



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2019년03월26일

(11) 등록번호 10-1962301

(24) 등록일자 2019년03월20일

(51) 국제특허분류(Int. Cl.)

G06F 17/22 (2006.01) G06F 16/00 (2019.01)

(52) CPC특허분류

G06F 17/2247 (2013.01)

G06F 16/9574 (2019.01)

(21) 출원번호 10-2017-7028030(분할)

(22) 출원일자(국제) 2014년02월27일

심사청구일자 2019년02월07일

(85) 번역문제출일자 2017년09월29일

(65) 공개번호 10-2017-0117227

(43) 공개일자 2017년10월20일

(62) 원출원 특허 10-2015-7027002

원출원일자(국제) 2014년02월27일

심사청구일자 2017년05월17일

(86) 국제출원번호 PCT/US2014/018827

(87) 국제공개번호 WO 2014/134234

국제공개일자 2014년09월04일

(30) 우선권주장

13/782,937 2013년03월01일 미국(US)

(56) 선행기술조사문헌

US20080228772 A1

(뒷면에 계속)

전체 청구항 수 : 총 20 항

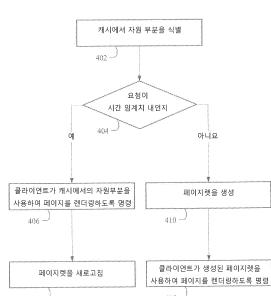
심사관 : 경연정

(54) 발명의 명칭 구조화 문서의 페이지렛의 캐싱

(57) 요 약

방법은 웹 페이지에 대한 요청을 수신하는 단계, 캐시에 저장된 하나 이상의 자원 부분을 식별하는 단계, 웹 페이지에 대한 이전의 요청 이후 기결정된 시간 임계치 내에 웹 페이지가 요청되었는지 여부를 결정하는 단계를 포함하며, 기결정된 시간 임계치 내에 웹 페이지가 요청되었다면, 캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 명령하는 단계, 및 하나 이상의 자원 부분을 새로고침하는 단계를 포함하고, 기결정된 시간 임계치 내에 웹 페이지가 요청되지 않았다면, 하나 이상의 자원 부분을 재생성하는 단계 및 재생성된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지가 렌더링되도록 명령하고 재생성된 하나 이상의 자원 부분을 캐시하는 단계를 포함하며, 상기 자원 부분 각각은 요청된 웹 페이지를 렌더링하는데 사용하기 위한 구조화 문서의 일부에 대응한다.

대 표 도 - 도4a



(52) CPC특허분류

G06F 17/2229 (2013.01)

(56) 선행기술조사문헌

US20110055683 A1

US7376653 A

US20110022984 A1

KR1020100057505 A

명세서

청구범위

청구항 1

서버 컴퓨팅 시스템에 의해, 클라이언트 컴퓨팅 장치로부터 웹 페이지에 대한 요청을 수신하는 단계;

서버 컴퓨팅 시스템에 의해, 캐시에 저장된 하나 이상의 자원 부분을 식별하는 단계;

서버 컴퓨팅 장치에 의해, 클라이언트 컴퓨팅 장치가 캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 명령하는 단계; 및

서버 컴퓨팅 장치에 의해, 하나 이상의 자원 부분을 새로고침(refresh)하는 단계를 포함하고, 새로고침하는 단계는:

만약 존재한다면, 하나 이상의 자원 부분에 대하여, 차분 데이터를 생성하는 단계;

클라이언트 컴퓨팅 장치에게, 차분 데이터의 생성과 병행하여, 하나 이상의 자원 부분을 렌더링하도록 명령하는 단계;

클라이언트 컴퓨팅 장치에게 차분 데이터를 전송하는 단계; 및

클라이언트 컴퓨팅 장치에게, 차분 데이터에 기반하여, 캐시에 저장된 자원 부분을 업데이트하고 업데이트된 자원 부분을 렌더링하도록 명령하는 단계를 포함하고,

자원 부분 각각은 클라이언트 컴퓨팅 장치에 의해 요청된 웹 페이지를 렌더링하는데 사용되기 위한 구조화 문서의 일부에 대응하고,

하나 이상의 자원 부분에 대한 차분 데이터는 하나 이상의 자원 부분에 만들어지는 변화를 표시하는 방법.

청구항 2

제 1 항에 있어서,

하나 이상의 자원 부분을 새로고침하는 단계는:

만약 존재한다면, 하나 이상의 자원 부분에 대한 업데이트된 데이터를 포함하는 하나 이상의 자원 업데이트 부분을 생성하는 단계; 및

클라이언트 컴퓨팅 장치로 하나 이상의 자원 업데이트 부분을 전송하는 단계를 더 포함하는 방법.

청구항 3

제 1 항에 있어서,

하나 이상의 자원 부분을 새로고침하는 단계는:

하나 이상의 대체 자원 부분을 생성하는 단계; 및

클라이언트 컴퓨팅 장치로 하나 이상의 대체 자원 부분을 전송하는 단계를 더 포함하고,

하나 이상의 대체 자원 부분은 하나 이상의 자원 부분 및 만약 존재한다면, 하나 이상의 자원 부분에 대한 업데이트된 데이터를 포함하는 방법.

청구항 4

제 1 항에 있어서,

하나 이상의 자원 부분의 상태에 영향을 주는 사용자 이벤트가 하나 이상의 자원 부분이 캐시에 마지막으로 저장된 이후에 발생했는지 여부를 결정하는 단계; 및

사용자 이벤트가 발생했다면, 하나 이상의 자원 부분의 상태에 영향을 주는 사용자 이벤트에 기초하여 캐시에

저장될 하나 이상의 자원 부분을 재생성하는 단계를 더 포함하는 방법.

청구항 5

제 1 항에 있어서,

캐시는 클라이언트 컴퓨팅 장치의 영속적인 저장소에 저장되는 방법.

청구항 6

제 5 항에 있어서,

하나 이상의 자원 부분은 암호화된 포맷으로 캐시에 저장되고,

캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하는 단계는 하나 이상의 자원 부분을 해독(decrypt)하는 단계를 포함하는 방법.

청구항 7

제 5 항에 있어서,

사용자 이벤트에 응답하여 캐시를 삭제하는 단계를 더 포함하는 방법.

청구항 8

제 7 항에 있어서,

사용자 이벤트는 웹 페이지가 요청된 웹사이트로부터 로그오프하는 것을 포함하는 방법.

청구항 9

하나 이상의 프로세서; 및

하나 이상의 프로세서와 연결되고 명령어를 포함하는 하나 이상의 컴퓨터 판독가능한 비일시적 저장 매체를 포함하고,

명령어는 하나 이상의 프로세서에 의해 실행될 때, 시스템으로 하여금:

클라이언트 컴퓨팅 장치로부터 웹 페이지에 대한 요청을 수신하고;

캐시에 저장된 하나 이상의 자원 부분을 식별하고;

클라이언트 컴퓨팅 장치가 캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 명령하고; 및

하나 이상의 자원 부분을 새로고침(refresh)하도록 야기하도록 동작하고, 새로고침하는 것은:

만약 존재한다면, 하나 이상의 자원 부분에 대하여, 차분 데이터를 생성하고;

클라이언트 컴퓨팅 장치에게, 차분 데이터의 생성과 병행하여, 하나 이상의 자원 부분을 렌더링하도록 명령하고;

클라이언트 컴퓨팅 장치에게 차분 데이터를 전송하고; 및

클라이언트 컴퓨팅 장치에게, 차분 데이터에 기반하여, 캐시에 저장된 자원 부분을 업데이트하고 업데이트된 자원 부분을 렌더링하도록 명령하는 것을 포함하고,

자원 부분 각각은 클라이언트 컴퓨팅 장치에 의해 요청된 웹 페이지를 렌더링하는데 사용되기 위한 구조화 문서의 일부에 대응하고,

하나 이상의 자원 부분에 대한 차분 데이터는 하나 이상의 자원 부분에 만들어지는 변화를 표시하는 시스템.

청구항 10

제 9 항에 있어서,

명령어는 실행될 때:

만약 존재한다면, 하나 이상의 자원 부분에 대한 업데이트된 데이터를 포함하는 하나 이상의 자원 업데이트 부분을 생성하고;

클라이언트 컴퓨팅 장치로 하나 이상의 자원 업데이트 부분을 전송하고; 및

클라이언트 컴퓨팅 장치가 캐시에 자원 업데이트 부분을 저장하고 하나 이상의 자원 부분과 하나 이상의 자원 업데이트 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 명령함으로써 하나 이상의 자원 부분을 새로고침하도록 더 동작가능한 시스템.

청구항 11

제 9 항에 있어서,

명령어는 실행될 때:

하나 이상의 대체 자원 부분을 생성하고;

클라이언트 컴퓨팅 장치로 하나 이상의 대체 자원 부분을 전송하고; 및

클라이언트 컴퓨팅 장치가 캐시의 하나 이상의 자원 부분을 하나 이상의 대체 자원 부분으로 대체하고 하나 이상의 대체 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 명령함으로써 하나 이상의 자원 부분을 새로고침하도록 더 동작가능하고,

하나 이상의 대체 자원 부분은 하나 이상의 자원 부분 및 만약 존재한다면, 하나 이상의 자원 부분에 대한 업데이트된 데이터를 포함하는 시스템.

청구항 12

제 9 항에 있어서,

명령어는 실행될 때:

하나 이상의 자원 부분의 상태에 영향을 주는 사용자 이벤트가 하나 이상의 자원 부분이 캐시에 마지막으로 저장된 이후에 발생했는지 여부를 결정하고; 및

사용자 이벤트가 발생했다면, 하나 이상의 자원 부분의 상태에 영향을 주는 사용자 이벤트에 기초하여 캐시에 저장될 하나 이상의 자원 부분을 재생성하도록 더 동작가능한 시스템.

청구항 13

제 9 항에 있어서,

캐시는 클라이언트 컴퓨팅 장치의 영속적인 저장소에 저장되는 시스템.

청구항 14

제 13 항에 있어서,

하나 이상의 자원 부분은 암호화된 포맷으로 캐시에 저장되고,

소프트웨어는 하나 이상의 자원 부분을 해독(decrypt)함으로써 캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 더 구성되는 시스템.

청구항 15

제 13 항에 있어서,

명령어는 실행될 때 사용자 이벤트에 응답하여 캐시를 삭제하도록 더 동작가능한 시스템.

청구항 16

제 15 항에 있어서,

사용자 이벤트는 웹 페이지가 요청된 웹사이트로부터 로그오프하는 것을 포함하는 시스템.

청구항 17

소프트웨어를 포함하는 하나 이상의 컴퓨터 판독가능한 비일시적 저장 매체로서,

소프트웨어는 실행될 때:

서버 컴퓨팅 시스템에 의하여, 클라이언트 컴퓨팅 장치로부터 웹 페이지에 대한 요청을 수신하고;

서버 컴퓨팅 시스템에 의하여, 캐시에 저장된 하나 이상의 자원 부분을 식별하고;

서버 컴퓨팅 시스템에 의하여, 클라이언트 컴퓨팅 장치가 캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 명령하고; 및

서버 컴퓨팅 시스템에 의하여, 하나 이상의 자원 부분을 새로고침(refresh)하도록 구성되고, 새로고침하는 것은:

만약 존재한다면, 하나 이상의 자원 부분에 대하여, 차분 데이터를 생성하고;

클라이언트 컴퓨팅 장치에게, 차분 데이터의 생성과 병행하여, 하나 이상의 자원 부분을 렌더링하도록 명령하고;

클라이언트 컴퓨팅 장치에게 차분 데이터를 전송하고; 및

클라이언트 컴퓨팅 장치에게, 차분 데이터에 기반하여, 캐시에 저장된 자원 부분을 업데이트하고 업데이트된 자원 부분을 렌더링하도록 명령하는 것을 포함하고,

자원 부분 각각은 클라이언트 컴퓨팅 장치에 의해 요청된 웹 페이지를 렌더링하는데 사용되기 위한 구조화 문서의 일부에 대응하고,

하나 이상의 자원 부분에 대한 차분 데이터는 하나 이상의 자원 부분에 만들어지는 변화를 표시하는 컴퓨터 판독가능한 비일시적 저장 매체.

청구항 18

제 17 항에 있어서,

소프트웨어는:

하나 이상의 자원 부분의 상태에 영향을 주는 사용자 이벤트가 하나 이상의 자원 부분이 캐시에 마지막으로 저장된 이후에 발생했는지 여부를 결정하고; 및

사용자 이벤트가 발생했다면, 하나 이상의 자원 부분의 상태에 영향을 주는 사용자 이벤트에 기초하여 캐시에 저장될 하나 이상의 자원 부분을 재생성하도록 더 구성되는 컴퓨터 판독가능한 비일시적 저장 매체.

청구항 19

제 17 항에 있어서,

캐시는 클라이언트 컴퓨팅 장치의 영속적인 저장소에 저장되는 컴퓨터 판독가능한 비일시적 저장 매체.

청구항 20

제 19 항에 있어서,

하나 이상의 자원 부분은 암호화된 포맷으로 캐시에 저장되고,

소프트웨어는 하나 이상의 자원 부분을 해독(decrypt)함으로써 캐시에 저장된 하나 이상의 자원 부분을 사용하여 요청된 웹 페이지를 렌더링하도록 더 구성되는 컴퓨터 판독가능한 비일시적 저장 매체.

발명의 설명

기술 분야

[0001]

본 명세서는 일반적으로 (웹 페이지와 같은) 구조화 문서의 렌더링에 관한 것이며, 특히 구조화 문서의 페이지 렌더링을 캐싱하고 효율적으로 구조화 문서를 렌더링하기 위해 캐시된 자원을 사용하며 반응 렌더링 시간을 감소시키는 것에 관한 것이다.

