



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0086160
(43) 공개일자 2022년06월23일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>G06F 16/957</i> (2019.01) <i>G06F 16/901</i> (2019.01)
 <i>G06F 16/958</i> (2019.01) <i>H04L 65/40</i> (2022.01)</p> <p>(52) CPC특허분류
 <i>G06F 16/9574</i> (2019.01)
 <i>G06F 16/9024</i> (2019.01)</p> <p>(21) 출원번호 10-2020-0176352
 (22) 출원일자 2020년12월16일
 심사청구일자 2020년12월16일</p> | <p>(71) 출원인
 고려대학교 산학협력단
 서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)</p> <p>(72) 발명자
 이원준
 서울특별시 강남구 선릉로 221, 104동 2005호(도곡동, 도곡렉슬아파트)</p> <p>정예지
 서울특별시 용산구 한강대로43길 8, 102동 2703호(한강로2가, 벽산메가트리움)</p> <p>(74) 대리인
 윤귀상</p> |
|---|---|

전체 청구항 수 : 총 15 항

(54) 발명의 명칭 웹 리소스 의존성 기반 리소스 우선순위화 방법, 이를 수행하기 위한 기록 매체 및 장치

(57) 요약

웹 리소스 의존성 기반 리소스 우선순위화 방법은, 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 단계; 의존성 그래프로부터 하나의 노드가 하나의 리소스에 대응하는 오브젝트 로딩 액티비티 간 리소스 의존성 그래프를 추출하는 단계; 상기 리소스 의존성 그래프와 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기와 리소스가 중요 경로(critical path)에 속하는지 여부로 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정하는 단계; 및 웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드하는 단계를 포함한다. 이에 따라, 웹 페이지 로딩 액티비티 간 의존성을 고려한 리소스 우선순위화에 따라 빠른 렌더링을 제공할 수 있다.

대표도 - 도7

```

G = (V, E), G is a directed acyclic graph
1: function TopSort(G)
2:   stack ← [...]
3:   Initialize visited[u] = False for all vertices u
4:   for each vertex v in V in ascending order of Scores(u) do
5:     if visited[v] = False then
6:       RecursiveTopSort(v, visited, stack)
7:     end if
8:   end for
9:   Return stack
10: end function

11: function RecursiveTopSort(u, visited, stack)
12:   visited[u] = True
13:   for each adjacent vertex adjv of u in ascending order of Scores(u) do
14:     if visited[adjv] = False then
15:       RecursiveTopSort(adjv, visited, stack)
16:     end if
17:   insert v to stack
18: end for
19: end function

20: function Scores(u)
21:   subgraphsize ← number of vertices in subgraph of G when u is regarded
   as the root node of the subgraph
22:   if u is in critical path then
23:     Criticalpath ← 1
24:   else
25:     Criticalpath ← 0
26:   end if
27:   weight ← 10
28:   relativestarttime ← relative starting time of loading that corresponds to u
29:   score = criticalpath x 10 + subgraphsize - relativestarttime
30:   Return score
31: end function

```

(52) CPC특허분류

G06F 16/958 (2019.01)

H04L 67/02 (2022.05)

H04L 67/61 (2022.05)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711104878
과제번호	2017M3C4A7083676
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	차세대정보·컴퓨팅기술개발(R&D)
연구과제명	Web 응용을 위한 Seamless 자율 전송 네트워킹 제어 프레임워크
기 여 율	1/1
과제수행기관명	고려대학교
연구기간	2017.11.01 ~ 2020.12.31

명세서

청구범위

청구항 1

브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 단계;

의존성 그래프로부터 하나의 노드가 하나의 리소스에 대응하는 오브젝트 로딩 액티비티 간 리소스 의존성 그래프를 추출하는 단계;

상기 리소스 의존성 그래프와 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기와 리소스가 중요 경로(critical path)에 속하는지 여부로 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정하는 단계; 및

웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드하는 단계;를 포함하는, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 2

제1항에 있어서,

브라우저에 의해 상기 리소스의 우선순위 순서대로 리소스를 요청하는 단계;를 더 포함하는, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 3

제1항에 있어서, 상기 리소스 의존성 그래프를 추출하는 단계는,

오브젝트 로딩 액티비티는 모두 네트워킹인, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 4

제1항에 있어서,

상기 중요 경로(critical path)는 페이지 로딩 중 가장 시간이 오래 걸리는 일련의 액티비티 순서인, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 5

제1항에 있어서,

상기 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)인, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 6

제5항에 있어서, 상기 리소스의 우선순위를 결정하는 단계는,

토폴로지컬 정렬(topological sorting) 알고리즘을 사용하여 상기 유향 비순환 그래프로부터 하나의 순서를 생성하는, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 7

제6항에 있어서, 상기 리소스의 우선순위를 결정하는 단계는,

상기 루트 노드(root node)부터 시작하여 인접한 모든 노드에 대해 재귀적으로 토폴로지컬 정렬을 호출하는 단계;

노드에 인접한 모든 노드가 스택(stack)에 있는 경우, 스택에 해당 노드를 삽입하는 단계; 및

스택은 그래프의 의존성을 어긋나지 않는 하나의 순서를 생성하는 단계;를 포함하는, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 8

제1항에 있어서, 상기 리소스의 우선순위를 결정하는 단계는,

리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기를 나타내는 subgraphsize 함수, 리소스의 중요 경로 여부를 나타내는 criticalpath 함수 및 각 리소스가 요청된 시간을 나타내는 relativestarttime 함수를 사용하여 각 리소스의 중요도 점수를 부여하는, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 9

제8항에 있어서,

상기 subgraphsize 함수와 criticalpath 함수는 정수이고 relativestarttime은 0과 1사이의 값인, 웹 리소스 의존성 기반 리소스 우선순위화 방법.

청구항 10

제1항 내지 제9항 중 어느 하나의 항에 따른 상기 웹 리소스 의존성 기반 리소스 우선순위화 방법을 수행하기 위한 컴퓨터 프로그램이 기록된 컴퓨터로 판독 가능한 저장 매체.

