

# MMORPG에서의 부하 분산을 위한 가상 영역 정보 기반 동적 지역 분할

김 법 균<sup>†</sup> · 안 동 언<sup>\*\*</sup> · 정 성 종<sup>\*\*\*</sup>

## 요 약

MMORPG(Massively Multiplayer Online Role-Playing Game)은 대규모의 인원들이 동일한 가상공간에서 동일한 시각에 상호작용하면서 활동하는 온라인 롤플레이팅 게임이다. 대부분의 경우, 서버와 대역폭 등에서 상당한 수준의 하드웨어 사양과 지원 스태프를 필요로 한다. 수많은 개발자들의 노력에도 불구하고 사용자들은 여전히 플레이어의 과밀, 락, 그리고 부족한 지원 등에 대해 불만을 가지는 경우가 많다. 본 논문에서는 MMORPG에서의 부하 분산을 위한 동적 지역 분할 방법을 제안한다. Field, sector group, sector, cell 등으로 구성되는 가상 영역 정보를 가진 VML(Virtual Map Layer)를 이용하여 지역을 분할함으로써 부하 분산을 시도한다. 실험 결과, 제안된 기법이 23~67% 정도의 부하를 줄이는 효과를 가져왔다. 맵 데이터, 자원의 상태, 그리고 사용자들의 행동 패턴의 변화 등에 대해서는 VML에 대한 간단한 수정으로 유연하게 적용할 수 있다.

키워드 : 게임, 부하 분산, MMORPG

## A Dynamic Map Partition for Load Balancing of MMORPG based on Virtual Area Information

Beob-Kyun Kim<sup>†</sup> · Dong-Un An<sup>\*\*</sup> · Seung-Jong Chung<sup>\*\*\*</sup>

## ABSTRACT

A MMORPG(Massively Multiplayer Online Role-Playing Game) is an online role-playing game in which a large number of players can interact with each other in the same world at the same time. Most of them require significant hardware requirements(e.g., servers and bandwidth), and dedicated support staff. Despite the efforts of developers, users often cite overpopulation, lag, and poor support as problems of games. In this paper, a dynamic load balancing method for MMORPGs is proposed. It tries to adapt to dynamic change of population by using dynamic map-partition method with VML(Virtual Map Layer) which consists of fields, sector groups, sectors, and cells. From the experimental results, our approach achieves about 23~67% lower loads for each field server. By the modification to Virtual Area Layer, we can easily manage problems that come from changes of map data, resources' status, and users' behavior pattern.

Key Words : Game, Load Balancing, MMORPG

## 1. 서 론

최근 게임 시장에서는 그 동안 강자로 군림해 오던 아케이드 게임과 개인용 컴퓨터 게임의 시장 규모는 줄어들고 있는 추세이지만, 온라인 게임과 모바일 게임이 새로운 시장을 창출하며 점차 규모가 커지고 있다[1]. 이렇게 놀라운 성장을 보이는 이유는 현실 세계를 온라인 게임의 가상공간으로 옮겨놓음으로써 현실에서 이루기 힘든 꿈과 능력을 이

루게 해주고 커뮤니티 활성화를 통해 인간 사회와 비슷하게 만들어준다는 점 때문이다.

그러나 이와 같이 사람들의 관심과 수요가 커지면서 폭증하는 사용자와 트래픽을 감당하지 못해 인구 과밀, 락, 스태프의 지원 부족 등으로 인한 사용자의 불편 등을 해결하기 위한 노력을 여전히 필요로 하고 있다. 이에 대비하여 그동안 온라인 게임 업계에서는 다양한 형태의 게임 서버 구조를 개발하여 왔으며 이제는 수만 명의 동시 접속자도 원활히 대처할 수 있다. 하지만 아직도 사용자가 급격히 늘어날 때마다 게임의 속도와 안정성에 문제를 보일 뿐 만 아니라 향후 빠르게 늘어날 사용자를 감당할 만한 뚜렷한 대책이 없다.

<sup>†</sup> 준 회원 : 전북대학교 공학연구원 연구원  
<sup>\*\*</sup> 중신회원 : 전북대학교 컴퓨터공학과 교수  
<sup>\*\*\*</sup> 정 회원 : 전북대학교 컴퓨터공학과 교수  
논문접수 : 2006년 3월 7일, 심사완료 : 2006년 5월 15일

이러한 특성이 뚜렷하게 나타나는 분야가 MMORPG(Massively Multi-player Online Role Playing Game) 장르의 온라인 게임이다[2]. MMORPG는 대규모의 인원들이 동일한 가상공간에서 동일한 시각에 상호작용하면서 활동하는 온라인 톨플레잉 게임이다. 대부분의 경우, 서버와 대역폭 등에서 상당한 수준의 하드웨어 사양과 지원 스태프를 필요로 한다. 따라서 설계 시 많은 제약이 가해질뿐더러 고려해야 할 사항들이 많다. 성능, 안정성, 확장성, 개발 주기, 관리의 편의성, 비용 등에 대해 균형을 이루어야 하며 부하의 상당부분을 차지하는 DBMS에 대한 접근을 최소화할 수 있어야 한다. 대용량 메모리 복사가 빈번하게 이루어지므로 이에 대한 고려가 있어야 하며 네트워크 대역폭 및 지연을 고려하여 사용자들을 공정하게 취급할 수 있어야 하고 시스템을 구성하는 장비들의 특성을 고려해야 한다. 마지막으로 시스템이 항상 실시간으로 응답해야 한다는 점과 확장성을 고려해야 한다.

