

대규모 사용자들을 수용하기 위한 효율적인 서버 확장이 가능한 온라인 게임 서버 구조

김정훈⁰
(주)소프트젠
crony@softzen.co.kr

Efficient and scalable Online Game Server Architecture for Massive Users.

Jeong-Hoon Kim⁰
SoftZen Co., Ltd.

요 약

본 논문에서는 사용자 수 증가에 의한 네트워크 게임 서버 부하를 줄일 수 있는 효율적인 서버 구조를 제시하였으며, 이를 위해 대칭형 분산 서버 구조와 비대칭형 분산 서버 구조를 적절히 혼용하는 방식을 채택하였다. 대칭형 분산 서버 구조 방식은 서버들간의 부하 분배를 효율적으로 운영할 수 있다는 장점이 있고 비대칭형 분산 서버 구조는 한 서버 그룹 내에서 서로 다른 게임 서버로 자유롭게 이동할 수 있다는 장점이 있다. 본 제안 서버 구조에서는 이러한 장점들을 적절히 이용하여 온라인 게임 서버의 효율성을 향상시켰다.

Abstract

In this paper, I tried to propose efficient online game server architecture that decrease server overload. I used replicated distributed sever type together with non-replicated distributed sever type. Replicated distributed sever type has the advantage to maintain efficient distribution of server overload and non-replicated distributed sever type has the advantage to move a game server to a different game server freely in one server group. I used these advantages suitably in this proposal, and I will show the efficiency of online game server architecture.

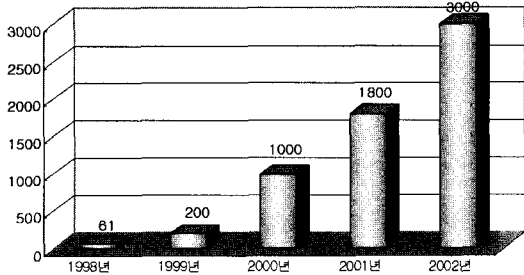
1. 서론

온라인 게임이란 다수의 사용자가 동시에 인터넷에 접속하여 상호 관계를 맺으면서 즐기는 게임이다. 지난 1980년 영국의 바틀 트루쇼(Bartle Trubshaw)가 처음으로 텍스트 머드 게임을 선보였는데, 이것을 온라인 게임의 시작으로 볼 수 있으며, 1988년 영국의 알랜 콕스(Alan Cox)가

AberMUD를 개발, 서비스하면서 일반에게 널리 알려지기 시작했다. 국산 온라인 게임의 효시는 1993년 개발된 텍스트 머드 게임인 주라기 공원이다. 최근에는 텍스트 머드 게임에서 한단계 발전된 그래픽 머드 게임과 3D 머드 게임이 일반화되어 많은 사용자들을 온라인 게임 속으로 유인하고 있다[1].

국내 게임 시장의 규모는 2004년의 경우 출하규모 기준으

로 1조 7,632억원, 최종소비자 시장 기준으로 4조 2,657억원이 될 전망이다. 각 플랫폼별 2004년 출하시장 규모는 온라인 게임이 연평균 22%의 성장으로 4,636억원을, PC게임도 연평균 12% 이상의 성장으로 2,698억원을, 비디오 게임은 2,404억원에 이를 것으로 전망된다[2].



(그림 1) 온라인게임시장 (단위: 억원)

청소년들의 점유율로만 인식되어 온 온라인 게임이 이제 발전 가능성이 아주 높은 산업으로서 주목 받기 시작했고, 최근에는 IT, BT에 이어 CT를 정부 차원에서도 집중적으로 지원하기에 이르렀다. 여기에 화답이라도 하듯 온라인 게임 사용자들은 기하급수적으로 늘어나고 있는데, 이는 인터넷 인프라의 급속한 발전과 PC 방 보급에 기인한다고 볼 수 있을 것이다.

그러나 급속히 증가하는 사용자들을 수용하기 위한 온라인 게임 서버 기술은 초보적인 수준에 머물고 있어서 대부분의 온라인 게임 사들은 단순한 방법으로 서버를 확장하고 서비스하고 있는 실정이다.