청구항 11

브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 정보 추출부;

의존성 그래프로부터 하나의 노드가 하나의 리소스에 대응하는 오브젝트 로딩 액티비티 간 리소스 의존성 그래프를 추출하는 의존성 그래프 추출부;

상기 리소스 의존성 그래프와 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기와 리소스가 중요 경로(critical path)에 속하는지 여부로 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정하는 리소스 우선순위화부; 및

웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드하는 리소스 프리로드부;를 포함하는, 웹 리소스 의존성 기반 리소스 우선순위화 장치.

청구항 12

제11항에 있어서,

상기 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)인, 웹 리소스 의존성 기반 리소스 우선순위화 장치.

청구항 13

제12항에 있어서, 상기 리소스 우선순위화부는,

토폴로지컬 정렬(topological sorting) 알고리즘을 사용하여 상기 유형 비순환 그래프로부터 하나의 순서를 생성하는, 웹 리소스 의존성 기반 리소스 우선순위화 장치.

청구항 14

제13항에 있어서, 상기 리소스 우선순위화부는,

상기 루트 노드(root node)부터 시작하여 인접한 모든 노드에 대해 재귀적으로 토폴로지컬 정렬을 호출하고, 노드에 인접한 모든 노드가 스택(stack)에 있는 경우, 스택에 해당 노드를 삽입하는, 웹 리소스 의존성 기반 리소스 우선순위화 장치.

청구항 15

제11항에 있어서, 상기 리소스 우선순위화부는,

리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기를 나타내는 subgraphsize 함수, 리소스의 중요 경로 여부를 나타내는 criticalpath 함수 및 각 리소스가 요청된 시간을 나타내는 relativestarttime 함수를 사용하여 각 리소스의 중요도 점수를 부여하는, 웹 리소스 의존성 기반 리소스 우선순위화 장치.

발명의 설명

기술 분야

[0001] 본 발명은 웹 리소스 의존성 기반 리소스 우선순위화 방법, 이를 수행하기 위한 기록 매체 및 장치에 관한 것으로서, 더욱 상세하게는 각 웹 리소스를 어떠한 순서로 요청 및 응답해야 하는지에 대한 결정을 하는 리소스 우선순위화(resource prioritization) 문제에 있어서 리소스 의존성(dependency) 정보를 활용하여 리소스 우선순위를 수행하는 기술에 관한 것이다.

배경 기술

[0003] 웹 페이지 로딩(Web page loading) 과정은 매우 복잡하다. 웹 페이지 로딩은 오브젝트 로딩(object loading), HTML 파싱(hypertext markup language parsing), JavaScript/CSS 실행 (evaluation), 렌더링(rendering) 액티비티들의 복잡한 상호과정으로 이루어져 있다.

[0004] 오브젝트 로딩은 오브젝트 혹은 사용자가 요청한 오브젝트를 가져온다. 오브젝트는 캐시(cache)나 로컬 저장소(local storage)에서 얻거나 네트워킹(networking)을 통해 서버로부터 전달받을 수 있다. HTML 파싱 액티비티는 HTML 페이지를 DOM(document object model)으로 변환시킨다. JavaScript/CSS 실행은 웹 오브젝트 중 JavaScript 혹은 CSS 오브젝트를 실행하여 DOM을 변경한다. 렌더링 액티비티는 DOM을 기반으로 웹 페이지를 시각적으로 표현한다.

[0005] 페이지 로딩 액티비티들은 서로가 서로에 영향을 받는다. 예를 들어, 한 액티비티는 다른 액티비티가 끝난 후에 실행될 수 있다. 또는, 한 액티비티는 다른 액티비티의 결과물 중 일부를 입력으로 필요로 한다. 이러한 관계성을 웹 페이지 로딩 액티비티 간 의존성이라 표현할 수 있다. 웹 페이지 로딩을 가속화하기 위한 여러 웹 기술들이 도입되었다.

[0006] 리소스 우선순위화는 웹 페이지 로딩 가속화를 위한 방법론 중 하나이다. 웹 페이지 로딩 시 리소스의 요청은 주로 브라우저의 리소스 발견 순서에 따라 결정된다. 기존에는 리소스를 발견하면 즉시 리소스 요청을 보냈다.

하지만, 리소스들 간 중요도가 다르다는 것이 인식되고 리소스 우선순위화 기법이 도입되었다. 리소스 요청 혹은 응답의 순서를 그 중요도에 따라 변경하는 것을 리소스 우선순위화라 칭한다.

[0007] 이는 일반적으로 브라우저 내에서 이루어지지만 다른 방식으로도 가능하다. HTTP/2의 PRIORITY 프레임은 서버와 클라이언트(client)가 스트림(stream)의 중요도와 스트림 간 의존성을 설정할 수 있게 한다. 리소스는 하나의 양방향 스트림과 대응하기에 스트림 우선순위화는 프로토콜을 이용한 리소스 우선순위화이다.

[0008] 그러나, 기존의 리소스 우선순위화의 결정 기준은 각 리소스의 특성이었다. 이는 리소스의 종류, 리소스가 'async' 속성을 포함하는지 등 각 단일 리소스에 관한 정보를 의미한다. 하지만, 웹 페이지 로딩의 복잡성을 고려한 리소스 우선순위화가 필요한 실정이다.

선행기술문헌

특허문헌

- [0010] (특허문헌 0001) KR 10-2008-0033403 A
(특허문헌 0002) KR 10-2151457 B1
(특허문헌 0003) KR 10-1711528 B1
(특허문헌 0004) US 10,389,592 B2

비특허문헌

- [0011] (비특허문헌 0001) M. Wijnants and R. Marx and P. Quax and W. Lamotte, "HTTP/2 Prioritization and its Impact on Web Performance," in Proc. of WWW, April 2018.
(비특허문헌 0002) R. Marx, T. D. Decker, P. Quax and W. Lamotte, "Of the Utmost Importance: Resource Prioritization in HTTP/3 over QUIC," in Proc. of WEBIST, September 2019.

발명의 내용

해결하려는 과제

- [0012] 이에, 본 발명의 기술적 과제는 이러한 점에서 착안된 것으로 본 발명의 목적은 웹 리소스 의존성 기반 리소스 우선순위화 방법을 제공하는 것이다.
- [0013] 본 발명의 다른 목적은 상기 웹 리소스 의존성 기반 리소스 우선순위화 방법을 수행하기 위한 컴퓨터 프로그램이 기록된 기록 매체를 제공하는 것이다.
- [0014] 본 발명의 또 다른 목적은 상기 웹 리소스 의존성 기반 리소스 우선순위화 방법을 수행하기 위한 장치를 제공하는 것이다.

