

그래픽 서브시스템 최적화를 통한 CAD 시스템 성능 개선에 관한 연구

김희중⁺, 정재현⁺⁺

Performance Tuning of CAD Application by optimizing Graphics Subsystem

Kim Heui Jung⁺, Jeong Jae Hyun⁺⁺

Key words : Graphics Subsystem, OpenGL 3D-API, SPECViewperf, SPECapc for Pro/E 2001

1. 서론

본 연구에서는 3D CAD 어플리케이션을 운용하는 워크스테이션을 위한 퍼포먼스 튜닝 키트 제작에 있어 하드웨어 및 소프트웨어 환경에 따른 OpenGL 가속 그래픽 서브시스템의 최적 운용 환경 개선에 관한 연구를 수행하였다.

OpenGL은 그래픽 가속 프레임 버퍼를 내장한 그래픽 서브시스템을 필요로 하며, 그래픽 서브시스템의 OpenGL 가속 제어는 그래픽스 드라이버와 어플리케이션에서 설정되며, 어플리케이션과 드라이버의 OpenGL 가속 성능은 별개의 범위로 다뤄진다.

그래픽 서브시스템의 드라이버 및 제어 요소 설정에 대한 OpenGL 및 그래픽스 어플리케이션 성능 검증은 SPECopc의 SPECViewperf 8.1과 SEPCapc의 SPECapc for Pro/E 2001을 사용하였으며, SPEC 기준을 충족하는 환경에서 진행되었다^{1,2}. 사용된 그래픽 서브시스템은 nVidia의 Quadro4 GPU를 탑재한 ELSA의 GLoria4 750XGL이다.

2. 하드웨어 성능 분석

2.1 하드웨어 병목 현상 분석

그래픽 서브시스템의 제어 요소는 OpenGL 그래픽 서브시스템이 개선으로 설정 항목은 증가하고 있다. 때문에 운용하는 그래픽스 어플리케이션에 대한 설정 사항도 비례하여 증가할 수 있다. OpenGL 그래픽 서브시스템의 구조를 Fig. 1과 같이 정의하였다. OpenGL 그래픽 서브시스템 최적화는 그래픽스 파이프라인의 병목 현상 문제를 어플리케이션 및 드라이버 운용에 있어 각 병목현상의 발생을 최소화하는 것으로 볼 수 있다.

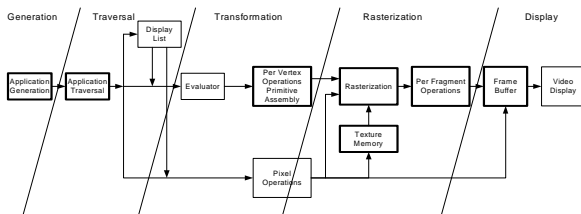


Fig. 1 OpenGL Graphics Pipelines

1) 지오메트리 변환 단계 : 그래픽스 데이터를 윈도우 공간

에 투영하는 과정은 많은 정점과 반복 계산으로 병목 현상을 발생하게 되므로, 폴리곤 렌더링 가속이 요구된다. 하지만 특정 하드웨어 고정된 시스템은 실제 효율이 떨어진다. 지오메트리 변환 하드웨어는 소프트웨어로 운용된다.

2) 레스터 단계 : 윈도우-공간 프리미티브의 프레임 버퍼로의 렌더링 과정은 병목 현상이 상대적으로 적지만, 정점 수와 프리미티브 크기에 영향을 받게 된다.

3) 텍스처 단계 : 텍스처 이미지를 다루는 픽셀 작업에는 빠른 메모리가 필요하며, 메모리가 소모되면 병목 현상이 발생하고, 메모리 캐싱 효율도 문제가 될 수 있다.

4) 프래그먼트 단계 : 프래그먼트 단위는 각 프래그먼트 픽셀에 대한 개별 갱신을 수행한다. 효율적인 픽셀 갱신을 위해 각 프로세서가 스크린의 픽셀 서브 세트를 갱신한다. 하지만 스크린의 모든 픽셀이 독자적인 프로세서를 가지는 것은 그래픽 서브시스템에 있어 비효율적이다. 프래그먼트가 많이 증가되거나 단위 계산이 많으면 병목 현상이 발생한다.

5) 디스플레이 단계의 병목 현상 : 프레임 버퍼에 픽셀을 모니터로 내 보내는 과정에서도 병목 현상이 발생할 수 있다. 일반적으로 컬러 채널에 대한 조정을 중지한다.

2.2 그래픽 서브시스템 설정에 따른 변화

그래픽 서브시스템의 변화가 실제 3D 그래픽스 어플리케이션의 성능에 미치는 영향은 상대적으로 적다. Fig. 3의 SPECapc for Pro/E 2001의 결과에서는 OpenGL 그래픽 서브시스템의 제어 요소 변화가 마이크로프로세서 성능과 데이터 입출력 성능에 영향을 미치기는 하지만, 실제 3D 그래픽스 성능에서는 거의 영향을 미치지 않고 있다는 것을 알 수 있다. 어플리케이션 대형화되면서 그래픽 서브시스템에 비해 시스템에 걸리는 부하가 더 증가된 결과로 볼 수 있다.

