



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0076954
(43) 공개일자 2011년07월06일

(51) Int. Cl.

H04L 12/56 (2006.01) H04W 74/06 (2009.01)

H04W 52/02 (2009.01) G06F 15/16 (2006.01)

(21) 출원번호 10-2011-7009123

(22) 출원일자(국제출원일자) 2009년09월23일

심사청구일자 2011년04월25일

(85) 번역문제출일자 2011년04월21일

(86) 국제출원번호 PCT/IB2009/006928

(87) 국제공개번호 WO 2010/035108

국제공개일자 2010년04월01일

(30) 우선권주장

12/235,744 2008년09월23일 미국(US)

(71) 출원인

노키아 코포레이션

핀란드핀-02150 에스푸 카일알라덴티에 4

(72) 발명자

리 두

미국 캘리포니아 94303 팔로 알토 플로버 래인
1004

찬드라 우메쉬

미국 캘리포니아 94089 서니베일 포포이즈 베이
터 #디 517

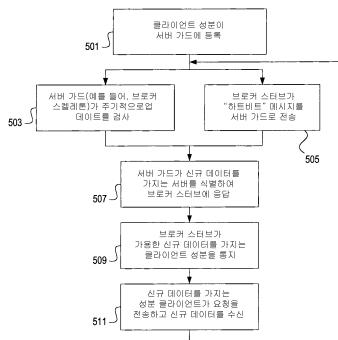
(74) 대리인

리엔목특허법인

전체 청구항 수 : 총 74 항

(54) 저자원 장치에서의 최적화 폴링**(57) 요 약**

이동 클라이언트에 의한 서버 폴링을 최적화함으로써 이동 단말기로 하여금 이동 단말기 내의 프로세서 및 송수신기와 같은 자원을 더 효율적으로 이용하여 배터리 수명을 보전하도록 하는 방법 및 시스템이 개시된다. 브로커 시스템은 폴링을 위하여 이용되는 무선 통신 트래픽을 최소화하는데 이용될 수 있다. 브로커 스터브는 클라이언트에서 서버 폴링 메시지를 인터셉트하고, 서버 요청들을 서로 다중화하며, 다중화된 메시지를 해당 메시지를 적절하게 역다중화 및 전달하는 브로커 스케лет론으로 전달한다. 또한, 폴링은 사용자 동작에 기반하여 동적으로 적용될 수 있고, 또는 서버 가드가 데이터의 변경을 모니터링하여, 서버 가드가 해당 클라이언트에 대한 서버에 존재하는 신규 또는 갱신된 데이터를 검출하면 그 클라이언트로 하여금 자신의 개별 서버를 폴링하도록 통지할 수 있다.

대 표 도 - 도5

특허청구의 범위

청구항 1

장치에 있어서:

상기 장치의 몇 가지 동작을 제어하도록 구성되는 프로세서; 및

컴퓨터 독출가능 명령들을 포함하는 메모리를 포함하며, 상기 컴퓨터 독출가능 명령들은, 상기 프로세서에 의하여 실행되면:

소프트웨어 모듈이 복수 개의 서버 요청 메시지들을 인터셉트(intercepting)하는 동작으로서, 각 서버 요청 메시지는 상기 장치에서 실행되는 상이한 클라이언트 성분으로부터 수신되는 단계;

복수 개의 상기 서버 요청 메시지들을 브로커 요청 메시지(broker request message) 내에 다중화하는 단계; 및

상기 브로커 요청 메시지를 송신하는 단계에 의하여, 상기 장치로 하여금 서버 폴링(server polling)을 실행하도록 야기하는 것을 특징으로 하는 장치.

청구항 2

제1항에 있어서, 상기 명령들은 더 나아가:

브로커 응답 메시지(broker response message)를 수신하는 단계;

상기 브로커 응답 메시지를 역다중화하여 복수 개의 서버 응답 메시지를 획득하는 단계로서, 각 서버 응답 메시지는 상기 장치에서 실행되는 상이한 클라이언트 성분에 상응하는 단계; 및

복수 개의 서버 응답 메시지를 각각을 그 상응하는 클라이언트 성분으로 전달하는 단계를 수행하는 것을 특징으로 하는 장치.

청구항 3

제1항에 있어서, 상기 명령들은 더 나아가:

인터셉트, 다중화, 송신, 수신, 역다중화, 및 전달 단계를 선결된 간격(interval)으로 반복하는 것을 특징으로 하는 장치.

청구항 4

제3항에 있어서,

상기 서버 폴링은 더 나아가 상기 반복 단계를 상기 브로커 요청 메시지에 대한 사전 커넥션(previous connection)이 완결될 때까지 수행 대기함으로써 수행되는 것을 특징으로 하는 장치.

청구항 5

제1항에 있어서,

이동 전화기를 포함하는 것을 특징으로 하는 장치.

청구항 6

제1항에 있어서,

상기 소프트웨어 모듈은 브로커 스타브(broker stub)를 포함하는 것을 특징으로 하는 장치.

청구항 7

제1항에 있어서,

상기 브로커 요청 메시지를 송신하는 단계는 상기 브로커 요청 메시지를 브로커 스켈레톤(broker skeleton)으로

송신하는 단계를 포함하는 것을 특징으로 하는 장치.

청구항 8

제2항에 있어서,

서버 응답 메시지를 중 하나는 폴링 간격(polling interval) 내의 변경을 요청하는 피드백을 포함하고,

상기 명령들은 상기 폴링 간격을 상기 피드백에 기반하여 적응시키는 단계를 더 포함하는 것을 특징으로 하는 장치.

청구항 9

복수 개의 서버 요청 메시지들을 인터셉트하는 단계로서, 각 서버 요청 메시지는 장치에서 실행되는 상이한 클라이언트 성분으로부터 수신되는 단계;

복수 개의 상기 서버 요청 메시지들을 브로커 요청 메시지 내에 다중화하는 단계; 및

상기 브로커 요청 메시지를 송신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 10

제9항에 있어서,

브로커 응답 메시지를 수신하는 단계;

상기 브로커 응답 메시지를 역다중화하여 복수 개의 서버 응답 메시지를 획득하는 단계로서, 각 서버 응답 메시지는 상기 장치에서 실행되는 상이한 클라이언트 성분에 상응하는 단계; 및

복수 개의 서버 응답 메시지를 각각을 그 상응하는 클라이언트 성분으로 전달하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 11

제9항에 있어서,

인터셉트, 다중화, 송신, 수신, 역다중화, 및 전달 단계를 선결된 간격으로 반복하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 12

제11항에 있어서,

상기 반복 단계를 상기 브로커 요청 메시지에 대한 사전 커넥션이 완결될 때까지 수행 대기하는 더 포함하는 것을 특징으로 하는 방법.

청구항 13

제10항에 있어서,

상기 서버 응답 메시지들 중 하나는, 폴링 간격 내의 변경을 요청하는 피드백을 포함하고,

상기 방법은, 상기 폴링 간격을 상기 피드백에 기반하여 적응시키는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 14

컴퓨터 독출가능 명령들을 포함하는 하나 또는 그 이상의 컴퓨터 독출가능 매체에 있어서,

상기 컴퓨터 독출가능 명령들은, 실행되면:

복수 개의 서버 요청 메시지들을 인터셉트하는 단계로서, 각 서버 요청 메시지는 상기 장치에서 실행되는 상이한 클라이언트 성분으로부터 수신되는 단계;

복수 개의 상기 서버 요청 메시지들을 브로커 요청 메시지 내에 다중화하는 단계; 및

상기 브로커 요청 메시지를 송신하는 단계에 의하여 서버 폴링을 수행하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 15

제14항에 있어서, 상기 명령들은:

브로커 응답 메시지를 수신하는 단계;

상기 브로커 응답 메시지를 역다중화하여 복수 개의 서버 응답 메시지를 획득하는 단계로서, 각 서버 응답 메시지는 상이한 클라이언트 성분에 상응하는 단계; 및

복수 개의 서버 응답 메시지를 각각을 그 상응하는 클라이언트 성분으로 전달하는 단계를 더 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 16

제14항에 있어서, 상기 명령들은:

인터넷, 다중화, 송신, 수신, 역다중화, 및 전달 단계를 선결된 간격으로 반복하는 단계를 더 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 17

제16항에 있어서,

상기 반복 단계를 상기 브로커 요청 메시지에 대한 사전 커넥션이 완결될 때까지 수행 대기하는 R을 더 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 18

제14항에 있어서,

소프트웨어 모듈은 브로커 스터브를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 19

제14항에 있어서,

브로커 요청 메시지를 송신하는 상기 단계는, 상기 브로커 요청 메시지를 브로커 스켈레톤으로 송신하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 20

제15항에 있어서,

서버 응답 메시지를 중 하나는 폴링 간격 내의 변경을 요청하는 피드백을 포함하고,

상기 명령들은 상기 폴링 간격을 상기 피드백에 기반하여 적응시키는 단계를 더 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 21

다중화된 브로커 요청 메시지를 수신하는 단계;

상기 브로커 요청 메시지를 복수 개의 성분 요청 메시지(component request message)들로 역다중화하는 단계;

각 성분 요청 메시지를 전달하는 단계;

각 성분 요청 메시지에 상응하는 성분 응답 메시지(component response message)를 수신하는 단계;

상기 성분 응답 메시지들을 브로커 응답 메시지 내에 다중화하는 단계; 및

상기 브로커 응답 메시지를 전송하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 22

제21항에 있어서,
다중화된 상기 브로커 요청 메시지는 이동 단말기로부터 수신되고,
상기 브로커 응답 메시지는 상기 이동 단말기로 전송되는 것을 특징으로 하는 방법.

청구항 23

제21항에 있어서,
각 성분 요청 메시지를 전달하는 상기 단계는, 각 성분 요청 메시지를 상기 성분 요청 메시지 내에서 식별된 서버 또는 성분으로 전달하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 24

제21항에 있어서,
각 성분 요청 메시지에 상응하는 성분 응답 메시지를 수신하는 상기 단계는,
식별된 상기 서버 또는 성분들 각각으로부터 성분 응답 메시지를 수신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 25

제21항에 있어서,
상기 성분 응답 메시지들 중 하나는 풀링 간격 내의 변경을 요청하는 피드백 값을 포함하는 것을 특징으로 하는 방법.

청구항 26

컴퓨터 독출가능 명령들을 포함하는 하나 또는 그 이상의 컴퓨터 독출가능 매체에 있어서,
상기 컴퓨터 독출가능 명령들은, 실행될 때:
다중화된 브로커 요청 메시지를 수신하는 단계;
상기 브로커 요청 메시지를 복수 개의 성분 요청 메시지들로 역다중화하는 단계;
각 성분 요청 메시지를 전달하는 단계;
각 성분 요청 메시지에 상응하는 성분 응답 메시지를 수신하는 단계;
상기 성분 응답 메시지들을 브로커 응답 메시지 내에 다중화하는 단계; 및
상기 브로커 응답 메시지를 전송하는 단계를 수행하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 27

제26항에 있어서,
다중화된 상기 브로커 요청 메시지는 이동 단말기로부터 수신되고,
상기 브로커 응답 메시지는 상기 이동 단말기로 전송되는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 28

제26항에 있어서,
각 성분 요청 메시지를 전달하는 상기 단계는, 각 성분 요청 메시지를 상기 브로커 요청 메시지 내에서 식별된 서버 또는 성분으로 전달하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 29

제29항에 있어서,

각 성분 요청 메시지에 상응하는 성분 응답 메시지를 수신하는 상기 단계는,

식별된 상기 서버 또는 성분들 각각으로부터 성분 응답 메시지를 수신하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 30

제26항에 있어서,

상기 성분 응답 메시지들 중 하나는 폴링 간격 내의 변경을 요청하는 피드백 값을 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 31

장치에 있어서:

상기 장치의 몇 가지 동작을 제어하도록 구성되는 프로세서; 및

컴퓨터 독출가능 명령들을 저장하는 메모리를 포함하며, 상기 명령들은 상기 프로세서에 의하여 실행되면:

다중화된 브로커 요청 메시지를 수신하는 단계;

상기 브로커 요청 메시지를 복수 개의 성분 요청 메시지들로 역다중화하는 단계;

각 성분 요청 메시지를 전달하는 단계;

각 성분 요청 메시지에 상응하는 성분 응답 메시지를 수신하는 단계;

상기 성분 응답 메시지들을 브로커 응답 메시지 내에 다중화하는 단계; 및

상기 브로커 응답 메시지를 전송하는 단계를 수행하는 것을 특징으로 하는 장치.

