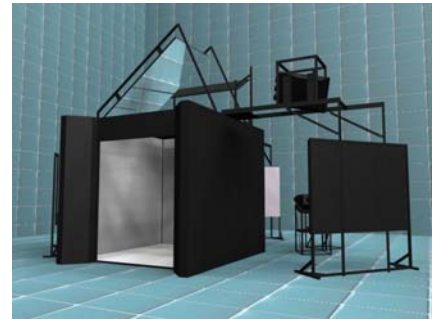
그림 차례

[그림 1-1] CAVE	1
[그림 1-2] Wall screen	1
[그림 1-3] Edge blending이 제대로 구현되지 않은 스크린(좌)과	2
[그림 1-4] Tiled display@UIC	2
[그림 1-5] Tiled display@Cal-(IT)2	2
[그림 2-1] Quad-SLI	4
[그림 2-2] LCD 모니터를 이용한 출력장치	8
[그림 2-3] 프로젝터와 스크린을 이용한 출력장치(NCHC)	9
[그림 2-4] 출력장치를 위한 다양한 형태의 구조물	9
[그림 3-1] /etc/inittab의 내용	12
[그림 3-2] 방화벽 설정 스크립트 예	12
[그림 3-3] /etc/dmx/dmx.conf의 내용	13
[그림 3-4] DMX의 실행	14
[그림 3-5] 애플리케이션의 실행 방법 1	14
[그림 3-6] tiled display를 이용한 애플리케이션의 실행	15
[그림 3-7] options.mk의 수정	16
[그림 3-8] DMX와 Chromium을 연계하면서 자동실행까지 담당하는 mothership 설정	16
[그림 4-1] COVISE 사용 예	17
[그림 4-2] 비행선 유동장 데이터 가시화	17
[그림 4-3] iGrid2005 시연	18

1. 서론

Visualization 시스템은 고성능 컴퓨터나 실험 장비를 이용해서 만들어낸 대용량의 수치 데이터를 시각적으로 표현하기 위해 사용되는 장비와 소프트웨어 일체를 지칭한다. 일반적으로 visualization 시스템은 데이터 저장 및 가공, 영상 생성 등을 담당하는 전용 컴퓨터와 생성된 영상을 보여주는 출력장치로 구성되고, 필요에 따라서 가상현실 입력 장치와 음향 입/출력 장치 등 다양한 장비가 추가된다. Visualization 시스템은 여타의 계산 시스템과 달리 주 활용목적이 영상 생성과 출력에 있기 때문에 출력장치의 형태에 따라서 그 이름을 달리하는데, 현재는 CAVE, wall screen, tiled display 등으로 구분된다.

CAVE(CAVE Automatic Virtual Environment) [그림 1-1]은 UIC(University of Illinois at Chicago)에서 개발한 visualization 시스템이다. 이 장비는 사용자에게 완전한 몰입감을 제공해 준다는 장점이 있다. 하지만 초기 도입비용과 유지보수비가 매우 높고, 넓은 설치공간을 필요로 하기 때문에 충분한 자원을 확보하기 어려운 연구기관이나 학교 등에서는 도입이 어렵다는 단점을 갖고 있다.



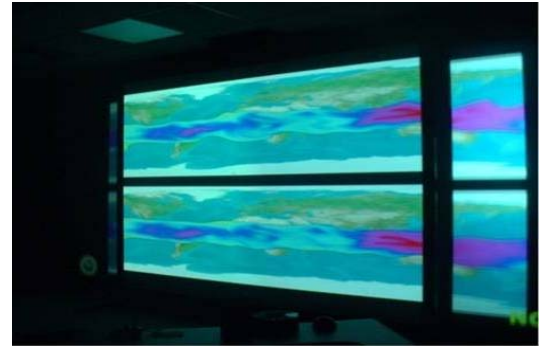
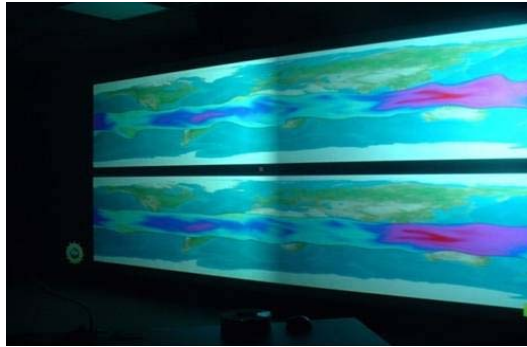
[그림 1-1] CAVE

Wall screen은 대형 스크린과 소수(3대~5대 전후)의 프로젝터로 구성된다[그림 1-2]. Wall screen 형태의 시스템은 CAVE에는 미치지 못하더라도 일정수준 이상의 몰입감을 제공하고, 무엇보다도 CAVE에 비해서 높은 범용성을 갖고 있다. 또한 소규모 강의실부터 대형 강당에 이르기까지 다양한 크기와 형태로 구축할 수 있으며, 구축비용도 CAVE보다 저렴하기 때문에 많이 사용되고 있다. 최근에는