Fig. 2의 SPECViewperf 8.1을 통해 수행한 결과에서 OpenGL 그래픽 서브시스템이 제공하는 제어 요소 가운데, Vertical Sync. 항목과 Texture Mem. 항목에서 차이를 보여

+ 김희중 (동현시스텍), E-mail: spix@mail.hhu.ac.kr, Tel: 051)410-4292

++ 정재현 (한국해양대학교 기계정보공학부)

주고 있다. 특히, Texture 항목에서는 CAD 어플리케이션의 경우 DCC 어플리케이션에 비해 성능 저하가 매우 크다. Vertical Sync.는 모든 어플리케이션에서 고른 성능 향상을 보여 준다.

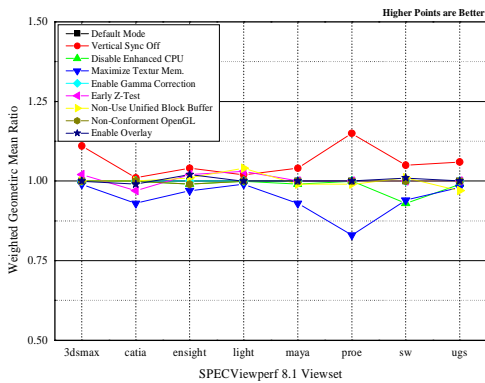


Fig. 2 SPECViewperf 8.1 Performance Test

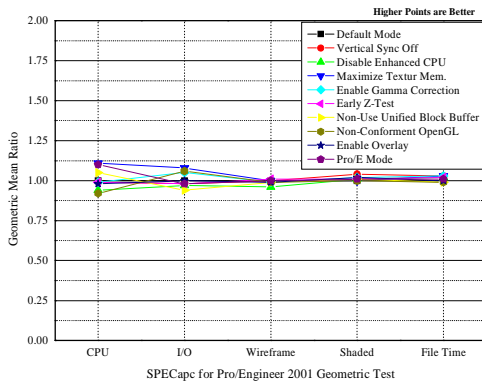


Fig. 3 SPECcapc for Pro/E 2001 Performance Test

3. 드라이버 개정에 의한 성능 분석

OpenGL 그래픽 서브시스템은 OpenGL 3D-API와는 별개로 드라이버에 의해 제어되며, 드라이버에 의해 각 API의 함수와 명령을 가속할 수 있게 된다. OpenGL 그래픽 서브시스템의 급속한 발전은 OpenGL 스펙의 개정을 촉진시켰지만, 실제로 그래픽 서브시스템에서 지원하는 스펙과 3D 그래픽스 어플리케이션에서 지원하는 스펙은 상당한 시간적인 격차를 두고 있다. 대부분의 어플리케이션이 아직도 OpenGL 스펙 1.1/1.2에 기반으로 작성되고 있기 때문이다. 그러므로 드라이버에 의해 그래픽 서브시스템의 지원이 실제 어플리케이션의 성능 개선에 절대적인 상승효과를 보증하지 못하며, 심지어 성능 저하의 요인이 되기도 한다.

각 OpenGL 기반 3D 그래픽스 어플리케이션에 가장 적합한 그래픽 서브시스템의 드라이버의 선택과 그 설정은 실제 현실적인 성능 향상에 있어 가장 중요하다. 본 연구에서는 SPECcapc for Pro/E 2001에 대한 그래픽 서브시스템의 성능을 분석하여

Fig. 4에서와 같은 결과를 얻었으며, 각 드라이버 개정에 따른 성능 변화가 상당함을 알 수 있다.

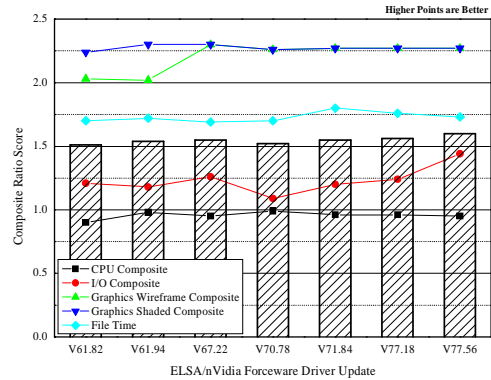


Fig. 4 Comparison of Updating Drives on Graphics Subsystem

4. 결론

본 연구의 OpenGL 그래픽 서브시스템에 대한 하드웨어 환경 제어를 위한 설정과 3D 그래픽스 어플리케이션 운용에 대한 분석을 통한 결과는 다음과 같다.

1. OpenGL 그래픽스 환경에서 그래픽 서브시스템의 가속 성능에 비해 실제 시스템 구성 요소에 의해 영향이 상대적으로 더욱 크다. 그러므로 일정 수준 범위 이내에서의 그래픽 서브시스템 개선은 효과가 매우 낮다.

2. 그래픽 서브시스템의 OpenGL 가속 성능 향상을 위해서는 오히려 성능 제어 요소와 함께 운용 3D 그래픽스 어플리케이션에 적합한 드라이버의 선택이 상대적으로 높은 생산성을 제공한다.

실제 사용자 입장에서 적용하기 위해서는 주요 어플리케이션에 대한 그래픽 서브시스템의 제어 요소 및 드라이버 호환성에 대한 정보를 지속적으로 제공할 수 있다면, 투자한 비용에 대한 높은 3D 그래픽스 성능을 유지할 수 있다.

REFERENCE

1. SPECopc, The OpenGL Performance Characterization Project Group Rules, SPEC, 2005, <http://www.spec.org/gpc/opc.static/Rulestv20.html>
2. SPECapc, The Application Performance Characterization Project Rules, SPEC, 2005, <http://www.spec.org/gpc/apc.static/current.html>