과제의 해결 수단

- [0016] 상기한 본 발명의 목적을 실현하기 위한 일 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 방법은, 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 단계; 의존성 그래프로부터 하나의 노드가 하나의 리소스에 대응하는 오브젝트 로딩 액티비티 간 리소스 의존성 그래프를 추출하는 단계; 상기 리소스 의존성 그래프와 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기와 리소스가 중요 경로(critical path)에 속하는지 여부로 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정하는 단계; 및 웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드하는 단계;를 포함한다.

- [0017] 본 발명의 실시예에서, 상기 웹 리소스 의존성 기반 리소스 우선순위화 방법은, 브라우저에 의해 상기 리소스의 우선순위 순서대로 리소스를 요청하는 단계;를 더 포함할 수 있다.
- [0018] 본 발명의 실시예에서, 상기 리소스 의존성 그래프를 추출하는 단계는, 오브젝트 로딩 액티비티는 모두 네트워킹될 수 있다.
- [0019] 본 발명의 실시예에서, 상기 중요 경로(critical path)는 페이지 로딩 중 가장 시간이 오래 걸리는 일련의 액티비티 순서일 수 있다.
- [0020] 본 발명의 실시예에서, 상기 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)일 수 있다.
- [0021] 본 발명의 실시예에서, 상기 리소스의 우선순위를 결정하는 단계는, 토폴로지컬 정렬(topological sorting) 알고리즘을 사용하여 상기 유향 비순환 그래프로부터 하나의 순서를 생성할 수 있다.
- [0022] 본 발명의 실시예에서, 상기 리소스의 우선순위를 결정하는 단계는, 상기 루트 노드(root node)부터 시작하여 인접한 모든 노드에 대해 재귀적으로 토폴로지컬 정렬을 호출하는 단계; 노드에 인접한 모든 노드가 스택(stack)에 있는 경우, 스택에 해당 노드를 삽입하는 단계; 및 스택은 그래프의 의존성을 어긋나지 않는 하나의 순서를 생성하는 단계;를 포함할 수 있다.
- [0023] 본 발명의 실시예에서, 상기 리소스의 우선순위를 결정하는 단계는, 리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기를 나타내는 subgraphsize 함수, 리소스의 중요 경로 여부를 나타내는 criticalpath 함수 및 각 리소스가 요청된 시간을 나타내는 relativestarttime 함수를 사용하여 각 리소스의 중요도 점수를 부여할 수 있다.
- [0024] 본 발명의 실시예에서, 상기 subgraphsize 함수와 criticalpath 함수는 정수이고 relativestarttime은 0과 1사이의 값일 수 있다.
- [0025] 상기한 본 발명의 다른 목적을 실현하기 위한 일 실시예에 따른 컴퓨터로 판독 가능한 저장 매체에는, 상기 웹 리소스 의존성 기반 리소스 우선순위화 방법을 수행하기 위한 컴퓨터 프로그램이 기록되어 있다.
- [0026] 상기한 본 발명의 또 다른 목적을 실현하기 위한 일 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 장치는, 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 정보 추출부; 의존성 그래프로부터 하나의 노드가 하나의 리소스에 대응하는 오브젝트 로딩 액티비티 간 리소스 의존성 그래프를 추출하는 의존성 그래프 추출부; 상기 리소스 의존성 그래프와 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기와 리소스가 중요 경로(critical path)에 속하는지 여부로 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정하는 리소스 우선순위화부; 및 웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드하는 리소스 프리로드부;를 포함한다.
- [0027] 본 발명의 실시예에서, 상기 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)일 수 있다.
- [0028] 본 발명의 실시예에서, 상기 리소스 우선순위화부는, 토폴로지컬 정렬(topological sorting) 알고리즘을 사용하여 상기 유향 비순환 그래프로부터 하나의 순서를 생성할 수 있다.
- [0029] 본 발명의 실시예에서, 상기 리소스 우선순위화부는, 상기 루트 노드(root node)부터 시작하여 인접한 모든 노드에 대해 재귀적으로 토폴로지컬 정렬을 호출하고, 노드에 인접한 모든 노드가 스택(stack)에 있는 경우, 스택에 해당 노드를 삽입할 수 있다.
- [0030] 본 발명의 실시예에서, 상기 리소스 우선순위화부는, 리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기를 나타내는 subgraphsize 함수, 리소스의 중요 경로 여부를 나타내는 criticalpath 함수 및 각 리소스가 요청된 시간을 나타내는 relativestarttime 함수를 사용하여 각 리소스의 중요도 점수를 부여할 수 있다.

발명의 효과

- [0032] 이와 같은 웹 리소스 의존성 기반 리소스 우선순위화 방법에 따르면, 웹 페이지 로딩 액티비티 간 의존성을 고

려하므로, 빠른 렌더링을 유도하게되어 결론적으로 좋은 사용자 경험을 제공할 수 있다.