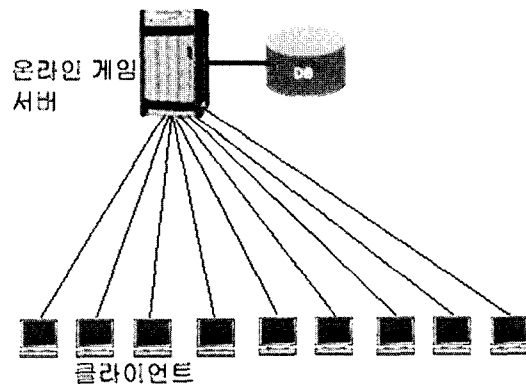
이에 본 논문에서는 기존의 온라인 게임 서버 구조를 살펴보고 각각의 장단점을 분석해서 급속히 증가하는 사용자들을 효율적으로 수용하기 위한 서버 확장 구조를 제안한다. 효율적으로 서버가 확장될 수 있다면 투자대비 성능비를 한층 더 높일 수 있을 것이다. 본 논문은 다음과 같이 기술된다. 2장에서는 일반적인 온라인 게임 서버 구조를 살펴보고, 3장에서는 본 논문에서 제안하는 효율적인 온라인 게임 서버 구조에 대해서 상세히 기술한다. 마지막으로 4장에서 본 논문의 결론을 맺는다.

2. 온라인 게임 서버 구조

2.1 중앙 집중형 구조(단일 서버 구조)

현재 온라인 게임에서 클라이언트와 서버간에 많이 사용되고 있는 연결 방식은 2-tier 방식이다. 2-tier 방식은 인터넷을 통해 클라이언트와 서버간에 다대일 관계를 유지하는데, 이러한 방식을 중앙 집중형 구조 또는 단일 서버 구조라고 한다.

중앙 집중형 구조인 경우 서버의 구현과 운영이 쉬운 장점이 있으나, 수용 가능한 사용자 수가 제한되어 있어 사용자 수 증가에 따른 시스템 부하를 적절히 감당해 낼 수 없다는 단점이 있다. 또한 CPU 및 네트워크 트래픽(network traffic), 메모리 사용량은 사용자 수 증가에 따라 급격히 증가할 수 있는데 중앙 집중형 구조인 경우는 규모의 확장에 따른 대비를 적절히 할 수 없다는 단점이 있다[3,4,5].



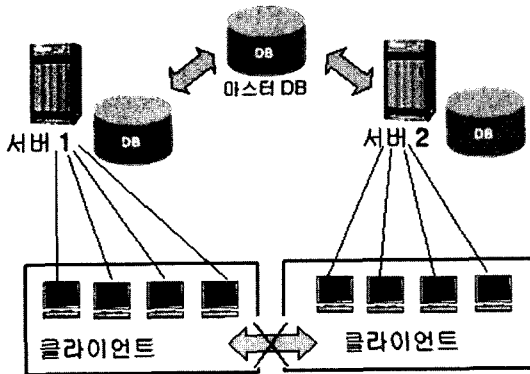
(그림 2) 중앙 집중형 구조

2.2 대칭(Replicated) 형 분산 서버 구조

분산 서버 구조는 중앙 집중형 서버 구조와는 달리 서버가 한 대 이상일 때 선택될 수 있는 구조이다. 온라인 게임을 즐기는 사용자들이 급증하면서, 단일 서버로만 운영되는 시스템이 한계에 달하게 되었는데, 이러한 이유로 분산 서버 구조가 제안되었다. 분산 서버 구조는 2개 이상의 서버로 구성되며, 이 서버들은 서버들 간에 데이터 공유가 되는 방식과 그렇지 않은 방식으로 나누어 질 수 있다. 데이터가 공유되는 방식은 대칭 서버(Replicated sever)라고 하고 그렇지 않은 경우를 비대칭 서버(Non-Replicated sever)라고 한다. 비대칭 서버에 대해서는 2.3절에서 살펴보기로 한다.