배경기술

[0002]

종래에는 클라이언트 장치에 의해 송신되는 웹 페이지 또는 다른 구조화 문서에 대한 요청이 웹 페이지를 호스팅하는 컴퓨팅 시스템 또는 서버에 의해 수신될 때, 호스팅 시스템은 통상적으로 하이퍼텍스트 마크업 언어 (HTML), 확장형 마크업 언어(XML) 또는 다른 웹 브라우저-지원 구조화 문서의 형태로 베이스(base) 웹 페이지를 생성한다. 이후, 생성된 구조화 문서는 하이퍼텍스트 전송 프로토콜(HTTP)이나 다른 적절한 연결을 통해 요청중인 클라이언트 장치에 응답하여 클라이언트 장치에서 렌더링을 위해 전송된다. 구조화 문서는 하나 이상의 자원 (예컨대, 자바스크립트(JavaScript)의 스크립트 또는 자원, 캐스케이딩 스타일 시트(Cascading Style Sheet, CSS) 자원, 이미지, 비디오 등) 또는 전송되는 문서 내에 내장되는 그런 자원에 대한 레퍼런스(reference)를 포함할 수 있다. 예로서, HTML 문서에 내장되는 자원은 일반적으로 자원의 유형에 따라 스크립트 구성요소, 이미지 구성요소 또는 객체 구성요소 등 내에 포함되거나 특정될 수 있다. 자원을 참조하거나 특정하는 구성요소는 자원의 위치를 웹 페이지를 요청중인 클라이언트 장치로 식별하는 소스 속성(예컨대, src)을 포함할 수 있다. 통상적으로, 응답을 수신하자마자, 클라이언트 장치에서 실행되는 웹 브라우저나 다른 클라이언트 애플리케이션은 이후 수신되는 구조화 문서의 DOM(Document Object Model) 표현을 구성하며, 그 문서에 내장된 (하나 이상의 다른 외부 위치에 있을 수 있는) 자원(들)을 요청한다.

[0003]

제다가, 원격 클라이언트에서 웹 컨텐츠를 열람하는 사용자가 현재 렌더링된 웹 페이지(예컨대, 현재 렌더링된 웹 페이지 내의 링크를 클릭하거나, 브라우저 애플리케이션의 뒤로 가기 또는 앞으로 가기 버튼을 클릭하거나, 타겟 웹 페이지의 URL을 입력함으로써)로부터 새로운(또는 "타겟") 웹 페이지로 탐색하기를 희망할 때, 웹 컨텐츠 렌더링을 담당하는 브라우저는 새로운 웹 페이지에 대한 요청을 형성(formulate)하고 새로운 웹 페이지를 호스팅하는 서버로 요청을 전송한다. 따라서, 종래에는 사용자가 새로운 웹 페이지를 탐색하기 위해 요청할 때마다 브라우저는 완전히 새로운 웹 페이지에 대한 요청을 서버로 전송하고, 현재 렌더링된 페이지를 언로드하며, 서버로부터 수신된 새로운 웹 페이지를 전체로서 렌더링한다. 종래에는, 이러한 완전한 페이지 로딩과 언로딩 기법은 사용자가 요청하는 이후의 페이지 각각에 대해 사실이다. 아래 놓여진 웹 페이지 내에 내장된 특정 자원 및 웹 페이지는 브라우저 캐시(cache)에 위치할 수 있고 국지적으로 검색될 수 있다. 하지만, 많은 동적 또는 상호작용적 웹 페이지는 그들이 최종적으로 렌더링된 이후 자주 업데이트되고 변경될 수 있는 다른 자원 및 컨텐츠를 포함한다. 종래에는 캐시된 페이지의 임의의 부분이 변경된다면, 캐시된 페이지 전체가 무효화되고 캐시로부터 지워진다.

발명의 내용

해결하려는 과제

[0004]

캐시된 발명의 대상은 원격 및/또는 로컬 데이터 스토어로부터의 캐시된 자원에 대한 업데이트를 검색하기 위해 비동기식 기술과 함께 캐시된 자원을 사용하여 웹 페이지 및 다른 구조화 문서를 효과적으로 렌더링하는 것에 관한 것이다. 특정 실시예는 또한, 하나 이상의 이전에 렌더링된 웹 페이지와 함께 다운로드 되었던 스크립트들을 재실행하고 (서버로부터 원격으로 접근 및/또는 캐시로부터 로컬하게 접근되던) 컨텐츠를 다시 로딩하는 것과 연관된 브라우저 오버헤드를 제거 또는 감소시키는 웹 페이지 또는 다른 구조화 문서를 렌더링하기 위한 시스템, 방법 및 논리에 관한 것이다. 특정 실시예는 브라우저 또는 밀리 놓여지는 클라이언트 애플리케이션이 불필요하게 원천적으로 전체 웹 페이지를 다시 렌더링하지 않도록 타겟 웹 페이지를 렌더링하는데 필요한 새로운 컨텐츠 및 자원을 요청하는 다양한 기술을 사용한다.

과제의 해결 수단

[0005]

다양한 실시예로, 캐시에 저장된 이후 캐시된 자원이 변경되었다면, 하나 이상의 증분적(incremental) 업데이트가 캐시된 자원을 새로고침(refresh)하기 위해 생성될 수 있다.

[0006]

다양한 실시예로, 캐시에 저장된 이후 캐시된 자원이 변경되었다면, 하나 이상의 새로운 대체 자원이 캐시된 자원을 새로고침하기 위해 생성될 수 있다.

발명의 효과

[0007]

본 발명의 내용 중에 포함되어 있다.

도면의 간단한 설명

[0008]

도 1은 예시적인 네트워크 환경을 도시한다.

도 2는 도 1의 예시적인 네트워크 환경의 예시적인 컴포넌트들의 블록도를 도시한다.

도 3은 웹 페이지에 대한 요청을 서빙(serve)하는 예시적인 방법을 도시하는 흐름도(flow chart)를 도시한다.

도 4a는 웹 페이지를 위한 페이지렛을 서빙하는 예시적인 방법을 도시하는 흐름도를 도시한다.

도 4b는 페이지렛을 새로고침하는 예시적인 방법을 도시하는 흐름도를 도시한다.

도 4c는 페이지렛을 새로고침하는 다른 예시적인 방법을 도시하는 흐름도를 도시한다.

도 5는 페이지렛을 렌더링하는 예시적인 방법을 도시하는 흐름도를 도시한다.

도 6은 예시적인 컴퓨터 시스템 구조를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0009]

특정 실시예는 원격 및/또는 로컬 데이터 스토어로부터의 캐시된 자원에 대한 업데이트를 검색하기 위한 비동기식 기술과 함께 캐시된 자원을 사용하여 웹 페이지 및 다른 구조화 문서를 효과적으로 렌더링하는 것에 관한 것이다. 특정 실시예는 또한, 컨텐츠를 다시 로딩(서버로부터 원격 및/또는 캐시로부터 로컬하게 접근되던지)하고 하나 이상의 이전에 렌더링된 웹 페이지와 함께 다운로드되는 스크립트를 재실행하는 것과 연관된 브라우저 오버헤드를 제거 또는 감소시키는 웹 페이지 또는 다른 구조화 문서를 렌더링하기 위한 시스템, 방법 또는 논리에 관한 것이다. 특정 실시예는 브라우저 또는 밑에 놓여지는 클라이언트 애플리케이션이 불필요하게 원천적으로 전체 웹 페이지를 다시 렌더링하지 않도록 타겟 웹 페이지를 렌더링하는데만 필요한 새로운 컨텐츠 및 자원을 요청하는 비동기식 자바스크립트 및 XML(AJAX) 기술을 사용한다. 특정 실시예는 클라이언트 장치에서 동작되는 브라우저의 컨텍스트 내에서 실행될 수 있고, 브라우저로부터 웹 페이지에 대한 요청을 가로채며 웹 페이지를 업데이트된 형태로 렌더링하기 위해 하나 이상의 캐시된 자원에 대한 충분적 업데이트에 대한 요청을 전송하는 프로세스에 관한 것이다. 다양한 예시적인 실시예에서, 하나 이상의 상술된 웹 페이지는 웹사이트 내에서 실행되는 소셜 네트워킹 서비스 또는 소셜 네트워킹 시스템과 연관될 수 있다. 하지만, 개시된 발명의 대상의 실시예는 네트워크 주소화 자원 또는 웹사이트의 임의의 유형에 의해 호스트되는 구조화 문서의 렌더링 및 검색에 대한 활용을 갖는다. 본 명세서에서 사용되는 것처럼, "사용자"는 개인, 그룹 또는 (사업체 또는 제 3 자 애플리케이션과 같은) 엔티티일 수 있다.

[0010]

특정 실시예는 다중 네트워크 주소화 시스템을 포함하는, 가령 인터넷과 같은 광역 네트워크 환경에서 동작할 수 있다. 도 1은 다양한 예시적인 실시예들이 동작할 수 있는 예시적인 네트워크 환경을 도시한다. 네트워크 클라우드(60)는 일반적으로 본 명세서에 기술된 다양한 시스템 및 호스트가 통신할 수 있는 하나 이상의 상호연결된 네트워크를 나타낸다. 네트워크 클라우드(60)는 패킷-기반 광역 네트워크(가령 인터넷), 사설 네트워크, 무선 네트워크, 위성 네트워크, 셀룰러 네트워크, 무선호출(paging) 네트워크 등을 포함할 수 있다. 도 1이 도시하는 바와 같이, 특정 실시예는 소셜 네트워킹 시스템(20)과 하나 이상의 클라이언트 장치(30)를 포함하는 네트워크 환경에서 동작할 수 있다. 클라이언트 장치(30)는 네트워크 서비스 제공자, 무선 캐리어 또는 임의의 적절한 다른 수단을 통해 네트워크 환경으로 동작가능하게 연결된다.

[0011]

예시적인 일실시예로, 소셜 네트워킹 환경(20)은 본 명세서에 기술된 바와 같이 사용자들이 서로 통신하거나 상호작용하며 가령 사용자 프로필과 같은 컨텐츠에 접속할 수 있도록 해주는 컴퓨팅 시스템을 포함한다. 소셜 네트워킹 시스템(20)은 다양한 예시적인 실시예에서 하나 이상의 물리적 서버(22) 및 데이터 스토어(24)를 포함하는 네트워크 주소화(network addressable) 시스템이다. 하나 이상의 물리적 서버(22)는 예컨대, 한 세트의 라우터 또는 네트워킹 스위치(26)를 통해 컴퓨터 네트워크(60)와 동작가능하게 연결된다. 예시적인 일실시예로, 하나 이상의 물리적 서버(22)에 의해 호스팅되는 기능은 웹 또는 HTTP 서버, FTP 서버뿐만 아니라, 제한 없이, 공통 게이트웨이 인터페이스(CGI) 스크립트를 사용하여 구현되는 웹 페이지와 애플리케이션, PHP 하이퍼-텍스트 전처리기(PHP), 액티브 서버 페이지(ASP), 하이퍼 텍스트 마크업 언어(HTML), 확장형 마크업 언어(XML), 자바,

자바스크립트, 비동기 자바스크립트와 XML(AJAX) 등을 포함할 수 있다.

[0012]

물리적 서버(22)는 소셜 네트워킹 시스템(20)의 동작에 관한 기능을 호스트 할 수 있다. 예로서, 소셜 네트워킹 시스템(20)은 하나 이상의 클라이언트 장치(30)에서 하나 이상의 사용자들이 정보를 열람 및 게시할 뿐만 아니라 웹사이트를 통해 서로 통신할 수 있도록 해주는 웹사이트를 호스트 할 수 있다. 서버(22)가 예컨대 소셜 네트워킹 시스템(20)을 호스팅하는 많은 서버들뿐만 아니라 다른 컨텐츠 분배 서버, 데이터 스토어 및 데이터베이스를 포함할 수 있지만, 이하에서, 서버들(22)은 서버(22)라고 한다. 데이터 스토어(24)는 디지털 데이터 객체로서 소셜 네트워킹 시스템의 동작과 관련되고 동작가능하게 해주는 컨텐츠 및 데이터를 저장할 수 있다. 특정 구현예로, 데이터 객체는 데이터 파일, 데이터베이스 또는 레코드에 통상 저장되거나 포함되는 디지털 정보의 아이템이다. 컨텐츠 객체는 텍스트(예컨대, ASCII, SGML, HTML), 이미지(예컨대, jpeg, tif 및 gif), 그래픽(벡터-기반 또는 비트맵), 오디오, 비디오(예컨대, mpeg) 또는 다른 멀티미디어, 및 이들의 조합을 포함하는 많은 형태를 취할 수 있다. 또한, 컨텐츠 객체 데이터는 실행가능한 코드 객체(예컨대, 브라우저 윈도우 또는 프레임 내의 실행가능한 게임), 팟캐스트(podcast) 등을 포함할 수 있다. 논리적으로, 데이터 스토어(24)는 하나 이상의 물리적 시스템에 저장된 논리적으로 관계된 레코드 또는 파일의 통합 모음으로서 정보를 유지하는, 가령 관계형 데이터베이스 및 객체-지향형 데이터베이스와 같은 하나 이상의 다양한 개별 또는 통합 데이터베이스에 해당한다. 구조적으로, 데이터 스토어(24)는 하나 이상의 많은 계층의 데이터 저장 및 관리 시스템을 일반적으로 포함할 수 있다. 특정 실시예로, 데이터 스토어(24)는 가령 하나 이상의 데이터베이스 서버, 대용량 저장 매체, 미디어 라이브러리 시스템, 저장 영역 네트워크, 데이터 저장 클라우드 등과 같은 컴퓨트를 포함하는 임의의 적합한 물리적 시스템(들)에 의해 구현될 수 있다. 예시적인 일실시예에서, 데이터 스토어(24)는 하나 이상의 서버, 데이터베이스(예컨대, MySQL) 및/또는 데이터 웨어하우스(warehouses)를 포함할 수 있다.

