

유무선 동시지원 게임 서버의 구현 및 성능분석

김용빈^o, 신동규, 신동일
세종대학교 컴퓨터공학과
{kimybin^o, shindk, dshin }@gce.sejong.ac.kr

Implementation and Performance Evaluation for a Wire/Wireless Game Server

Kim Yong Bin^o, Shin Dong Kyoo, Shin Dong Il
Dept. of Computer Engineering, Sejong University

요약(abstract)

최근 온라인 게임은 많은 호황을 누리고 있다. 현재는 PC 기반의 온라인 게임이 주류를 이루고 있지만 무선 단말기를 통하여 온라인 게임을 즐기는 사람들이 급증하는 추세이다. 기존의 온라인 게임서버의 구조를 살펴보면 각각의 게임의 환경에 맞게 서버를 설계하여 많은 시간과 자원의 낭비를 초래하게 된다. 유사한 Network Game을 지원하기 위해 헤더 영역과 Data영역을 분리 시켜 보다 유사한 게임들은 동일한 게임서버를 통하여 지원을 가능하게 하여 효율적인 게임서버를 운영할 수 있을 것이다. 따라서 본 논문에서는 유선과 무선상에서의 Network delay를 고려한 유무선 네트워크를 동시에 지원하는 Game Server 및 미들웨어를 구현하고 구현된 Game Server의 성능분석을 통해 본 논문에서 제안한 게임서버의 분산처리 성능을 평가하였다.

1. 서론

최근 온라인 게임은 많은 호황을 누리고 있다. 과거 리니지 게임부터 온라인 게임은 급성장하여 현재는 많은 온라인 게임이 존재한다. 하지만 이러한 온라인 게임들은 가정의 PC나 게임방같은 한정된 곳에서만 게임을 할 수 있다. 그러나 현재 PDA, 모바일 폰의 발전으로 지하철이나 버스에서도 무선 Network를 이용하여 World Wide Web Service, Vod등 서비스의 이용이 가능해졌다. 특히 단말기의 하드웨어 성능이 발전함에 따라 초기의 퍼즐위주의 단순한 게임에서 고사양의 하드웨어에 힘입어 점차 화려한 Graphic 기반인 슈팅이나, 액션게임이 가능해졌다[1][2][9].

특히, 현재까지 서비스되고 있는 거의 대부분의 PDA용 게임은 Stand Alone Game이 주종을 이루고 있으나, 값비싼 통신 요금이 점차적으로 인하되는 추세고, 주요 건물의 Hotspot 지역에서는 무선 LAN카드를 장착하여 인터넷을 이용할 수 있게 되면서, 점차적으로 Stand Alone Game에서 Network Game으로 발전되고 있는 추세이다.

대부분의 PC 사용자는 고사양의 하드웨어와 고속 인터넷망을 통하여 Game을 즐기므로, 저사양의 하드웨어와 저속의 인터넷망에서 Game을 플레이해야만 하는 무선 게임 사용자와의 동기화 작업은 필수적이다. 국내 대부분 CDMA망을 사용하여 최대 전송속도가 144kbps임을 감안할 때 수 Mbps급의 PC Network망과의 속도차이에 따른 PDA사용자와 PC사용자의 Game 환경은 크게 다를 수 밖에 없다[3].

따라서 PC사용자와 PDA 사용자간의 패킷 송수신시에 따른 Network Delay를 고려한 동기화 작업은 반드시 고려되어야 하는 사항이다. 또한, 모든 Game들이 각각의 Game을 지원하기 위해 서버를 개발하고 고성능 서버를

사용한다면 이 또한, 자원과 시간의 낭비일 수밖에 없다.

모든 Network Game을 지원하기 위해 헤더 영역과 Data영역을 분리시킴으로 유사한 게임은 동일한 게임서버를 통하여 지원을 가능하게 하여 자원과 시간의 낭비를 줄일 수 있도록 할 수 있다[4]. 따라서 본 논문에서는 유선과 무선상에서의 Network delay를 고려한 유무선 네트워크를 동시에 지원하는 Game Server를 구현하고 게임 서버의 성능분석을 통해 게임서버의 성능을 평가하였다.

