

# Seamless Games 에서의 부하 분산을 위한 MigAgent 시스템 설계

## Degine of MigAgent System for Load Balancing in Seamless Game

원동기\*, 이정진\*, 황호전\*\*, 두길수\*\*, 김법균\*\*\*, 안동언\*\*\*\*, 정성종\*\*\*\*  
 Dong Kee Won\*, Jeong Jin Lee\*, Ho Jeon Hwang\*\*, Gil Su Doo\*\*,  
 Beob Kyun Kim\*\*\*, Dong Un An\*\*\*\*, Seung Jong Chung\*\*\*\*

**Abstract** - MMORPG enables thousands of players to play in an evolving virtual world at the same time over the internet. So, the load of this kind of games is heavier than that of any other precedents. One of load balancing methods is map-partition to divide the load of entire system which is vulnerable to delay message between clients and servers. In this paper, we propose a game architecture using MigAgent to help migration of player character between field servers and to reduce message traffic between clients and field servers.

**Key Words** :Seamless, MigAgent, Field Server, PlayerCharacter, MMORPG

### 1. 서론

최근 온라인 RPG게임은 MMORPG(Massively Multiplay on-line Role Playing Game)[1][2]가 주종을 이루고 있다. 이러한 환경은 과거의 한정된 게임 공간 보다 넓은 지형을 지원하며 사용자가 이러한 넓은 지형을 여행할 때 끊임없이 트인 공간을 제공해야 한다. 이때 개개의 서버가 관리하는 위치와는 관계 없이 게임 유저는 개개의 서버들로 구성된 게임 공간을 하나의 지역으로 인식하게 하고 분산 서버들이 구성된 게임 월드를 자유롭게 이동 할 수 있는 분산 게임 서버 방식을 Seamless 라고 한다. 이 방식은 게임 유저에게 넓은 환경을 제공 한다는 장점과 함께, 게임 서버들 간의 영역들이 서로 인접하고 있는 경계 부분에서의 처리가 매우 복잡하다는 단점을 갖는다[3].

이러한 문제를 해결 하기위해서 [4]에서는 Migration 서버를 제안한다. AC(Adjacency Cell)에 위치한 PC(Player Character)들을 Migration 서버가 관리하므로 경계 부분에서의 처리를 원활하게 하지만, AC 내의 PC가 집중될 경우 좋지 못한 성능을 보인다. [5][6]에서는 Grid 형식으로 맵을 나누고 나누어진 맵들 끼리 동기화함으로 동기화로 인한 성능저하를 어느 정도 예방 할 수 있지만, AOI(Area Of Interest)[5]내 동기화를 위해 많은 연산을 필요로 한다. 본 논문에서는 Field Server 들 간의 트래픽을 줄이기 위해 기존 방법[4]에 PC에 대한 에이전트 역할을 추가한 MigAgent를 제안한다. 이를 통해, PC의 Field Server간 이동 시 효율적인 서버간 트래픽 관리가 가능하며, PC에 대한 에이전트 역할을

통해 클라이언트와 게임 서버간의 트래픽을 줄일 수 있다.

### 2. Seamless 게임 서버의 구조

분산서버 방식은 게임공간을 연결하는 방식에 있어 두 가지 방식으로 나눌 수가 있다[4].

첫 번째 방식은 게임 공간내의 여러 영역이 단절되어 있어서 영역간의 상호작용은 불가능하지만 특정 포탈을 이용해서 다른 영역으로 이동이 가능한 방식이다. 그러나 이 방식은 쉽게 확장이 가능하지만 다중 서버 방식에 비해 크게 장점이 없다. 다수의 사용자들을 게임 공간에서 서비스가 가능하지만 넓은 게임 지역을 제공할 수가 없고 공간을 이동 시 포탈을 이용해야 하므로 공간별로 단절된 느낌을 제공하여 분산 서버 방식을 채택한 반면 게임 유저들에게는 다중서버 방식과 비교하여 큰 변화를 제공할 수가 없다.