[0013]

데이터 스토어(24)는 상이한 소셜 네트워크 시스템(20) 사용자 및/또는 클라이언트 장치(30)를 포함할 수 있다. 특정 실시예로, 소셜 네트워킹 시스템(20)은 시스템(20)의 각각의 사용자의 사용자 프로필을 유지한다. 사용자 프로필은 예컨대, 적절한 이름(사람의 이름, 중간 이름 및 성, 상호 및/또는 사업체의 회사명 등), 인명, 신상 또는 이력, 학력, 취미 또는 기호, 지리적 위치 및 추가적인 설명 데이터를 포함할 수 있는 소셜 네트워크의 사용자를 설명하는 데이터를 포함한다. 예컨대, 사용자 프로필은 사용자의 생일, 관계 상태, 주거 도시 등을 포함할 수 있다. 시스템(20)은 또한 상이한 사용자들 사이의 하나 이상의 관계 또는 연결을 설명하는 데이터를 저장할 수 있다. 관계 정보는 유사한 또는 공통의 경력, 그룹 멤버쉽, 취미 또는 학력을 가진 사용자들을 표시할 수 있다. 사용자 프로필은 또한 다른 사용자들에 대한 사용자의 정보에 대한 접근을 통제하는 개인정보 설정을 포함할 수 있다.

[0014]

클라이언트 장치(30)는 일반적으로 컴퓨터 네트워크 상에서 (예컨대 원격으로) 통신하기 위한 기능을 포함하는 컴퓨터 또는 컴퓨팅 장치이다. 클라이언트 장치(30)는 다른 적절한 컴퓨팅 장치 외에도 데스크톱 컴퓨터, 노트북 컴퓨터, PDA(personal digital assistant), 차량 내외부의 내비게이션 시스템, 스마트폰 또는 다른 셀룰러 또는 휴대 전화 또는 모바일 게임 장치일 수 있다. 클라이언트 장치(30)는 가령 웹 브라우저(예컨대, 마이크로 소프트 윈도우 인터넷 익스플로러, 모질라 파이어폭스, 애플 사파리, 구글 크롬 및 오페라 등)와 같은 하나 이상의 클라이언트 애플리케이션을 실행하여 컴퓨터 네트워크를 통해 컨텐츠에 접근하고 컨텐츠를 열람할 수 있다. 특정 구현예로, 클라이언트 애플리케이션은 클라이언트 장치(30)의 사용자가 가령 소셜 네트워킹 시스템(20)에 의해 호스팅된 자원과 같은 검색되는 특정 네트워크 자원의 주소로 들어갈 수 있도록 해준다. 이를 주소는 URL(Uniform Resource Locators)일 수 있다. 추가적으로, 일단 페이지나 다른 자원이 검색되었다면, 클라이언트 애플리케이션은 사용자가 다른 자원으로의 하이퍼링크를 "클릭"할 때 다른 페이지나 레코드로의 접속을 제공할 수 있다. 예로서, 이런 하이퍼링크는 웹 페이지 내에 위치할 수 있으며, 사용자가 또 다른 페이지의 URL에 들어가고 그 페이지를 검색하는 자동화 방식을 제공할 수 있다.

[0015]

웹 페이지 내에 내장된 웹 페이지 또는 자원은, 자체적으로 다수의 내장된 자원을 포함할 수 있으며, 소프트웨어 프로그램 또는 다른 코드 객체, 그래픽, 이미지, 음성 신호, 비디오 등과 같은 보다 복잡하게 디지털 인코딩된 멀티미디어 컨텐츠 또는 평이한 텍스트 정보와 같은 데이터 레코드를 포함할 수 있다. 웹 페이지를 생성하기 위한 하나의 널리 알려진 마크업 언어는 HTML(Hypertext Markup Language)이다. 다른 일반적인 웹 브라우저가 지원되는 언어 및 기술은 XML(Extensible Markup Language), XHTML(Extensible Hypertext Markup Language), 자바스크립트, CSS(Cascading Style Sheet), 및 빈번하게 자바를 포함한다. 예컨대, HTML은 페이지 개발자가 페이지 내에 내장될 수 있는 이미지, 웹 애플리케이션 및 다른 객체뿐만 아니라 텍스트 및 링크에 대한 구조적 시맨틱(semantics)을 기재함으로써 구조화 문서를 생성할 수 있게 한다. 일반적으로, 웹 페이지는 정적 문서로 클라이언트에게 전달될 수 있지만, 페이지에 내장된 웹 구성요소의 사용을 통해, 상호작용적 경험이 페이지 또는

일련의 페이지로 이루어질 수 있다. 클라이언트에서의 사용자 세션 동안, 웹 브라우저는 잠재적으로 다른 웹사이트로부터의 자원뿐만 아니라, 페이지를 호스팅하는 웹사이트로부터 수신 또는 검색된 페이지 및 연관된 자원을 해석하고 디스플레이한다.

[0016] 더욱 특히, HTML은 개발자들이 웹 페이지와 같은 구조화 문서 내에 웹 애플리케이션, 이미지 또는 비디오를 포함하는 자원 또는 객체를 내장할 수 있게 한다. 일반적으로, HTML 구조화 문서는 구조화 문서 컨텐츠 내의 (꺾쇠 팔호안에 있는) 태그로 구성된 HTML 구성요소의 형태로 쓰여지는데, 이는 문서가 어떻게 웹 브라우저에 의해 해석되고 최종적으로 사용자 디스플레이 상에 제시되는지에 관해 구조화 문서를 렌더링하는 웹 브라우저에 대한 지시자(indicator)로서의 역할을 한다. 예로서, HTML 구성요소는 헤딩, 단락, 하이퍼텍스트 링크, 내장된 매체 및 다양한 다른 구조를 표현할 수 있다. HTML은 종래의 웹 브라우저와 같은 HTML 프로세서의 행동에 영향을 주는 자바스크립트와, 다른 컨텐츠 및 텍스트의 레이아웃 및 외관을 정의하는 CSS(Cascading Style Sheet)와 같은 언어의 스크립트를 로딩하거나 포함할 수 있다. HTML 구성요소는 HTML의 기본적인 컴포넌트이며 2 개의 기본 특성, 즉 속성(attribute) 및 컨텐츠를 갖는다. 각각의 구성요소의 속성 및 컨텐츠는 HTML 구성요소가 유효하다고 인정되는데 따른 일정한 제약을 가진다. HTML 구성요소는 일반적으로 시작 태그(예컨대, <element-name>) 및 종료 태그(예컨대, </element-name>)를 갖는다. 구성요소의 속성은 시작 태그에 포함되며, 컨텐츠는 태그 사이에(예컨대, <element-name attribute="value">Content</element-name>와 같이) 위치한다.

[0017] 예로서, HTML 구성요소는 (예컨대, 텍스트 또는 다른 컨텐츠의 목적을 설명하는) 구조적 구성요소, (예컨대, 기능과 무관하게 텍스트 또는 다른 컨텐츠의 외관을 설명하는) 프레젠테이션용 구성요소 및 (예컨대, 문서의 일부를 다른 문서에 대한 링크로 만드는) 하이퍼텍스트 구성요소를 포함한다. 대부분의 구성요소는 임의의 몇몇 일반적인 속성을 취한다. 예로서 id 속성은 구성요소에 대한 문서 전체에 고유한 식별자를 제공하고, 클래스 속성은 유사한 구성요소를 분류하는 방법을 제공하며, 타이틀 속성은 구성요소에 대한 서브텍스트 설명을 첨부하는데 사용된다. HTML은 또한 예컨대, ID들, 이름들, URI들, URL들, 개수, 길이의 단위, 언어, 매체 설명자, 색상, 문자 인코딩, 날짜 및 시간 등을 포함하는 속성 값들에 대한 다수의 유형과 스크립트 데이터 및 스타일시트 데이터와 같은 구성요소 컨텐츠에 대한 몇몇의 데이터 유형을 정의한다.

[0018] 문서 구조 구성요소는 (시작 및 종료 태그 <html> 및 </html> 각각에 의해 정의되는) 루트 구성요소, (시작 및 종료 태그 <head> 및 </head> 각각에 의해 정의되는) 헤드 구성요소, (시작 및 종료 태그 <body> 및 </body> 각각에 의해 정의되는) 바디 구성요소를 포함한다. 루트 구성요소 태그 <html> 및 </html>은 각각 HTML 문서의 시작과 끝의 범위를 정한다. 주어진 HTML 문서의 모든 다른 HTML 구성요소들은 루트 구성요소 내에 포함된다. 헤드 구성요소 태그 <head> 및 </head>는 일반적으로 HTML 문서에 대한 메타데이터 및 프로세싱 정보를 위한 컨테이너를 정의한다. 헤드 구성요소 컨테이너 내의 예시적인 문서 헤드 구성요소는 예로서 제한없이, (시작 및 종료 태그 <html> 및 </html> 각각에 의해 정의되고) HTML 문서에서 다른 링크들 및 모든 상대적 href에 대한 베이스 URL(uniform resource locator)을 특정하는 베이스 구성요소, (시작 및 종료 태그 <link> 및 </link> 각각에 의해 정의되고) 다른 문서(예컨대, 외부 CSS 파일에 대한)에 대한 링크를 특정하는 링크 구성요소, (시작 및 종료 태그 <meta> 및 </meta> 각각에 의해 정의되고) HTML 문서에 대한 추가적인 메타데이터를 특정하는데 사용되는 메타 구성요소, (시작 및 종료 태그 <object> 및 </object> 각각에 의해 정의되고) 문서 헤더 내의 일반적인 객체를 포함하도록 사용되는 객체 구성요소, (시작 및 종료 태그 <script> 및 </script> 각각에 의해 정의되고) (예컨대 자바스크립트 같은) 스크립트 명령어들을 위한 컨테이너 또는 src(소스) 속성을 갖는 외부 스크립트에 대한 링크로서 역할을 할 수 있는 스크립트 구성요소, (시작 및 종료 태그 <style> 및 </style> 각각에 의해 정의되고) 문서에 대한 스타일을 특정하고 (예컨대, 인라인된 CSS 규칙)과 같은 스타일 명령어들에 대한 컨테이너로서 역할을 할 수 있는 스타일 구성요소 및 (시작 및 종료 태그 <title> 및 </title> 각각에 의해 정의되고) 문서 타이틀을 정의하는 타이틀 구성요소를 포함한다.

[0019] 바디 구성요소 <body>는 HTML 문서의 디스플레이 가능한 컨텐츠를 위한 컨테이너를 표현한다. 예시적인 바디 구성요소는 예로서 제한 없이 (예컨대, 다른 것들뿐만 아니라 기본적인 텍스트 및 리스트 구성요소와 같은) 블록 구성요소, (예컨대, 앵커(anchor) 및 구(phrase)의 구성요소와 같은) 인라인 구성요소, 이미지 및 객체 구성요소를 포함한다. 바디 구성요소 내에 위치하는 스크립트 구성요소는 문서에서 스크립트를 배치하는데 사용될 수 있다(예컨대 스크립트 구성요소는 동적으로 블록 또는 인라인 컨텐츠를 생성하는 명령어들을 포함함). 이미지 구성요소는 (시작 및 종료 태그 및 각각에 의해 정의되며) 이미지를 문서 속에 삽입하는데 사용될 수 있다. 예컨대, 이미지 구성요소는 이미지가 위치하는 URL을 특정하는 소스 속성을 포함할 수 있다. 객체 구성요소는 (시작 및 종료 태그 <object> 및 </object> 각각에 의해 정의되며) 포함된 유형 속성에서 특정된 유형의 문서 속에 객체를 삽입하는데 사용될 수 있다. 또 다른 빈번하게 사용되는 HTML 구성요소는 프레임세트 구성

요소로서, 바디 구성요소에 대한 대안으로 사용될 수 있다.

[0020] 일반적으로 웹 애플리케이션은 네트워크 상에서 웹 브라우저 또는 다른 클라이언트 애플리케이션을 통해 접근될 수 있는 애플리케이션 또는 애플리케이션을 실행가능하게 만들기 위해 웹 브라우저에 의존적이며 웹 브라우저를 지원하는 언어로 코딩된 컴퓨터 소프트웨어 애플리케이션이다. 웹 애플리케이션은 웹 브라우저의 유비쿼터(ubiquity), 클라이언트(때때로, 썬 클라이언트(thin client)라고도 하는)로서 원격 컴퓨팅 장치에서 런치되는 웹 브라우저를 사용하는 편리함 및 원격 클라이언트 상에서 소프트웨어를 설치하고 배포함이 없이 웹 애플리케이션을 유지하고 업데이트할 대응하는 능력으로 인하여 널리 인기를 얻고 있다. 종종 웹 애플리케이션을 구현하기 위해, 웹 애플리케이션은 연관된 웹사이트의 후단 서버에서 제공되는 하나 이상의 자원에 대한 액세스를 요구한다. 추가적으로, 웹 애플리케이션은 종종 다른 애플리케이션과 연관된 추가적인 자원에 대한 액세스를 요구한다.

[0021] 소셜 네트워킹 시스템(20)은 다수의 특징들을 포함하며, 원격 클라이언트(30)에서의 사용자는 특징들로 사용자 세션 동안 상호작용할 수 있다. 특정 실시예로, 이러한 특징들은 웹 애플리케이션으로서 구현될 수 있고, 다른 외부 서버 또는 데이터 스토어 뿐만 아니라 서버(22)로부터 요청된 자바스크립트 및 CSS 자원을 사용할 수 있다. 웹 애플리케이션 또는 자원은 프레임 또는 iFrame, 섹션 또는 "divs" 등에서 원격 클라이언트에 대해 서빙되는 다양한 웹 페이지에서 내장될 수 있다. 특정 실시예로, 소셜 네트워킹 시스템(20)은 상이한 종류의 아이템들을 위한 많은 수의 객체들을 데이터 스토어(24)에 유지할 수 있고, 아이템들로 사용자는 소셜 네트워킹 시스템(20)에 접근하는 동안 상호작용할 수 있다. 일실시예로, 이러한 객체는 사용자 프로필, 애플리케이션 객체, 및 (담벼락 게시물, 이메일 및 다른 메시지와 같은) 메시지 객체를 포함한다. 일실시예로, 객체는 그것의 연관된 아이템의 각각의 인스턴스(instance)에 대해 시스템(20)에 의해서 저장된다. 본 명세서에서 논의되는 이러한 객체 및 행위는 예시적인 목적으로 제공되는 것이며, 제한되지 않은 개수의 변형 및 특징들이 소셜 네트워킹 시스템(20) 상에 제공될 수 있는 것으로 이해된다.

