

MMORPG 서버를 위한 게임 사용자 중심의 부하 분산 기법에 관한 연구*

최종관^{*}, 김혜영^{*}, 우원식^{**}
홍익대학교 게임학부 게임 소프트웨어 전공^{*}, (주)엔씨소프트^{**}
kryy1987@nate.com, hykim@hongik.ac.kr, wswoo@ncsoft.com

A Study of a Game User Oriented Load Balancing Scheme on MMORPG

Jong-Gwan Choi^{*}, Hye-Young Kim^{*}, Won-Sik Woo^{**}
Major in Game Software, School of Games, Hongik University^{*}, NC Soft Corp.^{**}

요 약

동일한 가상공간에서 대규모 게임 사용자들 간의 상호작용이 활발히 일어나는 MMORPG(Massively Multi-player Online Role-Playing Games)에서는 많은 부하가 발생하며 이로 인하여 지연현상 및 자원부족 등의 문제를 발생시킨다. 이를 해결하기 위해 많은 개발자들이 분산서버 연구에 노력하고 있지만 주로 정적, 동적 맵 분할에 의한 분산 서버 연구개발에 편중되어있다. 본 논문은 MMORPG에서 서버의 부하 균등화를 위한 게임 사용자 중심의 분산 처리 기법을 제안하고, 4장에서의 구현을 통한 성능평가를 통해 본 논문에서의 제안 기법이 심리스(seamless) 서버 모델과 존 서버 모델에서 생기는 핫스팟(hot-spot) 및 특정 서버에 대한 게임 사용자 몰림 현상에 대해 효율적으로 대응할 수 있음을 보였다.

ABSTRACT

At MMORPGs that has the numerous game users with many interactions at the same virtual world, massive loading that results in delays, resource shortages, and other such problems occur. To solve this, many developers devote research to load-balancing servers, yet due to steady and dynamic map divisions, such research is unreliable. This paper proposes a game user-oriented load balancing scheme for the load balancing of MMORPG servers. This scheme shows effectiveness at dealing with hot-spots and other gatherings of game users at specific servers compared to previous methods.

Keywords : game user, MMORPG, seamless, map, load balancing, game user oriented

접수일자 : 2012년 05월 21일 심사완료 : 2012년 06월 12일

교신저자(Corresponding Author) : 김혜영

* 이 논문은 2011년도 9월~2012년도 2월 (주)엔씨소프트에 의해 지원되었음.

1. 서론

MMORPG(Massively Multiplayer Online Role-Playing Games)시장이 확대됨에 따라 게임 사용자 수가 증가함으로써 하나의 서버에서 처리할 수 있는 데이터는 한계가 있으므로 이를 해결하기 위한 방법으로 분산 서버 구조를 사용하게 되었다[1]. 일반적으로 MMORPG 서버의 로드는 게임 사용자의 수에 비례하여 증가하므로 분산 서버 구조는 효과적으로 게임 사용자를 각 게임 서버들에 분산하는 기법이라고 할 수 있다. MMORPG에서 가장 중요한 게임 사용자들 간의 상호작용은 게임 월드 맵 상에서 지형적으로 근접한 게임 사용자들끼리 가장 많이 일어나는데, 이때 근접 게임 사용자들이 각각 다른 서버에 속해 있다면 중간에 서버 간 통신이 한 번 더 일어나므로 하나의 서버에 근접 게임 사용자들이 속해 있을 때보다 큰 오버헤드가 발생하게 된다[2].

이와 같은 문제를 최소화시키기 위해서는 근접 게임 사용자들을 하나의 서버에서 처리하는 것이 효율적이므로, 분산 서버 구조 중에서 가장 활발히 연구개발 되고 있는 분야는 맵 분할을 통한 분산 서버 기법이다. 대표적으로 cell, sector 등으로 구성되는 영역 정보를 이용하여 맵을 분할하는 심리스 서버 모델과 월드를 각각 존(Zone)으로 분할하여 각각의 존들에 대한 처리를 서버에 분산하는 존 서버 모델을 들 수 있다[3]. 그러나 월드 맵 상에서 어느 지역에 게임 사용자가 편중 될지에 대한 예측이 힘들다는 점에서 맵 분할을 통한 부하 분산 서버 구조는 문제점을 가지고 있다[3].

