



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년03월14일
(11) 등록번호 10-1824222
(24) 등록일자 2018년01월25일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G06F 17/21 (2006.01)
G06F 17/22 (2006.01) G06F 17/27 (2006.01)
(52) CPC특허분류
G06F 17/30905 (2013.01)
G06F 17/218 (2013.01)
(21) 출원번호 10-2017-7001770
(22) 출원일자(국제) 2015년07월25일
심사청구일자 2017년10월18일
(85) 번역문제출일자 2017년01월19일
(65) 공개번호 10-2017-0038793
(43) 공개일자 2017년04월07일
(86) 국제출원번호 PCT/US2015/042146
(87) 국제공개번호 WO 2016/015031
국제공개일자 2016년01월28일
(30) 우선권주장
62/029,398 2014년07월25일 미국(US)
14/809,046 2015년07월24일 미국(US)
(56) 선행기술조사문헌
KR1020110131673 A

(73) 특허권자
웹컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
리우, 보진
미국 92121 캘리포니아 샌 디에고 모어하우스 드라이브 5775
리브니코브, 다니스
미국 92121 캘리포니아 샌 디에고 모어하우스 드라이브 5775
(74) 대리인
(뒷면에 계속)
특허법인 남앤드남

전체 청구항 수 : 총 20 항

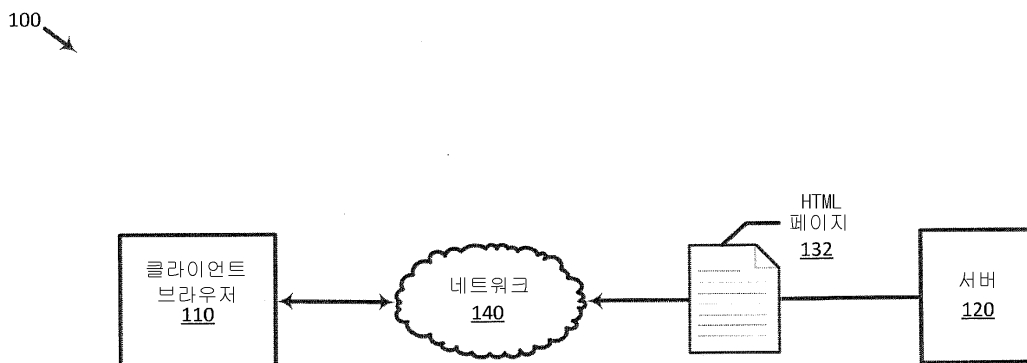
심사관 : 이복현

(54) 발명의 명칭 동적 콘텐츠 및 스타일 콘텐츠를 포함하는 웹사이트들의 고속 렌더링

(57) 요약

웹페이지를 렌더링하는 예시적인 방법은, 위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여, 제 1 직렬화된 최종 문서 오브젝트 모듈(DOM)을 생성하는 단계를 포함한다. 제 1 직렬화된 최종 DOM은, 제 1 인-라인된 동적 콘텐츠를 포함한다. 방법은 또한, 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여, 제 2 직렬화된 최종 DOM을 생성하는 단계를 포함한다. 제 2 직렬화된 최종 DOM은, 제 2 인-라인된 동적 콘텐츠를 포함한다. 방법은 추가로, 제 1 및 제 2 직렬화된 최종 DOM들 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성하는 단계를 포함한다. 공통 직렬화된 최종 DOM은 제 1 및 제 2 직렬화된 최종 DOM들에 포함되는 정적인 동적 콘텐츠를 포함하고, 차이는 트루 동적 콘텐츠이다. 방법은 또한, 공통 직렬화된 최종 DOM을 디바이스의 디스플레이에 렌더링하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

G06F 17/2211 (2013.01)

G06F 17/2247 (2013.01)

G06F 17/227 (2013.01)

G06F 17/272 (2013.01)

(72) 발명자

페렐만, 바레리아

미국 92121 캘리포니아 샌 디에고 모어하우스 드라
이브 5775

자크라제크, 토마스

미국 92121 캘리포니아 샌 디에고 모어하우스 드라
이브 5775

명세서

청구범위

청구항 1

웹페이지를 렌더링(render)하는 방법으로서,

위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여, 제 1 직렬화된(serialized) 최종(eventual) 문서 오브젝트 모델(DOM; document object model)을 생성하는 단계 - 상기 제 1 웹 문서는 정적인 동적 콘텐츠(static dynamic content) 및 제 1 트루 동적 콘텐츠(true dynamic content)를 포함하는 제 1 클라이언트-측 생성 동적 콘텐츠(client-side generated dynamic content)와 연관되고, 그리고 상기 제 1 클라이언트-측 생성 동적 콘텐츠는 상기 제 1 직렬화된 최종 DOM에 인-라인(in-line)됨 -;

상기 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여, 제 2 직렬화된 최종 DOM을 생성하는 단계 - 상기 제 2 웹 문서는 상기 정적인 동적 콘텐츠 및 제 2 트루 동적 콘텐츠를 포함하는 제 2 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 2 클라이언트-측 생성 동적 콘텐츠는 상기 제 2 직렬화된 최종 DOM에 인-라인됨 -;

상기 제 1 직렬화된 최종 DOM에 인-라인된 상기 제 1 트루 동적 콘텐츠와 상기 제 2 직렬화된 최종 DOM에 인-라인된 상기 제 2 트루 동적 콘텐츠 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성하는 단계 - 상기 공통 직렬화된 최종 DOM은 상기 정적인 동적 콘텐츠를 포함하고 그리고 상기 제 1 트루 동적 콘텐츠를 결여(absent)함 -; 및

디바이스의 디스플레이에 상기 공통 직렬화된 최종 DOM을 렌더링하는 단계를 포함하는, 웹페이지를 렌더링하는 방법.

청구항 2

제 1 항에 있어서,

상기 공통 직렬화된 최종 DOM 생성하는 단계는, 상기 제 1 직렬화된 최종 DOM과 상기 제 2 직렬화된 최종 DOM 간의 하나 또는 그 초과와 차이들을 식별하는 단계, 및 상기 제 1 직렬화된 최종 DOM으로부터 상기 하나 또는 그 초과와 차이들을 제거하는 단계를 포함하는, 웹페이지를 렌더링하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 웹 문서를 파싱(parse)하는 단계 - 상기 제 1 웹 문서는 마크업(markup) 언어로 되어 있고, 그리고 엘리먼트들의 세트 및 규칙들의 세트를 포함함 -;

상기 엘리먼트들의 세트의 하나 또는 그 초과와 엘리먼트들에 상기 규칙들의 세트를 적용하는 단계;

상기 파싱하는 단계 및 상기 적용하는 단계에 따라 최종 DOM을 생성하는 단계 - 상기 최종 DOM은, 상기 규칙들의 세트를 적용하는 단계에 기초하여 생성되는 상기 제 1 클라이언트-측 생성 동적 콘텐츠를 포함함 -; 및

상기 제 1 직렬화된 최종 DOM을 생성하기 위해 상기 최종 DOM을 직렬화하는 단계를 더 포함하는, 웹페이지를 렌더링하는 방법.

청구항 4

제 3 항에 있어서,

상기 제 1 직렬화된 최종 DOM을 생성하는 단계는, 상기 제 1 직렬화된 최종 DOM 내의 하나 또는 그 초과와 인-라인(non-in-line) 규칙들을 인-라인 규칙들로 이동시키는 단계를 포함하는, 웹페이지를 렌더링하는 방법.

청구항 5

제 4 항에 있어서,

규칙이 클라이언트-측 스크립트(script)를 포함하는, 웹페이지를 렌더링하는 방법.

청구항 6

제 4 항에 있어서,

규칙은, 상기 제 1 웹 문서 내의 하나 또는 그 초과와 엘리먼트들에 적용되는 포매팅(formatting) 규칙인, 웹페이지를 렌더링하는 방법.

청구항 7

제 3 항에 있어서,

상기 제 1 직렬화된 최종 DOM 및 상기 제 2 직렬화된 최종 DOM은, 상기 최종 DOM의 상이한 버전들인, 웹페이지를 렌더링하는 방법.

청구항 8

제 1 항에 있어서,

상기 제 1 직렬화된 최종 DOM을 파일로 직렬화하는 단계; 및

상기 파일을 캐싱(chche)하는 단계를 더 포함하는, 웹페이지를 렌더링하는 방법.

청구항 9

제 1 항에 있어서,

상기 공통 직렬화된 최종 DOM을 렌더링하는 단계는, 상기 공통 직렬화된 최종 DOM을 포어그라운드(foreground)에서 렌더링하는 단계를 포함하며,

상기 방법은,

캐시로부터 상기 공통 직렬화된 최종 DOM을 검색(retrieve)하는 단계;

상기 위치에 의해 참조되는 제 3 웹 문서에 대한 현재 요청을 수신하는 단계;

백그라운드(background) 프로세스에서, 상기 현재 요청에 대한 현재 DOM을 구성하는 단계 및 상기 현재 DOM을 렌더링하는 단계;

상기 현재 DOM이 렌더링된 이후, 렌더링된 현재 DOM을 렌더링된 공통 직렬화된 최종 DOM으로 대체하는 단계

를 더 포함하는, 웹페이지를 렌더링하는 방법.

청구항 10

제 1 항에 있어서,

상기 위치에 의해 참조되는 제 3 웹 문서를 프로세싱하는 것에 기초하여, 제 3 직렬화된 최종 DOM을 생성하는 단계 — 상기 제 3 직렬화된 최종 DOM은 인-라인된 동적 콘텐츠를 포함함 —;

상기 제 2 직렬화된 최종 DOM과 상기 제 3 직렬화된 최종 DOM 간의 제 2 차이에 기초하여, 제 2 공통 직렬화된 최종 DOM을 생성하는 단계 — 상기 제 2 공통 직렬화된 최종 DOM은, 상기 제 2 직렬화된 최종 DOM 및 상기 제 3 직렬화된 최종 DOM에 포함된 상기 정적인 동적 콘텐츠를 포함하고, 그리고 상기 제 2 트루 동적 콘텐츠를 결여함 —; 및

상기 디바이스의 상기 디스플레이에 상기 제 2 공통 직렬화된 최종 DOM을 렌더링하는 단계를 더 포함하는, 웹페이지를 렌더링하는 방법.

청구항 11

제 1 항에 있어서,

상기 위치는 URL(uniform resource locator)인, 웹페이지를 렌더링하는 방법.

청구항 12

웹페이지를 렌더링하기 위한 시스템으로서,

위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여 제 1 직렬화된 최종 문서 오브젝트 모델(DOM)을 생성하고 그리고 상기 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여 제 2 직렬화된 최종 DOM을 생성하는 클라이언트 브라우저를 포함하고,

상기 제 1 웹 문서는 정적인 동적 콘텐츠 및 제 1 트루 동적 콘텐츠를 포함하는 제 1 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 1 클라이언트-측 생성 동적 콘텐츠는 상기 제 1 직렬화된 최종 DOM에 인-라인되고, 그리고 상기 제 2 웹 문서는 상기 정적인 동적 콘텐츠 및 제 2 트루 동적 콘텐츠를 포함하는 제 2 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 2 클라이언트-측 생성 동적 콘텐츠는 상기 제 2 직렬화된 최종 DOM에 인-라인되고,

상기 클라이언트 브라우저는, 상기 제 1 직렬화된 최종 DOM에 인-라인된 상기 제 1 트루 동적 콘텐츠와 상기 제 2 직렬화된 최종 DOM에 인-라인된 상기 제 2 트루 동적 콘텐츠 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성하고, 그리고 상기 공통 직렬화된 최종 DOM은 상기 정적인 동적 콘텐츠를 포함하고 그리고 상기 제 1 트루 동적 콘텐츠를 결여하고,

상기 클라이언트 브라우저는, 상기 공통 직렬화된 최종 DOM을 디바이스의 디스플레이에 렌더링하는, 웹페이지를 렌더링하기 위한 시스템.

청구항 13

제 12 항에 있어서,

상기 클라이언트 브라우저는, 상기 제 1 직렬화된 최종 DOM과 상기 제 2 직렬화된 최종 DOM 간의 하나 또는 그 초과된 차이들을 식별하고, 상기 제 1 직렬화된 최종 DOM으로부터 상기 하나 또는 그 초과된 차이들을 제거하는, 웹페이지를 렌더링하기 위한 시스템.

청구항 14

제 12 항에 있어서,

상기 클라이언트 브라우저는 상기 제 1 웹 문서를 파싱하고, 그리고 상기 제 1 웹 문서는 마크업 언어로 되어 있고 그리고 엘리먼트들의 세트 및 규칙들의 세트를 포함하고,

상기 클라이언트 브라우저는, 상기 엘리먼트들의 세트의 하나 또는 그 초과된 엘리먼트들에 상기 규칙들의 세트를 적용하고, 그리고 상기 파싱하는 것 및 상기 적용하는 것에 따라 최종 DOM을 생성하고, 그리고

상기 최종 DOM은, 상기 규칙들의 세트를 적용하는 것에 기초하여 생성된 제 1 동적 콘텐츠를 포함하는, 웹페이지를 렌더링하기 위한 시스템.

청구항 15

제 14 항에 있어서,

상기 클라이언트 브라우저는, 제 1 최종 DOM으로부터 하나 또는 그 초과된 규칙들을 제거하고, 그리고 하나 또는 그 초과된 제거된 규칙들을 상기 제 1 최종 DOM에 인-라인으로 포함시키는, 웹페이지를 렌더링하기 위한 시스템.

청구항 16

제 15 항에 있어서,

규칙이 클라이언트-측 스크립트를 포함하는, 웹페이지를 렌더링하기 위한 시스템.

청구항 17

제 16 항에 있어서,

상기 클라이언트-측 스크립트는 JAVASCRIPT 프로그래밍 언어로 되어 있는, 웹페이지를 렌더링하기 위한 시스템.

청구항 18

제 14 항에 있어서,

상기 제 1 직렬화된 최종 DOM 및 상기 제 2 직렬화된 최종 DOM은, 최종 DOM의 상이한 버전들인, 웹페이지를 렌더링하기 위한 시스템.

