

ISBN 978-89-5884-995-7 98560



특수 목적 클러스터의 벤치마크 방안

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. Visualization 시스템을 위한 벤치마크	2
가. 개별요소 벤치마크	3
1) CPU	3
2) GPU	3
3) 메모리 대역폭	4
4) 네트워크	4
5) 디스크 I/O	5
6) 프로젝터 / 스크린	5
7) 기타	6
나. 종합 벤치마크	7
1) CPU / 메모리 / 네트워크	7
2) 그래픽 / 네트워크	7
3) 디스크 I/O	8
4) Synchronization	9
5) 스트레스 테스트	9
다. 이슈	10
1) CPU와 GPU의 적정한 비율	10
3. 차세대 가시화 시스템의 벤치마크 기준	11
4. 벤치마크 결과	12
가. 노드 간 네트워크 (Interconnection network)	12
1) Infiniband 벤치마크 결과	13
2) Quadrics	15
나. GPU	17
5. 결론 및 향후계획	20

표 차례

[표 2-1] Visualization 시스템을 위한 벤치마크 분류 및 평가 프로그램	2
[표 2-2] Chromium의 PerfSPU에 대한 옵션	8
[표 3-1] 벤치마크 테스트 프로그램 및 내용	11
[표 4-1] NVIDIA QuadroFX5600의 SpecViewPerf 수행결과	17
[표 4-2] NVIDIA QuadroFX4400의 SPECViewPerf 수행결과	18
[표 4-3] NVIDIA GeForceFX8800GTX의 SPECViewPerf 수행결과	18
[표 4-4] NVIDIA GeForceFX7800GS의 SPECViewPerf 수행결과	19
[표 4-5] NVIDIA GeForceFX6800 Ultra의 SPECViewPerf 수행결과	19

그림 차례

[그림 4-1] Infiniband, blocking I/O, ping-pong	11
[그림 4-2] Infiniband, non-blocking I/O, ping-pong	11
[그림 4-3] Infiniband, blocking I/O, streaming	11
[그림 4-4] Infiniband, non-blocking I/O, streaming	11
[그림 4-5] Quadrics, blocking I/O, ping-pong	11
[그림 4-6] Quadrics, non-blocking I/O, ping-pong	11
[그림 4-7] Quadrics, blocking I/O, streaming	11
[그림 4-8] Quadrics, non-blocking I/O, ping-pong	11

1. 서론

컴퓨터의 벤치마크는 주어진 예산 한도 내에서 구매 가능한 시스템의 최적의 규모와 성능을 예측하거나, 설치 후의 실제 성능이나 안정성 등을 측정하기 위해 수행한다. CPU, 메모리 등 컴퓨터의 각 구성요소의 성능을 측정하기 위해 실행하는 벤치마크 프로그램은 이미 많이 개발됐고, 거의 대부분 인터넷을 통해 쉽게 구할 수 있다. 특히 몇몇 대표적인 벤치마크 프로그램은 오픈 소스 프로그램이면서도 충분히 높은 정확도와 안정성을 제공하기 때문에 일정부분 공식적인 성능 자료로 인정받기도 한다.

한편, 클러스터가 고성능 컴퓨터의 주요 아키텍처 중 하나로 자리 잡으면서 대형 visualization 시스템 또한 클러스터 형태로 구축되는 사례가 증가하고 있다. 일반적으로 visualization 시스템에는 GPU, 프로젝터, 스크린 등 계산 전용 시스템에서는 보기 어려운 구성요소가 추가되며, 대용량의 데이터를 처리하면서도 많은 작업들이 빠른 응답시간(response time) - 실시간 렌더링 - 을 요구하기 때문에 짧은 시간 내에 시스템 리소스(resource)를 대량으로 소비한다는 특징을 갖고 있다. 따라서 visualization 시스템의 벤치마크는 단순히 CPU, 메모리 등 개별 구성요소의 성능뿐만 아니라 시스템의 모든 리소스를 얼마나 효율적으로 동원할 수 있는지, interactive 작업을 얼마나 효과적으로 지원하는지에 대한 평가 방안도 제시해야 한다.

본 보고서에서는 최적의 visualization 시스템을 구축하기 위한 지표를 제공하기 위해 visualization 시스템에 적용할 수 있는 벤치마크 항목과 이에 해당하는 프로그램을 제안하고자 한다. 비록 첫 번째 보고서에서 모든 내용을 다룰 수는 없지만 이후의 지속적인 내용개선을 통해 visualization 시스템의 종합적인 성능평가방안과 시스템 구축 과정에서 주의해야 할 사항 등을 체계적으로 정리해 나가 고자 한다.

2. Visualization 시스템을 위한 벤치마크

여타의 고성능 컴퓨터와 마찬가지로 해외 주요 연구기관의 대형 visualization 시스템은 많은 경우 클러스터 형태로 구축되고 있다. 특히 고성능 visualization 시스템의 대명사로 여겨졌던, SGI의 NUMA 구조를 갖는 시스템도 더 이상 공식적으로 판매되지 않기 때문에 이제는 사실상 모든 visualization 시스템이 클러스터 (엄밀하게는 SMP 클러스터) 구조를 갖고 있다고 해도 과언이 아니다. 따라서 visualization 시스템의 벤치마크도 많은 부분은 다른 클러스터와 유사한 형태로 진행된다. 하지만 visualization 시스템은 계산 전용 클러스터와 달리 그래픽 성능이 중요한 성능요인으로 작용하고, 데이터 visualization 작업의 특성상 CPU, 메모리, 디스크, GPU 등 시스템의 모든 구성요소의 최대성능을 동시에 끌어내기 때문에 개별 구성요소의 성능뿐만 아니라 전체 시스템의 종합적인 성능과 각 구성요소 사이의 성능균형도 중요하게 생각해야 한다.

본 보고서에서는 visualization 시스템을 위한 벤치마크를 개별요소 벤치마크와 종합 벤치마크의 두 가지로 분류해서 제안한다. 개별요소 벤치마크는 CPU, 네트워크 등 시스템을 구성하는 각 구성 요소만의 성능을 측정하는 데에 초점을 맞추고, 종합성능 벤치마크는 두 개 이상의 구성 요소를 같이 사용할 때의 성능을 측정하는 데에 중점을 둔다.