따라서 본 논문에서는 맵 분할이 아닌 게임 사용자를 서버에 균등화하게 분산시킴으로써 부하를 분배하는 기법을 통하여 기존의 연구들과 비교 시 보다 유연하게 대응할 수 있도록 제안한다. 또한 게임 사용자들 간 상호작용을 보다 원활하게 하기 위해 맵 상에 근접 게임 사용자들을 최대한 하나의 서버에서 처리하도록 설계함으로써, MMORPG 게임 서버의 부하 분산을 위해 게임 사용자 중심

의 부하 분산 기법에 맵 분할 기법을 접목함으로써 게임 진행시의 부하 분산의 효율성을 높였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 기존의 부하 분산 기법들의 특징과 장단점을 알아보고, 3장에서는 MMORPG 게임 서버에서의 효율적인 부하 분산 기법을 제안한다. 4장에서는 제안 기법과 기존 연구들과의 성능평가를 통해 비교 분석하고, 5장에서는 결론 및 향후 연구방향을 보였다.

2. 관련연구

온라인 게임에서 부하를 분산하기 위한 가장 손쉬운 방법은 다수의 서버군 제공과 게임 채널 제공 방법이 있다. 위의 두 방법은 독립된 게임서버 그룹을 여러 개 설치함으로써 게임서버 수용능력을 손쉽게 높이고, 특정 공간에 게임 사용자 밀집 가능성을 낮출 수 있으며 확장성도 좋기 때문에 효율성 측면에서 쉽고 효과적인 방법이라 할 수 있다. 그러나 서버군 제공 방법은 각 서버들이 모두 별개의 공간임으로 다른 서버 군에 참여중인 게임 사용자들 간 상호작용이 불가능하고 함께 게임을 즐길 수 없다는 점에서 커뮤니티 활성화 실패의 문제를 발생시킨다. 게임 채널 제공 방법은 일시적으로 제공되는 게임 공간으로서 게임 사용자들이 같은 채널로 참여할 경우 원하는 게임 사용자들과 함께 게임을 즐길 수 있는 장점이 있다. 하지만 이와 같은 방법은 아케이드 게임, FPS(First-person shooter) 게임 등의 소규모 게임 사용자가 모여서 즐기는 방식에 적합하며 가상공간이 거대한 MMORPG에는 맞지 않는 방법이다[4].

MMORPG에서는 게임 사용자들 간의 상호작용이 게임의 재미 요소 중 가장 중요하므로 최대한 많은 게임 사용자들을 하나의 월드 맵 상에서 처리하기 위한 연구가 활발하다. 일반적으로 맵 분할을 통한 부하 분산 연구가 가장 활발하며 맵 분할을 하기 위한 방법에 따라 전체 시스템의 효율성이 좌우된다.

맵을 효율적으로 나누기 위한 조건들은 아래와 같이 5가지로 분류 할 수 있다[5,6].

첫째, 분할 된 면적들의 차이가 적도록 분할한다.

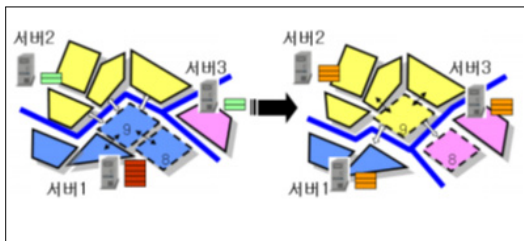
둘째, 각 서버에 할당 된 면적에 있는 오브젝트의 수가 균등하도록 분할한다.

셋째, 서버이전 부담률을 줄이기 위해 경계선 주변의 오브젝트 수가 적도록 분할한다.