[그림 1-2] Wall screen

는 wall screen에 바닥 스크린을 추가해서 더 높은 몰입감을 구현하거나, 입체영상을 지원하지 않는 프로젝터와 지원하는 프로젝터를 적절히 배분해서 낮은 가격과 입체영상을 동시에 실현하는 형태의 시스템도 등장하고 있다. Wall screen 형태의 시스템을 구축할 때 발생하는 가장 큰 문제점은 서로 인접한 프로젝터가 출력하는 영상 사이의 경계를 매끄럽게 처리하는 것(edge blending과 색상보정)과 평면이 아닌 스크린에 영상을 출력할 때 영상의 찌그러짐을



[그림 1-3] Edge blending이 제대로 구현되지 않은 스크린(좌)과

정상적으로 구현된 스크린(우)

방지(geometry correction)하는 것이다. 저가형 시스템의 경우 이러한 문제를 제대로 해결하지 못해서 낮은 image quality를 보여주는 사례가 적지 않다[그림 1-3].

한편 클러스터의 인기가 높아지고 GPU(Graphics Processing Unit)가 빠르게 발전하면서 이들을 활용한 tiled display를 구축하는 사례가 늘고 있다 [그림 1-4][그림 1-5]. Tiled display는 앞에서 설명한 시스템보다 훨씬 낮은 수준의 가격에 높은 해상도와 높은 성능을 구현할 수 있기 때문에 그 수가 급격히 증가하는 추세다.



[그림 1-4] Tiled display@UIC

여담이지만 CAVE, wall screen, tiled display 모

두 걸으로 드러나는 출력장치의 형태는 다르지만 대부분 리눅스 클러스터를 전용 컴퓨터로 이용하고, 소프트웨어 스택도 큰 차이가 없다. 굳이 차이를 말하자면 wall screen은 하나의 스크린에 여러 대의 프로젝터가 동시에 영상을 출력하고, 각 프로젝터의 영상이 서로 일정부분 겹치도록 해서 경계를 부드럽게 하는

경우가 대부분이다. 반면 tiled display는 모니터의 프레임 때문에 타일(tile) 사이의 경계가 명확히 구분되고, 서로 다른 tile에 출력되는 이미지 사이의 연속성이 깨지는 문제가 발생하지만 wall screen 형태의 장비보다 낮은 가격에 높은 해상도를 구현할 수 있다는 장점이 있다. 머지않은 미래에 이런 문제를 해결하는 출력장치가 나올 것으로 예상된다.



[그림 1-5] Tiled display@Cal-(IT)²

본 보고서는 지난 2005년에 발행한 기술보고서(Tiled Display 구축사례, ISBN 89-5884-331-4 9856)의 내용을 확대해서 tiled display를 구축할 때 필요한

하드웨어와 소프트웨어 구성, 구축 과정에서 발생할 수 있는 문제점 등 실무차원에서 tiled display 구축에 관한 전반적인 내용을 다룬다.

2. Tiled Display의 하드웨어 구성

하드웨어 관점에서 보면 tiled display는 렌더링, 데이터 가공 등을 담당하는 컴퓨팅 시스템(클러스터)과 다수의 모니터(또는 프로젝터와 스크린)를 연결한 출력장치로 구분되며, 여기에 더해서 트래킹 시스템, 음향 입/출력 장비 등을 설치할 수 있다. 대부분의 경우 tiled display에서 대량의 계산을 수행하지는 않기 때문에 전용 컴퓨터는 영상 출력에 최적화시켜서 구성한다. 이 장에서는 컴퓨팅 시스템과 출력장치 각각의 구성요소를 어떤 기준에 따라서 선택할 것인지에 대해 설명한다.

가. 컴퓨팅 시스템(클러스터)

(1) 메인보드

Visualization 시스템을 클러스터로 구축할 때에는 모든 노드에 비디오 카드가 반드시 포함되는데, 이로 인해서 각 노드에 대해 메인보드의 슬롯 구성, 전원(파워 서플라이), 본체 케이스(새시) 등을 신중하게 결정해야 한다.

단일 노드를 놓고 봤을 때, 낮은 CPU 성능을 갖더라도 그래픽 성능을 최대로 끌어올리려면 SLI(NVIDIA)나 CrossFire(AMD)를 지원하는 메인보드를 선택하는 것이 바람직하다. 극단적인 경우 Quad-SLI[그림 2-1]를 지원하는 메인보드를 사용하거나 QuadroPlex와 같은 고성능 장비를 장착할 수 있는데, Quad-SLI의 경우 네 대의 비디오 카드를 한 대의 컴퓨터에 설치해야 하므로 전기, 발열, 확장슬롯의 수, 본체 내의 공간소요 등 많은 제약이 따르기 때문에 신중히 선택해야 한다. QuadroPlex는 GPU 집적도가 높고, 외장형인데다가 필요에 따라서는 랙에 설치할 수도 있으며, 별도의 전원을 사용하는 등 많은 이점이 있지만 가격이 매우 높다는 단점이 있다.