다음 그림에서 보여주는 대칭형 분산 서버 구조는 단일 동일한 서버 구조를 가진 시스템을 2개 이상 연결하여, 클라이언트로 하여금 서버에 접속할 수 있는 사용자 수를 늘

리케 해준다. 이 구조에서는 동일한 서버라고 해도 각각에 연결되어 있는 클라이언트는 다른 서버의 클라이언트와 커뮤니케이션을 할 수 없다는 단점이 있다.



<그림 3> 대칭형 분산 서버 구조

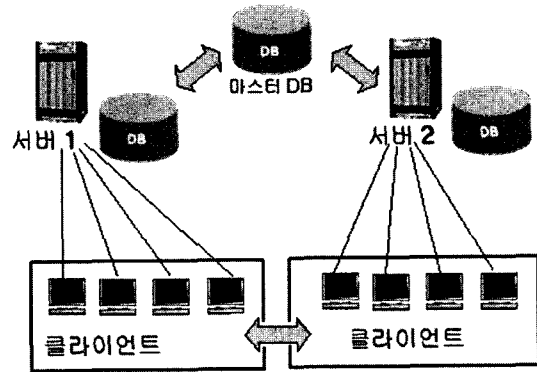
대칭형 분산 서버 방식은 동일한 서버를 연결하여 서비스 가능한 사용자 수를 늘리는 방식이므로 각 서버는 동일한 맵 데이터를 가지게 된다. 서버는 클라이언트가 실행하는 도중에 클라이언트의 관리를 다른 서버로 이전하지 않는다.

대칭형 분산 서버 방식을 선택하면 각 게임 서버에 걸리는 부하가 대체적으로 균등하게 분산되는 장점이 있다. 결국 각 서버의 평균 부하가 감소되어, 증가되는 서버의 수만큼 많은 사용자를 수용할 수 있게 된다. 하지만 각 게임 서버간의 동기화 문제가 발생할 수 있는 단점이 있는데 이는 어떤 서버에 접속하여도 맵 데이터와 게임 데이터를 동일한 게임 환경으로 사용자에게 서비스되어야 하기 때문에 발생된다.

2.3 비대칭(Non-Replicated) 형 분산 서버 구조

비대칭 형 분산 서버 구조는 서버간에 데이터베이스를 통해 데이터를 공유할 수 있으며, 하나의 서버에서 다른 서버로의 연결전환이 가능하다. 그러나 이에 따른 시간 동기화(Clock Synchronization), 서버들 간의 교착 상태(Deadlock), 게임 데이터의 무결성(integrity)과 일관성(Consistency) 문제가 발생할 수도 있다. 또한 클라이언트가 특정 서버에 많이 접속할 가능성이 있으며, 대칭형 분산 서버와는 달리 서버간의 부하가 균등하게 분산되지 않을 수 있다는 단점이 있다.

비대칭형 분산 서버 구조는 단일 서버 구조와 다중 서버 구조, 둘의 장점을 취한 형태이다. 단일 서버 구조는 중앙

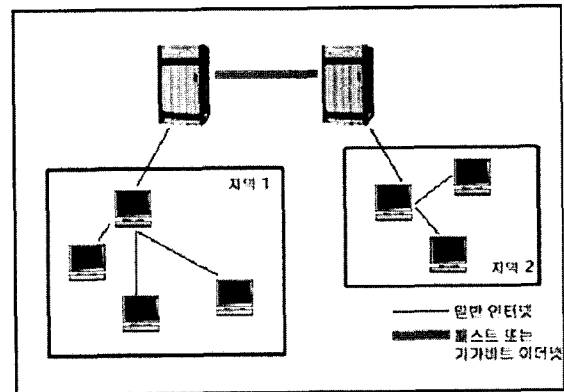


<그림 4> 비대칭형 분산 서버 구조

집중의 구조로 운영 및 관리가 용이한 장점이 있지만, 많은 사용자를 수용하기 위한 확장성이 낮았다. 반면 분산 서버 구조는 서버를 쉽게 추가함으로써 사용자 측면의 확장성을 높였지만, 서버간의 복잡한 동기화 과정과 데이터의 동시성 보장 등이 오히려 부담이 되는 단점이 있었다.