넷째, 분할한 경계선이 짧을수록 서버 이전 부담률을 더 줄일 수 있다.

다섯, 오브젝트들의 주 이동경로와 경계선이 많이 겹칠수록 서버이전 부담률이 커짐으로 이동경로와 경계선의 만남이 최소화 되도록 한다.

위의 방법들은 효율적인 맵 분할을 위한 좋은 조건들이다. 그러나 게임이 초기에 서비스 될 때에는 전체 맵의 어느 지형에 게임 사용자의 몰림 현상이 일어날지, 어떠한 이동경로가 게임 사용자에게 가장 많이 사용될지에 대한 예측이 힘들어 위와 같은 조건들을 활용하여 맵을 분할하는데 한계가 있다. 따라서 이를 해결하기 위한 방법으로 동적 맵 분할 기법이 연구되었다[7,8].



[그림 1] 동적 맵 분할 기법

[그림 1]은 Piece Cell을 이용한 동적 맵 분할 기법에 관한 설명 그림이다. 기본적으로 다수의 게임 사용자가 이용할 큰 맵을 다수의 조각 셀(Piece Cell)로 나누고 각 서버의 처리 능력에 따라 여러 개의 Piece Cell을 할당하여 처리하는 것을 기본으로 한다. [그림 2]와 같이 서버 1에 부하가 생겼을 경우 서버 1이 할당받은 일부 Piece Cell을 주변의 인접 서버에게 이전 시켜 서버 부하를 균등화 시

키는 방법이다. 동적 맵 분할 서버 구조에서는 초기 맵 분할을 어떻게 하는지에 따라 Piece cell의 서버이전 발생하는 횟수가 달라진다. 그러나 정적 맵 분할과 마찬가지로 예측이 어려워 유연한 대처가 힘들다. 또한 특정 Piece cell에 게임 사용자가 밀집할 경우 이를 분산하기 힘들고 서버이전 시 Piece cell에 포함된 모든 오브젝트 정보가 한 번에 이전되므로 이때 발생하는 오버헤드가 크다[9].

따라서 본 논문에서는 게임 사용자를 중심으로 서버 분산을 함으로써 초기 맵 분할과 같은 작업이 없고, Piece cell의 데이터 서버이전 시 발생하는 오버헤드를 한 번에 발생시키지 않고 시간차를 두어 분산시키는 기법을 제안하고자 한다.

3. 사용자 중심의 분산 기법

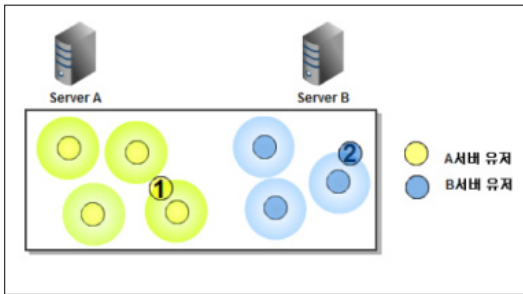
본 제안 기법에서는 MMOG의 게임서버 중 게임 사용자가 전혀 접속하지 않은 상태의 서버가 존재할 경우 라운드 로빈 방식으로 서버에 게임 사용자를 할당 하는 것을 기본으로 한다. 그러나 라운드 로빈 방식만을 사용할 경우 게임 사용자의 수는 균등하게 서버에 분산되나 무분별하게 분산됨으로써 서버 간 통신 증가로 인한 과부하를 발생시킬 수 있다[10]. 이와 같은 문제점을 해결하기 위해서는 최대한 서버 간 게임 사용자 균등화를 유지하면서도 근접 게임 사용자들을 한 서버에서 처리할 수 있도록 게임 사용자를 분산하는 기법이 필요하다. 여기서는 근접 게임 사용자들을 고려한 게임 사용자 분산 기법과 부하 분산 시에 임계영역 최소화를 유지하는 게임 사용자 서버이전 기법을 통해 게임 사용자 편중을 해결하고 서버 간 통신을 최소화 하는 방법을 제안한다.

