

# MMOG 서버 부하의 균등화를 위한 효과적인 분할

장수민\*, 유재수\*\*  
충북대학교 정보통신공학과  
e-mail: jsm@netdb.chungbuk.ac.kr\*, yjs@chungbuk.ac.kr\*\*

## An Efficient Partitioning Scheme for Load Balancing of MMOG Servers

Su-Min Jang, Jae-Soo Yoo  
Department of Computer and Communication Engineering, Chungbuk  
National University

### 요 약

MMOG 게임은 온라인 서비스들 중에 중요한 부분을 차지하고 있다. 최근에는 네트워크를 통한 온라인 서비스를 이용하는 사용자들의 증가로 인해 서버에 부하가 가중되고 있다. 본 논문에서는 이와 같은 문제를 해결하기 위하여 효과적인 분할방법을 통하여 서버의 부하를 낮추는 방법을 제안한다. 제안하는 방식은 많은 사용자들을 위한 분산 게임서버에 효과적인 해결책을 제공한다.

### 1. 서론

전 세계적으로 MMOG(Massively Multiplayer Online Game) 게임의 시장규모는 이미 PC게임, 비디오게임을 능가하고 있으며 이러한 온라인 게임의 성장 추세는 지속될 전망이다 [4, 5, 6]. MMOG 게임의 성공요인을 몇 가지로 분석할 수 있다. 첫째로 기존의 Standalone형 게임에서 사용자는 기획된 게임 패턴에 쉽게 적응하게 되는 반면, MMOG 게임은 다수의 참여자들 간의 발생하는 다양성 (variety) 때문에 무한대의 게임 패턴을 갖게 된다. 둘째로 온라인을 통한 자동 업데이트가 지속적으로 이루어지므로 MMOG 게임은 지속적인 확장성 및 유연성을 갖게 된다. 셋째로 이용자와 개발자간의 다양한 경로를 통한 의견교환을 통한 개발자의 의도가 아닌 실 이용자가 원하는 맞춤형 게임으로 만들기 적합하다. 그러나 최근 MMOG 게임은 이전 보다 넓은 가상의 다양한 공간에서, 보다 많은 사용자를 처리해야 하는 게임으로 발전 되어 가면서 게임 서버에 대한 부하를 가중시키고 있다. 이러한 문제점을 해결하기 위하여 이전의 방법에서는 가상공간의 크기를 일정한 크기로 분할하여 분산서버를 구축하는 방법을 사용하였다. 그러나 이러한 분할은 여러 가지 문제점을 가지고 있다. 본 논문에서는 보다 효율적인 분할 방식을 통하여 서버의 부하를 낮추는 효과적인 분할 방식을 제안한다. 게임서버의 분할할 때 고려해야 할 여러 가지 주요조건들을 제시하고 최적화하는 방법을 제안한다. 또한 본

논문에서 제시하는 분할방식은 다중 프로세스 방식에 적합하도록 설계하였다.

본 논문의 구성은 먼저 2장에서 관련연구에 대해서 알아보고, 3장에서는 제안하는 효율적인 MMOG게임 서버 분할을 제시한다. 4장에서는 성능평가 및 분석하고 마지막으로 5장에서는 결론을 맺고 향후연구방향을 제시한다.

