온라인게임 서버에서의 효율적인 클라이언트 접속 처리를 위한 비대칭 분산형 다중 서버 구조

황 도 연[†] · 이 남 재[††] · 곽 훈 성[†††]

요 약

온라인 게임 시스템은 크게 서버와 클라이언트 두 부분으로 나누어 볼 수 있다. 클라이언트는 게임 서버와의 접속 그리고 서버로부터 전송되는 패킷 분석 등의 역할을 하며, 게임 서버는 사용자 정보와 데이터베이스 등의 관리를 수행한다. 특히 게임 서버는 새로운 클라이언트가 게임을 수행하기 위해 서버로의 접속을 허용할 경우 기존에 연결된 사용자와의 온라인 연결을 유지하면서 새로운 클라이언트의 접속 요청을 받아들여야 한다.

본 논문에서는 서버의 다중 작업 처리를 위한 방법 중 프로세스(Process) 방법과 쓰레드(Thread) 방법의 처리 형태를 비교한 후, 현재 대부분의 게임 서버에 적용되어 있는 비대칭 분산형 구조에 적합한 비대칭 분산형 다중 서버 구조를 제안하였다.

제안한 비대칭 분산형 다중 서버 구조는 서버의 기능을 로그인 서버, 게임 서버, 통신 서버, 데이터베이스 서버 형태로 세부적으로 분리하여 각각의 기능을 독립적으로 수행한다. 따라서 다른 구조들과 비교하였을 경우 경제적, 기능적으로 더 나은 성능을 보이며 특히 서버의 안정성과 확장성이 향상되었다.

키워드 : 온라인게임, 클라이언트, 쓰레드, 프로세스

Asymmetric distributed multi server architecture for efficient method of client connection process at online game servers

Doh-Yeun Hwang[†] · Nam-Jae Lee[††] · Hoon-Sung Kwak[†††]

ABSTRACT

The online game system could be largely divided into two parts: servers and clients. Clients accesses to a game server and analyzes the packets transmitted from a server. A game server manages users information and database. If a game server allows a new client to access the server to execute a game, it should accept the access request of the new client maintaining the online connection of the existing users.

In this paper, we compare Process method and Thread method within the multiple jobs process methods of a server. Then we propose an asymmetric distributed multi server architecture that is adequate to asymmetric distributed architecture that is widely applied to most game servers.

The proposed asymmetric distributed multi server architecture includes login server, game server, communication server and database server to perform its own feature independently. Comparing with other architectures, it shows better performance economically and technically. Especially it improves the stability and expandability of a server.

Key Words : On-line Game, Client, Thread, Process

1. 서 론

현재 컴퓨터와 초고속 통신망의 비약적인 발전은 각각의 단말기 안에서 개별적으로 수행하던 게임들을 여러 단말기가 네트워크를 통하여 다수의 사용자들이 함께 게임을 공유하여 즐기는 네트워크를 지원하는 게임의 출현을 야기시켰다[1, 3, 4].

특히 다수의 사용자가 인터넷을 통하여 게임서버에 온라인으로 접속하여 게임을 즐기는 온라인 게임들은 2002년 이후 플랫폼 1위로 등극하여 2003년에는 전체 게임 수출의 80% 가량을 차지하는 등 크게 각광받고 있다[2].

※ 본 논문은 산학 협력결과물임. 본 논문에 제안된 방식의 모든 소유권은 (주)다이스넷엔터테인먼트에 있음.
† 준 회 원 : 전북대학교 컴퓨터공학과 박사과정
†† 정 회 원 : (주)다이스넷 엔터테인먼트 대표이사
††† 정 회 원 : 전북대학교 전자정보공학부 컴퓨터공학전공 교수 및 영상공학과(대학원) 주임교수
논문접수 : 2005년 1월 10일, 심사완료 : 2005년 6월 14일

온라인 게임의 형태는 바둑, 장기, 고스톱 등 고전의 보드 게임들과 같은 턴 방식의 게임과 스타크래프크와 같이 게임 참여자는 제한되지만 여러 명의 사용자들끼리 팀을 구성하여 팀별 대전이 가능한 베틀넷을 통한 전략 시뮬레이션 게임 및 수천, 혹은 수만 명 이상이 동시에 연결되어 게임을 수행하는 온라인 RPG(Role Playing Game) 등이 있다.