청구항 19

동작들을 수행하기 위한 컴퓨터-실행가능 명령들이 저장된 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 동작들은,

위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여, 제 1 직렬화된 최종 문서 오브젝트 모델(DOM)을 생성하는 것 - 상기 제 1 웹 문서는 정적인 동적 콘텐츠 및 제 1 트루 동적 콘텐츠를 포함하는 제 1 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 1 클라이언트-측 생성 동적 콘텐츠는 상기 제 1 직렬화된 최종 DOM에 인-라인됨 -;

상기 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여, 제 2 직렬화된 최종 DOM을 생성하는 것 - 상기 제 2 웹 문서는 상기 정적인 동적 콘텐츠 및 제 2 트루 동적 콘텐츠를 포함하는 제 2 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 2 클라이언트-측 생성 동적 콘텐츠는 상기 제 2 직렬화된 최종 DOM에 인-라인됨 -;

상기 제 1 직렬화된 최종 DOM에 인-라인된 상기 제 1 트루 동적 콘텐츠와 상기 제 2 직렬화된 최종 DOM에 인-라인된 상기 제 2 트루 동적 콘텐츠 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성하는 것 - 상기 공통 직렬화된 최종 DOM은 상기 정적인 동적 콘텐츠를 포함하고 그리고 상기 제 1 트루 동적 콘텐츠를 결합 함 -; 및

디바이스의 디스플레이에 상기 공통 직렬화된 최종 DOM을 렌더링하는 것을 포함하는, 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 20

웹페이지를 렌더링하기 위한 시스템으로서,

위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여, 제 1 직렬화된 최종 문서 오브젝트 모델 (DOM)을 생성하기 위한 수단 - 상기 제 1 웹 문서는 정적인 동적 콘텐츠 및 제 1 트루 동적 콘텐츠를 포함하는 제 1 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 1 클라이언트-측 생성 동적 콘텐츠는 상기 제 1 직렬화된 최종 DOM에 인-라인됨 -;

상기 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여, 제 2 직렬화된 최종 DOM을 생성하기 위한 수단 - 상기 제 2 웹 문서는 상기 정적인 동적 콘텐츠 및 제 2 트루 동적 콘텐츠를 포함하는 제 2 클라이언트-측 생성 동적 콘텐츠와 연관되고, 그리고 상기 제 2 클라이언트-측 생성 동적 콘텐츠는 상기 제 2 직렬화된 최종 DOM에 인-라인됨 -;

상기 제 1 직렬화된 최종 DOM에 인-라인된 상기 제 1 트루 동적 콘텐츠와 상기 제 2 직렬화된 최종 DOM에 인-라인된 상기 제 2 트루 동적 콘텐츠 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성하기 위한 수단 - 상기 공통 직렬화된 최종 DOM은 상기 정적인 동적 콘텐츠를 포함하고 그리고 상기 제 1 트루 동적 콘텐츠를 결합 함 -; 및

디바이스의 디스플레이에 상기 공통 직렬화된 최종 DOM을 렌더링하기 위한 수단을 포함하는, 웹페이지를 렌더링하기 위한 시스템.

발명의 설명

기술분야

- [0001] [0001] 본 개시내용은 일반적으로, 콘텐츠(content)를 렌더링(render)하는 것에 관한 것으로, 더 상세하게는, 웹페이지들을 렌더링하는 것에 관한 것이다.
- [0002] 관련 출원들에 대한 상호 참조
- [0003] [0002] 본 출원은, 2015년 7월 24일자로 출원된 미국 특허 출원 제 14/809,046호의 출원일에 대한 우선권을 주장하는 2014년 7월 25일자로 출원된 미국 가특허 출원 제 62/029,398호에 대한 우선권의 이득을 주장하며, 이들 출원들 둘 모두는 인용에 의해 본원에 통합된다.

배경기술

- [0004] [0003] 고속 웹페이지 디스플레이를 위한 종래의 기술들은, 애플리케이션 개발자들이, 최소한의 JavaScript 및 CSS 프로세싱이 첫번째 페인트(paint)에 대해 발생하도록 웹페이지들을 설계하는데 의존한다. 그러나, 이러한 접근법은, JavaScript에 의해 생성되는 많은 양의 동적(dynamic) 콘텐츠를 포함하는 웹사이트들에 대해 적합하지 않다. 예를 들어, 몇몇 웹사이트들은, 그들의 사용자 인터페이스(UI) 엘리먼트들 전부를 JavaScript에 의해 생성된 채로 가질 수 있다. 부가적으로, 웹페이지 개발자의 툴(tool)들 및/또는 라이브러리(library)들에 대한 선택들이 제한되기 때문에, 웹페이지 개발자들이 웹페이지들을 더 빠르게 디스플레이되게 하는 것에 의존하는 것은 충분하지 않다. 예를 들어, 웹페이지 개발자들은 종종, 통상적으로 JavaScript로 구현되는 제 3자 라이브러리들에 의존한다.
- [0005] [0004] 브라우저 또는 콘텐츠 전달 네트워크(CDN; content delivery network) 측 상에서의 동작의 속도를 향상시키기 위해 사용되는 종래의 기술은, 캐싱(cache)된 페이지들을 사용하는 것이다. 그러나, 이러한 기술은 고도로 동적인 웹사이트들에 대해 적합하지 않은데, 스테일(stale) 정보가 캐싱되어 이후에 사용자에게 제공될 수 있기 때문이다. "스테일" 정보가 표시되는 방지하기 위해 JavaScript에 의해 생성된 콘텐츠가 전혀 디스플레이되지 않으면, 많은 양의 동적 콘텐츠를 포함하는 이들 웹사이트들은 최소한의 것들을 표시할 것으로 가질 것이다. 이들 단점들을 회피하면서 웹페이지들을 신속하게 렌더링하는 것이 바람직할 수 있다.

발명의 내용

- [0006] [0005] 본 개시내용은 콘텐츠를 렌더링하는 것에 관한 것이다.
- [0007] [0006] 몇몇 실시예들에 따르면, 웹페이지를 렌더링하는 예시적인 방법은, 위치에 의해 참조되는 제 1 웹 문서(document)를 프로세싱하는 것에 기초하여, 제 1 직렬화(serialize)된 최종(eventual) 문서 오브젝트 모듈(DOM; document object module)을 생성하는 단계를 포함한다. 제 1 직렬화된 최종 DOM은, 제 1 인-라인(in-line)된 동적 콘텐츠를 포함한다. 방법은 또한, 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여, 제 2 직렬화된 최종 DOM을 생성하는 단계를 포함한다. 제 2 직렬화된 최종 DOM은, 제 2 인-라인된 동적 콘텐츠를 포함한다. 방법은 추가로, 제 1 및 제 2 직렬화된 최종 DOM들 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성하는 단계를 포함한다. 공통 직렬화된 최종 DOM은 제 1 및 제 2 직렬화된 최종 DOM들에 포함되는 정적인 동적 콘텐츠를 포함하고, 차이는 트루(true) 동적 콘텐츠이다. 방법은 또한, 공통 직렬화된 최종 DOM을 디바이스의 디스플레이에 렌더링하는 단계를 포함한다.
- [0008] [0007] 몇몇 실시예들에 따르면, 웹페이지를 렌더링하기 위한 시스템은, 위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여 제 1 직렬화된 최종 문서 오브젝트 모듈(DOM)을 생성하고 그리고 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여 제 2 직렬화된 최종 DOM을 생성하는 클라이언트 브라우저를 포함한다. 제 1 직렬화된 최종 DOM은 제 1 인-라인된 동적 콘텐츠를 포함하고, 제 2 직렬화된 최종 DOM은 제 2 인-라인된 동적 콘텐츠를 포함한다. 클라이언트 브라우저는, 제 1 및 제 2 직렬화된 최종 DOM들 간의 차이에 기초하여, 공통 직렬화된 최종 DOM을 생성한다. 공통 직렬화된 최종 DOM은 제 1 및 제 2 직렬화된 최종 DOM들에 포함되는 정적인 동적 콘텐츠를 포함하고, 차이는 트루(true) 동적 콘텐츠이다. 클라이언트 브라우저는, 공통 직렬화된 최종 DOM을 디바이스의 디스플레이에 렌더링한다.
- [0009] [0008] 몇몇 실시예들에 따르면, 동작들을 수행하기 위한 컴퓨터-실행가능 명령들이 저장된 컴퓨터-판독가능 매체는, 위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여 제 1 직렬화된 최종 문서 오브젝트 모듈(DOM)을 생성하는 것 - 제 1 직렬화된 최종 DOM은 제 1 인-라인된 동적 콘텐츠를 포함함 -; 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여 제 2 직렬화된 최종 DOM을 생성하는 것 - 제 2 직렬화된

최종 DOM은 제 2 인-라인된 동적 콘텐츠를 포함함 -; 제 1 및 제 2 직렬화된 최종 DOM들 간의 차이에 기초하여 공통 직렬화된 최종 문서 오브젝트 모듈(DOM)을 생성하는 것 - 공통 직렬화된 최종 DOM은, 제 1 및 제 2 직렬화된 최종 DOM들에 포함된 정적인 동적 콘텐츠를 포함하고, 차이는 트루 동적 콘텐츠임; 및 공통 직렬화된 최종 DOM을 디바이스의 디스플레이에 렌더링하는 것을 포함한다.

[0010] [0009] 몇몇 실시예들에 따르면, 웹페이지를 렌더링하기 위한 시스템은, 위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여 제 1 직렬화된 최종 문서 오브젝트 모듈(DOM)을 생성하기 위한 수단 - 제 1 직렬화된 최종 DOM은 제 1 인-라인된 동적 콘텐츠를 포함함 -; 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여 제 2 직렬화된 최종 DOM을 생성하기 위한 수단 - 제 2 직렬화된 최종 DOM은 제 2 인-라인된 동적 콘텐츠를 포함함 -; 제 1 및 제 2 직렬화된 최종 DOM들 간의 차이에 기초하여 공통 직렬화된 최종 DOM을 생성하기 위한 수단 - 공통 직렬화된 최종 DOM은, 제 1 및 제 2 직렬화된 최종 DOM들에 포함된 정적인 동적 콘텐츠를 포함하고, 차이는 트루 동적 콘텐츠임; 및 공통 직렬화된 최종 DOM을 디바이스의 디스플레이에 렌더링하기 위한 수단을 포함한다.

도면의 간단한 설명

[0011] [0010] 본 명세서의 일부를 형성하는 첨부된 도면들은 본 개시내용의 실시예들을 예시하고, 설명과 함께, 실시예들의 원리들을 설명하도록 추가적으로 기능한다. 도면들에서, 동일한 참조 번호들은 동일하거나 기능적으로 유사한 엘리먼트들을 표시할 수 있다. 엘리먼트들이 처음 나타나는 도면은 일반적으로, 대응하는 참조 번호의 최좌측 숫자에 의해 표시된다.

[0011] 도 1은, 몇몇 실시예들에 따른 예시적인 통신 시스템이다.

[0012] 도 2는 몇몇 실시예들에 따른, 최종 DOM을 직렬화하기 위한 프로세스 흐름이다.

[0013] 도 3은 몇몇 실시예들에 따른, 웹 문서 내의 트루 동적 콘텐츠를 식별하기 위한 프로세스 흐름이다.

[0014] 도 4는 몇몇 실시예들에 따른, 2개의 웹페이지들 간의 차이를 제공하기 위한 서버와 클라이언트들 간의 통신들의 예시적인 스윙 다이어그램(swim diagram)을 예시한다.

[0015] 도 5는 몇몇 실시예들에 따른, 듀얼 렌더링(dual render) 프로세스 접근법을 사용하여 실제 웹페이지 및 공통 직렬화된 최종 DOM을 디스플레이하기 위한 예시적인 프로세스 흐름을 예시한다.

[0016] 도 6은 몇몇 실시예들에 따른, 웹페이지를 렌더링하는 방법을 예시하는 흐름도이다.

[0017] 도 7은, 본 개시내용의 하나 또는 그 초과 실시예들을 구현하는데 적절한 컴퓨터 시스템의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0012] I. 개요

