
MMORPG 서버의 부하균등화를 위한 효율적인 분산처리 기법

Efficient Distributed Processing Scheme for Load Balancing of MMORPG Servers

장수민, 유재수
충북대학교 정보통신공학과

Su-Min Jang(rusu2004@naver.com), Jae-Soo Yoo(yjs@chungbuk.ac.kr)

요약

이 논문은 MMORPG(Massively Multiplayer Online Role-Playing Games)을 지원하는 새로운 분산처리 기법을 제안한다. 그러나 기존의 분산게임서버들은 핫스팟, 폭주 그리고 일반적으로 발생하는 서버오류와 같은 문제들을 갖고 있다. 이러한 문제점을 해결하기 위해서 제안하는 기법은 가상세계를 보다 작은 구역으로 나누고, 각 나누어진 구역은 다른 서버에 의해 처리된다. 본 논문은 모든 게임서버들과 사용자들 간에 요청되는 대역을 줄이는 기법을 제안한다. 또한 이 기법은 각각 다른 서버에 의해 제어되는 구역에 있는 사용자들 간에 끊임이 없는 통신을 제공한다. 우리는 다양한 성능평가를 통하여 제안한 기법의 우수성을 보인다.

■ 중심어 : | 온라인게임 | 분산처리 | MMORPG |

Abstract

In this paper, we propose a new distributed processing scheme to support MMORPG (Massively Multiplayer Online Role-Playing Games). But existing distributed game servers have some problems such as hotspot, congestion, server failure problems, and so on. To solve such problems, our proposed scheme begins by splitting the large virtual world into smaller regions, each region handled by a different server. We present a scheme that reduces the bandwidth requirements for both game servers and clients. Also our scheme allows seamless interaction among players residing on areas handled by different servers. We show an excellence of the proposed solution through various experiments.

■ keyword : | Online Game | Distributed Processing | MMORPG |

I. 서론

최근 MMORPG게임 서버는 이전 보다 넓은 가상의 다양한 공간에서, 보다 많은 사용자를 처리해야 하는 MMORPG게임으로 발전 되어 가면서 MMORPG게임

서버에 대한 부하를 가중 시키고 있다[1][2]. MMORPG 게임과 같이 수천 명이 동시 접속 가능한 게임에서의 서버의 처리는 모든 연산이 여러 개의 프로세스나 여러 개의 컴퓨터로 나누어 처리하는 분산 환경이 필요하며, 효율적인 처리를 위해서는 인접한 지역의 정보를 효과

* 본 연구는 2007학년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었습니다.

접수번호 : #070823-001

접수일자 : 2007년 08월 23일

심사완료일 : 2007년 10월 08일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

적으로 처리하는 방법이 필요하다[3]. 본 연구에서는 MMORPG게임의 분산처리를 위한 기존의 방식이 갖고 있는 문제점을 제시하고 이러한 문제점을 해결하는 효과적인 분산 처리 방식을 제안한다. 본 연구에서 제시하는 분산 처리 방식은 맵기반의 분할방식을 기반으로 설계한다. 또한 MMORPG게임서버의 효율적인 분산처리를 위하여, 서비스 영역의 분할을 최적화하기 위한 여러 가지 요소를 제안한다. 이러한 MMORPG 서버의 부하균등화를 위한 효율적인 분산처리 기법은 보다 다양한 게임서버의 개발을 지원한다. 또한 보다 효과적으로 게임서버의 부하를 낮추는 기법을 제공한다.

본 논문의 구성은 먼저 2장에서 관련연구에 대해서 알아보고, 3장에서는 제안하는 효율적인 MMORPG게임 서버 분할 및 처리기법을 제시한다. 4장에서는 성능평가를 수행하고 마지막으로 5장에서는 결론을 맺고 향후연구방향을 제시한다.

