

CORBA기반 분산 네트워크 게임 서버에 관한 연구

최재언*, 이해원**, 하수철*

*대전대학교 컴퓨터정보통신공학부

**한국전자통신연구원

regina@zeus.taejon.ac.kr*, hewlee@etri.re.kr**,

soocha@dragon.taejon.ac.kr*

A Study on CORBA-based Distributed Network Game

Jae-Eon Choi*, Hae-Won Lee**, Soo-Cheol Ha*

*Dept of Computer Engineering, Daejon University

**ETRI

요약

오늘날 게임은 초고속 통신망, 인터넷의 확산으로 많은 발전을 가져왔다. 그 중 온라인 네트워크 게임은 동시에 수백에서 수천명이 접속하여 게임을 즐기는 클라이언트/서버 모델 개념으로 발전하고 있다. 하지만 서버 내 접속하는 클라이언트의 수가 증가함에 따라 서버에서 다수의 클라이언트 게임머를 지원하기 위해서는 많은 문제점을 가지고 있다.

본 논문은 기존 네트워크 게임의 구조적 단점을 보완하고 CORBA기반 three-tier방식에 적용하여 다수의 클라이언트 게임머 접속을 효율적으로 지원할 수 있는 분산 네트워크 게임에 대한 제안이다

1. 서론

오늘날 컴퓨터의 빠른 보급과 초고속 통신망을 바탕으로 한 인터넷의 발전으로 인해 게임은 급속도로 확산·보급되었으며, 또한 고부가가치를 창출하는 미래의 핵심 산업으로 각광을 받고 있다. 그 중에 온라인 네트워크 게임은 TCP/IP망을 이용, 동시에 여러명이 접속하여 하나의 사이버 공간을 만들 수 있으며, 이렇게 만든 공간내에 또다른 사회를 형성하여 그들 나름대로의 인간관계를 맺을 수도 있다. 하지만 서버내 접속한 클라이언트 수가 증가함에 따라 네트워크 부하 가중과 게임 객체 및 데이터들을 저장하고 있는 게임서버의 시간적 동기화, 그리고 하나의 서버가 이상이 발생했을 경우, 다른 서버와의 connection문제의 해결책이 필요하다.

이에 따라 본 연구에서는 현재 분산 객체 기술로서 각광을 받고 있는 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture)를 이용하여 게임 데이터의 동시접근

제어 및 접속중인 서버가 이상이 있을 경우 다른 서버와의 connection, 그리고 네트워크 부하 가중의 문제점을 줄이기 위한 해결방안을 제안한다.

이상의 연구에 대한 본 논문의 구성은 다음과 같다. 2장에서는 일반적인 네트워크 게임서버에 대해 알아보고 3장에서는 분산 객체 기술중의 하나인 CORBA에 대한 관련 연구를 살펴보고 4장에서는 구현하고자하는 분산 네트워크 게임의 전체구성도 및 동작 메커니즘을 제시한다. 5장에서는 제안된 시스템의 특징 및 장점을 기술하며 마지막 6장에서는 결론 및 향후 연구과제를 알아본다.

2. 네트워크 게임

2.1 IPX기반의 네트워크 게임

IPX를 기반으로 하는 네트워크 게임의 경우에 게임에 참여한 한 사람의 컴퓨터가 서버처럼 동작하며 이 서버에 의해서 전체 게임의 동기가 조절되며 사용자의 수가 제한적인 반면에 전체 네트워크의 부하

가중에 의한 속도저하로부터 비교적 자유롭다. 따라서 네트워크의 성능이나 기타 사용자들간의 물리적인 위치에 대한 부담으로부터 자유로우며 하나의 서버에 부가되는 부하가 크지 않으므로 TCP/IP를 기반으로 한 네트워크 게임보다 개발에 필요한 자원이 적게 소요된다.

2.2 TCP/IP기반의 네트워크 게임

TCP/IP를 기반으로 하는 게임의 경우에 서버에 크게 의존하므로 서버의 성능이 전체 게임의 수행속도에 지대한 영향을 미치며, 원거리 사용자간에 게임이 이루어지므로 인터넷에서의 네트워크 속도와 같은 여러 가지 네트워크의 문제점에 대한 고려가 필요하다.

2.3 일반적인 네트워크 게임 서버구조[5]

현재 일반적으로 많이 사용되고 있는 방식은 two-tier방식으로서, 클라이언트와 서버간에 인터넷을 통해 일대다 관계를 유지한다. 이러한 방식을 사용하는 구조방식을 중앙 집중형 구조(단일 서버 구조)라 한다.