[0013] II. 예시적인 시스템 아키텍처

[0014] III. 웹페이지 렌더링

[0015] A. 오리지널(Original) DOM에 기초한 최종 DOM 구축

[0016] B. 최종 DOM 직렬화

[0017] 1. 최종 DOM의 시각적 표현에 대응하는 직렬화된 최종 DOM 구축

[0018] 2. 최종 DOM을 파일로 직렬화

[0019] C. 트루 및 정적인 동적 콘텐츠의 식별 및 프로세싱

[0020] D. 직렬화된 최종 DOM 관리 및 전달을 위한 메커니즘들

[0021] E. 요청된 웹페이지를 디스플레이하기 위한 메커니즘들

[0022] 1. 듀얼 렌더링 프로세스 접근법

[0023] 2. 임베디드(Embedded) JavaScript 접근법

- [0024] IV. 예시적인 방법
- [0025] V. 예시적인 디바이스
- [0026] I. 개요
- [0027] [0018] 다음의 개시내용이 본 개시내용의 상이한 특징들을 구현하기 위한 많은 상이한 실시예들 또는 예들을 제공한다는 것이 이해될 것이다. 몇몇 실시예들은, 이들 특정 세부사항들 중 일부 또는 전부 없이 실시될 수 있다. 본 개시내용을 간략히 하기 위해, 컴포넌트들, 모듈들, 및 어레인지먼트들의 특정 예들이 아래에 설명된다. 이들은, 물론, 단지 예들이고, 제한적인 것으로 의도되지 않는다.
- [0028] [0019] 본 개시내용은, 웹 개발자들의 그들의 웹사이트들을 변경하는 것을 요구하지 않으면서 가능한 한 많은 년-스테일(non-stale) 동적 콘텐츠들을 고속 렌더링하는 것의 모순되는(contradictory) 요건들을 달성하기 위한 브라우저/서버 프레임워크(framework)를 제공한다. 많은 대중적인 웹사이트들은, 클라이언트 측 스크립팅(scripting)을 사용하여 클라이언트 측에서 생성되는 동적 콘텐츠를 갖는다. 동적 콘텐츠들을 갖는 웹페이지들을 신속하게 그리고 가능한 한 많이 렌더링하면서 최소한의 스테일 정보를 표시하는 것이 바람직할 수 있다. 정적인 동적 콘텐츠는, 예를 들어, 프레임들, 레이아웃, 사이드 바, 메뉴들, 포스트 템플릿(post template)들, 친구 리스트 등을 지칭할 수 있다. 트루 동적 콘텐츠는, 페이지 로드(load)들에 걸쳐 상당히 자주 변경되는 임의의 콘텐츠(예컨대, 뉴스피드(newsfeed))를 지칭할 수 있다.
- [0029] [0020] 웹페이지들을 더 신속하게 렌더링하기 위해, 본 개시내용은, 최종 DOM의 직렬화, (스테일 정보를 표시하는 것을 회피하기 위한) 동적 콘텐츠 식별 및 프로세싱, (콘텐츠의 관리 및 전달을 위한) 직렬화된 최종 DOM을 생성하기 위한 메커니즘들, 및 직렬화된 최종 DOM 및 실제 웹페이지를 디스플레이하기 위한(또한, 최종 DOM을 어떻게 사용하는지에 대한) 메커니즘들과 같은 기술들을 제공한다. 웹페이지가 완전히 로딩된 이후의 DOM이 웹페이지의 "최종 DOM"으로 지칭될 수 있다.
- [0030] [0021] 본 개시내용에 설명된 하나 또는 그 초과 기술들을 포함하는 실시예는, 가장 동적인 웹사이트들조차도 더 고속으로 디스플레이하고, 스테일 정보가 사용자에게 디스플레이 되는 것을 회피하고, 그리고/또는 동적으로 생성되지만 고속 디스플레이를 위해 변경되지 않는 아이템(item)들을 보존할 수 있다. 부가적으로, 솔루션은, 서버의 지원으로 또는 지원 없이 브라우저에서 구현될 수 있다. 따라서, 개발자가 오리지널 웹사이트를 변경할 필요가 없을 수 있다.
- [0031] II. 예시적인 시스템 아키텍처
- [0032] [0022] 도 1은, 몇몇 실시예들에 따른 예시적인 통신 시스템(100)이다. 시스템(100)은, 네트워크(140)를 통해 서버(120)에 통신가능하게 커플링되는 클라이언트 브라우저(110)를 포함한다. 하나의 클라이언트 브라우저 및 하나의 서버를 참조하여 시스템(100)이 본원에 설명되지만, 당업자는, 시스템(100)이 하나 초과 클라이언트 브라우저 및/또는 서버를 포함할 수 있다는 것을 인지할 것이다.
- [0033] [0023] 브라우저들은, World Wide Web 상의 웹 애플리케이션 또는 웹사이트 상에서와 같이 네트워크 상에서 이용가능한 콘텐츠를 사용자가 보거나 다운로드하는 것을 허용하는 소프트웨어 애플리케이션들이다. 콘텐츠는, 텍스트, 파일들, 이미지들, 오디오, 비디오, 또는 개인 통신들을 포함할 수 있다. 웹페이지들은 그러한 콘텐츠를 제공하고, URL(uniform resource locator)과 같은 URI(uniform resource identifier)들을 사용하여 로케이팅된다. 웹페이지들은, 웹페이지 콘텐츠를 보유하는 컴퓨터의 인터넷 프로토콜(IP) 어드레스를 사용하여 검색(retrieve)될 수 있다. 브라우저들은, 콘텐츠 플로우(content flow)를 획득하거나 또는 관리하기 위해 다수의 프로토콜들 및 표준들을 사용할 수 있다. 예를 들어, 브라우저들은, 콘텐츠 및 웹페이지들을 페칭(fetch)하기 위해 HTTP(hypertext transfer protocol)을 사용할 수 있다. HTTP는, 기본 요청/응답 의미론(semantics)을 제공하는 애플리케이션-레벨 프로토콜이다. HTTP는, 서버에 오브젝트들을 요청하고 그리고 서버로부터 오브젝트들을 검색하는데 사용될 수 있다.
- [0034] [0024] 서버(120)는, 예를 들어, 제한 없이, 웹 서버, 원격통신 서버, 데이터베이스 서버 등일 수 있다. 웹 서버는, 클라이언트로부터의 HTTP 요청에 대한 응답으로 콘텐츠를 생성하는 웹 애플리케이션들을 포함할 수 있다. 웹 서버는, 생성된 콘텐츠를 패키징(package)하고, HTTP 응답의 형태로 그 콘텐츠를 클라이언트에 서빙(serve)한다. 웹 서버는, HTTP 응답으로 HTTP 요청에 대해 응답하는 소프트웨어 컴포넌트일 수 있다. 예시적인 예들로서, 웹 서버는, 제한 없이, Apache HTTP Server, Apache Tomcat, Microsoft Internet Information Server, JBoss Application Server, WebLogic Application Server, 또는 Sun Java System Web Server일 수 있다.

다. 상표들은 그들 각각의 소유자들의 재산들이다. 웹 서버는, HTML(hypertext markup language), XML(extendable markup language), 문서들, 비디오들, 이미지들, 멀티미디어 피쳐(feature)들, 또는 이들의 임의의 결합과 같은 콘텐츠를 서빙할 수 있다. 이들 예들은 절대적으로 예시적이며, 본 개시내용의 범위를 제한하지 않는다.

[0035] [0025] 클라이언트 브라우저(110)는, 임의의 컴퓨팅 디바이스 상에서 실행될 수 있다. 유사하게, 서버(120)는, 클라이언트에 데이터를 서빙하는 것이 가능한 임의의 컴퓨팅 디바이스를 사용하여 구현될 수 있다. 컴퓨팅 디바이스들의 예들은, 중앙 프로세싱 유닛, 주문형 집적 회로, 컴퓨터, 워크스테이션(workstation), 분산형 컴퓨팅 시스템, 컴퓨터 클러스터(cluster), 임베디드 시스템, 독립형(stand-alone) 전자 디바이스, 네트워크형(networked) 디바이스, 모바일 디바이스(예컨대, 모바일 폰, 스마트 폰, 개인 휴대 정보 단말(PDA), 내비게이션 디바이스, 태블릿, 또는 모바일 컴퓨팅 디바이스), 랙 서버(rack server), 셋-톱(set-top) 박스, 또는 적어도 하나의 프로세서 및 메모리를 갖는 다른 타입의 컴퓨터 시스템을 포함하지만 이에 제한되지 않는다. 클러스터링된 컴퓨팅 환경 또는 서버 팜(farm)에 의해 수행되는 컴퓨팅 프로세스는, 동일한 또는 상이한 위치들에 로케이팅된 다수의 프로세서들에 걸쳐 수행될 수 있다. 그러한 컴퓨팅 디바이스는, 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 결합을 포함할 수 있다. 소프트웨어는, 하나 또는 그 초과 애플리케이션들 및 운영 시스템을 포함할 수 있다. 하드웨어는, 프로세서, 메모리, 및 사용자 인터페이스 디스플레이를 포함할 수 있지만 이에 제한되지 않는다.

[0036] [0026] 네트워크(140)는, 데이터 통신을 반송(carry)할 수 있는 임의의 네트워크 또는 네트워크들의 결합일 수 있고, 본원에서 컴퓨터 네트워크로서 지칭될 수 있다. 그러한 네트워크(140)는, 유선 네트워크(예컨대, 이더넷) 또는 무선 네트워크(예컨대, Wi-Fi 및 3G) 네트워크, 로컬 영역 네트워크, 중규모 영역(medium area) 네트워크, 및/또는 광역 네트워크, 이를테면 인터넷을 포함할 수 있지만 이에 제한되지 않는다. 네트워크(140)는, World Wide Web 프로토콜들 및/또는 서비스들을 포함하지만 이에 제한되지 않는 프로토콜들 및 기술을 지원할 수 있다. 특정 애플리케이션 또는 환경에 의존하여, 중간(Intermediate) 웹 서버들, 게이트웨이들, 또는 다른 서버들이 시스템(100)의 컴포넌트들 간에 제공될 수 있다.

[0037] [0027] 클라이언트 브라우저(110)는, 서버에 대한 일련의 요청들 및 서버로부터의 응답들을 통해 웹페이지들을 검색할 수 있다. 일 예에서, 클라이언트 브라우저(110)는, 네트워크(140)를 통해 웹페이지에 대한 요청을 서버(120)에 전송한다. 서버(120)는, 요청을 수신하고, 요청을 프로세싱하며, 그리고 요청된 웹페이지에 대한 콘텐츠와 같은 응답을 리턴(return)한다. 일 예에서, 서버(120)는, 클라이언트 브라우저(110)의 요청에 대한 응답으로 HTML 페이지(132)를 클라이언트 브라우저(110)에 리턴한다. HTML 페이지(132)는, 마크업 언어(markup language)(HTML)로 되어 있고 하나 또는 그 초과 엘리먼트들(예컨대, 태그(tag)들)의 세트, 및 하나 또는 그 초과 규칙들의 세트를 포함하는 웹 문서이다. 규칙은, 예를 들어, 하나 또는 그 초과 엘리먼트들 또는 CSS(Cascading Style Sheet) 엘리먼트에 적용된 JavaScript일 수 있다. CSS는, 웹 문서들을 포맷팅(format)하고 렌더링하는데 사용되는 프레젠테이션(presentation) 규칙들, 속성들, 및 부수적인 구성들을 정의하기 위한 선언적 구문(declarative syntax)이다. CSS를 사용함으로써, 문서(예컨대, HTML 페이지(132)) 내의 엘리먼트들(예컨대, HTML 엘리먼트들)의 외관(appearance) 및 포맷팅은, 엘리먼트들을 부가하거나 또는 제거하지 않으면서 신속하게 변경될 수 있다. 따라서, HTML 페이지(132)의 사이즈는 작을 수 있고, 문서를 조작하는 스크립트들을 고속일 수 있다.

[0038] [0028] 클라이언트 브라우저(110)는, HTML 페이지(132)를 렌더링하기 위해 외부 리소스들을 다운로드할 수 있다. 예를 들어, HTML 페이지(132)는, CSS, JavaScript 파일들, 이미지들, 및 다른 리소스들과 같은 다른 컴포넌트들의 어드레스들을 포함할 수 있다. 클라이언트 브라우저(110)는, 이들 각각을, 차례로 서버에 요청함으로써, 이들을 웹페이지의 최종 렌더링으로 결합시킨다. 클라이언트 브라우저(110)가 HTML 페이지(132)를 렌더링하는 것은, 클라이언트 브라우저(110)가 HTML 페이지(132)를 렌더링할 수 있기 이전에 다수의 동작들을 수행할 수 있기 때문에, 시간 소모가 클 수 있다. 예를 들어, HTML 페이지(132)는 JavaScript를 포함할 수 있는데, 이러한 경우에서, 클라이언트 브라우저(110)는, JavaScript를 다운로드 및 실행하여 클라이언트에 동적 콘텐츠를 생성할 수 있는데, 이는 시간을 소요한다. 그 후, 생성된 동적 콘텐츠는 웹페이지의 일부가 된다.

[0039] [0029] HTML 페이지(132)내의 일부 콘텐츠는, 각각의 페이지 로드 대해 변경될 가능성이 있는 "트루" 동적 콘텐츠(예컨대, 뉴스 피드들)일 수 있다. 트루 동적 콘텐츠는, 동일한 URL에 의해 참조되는 페이지들의 로딩에 걸쳐 상이할 수 있고, 통상적으로, 각각의 페이지 로드 이후에 스테일하게 될 수 있다. 예를 들어, 제 1 일에 사용자는 소셜 미디어 웹사이트를 열람하고, 사용자의 친구들에 의해 포스팅된 뉴스 피드들을 디스플레이하는 웹페이지를 요청할 수 있다. 레이아웃, 사이드바들, 및 템플릿들의 디스플레이는 페이지 로드들에 걸쳐 동일할

수 있지만, 사용자에게 디스플레이되는 포스팅된 뉴스 피드들은 사용자가 웹페이지를 리프레시(refresh)하거나 또는 요청할 때마다 변경될 수 있다.

[0040] [0030] HTML 페이지(132) 내의 일부 콘텐츠는 "정적인" 동적 콘텐츠일 수 있는, 이는, 정적 콘텐츠가 페이지 로드들 간에 변경되지 않는다는 관점에서는 "정적"이지만, 콘텐츠가 클라이언트 측 스크립트(예컨대, JavaScript로 작성된 스크립트)에 의해 생성된다는 관점에서는 "동적"이다. 정적인 동적 콘텐츠들의 예들은, 사용자가 로그-인한 이후의 웹사이트의 레이아웃, 크롬(chrome), 배너, 양식 제어부들(텍스트 박스들 및 버튼들), 메뉴들, 백그라운드(background) 등이다. 그러한 웹사이트들의 전체 사용자 인터페이스는, 예를 들어, JavaScript를 사용하여 생성될 수 있으며, 이러한 콘텐츠는, 주요한 설계 변경들(이는 빈번하게 발생하지 않음)이 발생하지 않는 한 동일하게 유지된다.

[0041] [0031] 단일 웹페이지가 트루 동적 콘텐츠 및 정적인 동적 콘텐츠 둘 모두를 갖는 것은 일반적이다. 트루 동적 콘텐츠 및 정적인 동적 콘텐츠 둘 모두는 그들 둘 모두가 온-더-플라이(on-the-fly)로 생성된다는 관점에서는 "동적 콘텐츠"이지만, 정적인 동적 콘텐츠는 페이지 로드들에 걸쳐 동일할 더 높은 확률을 갖는다. 웹페이지들을 더 고속으로 렌더링하기 위해서는, 웹페이지 내의 트루 동적 콘텐츠 및 정적인 동적 콘텐츠를 식별하고 그들을 서로 구별하는 것이 바람직할 수 있다.

[0042] [0032] 웹 페이지를 더 고속으로 렌더링하기 위해, 클라이언트 브라우저(110)는 웹페이지들을 캐싱할 수 있다. 클라이언트 브라우저(110)가 캐싱된 웹페이지에 대응하는 HTML 페이지(132)를 서버(120)로부터 수신하면, 클라이언트 브라우저(110)는, HTML 페이지(132)에 대응하는 캐싱된 페이지를 디스플레이할 수 있다. 그러나, 스테일 정보가 사용자에게 제공될 수 있다(예컨대, 어제의 뉴스와 같은 잘못된 "트루" 동적 콘텐츠들이 캐시로부터 디스플레이될 수 있음). 본 개시내용은, 브라우저/서버 프레임워크를 사용하여 최소한의 스테일 정보를 표시하면서 년-스테일 동적 콘텐츠들의 고속 렌더링에 대한 모순되는 요건들을 달성하기 위한 기술들을 제공한다.

