

# 멀티 플랫폼 기반 네트워크 응용을 위한 기반 구조

김진덕 · 진교홍

동의대학교

## Infrastructure for Network Applications based on Multi-Platform

Jin-Deog Kim · Kyo-Hong Jin

Donggeui University

E-mail : {jkd, khjin}@donggeui.ac.kr

### 요 약

기존의 고정 단말기(PC)에서만 주로 행해지던 각종 다중 사용자 접속 온라인 응용이 최근 모바일 단말기의 급속한 확산으로 기존 PC와 PDA, 휴대폰 등이 공동 작업을 수행하는 멀티 플랫폼 기반 온라인 응용으로 전환되고 있다. 이 논문에서는 멀티 플랫폼 기반 네트워크 응용을 위한 기반 구조를 제안하였다. 그리고 다양한 클라이언트의 프로세서 처리 능력과 통신 속도의 비대칭이라는 멀티 플랫폼 응용의 특징을 고려한 중복 일관성 제어 기법, 다양한 클라이언트간의 변경 전파 프로토콜을 제안하였다. 그리고 멀티 플랫폼 기반 채팅 프로그램을 제작하여 제안한 구조 및 기법들이 적절히 동작함을 보였다.

### ABSTRACT

Several multi-user online applications which are operated by the existing fixed terminals(PC) are being changed into online application based on multi-platform operated by the several PC, PDA and mobile phones to perform concurrent works due to the popularization of mobile devices recently. This paper proposed the infrastructure for network applications based on multi-platform. We also proposed the methods of consistency control based on multi-platform, update propagation protocols among diverse clients which take the asymmetry of computing power and network bandwidth among each client into account. Moreover, we implemented a chatting application based on multi-platform and it showed the proposed infrastructure and methods perform well.

### 키워드

멀티 플랫폼, 온라인 응용, 병행 제어, 변경 전파

### 1. 서 론<sup>1)</sup>

기존의 PC에서 주로 행해지던 각종 프로그램이 이제 휴대폰이나 PDA와 같은 모바일/무선 단말기로도 확산될 것으로 예상된다[7]. 그러므로 앞으로 급속히 증가될 것으로 예상되는 모바일/무선 단말기용 온라인 응용을 위한 기반 구축 기술이 개발되어야 한다. 다수의 PC, PDA, 휴대폰을 통해 유무선으로 동시에 수행되는 멀티 플랫폼 기반 온라인 응용 프로그램이 원활

히 서비스된다면 그 가치는 보다 증대될 것이다.

그러나 이러한 멀티 플랫폼(PC, PDA, 휴대폰) 기반 온라인 환경은 모바일/무선 단말기의 특성상 다음과 같은 이유로 기존의 온라인 응용 환경을 그대로 활용할 수가 없다. 우선, 각 클라이언트 기기의 사용 가능한 리소스의 차이가 심하고 프로세서의 처리 능력의 편차가 심하다. PC와 같은 고정 단말장치에 비해 휴대폰과 같은 모바일 휴대 장치는 극히 작은 용량의 메모리가 사용가능하며, 중앙 처리 장치의 처리 능력 또한 현격한 차이가 있다. 둘째, 통신 대역폭의 차이가 심하다. 모바일/무선 단말기는 광대역의 통신 대역폭을 사용할 수 없을 뿐만 아니라 열악한 통신 환경으로 인해

<sup>1)</sup> 이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2002-003-D00277)

새로운 통신 알고리즘을 고려해야 한다.

이 논문에서는 멀티 플랫폼 환경에서 온라인 응용의 원활한 서비스 제공을 위한 인프라 구축 기술을 개발하는 것을 목적으로 한다. 구체적으로 위와 같이 특성을 갖는 멀티 플랫폼 기반 다중 사용자의 동시 접속 온라인 응용 프로그램은 새로운 병행 제어(Concurrency Control) 기법과 변경 전파(Update Propagation) 기법을 제안하고자 한다.

