

블록을 이용한 대규모 사용자 지원 분산 서버 시스템 설계

김순곤*

요약

본 논문에서는 MMORPG와 같은 게임에서 대규모의 사용자 처리를 유연하게 할 수 있는 블록 기법을 이용한 분산형 게임서버구조를 제안하였다. 제안된 분산형 구조에서의 게임서버들은 각 맵(map)을 담당하는 존(zone)이라는 형태로 그룹화되며 각 존 내의 서버들은 각 존에 할당된 게임 맵을 여러 개 일정크기의 블록으로 나누어 각 블록들을 게임 서버의 성능에 따라 적절하게 분배하여 처리한다. 제안된 블록 기법을 도입한 분산형 게임서버구조는 게임서버와 통신서버에서의 블록처리 방식을 통해 게임배경의 추가/삭제를 용이하게 할 수 있어 대규모의 사용자를 수용할 수 있음은 물론 추후 사용자가 직접 만드는 배경도 기존 게임 배경에 연동될 수 있도록 하였다. 일정한 블록으로 나눈 게임배경을 서버의 성능이나 처리하는 캐릭터의 수에 따른 부하를 동적으로 배분하여 시스템의 효율성을 증가시키고 안정성을 높였다.

Design of A Distributed Server System for Supporting Enormous Users Using Blocks

Soon-Gohn Kim*

Abstract

In this paper, I propose a distributed game server system which has flexibility of enormous game character's processing in MMORPG using block scheme. I proposed game servers in distributed system grouped with zone which charge of each map, and the game servers in this zone divide game map into several unit blocks and process distributing blocks to game servers according to its performance. I proposed distributed game server system using block scheme can add/drop game map to/from easily through block process method in game and communication servers. Therefore, my proposed system can enormous character processing and add user created game map to existed game map easily in the future as well. Also, my scheme increases the system utilization and upgrades system stability by dynamic distribution of game map according to the number of characters and performance of servers.

Keywords : Distributed Server System, Online Game server, MMORPG, Block Map Scheme

1. 서론

현재 네트워크를 이용한 국내외 온라인 게임 시장은 크게 3가지 형태의 장르로 구분될 수 있다. 즉, 첫째 기존 테트리스나 바둑 등 고전게임을 온라인화 한 게임, 두 번째 스타크래프트나 FIFA와 같이 기본적으로 PC에서 동작하지만 추가로 네트

워크기능을 지원하는 리테일 하이브리드 게임 및 마지막으로 리니지나 월드오브워크래프트와 같은 MMORPG 계열의 PW(Persistent World)게임 등이 있다[1][4][5].

이중 PW 게임은 가장 많은 인기를 끌고 있는 게임들이며 인류세계와 비슷한 가상세계를 게임 배경으로 삼기 때문에 그 배경의 규모가 계속 커지면서 복잡해지는 양상이 되고 있다.

PW 게임들은 일반적으로 클라이언트 서버구조의 형태를 가지고 있으며 캐릭터나 오브젝트들의 현 좌표나 지형좌표 등을 처리하는 게임 배경 처리는 대부분 게임서버에서 수행하게 된다.

온라인 게임서버구조는 크게 단일 서버구조와

※ 제일저자(First Author) : 김순곤
접수일:2011년 10월 22일, 수정일:2011년 11월 26일,
완료일:2011년 12월 26일
* 중부대학교 컴퓨터학과: sgkim@joongbu.ac.kr
■ 이 논문은 2010년도 중부대학교 학술연구비 지원에 의하여 이루어진 것임.

다중서버구조로 나누어지며, 다중서버구조는 클러스터링 기법을 이용한 대칭구조와 분산기법을 이용한 비대칭구조로 다시 세분화 할 수 있다 [2].

클러스터링 기법을 이용한 대칭 구조는 사용자의 증가에 쉽게 대응할 수 있다는 장점이 있으나 기본적으로 구축해야할 클러스터링 내 서버군의 구축비용이 크다. 특히, 게임의 시나리오 추가에 따른 게임 배경의 추가나 마케팅 전략 변경에 의한 배경의 변경/삭제 등이 거의 불가능하며 기존 게임 배경과 추가된 게임 배경간의 상호 연동도 매우 어렵다[6][7].