II. 관련연구

대부분의 온라인 게임은 클라이언트와 서버간의 통신을 통하여 처리하거나 클라이언트간의 통신을 통하여 처리한다[4][5]. 이러한 방식으로는 여러 가지 게임서버의 구성이 있다. [그림 1]은 일반적인 온라인게임서버의 구성도이다. [그림 1(a)]과 같은 형태를 P2P방식 [6][7]이라고 한다. 이러한 구성도의 서버는 게임사용자들의 계정관리와 게임 결과만을 서버에서 처리하고 실제 게임은 각 게임사용자의 컴퓨터에 설치된 프로그램으로 처리한다. 이러한 대표적인 게임으로는 블리자드에서 서비스 하는 배틀넷 형식을 갖는 스타크래프트가 있다.

또한 [그림 1(b)]과 같은 형태는 단일서버구조로 여러 개의 클라이언트가 하나의 서버에 연결되어 있는 형태이다. 단일 서버 구조는 하나의 게임서버에서 모든 클라이언트가 서비스를 제공받는다. 사용자의 계정관리 및 사용자간의 통신, 게임의 로직을 처리하는 것이다. 단점으로는 서버의 한계가 있기 때문에 동시접속자 수가 제한된다. 그래서 소수의 클라이언트가 접속하는 단

순한 게임을 서비스하는데 주로 사용한다. 그러나 이 구조의 장점은 하나의 서버에서 모든 게임 로직이 처리되므로 여러 개의 서버로 분산하여 처리하는 분산서버구조의 게임에서 발생하는 서버간의 시간동기화, 교착상태, 동시성, 무결성 등의 문제가 발생하지 않는다.

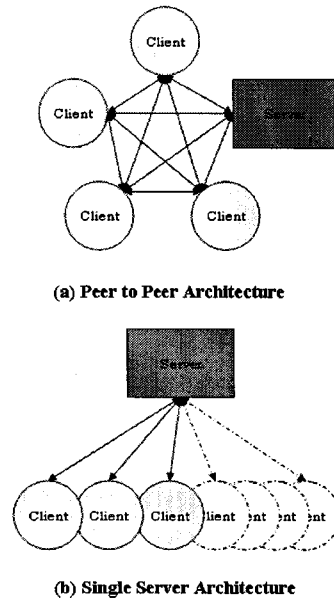


그림 1. 일반적인 온라인게임서버 구성도

그러나 MMORPG 서버는 동시에 같은 지역에서 많은 사용자가 게임을 할 수 있어야 하는 구조를 요구한다. 그래서 단일서버구조와 같은 형태의 구성으로는 원활한 서비스를 제공하기 힘들다. 그래서 그림 2와 같은 대칭형 서버구조(Replicated Server Architecture or Mirrored Server Architecture)와 비대칭형 서버구조(Non-Replicated Server Architecture)를 이용한 서비스가 최근에 이루어지고 있다. 대칭형 서버구조는 서버의 자원을 그대로 복사하는 방식으로 매우 간결한 구조이다. 또한 서버의 부하를 매우 균등하게 분할한다. 그러나 대칭형 서버 구조는 복제된 서버들 간의 동시성제어비용이 매우 높은 문제점이 있다. 최근에는 일반적으로 게임공간의 크기에 제한을 받지 않으면서 게임에서 발생하는 데이터와 메시지 트래픽을 효율적으로 처리하는 구조인 비대칭형 서버구조는 다른 이름으로는 맵 기반 분산서버구조(Map-based distributed server architecture)이라 한다. 이 방식은 게임공간을 처리가

능한 지역단위로 분할하고 각 지역마다 담당하는 서버를 두어 가상공간의 확장성을 높여주는 기법이다. 게임 공간을 분할하는 방식에 따라 전체시스템의 효율성이 좌우한다. 또한, 비대칭형 서버구조의 단점인 한 서버로 편중될 때 과부하를 갖는 문제점이 있다.

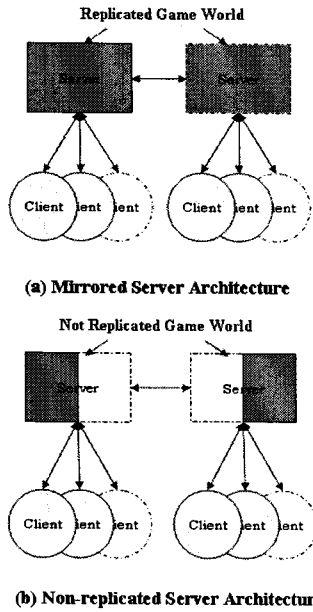


그림 2. MMORPG의 게임서버 구성도

서버로 분할하는 관점에서 고려해야 할 조건들로 이전의 지도의 면적, 오브젝트 수, 서버이전 부담률, 경계선의 길이등과 같은 네 가지 요소를 이용한 방식을 통하여 보다 효율적인 맵 분할이 되었다.