[0022] 전술한 바와 같이, 웹 페이지 또는 아래 놓여지는 구조화 문서는 논리적, 시각적 또는 다른 방식으로 섹션으로 세그먼트화 또는 분할될 수 있다. 예로서, 웹 페이지를 인코딩하는데 사용되는 구조화 문서는 시작 및 종료 HTML <div> 태그에 의해 나타나는 하나 이상의 블록-수준의 구성요소를 포함할 수 있다. 배경으로서 일반적인 웹 브라우저에 있어서 웹 페이지의 디스플레이가능한 구성요소는 블록 또는 인라인으로서 렌더링될 수 있다. 모든 구성요소가 문서 시퀀스의 일부인 한편, 블록 구성요소는 그들의 부모 구성요소 내에서 사각형의 객체로서 나타나며, 라인을 가로지르지 않으며, 블록 마진, 폭 및 높이 특성을 갖고, 이러한 특성은 주위 구성요소와는 독립적으로 설정될 수 있다. 반면에, 인라인 구성요소는 문서 텍스트의 흐름의 일부로서 취급되며, 그들은 마진, 폭 또는 높이 세트를 가질 수 없으며, 선을 가로지를 수 있다. 인라인 구성요소는 바디 구성요소 내에 직접 놓일 수 없고, 그들은 블록-수준의 구성요소 내에 온전히 네스트(nest) 되어야만 한다.

[0023] 다른 예로서, 웹 페이지는 또한 (예컨대, 시작 및 종료 태그 <frame> 및 </frame> 각각에 의해 나타나는)HTML 프레임 구성요소를 사용하여 구조적으로 특정된 하나 이상의 프레임으로 분할될 수 있다. 프레임은 웹브라우저 디스플레이 윈도우가 세그먼트(segment)로 분할되도록 하며, 세그먼트들 각각은 상이한 문서를 보여준다. 다른 프레임 구성요소는 (시작 및 종료 태그 <iframe> 및 </iframe> 각각에 의해 나타나는) 인라인 프레임 구성요소이다. 인라인 프레임은 다른 HTML 구조화 문서를 프레임에 배치한다. 객체 구성요소와는 달리, 인라인 프레임은 다른 구성요소에 의해 정의된 링크에 대해 "타겟" 프레임일 수 있다.

[0024] 이하에서, 그 중에서도 블록-수준 구성요소, 프레임 또는 인라인 프레임과 같은 웹 페이지를 인코딩하는데 사용되는 구조화 문서 또는 웹 페이지의 임의의 논리적, 구조적 또는 시각적 섹션 또는 부분 각각은 이하에서 "페이지렛(pagelet)"이라고 언급될 수 있다.

[0025] 클라이언트 장치(예컨대, 클라이언트 장치(30)와 같은) 클라이언트 장치에서의 사용자가 적어도 부분적으로 소셜 네트워킹 시스템(20)에 의해 호스트되는 특정 웹 페이지(이하 "타겟 구조화 문서"라고도 한다)를 보기를 희망할 때, 사용자의 웹브라우저 또는 다른 클라이언트측 문서-렌더링 엔진 또는 적절한 클라이언트 애플리케이션은 소셜 네트워킹 시스템(20)으로 요청을 생성하고 전송한다. 요청은 일반적으로 메타데이터 또는 다른 정보뿐만 아니라 URL 또는 다른 문서 식별자를 포함한다. 예로서, 요청은 사용자의 클라이언트 컴퓨팅 장치(30)에서 실행되는 운영 체제 또는 웹 브라우저를 식별하거나 특징짓는 정보뿐만 아니라 사용자 ID와 같은 사용자를 식별하는 정보를 포함할 수 있다. 요청은 또한, 사용자 클라이언트 장치의 논리 네트워크 위치 또는 사용자의 클라이언트 장치의 지리적 위치를 식별하는 위치 정보를 포함할 수 있다. 요청은 또한 요청이 언제 전송되었는지를 식별하는 타임스탬프(timestamp)를 포함할 수 있다.

[0026]

이제 도 2의 블록도 및 도 3의 흐름도를 참조하여 웹 페이지에 대한 요청을 서빙하는 방법이 설명될 것이다. 예시적인 실시예로, 방법은 소셜 네트워킹 시스템(20)에서의 다른 컴퓨팅 시스템 또는 서버(22)에 의해 클라이언트 장치(30)에 있는 클라이언트 애플리케이션으로부터 소셜 네트워킹 시스템(20)에 의해 적어도 부분적으로 호스팅되는 웹 페이지에 대한 요청을 수신하는 단계(302)로 시작한다. 전술한 바와 같이, 요청은 일반적으로 메타데이터 또는 다른 정보뿐만 아니라 웹 페이지 위치에 대응하는 다른 문서 식별자 또는 URL을 포함한다. 예로서, 요청은 사용자의 클라이언트 컴퓨팅 장치(30)에서 실행되는 운영 체제 또는 클라이언트측 문서-렌더링 애플리케이션(예컨대, 웹 브라우저)(202)를 특징짓거나 식별하는 정보뿐만 아니라 사용자 ID와 같은 요청을 하는 클라이언트 애플리케이션의 사용자를 식별하는 정보를 포함할 수 있다. 요청은 또한, 사용자의 클라이언트 장치의 지리적 위치 또는 사용자의 클라이언트 장치의 논리 네트워크 위치를 식별하는 위치 정보를 포함할 수 있다. 요청은 또한 언제 요청이 전송되었는지를 식별하는 타임스탬프를 포함할 수 있다. 인증 프로세스(204)는 우선 요청을 생성한 사용자가 요청된 웹 페이지를 수신하도록 인증되는지 여부(예컨대, 사용자가 성공적으로 로그인했고 사용자가 접근 또는 관리 권한을 갖는 페이지를 요청하는지 여부)를 결정(304)한다.

[0027]

특정 실시예로, 웹 페이지 서빙 프로세스(206)는 이후 요청을 분석하고, 마크업 언어 코드(예컨대, HTML, XML 또는 다른 적절한 마크업 언어 코드)를 사용하여 베이스(뼈대) 구조화 문서를 생성(306)한다. 특정 실시예로, 베이스 구조화 문서는 헤드 구성요소(예컨대, HTML 헤드 구성요소) 및 바디 구성요소(예컨대, HTML 바디 구성요소)를 포함한다. 특정 실시예로, 헤드 구성요소는 요청하는 클라이언트(30)에서 페이지 조립 프로세스(208)를 구현하기 위한 실행가능한 코드 세그먼트를 포함한다. 특정 실시예로, 페이지 조립 프로세스(208)를 구현하기 위한 실행가능한 코드 세그먼트는 자바스크립트 코드 세그먼트이며, 자바스크립트 함수 라이브러리를 포함한다. 헤드 구성요소는 하나 이상의 최초 자원(예컨대, 자바스크립트 또는 CSS) 또는 외부 위치(예컨대, 제 3 자 사이트)로부터 다운로드되는 이러한 자원들에 대한 (예컨대, 선택된 자원을 찾기 위한 대응하는 소스(src) 식별자를 갖는 객체 구성요소, 이미지 또는 스크립트의 형태의) 레퍼런스를 포함한다. 특정 실시예로, (예컨대, 베이스 구조화 문서를 요청하는 클라이언트(30)에게 송신하는 것을 가속화하기 위해) 베이스 구조화 문서에 포함된 최초 자원의 개수를 최소화하는 것이 바람직하다. 특정 실시예로, 바디 구성요소는 요청하는 클라이언트(30)에 의해 일단 수신되는 하나 이상의 플레이스-홀더(place-holder)를 렌더링하기 위한 마크업 언어 코드를 포함한다. 특정 실시예로, 페이지 서빙 프로세스(206)는 이하에서 보다 상세히 설명되는 것과 같이 대응하는 2차 반응에서 클라이언트(30)로 전송되는 하나 이상의 (일반적으로 복수의) 페이지셋 각각에 대해 플레이스-홀더를 베이스 구조화 문서 내에 추가한다. (비록 특정 실시예에서, 자원, 자원에 대한 레퍼런스 또는 컨텐츠는 베이스 구조화 문서에서 최소화되지만) 바디 구성요소는 또한 자원, 자원에 대한 레퍼런스 또는 심지어는 요청하는 클라이언트(30)에 의해 일단 수신된 렌더링될 컨텐츠를 포함할 수 있다. 특정 실시예로, 웹 페이지 서빙 프로세스(206)는 308에서 이후 베이스 구조화 문서를 포함하는 초기 응답 또는 초기 응답 부분(이하에서 "제 1 응답" 또는 "제 1 응답 부분"이라고도 함)을 생성하고 요청하는 클라이언트(30)에게 전송한다(일부 실시예에서, 예컨대, 프리-페칭(pre-fetching) 명령어가 최초/제 1 응답/응답 부분을 전송하기 전에 클라이언트로 전송될 수 있음을 유의해야 한다). 초기 응답은 HTTP 또는 임의의 다른 적절한 연결로 요청하는 클라이언트(30)에게 전송될 수 있다. 특정 실시예로, 초기 응답이 클라이언트(30)로 전송되는 연결은 영속적 전송 제어 프로토콜 연결이다. 후술한 바와 같이, 이것은 클라이언트(30)가 베이스 구조화 문서의 프로세싱을 시작하고 수신하며, 요청된 웹 페이지의 나머지가 생성되는 동안 초기 응답에서의 자원을 다운로드 또는 초기화할 수 있게 한다.

[0028]

특정 실시예로, 306에서 베이스 구조화 문서를 생성하거나 308에서 초기 응답을 전송하는 것 이후 또는 심지어는 적어도 부분적으로 병렬적으로, 웹 페이지 서빙 프로세스(206)는 페이지 생성 프로세스(210)를 310에서 런칭하고 그것이 요청된 웹 페이지의 나머지를 생성하도록 명령한다. 특정 일실시예로, 페이지 생성 프로세스(210)는 웹 페이지의 할당된 하위 부분을 생성하도록 각각 구성된 복수의 서브페이지 생성 프로세스를 관리하는 마스터 프로세스를 포함한다. 즉, 특정 실시예로, 하나 이상의 서브페이지 생성 프로세스 각각은 다른 서브페이지 생성 프로세스와 병렬적으로 할당된 페이지셋을 생성하는 것을 개시한다. 특정 실시예로, 각각의 서브페이지 생성 프로세스가 그것의 할당된 페이지셋의 생성을 완료함에 따라 페이지 생성 프로세스(210)는 페이지셋을 페이지 서빙 프로세스(206)로 전달하고, 이후 페이지셋을 2차 응답 또는 2차 응답 부분(이하 "제 2 응답" 또는 "제 2 응답 부분"이라고도 함)으로서 생성하고 요청하는 클라이언트(30)로 전송(312)한다. 대안적인 실시예로, 페이지 서빙 프로세스(206)는 페이지 생성 프로세스(210)를 포함할 수 있고, 일반적으로 하나 이상의 다른 페이지 생성 프로세스를 런칭하는 것을 수반하지 않고 인-프로세스로 연속적으로 각각의 2차 부분을 생성할 수 있다. 단계(312)는 단일의 단계(312)로 설명되지만, 일반적으로 일부 실시예에서, 각각의 페이지셋이 생성됨에 따라 클라이언트(30)로 직렬 또는 병렬로 전송되는 대응하는 복수의 2차 응답에서 복수의 페이지셋을 생성하고 전송하는 것을 포함할 수 있다. 전술한 바와 같이, 각각의 2차 응답은 초기 응답이 전송된 다른 적절한 연결 또는

동일한 영속적 TCP 상에서 클라이언트(30)로 전송될 수 있다. 일실시예로, 베이스 구조화 문서의 바디 구성요소는 종료 태그로 마감되지 않을 수 있고, 따라서, 각각의 2차 응답은 베이스 구조화 문서의 바디 구성요소의 이후에 생성되고 이후에 전송된 부분으로서 전송될 수 있다. 다른 실시예로, 하나 이상의 2차 응답은 하나 이상의 독립적인 HTML 응답으로서 전송될 수 있다.

[0029] 특정 실시예로, 요청하는 클라이언트(30) 및 특히 문서-렌더링 애플리케이션(202)(예컨대, 웹브라우저)이 초기 응답을 수신할 때, 문서-렌더링 애플리케이션(202)은 초기 응답 및 베이스 구조화 문서를 프로세스하고, 클라이언트(30)에서의 메모리에 베이스 구조화 문서의 모델 표현을 생성한다. 예로서, 문서-렌더링 애플리케이션(202)은 베이스 구조화 문서의 문서 객체 모델(Document Object Model; DOM) 표현(212)을 생성한다. 통상의 기술자에 의해 이해될 것처럼, 문서-렌더링 애플리케이션(202)이 DOM 표현(212)을 생성할 때, 그것은 필연적으로 마크업 언어 코드를 DOM 트리 또는 DOM 구성요소 또는 노드의 DOM 계층구조로 해석하고, 이를 각각은 다른 가능성들 중에서도 자원, 자원에 대한 레퍼런스 또는 컨텐츠를 포함하거나 수록할 수 있다. 특정 실시예로, 베이스 구조화 문서는 문서-렌더링 애플리케이션(202)이 베이스 구조화 문서에서 특정된 플레이스-홀더 각각에 대한 하나 이상의 DOM 노드를 생성하도록 구성된다. 이하에서, 이러한 DOM 노드들은 "플레이스-홀더 DOM 노드"라고 불리운다. 일반적으로, 각각의 플레이스-홀더 DOM 노드는 임의의 자원, 자원에 대한 레퍼런스 또는 디스플레이될 컨텐츠를 포함하지 않지만, 이러한 플레이스-홀더 DOM 노드는 문서-렌더링 애플리케이션(202)에 의해 렌더링되고, 일반적으로 디스플레이될 때, 이하에서 보다 자세히 후술할 것처럼, 다음의 수신된 자원, 자원에 대한 레퍼런스 또는 다음의 수신된 2차 응답에서의 연관된 페이지렛에서의 컨텐츠가 디스플레이, 초기화 또는 실행될 렌더링된 베이스 구조화 문서에서의 위치를 필수적으로 예약한다.