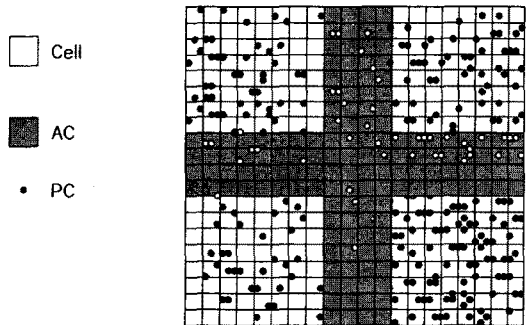


그림 1 Seamless 방식의 맵 분할

두 번째 방식은 (그림1)과 같이 게임공간내의 분할된 영역들이 서로 끊임없는(Seamless) 하나의 게임공간을 구성하는 방식이다. 이 방식은 사용자들은 서로 다른 서버가 담당하는

### 저자 소개

- \* 全北大學 컴퓨터學科 碩士課程
- \*\* 全北大學 컴퓨터學科 博士課程
- \*\*\* 全北大學 工學硏究員 硏究員
- \*\*\*\* 全北大學 컴퓨터學科 教授 · 工博

영역들 간을 자유롭게 이동한다. 사용자들은 자신이 다른 영역을 넘어갈 때 약간의 지연을 제외하고는 거의 느끼지 못한다. 그러므로 게임 유저들의 입장에서는 게임 지역들이 비록 분산 서버로 인해 지역별로 분할이 되어 있지만 하나의 넓은 게임 공간을 제공 하는 것으로 느끼게 된다.

하지만 이 방식은 서버들이 서로 인접한 지역에 있어서의 유저 이동처리의 복잡성을 가지고 있다. (그림1)에서 진하게 표시된 부분이 AC이며 인접 Field Server와의 경계에 해당하는 부분이다. PC가 AC에 있을 경우, 상태 동기화에 있어서 보다 복잡한 처리 과정이 필요하고 그 양 또한 증가한다[3].

### 3. MigAgent 를 이용한 부하 분산

맵은 기본적으로 cell이라 불리는 기본 단위로 구성되어 있으며 각 Field Server는 전체 맵 중 독립적으로 자신이 관리하는 영역을 가지고 있으며 이를 Field라 부른다. 이때 각 cell 내에 존재하는 PC 및 NPC는 Field Server에 의해 관리 되고 정보는 동기화 된다. 이 cell 중에 다른 Field Server가 관리하는 Field와 인접한 cell을 AC(Adjacent Cell)이라 부르며 이 cell 내에 위치한 PC들은 일반적으로 다른 Field Server로 이동할 가능성이 높은 PC들이다.

본 논문에서는 AC에 존재하는 PC들을 관리하기 위한 별도의 기능을 가진 MigAgent를 두고 이를 통해 인접 Field Server로 이동할 가능성이 높은 PC들의 정보를 관리함으로써 인접 Field Server로 이동할 경우 발생하는 트래픽을 제어하고자 한다. AC에 위치한 PC들의 정보는 MigAgent에 등록 되고 인접 Field Server의 안전한 cell로 이동할 때까지 MigAgent가 이 정보를 유지 및 필요한 동기화 작업을 수행한다. 이러한 과정을 통해 각 Field Server가 독립적으로 관리하는 지역별 관리가 가능하다.