[0043] III. 웹페이지 렌더링

[0044] [0033] 웹페이지를 완전히 로딩하여 디스플레이하기 위해, 클라이언트 브라우저(110)는, HTML 페이지(132)를 다운로드하고, HTML 페이지(132)를 파싱(parse)하고, HTML 페이지(132)의 구조의 문서 오브젝트 모듈(DOM)을 구축하며, 그리고 DOM을 렌더링한다. 많은 양의 동적 콘텐츠를 갖는 웹페이지에서, 가장 시간 소모가 큰 동작들은, 동적 콘텐츠(예컨대, 외부 소스로부터의 JavaScript 스크립트)를 다운로드하는 것, 동적 콘텐츠를 실행하는 것(예컨대, JavaScript를 실행하는 것), 및 스타일 계산할 수 있다. 이들 동작들이 수행된 이후, 클라이언트 브라우저(110)는 DOM을 렌더링할 수 있다. DOM을 렌더링하는 것은, 실제 웹페이지 콘텐츠를 스크린 상에 페인팅(paint)한다.

[0045] [0034] 도 2는 몇몇 실시예들에 따른, 최종 DOM을 직렬화하기 위한 프로세스 흐름(200)이다. 도 2는, 렌더링을 위해 서버(120)에 의해 클라이언트 브라우저(110)로 전송되는 "미가공(raw)" HTML 페이지(132)의 예를 제공한다. 클라이언트 브라우저(110)는, HTML 페이지(132)를 프로세싱하고, HTML 페이지(132)의 표현인 오리지널 DOM(202)을 속성들 및 체계성(method)들을 갖는 구조화된 그룹의 노드들 및 오브젝트들로서 구축한다. 오리지널 DOM(202)은, HTML 페이지(132) 내의 마크업 엘리먼트들의 계층(hierarchy)에 대한 트리(tree) 구조로서 시각화된다.

[0046] [0035] HTML 페이지(132) 내의 오브젝트 속성들은, HTML 엘리먼트들의 속성들에 대한 직접적인 맵핑(mapping)들이다. HTML 구문과 DOM 간에 직접적인 맵핑이 존재한다. 오리지널 DOM(202)에서, "html" 노드(204)는 HTML 페이지(132) 내의 <html> 엘리먼트에 대응하고, "body" 노드(206)는 HTML 페이지(132) 내의 <body> 엘리먼트에 대응하고, "Text1" 노드(208)는 HTML 페이지(132) 내의 "Text1"에 대응하고, "Js1.js" 노드(210)는 HTML 페이지(132) 내의 <js1.js> 엘리먼트에 대응하며, 그리고 "Css1.css" 노드(212)는 HTML 페이지(132)의 <link css1.css> 엘리먼트에 대응한다. HTML 페이지(132)는 또한, 2개의 종료(ending) 태그들 </body> 및 </html>을 포함한다. DOM은, 스타일들에 대한 프로그래밍적 액세스를 제공한다. DOM 내의 CSS 오브젝트들은 "라이브(live)"인데, 즉, 스타일 시트에서의 변경이, 컴퓨팅된 스타일 및 실제의 스타일에 반영된다. 따라서, 개별적인 엘리먼트들 상의 인라인 스타일들 및 스타일 규칙들은, 단순한 스크립트-기반 프로그래밍을 사용하여 변경될 수 있다. 스크립트들은, 예를 들어, JavaScript, Microsoft JScript, 또는 VBScript(Microsoft Visual Basic Scripting Edition)로 작성될 수 있다.

[0047] A. 오리지널 DOM에 기초한 최종 DOM 구축

[0048] [0036] 클라이언트 브라우저(110)는, HTML 페이지(132)에서 탐색되는 서브-리소스들을 다운로드한다. 예시적

인 서버-리소스는, HTML 페이지(132)에 포함된 하나 또는 그 초과와 엘리먼트들에 적용될 수 있는 규칙이다. 예시적인 규칙들은, CSS 엘리먼트들 및 JavaScript 스크립트들을 포함한다. 클라이언트 브라우저(110)는, HTML 페이지(132)가 로딩되는 동안 또는 HTML 페이지(132)가 로딩된 직후에 서버-리소스들을 컴파일(compile) 및 실행할 수 있다. 서버-리소스들의 컴파일 및 실행은 동기식 동작일 수 있다. 서버-리소스들의 컴파일 및 실행 동안, 클라이언트 브라우저(110)는, 예컨대 CSS 엘리먼트들을 프로세싱하고, 이미지들을 프로세싱하고, 그리고/또는 JavaScript 스크립트들을 실행할 수 있다.

[0049] [0037] 서버-리소스를 실행하는 것의 결과로서, 클라이언트 브라우저(110)는, 동적 콘텐츠를 생성하거나, HTML 태그들로부터 추가적인 DOM 노드들을 구축하거나, 또는 DOM으로부터 DOM 노드들을 제거할 수 있다. 동적 콘텐츠를 갖는 웹페이지가 완전히 로딩된 이후, 그 스테이지에서의 DOM은 최종 DOM으로 지칭될 수 있는데, 이는, 오리지널 HTML 페이지(132)에 정의된 정적 콘텐츠 뿐만 아니라 서버-규칙들(예컨대, JavaScript 스크립트들, CSS에서 명시된 스타일들 등)에 의해 생성된 새롭게 부가된 그리고/또는 수정된 동적 콘텐츠를 또한 포함한다. 이러한 스테이지에서, 최종 DOM 엘리먼트의 스타일들은 이미 리졸빙(resolve)되어 있다.

[0050] [0038] 도 2에서, 클라이언트 브라우저(110)는, HTML 페이지(132)를 프로세싱하고, 웹페이지의 동적 콘텐츠를 포함하는 최종 DOM(220)을 생성한다. JavaScript를 실행하는 것은, 서버(120)로부터 전송된 오리지널 HTML 페이지(132) 내에 있지 않은 추가적인 DOM 노드들을 부가할 수 있다. 최종 DOM(220)을 생성하기 위해, 클라이언트 브라우저(110)는, (HTML 페이지(132)의 "js1.js" 엘리먼트와 연관된) "js1.js" 노드(210)에 의해 표현되는 JavaScript를 실행하고, 그 스크립트를 실행하는 것에 대한 응답으로, "form" 노드(222)에 의해 표현되는 동적 콘텐츠를 생성한다. "js1.js" 노드(210)에 의해 표현되는 JavaScript 스크립트는, HTML 바이트 스트림(예컨대, document.write())에 "<form>" 태그를 부가하거나 또는 오리지널 DOM(202)에 "form" DOM 노드를 직접 부가함으로써, 오리지널 DOM(202)에 "form"을 동적으로 부가한다. 도 2에 예시된 바와 같이, 최종 DOM(220)은, 오리지널 DOM(202)에 포함되지 않은 "form" 노드(222)를 포함한다. 오리지널 DOM(202)과 유사하게, 최종 DOM(220)은, 노드들 "html" 노드(204), "body" 노드(206), "Text1" 노드(208), "Js1.js" 노드(210), 및 "Css1.css" 노드(212)를 포함한다. 이들 노드들에 부가하여, 최종 DOM(220)은 또한, "form" 노드(222)를 포함한다. 클라이언트 브라우저(110)는, "Css1.css" 노드(212)에 의해 특정되는 스타일들을 HTML 페이지(132)의 적절한 HTML 태그들에 적용하고, 최종 DOM(220)을 사용자에게 렌더링한다.

[0051] B. 최종 DOM 직렬화

[0052] [0039] 클라이언트 브라우저(110)는, 최종 DOM의 시각적 표현에 대응하는 DOM을 구축함으로써 최종 DOM(220)을 렌더링하고, 이를, 클라이언트 브라우저(110)에 커플링되는 디바이스의 디스플레이 상에 페인팅한다. 클라이언트 브라우저(110)는, 직렬화된 최종 DOM(230)을 구축하고 그것을 파일로 직렬화함으로써 최종 DOM(220)을 직렬화할 수 있다. 본 개시내용에서 추가로 논의되는 바와 같이, 트루 동적 콘텐츠 및 정적인 동적 콘텐츠를 식별하고 웹페이지들을 신속하게 렌더링하기 위해, 직렬화된 최종 DOM을 직렬화하는 것이 유리할 수 있다.

[0053] 1. 최종 DOM의 시각적 표현에 대응하는 직렬화된 최종 DOM 구축

[0054] [0040] 클라이언트 브라우저(110)는, 최종 DOM(220)의 동적 콘텐츠를 식별하고, 최종 DOM(220)으로부터의 동적 콘텐츠를 비롯하여 규칙들의 세트(예컨대, "Js1.js" 노드(210) 및 "Css1.css" 노드(212))를 표현하는 노드들을 제거하고, 그리고 동적 콘텐츠를 직렬화된 최종 DOM(230)으로 인-라인시킴으로써 직렬화된 최종 DOM(230)을 생성할 수 있다. 따라서, 클라이언트 브라우저(110)는, 최종 DOM(220)으로부터 JavaScript 스크립트 및 CSS 스타일들을 제거하고, 직렬화된 최종 DOM(230)에서 이러한 동적 콘텐츠를 인-라인시킨다. 몇몇 예들에서, 클라이언트 브라우저(110)는, 최종 DOM(220)에서의 하나 또는 그 초과와 인-라인(non-in-line) 규칙들을 인-라인 규칙들로 이동시킴으로써 직렬화된 최종 DOM(230)을 생성한다.

[0055] [0041] 직렬화된 최종 DOM(230)은 노드들(204, 206, 208', 및 222')을 포함한다. 최종 DOM(220)의 노드들(204 및 206)은 직렬화된 최종 DOM(230)의 노드들(204 및 206)과 동일한 것일 수 있다. 최종 DOM(220)의 "Css1.css" 노드(212)에 의해 표현되는 적절한 스타일들은, 적절한 HTML 태그들/DOM 노드들(예컨대, 최종 DOM(220)의 "Text1" 노드(208) 및/또는 "form" 노드(222)에 의해 표현되는 텍스트)에 적용된다. 인라인 스타일들은, 스타일 속성을 사용하여 HTML 엘리먼트(들)에 적용된 CSS 스타일 지정(assignment)들이다. 몇몇 예들에서, 클라이언트 브라우저(110)는, HTML 페이지(132)의 "<link css1.css>" 엘리먼트에 의해 표현되는 위치에 저장된 외부 CSS 파일의 콘텐츠들을 카피(copy)하여 HTML 페이지(132)의 헤드 섹션의 스타일 태그들 간에 그 콘텐츠들을 페이스팅(paste)함으로써 CSS 스타일들을 인라인시킬 수 있다.

[0056] 2. 최종 DOM을 파일로 직렬화

[0057] [0042] 클라이언트 브라우저(110)는, 직렬화된 최종 DOM(230)을 직렬화된 파일(240)(예컨대, HTML 파일)로 직렬화시킬 수 있으며, 엘리먼트들의 스타일들 전부가 인-라인된다. 최종 DOM(220)의 직렬화된 직렬화된 파일(240)은, 올바른(correct) 스타일들로 정적 및 동적 콘텐츠 둘 모두를 포함하는 "평탄화된(flattened)" 파일이다. 직렬화된 파일(240)은, 최종적으로 사용자에게 디스플레이되는 "실제 웹페이지"와 동일한 콘텐츠들을 표시하는 HTML 파일일 수 있다. 클라이언트 브라우저(110)는, 이후의 검색 및 로딩을 위해, 직렬화된 파일(240)을 캐시에 저장한다. 다음 번에, HTML 페이지(132)와 연관된 URL에 의해 참조되는 웹페이지를 디스플레이하라는 요청을 수신할 시, 클라이언트 브라우저(110)는 직렬화된 파일(240)을 로딩할 수 있다. 직렬화된 파일(240)은 HTML 페이지(132)와 동일한 외관을 가질 수 있지만, 규칙들을 적용하고 그리고/또는 외부 소스들을 서치(search)할 필요가 없기 때문에 훨씬 더 고속으로 디스플레이될 수 있다. 예를 들어, 클라이언트 브라우저(110)는 JavaScript를 실행하거나 CSS 엘리먼트들을 적용할 필요가 없을 수 있는데, 이들은 통상적으로, 웹페이지를 디스플레이하는데 있어 시간 소모가 큰 단계들이다. 따라서, 다음 번에 웹페이지가 요청될 때 더 고속의 페이지 디스플레이가 달성될 수 있다.

[0058] [0043] 동일한 웹페이지의 직렬화된 최종 DOM이 이용가능하면, 직렬화된 파일이 먼저 디스플레이(더 고속임)될 수 있는 한편, 실제 웹페이지가 백그라운드에서 로딩되고 있다. 실제 웹페이지는, 백그라운드에서의 로딩이 완료된 경우 디스플레이될 수 있다. 그러나, 직렬화된 파일(240)은 스타일 정보를 표시할 수 있다. 예를 들어, 웹사이트가 뉴스피드를 포함하면, 사용자에게 오래된(outdated) 뉴스피드가 표시될 수 있다. 스타일 정보의 디스플레이를 회피하기 위해, 클라이언트 브라우저(110)는, 직렬화된 파일(240) 내의 동적 콘텐츠를 식별하고, 동적 콘텐츠 중 일부를 제거하거나 또는 그것을 범용 및/또는 더미(dummy) 플레이스 홀더(place holder)들로 대체할 수 있다.

[0059] [0044] 동적 콘텐츠를 식별하기 위해, 클라이언트 브라우저(110)는, HTML 문서 내에 규칙들 중 임의의 규칙을 (예컨대, JavaScript 및 CSS 엘리먼트들) 적용하기 이전에, 최종 DOM과 오리지널 메인 HTML의 대응 오리지널 DOM 간의 차이를 식별할 수 있다. 클라이언트 브라우저(110)는, 최종 DOM(220)과 오리지널 DOM(202) 간의 차이를 결정하여 최종 DOM으로부터 차이를 제거하고, 그 후에 최종 DOM을 디스플레이할 수 있으며, 여기서, 차이는 동적 콘텐츠이다. 그러나, 이러한 접근법은, 모든 동적 콘텐츠(트루 동적 콘텐츠 및 정적인 동적 콘텐츠 둘 모두)를 제거할 수 있다. 순수 효과(net effect)는, 브라우저에서 JavaScript를 디스에이블링(disable)하여 그 후 오리지널 HTML을 로딩하는 것과 유사할 수 있다. 추가로, 스타일들이 JavaScript에 의해 또한 수정될 수 있는데, 이는 최종 DOM에서는 유지되지만 오리지널 미가공 HTML에서는 유지되지 않기 때문에 순수 효과가 상이할 수 있다. 많은 정적인 동적 콘텐츠를 갖는 웹사이트들의 경우, 사용자에게 거의 전부가 표시되지 않을 수 있는데, 사용자 인터페이스 엘리먼트들 중 어떠한 것도 또는 거의 전부가 사용자에게 디스플레이되지 않을 것이기 때문이다. 따라서, 트루 동적 콘텐츠와 정적인 동적 콘텐츠를 구별하여 오직 트루 동적 콘텐츠만을 제거하는 것이 바람직할 수 있다.

