언리얼 클라이언트를 연동한 IOCP모델서버 성능 테스트

나장호, 김혜영*, 오성현
홍익대학교 게임학부 게임소프트웨어전공
jang1na2@gmail.com, hykim@hongik.ac.kr, yb2180@naver.com

IOCP model server performance test linked with Unreal Client

Jang-Ho Na, Hye-Young Kim*, Sung-Hyun Oh
Major in Game Software, School of Games, Hongik University

요 약

본 논문은 언리얼 엔진에서 제공하는 Dedicated server를 사용하지 않고 자체적으로 제작한 IOCP모델 서버를 구축하고, 이를 언리얼 클라이언트와 MySQL데이터베이스에 연동하여, 완성된 서버의 성능 테스트를 진행한 결과를 보였다.

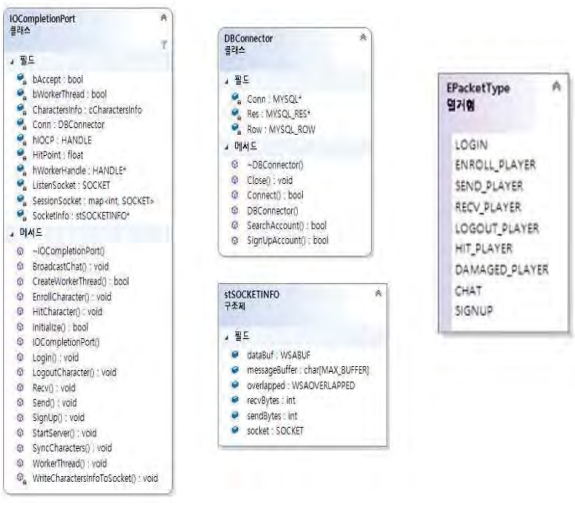
1. 서 론

네트워크 게임의 발달과 증가로 서버의 중요성은 나날이 커지고 있다. 그 중에서도 게임서버에서 최적인 IOCP서버를 캡스톤디자인에 입각하여 구현을 해보고 성능테스트까지 진행하여 게임서버를 만드는데 도움이 되고자한다. IOCP는 서버의 I/O처리 성능을 극대화 시켜 하드웨어 비용을 감소시키는데 크게 기여하고 있다[1, 2].

상용 게임 엔진에서는 독자적으로 만든 서버를 지원하여 효율성을 높여 게임 엔진의 장점인 개발 기간의 단축 및 개발비용을 절감하고 여러 가지 네트워크 기능을 제공한다[3]. 실제로 언리얼 엔진에서는 제공하고 있는 Dedicated Server를 많이 사용 하고 있는데 자체적으로 서버를 직접 제작해 봄으로써 서버를 구현하는데 필요한 기술적요소나 구조를 경험하고 후에 서버를 설계하는데 효율적으로 하는데 도움이 될 것이다.

2. 작품 구조 및 기술요소

2.1. 클래스다이어그램



(그림 1) IOCP서버 클래스 다이어그램

IOCP란 Input/Output Completion Port의 약자이며 여러 개의 쓰레드를 비동기식 입/출력으로 수행하기 위한 API이다.

서버는 (그림 1)과 같이 IOCP서버, 데이터베이스 형태로 구성되어있다.

* 교신저자: hykim@hongik.ac.kr

2.2. 서버의 모듈화 및 각 기능 함수분할

기본 Echo서버는 Main함수에 모든 서버내용이 다 들어가 있는데 이것을 Class로 모듈화 시키고 각 기능들에 따라 함수로 따로 분할하여 만든다. (그림 2)에서 IOCP모델 서버에 필수로 들어가야 하는 Echo서버 기능을 하는 함수들은 빨간 줄로 표시하였고 그렇지 않은 함수는 클라이언트의 기능들을 추가한 함수들이다.

