이동객체에 대한 위치정보서비스를 위한 효율적인 분산처리기법

장수민°, 복경수*, 유재수**
충북대학교 정보통신공학과°,**, 한국과학기술원 전산학과*
jsm°@netdb.chungbuk.ac.kr, ksbok*@dbserver.kaist.ac.kr, yjs**@chungbuk.ac.kr

An Efficient Distributed Processing Scheme for Location Based Service on Moving Objects

Sumin Jang°, Kyoungsoo Bok*, Jaesoo Yoo**
Dept of Computer and Communication Engineering, ChungBuk National University°,**
Department of Computer Science, Korea Advanced Institute of Science and Technology*

요 약

최근에 이동객체에 대한 위치기반서비스는 서비스영역이 확대되고, 이를 사용하는 사용자가 늘어남으로써 원활한 서비스를 위하여 시스템의 분산처리가 요구되고 있다. 기존의 방식은 분산처리를 위하여 서비스영역을 분할하는 방식으로 전체 서비스의 영역을 보다 적은 영역으로 분할하여 여러 개의 서버가 각각의 영역을 담당하는 형태로 처리한다. 그러나 이러한 기존의 방식에서는 사용자의 핫스팟, 폭주로 인한 특정서버에 과부하가 발생되는 일어나는 문제점을 갖고 있다. 본 논문에서는 이러한 문제점을 해결하기 위하여 또한 특정서버에서 과부하 발생하면 동시성제어를 필요로 하지 않는 모듈만을 비 과부하 서버에 이전하는 기법을 통하여 효율적으로 과부하를 분산하여 처리한다. 또한 제안하는 기법은 각각 다른 서버에 의해 제어되는 구역에 있는 사용자들 간에 실시간 서비스를 제공한다. 제안된 기법은 다양한 성능평가를 통하여 제안한 기법의 우수성을 보인다.

1. 서 론

위치기반서비스는 휴대폰 속에 기지국이나 위성항법장치(GPS)와 연결되는 칩을 부착해 위치추적 서비스, 공공안전 서비스, 위치기반정보 서비스 등 위치와 관련된 각종 정보를 제공하는 서비스로 최근에 이슈화되고 있다. 위치기반서비스로 사람·차량 등의 위치를 파악하고 추적할 수 있음은 물론, 산속이나 사막과 같은 오지에서 위험에 처했을 때 휴대폰의 응급 버튼을 누르면 구조기관에 연결되어 구조를 받을 수 있다. 또 휴대폰 사용자가 있는 특정 장소의 날씨 서비스, 일정한 지역의 가입자에 대한 일괄 경보 통지 서비스, 지름길을 찾을 수 있는 교통정보 서비스, 주변의 백화점·의료기관·극장·음식점 등 생활정보 서비스, 이동 중에 정보가 제공되는 텔레매틱스 서비스 등 각종 서비스가 가능하다.

이러한 이동객체에 대한 위치기반서비스는 이전 보다 넓은 서비스 영역에서, 보다 많은 사용자를 처리해야 하는 형태로 발전되어 가면서 위치기반서비스를 위한 서버에 대한 부하가 가중되고 있다[1]. 이와 같이, 수천 명이 동시 접속 가능한 위치기반서비스에서의 서버의 처리는 모든 연산이 여러 개의 프로세스나 여러 개의 컴퓨터로 나누어 처리하는 분산 환경이 필요하며, 효율적인 처리를 위해서는 인접한 지역의 정보를 효과적으로 처리하는 방법이 필요하다.

본 연구는 다양한 형태의 위치기반서비스가 있지만 이동하는 객체가 무선단말기를 이용하여 서버로부터 서비스를 받는 구조이고, 객체는 자기 자신을 중심으로 근거리의 다른 객체에 대한 위치정보 및 객체의 정보를 요구하는 형태로 제한하다. 그림 1과 같이 정적인 객체와 동적인 객체가 있으며, 객체들은 주변의 객체에 대한 정보를 통하여 택시를 호출하거나 주변의 음식점이나 호텔의 정보를 이용하는 형태로 다양한 서비스로 응용되는 형태이다.

