온라인 게임서버 구현에 관한 연구

배성길, 김혜영,
*홍익대학교 게임학부 소프트웨어학과

e-mail : ddinggill@naver.com
hykim@hongik.ac.kr

A Study for Online-GameServer Development

요 약

온라인 게임서버는 다수의 클라이언트들의 접속 요청 및 여러 작업요청을 실시간으로 빠르게 처리할 수 있어야 하기 때문에 동시 접속 사용자의 수에 비례하여 처리할 네트워크 양도 많아지고 서버 프로그램의 부하도 그만큼 높아지게 된다. 따라서, 본 논문에서는 온라인 게임서버에서의 게임서버의 기능을 효율적으로 수행하기 위한 기술들을 제시하고 AOS 장르의 게임의 프로토타입의 클라이언트들을 IOCP 로 구현한 온라인게임 서버에 연동하여 네트워크 통신과 클라이언트들간의 동기화 작업을 구현하였다.

1. 서론

온라인 게임서버는 다수의 클라이언트들의 접속 요청 및 여러 작업요청을 실시간으로 빠르게 처리할 수 있어야 한다. 또한 사용자들의 그룹형성이나 사용자들이 게임을 즐길 환경의 제공 및 게임의 맵, NCP, AI 등의 데이터 처리 하는 역할까지 수행해야 한다.[1]

게임서버의 역할은 처리단계, 기능, 데이터 등의 기준으로 나누어져 효율적으로 수행 할 수 있게 설계되고, 확장성도 고려해야 한다.

온라인 게임은 다른 게임들과는 다르게 클라이언트는 자료 혹은 인증을 요청하고, 서버는 클라이언트의 요청을 판단하고 처리하는 작업을 수행하기 때문에 동시 접속 사용자의 수에 비례하여 처리할 네트워크 양도 많아지고 서버 프로그램의 부하도 그만큼 높아지게 된다.

따라서, 본 논문에서는 온라인 게임서버에서의 게임서버의 기능을 효율적으로 수행하기 위해 게임서버에 적용한 몇 가지 기술들을 제시하고, 이를 적용하여 온라인 게임서버를 구현하였다. .

2. 온라인 게임서버에 적용한 기술들

(1) C/C++관련 기술들

온라인 게임서버는 여러 가지 역할들로 나누어진 객체클래스들로 이루어진다고 할 수 있는데 거기서 클래스를 효율적으로 활용하기 위한 기법들이 있다.
이를 위한 기법들로는 여러 가지 디자인 패턴사용(ex 싱글톤패턴), 클래스 상속, 포함, 참조, 및 템플릿, 멤버함수들의 오버로딩 및 오버라이딩, 함수포인터, 메모리 동적 및 정적 생성 및 메모리 누수 체크, 가상함수의 사용 등이 있다. 온라인 게임서버에서는 설계적인 부분이 매우 중요하므로 이러한 클래스들을 효율적으로 사용하기 위한 기법들의 역할과 활용이 큰 비중을 차지하게 된다

좀 더 효율적인 데이터 관리 및 참조를 위해서 사용하는 기술들이 있는데 바로 STL 의 사용이다. STL 의 iterator, vector, list 등의 다양한 함수들을 통해 손쉽게 다양한 객체들을 관리할 수 있다.

서버측에서는 IOCP 기술을 사용하여 데이터를 수신하고 이를 WorkThread 들을 통해 처리하며, 게임 사용자의 수에 따라 서버에 도착하는 패킷의 양은 기하급수적으로 늘게 되므로, 기존의 단일버퍼만으로는 패킷이 도착하는 양을 서버의 작업속도만으로는 충분히 만족시킬 수가 없다. 그래서 원형구조로 된 하나의 버퍼를 만들어 도착한 패킷들을 서버가 처리하기 전에 저장하고 버퍼에 저장된 패킷들을 서버가 꺼내어 처리하는 형태의 기술을 사용하였다. 이를 통해 서버의 비동기적인 작업 수행능력과 효율을 높일 수가 있다.