이러한 특성을 반영하기 위해 대부분 분산된 형태의 구조를 가지는 서버 구조로 설계되고 있으며 가상공간의 설계 단계에서부터 부하의 집중을 최소화할 수 있는 구조로 설계되고 있다. 특히, 가상공간의 설계에 있어 MMORPG의 특성을 극대화시키기 위해 분리된 다수의 가상공간을 유지하는 채널 방식 보다는 공간 영역을 분할하여 분리된 서버를 통해 관리하도록 함으로써 대규모의 단일 가상공간을 유지하는 Seamless 게임 방식이 선호되고 있다. Seamless 게임 방식으로 가상공간을 구성하는 경우 게임 시스템의 구동 전에 각 서버가 관할하게 될 맵 영역이 고정적으로 할당되어 관리하게 된다. 그러나 이러한 방식의 게임 시스템은 유동적인 사용자들의 활동 패턴에 대한 대처 능력이 떨어진다. MMORPG의 특성상 전체 맵 영역 중 사용자들에게 인기 있는 특정 영역이 여러 가지 이유로 인해 존재하게 되며 특정 이벤트가 있을 경우 그 집중은 더욱 심화된다. 이렇게 특정 영역 내의 PC(Player Character) 과밀 현상이 심화될 경우 이 영역을 관할하는 서버는 사용자의 액션에 대한 응답을 처리하는 데 더 많은 자원을 소비하게 되며 다른 영역의 서버들은 비효율적인 운영을 피할 수 없다.

본 논문에서는 Seamless한 MMORPG를 지원하기 위해 정적인 지역 분할 대신 동적인 지역 분할을 통한 부하 분산 방식을 제안한다. Field, sector group, sector, cell 등으로 구성되는 가상 영역 정보를 갖는 VML(Virtual Map Layer)를 이용하여 지역을 분할함으로써 부하 분산을 시도한다. VML은 전체 맵 영역에 대해 하위 단위의 영역 경계 정보를 담고 있으며 각 단위 영역 내의 부하를 측정하여 영역의 분할 또는 다른 영역과의 병합을 수행한다. 이를 통해 각 게임 서버의 부하에 따라 동적으로 관할 맵 영역의 분할 및 병합이 가능하고 사용자들에 대한 응답 시간을 허용 가능한 수준으로 조절할 수 있다. 또한 맵 데이터, 게임 서버의 상태, 그리고 사용자들의 행동 패턴의 변화 등에 대해서는 VML에서의 간단한 수정만으로도 유연하게 대처할 수 있다.

논문의 구성은 2장에서는 MMORPG에서의 부하 분산을

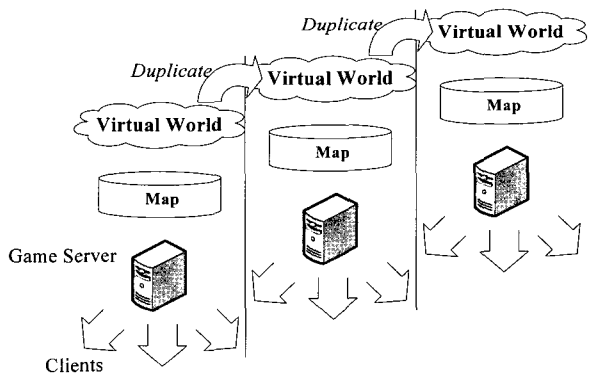
위한 가상공간의 구성 방법 및 관련 연구에 대해 살펴보고 3장에서는 동적 부하 분산을 위한 VML 기반 MMORPG 게임 시스템을 설계한다. 4장에서는 설계된 시스템을 구현 및 고찰하며 5장에서 결론을 맺는다.

## 2. MMORPG에서의 부하 분산

### 2.1 MMORPG에서의 부하 분산을 위한 게임 시스템의 유형

MMORPG의 경우 각 사용자들이 동일 시각에 동일 가상공간에 모여 상호작용을 하게 되므로 게임의 성격과 가상공간의 운용 방식에 따라 극명한 성능의 차이를 나타낸다. 게임 내에서는 각 사용자를 대신하는 PC가 그 역할을 대신하게 되며 일종의 아바타(Avatar) 역할을 한다. 기존의 MMORPG 시스템들을 가상공간의 운용 방식에 따라 구분하면 중앙집중식과 지역분할방식으로 나눌 수 있다[3, 4].

중앙집중식 게임 시스템은 하나의 고성능 서버에서 하나의 가상공간 전체를 처리하는 방식이다. 최근에는 사용자들의 증가로 인해 가상공간의 수를 늘리는 방법을 사용하여 부하를 조절하는 경우가 많다. 즉, 여러 개의 복제된 서버에서 각각 독립된 가상공간을 운영하는 채널 방식 구조로 운용(그림 1)하는 방식으로 각종 온라인 카드 게임이나 대전 격투 게임 등에 많이 이용된다. 이 방식의 경우 설계와 개발이 용이하며 사용자의 수가 늘어날 경우 게임 서버의 추가로 쉽게 대처할 수 있다. 그러나 MMORPG의 특성상 거의 무제한에 가까운 PC와 하나의 서버로는 감당하기 힘든 크기의 가상공간을 유지해야 하므로 MMORPG를 위한 게임 시스템으로는 적합하지 않다.

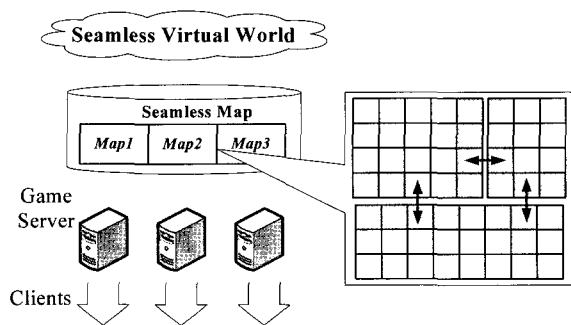


(그림 1) 채널 방식의 게임 시스템

지역분할방식의 게임 시스템은 하나의 가상공간을 다수의 영역으로 분할하고 분할된 영역만을 담당하는 다수의 게임 서버들을 모아 하나의 거대한 가상공간을 구성하는 방식이다. 앞서 설명한 채널방식의 경우 다수의 독립된 작은 크기의 가상공간을 유지하는데 반해 이 방식은 하나의 거대한 가상공간을 유지하므로 보다 많은 PC를 감당할 수 있다는 장점이 있지만 복잡한 시스템 구조와 PC/NPC(Non-PC) 정보의 동기화 문제 등으로 인해 개발이 어렵다는 단점이 있다.

지역분할방식은 분할 지역 사이의 이동 방식과 지역분할 시점에 따라 분류할 수 있다.