청구항 32

제31항에 있어서,

다중화된 상기 브로커 요청 메시지는 이동 단말기로부터 수신되고,

상기 브로커 응답 메시지는 상기 이동 단말기로 전송되는 것을 특징으로 하는 장치.

청구항 33

제31항에 있어서,

각 성분 요청 메시지를 전달하는 상기 단계는, 각 성분 요청 메시지를 상기 브로커 요청 메시지 내에서 식별된 서버 또는 성분으로 전달하는 단계를 포함하는 것을 특징으로 하는 장치.

청구항 34

제31항에 있어서,

각 성분 요청 메시지에 상응하는 성분 응답 메시지를 수신하는 상기 단계는,

식별된 상기 서버 또는 성분들 각각으로부터 성분 응답 메시지를 수신하는 단계를 포함하는 것을 특징으로 하는 장치.

청구항 35

제31항에 있어서,

상기 성분 응답 메시지들 중 하나는 폴링 간격 내의 변경을 요청하는 피드백 값을 포함하는 것을 특징으로 하는 장치.

청구항 36

장치에 있어서:

상기 장치의 몇 가지 동작을 제어하도록 구성되는 프로세서; 및

컴퓨터 독출가능 명령들을 저장하는 메모리를 포함하며, 상기 명령들은 상기 프로세서에 의하여 실행되면:

클라이언트 성분을 실행하여 사용자 상호작용 기준(user interaction criterion)이 만족되는 제1 속도에서 서버를 폴링(poll)하도록 하는 단계로서, 상기 서버는 상기 클라이언트 성분에 대한 데이터를 제공하는 단계; 및

상기 클라이언트 성분을 실행하여 상기 사용자 상호작용 기준이 만족되지 않고 상기 제1 속도와 상이한 제2 속도에서 상기 서버를 폴링하도록 하는 단계에 의하여,

상기 장치가 서버 폴링을 수행하도록 구성하는 것을 특징으로 하는 장치.

청구항 37

제36항에 있어서,

상기 사용자 상호작용 기준은 상기 장치의 디스플레이 스크린 상에 디스플레이되는 클라이언트 성분을 포함하는 것을 특징으로 하는 장치.

청구항 38

제36항에 있어서,

상기 사용자 상호작용 기준은, 상기 클라이언트 성분이 상기 장치 상에서 실행되는 제2 클라이언트 성분보다 더 높은 레벨의 현저함(prominence)으로 상기 디스플레이 스크린 상에 디스플레이되는 것을 포함하는 것을 특징으로 하는 장치.

청구항 39

제36항에 있어서,

상기 제1 속도는 제2 속도보다 빠른 것을 특징으로 하는 장치.

청구항 40

제36항에 있어서,

이동 전화기를 포함하는 것을 특징으로 하는 장치.

청구항 41

클라이언트 성분을 실행하는 단계;

상기 클라이언트 성분이 사용자 상호작용 기준이 만족되는 제1 속도에서 서버를 폴링하는 단계로서, 상기 서버는 상기 클라이언트 성분에 대한 데이터를 제공하는 단계; 및

상기 클라이언트 성분이 상기 사용자 상호작용 기준이 만족되지 않고 상기 제1 속도와 상이한 제2 속도에서 상기 서버를 폴링하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 42

제41항에 있어서,

상기 사용자 상호작용 기준은 사용자 인터페이스 상에 디스플레이되는 클라이언트 성분을 포함하는 것을 특징으로 하는 방법.

청구항 43

제41항에 있어서,

제2 클라이언트 성분을 실행하는 단계를 더 포함하며,

상기 사용자 상호작용 기준은, 상기 클라이언트 성분이 상기 장치 상에서 실행되는 제2 클라이언트 성분보다 더

높은 레벨의 현저함으로 상기 디스플레이 스크린 상에 디스플레이되는 것을 포함하는 것을 특징으로 하는 방법.

청구항 44

제41항에 있어서,

상기 제1 속도는 제2 속도보다 빠른 것을 특징으로 하는 방법.

청구항 45

제41항에 있어서,

클라이언트 성분을 실행하는 상기 단계는, 상기 클라이언트 성분을 이동 전화기를 포함하는 장치에서 실행하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 46

제42항에 있어서,

상기 사용자 인터페이스를 상기 이동 전화기의 디스플레이 스크린 상에 디스플레이하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 47

컴퓨터 독출가능 명령들을 포함하는 하나 또는 그 이상의 컴퓨터 독출가능 매체에 있어서,

상기 명령들은 실행되면:

클라이언트 성분을 실행하여 사용자 상호작용 기준이 만족되는 제1 속도에서 서버를 폴링하는 단계로서, 상기 서버는 상기 클라이언트 성분에 대한 데이터를 제공하는 단계; 및

상기 클라이언트 성분을 실행하여 상기 사용자 상호작용 기준이 만족되지 않고 상기 제1 속도와 상이한 제2 속도에서 상기 서버를 폴링하는 단계에 의하여 서버 폴링을 수행하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 48

제47항에 있어서,

상기 사용자 상호작용 기준은 장치의 디스플레이 스크린 상에 디스플레이되는 클라이언트 성분을 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 49

제47항에 있어서,

상기 사용자 상호작용 기준은, 상기 클라이언트 성분이 상기 장치 상에서 실행되는 제2 클라이언트 성분보다 더 높은 레벨의 현저함으로 상기 디스플레이 스크린 상에 디스플레이되는 것을 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 50

제47항에 있어서,

상기 제1 속도는 제2 속도보다 빠른 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 51

장치에 있어서:

상기 장치의 몇 가지 동작을 제어하도록 구성되는 프로세서; 및

컴퓨터 독출가능 명령들을 저장하는 메모리를 포함하며,

상기 명령들은 상기 프로세서에 의해 실행되면:

등록 메시지(registration message)를 수신하는 단계로서, 상기 등록 메시지는 클라이언트 성분에 상응하는 서버 성분에 대한 서버 정보를 제공하는 단계;

상기 서버 정보를 데이터베이스에 등록하는 단계;

선결된 스케줄에 따라서 상기 서버 성분을 폴링하여, 상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가지는지 여부를 결정하는 단계; 및

상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가진다면, 상기 서버가 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가지고 있다는 것을 표시하며 상기 클라이언트 성분으로 어드레싱된 (addressed) 메시지를 전송하는 단계에 의하여, 상기 장치를 서버 가드(server guard)로서 동작하도록 구성하는 것을 특징으로 하는 장치.

청구항 52

제51항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터(invariant data)를 포함하며, 상기 클라이언트 성분에 기반한 가변 데이터(variant data)를 포함하는 것을 특징으로 하는 장치.

청구항 53

제51항에 있어서,

상기 서버 정보는 상기 서버에 고유한 불변 데이터만을 포함하며, 상기 장치는:

상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 포함한다면, 상기 서버 성분으로 질의(query)를 전송하는 단계로서, 상기 질의는 상기 불변 데이터를 포함하는 단계; 및

상기 메시지 내에서 상기 클라이언트 성분으로, 상기 질의에 응답하여 상기 서버 성분으로부터 수신되는 모든 데이터를 전달하는 단계를 수행하도록 더욱 구성되는 것을 특징으로 하는 장치.

청구항 54

제51항에 있어서,

상기 선결된 스케줄은 상기 클라이언트 성분으로부터 수신되는 하트비트 메시지(heartbeat message)에 기반하는 것을 특징으로 하는 장치.

청구항 55

컴퓨터 독출가능 명령들을 저장하는 하나 또는 그 이상의 컴퓨터 독출가능 매체에 있어서,

상기 명령들은 실행되면:

등록 메시지를 수신하는 단계로서, 상기 등록 메시지는 클라이언트 성분에 상응하는 서버 성분에 대한 서버 정보를 제공하는 단계;

상기 서버 정보를 데이터베이스에 등록하는 단계;

선결된 스케줄에 따라서 상기 서버 성분을 폴링하여, 상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가지는지 여부를 결정하는 단계; 및

상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가진다면, 상기 서버가 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가지고 있다는 것을 표시하며 상기 클라이언트 성분으로 어드레싱된 메시지를 전송하는 단계를 수행하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 56

제55항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터를 포함하며, 상기 클라이언트 성분에 기반한 가변 데이터를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 57

제55항에 있어서,

상기 서버 정보는 상기 서버에 고유한 불변 데이터만을 포함하며, 상기 장치는:

상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 포함한다면, 상기 서버 성분으로 질의를 전송하는 단계로서, 상기 질의는 상기 불변 데이터를 포함하는 단계; 및

상기 메시지 내에서 상기 클라이언트 성분으로, 상기 질의에 응답하여 상기 서버 성분으로부터 수신되는 모든 데이터를 전달하는 단계를 수행하도록 더욱 구성되는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 58

제55항에 있어서,

상기 선결된 스케줄은 상기 클라이언트 성분으로부터 수신되는 하트비트 메시지에 기반하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 59

등록 메시지를 수신하는 단계로서, 상기 등록 메시지는 클라이언트 성분에 상응하는 서버 성분에 대한 서버 정보를 제공하는 단계;

상기 서버 정보를 데이터베이스에 등록하는 단계;

선결된 스케줄에 따라서 상기 서버 성분을 폴링하여, 상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가지는지 여부를 결정하는 단계; 및

상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가진다면, 상기 서버가 상기 클라이언트 성분에 대해 의도된 신규 데이터를 가지고 있다는 것을 표시하며 상기 클라이언트 성분으로 어드레싱된 메시지를 전송하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 60

제59항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터를 포함하며, 상기 클라이언트 성분에 기반한 가변 데이터를 포함하는 것을 특징으로 하는 방법.

청구항 61

제59항에 있어서,

상기 서버 정보는 상기 서버에 고유한 불변 데이터만을 포함하며, 상기 장치는:

상기 서버 성분이 상기 클라이언트 성분에 대해 의도된 신규 데이터를 포함한다면, 상기 서버 성분으로 질의를 전송하는 단계로서, 상기 질의는 상기 불변 데이터를 포함하는 단계; 및

상기 메시지 내에서 상기 클라이언트 성분으로, 상기 질의에 응답하여 상기 서버 성분으로부터 수신되는 모든 데이터를 전달하는 단계를 수행하도록 더욱 구성되는 것을 특징으로 하는 방법.

청구항 62

제59항에 있어서,

상기 선결된 스케줄은 상기 클라이언트 성분으로부터 수신되는 하트비트 메시지에 기반하는 것을 특징으로 하는 방법.

청구항 63

장치에 있어서:

프로세서; 및

컴퓨터 실행가능 명령들을 저장하는 메모리를 포함하며, 상기 명령들은 상기 프로세서에 의해 실행되면:

서버 가드 모듈(server guard module)로 어드레싱된 등록 메시지를 무선으로 전송하는 단계로서, 상기 등록 메시지는 상기 장치에서 실행되는 클라이언트 성분으로 데이터를 제공하는 서버 성분에 상응하는 서버 정보를 포함하는 단계;

상기 서버 가드 모듈로 어드레싱된 복수 개의 하트비트 메시지들을 연결된 스케줄에 따라서 전송하는 단계; 및
상기 하트비트 메시지들 중 하나에 대한 응답을 수신하는 단계로서, 상기 응답은 상기 서버 성분이 상기 장치가 아직 수신하지 않은 신규 데이터를 가지고 있음을 표시하는 단계에 의하여,
상기 장치가 서버를 폴링하도록 구성하는 것을 특징으로 하는 장치.

청구항 64

제63항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터 및 상기 장치에 기반한 가변 데이터를 포함하고, 및
상기 장치는,

상기 응답의 수신에 응답하여, 상기 불변 및 가변 데이터 모두에 기반하는 폴링 메시지를 상기 서버 성분으로 전송함으로써, 상기 서버 성분을 폴링하도록 더욱 구성되는 것을 특징으로 하는 장치.

청구항 65

제63항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터만을 포함하고, 및

상기 서버 가드 모듈로부터 수신된 상기 응답은, 상기 서버 정보에 기반하여 질의에 응답해 상기 서버 성분에 의하여 제공된 데이터를 포함하는 것을 특징으로 하는 장치.

청구항 66

제63항에 있어서,

응답을 수신하는 상기 단계는, 응답을 상기 서버 가드 모듈로부터 수신하는 단계를 포함하는 것을 특징으로 하는 장치.