분산 환경에서 중복 일관성 제어 문제는 전통적으로 동기식 제어[1]와 비동기 제어[3] 접근 방법이 있다. 동기식 제어 기법은 잠금과 2PC를 이용하여 모든 수정을 동기화 시켜서 직렬화(serializability)를 보장하는 방법이다. 이 기법은 공유 데이터의 동시 수정에 의한 불일치 문제(concurrency anomalies)가 발생하지 않는다는 장점이 있으나, 엄격한 제약으로 인하여 응답시간과 성능이 심각하게 저하되는 문제점이 있다. 비동기식 제어 기법은 동기식 제어 기법에 비해 성능이 향상되었지만 일정 기간 동안 데이터베이스의 일관성이 위배되는 단점이 있다.

동시 작업에 의해 공동 작업 환경에서 동기화를 지원하는 방식으로서 병행 엔지니어링(Concurrent Engineering) 방식[2, 4]이 있다. 병행 엔지니어링 방식에서 모듈간 종속성이 존재하면 사용자간 변경 상충(Update Conflict)이 발생하여 작업이 다시 피드백되어 재작업하는 기법이다.

공동 작업 환경에서 객체들에 대해 사용자-제어 잠금에 의해 동기화 작업을 지원하는 방식으로 CSCW(Computer Supported Cooperative Work) 방식[5, 6]이 있다. 그러나 이 논문에서 다루고자 하는 온라인 응용 실시간 처리용 변경 전파 기법과는 차이가 있다.

이 논문에서는 또한 제안한 기법이 올바르게 동작함을 확인하고자 멀티 플랫폼 기반 채팅 프로그램을 제작하여 테스트한다. 채팅 프로그램은 클라이언트로 PC, PDA, Phone 버전이 각각 존재하며 개발 환경 또한 전혀 다르지만 서로 데이터를 공유하거나 송수신이 가능하다. 이 논문에서 채팅 프로그램을 통해 제안한 기법을 구현하였지만 멀티 플랫폼 기반 다중 접속자 응용인 게임, 온라인 학습, 모바일 커머스 등에서도 쉽게 활용할 수 있다.

이 논문의 구성은 다음과 같다. 제 2장에서는 온라인 응용의 데이터 제어를 위한 전체 시스템 구조 및 이 논문에서 제안하는 멀티 플랫폼 기반 동시성 제어, 변경 전파 등에 대해 설명한다. 제 3장에서는 제안한 기법을 온라인 채팅 프로그램에 적용한 예를 들며, 제 4장에서는 결론을 맺는다.

## II. 멀티 플랫폼 응용의 기반 구조

### 2.1 시스템구조

멀티 플랫폼 응용을 위한 시스템구조는 크게 클라이언트/서버 구조와 피어-투-피어 구조로 구분할 수 있다. 클라이언트/서버 구조에서 서버를 담당하는 컴퓨터는 서버의 역할만을 지속적으로 수행하고 모든 클라이언트는 응용 프로그램의 입력을 서버에 보내며 서버는 이러한 입력을 기반으로 모든 진행 결과를 전체 클라이언트에게 브로드캐스팅한다. 메시지의 수는 클라이언트의 수에 비례하기 때문에 비교적 많은 수의 클라이언트를 수용할 수 있다. 2.3절에서 설명할 일관성 제어시에도 클라이언트/서버 구조가 사용된다.