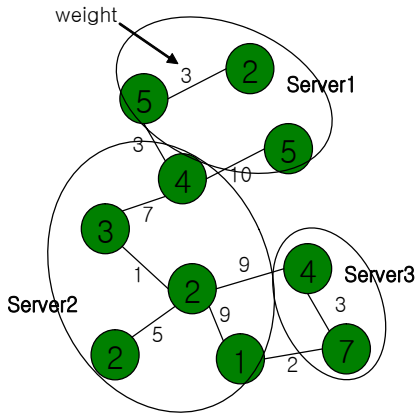
### 2. 관련연구

MMOG는 일반적으로 게임공간의 크기에 제한을 받지 않으면서 게임에서 발생하는 데이터와 메시지 트래픽을 효율적으로 처리하는 구조인 맵기반 분산 서버 구조 (map-based distributed server architecture)를 사용한다. 이 방식은 게임공간을 처리 가능한 지역단위로 분할하고 각 지역당 담당 서버를 두어 가상공간의 확장성을 높여주는 기법이다. 게임공간을 분할하는 방식에 따라 전체시스템의 효율성이 좌우된다고 할 수 있다. 공간다중분할방식 (multi-resolution spatial partitioning)은 온라인게임의 공간을 top-down방향으로 세분화하여 관리하는 방법[1]이다. 전체 게임공간은 상호작용이 불가능하고 서로 독립적인 world의 모임으로 구성된다. 하나의 world는 여러 개의 zone으로 분할되고 각각의 zone은 담당서버에 의하여 관리된다. zone과 zone간의 경계는 시스템만이 알고 있는 정보로 참여자는 의식하지 못한다. zone은 또다시 최소공간단위인 cell로 나누어지며 참여자가 속한 cell과 주변 cell의 집합으로 관심영역(area of interest : AOD)이 정의된다.

게임공간에서 참여자들은 전체 공간에 균일하게 분포하는 경우보다는 특정지역에 편중되어 분포하는 경우가

\* 본 연구는 한국과학재단 특정기초연구(R01-2006-000-10809-0)와 산업자원부 지역혁신인력양성사업의 연구결과물입니다.

더 일반적이다. 맵기반 분산서버방식에서 그림 4와 같이 참여자의 밀집도가 높은 지역을 담당하는 서버에게는 메시지 트래픽 처리 부하가 증가되며, 상대적으로 밀집도가 낮은 지역의 서버는 처리할 데이터가 적은 부하 불균형 (workload unbalance)의 문제가 발생한다. 이러한 분산 서버의 처리부하를 균등화(load balancing)하여 시스템을 안정화하는 문제는 최근 온라인게임분야의 주요 이슈가 되고 있다 [2]. 맵기반 분산서버방식에서 서버 부하의 불균형은 불가피하며 효과적으로 균등화하기 위하여 새로운 기준을 적용하여야 한다. 분산서버간의 부하균등화 문제는 최적의 분할을 탐색하기까지 너무 많은 경우의 수가 발생하는 NP-complete problem로 알려져 있다 [3]. Lui와 Chen이 제안한 Linear Optimization Technique(LOT)기법 [4]은 그래프 이론(graph theory)을 서버 부하 균등화 문제에 적용한 것으로 가장 효율적인 방법으로 평가되고 있다[3]. 그림 1과 같이 LOT기법은 참여자들 간에는 노드를 edge로 연결하고 떨어진 거리를 edge의 weight로 표시하여 weight의 비중을 두고 클러스터링 한다. 그러나 MMOG의 참여자들 간에 weight는 매우 균등하여 LOT기법을 MMOG에 적용하기에는 적합하지 않다.



(그림 1) LOT 방법에 의한 가상공간의 분할

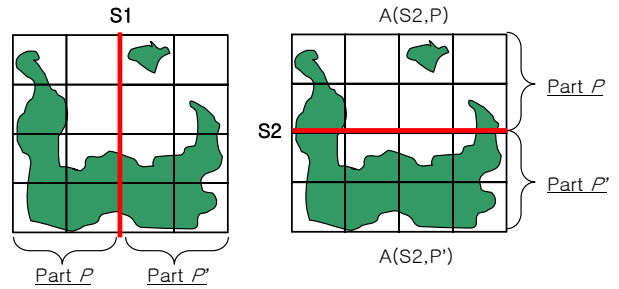
### 3. 제안하는 효과적인 분할 방식

MMOG 한 개의 서버가 아주 잘 구성되었다하여도 사용자의 엄청난 증가와 가상의 공간이 커지게 되면 부하가 가중되어 원활한 서비스를 할 수 없다. 그래서 다양한 방법을 사용하여 이러한 서버의 부하를 load balancing 하고자 하는 연구가 있다 [3, 6, 7, 8, 9, 10]. 본 논문에서는 분산 MMOG을 위한 효과적인 분할하기 위하여 다섯 가지의 주요조건을 제시한다. 여러 개의 서버로 분할할 때 고려해야 할 조건들로 지도의 면적, 오브젝트 수, 경계선 부분의 오브젝트 수, 경계선의 길이, 오브젝트의 이동경로이다.