[그림 2-1] Quad-SLI

메인보드를 선택할 때 주의해야 할 또 다른 사항은 PCI express 슬롯의 지원문제다. 시판되고 있는 메인보드 중 상당수가 물리적으로 PCI express x16 슬롯 규격을 따르고 있지만 실제 성능(대역폭)은 PCI express x8 성능(4GB/s)에 불과한데, 이는 time varying data의 렌더링과 같이 지속적으로 그래픽 카드에 데

이터를 전송해야 하는 애플리케이션의 성능을 저하시키는 요인이 될 수 있다. 특히 SLI를 지원하는 메인보드는 두 개의 PCI express x16 슬롯을 갖고 있다고 해도 그 중 하나를 사용할 때만 8GB/s의 대역폭을 제공하고, 두 개를 다 사용할 때에는 각각 4GB/s의 대역폭을 제공하는 경우가 많으므로, 이에 대한 자세한 검증이 필요하다.

마지막으로 최신 그래픽 카드는 전원 커넥터의 수가 두 개로 늘었기 때문에 전원 공급 장치(power supply)가 충분한 전력을 공급해줘야 한다. 여기에 더해서 full-size의 크기를 갖고 있고, 기본적으로 두 개의 확장슬롯을 차지하는 등 물리적인 공간을 많이 차지하기 때문에 추가의 주변장치(인피니밴드, RAID 컨트롤러 등)를 설치할 때 확장슬롯 소요와 배치 등을 꼼꼼히 살펴보고 모델을 정해야 한다.

(2) GPU / 그래픽카드

2007년 현재 양대 GPU 제조업체인 NVIDIA와 AMD(ATI)에서 출시하는 그래픽 카드는 대부분 한 대의 그래픽 카드가 두 대의 모니터를 운영할 수 있다. 여기에 더해서 최신 메인 보드는 동시에 두 개 또는 네 개의 그래픽 카드를 장착할 수 있으므로, 이를 바탕으로 산정해보면 한 대의 노드가 최대 8대의 모니터를 운영할 수 있음을 의미한다. 하지만 모니터가 많이 연결되는 만큼 메인보드와 PCI express 슬롯, GPU 자체의 성능 등이 병목으로 작용할 소지가 있으므로 가능하면 한 대의 노드에 연결하는 모니터의 수는 2대를 초과하지 않는 것이 바람직하다.

그리고 최근 2560 x 1600의 초고해상도를 지원하는 모니터가 출시되고 있는데, 이 해상도를 정상적으로 지원하는 GPU는 한정돼 있으므로 주의해야 한다.

또 다른 문제는 CPU와 GPU의 비율을 어떻게 산정해야 가장 균형 잡힌 시스템으로 구성하느냐 하는 것이다. 대부분의 visualization 애플리케이션이 내부적으로 'CPU 제어 ⇒ GPU 실행'의 형태로 구현되고 있고, CPU(쿼드 코어)의 성능과 GPU의 성능이 1 : 10 이상 벌어져있기 때문에 적절한 CPU와 GPU의 비율은 시스템의 가격, 전체적인 애플리케이션 성능 등을 좌우하는 문제가 된다. 이에 대해서 NVIDIA의 관계자에게 문의한 결과 다음과 같은 답을 얻었다.

This is a very good point and to be honest we don't know since it is very

dependent on application and data. Yes I would agree that 1 CPU per GPU is a good start although, I have heard of apps that work well with fewer CPUs than GPUs so it maybe worth experimenting with the actual applications although I realize this is difficult when you have to decide on a hardware config.

마지막으로, 비록 NVIDIA와 AMD의 GPU 모두 OpenGL 표준 2.0을 지원하고 있지만 대부분의 tiled display가 MS-Windows보다는 Linux를 OS로 채택하는 경우가 많다는 점과 Linux를 위한 디바이스 드라이버의 지원 등을 고려하면 NVIDIA 제품을 선택하는 것이 유리하다.

(3) 디스크 및 파일시스템

Tiled display는 그 목적이 고해상도 영상출력에 있기 때문에 대용량 스토리지에 대한 수요는 없는 편이라고 생각해도 큰 무리는 없다. 하지만 고성능 컴퓨팅 시스템의 스토리지를 tiled display가 공유하거나, 독자적인 파일 서버와 스토리지를 유지해서 어떻게든 계산 전용 시스템의 각 노드가 대용량 데이터를 접근할 수 있는 방법을 만들어놓는 것이 바람직하다. 해외 주요 연구기관에서는 고성능 컴퓨팅 환경을 구성할 때 주력 계산 시스템과 가시화 시스템 사이의 원활한 데이터 공유를 중요하게 생각하고 있다는 점을 참고해야 한다.