[표 2-1] Visualization 시스템을 위한 벤치마크 분류 및 평가 프로그램

벤치마크 분류	주 평가 대상	평가 프로그램
개별요소 벤치마크	CPU	FFTE
	GPU	SPECViewPerf, Quake, Doom
	메모리 대역폭	STREAM
	네트워크	NetPerf, HPCBench, NetPIPE, ...
	디스크 I/O	IOR, Bonnie, IOZONE, ...
	프로젝터	N/A
	스크린	N/A
종합 벤치마크	CPU / 메모리 / 네트워크	HPL
	그래픽 / 네트워크	Chromium
	디스크 I/O	IOR, Bonnie, IOZONE, ...
	시스템 안정성	stress

가. 개별요소 벤치마크

Visualization 시스템 역시 궁극적으로는 ‘클러스터’이기 때문에 대부분의 벤치마크 항목과 비슷한 프로그램을 동일하게 사용한다.

개별요소 벤치마크의 경우 이미 많은 자료가 인터넷에 공개돼있고, 문서상의 사양만 보고도 실제 성능이 어느 정도까지 나오는지 비교적 정확하게 예측할 수 있기 때문에 종합성능 벤치마크보다 상대적으로 우선순위를 낮출 수 있다. 개별요소 벤치마크 분야는 CPU, GPU, 메인 메모리, 네트워크, 디스크, 오디오 등으로 나뉘어서 생각할 수 있다.

1) CPU

최신 CPU(코어)는 한 클럭에 최대 네 번의 수치 연산을 할 수 있기 때문에 클럭 정보만 알면 이론 최대 성능을 알 수 있고, 어느 정도까지는 HPL(LINPACK) 성능도 예측할 수 있다. 하지만 HPL 성능에는 네트워크 등 다른 구성요소도 벤치마크 결과에 영향을 주기 때문에 다른 종류의 벤치마크 프로그램을 이용하는 경우도 종종 있다.

○ **FFT** : FFT는 HPC Challenge Benchmark 중 하나로 Discrete Fourier Transform을 통해서 실수(알고리즘 상으로는 복소수) 연산 성능을 측정한다.

2) GPU

사실 PC에서는 3D 게임을 이용해서 GPU의 성능을 측정하는 경우가 매우 흔하다. 특히 최신 게임은 다양한 특수효과를 구현하기 위해 shader와 같은 최신 OpenGL 사양에서 볼 수 있는 기능을 사용하기 때문에 GPU의 성능을 최대한 끌어내고, 그 결과 GPU의 실제 성능을 가장 잘 보여주는 프로그램으로 자리매김하고 있다.

하지만 scientific visualization 애플리케이션은 복잡한 shader를 사용하는 경우가 많지 않기 때문에 여전히 고전적인 polygon 처리능력을 성능측정의 주 기준으로 생각해도 무리가 없는 것이 사실이다. 하지만, 향후 scientific visualization 애플리케이션은 3D texture, shader 등 보다 개선된 기능을 사용해서 성능과 image

quality를 개선할 가능성이 높으므로 texture 성능, GPU 자체의 계산능력 등에 대한 평가도 염두에 두어야 한다.

- **Doom 3 / Quake 4** : 꽤 많은 종류의 3D 게임이 출시되었음에도 불구하고 이 두 게임을 굳이 언급하는 이유는 두 게임 모두 리눅스에서 실행이 가능하기 때문이다.
- **SPECViewPerf** : SPEC에서 제안하는 벤치마크 프로그램으로 다양한 애플리케이션에서 만들어낸 데이터를 여러 가지 방법으로 렌더링하면서 그 성능을 측정한다.
- **GPUbench** : Programmable GPU의 계산 성능을 측정하기 위한 일련의 프로그램을 담고 있다. 이 프로그램은 MS-Windows 계열 OS에서만 수행할 수 있다는 단점이 있다.

3) 메모리 대역폭

메모리 대역폭은 크게 두 가지로 나뉘어서 생각할 수 있다. 하나는 시스템의 메인 메모리 대역폭이고 다른 하나는 그래픽 메모리 대역폭이다.

- **메인 메모리 대역폭** : 메인 메모리 대역폭은 STREAM으로 수행한다. 이는 사실상의 표준 벤치마크 프로그램으로 자리 잡고 있기 때문에 이 프로그램을 채택하는 데에는 큰 무리가 없다.
- **그래픽 메모리 대역폭** : 그래픽 메모리 자체의 대역폭 또한 무시할 수 없는 성능요소 중의 하나다. 하지만 아직 리눅스에서 그래픽 메모리의 대역폭을 측정할 수 있는 벤치마크 프로그램은 존재하지 않는다.

4) 네트워크

여기서의 네트워크는 1:1 네트워크 대역폭과 지연시간(latency)을 중심으로 측정하며, 특히 한 노드 내부에서의 네트워크 성능, 두 노드 사이의 네트워크 성능, 다수의 노드를 이용한 네트워크 성능 등을 모두 측정할 수 있어야 한다. 네트워크 벤치마크에 사용할 수 있는 프로그램은 다음과 같다.

-
- **Netperf** : 매우 잘 알려져 있는 벤치마크 프로그램이다. TCP, UDP, UNIX domain socket 등 다양한 종류의 네트워크를 테스트할 수 있고, IPv6까지 지원한다.
 - **HPCbench** : HPCbench는 리눅스 기반의 네트워크 벤치마크 프로그램이다. 여타의 네트워크 벤치마크 프로그램과 마찬가지로 버퍼와 메시지 크기를 다양하게 설정해서 TCP, UDP, MPI 통신성을 각각 측정할 수 있다. 그리고 네트워크 성능측정에 더해서 네트워크 I/O를 수행하는 동안의 CPU와 메모리 사용량, 인터럽트 현황, 스와핑(swapping), 페이징, context 스위치, 네트워크 카드 통계 등 다양한 자료를 보여준다는 장점이 있다. 측정결과는 바로 GNU Plot을 이용해서 이미지 파일로 저장할 수 있다.