#### 3.1 Migratoin 서버

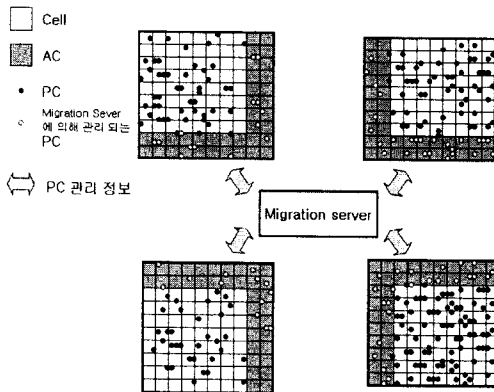


그림 2 Migration server

Migration 서버[4]의 구조는 (그림 2)와 같다. PC들이 AC 내부로 들어올 경우 Migration 서버에 등록 되게 되고 Migration 서버[4]는 AC에 들어오는 모든 PC를 처리하게 된다. 하지만, 실제로 다른 서버로 이동하지 않더라도, AC에 접근 하는 모든 PC에 대한 정보를 처리해야 하기 때문에 실제

Field Server의 필요 트래픽 이상의 트래픽을 발생시켜 사용자 응답시간이 길어지는 단점이 있다.

#### 3.2.Migration서버의 개선

본 논문에서는 PC의 방향성과 AC내에서의 잔류시간을 고려하여 트래픽 양을 조절한다. AC에 진입 또는 이탈하는 PC의 수가 많을 경우 오히려 Migration 서버로 인해 트래픽이 증가할 수 있다. 본 논문에서는 Migration 서버 대신 MigAgent를 도입하였으며 Migration 서버의 기본 기능에 변화를 줌으로써 성능의 개선을 시도한다.

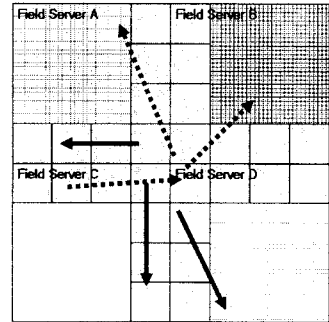


그림 3 PC의 이동 특성

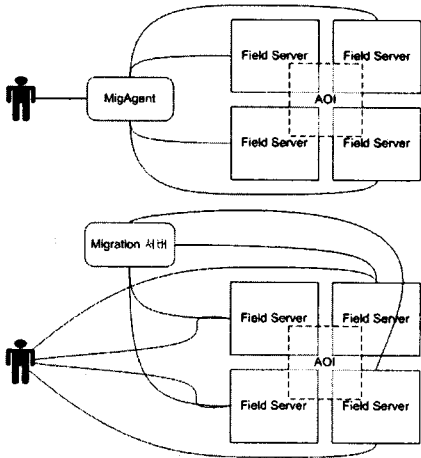
(그림 3)은 PC의 이동 방향 특성을 나타낸다. 실선으로 표현된 화살표는 PC가 Field Server 간 이동이 발생하지 않을 가능성이 높은 경우를 나타낸다. 일반적으로 PC들은 일정한 방향성을 갖고 이동 하게 된다[1][10]. AC에 PC가 진입하더라도 맵 경계면과 평행한 방향으로 진행하여 다른 서버로 향하지 않는 경우 다른 Field Server로 이동할 확률은 낮다고 판단하여 MigAgent가 관여하지 않는다. 또한, PC가 AC 내에 진입한 후 특정 시간이 지난 이후에 MigAgent가 관여함으로써 일시적인 AC 영역 진입에 대한 MigAgent의 반응을 줄인다. 따라서 PC가 AC 영역 내에 위치하더라도 인접 Field Server로 이동할 가능성이 없다면 Migration 서버를 통한 작업을 하지 않도록 함으로써 불필요한 Migration 서버의 관여를 최소화 한다.

#### 3.3Agent의 기능

PC가 Field Server를 옮기는 경우 Field Server내에서의 AOI 관리와는 달리 AOI에 포함된 Field Server 수에 비례하여 클라이언트와의 트래픽이 증가할 수 있으며 그 절차 또한 복잡해 질 수 있다.

본 논문에서는 이러한 상황의 PC에 대해 MigAgent가 PC의 Agent 기능을 수행하도록 함으로써 Field Server는 MigAgent를 PC로 간주하고 클라이언트는 MigAgent를 Field Server로 간주하도록 하여 부하를 줄이고 구성을 단순화 할 수 있도록 한다.