파일시스템의 경우, 8~16 노드로 구성되는 클러스터라면 일반적인 NFS로 충분히 수요를 감당할 수 있겠으나 그보다 큰 시스템의 경우에는 Lustre나 GPFS와 같은 병렬 파일시스템을 사용하는 것이 성능 면에서 유리해진다.

(4) 네트워크

네트워크는 크게 클러스터의 노드를 연결하는 노드 간 네트워크(interconnection network)와 외부 네트워크로 나뉘어 생각할 수 있다. Tiled display는 로컬 스토리지에 저장된 데이터를 렌더링하는 작업뿐만 아니라 외부로부터 전송되는 고해상도 영상을 출력하는 데에도 자주 사용되기 때문에 클러스터의 내부 노드(internal node)라고 해도 외부 네트워크로 연결하기 위한 방법이 있어야 한다.

노드 간 네트워크는 저가의 기가비트 이더넷부터 10G 이더넷, Quadrics, 인피니

밴드, 미리넷에 이르기까지 다양한 제품을 생각할 수 있다. 그러므로 tiled display를 구축할 때에는 용도와 예산 규모 등에 따라 적절한 솔루션을 적용해야 한다. Tile display에서 DMX와 Chromium 등을 주로 이용할 경우에는 네트워크 대역폭과 지연시간(latency)이 중요한 성능요인으로 작용하기 때문에 고성능 네트워크를 사용하는 것이 바람직하다. 하지만 CAVELib 기반 애플리케이션이나 SAGE와 같이 주로 외부로부터 비디오 스트림을 전송받는 애플리케이션을 사용할 경우에는 노드 사이의 데이터 교환은 거의 없고, 화면 사이의 동기화(synchronization)만 생각하면 되기 때문에 기가비트 이더넷으로 충분하다.

[표 2-1] 네트워크 사양 비교

	기가비트 이더넷	인피니밴드	Quadrics	10G 이더넷
최대 대역폭	1 Gbps	20 Gbps	10 Gbps	10 Gbps
드라이버 안정성	매우 안정	불안정	안정	안정
포트당 가격	매우 낮음	높음	매우 높음	매우 높음
Latency	나쁨	10ms 미만		
애플리케이션	CAVELib, 외부 A/V 스트림 등	병렬 렌더링, DMX, Chromium 등		

외부 네트워크는 계산 전용 시스템(특히 클러스터)에서는 찾아보기 어려운 구성 요소지만 tiled display와 같은 시스템에서는 필요한 경우가 종종 있다. 특히 외부로부터 A/V 스트림을 전송받을 경우에는 반드시 필요하게 된다. 보통 내부 노드가 외부 네트워크에 접속할 때에는 한 대의 게이트웨이 노드를 통해서 데이터를 전송하지만, tiled display에서는 모든 내부 노드가 독립적으로 외부 네트워크로 접근할 수 있도록 해서 필요한 경우 병렬 데이터 전송도 가능하도록 구축하는 것이 바람직하다.

나. 출력장치

Tiled display의 출력장치는 여러 대의 LCD 모니터를 평면이나 원형으로 연결해서 구성하거나, 프로젝터와 스크린을 이용해서 구축하기도 한다. 특히 프로젝터와 스크린을 이용해서 구축하는 tiled display는 이른바 wall screen이라고 불리는 시스템과 거의 차이가 없기 때문에 여기서 설명하는 내용 중 일부는 wall screen을 구축할 때 그대로 적용될 수 있다.

(1) LCD 모니터

LCD 모니터는 낮은 가격으로 매우 높은 해상도를 구현할 수 있기 때문에 tiled display의 출력장치로 가장 많이 사용되고 있다. LCD 모니터로 구성된 출력장치는 다음과 같은 특징을 갖고 있다.

- 장비 구축비용이 상대적으로 적게 소요되면서도 아주 높은 해상도를 구현할 수 있다.
- 모니터 사이의 색상보정, geometry 보정 등의 작업을 별도로 진행하지 않아도 일정 수준 이상의 image quality를 보장한다.
- 프로젝터로 구성하는 출력장치에 비해 공간소요가 적다.
- 규모 확장이 상대적으로 용이하다.
- 대부분의 경우 모니터의 프레임 (bezel)을 제거할 수 없기 때문에 이미지의 연속성을 보장하지 못한다.



[그림 2-2] LCD 모니터를 이용한 출력장치

(2) 프로젝터와 스크린

Tiled display는 앞에서 설명한 LCD 모니터 외에도 여러 대의 프로젝터와 스크린을 이용해서 출력장치를 구성하는 경우도 있다. 프로젝터를 이용한 출력장치는 다음과 같은 특징을 갖고 있다.