분할된 지역 사이의 이동 방식에 따라 분류하면 명시적 지역분할과 묵시적 지역분할로 나눌 수 있다. 명시적 지역분할방식은 가상공간을 각각 독립된 지역으로 나누고 지역 사이의 이동 통로를 다리나 좁은 길 등으로 제약을 가하는 형식이다. 이런 제약을 통해 지역 사이의 이동을 단순화시키고 이 때 발생하는 서버들 사이의 트래픽을 최소화시킬 수 있다. 묵시적 지역분할방식은 지역과 지역사이의 경계를 사용자들이 거의 인지하지 않을 수준으로 매끄럽게 연결된 맵을 가상공간으로 사용한다. (그림 2)와 같이 각 게임 서버들이 관리할 영역을 설계 단계에서부터 분할하고 사용자가 자유자재로 경계를 넘나들 수 있도록 한다. 따라서 사용자(PC & NPC)들이 지역 사이를 이동하는 경우가 빈번하게 발생하게 되며 이 때 각 사용자 정보와 서버 정보의 동기화에 상당히 많은 부하가 걸린다.



(그림 2) 묵시적 지역분할방식의 게임 시스템

지역분할방식을 지역분할이 일어나는 시점에 따라 분류하면 정적 지역분할방식과 동적 지역분할방식으로 구분할 수 있다. 정적 지역분할방식은 게임 설계 단계에서 모든 지역의 분할이 이루어진 후 게임의 운용 중에는 지역의 분할 및 병합이 이루어지지 않는 방식으로 예기치 못한 특정 지역에 대한 인구 과밀 등에 대해 적절하게 대처하지 못하는 단점이 있다. 동적 지역분할방식은 설계 단계에서 지역 분할이 이루어지는 것은 동일하지만 게임의 운용 중 서버의 부하에 따라 각 서버의 관리 영역을 동적으로 재조정하여 분할 및 병합할 수 있도록 하는 방식이다. 그러나 이 방식은 PC 및 NPC의 동기화 문제 뿐 만 아니라 게임 서버의 초기화 또는 병합 시 최적의 해결책을 제시하기 어렵다는 단점이 있으며 실제 적용된 경우는 아직 보고된 바 없다.

최근 MMORPG의 게임공간은 하나의 큰 월드를 여러 개의 영역으로 분할하여 각각 담당서버들을 따로 두는 구조이다. 이런 경우 가상공간의 각 영역 간 인접 지역에서의 다른 영역으로의 이동으로 인한 User Data 동기화 문제, User AOI(Area of Interest), 이동이 많은 경우 부하 등의 문제들을 고려해야 한다. 이런 가상공간을 기반으로 구축되는 게임 시스템을 Seamless 게임이라 하며 그 경계 영역에서의 다양한 문제에 관한 연구가 진행되고 있다.

## 2.2 온라인 게임에서의 부하 분산을 위한 연구 사례

온라인 게임에서의 부하 집중을 해소하기 위한 다수의 노력이 있어왔다. 대표적인 사례들로 Optimal Grid[5], CORBA(Common Object Request Broker Architecture) 기반 분산 컴퓨팅[6], EMERGENT 프로젝트[7] 등을 들 수 있다.

OptimalGrid[5]는 그리드 응용 프로그램을 쉽게 제작할 수 있도록 하고, 기존의 어플리케이션들을 그리드 컴퓨팅에 적용하고 관리할 수 있도록 하는데 목표를 두고 IBM Almden Research Lab에서 연구 개발된 미들웨어이다. OptimalGrid는 FEM(Finite Element Modeling)을 기반으로 데이터의 연결 상관관계에 따라 요소들을 분리하여 분리된 요소들을 하나의 독립된 문제로 규명하여 해결한 후 통합하는 방식이다. 분리된 요소들 사이의 정보 교환은 인접 요소로 제한되며 온라인 게임에 적용될 경우 분할된 가상공간으로 모델링 된다. 부하를 동적으로 분산시킬 수 있는 획기적인 방법이기도 하나 각 온라인 게임마다의 특성들을 반영하지 않은 채 산술적인 분산 알고리즘을 사용한다는 단점이 있다. 즉 일반적인 MMORPG에서의 맵들과는 상당히 다른 환경에서의 부하 분산을 의미하며 Seamless 게임에 적용할 경우 상당한 오버헤드가 발생한다.

이종의 분산된 환경 하에서 응용 프로그램들을 서로 통합할 수 있도록 CORBA[6]를 사용하는 경우도 있다. CORBA 자체적으로 제공하는 IIOP(Internet Inter-ORB Protocol)를 이용하고, 동적으로 쓰레드를 생성하고 및 다른 서버로 연결해 줌으로써 좀 더 효율적인 동시성 제어 및 네트워크 부하를 줄일 수 있도록 제안되었다. 그러나 CORBA 자체가 서버 쪽의 부하 분산과 오류 감내 구현이 아직 미미하고 설계 및 구현이 아직은 상대적으로 어렵다는 단점이 있다.

EMERGENT 프로젝트는 Globus Toolkit을 이용하여 그리드 컴퓨팅 기술을 도입한 Butterfly Grid[7]에서 출발하였으며, 과학 연산 분야에서만 쓰이던 그리드 컴퓨팅 기술을 게임 네트워크에 적용할 수 있는 시스템 구조로 설계 및 연구하였다. 오픈소스인 미들웨어 Globus Toolkit에 기반 하여 자체 개발한 시스템으로 구축하였으며, PC를 상황에 따라 서버를 넘나들 수 있도록 제안된 시스템이다. 묵시적 지역 분할 구조에 네트워크 트래픽, 데이터베이스 접근 시간 등을 줄이기 위해 자체적으로 시스템 구조를 최적화한 독특한 구조이지만 동적인 부하 분산이 제대로 이루어지지 못하고 있다는 단점이 있다.

각 프로세스의 종류에 따라 게임 서버를 독립적으로 구축하여 프로세스를 분산시키는 방식[8, 9]의 경우 맵의 확장이나 사용자의 급작스런 집중 또는 증가에는 여전히 취약한 면을 보인다.