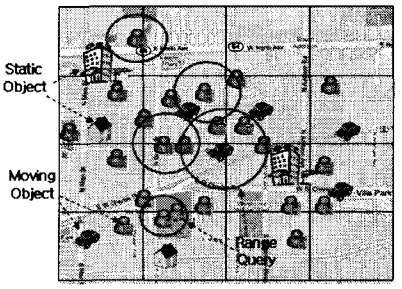


그림 1. 위치기반서비스의 예제

본 연구에서는 이동객체에 대한 위치기반서비스의 분

* 본 연구는 2006년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2006-000-1080900)

산처리를 위한 기존의 방식이 갖고 있는 문제점을 제시하고 이러한 문제점을 해결하는 효과적인 분산처리 기법을 제안한다. 위치기반서비스의 효율적인 분산처리를 위하여, 서비스 영역의 분할을 최적화하기 위한 여러 가지 요소를 제안한다. 이러한 위치기반서비스의 부하균등화를 위한 효율적인 분산처리 기법은 보다 다양한 위치기반서비스의 개발을 지원한다.

본 논문의 구성은 2장에서 관련연구에 대하여 기술하고, 3장에서는 제안하는 효율적인 분산처리 기법을 제안한다. 4장에서는 제안하는 방법의 우수성을 입증하기 위해 성능평가와 분석결과를 기술한다. 마지막으로 5장에서는 본 논문의 결론과 향후연구방향을 제시한다.

2. 관련 연구

대부분의 위치기반서비스는 이동객체와 서버간의 유선·무선통신을 통하여 처리한다. 위치기반서비스를 이용하는 객체들은 질의의 형태로는 범위 질의(Range Query), 최 근접 질의(Nearest Neighbor Query) 스카이라인 질의(Skyline Query) 등 다양하다. 또한 이러한 질의를 연속적으로 질의의 결과를 요구하는 연속질의형태로 실시간 처리를 요구한다. 이러한 서비스에 대한 확대로 많은 객체들을 처리하기 위한 연구로 가상공간에서 병렬처리하기 위한 클러스터링 연구[2][3]도 진행되었다. 그림 2는 일반적으로 사용하는 위치기반서비스의 구성도이다. 그림 2(a)와 같은 형태는 P2P방식[4][5]를 사용한 형태이다. 이러한 구성도의 서버는 객체들의 정보관리와 객체들 간의 연결정보를 제공하는 형태로, 모든 서비스가 이동객체들 간의 통신을 통하여 협력을 통하여 결과를 만들고 서비스한다.

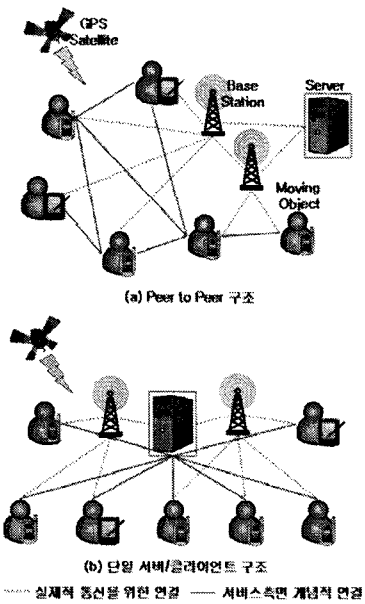


그림 2. 보편적인 위치기반서비스의 구조

또한 그림 2(b)와 같은 형태는 단일서버구조로 여러 개의 객체들이 하나의 서버에 연결되어 있는 형태이다. 단일 서버 구조는 하나의 서버에서 모든 객체들의 서비스를 제공한다. 객체들의 정보관리 및 사용자의 요청을 받고 처리하는 형태로 서버의 역할이 매우 중요하다. 단점으로는 서버가 갖고 있는 자원의 한계로 인하여 동시에 접속하여 처리할 수 있는 객체의 수가 제한적이다. 그래서 소수의 클라이언트가 접속하는 규모가 작은 서비스에 하는데 주로 사용한다. 그러나 이 구조의 장점은 하나의 서버에서 모든 서비스를 처리함으로써 여러 개의 서버로 분산하여 처리하는 분산서버구조에서 발생되는 서버간의 시간동기화, 교착상태, 동시성, 무결성 등의 문제가 발생하지 않는다.

그러나 위치기반서비스를 요청하는 객체들의 수가 크게 증가되면 단일 서버의 형태로는 원활한 서비스를 할 수 없다[6-9]. 이러한 경우 여러 개의 서버를 이용한 분산기법을 통하여 대수의 객체들을 위한 서비스가 가능하다. 이러한 분산기법으로 그림 3과 같은 대칭형 서버구조(Replicated Server Architecture or Mirrored Server Architecture)와 비대칭형 서버구조(Non-Replicated Server Architecture)를 이용한 서비스가 최근에 이루어지고 있다.