(2) Network 관련 기술들

기본적인 모든 통신에는 Socket 을 사용한 통신을 하였는데 이를 위한 WinSock 기본함수들을 사용하였다. WSAStartup(), Socket(), Send/Recv(), Accept, Connect() 함수 등이 이에 해당한다. 이들은 초기화, 소켓생성, 데이터 전송 및 수신, 접속요청, 접속요청 수락 등의 기능을 지원하여 기본적인 Network 통신이 가능하도

록 해준다.[2]

서버는 클라이언트의 요청 및 여러 가지 다른 작업을 실시간으로, 비동기적으로 수행하여야 하기 때문에 단순한 쓰레드 모델이나, 에코모델들로는 한계가 있고 작업을 수행하는 것에 또한 많은 부하가 발생한다. 클라이언트의 작업요청에 최대한 빨리 반응하여 데이터를 처리 및 전송하여야 하는 온라인게임에서 이러한 서버의 느린 성능은 아주 치명적인 약점으로 작용할 수 있다. 그래서 우리는 서버/클라이언트 모델에서, 서버 측에는 IOCP(Input Output Completion Port)라는 윈도우에서 제공하는 기본 입출력 관련 모델을 이용하였다. 기본적으로 IOCP 를 사용하여 중첩 I/O의 기술과 쓰레드 풀링 기술들을 이용하여 효율적으로 데이터를 처리하였다. IOCP 에 사용할 포트를 생성하여 관련 소켓들과 연결하고, 실시간 쓰레드을 이용하여 완료 포트 큐를 감시하여 클라이언트로부터 도착한 패킷들을 쓰레드들이 풀링기술을 이용하여 효과적으로 처리한 것이다. 이를 통해 서버는 패킷수신, 데이터처리, 접속처리 등의 다양한 작업들을 비동기적으로 수행한다. 즉 기존의 메인 쓰레드를 남겨두고 빈번하게 발생되는 네트워킹 작업을 위해서 전담 쓰레드를따로 생성해야 하는 것이다.[3]

클라이언트 네트워크 모듈은 서버에 비하여 그렇게 많은 데이터를 처리하지 않아도 되기 때문에 구현이 어렵고 복잡한 IOCP 기술을 사용하지 않고 좀 더 단순한 EventSelect 모델을 사용하여 구현하였다.

(3) 시스템 프로그래밍 및 기타 기술

앞에 언급한 기술과는 별도로 서버의 효율적인 수행을 위하여 꼭 필요한 기술로 동기화 기술들과 온라인 게임에서 필수라고 할 수 있는 로그인 및 자료관리를 가능하게 해주는 DB 기술들이 있다. 동기화 기술로는 다양한 기술들이 있지만 여기에서는 CriticalSection 과 Event 를 이용하여 서버의 동기화를 구현하였다.

CriticalSection 을 사용한 가장 큰 이유는 구현이 용이하고 유저영역의 동기화 기법으로써 서버에 걸리는 부하가 적기 때문이다. 서버는 앞에서도 언급했듯이 많은 네트워크 데이터를 처리하기 때문에 다른 부분에서 최대한 자원을 작게 사용하고 효율적으로 사용하는 것이 중요하다. 또 서버에서는 동적 할당을 사용하는 경우가 많아지는데 이를 위해서 꼭 메모리 누수체크를 하는 클래스를 생성하여 관리했다.

서버에 접속한 클라이언트들을 효과적으로 관리하기 위해 클라이언트 세션객체를 동적으로 생성하는 것이 아니라 정적으로 생성해놓고 효과적으로 돌려쓰는 객체 풀링 기법을 적용하였다.

3. 구현

클라이언트 측은 EventSelect 모델을 사용하여 AOS 장르의 게임에 대한 프로토타입의 클라이언트에 네트워크통신을 위한 모듈용으로 EventSelect 클래스를 만들어서 네트워크 작업들을 수행하도록 구현하였다.

