

멀티 플랫폼 기반 다중 사용자 온라인 응용을 위한 데이터 제어

김진덕*, 진교홍**

*동의대학교 컴퓨터공학과

**동의대학교 멀티미디어공학과

e-mail : jdk@dongeui.ac.kr

Data Control for Multi-User Online Application based on Multi-Platform

Jin-Deog Kim*, Kyo-Hong Jin**

*Dept. of Computer Engineering, Dongeui University

**Dept. of Multimedia Engineering, Dongeui University

요 약

최근 기존의 고정 단말기(PC)에서만 주로 행해지던 각종 다중 사용자 접속 온라인 응용이 PC와 PDA, 휴대폰 등이 공동 작업을 수행하는 멀티 플랫폼 기반 온라인 응용으로 전환되고 있다. 그러나 지금까지 멀티 플랫폼 기반 온라인 응용은 다중 사용자 환경 및 각 클라이언트의 처리 능력과 통신 속도의 비대칭을 고려하지 않아 컨텐츠가 매우 단순한 실정이다.

이 논문에서는 멀티 플랫폼 기반 온라인 응용의 인프라 구축을 위한 기술을 개발하고자 하였다. 구체적으로 멀티 플랫폼 단말 장치간의 데이터 송수신을 위한 네트워크 구조, 다중 사용자 환경의 일관성 제어 기법, 다양한 클라이언트간의 변경 전파 기술을 제안하였다. 그리고 여러 개의 PC, PDA, 휴대폰이 연동되어 동작하는 채팅 프로그램을 제작하여 제안한 기법들이 적절히 동작함을 보였다.

1. 서론

국내의 인터넷 및 휴대폰의 보급이 폭발적인 성장세를 보이며, 최근 PDA 사용자 역시 꾸준히 증가하고 있는 추세이다. 이와 같은 각각의 기기들은 유선 또한 무선의 형태로 온라인 응용 프로그램을 운용할 수 있다. 아울러 기존의 PC에서 주로 행해지던 각종 프로그램이 이제 휴대폰이나 PDA와 같은 모바일/무선 단말기로도 확산될 것으로 예상된다[5].

따라서 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 원활히 서비스 된다면 그 가치는

보다 증대될 것이다.

그러나 이러한 멀티 플랫폼(PC, PDA, 휴대폰) 기반 온라인 환경은 모바일/무선 단말기의 특성상 각 클라이언트 기기의 사용 가능한 리소스의 차이가 심하고 프로세서의 처리 능력의 편차가 크다는 점과 통신 대역폭의 차이가 심하다는 점 때문에 기존의 응용 프로그램에 적용한 기법을 그대로 적용할 수가 없다.

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 중복 일관성 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 있

* 본 논문은 2002년도 부산테크노파크 연구지원(BTP-BDE-9)으로 수행되었음

다. 동기식 제어 기법은 잠금(Lock)과 2PC 를 이용하여 모든 수정을 동기화 시켜서 직렬화(serializability)를 보장하는 방법이다. 비동기식 제어 기법은 수정은 이 서버에서만 허용한다. 이 후 적당한 시기에 서버의 책임으로 중복 일관성 제어를 위해 수정 전파 통신을 수행하는 방식이다. 전통적으로 연구된 두 가지 일관성 제어 기법은 각기 장단점이 있지만, 본 논문에서 연구하고자 하는 멀티 플랫폼 온라인 응용의 특수성을 고려하지 않아 다소 차이점이 있다. 멀티 플랫폼 환경은 통신 대역폭의 편차로 인해 공유 데이터를 서버에서 관리하는 것이 아니라 각 클라이언트에서 중복하여 관리하며 독자적으로 시나리오를 진행하다가 변경 전파를 하는 과정에서 일관성 제어 위반 사실이 발견되므로 새로운 중복 일관성 제어 기법이 도입되어야 한다. 현재 멀티 플랫폼 환경에서 가장 많은 콘텐츠를 제공하는 게임 분야에서도 다중 사용자의 동시 수행에 따른 일관성 제어 문제를 다루는 경우는 거의 없다.