5) 디스크 I/O

클러스터를 구축할 때 단일 노드에 장착된 하드디스크의 I/O 성능은 크게 중요하지 않지만, 클러스터의 모든 노드들이 동시에 접근할 수 있는 대용량 저장장치에 대해서는 수 GB/s 수준의 throughput이 요구되기도 한다. 특히 time-varying 데이터를 처리하는 경우, 1초에 수십 GB 이상의 데이터를 각 노드에 보내줘야 할 정도로 높은 I/O 성능을 필요로 한다.

- **IOR** : LLNL에서 제안한 디스크 벤치마크 프로그램이다. MPI를 이용해서 동시에 여러 노드가 디스크 입/출력을 수행하도록 제어할 수 있다.

6) 프로젝터 / 스크린

프로젝터/스크린의 성능은 image quality와 가장 밀접한 관계를 갖고 있다. 하지만 image quality는 해상도, 픽셀 사이의 간격, 픽셀의 크기 등 정량적으로 측정할 수 있는 부분도 있지만 '색감'과 같이 다분히 주관적인 평가도 반영해야 한다. 게다가 동일한 프로젝터라고 해도 같이 사용하는 스크린의 특성에 따라 광학적인 성질이 서로 다르게 나타날 수 있기 때문에 정형화된 평가 방안을 제시하는 것이 상당히 어렵다.

7) 기타

여기서는 정량적으로 그 성능을 평가할 수는 없지만 어떤 식으로든 평가가 필요한 항목들에 대해 정리한다.

- **PCI express 슬롯 대역폭** : Visualization 시스템은 마지막에는 데이터를 GPU의 그래픽 메모리에 올려놓고 처리하기 때문에 시스템의 메인 메모리와 PCI express 슬롯 사이의 대역폭 또한 성능에 큰 영향을 주고 있다. 2007년 11월 현재 메인보드의 FSB는 1333MHz의 clock으로 작동하며, PCI express 16배속 슬롯은 이론적으로 8GB의 대역폭을 갖고 있는 것으로 돼있으나, 실제로 PCI express의 root complex 구성에 따라서 많은 차이를 보여줄 수 있다. 특히 NVIDIA의 SLI, Quad-SLI와 AMD(ATI)의 CrossFire와 같이 단일 메인보드에 두 개 이상의 GPU를 장착/운용할 수 있는 기술이 나오면서 PCI express 슬롯의 대역폭도 전체 visualization 시스템의 성능의 중요한 부분을 차지하고 있다.
- **5.1채널 오디오** : 가상현실 애플리케이션 등을 수행할 때는 오디오 출력도 매우 중요한 요소가 되기 때문에 가상화 시스템을 구성할 때 양질의 오디오 출력을 지원할 수 있어야 한다. 오디오 테스트는 단순히 지원/미지원 여부만 측정하는 것으로 충분하며, 나머지는 음향기기 구성에 따라 달라진다.

나. 종합 벤치마크

앞에서 설명했듯이 종합 벤치마크는 두 개 이상의 시스템 구성요소를 같이 사용하거나 하나의 구성요소 여러 개를 동시에 사용할 때의 성능을 측정하는 데에 중점을 둔다. 사실 개별요소는 최신 하드웨어에 대해서도 그 성능자료를 인터넷에서 바로 찾아낼 수 있지만 전체 시스템의 구성요소를 활용하는 벤치마크는 그 평가기준을 정하기도 어렵고, 실제로 그 기준에 따라서 성능을 평가한 내용도 별로 없는 것이 사실이다. 지금도 visualization 시스템은 일반 계산 시스템과 동일한 Top500에 등록되는 것이 현실이다.

1) CPU / 메모리 / 네트워크

CPU, 메모리, 네트워크를 동시에 사용하는 벤치마크는 LINPACK이 가장 널리 사용된다. LINPACK 성능 자체가 시스템의 절대 성능을 보여주는 것은 아니지만 일반적으로 잘 튜닝 된 클러스터는 LINPACK 성능이 이론 성능의 80% 이상을 실현한다.

2) 그래픽 / 네트워크

Visualization 시스템을 클러스터로 구축하는 큰 이유 중 하나가 필요에 따라서 그래픽 처리 성능을 향상시키기 위한 것으로 볼 수 있다. 이 때 사용하는 대표적인 프로그램이 Chromium인데 이 프로그램이 제공하는 PerfSPU를 이용해서 각종 통계 데이터를 수집할 수 있다.

[표 2-2] Chromium의 PerfSPU에 대한 옵션

Parameter name	Value types	Description
dump_on_finish	Boolean	Will cause performance data to be 'dumped' when glFinish is called. Useful if SwapBuffers is not used. Default 0.
dump_on_flush	Boolean	Will cause performance data to be 'dumped' when glFlush is called. Useful if SwapBuffers is not used. Default 0.
dump_on_swap_count	Integer	Trigger when data is to be 'dumped' on hitting this amount of SwapBuffer calls. Default 0.
token	String	Mark 'interesting' performance data with tokens. Default is 'tab'.
log_separator	String	Field separator for logged data. Default is 'tab'.
mothership_log	Boolean	Send performance data to a mothership log file. Set CR_PERF_MOTHERSHIP_LOGFILE to a file on the mothership for data collection. This allows for gathering of data from multiple nodes. Default 0.
token	String	Mark 'interesting' performance data with tokens. Default is 'tab'.

3) 디스크 I/O

Visualization 작업도 디스크에 저장된 데이터를 메인 메모리로 읽어 들인 후 후 처리나 렌더링을 수행한다는 점에서는 여타의 계산 작업과 큰 차이가 나지 않는다. 하지만 time varying 데이터의 실시간 렌더링이나 interactive time 렌더링을 전제로 한다면 상황은 많이 달라진다. 특히 용량이 큰 time varying 데이터는 한 타입스텝이 수백 MB ~ 30GB이상의 크기를 갖는데, 이런 데이터를 실시간으로 렌더링하기 위해서는 디스크 I/O 성능이 초당 900GB라는, 터무니없이 큰 대역폭을 필요로 한다. 이 수준의 대역폭을 구현하는 것은 결코 쉬운 일이 아니므로 주어진 예산이나 애플리케이션의 특성 등을 고려해서 적절한 수준의 디스크 I/O 성능을 구현하는 것이 중요하다.

