

# 온라인 게임서버에서의 k-최근접 관심영역 관리기법

박일규, 심광현, 김종성  
한국 전자통신 연구원  
{xiao, shimkh, joskim}@etri.re.kr

## K-Nearest Interest Management in Online Game Server

Park Il Kyu, Kwang-Hyun Shim, Jong-Sung Kim  
Electronics and Telecommunications Research Institute

### 요 약

대규모 온라인 게임과 같이 수많은 사용자를 수용하는 클라이언트/서버 방식 응용에서는 네트워크의 대역폭을 효율적으로 사용하는 것이 중요하다. 각 클라이언트의 관심영역에 해당하는 데이터만을 보내어 트래픽을 줄이는 방법을 관심영역 관리라 하며, 클래스기반, 영역 기반, 격자기반 등 여러 가지의 방법이 제안되어 있다. 본 논문에서는 작은 영역에서 거리 기반으로 관심영역을 정하는 관심영역 관리 기법을 제안하고, 이를 이용하여 참가자가 편중된 영역에서 생기는 트래픽 집중 현상을 해소하는 방법을 제안한다.

### 1. 개요

대규모 온라인 게임과 같은 응용에서는 수천, 수만의 참가자가 가상 공간 안에서 상호작용을 한다. 이들의 하나하나의 행동은 메시지의 형태로 네트워크를 통해 전달된다. 이 메시지들을 모든 참가자들에게 보낼 경우에는 사용자의 수  $n$ 의 제곱에 비례하여 메시지가 증가하므로 사용자의 수에 따라 급격히 트래픽 양이 늘어나게 된다. 이 트래픽 양을 줄이기 위해서 각 참가자는 자신의 관심영역(Area of Interest)을 표시하고, 관심영역 관리자(Interest Manager)는 이에 따라 각 참가자에게 필요한 메시지만을 보내어 트래픽 양을 줄인다.

관심영역 관리 방법은 관심영역을 정의하는 방법에 따라 클래스 기반, 지역 기반, 격자 기반 방식으로 나뉜다[1,2]. 클래스 기반 방식에서는 업데이트 되는 데이터의 종류(클래스)에 따라 관심영역을 정하며, 지역 기반과 격자 기반의 방식에서는 참가자의 위치에 의하여 업데이트를 받는 영역이 정해지게 된다. 지역 기반의 관심영역 방식에서는 관심영역을 가입 영역과 갱신 영역으로 표시하고, 데이터를 보내는 참가자의 갱신 영역이 받을 참가자의 가입영역과 겹칠 때 데이터를 보내는 방식이다. 격자 기반의 방식에서는 지역 기반의 방식처럼 갱신 영역과 가입 영역을 사용하지만, 이

영역은 연속적으로 정할 수 있는 영역이 아니라 격자에 의해 미리 나뉘어진 영역 중에서 선택하는 방식이다.

이러한 관심영역 방법들은 중요하지 않은 데이터의 전송을 막아 네트워크의 트래픽을 낮출 수 있다. 그러나, 가상공간상에 참가자가 균등하게 분포하지 않고 좁은 공간에 몰려 있을 경우에는 지역 기반 또는 격자 기반 방식을 사용하더라도 전송되는 트래픽의 양이 많아지게 된다. 또한 참가자가 좁은 공간에 몰리는 경우는 빈번히 일어나기 때문에[1] 이러한 경우에 트래픽을 제한하는 방법이 필요하다. 본 논문에서는 격자 기반의 관심영역 관리 기법을 사용하는 환경에서, 참가자가 편중된 셀(cell)에서의 메시지 양을 제안할 수 있는 k-최근접 관심영역 관리 기법을 제안한다.

### 2. 격자 기반 관심영역 관리

격자 기반 관심영역 관리는 가상 공간을 정해진 셀들로 나누고, 참가자들이 속한 셀을 기준으로 메시지를 필터링하는 관심영역 관리 방법이다. 메시지가 발생하면 서버는 메시지를 받아 그 메시지가 발생한 셀과 근접한 셀들에 속한 참가자들에게 메시지를 포워딩한다. 근접한 셀에 속하지 않는 참가자들은 메시지를 받지 않으므로, 참가자의 위치가 균등하게 분포하는 경우에는 전체적인

네트워크 트래픽과 각 참가자에 대한 트래픽 모두 낮게 유지할 수 있다. 그러나, 온라인 게임의 경우 다양한 이벤트나 게임의 진행 상황에 의해 참가자가 좁은 영역에 모이는 일이 빈번하게 발생하고, 이에 따라 같은 셀이나 인접한 셀에 속한 사용자가 증가하여 트래픽 양이 증가하게 된다. 또한, 온라인 게임의 참가자는 대역폭의 크기가 낮은 경우가 대부분이므로 이렇게 사용자가 증가한 경우에 메시지의 전송에 장애를 가져올 수 있다.