동시 작업에 의해 공동 작업 환경에서 동기화를 지원하는 방식으로서 병행 엔지니어링(Concurrent Engineering) 방식[2]이 있다. 병행 엔지니어링에서는 사용자들이 동시에 두개 이상의 모듈을 작업하면서 상호 연관된 데이터를 송수신한다. 공동 작업 환경에서 객체들에 대해 사용자-제어 잠금에 의해 동기화 작업을 지원하는 방식으로 CSCW(Computer Supported Cooperative Work) 방식[4]이 있다. 그러나 이 논문에서 다루고자 하는 온라인 응용의 실시간 처리를 위한 변경 전파 기법과는 차이가 있다. 또한 응용 프로그램의 진행 정보에 대한 변경 발생시 접속되어 있는 다른 클라이언트에게 통보 해야 하는 변경 전파 기법 또한 각 클라이언트마다 통신 기법이 상이하므로 새로이 연구되어야 한다.

또한 이 논문에서는 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 있으리라 판단된다.

이 논문의 구성은 다음과 같다. 제 2 장에서는 온라인 응용의 데이터 제어를 위한 전체 시스템 구조 및 이 논문에서 제안하는 멀티 플랫폼 기반 동시성 제어, 변경 전파, 객체 처리 기법 등에 대해 자세히 설명한다. 제 3 장에서는 제안한 기법을 온라인 채팅 프로그램에 적용한 예를 들며, 제 4 장에서는 결론 및 향후 연구 과제를 다룬다.

2. 멀티 플랫폼 환경의 데이터 제어

2.1 멀티 플랫폼 응용의 시스템구조

여기서는 온라인 시스템 구조의 종류를 제시하고 각각의 특징을 살펴본다. 제시하는 구조들은 멀티 플

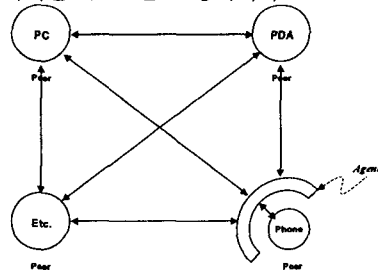
랫폼 온라인 응용에서 항상 고정된 것이 아니라 상황에 따라 적합한 구조를 선택하여 사용하게 된다.

(1) 클라이언트/서버 구조

이 구조에서 서버를 담당하는 컴퓨터는 서버의 역할만을 지속적으로 수행하고 모든 클라이언트는 응용 프로그램의 입력을 서버에 보내며 서버는 이러한 입력을 기반으로 모든 진행 결과를 전체 클라이언트에 게 브로드캐스팅한다. 메시지의 수는 클라이언트의 수에 비례하기 때문에 비교적 많은 수의 클라이언트를 수용할 수 있다[7]. 예를 들어 온라인 게임의 경우 주로 사용자 로그인 및 전체 게임 정보의 변경 전파시에 주로 사용되는 시스템 구조이다. 그리고 2.2 절에서 설명할 일관성 제어시에도 이 구조가 사용된다.

(2) 피어-투-피어(Peer-to-Peer) 구조

이 구조(그림 1)는 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[7]. 반면, 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 받는 방식이다.

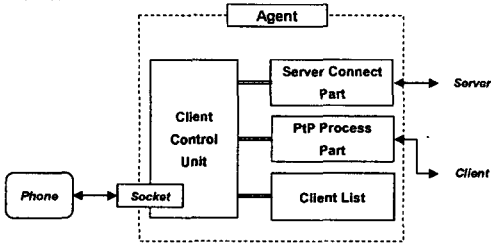


(그림 1) Peer-to-Peer 구조

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 하므로, 휴대폰은 오직 하나의 소켓만을 가질 수 있으므로 에이전트(Agent)를 두어 처리한다.

그림 2 는 에이전트의 구조도를 나타낸 것이다. 에이전트는 서버에 존재하는 것으로 하나의 휴대폰이 서버에 접속할 때마다 하나의 에이전트가 생성된다. 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 해결 하였으며 아울러 저성능이라는 휴대폰의 단점을 극복

할 수 있다.



(그림 2) 에이전트 구조

2.2 동시성 제어

다양한 유형의 다중 접속자 온라인 응용에서는 여러 개의 클라이언트가 동작하는 다중 사용자 환경에서 동일한 자원의 동시 접속 시 발생하는 문제를 해결하기 위한 동시성 제어 기법은 두 가지 방법이 존재한다. 우선, 항상 일관성을 유지해야 하는 경우에는 전통적인 잠금 기반 동시성 제어를 수행하는 반면, 실시간 실행을 위해 순간적인 일관성 위배를 허용하는 경우에는 타임 스탬프 기반 동시성 제어 기법[6]을 도입하여 각 클라이언트 연산의 병행성을 최대한 허용하면서 우선 권한이 없는 클라이언트의 연산을 철회하는 방법을 택해야 한다.