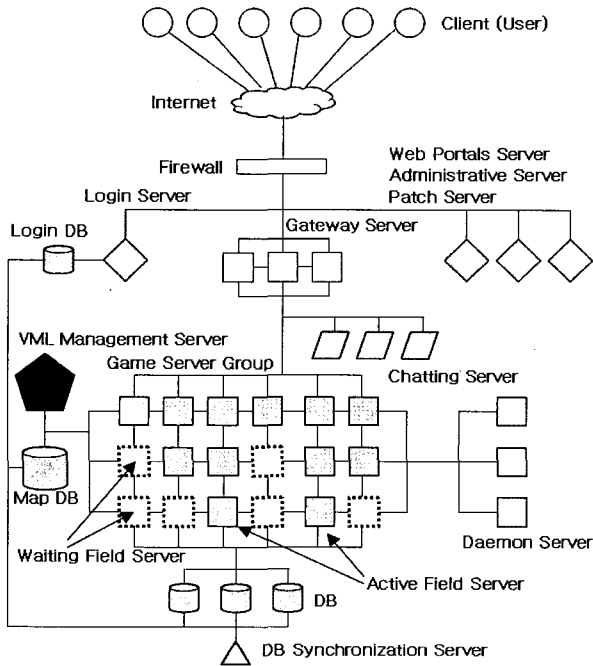
이와 같이 대부분의 연구들은 동적으로 변하는 PC들의 이동 패턴이나 부하의 집중 등에 대해 여전히 취약한 면을 가지고 있다. 본 논문에서는 이러한 상황에 능동적으로 대처할 수 있는 MMORPG 게임 시스템을 위해 설계단 계에서 각 월드의 특성에 따라 이웃한 월드들과의 관계를 고려하여 VML을 설계하도록 하고 분할 및 병합이 이루어질 때 VML

정보를 기반으로 동작하도록 함으로써 게임 시나리오에 적합하게 동작하도록 하는 게임 시스템을 제안한다.

### 3. 가상 영역 정보 기반 동적 지역 분할 시스템 설계

#### 3.1 동적 지역 분할을 지원하는 MMORPG 시스템

(그림 3)은 일반적인 MMORPG 시스템에 VML 지원을 위한 서버를 추가한 것을 근사화한 것이다. VML Management Server는 각 서버에서 일정 주기마다 보고하는 부하를 바탕으로 분할 및 병합을 시도한다.



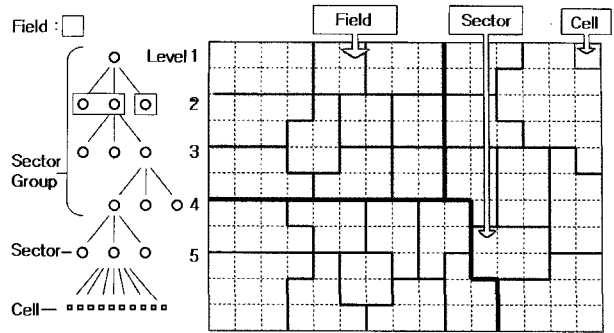
(그림 3) 동적 지역 분할을 지원하는 MMORPG 시스템의 구조

게임의 설계 단계에서, 게임의 시나리오에 맞는 가상공간 내의 각 지역마다 독특한 특성을 부여하게 되고 설계에 따라 PC가 많이 몰리는 영역과 그렇지 않은 영역이 어느 정도 구분되어 지며 VML 제작 단계에서도 이러한 정보를 활용해야 한다. VML 제작 단계에서 설정한 정보에 의해 동적 지역분할이 이루어지므로 게임의 설계 단계에서부터 이러한 사항을 고려해야 한다. 또한, 게임 운용 중 예상하지 못한 PC들의 활동 패턴 등이 등장했을 경우 맵 자체의 수정이 아닌 VML의 수정만으로도 쉽게 적용시킬 수 있다.

#### 3.2 VML

기존의 맵 구조에 가상 영역 정보를 담은 VML(Virtual Map Layer)이란 계층을 추가하고, 이 계층에서 나는 가상의 영역단위 마다 부하를 측정하여, 지역의 분할 및 병합을 시도한다.

VML은 field, sector group, sector, cell 등으로 구성(그림 4)된다. Field는 각 게임 서버가 관리하는 영역을 의미하며

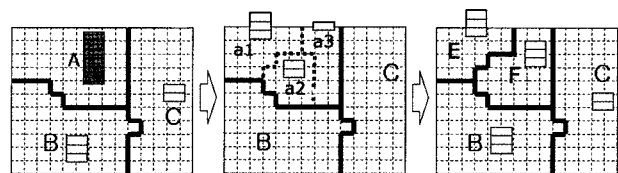


(그림 4) VML의 구성 요소

설계 방침에 따라 sector 또는 sector group의 집합으로 구성된다. 각 field 내의 부하를 측정하여 field를 구성하는 sector 또는 sector group의 소속 집합을 달리함으로써 지역 분할 및 병합을 수행한다. sector는 지역 분할 및 병합이 이루어지는 가장 작은 단위로 Optimal Grid의 OPC(Original Problem Cell)[5]와 유사한 개념이다. 게임 설계 단계에서 설계자가 맵의 특성이나 설계 철학에 따라 일정 영역에 포함된 cell들을 sector에 포함시킨다. 각 cell에서의 부하를 측정하여 포함된 sector의 부하에 합산한다. cell은 VML의 최소 단위로서 sector 정의의 편의를 돕기 위해 사용된다. 부하 측정이 용이하도록 격자 형태를 가지며 맵의 특성에 따라 sector를 정의할 때 포함되는 cell의 개수가 달라질 수 있다.

특정 field에 임계치( $MaxLoF_i$ ) 이상의 부하가 일정시간( $T_{term}$ ) 동안 발생하면 field를 구성하는 sector들의 부하를 합산하여 적절한 분할 방법을 선택한다. 병합과정에서도 2개 이상의 field의 부하의 합계가 임계치( $MinLoF_i$ ) 이하의 값으로 일정시간( $T_{term}$ ) 이상 유지되면 병합과정이 진행된다. 설계 단계에서 각 sector에 가중치를 두고 이 값을 field의 분할 또는 병합과정에서 활용할 수 있다.