[0060] [0045] 몇몇 실시예들에서, 직렬화된 DOM을 생성하는 방법은, 제 1 웹 문서를 파싱하는 단계를 포함한다. 제 1 웹 문서는, 마크업 언어로 되어 있고 엘리먼트들의 세트 및 규칙들의 세트를 포함할 수 있다. 방법은 또한, 규칙들의 세트를 엘리먼트들의 세트의 하나 또는 그 초과 엘리먼트들에 적용하는 단계, 및 웹 문서를 파싱하는 것 및 규칙들의 세트를 웹 문서에 적용하는 것에 따라 최종 DOM을 구축하는 단계를 포함할 수 있다. 방법은 추가로, 최종 DOM을 컴퓨팅 디바이스의 디스플레이 상에 렌더링하는 단계, 및 최종 DOM에 포함된 규칙들의 세트를 식별하는 단계를 포함할 수 있다. 방법은 또한, 최종 DOM으로부터 규칙들의 세트를 제거하는 단계, 및 최종 DOM을 제 2 웹 문서로 직렬화시키는 단계를 포함할 수 있다. 직렬화된 최종 DOM은, 브라우저의 캐시에 캐싱될 수 있다.

[0061] C. 트루 및 정적인 동적 콘텐츠의 식별 및 프로세싱

[0062] [0046] 스타일 정보의 디스플레이를 회피하기 위해, 클라이언트 브라우저(110)는, 직렬화된 파일(240) 내의 트루 동적 콘텐츠 및 정적인 동적 콘텐츠를 식별하고, 정적인 동적 콘텐츠 중 적어도 일부를 제거하거나 또는 그것을 범용 및/또는 더미 플레이스 홀더들로 대체한다. 트루 동적 콘텐츠를 식별하기 위해, 클라이언트 브라우저(110)는, 웹페이지 로드들과 연관된 최종 DOM들을 유지하고, 클라이언트 브라우저(110)가 동일한 URL들 중 하나 또는 그 초과에 대응하는 웹페이지를 로딩할 때마다 각각의 최종 DOM을 저장할 수 있다. 각각의 페이지 로드 이후, 클라이언트 브라우저(110)는 이후의 사용을 위해, 직렬화된 파일(240)(예컨대, HTML 파일로 직렬화된, 직렬화된 최종 DOM)을 저장할 수 있다.

- [0063] [0047] 일 예에서, 클라이언트 브라우저(110)는, 동일한 웹페이지(동일한 URL에 의해 참조되는 웹페이지)에 대한 요청들과 연관된 복수의 최종 DOM들을 생성하고, 클라이언트 브라우저(110)가 동일한 웹페이지를 로딩할 때마다 캐시에 각각의 최종 DOM을 저장한다. 클라이언트 브라우저(110)는 그 후, 동일한 웹페이지에 대한 복수의 최종 DOM들 간의 차이들을 결정한다. 비교된 복수의 최종 DOM들에서의 상이한 부분들은, 트루 동적 콘텐츠로 식별될 수 있다.
- [0064] [0048] 도 3은 몇몇 실시예들에 따른, 웹 문서 내의 트루 동적 콘텐츠를 식별하기 위한 프로세스 흐름(300)이다. 도 3은, 직렬화된 최종 DOM(302) 및 직렬화된 최종 DOM(304)을 포함한다. 직렬화된 최종 DOM들(302 및 304)은, 위에서 나타난 바와 같이 생성될 수 있고, 동일한 URL과 연관된다. 직렬화된 최종 DOM들(302 및 304)은, 동일한 웹페이지에 대한 직렬화된 최종 DOM의 버전들일 수 있고, 메모리에 유지 및 저장된다.
- [0065] [0049] 직렬화된 최종 DOM(302)은, "html" 노드(306), "body" 노드(308), "<Menu>" 노드(310), "<feed>" 노드(312), "Item1" 노드(314), "Item2" 노드(316), 및 "News A" 노드(318)를 포함한다. 직렬화된 최종 DOM(304)은, "html" 노드(306), "body" 노드(308), "<Menu>" 노드(310), "<feed>" 노드(312), "Item1" 노드(314), "Item2" 노드(316), 및 "News B" 노드(320)를 포함한다. "html" 노드(306), "body" 노드(308), "<Menu>" 노드(310), 및 "<feed>" 노드(312)는, HTML 엘리먼트들/서브트리(subtree)를 표현하는 "정적" 노드들일 수 있다. "Item1" 노드(314), "Item2" 노드(316), "News A" 노드(318) 및 "News B" 노드(320)는, JavaScript에 의해 생성되는 동적 콘텐츠일 수 있다. 클라이언트 브라우저(110)가 제 1 일 및 제 2 일에 소셜 미디어 웹사이트로부터 웹페이지를 로딩했다면, 클라이언트 브라우저(110)는, 이들 2개의 웹페이지 로드들과 연관된 직렬화된 최종 DOM들(302 및 304)의 버전들 둘 모두를 캐시 또는 데이터베이스에 저장할 수 있다.
- [0066] [0050] 일 예에서, 클라이언트 브라우저(110)는, 직렬화된 최종 DOM(302) 및 직렬화된 최종 DOM(304)을 비교하고, 변경된 하나 또는 그 초과 부분들을 식별한다. 하나 또는 그 초과 부분들은, 직렬화된 최종 DOM들에 걸친 트루 동적 콘텐츠이다. 직렬화된 최종 DOM(302)은 "News A" 콘텐츠를 포함하고, 직렬화된 최종 DOM(304)은 "News B" 콘텐츠를 포함한다. 따라서, "News X"는 트루 동적 콘텐츠로 식별된다. 트루 동적 콘텐츠(예컨대, 뉴스피드들 및 뉴스의 콘텐츠들)는, 2개의 직렬화된 최종 DOM들 간에 상이하지만, 정적인 동적 콘텐츠(예컨대, 사용자 인터페이스 엘리먼트들)는 동일하게 유지된다. 클라이언트 브라우저(110)가 직렬화된 최종 DOM(302)을 캐싱하고 그리고 직렬화된 최종 DOM(304)이 아니라 그것을 사용자에게 디스플레이했다면, 사용자는, ("News B" 콘텐츠에 의해 표현되는) 가장 최신 뉴스피드가 아니라 ("News A" 콘텐츠에 의해 표현되는) 스테일 뉴스피드를 제공받을 수 있다. 스테일 정보를 사용자에게 디스플레이하는 것을 회피하기 위해, 클라이언트 브라우저(110)는, 트루 동적 콘텐츠를 제거하여 공통 직렬화된 최종 DOM(330)을 생성한다. 공통 직렬화된 최종 DOM(330)은, 직렬화된 최종 DOM(302) 또는 직렬화된 최종 DOM(304)으로부터 제거된 트루 동적 콘텐츠를 갖는 DOM이다.
- [0067] [0051] 몇몇 예들에서, 클라이언트 브라우저(110)는, 직렬화된 최종 DOM(302)(또는 직렬화된 최종 DOM(304))으로부터 트루 동적 콘텐츠를 표현하는 노드를 제거함으로써 트루 동적 콘텐츠를 제거한다. 트루 동적 콘텐츠를 표현하는 노드가 제거되면, 웹페이지의 레이아웃이 변경될 수 있다. 예를 들어, 큰 배너 이미지가 트루 동적 콘텐츠일 수 있고, 큰 배너 이미지가 제거되면, 웹페이지의 레이아웃의 나머지는 사용자에게 이상하게 보일 수 있다. 도 3에 예시된 예에서, 클라이언트 브라우저(110)는, 트루 동적 콘텐츠를 표현하는 노드(예컨대, 노드(318))를 제거하는 것이 아니라, 트루 동적 콘텐츠를 표현하는 노드를 플래스홀더 엘리먼트로서 동작하는 더미 노드(332)로 대체할 수 있다. 더미 노드(332)는, 웹페이지 레이아웃을 이전과 동일하게 유지하기 위해, 동일한 사이즈(또는 심지어, 예를 들어, 큰 이미지이면, 동일한 색상 체계)를 가질 수 있다.
- [0068] [0052] 트루 동적 콘텐츠는 또한, 서브-DOM들로부터 제거될 수 있다. 예를 들어, iFrame들은 자신 고유의 서브-DOM을 가질 수 있고, HTML 문서들은 일반적으로 캐싱가능하지 않다. 따라서, 클라이언트 브라우저(110)는, iFrame을 공통 직렬화된 최종 DOM에 유지할 수 있거나, 또는 소스 URL이 변경되지 않으면, iFrame을 동일하게 사이징(size)된 더미 노드로 대체할 수 있다.
- [0069] [0053] 트루 동적 콘텐츠를 표현하는 노드를 제거하거나 또는 그 노드를 더미 노드로 대체함으로써 트루 동적 콘텐츠(예컨대, "News X")를 제거한 이후, 직렬화된 최종 DOM의 나머지가 가능한 한 적은 스테일 콘텐츠들로 디스플레이될 수 있다. 공통 직렬화된 최종 DOM(330)은, 잠재적인 스테일 콘텐츠를 포함하지 않는다. 오리지널 HTML 페이지(웹사이트로부터 다운로드되는 실제 HTML 페이지)와 직렬화된 최종 DOM의 HTML 간의 차이들을 식별하거나 또는 직렬화된 최종 DOM들의 상이한 버전들 간의 차이들을 식별함으로써, 어떠한 트루 동적 콘텐츠도 포함함이 없이 공통 직렬화된 최종 DOM이 생성될 수 있다. 직렬화된 최종 DOM들 간의 비교는, 트루 동적 콘텐츠

를 제외시키기 위해, 다른 시간들에서 수행되어 최종 DOM을 정교화(refine)할 수 있다. 부가적으로, 직렬화된 최종 DOM의 2개 초과 상이한 버전들이 비교될 수 있다.

[0070] D. 직렬화된 최종 DOM 관리 및 전달을 위한 메커니즘들

[0071] [0054] 몇몇 실시예들에서, 클라이언트 측 상에서, 클라이언트 브라우저(110)는, 로컬 캐시 또는 데이터베이스(웹페이지의 URL에 의해 인덱싱(index)됨)에 직렬화된 최종 DOM을 저장할 수 있다. 클라이언트 브라우저(110)는 또한, 직렬화된 최종 DOM에 대한 만료일을 셋팅할 수 있다. 직렬화된 최종 DOM들의 상이한 버전들 간의 차이(들)를 식별하는 것은, 백그라운드에서 주기적으로 수행될 수 있다. 일 예에서, 웹페이지가 로딩될 때마다, 동일한 URL의 직렬화된 최종 DOM이 사용될 수 있는지 여부를 결정하기 위해, URL을 사용하여 로컬 캐시 또는 데이터베이스가 질의를 받을 수 있다. 다음 번에 동일한 URL이 요청될 때 사용될 클라이언트들로부터의 메타데이터를 서버가 수용하면, 클라이언트 브라우저(110)는 직렬화된 최종 DOM을 또한 서버에 제출할 수 있다.

[0072] [0055] 위에 설명된 바와 같이, 각각의 페이지가 직렬화된 최종 DOM을 로딩한 이후, 이후의 사용을 위한 HTML이 생성 및 저장될 수 있다. 이러한 직렬화된 최종 DOM 생성 및 비교는, 클라이언트 측(클라이언트 브라우저(110)에서) 또는 서버 측(예컨대, 웹 프록시(proxy) 서버 또는 CDN 엣지 서버(edge server)) 중 어느 하나에서 수행될 수 있다. 따라서, 클라이언트 브라우저(110)가 최종 DOM들 및 직렬화된 최종 DOM들을 생성하고 그리고/또는 저장하는 것으로 설명되지만, 서버가(예컨대, CDN(content delivery network)에서) 최종 DOM 및/또는 직렬화된 최종 DOM들을 또한 생성 및/또는 저장할 수 있다는 것이 또한 이해되어야 한다. 예를 들어, 웹페이지가 요청될 때마다, CDN은, 히트(hit)시에, 직렬화된 최종 DOM들을 캐싱할 수 있다.

[0073] [0056] 몇몇 실시예들에서, 서버 측 상에서, 서버(예컨대, CDN 엣지 서버 또는 프록시 서버)는, 다수의 사용자들로부터 동일한 URL에 대한 요청을 수신할 수 있다. 일 예에서, URL 요청이 서버에 의해 서빙될 때마다, 서버는, 직렬화된 최종 DOM을 서버 측에서 생성하고, 직렬화된 최종 DOM을 그 URL에 대한 서버에 저장한다. 서버는, 예를 들어, PhantomJS와 같은 헤드리스(headless) 브라우저를 사용함으로써 최종 DOM을 생성할 수 있다.

[0074] [0057] 도 4는 몇몇 실시예들에 따른, 2개의 웹페이지들 간의 차이를 제공하기 위한 서버와 클라이언트들 간의 통신들의 예시적인 스윙 다이어그램을 예시한다. 도 4에서, 서버 "a.com"(406)은 클라이언트 브라우저들(110 및 410)로부터 다수의 요청들을 수신한다. 클라이언트 브라우저(110)는, URL "http://a.com"(402)에 의해 참조되는 웹페이지(408)에 대한 요청을 전송할 수 있다. 요청은, 클라이언트 브라우저(110)로부터 CDN 서버(404)로 전송될 수 있고, CDN 서버(404)는 그 요청을 서버 "a.com"(406)로 전달하며, 서버 "a.com"(406)은 URL "http://a.com"(402)에 의해 참조되는 웹페이지(408)를 저장한다. 서버 "a.com"(406)은, (index.html(1)에 의해 표현되는) 웹페이지(408)를 CDN 서버(404)에 전송하고, CDN 서버(404)는 그 웹페이지를 클라이언트 브라우저(110)로 전달한다. CDN 서버(404)는, "index.html(1)"에 기초하여, 직렬화된 최종 DOM(420)을 생성하고, 직렬화된 최종 DOM(420)을 URL "http://a.com"(402)에 대한 CDN 서버(404) 상에 저장할 수 있다.