4) Synchronization

많은 visualization 시스템이 두 대 이상의 프로젝터나 모니터를 출력장치로 갖고 있기 때문에 애플리케이션을 실행할 때 각 출력장치 사이의 동기화(synchronization) 문제도 중요하다. 이 문제를 성능 문제로 보기는 어렵지만 시스템의 활용성 등의 기준으로 생각해서 평가대상 중 하나로 제안하는 데에는 무리가 없을 것이다.

5) 스트레스 테스트

이 테스트는 클러스터 노드의 모든 구성요소(CPU, GPU, 메모리, 디스크 등)를 동시에 사용해서 일정 수준 이상의 부하를 지속적으로 발생시키는 시험으로 visualization 시스템뿐만 아니라 모든 종류의 컴퓨터에 공통적으로 적용할 수 있다. 이다. 24시간, 또는 48시간동안 쉬지 않고 스트레스 테스트를 수행함으로써 출시 상태가 불량한 제품을 조기에 가려낼 수 있을 것으로 기대한다.

다. 이슈

1) CPU와 GPU의 적절한 비율

한 대의 visualization 시스템을 구성할 때 가장 문제가 되는 것이 CPU와 GPU의 비율을 어떻게 해야 가장 균형 잡힌 시스템으로 구성하느냐 하는 것이다. 특히 대부분의 visualization 애플리케이션이 내부적으로 'CPU 제어 ⇒ GPU 실행'의 형태로 구현되고 있고, CPU(쿼드 코어 기준)의 성능과 GPU의 성능이 현재 1 : 10 이상 벌어져있기 때문에 적절한 CPU와 GPU의 비율은 시스템의 가격, 전체적인 애플리케이션 실행 등을 좌우하는 문제로 볼 수 있다. 이 문제에 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.

3. 차세대 가시화 시스템의 벤치마크 기준

이 장에서는 KISTI 슈퍼컴퓨팅센터가 ‘차세대 가시화 시스템’을 도입할 당시 적용했던 벤치마크 기준에 대해 설명한다. KISTI 슈퍼컴퓨팅센터가 지난 6월 ‘차세대 가시화 시스템’ 도입을 위한 제안요청서를 발표했다. 당시의 벤치마크 프로그램은 [표 3-1]과 같이 지정했다. 도입 프로젝트를 진행할 당시 시간이 매우 촉박했기 때문에 제안요청서를 진행할 당시의 벤치마크 프로그램이 본 보고서에서 제시하는 내용과 많은 차이가 있다.

[표 3-1] 벤치마크 테스트 프로그램 및 내용

구 분		프로그램 명	평가 내용
요소 성능	CPU	FFTE	○ Fast Fourier Transformation
	GPU	SpecViewPerf	○ OpenGL 일반성능 측정
	메모리	STREAM	○ 메모리 bandwidth 측정
	네트워크	NetPIPE	○ 다양한 프로토콜에 대한 네트워크 성능 측정 ○ 고속 네트워크 성능 시험에 사용
종합성능	LINPACK	HPL	○ 대형 연립방정식 해석 성능 ○ 렌더링/계산 네트워크 성능 시험에 사용
	디스크 I/O	IOR	○ 다양한 파일크기 및 전송크기에 대해 다중작업을 통한 디스크 입출력 성능 측정 ○ 파일 I/O 네트워크와 스토리지 성능 동시 측정
설치 후 벤치마크	스트레스	stress	○ 컴퓨터 시스템의 각 구성요소에 대해 제어 가능한 수준의 부하 발생 ○ 시스템 안정성 테스트
	LINPACK	HPL	○ 대형 연립방정식 해석 성능 ○ 렌더링/계산 네트워크 성능 시험에 사용
	디스크 I/O	IOR	○ 다양한 파일크기 및 전송크기에 대해 다중작업을 통한 디스크 입출력 성능 측정 ○ 파일 I/O 네트워크와 스토리지 성능 동시 측정

4. 벤치마크 결과

이 장에서는 KISTI 슈퍼컴퓨팅센터의 ‘차세대 가시화 시스템’ 도입 과정에서 진행했던 사전 벤치마크 중 노드 간 네트워크와 GPU에 대한 벤치마크 결과를 소개한다.

가. 노드 간 네트워크 (Interconnection network)

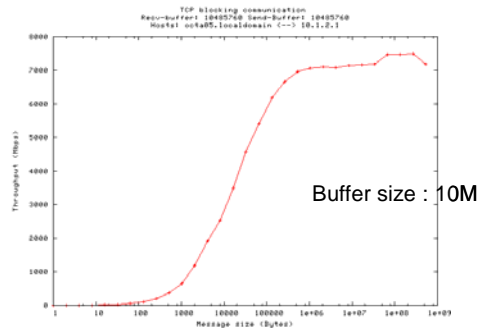
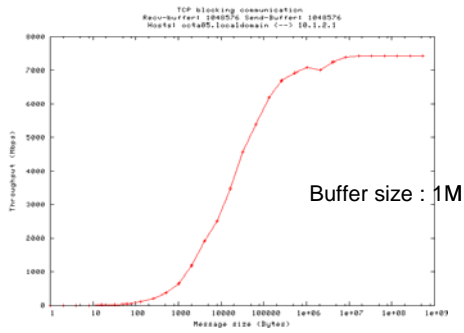
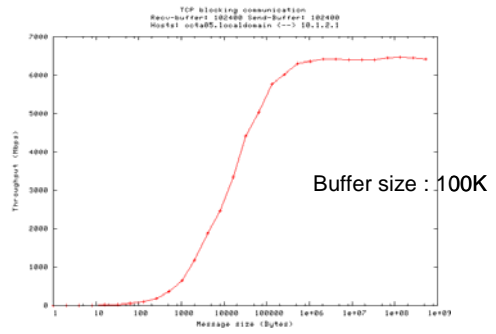
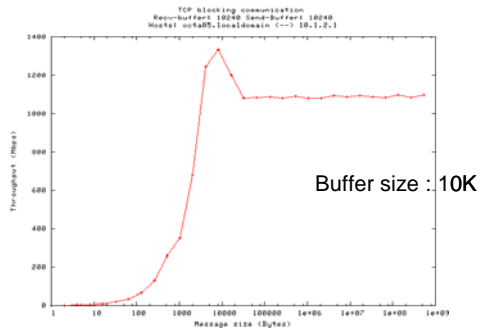
‘차세대 가시화 시스템’의 노드 간 네트워크로는 높은 대역폭과 낮은 지연시간(latency)을 갖는 제품을 사용하기로 했다. 이 때 검토했던 제품들은 10Gbps 이더넷, 인피니밴드, 퀴드릭스, 미리넷 등이었는데, 이 중 미리넷과 10Gbps 이더넷은 스위치 가격이 높아서 제외됐고, 20Gbps 수준의 대역폭을 갖는 인피니밴드와 낮은 지연시간과 비교적 안정적인 디바이스 드라이버를 제공하는 퀴드릭스가 최종 후보로 선정됐다.