청구항 67

서버 가드 모듈로 어드레싱된 등록 메시지를 무선으로 전송하는 단계로서, 상기 등록 메시지는 클라이언트 성분으로 데이터를 제공하는 서버 성분에 상응하는 서버 정보를 포함하는 단계;

상기 서버 가드 모듈로 어드레싱된 복수 개의 하트비트 메시지들을 연결된 스케줄에 따라서 전송하는 단계; 및
상기 하트비트 메시지들 중 하나에 대한 응답을 수신하는 단계로서, 상기 응답은 상기 서버 성분이 상기 클라이언트 성분이 아직 수신하지 않은 신규 데이터를 가지고 있음을 표시하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 68

제67항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터 및 장치에 기반한 가변 데이터를 포함하고, 및

상기 방법은,

상기 응답의 수신에 응답하여, 상기 불변 및 가변 데이터 모두에 기반하는 폴링 메시지를 상기 서버 성분으로 전송하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 69

제67항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터만을 포함하고, 및

상기 서버 가드 모듈로부터 수신된 상기 응답은, 상기 서버 정보에 기반하여 질의에 응답해 상기 서버 성분에 의하여 제공된 데이터를 포함하는 것을 특징으로 하는 방법.

청구항 70

제67항에 있어서,

응답을 수신하는 상기 단계는, 응답을 상기 서버 가드 모듈로부터 수신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 71

컴퓨터 실행가능 명령들을 포함하는 하나 또는 그 이상의 컴퓨터 독출가능 매체에 있어서,

상기 명령들은 실행되면:

서버 가드 모듈로 어드레싱된 등록 메시지를 무선으로 전송하는 단계로서, 상기 등록 메시지는 장치 상에서 동작하는 클라이언트 성분으로 데이터를 제공하는 서버 성분에 상응하는 서버 정보를 포함하는 단계;

상기 서버 가드 모듈로 어드레싱된 복수 개의 하트비트 메시지들을 연결된 스케줄에 따라서 전송하는 단계; 및

상기 하트비트 메시지들 중 하나에 대한 응답을 수신하는 단계로서, 상기 응답은 상기 서버 성분이 상기 장치가 아직 수신하지 않은 신규 데이터를 가지고 있음을 표시하는 단계를 수행하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 72

제71항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터 및 상기 장치에 기반한 가변 데이터를 포함하고, 및

상기 장치는, 응답의 수신에 기반하여, 상기 불변 및 가변 데이터 모두에 기반하는 폴링 메시지를 상기 서버 성분으로 전송함으로써, 상기 서버 성분을 폴링하도록 더욱 구성되는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 73

제71항에 있어서,

상기 서버 정보는 상기 서버 성분에 고유한 불변 데이터만을 포함하고, 및

상기 서버 가드 모듈로부터 수신된 상기 응답은, 상기 서버 정보에 기반하여 질의에 응답해 상기 서버 성분에 의하여 제공된 데이터를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

청구항 74

제71항에 있어서,

응답을 수신하는 상기 단계는, 응답을 상기 서버 가드 모듈로부터 수신하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 독출가능 매체.

명세서**기술분야**

[0001] 본 발명은 이를테면 휴대폰, 스마트폰, 개인 휴대 정보 단말기, UMPC(ultra-mobile PC)와 같은 이동식 저자원 장치들에서 자원을 보호하는데 일반적으로 관련된다. 특히, 본 발명은 클라이언트 및 그 서버 어플리케이션 간의 폴링을 최적화하여 클라이언트 및 서버 간의 활동적이고 정확한 유지하는 데 필요한 오버헤드를 감소시키는

기술을 제공한다.

배경기술

[0002] 이를테면 스마트폰 및 UMPC와 같은 지능형 이동식 연산 장치들은 사회 및 세계 전체에 걸쳐 점점 편재하게 되었다. 어떤 사용자들은 우선적으로 이동 장치를 가끔씩 이용하는데, 예를 들어 이러한 사용자가 전용 전력 및 유선 연결된 네 연결을 가질 수 있는 더 전형적인 컴퓨터로의 액세스를 가지지 않는 경우 공항이나 식당 내에서만 가끔 이용한다. 다른 몇몇 사용자들은 이동 연산 장치들을 그들의 주된 데이터 처리 장치로서 의존하는데, 그 이유는 이러한 사용자들은 그들의 생계 또는 전문적 필요성에 대해 더 일반적인 컴퓨터를 가지지 않거나 심지어는 필요로 하지 않기 때문이다. 이동 장치들은 통신용 무선 기술 및 전력용 배터리를 이용한다. 이들은 강력한 정보 단말기로서 구동하기 충분할 정도로 복잡해지고 있으며, 이들의 한계는 오직 배터리가 재충전되기 이전에 이동 장치의 배터리의 지속 시간뿐이다.

[0003] 예를 들어 Web과 같이 이동 장치와 관련된 몇 가지 환경에서, 서버는 보안 및 서버 확장성(server scalability)을 고려하지 않고는 클라이언트로의 통신을 개시할 수 없고, 또한 이러한 서버는 클라이언트와의 꽤 장시간 연결을 유지할 수도 없다. 그 대신에, 각 클라이언트는 데이터를 서버로 전송하거나 서버로부터 데이터를 수신하여 위하여 서버와의 연결을 개시 또는 설립(establish)해야 한다. 또한, 각 클라이언트는 신규하거나 생성된 데이터에 대하여 서버를 주기적으로 폴링해야 하는데, 바람직하게는 자주 폴링함으로써, 클라이언트가 가장 최근의 정보 및 데이터를 가지고 있도록 보장해야 한다. 그러므로, 클라이언트-서버 통신은 자원 소비원(resource hog)이 될 수 있고 성능의 병목 현상이 될 수 있고, 이에 따라 이동 장치의 배터리가 신속히 닳도록 야기할 수 있다.

발명의 내용

해결하려는 과제

[0004] 이하, 본 명세서에 설명된 다양한 측면들에 대한 기본적인 이해를 제공하기 위하여 본 발명에 대한 간략화된 개요가 제공된다. 이러한 개요는 본 발명에 대한 광범위한 개요가 아니며, 본 발명의 중요하거나 주된 구성 요소를 식별하거나, 본 발명의 기술적 범위를 한정하려는 의도를 가지는 것이 아니다. 후술되는 개요는 단지 후술되는 상세한 설명에 대한 전제로서 간략화된 형태로서 본 발명의 몇 가지 개념을 제공할 뿐이다.

과제의 해결 수단

[0005] 전술된 종래 기술의 한계를 극복하고, 본 명세서를 읽고 이해하는 동안에 명백해질 다른 한계들을 극복하기 위하여, 본 발명은 이동 클라이언트 및 그들이 데이터를 획득하는 개별 서버들 간의 클라이언트-서버 통신을 더 효율적으로 관리하는데 관한 것이다.

[0006] 본 발명의 제일의 양태는 이동 클라이언트(예를 들면, 이동 단말기) 및 이동 클라이언트에서 실행중인 하나 또는 그 이상의 상응하는 클라이언트 성분으로 데이터를 제공하는 하나 또는 그 이상의 서버들 간의 브로커-관리(broker-managed) 클라이언트-서버 통신을 제공한다. 이동 클라이언트에서 실행되는 브로커 모듈("스터브")은 이동 클라이언트에서 실행되는 어떤 클라이언트 성분에 의하여 전송된 서버 요청 메시지를 인터셉트한다. 브로커 스터브는 서버 요청 메시지를 브로커 요청 메시지 내로 다중화하고, 서버에서 실행되는 브로커 모듈 스켈레톤에 의해 수신되도록 브로커 요청 메시지를 송신한다.

[0007] 브로커 스터브가 브로커 스켈레톤으로부터 응답을 수신하면, 브로커 스터브는 브로커 응답 메시지를 이산 서버 응답 메시지로 역다중화하는데, 각 서버 응답 메시지는 장치에서 실행되는 상이한 클라이언트 성분에 상응한다.

[0008] 본 발명의 다른 양태에 따르면, 브로커 스켈레톤(예를 들면, 웹 서버에서 실행되는)이 이동 단말기로부터 다중화된 브로커 요청 메시지를 수신하면, 브로커 스켈레톤은 브로커 요청 메시지를 이산 서버 요청 메시지로 역다중화하고, 각 브로커 요청 메시지를 각 브로커 요청 메시지 내에서 식별된 서버로 전달한다. 브로커 스켈레톤이 식별된 서버의 각각으로부터 서버 응답 메시지를 수신하면, 스켈레톤은 단일 브로커 응답 메시지로 서버 응답 메시지를 다중화하고, 브로커 응답 메시지를 이동 단말기에 송신한다. 브로커 스켈레톤과 브로커 스터브 사이에서 전달된 단일 메시지는 다양한 패킷 또는 버스트(burst) 내에서 송신될 수 있지만, 논리적으로는 단일 통신에 상응할 수 있다.

[0009]

본 발명의 다른 양태에 따르면, 이동 단말기에서 실행되는 하나 이상의 이동 클라이언트는 이동 클라이언트 상의 각 어플리케이션을 이용한 사용자 동작에 기반한 적응적 폴링(adaptive polling)을 수행할 수 있다. 이동 단말기는, 사용자 상호작용 기준이 만족되면 클라이언트를 위해 데이터를 제공하고 있는 서버에 특정 간격(interval)으로 폴링하도록 클라이언트 성분을 실행할 수 있다. 이동 단말기는, 사용자 상호작용 기준이 만족되지 않으면, 상기 제1 간격과 다른 간격으로 서버에 폴링하도록 클라이언트 성분을 실행할 수 있다. 일 실시 예에서, 사용자 상호작용 기준은 클라이언트 성분이 장치의 디스플레이 스크린에 표시될 것을 포함할 수 있다. 다른 예에서 사용자 상호작용 기준은 이동 단말기에서 클라이언트 성분이 이동 단말기에서 실행되는 제2 클라이언트 성분보다 높은 수준의 현저함으로 디스플레이 스크린에 표시되는 것을 포함할 수 있다.

[0010]

본 발명의 다른 양태에 따르면, 서버 가드 모듈은 생성된 데이터가 있는지를 확인하기 위하여 하나 이상의 서버를 독립적으로 감시하기 위해 사용될 수 있다. 서버 가드 모듈은 웹 서버 또는 직접 전력 연결 및 물리적으로 접속되는 네트워크 접속을 가지고 있는 다른 데이터 처리 장치에서 실행될 수 있다. 그러나, 서버 가드가 상시 전력원 또는 직류 전력원을 가지는 장치에 위치하는 것에 비하여 절약되는 자원 및 얻어지는 효율성이 더 크다면, 서버 가드 모듈은 대안적으로 이동 단말기에 위치할 수 있다. 서버 가드 모듈은 이동 단말기에서 실행되는 클라이언트 성분으로부터 등록 메시지를 수신한다. 각 등록 메시지는 서버에게 서버 가드 정보, 이를테면 클라이언트 성분에 상응하는 서버의 주소 및 서버로부터 클라이언트 성분이 정보를 검색하기 위하여 이용할 질의 파라미터(query parameter)들을 제공한다. 서버 가드는 데이터베이스에서 정보를 등록한다. 각 서버 성분이 그 상응한 클라이언트 성분에 대해 의도된 신규 데이터를 가지는지 여부를 결정하기 위하여, 서버 가드는 선결된 스케줄에 따라서 각 서버 성분에게 주기적으로 폴링하거나, 또는 서버에 대해 수행된 신규 업데이트들이 서버 가드에 통지된다. 서버 성분이 그 상응한 클라이언트 성분에 대해 의도된 신규 데이터를 가지면, 서버 가드는 서버가 해당 클라이언트 성분에 대해 의도된 신규 데이터를 가지고 있다는 것을 상응하는 클라이언트 성분에게 표시하여 통지한다.

[0011]

본 발명의 또 다른 양태에 따르면, 이동 단말기는 서버 가드와 상호 작용하도록 적응될 수 있다. 이동 단말기의 각 클라이언트 성분은 등록 메시지를 서버 가드에게 송신한다. 각 등록 메시지는 서버 성분에게 데이터를 해당 메시지를 송신하는 클라이언트 성분으로 제공하는 서버에 상응하여 정보를 제공한다. 이동 단말기는 미리 정의되는 스케줄대로 서버 가드 모듈에 복수의 하트비트 메시지를 후속 송신하는데, 각 메시지는 새로운 데이터의 가용성(availability)에 관한 상태를 요청한다. 이동 단말기는 서버 가드 모듈로부터 응답을 수신하는데, 여기서의 응답은 하트비트 메시지 중의 하나에 응답하며, 이동 단말기가 아직 수신하지 않은 신규 데이터를 서버가 가지고 있다는 것을 표시한다.