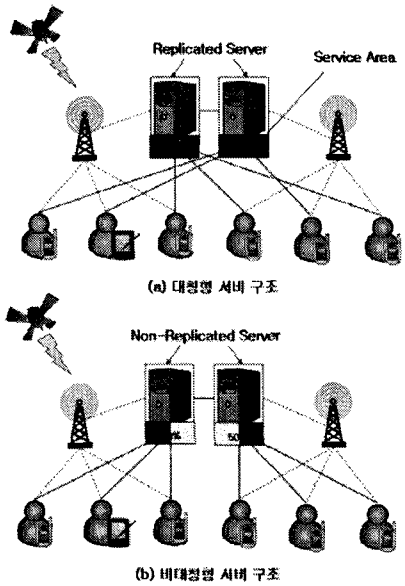


그림 3. 위치기반서비스를 위한 분산구조

대칭형 서버구조는 단일서버에서 필요한 자원을 그대로 복사하는 방식을 통하여 여러 대의 서버를 구축하는 형태이다. 이러한 구조는 객체의 위치에 상관없이 임의의 서버에 접속하여 서비스를 받을 수 있다. 그래서 각 서버마다 적절한 객체의 수를 할당하는 형태로 매우 균등하게 분할하는 장점이 있으나 복재된 서버들 간의 동시성제어 비용이 매우 크다는 문제점을 갖고 있다[6]. 그래서 서비스 영역을 분할하는 방식으로 비대칭형 서버구조를 더 많이 사용하고 있다. 이 방식은 처리할 서비

스영역을 분할하여 그 영역을 담당하는 서버를 미리 선정하여 서비스한다. 서비스영역의 확장성을 매우 높은 특징과 서비스영역의 효율적인 분할을 필요로 하는 기법이다. 또한 각 서비스 영역에 따른 분할된 서버는 독립적으로 서비스가 가능하며 이러한 서버들 간의 동시성제어 비용이 매우 적다는 큰 장점을 갖고 있다. 그러나 이러한 비대칭형 구조의 단점으로 객체들이 특정영역으로의 편중될 경우 편중된 영역을 담당하는 서버는 과부하가 발생하는 문제점이 있다.

3. 제안하는 효율적인 분산처리 기법

본 논문에서 제안하는 기법은 분산서버들 간의 갱신된 정보를 항상 유지해야 하는 비용이 많이 필요로 하는 대칭형 서버구조가 아닌 비대칭형 서버구조를 기본구조로 한다. 비대칭형 서버구조의 단점인 객체들의 편중으로 발생되는 특정영역의 과부하를 효과적으로 분산하기 위한 새로운 방법을 제안한다.

3.1 서버의 부하 분석

그림 4는 위치기반서비스의 구성모듈을 보여준다. 이처럼 객체의 현 위치정보데이터와 객체가 요구하는 질의를 받아주는 패킷수신처리기(Packet Receiver), 이러한 데이터를 받아 처리하는 패킷처리기(Packet Processor)와 그 결과를 객체에게 보내는 패킷송신처리기(Packet Sender)로 구성되어 있다. 이중에 패킷수신처리기와 패킷송신처리기는 서버의 전체처리비용중 많은 부분을 차지하는 부분으로 이를 분산처리하면 과부하를 분산시킬 수 있다. 또한 이 두 개의 모듈은 동시성제어와는 무관한 모듈로 따로 분리하여 처리할 수 있다. 또한 객체에서 중계기로 전달하는 네트워크 속도가 가장 늦고, 중계기에서 서버로 보내는 네트워크는 속도는 빠른 편이다. 또한 서버와 서버간의 네트워크는 고속망을 구축하여 사용하고 있다.

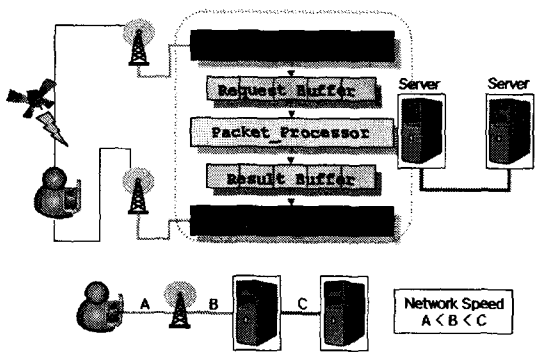


그림 4. 위치기반서비스의 구성모듈

3.2 효율적인 서비스영역의 분할

비대칭형 서버구조는 일반적인 효율적인 서비스영역의 분할이 필요한데, 분산서버방식에서는 서비스영역의 크기나 객체의 수를 고려한 분할 방식을 사용한다. 그러나 [10]에서는 서비스영역의 크기, 객체의 수뿐만 아니라 객체의 서버이전 부담률을 고려하여 네 가지의 요소를 고려한 방식으로 효율적인 서비스영역을 분할한다. 이러한 분할 방식은 서비스영역의 크기나 객체의 수를 고려한 분할 방식보다 우수한 성능을 보여주었다. 그래서 본 논문은 이 방식을 통하여 서비스영역을 분할한다.