2.4 피어 투 피어(Peer to Peer) 기반의 서버 구조

한 지역에 제일 처음 로그인하게 되면 서버는 그 클라이언트에게 맵 정보를 전송한다. 그리고 그 이후에 그 지역에 접속하는 클라이언트는 온라인 게임의 DNS 서버가 초기에 접속한 클라이언트에 연결시킨다. 초기 클라이언트가 일종의 서버 역할을 하는 것이다. 스타크래프트 배틀넷에서 이미 실행된 사항으로 6~8개 정도의 클라이언트가 무리없이



<그림 5> 피어 투 피어 기반의 서버 구조

접속되어 게임이 진행될 수 있다.

피어 투 피어 기반의 서버 구조는 게임 내에서의 아이템 보안이 문제될 수 있다. 아이템 보안 문제는 지역 내에 있는 클라이언트가 서버 역할을 하기 때문에 발생한다. 또 이 구조는 어느 한 클라이언트가 아이템을 보유하다가 메인 서버에 그 내용을 반영하기 전에 접속이 끊겼을 경우에도 문제가 될 수 있는 구조이다.

이러한 문제점을 해결하기 위해 게임내의 아이템 정보를 게임에 참여하는 모든 클라이언트가 같이 보유하고 있다가 어느 한 클라이언트와의 접속이 끊어지면 다른 클라이언트의 정보를 서로 비교, 가장 최근의 타임 스탬프(time stamp)가 찍혀 있는 정보를 서버에 반영시키는 방법을 선택하기도 한다.

3. 효율적인 확장이 가능한 서버 구조 제안

제안 시스템은 대칭형 분산 서버 구조를 변형한 것이다. 효율적인 로드 밸런싱을 위하여 서버 그룹(Server Group)이라는 개념을 도입한다.

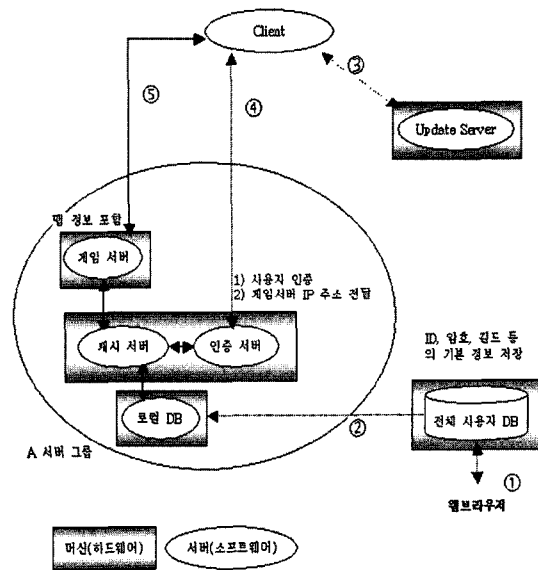
서버 그룹은 사용자들에게는 하나의 월드 개념으로 보이지만 각 서버 그룹은 업데이트 서버, 로그인 서버, 게임 서버, 캐시 서버, 로컬 데이터베이스 서버 등의 요소로 나누어진다. 제안 시스템 구조에서의 특징은 증가하는 사용자들을 수용하기 위해 게임 서버만을 추가로 설치하면 된다는 것이다. 추가되는 게임 서버가 많아져서 캐시 서버에 부하가 걸리게 되면 그때 또 다른 서버 그룹을 추가한다. 이렇게 함으로써 최소한의 비용 투입으로 최대의 효과를 거둘 수 있다.

또한 비대칭형 분산 서버 구조의 단점인 시간 동기화 문제나 서버들의 교착상태, 게임 데이터의 무결성과 일관성 문제도 본 제안 서버 구조에서는 해결된다. 본 제안 서버 구조에서는 서버간의 이동을 자유롭게 허용하지만 다수의 클라이언트와 연결되어 있는 게임 서버들은 캐시 서버를 통해 동일한 로컬 데이터베이스를 사용하므로 비대칭형 분산 서버 구조에서 나타나는 문제들을 해결할 수 있다.