이러한 온라인 게임들은 하나의 게임을 수행하는 인원이 2명에서 수십만 이상까지 다양하지만 2명과 같은 적은 수의 인원도 실제적으로는 게임 내 사이버 룸을 통하여 하므로 접속되는 클라이언트들의 수는 서버의 성능에 민감하나 대개 1,000명에서 2,000명 정도를 한 서버가 관리 운영한다.

따라서 온라인 게임 서버는 게임의 장르를 불문하고 동시에 접속 가능한 인원이 1,000명 이상이며 사용자 수가 증가하면 게임 서버의 수를 증가시켜 게임의 진행을 원활히 할 수 있게 한다.

게임 서버를 다중으로 구성할 경우 각 서버 간을 연동 시켜야 하며 그 연결 방식 및 역할방식에 따라 클러스터형, 분산형과 대칭형, 비대칭형으로 나눌 수 있다.

게임서버가 초기 한명의 게임 사용자로부터 연결된 후 계속해서 추가의 사용자를 받아들이기 위해서 기존 접속된 사용자의 연결은 유지시키면서 추가로 연결을 원하는 클라이언트에 대해서 동시에 연결 처리를 수행해야 한다. 반대로 기존 연결된 여러 사용자 중의 한명이 접속을 중단하려고 할 때에도 기존 연결된 사용자의 수행과는 별도로 동시처리 되어야 한다[1, 3~5].

서버가 동시에 두 가지 이상의 일을 처리하는 방법으로는 프로세스 방법과 쓰레드 방법 중 하나를 사용한다. 프로세스 방식은 대칭형 다중 프로세서 구조인 SMP(Symmetric Multi Processor)기능을 가진 컴퓨터에서 뛰어난 성능을 발휘하나 클라이언트를 처리 위한 프로세스 생성속도가 쓰레드 보다 느리다는 단점이 있다. 이에 비해서 쓰레드 방식은 SMP 구조를 제대로 이용하지 못하지만 프로세스보다 생성속도가 빠르며 구현이 간단하다는 장점을 가진다.

본 논문에서는 프로세스 방법과 쓰레드 방식을 비교하여 현재 대부분의 게임서버에 적용되어 있는 비대칭 분산형 구조에 적합한 방법을 제안하였다.

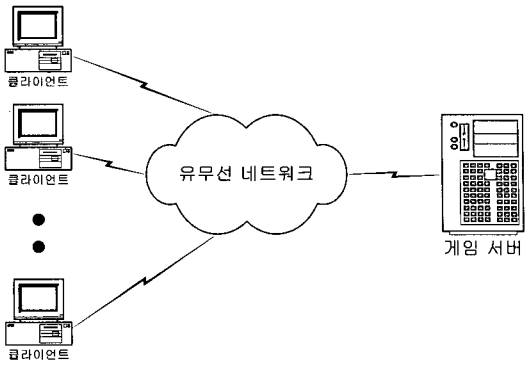
2. 게임 서버의 구성 형태

2.1 단일 게임서버 구조

온라인 게임을 위한 단일 서버구조는 (그림 1)과 같다. 이러한 구조는 하나의 게임서버가 게임처리, 연결, 데이터베이스 관리 등 모든 것을 한꺼번에 처리하는 구조로서 컴퓨터의 성능에 따라 접속자 수가 제한된다. 이러한 구조는 구현이 쉽다는 장점이 있으나 사용자 수의 증가에 따른 적절한 대응이 어려워 초기 시험 단계를 제외하고는 거의 사용되지 않는 구조이다[1].

2.2 다중 게임서버 구조

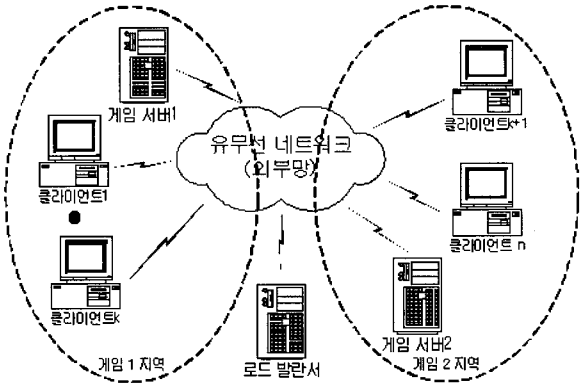
다중게임 서버구조는 서버간의 네트워크 연결 방식에 따라 다소간의 차이는 있지만 크게 대칭형과 비대칭형 및 분산형으로 나눌 수 있다.