분산 서버 구조는 중앙 집중형 서버 구조와 달리 서버가 한 대 이상일 때 필요하다. 네트워크 게임을 즐기는 사용자들이 급증하면서, 단일 서버로만 운영되는 시스템이 한계에 달하게 되었다. 분산 서버 구조는 2개 이상의 서버로 구성되어 있으며, 이 서버들은 서버들간에 데이터 공유가 되는 방식과 데이터 공유를 할 수 없는 방식으로 나뉜다. 즉 대칭 서버(Replicated Server)와 비대칭 서버(Non-Replicated Server)로 나뉜다.

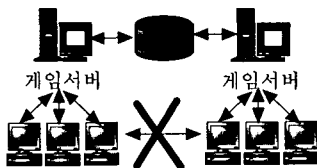


그림 1 대칭 서버 구조

그림 1에서 보여주는 대칭 서버구조는 단지 동일한 서버구조를 가진 시스템을 2개 이상 연결하여, 클라이언트로 하여금 서버에 접속할 수 있는 사용자수를 늘리는 방법중의 하나이다.

그렇기 때문에 동일한 서버라고 해도 각각에 연결되어 있는 클라이언트는 다른 서버의 클라이언트와 공유될 수가 없다는 단점이 있다.

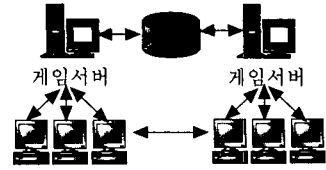


그림 2 비대칭 서버 구조

그림 2의 구조 형태는 서버간에 데이터 베이스를 통해 데이터를 공유할 수 있으며, 하나의 서버에서 다른 서버로의 연결전환이 가능하다. 하지만 이에 따른 시간 동기화(Clock Synchronization), 서버들간의 교착 상태(Deadlock), 게임 데이터의 무결성(integrity)과 동시성(Concurrency)이 발생할 수 있다.

이 서버 구조는 대칭 서버와는 달리 서버들간의 데이터 공유가 가능하다. 하지만 클라이언트가 특정 서버에 많이 접속할 경우가 발생할 수 있으며, 대칭 서버와는 달리 서버간의 부하가 완전히 균등하게 분산되지는 않는다는 단점이 있다.

3. CORBA

CORBA는 컴포넌트간에 서로를 발견하고, ORB(Object Request Broker)상에서 상호 운용될 수 있게 설계되어 있어, 객체를 생성 및 삭제하고, 객체의 이름을 통해 접근하며, 저장소인 IR(Implementation Repository)과 IR(Interface Repository)에 객체를 저장하고, 객체의 상태를 외형화하고, 그들간의 특별한 관계를 정의하기 위하여 ORB관련 서비스의 확장된 집합, COS(Common Object Services)을 지정한다.

CORBA는 일반 객체를 생성한 다음 생성된 객체를 적절한 서비스로부터 다중 상속받게 함으로써 트랜잭션, 보안성, 폐쇄성 및 지속성을 갖게한다. 이는 일반적인 컴포넌트를 일반함수를 제공할 수 있도록 설계한 다음, 컴포넌트 구현시 또는 런타임동안 메소드들을 동적으로 발견할 수 있다는 것을 의미한다. 기존의 클라이언트/서버 환경에서는 이러한 것을 할 수 없다[1][2].

ORB는 RPC(Remote Procedure Calls), MOM(Message-Oriented Middleware), 데이터베이스 저장 프로시저(database stored procedures), Peer-to-Peer 서비스등을 포함하는 클라이언트/서버 미들웨어 대체안들보다 훨씬 정교하다[1].

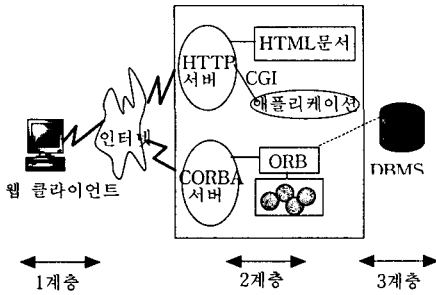


그림 3 CORBA기반 웹모델

4. 제안된 시스템의 구성도 및 동작 메커니즘

본 논문은 기존의 네트워크 게임 형태인 Client/Server 개념이 아닌, CORBA의 분산 Client/Server 개념을 이용한다. 제안된 시스템 구성도는 게임 서버 접속을 원하는 Client 측과 동적 스레드 생성 및 게임 객체의 동기화등을 관리하는 게임 서버, 그리고 network traffic으로 인한 Server/Client 간의 동기화 문제를 관리하는 CORBA의 ORB 엔진으로 분류한다.