이 논문에서는 클라이언트의 처리 능력 차이 및 네트워크 속도에 관계없이 잠금을 요청한 시간을 기준으로 잠금을 획득하는 알고리즘을 제안한다. 이 기법은 타임 스탬프 기법에도 적용 가능하다. 알고리즘에서 최초 서버는 각 클라이언트가 접속할 때마다 클라이언트 시간과 서버 시간의 차이값을 저장해 놓는다. 그 뒤 하나의 객체 object_i에 대한 잠금 요청이 왔을 때 이후 적정 기간(threshold period)동안 object_i에 대한 잠금 요청이 있는 클라이언트를 살펴서 각 클라이언트가 잠금을 요청한 시간을 기준으로 보정한 뒤 제일 먼저 요청한 클라이언트에게 잠금 권한을 부여하는 것이다. 그림 3 은 동시성 제어 알고리즘을 의사코드로 나타낸 것이다.

Algorithm Concurrency_Control_Multi_Platform

```

{
  Whenever Client(i) is connected
  {
    Receive ClientTime
    Calculate Diff(i) = ServerTime - ClientTime
    Register Diff(i)
  }
  ...
  When a Client(i) request Lock(objecti)
  {
    Calculate ReqestTime(i) = ClientTime(i)+Diff(i)
    During( threshold period )
    {
      Whenever Client(j) request Lock(objecti)
      Calculate ReqestTime(j) = ClientTime + Diff(i)
    }
    Select minimal ReqestTime(i)
    Client(i) acquire Lock(objecti)
  }
}
    
```

(그림 3) 동시성 제어 알고리즘

2.3 변경 정보의 동기화

이 논문에서는 온라인 실시간 변경 전파를 위해 3 가지 부류로 구분하여 변경 정보의 즉시 전파를 수행한다. 그림 4 에 나타난 바와 같이 서버에서 관리하는 종속성 도표는 전체 3 부분으로 나누어진다. ‘전체 클라이언트’ 섹션은 현재의 온라인 응용의 해당 개설 창구에 접속한 모든 클라이언트를 의미하는 것으로서 서버가 모든 클라이언트에게 브로드캐스팅하고자 할 때 사용된다. ‘정적 관련성’ 섹션은 현재 연결된 클라이언트 중 연결시 결정되는 정적 그룹정보를 의미하는 것으로서 해당 그룹별로 선택적인 메시지를 전파하고자 할 때 사용된다. 그룹의 조합은 필요에 따라 여러 번 반복가능하다. 예를 들어 온라인 게임 응용에서 A, B 두 그룹으로 나누어 진행될 때 각 그룹마다 변경 전파 메시지가 달라야 할 경우에 사용된다. ‘동적 관련성’은 온라인 응용 도중에 순간 순간 발생하며 변경 전파 과정을 거친 후에 사라지는 정보이다. 예를 들어 동시성 제어에서 타임 스탬프 기법을 사용했을 경우 동시에 같은 아이템을 요구한 클라이언트들은 모두 동적 관련성 정보로 생성이 되며, 아이템을 획득하지 못한 여러 개의 클라이언트는 서버로부터 메시지를 받아 복귀(Rollback)하게 된다. 그림 4 에서 동적 관련성은 응용 프로그램이 진행되는 동안 관련성 정보가 생성 후 삭제 과정이 반복되며, ‘Waiting’은 아이템에 대한 권한이 확정되지 않은 상태이며, ‘Acquire’는 아이템을 획득한 상태이며, ‘RollBack’은 아이템을 획득하지 못해 복귀해야 함을 의미한다. 그리고 각 클라이언트의 정보는 다음과 같다.

Info. of client C_i = [Login_ID, IP_Address, Socket_Port_No]

지금까지 설명한 종속성 도표는 서버에서 클라이언트/서버 구조 형태로 변경 정보를 전파하기 위한 메타 데이터입에 반해 각 클라이언트 또한 Peer-To-Peer 구조 형태로 직접 다른 클라이언트와의 통신을 위해 ‘전체 클라이언트’와 ‘정적 관련성’ 정보를 유지하고 있다.