[0012]

일 실시예에서, 이동 단말기 상의 클라이언트 성분은 후속하여 신규 데이터를 얻기 위하여 모든 필요한 질의 데이터(예를 들어, 해당 질의에 관련되는 이동 단말기의 현재 위치)를 제공하는 폴링 메시지를 자신의 상응하는 서버로 전송할 수 있다. 또는, 서버가 클라이언트 성분에 특이하거나 이에 기반한 질의 파라미터를 요구하지 않으면(예를 들어, 서버는 서버에 질의하는 것이 누구인지와 무관하게 단지 그리니치 평균 시간만을 제공한다), 서버 가드로부터 수신되는 응답은 직접적으로 서버에 의하여 제공되는 신규 데이터를 포함할 수 있다.

발명의 효과

[0013]

전술된 바와 같은 본 발명의 양태들은 폴링 요청의 개수 및 이동 클라이언트로부터 그 상응하는 서버로 전송되는 총 바이트수를 감소시킨다. 또한, 전술된 바와 같은 본 발명의 양태들은 클라이언트 장치의 CPU 처리, 메모리, 비트율, 및 배터리 수명을 보전한다. 또한, 본 발명의 양태들은 서버 부하를 감소시키고 다양한 이동 인터넷 서비스의 성능을 개선한다.

도면의 간단한 설명

[0014]

본 발명 및 본 발명의 장점들에 대한 더 완전한 이해는, 유사한 부재 번호가 유사한 특징들을 가리키는 첨부 도면을 고려하여 후술되는 상세한 설명을 참조함으로써 얻어질 수 있을 것이다.

도 1은 본 발명의 하나 이상의 실례가 되는 양태에 따라 사용될 수 있는 네트워크 구성을 도시한다.

도 2는 본 발명의 하나 이상의 실례가 되는 양태에 따라 사용될 수 있는 이동 단말기를 도시한다.

도 3은 본 발명의 하나 이상의 실례가 되는 양태에 따라 협력 클라이언트(collaboration client)와 협력 서버(collaboration server) 사이에서의 데이터 흐름을 도시한다.

도 4는 본 발명의 하나 이상의 실례가 되는 양태에 따른 브로커-관리(broker-managed) 서버 폴링의 방법에 대한 흐름도를 도시한다.

도 5는 본 발명의 하나 이상의 실례가 되는 양태에 따른 서버 가드-관리(server guard-managed) 서버 폴링의 방법에 대한 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0015]

다양한 실시예들에 대한 후술되는 상세한 설명에서, 본 명세서의 일부를 구성하는 첨부 도면들을 참조하는데, 이 도면에서는 본 발명이 실시될 수 있는 다양한 실시예들이 예시적으로 도시된다. 본 발명의 기술적 사상에서 벗어나지 않는 범위에서 다른 실시예들도 이용될 수 있으며, 구조적 및 기능적 수정이 이루어질 수 있다는 점이 이해되어야 한다.

[0016]

도 1은 본 발명의 창조적인 기술적 사상들이 실시될 수 있는 예시적인 통신 네트워크를 도시한다. 이동 통신 장치(105 및 110), 개인 디지털 보조(PDA)(120), 개인용 컴퓨터(PC, 115), 서비스 제공자(125) 및 콘텐츠 제공자(130)를 포함하는 복수 개의 컴퓨터 및 장치들은 네트워크(100)를 통해 서로 통신하고 다른 장치들과 통신할 수 있다. 네트워크(100)는 유선과 무선 접속을 포함할 수 있고, 네트워크를 통한 연결은 영구적 또는 일시적인 접속을 포함할 수 있다. 네트워크(100)를 통한 통신은 도시된 장치에 제한되지 않고, 추가적인 장치, 이를테면 홈 비디오 저장 시스템, 휴대용 오디오/비디오 플레이어, 디지털 카메라/캠코더, 위치 결정 장치, 이를테면 GPS(Global Positioning System) 장치 또는 위성, 이동식 텔레비전, 세트 탑 박스(STB), 디지털 비디오 리코더, 리모트 컨트롤 장치와 어떤 조합과 같은 추가적인 장치들을 포함할 수 있다.

[0017]

비록 간략화를 위해 도 1에서는 단일 네트워크로서 도시되었지만, 네트워크(100)는 상 인터넷 작업된 (internetworked) 통신을 제공하기 위해 연결된 다양한 네트워크를 포함할 수 있다. 이러한 네트워크는 하나 이상의 개인용 또는 공공 패킷 교환(packet-switched) 네트워크(예를 들면, 인터넷), 하나 또는 그 이상의 개인용 또는 공공 회선 교환(circuit-switched) 네트워크(예를 들면, 공중 교환 전화망), 이동 통신 장치(105와 110)로의 통신 및 그로부터의 통신을 용이하도록 구성되는 셀룰러 네트워크(예를 들면, 기지국, 이동 전화 교환국, 등의 사용을 통하여), 근거리 또는 중거리 무선 통신 연결(예를 들면, Bluetooth^(R), 초광대역(UWB), 적외선, WiBree, IEEE 표준 번호 802.11의 하나 이상의 버전에 따르는 무선 근거리 통신망(WLAN)), 또는 EV-DO(Evolution-Data Optimized) 네트워크, UMTS(Universal Mobile Telecommunications System) 네트워크, Long Term Evolution (LTE) 네트워크, 또는 EDGE(networks or Enhanced Data rates for GSM Evolution) 네트워크와 같은 고속 무선 데이터 네트워크를 포함할 수 있다. 장치들(105-120)은 당업계에 알려진 여러 가지 기술들 중에서 IP(Internet Protocol), TCP(Transmission Control Protocol), SMTP(Simple Mail Transfer Protocol)와 같은 다양한 통신 프로토콜을 이용할 수 있다. SMS(Short Messaging Service_와 같은 다양한 메시징 서비스도 역시 포함될 수 있다.

[0018]

장치들(105-120)은 이를테면 콘텐츠 서버(130) 또는 서비스 제공자(125)와 상호 작용하기 위해 구성될 수 있다. 일 실시예에서, 이동 장치(110)는 콘텐츠 제공자/서버(130)로의 송신 및 그로부터의 수신을 조정(coordinate)하도록 구성되는 클라이언트 소프트웨어(165)를 포함할 수 있다. 일 구현예에서, 클라이언트 소프트웨어(165)는 콘텐츠 서버(130)로부터 콘텐츠를 요청 및 수신하기 위한 어플리케이션 또는 서버 특이적 프로토콜을 포함할 수 있다. 예를 들면, 클라이언트 소프트웨어(165)는 웹 브라우저 또는 이들의 이동식으로 변형된 변형물을 포함할 수 있으며, 콘텐츠 제공자/서버(130)는 웹 서버를 포함할 수 있다. 또한, 접속 요금 또는 제공된 서비스의 요금을 청구하기 위해 청구 서비스(미도시)가 또한 포함될 수 있다. 서비스 제공자(125)가 셀룰러 네트워크 액세스(예를 들면, 무선 서비스 제공자)를 제공하는 일 구현예에서, 클라이언트 소프트웨어(165)는 셀룰러 네트워크를 통하여 액세스 및 통신하기 위한 명령어를 포함할 수 있다. 클라이언트 소프트웨어(165)는 장치(110) 내의 독출 전용 또는 임의 접근 메모리와 같은 컴퓨터 독출 메모리(160) 내에 저장될 수 있으며, 장치(110)의 하나 이상의 성분(예를 들면, 프로세서(155), 트랜시버 및 디스플레이)으로 하여금 본 명세서에서 설명되는 것을 포함한 다양한 기능 및 방법을 수행하도록 야기하는 명령어들을 포함할 수 있다.

[0019]

도 2는 컴퓨팅 장치, 이를테면 도 1의 네트워크(100)에서 사용될 수 있는 이동 장치(212)를 도시한다. 이동 장치(212)는 사용자 인터페이스 콘트롤(230)에 연결되는 콘트롤러(225), 디스플레이(236) 및 도시된 바와 같은 다른 요소들을 포함할 수 있다. 콘트롤러(225)는 소프트웨어(240)를 저장하는 하나 이상의 프로세서(228)와 메모리(234)를 포함할 수 있다. 또한, 이동 장치(212)는 배터리(250), 스피커(252) 및 안테나(254)를 포함할 수 있다. 사용자 인터페이스 콘트롤(230)은 키패드, 터치 스크린, 음성 인터페이스(예를 들면 마이크로폰(256)을 경

유하는), 기능 키, 조이스틱, 데이터 글러브(data glove), 마우스 등으로부터의 입력을 수신하거나 이들로 출력을 제공하도록 적응되는 콘트롤러 또는 어댑터들을 포함할 수 있다.

[0020] 이동 장치(212)의 프로세서(228) 및 다른 성분에 의해 사용되는 컴퓨터 실행가능 명령어와 데이터는 저장 시설, 이를테면 메모리(234)에 보관될 수 있다. 메모리(234)는 휘발성과 비휘발성 메모리, 이를테면 디스크를 포함하는 독출 전용 메모리(ROM) 모듈 또는 임의 접근 메모리 모듈의 모든 타입을 포함할 수 있다. 소프트웨어(240)는 메모리(234)내에 저장되어 프로세서(228)에게 명령어를 제공함으로써, 명령어들이 실행되면, 프로세서(228), 이동 장치(212) 및/또는 이동 장치(212)의 다른 성분이 각종 기능 또는 방법, 이를테면 본 명세서에서 기술되는 것들을 수행하도록 야기한다. 소프트웨어는 어플리케이션 및 운영체제 소프트웨어를 포함할 수 있고, 코드 세그먼트(code segment), 명령어, 애플릿(applets), 사전 컴파일(pre-compiled) 코드, 컴파일(complied) 코드, 컴퓨터 프로그램, 프로그램 모듈, 엔진, 프로그램 로직 및 이들의 조합을 포함할 수 있다. 더 나아가, 컴퓨터 실행가능 명령어와 데이터는 EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD 또는 자기 다른 광디스크 저장소, 자기 카세트, 자기 테이프, 자기 저장소 등과 같은 컴퓨터 독출 매체에 저장될 수 있다.

[0021] 본 명세서에 설명되는 방법 단계, 프로시저, 또는 기능들 모두는, 프로세서로 및 다른 성분으로 하여금 방법 단계들, 프로시저, 기능을 수행하도록 야기하는 실행가능 명령들과 조합되는 하나 또는 그 이상의 프로세서를 이용하여 구현될 수 있다는 점이 이해되어야 한다. 본 명세서에서, "프로세서" 및 "컴퓨터"라는 용어는 이들이 홀로 이용되거나 또는 컴퓨터 독출 저장 매체 또는 메모리에 저장된 실행 명령어들과 조합되어 이용되거나에 관계없이, 다양한 타입의 공지된 연산 구조들 모두를 포함하는 것으로 이해되어야 하는데, 여기에는 하나 또는 그 이상의 마이크로 프로세서, 특수 목적 컴퓨터 칩, 필드-프로그램 가능 게이트 어레이(field-programmable gate arrays, FPGAs), 컨트롤러, 주문형 집적 회로(application-specific integrated circuits, ASICs), 하드웨어/펌웨어의 조합 또는 특별 목적 또는 범용 처리 회로부를 포함한다.

[0022] 이동 장치(212) 또는 이것의 다양한 성분들은 다양한 타입의 송신을 수신, 디코딩 및 처리하도록 구성될 수 있는데, 이러한 송신에는 예를 들면 특수 방송 트랜시버(241)를 통한 DVB-H, DVB-H+ 또는 DVB-MHP와 같은 디지털 비디오 방송(DVB) 표준에 기반한 디지털 광대역 방송 송신을 포함한다. 또는, 다른 디지털 송신 포맷이 보충적 서비스의 가용성에 대한 콘텐츠 및 정보를 전달하기 위하여 이용될 수 있다. 이에 추가되거나 이것 대신에, 이동 장치(212)는 FM/AM 라디오 트랜시버(242), 무선 근거리 통신망(WLAN) 트랜시버(243), 및 원격 통신 트랜시버(244)를 통하여 송신을 수신, 디코딩 및 처리하도록 구성될 수 있다. 또는, 트랜시버(241, 242, 243 및 244)는 개별 송신기와 수신기 성분을 포함할 수 있다.

[0023] 비록 도 2에 대한 전술된 설명이 일반적으로 이동 장치에 관련되지만, 다른 장치 또는 시스템들도 동일 또는 유사한 성분들을 포함할 수 있으며, 동일 또는 유사한 기능 또는 방법을 수행할 수 있다. 예를 들면, PC(115)(도 1)와 같은 움직이지 않는 컴퓨터가 위에 기술된 성분을 포함할 수 있고, 이동 장치(212) 및 그 성분과 동일하거나 유사한 기능을 수행하도록 구성될 수 있다.