간단히 설명하면, [그림 3]에서 분할 S1과 S2 중에 지도의 면적 Part와 Part'의 차이 값이 적도록 분할한다. 그리고 오브젝트의 수가 균등하게 분할 되도록 분할 S3 보다는 S4가 효과적이다. 또한 서버의 이전비용이 적도록 S4보다는 S5가 유리하다. S6가 S5보다 유리한 조건은 경계선의 길이가 짧으면서 서버의 이전비용이 적게 발생하는 분할을 보여준다.

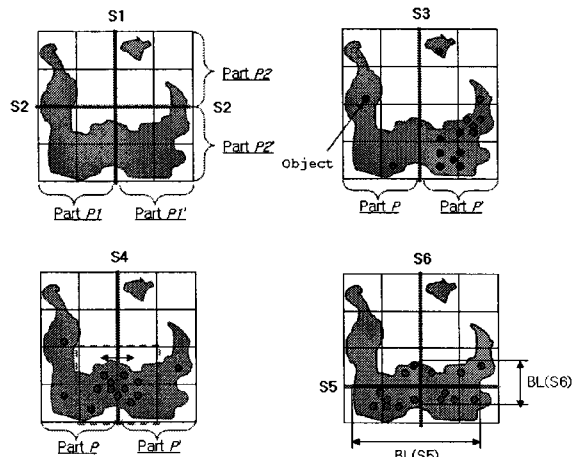


그림 3. 맵 분할시 고려사항

III. 제안하는 분산처리기법

제안하는 분산처리기법은 크게 두 가지의 기법을 사용하여 효율적으로 게임서버의 부하를 분산한다. 하나는 맵을 분할할 때 게임적인 요소뿐만 아니라 부하에 영향을 미치는 여러 가지 요소들을 고려하여 분할한다. 또 하나는 잘 분할된 서버의 형태로 서비스 한다하여도 일시적인 사용자의 폭주나 편중을 통한 서버의 부하를 효율적인 패킷처리 방식을 통하여 서버의 부하를 균등화하는 방식을 제공한다.

1. 효율적인 맵 분할 방식

일반적인 분산게임서버방식에서는 지도의 면적이나 오브젝트 수만을 고려한 분할 방식을 사용하였다[8]. 그러나 [9]에서는 분산서버를 구축하기 위해서 여러 개의

본 논문은 오브젝트의 주 이동경로라는 요소를 맵 분할 조건으로 추가적으로 고려한다. 오브젝트가 주로 이동하는 패스를 분할하는 것보다는 주 이동경로를 분할하지 않는 것이 더 효율적이다. [그림 4]에서 S7이 S8보다 서버 간에 클라이언트가 이동하는 이전비용이 적게 들어 서버의 부하가 감소한다.

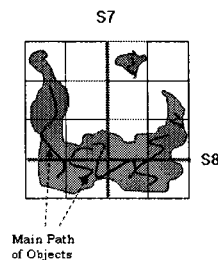


그림 4. 오브젝트의 주 이동경로를 이용한 분할

2. 효율적인 패킷처리 방식

제안하는 방식은 맵분할 방식의 분산서버의 구조이다. 이러한 방식은 한 서버로 오브젝트가 편중될 때 과부하가 생기는 문제점을 갖고 있다. 이를 해결하기 위하여 편중된 서버의 과부하를 분산하기 위한 새로운 기법을 제공한다. [그림 3]과 같이 편중에 따른 서버 C의 과부하를 분산 처리하기 위하여 게임서버의 메인 모듈 중에서 특정 모듈을 다른 서버로 이전하여 서버들의 부하를 균등화하는 방식을 제공한다. 그림 5는 제안하는 분산게임서버가 정상적인 균등화를 이루고 있는 일반상태와 특정지역에 사용자가 일시적으로 편중되어 과부하가 발생한 상태를 보여준다.