3.1 인접 게임 사용자 분산 기법

MMORPG는 게임 사용자들 간 상호작용이 중요하며 상호작용은 보통 시야거리 안에 있는 게임 사용자들 간 발생한다. 그러므로 게임 사용자들이

서로 시야거리 안에 있을 경우 이 게임 사용자들은 최대한 하나의 서버에서 관리하는 것이 바람직하다. 위와 같이 게임 사용자들을 관리하기 위해 게임 사용자들의 상호작용 발생범위 및 발생 가능성 범위까지 감안하여 모든 게임 사용자들의 위치를 중심으로 시야거리의 약 1.5배에서 2배를 근접 게임 사용자 영역으로 설정한다. 새로운 게임 사용자가 접속 시 최우선적으로 기존 게임 사용자들의 인접 게임 사용자 영역 안에 포함되는지 확인하고 포함될 시 해당 인접 게임 사용자가 속한 서버에 새로운 게임 사용자를 분산한다.

[그림 2]에서는 인접 게임 사용자 분산 기법의 예를 보여준다. 서버 A와 서버 B는 각각 4, 3명의 게임 사용자를 수용한 상태이고 게임 사용자 1과 게임 사용자 2가 접속하였다. 로드 밸런싱 서버는 기존에 접속된 게임 사용자들과 현재 접속한 게임 사용자 1, 2의 위치정보 값을 비교하여 게임 사용자 1과 2를 각각 근접 게임 사용자가 있는 서버 A와 B로 할당한다.



[그림 2] 인접 게임 사용자 분산 기법

3.2 서버 중심 좌표를 통한 사용자 분산 기법

부하 분산을 위해서는 각각의 게임 서버마다 1개의 중점을 구한 값을 부하분산 서버에서 최신화하여 가지고 있으며, 이 중점 좌표 값은 접속한 게임 사용자 좌표 값들의 평균으로 구해진다. 부하분산 서버는, 이 값을 통해 새로운 게임 사용자가 접속했을 때 모든 게임서버의 게임 사용자들의 근접영역에 포함되지 않을 경우 서버의 중점과의 거

리가 가장 가까운 서버로 해당 게임 사용자를 분산한다. 또한 각기 다른 서버 간 게임 사용자들의 근접 영역 부분이 중첩되어 있고 그 중첩 부분에 새로운 게임 사용자가 접속했을 경우도 서버의 중심과의 거리가 가까운 서버로 할당한다.

```

 $S_k$  : 서버  $k$ 를 의미
 $T_k$  :  $k$ 서버에 접속한 게임 사용자 수
 $U_n$  : 게임 사용자  $n$ 을 의미
 $P_n$  : 게임 사용자  $n$ 의 위치 좌표
 $A_k$  : 서버  $k$ 에 접속한 게임 사용자들의 위치 좌표
CS: 접속 중인 게임 사용자들을 관리하는 서버의 갯수
PS: 접속 대상 서버

 $L(P_i, P_j)$  : 두 위치 좌표 간의 거리
 $F(U_n)$  : 게임 사용자  $n$ 과 인접한 게임 사용자가 속한 서버를 리턴, 인접 게임 사용자가 없을 시 none을 리턴
 $Update A_k(U_n) = A_k \leftarrow A_k + \frac{P_n}{T_k + 1}$ 
:  $k$ 서버의 중심 좌표 값을 게임 사용자  $n$ 의 위치 좌표 값을 더하여 최신화 시킴

// 새로운 접속 게임 사용자  $n$ 의 인접 게임 사용자
// 탐색 및 대상 서버 탐색
 $PS \leftarrow F(U_n)$ 
// 인접 게임 사용자가 없을 시
if ( $PS = none$ )
    min  $\leftarrow -\infty$ 
    for  $i = 1$  to CS
        if ( $L(P_n, A_i) < min$ )
            min  $\leftarrow L(P_n, A_i)$ 
             $PS \leftarrow S_i$ 
        end
    end
end
// 탐색한 접속 대상 서버로 게임 사용자  $n$ 을 할당
Allocate  $U_n$  to PS
// 접속 대상 서버의 위치 좌표 값 최신화
Update  $A_{PS}(U_n)$ 
    
```