[0024] 예를 들어 인터넷을 통한 데이터 통신과 같은 웹 기반 시스템에서, 클라이언트-서버 통신은 일반적으로 요청-응답 패턴을 따르는데, 여기서는 HTTP 요청을 서버에 송신하는 클라이언트에 응답하여 서버가 클라이언트에게 응답을 다시 송신한다. 전형적으로 응답은 클라이언트에 의해 요청받는 데이터를 포함한다. 각 요청-응답 사이클 동안, 이동 클라이언트는 그 무선 통신망 내에 TCP 접속을 확립해야 하며, 이후에 TCP 접속을 사용하여 요청/응답 메시지를 통신한다. 각 사이클은 클라이언트 및 서버 간에 수 개의 메시지의 교환 또는 "왕복(round trips)"을 포함할 수 있으며, 비록 어플리케이션 페이로드(application payload)가 데이터의 수 바이트만을 가진다 하여도 HTTP 요청 헤더는 종종 수백(예를 들면, 600) 바이트를 가질 수 있다. 그러므로, 통신 레이턴시는 높아질 수 있고, 링크 이용율이 낮아질 수 있다.

[0025] Wi-Fi 및 3G와 같은 무선 기술이 높은 통신 레이턴시(communication latency) 및 낮은 비트율을 가지기 때문에, 장치가 이동성을 가지는 것은 HTTP 및 웹 통신의 문제를 악화시킨다. 보다 새로운 무선 데이터 통신 기술도 유사한 문제를 제공한다. 업링크 및 다운링크 속도는 전형적으로 비대칭이며, 업링크는 더 낮고 에너지를 더 소비한다. 각 송신 이후에, 무선 인터페이스는 고전력 소비 상태에 있으며, 비활성의 선결된 기간이 지난 이후에만 저전력 상태로 전이된다. 각 통신은 CPU 사이클, 메모리와 배터리 파워를 소비한다. 만일 이동 클라이언트가 서버와 통신해야 하는 다중 성분을 가진다면, 무선 인터페이스는 늘 비지(busy) 상태로 유지되고, 배터리는 빨리 빽빽이 되며, 이동 장치의 응답성이 점점 떨어질 수 있다.

[0026] 전술된 설명을 참조하면, 본 발명의 양태는 전력이 제한된 자원인 경우, 즉, 전화기가 전력원에 꽂혀있는 것이 아니라 그 대신에 배터리 전력으로 동작하는 경우에, 이동 장치를 통한 클라이언트-서버(client-server) 상호작

용을 위한 개선된 기술에 대한 것이다. 본 발명의 기술은 또한 심지어 장치가 전력원에 플러그인 된 경우에도 전력 소모를 절감하는데 이용함으로써, 전송된 바이트수를 줄임으로써 네트워크 대역폭을 더 잘 이용하게 하거나, 통신에서 이용되는 처리 사이클을 감소시킴으로써 이동 장치의 CPU를 더 잘 이용하도록 한다.

[0027]

본 발명의 다양한 양태를 설명하기 위해, 다음의 실례가 되는 시나리오가 사용된다. 도 1을 다시 참조하면, 장치 (110, 120)는, 모든 사용자들이 모일 수 있는 공유 화이트보드는 물론 채팅 서비스와 같은 협업 서비스를 제공하는 클라이언트 소프트웨어(165)를 실행시킬 수 있다. 도 3을 더 참조하면, 클라이언트 소프트웨어(165)는 협력 클라이언트(301)이라고 불릴 수 있다. 도 3은 각 협력 클라이언트와 그 적용가능 서버 간의 논리적 데이터 흐름 다이어그램을 도시한다. 협력 클라이언트(301)를 이용하면, 각 협력 서비스(303, 305)는 대항 서비스에 대한 고유 서버와 통신하는 고유 클라이언트의 역할을 할 수 있다. 예를 들면, 예를 들면 채팅 특징(feature)은, 예를 들어 서버(130)(도 1)에서 실행되는 논리적 서버와 같은 협업 서버(313)에서 실행되는 채팅 서비스 소프트웨어 모듈(317)과 통신하는 협력 클라이언트(301)에서 실행되는 채팅 클라이언트 소프트웨어 모듈(303)에 의하여 각 협력 클라이언트(301)에서 제공될 수 있다. 마찬가지로, 화이트 보드 특징은 예를 들면, 채팅 특징을 제공하는 논리적 서버와 구별되는 서버(130)(도 1)에서 실행되는 논리적 서버와 같은 협력 서버(313)에서 실행되는 화이트 보드 서비스 모듈(319)과 통신하는 협력 클라이언트(301)에서 실행되는 화이트 보드 클라이언트 소프트웨어 모듈(305)에 의해 각 협력 클라이언트(301)에서 제공될 수 있다. 추가적 협업 서비스들도 제공될 수 있지만, 본 명세서에서는 예시적 목적을 위하여 오직 두 개만이 참조되었다.

[0028]

채팅과 화이트 보드 성분의 각각은 클라이언트 측에서 자신의 사용자 인터페이스를 가지는데, 가끔은 브라우저 윈도우 내의 단일 웹 페이지 내에서 제공된다. 또는 다른 사용자 인터페이스가 사용될 수 있다. 고유 서버를 가지는 것에 추가하여, 각 성분은 해당 성분에 의하여 제공되는 서비스/특징에 상응하는 데이터를 저장하는 자신의 데이터베이스를 가질 수도 있다. 어플리케이션 서버(317, 319)는 요청이 클라이언트(303, 305)로부터 수신되면 예를 들어 아파치(Apache)를 이용하여 웹 서버(311)에 의하여 호출될 수 있다. 서버에서 가용한 최신 정보를 검색(retrieve)하기 위해 예를 들면 매 5초마다와 같은 스케줄에 따라서 클라이언트(303, 305)는 그들의 개별 서버(317, 319)를 전형적으로 폴링한다.

[0029]

본 발명의 일 양태에 따르면, 브로커 스터브(307) 및 브로커 스켈레톤(315)이 서버 요청을 단일 메시지로 다중화 및 결합함으로써, 이동 클라이언트 내의 자원을 절약하기 위하여 이용될 수 있다. 서버와 백그라운드로 통신하는 서비스를 제공하기 위해, 브로커 스터브(307)는 Ajax(Asynchronous JavaScript and XML)로 코딩될 수 있다. 브로커 스터브(307)는, 클라이언트 성분이 XMLHttpRequest(XHR)요청을 서버(313)로 전송할 수 있는 통로가 되는 API(Application programming interfaces)를 제공함으로써, 다중 요청들이 하나로서 통합 및 전송될 수 있도록 하고 폴링 간격이 업데이트의 가용성, 네트워크 상태, 및 서버 부하(workload)에 따라서 동적으로 적응되도록 할 수 있다.

[0030]

따라서, 도 3에서 도시된 예에서 그리고 도 4를 더 참조하면, 단계 401에서 사용자는 그 또는 그녀의 이동 전화기에서 브라우징하여 웹-기반 협업 서비스에 접근할 수 있으며, 이를 통하여 이동 클라이언트(110, 120)가 무선 네트워크(예를 들면, 하이-레이턴시(high-latency) 네트워크)를 통하여 다른 서버(313)를 경유하여 상호 통신할 수 있다. 이러한 프로세스 전체에 걸쳐서, 커뮤니케이션을 효율적이게 하기 위해 브로커 모듈(307, 315)은 클라이언트-서버 통신을 중재(mediate)한다. 브로커는 클라이언트 측의 모듈(스터브(307))과 서버 측의 모듈(스켈레톤(315))을 포함할 수 있다. 스터브(307)는 채팅(303)과 화이트 보드(305)와 같은 협력 클라이언트 성분으로서 동일한 웹 페이지에 있을 수 있고, 이들이 서버(313)와 통신하도록 하기 위한 API를 제공한다. 또는, 스터브(307)는 클라이언트 성분으로부터 독립해 있을 수 있다.

[0031]

단계 403에서, 협력 클라이언트(301)를 이용하는 사용자는 협력 서비스 안에서 채팅과 화이트 보드 서비스를 활성화한다. 단계 405에서, 각 클라이언트는 그 각각의 서버로부터 업데이트를 요청한다. 각 요청은 서버 요청 메시지(server request message) 또는 성분 요청 메시지(component request message)라고 불릴 수도 있다. 단계 407에서, 브로커 스터브(307)는 다중 서버 요청 메시지들을 인터셉트하고, 단계 409에서, 스터브(307)는 이러한 성분들로부터의 다중 요청을 하나의 브로커 요청 메시지로 다중화하고 다중화된 브로커 요청 메시지를 브로커 스켈레톤(315)에 전송할 수 있다. 단계 411에서, 스켈레톤(315)은 스터브(307)로부터 송신받는 다중화된 브로커 요청 메시지를 인터셉트하고, 해당 요청을 역다중화한다. 단계 413에서, 스켈레톤은 그들의 각각의 최초의 성분 서버(317, 319)로 역다중화된 요청(개별 성분 요청 메시지들)을 전송 또는 파견(dispatch)한다.

[0032]

단계 415에서, 하나 이상의 성분 서버(317, 319)는 그들의 각각의 성분 클라이언트(303, 305)에 대한 응답을 역으로 생성한다. 단계 417에서, 브로커 스켈레톤(315)은 응답을 인터셉트하는데, 이것은 각각 서버 응답 메시지

또는 성분 응답 메시지라고 불린다. 단계 419에서 브로커 스켈레톤(315)은 응답 메시지를 다중화하고, 다중화된 브로커 응답 메시지를 브로커 스터브(307)로 다시 전송한다. 단계 421에서, 브로커 스터브(307)는 다중화된 브로커 응답 메시지를 수신하고 해당 메시지를 개별 성분 응답 메시지로 역다중화한다. 단계 423에서 브로커 스터브(307)는 그들의 상응하는 클라이언트 성분(303, 305)으로 개별 응답 메시지들을 파견 또는 전달한다.

[0033] 본 발명의 다른 양태에 따르면, 브로커 시스템(예를 들면, 브로커 스켈레톤(315) 및 브로커 스터브(307))는 적응적 폴링을 수행할 수 있다. 일 실시예에서, 스터브는 주기적 폴링 요청을 해당 요청에 대한 최후 연결이 완성되기 이전에는(또는 해당 요청이 선결된 값에 기반하여 타임아웃 되기 이전에는) 전송하지 않음으로써, 결과적으로 그 간격(interval)이 현재 네트워크 상태 또는 서버 부하에 비하여 너무 빈도가 높은 경우에는 폴링 작업을 느리게 할 수 있다. 또한, 브로커 스터브는 업데이트의 가용성에 따라 주기적인 요청의 폴링 간격을 적응시킬 수 있다(속도를 늦추거나, 빠르게 할 수 있다). 클라이언트 성분이 스터브에 피드백을 스터브 API를 경유하여 데이터의 가용성을 표시하여 제공할 필요가 있는 한, 업데이트의 가용성은 어플리케이션에 특유할 수 있다.

[0034] 예시적인 일 실시예에서, Yahoo! 접속 매니저(Yahoo! connection manager, YCM)가 XHR을 프로그래밍하기 위한 크로스 플랫폼(crossplatform API)을 위해 이용될 수 있다. 메인 인터페이스는 `asyncRequest(method, URL, callback, data)`일 수 있는데, 이것은 비동기 요청을 서버로 제공한다. 4개의 인수들 중에서, `method`는 GET 및 POST와 같은 HTTP method이다; `url`은 목표 웹 서버의 주소이다; `callback`은 서버 응답을 핸들링하기 위한 오브젝트이다; `data`는 POST 요청 내에 전송될 데이터를 유지한다. `callback` 오브젝트는 다음과 같은 세 개의 멤버들을 제공할 수 있다: `success`는 성공적으로 반환된 서버 응답을 처리하기 위하여 호출되는 함수이다; `failure`는 문제가 있는 응답(예를 들어 통신 실패 또는 서버 에러와 같은)을 처리하기 위한 함수이다; `argument`는 서버 응답을 처리하기 위한 성공 및 실패 핸들러(handler)들에 필요한 데이터를 포함하는 모든 오브젝트이다.