도면의 간단한 설명

- [0034] 도 1은 본 발명의 일 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 장치의 블록도이다.
- 도 2는 웹 페이지의 예시를 보여주는 도면이다.
- 도 3은 도 2의 웹 페이지에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출한 결과를 보여주는 도면이다.
- 도 4는 도 3의 웹 페이지 로딩 액티비티 간 의존성 그래프 중 네트워킹 액티비티 간의 의존성 그래프를 추출한 결과를 보여주는 도면이다.
- 도 5는 WProfX를 통해 실제로 얻은 google.com 페이지 로딩 액티비티 간 의존성 그래프이다.
- 도 6은 도 5의 네트워킹 액티비티 간 의존성 그래프를 추출한 결과를 보여주는 도면이다.
- 도 7은 본 발명에 따른 수정된 리소스의 중요도 점수를 고려한 깊이 우선 탐색 토폴로지컬 정렬 알고리즘의 의사코드이다.
- 도 8은 본 발명에 따라 변경한 초기 HTML파일 indexB.html을 보여주는 도면이다.
- 도 9는 본 발명에서 제안한 방식으로 리소스 우선순위를 다른 리소스 우선순위화 방법과 그 성능을 비교한 그래프이다.
- 도 10은 본 발명의 일 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0035] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예에 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.
- [0036] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.
- [0037] 도 1은 본 발명의 일 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 장치의 블록도이다.
- [0038] 본 발명에 따른 웹 리소스 의존성 기반 리소스 우선순위화 장치(10, 이하 장치)는 웹 페이지 로딩 시 리소스를 어떤 순서로 요청 혹은 응답할 것인가에 대한 문제인 리소스 우선순위를 해결하기 위해 기존의 리소스 우선순위화 방법과는 달리 리소스 의존성(dependency) 정보를 활용하여 리소스 우선순위를 수행하는 새로운 방안을 제안한다.
- [0039] 도 1을 참조하면, 본 발명에 따른 장치(10)는 정보 추출부(110), 의존성 그래프 추출부(130), 리소스 우선순위화부(150) 및 리소스 프리로드부(170)를 포함한다.
- [0040] 본 발명의 상기 장치(10) 웹 리소스 의존성 기반 리소스 우선순위를 수행하기 위한 소프트웨어(애플리케이션)가 설치되어 실행될 수 있으며, 상기 정보 추출부(110), 상기 의존성 그래프 추출부(130), 상기 리소스 우선순위화부(150) 및 상기 리소스 프리로드부(170)의 구성은 상기 장치(10)에서 실행되는 상기 웹 리소스 의존성 기반 리소스 우선순위를 수행하기 위한 소프트웨어에 의해 제어될 수 있다.
- [0041] 상기 장치(10)는 별도의 단말이거나 또는 단말의 일부 모듈일 수 있다. 또한, 상기 정보 추출부(110), 상기 의존성 그래프 추출부(130), 상기 리소스 우선순위화부(150) 및 상기 리소스 프리로드부(170)의 구성은 통합 모듈로 형성되거나, 하나 이상의 모듈로 이루어 질 수 있다. 그러나, 이와 반대로 각 구성은 별도의 모듈로 이루어

질 수도 있다.

- [0042] 상기 장치(10)는 이동성을 갖거나 고정될 수 있다. 상기 장치(10)는, 서버(server) 또는 엔진(engine) 형태일 수 있으며, 디바이스(device), 기구(apparatus), 단말(terminal), UE(user equipment), MS(mobile station), 무선기기(wireless device), 휴대기기(handheld device) 등 다른 용어로 불릴 수 있다.
- [0043] 상기 장치(10)는 운영체제(Operation System; OS), 즉 시스템을 기반으로 다양한 소프트웨어를 실행하거나 제작할 수 있다. 상기 운영체제는 소프트웨어가 장치의 하드웨어를 사용할 수 있도록 하기 위한 시스템 프로그램으로서, 안드로이드 OS, iOS, 윈도우 모바일 OS, 바다 OS, 심비안 OS, 블랙베리 OS 등 모바일 컴퓨터 운영체제 및 윈도우 계열, 리눅스 계열, 유닉스 계열, MAC, AIX, HP-UX 등 컴퓨터 운영체제를 모두 포함할 수 있다.
- [0044] 상기 정보 추출부(110)는 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출한다.
- [0045] 본 발명을 수행하기 위해서는 우선으로 웹 페이지 로딩 시 발생하는 액티비티 간의 의존성 그래프가 필요하다. WProfX는 Google Chrome 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 기존의 도구이다.
- [0046] 캐시와 로컬 저장소가 비어 있는 상태에서 WProfX를 사용하여 웹 페이지 로딩 액티비티의 의존성 그래프를 얻는다. 캐시나 로컬 저장소에서 리소스를 얻어 오는 오브젝트 로딩을 제외하였기에 웹 페이지 로딩 액티비티들 중 오브젝트 로딩 액티비티는 모두 네트워킹이다.
- [0047] 상기 의존성 그래프 추출부(130)는 의존성 그래프에서 오브젝트 로딩(네트워킹) 액티비티 간 의존성만 추출한다. 추출한 그래프에서 한 노드는 하나의 리소스에 대응하기에 이를 리소스 의존성 그래프라 칭한다. 리소스 의존성은 추후의 리소스 우선순위화에서 한 리소스의 로딩이 어떤 리소스의 로딩보다 선행되어야 하는지에 대한 명확한 기준을 제시한다.
- [0048] 도 2는 예시 웹 페이지이다. 예시 웹 페이지의 초기 HTML 문서는 indexA.html이고, indexA.html은 1.js, 2.js를 <script> 태그를 통해 참조한다. 1.js, 2.js는 각각 1.img, 2.img를 참조한다.
- [0049] 해당 예시 웹 페이지에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출한 결과가 도 3이다. indexA.html의 앞 부분 x의 렌더링 관련 액티비티 이후를 살펴보면, indexA.html의 a 부분의 파싱 액티비티가 선행되어야 1.js가 참조된다는 것을 알고 1.js 오브젝트 로딩 액티비티가 시작될 수 있다. 1.js의 실행 액티비티가 선행되어야 1.js가 1.img를 참조한다는 것을 알고 1.img의 오브젝트 로딩이 시작될 수 있다.
- [0050] indexA.html의 a부분의 파싱 이후에 indexA.html의 b 부분의 파싱이 실행된다. indexA.html의 b 부분의 파싱 액티비티가 선행되어야 2.js가 참조된다는 것을 알고 2.js 오브젝트 로딩 액티비티가 시작될 수 있다. 2.js의 실행 액티비티가 선행되어야 2.js가 2.img를 참조한다는 것을 알고 2.img의 오브젝트 로딩이 시작될 수 있다.
- [0051] 해당 예시 페이지의 웹 페이지 로딩 액티비티 간 의존성 그래프 중 네트워킹 액티비티 간의 의존성 그래프를 추출한 결과가 도 4이다.
- [0052] 도 5는 WProfX를 통해 실제로 얻은 google.com 페이지 로딩 액티비티 간 의존성 그래프이다. 그래프에서 노란색 노드는 네트워킹 액티비티이고, 흰색 노드는 다른 종류의 페이지 로딩 액티비티들이다. 빨간 선의 노드들은 중요 경로(critical path)에 해당하는 액티비티들을 의미한다. 중요 경로는 페이지 로딩 중 가장 시간이 오래 걸리는 일련의 액티비티 순서를 의미한다.
- [0053] 도 5에서 네트워킹 액티비티 간 의존성 그래프를 추출한 결과는 도 6에서 보이는 google.com의 리소스 의존성 그래프이다.
- [0054] 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)이다. 토폴로지컬 정렬(topological sorting) 알고리즘을 사용하면 DAG에서 그 방향을 어긋나지 않는 하나의 순서를 만들 수 있다.
- [0055] 상기 리소스 우선순위화부(150)는 상기 리소스 의존성 그래프 자체로 리소스의 우선순위를 결정하거나 또는 상기 리소스 의존성 그래프 및 상기 리소스 의존성 그래프에서 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정한다.
- [0056] 다시 말해, 일 실시예에서는 상기 리소스 의존성 그래프 자체로 리소스의 우선순위를 결정할 수 있고, 다른 실시예에서는 리소스 의존성 그래프 자체와 리소스 중요도 점수를 모두 고려하여 우선순위를 결정할 수 있다.
- [0057] 상기 리소스 중요도 점수는 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로