그에 비해서 비대칭 방식의 분산 형 서버구조는 게임 진행에 필요한 로그인, 통신, 게임 물 처리, 데이터베이스 접근 등 모든 기능을 각각 독립적으로 수행할 수 있게 분리하여 구성되어 있다. 따라서 게임 배경의 추가/변경/삭제에 유연한 장점이 있다[3]. 즉, 분산 구조의 게임 서버 시스템에서는 사용자수의 증가에 통신서버만 증가시키고 게임배경 추가에는 각 배경을 담당하는 게임 서버만 추가하면 된다[4].

그러나 현재 컴퓨터 발전 속도를 볼 때 게임 배경을 처리하는 같은 가격대의 서버의 성능이 향상됨에도 불구하고 기존의 방식으로 서버의 성능에 따라 담당하는 배경 크기에 따른 가변적인 처리가 어렵다는 단점이 있다[8][9][10].

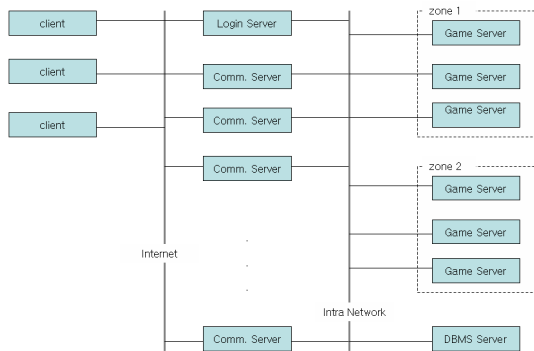
본 논문에서는 이를 해결하기 위해서 기존 분산시스템을 기반으로 게임의 각 배경을 일정 크기의 블록으로 나누어 처리함으로써 게임 배경처리에 대한 보다 큰 유연성을 부여 하였다. 이를 통하여 각 서버가 가지는 성능에 맞게 적절하게 부하를 조절 할 수 있으며 이에 따라 대규모 사용자의 증가에 점진적으로 유연히 대응할 수 있다.

본 논문은 2장에서는 제안한 게임서버 시스템의 기본구조에 대하여 기술하고, 3장에서는 제안한 게임서버의 배경 블록처리 방법을 설명한다. 그리고 4장에서는 제안한 통신서버에서의 패킷 처리 방식을 기술하며, 5장에서 결론을 맺는다.

2. 제안한 게임서버 시스템의 기본 구조

본 논문에서 제안한 게임서버 시스템의 기본 구조는 (그림 1)과 같다. (그림 1)에서 보는 바와 같이 제안한 시스템은 사용자가 사용하는 클라이언트와 사용자인증을 담당하는 로그인서버, 각 모듈간의 통신을 담당하는 통신서버, 게임 물 처리를 담당하는 게임 서버 및 사용자의 정보를 관리하는 데이터베이스 서버로 구성된다. 이중 게임 배경을 처리하는 게임서버들은 다시 처리하는 배경에 따라 존(zone) 이라는 그룹으로 구성된다. 이중 데이터베이스서버는 사용자의 정보를 바탕으로 게임을 진행하기 위한 인증 및 사용자 정보 갱신 등을 담당하며 기존 게임 구조와 동작이 같다.

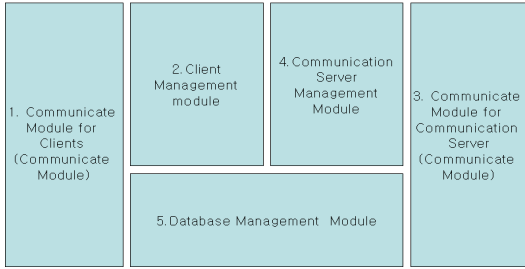
따라서 본 논문에서는 로그인 서버, 통신서버 게임 서버 구조에 대해서만 기술한다.