- 영상 사이의 간격이 존재하지 않기 때문에 이미지의 연속성을 보장할 수 있다.
- LCD 모니터에 비해 많은 비용이 소요되면서, 같은 가격에 구현할 수 있는 해상도는 낮다.
- 입체영상 구현이 용이하다.
- 저가형 프로젝터의 경우 색상보정, geometry 보정 등의 작업을 수행하기 어렵다.
- LCD 모니터에 비해 많은 공간을 차지한다.
- 한 번 크기가 결정되면 규모 확장이 쉽지 않다.



[그림 2-3] 프로젝터와 스크린을 이용한 출력장치(NCHC)

(3) 출력장치를 위한 구조물

출력장치를 구성하는 장비에 상관없이 모든 출력장치는 설치를 위한 별도의 구조물이 필요하다. LCD 모니터, 프로젝터 등 출력장치를 구성하는 장비에 따라서 다양한 형태의 구조물을 설계할 수 있는데, 장비의 무게를 가장 중요하게 생각해야 한다. 과거에는 별도로 주문제작을 해야 했지만 최근 이를 전문으로 제작해주는 회사들이 등장해서 사용자의 고민을 덜어주고 있다.



[그림 2-4] 출력장치를 위한 다양한 형태의 구조물

(4) 입체영상 지원

지금까지의 LCD 모니터는 입체영상을 지원하지 않았다. 하지만 varrier 등 입체 안경의 도움이 없어도 입체영상을 구현할 수 있는 기술이 개발되면서 아직은 소수에 불과하지만 입체영상을 지원하는 tiled display가 구축되고 있다.

다. 기타 장비

(1) 트래킹 장비

Tiled display를 사용할 때 사용자가 콘솔 데스크에서 키보드와 마우스를 사용하는 것보다는 - 비록 입체영상을 지원하지 않는 출력장치라고 해도 - 스크린 앞에서 트래킹 장비를 이용해서 애플리케이션의 실행을 제어하는 것이 더 편리한 것은 자명한 사실이다. 상용 visualization 소프트웨어들은 이미 다양한 종류의 트래킹 장비를 지원하고 있고, SAGE, TileViewer 등 실험실 차원에서 진행되는 프로젝트들도 조이스틱이나 트래킹 장비를 통해서 애플리케이션을 제어할 수 있다.

(2) 음향출력 장치

음향출력 장치는 화상회의와 같은 애플리케이션을 사용할 때 주로 사용된다. 요즘은 리눅스에서도 5.1채널 오디오 출력을 지원하기 때문에 사용자의 필요에 따라서 원하는 수준의 음향출력을 구현할 수 있다.

3. 소프트웨어

Tiled display는 소프트웨어 구성도 중요한 부분을 차지한다. LLNL은 이미 수년 전 고성능 visualization 시스템을 위한 소프트웨어 스택을 제안했는데, 오픈 소스 소프트웨어로 구성되었고, tiled display뿐만 아니라 CAVE나 wall screen 형태의 시스템에도 적용될 수 있기 때문에 대부분의 범용 visualization 시스템이 기본적으로 채택하고 있다. 새롭게 tiled display를 구축할 때에도 이 장에서 설명하는 소프트웨어 스택을 기본으로 해서 각자의 활용목적에 따라 변형을 주면 보다 빠른 시간 내에 안정적인 시스템을 구축할 수 있을 것이다.

가. OS / X-Windows

(1) 리눅스 배포판

Tiled display를 위한 OS는 MS-Windows를 사용하기도 하지만 거의 대부분 리눅스를 사용한다. 최신 리눅스 배포판들은 패키지가 세분화되었고 네트워크를 통한 패키지의 설치, 제거, 업그레이드 등을 지원하기 때문에 관리자/사용자의 요구에 따라서 필요한 패키지만을 선택해서 설치할 수 있다.

한편 마이크로소프트(Microsoft)에서도 Compute Cluster Server라는, 클러스터를 위한 OS를 내놓고 있는데 아직 전체적인 성능이 리눅스를 따라가지는 못하기 때문에 지금 당장 많이 사용되지는 않지만 마이크로소프트라는 회사를 생각해보면 지속적으로 그 동향을 살펴볼 필요는 있다.

(2) X-Windows의 설정

클러스터의 각 렌더링 노드는 사용자가 직접 로그인해서 작업하는 경우가 거의 없기 때문에 부팅할 때마다 자동으로 X-Window가 실행돼서 애플리케이션의 출력을 언제라도 보여줄 수 있도록 준비가 되어야 한다. 이를 위해서 렌더링 노드는 시스템이 부팅 될 때마다 X-Window가 실행되도록 했다.

```
# The default runlevel
id:5:initdefault:      # 부팅 시 X-Windows display manager 자동실행
(후략)
```

[그림 3-1] /etc/inittab의 내용

요즘 출시되는 대부분의 리눅스 배포판은 gnome이나 KDE를 기본 데스크탑으로 사용하기 때문에 위와 같이 설정하면 부팅할 때마다 gdm같은 X display manager가 실행된다. 하지만 디스플레이 매니저만 실행되는 것보다는 특정 계정으로 자동 로그인까지 하고, X-Windows의 access control을 해제(man xhost)해서 언제라도 필요한 애플리케이션을 띄울 수 있도록 설정해두는 것이 편리하다.

