

효과적인 자바 다중 쓰레드 병렬처리를 이용한 네트워크 가상환경의 구현

김우열, 박종구
성균관대학교 정보통신공학부
e-mail:kwy7134@ece.skku.ac.kr

Implementation of Networked Virtual Environment by Using Effective Java Multi-Thread Parallelism

Woo-Yeal Kim, Jong-Koo Park
School of Info. & Comm. Eng., SungKyunKwan University

요 약

대단위(large-scale) 가상현실 시스템이나 가상현실 시스템의 구성요소가 네트워크(일반적인 통신망)를 통하여 분산되어 있는 경우를 네트워크 가상현실 시스템(Net-VE : Networked Virtual Environment)이라 한다. 가상현실의 환경을 분산 처리하는 경우 가상현실 시각의 공유방법, 분산시스템의 구성방법, 저속 혹은 고속통신망을 통한 대량의 데이터를 전송하는 방법, 가상현실 내의 이용자의 상호작용 방법, 데이터 공유방법, 분산처리 방법 등 가상현실 기술에 부가하여 통신망기술, 프로토콜 공학, 분산 처리기술 등 다양한 주변 기술이 필요하다. 따라서 본 논문에서는 더 유연성 있고 모듈러한 네트워크 가상 환경을 구축하는데 있어 효율적인 객체 지향 프로그래밍 기술인 자바의 다중 쓰레드 병렬처리 기법을 사용하고 자 한다. 자바의 쓰레드 구조를 사용함으로써 네트워크 가상환경에서 여러 클라이언트에서 요청하는 데이터를 네트워크를 통한 지연에 대한 효과적인 대안으로서 자바의 다중 쓰레드 병렬처리 기법을 제안하고자 한다.

1. 서론

가상현실(Virtual Reality, VR)이란 컴퓨터가 만들어낸 가상의 세계를 사용자에게 다양한 감각 채널을 통해 제공함으로써 사용자로 하여금 생성된 가상 세계에 몰입(immerse)하도록 하는 것과 동시에, 가상 세계 내에서 현실 세계에서와 같은 자연스러운 상호작용(interaction)을 가능하도록 하는 제반 기술과 이러한 기술에 필요한 이론적 바탕을 지칭한다. 가상현실감 기술은 컴퓨터 그래픽스, 컴퓨터 비전, 인공지능, 네트워크 등 다양한 학문적 지식을 기반으로 하는 미래 지향적 첨단 기술이다. 가상현실의 첫째 목표는 가상 세계에 참여하는 사용자들이 가상 세계에 몰입하고 자연스러운 상호작용을 통해 높은 수준의 현실감(reality)을 경험하여 '자신이 마치 그곳에

실재하는 듯(sense of being there)'한 착각을 느끼도록 하는 것이다.

실시간 상호작용은 Net-VE 응용프로그램의 프로세스 및 Thread 아키텍처를 정의하는데 있어 중요한 고려사항이다. 많은 작업들이 동시에 CPU를 할당받기 위해, 즉 실행되기 위해 경쟁한다. 일반적인 시스템들과 달리 Net-VE의 모든 작업들은 높은 수준의 실시간 제약을 가진다. 다양하고 많은 작업들의 실시간 요구를 만족시킨다는 것은 Net-VE 설계자들에게 하나의 도전이다. 먼저, Net-VE는 로컬 사용자의 실시간 상호작용을 지원할 필요가 있다. 사용자들이 키보드와 마우스, HMD 등을 이용하여 행하는 행동들은 재빨리 감지되고 처리되도록 배려되어야 한다. 이를 처리하는데 지연이 있다면 사용

자들은 지루함과 실망감을 느낄 것이다. 따라서 Net-VE 설계자들은 이러한 작업들을 관리하기 위하여 많은 기술적 노력을 기울여야 한다. 본 논문에서는 그에 대한 한 가지 방법으로 자바의 다중 쓰레드(multi-thread) 병렬처리 기법을 제안한다. 서버에 여러 클라이언트에서 들어온 요청들을 각각 하나의 thread로 할당하여 Net-VE의 자원을 효율적으로 공유하고 CPU 자원을 효율적으로 사용할 수 있게 된다. 하지만, 다중 쓰레드 사용으로 인한 deadlock문제와 공유 데이터에 대한 동기화 처리문제가 발생한다.

본 논문의 구성은 다음과 같다. 2장에서는 Net-VE를 개발할 때 고려해야 할 방법들에 대해서 기술하며, 3장에서는 Java multi-thread parallel 기법과 scheduling 기법을 제안하고, 마지막으로 4장에서는 결론 및 향후 연구 과제를 제시한다.