3.1 서비스 초기 시스템 구성도

서비스 초기에는 많은 장비가 투입될 수 없다. 게임의 성공 여부를 판단하기 힘든 시점이기 때문에 최소한의 장비

만이 투자되어 서비스되어야 한다. 본 제안 시스템은 초기에 5개의 서버 머신(machine)이 설치된다. 물론 각 서버들은 물리적으로 같은 머신에 설치될 수도 있다. 이 경우 5개 미만의 머신으로도 초기에 서비스가 가능할 것이다. 다음은 이 내용을 그림으로 표현한 것이다. 웹 브라우저를 통해 가입된 회원 정보는 마스터 데이터베이스에 저장된다. 아래 그림에서 점선은 필요에 의해 연결 후 끊어지는 것을 표현하고 실선은 계속적으로 연결되어 있는 상태를 표현한다.



〈그림 6〉 초기 서비스 서버 구조

1단계

사용자는 웹 브라우저를 통해 회원에 가입한다. 사용자는 회원 가입 시 서버 그룹을 선택하는데, 선택된 서버 그룹 정보를 포함하여 회원정보는 마스터 데이터베이스에 저장된다.

2단계

사용자가 선택한 서버 그룹의 로컬 데이터베이스에 회원 정보가 전송된다.

3단계

클라이언트는 업데이트 서버에 연결해서 변경사항(추가된 서버 그룹 리스트 정보나 클라이언트의 업데이트 정보) 있다면 변경사항을 받는다. 이 작업이 끝나면 업데이트 서버와는 연결을 끊는다.

4단계

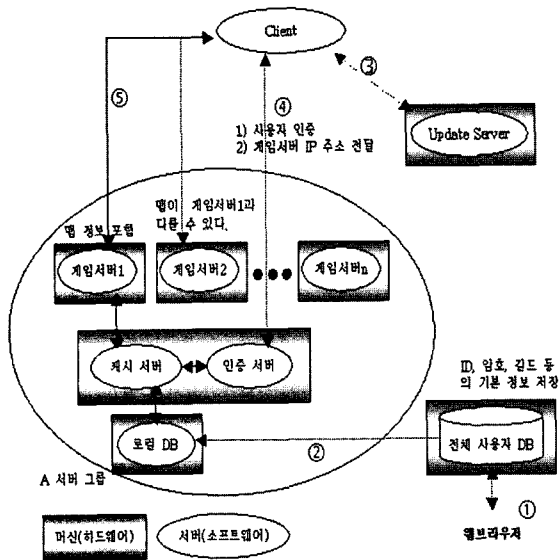
클라이언트는 인증서버에 연결되어 로컬 데이터베이스 서버를 통해 인증을 받는다.

5단계

성공적으로 인증이 끝나면 게임 서버에 연결된다.

3.2 하나의 서버 그룹 내에서의 서버 확장 시

위 3.1의 단계에서 사용자들이 증가하게 되면 이를 수용하기 위해서 다음 그림과 같이 게임 서버를 추가한다. 다음 그림은 n 개의 게임 서버를 추가로 설치한 그림이다. 이 구조에서는 게임 서버1과 게임 서버2가 서로 다른 맵 정보를 가질 수 있다. 따라서 이 부분만을 보면 위 2.3 절에서 살펴본 비대칭형 분산 서버 구조가 된다. 사용자들은 게임 서버 1에 접속되었다가 자연스럽게 게임 서버2로 접속이 바뀔 수 있는데 이때의 맵 이동은 내부적으로 이루어지기 때문이다. 이렇게 서버 접속이 변경되는 것을 사용자가 알게 구현할 수도 있고 프로그램에서 자동으로 처리해주어 모르게 구현할 수도 있다. 예를들어 게임서버1이 던전1의 맵을 가지고 있고 게임서버 2가 던전2의 맵을 가지고 있다면 어떤 사용자가 던전1에서 던전2의 입구를 통해 이동하면 자연스럽게 서버를 이동하게 되는 것이다. 게임을 같이 진행하고 있는 다른 사용자들이 보기에는 단지 다른 장소로 이동하는 것처럼 보일 것이다. 만약 본 게임으로 들어오기 전 대기



〈그림 7〉 게임 서버의 추가

실과 같은 부분이 있다면 이 대기실에서 게임 서버를 선택하게 할 수도 있다. 결국 이 과정은 사용자가 직접 맵을 선택하는 형태이다. 이렇게 함으로써 로드 밸런싱을 자연스럽게 유도할 수 있고 더 많은 사용자들을 수용할 수 있게 된다. 게임 서버들간의 아이템은 그대로 유지되어 지고 아이템 변동 내용은 즉시 캐시서버를 통해 데이터베이스에 반영된다. 아래 그림에서 점선은 필요에 의해 연결 후 끊어지는 것을 표현하고 실선은 계속적으로 연결되어 있는 상태를 표현한다.