[0030] 특정 실시예로, (예컨대, 자바스크립트와 같은) 페이지 조립 프로세스(208)를 구현하기 위한 코드 세그먼트는 문서 렌더링 애플리케이션(202)에 의해 프로세싱되자 마자 초기화된다. 초기화되자 마자, 페이지 조립 프로세스(208)는 이후 전술한 다음으로 수신된 2차 응답을 기다릴 수 있다. 초기 응답에 포함된 임의의 다른 자원 또는 자원에 대한 레퍼런스는 또한 문서 렌더링 애플리케이션(202)에 의해 수신되자 마자 프로세싱, 초기화, 실행 또는 다운로드 될 수 있다.

[0031] 특정 실시예로, 2차 응답의 각각이 문서 렌더링 애플리케이션(202)에 의해 수신됨에 따라, 페이지 조립 프로세스(208)는 (페이지 조립 프로세스(208) 또는 문서 렌더링 애플리케이션(202)에 의해 접근가능한 임시 또는 다른 적절한 메모리 위치에서 구현될 수 있는) 큐(214)에 각각의 대응하는 페이지렛의 컨텐츠를 배치할 수 있다. 다음으로, 페이지 조립 프로세스(208)는 각각의 수신된 2차 응답에 대해, DOM 표현(212)에서 대응하는 플레이스-홀더 DOM 노드 (또는 노드들)을 식별하고, 대응하는 플레이스-홀더 DOM 노드에서 또는 그 대신에 페이지렛의 컨텐츠를 DOM 표현(212)으로 동적으로 삽입 또는 스왑할 수 있다. 특정 실시예로, 페이지 조립 프로세스(208)는 문서 렌더링 애플리케이션(202)과 함께 페이지렛의 컨텐츠를 DOM 표현(212)에 동적으로 삽입 또는 스왑할 수 있다; 즉, 문서 렌더링 애플리케이션(202)은 페이지 조립 프로세스(208)로부터의 호출에 응답하여 실제로 컨텐츠를 삽입 또는 스왑할 수 있다. 삽입 또는 스왑은 (예컨대, 페이지렛에 대응하는 플레이스-홀더 DOM 노드의 레벨에서 또는 그보다 낮게 낮은 순위의 DOM 노드를 DOM 표현(212)에서 생성하는 것과 같은) DOM 표현(212)에서의 추가적인 DOM 노드의 생성을 수반할 수 있음을 유의해야 한다. 각각의 2차 응답의 컨텐츠가 DOM 표현(212)으로 삽입됨에 따라, 삽입되는 컨텐츠 또는 자원은 이후 렌더링, 실행, 초기화 또는 문서 렌더링 애플리케이션(202)에 의해 렌더링하기 위해 다운로드될 수 있고, 사용자에게 디스플레이될 수 있다.

[0032] 특정 실시예로, 장치 캐시 컴포넌트(220)는 예컨대, 클라이언트(30)에서의 임시 메모리 또는 영속적 저장소 내에서 문서 렌더링 애플리케이션(202)에 의해 수신된 페이지렛을 캐시할 것이고, 사용자가 예컨대, 후술할 바와 같이 현재 렌더링된 웹 페이지를 다시 로딩하거나 현재 렌더링된 웹 페이지로부터 다른 웹 페이지로 탐색하고 이후 이전에 렌더링된 웹 페이지로 돌아가기를 희망할 때, 하나 이상의 자원이 이용될 수 있다. 일부 실시예로, 장치 캐시 컴포넌트(220)는 페이지렛이 캐시가능한지, 페이지렛이 이전에 캐시되었는지, 언제 페이지렛이 이전에 캐시되었는지 및/또는 캐시된 페이지렛이 서버(22) 상에 업데이트되었는지 여부 및 그에 따라 클라이언트(30) 상에 업데이트 되어야 하는지 여부와 같은 정보를 저장할 수 있다.

[0033] 특정 실시예로, 각각의 페이지렛은 클라이언트(30)에 의해 일단 수신된 페이지 조립 프로세스(208)를 호출하도록 구성된 함수 호출을 포함하거나 함수 호출의 포맷이다. 특정 실시예로, 페이지 조립 프로세스(208)는 이후 수신되는 2차 응답에서 함수 호출에 의해 호출될 또는 호출될 수 있는 (예컨대, 자바 스크립트) 함수를 포함하는 (예컨대, 자바스크립트 함수 라이브러리와 같은) 함수 라이브러리를 포함한다. 특정 실시예로, 각각의 페이지렛은 대응하는 함수 호출을 구현하기 위한 스크립트를 포함하며, 페이지 조립 프로세스(208)의 함수 라이브러리에 의해 요구되는 함수를 식별하는 (예컨대, HTML과 같은) 몇몇 마크업 언어 코드를 포함한다. 특정

실시예로, 각각의 함수 호출은 HTML 또는 다른 마크업 언어 코드, 자바스크립트 자원의 어레이, CSS 자원의 어레이 및 예컨대, 페이지렛이 디스플레이되고 페이지렛에서 식별된 자원이 다운로드된 후에 실행될 (예컨대, 자바스크립트와 같은) 스크립트 또는 코드 세그먼트의 리스트를 포함하는 레지스터를 포함하는 하나 이상의 콜백 함수에서 래프된(wrapped) 인수(argument)를 포함한다.

[0034] 특정 실시예로, 각각의 함수 호출은 또한 페이지렛의 몇몇의 특성을 설명하고 페이지렛의 상대적인 우선순위(priority)를 결정하기 위해 페이지 조립 프로세스(208)에 의해 사용될 수 있는 몇몇의 메타데이터를 포함한다. 이것은 페이지 조립 프로세스(208)가 2차 응답에서 수신된 각각의 페이지렛의 상대적인 디스플레이 순서를 스케줄하는 것을 가능하게 한다. 예컨대, 일부 페이지렛은 피처가 중요하지 않거나 다른 피처 또는 컨텐츠만큼 중요하지 않다고 미리결정할 수 있고, 그리하여 이러한 페이지렛은 다른 더 중요한 페이지렛 이후에 렌더링될 수 있다. 단순한 일구현예에서, 각각의 함수 호출은 또한 "지연 플래그"를 포함한다. 이 지연 플래그가 트루(true)로 설정되면, 이것은 페이지 조립 프로세스(208)에게 페이지렛이 상대적으로 중요하지 않고 다른 높은 우선순위의 페이지렛이 삽입되고 렌더링될 때까지 그것의 DOM 표현(212)으로의 삽입 및 문서 렌더링 애플리케이션(202)에 의한 이후의 렌더링이 지연될 수 있다고 명령한다. 더욱 특히, 이하에서 더 자세히 후술되듯이, 각각의 페이지렛 콜백 함수는, 클라이언트(30)에서 수신될 때, 다른 높은 우선순위의 페이지렛 콜백 함수가 프로세싱 또는 실행될 때까지 큐(214)에서 수용될 수 있다. 하지만, 일부 실시예, 큐(212)에 다른 더 높은 우선순위의 페이지렛이 없다면, 더 낮은 우선순위의 페이지렛 콜백 함수가 프로세싱될 수 있다.

[0035] 특정 실시예로, 전술한 바와 같이, 2차 응답에서 수신된 특정 페이지렛의 컨텐츠를 DOM 표현(212)으로 삽입하기 위해, 페이지 조립 프로세스(208)는 DOM 표현(212)에서 대응하는 플레이스-홀더 DOM 노드를 식별해야 한다. 특정 실시예로, 이것은 페이지 서빙 프로세스(206)에 의해 베이스 구조화 문서에서 각각의 플레이스-홀더와 함께 식별자(ID) 파라미터를 포함함으로써 이루어질 수 있다. 유사하게, 페이지 생성 프로세스(210)에 의해 생성된 각각의 페이지렛은 페이지렛의 컨텐츠가 삽입 또는 스왑되는 DOM 표현(212)에서의 일치 또는 대응하는 플레이스-홀더 DOM 노드를 식별하기 위해 페이지 조립 프로세스(208)에 의해 이용가능한 ID 파라미터를 포함한다. 구조화 문서의 생성 및 전달에 관한 추가적인 세부사항은 본 명세서에 전체로서 참조로 통합되는 미국 특허출원 번호 제12/754,549호에서 찾을 수 있을 것이다.

[0036] 페이지렛을 서빙하는 방법이 도 4a의 흐름도를 참조하여 이제 설명될 것이다. 전술한 바와 같이, 웹 페이지 요청에 응답하여, 베이스 구조화 문서를 생성(306)하거나 초기 응답을 전송(308)하는 것 이후 또는 병렬적으로, 웹 페이지 서빙 프로세스(206)는 310에서 페이지 생성 프로세스(210)를 런칭(206)하고 그것이 요청된 웹 페이지의 잔존물을 생성하도록 명령한다. 특정 실시예로, 요청된 웹 페이지가 이전에 렌더링되어서 웹 페이지를 렌더링하는데 사용하기 위한 클라이언트(30)에 의한 캐시에서 캐시 또는 저장이 된 대응하는 페이지렛을 갖는 경우 ("캐시 히트")에, 웹 페이지 서빙 프로세스(206)은 클라이언트(30)가 캐시된 자원을 적어도 부분적으로 사용하여 페이지렛을 렌더링하도록 명령할 수 있다. 예컨대, 도 4a를 참조하면, 클라이언트 장치(30)가 이미 렌더링된 웹 페이지를 요청할 때, 웹 페이지 서빙 프로세스(206)는 캐시 히트가 있는지 여부를 먼저 결정(402)할 수 있다; 즉, 요청된 웹 페이지가 이미 클라이언트(30)에 의해 렌더링되었고 캐시에 저장되었다면, 웹 페이지 서빙 프로세스(206)는 웹 페이지를 렌더링하는데 사용될 캐시에서 저장된 하나 이상의 페이지렛 또는 페이지렛 자원을 식별할 수 있다. 예시의 실시예에서, 캐시 히트가 없다면, 웹 페이지 서빙 프로세스(206)는 페이지 생성 프로세스(210)를 런칭하고 그것이 본 명세서에 기술되는 것처럼 요청된 웹 페이지의 나머지를 생성하도록 명령한다.

[0037] 특정 실시예로, 웹 페이지 서빙 프로세스(206)는 언제 웹 페이지에 대한 요청이 서버로 전송되었는지, 언제 응답이 서버(22)로부터 제공되었는지 및/또는 언제 웹 페이지가 클라이언트(30)에 의해 렌더링되었는지를 표시하는 타임스탬프를 저장할 수 있다. 캐시에 저장된 하나 이상의 페이지렛이 식별된 후에, 웹 페이지 서빙 프로세스(206)는 캐시된 페이지렛의 일부 또는 전부가 유효기간이 지나서 쓸모가 없어진 것인지 결정할 수 있다. 예컨대, 웹 페이지 서빙 프로세스(206)는 404에서 현재의 요청에 대한 타임스탬프와 이전의 요청에 대한 타임스탬프 사이의 경과 시간이 기결정된 시간 임계치를 초과하는지 여부를 결정하기 위해, 웹 페이지에 대한 현재 요청의 타임스탬프와 웹 페이지에 대한 이전의 요청에 대해 이루어진 타임스탬프를 비교할 수 있다. 기결정된 시간 임계치는 예컨대 제한 없이 5분과 같이 임의의 소정의 시간일 수 있다.

[0038] 특정 실시예로, 시스템 캐시 캠포넌트(218)는 추가적인 데이터를 쓰는 것과 같은 장치(30)의 캐시에 저장된 페이지렛 상에 수행되는 하나 이상의 상태 변화 동작을 모니터 및/또는 기록할 수 있다. 예컨대, 시스템 캐시 캠포넌트(218)는 동작이 수행된 페이지렛에 대해 저장된 대응하는 데이터 객체 내에 내장된 스크립트에 의해 레지스터되는 콜백 함수와 연관하여 (일실시예로, 내장된 스크립트에서 객체로서 구체화되는) 상태 변화 동작을 저

장할 수 있다. 이러한 방식으로, 사용자가 동작이 수행된 페이지로부터 떠나 탐색하고, 이후에 동작이 수행된 페이지로 돌아와 탐색할 때에, 시스템 캐시 컴포넌트(218)는 콜백 함수를 클라이언트(30)에 2차 응답 부분으로서 전송하도록 웹 페이지 생성 프로세스(206)에 명령할 수 있고, 이후 캐시된 페이지의 문맥에서 동작을 재실행 한다. 상태 변화 동작이 이전에 수행된 페이지에 대한 캐시 히트에 응답하여, 페이지 생성 프로세스(208)는 대응하는 데이터 객체를 접근할 수 있고, 본 명세서에 기술되는 것처럼 페이지에 대한 캐시된 자원을 다시 로딩하거나 다시 실행할 수 있다. 일부 실시예에서, 웹 페이지 서빙 프로세스(206) 또는 시스템 캐시 컴포넌트는 수신된 상태 변화 정보에 적어도 부분적으로 기초하여 캐시된 페이지렛이 쓸모가 없다(obsolete)고 결정할 수 있다.

[0039] 특정 실시예로, 404에서 캐시된 페이지렛이 쓸모 없지 않다고 결정된다면, 웹 페이지 서빙 프로세스(206)는 도 5와 관련하여 후술되는 것처럼 클라이언트(30)가 캐시로부터의 페이지렛의 일부 또는 전부를 렌더링하도록 명령할 수 있다. 예컨대, 웹 페이지 서빙 프로세스(206)는 페이지렛을 렌더링하는데 사용될 자원의 일부 또는 전부가 캐시에 저장된다는 것을 표시하는 2차 응답을 클라이언트(30)에게 전송할 수 있다.

[0040] 특정 실시예로, 408에서 클라이언트(30)가 캐시로부터 페이지렛의 일부 또는 전부를 일단 렌더링하면, 웹 페이지 서빙 프로세스(206)는 도 4b 및 4c와 관련하여 이하에서 후술되는 것처럼 페이지 생성 프로세스(210)가 페이지렛을 새로고침하도록 명령할 수 있다. 일부 실시예로, 웹 페이지 서빙 프로세스(206)는 페이지렛을 새로고침하기 위해 클라이언트(30)에 의한 요청에 응답하여 페이지렛이 새로고침되도록 할 수 있다.