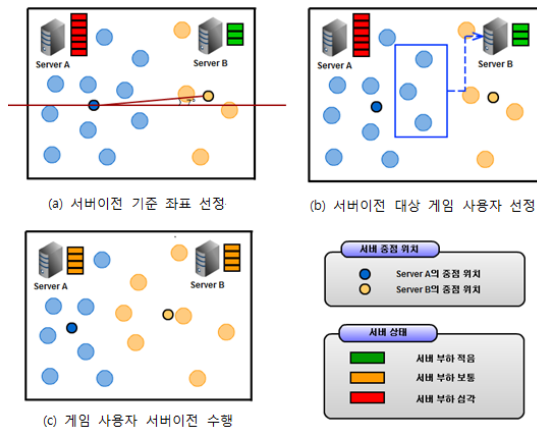
[그림 3] 사용자 중심의 부하 분산 제안 기법

3.3 부하 분산 시 서버의 중점들의 기울기를 통한 최적화

서버의 부하 분산은 서버들의 부하 측정을 통해 인접한 서버들 간의 부하가 특정 수치 이상으로 차이가 나는 경우 발생하며 이때 과부하 서버에서 타 인접 서버 중 부하가 제일 적은 서버로 게임

사용자를 서버이전 시키게 된다. 게임 사용자 서버이전 시 과부하 서버가 할당 받은 게임 사용자들에 최대한 멀리 있는 게임 사용자를 이전시키되 서버간 공유 임계 영역이 최대한 직선모양을 유지할 수 있도록 게임 사용자들을 할당 받아야 공유 임계영역에서 생기는 부하를 최소화 시킬 수 있다. K-mean 알고리즘을 통해 구한 중심 값들 간의 기울기가 가로축에 가까우면 X좌표를 기준으로, 세로축에 가까우면 Y좌표를 기준으로 이전 시킬 서버에 가장 근접해 있는 게임 사용자를 최우선으로 하여 서버이전을 수행한다.

[그림 3]과 [그림 4]에서 본 논문의 제안기법에 대한 알고리즘 및 부하 분산 과정을 보여 주고 있다.

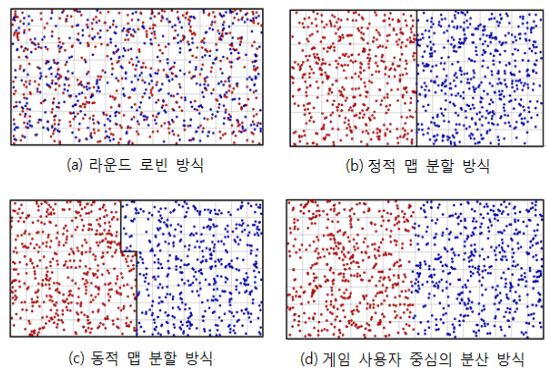


[그림 4] 제안기법의 부하 분산 수행과정

4. 구현 및 성능평가

본 논문에서는 무작위의 위치 좌표 x, y 를 갖는 가상의 게임 사용자 접속을 지속적으로 발생시켜 정적 맵 분할 기법, 동적 맵 분할 기법, 본 논문에서 제안하는 게임 사용자 중심의 부하 분산 기법에 따른 성능을 비교하였다. 성능평가는 2개의 서버가 있다는 가정을 하고 가상의 게임 사용자 증가 시뮬레이션을 통해 진행된다. 총 월드 맵의 크