퀴드릭스와 인피니밴드 모두 고유의 네트워크 프로토콜을 유지하고 있고, 다른 소프트웨어와의 호환성을 위해 OSI 7 layer 중 network layer의 IP 프로토콜을 지원하는 별도의 커널 모듈을 제공한다. 따라서 고유의 네트워크 프로토콜을 사용하지 않는 만큼 성능이 낮게 나온다는 단점이 있지만 일반적인 TCP/IP 프로토콜을 사용하는 애플리케이션도 실행할 수 있도록 해준다는 장점을 갖고 있다.

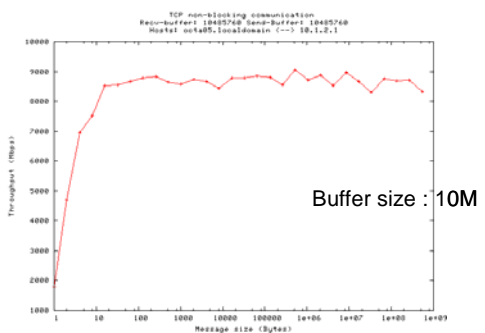
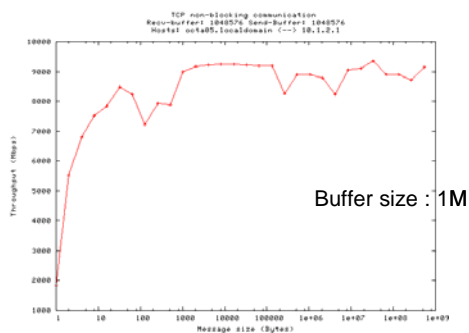
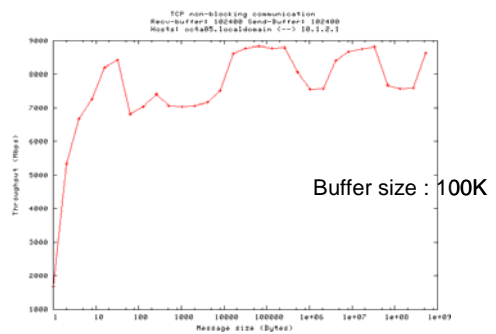
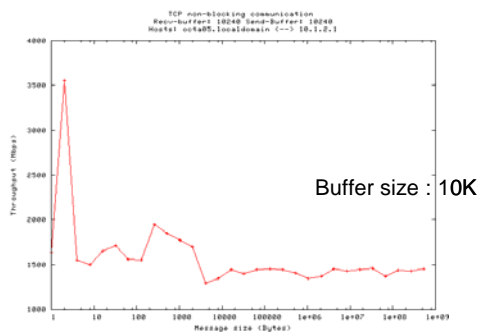
벤치마크 프로그램은 HPCbench를 이용했으며, 각 네트워크 장비에 대해 TCP 프로토콜의 blocking I/O, non-blocking I/O와 ping-pong, 1-way streaming을 조합해서 성능을 측정했다.

1) Infiniband 벤치마크 결과

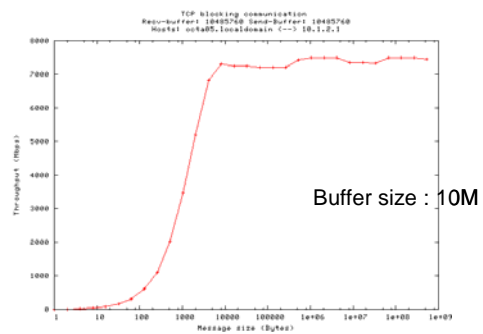
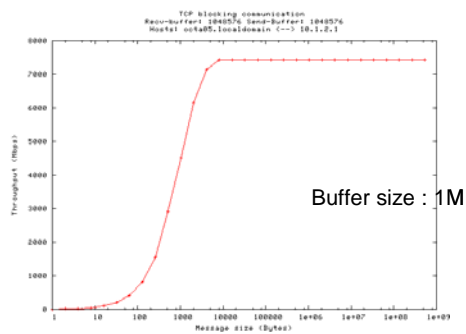
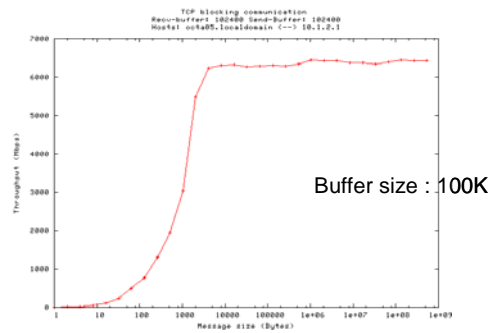
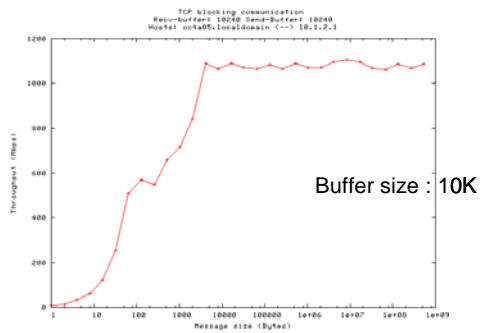
○ Blocking I/O, ping-pong 성능