네트워크 게임은 TCP/IP 기반의 게임과 IPX(Internet Packet Exchange) 기반의 게임으로 나눌 수 있다. 이는 하나 이상의 서버를 필요로 하며, 각각의 서버는 또한 하나 이상의 Client의 접속을 처리할 수 있다. 이러한 서버 구현은 다수의 Client 동시 접속에 의한 네트워크 부하 가중에 대한 해결책을 제시하지 못하고 있다. 또한, 접속 중인 서버가 이상이 발생하였을 경우 Client는 게임을 계속 유지시킬 수 없다.

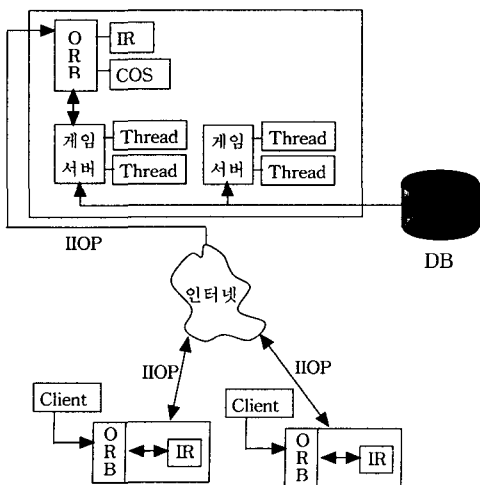


그림 4 제안된 시스템의 전체 구성도

위의 그림 4은 제안된 시스템의 전체 구성도로서 다수의 클라이언트 요청을 동시에 서비스해주는 문제 및 하나의 서버가 이상이 발생했을 경우, CORBA의 ORB엔진을 통해 다른 서버와 연결하여 접속한 클라이언트는 계속 게임을 할 수 있도록 한다. 물론, 게임을 계속하기 위해 각각의 서버는 계속적으로 정보를 제공 및 공유하여야 한다.

이 시스템의 동작 메커니즘을 보면 다음과 같다.

- ① 클라이언트가 서버를 접속할 경우, 클라이언트측 ORB IR에 그 클라이언트의 정보 및 게임 객체들의 데이터 정보를 등록한다.
- ② CORBA의 ORB엔진을 통해 클라이언트측 IR의 정보는 서버측 IR에 저장되어 있는 서버들의 정보를 참조하여 네트워크 게임이 이루어진다. 또한, 클라이언트측에 보여질 게임 데이터(Text, Graphic, Audio등)를 결정한다.
- ③ 클라이언트측으로 보내질 데이터가 결정되면, 게임 객체들은 시간순서대로 동기화시키며, 게임 속도 저하 및 네트워크 부하를 줄이기 위해 멀티스레딩 기법을 이용 또 다른 하나의 스레드를 dynamic하게 생성하여 connection을 시켜준다.
- ④ 서버마다 접속 임계치를 두어 임계치 이상의 클라이언트가 접속할 경우, 또는 접속 중인 서버에 문제가 발생할 경우 다른 서버로 routing을 시켜줌으로써 네트워크 및 시스템 부하를 줄일 수 있다

게임을 시작하기 위해, 클라이언트는 서버에 접속하여 사용자 인증 과정을 거친 후에, 게임그룹을 만들고, 게임을 시작한다. 게임이 시작되면 서버는 각 클라이언트의 정보를 유지하며, 클라이언트가 게임 중에 총을 발사하거나, 이동/채팅등의 이벤트를 발생시키면 서버에게 각 이벤트의 플러그와 정보값을 보내게 된다. 이러한 client/server 모델에서 클라이언트가 증가함에 따라 서버의 처리능력이 떨어지는데, 이를 방지하기 위해 본 논문에서는 CORBA의 ORB엔진을 게임서버마다 둬으로써 게임객체의 정보를 저장하고 있는 IR과 동시성 제어 서비스등을 제공하는 COS와 연결하며, 하나의 게임서버에 여러개의 thread를 생성하여 connection시켜준다. 하나의 게임서버에 임계치를 두고, 임계치 이상의 클라이언트 접속이 있을 경우나 서버에 문제가 발생할 경우 다른 서버로 연결시켜줌으로써 클라이언트는 게임을 계속 유지시킬 수 있다.

제안한 시스템에 CORBA 미들웨어 제품중의 하

나인 IONA의 Orbix[4]를 사용할 경우, Server Manager라는 창을 통해 서버를 공유할 수 있는 클라이언트 수를 제한하는 부분이 따로 있다.

여기서 컴퓨터 시스템 및 네트워크 부하를 감안하여 임계값을 정하고, 또한 서버 bind나 invoke는 option 사항에서 선택할 수 있다. 임계값이상의 클라이언트 접속이 있을 경우 더 이상 connection을 수용하지 않는다. 이때, 동적으로 스레드를 생성하거나 다른 서버로 routing을 시켜줌으로써 서버 및 클라이언트들을 좀더 효율적이고 안정되게 관리할 수 있다.