**3. k-최근접 관심영역 관리 기법**

격자 방식의 관심영역 관리 기법에서 셀 안의 메시지는 같은 셀 안의 모든 참가자에게 전달되므로, 메시지 전송량은 셀 안에 속한 참가자의 수 n의 제곱에 비례한다. 따라서 한 셀에 많은 사용자가 편중되면 메시지의 전송량이 빠른 속도로 증가하게 된다. 본 논문에서는 격자 방식의 관심영역 관리기법에서 한 셀에 사용자가 편중되는 경우의 네트워크 부하를 줄이기 위해, 셀 내에서 참가자들의 가상공간 상 거리를 기준으로 관심영역을 제한하는 k-최근접 관심영역 관리 기법을 제안한다. 제안하는 방법은 셀 단위에서는 격자 방식의 관심영역 관리 기법을 적용하며, 참가자가 편중된 셀에서는 거리 기반으로 관심영역을 제한함으로써 네트워크의 부하를 줄이고자 한다.

셀 내에서 k-최근접 관심영역 관리 기법은 각 메시지가 발생할 때에 메시지를 받는 참가자를 메시지가 발생한 지점에서 가장 가까운 k명으로 정의한다. 셀 안에 속한 참가자의 수가 n이면, 셀 내의 메시지 전송량은 n\*k가 된다. k는 n과 독립적이므로, 메시지의 전송량은 선형적으로 증가하게 된다.

참가자의 위치 변화에 따른 계산 복잡도를 낮추기 위해서 k-최근접 관심영역 관리 기법에서는 다음과 같이 거리의 근사치를 사용한다. 참가자의 밀도가 높은 셀을 m\*m개의 정사각형 모양 bucket으로 나누고, 참가자간의 거리의 근사치로 bucket간의 거리를 사용하여 관심영역을 관리한다. m값은 셀의 참가자 밀도가 높을 때 예상되는 참가자 수에 비해 m\*m이 충분히 크도록 정하여 거리의 오차로 생기는 영향이 작게 한다.

각 bucket에는 현재 그 bucket에 속한 사용자의 목록은 연결 리스트 형태로 저장한다. 서버는 사용자의 이동 메시지를 받을 때마다 사용자의 위치를 갱신하는 동시에 사용자가 이전에 속했던 bucket에서 사용자를 빼고 새로운 bucket에 사용자를 추가하여 사용자의 목록을 유지한다. Bucket은 정사각형 모양이므로, 참가자의 위치에 해당하는 bucket을 찾는 작업은 좌표마다 나누기 연산을 수행하는 것으로 구할 수 있다. 거리에 기반하여 관심영역을 관리하기 위하여 각 bucket간 거리를 계산하여 거리 순으로 bucket을 정렬한다. 정사각형의 bucket의 거리 순서는 bucket의 크기에 관계가 없으므로 이 계산은 서버가 서비스를 수행하기 전에 다음과 같이 계산하여 사용한다.

거리는 bucket의 한 변의 길이를 한 단위로 하여 계산한다. 기준 bucket의 좌표를 (0,0)이라 하고, x 좌표의 범위를 [-m, m], y좌표의 범위를 [-m, m] 으로 하여 각 bucket과 기준 bucket사이의 거리  $\sqrt{x^2 + y^2}$  을 계산하면, 2m\*2m개의 bucket에 대해 거리의 list가 만들어진다. 이를 다시 거리를 기준으로 정렬하면, 표 1 과 같이 거리의 순서대로 bucket의 좌표 리스트가 만들어지게 된다.

거리	0	1	1	1	1	$\sqrt{2}$	...
좌표	0,0	1,0	0,1	-1,0	0,-1	1,1	...

표 1 거리순서 bucket 좌표 리스트

이 자료구조를 이용하여 실제 메시지의 전송이 일어날 때는 다음과 같이 데이터를 보낼 참가자를 선택할 수 있다. 거리 목록에서 하나씩 좌표를 가져오고, 현재의 좌표에 이 좌표를 더하여 해당 bucket을 얻는다. 이 bucket에 있는 참가자들을 갱신 목록에 추가하고, 참가자의 총 합이 k를 초과하였는지 검사한다. 갱신 목록의 참가자의 수가 k를 초과하지 않았으면 거리 list에서 다음 좌표를 가져와 참가자를 추가하는 작업을 반복한다. 갱신 목록에 속한 참가자의 수가 k를 초과하였으면 그 갱신 목록이 현재의 갱신 목록이 된다. 좌표의 리스트는 거리순으로 정렬되어 있으므로, 얻어진 갱신 목록이 근접한 k명의 목록이 된다. 그림 1에 전체적인 동작이 도시되어 있다.