(그림 2) IOCP서버 주요 함수

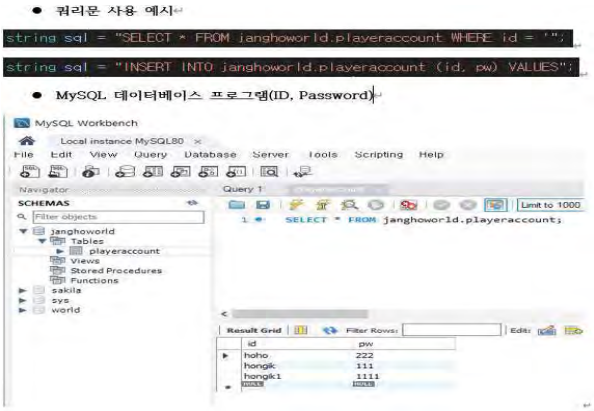
2.3. DLL(Dynamic Link Library)모듈 사용

헤더파일을 만들어 (그림 3)과 같이 변수 및 함수들을 정적으로 관리하지 않고 DLL프로젝트를 따로 생성 후 패킷을 관리하였다. DLL파일로 관리함으로써 정적으로 사용할 때보다 더 빠른 빌드테스트가 가능하며 게임 특성상 업데이트가 자주 있는 클라이언트에 적합하다.

(그림 3) DLL 필수 코드

2.4. MySQL 데이터베이스 연동 및 로그인

MySQL 데이터베이스를 연동 후 로그인 및 회원가입 기능을 만들어 최대한 게임에 필수인 요소들을 (그림 4)와 같이 구현하였다. MySQL 헤더파일의 함수들과 쿼리문에 익숙해져야 구현할 수 있다.



(그림 4) 데이터베이스 사용 예시

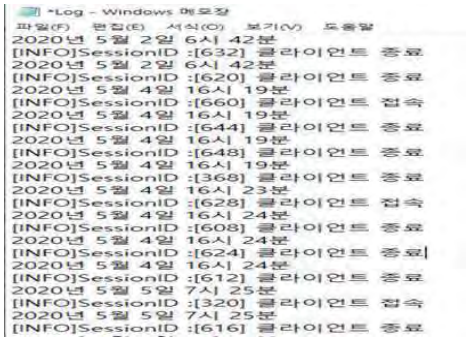
3. IOCP 게임서버 성능 테스트

<표 1> 테스트 하드웨어 환경

CPU	I7-6700
RAM	8GB
운영체제	Windows 10
GPU	NVIDIA GTX960

3.1. 72시간 서버 가동 상태 확인

하드웨어 환경은 <표 1>과 같고 제작된 게임서버의 가동성을 검증하기 위해 약 3일정도의 시간동안 서버를 계속 작동시킨 상태에서 무작위로 언리얼 클라이언트를 접속하여 서버가 과부하가 되지는 않는지, 게임 플레이는 문제없이 작동하는지의 테스트를 위해 72시간이 넘는 약 80시간 정도를 진행하였으며 테스트 상에서 도출되는 문제는 없었다. 테스트 결과는 (그림 5)와 같다.

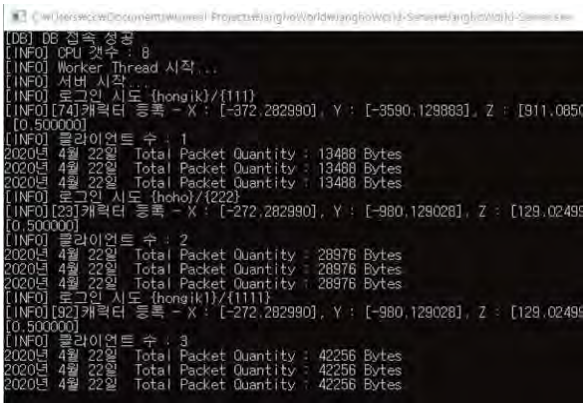


(그림 5) 클라이언트 로그 기록

3.2. 언리얼 클라이언트와 IOCP서버 송수신 Byte처리량 측정

초당 세션 접속 수, 초당 바이트 처리량들에 대해 테스트를 통해 서버에 대한 대략적 성능을 확인 할 수 있다[4].