기는 1024×562 로 가정하며 맵 분할 기법 사용 시 각 서버는 기본적으로 562×562 의 맵 영역을 할당 받는다. 동적 맵 분할 시 월드 맵의 piece cell은 4×8 로 가정하며 두 서버 간 부하 측정량의 차가 하나의 piece cell에 있는 게임 사용자 수 평균의 2배 이상 발생 했을 경우 1개의 piece cell을 부하가 적은 서버로 이전한다. 게임 사용자 중심의 부하 분산 기법의 경우 부하 측정량의 차가 특정 수치 이상으로 발생 시 제안기법에 따라 서버 균등화가 될 때 까지 게임 사용자를 부하가 적은 서버로 이전한다. 본 논문에서는 서버 간 게임 사용자 수의 차가 6이상 발생 했을 시 게임 사용자 서버이전을 수행하였다.

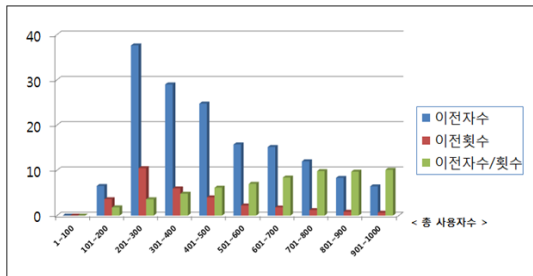


[그림 5] 서버 분산 방식에 따른 유저 분포 비교

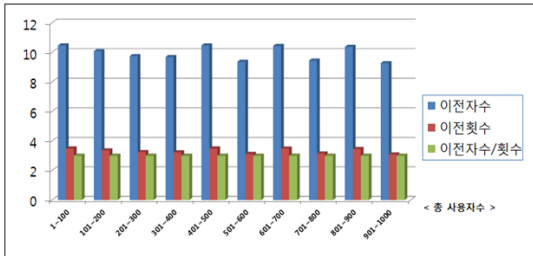
4.1 분산 방식에 따른 게임 사용자 증가 시 서버이전 비교

[그림 5]에서는 1024×512 크기의 맵에 게임 사용자들이 고른 분포로 접속 했을 때 서버 분산 방식에 따라 발생하는 서버 이전이 일어나는 것을 분포도로 표현하였으며, [그림 6]에서는 1024×512 크기의 맵에 게임 사용자들이 고른 분포로 접속 했을 때 동적 맵 분할 방식과 제안하는 게임 사용자 중심의 분산서버에서 발생하는 서버이전을 각각 그래프로 나타낸 것이다. 각각의 막대그래프는 총 게임 사용자가 증가함에 따른 서버이전 게임 사용자 수, 이전 횟수, 서버이전 발생 시 평균 이전자수를

보여준다. 맵 분할 방식의 그래프 [그림 6](a)를 보면 총 게임 사용자수가 200~300일 때 서버이전이 가장 활발하게 일어나며 그 이후로 총 게임 사용자수가 증가함에 따라 맵 분할이 점차 안정을 찾으며 이전횟수와 이전자수 모두 서서히 감소하는 것을 보여준다. 그러나 총 게임 사용자 수가 증가하면 한 Cell에 있는 게임 사용자수도 증가하므로 한 번의 서버이전 시 발생하는 이전자수도 증가함으로써 순간적으로 큰 오버헤드를 발생시킬 위험부담은 커지게 된다.