(3) Netfilter 설정

클러스터의 내부 노드가 외부와 직접 통신할 필요가 있을 경우 섬세하게 방화벽을 설정할 필요가 있다.

```
/sbin/iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
/sbin/iptables -A INPUT -s 127.0.0.0/8 -j ACCEPT      # Localhost
/sbin/iptables -A INPUT -s 10.1.2.0/24 -j ACCEPT      # Local network
/sbin/iptables -A INPUT -s 10.1.4.0/24 -j ACCEPT      # Local network
/sbin/iptables -A INPUT -s *.*.0.0/16 -j ACCEPT # KISTI
/sbin/iptables -A INPUT -j REJECT

/sbin/iptables -A FORWARD -j REJECT

/sbin/iptables -A OUTPUT -m state --state NEW,ESTABLISHED -j ACCEPT
```

[그림 3-2] 방화벽 설정 스크립트 예

나. Tiled display를 위한 미들웨어

미들웨어라는 호칭이 부적절할 수도 있겠으나, 여기서는 오픈 소스로 공개되었고, 클러스터나 tiled display에서 다른 애플리케이션을 실행하기 위한 기반이 되는 소프트웨어를 모두 미들웨어로 분류했다.

(1) DMX

DMX는 Distributed Multi-head X의 약자로, 서로 다른 컴퓨터에 연결된 독립된 출력장치들을 하나로 묶어서 단일 데스크톱 화면을 구성하기 위해 사용하는 프락시 X 서버다. 일반적인 X-Windows 시스템은 한 대의 컴퓨터에 연결된 다수의 출력장치를 지원하도록 구현되어 있으나, 여러 대의 독립적인 컴퓨터에 연결된 다수의 출력장치를 다루지는 못한다. DMX는 이 문제를 해결하기 위해서 만들어졌으며, <http://dmx.sourceforge.net>을 통해서 공개 소스로 배포되고 있다.

- DMX의 설치

거의 모든 리눅스 배포판이 DMX를 기본 패키지로 포함하기 때문에 특별한 경우가 아니고서는 소스를 가져와서 별도로 설치할 필요는 없다. 하지만 x86-64 전용으로 나온 일부 리눅스 배포판에서 Chromium과 DMX를 같이 사용하려면 DMX의 소스를 가져와서 별도의 설치작업을 진행해야 하는 경우도 있다. 이는 리눅스 배포판에서 제공하는 libdmx가 shared object로 되어있지 않으면 Chromium의 tilesort SPU가 libdmx를 제대로 링크하지 못하기 때문이다. 따라서 리눅스 배포판에서 정확히 어떤 형태의 libdmx를 제공하는지 확인할 필요가 있다. 예를 들어서 Fedora Core는 static library 형태의 libdmx(libdmx.a)를 제공하고, RedHat Enterprise Linux 5는 shared object 형태의 libdmx(libdmx.so)를 제공한다.

- DMX의 설정

[그림 3-3]은 KISTI 슈퍼컴퓨팅센터의 tiled display를 위한 DMX 설정 파일이다.

```
% cat /etc/dmx/dmx.conf
virtual OctaVision 5600x2100 {
    display octa01-ib0:0.0 1400x1050;
    display octa02-ib0:0.0 1400x1050 @1400x0;
    display octa03-ib0:0.0 1400x1050 @2800x0;
    ...
}
```

[그림 3-3] /etc/dmx/dmx.conf의 내용

- DMX의 실행

DMX의 설정이 모두 끝나면 [그림 3-4]와 같이 DMX를 실행한다. DMX의 실행 파일명은 Xdmx이며, 로그인 노드에서만 실행시키면 클러스터의 나머지 노드들을 제어할 수 있게 된다. 이 때 각 노드는 xhost를 이용해서 ACL을 해제(disable)시켜놓아야 한다.

```
Xdmx :1 # display 번호 할당
+xinerama # 전체 tile을 하나의 데스크톱으로 사용
-configfile /etc/dmx/dmx.conf # 설정 파일 지정
-input localhost:0 # 키보드/마우스 입력
-ac # disable access control
-br # 검정색으로 root window 설정
-ignorebadfontpaths # 잘못된 font path 무시
-norender & # 필요에 따라 사용
```

[그림 3-4] DMX의 실행

- X-Windows 애플리케이션의 실행

앞에서 설명한 방법으로 Xdmx가 정상적으로 실행되면 사용자는 X-Window 애플리케이션을 다음과 같은 방법으로 tiled display에 출력시킬 수 있다.

```
application -display :1

또는

export DISPLAY=:1 (bash 기준)
application
```