(그림 1) 제안한 게임서버 시스템의 기본구조

2.1. 로그인 서버

로그인 서버는 참고문헌 [2]에 기술된 분산 형 게임서버구조안의 로그인 서버와 거의 같은 기능을 수행하며 추가로 통신서버의 부하를 조절하는 로드밸런서 기능을 위한 통신서버 부하 리스트를 관리한다. 즉, 로그인 서버는 사용자 클라이언트와 접속하여 인증을 실시하며 그 후 클라이언트를 통신서버에 접속시킨 후 클라이언트와의 접속을 끊고 다음 사용자 인증을 위해 대기한다. 이를 위해 로그인 서버는 (그림 2)에서 보는 바와 같이 크게 5가지 기능을 수행한다.

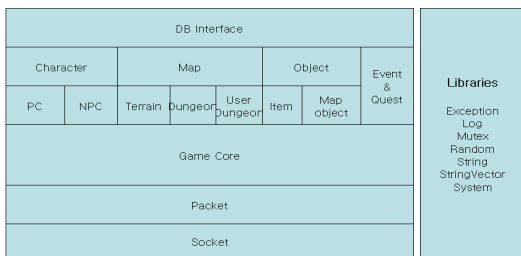


(그림 2) 로그인 서버의 기능 블록도

2.2 게임서버

게임서버는 캐릭터, 지형, 오브젝트 및 이벤트나 퀘스트 등을 처리하는 서버로서 기능 블록은 (그림 3)과 같다. 본 논문에서 제안한 게임서버 방식은 기본적으로 참고 문헌[2]에 기술된 분산형 게임 서버의 역할과 비슷하나 각 배경을 일정크기의 블록으로 나누어서 각 블록을 담당하는 서버를 동적으로 지정하여 운영하게 함으로써 게임서버 하드웨어 발전에 의한 배경의 크기를 유연하게 처리할 수 있게 했 하였다. 또한 통신서버와의 연동을 통하여 게임 배경의 추가/삭제/변경을 쉽게 할 수 있게 하여 대규모의 사용자를 수용할 수 있게 하였다.

게임 배경을 온라인으로 추가/삭제/변경할 수 있는 기능은 추후 게임 배경을 개발자뿐만 아니라 사용자도 직접 만들어 기존게임에 반영시킬 수 있는 UCC 개념의 게임 기획이 가능하도록 하였다. 이 같은 사용자 중심의 배경을 본 논문에서는 사용자 던전이라고 정의하였다. 즉 사용자 던전이란 게임 개발자가 기획하여 제공하는 게임의 배경이 아니고 그 게임을 수행하는 사용자가 3D 제작프로그램이나 개발사가 제공한 맵 에디터를 통해 직접 제작한 배경을 게임에 연동시켜 일부지만 게임의 운영에 직접 관여할 수 있는 게임공간을 의미한다.



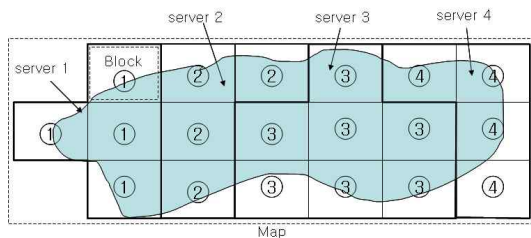
(그림 3) 게임 서버의 기능 블록도

2.2.1 게임서버에서의 배경처리

게임서버에서의 배경처리는 크게 기본 지형과 특수지형(던전) 및 사용자가 직접 제작하는 지형(사용자던전)처리 3가지로 구분된다. 본 논문에서 제안한 배경처리 방법은 다음과 같다. 맵은 3차원으로 그려진 지형을 말하며 기본지형, 특수지형 사용자 제작 지형을 포함한다. (그림 4)에 배경처리 예에 대한 개념을 제시하였다.

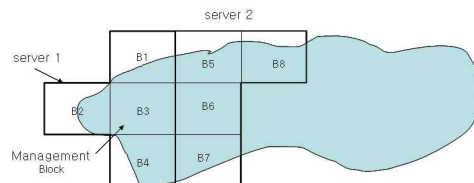
먼저 생성된 맵을 일정크기의 정방형 블록으로 나눈 다음 (그림 4)와 같이 각 서버가 안정적으로 수행할 수 있는 배경처리 능력에 따라 블록의 수를 고정적으로 미리 할당한다. 각 서버는 할당된 블록에 대한 배경처리를 독립적으로 담당해야 하며 일반적으로 블록사이즈는 사용자가 게임 그래픽 내에서 볼 수 있는 가시거리로 범위를 정한다.