(그림 5)는 지역의 분할과정을 보여주고 있다. 지역 분할이 이루어질 때 그 후보가 되는 하위 sector 또는 sector group은 초기에 구성된 VML의 계층 구조를 따른다. 산술적으로 계산된 부하를 바탕으로 특정 지역만을 따로 떼어내는 방법도 가능하지만 유지 보수가 어렵고 추후 병합이 이루어지는 과정에서 복원이 어렵다는 단점이 있으며 분할 및 병합이 이루어지는 과정에서 많은 양의 트래픽이 일어나는 부작용이 있다. 따라서 게임의 설계 단계에서 게임의 설계 철학 및 시나리오 등과 관련된 맵의 영역을 지정한 후 분할 및 병합이 이루어지는 순서를 계층구조로 표현해 놓는 것이 더 현실적이며 VML이 이러한 방식을 표현해 놓은 것이다.



(그림 3) 지역 분할 과정

(그림 6)에서 하위 단위의 sector를 구할 때 VML에서 그 정보를 획득하며 분할 정보를 기록으로 남아 추후 병합이 이루어질 경우 이 정보를 바탕으로 분할된 field들이 다시 병합되는 방식으로 진행된다. VML의 계층구조는 일종의 트리 구조와 같으며 부모 노드(A)의 영역은 자식 노드(a, b, c)의 영역의 합집합과 동일(식1)하고 부모 노드의 영역을 분할할 경우 자식 노드들의 부분 집합이 그 후보가 된다.

$$\begin{aligned} A &= \{a_1, a_2, a_3, \dots, a_n\} \\ A &\supset E \\ F &= A - E \end{aligned} \quad (1)$$

```

FieldPartition (field A)

s1, s2, s3 = get sub-sectors
ls1, ls2, ls3 = get load of s1, s2, s3
(sc.a, sc.b) = get best combination of s1, s2, s3
               based on ls1, ls2, ls3

allocate new field (sc.a)
reduce the existing field (sc.b)
synchronize field servers (sc.a, sc.b)
    
```

(그림 4) Field Partition Algorithm

부하를 측정할 때는 PC의 활동성, 데이터베이스 접근을 요하는 이벤트, 영역별 가중치, 서버의 성능 등 다양한 정보가 필요하다[10]. 특히, 데이터베이스 접근을 요하는 이벤트[11]의 경우 주기억장치에 대한 접근 시간과 수십 배 정도 차이를 보이므로 이에 대한 고려가 필요하다[12]. Field의 부하를 측정하기 위해서는 field 내에 위치한 sector들의 부하와 각 sector들의 가중치를 곱한 값을 합산하면 된다(식2). Sector의 가중치는 게임 설계 단계에서 각 sector의 중요도에 따라 값을 부여하며 값이 클수록 sector의 중요도가 높아 더 원활한 동작을 요구함을 의미한다. 각 sector의 부하(식3)를 측정할 때는 각 PC의 활동성 요소( $A_i$ )가 기본적으로 필요하며 이 값은 다양한 정보의 조합으로 구성된다[10]. 데이터베이스 접근을 요하는 이벤트의 수( $E_j$ ) 및 데이터베이스 접근 시간과 주기억 장치 접근 시간의 비율( $W_{event}$ )도 사용된다.

$$LoF_f = \sum_s^n (LoS_s \times W_s) \quad (2)$$

$$LoS_s = \sum_{i=0}^l A_i + W_{event} \times \sum_{j=0}^m E_j \quad (3)$$

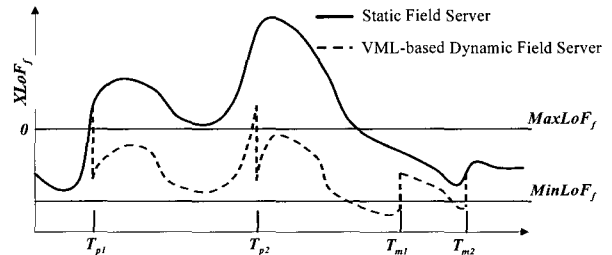
각 field의 부하는 field를 담당하는 field server가 수용할 수 있는 최대 부하를 넘지 않아야 하며 이 수치를 넘게 되면 field의 분할이 수행된다. 이를 위해 각 field의 부하를 계산한 후 field server가 수용할 수 있는 최대 부하와의 관계를 수치로 표현해야 한다(식3).  $MaxLoF_f$ 는 field server  $f$ 가 수용할 수 있는 최대 한계치를 의미하며  $XLoF_f$ 는 한계치를 초과한 부하이다(식4). 즉  $XLoF_f$  자체는 각 field server의 상태를 의미하고 이 값들의 합은 전체 게임 시스템의

상태라 볼 수 있으며 이는 사용자들이 느끼는 체감 속도가기도 하다.  $RXLoF_f$ 은  $XLoF_f$ 를  $MaxLoF_f$ 로 나눈 것으로 일반화시킨 수치이다(식5).

$$XLoF_f = \begin{cases} LoF_f - MaxLoF_f \\ 0 \end{cases}, \text{ if } LoF_f \leq MaxLoF_f \quad (4)$$

$$RXLoF_f = \frac{XLoF_f}{MaxLoF_f} \quad (5)$$

본 논문에서 제안한 기법을 사용할 경우 각 field server의 부하는 (그림 7)과 같이 그 변화를 예측할 수 있다. 분할이 이루어지는  $T_{p1}$ 과  $T_{p2}$  직후 각 field server는  $MaxLoF_f$ 를 넘지 않는 부하로 떨어지며 병합이 이루어지는  $T_{m1}$ 과  $T_{m2}$  직후  $MinLoF_f$ 보다 높은 부하를 가지게 된다. 그래프에서  $MaxLoF_f$ 를 넘는 부하는  $XLoF_f$ 로 정의되며 제안된 기법에서는  $XLoF_f$ 가 0보다 큰 값을 가지는 경우가 현저하게 줄어들 것으로 예측할 수 있다.