[그림 3-5] 애플리케이션의 실행 방법 1

DMX를 많이 사용하는 tiled display는 사용할 때는 높은 성능의 노드 간 네트워크를 설치해야 좋은 응답시간(response time)을 구현할 수 있다. DMX를 통해서 실행되고 있는 애플리케이션이 모니터(tile)의 경계선을 지나서 움직일 경우, 내부적으로는 해당 모니터를 담당하는 두 대 이상의 노드가 네트워크를 통해서 X-Windows 메시지를 지속적으로 교환하는데, 이는 생각보다 큰 부담으로 작용

한다. 특히 애플리케이션의 화면이 여러 대의 모니터에 걸쳐있고 그 내용이 지속적으로 바뀐다면 애플리케이션을 그려주는 노드 사이의 네트워크 통신이 계속 발생하기 때문에 노드 간 네트워크가 충분한 성능을 제공하지 않으면 응답속도가 느려진다.



[그림 3-6] tiled display를 이용한 애플리케이션의 실행

(2) Chromium

Chromium은 클러스터로 분산 렌더링 시스템을 구축할 수 있도록 해주는 소프트웨어다. 이 소프트웨어는 원래 미국 Stanford 대학교의 그래픽스 연구실에서 WireGL이라는 이름의 프로젝트로 시작했으나, 어느 정도 안정화가 이루어지면서 소스를 일반에게 공개¹⁾하고 명칭을 Chromium으로 변경했다.

Chromium은 일반 OpenGL 애플리케이션을 아무 수정 없이 tiled display로 출력할 수 있도록 해주는 역할을 한다. DMX는 일반 애플리케이션을 tiled display로 출력하는 것은 가능하지만 OpenGL 애플리케이션을 tiled display로 출력하는 것을 제대로 지원하지 못하는데, 이 문제를 Chromium을 이용해서 해결할 수 있다.

• Chromium의 설치

Chromium은 소스 파일 형태로만 배포되기 때문에 사용자가 직접 컴파일을 해야 하지만 UNIX의 기본 명령어만 알고 있으면 비교적 손쉽게 컴파일/설치가 가능하다. 다만, DMX와 Chromium이 연동되도록 하기 위해서 컴파일 전에 설정 파일을 수정해야 한다. Chromium의 소스파일을 가져와서 압축을 풀면 새로운 디렉터리가 생성되는데, 이 디렉터리 내에 있는 options.mk라는 파일의 내용 중 USE_DMX=0이라는 부분을 찾아서 [그림 3-7]과 같이 수정한 후에 컴파일을 진행한다.

1) <http://chromium.sourceforge.net>에서 다운받을 수 있다.

```
# Set USE_DMx to 1 if you want to enable DMX support.
# You'll need the libdmx.a library and dmxext.h header file in the usual X
# directories.
USE_DMx=1
```

[그림 3-7] options.mk의 수정

- Chromium의 실행

Chromium으로 OpenGL 애플리케이션을 실행할 때에는 렌더링 노드에 crserver를 띄우고, 애플리케이션 노드에 crappfaker를 일일이 띄워줘야 한다. 하지만 매번 애플리케이션을 실행할 때마다 이런 작업을 해야 한다면 매우 귀찮은 일이다. 그래서 Chromium은 crserver와 crappfaker 등을 자동으로 실행할 수 있는 방법을 제공한다[그림 3-8].

```
for tile in dmx:
    ...

    renderspu.Conf('render_to_app_window', 1)
    servernode = CRNetworkNode( host )
    servernode.AddTile( 0, 0, tile['width'], tile['height'] )

    ...

    # optimize_bucket is of no use w/ DMX.
    servernode.Conf('use_dmx', 1)
    cr.AddNode( servernode )
    if AUTOSTART:
        servernode.AutoStart( ["/usr/bin/rsh", host,
                                "/bin/sh -c 'DISPLAY=:0.0 LD_LIBRARY_
PATH=%s %s/crserver -motherhip %s:%d'" % (crlibdir, crbindir, local
Hostname, mothershipPort) ] )

    tilesortspu.AddServer( servernode, protocol='tcpip', port=server
Port )
```

[그림 3-8] DMX와 Chromium을 연계하면서 자동실행까지 담당하는 mothership 설정

4. 활용사례

이 장에서는 KISTI 슈퍼컴퓨팅센터에서 tiled display의 활용한 사례를 설명한다. KISTI 슈퍼컴퓨팅센터는 2004년부터 tiled display를 구축했으며, 국제시연 등 다양한 형태의 애플리케이션에 사용해왔다. 그리고 tiled display를 구축하면서 축적한 노하우(know-how)는 ‘차세대 가시화 시스템’의 구축에 적용되는 등 tiled display 관련 기술이 다양한 형태로 활용됐다.