[0041] 특정 실시예로, 410에서 캐시된 페이지렛이 쓸모가 없다고 결정되면, 웹 페이지 서빙 프로세스(206)는 임의의 캐시된 자원을 이용함이 없이 페이지렛을 새로운 페이지렛으로 생성하도록 페이지 생성 프로세스에게 명령할 수 있다. 웹 페이지 서빙 프로세스는 412에서 클라이언트(30)가 본 명세서에 기술되는 것처럼, 예컨대, 페이지렛을 클라이언트(30)에 대한 2차 응답의 일부로서 캐시에 전송함으로써, 캐시로부터가 아닌 410에서 생성된 페이지렛을 사용하여 웹 페이지를 렌더링하도록 명령할 수 있다.

[0042] 도 4b의 흐름도와 관련하여 페이지렛을 새로고침하기 위한 예시적인 방법이 이제 설명된다. 웹 페이지 서빙 프로세스(206)는 적어도 부분적으로 캐시로부터 렌더리된 페이지렛을 페이지 생성 프로세스(210)가 새로고침하도록 명령할 수 있다. 예컨대, 클라이언트(30)는 페이지가 더욱 특히, 페이지렛이 클라이언트(30)에 의해 마지막으로 렌더링된 이후에 변경된 대응하는 애플리케이션에 의해 요구되는 자원에 대한 충분적 업데이트를 요청할 수 있다. 특정 실시예로, 페이지렛 함수 호출은 요청 내의 대응하는 애플리케이션을 렌더링하는데 요구되는 자원에 대한 대응하는 타임스탬프를 포함한다. 특정 실시예로, 웹 페이지 서빙 프로세스(206) 및/또는 시스템 캐시 컴포넌트(218)는 대응하는 타임스탬프에 기초하여 자원이 클라이언트(30)로 마지막으로 전송되고 렌더링된 이후에 대응하는 자원에 대한 임의의 업데이트가 있었는지 여부를 결정할 수 있다.

[0043] 특정 실시예로, 420에서 웹 페이지 서빙 프로세스(206)는 페이지 생성 프로세스(210)가 예컨대, 그 자원이 타임스탬프가 요청과 함께 전송된 이후에 변경되었다면, 페이지렛에서 클라이언트(30)로 캐시된 자원 또는 자원에 대한 하나 이상의 충분적 업데이트를 생성하도록 명령할 수 있다. 즉, 웹 페이지가 전체로서 고려되고 웹 페이지에서의 임의의 변경이 클라이언트측 캐시로부터의 페이지의 플러싱(flushing)의 결과를 낳는 종래의 캐싱 절차에 대조적으로, 개별적 페이지렛은 전체 웹 페이지를 다시 요청함이 없이 그들의 대응하는 함수를 통해 충분적인 업데이트를 요청할 수 있다. 예컨대, 자원이 변화하거나 업데이트되었다면, 페이지 생성 프로세스(210)는 클라이언트(30)에서 캐시된 자원을 업데이트하는데 사용되는 충분적 데이터 (또는 아래 놓여지는 데이터에 대한 변화를 표시하는 차분 데이터(difference data))를 생성할 수 있다.

[0044] 특정 실시예로, 422에서 웹 페이지 서빙 프로세스(206)는 (만약 존재한다면) 페이지 생성 프로세스(210)로부터의 업데이트된 자원을 생성하자마자, 업데이트된 자원을 예컨대, 본 명세서에 기술되는 것처럼 2차 응답 부분으로서 클라이언트(30)로 전송할 수 있다. 424에서, 웹 페이지 서빙 프로세스(206)는 클라이언트(30)가 업데이트된 자원 및 캐시된 자원을 사용하여 페이지를 렌더링하도록 명령할 수 있다. 예컨대, 2차 응답 부분은 본 명세서에 기술되는 것처럼 페이지 조립 프로세스(208)에 의해 업데이트된 자원을 페이지로 삽입하는 명령어를 포함할 수 있다. 예컨대, 이것은 업데이트된 HTML 또는 다른 컨텐츠를 대응하는 페이지렛 또는 렌더링된 페이지의 다른 섹션으로 삽입하거나 페이지를 렌더링하는데 사용하기 위한 업데이트된 스크립트를 실행하는 것을 수반할 수 있다. 이러한 방식으로, 캐시된 재원은 적어도 부분적으로 업데이트된 자원의 생성과 병렬적으로 렌더링되고 사용자에게 디스플레이될 수 있다. 업데이트된 자원이 먼저 수신되면, 페이지 조립 프로세스는 업데이트된 자원을 포함하는 페이지렛을 렌더링할 수 있다. 추가적으로, 2차 응답 부분은 클라이언트(30)의 캐시에서 업데이트된 자원을 캐시하는 명령어를 포함할 수 있다.

[0045] 페이지렛을 새로고침하기 위한 추가로 예시적인 방법이 도 4c의 흐름도를 참조하여 이제 설명된다. 페이지렛의

캐시된 자원에 대한 하나 이상의 충분적 업데이트를 생성하기 보다는, 426에서 웹 페이지 서빙 프로세스(206)는 페이지 생성 프로세스(210)가 캐시된 자원을 새로운 버전으로 대체하기 위한 하나 이상의 새로운 대체 자원을 생성하도록 명령할 수 있다. 이것은 본 명세서에 기술되는 것처럼 타임스탬프가 요청과 함께 전송된 이후 및/또는 클라이언트(30)로부터의 요청에 있자 마자, 캐시된 자원이 변경되었다면, 웹 페이지를 렌더링하기 위해 캐시된 자원이 사용되는 때마다 발생할 수 있다.

[0046] 특정 실시예로, 428에서, 웹 페이지 서빙 프로세스(206)는 페이지 생성 프로세스(210)로부터 새로운 대체 자원이 (존재한다면) 생성되자 마자, 본 명세서에 기술되는 것처럼 예컨대, 2차 응답 부분의 일부로서 업데이트된 자원을 클라이언트(30)로 전송할 수 있다. 430에서, 웹 페이지 서빙 프로세스(206)는 클라이언트(30)가 새로운 대체 자원을 사용하여 페이지를 렌더링하도록 명령할 수 있다. 예컨대, 2차 응답 부분은 본 명세서에 기술되는 것처럼 페이지 조립 프로세스(208)에 의해 새로운 대체 자원을 페이지로 삽입하는 명령어를 포함할 수 있다. 예컨대, 이것은 대응하는 페이지렛의 HTML 또는 다른 컨텐츠 또는 렌더링된 페이지의 다른 섹션을 대체함으로써 대응하는 페이지렛을 대체하는 것 또는 페이지를 렌더링하는데 사용하기 위한 대체 스크립트를 실행하는 것을 수반할 수 있다. 이러한 방식으로, 캐시된 자원은 대체 자원의 생성 이전에 또는 적어도 부분적으로 병렬로 렌더링되고 사용자에게 디스플레이될 수 있다. 대체 자원이 일단 수신되면, 페이지 조립 프로세스는 대체 자원을 사용하여 페이지렛을 렌더링할 수 있다. 추가적으로 2차 반응 부분은 클라이언트(30)의 캐시에서 대응하는 페이지렛의 위치에 새로운 대체 자원을 캐시하는 명령어를 포함할 수 있다.

[0047] 페이지렛을 렌더링하는 방법이 도 5의 흐름도와 관련하여 이제 설명된다. 특정 실시예로, 502에서 페이지렛이 2차 응답에서 수신되면, 대응하는 함수 호출은 페이지 조립 프로세스(208)를 호출한다. 504에서, 페이지 조립 프로세스는 독자적으로 또는 장치 캐시 컴포넌트(220)와 통신하여, 수신된 페이지렛이 캐시 가능한 페이지렛인지 여부에 대한 체크를 수행할 수 있다. 예컨대, 함수 호출은 페이지렛이 캐시 가능한 자원을 포함한다는 것을 표시하기 위해 플래그와 같은 명령어를 포함할 수 있다. 예컨대, 캐시 가능한 자원은 네비게이션 메뉴, 뉴스피드 정보 또는 사용자가 웹 페이지를 떠난 이후 돌아올 때에 일반적으로 다시 로딩되는 다른 자원과 같은 자원을 포함할 수 있다. 일부 페이지렛은 캐시 가능한 자원을 포함하지 않을 수 있다. 예컨대, 일부 페이지렛에 대해서는, 사용자가 동일한 페이지로 돌아올 때마다 상이한 광고를 렌더링하는 것이 바람직한 광고를 디스플레이하는 페이지렛과 같이 페이지가 로딩될 때마다 새로운 자원을 요청하는 것이 바람직할 수 있다.

[0048] 특정 실시예로, 504에서 장치 캐시 컴포넌트(220)는 페이지렛이 캐시 가능하지 않다고 결정하고, 506에서 페이지 렌더링 컴포넌트(208)는 페이지렛을 캐싱하지 않고, 페이지렛을 렌더링하기 위한 임의의 캐시된 자원을 사용하지 않고 본 명세서에 기술되는 것처럼 페이지렛을 렌더링하도록 진행할 수 있다.

[0049] 특정 실시예로, 504에서 장치 캐시 컴포넌트가 페이지렛이 캐시 가능하다고 결정한다면, 508에서 장치 캐시 컴포넌트(220)는 이후 수신된 페이지렛이 이전에 캐시되었는지 여부를 결정하는 체크를 수행한다. 예컨대, 함수 호출은 페이지렛이 이전에 캐시되었다고 표시하는 플래그와 같은 명령어를 포함할 수 있다. 추가적으로 또는 대안적으로, 장치 캐시 컴포넌트(220)는 각각의 페이지렛에 대한 고유의 식별 정보로 각각의 캐시된 페이지렛을 등록하고, 페이지렛이 이전에 캐시되었는지 여부를 고유의 정보로부터 결정할 수 있다. 함수 호출 및/또는 장치 캐시 컴포넌트(220)는 또한 언제 페이지렛이 캐시되었는지에 관한 정보를 포함할 수 있다.

[0050] 특정 실시예로, 508에서 장치 캐시 컴포넌트(220)가 수신된 페이지렛이 이전에 캐시되지 않았다고 결정한다면, 장치 캐시 컴포넌트(220)는 클라이언트(30)의 캐시에 페이지렛을 저장할 수 있다. 예컨대, 캐시에 수신된 페이지렛을 저장하는 것은 대용량 저장 드라이브 상에서와 같이 클라이언트(30)의 영속적 메모리에 수신된 페이지렛을 저장하는 것을 포함할 수 있다. 수신된 페이지렛을 저장하는 것은 또한, 페이지렛을 암호화하고 암호화된 페이지렛을 캐시에 저장하는 것을 포함할 수 있다. 페이지렛을 암호화하는 것은 페이지렛의 복제, 간섭(tampering), 리버스 엔지니어링을 방지할 수 있고, 페이지렛 캐시 데이터는 웹 브라우징 세션이 종료한 후에 용속적 메모리에 잔존해야 한다. 페이지렛의 암호화는 예로서 제한 없이, 스텐포드 자바스크립트 크립토 라이브러리(Stanford Javascript Crypto Library)와 같은 암호화 라이브러리를 사용하여 페이지렛 데이터를 암호화하는 것을 포함할 수 있다. 추가적으로 또는 대안으로, 페이지렛을 저장하는 것은 클라이언트(30)의 시스템 메모리 또는 온-칩 프로세서 캐시 메모리와 같은 비영속적 메모리에 페이지렛을 저장하는 것을 포함할 수 있다. 506에서 페이지 렌더링 컴포넌트(208)는 이후 본 명세서에 기술되는 것처럼 페이지렛을 렌더링하는 것을 진행한다.

[0051] 특정 실시예로, 508에서 장치 캐시 컴포넌트(220)가 수신된 페이지렛이 이전에 캐시되었다고 결정한다면, 512에서 장치 캐시 컴포넌트(220)는 클라이언트(30)의 캐시로부터 캐시된 페이지렛 또는 캐시된 페이지렛의 일부에 대응하는 캐시된 자원을 검색할 수 있다. 캐시된 페이지렛이 암호화된다면, 캐시된 자원을 검색하는 것은 그들

이 페이지 생성 프로세스(208)에 의해 이용되기 전에 캐시된 자원을 해독하는 것을 포함할 수 있다. 캐시된 자원이 일단 수신되면, 506에서 페이지 렌더링 컴포넌트(208)는 본 명세서에 기술되는 것처럼 캐시된 자원을 사용하여 페이지렛을 렌더링하도록 진행할 수 있다.

[0052] 특정 실시예로, 페이지렛의 렌더링과 적어도 부분적으로 병렬로 또는 그 후에, 514에서 장치 캐시 컴포넌트(220)는 페이지렛의 스냅샷(snapshot)을 찍을 수 있다. 이러한 스냅샷은 각각의 수신된 페이지렛에 대한 DOM 정보 및/또는 함수 호출 데이터의 스냅샷을 찍는 것을 포함한다. 스냅샷은 예컨대 페이지렛에 대한 업데이트를 위한 요청의 일부로서 서버(22)로 전송될 수 있다. 캐시된 자원을 사용하여 렌더링된 페이지렛은 본 명세서에 기술되는 것처럼 페이지렛 자원이 캐시된 이후에 추가된 새로운 정보를 포함하기 위해, 새로고침될 수 있다. 이러한 방식으로, 캐시된 페이지렛은 캐시된 페이지렛의 렌더링과 적어도 부분적으로 병렬로 또는 그 후에 렌더링되고 사용자에게 디스플레이될 수 있으며, 페이지렛은 새로운 정보가 있다면 이를 포함하기 위해 새로고침될 수 있다. 그리하여, 사용자는 캐시로부터 검색되는 것이 아닌 서버로부터 모든 페이지렛 자원이 다운로드되는 경우 보다 페이지렛이 더욱 먼저 로딩되는 것을 지각할 수 있다.