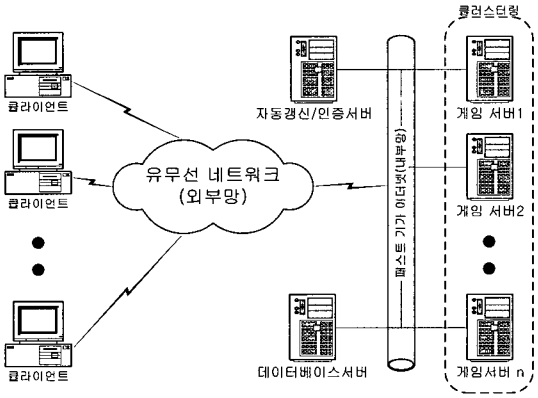
(그림 1) 단일 게임서버 구조

대칭형 구조는 (그림 2)에 나타난 바와 같이 똑 같은 구조의 게임서버 집단이 단순히 클라이언트에 대한 연결 부하를 조정하는 로드발란서(Load-Balancer)를 중심으로 연결되어 있는 형태로 모든 게임서버 구성이 동일하므로 단순한 확장성이 용이하나 기존게임과 새로이 개발된 게임의 연동이 거의 불가능하다. 또한 하나의 서버가 변경되면 모든 서버들이 같이 변경되어야한다[1].

(그림 3)에 표현된 비대칭형의 구조는 공통으로 사용하는 인증서버와 데이터베이스서버는 따로 구성하고 사용자 증가에 따른 부하를 여러 대의 게임서버를 클러스터링(Cluste-



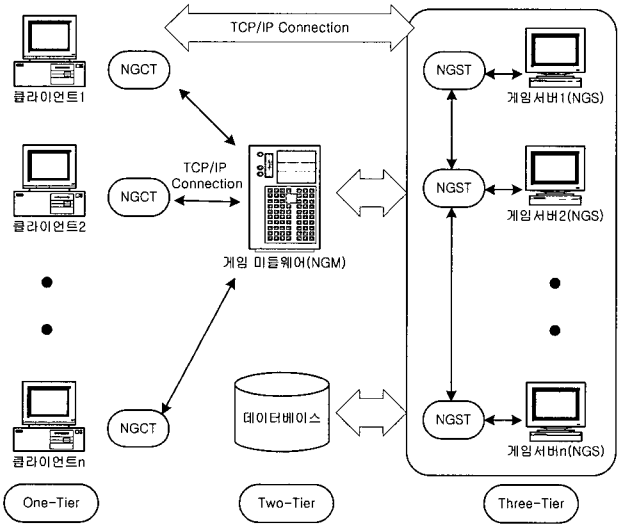
(그림 2) 대칭형 다중 게임서버 구조



(그림 3) 비 대칭형 다중 게임서버 구조

ring)을 통하여 연동될 수 있게 한 구조이다. 이는 상용화된 많은 게임 업체가 대부분 수용하고 있는 형태로 게임의 확장이 쉽고 게임서버만의 추가로 운영이 용이하나 새로 개발되는 게임하고의 연동은 매우 어려운 형태이다[1].

분산형 게임 서버는 (그림 4)에 나타내었다. 이와 같은 구조는 온라인 게임 서버가 담당하는 모든 기능들을 분리하여 별도의 서버들로 구성하여 각각의 기능을 독립적으로 수행할 수 있게 내부 네트워크를 통하여 연결된 형태로서 초기 게임 서버의 수가 많고 내부 네트워크 구성을 위한 별도의 구성이 필요해진다는 단점이 있으나 기본 구성이 완료될 경우, 가장 저렴한 가격의 서버를 이용하여 가장 많은 수의 접속자 처리를 하는 장점이 있으며, 게임의 연속성에 대해서도 유연하게 대응할 수 있다[1].



(그림 4) 분산형 다중 게임서버 구조

NGM : Network Game Middleware NGS : Network Game Server
NGST : Network Game Server Terminal NGCT : Network Game Client Terminal

3. 클라이언트의 접속처리 방법

온라인 RPG에서 한 클라이언트가 게임을 수행하기 위해 게임 서버로의 새로운 접속을 허용할 때 서버는 기존의 연결을 유지하면서 클라이언트의 접속 요청을 받아드려야 한다. 이 때 일반적으로 서버는 다중 작업 처리를 위한 방법으로 프로세스 방법과 쓰레드 방법 중 하나를 사용한다.