피어-투-피어(Peer-to-Peer)구조는 온라인 응용에서 각 클라이언트 책임 하에 모든 변경 정보 메시지를 해당 클라이언트에 보내며, 정보를 받은 클라이언트는 개별적으로 연산을 수행하여 진행한다. 따라서 네트워크의 메시지 트래픽은 클라이언트의 수가 증가하면 기하급수적으로 증가하기 때문에 클라이언트의 수는 자연적으로 제한된다. 개별적인 메시지 전송으로 인해 부하가 각 클라이언트에 분산되며 서버의 부하를 줄일 수 있다는 장점이 있다. 예를 들어 실시간 전략 게임이나 액션 게임의 경우 각 클라이언트는 연결된 상대방의 클라이언트를 인식한 뒤 다른 서버의 도움 없이 서로 메시지를 주고 받는다. 그러나 이 구조는 각기 다른 플랫폼 클라이언트간의 통신이 요구되므로 메시지 전송방식의 적절한 변환 모듈이 요구된다. 특히 클라이언트/서버 구조와는 달리 하나의 클라이언트가 여러 개의 소켓을 가져야 하지만, 휴대폰은 오직 하나의 소켓만을 가지므로 에이전트(Agent)를 두어 처리한다. 에이전트는 2.2절에서 다룬다.

### 2.2 서버 및 클라이언트의 구조도

그림 1은 온라인 응용을 위한 서버의 구조를 설계한 것이다. 서버는 클라이언트들이 모든 서비스의 요청을 위해 최초로 접속하는 모듈로서 접속 및 종료하는 클라이언트의 관리와 함께 각 클라이언트의 요청을 수행하는 역할을 한다. 서버는 크게 소켓 관리기와 디스플레이 모듈과 데이터 제어기 및 소켓 리스트로 나뉜다. 데이터 제어기는 다중 사용자 환경의 동시성 제어와 변경전파 및 객체 관리의 역할을 수행한다. 소켓 리스트는 현재 서버에 연결된 클라이언트의 각 소켓 정보를 포함한다.

그림 2는 PC와 PDA용 클라이언트의 기본 구조로서 크게 제어부, 디스플레이 모듈, 서버 연결부, Peer-to-Peer 처리부, 클라이언트 리스트로 구성된다. 서버 연결부는 초기 로그인 과정, 동시성 제어(2.3절)등과

그림 4는 에이전트의 구조도를 나타낸 것이다. 에이전트는 서버에 존재하는 것으로 하나의 휴대폰이 서버에 접속할 때마다 하나의 에이전트가 생성된다. 에이전트의 구조는 PC 클라이언트와 거의 흡사하지만 휴대폰과의 통신 소켓을 가지고 있는 것이 차이점이다. 휴대폰 클라이언트는 하나의 소켓을 이용하여 에이전트에게 스트림 명령을 내리면 에이전트는 명령을 해석하여 수행한 뒤 결과를 다시 휴대폰에게 넘겨준다. 이렇게 에이전트를 두는 이유는 휴대폰이 오직 하나의 소켓만 생성이 가능하여 Peer-to-Peer 통신을 할 수 없기 때문이며, 에이전트를 도입하여 이 문제를 해결하였으며 아울러 저성능이라는 휴대폰의 단점을 극복할 수 있다.

그림 1 멀티 플랫폼 환경의 서버 구조

같은 클라이언트/서버 구조의 데이터 통신을 위한 모듈이며, 클라이언트 리스트는 변경 정보의 동기화(2.3절)를 위한 정보를 관리하는 모듈로서 연결된 전체 클라이언트와 해당 그룹에 속하는 클라이언트 리스트 정보를 유지한다.

그림 4 에이전트 구조

### 2.3 병행 제어 및 변경 정보의 동기화

전통적으로 제시된 두 가지 일관성 제어 기법은 각기 장단점이 있지만, 멀티 플랫폼 온라인 응용에서는 적용하기 어렵다. 왜냐하면 멀티 플랫폼 기반 응용에서는 각 클라이언트마다 프로세서의 처리 능력 차이와 네트워크 속도 차이로 인해 잠금 기법은 그 정확성을 보장할 수 없다. 다시 말해, 서버에서 중앙 관리하는 잠금 기법은 전술한 차이로 인해 클라이언트에서 잠금을 요구(Request)한 순서와 잠금을 획득한 순서가 일치하지 않을 수도 있다는 점이다. 이 문제는 타임 스탬프 기법에서도 마찬가지로 적용된다. 단, 타임 스탬프 기법은 각 클라이언트 내의 대기 시간은 생략되지만, 공동 작업 결과의 동기화 처리를 해주어야 한다.