설정하여 하위 그래프의 크기 및 리소스가 중요 경로(critical path)에 속하는지 여부에 따라 결정될 수 있다.

- [0058] 도 7은 수정된 리소스의 중요도 점수를 고려한 깊이 우선 탐색 토폴로지컬 정렬 알고리즘의 의사코드이다.
- [0059] 토폴로지컬 정렬 알고리즘은 도 7의 6줄과 15줄에서 보이는 것처럼 루트 노드(root node)부터 시작하여 인접한 모든 노드에 대해 재귀적으로 토폴로지컬 정렬을 호출한다. 도 7의 17줄처럼 어떤 노드에 인접한 모든 노드가 스택(stack)에 있으면, 스택에 해당 노드를 삽입한다. 스택은 그래프의 의존성을 어긋나지 않는 하나의 순서를 만들어준다.
- [0060] 본 발명은 리소스 의존성 정보뿐 아니라 각 리소스의 중요도 또한 리소스 우선순위에 활용하였다. 우선 각 리소스에게 점수를 리소스 중요 경로 여부, 리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기를 기준으로 점수를 부여하였다.
- [0061] 도 7의 각 리소스의 중요도 점수를 부여하는 함수인 SCORE에서 subgraphsize는 리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기, criticalpath는 리소스의 중요 경로 여부, 그리고 relativestarttime은 각 리소스가 요청된 시간을 의미한다.
- [0062] subgraphsize와 criticalpath 정수이고 relativestarttime은 0과 1사이의 값이기 때문에 subgraphsize와 criticalpath에 의해 우선순위화 순서는 결정된다. relativestarttime은 동일한 점수의 리소스 간 순서를 만드는 1보다 작은 값이다.
- [0063] 기존의 깊이 우선 탐색 토폴로지컬 정렬 알고리즘에서 각 노드가 인접한 노드에 대한 토폴로지컬 정렬을 노드의 중요도 점수가 낮은 순서로 재귀적으로 호출하게 하였다. 이는 도 7의 4줄과 14줄에서 확인할 수 있다. 이 방식을 사용하면 중요도 점수가 높은 노드가 나중에 탐색되어 스택에 먼저 삽입될 확률을 낮춘다.
- [0064] 상기 리소스 프리로드부(170)는 웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드한다.
- [0065] 프리로드란 곧 필요로 하는 리소스 정보를 브라우저에게 미리 알려주는 방법이다. 주로 HTML 파일의 앞부분 <head> 안에 <link> 태그를 이용하여 적시한다. 이와 같은 프리로드를 사용하면 리소스의 발견까지 걸리는 시간을 단축해 필요로 하는 리소스의 요청을 빠르게 보낼 수 있다.
- [0066] 리소스 우선순위화는 서버가 클라이언트에 보내는 리소스 내에 우선순위를 삽입하여 클라이언트가 이를 바탕으로 우선순위화를 수행하도록 하였다. 이 때, 프리로드가 사용되었는데, 클라이언트가 웹 페이지 로딩 시 받는 초기 HTML 파일의 <head>에 리소스 우선순위화 순서대로 각 리소스를 프리로드하게 하였다. 브라우저는 프리로드된 순서대로 리소스 요청을 보낸다.
- [0067] 도 2의 테스트 페이지의 indexA.html을 본 발명을 이용하여 리소스 우선순위화 순서를 도출하고, 도출한 리소스 우선순위화 순서대로 초기 HTML 파일의 <head>에 preload로 삽입한다. 도 8은 본 발명의 방법론대로 변경한 초기 HTML파일 indexB.html을 보여준다. indexB.html을 서버가 서비스하면 브라우저는 우선순위 순서대로 리소스 요청을 보낸다.
- [0068] 도 9는 본 발명에서 제안한 방식으로 리소스 우선순위화를 다른 리소스 우선순위화 방법과 그 성능을 비교한 그래프이다. NO-PRELOAD는 우선순위화 없이 웹 페이지 로딩을 수행한다.
- [0069] DEFAULT[선행기술문헌의 비특허문헌 1]은 WProfX 측정 시 기록된 순서대로 리소스를 프리로드한다. 이는 Chrome의 리소스 우선순위화의 결과대로 로딩을 수행하는 것이다. SIMPLETOP[본 발명에 따라 리소스 의존성 그래프만으로 우선순위를 결정하는 알고리즘]은 기본 토폴로지컬 정렬의 순서대로 리소스를 프리로드하여 웹 페이지 로딩을 수행한다. REVTOP[본 발명에 따라 리소스 의존성 그래프 자체와 리소스 중요도 점수를 둘다 고려하여 우선순위를 결정하는 알고리즘]은 본 발명에서 제안한 우선순위화의 방식대로 프리로드하여 웹 페이지 로딩을 수행하였다. 도 9에서 본 발명에 따른 REVTOP이 RelativeRUM-SpeedIndex가 가장 작은 것을 확인할 수 있다.
- [0071] 도 10은 본 발명의 일 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 방법의 흐름도이다.
- [0072] 본 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 방법은, 도 1의 장치(10)와 실질적으로 동일한 구성에서 진행될 수 있다. 따라서, 도 1의 장치(10)와 동일한 구성요소는 동일한 도면부호를 부여하고, 반복되는 설명은 생략한다.