프로세스 방식은 대칭형 다중 프로세서 구조인 SMP 기능을 가진 컴퓨터에서 뛰어난 성능을 발휘하나 클라이언트를 처리 위한 프로세스 생성속도가 쓰레드 보다 느리다는 단점이 있다.

이에 비해서 쓰레드 방식은 SMP 구조를 제대로 이용하지 못하지만 프로세스보다 생성속도가 빠르며 구현이 간단하다는 장점을 가진다. 이를 자세히 기술하기 위하여 다음과 같은 표기를 가정한다.

P : Process **T : Thread**

본 논문에서 P와 T앞에 나오는 숫자는 해당 방식의 생성 횟수를 나타낸다.

i) 0T 방식

이 방식은 클라이언트 연결에 쓰레드를 사용하지 않으며 단지 서버의 관리 목적으로 한 두 개의 쓰레드를 제한적으로 사용하는 방식을 말한다.

ii) 1P0T 방식

이 방식은 main 함수에서 loop가 돌면서 select로 소켓을 감지하여 accept와 recv/send 명령어를 처리하는 방식으로 하나의 프로세스만 생성하는 방식으로 구현하기 간단하고 한 번에 한 클라이언트만 처리하기 때문에 자원공유가 매우 쉽다. 또한 새로운 프로세스나 쓰레드를 만들 필요가 없다는 장점이 있으나 SMP 구조의 컴퓨터를 효율적으로 이용할 수 없고 오류가 나면 즉시 서버가 다운되는 단점이 있다.

iii) nP0T 방식

이 방식은 main 함수에서 accept 요청만 처리하고, 새로운 접속 요청이 들어오면 fork를 호출하여 새로운 프로세스(Child P-)를 만드는 방식으로 Apache를 비롯하여 대부분의 웹 서버가 이 방식을 취하고 있다. 이 방식은 SMP를 효율적으로 이용할 수 있으며 자식 프로세스(Child P-)가 소멸되더라도 부모 프로세스(Parent P-)는 소멸되지 않으므로 서비스는 계속할 수 있고 하나의 클라이언트가 느려져도 전체 속도에 큰 영향을 끼치지 않는 장점을 가진다.

그러나 이 방식은 프로세스간의 자원 공유가 공유메모리 등을 이용해야 하기 때문에 자원에 대한 공유가 어렵고 또한 구현 자체도 어려우며 프로세스를 만드는 비용이 상당히 크다. 이를 해결하기 위해서 Pool을 구현하여 사용한다.

Pool이란 미리 일정량의 프로세스나 쓰레드를 생성해놓고 새로운 프로세스나 쓰레드를 만들 요청이 오면 이미 만들어진 프로세스나 쓰레드를 할당하는 방식으로 사용할 수 있는 미리 만들어놓은 프로세스나 쓰레드가 없다면 새로 만든다.

iv) 1PnT 방식

이 방식은 main 함수에서 accept요청을 받아들이면 새로운 스레드를 생성하여 클라이언트를 관리하게 하는 방식으로 공유가 쉬운 편이며 공유메모리를 쓸 필요가 없고 하나의 클라이언트가 느려져도 전체 속도에 큰 영향을 끼치지 않는다. 또한 새로운 쓰레드 생성 비용이 프로세스 보다 작다. 하지만 이 방식 역시 SMP를 효율적으로 이용할 수 없으며 한 쓰레드에서 문제가 발생하면 모든 서비스가 중단된다.

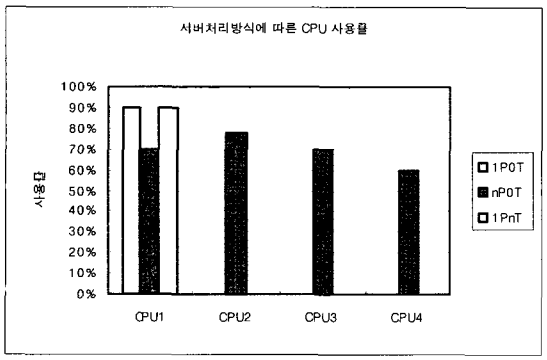
위 네 가지 방법 중 서버가 클라이언트와의 접속을 위해 사용하는 방식은 일반적으로 ii), iii), iv)의 방법이 많이 채택된다.