3.3 객체의 편중에 따른 과부하

비대칭형 서버구조는 효율적인 서비스영역의 분할로 분산서버들 간의 부하가 균등화되는 효과뿐만 각 서버마다 독립적으로 서비스할 수 있는 장점을 갖고 있다. 그러나 객체들이 서버들 간에 이동할 수 있는 구조이기 때문에 특정서버에 객체들이 편중되는 경우가 발생된다. 이러한 편중현상은 일시적인 현상이지만 서버의 과부하로 인한 서비스 장애가 생긴다. 그림 5는 객체들의 편중에 따른 과부하를 보여준다. 비편중 상태일 경우에는 각각의 서버가 과부하가 발생되지 않고 원활한 서비스를 한다. 그러나 이처럼 서버 C의 서비스영역으로 객체들이 편중될 때 서버 C는 과부하가 발생된다.

만약 이러한 편중에 따른 과부하를 각 서버의 서비스영역을 자동적으로 변경하여 처리하고자 한다면 각 서버가 갖고 있는 정보데이터를 이전해야 하는 비용이 필요하고 각 서버가 담당하는 서비스영역의 변화를 통보해야 한다. 이러한 동기화는 실시간 서비스를 요구하는 위치기반서비스에 적합하지 않다.

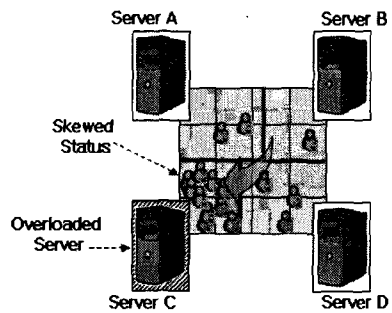


그림 5. 편중 상태에 따른 서비스의 과부하

3.4 객체의 편중에 따른 과부하 해결방법

객체의 편중으로 발생하는 과부하의 원인을 분석하면 일단 처리해야 할 객체의 수가 늘어난 것이 직접적인 원인이다. 대칭형 서버(Replicated Server)를 추가하는 방식은 각 객체의 위치 정보와 같은 실시간 갱신데이터를 반영해야하는 매우 높은 동시성제어비용을 요구하게 된다. 제안한 기법은 이를 해결하기 위하여 과부하가 된 서버 C에서 처리해야 하는 객체의 패킷수신처리기와 패킷송신처리기 부분을 과부하가 없는 서버 A, B로 이전하는 방법으로 과부하를 분산시킨다. 이 두 가지의 처리부

분은 서버부하 중 네트워크에 해당하는 부분으로, 분산처리하면 서버의 부하가 분산에 효과가 크다. 그림 6과 같이 서버 C에 접속되어 있는 객체들의 중계기를 통한 데이터전송이 하나로 묶어 마치 서버 A나 서버 B가 하나의 객체에서 데이터를 요구하는 것처럼 처리한다. 이러한 처리방식은 분산서버구조의 서버들 간에 발생되는 서버간의 시간동기화, 동시성, 무결성 등의 문제가 발생하지 않는 장점을 갖으면서 부하를 분산하기 때문에 매우 효율적이다.

위치기반서비스의 특징으로 객체들은 중계기를 통하여 서버에 데이터를 전송하기 때문에 중계기 단위의 일처리를 고려하여야 한다. 그래서 제안하는 기법에서는 과부하가 발생된 서비스영역에 해당하는 중계기에 세션을 갖고 있는 객체들을 비 과부하의 서버에서 담당하는 형태로 부하를 분산하여 처리한다. 이러한 방식으로 처리하게 되면 과부하의 서버는 객체의 패킷수신처리기와 패킷송신처리기부분이 분산되지만 질의의 결과를 산출하기 위한 정보는 분할된 서비스영역 그대로 유지함으로써 일관성 있는 질의처리를 할 수 있다. 그래서 편중에 따른 객체의 수가 증가한 영역을 담당하는 서버는 다소 부하가 있다. 그러나 객체의 정보를 이전하거나 복제할 경우 발생되는 동기화 비용이 없기 때문에 서비스영역을 확대하거나 사용자가 증가하여도 서비스를 위한 확장성이 매우 효과적인 방법이다.