게임서버는 블록단위를 기본으로 배경을 관리한다. (그림 5)에서 볼 수 있는 바와 같이 인접블록은 한 게임 서버 안에 존재할 수도 있고 다른 서버 안에 존재할 수도 있다.



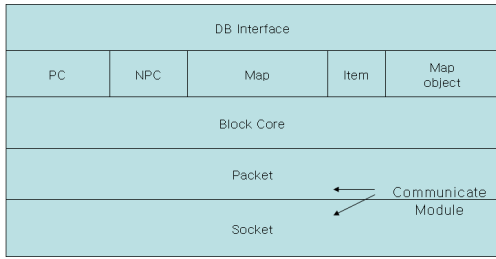
(그림 4) 배경 처리 예

만약 인접블록이 한 서버 안에 존재할 경우 데이터들의 전송은 함수 호출 방식에 의해 이루어지고 그렇지 않은 경우 통신서버를 통하여 서버 간 자료를 전송하게 된다. (그림 5)의 예에서 보듯 현재 캐릭터가 속해있는 지형의 블록이 B3 일 경우 B1, B2, B4는 같은 서버에 B5, B6, B7은 다른 서버에서 담당한다.



(그림 5) 인접블록의 예

(그림 6)은 이러한 블록 관리를 위한 모듈의 기능 블록도를 보여준다.



(그림 6) 블록관리모듈의 기능 블록도

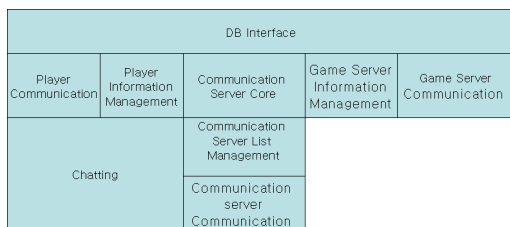
2.3 통신서버

통신 서버는 클라이언트와 게임서버간의 자료 전달을 하는 매개역할을 한다. 클라이언트는 통신서버와 일단 접속하면 맵의 종류나 크기에 관계없이 게임에 접속을 계속 유지하며 맵을 자유로이 이동할 수 있다.

또한 통신 서버는 게임서버에서 받은 자료를 다른 통신서버에 방송함으로써 해당 사용자와 관련된인접한 모든 클라이언트들이 패킷을 받아 볼 수 있도록 한다.

통신 서버는 각 클라이언트들의 맵 및 좌표를 알고 있으므로 채팅 패킷을 게임 서버를 거치지 않고 자체적으로 처리할 수 있다. 따라서 게임 서버는 게임 진행에만 몰두할 수 있게 되어 게임서버의 부하를 줄이고 게임서버의 처리 능력을 향상시킨다.

통신 서버는 게임서버들의 맵과 블록들에 대한 정보를 관리하여 맵과 블록의 등록 및 삭제를 함으로써 맵의 추가 및 삭제가 용이하다. 이는 게임 배경을 순차적으로 추가 시킬 수 있어 대규모 사용자를 수용하는 거대한 게임 배경 제작 서비스가 가능하다. (그림 7)은 이러한 통신 서버의 기능 블록도를 보여준다.



(그림 7) 통신 서버의 기능 블록도

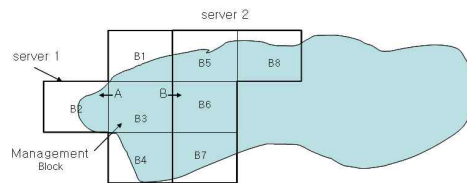
3. 제안한 게임서버의 배경 블록처리

제안한 게임 서버에서의 배경 블록 처리는 본 논문의 핵심이 되는 부분으로 게임 내 배경 안에 존재하는 게임 캐릭터들을 처리하기 위한 서버 자원을 효율적으로 유연하게 사용할 수 있는 방법을 말한다.

분산형 구조의 게임 내에서 움직이는 게임 캐릭터(NPC포함)가 블록 간을 이동할 때, 동일 서버 내 블록 간 이동이나 서로 다른 서버로의 블록 간 이동이냐에 따라 처리가 달라진다.