- [0073] 또한, 본 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 방법은 웹 리소스 의존성 기반 리소스 우선순위화를 수행하기 위한 소프트웨어(애플리케이션)에 의해 실행될 수 있다.
- [0074] 본 발명은 웹 페이지 로딩 시 리소스를 어떤 순서로 요청 혹은 응답할 것인가에 대한 문제인 리소스 우선순위화를 해결하기 위해 기존의 리소스 우선순위화 방법과는 달리 리소스 의존성(dependency) 정보를 활용하여 리소스 우선순위화를 수행하는 새로운 방안을 제안한다.
- [0075] 도 10을 참조하면, 본 실시예에 따른 웹 리소스 의존성 기반 리소스 우선순위화 방법은, 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출한다(단계 S10).
- [0076] 본 발명을 수행하기 위해서는 우선으로 웹 페이지 로딩 시 발생하는 액티비티 간의 의존성 그래프가 필요하다. WProfX는 Google Chrome 브라우저에서 웹 페이지 로딩 액티비티 간 의존성 그래프를 추출하는 기존의 도구이다.
- [0077] 캐시와 로컬 저장소가 비어 있는 상태에서 WProfX를 사용하여 웹 페이지 로딩 액티비티의 의존성 그래프를 얻는다. 캐시나 로컬 저장소에서 리소스를 얻어 오는 오브젝트 로딩을 제외하였기에 웹 페이지 로딩 액티비티들 중 오브젝트 로딩 액티비티는 모두 네트워킹이다.
- [0078] 의존성 그래프로부터 하나의 노드가 하나의 리소스에 대응하는 오브젝트 로딩 액티비티 간 리소스 의존성 그래프를 추출한다(단계 S20).
- [0079] 상기 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)이다. 리소스 의존성은 추후의 리소스 우선순위화에서 한 리소스의 로딩이 어떤 리소스의 로딩보다 선행되어야 하는지에 대한 명확한 기준을 제시한다.
- [0080] 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기 및 리소스가 중요 경로(critical path)에 속하는지 여부를 기초로 리소스의 우선순위를 결정한다(단계 S30).
- [0081] 상기 중요 경로(critical path)는 페이지 로딩 중 가장 시간이 오래 걸리는 일련의 액티비티 순서이다. 리소스 의존성 그래프는 하나의 노드가 리소스에 대응하는 유향 비순환 그래프(directed acyclic graph, DAG)이다. 토폴로지컬 정렬(topological sorting) 알고리즘을 사용하면 DAG에서 그 방향을 어긋나지 않는 하나의 순서를 만들 수 있다.
- [0082] 상기 리소스의 우선순위를 결정하는 단계는, 상기 리소스 의존성 그래프 자체로 리소스의 우선순위를 결정하거나 또는 상기 리소스 의존성 그래프 및 상기 리소스 의존성 그래프에서 결정된 리소스 중요도 점수를 기초로 리소스의 우선순위를 결정한다.
- [0083] 다시 말해, 일 실시예에서는 상기 리소스 의존성 그래프 자체로 리소스의 우선순위를 결정할 수 있고, 다른 실시예에서는 리소스 의존성 그래프 자체와 리소스 중요도 점수를 모두 고려하여 우선순위를 결정할 수 있다.
- [0084] 상기 리소스 중요도 점수는 상기 리소스 의존성 그래프에서 리소스에 대응하는 노드를 루트 노드(root node)로 설정하여 하위 그래프의 크기 및 리소스가 중요 경로(critical path)에 속하는지 여부에 따라 결정될 수 있다.
- [0085] 상기 루트 노드(root node)부터 시작하여 인접한 모든 노드에 대해 재귀적으로 토폴로지컬 정렬을 호출하고, 노드에 인접한 모든 노드가 스택(stack)에 있는 경우, 스택에 해당 노드를 삽입하여, 그래프의 의존성을 어긋나지 않는 하나의 순서를 생성할 수 있다.
- [0086] 상기 리소스의 우선순위를 결정하기 위해, 리소스에 해당하는 노드를 루트 노드로 볼 때 하위 그래프의 크기를 나타내는 subgraphsize 함수, 리소스의 중요 경로 여부를 나타내는 criticalpath 함수 및 각 리소스가 요청된 시간을 나타내는 relativestarttime 함수를 사용하여 각 리소스의 중요도 점수를 부여할 수 있다.
- [0087] 예를 들어, 상기 subgraphsize 함수와 criticalpath 함수는 정수이고 relativestarttime은 0과 1사이의 값일 수 있다.
- [0088] 웹 페이지를 로딩하는 경우 서버가 클라이언트에 보내는 초기 파일에 결정된 리소스의 우선순위 순서대로 각 리소스를 프리로드한다(단계 S40).
- [0089] 리소스 우선순위화는 서버가 클라이언트에 보내는 리소스 내에 우선순위를 삽입하여 클라이언트가 이를 바탕으로 우선순위화를 수행하도록 하였다. 예를 들어, 클라이언트가 웹 페이지 로딩 시 받는 초기 HTML 파일의 <head>에 리소스 우선순위화 순서대로 각 리소스를 프리로드하게 할 수 있다.

- [0090] 본 발명은 브라우저에 의해 상기 리소스의 우선순위 순서대로 리소스를 요청하게 된다. 다시 말해, 기존의 깊이 우선 탐색 토폴로지컬 정렬 알고리즘에서 각 노드가 인접한 노드에 대한 토폴로지컬 정렬을 노드의 중요도 점수가 낮은 순서로 재귀적으로 호출하게 하였다. 이 방식을 사용하면 중요도 점수가 높은 노드가 나중에 탐색되어 스택에 먼저 삽입될 확률을 낮출 수 있다.
- [0091] 이와 같은 웹 리소스 의존성 기반 리소스 우선순위화 방법에 따르면, 웹 페이지 로딩 액티비티 간 의존성을 고려하였기에 이에 따른 빠른 렌더링으로 결론적으로 좋은 사용자 경험을 제공할 것이다. 본 발명은 사용자 경험과 밀접히 관련된 지표인 QoE(quality of experience) 지표 중 RUM-SpeedIndex 지표에서 좋은 성능을 보였다. Speed Index는 시간에 따른 웹 페이지 로딩의 시각적 완성도를 수량화한 지표로 RUM-SpeedIndex는 Speed Index를 Resource Timing API를 통해 계산한 지표이다.
- [0092] 이와 같은, 웹 리소스 의존성 기반 리소스 우선순위화 방법은 애플리케이션으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.
- [0093] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0094] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.
- [0095] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0096] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