2. Net-VE 설계와 개발에 있어서의 고려사항

Net-VE는 매우 복잡한 시스템으로 정확하고 효율적으로 구현하기가 매우 힘들다고 알려져 있다. 이 시스템은 여러 가지 소프트웨어들의 성격을 함께 갖고 있으며, 각자가 하나의 응용프로그램으로 동작할 수 있는 수준의 것들이다. Net-VE 시스템의 설계자들은 다음과 같은 시스템의 성격을 이해하고 고려해야 한다.

① 분산 시스템(Distributed systems) : Net-VE의 설계자들은 네트워크 자원을 관리하고, 데이터 손실과 네트워크 고장에 대비하여야 하며, 시스템의 동시성(concurrency)을 고려해야 한다.

② 그래픽 응용프로그램(graphical application) : 설계자들은 끊임없는 실시간 화면 갱신을 위해, 프레임 갱신 비율(frame rates)을 일정 수준으로 유지하여야 한다. 이를 위해서는 렌더링과 같은 관련 작업을 위해 CPU를 주의 깊게 할당하여야 한다.

③ 상호작용 응용프로그램(Interactive application) : Net-VE 시스템은 사용자로부터 끊임없이 데이터 입력이 들어온다. 설계자들은 이러한 실시간 입력을 고려하여 적절하게 처리해 주어야 한다. 비록, 사용자들이 멀리 떨어진 원격 호스트에서 시스템에 접속하지만, 사용자들에게는 가상 환경에 직접 연결되어 있는 것처럼 느껴져야 하기 때문이다.

Net-VE 시스템을 구체적으로 설계하는 것은 실제로 더 복잡한 일이다. 이는 기존에 존재하는 여러 가지 응용 서비스들과 연계되어 시스템이 동작하기

때문이다.

요즘 네트워크 가상 환경의 개발 추세가 점차 성장하고 있으며, 이는 프로세서 가격의 하락이 큰 요인이기도 하다. 네트워크 가상 환경 시스템은 적지 않은 양의 프로세서 수용능력(processor capacity)을 요구한다. 프로세서는 사용자들의 입력 장치를 통해 이벤트를 받아 이를 처리한다. 이러한 입력 이벤트들이 가상 환경 내에서 사용자들의 위치나, 다른 객체들의 위치에 어떠한 영향을 주는지 계산해내는 것이다. 그 다음, 프로세서는 이러한 변화를 다른 참여자들에게 언제 어떠한 방법으로 알릴지 결정한다. 마찬가지로, 프로세서는 다른 사용자들로부터 그들의 위치 변화나 그들이 취한 행동들에 대한 정보를 받는다. 그리고 사용자들의 컴퓨터에서 자율적으로 제어되고 있는 가상 환경 내의 객체들에 대한 모델링도 수행한다. 마지막으로, 프로세서는 가상 환경에 대한 디스플레이를 가장 최신으로 유지하기 위해 그래픽 디스플레이를 변화시킨다. 네트워크 가상 환경을 유지하기 위해 요구되는 무수히 많은 작업들 사이에서 사용 가능한 프로세서 사용 시간을 어떻게 할당해 내느냐는 것이다.

이러한 한정된 프로세서의 수용능력을 극복하고 네트워크 가상 환경의 성능을 개선시키기 위해 여러 가지 하드웨어적인 기술과 소프트웨어적인 기술이 필요하다.

3. 자바 다중 쓰레드 병렬처리와 동기화 기법

3.1. Java multi-thread parallelism

쓰레드는 프로세스와 유사성을 가지고 있다. 일단, 쓰레드는 프로세스와 비교될 수 있고, 하나의 자바 가상 머신 상에서 수행되는 여러 쓰레드를 가진 프로그램은 운영체제에서 수행중인 여러 프로세스와 비교될 수 있다. 자바에서 다중 쓰레드는 다음과 같은 속성을 가지고 있다.

① 쓰레드는 미리 정의된 위치에서 수행을 시작한다. 프로그램 내에서 쓰레드들 중 하나의 시작 지점은 main() method이다. 나머지 쓰레드의 시작 지점은 프로그래머가 코딩할 때 임의로 결정한다.

② 쓰레드는 시작 지점으로부터 미리 정의된 순차적인 순서대로 수행된다.

③ 쓰레드는 프로그램 내의 다른 쓰레드와 서로 독립적으로 수행된다.

④ 쓰레드는 동시성을 가진다. 동시성의 정도는 여

러 요인에 의해 좌우되지만, 여러 쓰레드들 중 상대적으로 중요한 쓰레드를 어떤 것으로 결정하는지에 따라 좌우되기도 하고, 운영체제가 다양한 기능을 어느 정도 지원 해주는냐에 따라 좌우되기도 한다.

