

모바일 예약 시스템에서 데이터 일관성을 위한 동기화 메커니즘

정영섭*, 김강석*, 김재훈*, 홍만표*

*아주대학교 지식정보공학과

E-mail : richfamily@ajou.ac.kr

Synchronous Mechanism for Ensuring Data Consistency in Mobile based Reservation System

Youngsub Jeong*, Kangseok Kim*, Jai-hoon Kim*, Manpyo Hong*

*Dept. of Knowledge Information Engineering, Graduate School of Ajou University

요 약

최근 모바일 기기의 발달로 모바일 환경에서도 예약 서비스를 이용할 수 있다. 모바일 예약 시스템은 서버와 사용자간에 예약 정보 데이터를 일관성 있게 유지하여 사용자에게 실시간 예약 현황을 제공해야 한다. 현재 모바일 예약 시스템은 어플리케이션을 기반으로 서비스를 제공하고 있고, 웹 서비스와 같이 요청 대 응답 방식으로만 데이터를 처리하기 때문에 사용자가 요청하지 않으면 변화된 데이터를 알 수 없다는 문제점이 있다. 이는 데이터 일관성의 문제로 연결되어 사용자에게 실시간으로 예약 현황을 제공하지 못하게 된다. 이러한 문제점을 해결하기 위해 본 논문에서는 모바일 예약 시스템에서 스트리밍 방식을 개선한 Push를 통해 데이터 일관성을 유지하는 동기화 메커니즘을 제안한다.

1. 서론

현재 모바일 환경에서의 예약 시스템은 PC 환경과 동일하며 기차, 영화, 버스 등의 예약 서비스가 다양하게 제공되고 있다. 모바일 예약 시스템은 기존의 예약 웹 페이지를 연동하거나 자체 어플리케이션으로 서비스를 제공하고 있다. 예약 시스템에서 중요한 것은 어느 시점에서든지 사용자의 단말기는 서버와 예약 정보 데이터의 일관성을 유지하면서 사용자에게 예약 현황을 보여줘야 한다는 것이다. 그러나 기존의 모바일 예약 시스템은 사용자가 요청할 경우에만 예약 정보 데이터를 응답하는 방식으로 사용자의 단말기와 서버간의 예약 정보 데이터 일관성을 유지하지 못한다. 사용자의 단말기와 서버간의 예약 정보 데이터의 일관성을 유지하기 위해서는 사용자의 단말기가 서버에게 지속적인 요청을 해야 하고, 이로 인해 사용자에게 편리한 예약 서비스를 제공할 수 없다는 문제점이 있다.

이러한 문제점을 해결하기 위해 본 논문에서는 스트리밍 방식을 개선한 Push를 통해 데이터의 일관성을 유지하는 동기화 메커니즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 Push 기술, Ajax를 이용한 실시간 예약 시스템 방식, 3G(3Generation) 모바일 네트워크에서의 Push 시스템에 대해 살펴본다. 3장에서는 모바일 예약 시스템의 설계, 4장에서는 데이터의 일관성을 보장하는 동기화 메커니즘에 대해 설명한다. 5장에서는 검증 결과, 6장은 결론 및 향후 연구에 대해 설명한다.

2. 관련연구

2.1 Push 기술

Push 기술은 구현 방식에 따라 스트리밍 방식과 룽 폴링 방식으로 나눌 수 있다. 스트리밍 방식은 클라이언트가 서버에게 최초 요청 시에만 요청 메시지를 보낸다. 그 이후 클라이언트의 요청이 없이도 서버에 이벤트가 있을 때마다 응답을 해주는 방식이다. 룽 폴링 방식은 서버에게 요청을 하면 이벤트가 발생하기 전까지 대기상태에 있다가 이벤트 발생 시 응답하고 다시 요청하는 방식이다[1][2].