산업상 이용가능성

- [0098] 본 발명은 더 나은 사용자 웹 경험을 제공하고자 하는 웹 서비스를 제공하는 사업자가 활용할 수 있다. 현재 웹 서비스를 하는 서버를 가진 사업자의 경우, 현재 서비스 중인 웹 콘텐츠를 위 방식으로 수정하여 서비스하면 리소스 의존성을 고려한 리소스 우선순위화된 웹 서비스를 제공할 수 있다. 브라우저 내에서 리소스 우선순위를 수행하는 것 또한 가능하다. 웹 콘텐츠의 수정 없이 브라우저 내에서 리소스 의존성 정보를 고려한 리소스 우선순위를 수행할 수 있다. 웹 서비스를 제공하는 대부분 기업에서 활용할 수 있으며, 사용자가 겪는 웹 서비스의 품질 향상이 이익과 직접 연결되는 시장에서 활용이 더욱 기대된다.

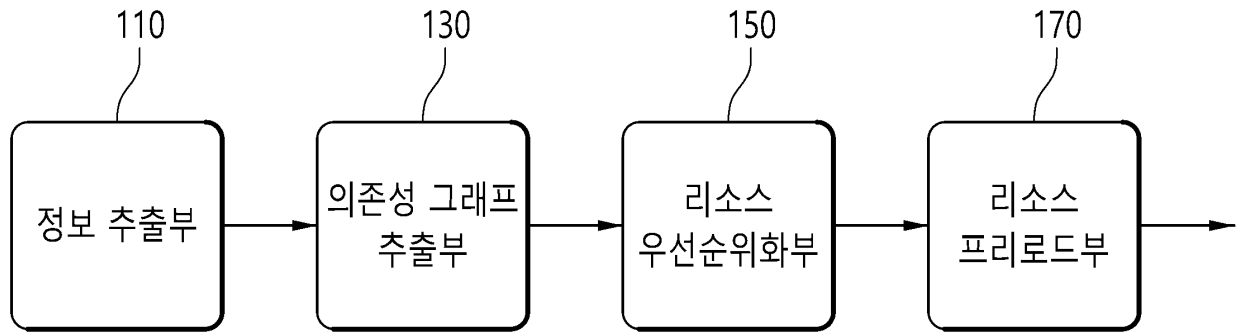
부호의 설명

- [0100] 10: 웹 리소스 의존성 기반 리소스 우선순위화 장치
 110: 정보 추출부
 130: 의존성 그래프 추출부
 150: 리소스 우선순위화부
 170: 리소스 프리로드부