2. 분산 서버 아키텍처 관련연구

클라이언트-서버 아키텍처의 경우 많은 수의 클라이언트가 참여할 수 있으나 MMORPG(Massively Multiplayer Online Role-Playing Game) 방식에서와 같이 하나의 게임에 수백 내지 수천 명이 참여할 경우 하나의 서버로 이를 담당하기 어렵게 된다. 따라서 서버의 작업량을 여러 서버에 나누는 방식이다.

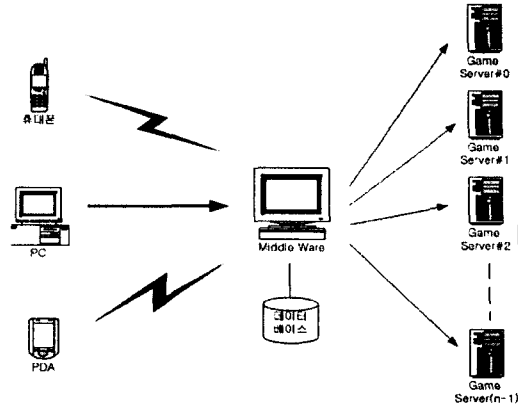
분산 서버 방식에는 부하분산 방식과 맵 서버 방식의 두 가지방식이 사용된다. 부하분산 방식의 경우에는 게이머의 서버 접속을 여러 대의 접속 서버가 담당하며 메인 서버가 이를 제어하는 방식이다. 부하 분산 방식은 비록 접속 기능을 여러 대의 서버에 나누었지만 하나의 메인 서버에 의해 게임이 진행되기 때문에 사용자의 수가 증가할 경우 새로운 한계에 부딪히게 된다.

맵 서버 방식은 게임 월드를 여러 개의 맵으로 나누고 게이머가 어느 맵에 있는가에 따라 해당 맵을 담당하는 서버에 접속하게 하는 방법이다. 게임 월드가 넓어질 경우 서버의 수를 증가시키면 되기 때문에 더 발전된 방식이다. 그러나 맵의 경계를 넘게 되면 해당되는 캐릭터를 다른 서버로 보내 주어야 하기 때문에 게임 진행이 지연될 가능성이 있다.

3. 유무선 동시지원 게임서버의 구축

3.1 유/무선 게임 서버의 전체 구조

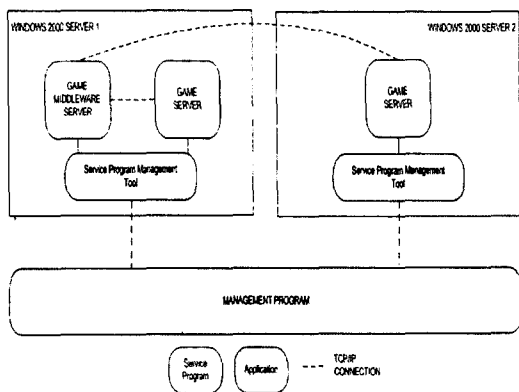
본 논문에서 제안한 유/무선 게임을 위한 Game Server의 전체 구조는 다음과 같다.[그림 1]



[그림 1] 유/무선 게임서버의 전체구조

각각의 게임서버는 미들웨어를 통하여 접속하게 되고 미들웨어에 등록된 게임서버의 Load Balancing을 측정하여 게임서버에 접속을 하는 구조이다.