2.2 Ajax를 이용한 실시간 예약 시스템

Ajax(Asynchronous JavaScript And XML)를 이용한 실시간 예약 시스템은 Ajax Push 기술을 이용하여 데이터 일관성을 유지한다[3]. Ajax Push는 HTTP상에서 Push를 하는 방식을 가리킨다. 일반적인 웹 방식은 웹 브라우저가 서버에게 요청한 뒤, 응답받는 방식이었지만 Ajax Push의 경우 룽 폴링과 스트리밍 방식을 이용하여 Push가 가능하다[4]. 그러나 HTTP는 프로토콜 특성상 지속적인 연결을 갖지 못한다. 이를 위해 Ajax Push는 요청할 때마다 쓰레드를 생성하여 연결을 끊지 않고 유지하면서 Push한다. 그 결과 쓰레드가 많아지면 문맥교환 시간이 길어져 시스템의 전체적인 효율이 떨어지고 이는 실시간 예약 서비스를 제공하기에 무리가 있다.

* 본 연구는 지식경제부 및 한국인터넷진흥원의 "고용계약형 지식정보보안 석사과정 지원사업"의 연구결과로 수행되었음

2.3 3G 모바일 네트워크에서의 Push 시스템

모바일에 할당되는 IP 주소는 사실 IP이다. 이 사실 IP는 GGSN(Gateway GPRS Support Node)내의 네트워크에서만 유효한 IP이다. GGSN은 모바일의 사실 IP와 공인 IP를 변환해주는 중계 역할을 한다[5]. 이러한 환경에서는 서버가 모바일로 접근할 수 없다는 문제점을 갖게 된다. 따라서 모바일 Push 시스템의 경우 모바일에서 세션을 만들고 서버는 이 세션을 이용하여 모바일에게 정보를 Push한다. 정보를 Push 하기 위해서는 이 세션을 계속 유지하고 있어야 한다.

3. 제안하는 모바일 예약 시스템 설계

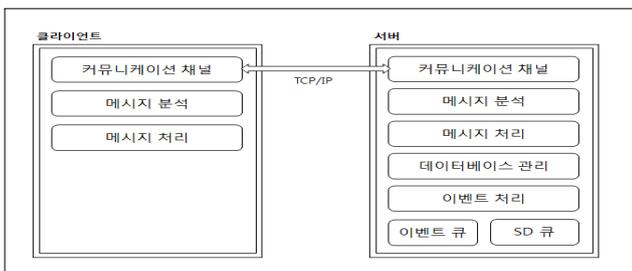
제안하는 모바일 예약 시스템은 영화관 시스템에 적용할 수 있도록 설계한다. 제안하는 시스템은 이벤트 기반의 Push 방식을 사용하므로 이벤트를 메이저 이벤트와 마이너 이벤트로 나눈다. 메이저 이벤트는 데이터 일관성을 유지하기 위한 이벤트로써 (그림 1)과 같이 메이저 이벤트를 수신한 서버는 예약을 진행하고 있는 클라이언트들에게 변화된 데이터를 Push 한다. 마이너 이벤트는 데이터 일관성과 무관한 이벤트로써 사용자들에게 Push 하지 않는다. 제안한 모바일 예약 시스템에서는 예약/취소 요청 이벤트를 메이저 이벤트로 정의하고, 예약/취소 요청 외의 모든 이벤트들을 마이너 이벤트로 정의한다.



(그림 1) 데이터 일관성을 유지하기 위한 Push 방식

3.1 시스템 구조

모바일 예약 시스템에서 서버는 클라이언트의 요청에 대한 응답, 예약을 진행하고 있는 단말기의 세션 관리, 이벤트에 따른 Push 기능이 필요하다. 클라이언트는 안드로이드 단말기 환경으로 요청, Push를 수신하는 기능이 필요하다. 이와 같은 사항을 고려하여 각 기능을 모듈로 나누어 설계하였고 서버와 클라이언트의 모듈 구조도는 (그림 2)와 같다.