또한 캐릭터가 블록 간 이동을 한다는 것은 크게 단순한 이동과 공격을 위한 이동 두 가지로 구분할 수 있다. 공격을 위한 이동은 단순한 이동에 대한 기능 수행 후 해당 공격에 대한 처리를 추가 하는 것으로 단순한 이동의 공격에 대한 추가 처리의 개념이다.

블록 간 이동 관리는 게임 서버의 블록 코어 부분에서 이루어지며 (그림 8)에 나타난 바와 같이 동일 서버 내 블록 간 이동 (A의 경우)과 서로 다른 서버로 블록 간 이동 (B의 경우)로 크게 나눌 수 있다.



(그림 8) 동일 서버 내 블록 간 이동

위 (그림 8)에 도시 되었듯이 게임 서버 1(server1)은 블록 B1, B2, B3, B4를 담당하고 게임서버2(server2)는 B5, B6, B7, B8을 담당한다고 할 때, A의 경우는 B3에서 B2로 이동하는 동일 서버 내 블록 간 이동이다.

이 경우 게임 서버1은 해당 블록(B2)에 입장함을 통해 플레이어의 블록 간 이동이 발생했음을 알리고 플레이어 또는 NPC의 정보를 해당 블록에 함수방식으로 전달한다.

이에 비해서 경우 B는 캐릭터가 서버1의 B3 블록에서 서버2의 B6블록으로 서로 다른 서버로

블록 간 이동을 실시한 경우이다. 이 경우 게임 서버1은 플레이어 또는 NPC의 정보를 데이터베이스에 저장하고 통신서버에 플레이어 또는 NPC가 이동했음을 통신 패킷을 이용하여 통보한 후 해당 플레이어 또는 NPC의 정보를 삭제한다. 통신서버에 의해 패킷으로 게임 서버2에 전달된 정보에 따라 이 후 게임 서버2는 해당 처리 블록 안으로 이동해 온 캐릭터에 대한 이동 처리를 담당한다.

블록 간 공격관리는 기본적으로 이동과 비슷하며 블록 간 공격이 동일 서버 내 이루어졌을 경우(A의 경우) 게임 서버1은 해당 블록(B2)에 공격이 발생했음을 알리고 공격 정보를 해당 블록에 함수 호출 방식을 이용해서 전달한다.

B의 경우와 같이 다른 서버로 블록 이동하면서 공격을 하였을 경우는 게임 서버1은 해당 통신서버에 공격이 발생했음을 통신패킷을 이용하여 게임서버2에 통보한다.

4. 통신서버 패킷처리

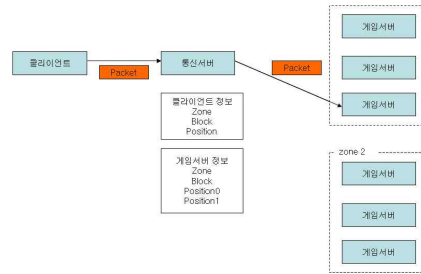
본 논문에서 제안한 분산형 게임 서버 구조에서 통신 서버는 로드발란서 역할 뿐만 아니라 사용자간 채팅, 재접속이 없는 사용자의 게임 배경 간 이동, 게임배경의 추가 삭제 등 가장 핵심적인 역할을 담당하며 서버 간 패킷 전송을 통해 분산된 각 서버와의 정보교환을 통해 역할을 수행한다.

통신서버에서의 패킷처리 방식은 클라이언트와의 패킷처리, 클라이언트 채팅에 관한 패킷처리, 게임서버와의 패킷처리 및 게임 서버가 담당하는 블록처리를 위한 패킷처리 등 크게 4가지로 분류할 수 있다.

4.1 클라이언트 패킷처리

클라이언트 패킷처리는 클라이언트의 동작이 게임 서버에 요청하는 패킷을 발생시키는 경우로 사용자간 채팅이 게임서버를 통하지 않고 이루어지는 채팅 패킷은 제외한다. 통신서버는 (그림 9)와 같이 클라이언트에서 패킷이 도착하면 클라이언트의 정보를 확인하여 해당 클라이언트가 위치한 블록을 소유한 게임서버에게 패킷을

전달한다.