[0035] API는 XHR 작업의 다음 네 가지 타입들을 구별함으로써, 내재적인 XHR 통신을 위한 YCM을 레버리지(leverage)할 수 있다: 예를 들어 성분이 초기화될 때 공유 데이터를 탐색하도록 서버에 질의(query)하기 위한 일회적 폴링(one-time polling), 서버로 하여금 몇 가지 공유 데이터의 신규 업데이트에 대해서 주기적으로 질의하는 주기적 폴링(periodic polling), 서버에게 몇 가지 공유 데이터에 대한 지역적 상태 변화를 통지하는 일회적 쟁신(one-time updating), 및 몇 가지 공유 데이터의 업데이트를 서버로 주기적으로 쟁신하는 주기적 쟁신(periodic updating).

[0036] 모든 작업은 고유 ID, 그 `url`을 획득하기 위한 방법, 및 콜백 오브젝트를 정의할 수 있다. 쟁신 작업은 추가적으로 전송될 데이터를 획득하기 위한 방법을 정의한다. 또한, 주기적 작업은 해당 요청이 전송되는 빈도를 특정하기 위한 간격을 정의할 수도 있다. 일부 실시예에서, 주기적 쟁신이 이용될 수 있는데, 예를 들어 웹 서비스 API가 주기적인 업데이트를 생성할 수 있는 마이크, 카메라, GPS와 같은 입력원을 지원하는 경우에 이용될 수 있다.

[0037] 일회성 폴링 및 쟁신 작업은 Ajax 브로커 방법을 호출함으로써 특정 성분에서 전송될 수 있으며, 이것은 각각 `send_polling(task)`과 `send_updating(task)`인데, 여기서 `task` 오브젝트는 위에 정의된 바와 같다. 주기적 폴링 작업은 성분이 Ajax 브로커 방법 `register_polling(task)`를 호출함으로써 등록한 이후에 브로커 스터브에 의해 전송된다.

[0038] 본 발명의 일 양태에 따르면, 브로커 스터브(307)는 요청을 다중화하고 요청 간격을 적응시킴으로써 주기적 폴링 요청을 관리할 수 있다. 일 실시예에서, 브로커 시스템은 일회적 폴링 및 쟁신 요청을 개별 메시지로서 전송하면서 오직 주기적 폴링 요청만을 다중화할 수 있다. 요청들을 다중화하는 것은 필수적으로 더 많은 런타임 부하(runtime overhead)를 가져오게 된다. 일회적 요청들이 일반적으로 성분 UI를 이용한 사용자 상호작용에 의하여 트리거링되기 때문에, 사용자는 흔히 짧은 시간 내에 몇 가지 UI 피드백을 보고자 기대한다. 따라서, 시스템은 상호작용이 발생하자마자 즉시 일회적 요청을 전송하고, 그 결과 해당 요청이 최소한의 지연으로서 서버에 도달하고 응답을 다시 얻어낼 수 있다. 반면에, 주기적 폴링 작업은 원격 자원을 끌어오는데 이용되는 백그라운드 활동일 수 있으며, 더 많은 지연을 참아야 할 수 있다. 따라서, 시스템은 주기적 폴링 작업만을 다중화 할 수 있다.

[0039] 브로커 스터브(307)에서, 메타의 시스템 타이머는 주기적 요청을 규칙적으로 외부로 전송한다. 바람직하게는, 타이머 간격(시스템 파라미터 `meta_interval`로 표시됨)은 모든 주기적 요청 간격의 최대 공약수(gcd)여야만 한다. 실제로, 1,000 ms(또는 다른 값)와 같은 값이 사용될 수 있다. 주기적 폴링 작업은 내부 큐(queue)인 `polling_tasks`에 등록될 수 있다. 각 작업 `t`는 그 간격을 표시하기 위하여 파라미터 `t.interval`을 사용할 수

있고, 그것이 전송된(그 자체로서 또는 다중화되어) 최후 시각을 표시하기 위하여 파라미터 `t.last_time`을 이용할 수 있으며, 그것이 전송된 연결을 표시하기 위하여 파라미터 `t.connection`을 이용할 수 있다.

[0040] 메타 타이머는 다음 2의 상태를 동시에 만족시키는 모든 작업 `t`를 위해 폴링 작업을 스캔할 수 있다: (1) `now - t.last_time >= t.interval`일 것, 여기서 `now`는 현재 시간임, (2) `t.connection`이 진행중이 아닐 것. 요청이 전송되었지만 성공 또는 실패로서 완성되지 않으면 어떤 연결은 진행중(in progress)이다. 이러한 정격 작업(qualified task)의 URL은 YCM `asyncRequest` 방법을 경유하여 브로커 스켈레톤으로 하나의 요청 내에서 전송된다. 한편, 그들의 `last_time` 파라미터는 그들이 전송된 현재 시간 `now`로 설정된다.

[0041] 메타 타이머는 서버로부터의 응답을 처리하기 위한 자신의 콜백 오브젝트를 가질 수 있다. 그 콜백 오브젝트의 인수 파라미터는 어떤 폴링 작업이 다중화되고 전송되었는지를 추적한다. 다중화된 응답이 스켈레톤으로부터 수신되면, 메타 타이머의 응답 핸들러(response handler)가 해당 메시지를 파싱(parse)하고, 개별 폴링 작업에 대한 응답을 찾아내고, 이들을 그들의 응답 핸들러로 파견하는데, 이들은 다시 데이터를 파싱하고 응답을 그들의 성분 UI로 반영한다.

[0042] 본 명세서에서 전술된 바와 같은 다중화는 전송되는 바이트수를 줄이고 전력 소모를 감소시킨다. 그러나, 다중화되지 않는 경우에 비하여 개별 작업이 서버로부터 응답을 수신하는데 더 오랜 시간이 걸릴 수 있다. 다소 커진 다중화 부하가 송신 시간을 많이 증가시키지는 않지만, 서버가 하나가 아니라 다중 요청을 처리하는 데는 더 오랜 시간이 걸린다. 그 결과로서, 일회적 요청들은 전술된 바와 같이 다중화되지 않는다.

[0043] 다중화되건 아니건 간에 관계없이 폴링 요청은 서버에 업데이트가 없다면 여전히 자원을 낭비시킬 수 있기 때문에, 본 발명의 다른 실시예는 적응적 폴링의 다른 형태를 이용할 수 있다. 다중화의 제2 조건, 즉, 메타 타이머가 소정 연결이 진행중인지 여부를 결정하는 것은 이미 몇 가지 적응적 동작을 예시한다. 요청이 개시된 이후에, 자원들은 응답이 수신되거나 타임아웃 이벤트가 발생하기 전에는 릴리스되지 않는다. 응답은 여러 가지 이유에 의하여 지연될 수 있다. 예를 들어, 몇 가지를 언급하면 웹 브라우저는 클라이언트 및 서버 간에 확립된 허용된 활성 연결의 최대 개수에 도달할 수 있고; 네트워크가 혼잡하거나; 또는 서버가 포화되었을 수 있다. 이러한 상태에서는, 후속 라운드(round)로 요청을 연기시키는 것이 시스템 전체의 성능에 유리할 수 있다.

[0044] 부가적으로, 스터브는 성분으로 하여금 서버 응답이 수신되면 스터브에게 긍정 또는 부정 피드백을 제공하도록 하는 방법을 제공할 수 있다: `polling_feedback(ID, new_data)`, 여기서 `ID`는 등록된 주기적 폴링 작업의 `id`이고 `new_data`는 응답 내에 업데이트가 적용한지를 표시한다. 긍정 피드백은 스터브로 하여금 그 간격을 감소시킴으로써 폴링 작업을 가속화하도록 요구하고, 부정 피드백은 스터브로 하여금 그 간격을 증가시킴으로써 테스크를 감속하라고 요구한다. 그러나, 얼마나 많이 가속 또는 감속할지는 해당 작업에 대해 선택된 적응적 방법에 의존한다.

[0045] 즉, 각 주기적 폴링 작업은 피드백이 주어졌을 때 어떻게 그 간격을 적응시킬지를 스터브에게 표시하는 파라미터 `adaptive_method`를 가질 수 있다. 적응적 방법은 작업의 간격이 적응될 수 있는 범위를 정의하는 `min_interval` 및 `max_interval`을 포함하는 두 개의 파라미터를 이용할 수 있다. 주기적 폴링 작업은 최초 간격에서 개시될 수 있으며, 해당 간격은 시간이 흐름에 따라 적응될 수 있다. 디폴트로, 복합 적응적 방법(hybrid adaptive method)이 사용될 수 있는데, 여기서 간격은 긍정 피드백에 대해서는 `min_interval`로 설정되고 부정 피드백이 수신되면 상수 또는 가변량(meta_interval)만큼 감소된다. 예시적 일 실시예에서, 복합형 방법은 이진 가속(binary speedup)을 이용할 수 있는데, 이것은 더 공격적이며, 선형 감속(linear slowdown)을 이용할 수 있는데 이것은 가장 보수적인 것이다. 이처럼, 업데이트가 수신되자마자, 수 개의 신규 업데이트들이 협업시스템 내에서 첫 번째 것에 응답하여 후속될 가능성성이 있다는 기대 하여 다중 폴링 요청들이 한 줄로 전송될 수 있다.

[0046] `min_interval` 파라미터는 스터브가 주기적 요청을 전송하는 최고 빈도인 `meta_interval`과 동일한 값을 가질 수 있다. `meta_interval`은 클라이언트 및 서버 간의 평균 왕복 통신 시간에 의존할 수 있다. 그러나, 간략화를 위하여 셀룰러 네트워크 환경에서의 일 실시예는 1,000 ms에 `meta_interval`을 설정할 수 있다. 그러나, `max_interval` 파라미터는 바람직하게는 사용자의 허용 가능한 피드스루 지연(feedthrough delay)을 반영하며, 예를 들어 이것은 이루어진 업데이트가 한 사용자로부터 다른 사용자에게 도달하는데 걸리는 시간이다. 실시간 작업에서부터 비실시간 작업까지와 같은 상이한 타입의 협업 작업들은 상이한 허용 가능 피드스루 지연을 가질 수 있다. 예를 들면, 근사-실시간(near-realtime) 협력을 지원하기 위하여, 디폴트 성분 수준의 `max_interval`은 10,000 ms일 수 있다. 따라서, 시스템은 사용자로 하여금 스터브 수준에서(모든 성분에 적용함으로써) 및 성분 수준에서(오직 그 성분에서 적용함으로써) 최저 폴링 빈도를 특정하도록 허용하는 `set_max_interval(b)` 방

법을 제공할 수 있다. 그러면, 각 성분의 실효 max_interval은 자신의 폴링 간격이 적응될 때 이러한 두 개의 극한의 최소값이다. 그러므로, 적응적 폴링은 주기적 폴링 작업의 "최적" 간격을 제공하는 문제도 용이하게 만들 수 있는데, 그 이유는 사용자는 오직 특정값 대신에 범위를 특정하면 되기 때문이다. 만일 어떤 성분이 피드스루 지연에 대해 허용할 수 있는 수준이 더 낮으면, 사용자는 예를 들어 3000ms와 같은 더 작은 최대값을 특정할 수 있다.

[0047] 전술된 예시적 실시예에서, 브로커 스터브(307)로부터의 다중화된 요청이 수신되면, 브로커 스켈레톤(315)은 해당 요청을 파싱하고, 원본 폴링 요청을 추출한다. 그러면, 해당 요청이 제공되고, 응답들은 단일 메시지 내에서 스터브로 되전송되기 위하여 다중화된다.

[0048] 이러한 요청을 처리(serve)하기 위해, 브로커 스켈레톤은 그들의 URL을 그들의 최초의 성분 서버에 전달하거나, 또는 직접적으로 서버 함수에 대한 함수 호출을 수행할 수 있다. 어느 경우에나, 이러한 동작들은 직렬로 동기적으로 실행되거나 또는 병렬로 비동기적으로 실행될 수 있다. 따라서, 전체적으로 네 개의 가능한 실행 정책들이 존재하는데, 이들은 call-serial(직렬적 함수 호출들), call-async(병렬적 호출), url-serial(URL들을 직렬로 전달), 및 url-async(URL을 병렬로 전달)들이다.

[0049] 전술된 내용에 기반하여, 본 발명의 예시적인 일 실시예는 다음 표 1에 예시된 바와 같은, 성능에 영향을 미치는 정책 및 파라미터들을 포함할 수 있다.