위에서 제시한 기능을 현재 Server/Client형태의 게임으로 구현시 개발자의 어려움과 각각의 플랫폼간의 환경 설정부분들에 많은 노력이 필요로 한다. 하지만 이러한 기능들을 CORBA로 대체할 경우 플랫폼간에 독립적이며, 프로그램 구현시 개발자의 어려움 또한 줄일 수 있다. 무엇보다도 CORBA의 최대 장점인 동적 메커니즘을 이용하여 클라이언트 접속을 관리한다면 시스템이 좀더 안정적이며 효율적일 것이다.

CORBA의 동시성 제어 서비스 API는 동시에 여러 객체들의 특정자원에 접근할 때 lock을 통해 원자성을 보장해주는 서비스이며 lock모드로는 read, write, intension read, intension write, upgrade 등이며 lock 조정자를 통해 설정된다[1][6].

이벤트 서비스(Event Service)는 버스상의 컴포넌트들이 특정한 이벤트에 대한 그들의 관심을 동적으로 등록 또는 등록 취소하는 것을 허용한다[1]. 공급자와 소비자 사이에 이벤트 채널(Event Channel) 객체를 만들어 비 동기적인 이벤트 통신을 가능하게 한다. 이벤트 채널안의 프락시 객체(Proxy Object)는 공급자로부터 받은 이벤트를 소비자에게 전송한다[3]. 이처럼 CORBA에서 제공하는 서비스를 이용, 함께 설계 및 구현한다면 좀더 안정적이며, 효율적일 것이다.

다수의 클라이언트가 동시접근시 서버내에서는 클라이언트 접속 수 즉, 임계치를 정해 각각의 스레드를 dynamic하게 생성하여 connection을 시켜주고, CPU 처리율, 게임 속도저하 또는 접속중인 서버에 이상이 발생할 경우 다른 서버로 routing 시켜줌으로써 동시접근제어 및 네트워크 부하를 효율적으로 관리할 수 있다.

5. 제안된 시스템의 특징 및 장·단점

제안된 시스템은 three-tier로서 이종의 분산된 환

경 하에서 응용 프로그램들을 서로 통합할 수 있는 CORBA를 사용하였기 때문에 하나 이상의 서버와 쉽게 연결할 수 있다. 즉, 클라이언트측은 여러 개의 이질적인 서버와 연결하기 위해 별도의 프로토콜이 필요 없이 CORBA가 제공하는 IIOP를 이용하고, 동적인 스레드 생성 및 다른 서버로 connection을 해 줌으로써 좀더 효율적인 동시성 제어 및 네트워크 부하를 줄일 수 있다.

또한 게임서버의 위치 투명성을 최대한 보장할 수 있는 네이밍 서비스(Naming Server)를 이용한다.

6. 결론 및 향후 연구 방향

본 논문은 CORBA를 사용한 네트워크 게임에 대한 제안이다. 이를 운영함에 있어서 게임 객체의 동시성 문제와 하나의 서버가 이상이 생겼을 경우 다른 서버와의 connection문제, 그리고 다수의 클라이언트 서버접속으로 인한 네트워크 부하를 줄이기 위해서 위에 제시한 시스템구성 및 동작 메커니즘을 통해 이를 해결한다.

또한, CORBA를 기반으로 하였기 때문에, 게임 객체를 생성한 다음 생성된 객체를 적절한 서비스로부터 다중 상속받게 함으로써 트랜잭션, 보안성, 폐쇄성 및 지속성을 갖게 한다.

앞으로 연구방향은 제안된 사항의 구현이며 여러 개의 서버를 네트워크 속도에 따라 클라이언트와 연결을 원활하게 하며, 클라이언트간의 동기화에 대한 해결책이 필요하다.

참고문헌

- [1] Robert Orfali and Dan Harkey, "Client/Server Programming with JAVA and CIRBA", John Wiley and Sons Inc., 1998.
- [2] Jon Siegel, "CORBA Fundamentals and Programming", John Wiley and Sons Inc., 1996.
- [3] 나길성, 이상호, "CORBA 이벤트 서비스의 구현 기법", 정보과학회 논문지:데이터베이스 제27권 제3호 pp.530-538, 2000.9.
- [4] IONA Technologies PLC. "Orbix 3.0.1 Programmer's guide", Feb. 1999.
- [5] 신동일, 신동규, "다수 사용자 기반의 온라인 게임 서버의 설계 및 제작", 전자공학회지, 제27권 제9호, pp954-960, 2000.9.
- [6] CORBA concurrency service, http://www.omg.org/technology/documents/formal/concurrency_service.htm.