[0053] 특정 실시예로, 506에서 본 명세서에 기술되는 것처럼 수신된 페이지렛 및/또는 캐시된 자원이 페이지 조립 프로세스(208)로 전송되고 나면, 페이지 조립 프로세스(208)는 이후 이전의 2차 응답에서 수신된 이전의 수신된 페이지렛에 대응하는 페이지렛 콜백 함수를 이미 포함할 수 있는 큐(214)에 대응하는 페이지렛 콜백 함수를 배치할 수 있다. 페이지렛을 렌더링하고 페이지렛을 웹 페이지에 삽입하는 것을 포함하는 페이지렛 콜백 함수를 다루는 것에 관한 세부사항은 예컨대, 본 명세서에 전체로서 참조로 통합되는 미국 특허출원 번호 제12/754,549호에서 발견될 수 있다.

[0054] 특정 실시예로, 장치(30)의 캐시는 특정 이벤트 후 또는 주기적으로 삭제될 수 있다. 예컨대, 장치 캐시 컴포넌트(220)는 캐시를 삭제하라는 사용자에 의한 명령이 있자마자 캐시를 삭제하도록 구성될 수 있다. 추가적으로 또는 대안으로, 장치 캐시 컴포넌트는 사용자가 웹 사이트로부터 로그오프하거나 사용자가 웹 사이트에 로그인한 이후 일정 시구간이 경과하거나 캐시가 생성된 때와 같은 특정 이벤트 후에 캐시를 삭제하도록 구성될 수 있다.

[0055] 본 명세서에서 기술되는 것처럼, 임의의 기술된 프로세스 또는 방법은 전술한 동작을 하나 이상의 프로세서가 구현하도록 실행시 동작가능한 실체적 데이터 저장 매체 내에 구체화 또는 인코딩된 일련의 컴퓨터판독가능한 명령어로서 구현될 수 있다. 더 작은 데이터세트를 위해, 전술한 동작은 단일의 컴퓨팅 플랫폼 또는 노드 상에서 실행될 수 있다. 예컨대, 특정 실시예로, 도 2 및 3을 참조하여 전술한 단계화 생성 프로세스는 서버(22)에서 실행되는 단일 서버 프로세스에 의해 구현될 수 있다. 즉, 전술한 웹 페이지 생성 및 서빙 프로세스는 서버(22) 상에서 구현될 수 있다. 더 큰 시스템 및 결과적 데이터세트에 대해서는, 병렬 컴퓨팅 플랫폼이 사용될 수 있다.

[0056] 도 1은 마스터 서버(22a) 및 2 개의 슬레이브 서버(22b)로 구성된 예시적인 분산형 컴퓨팅 시스템을 도시한다. 일부 실시예에서, 분산형 컴퓨팅 시스템은 슬레이브 서버가 통상적으로 노드라고 불리는 상업용 서버의 고가용성 클러스터를 포함한다. 도 1에서 비록 2 개의 노드만이 도시되지만, 노드의 개수는 백 또는 십지어는 일부 실시예에서 천 이상을 초과할 수 있다. 일상적으로 고가용성 클러스터의 노드는 중복적이어서 특정 애플리케이션을 수행하는 동안 하나의 노드가 충돌한다면 클러스터 소프트웨어는 하나 이상의 다른 노드에서 애플리케이션을 재시작할 수 있다.

[0057] 다수의 노드는 또한 큰 데이터베이스의 병렬 프로세싱을 용이하게 한다. 일부 실시예에서, 22a와 같은 마스터 서버는 클라이언트로부터 작업을 수신하고 마스터로부터의 명령이 있자마자 할당된 과제를 실행하고, 이후 과제 사이에서 데이터를 이동시키는 실제 작업을 하는 서버(22b)와 같은 슬레이브 서버 또는 노드에게 그 작업으로부터 발생되는 과제를 할당한다. 일부 실시예에서, 클라이언트 작업은 전술한 바와 같은 하둡의 맵리듀스 기능(Hadoop's MapReduce functionality)을 불러온다(invokes).

[0058] 유사하게, 일부 실시예에서, 22a와 같은 마스터 서버는 대형 데이터베이스의 병렬 프로세싱을 지원하는 분산형 파일 시스템을 지배한다. 특히, 마스터 서버(22a)는 서버(22b)와 같은 슬레이브 서버 또는 노드 상에 실제로 저장된 파일에 대한 클라이언트의 접근뿐만 아니라 파일 시스템의 네임스페이스 및 노드에 대한 블록 맵핑을 관리한다. 교대로, 일부 실시예에서, 슬레이브 서버는 클라이언트로부터의 읽기 및 쓰기 요청을 실행하고 마스터 서버로부터의 명령이 있자마자 블록의 생성, 삭제 및 복사를 수행하는 실제 작업을 한다.

[0059] 전술한 프로세스 및 기구가 광범위한 물리적 시스템에 의해 광범위한 네트워크 및 컴퓨팅 환경에서 구현될 수

있는 한편, 후술될 서버 또는 컴퓨팅 시스템은 제한하는 것이 아닌 훈시적인 목적으로 예시적인 컴퓨팅 시스템 구조(computing system architecture)를 제공한다.

[0060] 도 6은 서버(22a, 22b) 또는 클라이언트 장치(30)를 구현하는데 사용될 수 있는 예시적인 컴퓨팅 시스템 구조를 도시한다. 일실시예로, 하드웨어 시스템(600)은 프로세서(602), 캐시 메모리(604) 및 유형의 컴퓨터 판독가능한 매체에 저장되고 본 명세서에 기술된 기능에 관한 하나 이상의 실행가능한 모듈과 드라이버를 포함한다. 추가로, 하드웨어 시스템(600)은 고성능 입력/출력(I/O) 버스(606) 및 표준 I/O 버스(608)를 포함한다. 호스트 브릿지(610)는 프로세서(602)와 고성능 I/O 버스(606)를 연결하는 한편, I/O 버스 브릿지(612)는 2개의 버스(606, 608)를 서로 연결한다. 시스템 메모리(614) 및 하나 이상의 네트워크/통신 인터페이스(616)는 버스(606)에 연결된다. 하드웨어 시스템(600)은 비디오 메모리(미도시) 및 비디오 메모리에 연결된 디스플레이 장치를 더 포함할 수 있다. 대용량 저장소(618) 및 I/O 포트(620)가 버스(608)에 연결된다. 하드웨어 시스템(600)은 키보드와 포인팅 장치 및 버스(608)에 연결된 디스플레이 장치(미도시)를 선택적으로 포함할 수 있다. 일괄적으로, 이를 구성요소들은 캘리포니아, 산타 클라라의 인텔(Intel)사에 의해 제조된 x86-호환가능 프로세서와 캘리포니아, 서니베일의 AMD(Advanced Micro Devices)사에 의해 제조된 x86-호환가능 프로세서 및 임의의 다른 적합한 프로세서에 기반한 범용 컴퓨터 시스템을 포함하나 이에 제한되지 않는, 컴퓨터 하드웨어 시스템의 광범위한 카테고리를 나타내는 것으로 의도된다.

[0061] 하드웨어 시스템(600)의 구성요소는 하기와 더 상세히 기술된다. 특히, 네트워크 인터페이스(616)는 하드웨어 시스템(600)과 가령 이더넷(Ethernet)(예컨대, IEEE 802.3) 네트워크, 백플레인(backplane) 등과 같은 임의의 광범위한 네트워크 사이의 통신을 제공한다. 대용량 저장소(618)는 서버(22a, 22b)에서 구현되는 상술한 기능을 수행하기 위한 데이터 및 프로그래밍 명령어의 영구 저장을 제공하는 반면, 시스템 메모리(614)(예컨대, DRAM)는 프로세서(602)에 의해 실행될 때의 데이터 및 프로그래밍 명령어의 일시 저장을 제공한다. I/O 포트(620)는 하드웨어 시스템(600)에 연결될 수 있는 추가적인 주변 장치들 사이의 통신을 제공하는 하나 이상의 직렬 및/또는 병렬 통신 포트이다.

[0062] 하드웨어 시스템(600)은 다양한 시스템 구조(architecture)를 포함할 수 있으며, 하드웨어 시스템(600)의 다양한 구성요소는 재배열될 수 있다. 예컨대, 클라이언트(30)의 캐시(604)는 프로세서(602)에 내장(on-chip)되어 구현될 수 있다. 대안으로, 캐시(604)는 클라이언트(30)의 시스템 메모리(614)에서 구현될 수 있다. 이러한 방식으로 프로세서 칩 또는 시스템 메모리(614)에서 구현되는 캐시(604)는 일반적으로 일시적 저장소에서 구현될 수 있다. 추가적인 대안으로, 캐시(604)는 영속적 저장 장치일 수 있는 클라이언트(30)의 대용량 저장소(618)에서 구현될 수 있다.

[0063] 게다가, 개시된 발명의 대상의 특정 실시예는 상기의 컴포넌트들의 전부를 포함하지 않거나 요구하지 않을 수 있다. 예컨대, 표준 I/O 버스(508)에 연결되는 것으로 도시된 주변 장치는 고성능 I/O 버스(606)에 연결될 수 있다. 추가적으로, 일부 실시예로, 오로지 단일의 버스가 단일의 버스에 연결된 하드웨어 시스템(600)의 컴포넌트와 함께 존재할 수 있다. 더욱이, 하드웨어 시스템(600)은 추가적인 프로세서, 저장 장치 또는 메모리와 같은 추가적인 컴포넌트를 포함할 수 있다.

[0064] 하나의 구현으로, 본 명세서에 기술된 실시예의 동작은 분산형 컴퓨팅 환경에서 개별적으로 또는 일괄적으로 하드웨어 시스템(600)에 의해 운영되는 일련의 실행가능한 모듈로 구현된다. 특정 실시예로, 한 세트의 소프트웨어 모듈 및/또는 드라이버는 네트워크 통신 프로토콜 스택, 병렬 컴퓨팅 기능, 브라우징과 다른 컴퓨팅 기능, 최적화 프로세스 등을 구현한다. 상술한 기능형 모듈은 하드웨어, 컴퓨터 판독가능한 매체에 저장된 실행가능한 모듈 또는 이 둘의 조합에 의해 실현될 수 있다. 예컨대, 기능형 모듈은 가령 프로세서(602)와 같은 하드웨어 시스템의 프로세서에 의해 실행되는 복수의 또는 일련의 명령어를 포함할 수 있다. 초기에, 일련의 명령어는 가령 대용량 저장소(618)와 같은 저장장치에 저장될 수 있다. 그러나, 일련의 명령어는 가령 디스크, CD-ROM, ROM, EEPROM 등과 같은 임의의 적절한 저장 매체에 실체적으로 저장될 수 있다. 게다가, 일련의 명령어는 국부적으로 저장될 필요는 없으며, 네트워크/통신 인터페이스(616)를 통해 가령 네트워크상의 서버와 같은 원격 저장장치로부터 수신될 수 있다. 명령어는 가령 대용량 저장소(618)와 같은 저장장치로부터 메모리(614)로 복사된 후 프로세서(602)에 의해 접근되고 실행된다.

[0065] 운영 시스템은 소프트웨어 애플리케이션(미도시)으로의 및 소프트웨어 애플리케이션으로부터의 데이터의 입력 및 출력을 포함하는, 하드웨어 시스템(600)의 동작을 관리 및 제어한다. 운영 시스템은 시스템상에서 실행되는 소프트웨어 애플리케이션과 시스템의 하드웨어 컴포넌트 사이의 인터페이스를 제공한다. 가령, 리눅스(LINUX) 운영 시스템, 캘리포니아, 쿠퍼티노의 애플 컴퓨터(Apple Computer)사로부터 이용가능한 애플 매킨토시(Apple

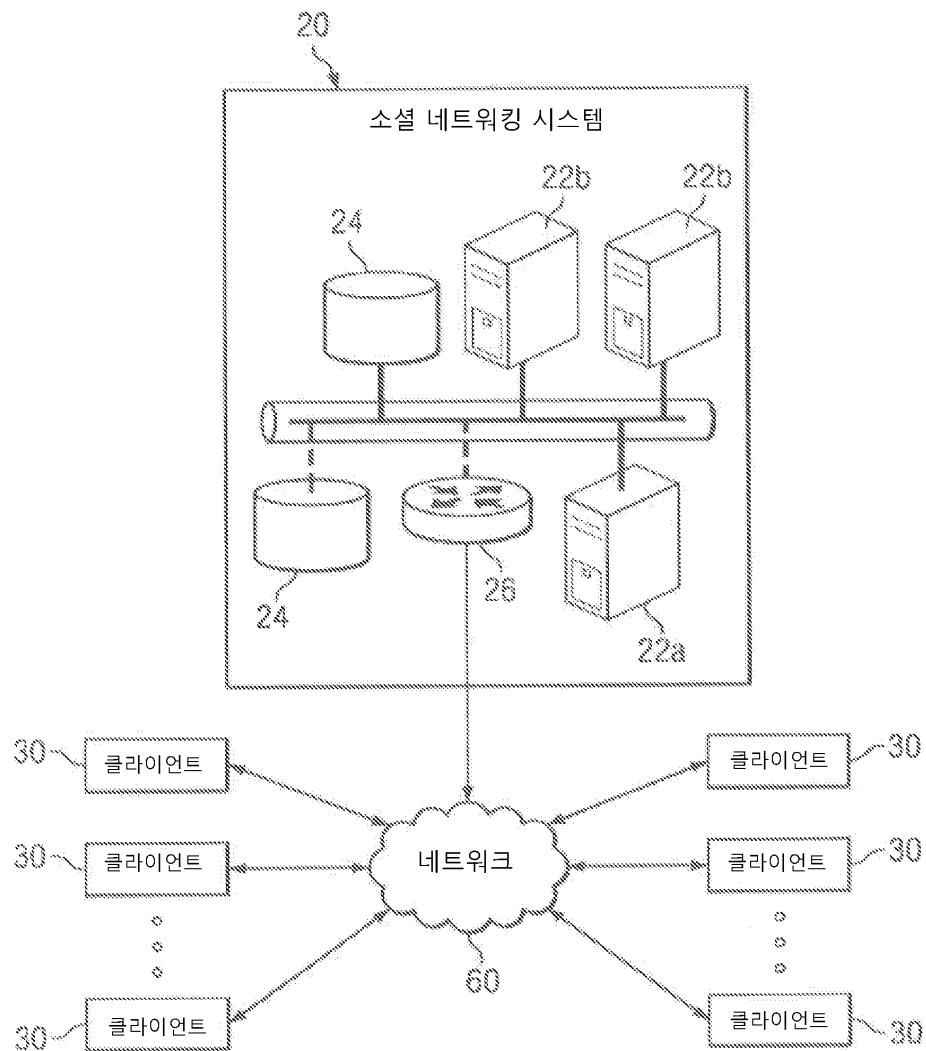
Macintosh) 운영 시스템, 유닉스(UNIX) 운영 시스템, 마이크로소프트(r) 윈도우(r) 운영 시스템, BSD 운영 시스템 등과 같은 임의의 적합한 운영 시스템이 사용될 수 있다. 물론, 다른 구현도 가능하다. 예컨대, 본 명세서에 기술된 기능은 펌웨어(firmware)에서 또는 애플리케이션 주문형 집적회로(application specific integrated circuit) 상에서 구현될 수 있다.