#### 3.1 가상공간의 면적

그림 2에서 첫 번째 분할 S1과 S2를 보면 각각 세로

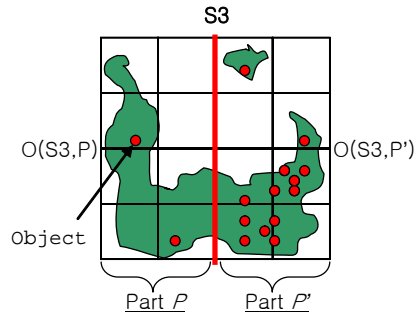
와 가로로 분할하였는데 분할로 인하여 나누어진 셀에 해당하는 지도의 면적이 서로 다르다. 지도의 면적은 오브젝트들의 이동할 수 있는 공간이 되기 때문에 지도의 면적의 크기가 고려되어야 한다. 분할할 경우 지도의 면적  $A(S, P)$ 와  $A(S, P')$ 의 차이 값이 적도록 분할해야 한다.



(그림 2) 지도의 면적크기를 고려한 분할

#### 3.2 오브젝트 수

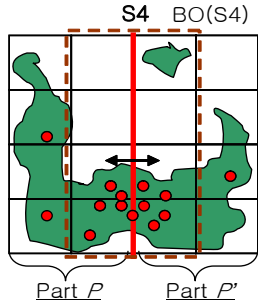
S3에서처럼 분할된 지도에  $O(S3, P)$ 에는 2개의 오브젝트가 있고  $O(S3, P')$ 에는 13개 있는 형태로 분할이 되었다. 분할할 경우 오브젝트수가 균등하게 나누어지도록 분할하여야 한다. 오브젝트의 수는 분할시점에 오브젝트수를 고려할 수도 있지만 일정한 시간간격으로 데이터를 수집하여 표본 데이터를 산출하여 오브젝트수를 고려한다.



(그림 3) 오브젝트 수에 따른 분할

#### 3.3 경계선 부분의 오브젝트 수

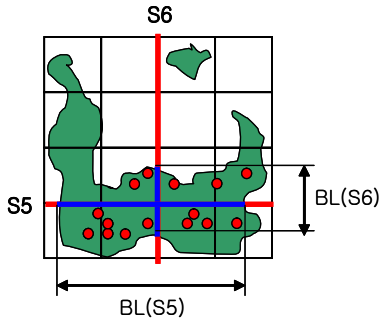
그림4은 오브젝트 수는 거의 균등하게 잘 분할하였다 그러나 오브젝트들이 분할된 경계선에 몰려있기 때문에 서버에서 서버로 이전할 확률이 높다. 이처럼 서버이전 부담률이 높지 않는 곳을 분할하여야 한다. 서버이전 부담률은 경계선 주변 셀의 오브젝트수인  $BO(S, P)$  값이 적어야 한다. S4로 나누어진  $BO(S4, P)$ 는 11개 이다.



(그림 4) 경계선 부분의 오브젝트 수를 고려한 분할

3.4 경계선 길이

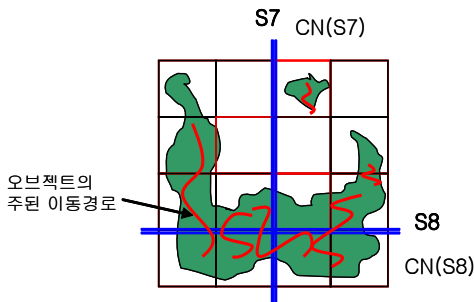
분할된 경계선의 길이는 서버이전 확률과 밀접한 관계를 가지고 있다. 경계선이 길수록 서버이전 부담률은 높아진다. 분할 S5의 BL(S5)과 분할 S6의 BL(S6)를 비교하면 BL(S6)가 경계선의 길이가 짧다. 분할되어지는 경계선은 짧을수록 서버이전 부담률이 적어진다.