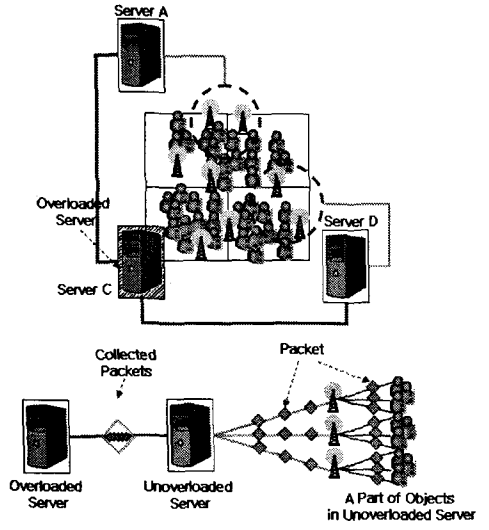


그림 6. 과부하 해결을 위한 처리절차

4. 성능평가

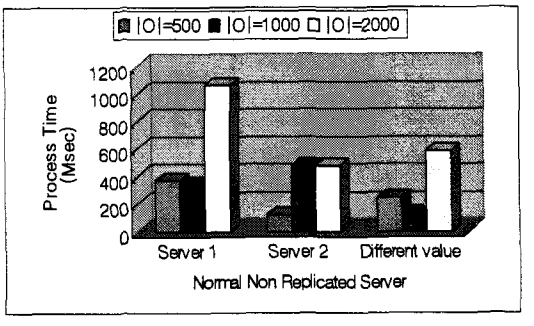
성능평가는 가상의 객체들을 생성하여 로컬 네트워크를 통한 통신을 기반으로 하면서 동시에 접속하여 서버에 패킷을 요청하는 형식으로 수행하였다. 성능평가의 조건으로는 객체의 수를 500에서 2000까지 변화를 주면서 한쪽 특정서버로 객체가 점차적으로 편중되는 비율을 높이는 과정을 통하여 성능평가 하였다. 표 1은 성능평가를 위한 상세한 조건들이다. 실험환경은 펜티엄 IV 2.0GHz에 1024Mbyte의 메모리를 가진 컴퓨터에 LINUX 환경에서 측정하였다.

<표 1> 성능평가를 위한 상세한 조건

Object	객체(Object)는 임의의 방향으로 이동한다. 다양한 이동패턴을 구사하기 위하여 객체(50%)는 최대 이동크기는 2이고 나머지(50%)는 5이다. 특정시점에 편중이 되도록 한다.
Packet, VS	패킷의 최대 크기는 1024byte, 서비스영역의 크기(VS)는 그리드형태로 1024*1024의 크기로 설정한다. 중계기는 균등하게 서비스영역에 분포시켰다.
|O|	객체의 수로 500 ~ 2000개 까지 동시에 접속
Query	객체가 요구하는 질의는 객체의 위치를 중심으로 정사각형 형태의 범위질의를 사용한다. 범위질의의 크기는 5로 한다.
S%	특정 서버로 객체의 편중의 비율로 전체 객체 중 S%가 편중되도록 설정한다.

4.1 객체의 수 증가가 미치는 영향

객체의 수가 증가함에 따라 제안하는 방식과 일반적인 비대칭구조의 처리시간을 비교한다. 객체의 수는 500개, 1000개, 2000개로 설정하고 편중비율 S%는 20%로 설정하였다. 그림 7과 같이 일반적인 비대칭 서버구조의 경우를 보면 객체의 수가 증가할수록 서버들이 처리시간은 증가하였고, 두 개의 서버들의 처리시간이 많이 차이나는 것을 보여준다. 편중된 서버에 있는 객체에 대한 질의처리가 원활하지 않다는 것을 알 수 있다. 그러나 본 논문에서 제안하는 효율적인 분산처리 방식을 적용한 경우에는 부하가 객체의 수가 증가할수록 서버의 전체 처리시간이 증가하지만 각 서버의 처리시간을 보면 거의 차이가 없음을 보여준다. 이는 서버의 부하가 매우 균등화된 것을 보여준다.