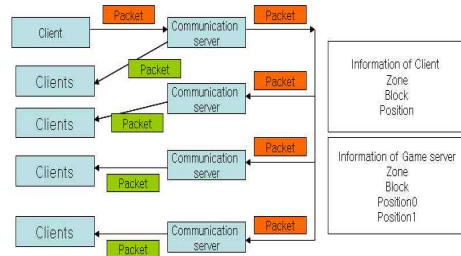
(그림 9) 클라이언트 패킷처리(채팅패킷 제외)

4.2 클라이언트 채팅 패킷처리

채팅은 인스턴스 메시지로서 게임처리에는 영향을 미치지 아니하고 사용자간 커뮤니티형성에만 관여한다. 이러한 채팅은 1대1, 1대다, 다대다 등의 형태가 있으며 본 분산구조에서는 통신서버에서만 이 기능을 담당한다.

통신서버는 클라이언트에서 채팅 패킷이 도착하면 채팅 패킷을 모든 통신서버에 방송한다. 통신서버들은 채팅 패킷과 관련된(채팅 방법에 따라 인접 블록에 존재, 동일 존에 존재, 모든 클라이언트) 클라이언트들에게 처리된 패킷을 전달한다.

채팅 패킷은 게임 패킷과 관련 없이 동작한다. (그림 10)은 이러한 클라이언트에서의 채팅패킷처리과정을 요약하여 보여준다.

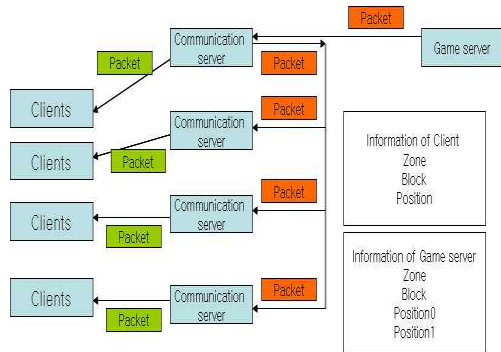


(그림 10) 클라이언트 채팅 패킷처리

4.3 게임서버 패킷처리

게임서버에서 생성되는 정보를 각 사용자에게 맞게 클라이언트로 전송하기 위하여 통신서버는 게임 서버에서 클라이언트로 전송을 하는 패킷이 도착하면 패킷을 모든 통신서버에 방송한다. 통신서버들은 먼저 패킷에 포함된 사용자의 위치 정보를 갱신하고 패킷과 관련된(인접 블록에

존재) 클라이언트들에게 처리된 패킷을 전달한다. (그림 11)에서는 이러한 게임 서버의 패킷처리 과정을 보여준다.

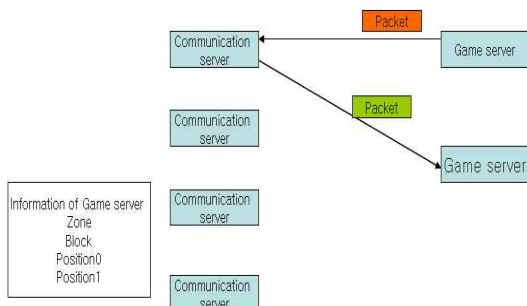


(그림 11) 게임서버 패킷처리

4.4 게임서버 블록 패킷처리

게임 서버에서 PC 및 NPC 의 이동 및 공격이 다른 서버의 블록으로 발생했을 때는 게임 서버에서 발생된 패킷을 처리하여 대상 블록을 소유한 게임서버로 전송한다.

이와 같은 방식은 게임서버의 성능과 처리하고 있는 사용자의 수를 감안하여 적절하게 게임 서버 간에 배분 될 수 있다. 즉, 블록을 이용한 분산처리 방식으로 게임서버의 동적인 관리가 가능하며 이에 따라 대규모의 사용자를 점진적으로 수용할 수 있어 서비스 론칭 시 비용을 사용자의 반응에 따라 적절하게 투입할 수 있다. (그림 12)에서는 이러한 게임 서버의 패킷처리 과정을 요약하여 보여준다



(그림 12) 게임서버 블록 패킷처리

5. 결론

본 논문에서는 MMORPG에서 대규모 사용자 처리를 위한 블록 기법을 이용한 분산형 게임서버구조를 제안하였다. 제안한 분산형 구조는 게임의 배경을 기본적으로 여러 개의 존으로 구분하고 하나의 맵을 존에 속한 서버들이 나누어 처리한다.