(그림 5) 경계선 길이를 고려한 분할

3.5 오브젝트의 이동경로

오브젝트들이 다양한 패턴으로 이동하게 되지만 주된 이동경로가 있다. 영역을 분할할 때 주된 이동경로를 분할하게 되면 서버의 이전부담률이 높아진다. 그래서 주된 이동경로가 아닌 곳을 분할한다. 오브젝트의 이동경로 또한 일정한 시간간격으로 수집한 데이터에서 표본 데이터를 산출하여 적용한다. S8처럼 분할하면 주된 경로 4개가 끊어지는 것으로 오브젝트가 서버에서 서버로 이전할 부담이 커진다. 그래서 S8보다는 S7로 분할하는 것이 서버 이전부담률을 적게 한다.



(그림 6) 오브젝트의 이동경로를 고려한 분할

3.6 가상공간의 최적화된 분할

서버의 부하를 최적의 균등화하기 위한 수식은 아래와 같이 OPT(S)의 값을 구하여 OPT(S)값이 적은 분할을 선택한다. 각각 조건들의 중요도는 상수를 통하여 정한다.

$$OPT(S) = i*|O(S,P)-O(S,P')| + j*|A(S,P)-A(S,P')| + k*BO(S) + l*BL(S) + m*CN(S)$$

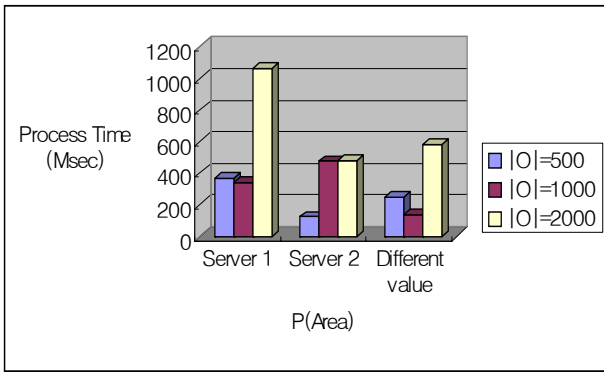
4. 성능평가 및 분석

성능평가는 2개의 서버를 가지고 진행하였다. 성능평가의 조건으로는 오브젝트의 수를 500에서 2000까지 변화를 주면서 면적만을 고려한 분할에서 각각의 서버의 처리속도와 다섯 가지의 조건을 모두 고려한 각각의 서버의 처리속도를 측정하였다. 서버 1과 서버2의 처리시간의 차이 값을 적을수록 서버의 부하를 균등하게 분할한 것이다. 표. 1은 성능평가를 위한 조건들을 나열한 것이다.

<표.1>성능평가를 위한 조건들

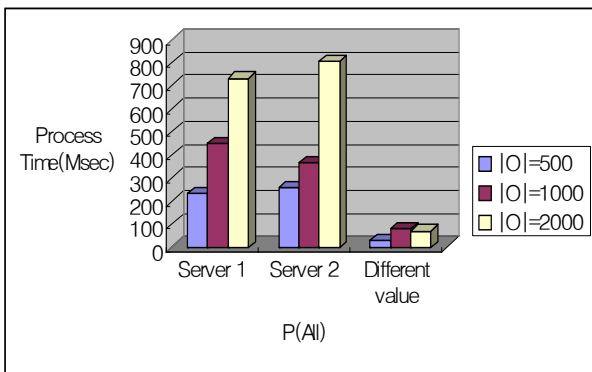
Player	player은 임의의 방향으로 이동한다. player(50%)는 max movement는 2이고 나머지는 5이다. 같은 셀에 monster가 있다면 player는 monster를 공격한다. player의 업데이트정보는 player의 자신의 셀과 주변 셀 8개를 포함하여 9개 셀의 정보를 보낸다.
Monster	max movement 1 이다; player가 공격하면 랜덤한 방향으로 이동한다.
Packet	max size 는 1024byte이다
NPC	랜덤한 위치에 세팅되고 움직이지 않는다.
O	오브젝트의 전체수는 players(70%), monsters(20%), NPCs(10%)으로 구성한다.
VS	가상의 게임공간은 30%의 빈 공간으로 갖는다.
P(Area), P(All)	P(Area)는 면적만을 고려한 분할이고 P(All)는 다섯 가지를 모두 고려한 분할이다.