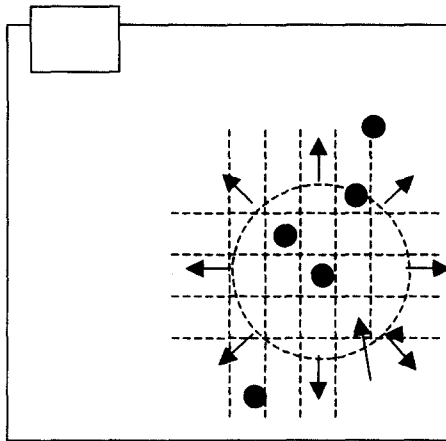


그림 1 k-최근접 알고리즘의 동작

#### 4. 메모리 및 CPU 오버헤드

본 논문에서 제안한 기법은 참가자가 편중된 셀마다 많은 메모리를 할당하나, 메시지를 전송하고 관심영역을 관리하는데 필요한 CPU비용은 높지 않다. 제안한 기법은 참가자의 밀도가 높은 셀에  $m \times m$ 개의 bucket을 만들고 각 bucket에는 해당 bucket에 속하는 참가자의 연결 리스트를 유지한다. Bucket마다 bucket에 속하는 참가자의 연결 리스트를 가지고 있으므로, 연결 리스트의 선두 포인터를 저장할 메모리가 요구된다. 포인터의 크기를  $p$ 라 하면, 이 비용은  $pm^2$ 이 된다. 연결 리스트를 만들기 위해서는 참가자마다  $id$ 와 포인터가 한 개씩 필요하므로, 셀 내의 참가자 수를  $n$ ,  $id$ 의 저장에 필요한 메모리 크기를  $q$ 라 하면, 이에 필요한 메모리는  $n(p+q)$ 이다. Bucket은 참가자를 구분할 수 있도록 작게 설정하므로,  $m^2 \gg n$ 가 되고, 따라서 총 메모리 요구량은  $pm^2 + c(p+q) \approx pm^2$  만큼 증가한다.

Bucket간 거리는 offline으로 계산하므로 실행시에는 CPU비용이 들지 않는다. 메시지를 보낼 때에는 거리-좌표 리스트를 참조하고, 각 bucket에 저장된 연결 리스트를 참조하므로 CPU의 오버헤드는 최악의 경우  $m^2 + c$  번의 메모리 참조비용이 된다.

Bucket을 갱신하는 오버헤드는 참가자가 이동할

때마다 발생한다. 참가자의 이동 메시지를 받으면 서버는 새로운 좌표에서 새로운 bucket을 얻고, 기존의 bucket에 저장된 연결 리스트에서 참가자를 삭제하고 새로운 bucket으로 옮기는 연산이 요구된다. 연결 리스트에서 원소를 추가하고 삭제하는 작업은  $O(1)$ 의 시간에 수행되므로 다른 작업에 비해 무시할 수 있다.

#### 5. 결론

본 논문에서는 대규모 온라인 게임에서 혼잡한 영역의 네트워크 메시지를 줄이기 위한 기법을 제안하였다. 제안한 방법은 격자기반 관심영역 관리 기법을 사용하며, 참가자가 편중된 셀에 거리 기반의 기법을 사용하였다. 제안한 기법에서는 bucket을 사용하여 거리를 미리 계산하고, 메시지를 전송할 때는 메모리 참조만을 사용하여 근접한  $k$ 명의 참가자의 목록을 얻는다. 근접한  $k$ 명에게만 메시지를 보내므로 네트워크의 트래픽이 사용자 수의 제곱에 비례하여 증가하는 것을 막고  $k$ 로 정해지는 일정한 부하 이하로 유지할 수 있다. 제안한 방법은 혼잡한 셀마다 많은 양의 메모리를 필요로 하지만, 필요한 계산을 미리 수행하여 실행시의 CPU비용은 낮다.

#### 참고문헌

- [1] Katherine L. Morse, "Interest Management in Large-Scale Distributed Simulations," UC Irvine, Information and Computer Science Technical Report, ICS-TR-96-27.
- [2] Gary Tan et al., "Grid-Based Data Management in Distributed Simulation," Proceedings of the 33rd Annual Simulation Symposium, 2000
- [3] Sandeep K. Singhal et al., "InVerse: Designing an Interactive Universe Architecture for Scalability and Extensibility," Proceedings of the 6th International Symposium on High Performance Distributed Computing, August 1997
- [4] Daniel J. Van Hook et al., "Approaches to Relevance Filtering," 11th DIS Workshop, September 1994, pp. 367-369