각 맵은 여러 개 일정크기의 블록으로 다시 나누어지고 게임 서버들은 이러한 블록들을 서로 나누어 처리하여 게임 서버의 성능에 따라 적절하게 배경처리를 배분하였다.

각 블록들의 기본 크기는 사용자가 게임 상에서 보여지는 가시거리 범위를 기본범위로 하였으며, 사용자 캐릭터가 블록 간 이동을 할 때 단순이동이나 공격을 위한 이동이나를 구분하여 처리 될 수 있도록 하였다.

본 논문에서 제안한 블록 기법을 도입한 분산형 게임 서버 구조는 게임서버와 통신서버를 통한 블록처리 방식을 통해 게임배경의 추가/삭제를 용이하게 할 수 있어 대규모의 사용자를 수용할 수 있음은 물론 추후 사용자가 직접 만드는 배경도 기존 게임 배경에 연동될 수 있도록 하였다.

또한 게임배경을 일정한 블록으로 나눔으로써 서버의 성능이나 처리하는 부하량에 따라 적절하게 부하를 동적으로 배분하여 시스템의 효율성을 증가시키고 안정성을 높였다. 향후 연구로는 게임배경을 Plug-Play 기능처럼 자동으로 추가 삭제할 수 있는 프로토콜에 대한 연구가 필요하다.

참 고 문 헌

- [1] 제시카 멀리건, 브라짓 페트로브스키, “온라인게임기획, 이렇게 한다”, 제우미디어, 2006.
- [2] 이남재 외 2, “온라인 게임에서의 효율적인 클라이언트 접속처리를 위한 비대칭 분산형 다중 서버 구조”, 정보처리학회 논문지 제12-B권 제4호(통권 제100호), pp.27-38, 2005.
- [3] 이남재, 곽훈성, “진화하는 온라인 롤플레이 게임을 위한 분산형 게임 서버 모델”, 한국게임학회 논문지 제2권 1호, pp 36~41, 2002.
- [4] 김범균, 안동연, 정성중, “MMORPG에서의 부하 분산

을 위한 가상 영역 정보 기반 동적 지역 분할”, 한국정보처리학회 논문지 A, VOL. 13 NO. 03, pp. 223 ~ 230, 2006.

[5] IBM Grid Research Lab "Butterfly.net : Powering Next-Genertation Gaming with Computing On-Demand", <http://www.butterfly.net>, 2004.

[6] S. Y. Hu, G. M. Liao:"Scalable peer-to-peer networked virtual environment", Proc. of the 3rd Workshop on Network and System Support for Games (NETGAMES 2004), pp.129-133, 2004.

[7] T. Henderson: "Latency and Behaviour on a Multiplayer Game Server", Proc. of 3rd Int'l. Workshop on Networked Group Communication (NGC2001), LNCS 2233, pp.1-13, 2001.

[8] A. Abdelkhalek, and A. Bilas, "Parallelization and performance of interactive multiplayer game servers," IEEE, pp.26-30, 2004.

[9] G. Huang, M. Ye, and L. Cheng, "Modeling system performance in MMOG," IEEE Communications Society Globecom Workshops 29, pp.512-518, 2004.

[10] A. Abdelkhalek, A. Bilas, and A. Moshovos, "Behavior and performance of interactive multi-player game servers," ISPASS01(2001), Vol.4, No.6, pp.137-146, 2001.



김 순 곤

1987년 : 동국대학교 교육대학원
전산교육학과(교육학석사)

1999년 : 전북대학교 일반대학원
전자계산기공학과 공학박사)

1987년 1995년 3월 : 한국원자력연구소 선임연구원

1995년 3월 ~ 현재 : 중부대학교 컴퓨터학과 교수
관심분야 : 온라인게임, 게임데이터베이스, 게임보안, 분산형 온라인 게임서버구조