표 1

파라미터	설명
Meta_interval	메타 타이머가 주기적 폴링 요청을 얼마나 자주 다중화 및 전송해야 하는지
Max_interval	스터브 레벨에서의 최대 간격, 또는 스터브가 주기적 폴링 요청을 전송할 최저 빈도
Multiplexing	주기적 폴링 요청이 다중화되어야 하는지 여부; 만일 다중화가 없다면, 모든 요청들은 원래의 성분 서버로 개별적으로 전송된다
Adaptive	주기적 폴링 작업의 간격이 적응되어야 하는지 여부; 만일 비-적응적이라면 모든 폴링은 그들의 최초 간격으로 전송된다; 만일 적응적이라면, 적응적 방법이 이용되며, 범위를 설정하기 위하여 min_interval 및 max_interval 파라미터들이 이용됨
Exec_mode	다중화된 요청을 서빙하기 위하여 스켈레톤 내에서 어떤 실행 정책(예를 들어, call-serial, call-async, url-serial, url-async 등)이 이용되어야 하는지

[0050] [0051] 디폴트로, 메타 간격은 1,000 ms에 조정될 수 있다; 스터브-수준 최대 간격은 10,000 ms일 수 있다; 다중화와 적응적 파라미터는 "진실(true)"에 조정될 수 있다; 모든 주기적 폴링 작업의 adaptive_method가 전술된 바와 같은 "복합형(Hybrid)" 방법으로 디폴트로서 설정되고 폴링 간격은 1,000 MS 내지 10,000 ms 사이에서 변동할 수 있다. server exec 모드는 "call-serial"로 디폴트로서 설정될 수 있다.

[0052] 전술된 브로커 시스템에 다양한 수정 및 변형들이 선택적으로 이루어질 수 있다. 본 발명의 일 양태에 따르면, 브로커는 사용자 행동에 기반한 적응적 폴링의 대안적 형태를 수행할 수 있다. 특히, 클라이언트 장치를 사용하는 동안 폴링 간격은 하는 것은 사용자 행동을 근거로 하면서 동적으로 적응될 수 있다. 예를 들면, 위에 기술된 협력 예에서, 이동성 장치의 스크린은 너무 작아서 모든 성분들이 동시에 표시되거나 동일한 크기로 디스플레이될 수 없다. 따라서, 성분이 보이지 않게 되거나 사용자에게 명백하게 보이지 않게 되면, 해당 성분의 폴링 간격은 적응되어 해당 성분이 그 상응하는 서버를 덜 자주 폴링하도록 할 수 있다. 반대로, 어느 성분이 보이게 되거나 또는 사용자의 시선을 더 끌게 되면, 해당 성분의 폴링 간격은 적응되어 이것이 더 자주 폴링하도록 할 수 있다. 사용자 행동에 기반한 적응적 폴링은 성분 클라이언트 자체에 의하여 수행되거나 또는 브로커 시스템(307, 315)과 같은 클라이언트 프록시에 의하여 수행될 수 있다. 따라서, 폴링 간격의 적응 동작은, 예를 들면 어느 성분을 더 또는 덜 가시화하는 것과 같은 사용자의 동작에 기반할 수 있다. 예를 들어, 사용자

가 어떤 성분 클라이언트의 사용자 인터페이스(UI)의 크기를 변경하거나 해당 UI의 가시성 또는 다른 시각/음향 속성을 수정하면, 해당 성분의 폴링 간격은 이에 따라서 자동적으로 조절될 수 있다.

[0053] 도 5를 참조하면, 서버 가드를 이용하는 본 발명의 다른 양태는 블라인드 주기적 폴링(blind periodic polling)을 제거할 수 있다. 웹서버와 성분 서버(들) 사이에서 "가드" 서비스가 서버 측에 있을 수 있다. 서버 가드는 브로커 스켈레톤(315)과 같을 수도 있다. 단계 501에서, 성분 클라이언트는 서버 가드에 그 URL을 등록하여 해당 URL의 불변부 및 가변부를 구별한다. 예를 들면, URL "http://web.address.com/server_name?a=1&b=2"에서, 불변부는 "http://web.address.com/server_name"이고, 가변부는 "a=1&b=2"이다. 전형적으로, 불변부는 성분 서버를 가리키는 반면에, 가변부는 어떤 질의 파라미터를 포함할 수 있다. 각 클라이언트 성분은 등록 메시지를 서버 가드로 제공하여 불변 및 가변 URL 정보를 제공할 수 있다.

[0054] 단계 503과 505는 동시에 일어나거나 적어도 서로 병렬적으로 발생할 수 있다. 다른 방식으로 말하면, 단계 503과 505의 어느 쪽도 다른 단계가 발생되기 이전에 발생되어야 하는 것처럼 서로 의존하지 않는다. 단계 503에서, 서버 가드는 각 서버에 의하여 제공된 데이터에서 어떤 업데이트가 존재하는지 여부를 결정한다. 이것은 몇 가지 방식으로 달성될 수 있는데, 예를 들어 신규 데이터에 대해서 서버에게 주기적으로 폴링하는 서버 가드에 의하여, 또는 서버가 신규 데이터에 대하여 서버 카드에게 사전에 통지하는 것에 의하여 달성될 수 있다. 단계 505에서, 어떤 서버가 신규하거나 갱신된 데이터를 가지고 있는지를 발견하기 위하여, 브로커 스터브(307)는 동시에 또는 병렬적으로 주기적 "하트비트(heartbeat)" 메시지를 송신하고 있다. 하트비트 요청은 바람직하게는 어떤 특정의 폴링 요청도 포함하지 않고, 오히려 어느 서버가 상응하는 성분 클라이언트를 위한 갱신된 데이터를 가지거나 신규 데이터를 가지는지를 찾기 위한 단순한 질의를 포함한다.

[0055] 단계 507에서, 서버 가드는 브로커 스터브에 응답하며, 어떤 서버가 포스팅된(posted) 신규/갱신 데이터를 가지는지에 대한 표시(indication)를 제공한다. 단계 509에서, 브로커 스터브는 메시지를 자신에 대한 신규/갱신 데이터가 존재하는 클라이언트 성분에게 전송하고, 신규/갱신 데이터의 가용성을 표시한다. 끝으로, 단계 511에서, 신규 데이터의 가용성에 관련된 통지를 받은 바 있는 모든 성분 클라이언트는, 해당 성분 클라이언트에 고유한 가변 파라미터를 이용하여 해당 신규 데이터를 획득하기 위한 폴링 요청을 자신의 개별 서버로 전송한다. 질의에 가변부가 없는 경우, 예를 들어 동일한 질의가 언제나 데이터베이스에 인가되는 경우에는, 서버 가드는 선택적으로 해당 데이터를 데이터베이스로부터 검색하고, 상응하는 성분 클라이언트에 대한 갱신이 존재할 경우에는 하트비트 메시지에 응답하여 해당 데이터를 제공함으로써, 해당 성분 클라이언트에 대한 업데이트 프로세스를 가속화할 수 있다.

[0056] 전술된 서버 가드를 이용하는 본 발명의 실시예에 따르면, 서버 가드는 소형 데이터베이스 테이블을 작성 및 사용할 수 있고, 이것은 선택적으로 고속 액세스를 위해서 서버 가드의 메인 메모리에 상주할 수도 있다. 후속하여, 성분 클라이언트 X1으로부터의 어떤 최신 요청이 잠재적으로 성분 클라이언트 X2로부터의 폴링 요청에 대한 응답을 변경하면, 소형 테이블이 갱신되어 해당 업데이트의 가용성을 표시한다. 예를 들면, 이동 단말기(110)(도 1)를 사용하고 있고, 채팅 클라이언트 X1을 실행하는 제1 사용자가 이동 단말기(120)를 사용하고 있고, 채팅 클라이언트 X2를 실행중인 제2의 사용자와 통신하고 있을 수 있다. 제2의 사용자가 제1 사용자에 의한 채팅 본문 입력을 볼 수 있기 위하여, 채팅 클라이언트에서 약간의 본문을 타이프하는 제1 사용자는 채팅 클라이언트 X2에 대해 포스팅되는 업데이트를 만들 수 있다. 그러면, 하트비트가 제2 사용자의 성분 클라이언트 X2에 상응하는 클라이언트 프록시로부터 오면, 서버 가드는 해당 하트비트 요청에 대한 자신의 응답 내에 업데이트의 가용성을 표시한다. 만일 X2의 URL에는 어떠한 가변 파라미터도 없으면, 업데이트는 서버 가드에 의하여 검색될 수 있는데, 서버 가드는 해당 채팅 클라이언트의 등록 URL을 실행하고, 해당 응답을 해당 하트비트에 대한 응답 상에 삽입(piggy-backing)한다. 그렇지 않으면, 서버 가드는 성분 클라이언트 X2로 하여금 자신의 전체 URL(불변부 및 가변부를 가지는)을 이용하여 자신의 개별 서버를 폴링하도록 명령함으로써, X2의 하트비트 요청에 응답한다.

[0057] 예시되는 실시예에 따르면, 데이터베이스 테이블은 다음 2 필드를 포함할 수 있다: (component_id, last_update_timestamp). 그러면, 성분 데이터베이스(또는 성분 X2의 폴링 요청에 대한 응답에 영향을 주는 모든 것)에 업데이트가 이루어지면, 성분 X2의 엔트리는 해당 갱신이 발생한 서버 시간(예를 들어 T1)을 이용하여 갱신된다.

[0058] X2의 클라이언트 프록시로부터의 하트비트 메시지는 X2가 수신한 가장 최근의 업데이트의 타임스탬프(timestamp)인 타임스탬프(예를 들어 T2)를 포함한다. 해당 하트비트를 수신하면, 가드는 T1과 T2를 비교하고,

만일 T1 > T2라면 업데이트의 가용성에 관련된 표지(indication)가 해당 하트비트에 대한 응답 상에 삽입된다. 그러면, 클라이언트 프록시(브로커 스터브)는 X2에 폴링 요청을 송신하도록 명령한다. 반면에, 질의에 만일 어떠한 가변부도 존재하지 않으면, 신규 데이터는 서버 가드에 의해 직접 검색되고 해당 하트비트 메시지에 대한 그 응답에 삽입된다.

[0059] 전술된 바와 같은 본 발명의 양태들은 폴링 요청의 개수 및 이동 클라이언트로부터 그 상응하는 서버로 전송되는 총 바이트수를 감소시킨다. 또한, 전술된 바와 같은 본 발명의 양태들은 클라이언트 장치의 CPU 처리, 메모리, 비트율, 및 배터리 수명을 보전한다. 또한, 본 발명의 양태들은 서버 부하를 감소시키고 다양한 이동 인터넷 서비스의 성능을 개선한다.

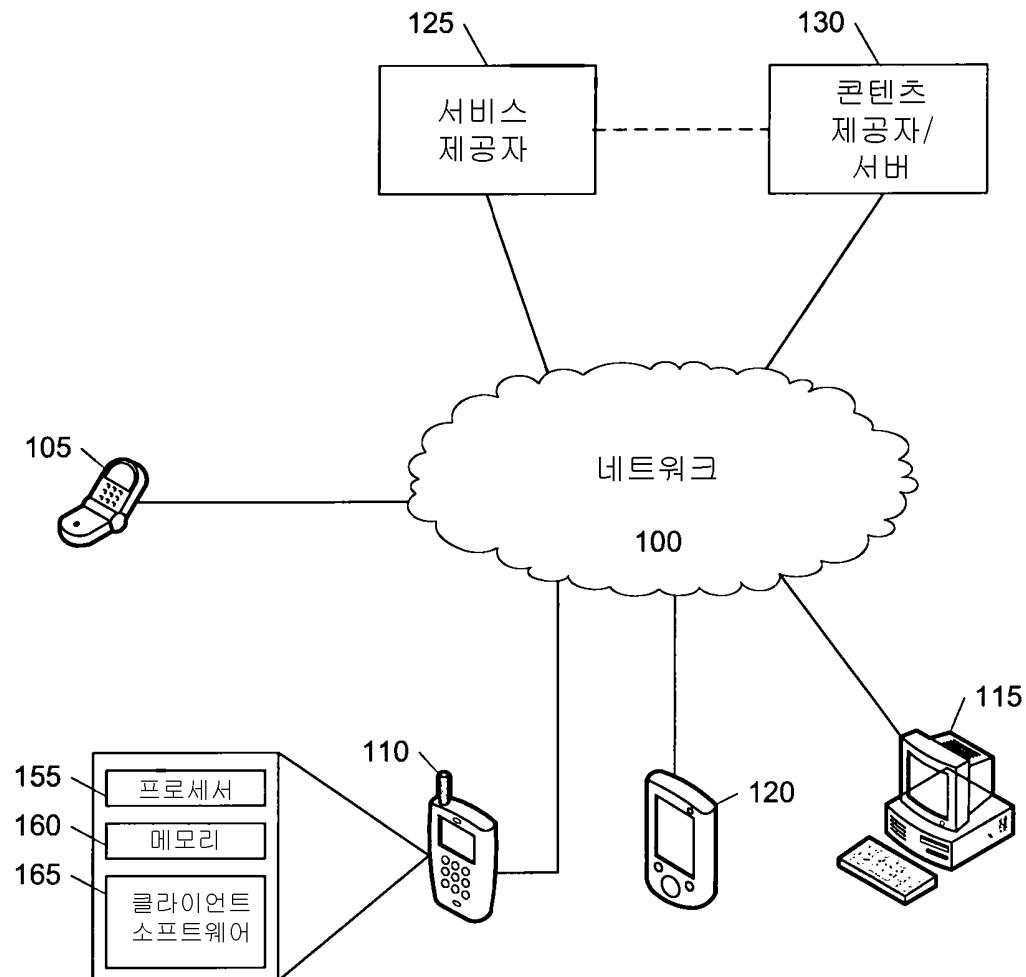
[0060] 비록 본 발명의 보호 대상(subject matter)은 구조적 특징 및/또는 방법론적 동작에 특이적인 언어를 이용하여 설명되었으나, 첨부된 특허청구범위 내에 정의된 보호 대상들은 반드시 전술된 특이적 특징 또는 동작에 한정되는 것이 아님이 이해되어야 한다. 오히려, 전술된 특이적 특징 및 동작들은 청구항을 구현하기 위한 예시적인 형태로서 개시된다.