이 논문에서는 클라이언트의 처리 능력 차이 및 네트워크 속도에 관계없이 잠금을 요청한 시간을 기준으로 잠금을 획득하는 알고리즘을 제안한다. 알고리즘에서 최초 서버는 각 클라이언트가 접속할 때마다 클라이언트 시간과 서버 시간의 차이값을 저장해 놓는다. 그 뒤 하나의 객체 object에 대한 잠금 요청이 왔을 때 이후 적정 기간(threshold period)동안 object에 대한 잠금 요청이 있는 클라이언트를 살펴서 각 클라이언트가 잠금을 요청한 시간을 기준으로 보정한 뒤 제일 먼저 요청한 클라이언트에게 잠금 권한을 부여하는 것이다.

그림 2 PC 클라이언트 구조

그림 3은 휴대폰 클라이언트의 구조를 나타낸 것이다. 그림에서 오직 하나의 소켓만이 에이전트와 통신을 수행하며 연결된 클라이언트의 식별자 리스트만 가지며, 구체적인 IP 주소와 상태값은 에이전트에서 관리한다. 그리고 데이터 모듈은 에이전트와의 통신을 관리하는 것으로서 소켓 생성, 입력, 출력, 종료 등과 같은 일련의 동작을 수행한다. 클라이언트간에 이루어지는 소켓 생성 및 모든 작업은 에이전트를 거쳐 처리되므로 대부분의 작업을 에이전트가 수행하게 된다.

그림 3 PDA 클라이언트 구조

그리고 온라인 변경 전파를 위해 3가지 부류로 구분하여 변경 정보의 즉시 전파를 수행한다. 첫째, 전체 클라이언트에게 브로드캐스팅, 둘째, 정적 관련성을 갖는 클라이언트에게 변경 전파, 셋째, 동적 관련성을 갖는 클라이언트에게 변경 전파 방법으로 구분한다. '정적 관련성' 섹션은 현재 연결된 클라이언트 중 연결 시 결정되는 정적 그룹정보를 의미하는 것으로서 해당 그룹별로 선택적인 메시지를 전파하고자 할 때 사용된다. 그룹의 조합은 필요에 따라 여러 번 반복가능하다. '동적 관련성'은 온라인 응용 도중에 순간 순간 발생하며 변경 전파 과정을 거친 후에 사라지는 정보이다. 예를 들어 동시에 같은 아이템을 요구한 클라이언트들은 모두 동적 관련성 정보로 생성이 되며, 아이템을 획득하지 못한 여러 개의 클라이언트는 메시지를 받아 복귀(Rollback)하게 된다.

### III. 구현 결과

이 논문에서 제안한 시험 구조와 데이터 제어 기법을 도입하여 멀티 플랫폼 기반 온라인 채팅 프로그램을 구현해 보았다. 환경은 다음과 같다.

- 서버 : Pentium 4, C++ Program
- PC 클라이언트 : Pentium 4, C++ Program
- PDA 클라이언트 : Compaq iPaq 3850, Embedded Visual Tool
- Phone 클라이언트 : SK-VM Phone Emulator Ver 1.1, Java(J2ME)

이 논문에서 제안한 동시성 제어와 변경 전파 모듈의 수행이 요구되는 상황으로서 두개의 PC 클라이언트(star, s004161)가 동시에 하나의 PDA 클라이언트(cosmos)에게 대화 요청을 하였을 경우 서버의 종속성 도표에는 그림 5와 같은 동적 관련성이 생성이 되며, 2.3절의 동시성 제어 알고리즘에 의해 star가 cosmos에 대한 대화 권한을 획득하여 대화 상대로 추가되었고(그림 6), s004161의 수행화면에서는 대화 불가능이란 메시지가 서버에서 즉시 변경 전파됨을 알 수 있다(그림 7).