(그림 2) 서버와 클라이언트의 모듈 구조도

각 모듈별 설명은 다음과 같다.

- 1) 커뮤니케이션 채널
서버와 클라이언트간의 연결을 담당한다. TCP 방식으로 각 사용자 요청마다 쓰레드를 생성하게 되고 예약 현황을 요청 받게 되면 예약 종료 메시지를 수신 받을 때까지 세션을 유지한다.
- 2) 메시지 분석
수신하는 메시지를 파싱하여 메시지 관리 모듈에 그 정보를 넘겨주는 기능을 한다.
- 3) 메시지 처리
파싱된 정보를 통해 메시지 유효성 체크와 메시지 타입에 맞는 처리를 한다.
- 4) 데이터베이스 관리
메시지 타입에 따라 예약 요청 또는 취소한 정보에 대해 등록/삭제 기능이 있다.
- 5) 이벤트 처리
이벤트 큐를 모니터링 하고, 이벤트를 탐지하게 되면 SD 큐에 있는 소켓 정보를 이용해 Push 하는 기능을 한다.
- 6) 이벤트 큐
메이저 이벤트를 받으면 해당 좌석 정보를 저장한다. 이벤트 큐는 상영 시간대 개수만큼 생성된다.
- 7) SD 큐
예약 현황을 요청받게 되면 해당 소켓 정보를 SD 큐에 저장하고, 예약 종료를 요청받게 되면 해당 소켓 정보를 SD 큐에서 제거한다. SD 큐도 이벤트 큐처럼 상영 시간대 개수만큼 생성된다.

설계한 시스템에서 각 쓰레드마다 이벤트 큐를 생성하지 않고 각 상영시간에 해당하는 이벤트 큐를 공유하면서 사용하는 이유는 다음과 같다. 쓰레드마다 이벤트 큐를 갖게 될 경우 사용자 수에 따라 그만큼의 큐를 생성해야 하고, 다수의 이벤트가 동시에 발생할 경우에는 해당 이벤트를 사용자 수만큼 처리해야 한다. 예를 들어 k명의 사용자가 예약 진행 중이고 메이저 이벤트가 동시에 n개 발생 시 k개의 이벤트 큐에 이벤트를 n번 저장하여야 한다. 즉, 공유된 1개의 이벤트 큐에 저장하는 것보다 k배만큼 처리량이 많아지고 이는 처리시간이 길어진다는 문제점을 발생시킨다.

4. 제안하는 동기화 메커니즘

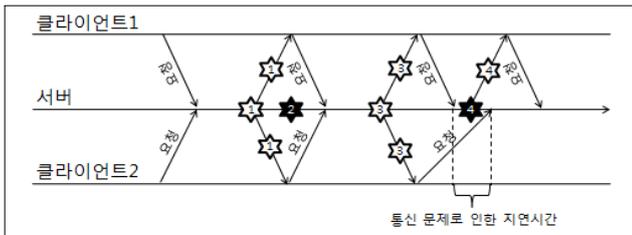
데이터의 일관성을 유지하는데 있어 동기식과 비동기식이 있다. 동기식은 어떠한 이벤트가 발생했을 때, 제한된 시간에 맞춰 이벤트 처리를 가능하게 하는 방식이다[9]. 비동기식은 이벤트 발생 시 곧바로 처리하지 않고 일정 시간이 흐른 뒤에 데이터의 일관성이 유지되는 방식이다.