(그림 7) Field server 내에서의 이론적인 부하 및 초과 부하 변화 추이

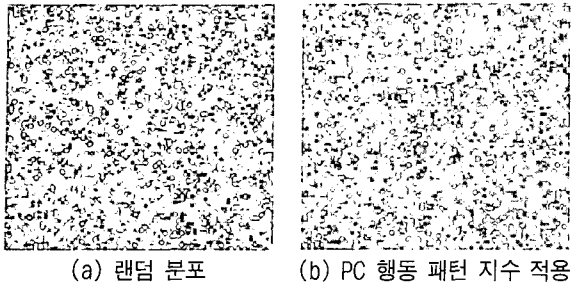
## 4. 구현 및 고찰

### 4.1 PC 행동 패턴 지수

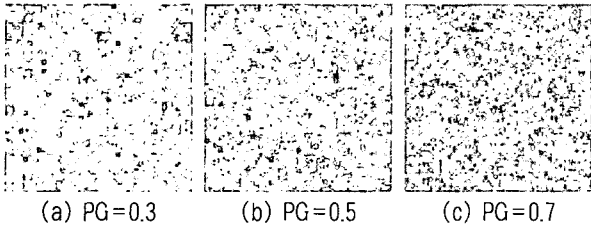
본 논문에서는 시스템의 성능 평가를 위해 PC의 행동 패턴을 조절하기 위한 척도로 PC의 활동성과 그룹핑 정도를 뜻하는 PA(PC's Activity)와 PG(PC's Grouping degree)를 도입하여 실험을 진행하였다. 실험에 사용된 PA와 PG는 다음 표와 같다.

<표 1> PC의 행동 패턴 지수

지수 구분	PA	PG
설명	PC의 활동성(턴당 정지/이동 확률 및 거리)	PC의 활동 패턴(동일 패턴으로 활동하는 PC 비율)
최소값	모든 PC의 이동을 멈춤	모든 PC가 동일 그룹에 소속
최대값	모든 PC는 매턴마다 최대거리 이동	그룹 당 PC의 수는 1
계산식	$\frac{\sum_i^n PC_i \times PCA_i}{n}$	$1 - \frac{\sum_j (\sum_i PC_{ji})}{n}$



(그림 8) PC 행동 패턴 지수 적용에 따른 PC 분포



(그림 9) PC의 분포 (PA=0.3)

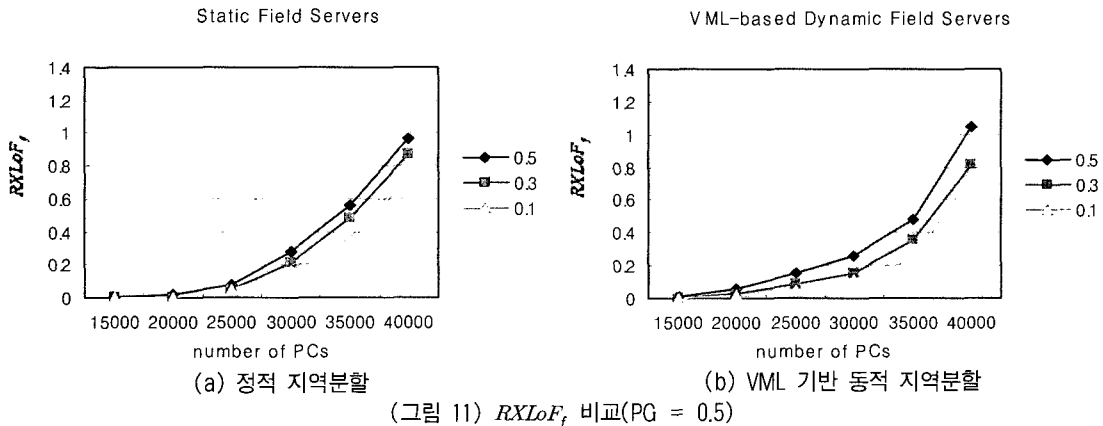
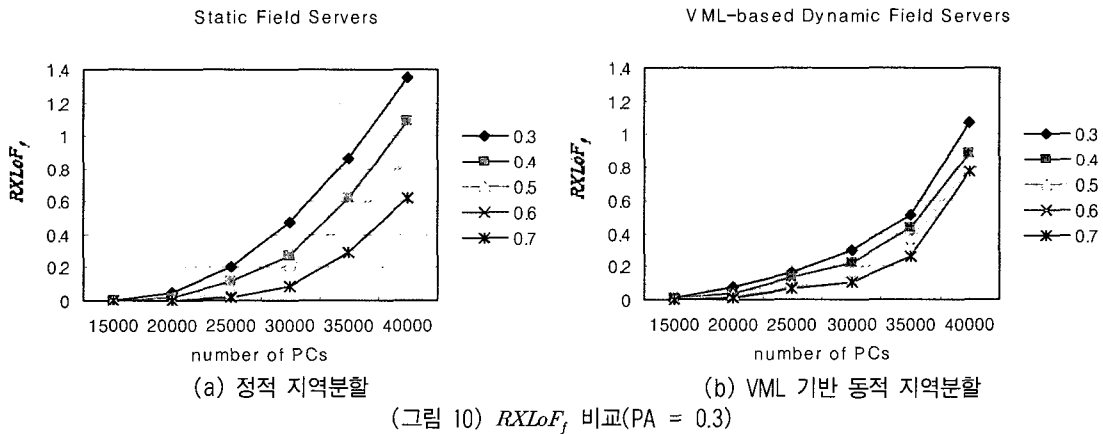
PC 행동 패턴 지수에 따라 가상월드에서의 PC의 분포를 살펴보면 (그림 8)과 같다. (그림 8)에서 (a)는 랜덤하게 PC를 배치시킨 것으로 전체 영역에 걸쳐 PC가 골고루 배치된 상황으로 해석되며 실제 게임에서는 맵의 특성 또는 이벤트