산업상 이용가능성

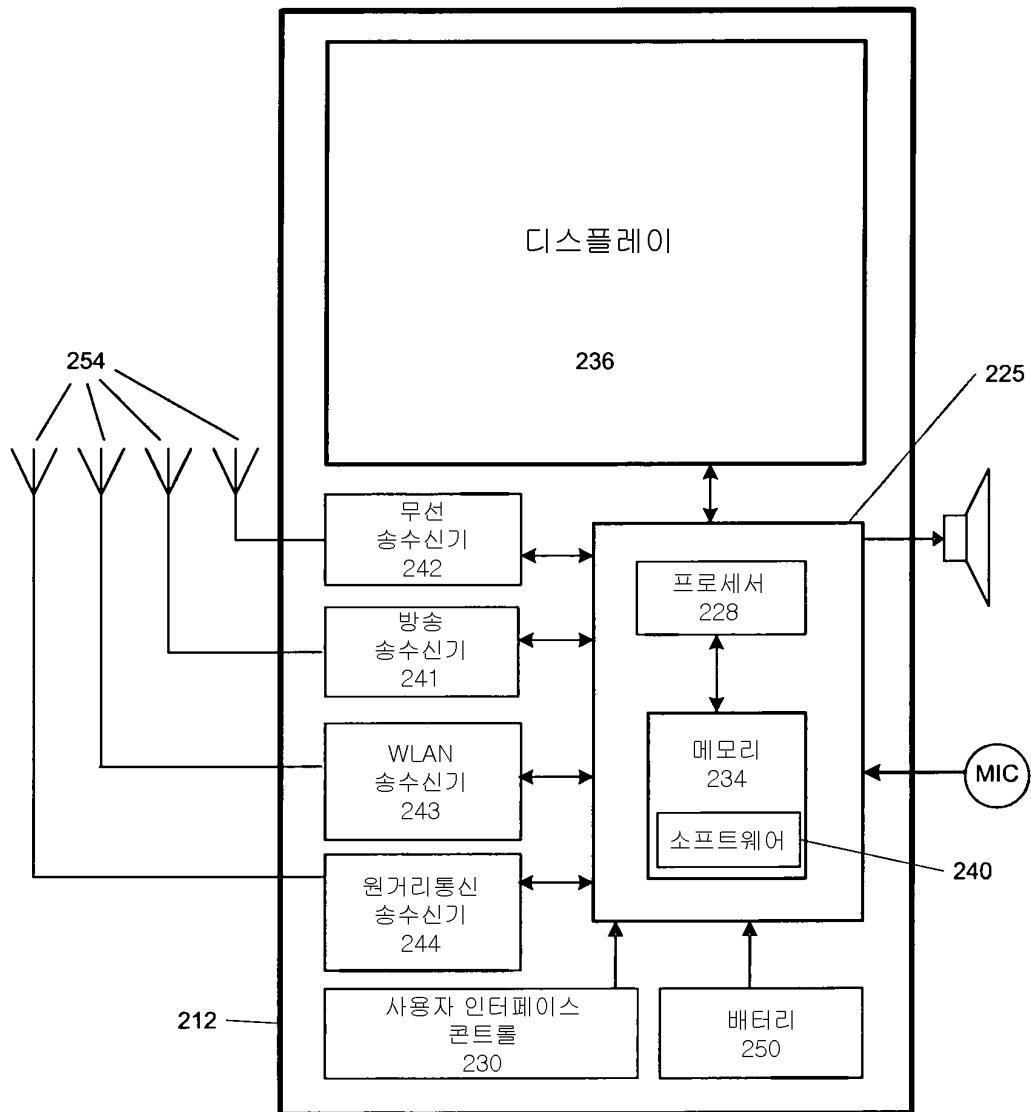
[0061] 본 발명은 이를테면 휴대폰, 스마트폰, 개인 휴대 정보 단말기, UMPC(ultra-mobile PC)와 같은 이동식 저자원 장치들에서 자원을 보호하는데 적용될 수 있다.

도면

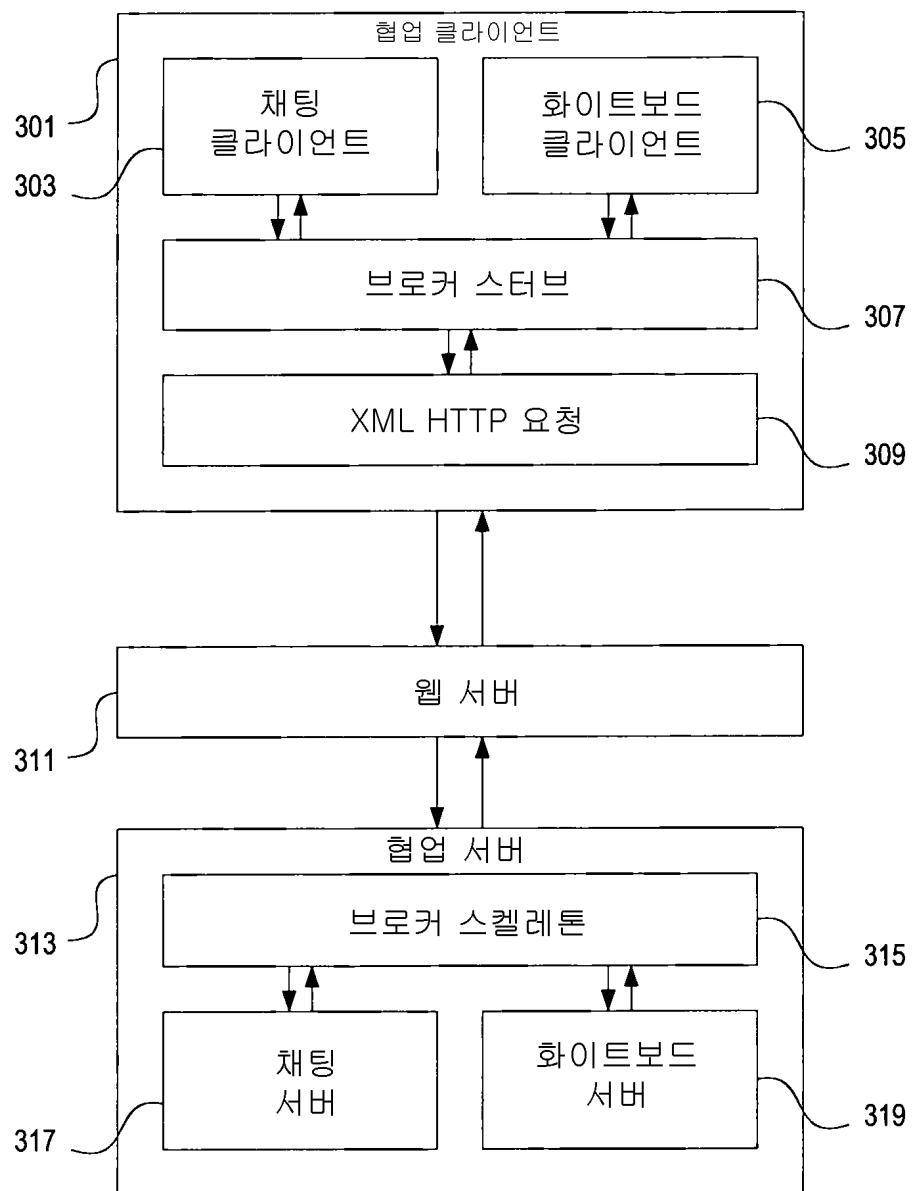
도면1



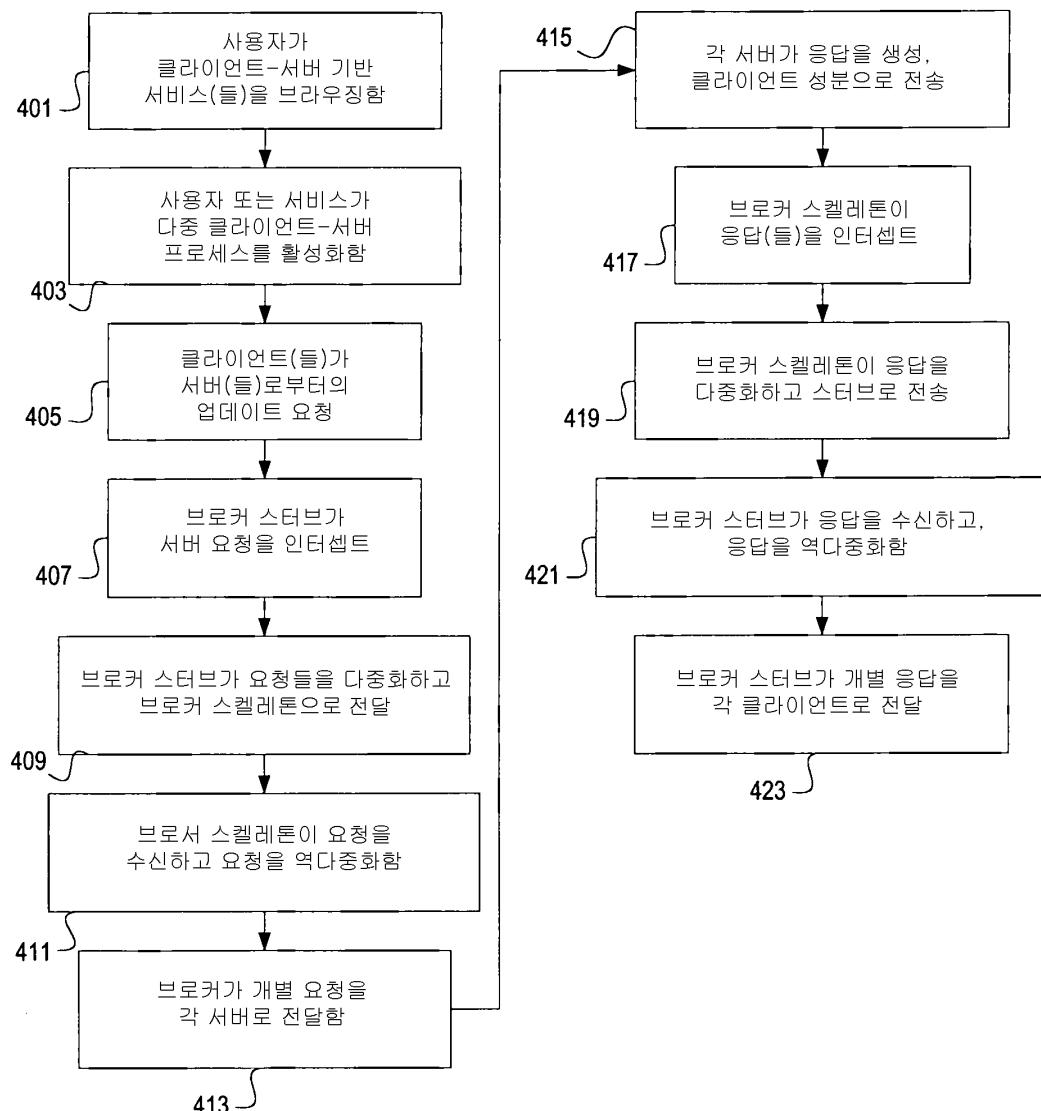
도면2



도면3



도면4



도면5