3.2 Middleware를 통한 게임 서버의 전체구조



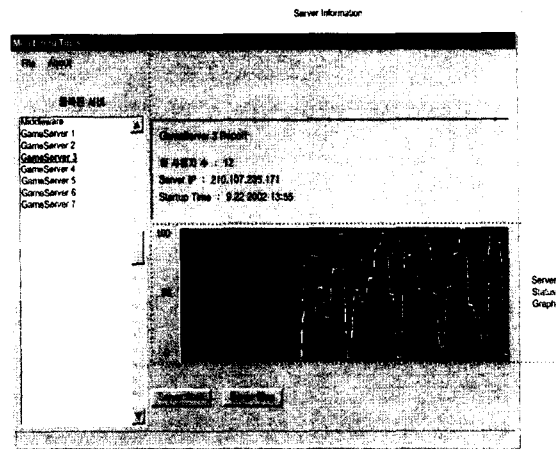
[그림 2] Game Middleware와 GameServer의 전체 구조

게임 서버와 Game Middleware가 운영되는 환경은 Windows 2000 Server이며, 각각의 서버들은 별도의 Management Program으로 원격지에서 특정 게임 서버와 Game Middleware를 통제하거나 또는 Server의 부하를 모니터링 할 수 있다.

Management Program과 Server사이에 "Service Program Management Tool"이란 중계 프로그램이 있는데 이는 Server가 Service프로그램으로 운영되기 때문에 중간에서 Management Program과 연결되어 Server를 관리한다.

Management Program은 다음과 [그림 3] 같은 GUI를 가지며 이를 통해 Game Server 및 Game Middleware

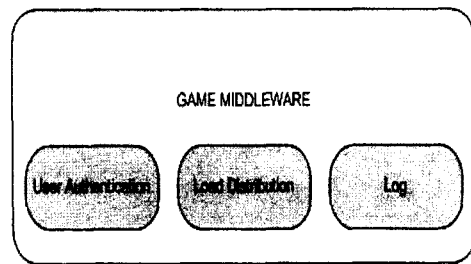
등을 모니터링 및 Server Start, Stop등의 기능을 제공한다.



[그림 3] Management Tools의 GUI

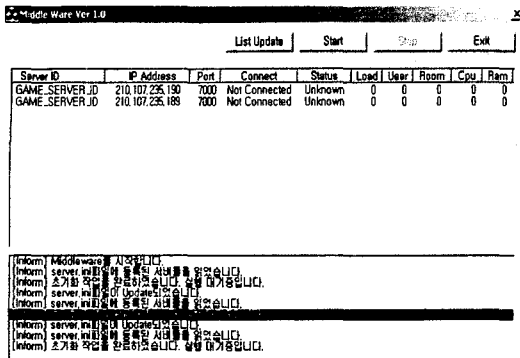
3.3 Game Middleware의 세부 구조

Game Middleware는 Game을 이용하여 User와 Game을 중계하는 게임 서버 사이에 존재하며 구조는 다음과 같다.



[그림 4] Game Middleware의 구조

Game User는 먼저 Game Middleware에 접속하여 사용자 인증 과정을 먼저 거치게 되며, 정상적인 사용자 일 경우 Login Time과 Logout Time을 Log File에 기록하며, User가 접속해야할 게임 서버를 선택하게 되는데 이때, Game Middleware는 현재 가동중인 게임 서버의 Load를 파악하여 User로부터 게임 서버를 선택하도록 한다. 이때 서버의 Load를 판단하는 요소로는 현재 게임 서버의 Resource Usage Rate, CPU Usage Rate, 게임 서버에 접속된 User 숫자 등의, 세 가지 요소를 종합적으로 판단하여 100분율로 환산하여, Load가 가장 적은 서버를 선택할 수 있도록 한다. 특히, Load가 80%가 넘는 Server는 User가 게임 서버를 선택하는 List에 나타나지 않도록 하여, 그 게임 서버에 더 이상 Load가 증가하지 않도록 한다.[5]