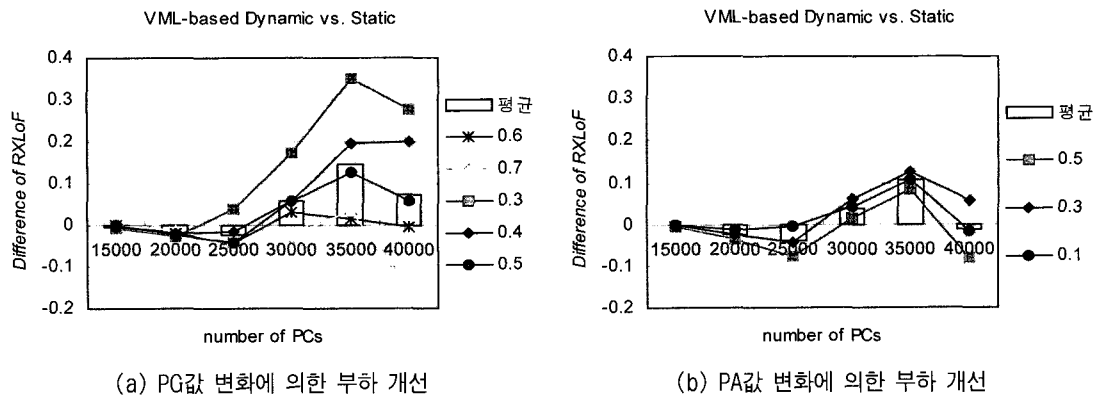
에 의해 특정 영역에 PC의 집중 현상이 발생하게 되므로 비현실적인 상황이다. (b)는 일정한 PA 및 PG 값을 부여한 후 임의의 턴이 지난 후의 상황으로 실제 게임 환경처럼 특정 영역에서의 밀도가 높아짐을 확인할 수 있다. (그림 9)를 보면 동일한 PA에 대해 PG의 값이 커질수록 골고루 분포된 후 활동하는 것을 확인할 수 있으며 특정 영역에 몰려있는 경우  $XL_{field}$ 의 평균값이 커질 것으로 예상할 수 있다.

(그림 10)은 PA의 값을 0.3으로 고정시킨 후 VML 기반의 동적 지역 분할을 적용한 경우와 그렇지 않은 경우에서의 초과 부하를 측정해본 것이다. PC의 수가 많아질수록 효과적임을 확인할 수 있다. 또한 PG의 값이 작을수록, 즉 특정 영역에 몰리는 정도가 강할수록 전체적으로 초과부하의 크기가 커진다. VML기반의 동적 지역분할을 사용함으로써 PG의 값이 0.3일 경우 23~67% 정도의 성능향상을 보였다.

(그림 11)은 PG의 값을 0.5로 고정시킨 후 PA의 값을 조절하면서 초과 부하를 측정해본 것이다. PA는 PC의 활동성을 의미하므로 그 수치가 커질수록 활동성이 높아 부하 증가의 한 원인이 될 수 있다.

(그림 12)는 (그림 10)과 (그림 11)에서 나타난 결과를 바탕으로  $RXLoF_f$ 의 변화량과 평균을 산출한 것이다. PA 값의 고정 후 PG값의 변화에 따라 부하를 측정해본 결과 (그림 12) (a)에서 VML 기반의 동적 지역분할에 의한 효과 크음을 확인할 수 있다. 즉, 특정 지역에 밀집하는 정도(PG)가



(그림 12)  $RXLoF_i$  변화 요인

클 경우에 더 큰 위력을 발휘한다고 할 수 있다. 전체적으로 VML 기반의 동적 지역분할을 한 경우 field server당 초과부하의 크기는 줄어들었으나 PC의 수가 커진 경우 오히려 성능 저하를 가져왔다. 이는 실험에서의 맵의 크기에 비해 PC의 턴 당 이동속도가 커서 나타나며 휴면 대기 서버의 부족으로 더 이상 분할이 일어날 수 없는 상황에서 기인하는 현상으로 분석된다.

#### 4.2 분할 및 병합으로 인한 오버헤드

본 논문에서 제안한 기법을 적용할 경우 VML에서 지정된 각 필드의 부하를 모니터링하고 필요시 분할 및 병합을 수행하는데 오버헤드가 발생할 수 있다.

본 논문에서 제안한 기법의 경우 VML의 각 필드는 독립된 필드 서버가 관리하며 이는 기존의 정적 지역 분할 방식의 MMORPG와 거의 유사하다. 기존의 MMORPG에서도 상시 모니터링을 통해 각 게임서버의 부하를 계산하여 측정하고 측정된 결과를 차후 분석을 위해 기록으로 남기고 있다. 본 논문에서 제안한 기법의 경우 이 과정의 중간에 VML Management Server가 추가되어 중계하도록 하고 있으므로 이 부분으로 인한 오버헤드는 없다고 할 수 있다.

분할 및 병합이 이루어지는 과정에서 오버헤드가 발생하는 경우는 각 필드 서버가 유지하는 PC 정보의 동기화, 새로 분할되거나 병합된 서버에서의 맵 데이터 로딩, 클라이언트 및 데몬 서버의 접속 방향 전환 등으로 구분할 수 있다. 기존의 게임서버에서도 각 필드 서버가 유지하는 PC 정보를 주기적으로 데이터베이스와 동기화하는 프로세스를 두고 있으며 본 논문의 기법에서 분할 또는 병합이 이루어질 경우 1회 증가한다. 맵 데이터는 정적인 데이터이므로 모든 필드 서버가 기본적으로 보관하고 있으며 해당 영역을 자신의 활성 영역으로 인식하느냐의 여부만이 달라지므로 이로 인한 오버헤드는 없다고 할 수 있다. 클라이언트 및 데몬 서버의 경우 접속 방향(필드 서버 변경)은 게이트웨이 서버에서 필드 사이를 이동할 때와 동일한 방식으로 이루어지므로 이로 인해 새로 발생하는 오버헤드는 없으며 이동하는 PC의 증가 정도로 인식할 수 있으므로 서버 구조의 개선이나 이동 알고리즘의 개선으로 해결할 수 있다.

## 5. 결론

MMORPG는 대규모의 인원들이 동일한 가상공간에서 동일한 시각에 상호작용하면서 활동하는 온라인 롤플레이팅 게임이다. 대부분의 경우, 서버와 대역폭 등에서 상당한 수준의 하드웨어 사양과 지원 스템프를 필요로 한다. 수많은 개발자들의 노력에도 불구하고 사용자들은 여전히 플레이어의 과밀, 락, 그리고 부족한 지원 등에 대해 불만을 가지는 경우가 많다.