1단계

사용자는 웹 브라우저를 통해 회원에 가입한다. 사용자는 회원 가입 시 서버 그룹을 선택하는데, 선택된 서버 그룹 정보를 포함하여 회원정보는 마스터 데이터베이스에 저장된다.

2단계

사용자가 선택한 서버 그룹의 로컬 데이터베이스에 회원 정보가 전송된다.

3단계

클라이언트는 업데이트 서버에 연결해서 변경사항(추가된 서버 그룹 리스트 정보나 클라이언트의 업데이트 정보) 있다면 변경사항을 받는다. 이 작업이 끝나면 업데이트 서버와는 연결을 끊는다.

4단계

클라이언트는 인증서버에 연결되어 로컬 데이터베이스 서버를 통해 인증을 받는다.

5단계

성공적으로 인증이 끝나면 게임 서버에 연결된다.

6단계

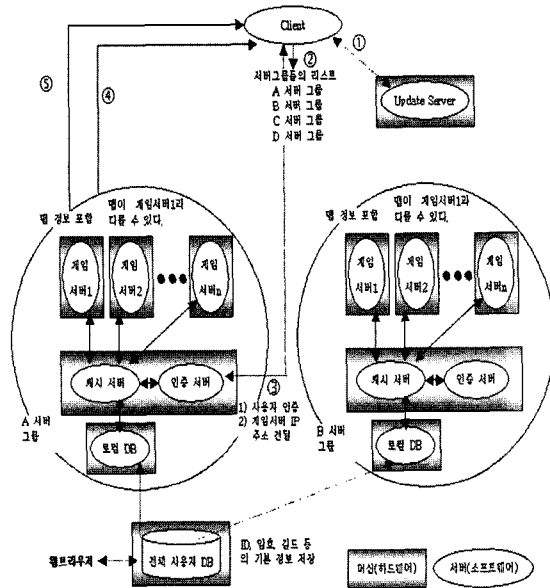
만약 사용자들이 다른 게임 서버에 접속하고 싶으면 한 서버 그룹 내의 타 게임 서버에 자유롭게 접속할 수 있다.

3.3 별개의 서버 그룹이 확장될 경우

별개의 서버 그룹이 추가되는 구조에서는 한 서버 그룹 내에서의 자유로운 이동은 허용하나 타 서버 그룹으로의 이동은 허용되지 않는다. 만약 A 서버 그룹에서 게임을 하던 사용자가 B 서버 그룹에서 게임을 하고 싶다면 A 서버 그룹에서 획득한 아이템이나 경험치를 B 서버 그룹에서는 포기해야 한다. 한 서버그룹에서는 아이템 이동이 허용되

나 다른 서버 그룹으로 이동시에는 기존 아이템들을 사용할 수 없다는 말이다. 현재 엔씨소프트의 리니지도 이와 같은 형태로 구현되어 있다.

이와 같은 구조는 효율적인 서버 확장을 지원하면서도 하나의 서버 그룹 내에서는 자유로이 게임 서버 간에 이동을



〈그림 8〉 별개의 서버 그룹이 추가된 경우

할 수 있다는 장점이 있다. 서버 그룹을 독립적으로 운영함으로써 A 서버 그룹 사용자는 B 서버 그룹에서 다른 게임을 즐기는 것과 같은 부가적인 효과를 누릴 수 있다. 아래 그림에서 점선은 필요에 의해 연결 후 끊어지는 것을 표현하고 실선은 계속적으로 연결되어 있는 상태를 표현한다.