Connected Clients	Info. of client C1		
	Info. of client C2		
	Info. of client Cn		
Static Dependency	Group A	Info. of client C1	
		Info. of client C9	
	Group B	Info. of client C3	
		Info. of client C6	
Dynamic Dependency	Item A	Info. of client C5	Waiting
		Info. of client C3	Waiting
	Item B	Info. of client C2	Acquire
		Info. of client C8	RollBack
		Info. of client C7	RollBack

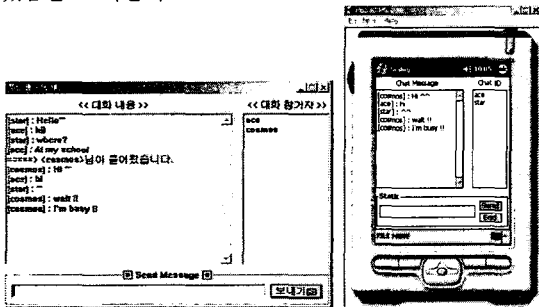
(그림 4) 종속성 도표

3. 프로그램의 구현 예

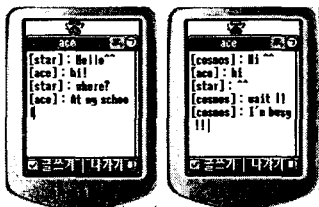
이 논문에서 제안한 시스템 구조와 데이터 제어 기법을 도입하여 멀티 플랫폼 기반 온라인 채팅 프로그램을 구현해 보았다. 구현 환경은 다음과 같다.

- 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)

그림 5 는 Peer-to-Peer 구조로 각 클라이언트간의 데이터를 송수신하는 과정을 보여주는 것이다. (a)는 PC(ID : star) 클라이언트이며, (b)는 PDA(ID : cosmos) 클라이언트이며, (c)는 휴대폰(ID : ace) 클라이언트이다. 최초 star 와 ace 가 대화를 하고 있는 상황에서 cosmos 가 추가되어 3 개의 클라이언트가 N:M 통신을 하고 있음을 보여준다.



(a) PC 클라이언트 (b) PDA 클라이언트



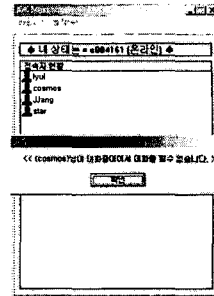
(c) 휴대폰 클라이언트

(그림 5) 각 클라이언트의 대화 내용

다음은 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 6 과 같은 동적 관련성이 생성이 되며, 2.2 절의 동시성 제어 알고리즘에 의해 star 가 cosmos 에 대한 대화 권한을 획득하여 대화 상대로 추가되었고, s004161 의 수행화면에서는 대화 불가능이란 메시지가 서버에서 즉시 변경 전파됨을 알 수 있다 (그림 7).

Dynamic Dependency	cosmos	Info. of star	Acquire
		Info. of s004161	RollBack

(그림 6) 동시 요청으로 생성된 동적 관련성



(그림 7) 대화 불가능 정보의 즉시 전파

4. 결론

멀티 플랫폼 다중 사용자 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. 이 논문에서는 멀티 플랫폼 기반 다중 사용자의 동시 접속 온라인 응용 프로그램에서 효과적인 데이터 제어를 위해 새로운 동시성 제어 기법과 변경전파 기법을 제안하였다. 동시성 제어 기법은 각 클라이언트의 잠금 요구 시간을 기준으로 잠금 권한을 부여하는 방법을 제안하였고, 실시간 변경 전파를 위해 3 단 계로 구성된 종속성 도표를 이용하였다.

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 게임, 학습 등 다양한 분야에서도 쉽게 적용할 수 있으리라 판단된다. 앞으로는 보다 엄격한 제약 조건을 요구하는 멀티 플랫폼 기반 온라인 게임을 개발하여 본 논문에서 제안한 기법들을 적용하고자 한다.

참고문헌

- [1] P.A. Bernstein, N. Goodman, " An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases ", ACM Transactions on Database Systems, vol. 9, no. 4, pp. 596-615, 1984
- [2] D.R. Brown, M.R. Cutkosky, J.M. Tenenbaum, " Next-Cut : A Computational Framework for Concurrent Engineering ", Int. Symposium on Concurrent Engineering, 1990
- [3] G. Coulouris, J. Dollimore, T. Kindberg, " Distributed Systems : Concepts and Design ", Addison-Wesley Publishing, 1994
- [4] P. Dewan, J. Riedl, " Toward Computer Supported Concurrent Software Engineering ", IEEE Computer, pp. 17-27, 1993
- [5] J. Jing, A. Elmagarmid, "Client-Server Computing in Mobile Environments", ACM Computing Surveys, Vol 31, No 2, 1999
- [6] R. Ramakrishnan, J. Gehrke, " Database Management Systems ", McGraw-Hill, pp. 523-570, 2000
- [7] 이만재, " 온라인 게임 엔진 기술 동향 ", 한국정보과학회 학회지, 20 권, 1 호, pp.12-18, 2002