가. COVISE (일반 가시화 애플리케이션의 실행)

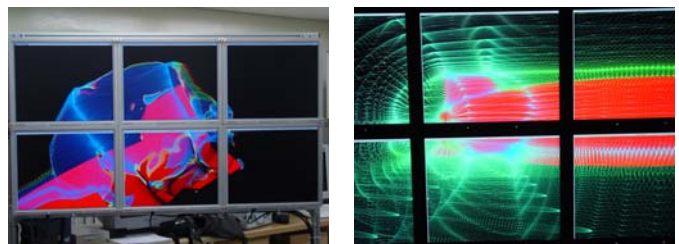
COVISE는 COLlaborative VISualization and SIMulation Environment의 약자로 독일 HLRS에서 개발한 가시화 소프트웨어이다. 가시화팀에서는 이 소프트웨어를 이용해서 응용과학자들이 생성한 데이터 가시화를 지원하고 있는데, 특히 몇몇 데이터에 대해서는 tiled display를 이용해서 성공적으로 가시화 작업을 수행할 수 있었다. 대상 데이터는 KTX(한국형 고속 열차)의 유동장 해석 결과로, 비행기의 속도에 따른 표면의 공기의 밀도의 변화를 보여주고 있다.



[그림 4-1] COVISE 사용 예

나. 유체 데이터의 가시화 (애플리케이션 개발)

가시화팀에서 생성한 데이터 중 비행선 주위의 유동장 해석 결과를 렌더링하는 데에 tiled display를 이용했다. 이 데이터는 비행선이 steady state로 있을 때 측풍이 불어오는 상황을 계산한 것으로, 57개의 timestep을 갖고 있는 240GB 크기의 curvilinear grid 볼륨 데이터이다. 이 실험의 특징은 Chromium이 아닌, 자체개발한 개발한 분산 렌더링 애플리케이션을 이용해서 렌더링을 수행했다는 점인데 QuadroFX 3000G의 frame lock 기능을 활용해서 tile 사이의 synchronization을 맞추

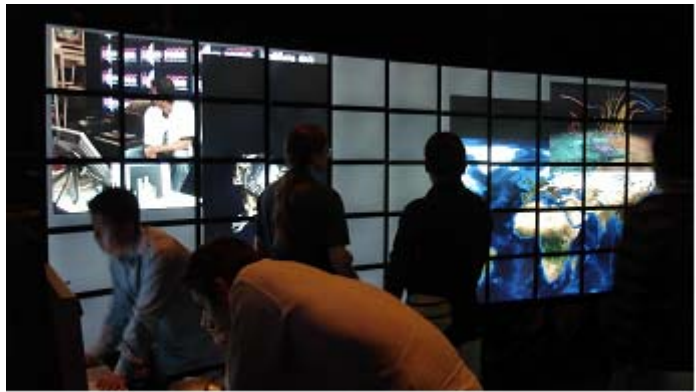


[그림 4-2] 비행선 유동장 데이터 가시화

고 있다. 현재는 다각형(polygon) 데이터만을 렌더링하고 있으나 향후 직접 볼륨 렌더링 기능을 추가할 예정이다.

다. 고해상도 화상회의

2005년 iGrid2005에서는 SAGE라는 애플리케이션을 이용해서 다수의 고해상도 영상을 전송하고, 그 결과를 1억 픽셀 해상도의 tiled display에 출력하는 시연이 있었다. 그 당시 iGrid2005 행사장(미국 샌디에이고)에는 55개의 LCD 모니터로 구성된 초고해상도 tiled display가 설치되어 있었으며, 네덜란드, 한국, 미국 시카고 등 전 세계에서 실시간으로 전송되는 고해상도 영상이 화면에 출력되었다. 이 시연을 통해서 향후 고성능 컴퓨팅 환경이 어떤 식으로 바뀔 것인지 간접적으로나마 예상할 수 있는 계기가 되었다.



[그림 4-3] iGrid2005 시연

5. 결론

Tiled display는 크게 클러스터 형태의 전용 컴퓨터와 LCD 모니터나 프로젝터로 구성된 출력장치로 구분할 수 있는데, 본 보고서는 tiled display를 구축하고자 하는 사람들에게 실질적인 도움이 될 수 있도록 전용 컴퓨터와 출력장치 각각을 구성할 때 고려해야 할 사항을 중심으로 설명했다.

Tiled display는 낮은 가격으로 높은 해상도를 구현할 수 있고, 확장성이 뛰어나기 때문에 세계의 우수 대학교 및 연구기관에서는 이미 일정수준 이상의 고성능 tiled display를 구축해서 활용하고 있으며, 앞으로 이러한 종류의 시스템은 더욱 많이 보급될 것이다. 앞으로 tiled display가 국내 연구진들에게 많이 보급돼서 손쉽게 대용량 데이터를 가공해서 고해상도 영상으로 접할 수 있을 것으로 기대한다.