기존의 모바일 예약 시스템은 비동기식이기 때문에 사용자가 다시 예약 현황을 요청해야 데이터의 일관성을 유지할 수 있다. 기존 예약 시스템이 Push를 사용하지 않은 이유는 서버가 모바일로 접근하기 위해 지속적인 연결이 필요하고 이를 위해 사용자마다 쓰레드를 생성해야 하기

때문이다. 이는 서버 성능과 시스템의 전체적인 효율성 문제점을 갖게 된다[4][7]. 제안 시스템은 Ajax Push 기술처럼 사용자 수만큼 쓰레드를 생성하지만 쓰레드 개수를 제한한다. 즉, 동시 접속자수 제한을 통해 쓰레드 생성에 따른 서버 성능 문제와 잦은 문맥 교환으로 인해 낮아지는 효율 문제를 해결한다. 그리고 기존의 Push 기술을 개선하여 다수의 이벤트를 Push할 때 지연시간 문제점을 해결하고, 동기식 처리로 데이터의 일관성을 유지한다.

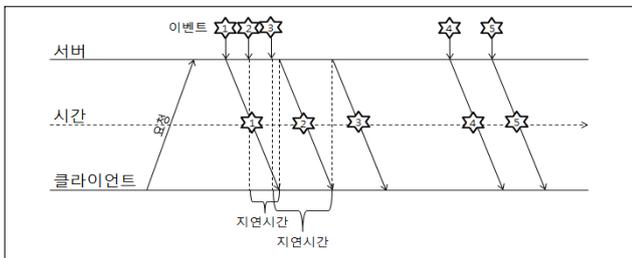
4.1 이벤트 손실과 지연시간을 줄이는 스트리밍 방식 제안

Push 기술 중 하나인 룬 폴링 방식은 이벤트 응답에 따른 요청을 매번 해야 하기 때문에 데이터의 변경이 빈번한 시스템에서는 적합하지 않다[7]. 또한 1개의 이벤트 큐를 공유하면서 사용하기 때문에 제한한 모바일 예약 시스템에서는 (그림 3)과 같은 손실 문제가 생긴다. 손실 문제는 서버가 응답하는 순간부터 클라이언트가 요청할 때까지 잠시 연결이 끊어져 있기 때문에 이와 같은 문제가 발생한다[8]. 따라서 제안 시스템에서는 룬 폴링 방식이 아닌 스트리밍 방식을 사용한다.



(그림 3) 룬 폴링 방식의 문제점

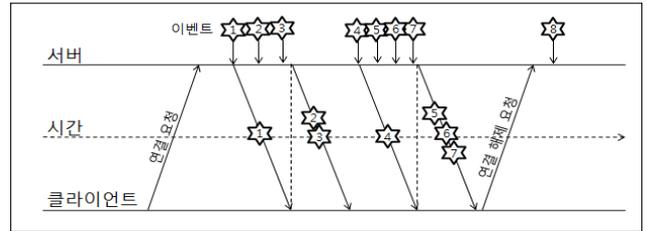
스트리밍 방식은 최초 한번만 요청하고 이벤트가 발생할 때마다 클라이언트에게 Push하게 된다. 그러나 스트리밍 방식은 서버와 클라이언트가 지속적으로 연결되고 있는지의 여부를 알 수 없고, 이벤트가 동시다발적으로 발생할 경우 이벤트들은 지연시간을 갖는다는 문제점이 있다. 지연시간은 이벤트가 발생한 시점부터 이벤트를 처리하는 시점까지의 시간을 말한다. 스트리밍 방식의 지연시간 문제점은 (그림 4)와 같다.



(그림 4) 스트리밍 방식의 문제점

본 논문에서는 스트리밍 방식의 지연시간을 최소화하기 위해 Enhanced 룬 폴링 방식[9]처럼 스트리밍 방식에서 이벤트를 모아서 보내는 방식을 제안한다. 예약 시스템에서는 동일한 형식의 이벤트만 발생하기 때문에 하나의 이벤트가