[0075] [0058] 이후 시점에서, 클라이언트 브라우저(410)는, URL "http://a.com"(402)에 의해 참조되는 웹페이지(408)에 대한 요청을 전송할 수 있다. 요청은, 클라이언트 브라우저(410)로부터 CDN 서버(404)로 전송될 수 있고, CDN 서버(404)는 그 요청을 서버 "a.com"(406)로 전달한다. 서버 "a.com"(406)은, (index.html(2)에 의해 표현되는) 웹페이지(408)를 CDN 서버(404)에 전송하고, CDN 서버(404)는 그 웹페이지를 클라이언트 브라우저(110)로 전달한다. CDN 서버(404)는, "index.html(2)"에 기초하여, 직렬화된 최종 DOM(430)을 생성하고, 직렬화된 최종 DOM(430)을 URL "http://a.com"(402)에 대한 CDN 서버(404) 상에 저장할 수 있다.

[0076] [0059] 동일한 URL(예컨대, "http://a.com")에 대한, 그러나 상이한 사용자들(예컨대, 클라이언트 브라우저들(110 및 410))에 대한 상이한 버전들의 직렬화된 최종 DOM(예컨대, 직렬화된 최종 DOM(420) 및 직렬화된 최종 DOM(430))이 서로 비교되어 상이한 콘텐츠를 식별할 수 있으므로, 최후의(final) 직렬화된 최종 DOM은, 모든 사용자들에 대해 공통인 콘텐츠만을 포함할 수 있다. 트루 동적 콘텐츠를 포함하지 않는 최후의 직렬화된 최종 DOM은, 공통 직렬화된 최종 DOM으로 지칭될 수 있다.

[0077] [0060] CDN 서버(404)는, 직렬화된 최종 DOM들을 생성하여 상이한 버전들의 직렬화된 최종 DOM들을 서로 비교하는 것을 계속할 수 있다. 도 4에서, CDN 서버(404)는, 직렬화된 최종 DOM들(420 및 430)을 생성하고 그들을 비교하여, 도 3에 관하여 논의된 바와 같이, 그들의 차이(들)를 결정한다. CDN 서버(404)는, 직렬화된 최종 DOM들(420 및 430) 간의 차이(들)(이는, "Diff=S(index 1)-(index 2)"로 작성될 수 있음)를 트루 동적 콘텐츠인 것으로 식별하고, (예컨대, 트루 동적 콘텐츠를 표현하는 노드를 직렬화된 최종 DOM(420 또는 430)으로부터

제거하거나, 또는 직렬화된 최종 DOM(420 또는 430)에서 트루 동적 콘텐츠를 표현하는 노드를 더미 노드로 대체함으로써) 직렬화된 최종 DOM들(420 또는 430) 중 하나로부터 트루 동적 콘텐츠를 제거할 수 있다.

[0078] [0061] 직렬화된 최종 DOM의 변경되지 않는 부분(들)이 안정화된 이후, CDN 서버(404)는, 직렬화된 최종 DOM들(420 및 430) 간의 차이(들)를 제거한 공통 직렬화된 최종 DOM(440)을 생성하여 공통 직렬화된 최종 DOM들을 클라이언트들에 서빙할 수 있다. 공통 직렬화된 최종 DOM(440)은, "s.index.html=S(index1)-Diff."에 의해 표현된다. 이후 시점에서, 클라이언트 브라우저(410)는, 웹페이지(408)에 대한 다른 요청을 전송할 수 있다.

[0079] E. 요청된 웹페이지를 디스플레이하기 위한 메커니즘들

[0080] [0062] 본 개시내용에서 논의된 기술들을 사용하여, 웹페이지들은 다양한 방식으로 렌더링될 수 있다. 일 예에서, 이러한 요청에 대한 응답으로, CDN 서버(404)는, 어떠한 동적 콘텐츠도 포함하지 않는 공통 직렬화된 최종 DOM(440)을 클라이언트 브라우저(410)에 서빙할 수 있다. CDN 서버(404)는, 상이한 URL(442)을 사용하여, 공통 직렬화된 최종 DOM(440)을 클라이언트 브라우저(410)에 전달할 수 있다. 다른 예에서, CDN 서버(404)는, JavaScript 디스플레이 관리자에 의해 사용될 메인 HTML에 트루 동적 콘텐츠를 임베딩 또는 인라인시킬 수 있다. 따라서, 공통 직렬화된 최종 DOM은, 클라이언트 및/또는 서버에서 생성되고, 별개의 URL을 통해 클라이언트에 전달되거나 또는 메인 HTML 문서 내에 임베딩될 수 있다.

[0081] 1. 듀얼 렌더링 프로세스 접근법

[0082] [0063] 몇몇 실시예들에서, 클라이언트 브라우저(110)는, 웹페이지를 디스플레이하는데 듀얼 렌더링 프로세스 접근법을 사용할 수 있다. 도 5는 몇몇 실시예들에 따른, 듀얼 렌더링 프로세스 접근법을 사용하여 실제 웹페이지 및 공통 직렬화된 최종 DOM을 디스플레이하기 위한 예시적인 프로세스 흐름을 예시한다. 도 5에서, 클라이언트 브라우저(110)는, 서버에 "foo.com"을 요청한다. 요청에 대한 응답으로, 클라이언트 브라우저(110)는, 2개의 렌더링 프로세스들, 즉, 캐시 또는 서버(예컨대, CDN 서버(404) 또는 프록시 서버) 중 어느 하나로부터의 공통 직렬화된 최종 DOM을 요청 및 렌더링하기 위한 하나, 및 사용자에게 디스플레이될 실제 웹페이지에 대한 하나를 생성할 수 있다.

[0083] [0064] 제 1 렌더링 프로세스는 "고속 프로세스"이고, 웹페이지가 신속하게 렌더링되고 있다는 착각의 달성을 위해 사용된다. 제 1 렌더링 프로세스에서, 클라이언트 브라우저(110)는, ("s.foo.com"을 사용하여) 캐시 또는 서버 중 어느 하나로부터 공통 직렬화된 최종 DOM을 로딩하고, 공통 직렬화된 최종 DOM을 사용자에게 먼저 디스플레이한다. 따라서, 클라이언트 브라우저(110)는, "s.foo.com"에 의해 참조되는 웹페이지를 먼저 디스플레이한다. 사용자는, 웹페이지 내의 정적인 동적 콘텐츠를 신속하게 보는 것이 가능한데, 이는, 동적 콘텐츠(예컨대, JavaScript 또는 CSS 엘리먼트들)가 제거되었기 때문이다. 제 2 렌더링 프로세스는 "풀(full) 프로세스"이고, 실제 웹페이지를 요청하여 백그라운드에서 렌더링하는데 사용된다. 제 2 렌더링 프로세스에서, 클라이언트 브라우저(110)는, ("foo.com"을 사용하여) 실제 웹페이지를 로딩하지만, 아직 사용자에게 표시되진 않는다. 클라이언트 브라우저(110)는, 실제 웹페이지에 대한 자신의 렌더링을 완료한 경우에 사용자에게 실제 웹페이지를 디스플레이하며, 사용자에게 최종적으로 표시되는 웹페이지가 깨짐(broken)이 없다는 것이 보장된다.

[0084] [0065] 다시 도 4를 참조하고 듀얼 렌더링 프로세스 접근법을 사용하면, 다음 번에 URL "http://a.com"(402)에 의해 참조되는 웹페이지가 로딩될 때, 클라이언트 브라우저(110)는, 공통 직렬화된 최종 DOM(440)을 사용자에게 포어그라운드(foreground)에서 디스플레이할 수 있는 한편, 백그라운드 프로세스는, 실제의 웹페이지를 백그라운드에서 렌더링한다. 공통 직렬화된 최종 DOM(440)은 직렬화된 최종 DOM의 현재 버전이고, 클라이언트 브라우저(110)는, 직렬화된 최종 DOM의 이전 버전과 직렬화된 최종 DOM의 현재 버전 간의 다른 차이를 백그라운드에서 수행한다. 따라서, 클라이언트 브라우저(110)는, 직렬화된 최종 DOM 및 실제 웹페이지 둘 모두를 프로세싱하지만 직렬화된 최종 DOM을 먼저 표시하도록 프로세스들을 렌더링할 수 있다.

[0085] 2. 임베디드 JavaScript 접근법

[0086] [0066] 몇몇 실시예들에서, 클라이언트 브라우저(110)는, 임베디드 JavaScript 접근법을 사용할 수 있다. 임베디드 JavaScript 접근법에서, 직렬화된 최종 DOM 콘텐츠는, 웹페이지의 메인 HTML에 임베딩된다. 따라서, 웹페이지는 JavaScript를 또한 포함한다. 클라이언트 브라우저(110)는, 직렬화된 최종 DOM 콘텐츠들을 먼저 디스플레이하고, 그 후, 실제 웹페이지를 디스플레이하는 것으로 전환(transit)한다. 임베디드 JavaScript 접근법은, 직렬화된 최종 DOM이 서버 측으로부터 이용가능한 경우 수행될 수 있다. 서버는, 직렬화된 최종 DOM을 오리지널 HTML에 임베딩함으로써 오리지널 HTML을 수정할 수 있고, 클라이언트 브라우저(110)는, JavaScript를 사용하여, 직렬화된 최종 DOM 및 실제 웹페이지를 상이한 시간들에서 로딩 및 디스플레이할 수 있다.

- [0087] IV. 예시적인 방법
- [0088] [0067] 도 6은 몇몇 실시예들에 따른, 웹페이지를 렌더링하는 방법(600)을 예시하는 흐름도이다. 방법(600)은 제한적인 것으로 의도되지 않으며, 다른 애플리케이션들에서 사용될 수 있다.
- [0089] [0068] 도 6에서, 방법(600)은 블록들(602-608)을 포함한다. 블록(602)에서, 위치에 의해 참조되는 제 1 웹 문서를 프로세싱하는 것에 기초하여 제 1 직렬화된 최종 DOM이 생성되며, 제 1 직렬화된 최종 DOM은 제 1 인-라인된 동적 콘텐츠를 포함한다. 일 예에서, 클라이언트 브라우저(110)는, URL에 의해 참조되는 HTML 페이지(132)를 프로세싱하는 것에 기초하여, 직렬화된 최종 DOM(302)을 생성하며, 직렬화된 최종 DOM(302)은 인-라인된 동적 콘텐츠를 포함한다. 블록(604)에서, 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여 제 2 직렬화된 최종 DOM이 생성되며, 제 2 직렬화된 최종 DOM은 제 2 인-라인된 동적 콘텐츠를 포함한다. 일 예에서, 클라이언트 브라우저(110)는, 위치에 의해 참조되는 제 2 웹 문서를 프로세싱하는 것에 기초하여, 직렬화된 최종 DOM(304)을 생성하며, 직렬화된 최종 DOM(304)은 제 2 인-라인된 동적 콘텐츠를 포함한다.
- [0090] [0069] 블록(606)에서, 제 1 및 제 2 직렬화된 최종 DOM들 간의 차이에 기초하여, 공통 직렬화된 최종 DOM이 생성되며, 공통 직렬화된 최종 DOM은, 제 1 및 제 2 직렬화된 최종 DOM들에 포함된 정적인 동적 콘텐츠를 포함하고, 차이는 트루 동적 콘텐츠이다. 일 예에서, 클라이언트 브라우저(110)는, 직렬화된 최종 DOM들(302 및 304) 간의 차이에 기초하여, 공통 직렬화된 최종 DOM(330)을 생성하며, 공통 직렬화된 최종 DOM(330)은, 직렬화된 최종 DOM들(302 및 304)에 포함된 정적인 동적 콘텐츠를 포함하고, 차이는 트루 동적 콘텐츠이다. 블록(608)에서, 공통 직렬화된 최종 DOM이 디바이스의 디스플레이에 렌더링된다. 일 예에서, 클라이언트 브라우저(110)는, 공통 직렬화된 최종 DOM(330)을 디바이스의 디스플레이에 렌더링한다.
- [0091] [0070] 몇몇 실시예들에서, 블록들(602-608)에 예시된 하나 또는 그 초과 동작들은, 클라이언트 브라우저(110)에 의해 로딩되는 임의의 수의 웹페이지들에 대해 수행될 수 있다. 부가적으로, 위에 논의된 블록들(602-608) 이전에, 그 동안에, 또는 이후에 부가적인 프로세스들이 삽입될 수 있다는 것이 또한 이해된다. 본원에 설명된 방법(600)의 블록들 중 하나 또는 그 초과가, 필요한 경우 생략, 결합, 또는 상이한 시퀀스로 수행될 수 있다는 것이 또한 이해된다.
- [0092] [0071] 도 1-6은 단지 예들이며, 이들이 청구항들의 범위를 과도하게 제한해서는 안 된다. 예를 들어, 간략화를 위해, 본 개시내용은 클라이언트 애플리케이션을 클라이언트 브라우저인 것으로 설명할 수 있지만, 이것은 제한적인 것으로 의도되지 않으며, 다른 클라이언트 애플리케이션들이 본 개시내용의 범위 내에 있다는 것이 이해되어야 한다.
- [0093] [0072] 본 개시내용은, 고도로 동적인 웹사이트들에 대한 고속 렌더링을 달성하면서 스타일 콘텐츠를 회피하기 위한 기술들을 제공한다. 부가적으로, 상이한 양상들로부터의 접근법들이 결합되어 작업 프레임워크를 형성할 수 있다. 일 예에서, 클라이언트 브라우저는, 서버 측 상에서 생성된 직렬화된 최종 DOM을 활용하기 위해 듀얼 렌더링 접근법을 이용할 수 있으며, 모든 트루 동적 콘텐츠들이 제거된다. 다른 예에서, 브라우저는, 어떠한 동적 콘텐츠들도 갖지 않는 직렬화된 최종 DOM들 전부를 캐싱할 수 있다. 부가적으로, 캐시 내의 콘텐츠 및 클라이언트 브라우저에 의해 요청된 콘텐츠 및 서버에 의해 리턴된 콘텐츠를 검토함으로써, 클라이언트 브라우저(110)는, 고속 렌더링을 가능하게 하기 위한 임의의 직렬화된 최종 DOM들 또는 임의의 형태의 리소스들이 전달 및 프로세싱되고 있는지를 식별할 수 있다.
- [0094] V. 예시적인 디바이스
- [0095] [0073] 도 7은, 본원에 개시된 실시예들 중 임의의 실시예를 구현하기에 적절한 예시적인 컴퓨터 시스템(700)의 블록도이다. 다양한 구현들에서, 컴퓨터 시스템(700)은 서버일 수 있다. 다양한 구현들에서, 클라이언트 브라우저(110)는 컴퓨터 시스템(700)에서 실행될 수 있다.
- [0096] [0074] 컴퓨터 시스템(700)은, 입력/출력(I/O)(704) 컴포넌트에 커플링되는 제어 유닛(701)을 포함한다. 제어 유닛(701)은, 하나 또는 그 초과 중량 프로세싱 유닛(CPU)들(770)을 포함할 수 있고, 부가적으로, 플로피 디스크, 플렉서블(flexible) 디스크, 하드 디스크, 자기 테이프, 임의의 다른 자기 매체, CD-ROM, 임의의 다른 광학 매체, 랜덤 액세스 메모리(RAM), 프로그래밍가능 판독-전용 메모리(PROM), 소거가능 ROM(EPROM), FLASH-EPROM, 임의의 다른 메모리 칩 또는 카트리지, 및/또는 프로세서 또는 컴퓨터가 판독하도록 적응되는 임의의 다른 매체를 포함하는 그룹으로부터 각각 선택된 하나 또는 그 초과 저장 디바이스들을 포함할 수 있다. 하나 또는 그 초과 저장 디바이스들은, 컴퓨터 네트워크(도시되지 않음)를 사용하여 컴퓨터 시스템(700)에 커플링되는 하나 또는 그 초과 컴퓨팅 디바이스들 및/또는 컴퓨터 프로그램들(예컨대, 클라이언트들)에게 이용 가능

하게 될 수 있는 저장된 정보를 포함할 수 있다. 컴퓨터 네트워크는, LAN, WAN, 인트라넷, 인터넷, 클라우드, 및/또는 시스템 내의 컴퓨팅 디바이스들 및/또는 컴퓨터 프로그램들을 상호연결하는 것이 가능한 이들 네트워크들의 임의의 결합을 포함하는 임의의 타입의 네트워크일 수 있다.