1단계

사용자는 브라우저를 통해 회원에 가입한다. 사용자는 회원 가입 시 서버 그룹을 선택하는데, 선택된 서버 그룹 정보를 포함하여 회원정보는 마스터 데이터베이스에 저장된다.

2단계

사용자가 선택한 서버 그룹의 로컬 데이터베이스에 회원 정보가 전송된다.

3단계

클라이언트는 업데이트 서버에 연결해서 변경사항(추가된 서버 그룹 리스트 정보나 클라이언트의 업데이트 정보)

있다면 변경사항을 받는다. 이 작업이 끝나면 업데이트 서버와는 연결을 끊는다.

4단계

사용자는 서버들의 리스트 중 하나를 선택한다(A서버 그룹, B서버 그룹, C서버 그룹 등).

5단계

클라이언트는 선택된 서버 그룹의 인증 서버에 연결되어 로컬 데이터베이스 서버를 통해 인증을 받는다.

6단계

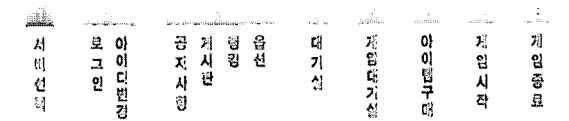
성공적으로 인증이 끝나면 게임 서버에 연결된다.

7단계

만약 사용자들이 다른 게임 서버에 접속하고 싶으면 한 서버 그룹 내의 타 게임 서버에 자유롭게 접속할 수 있다.

3.4 게임 흐름도

사용자 측면에서 보면 본 제안 시스템의 서버 구조는 다음 그림과 같은 게임 흐름도를 유지한다.



〈그림 9〉 제안 시스템 게임 흐름도

사용자는 먼저 서버를 선택한다. 사실 서버는 앞에서 살펴본 바와 같이 업데이트 서버, 로그인 서버, 게임 서버, 캐시 서버, 로컬 데이터베이스 서버 등의 요소로 이루어지지만 사용자가 보기에는 하나의 머신으로 구성되어 있다고 생각할 것이다. 둘째, 아이디와 암호를 입력함으로써 인증처리를 하고 게임 대기실에서 적절한 게임 서버를 선택한다. 셋째, 선택된 게임 서버에서 게임을 하다가 다른 게임 서버로 접속하고 싶으면 다시 대기실로 나와 다른 게임 서버를 선택할 수 있다. 구현상의 문제가 있겠지만 이 경우 굳이 대기실로 나오지 않고 자연스럽게 다른 게임 서버로 접속할 수도 있다.

4. 결론

2장과 3장에서 소개한 여러가지 서버 구조들을 서로 비교 및 요약하면 다음 표와 같다.

형태	장점	단점
① 중앙 집중형 구조	<ul style="list-style-type: none"> • 서버 구현과 운영이 용이 • 소수의 사용자일 경우 적합 	<ul style="list-style-type: none"> • 증가하는 사용자들 감당하지 못함 • 규모가 확장될 경우 부적절
② 대칭형 분산 서버 구조	<ul style="list-style-type: none"> • 대체적으로 로드가 분산됨 	<ul style="list-style-type: none"> • 서버간의 커뮤니케이션이 불가
③ 비대칭형 분산 서버 구조	<ul style="list-style-type: none"> • 다른 서버로 연결이 가능 • 로드 분산 	<ul style="list-style-type: none"> • 시간 동기화 문제 • 게임 데이터의 무결성과 일관성이 문제됨 • 무하가 균등하게 분산되지 않을 수도 있다
④ 피어 투 피어 기반의 서버 구조	<ul style="list-style-type: none"> • 로드 분산 	<ul style="list-style-type: none"> • 아이템 보안 문제
⑤ 본 논문의 제안 서버 구조	<ul style="list-style-type: none"> • 로드 분산 • 다른 서버로 이동 가능 • 증가하는 사용자들을 위한 확장성이 좋음 • 초기 투자비용이 저렴 	<ul style="list-style-type: none"> • 서버 그룹간의 커뮤니케이션 불가

〈표 1〉 각 서버 구조별 장단점

본 논문에서는 사용자 수 증가에 의한 네트워크 게임 서버 부하를 줄일 수 있는 효율적인 서버 구조를 제시하였으며, 이를 위해 대칭형 분산 서버 구조와 비대칭형 분산 서버 구조를 적절히 혼용하는 방식을 채택하였다.