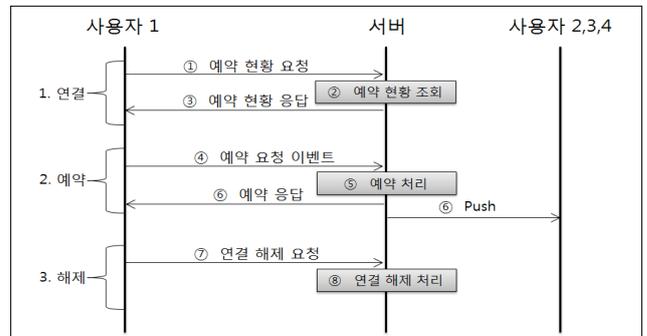
아닌 다수의 이벤트를 한 번에 보냄으로써 서버의 부하를 줄일 수 있고 이벤트 발생에 따른 지연시간을 줄일 수 있다. 또한, 사용자가 예약 종료 시 연결 해제 요청을 보냄으로써 서버는 클라이언트의 정상적인 종료로 알 수 있고 이에 이벤트 발생에 따른 응답을 보내지 않는다. 비정상적인 종료일 경우에는 타임아웃을 기능을 통해 일정 시간동안 예약 또는 취소 요청이 없을 경우 세션을 끊는다. 이를 통해 서버는 사용자의 연결 상태를 알 수 있다. 제안한 스트리밍 방식은 (그림 5)와 같다.



(그림 5) 제안하는 스트리밍 방식

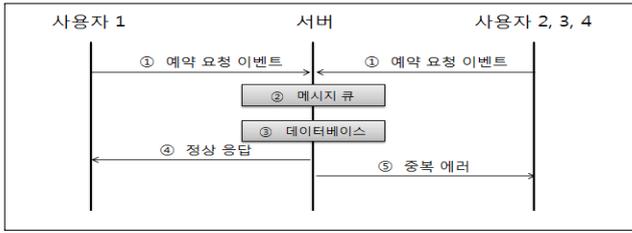
4.2 제안 방식에서의 메시지 흐름

(그림 6)은 제안한 모바일 예약 시스템의 메시지 흐름이다. 연결, 예약, 해제의 총 3단계로 구성된다. 연결 단계는 최초 서버에게 연결하는 단계로써 예약 현황을 요청한다. 이 연결은 사용자가 연결 해제 요청할 때까지 유지되어야 한다. 예약 단계는 예약 요청 이벤트를 발생시킴으로써 예약을 진행 중인 단말기들에게 멀티캐스팅 방식으로 Push 한다. 해제 단계에서는 연결을 해제함으로써 해제를 요청한 단말기는 Push 받지 않는다.



(그림 6) 제안한 모바일 예약 시스템의 메시지 흐름

동시에 여러 단말기가 동일한 좌석을 예약하려고 할 때의 처리과정은 (그림 7)과 같다. 동일한 정보를 담고 있는 예약 요청 이벤트를 동일한 시간에 서버가 받게 될 경우 메시지 큐를 통해 순차적으로 저장하게 된다. 그 후 가장 빠른 요청부터 데이터베이스에 등록되고, 나머지의 요청들은 개체 무결성 규칙에 위배되어 데이터베이스 에러가 발생한다. 요청이 완료된 사용자에게는 정상 응답을 주고, 다른 사용자들에게는 요청한 예약 좌석의 중복 에러를 준다. 이를 통해 사용자는 요청한 좌석을 갱신하게 된다. 따라서 동시에 동일한 좌석을 예약하여도 데이터의 일관성을 유지할 수 있게 된다.



(그림 7) 동시에 동일한 좌석 예약 시 처리과정

5. 검증 결과

데이터 일관성을 유지하는지에 대한 실험 방법은 이벤트 개수에 따라 각각 10회씩 측정하여 평균값을 계산하였다. (그림 8)은 동시에 발생한 이벤트에 따른 평균 Push 처리 시간이다.