[0097] [0075] 컴퓨터 시스템(700)은, 컴퓨터 시스템(700)의 다양한 컴포넌트들 사이에서 정보 데이터, 신호들, 및 정보를 통신하기 위한 버스(702) 또는 다른 통신 메커니즘을 포함한다. 컴포넌트들은, 키패드/키보드로부터의 키들을 선택하는 것 또는 하나 또는 그 초과와 버튼들 또는 링크들을 선택하는 것 등과 같은 사용자 동작들을 프로세싱하기 위한 입력/출력(I/O) 컴포넌트들(704)을 포함하고, 대응하는 신호를 버스(702)에 전송한다. I/O 컴포넌트(704)는 또한, 디스플레이(711)와 같은 출력 컴포넌트, 및 커서 제어부(713)(이클테면, 키보드, 키패드, 마우스 등)와 같은 입력 제어부를 포함할 수 있다. 오디오 신호들을 정보 신호들로 변환함으로써 사용자가 정보를 입력하기 위해 보이스를 사용하는 것을 허용하는 오디오 I/O 컴포넌트(705)가 또한 포함될 수 있다. 오디오 I/O 컴포넌트(705)는 사용자가 오디오를 듣는 것을 허용할 수 있다.

[0098] [0076] 트랜시버 또는 네트워크 인터페이스(706)는, 통신 링크(718)를 통해 컴퓨터 시스템(700)과 다른 디바이스들 사이에서 네트워크로 신호들을 송신 및 수신한다. 일 실시예에서, 송신은 무선이지만, 다른 송신 매체들 및 방법들이 또한 적합할 수 있다. 부가적으로, 디스플레이(711)는, 통신 링크(718)를 통해 제어 유닛(701)에 커플링될 수 있다.

[0099] [0077] 마이크로-제어기, DSP(digital signal processor), 또는 다른 프로세싱 컴포넌트일 수 있는 CPU(770)는, 이들 다양한 신호들, 이클테면 컴퓨터 시스템(700)의 디스플레이(711) 상의 디스플레이 또는 통신 링크(718)를 통한 다른 디바이스들로의 송신을 프로세싱한다. CPU(770)는 또한, 쿠키들 또는 IP 어드레스들과 같은 정보의 다른 디바이스들로의 송신을 제어할 수 있다. 컴퓨터 시스템(700)의 컴포넌트들은 또한, 시스템 메모리 컴포넌트(714)(예컨대, RAM), 정적 저장 컴포넌트(716)(예컨대, ROM) 및/또는 컴퓨터 판독가능 매체(717)를 포함한다. 컴퓨터 시스템(700)은, 시스템 메모리 컴포넌트들(714)에 포함된 명령들의 하나 또는 그 초과와 시퀀스들을 실행함으로써 CPU(770) 및 다른 컴포넌트들에 의해 특정 동작들을 수행한다. 로직은, 실행을 위해 명령들을 CPU(770)에 제공하는데 참여하는 임의의 매체를 지칭할 수 있는 컴퓨터 판독가능 매체(717)에 인코딩될 수 있다. 그러한 매체는, 비-휘발성 매체들, 휘발성 매체들, 및 송신 매체들을 포함하는 많은 형태들을 취할 수 있지만 이에 제한되지 않는다. 다양한 구현들에서, 비휘발성 매체들은 광학 또는 자기 디스크들 또는 고체-상태 드라이브들을 포함하고, 휘발성 매체들은 시스템 메모리 컴포넌트(714)와 같은 동적 메모리를 포함하고, 송신 매체들은 버스(702)를 포함하는 와이어들을 비롯하여, 동축 케이블들, 구리 와이어, 및 광섬유들을 포함한다. 일 실시예에서, 로직은 비-일시적인 컴퓨터 판독가능 매체에 인코딩된다. 컴퓨터 판독가능 매체(717)는, CPU(770)에 의해 또는 그와 관련하여 사용되는 명령들을 포함, 저장, 통신, 전파, 또는 전송할 수 있는 임의의 장치일 수 있다. 컴퓨터 판독가능 매체(717)는, 전자, 자기, 광학, 전자기, 적외선, 또는 반도체 디바이스 또는 전파 매체, 또는 임의의 다른 메모리 칩 또는 카트리지, 또는 컴퓨터가 판독하도록 적응되는 임의의 다른 매체일 수 있다. 일 예에서, 송신 매체들은, 라디오파, 광학 및 적외선 데이터 통신들 동안에 생성되는 것들과 같이 음향 또는 광파들의 형태를 취할 수 있다.

[0100] [0078] 본 개시내용의 다양한 실시예들에서, 본 개시내용을 실시하기 위한 명령 시퀀스들(예컨대, 방법(600))의 실행은 컴퓨터 시스템(700)에 의해 수행될 수 있다. 본 개시내용의 다양한 다른 실시예들에서, 통신 링크(718)에 의해 네트워크(예컨대, 이클테면, LAN, WLAN, PTSN, 및/또는 원격통신들, 모바일, 및 셀룰러 폰 네트워크들을 포함하는 다양한 다른 유선 또는 무선 네트워크들)에 커플링된 복수의 컴퓨터 시스템들(700)은, 서로 합동하여 본 개시내용을 실시하기 위한 명령 시퀀스들을 수행할 수 있다.

[0101] [0079] 적용 가능한 경우, 본 개시내용에 의해 제공된 다양한 실시예들은, 하드웨어, 소프트웨어, 또는 하드웨어 및 소프트웨어의 결합들을 사용하여 구현될 수 있다. 또한 적용 가능한 경우, 본원에 기재된 다양한 하드웨어 컴포넌트들 및/또는 소프트웨어 컴포넌트들은, 본 개시내용의 사상으로부터 벗어나지 않으면서 소프트웨어, 하드웨어, 및/또는 둘 모두를 포함하는 합성 컴포넌트들로 결합될 수 있다. 적용 가능한 경우, 본원에 기재된 다양한 하드웨어 컴포넌트들 및/또는 소프트웨어 컴포넌트들은, 본 개시내용의 사상으로부터 벗어나지 않으면서 소프트웨어, 하드웨어, 및/또는 둘 모두를 포함하는 서브-컴포넌트들로 분리될 수 있다. 부가하여, 적용 가능한 경우, 소프트웨어 컴포넌트들이 하드웨어 컴포넌트들로서 구현될 수 있고, 하드웨어 컴포넌트들이 소프트웨어 컴포넌트들로 구현될 수 있다는 것이 고려된다.

[0102] [0080] 본 개시내용에 따른 애플리케이션 소프트웨어는, 하나 또는 그 초과와 컴퓨터 판독가능 매체들 상에 저장될 수 있다. 본원에 식별된 애플리케이션 소프트웨어가 하나 또는 그 초과와 범용 또는 특수 목적 컴퓨터들

및/또는 컴퓨터 시스템들을 사용하여 구현되고, 네트워킹되고, 그리고/또는 달리 구현될 수 있음이 또한 고려된다. 적용 가능한 경우, 본원에 설명된 다양한 블록들의 순서는 변경되고, 합성 블록들로 결합되고, 그리고/또는 서브-블록들로 분리되어 본원에 설명된 특징들을 제공할 수 있다.

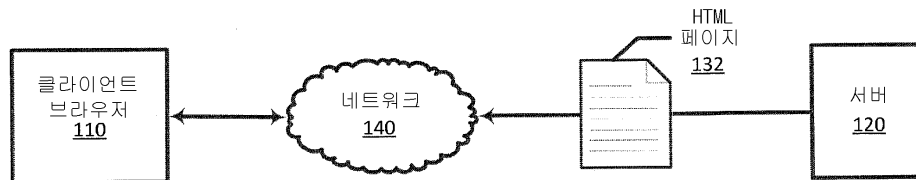
[0103]

[0081] 전술한 개시내용은, 개시된 특정 사용 분야들 또는 정확한 형태들로 본 개시내용을 제한하도록 의도되지 않는다. 이로써, 본원에 명시적으로 설명되든지 또는 암시되든지 간에, 본 개시내용에 대한 다양한 대안적인 실시예들 및/또는 수정들이 본 개시내용을 고려하여 가능하다는 것이 고려된다. 본 개시내용의 범위를 벗어나지 않으면서 형태 및 세부사항에서 변경들이 이루어질 수 있다. 따라서, 본 개시내용은 청구항들에 의해서만 제한된다.