본 논문에서는 MMORPG에서의 부하 분산을 위한 동적 지역 분할 방법을 제안하였다. Field, sector group, sector, cell 등으로 구성되는 VML을 이용하여 지역을 분할함으로써 부하 분산을 시도하였다. PC들의 활동성을 나타내는 PA값과 특정 방향으로의 이동성을 나타내는 PG값의 변화에 따른 초과 부하의 변화를 측정하였다. PG의 값이 0.3일 경우 23~67% 정도의 성능향상을 보였다. 실험 결과 PA값보다는 PG값에 의한 변화가 더 큰 것을 확인할 수 있었으며 PG값에 변화를 주었을 때 VML 기반의 동적 지역분할이 더 큰 효과를 발휘하였다. 결론적으로 본 논문에서 제안한 VML 기반의 동적 지역분할 기법은 특정지역에 많은 PC들이 몰리는 상황에서 정적 지역분할 기법에 비교하여 더 큰 효과를 발휘하였다.

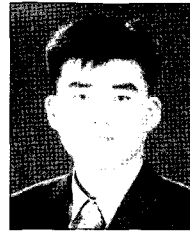
본 논문에서 제안한 VML 기반의 게임 시스템은 시스템의 유지 보수 측면에서도 장점을 가진다. 게임 중 유동적으로 변하는 PC들의 행동 패턴의 변화에 따른 인구 과밀 현상에 대해서도 능동적으로 대처할 수 있을 뿐만 아니라 수용할 수 있는 맵의 크기와 PC 수를 크게 늘릴 수 있다. 또한 게임 운영 도중 얻을 수 있는 시스템 분석 자료, 맵의 변경, PC들의 행동 패턴 변화 등의 자료를 바탕으로 지역 분할 방법을 변경하고자 할 때 VML의 계층 구조에 간단한 변화를 줌으로써 곧바로 적용가능하다. 그러나 이 방식의 경우 게임 설계단계에서 동적 지역분할을 위한 고려를 해야 하며 각 게임에 따라 동적 지역분할이 필요하지 않은 경우에는 적합하지 않다.

## 참고 문헌

- [1] World of Warcraft Community Site, <http://www.worldofwarcraft>.

com

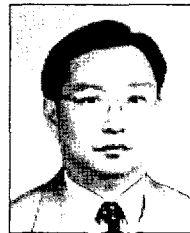
- [2] MMOG genre challenges, [http://en.wikipedia.org/wiki/MMOG\\_genre\\_challenges](http://en.wikipedia.org/wiki/MMOG_genre_challenges)
- [3] 이만재, “온라인 게임 엔진 기술 동향”, 정보과학회지, 제20권 제1호, pp.12-18, Jan. 2002.
- [4] 이남재, 박훈성 “진화하는 온라인 롤플레이밍 게임을 위한 분산형 게임 서버 모델” 한국게임학회 논문지 제2권 제1호, pp. 36-41, 05. 2002.
- [5] Tobin J. Lehman, James H. Kaufman, “OptimalGrid: middleware for automatic deployment of distributed FEM problems on an Internet-based computing grid,” Cluster Computing, Proceedings, 2003 IEEE International Conference on cluster computing, pp.164-171, Dec., 2003.
- [6] 최재연, 이해원, 하수철, “CORBA 기반 분산 네트워크 게임 서버에 관한 연구”, 한국정보처리학회 2001년 춘계학술대회, 제8권 제1호, pp.159-162, 04. 2001.
- [7] IBM Grid Research Lab, “Butterfly.net : Powering Next-Generation Gaming with Computing On-Demand,” <http://www.butterfly.net>, 2004
- [8] 이철민, 박홍성. “다중 사용자 게임 성능 향상을 위한 데이터 가상 그룹핑 방법”, 정보과학회논문지, 소프트웨어 및 응용 제 30권 제3호, pp.231-238, 04. 2003.
- [9] 이현진, “프로세스 기반 분산 게임 서버 아키텍처의 연구”, 인제대학교 석사학위논문, 02. 2004.
- [10] 문성원, “분산 seamless 게임 서버에서의 효율적인 게임 공간 관리 기술”, 서강대 정보통신대학원 학위 논문, 02. 2005.
- [11] 남재욱, “온라인 게임서버 프로그래밍”, 한빛미디어, 05. 2004.
- [12] 원동기 외 6명, “Seamless Games에서의 부하 분산을 위한 MigAgent 시스템 설계”, 정보 및 제어 심포지엄, pp.466-469, 10. 2005.



### 김 법 균

e-mail : kyun@chonbuk.ac.kr

1994년 전북대학교 컴퓨터공학과(공학사)  
 1997년 전북대학교 컴퓨터공학과(공학석사)  
 2005년 전북대학교 컴퓨터공학과(공학박사)  
 2005년~현재 전북대학교 공학연구원  
 관심분야 : 분산 및 병렬처리, Grid



### 안 동 언

e-mail : duan@moak.chonbuk.ac.kr

1981년 한양대학교 전자공학과(공학사)  
 1987년 KAIST 전산학과(공학석사)  
 1995년 KAIST 전산학과(공학박사)  
 1995년~현재 전북대학교 전자정보공학부  
 교수

2001년~2002년 전북대학교 정보검색시스템연구센터 센터장  
 관심분야 : 정보검색, 한국어정보처리, 문서분류, 문서요약, Grid



### 정 성 종

e-mail : sjchung@moak.chonbuk.ac.kr

1975년 한양대학교 전기공학과(공학사)  
 1981년 Houston대학교 전자공학과  
 (공학석사)  
 1988년 충남대학교 전산공학과(공학박사)  
 1985년~현재 전북대학교 전자정보공학부  
 교수

1996년~1998년 전북대학교 전자계산소 소장  
 2001년~2005년 전북대학교 BK21 전자정보사업단 단장  
 관심분야 : 정보검색, Grid