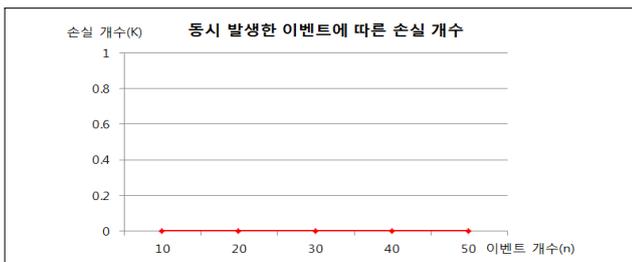
(그림 8) 동시에 발생한 이벤트에 따른 평균 Push 처리시간

(그림 9)는 동시에 발생한 이벤트에 따른 평균 Push 횟수이다. 이 결과를 통해 이벤트를 한 번에 보냄으로써 Push하는 횟수를 줄인 것을 확인할 수 있다.



(그림 9) 동시에 발생한 이벤트에 따른 평균 Push 횟수

(그림 10)은 동시에 발생한 이벤트에 따른 손실 개수이다. 이 결과를 통해 이벤트의 손실 없이 데이터 일관성을 유지하는 것을 검증하였다.



(그림 10) 동시에 발생한 이벤트에 따른 손실 개수

6. 결론

현재 웹 브라우저를 통한 예약 시스템이나 모바일 어플리케이션을 통한 예약시스템 모두 서버와 사용자간에 예약 정보 데이터의 일관성을 유지하지 못한다. 본 논문에서는 이에 대한 문제점을 분석하고 모바일 환경에서 예약 정보 데이터의 일관성 유지하는 동기화 메커니즘을 제안하였다.

모바일 예약 시스템에서는 데이터의 변화가 빈번하기 때문에 롱 폴링 방식 대신 스트리밍 방식을 사용하였다. 이 때 이벤트들을 한 번에 처리함으로써 기존의 스트리밍 방식보다 지연시간을 줄였다. 또한, 정상적인 종료일 경우 연결 해제 요청을 추가하고 비정상적인 종료일 경우에는 타임아웃 기능을 통해 사용자의 연결 유무를 알 수 없었던 기존의 스트리밍 방식 문제점을 해결하였다.

향후 연구 계획으로는 기존의 롱 폴링 방식과 스트리밍 방식을 모바일 예약 시스템에 적용하여 제안 방식과 비교 분석할 것이다. 이를 통해 기존 방식의 문제점들을 검증함과 동시에 제안한 동기화 메커니즘의 지연시간과 Push 횟수를 단축할 수 있을 것이다.

참고문헌

- [1] E. Bozdog, A. Mesbah, "A Comparison of Push and Pull Techniques for Ajax", WSE 2007. 9th IEEE International Workshop on, pp.15-22, 2007.
- [2] G. Pour, "The push to make software engineering respectable" IEEE Computer, vol.33, no.5, pp.35-43, 2000
- [3] Kookhyun Jun, S.J. Moon, Y.H. Eom, Y.K. Kook, K.D. Jung Y.G. Choi, "A Design of Realtime Reservation System Using AJAX", 한국정보과학회, Vol.33, No.2, 2006.
- [4] Uwiki Comet 구현 기법, http://www.uengine.org:8088/wiki/index.php/Comet_%EA%B5%AC%ED%98%84_%EA%B8%B0%EB%B2%95, 2011
- [5] 장재완, 김일환, "3G 모바일 네트워크의 이해", <http://helloworld.naver.com/helloworld/111111>, 2012
- [6] 이원사, "자바 분산 실시간 시스템 연구", 건국대학교 대학원 석사학위 논문, 2010
- [7] 김경윤 이창근, "RTCS 실시간 웹 서비스를 위한 도전", <http://helloworld.naver.com/helloworld/1052>, 2011
- [8] RFC 6202, Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP, <http://tools.ietf.org/html/rfc6202>, 2011
- [9] 김규완, "모바일 환경에서 효율적인 Push 메시지 전달에 관한 연구", 아주대학교 대학원 석사학위 논문, 2012