그림 7은 가장 많이 분할할 때 사용하는 면적을 기준으로 분할한 경우에 서버 두 개의 프로세싱 시간을 구하여 각각의 서버의 프로세싱 시간의 차이 값을 보여주고 있다. 면적 조건만을 고려한 서버의 분할은 서버의 부하를 아주 균등한 분할하지 못하고 있다.



(그림 7) 가상공간의 면적만을 고려한 분할에 대한 처리시간 측정

그림 8은 본 논문에서 제시한 다섯 가지의 조건을 모두 고려한 서버의 분할은 두 개의 서버의 프로세싱 시간의 차이 값이 매우 적은 결과를 보여주고 있다. 차이 값이 적다는 것은 서버의 부하를 아주 균등하게 분할된 것을 나타낸다. 이처럼 본 논문에서 제시한 지도의 면적, 오브젝트 수, 경계선 부분의 오브젝트 수, 경계선의 길이, 오브젝트의 이동경로를 모두 고려한 분할이 서버의 부하를 균등하게 분할한다는 것을 증명한다.



(그림 8) 모든 조건을 고려한 분할에 대한 처리시간 측정

### 5. 결론

네트워크를 통한 온라인 서비스를 이용하는 사용자들의 증가로 인해 서버에 부하가 가중되고 있다. 이러한 문제를 해결하기 위하여 MMOG 서버의 부하를 여러 개의 서버를 이용하여 서버의 부하를 균등화하는 효과적인 분할이 필수적이다. 본 논문은 서버의 부하를 균등화하는 효과적인 분할방법으로 다섯 가지의 주요 조건을 제안하였다. 그 조건으로는 지도의 면적, 오브젝트 수, 경계선 부분의 오브젝트 수, 경계선의 길이, 오브젝트의 이동경로를 제안하고 이를 성능평가를 통하여 효율적인 분할임을 보였다. 향후연구방향으로는 실 게임데이터를 사용한 성능평가와 다양한 조건을 통한 성능평가를 하고자 한다.

### 참고문헌

[1] B. Ng, A. Si, R. W.H. Lau, F. W.B. Li , A Multi-Server Architecture for Distributed Virtual Walkthrough, Proceedings of the ACM symposium on Virtual Reality Software and Technology , November 2002, pp.163-170.

[2] C.S. John, M. Lui, F. Chan, An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems, IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 3, March 2002, pp.193-211.

[3] P. Morillo, J. M. Orduna, M. Fernandez, A Comparison Study of Evolutive Algorithms for Solving the Partitioning Problem in Distributed Virtual Environment Systems, Parallel Computing Vol. 30, 2004, pp. 585-610.

[4] J.C.S. Lui, M.F. Chan, K.Y. Oldfield, Dynamic Partitioning for a Distributed Virtual Environment, Department of Computer Science, Chinese University of Hong Kong, 1998.

[4] Ahmed Abdelkhalek, Angelos Bilas: Parallelization and performance of interactive multiplayer game servers. April IEEE (2004) 26-30

[5] A. Abdelkhalek, A. Bilas, and A. Moshovos. Behavior and performance of interactive multi-player game servers. Nov. 4-6, ISPASS01(2001)137 - 146

[6] Lui, J.C.S.; Chan, M.F. : An efficient partitioning algorithm for distributed virtual environment systems Parallel and Distributed Systems. IEEE Transactions on Volume 13,3,(2002)193 - 211

[7] John C.S Lui, Member, IEEE and M.F. Chan, Student Member, IEEE A :An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems.IEEE Transactions on parallel and distributed systems Volume 13, Issue 3, March( 2002)193 - 211

[8] D. Min, E. Choi, Donghoon Lee and B. Park :A Load Balancing Algorithm for a Distributed Multimedia Game Server Architecture. Proc. of the IEEE International Conference on Multimedia Computing and Systems, June (1999) 882-886

[9] N. Baughman,B. Levine, :Cheat-proof palyout for centralized and distributed online games. In Proc. Infocom (2001)104-113

[10] Merabti. M, El Rhalibi. A,: Peer-to-peer architecture and protocol for a massively multiplayer online game", 29 Nov.-3 Dec GlobeCom Workshops (2004)519 - 528