(a) Cell 방식의 동적 맵분할 서버에서 게임 사용자의 서버이전



(b) 게임 사용자 중심의 분산 서버에서 게임 사용자의 서버이전

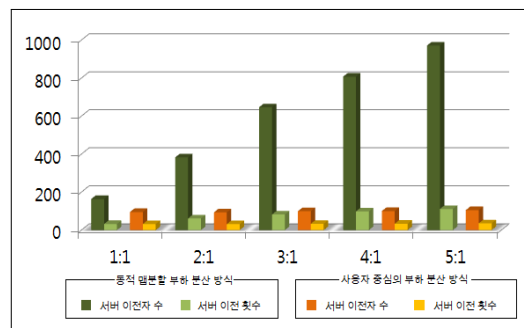
[그림 6] 분산 방식에 따른 게임 사용자 증가 시 서버이전 비교

게임 사용자 중심의 분산 서버는 그래프 [그림 6](b)에서 보듯이 총 게임 사용자 수에 따라 큰 변동 없이 고른 서버 이전자수와 이전 횟수를 가진다. 또한 총 게임 사용자의 수가 1000명까지 접속하는 동안 맵 분할은 평균 156명, 게임 사용자 중심의 분산 서버는 99명의 이전게임 사용자가 발생했다. 이는 게임 사용자 중심으로 서버이전을 수행하기 때문에 총 게임 사용자 증가에 따른 하나의 piece cell에 속하는 게임 사용자의 밀도 증가와 관

계없이 고른 서버이전 분포를 나타낸다. 이를 통해 본 논문에서 제안하는 게임 사용자 중심의 분산 서버가 서버이전 시 발생하는 서버부하를 게임 사용자 증가 시에도 고르게 분산시킴으로써 비교적 안정적이면서도 효율적으로 서버부하를 균등화시킬 수 있다.

4.2 게임 사용자 편중에 따른 서버이전 비교

MMORPG에서는 게임 사용자들의 선호하는 장소나 게임 맵 상의 지리적 문제로 인하여 게임 사용자들의 위치 분포가 고르게 나오지 않으며 특정 지역에 밀집되는 현상도 빈번히 발생한다. [그림 7]은 1024*512 크기 맵의 위치 좌표가 (0, 0) ~ (512, 512) 부분과 그 외 부분의 게임 사용자의 접속 빈도수 비율을 1:1, 2:1, 3:1, 4:1, 5:1 로 증가시키면서, 이때 발생하는 서버이전 상황을 그래프로 표현하였다. [그림 7]은 동적 맵 분할 부하 분산 방식의 경우 특정 영역에 게임 사용자들의 몰림 현상이 커질수록 서버 이전자수와 이전 횟수가 비례하여 증가함을 보여준다. 반면에 게임 사용자 중심의 부하 분산 방식의 경우 영역들 간 게임 사용자 빈도수의 차가 커짐에도 불구하고 서버 이전자수와 이전 횟수가 큰 차이를 보이지 않았다. 따라서 게임 사용자 중심의 부하 분산 방식이 동적 맵 분할 부하 분산 방식보다 유동적인 게임 사용자 위치 분포에도 보다 효율적으로 서버이전을 수행하고 있음을 보여준다.



[그림 7] 게임 사용자 편중에 따른 서버이전 비교

5. 결론 및 향후 과제

본 논문에서는 기존의 MMORPG에서 사용되던 맵 분할 방식의 부하 분산 방식이 아닌 게임 사용자 중심의 부하 분산 방식을 사용하여 보다 효율적이고 게임 사용자 분포 변화에 안정적으로 대응할 수 있는 분산처리 기법을 제안하였다. 또한 접속 시 접속 게임 사용자 주위의 게임 사용자들을 검색하여 인접게임 사용자들을 한 서버에서 처리하도록 분산하고 하고, 서버 간 중점의 기술기를 이용하여 서버이전을 시행함으로써 공유 임계영역과 서버이전 시 발생하는 부하를 최소화하는 기법을 제안하였다. 분산 방식에 따른 게임 사용자 증가 시 서버이전은 총 게임 사용자의 수가 1000명까지 접속하는 동안 맵 분할은 평균 156명, 게임 사용자 중심의 분산 서버는 99명의 이전게임 사용자가 발생함으로써 약 40%정도의 효율성을 보였으며, 게임 사용자 편중에 따른 서버이전에서도 동적 맵 분할 부하 분산 방식의 경우 특정 영역에 게임 사용자들의 몰림 현상이 커질수록 서버 이전자수와 이전 횟수가 비례하여 증가하는 반면에 게임 사용자 중심의 부하 분산 방식의 경우 영역들 간 게임 사용자 빈도수의 차가 커짐에도 불구하고 서버 이전자수와 이전 횟수가 큰 차이를 보이지 않음으로써 본 제안 기법의 효율성을 보였다.