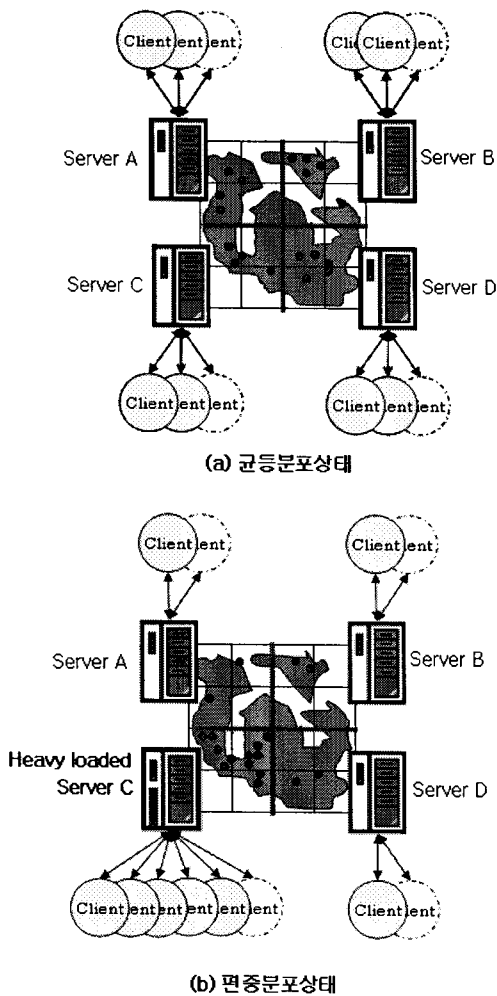


그림 5. 분산게임서버에서 정상상태와 편중에 따른 과부하 상태

과부하가 된 서버 C에서 처리해야 하는 클라이언트 일부의 특정 모듈을 과부하가 없는 서버 A, B에 분산시키는 방법이다. 여기서 특정모듈은 패킷의 전송에 관련된 클라이언트가 요청하는 패킷입력모듈과 게임로직을 통하여 결과를 보내주는 패킷출력모듈이다. 이 두 가지의 모듈은 서버부하의 대부분을 차지하는 부분으로, 분산처리하면 부하균등화에 효과가 크다. [그림 6]과 같이 서버 C에 접속되어 있는 클라이언트가 개별적으로 서버 C로 보내야 할 패킷들을 서버 A나 서버 B가 받아서 하나의 패킷으로 만들어 처리한다. 서버 C의 측면에서 보면 하나의 클라이언트를 처리하는 것과 같이 처리된다. 이러한 처리방식은 분산서버구조의 게임에서 발생하는 서버간의 시간동기화, 동시성, 무결성 등의 문제가 발생하지 않는 장점을 갖고면서 부하를 분산하기 때문에 매우 효율적이다.

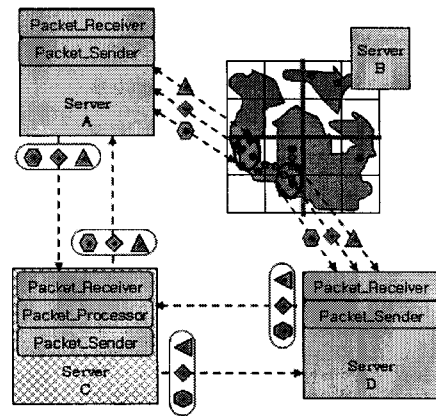


그림 6. 편중에 따른 과부하 분산을 위한 패킷처리방식

IV. 성능평가

성능평가는 가상의 클라이언트를 생성하여 로컬 네트워크를 통한 통신을 기반으로 하면서 동시에 접속하여 서버에 패킷을 요청하는 형식으로 수행하였다. 성능평가의 조건으로는 오브젝트의 수를 500에서 2000까지 변화를 주면서 면적만을 고려한 분할에서 각각의 서버의 처리속도와 다섯 가지의 조건을 모두 고려한 각각의 서버의 처리속도와 서버간의 처리시간의 차이 값을 측정하였다. 또한 한쪽 서버로 사용자가 편중되는 과정을