<표 1>과 (그림 5)에 나타난 도표는 각 접속 방식에 따른 최대 접속 클라이언트 수와 이에 따른 CPU 사용률을 나타낸 것이며 (그림 6)에서는 각 서버 처리 방식에 따른 클라이언트의 처리속도를 비교해 보았다. 각 서버 접속 방식에 따른 최대 접속 클라이언트 수와 CPU의 사용률을 알아

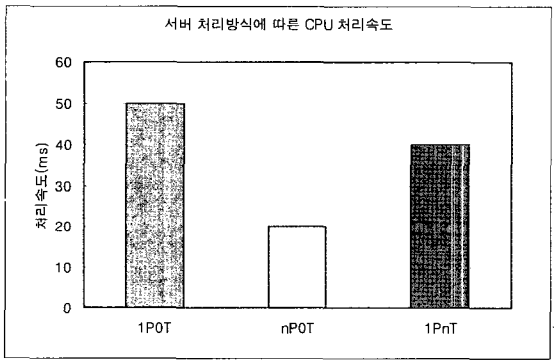
보기 위하여 CPU 펜티엄III 1GHz, RAM 512MB의 하드웨어 사양에 운영체제는 리눅스를 탑재하여 시스템을 구축하였으며, 가상의 시뮬레이션 프로그램을 사용하여 클라이언트를 생성한 후 자료를 구하였다.

〈표 1〉 처리 방식에 따른 접속 클라이언트 수와 CPU 사용률

처리 방식	최대 접속 클라이언트 수	CPU 사용률
1P0T	1,000개 이상	1 : 90% 이상
nP0T	423개	4 : 80% 이하
1PnT	255개 이상	1 : 90% 이상



(그림 5) 서버 처리방식에 따른 CPU 사용률



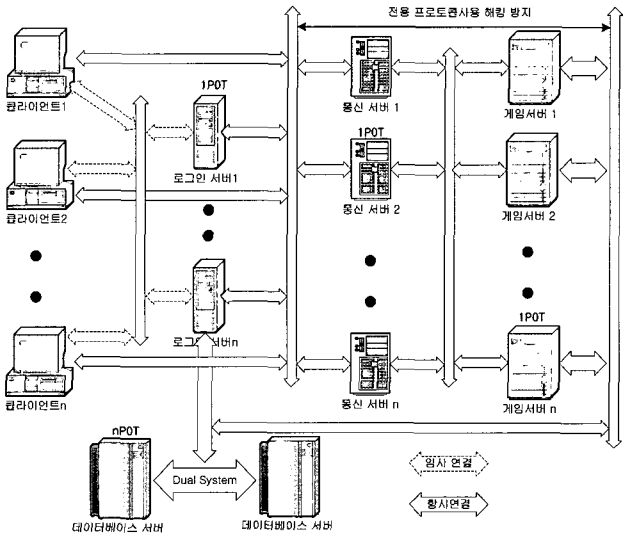
(그림 6) 서버 처리방식에 따른 CPU 처리속도

실험을 통하여 <표 1>과 (그림 5)에서 비교된 바와 같이 3가지 서버의 접속처리 방식 중에서 1P0T 방식은 최대 클라이언트 접속 수에서 가장 우수하였으며, CPU 사용률에서는 수치상으로 1PnT 방식과 유사한 결과를 나타내지만 접속한 클라이언트의 수가 많은 점을 감안하면 1P0T 방식이 더 우수한 방식이다. 1P0T 방식은 nP0T 방식과 1PnT 방식에 비하여 상대적으로 많은 수의 클라이언트가 접속 가능하였으며 nP0T 방식과 달리 SMP를 이용하지 않기 때문에 여러 개의 CPU를 사용하는 고가의 추가적인 장비 사용이 불필요하므로 서버 구축비용이 저렴하게 된다. (그림 6)에서 1P0T 서버 처리방식의 CPU 처리속도가 가장 늦게 나오는 이유는 <표 1>에서 보는 바와 같이 최대 접속 클라이언트의 수가 동일하지 않아서 발생하는 것이며, 클라이언트의 수를 동일하게 확대하여 처리속도를 비교하면 1P0T 방식이 다른 방식에 비하여 우수하다. 실험 결과에서 보는 바와 같이 서버의 성능은 CPU의 개수보다는 처리속도에 더욱 민감하다는 것을 알 수 있다.