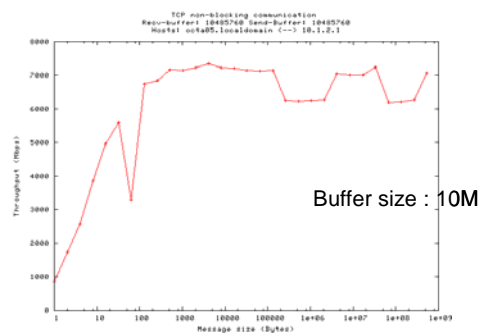
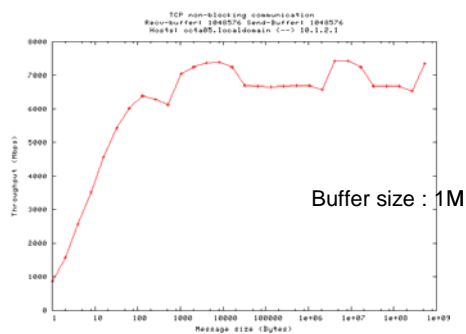
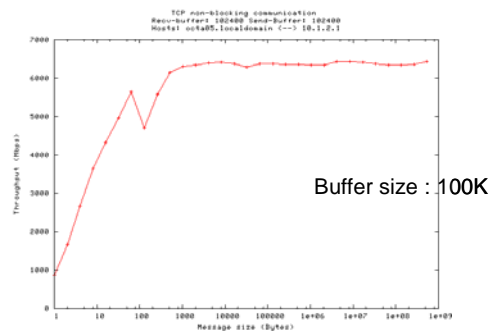
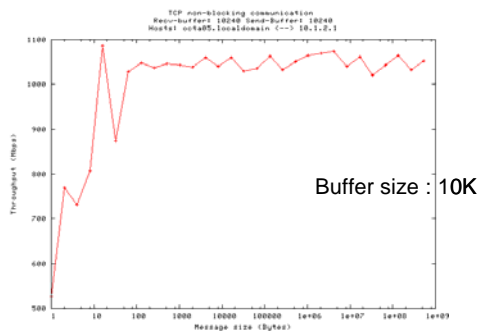
○ Non-blocking I/O, ping-pong 성능



○ Blocking I/O, streaming 성능

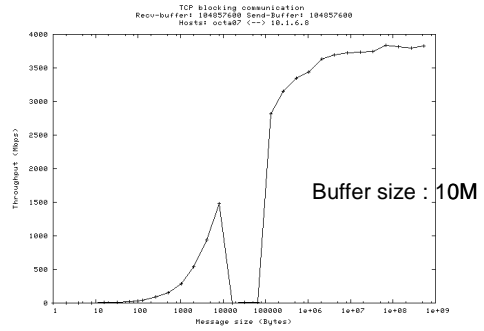
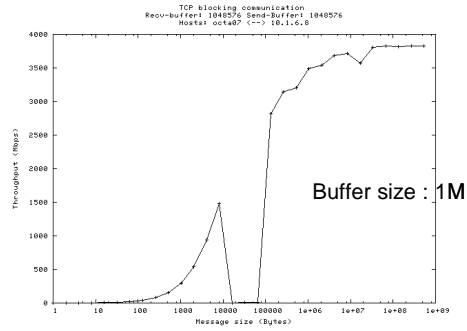
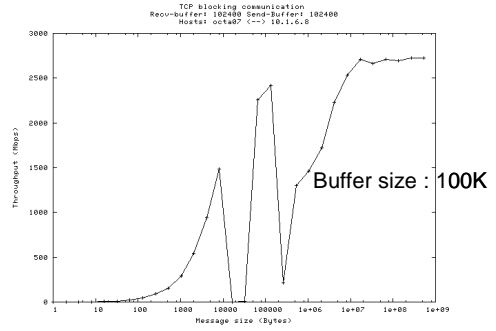
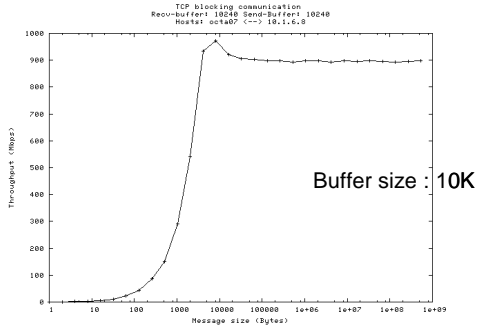


○ Non-blocking I/O, streaming 성능

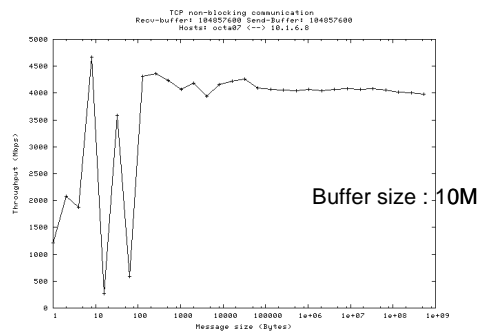
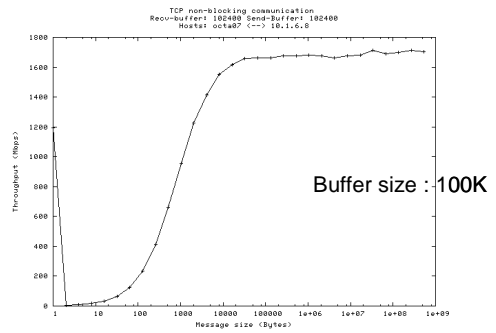
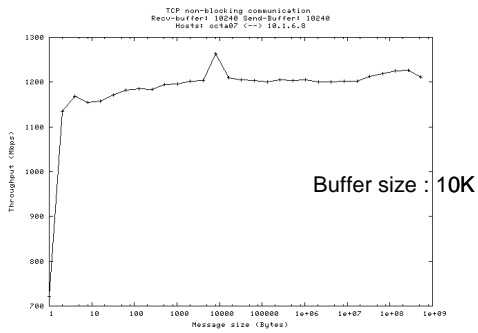