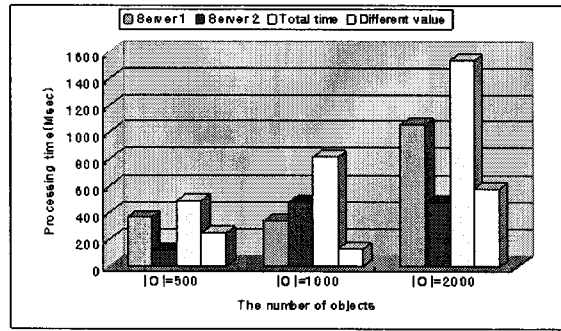
만들기 위하여 특정시간에 점차적으로 편중비율을 높이는 과정을 통하여 성능평가 하였다. [표 1]은 성능평가를 위한 상세한 조건들이다. 실험환경은 펜티엄 IV 2.0GHz에 512Mbyte의 메모리를 가진 컴퓨터에 LINUX환경에서 측정하였다.

표 1. 성능평가를 위한 상세한 조건

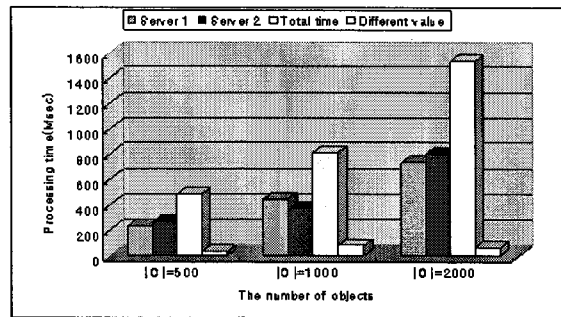
Player	player은 임의의 방향으로 이동한다. player(50%)는 최대 이동크기는 2이고 나머지(50%)는 5이다. 같은 셀에 monster가 있다면 player는 monster를 공격한다. player의 업데이트정보는 player의 자신의 셀과 주변 셀 8개를 포함하여 9개 셀의 정보를 보낸다.
Monster	max movement 1 이다. player가 공격하면 임의의 방향으로 이동한다.
Packet	최대 크기는 1024byte이다
IO	오브젝트의 전체수는 players(70%), monsters(30%)으로 구성한다.
IVSI	가상게임공간의 크기 맵의 30%의 이동 제한 구역으로 설정하였다.
P(Area), P(All)	P(Area)는 면적만을 고려한 분할이고 P(All)는 다섯 가지를 모두 고려한 분할이다.

1. 맵 분할 방식에 따른 처리시간

제안하는 방식은 맵을 분할시 지도의 면적, 오브젝트 수, 서버이전 부담률, 경계선의 길이, 오브젝트의 주 이동경로를 고려하였다. 맵 분할 방식에 따른 처리시간의 성능평가는 본 논문에서 제안하는 효율적인 패킷처리 방식을 적용하지 않은 상태에서 실행되었다. [그림 7]은 면적만을 고려한 맵 분할 방식과 본 논문에서 제시한 다섯 가지 조건을 모든 고려한 분할 방식의 처리시간을 비교하여 보여준다. 면적만을 고려한 [그림 7(a)]을 보면 클라이언트의 수가 증가할수록 서버들이 처리시간은 증가하였고, 두 개의 서버들의 처리시간이 많이 차이나는 것을 보여준다. 각 지역의 클라이언트가 다른 지역으로 이동하는 서버이전 비용이 많음을 알 수 있다. 그러나 모든 조건을 고려한 [그림 7(b)]의 경우에는 부하가 클라이언트의 수가 증가할수록 서버의 전체 처리시간이 증가한 것은 같지만 각 서버의 처리시간을 보면 거의 차이가 없음을 보여준다. 이와 같이 맵 분할시 본 논문에서 제시하는 조건들을 고려한 방식이 서버부하의 균등화에 매우 효과적임을 보여준다.