따라서 게임을 위한 서버 시스템의 구축 중에서 가장 중요한 사용자의 게임 데이터를 관리하는 데이터베이스 서버 시스템을 제외하고는 1개의 CPU를 갖는 300만원이하의 PC 서버급으로 게임 서버 시스템을 구현할 수 있다.

4. 제안한 비대칭 분산형 다중 서버 구조

앞 절에서 살펴본 바와 같이 결과적으로 클라이언트의 연결을 처리하기 위한 서버의 성능은 CPU의 개수가 아니라 초당 CPU가 처리할 수 있는 클럭수에 더욱 민감하다. 따라서 이 같은 결과 자료를 토대로 구현할 수 있는 가장 적합한 서버구조는 (그림 7)에 제안한 바와 같이 각 서버가 하는 일을 가능한 한 세부적으로 분산시킨 형태이다.



(그림 7) 제안된 비대칭 분산형 다중 서버구조

본 논문에서 제안한 비대칭 분산형 다중 서버 구조는 각 기능을 물리적으로 독립시켜 전체적인 서버의 안정성을 증대시킬 수 있으며 통신 서버, 로그인 서버 그리고 게임 서버에는 1P0T 방식을 데이터베이스 서버에는 nP0T 방식을 적용하였다. 로그인 서버, 통신 서버 그리고 게임서버에 1P0T방식을 채택한 것은 만약 각 서버에 과부하 등의 문제가 발생할 경우 게임의 전체적인 흐름으로 보아 새로운 사용자의 로그인이 지연되는 등의 비교적 사소한 경우의 문제가 발생한다고 볼 수 있다. 하지만 데이터베이스 서버의 경우 만약 문제가 발생한다면 사용자의 게임 캐릭터, 게임 단계 등 게임 사용자에게는 중대하고 민감한 문제가 발생할 수 있으므로 다중 프로세스 방식을 사용하여 하나의 프로세스에 이상이 발생하는 경우 다른 프로세스가 서버의 기능을 수행할 수 있으므로 서버의 안정성을 확보할 수 있다. 또한 다중 통신서버를 사용하여 로드 밸런싱 및 게임 서버의 부

하를 최소화하였다. 게임처리에 실질적 관계가 없는 채팅 부분과 같은 메시지는 통신서버에서 처리하도록 하였으며 통합된 단일의 데이터베이스 관리로 게임 배경의 추가 삭제가 용이하다는 장점을 가진다. 즉, 확장성 있는 유연한 게임 서버 구조를 확보 할 수 있다. 또한 게임 서버의 경우 사용자의 증가에 따른 점진적인 투자를 할 수 있는 구조이므로 게임 개발 시 초기 투자비용이 절감되는 효과가 있다. 본 논문에서는 제안한 비대칭 분산형 다중 서버 구조는 대칭형 서버 구조일 경우에는 불가능한 하나의 캐릭터로 여러 배경을 돌아다닐 수 있으며 게임 배경의 물리적 공간과 논리적 공간이 1:1로 매치된 형태를 가지게 되며, 하나의 게임 서버가 하나의 게임 배경만을 처리하게 된다.