[0066] 게다가, 상술한 구성요소 및 동작은 저장 매체에 저장된 명령어들로 구성될 수 있다. 명령어는 프로세싱 시스템에 의해 검색되고 실행될 수 있다. 명령어의 일부 예는 소프트웨어, 프로그램 코드 및 펌웨어이다. 저장 매체의 일부 예는 메모리 장치, 테이프(tape), 디스크, 집적회로 및 서버이다. 명령어는 프로세싱 시스템이 본 발명과 부합하여 동작하도록 프로세싱 시스템에 의해 실행되는 경우 동작한다. "프로세싱 시스템"이란 용어는 단일 프로세싱 장치 또는 상호-동작형 프로세싱 장치의 그룹을 뜻한다. 프로세싱 장치의 일부 예는 집적회로들 및 논리회로부(logic circuitry)이다. 통상의 기술자에게 명령어, 컴퓨터 및 저장 매체는 자명하다.

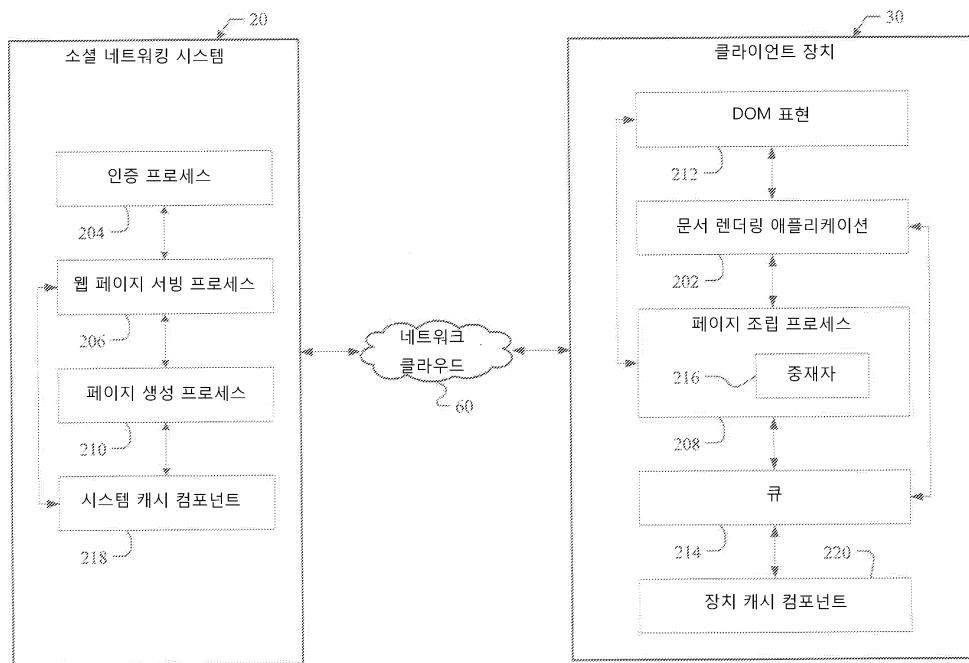
[0067] 본 명세서는 당업자가 이해할 수 있는 본 명세서의 예시적인 실시예에 대한 모든 변화, 치환, 변형, 대체 및 변경을 포함한다. 마찬가지로, 적절한 경우에, 첨부된 청구항들은 당업자가 이해할 수 있는 본 명세서의 예시적인 실시예에 대한 모든 변화, 치환, 변형, 대체 및 변경을 포함한다. 예로서, 본 발명의 실시예들이 소셜 네트워킹 웹사이트와 연결되어 동작하는 것으로 기술되었으나, 본 발명은 웹 애플리케이션을 지원하는 임의의 통신 시설과 연결되어 사용될 수 있다. 게다가, 일부 실시예로, "웹 서비스" 및 "웹 사이트"란 용어는 교환하여 사용될 수 있으며, 서버로 직접 API를 호출하는, 가령 모바일 장치(예컨대, 셀룰러폰, 스마트폰, 개인용 GPS, 개인용 정보단말기(personal digital assistance), 개인용 게임장치 등)와 같은 장치상의 맞춤형(custom) API 또는 일반형 API를 추가적으로 뜻할 수 있다.

도면

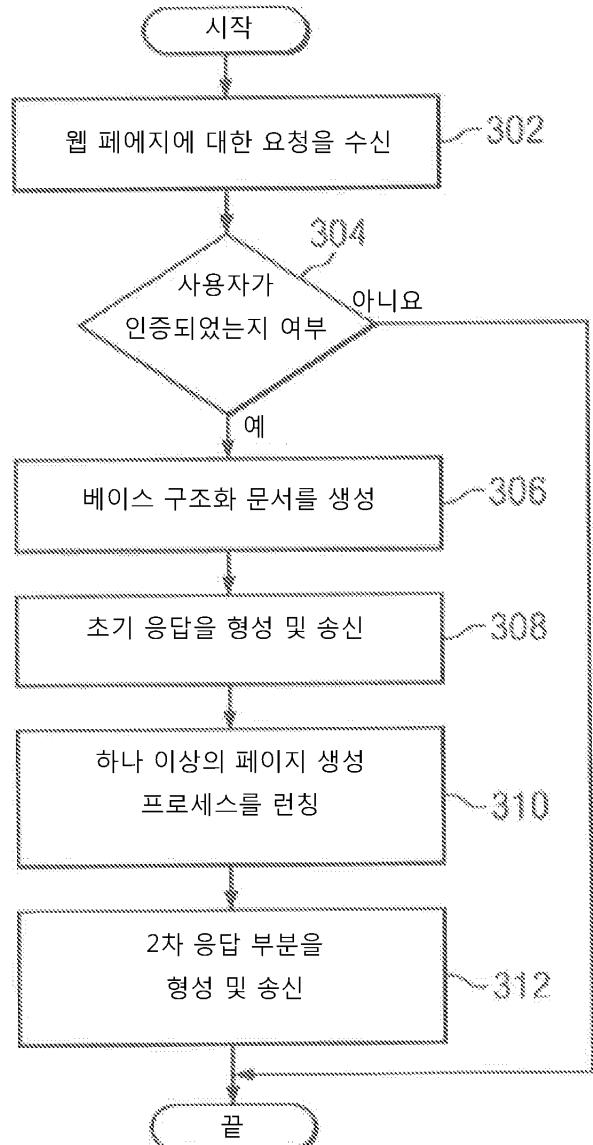
도면1



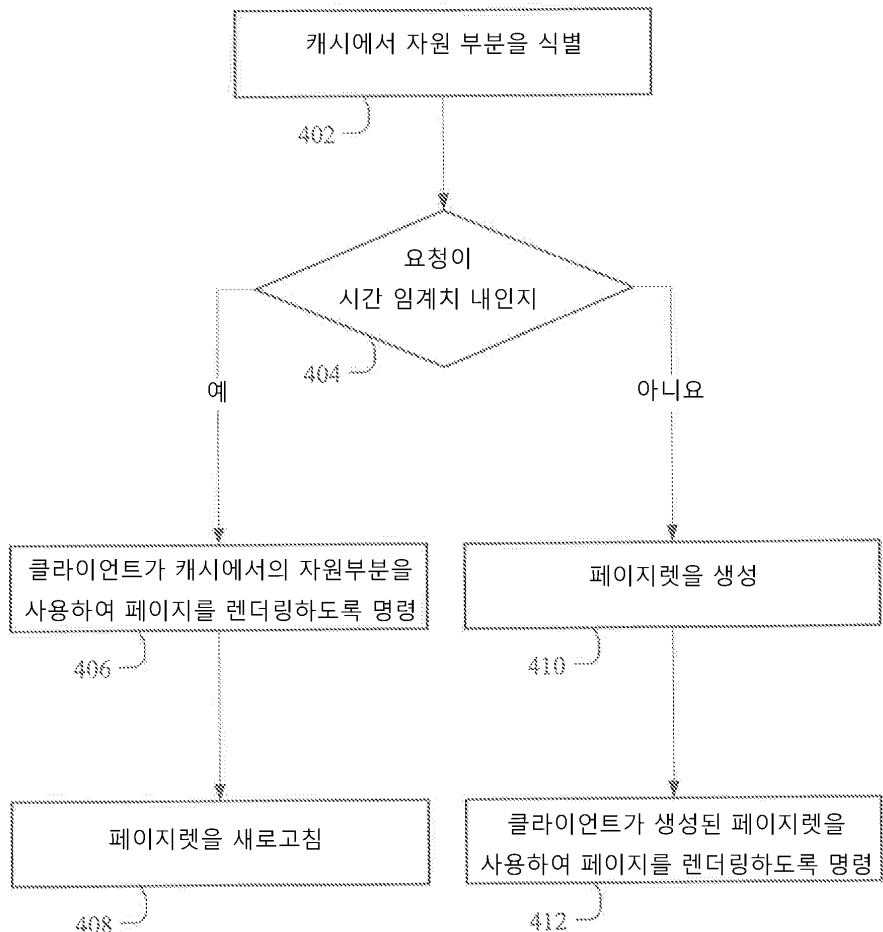
도면2



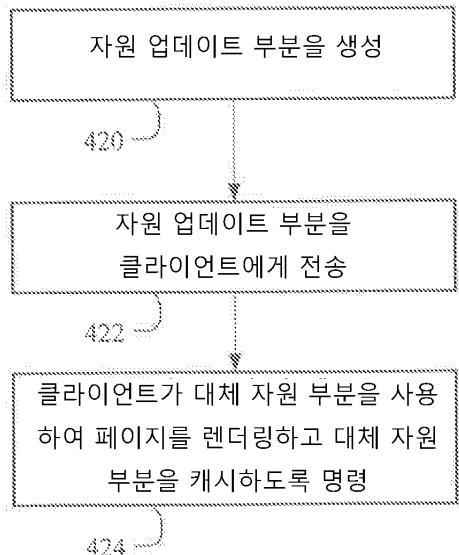
도면3



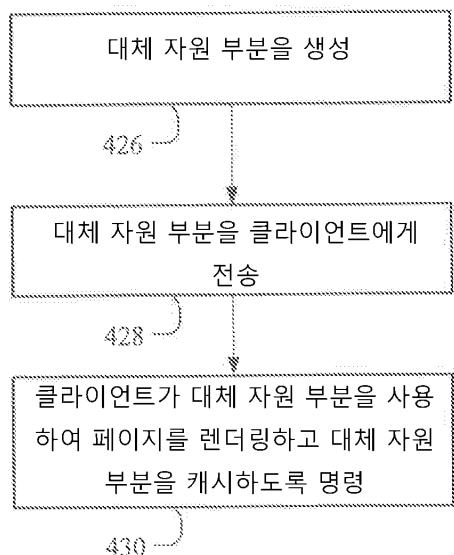
도면4a



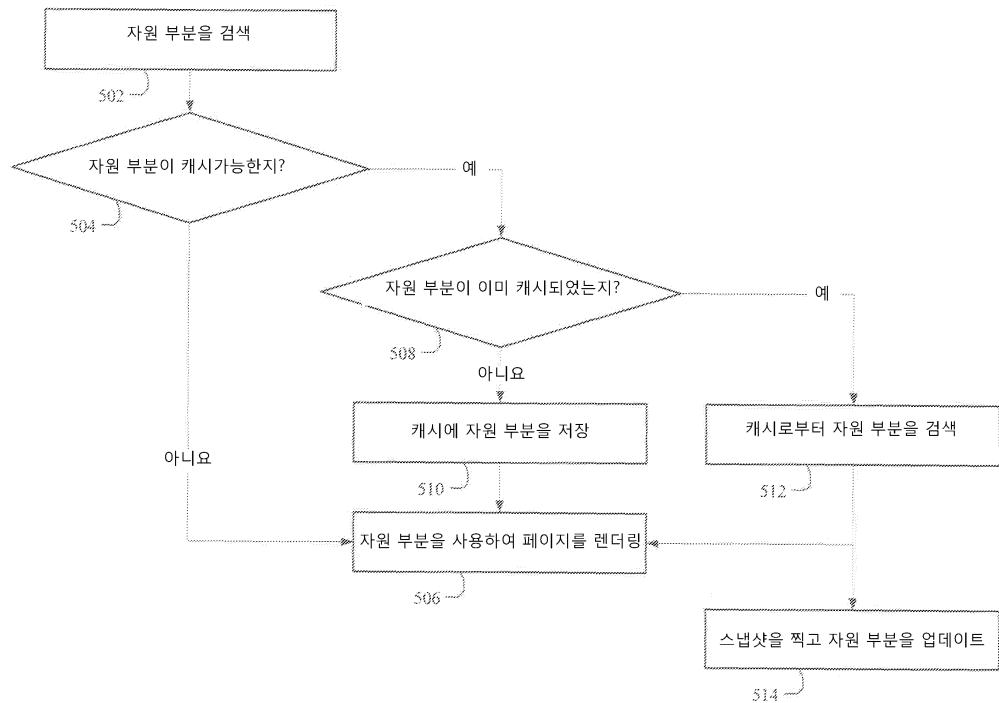
도면4b



도면4c



도면5



도면6