대칭 서버 구조 방식은 서버들간의 부하 분배를 효율적으로 운영할 수 있다는 장점이 있었고 비대칭형 분산 서버 구조는 한 서버 그룹 내에서 서로 다른 게임 서버에 자유롭게 이동할 수 있다는 장점이 있었다. 본 제안 서버 구조에서는 이러한 장점들을 적절히 취함으로써 온라인 게임 서버로는 최적의 효율을 나타낼 수 있게 하였다. 또한 본 제안은 초기 구축비가 저렴하여 손쉽게 도입할 수 있으며 대규모 사용자들을 무리없이 수용할 수 있다는 장점이 있다.

대한민국에서 생산 또는 제작되는 제품들 중 세계 시장 1위의 제품은 흔하지 않다. 반도체 및 LCD, 조선 등 소수의 업종이나 제품만이 세계 시장에서 1위를 차지하고 있을 뿐이다. 특히나 제조 산업이 아닌 서비스 산업이나 문화 산업 중 세계 1위를 노크할 수 있는 분야는 단연 게임 분야가 될 것으로 생각된다. 게임 분야에서도 온라인 게임 분야는 가히 세계적이라고 표현해도 좋을 것이다. 물론 사회적인 부작용도 없지 않다. 그러나 온라인 게임의 부정적인 요소들을 하나하나 해결해 나간다면 온라인 게임 분야는 상당히 희망적인 분야임에 틀림없다. 우리나라에서 개발되는 온라인 게임이 전세계인이 즐긴다면 얼마나 멋진 일이 되겠는

가.

이를 위해 확실한 진입장벽을 구축하는 것은 시장 선점의 효과를 오래 누릴 수 있기 위한 좋은 방안이다. 확실한 진입장벽 중에 하나가 바로 서버 구조 기술이다. 얼마나 많은 동시 사용자들을 효율적으로 수용하느냐는 온라인 게임 분야의 핵심적인 부분이라고 해도 과언이 아닐 것이다. 우리나라의 온라인 게임을 전세계인이 즐기는 미래를 예상해본다.

참고문헌

- [1]. 한국게임산업개발원, 온라인게임서버 기술, 대한민국 게임백서, 564-575, 2002
- [2]. 정보통신부 한국첨단게임산업협회, 세계모바일게임 시장, 게임산업연차보고서, 173-193, 2002
- [3]. 구본철외7, 게임프로그래밍 전문가 수험대비서, 형설출판사, pp.353-396, 2002
- [4]. <http://www.synczone.net/>
- [5]. Wen tong Cai, Percival Xavier, Stephen J. Turner, Bu-Sung Lee, "A Scalable Architecture for Supporting Interactive Games on the Internet, 16th Workshop on Parallel and Distributed Simulation", p60, May 12 - 15, 2002
- [6]. Edward Chang, Hector Garcia-Molina, "Reducing Initial Latency in Media Servers, IEEE Multimedia", Vol. 4, No. 3, pp. 50-61, July-September 1997
- [7]. Christoforos E. Kozyrakis, David A. Patterson, "A New Direction for Computer Architecture Research", IEEE COMPUTER, Vol. 31, No. 11, pp. 24-32, November 1998
- [8]. H.I. Bozma, J.S. Duncan, "A Game-Theoretic Approach to Integration of Modules", IEEE TRANSACTION ON Pattern Analysis And Machine Intelligence, Vol. 16, No. 11, pp. 1074-1086, November 1994



김정훈

1991.2 서울시립대학교 컴퓨터통계학과 졸업(이학사)
1994.2 연세대학원 컴퓨터과학과 졸업(공학석사)
1994.1~1999.5 ㈜현대전자, ㈜현대정보기술
1999.6~2001.11 ㈜엔씨소프트
2002.3~현재 ㈜소프트젠 이사
관심분야 : 온라인게임, 모바일게임, 인터넷, 하이퍼미디어