(그림 4)의 경우와 같이 고속의 안전한 통신망을 이용하는 Field Server군 사이에서는 트래픽이 동일하지만 클라이언트와 서버군 사이의 트래픽은 (AOI에 포함된 Field Server의 수 - 1) 만큼 감소한다.



AOI에 포함된 필드의 수에 따른 트래픽의 변화는 표 1과 같다. 일반적인 방법(MigAgent를 사용 하지 않을 때) AOI가 몇 개의 필드에 포함 되느냐에 따라서 PC와 Field Server 간 트래픽은 크게 달라진다. 하지만 MigAgent를 사용함으로써 트래픽의 양은 일정 수준이상으로 증가 하지 않는다.

AOI에 포함된 필드 수	MigAgent 사용	MigAgent 사용 하지 않음
1	NOP/MOFS*1	NOP/MOFS*1
2	NOP/MOFS*1	NOP/MOFS*2
4	NOP/MOFS*1	NOP/MOFS*4

NOP : Number of PC  
MOFS : Number of FS

표 1 AOI에 포함된 필드의 수에 따른 트래픽의 변화

#### 4. 결론 및 향후 연구 방향

Seamless 한 넓은 지형에 대한 욕구는 앞으로도 계속될 것이다. 단순히 넓은 지형이 아닌, 고수준의 그래픽에 대한 욕구와 빠른 응답 속도에 대한 열망은 컴퓨터의 발전과 함께 증가할 것이다. 이러한 요구에 맞춰나가기 위해서 효율적인 서버 구조는 필수적인 것이다. 본 논문에서는 Seamless한 서버 구조에서 성능을 최적화하기 위해서 MigAgent를 제안하였다. Seamless 환경에서 동기화를 위한 Field Server 간의 데이터 교환은 PC의 응답 시간에 대한 지연을 가져온다. Field Server간의 데이터 교환 횟수를 줄일 수 있는 MigAgent는 게임에 대한 사용자의 접근성을 높일 수 있을 것으로 예상된다.

향후에는 Field Server간의 동기화에 대한 연구와 Field Server 서버 간에 실질적인 데이터양을 줄임으로 MigAgent의 효율을 높일 수 있는 방향에 대한 연구가 필요하다.

#### 참 고 문 헌

[1]전일근, 대용량 물-플레이팅 게임을 위한 AOI에 관한 연구,

정보처리학회 2004.5

[2]이승욱, "MMORPG 게임엔진의 성능개선을 위한 분할공간에서의 충돌검출", 한국정보처리학회 논문지 B,2003

[3] 신동원, A study on Online Game Server Architecture, Synczone.Net

[4] 문성원, 분산 seamless 게임 서버에서의 효율적인 게임 공간 관리 기술, 서강대 정보통신 대학원 ,2004

[5] Jesse Aronson "Dead Reckoning: Latency Hiding for Networked Games", [http://www.gamasutra.com/features/19970919/aronson\\_01.htm](http://www.gamasutra.com/features/19970919/aronson_01.htm), September 19.1997

[6] Zou, L., Ammar M., Diot, C, "An evaluation of grouping techniques for state dissemination in networked multi-user games", Modeling, analysis and simulation of computer and telecommunication systems,2001

[7] 이정진 외 3명 "MMORPG 부하 분산을 위한 동적 맵 분할 시스템 설계" 정보과학회 2005년 춘계 학술대회 논문집

[8]이남재 박훈성 "진화하는 온라인 블블레이팅 게임을 위한 분산형 게임 서버 모델" 한국게임학회 논문집

[9] IBM Grid Research Lab "Butterfly.net: Powering Next-Generation Gaming with Computing On-Demand"

[10] [http://www.gameis.org/Korean/Game\\_Intro/Game\\_Genre/OnlineGame.htm](http://www.gameis.org/Korean/Game_Intro/Game_Genre/OnlineGame.htm)