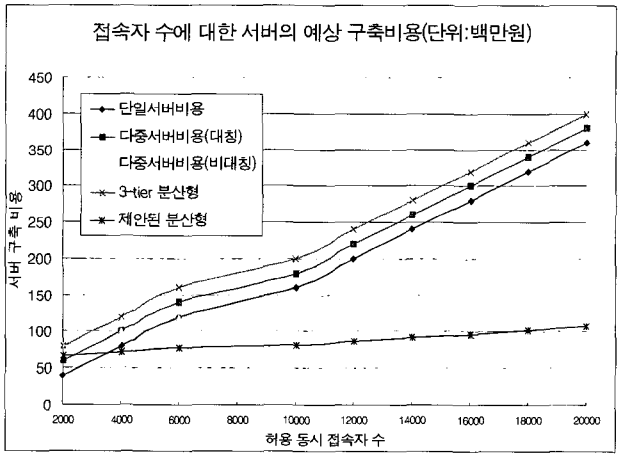
<표 2>에서는 여러 형태의 서버 구조에 따른 특징과 구축비용에 대하여 비교하였다. 서버의 구축비용은 서버에 동시 접속할 수 있는 사용자 수를 기준으로 하였으며, 본 논문에서는 서버의 속도와 용량보다는 동시에 접속할 수 있는 사용자 수를 기준으로 하였으며, 각 구조에 따른 서버 구축비용은 2,000명의 사용자가 동시에 접속할 수 있는 서버를 기준으로 비용을 산출하였다. 단일 서버의 경우 한 대의 서버 내에서 로그인, 통신, 데이터베이스, 게임 서버 등 게임에 관한 모든 부분이 한 대에서 운영이 되므로 초기 구축은 쉬운 반면 보안성이 약하며 서버의 확장이 어려운 단점이 있다. 대칭형 다중 서버의 경우에는 로드 발란서를 통하여 게임 사용자를 분산시킬 수 있다는 장점이 있으나 로드 발란서를 추가하여야 하므로 추가적인 구축비용이 발생하며 보안성과 확장성이 약하다. 비대칭형 다중 서버의 경우에는 인증 서버와 데이터베이스 서버 등이 추가되어 독립적으로 존재하므로 보안성이 높아지고 확장성 또한 개선된다. 그러나 초기 설치가 까다로우며, 새로운 게임을 추가할 경우 게임 서버의 확장에 어려움이 있다. 기존 3-Tier 분산형은 시스템의 초기 구축이 어렵고 보안성이 약한 단점이 있으나 확장성이 우수하며 게임 에피소드간 이동이 일부 가능하다는 장점을 가지고 있다. <표 2>에서 보는 바와 같이 제안한 4-Tier 비대칭 분산형 다중 서버 구조는 서버의 추가 비용이 적게 든다는 장점 즉, 여러 대의 저가형 PC를 이용하여 서버로 활용할 수 있으므로 시설 투자비용에 대한 부담을 줄일 수 있다. (그림 8)에는 각 서버 구조별로 접속자 수에 따르는 예상 서버 구축비용을 도표화 하였다.

제안한 4-Tier 비대칭 분산형 구조는 앞에서 설명한 바와 같이 초기 시스템 구축에 어려움이 있는 단점을 가지고 있으나, 초기 설치비용과 서버 추가 비용이 저렴하며 확장성과 보안성이 뛰어나고 실시간 이벤트 그리고 게임 에피소드간 이동이 가능하여 다른 서버구조에 비해 경제적, 성능적인 면에서 우수하다.

본 논문에서 제안한 비대칭 분산형 다중 서버 구조는 서버운영에 있어서의 사용자 수에 따른 점진적인 서버의 확장이 가능하며, 필요한 부분의 서버만 추가하면 되므로 단일 서버 구조 등 다른 서버 구조에 비하여 경제성을 가지게 되며, 게임의 각 기능이 로그인 서버, 통신 서버, 데이터베이스 서버 그리고 게임 서버 등으로 나누어져 있으므로 각 서버의 안정성을 확보할 수 있는 장점을 가진다.

<표 2> 서버 구조별 특성 비교

비교항목 \ 서버구조	단일서버	다중서버 (대칭)	다중서버 (비대칭)	기존 분산형 서버 (3-Tier)	제안한 분산형 서버 (4-Tier)
초기구축 비용	고성능 서버 1대 약 4,000만원	단일서버+ 로드발란서 약 6,000만원	대칭서버+DB+인증서버 약 8,000만원	고성능 서버 2대 약 8,000만원	통신+로그인+게임서버+DB 약 6,700만원
초기구축 용이성	쉬움	비교적 쉬움	비교적 어려움	노하우 필요	노하우 필요
보안성	취약함	취약함	약간 높음	취약함	높음
확장성	취약함	취약함	약간 높음	높음	매우 높음
실시간 이벤트	불가능	매우 어려움	쉬움	어려움	쉬움
게임 에피소드 간 연동	불가능	불가능	일부 가능	불가능	가능
게임 서버 간 캐릭터 이동	재접속 필요	재접속 필요	재접속 필요	일부 가능	가능
기본 서버 추가 비용	4,000만원	4,000만원	2,000만원	4,000만원	기본 500만원
서버 당 예상 동시 접속자 수	2,000명	2,000명	2,000명	2,000명	2,000명



(그림 8) 접속자 수에 대한 서버의 예상 구축비용

5. 결 론

본 논문에서는 클라이언트 연결을 처리하는 게임 서버의 처리 형태 중 1P0T, nP0T 그리고 1PnT 방식의 성능을 비교, 분석하였고 이를 기반으로 1P0T 방식이 게임 서버에서 클라이언트의 연결 처리에 가장 적합한 방식임을 실험 결과를 통하여 알 수 있었다.