(그림 6)에서 Sending Packet Quantity는 서버가 송신하는 패킷의 총 Byte량을 가리키고 Receive Packet Quantity는 서버가 수신하는 패킷의 총 Byte량을 나타낸다. 마지막으로 Total Packet Quantity는 총 송수신 패킷 처리량을 보인다.

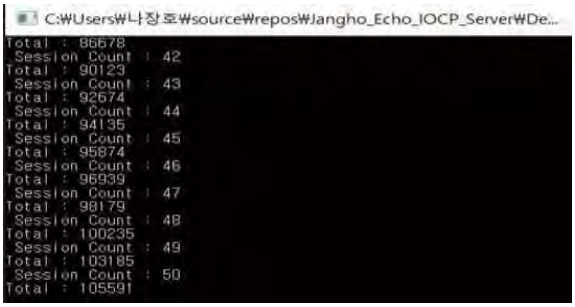


(그림 6) 클라이언트와 서버 송수신 패킷량

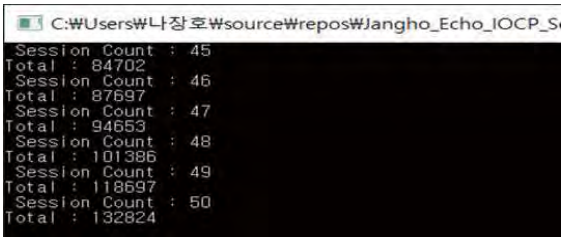
송수신 Packet 출력은 10초 동안의 처리량을 나타내며 클라이언트 1개당 약 14000Bytes를 처리하고 이는 1400Bytes/s 정도 처리량이 되는 것으로 알 수 있다. 하지만 이것은 오로지 언리얼 클라이언트에서 캐릭터의 위치, 회전방향, 속도값 등 캐릭터 Move패킷만을 Tick을 사용하여 주고받는 것을 측정했기 때문에 실제 상용화된 게임 환경을 반영하지 못한 한계가 있다.

3.3. 더미클라이언트를 사용한 다수 유저 접속 한 패킷 Byte처리량

언리얼 클라이언트는 용량이 너무 커서 서버 테스트 진행에 따른 어려움으로 별도의 c++ 더미 클라이언트를 만들어 다수의 접속자들의 연결을 가정해 서버가 처리하는 Byte량을 시뮬레이션 하여 테스트함으로써 과부하여부를 확인하였다[5].

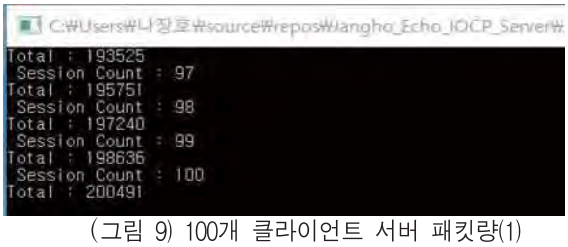


(그림 7) 50개 클라이언트와 서버 패킷량(1)

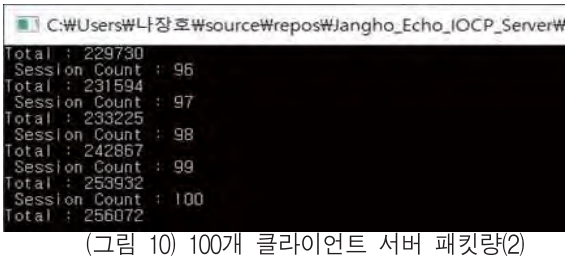


(그림 8) 50개 클라이언트와 서버 패킷량(2)

(그림 7, 8)은 50개 클라이언트를 첫 번째 테스트에선 10만Bytes 두 번째에선 13만Bytes로 테스트를 할 때 마다 차이를 조금씩 보였고 평균 초당 10만Bytes 정도를 처리하는 것으로 나타났다. 그 후 100개 클라이언트도 동일하게 테스트를 진행하였고 첫 번째는 20만Bytes 두 번째는 25만Bytes로 50개 클라이언트의 비해 다소 큰 차이가 나타났고 약 20만Bytes 정도로 나타났다. 결과는 (그림 9, 10)과 같다.