향후 연구로 동적으로 움직이는 게임 사용자들을 관리하는 기법과 많은 수의 서버를 위의 분산 처리 기법으로 적용 할 때 필요한 최적화 알고리즘 및 서버 추가 연동 시 최적화 방법을 연구하고자 한다.

games”, *Multimed Tolls Appl*, 45:263-289, 2009.

- [3] 임정열, 박일규, 정재용, 심광현, “분산 게임서버 기술 동향”, *전자통신동향분석*, 제20권, 제4호, pp. 93-102, 2005. 8.
- [4] 김혜영, 임영중, “실시간 전략 시뮬레이션 게임에서의 효율적인 동기화 기법”, *한국게임학회논문집*, 제10권, 3호, pp.83~92, 2010. 6
- [5] 장수민, 유재수, “MMOG 서버 부하의 균등화를 위한 효과적인 분할”, 제 27회 한국정보처리학회 춘계학술발표대회 논문집, 제14권, 제1호, pp. 16-19, 2007. 5.
- [6] 장수민, 유재수, “MMORPG 서버의 부하균등화를 위한 효율적인 분산처리 기법”, *한국콘텐츠학회 논문집*, Vol.7, No.11, pp. 69-75, 2007.
- [7] 이정진, 두길수, 안동연, 정성중, “MMORPG 부하 분산을 위한 동적 맵 분할 시스템 설계”, *한국컴퓨터종합학술대회*, Vol.32, No.1, pp. 802-804, 2005.
- [8] Dong Jun Kwak, “Probability Map Partitioning for Multi-player Pursuit-Evasion-Evasion Game”, *Control Automation and Systems(ICCAS)*, 2010 International Conference, pp. 294-298, 2010. 10. 27-30.
- [9] Santosh Kulkarni, “Badumna Network Suite: A Decentralized Network Engine for Massively Multiplayer Online Applications”, *Peer-to-Peer Computing*, 2009. P2P '09. IEEE Ninth International Conference, pp. 178-183, 2009. 9. 9-11.
- [10] 김범균, 안동연, 정성중, “MMORPG에서의 부하 분산을 위한 가상 영역 정보 기반 동적 지역 분할”, *정보처리학회 논문집*, 제13-A권, 제3호, pp. 223-230, 2006. 6

참고문헌

- [1] 이기훈, “게임 산업의 전망 및 최근 동향”, *동향*, 제23권, 22호, 통권 521호, pp. 74-89, 2001. 12. 1.
- [2] Carlos Eduardo Benevides Bezerra, Claudio Fernando Resin Geyer, “A load balancing scheme for massively multiplayer online



최 종 관 (Choi, Jong Jwan)

2008년 3월 2일-현재 홍익대학교 게임학부
게임소프트웨어전공 학부생

관심분야 : 게임서버, 물리엔진, 게임제작



김 혜 영 (Hye-Young Kim)

2005년 2월 고려대학교 컴퓨터학과 이학박사
2005년 3월-2006년 8월 Wright State Uni.Post-Doc.
2007년 3월-현재 홍익대학교 게임학부
게임소프트웨어전공 조교수

관심분야 : 모바일 게임, 온라인 게임서버, 게임엔진



우 원 식 (Won-Sik Woo)

2002년-현재 엔씨소프트 CTO 전무
1991년-1993년 서울대학교 제어계측 석사
1987년-1991년 서울대학교 제어계측 학사
1998년-2002년 나모인터랙티브 이사 (전무/상무)
1993년-1998년 국방과학연구소
1989년-1993년 한글과 컴퓨터 이사 (전무/상무)

관심분야 : 게임엔진설계, MMOPRG, 설계, 게임제작