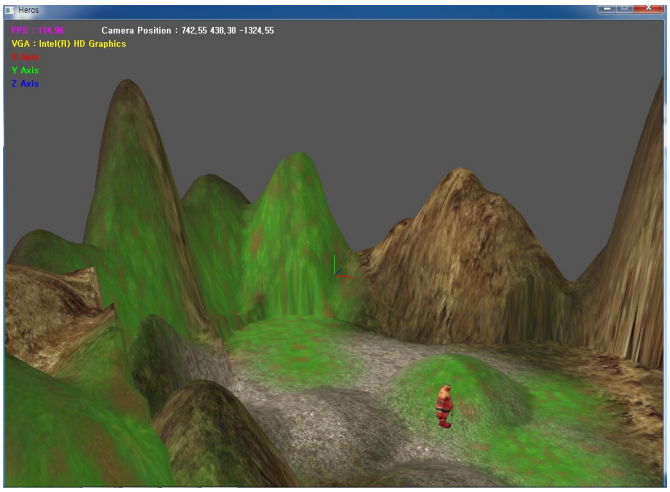
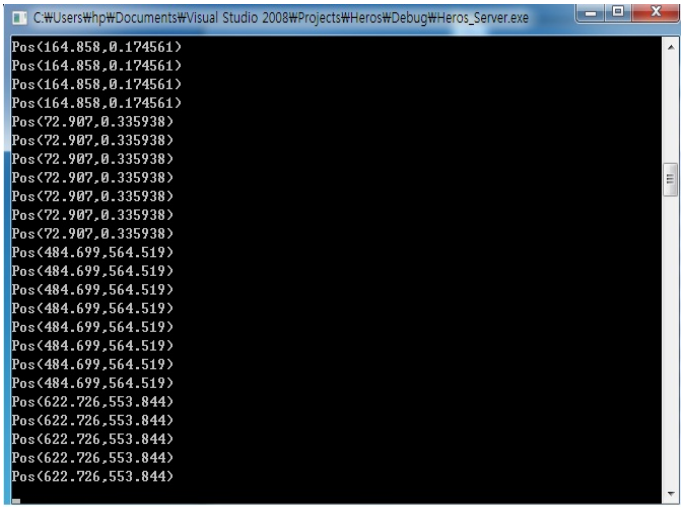


그림 1 클라이언트 화면

서버 측은 게임서버모델에서 가장 효율적인 IOCP 모델을 사용하여 구현하였다. 네트워크 통신을 위한 IOCP 기법들과 로그인 과정을 위한 데이터베이스를 위해 MSSQL 을 서버와 ODBC 로 연동하여 구현하였다. 클라이언트로부터 수신한 패킷을 데이터처리를 하여 다른 클라이언트에게 브로드캐스팅을 통해 패킷을 전송하고, 실시간으로 서버에서 로그를 실행하였다. 서버는 비동기적으로 IOCP 에게 데이터를 수신하여 원형 큐에 전송된 패킷을 추가하고 WorkThread 를 쓰레드풀링 방법으로 생성하여 실시간으로 버퍼를 감시하여 추가된 패킷을 처리한다.



(그림 1) 서버 화면

5. 결론

본 논문에서는 온라인 게임서버의 기능을 효율적으로 수행하기 위한 기술들을 제시하고 AOS 장르의 게

임의 프로토타입의 클라이언트들을 IOCP 로 구현한 온라인게임 서버에 연동하여 네트워크 통신과 클라이언트들간의 동기화 작업을 구현하였다. 기존의 다른 모델들 보다 IOCP 를 사용한 서버 모듈은 훨씬 더 많은 네트워크 작업을 수행 할 수 있었고, 비동기적인 작업들과 서버에서의 클라이언트들의 관리 및 클라이언트들 간의 동기화 작업이 효율적으로 수행되었다.

.

참고문헌

[1] 한 동훈, 온라인 게임 서버 프로그래밍 , 정보문화사, 2007

[2] 분산 게임서버 기술 동향, 정보통신동향 분석 제 20 권 4 호 2005 년 8 월

[3] 김혜영, 함대현, “온라인 게임서버를 위한 객체 폴링 기법에 관한 연구”, 한국 게임 학회 논문지, 제 9 권 제 6 호, 2009 년 12 월