그림 5 동시 요청으로 생성된 동적 관련성

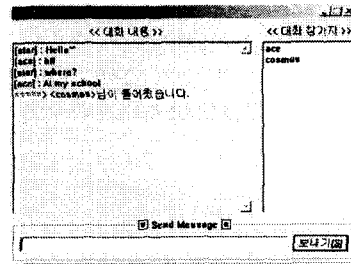


그림 6 대화 권한 획득 후 변경 전파 결과

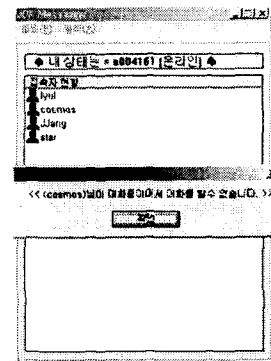


그림 7 대화 불가능 정보의 즉시 전파

### IV. 결론

이 논문은 멀티 플랫폼(PC, PDA, Phone) 온라인 응용의 원활한 서비스를 위해 클라이언트/서버 인프라 구축 기술을 개발하고자 하였다. 이러한 온라인 환경은 각 클라이언트 기기의 사용 가능한 리소스(메모리 용량, 프로세서 처리 능력) 차이가 심하고 통신 대역폭의 차이가 심하여 기존의 데이터 제어 기법을 그대로 적용할 수 없다. 이러한 문제점을 해결하기 위해 이 논문에서는 멀티 플랫폼 기반 다중 사용자의 동시 접속 온라인 응용 프로그램에서 효과적인 데이터 제어를 위해 새로운 동시성 제어 기법과 변경전파 기법을 제안하였다.

구체적으로 다양한 콘텐츠를 제공하는 온라인 응용의 동시성 제어를 위해 다양한 클라이언트의 처리 능력 차이 및 네트워크 속도에 관계없이 잠금을 요청한 시간을 기준으로 잠금을 획득하는 알고리즘을 제안하였다. 그리고 변경 정보의 동기화를 위해 시나리오에 따른 종속성 도표를 근거로 변경 정보를 전파하는 방법을 제안하였다.

또한 이 논문에서 서버, PC 클라이언트, PDA 클라이언트, 휴대폰 클라이언트 모듈을 개발하여 적용해본 채팅 프로그램을 통해 제안한 기법이 적절히 동작함을 알 수 있었다. 이 논문에서 제안한 기법들은 온라인 계

임, 학습 등 다양한 분야에서도 쉽게 적용할 수 있으리라 판단된다.

### 참고문헌

- [1] P.A. Bernstein, N. Goodman(1984), "An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases", ACM Transactions on Database Systems, vol. 9, no. 4, pp. 596-615
- [2] D.R. Brown, M.R. Cutkosky, J.M. Tenenbaum(1990), "Next-Cut : A Computational Framework for Concurrent Engineering", Int. Symposium on Concurrent Engineering
- [3] G. Coulouris, J. Dollimore, T. Kindberg(1994), "Distributed Systems : Concepts and Design" , Addison-Wesley Publishing
- [4] M.R. Cutkosky, R.S. engelmores, R.E. Fikes, M.R. Genesereth, T.R. Gruber, W.S. Mark, J.M. Tenenbaum, J.C. Weber(1993), "PACT : An Experiment in Integrating Concurrent Engineering Systems", IEEE Computer, pp.28-37
- [5] P. Dewan, J. Riedl(1993), "Toward Computer Supported Concurrent Software Engineering" , IEEE Computer, pp. 17-27
- [6] K. Dittrich, R. Lorie(1988), "Version Support for Engineering Database Systems", IEEE Transactions on Software Engineering, Vol. 14, No. 4, pp. 429-427
- [7] J. Jing, A. Elmagarmid(1999), "Client-Server Computing in Mobile Environments", ACM Computing Surveys, Vol 31, No 2