도출된 접속처리 방식은 서버의 기능을 세부적으로 분리하여 각각의 기능을 독립적으로 수행하는 비대칭 분산형 다중 서버 구조에 적합하였으며 이를 위하여 이 방식에 가장 알맞은 서버 구조를 제안하였다.

제안한 비대칭 분산형 다중 서버 구조는 서버의 기능을

로그인 서버, 게임 서버, 통신 서버, 데이터베이스 서버 형태로 세부적으로 분리하여 각각의 기능을 독립적으로 수행한다. 따라서 다른 구조들과 비교하였을 경우 서버의 점진적인 확장이 가능하여 경제적이며, 각 서버의 기능을 분리하여 각각의 기능을 독립적으로 수행하므로 기능적으로 더 나은 성능을 보인다. 게임 처리와 실질적으로 관계가 적은 채팅 부분과 같은 메시지는 통신 서버에서 처리하며, 통합된 단일의 데이터베이스 관리로 게임의 배경에 대한 추가, 삭제가 용이하다. 또한 대칭형 구조일 경우에는 불가능한 여러 배경을 돌아다닐 수 있고, 하나의 게임 서버가 하나의 배경만을 처리하므로 확장성이 용이한 게임 서버 구조이다. 추가적으로 실시간 이벤트 등을 실시할 수 있으며, 사용자의 직접적인 서버 접근을 막을 수 있으므로 해킹을 근본적으로 해결할 수 있다.

제안한 비대칭 분산형 다중 서버 구조는 위에서 살펴본 바와 같이 게임 서버 구조를 위한 경제성과 기능성을 갖추었으며, 특히 서버의 안정성을 향상시킬 수 있으며, 필요한 시기에 필요한 부분의 서버만을 추가하여 확장할 수 있으므로 서버 시스템의 확장성과 경제성이 크게 향상되었다.

참 고 문 헌

[1] 이남재 "온라인롤플레잉 게임을 위한 FULL 3D 맵 관리방법에 관한 연구" *박사학위논문 전북대학교 2003. 8.*

[2] 2004 대한민국 게임백서, 문화관광부 한국게임산업개발원, pp.65-81, 2004. 6.

[3] 2003 대한민국 게임백서, (재)한국게임산업개발원, pp.66-75, 2003. 6.

[4] 2002 대한민국 게임백서, (재)한국게임산업개발원, pp.28-39, 2002. 4.

[5] 이남재, 곽훈성, 온라인 RPG의 시리즈 시나리오(캠페인)를 위한 분산형 게임 서버 적용방법, 한국게임학회 2002년 동계학술대회 논문집, pp.353-357, 2002. 1.

황 도 연

e-mail : ehdus@chonbuk.ac.kr

1997년 전주대학교 전자계산학과(이학사)

1999년 전주대학교 컴퓨터공학과(공학석사)

2003년 전북대학교 컴퓨터공학과 박사과정 수료

관심분야 : 영상처리, 컴퓨터 게임

이 남 재

e-mail : njlee@dicenet.co.kr

1988년 전북대학교 전자계산기공학과 (공학사)

1991년 전북대학교 전산통계학과(이학석사)

2003년 전북대학교 컴퓨터공학과(공학박사)

1999년~현재 (주)다이스넷엔터테인먼트 대표이사

2001년~현재 한국게임학회 이사, 온라인 게임 분과 위원장

1999년~현재 전주국제게임엑스포 조직위원

2004년~현재 한국게임고등학교 감사

관심분야 : 인공지능, 컴퓨터게임

곽 훈 성

e-mail : hskwak@chonbuk.ac.kr

1970년 전북대학교 전기공학과(공학사)

1979년 전북대학교 전자공학과(공학박사)

1981년~1982년 미국 텍사스 주립대학 연구교수

1994년~1998년 국가교육연구 전산망 추진위원

1998년 과학기술법령정비정책위원

1999년~현재 조달청 우수제품(정보통신)심사위원

1997년~현재 (사)영상산업연구센터 대표

2005년 현재 전북대학교 전자정보공학부 컴퓨터공학전공 교수 및 영상공학과(대학원) 주임교수

관심분야 : 영상신호처리, 인공지능, 컴퓨터비젼, 컴퓨터게임