[그림 5] 미들웨어 인터페이스화면

4. 성능분석

4.1 성능 분석 환경

구현된 게임 서버의 성능을 측정하기 위해 멀티스레드를 이용한 가상 클라이언트 접속프로그램을 구현하였다. 임의의 값, 즉 접속인원을 입력한 후에 동작을 하게 되면 입력한 수 만큼의 스레드가 생성되어 미들웨어에 접속을 하게 되고 미들웨어를 통해서 등록된 게임서버의 Load Balancing을 측정하여 각각의 게임서버에 접속한다. 서버에 접속한 후 클라이언트에서 키 이벤트를 랜덤하게 발생하여 게임이 동작하는 형태와 유사하게 한다. 이러한 상태를 기반으로 Load 상태와, 메모리 점유율, CPU Rate, 네트워크 트래픽등을 측정하게 된다.

구현된 게임서버는 유/무선을 지원하기 때문에 무선 접속은 무선랜을 사용하여 서버에 접속을 하고 WindowCE3.0을 기반으로한 멀티 스레드 소켓통신을 이용하였다.[7][8] 유선 접속은 Visual Studio 6.0을 사용하여 멀티스레드 소켓프로그램을 작성하였다.

수행시간	1.3초
접속횟수	30회
재시도횟수	3회
서버갯수	5개

[표 1] 성능평가 요소 및 환경

4.2 성능 분석 결과

본 논문에서 제안된 게임서버의 분산처리 능력을 평가하기 위해서 총 5대의 컴퓨터를 이용하여 테스트를 하였다. 다음[표 2]를 보면 100명의 사용자가 접속했을때 각 서버의 CPU Rate와 메모리 액세스등을 나타내고 있다. 100명의 유저가 접속했을때 미들웨어에서 게임서버의 Status값을 체크하여 각 게임서버에 접속을 하게 되며 컴퓨터 성능에 따라 접속인원, CPU, 메모리 점유율등의 값이 약간의 차이는 보여주지만, 구현된 서버가 분산처리를 효과적으로 해준다는 것을 알수 있다.

서버	접속인원	CPU Usage Rate	Resource Usage Rate	Status
Server#1	40명	25%	20%	GOOD
Server#2	50명	15%	13%	HIGH
Server#3	30명	30%	27%	GOOD

[표 2] 게임서버의 성능평가 결과

- Game Server#1 P3, CPU : 1G, RAM: 512MB
- Game Server#2 P4, CPU : 1.8G RAM : 512MB
- Game Server#3 P3, CPU : 866 RAM : 256MB

5. 결론

본 논문에서는, PC와 PDA 사용자가 같은 Session 내에 원활하게 동일한 Game을 즐길 수 있도록 하는 유선 네트워크를 동시에 지원하는 Game Server와 이를 위한 Game Middleware를 구현하였다. 구현된 게임서버의 Load Balancing을 검사하여 Middleware에서 자동으로 서버에 연결시켜주는 기능을 하였으며, 가상 클라이언트 시뮬레이션 프로그램을 통해 분산기능을 테스트하였다. 현재는 슈팅, 보드, 아케이드, 시뮬레이션과 같은 Game Data의 발생량이 작은 Game에 대해서만 고려를 하였으나 앞으로, 서버 안정화 작업과 더불어 MMORPG와 같은 거대한 Game을 지원할 수 있도록 확장해야할 필요성이 있다.

6. 참고문헌

- [1] Windows CE 개발자사이트, <http://wecom.dstcorp.com>
- [2] CEcraft, <http://www.cecraft.com>
- [3] Qualcomm, <http://www.cdma.com>
- [4] PDA와 게임, <http://www.pdacircle.com/>
- [5] Ben Calica, "Multi-player Lobbying", Gamasutra, Sep. 4. 1998
- [6] S. Singhal and M. Zyda, Networked Virtual Environments Design and Implementation, Addison Wesley, 1999
- [7] Boling, "Programming Microsoft Windows CE second edition", Microsoft press, 2001
- [8] 고재관, "Mobile PDA Programing", 삼각형프레스, 2001.
- [9] 엠브레인, "게임성향 조사", <http://www.embrain.com>