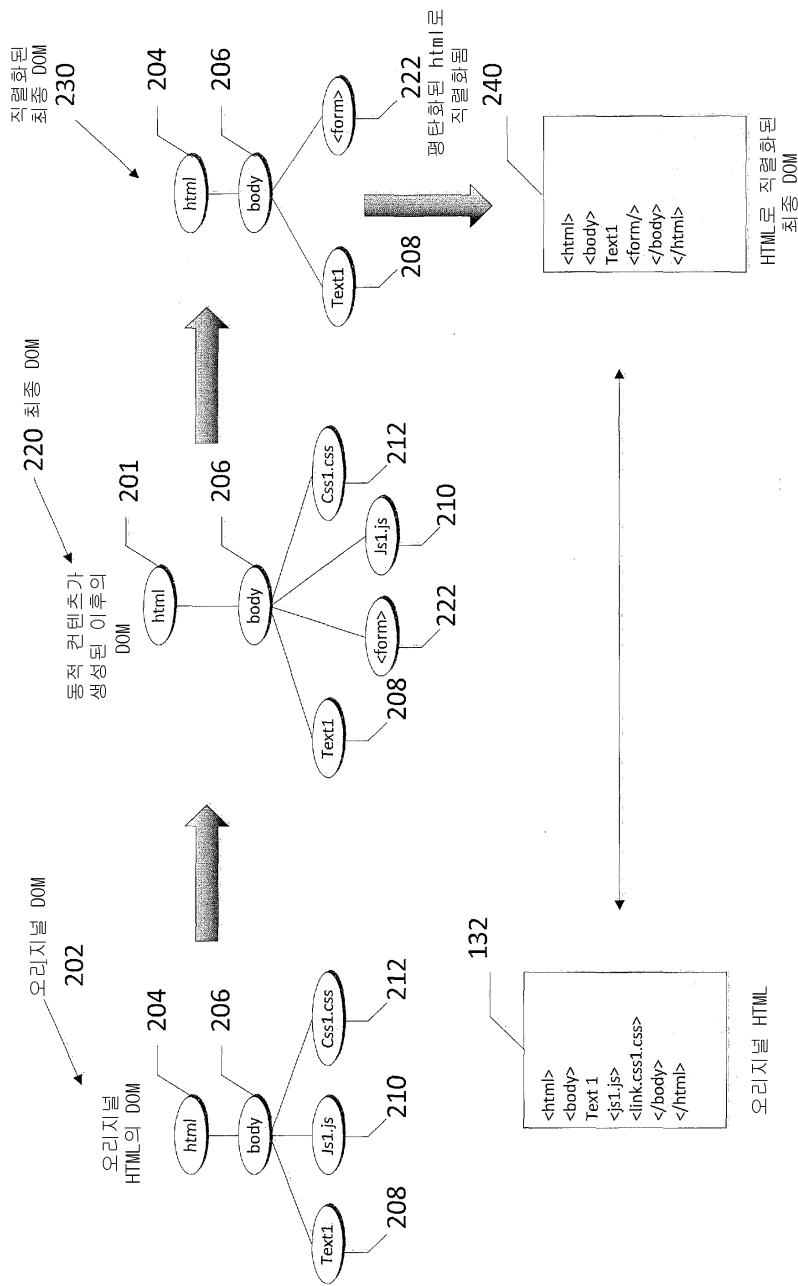
도면

도면1

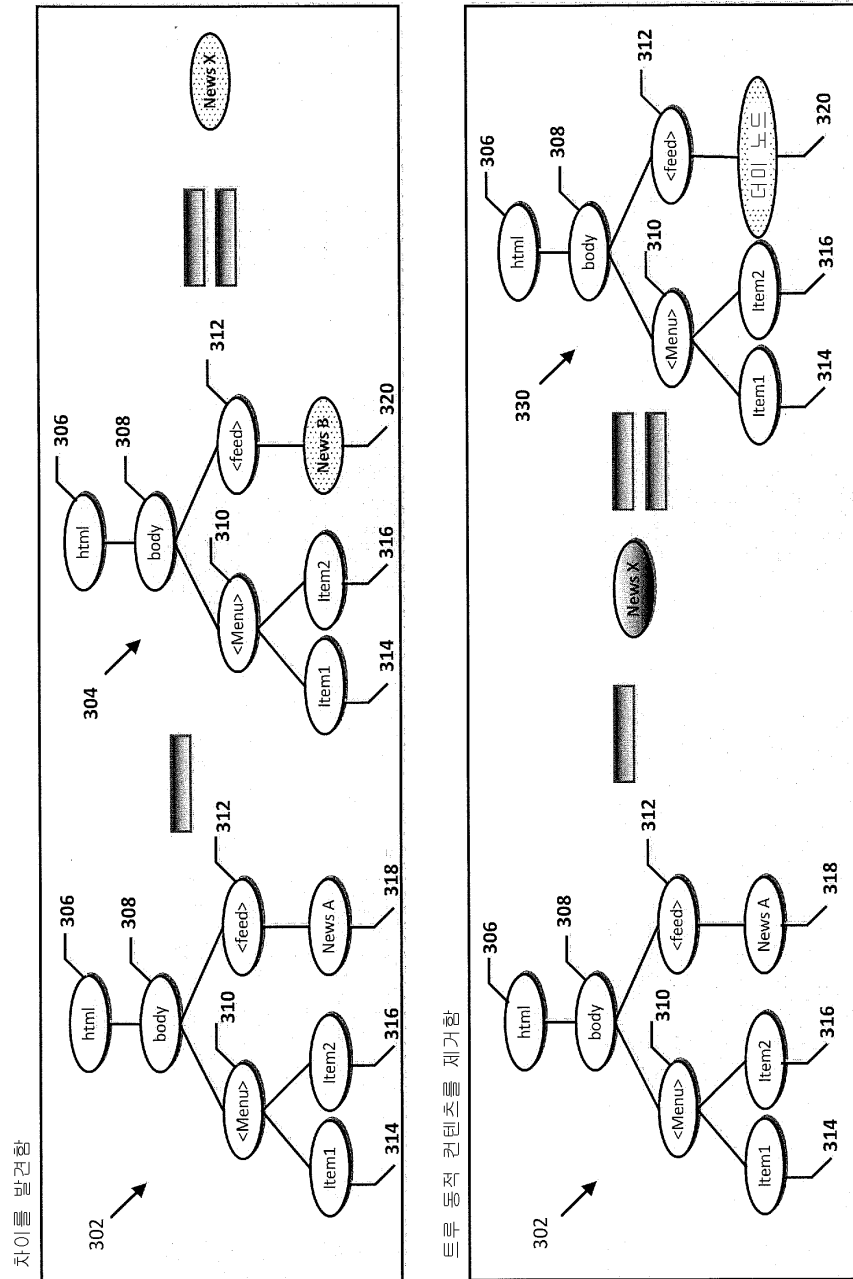
100 ↘



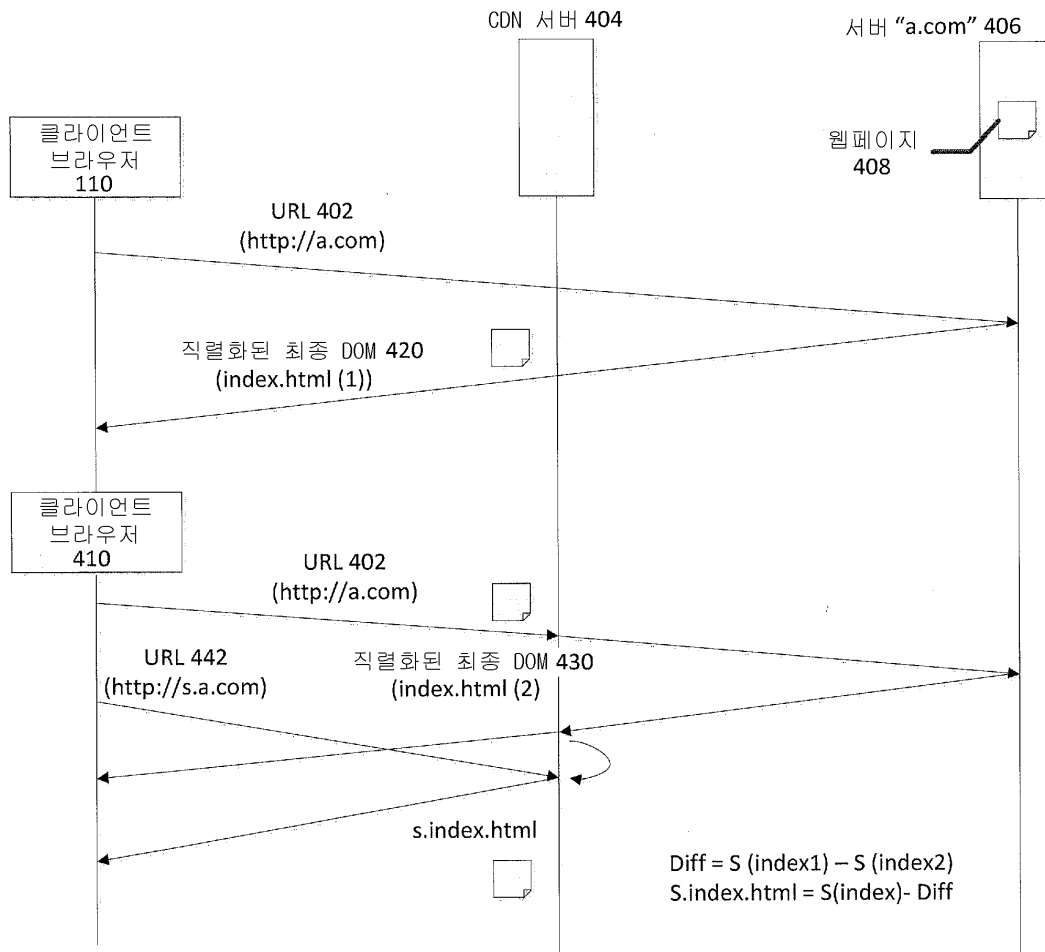
도면2



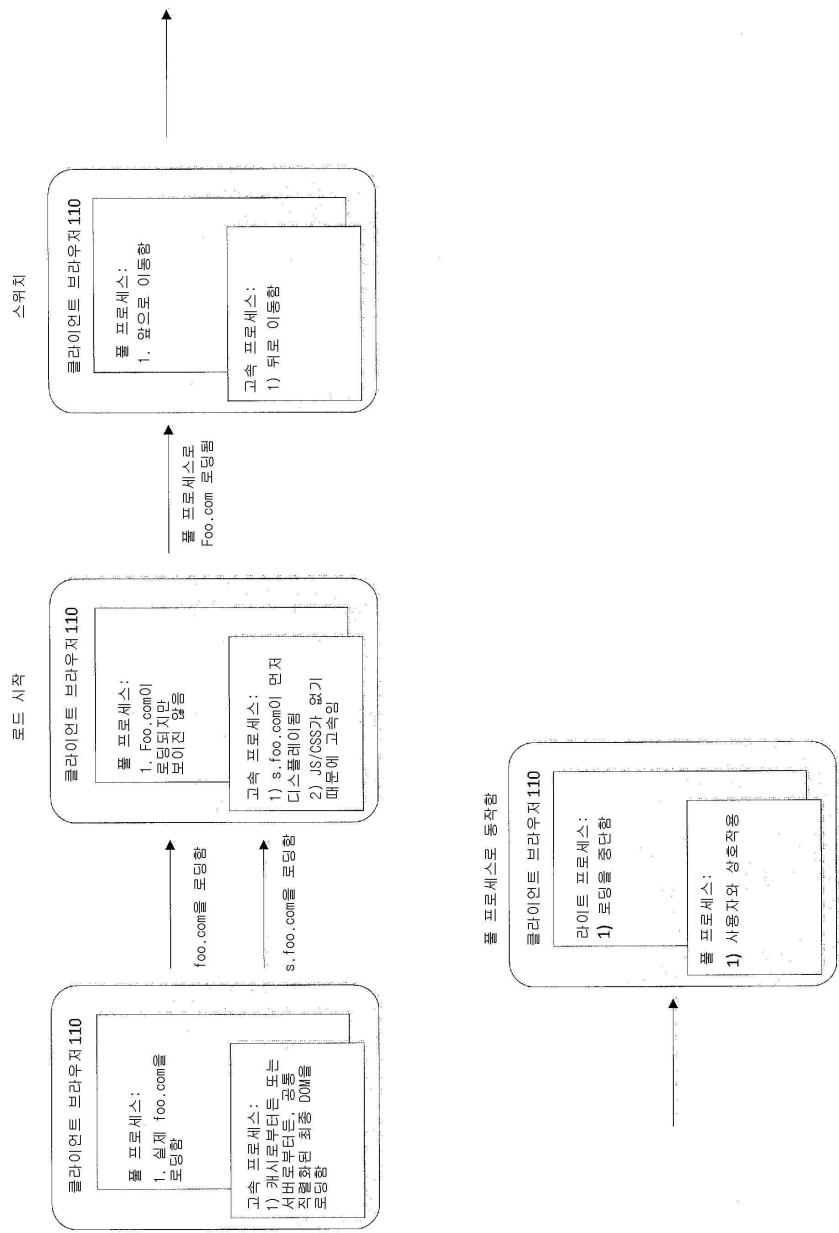
도면3



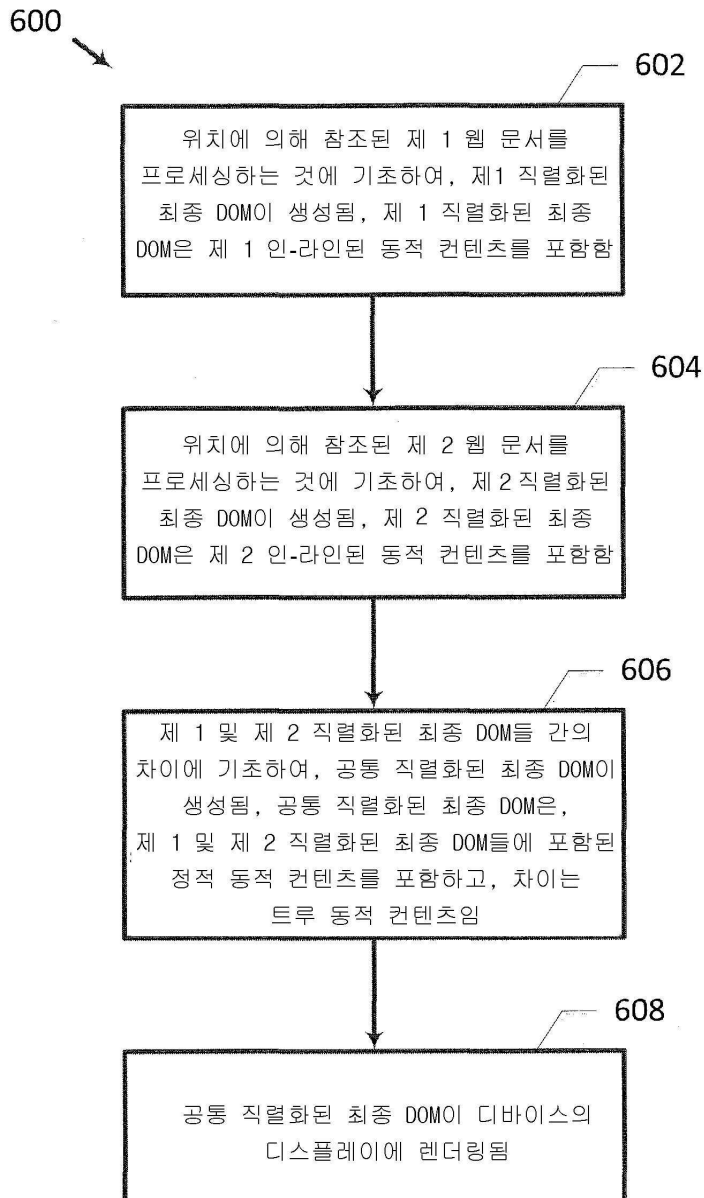
도면4



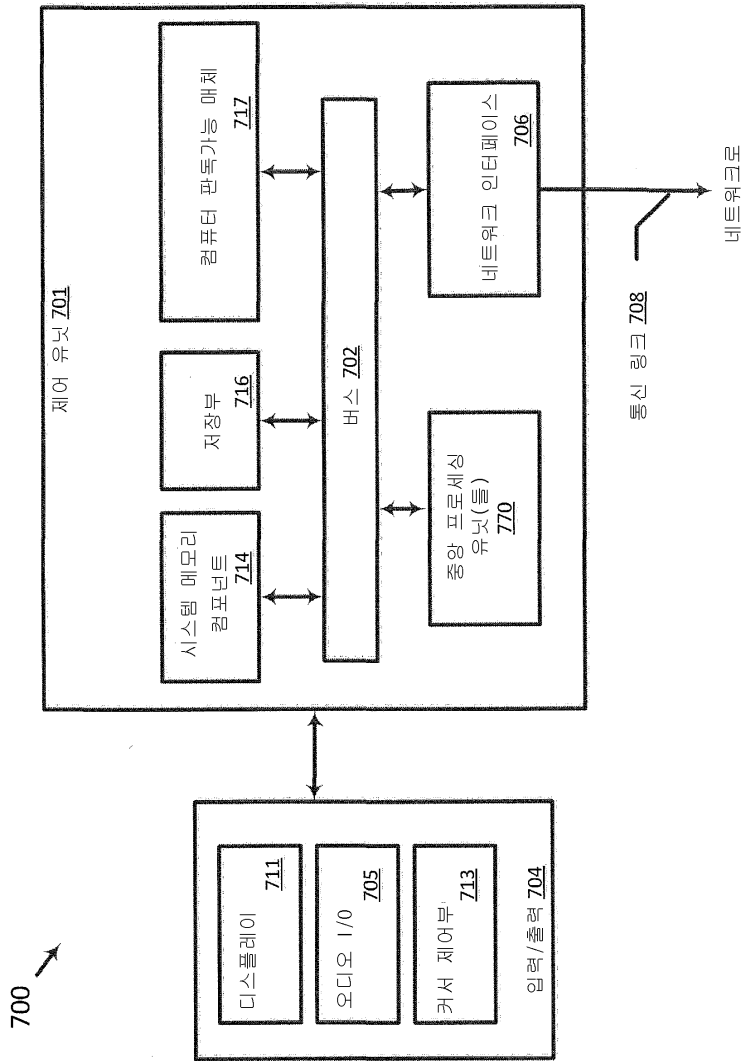
도면5



도면6



도면7



- 【심사관 직권보정사항】
- 【직권보정 1】
- 【보정항목】 청구범위
- 【보정세부항목】 청구항 4항
- 【변경전】
 - 제 1 최종 DOM
- 【변경후】
 - 제 1 직렬화된 최종 DOM