⑤ 쓰레드는 여러 종류의 데이터에 접근 가능하다.

Net-VE에서 서로 독립적인 여러 개의 작업들을 처리해야 할 경우가 자주 있다. 예를 들면, 가상 환경 내에서 두 개의 아바타의 애니메이션을 수행해야 할 필요가 있을 때, 또 여러 클라이언트의 요청을 받아 계산 작업을 수행하는 서버의 경우에도 마찬가지이다. 물론 이런 다중 작업을 처리하기 위해 단일 쓰레드를 사용하는 것도 가능하지만, 각 작업을 하나의 쓰레드로 분리해 처리하면 좀 더 효율적으로 수행할 수 있다. Pthreads 혹은 Cthreads와 같은 특정 Thread Library를 가지고 Thread management를 수행하는 일반적인 다른 많은 프로그래밍 언어와는 반대로 자바 프로그래밍 언어는 그 자체 내에서 직접적인 다중 쓰레드 기법을 지원한다. 자바 프로그래밍 언어는 java.lang package안에 class Thread와 interface Runnable을 가진 Thread API가 있다. Java 프로그래밍 언어는 쓰레드를 생성하는 두 가지 기본적인 방법이 있다.

쓰레드를 생성하는 방법 중 하나인 Thread 클래스를 이용하는 방법에 대해 살펴보면, 먼저 Thread class를 상속하여 새로운 클래스 하나를 만들고 Thread의 run() method를 overriding 한다. 다시 말해서, Thread를 시작 시키면, run() method가 호출되고, run() method의 수행이 끝나면 thread는 종료하게 된다. Thread를 생성하기 위한 다른 방법인 Runnable interface를 이용하는 방법은 Runnable interface 내에 단 하나밖에 없는 run() method를 구현하여 하나의 클래스를 만들고 그 클래스의 인스턴스를 생성한 후, 이 인스턴스를 thread를 생성할 때 매개변수로 넘겨준다. Runnable interface는 run() method 밖에 없기 때문에 표준 Thread 클래스를 이용하여 객체를 만들 때 Runnable interface를 Thread 생성자의 매개변수로 넣어주고, Runnable interface는 Thread 클래스의 객체를 생성할 때 생성자의 매개변수형식으로 들어가서 하나의 온전한 쓰레드가 된다.

3.2. Thread synchronization

하나의 프로그램 내에서 쓰레드들은 common address space를 가진다. 그러므로 여러 쓰레드들이

같은 데이터에 접근할 때 synchronization mechanism에 의해 올바르게 수행된다. 자바는 각각의 쓰레드들에게 내부적으로 lock을 할당하여 synchronization을 지원한다. Thread synchronization을 구현하는 가장 쉬운 방법은 Thread object를 synchronized로 선언하는 것이다. 쓰레드로 구현된 method는 lock을 얻으면 수행하고, 수행이 종료되면 lock을 해제한다. 쓰레드로 구현된 method의 수행이 정상적인 방법이 아닌 예외에 의해 종료되었다고 하더라도 lock은 해제된다. 또 다른 방법은 waiting thread의 목록을 지원하는 Object class의 wait() 와 notify() method에 의해서 쓰레드의 synchronization을 수행할 수 있다. 자바시스템에서 모든 객체들은 직-간접적으로 Object class로부터 상속받기 때문에 synchronization mechanism을 수행할 수 있다. 하나의 쓰레드가 하나의 객체의 wait() method를 호출했을 때, 그 쓰레드는 객체의 waiting thread의 목록에 더해지고 실행을 멈춘다. 다른 Thread가 같은 object의 notify() method를 호출하면 waiting thread들 중 하나가 활성화되어 실행을 수행한다. 이러한 간단한 synchronization 특징이 더 복잡한 synchronization mechanism을 구현하는데 사용된다.

3.3. 구현 시스템 구조

다수의 클라이언트측에서 요청한 작업들을 서버에서는 클라이언트의 요청을 쓰레드로 할당하여 처리한다.

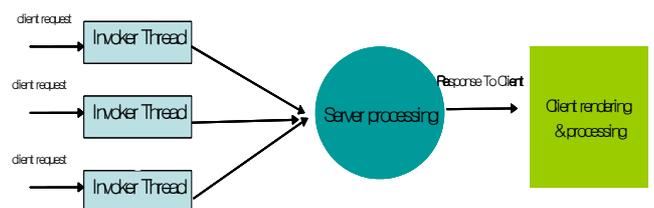


그림 1. 쓰레드 처리 방식

클라이언트측은 자바 servlet을 사용하여 그래픽 유저 인터페이스를 구축하고 데이터베이스 서버에 있는 가상현실의 공유 데이터들을 요청하여 웹 기판하에서 구현한다.