2) Quadrics

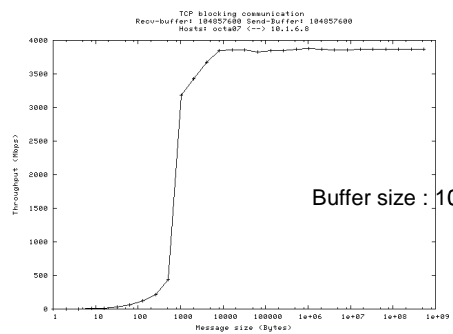
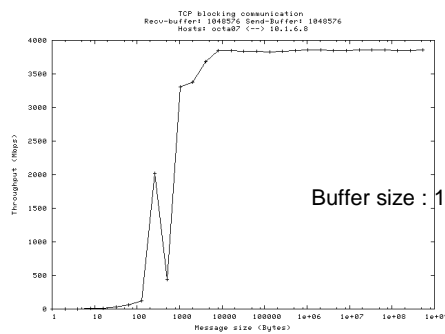
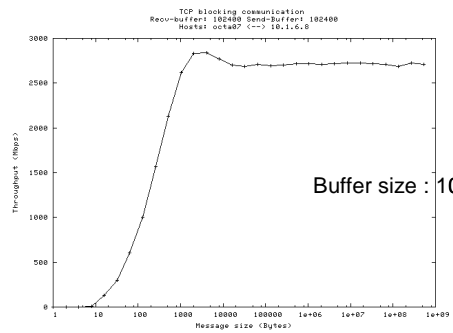
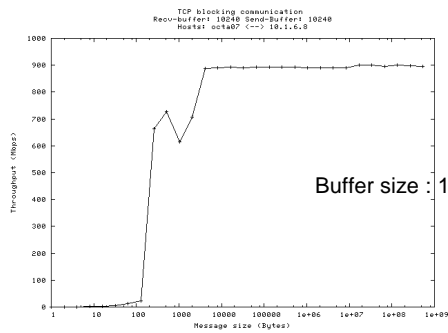
○ Blocking I/O, ping-pong 성능



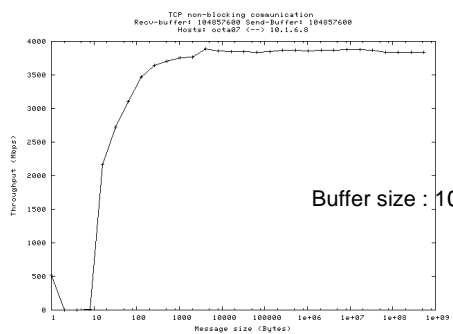
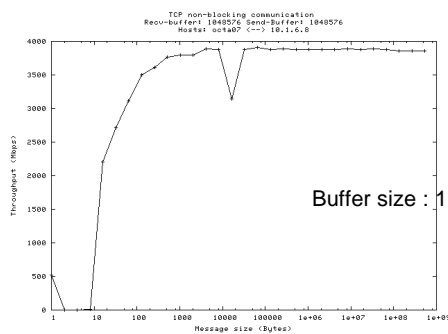
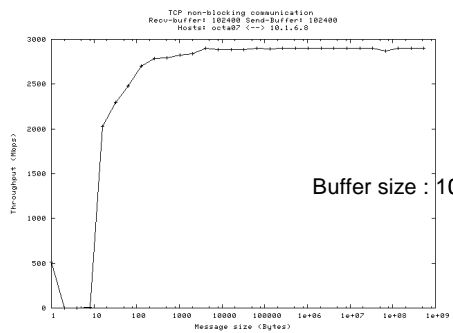
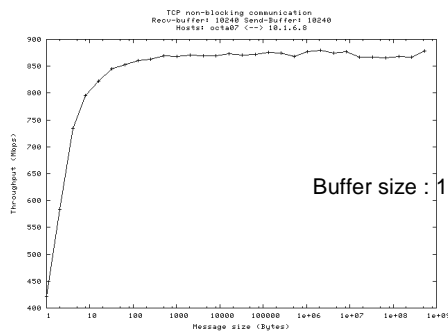
○ Non-blocking I/O, ping-pong 성능



○ Blocking I/O, streaming 성능



○ Non-blocking I/O, streaming 성능



나. GPU

GPU의 벤치마크에는 SPECviewperf를 이용했다. 사실 ‘차세대 가시화 시스템’ 도입 프로젝트 진행 당시 GPU는 가장 높은 성능을 갖는 모델로 이미 지정한 상태였기 때문에 서로 다른 종류의 벤치마크를 실행해서 성능을 비교하는 것 자체가 별다른 의미가 없는 상황이었다. 따라서 다른 종류의 GPU와 비교했을 때 우리가 도입하고자 하는 GPU가 어느 정도의 성능을 갖고 있는지 측정하는 것만으로 충분했다.

SPECViewPerf를 실행할 당시 모든 출력 윈도우의 해상도를 1024x768로 고정했고, 각각의 벤치마크를 5번 실행한 후의 산술평균값을 산정했으며, 최대 값과 최소 값에 대한 고려는 별도로 하지 않았다.

○ **QuadroFX5600** : ‘차세대 가시화 시스템’에서 채택한 GPU로 2007년 6월 현재 가장 높은 OpenGL 성능을 보여주고 있다.