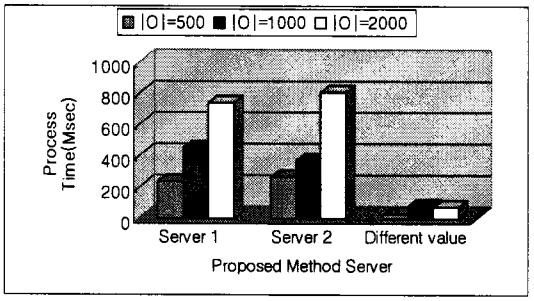


그림 7. 객체의 수 변화에 대한 처리시간

4.2 편중비율의 증가가 미치는 영향

효율적인 분할방식을 통하여 분할된 영역을 각 서버가 맡아서 처리하여도 일시적인 편중으로 객체가 특정한 지역으로 모일 경우 그 서버는 과부하가 생긴다. 그림 8은 이러한 객체들의 편중비율변화에 따른 처리시간을 비교하였다. 편중비율 S%는 0%에서 60%까지 변화하도록 설정하였다. 이때 서비스영역의 크기 VS는 1024*1024로 설정하고 객체의 수|O|는 2000개로 설정하였다. 제안하는 방식은 객체의 편중이 늘어남에 따라 서버 2의 처리시간이 증가하는 형태이지만 서버들 간의 처리속도를 비교해보면 높은 비율의 편중에도 매우 균등하게 분산된 것을 보여준다.

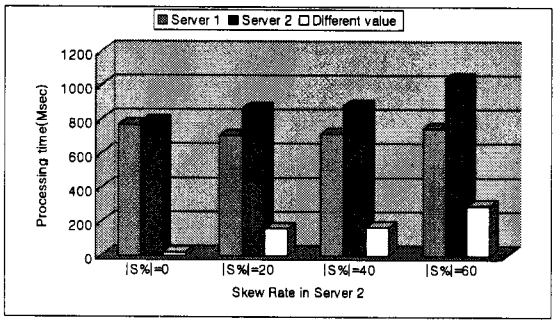


그림 8. 편중비율의 변화에 대한 처리시간

5. 결론

본 논문에서는 최근에 이슈화 되고 있는 위치기반서비스를 위한 기존서버의 구성에 따른 문제점을 제시하고 이를 해결하는 기법으로 효율적인 분산처리기법을 제안하였다. 기존의 분산처리 기법들은 사용자의 증가, 핫스팟, 폭주 그리고 일반적으로 발생하는 서버오류와 같은 문제들을 갖고 있다. 이러한 문제점을 해결하기 위해 비대칭형 서버구조를 기반으로 하면서 비대칭형 서버의 단점을 보완하기 위해 효율적인 패킷처리 기법을 제안하였다. 다양한 성능평가는 제안한 기법의 우수성을 보여주었다. 향후연구방향은 실 서비스에 적용하여 보다 세밀한 성능평가와 보다 향상된 서버를 구현하고자 한다.

참고 문헌

[1] Colbert, M. A diary study of rendezvousing: implications for position-aware computing and communications for the general public, in Proceedings of Supporting Group Work, (2001) 15-23.

[2] J. Xu, B. Zheng, W.-C. Lee, and D. L. Lee. Energy efficient index for querying location-dependent data in mobile broadcast environments. In Proceedings of the 19th IEEE International Conference on Data Engineering (ICDE'03), Bangalore, India, March 2003.

[3] E.Lety, T. Turletti, and F.Baccelli, : Cell-based grouping in large-scale virtual environments. Tech. Rep. 3729. INRIA, July (1999)

[4] 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

[5] Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. of the ACM SIGCOMM Conference, San Diego, CA (2001) 149-160

[6] D. Eager, E. Lazowska, and J. Zahorjan. Adaptive Load Sharing in Homogeneous Distributed Systems. IEEE Transactions on Software Engineering, 12(5):662-675, 1986.

[7] W. Zhu, C. Steketee, and B. Muilwijk. Load balancing and workstation autonomy on Amoeba. Australian Computer Science Communications, 17(1):588--597, 1995.

[8] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In 29th Annual ACM Symposium on Theory of Computing, El Paso, Texas, 1997.

[9] B. Liu and E. A. Rundensteiner. Revisiting pipelined parallelism in multi-join query processing. In VLDB, page to appear, 2005.

[10] Su-Min Jang, Jae-Soo Yoo: An Efficient Distributed MMOG Server Using 2Layer-Cell Method. Edutainment 2007: 864-875.