(그림 9) 100개 클라이언트 서버 패킷량(1)



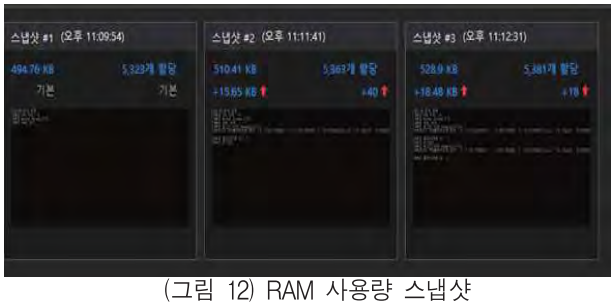
(그림 10) 100개 클라이언트 서버 패킷량(2)

3.4. 서버 CPU, RAM 사용량 프로파일링

(그림 11)의 CPU사용량에서 탐색 불가능부분은 DB Connector로 데이터베이스 코드이다. 가

장 많이 사용되는 부분은 역시 WorkThread가 포함된 IOCP클래스 부분이다. (그림 12)의 RAM 사용량은 서버만 작동 중일 때, 클라이언트가 접속해 있을 때를 구분해서 총 3개의 스냅샷을 찍어 프로파일링 하였다. 용량이 큰 언리얼 클라이언트를 사용하더라도 서버의 메모리를 많이 차지하는 것은 아닌 것을 알 수 있었다.

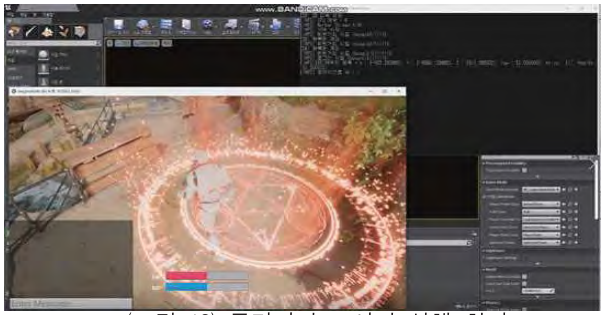
(그림 11) CPU 사용량 보고서



(그림 12) RAM 사용량 스냅샷

4. 결론 및 향후 연구 방향

서버의 재사용성을 위해 코드를 객체지향 클래스로 만들었으며 DLL파일을 사용하여 동적으로 패킷을 관리 및 기술요소들을 구현하였다. (그림 13)은 실행화면이며 새로 소켓을 추가하여 패킷타입과 추가 함수기능을 수정해 동기화를 시켜준다면 어떤 프로그램에도 사용이 가능한 다형성 특성도 갖고 있다. 향후 연구로 더 정확한 결과 도출을 위해 많은 패킷테스트와 실제 상용화된 게임 환경을 구현하고자 한다.


(그림 13) 클라이언트-서버 실행 화면

Acknowledgement

본 연구는 대학혁신지원사업비를 지원받았으며, 그 저작권은 홍익대학교로 귀속됨.

참고 문헌

[1] Gyu-baek Kim, "An Effective Processing Server for Various Database Operations of Large-scale On-line Games", IASTED International Conference on Information and Knowledge Sharing(IKS), Arizona, U.S.A November 2003.

[2] Jin-Wook Han, Hye-Young Kim, "A Study of Modueled Gameserver with IOCP", Proceedings of KIIT Conference, 565-568(4 pages), 2018.11

[3] Hun-Joo Lee, Tae-Joon Park, Hyun-Bin Kim, "A Development Case of 3D Game : Dream3D Prototype Game Contents", Academic Conference of the Korean Electronics Engineering Association, 1431-1434(4 pages), 2003.7.

[4] D. H. HAN, "Online Game Server Benchmark", Information Culture History, October 2008

[5] Ki-Nam Kim, Hye-Young Kim, "A study on gaming server Implementation as IOCP model applying load balance scheme in MMORPG", Proceedings of KIIT Conference, 612-615(4 pages), 2019.6