아래 그림은 클라이언트에서 서버로 하나의 작업을 요청하고, 서버는 쓰레드로 할당하여 처리하는 모습을 보여주고 있다.

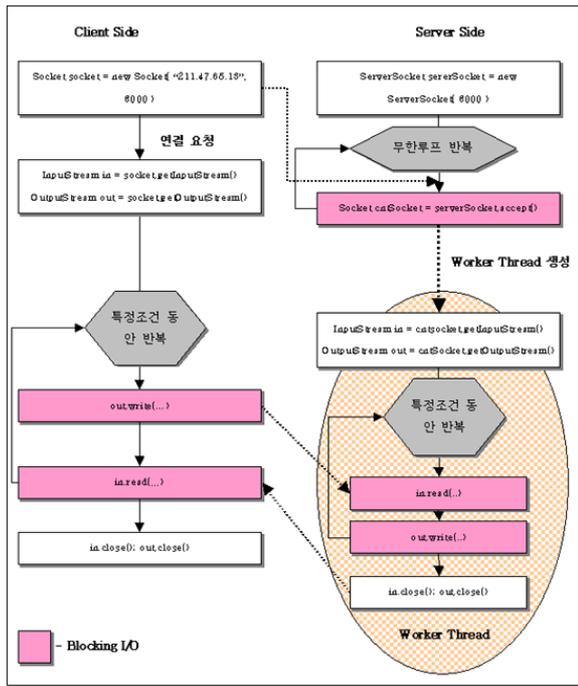


그림 2. 쓰레드 구현 방식

본 논문에서 구현한 시스템은 자바 Servlet을 사용하여 네트워크 가상 환경 참여자에게 View를 제공하고 Oracle DBMS를 사용하여 데이터베이스 서버를 구축하여 각각의 클라이언트에서 요청하는 데이터를 저장하고 있다.

4. 결론 및 향후 연구과제

Synchronization method와 여러 클라이언트들의 작업을 쓰레드로 할당한 방법으로 본 논문에서 구현한 Net-VE 시스템은 더 향상된 네트워크 전송 속도와 deadlock 문제를 피할 수 있었다. 자바에서 다중 쓰레드 병렬처리 방법은 서버에서 클라이언트로 데이터를 전송할 때 message passing 처리 실행시간을 단축할 수 있었다.

다중 쓰레드 병렬처리 방법을 사용하여 구현한 Net-VE 시스템의 보다 나은 성능 개선을 위해서 Thread scheduling과 Signal handling의 기술이 추가적으로 수행되어야 하고, 보다 넓은 의미의 Net-VE를 위해 확장성과 안정성 그리고 사용자에게 더욱 현실감과 몰입감 그리고 실제감을 줄 수 있도록 앞으로 더 많은 연구가 필요하다.

참고 문헌

[1] S. Singhal and M. Zyda, *Networked Virtual Environments -Design and Implementation*, ACM Press Books, SIGGRAPH Series, 23 July 1999.

[2] Pradeep K. Sinha, *Distributed Operating Systems Concepts and Design*, IEEE PRESS, 1997.
 [3] C. Greenhalgh, "Large Scale Collaborative Virtual Environments", *Springer*, 1999.
 [4] J. Vince and R. Earnshaw, Eds., *Virtual Worlds on the Internet*, Los Alamitos, CA:IEEE Comput. Soc. Press, 1998.
 [5] S. Oaks and H. Wong, *Java Threads*, O'Reilly & Associates, Inc., 1999.
 [6] J. E. Berger, L. T. Dinh, M. F. Masiello, and J. N. Schell, "NVR: A System for Networked Virtual Reality", *IEEE*, 1994.
 [7] D. I. Ko, B.R. Lee, J. S. Kim and W.G. Oh, "Implementation of virtual architectural engineering and design of real-time state server for internet virtual collaboration", *2000 IEEE Fall Symposium*, Nov. 2000.
 [8] M. Deering and H. Sowizral, *Java3D Specification, Version 1.0*, Sun Microsystems, August 1997.
 [9] R. F. Boisvert, J. Moreira, M. Philippsen, and R. Pozo, "Java and Numerical Computing" *IEEE Computing in Science and Engineering*, 2001.
 [10] H. Blaar, M. Legeler and T. Rauber, "Efficiency of Thread-parallel Java Programs from Scientific Computing", *Proceedings of the International Parallel and Distributed Processing Symposium*, 2002.
 [11] B. Joshua, *Effective Java™ Programming Language Guide*, Addison-Wesley, 2001.