(a) 면적만을 고려하여 분할한 서버의 처리시간



(b) 모든조건을 고려하여 분할한 서버의 처리시간

그림 7. 맵 분할 방식에 따른 처리시간

2. 특정지역의 편중에 따른 처리시간

효율적인 분할방식을 통하여 분할된 영역을 각 서버가 맡아서 처리하여도 일시적인 편중으로 사용자가 특정 지역으로 모일 경우 그 서버는 과부하가 생긴다. 클라이언트의 편중비율변화에 따른 처리시간을 평가하였다. IS%는 서버 중에 한쪽 서버의 편중비율을 나타낸다. 편중비율은 0%에서 60%까지 변화하도록 하였다. 이때 IVSI= 1024이고 IOI=2000이다. 제안하는 방식은 클라이언트의 편중이 늘어남에 따라 다소 증가하지만 매우 균등함을 보여준다.

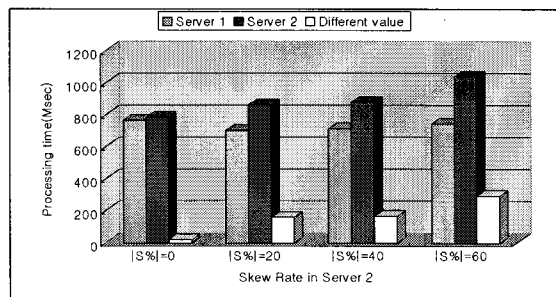


그림 8. 라이언트의 편중에 따른 처리시간

V. 결 론

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

- [7] M. Merabti and A. E. Rhalibi, "Peer-to-peer architecture and protocol for a massively multiplayer online game," *GlobeCom Workshops*, Vol.29, pp.519-528, Nov.-3 Dec. 2004.
- [8] J. C. S. Lui and M. F. Chan, "An efficient partitioning algorithm for distributed virtual environment systems *Parallel and Distributed Systems*," *IEEE Transactions*, Vol.13, No.3, pp.193-211, 2002.
- [9] S. M. Jang and J. S. Yoo, "An Efficient Distributed MMOG Server Using 2Layer-Cell Method," *Edutainment 2007, LNCS*, Vol.4469, pp.864-875, 2007.

참 고 문 헌

- [1] A. Abdelkhalek, A. Bilas, and A. Moshovos, "Behavior and performance of interactive multi-player game servers," *ISPASS01(2001)*, Vol.4, No.6, pp.137-146, Nov. 2001.
- [2] E. Lety, T. Turletti, and F. Baccelli, *Cell-based grouping in large-scale virtual environments*, Tech. Rep. 3729. INRIA, July, 1999.
- [3] A. Abdelkhalek and A. Bilas, "Parallelization and performance of interactive multiplayer game servers," *IEEE*, pp.26-30, April 2004.
- [4] G. Huang, M. Ye, and L. Cheng, "Modeling system performance in MMOG," *IEEE Communications Society Globecom Workshops 29*, pp.512-518, Nov.-3 Dec. 2004.
- [5] <http://www.igda.org/online>
- [6] S. Ito, H. Saito, H. Sogawa, and Y. Tobe, *A Propagation of Virtual Space Information Using a Peer-to-peer Architecture for Massively Multiplayer*, *Online Games ICDCS Workshops*, pp.04-07, 2006.

저 자 소 개

장 수 민(Su-Min Jang)

정회원



- 1997년 2월 : 목포대학교 전산통계학과(이학사)
- 1999년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2007년 8월 : 충북대학교 정보통신공학과 (공학박사)

▪ 2007년 9월 ~ 현재 : BK21 Post Doc

<관심분야> : 분산처리, 데이터베이스, 게임, 정보검색, 분산 객체 컴퓨터

유 재 수(Jae-Soo Yoo)

종신회원



- 1989년 2월 : 전북대학교 컴퓨터공학과 (공학사)
- 1991년 2월 : 한국과학기술원 전산학과 (공학석사)
- 1995년 2월 : 한국과학기술원 전산학과 (공학박사)

- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사
 - 1996년 8월 ~ 현재 : 충북대학교 전기전자컴퓨터공학부 교수
- <관심분야> : 데이터베이스 시스템, 정보검색, 멀티미디어 데이터베이스 분산 객체 컴퓨팅, 센서 네트워크