[표 4-1] NVIDIA QuadroFX5600의 SpecViewPerf 수행결과

Dataset	Weighted geometric mean					AVG
	1	2	3	4	5	
3DS Max	16.83	17.77	17.74	17.71	17.75	17.56
CATIA	19.74	20.54	20.59	20.55	20.51	20.38
EnSight	33.62	33.63	33.62	33.69	33.80	33.67
Lightscape	22.40	22.97	22.87	22.93	22.82	22.79
MAYA	35.82	37.42	37.04	37.28	37.22	36.95
Pro/ENGINEER	13.50	14.22	14.20	14.21	14.23	14.07
SolidWorks	21.45	22.18	22.16	22.27	22.20	22.05
UGS NX	30.85	30.86	30.86	30.85	30.86	30.85
Teamcenter Visualization	8.716	8.732	8.714	8.709	8.711	8.716

○ **QuadroFX4400** : KISTI 슈퍼컴퓨팅센터에서 운영하고 있는 tiled display용 클러스터에 장착된 GPU로 QuadroFX5600의 이전 세대 GPU에 해당된다.

[표 4-2] NVIDIA QuadroFX4400의 SPECViewPerf 수행결과

Dataset	Weighted geometric mean					
	1	2	3	4	5	AVG
3DS Max	12.45	12.52	12.71	12.37	12.65	12.54
CATIA	15.67	15.72	15.88	15.54	15.90	15.74
EnSight	12.75	12.80	12.86	12.75	12.84	12.80
Lightscape	19.56	19.58	19.22	19.70	19.35	19.48
MAYA	21.30	21.86	22.30	21.17	21.77	21.68
Pro/ENGINEER	10.17	10.27	10.34	10.17	10.28	10.24
SolidWorks	13.95	14.25	14.35	13.93	14.20	14.13
UGS NX	12.01	12.01	12.01	12.01	12.01	12.01
Teamcenter Visualization	4.837	4.897	4.845	4.820	4.893	4.858

○ **GeForceFX8800GTX** : NVIDIA에서 데스크톱 PC용으로 출시한 GPU 중 가장 높은 성능을 갖고 있는 모델 중 하나다.

[표 4-3] NVIDIA GeForceFX8800GTX의 SPECViewPerf 수행결과

Dataset	Weighted geometric mean					
	1	2	3	4	5	AVG
3DS Max	13.07	14.46	13.79	15.05	14.97	14.26
CATIA	17.43	15.87	17.35	17.39	17.06	17.02
EnSight	21.09	21.14	21.08	21.17	20.99	21.09
Lightscape	14.68	14.66	14.60	14.43	14.62	14.59
MAYA	39.02	38.44	38.57	38.60	38.60	38.64
Pro/ENGINEER	12.33	12.27	12.03	12.24	12.22	12.21
SolidWorks	22.02	22.04	22.06	21.98	21.90	22.00
UGS NX	5.365	5.369	5.366	5.363	5.367	5.366
Teamcenter Visualization	6.471	6.442	6.442	6.445	6.442	6.448

○ GeForceFX7800GS

[표 4-4] NVIDIA GeForceFX7800GS의 SPECViewPerf 수행결과

Dataset	Weighted geometric mean					
	1	2	3	4	5	AVG
3DS Max	6.251	6.281	6.360	6.303	6.386	6.316
CATIA	6.766	6.906	6.885	6.947	6.990	6.899
EnSight	6.388	6.394	6.371	5.954	6.370	6.295
Lightscape	5.271	5.262	5.387	5.287	5.393	5.320
MAYA	19.54	19.52	19.55	19.56	19.54	19.54
Pro/ENGINEER	5.889	5.961	5.960	5.879	5.916	5.921
SolidWorks	10.84	10.68	10.79	10.78	10.76	10.77
UGS NX	2.037	2.041	2.037	2.037	2.034	2.037
Teamcenter Visualization	2.563	2.569	2.569	2.569	2.569	2.568

○ GeForceFX6800 Ultra

[표 4-5] NVIDIA GeForceFX6800 Ultra의 SPECViewPerf 수행결과

Dataset	Weighted geometric mean					
	1	2	3	4	5	AVG
3DS Max	5.796	5.698	5.725	5.876	5.956	5.810
CATIA	7.139	7.123	6.918	7.063	7.079	7.064
EnSight	3.779	3.834	3.859	3.842	3.821	3.827
Lightscape	5.695	5.507	4.794	4.908	5.444	5.269
MAYA	18.90	18.70	18.76	18.67	18.72	18.75
Pro/ENGINEER	6.037	5.987	5.990	6.002	6.024	6.008
SolidWorks	9.664	9.604	9.620	9.616	9.584	9.617
UGS NX	1.337	1.353	1.352	1.350	1.354	1.349
Teamcenter Visualization	2.033	2.031	2.035	2.031	2.035	2.033

5. 결론 및 향후계획

Visualization 시스템과 같은 특수목적 시스템은 하드웨어 구성, 애플리케이션의 실행형태 등이 계산 시스템과 다르기 때문에 최적의 시스템 구성을 위해서는 일반적인 계산 시스템의 벤치마크에 더해 visualization 시스템만의 벤치마크 방안도 마련되어야 한다. 본 보고서에서는 visualization 시스템을 구축할 때 수행할 수 있는 벤치마크를 개별요소 벤치마크와 종합 벤치마크로 나누어서 제시하고, 일부분이지만 KISTI 슈퍼컴퓨팅센터가 ‘차세대 가시화 시스템’ 도입 프로젝트를 진행하면서 수행했던 벤치마크의 종류와 그 결과를 일부 소개했다.

벤치마크는 대상 시스템의 구성요소 각각에 대한 성능을 측정하는 것도 중요하지만, 실제 애플리케이션이 단일 구성요소만 사용하는 경우는 존재하지 않기 때문에 서로 다른 종류의 구성요소를 동시에 사용할 때의 성능을 측정하는 것이 실 성능을 더 정확하게 반영할 수 있다. 하지만 visualization 시스템은 인간의 ‘시각’을 대상으로 한다는 점에서 객관적인 성능측정을 할 수 없는 항목들이 존재한다. 따라서 향후 보고서에서는 정량적으로 평가가 가능한 부분에 대해서는 보다 세분화된 평가 항목을 제시하고, 객관적인 성능측정이 어려운 항목에 대해 보다 체계적인 평가 방안을 제안하고자 한다.