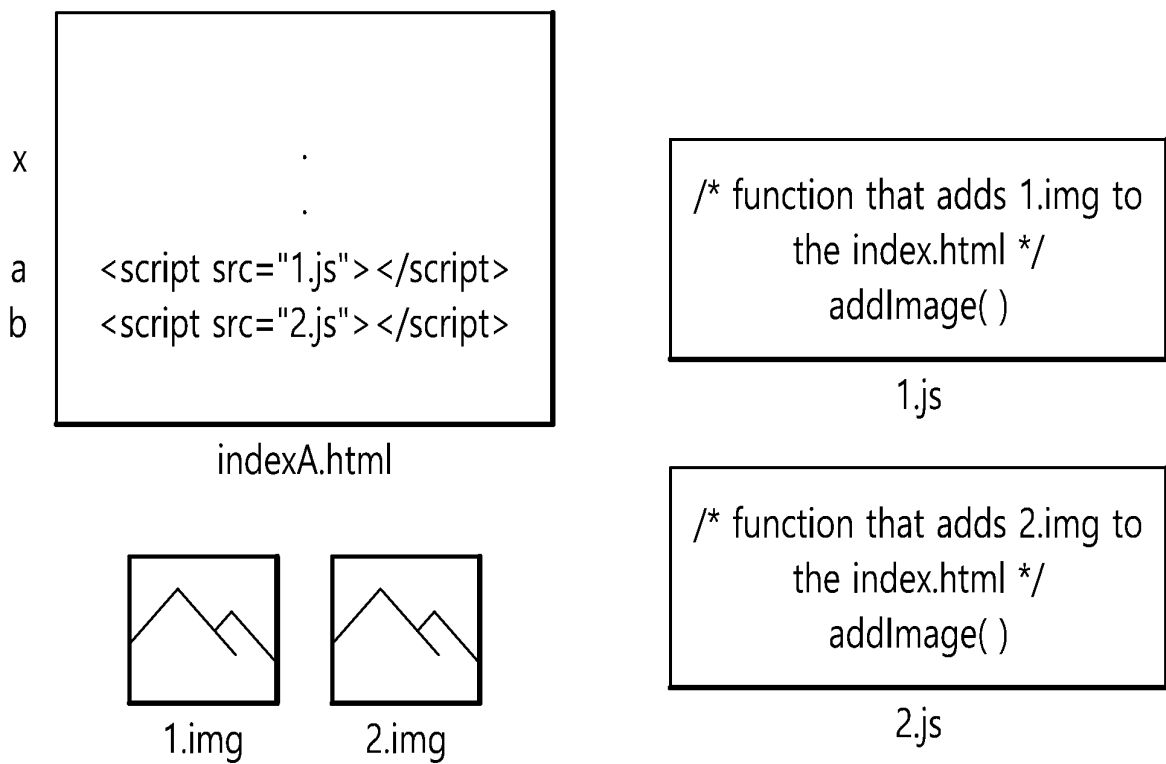
도면

도면1

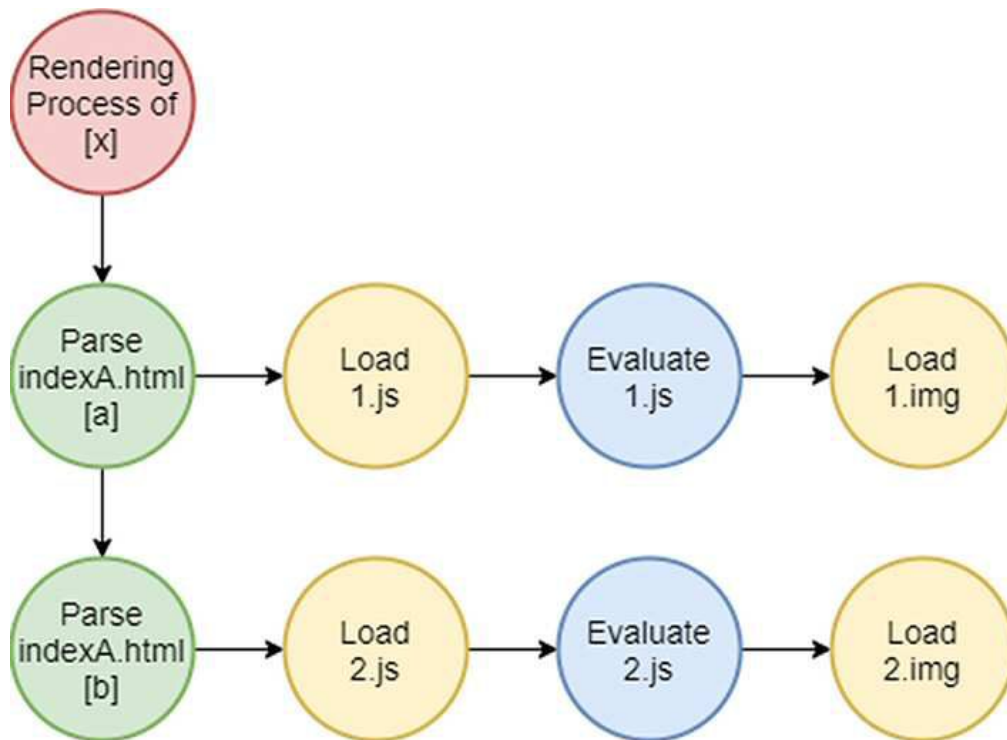
10



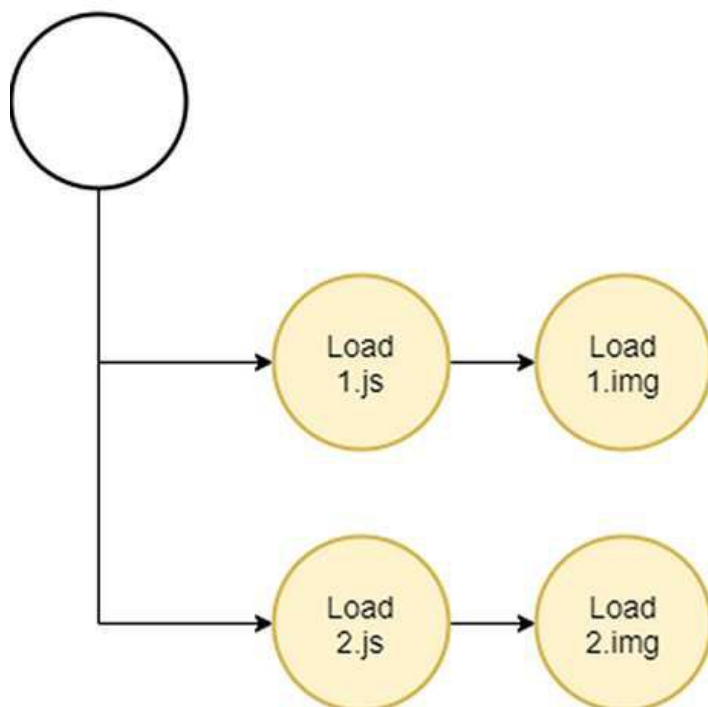
도면2



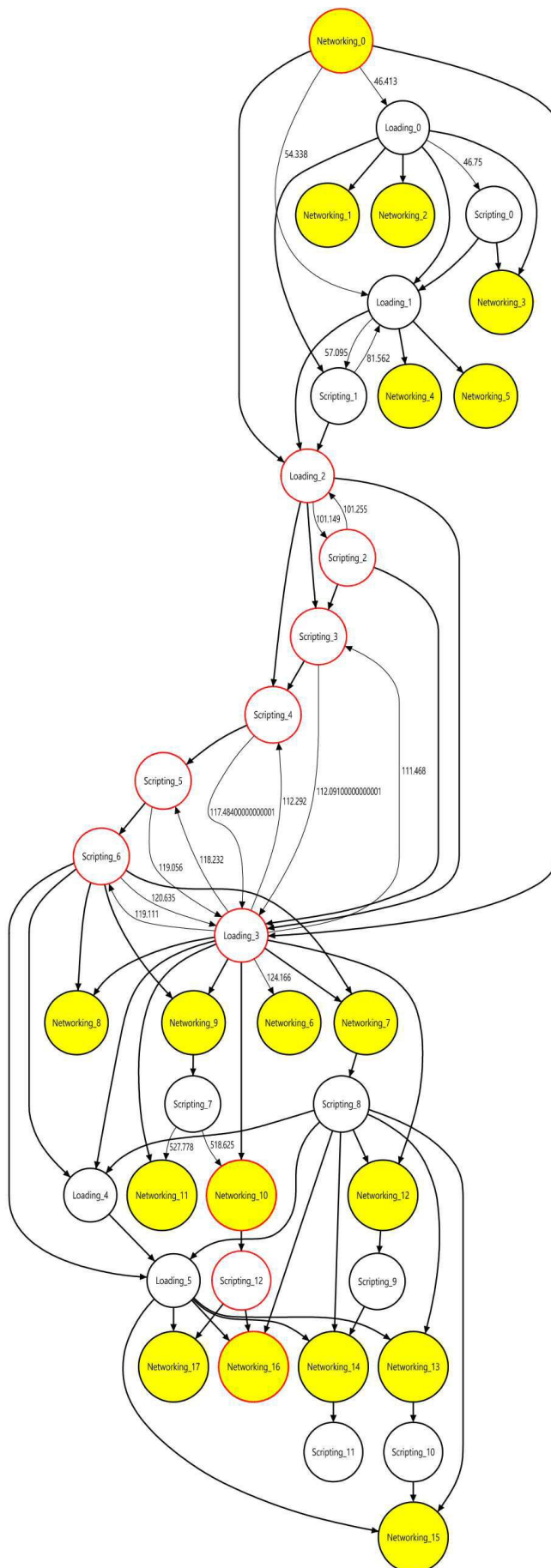
도면3



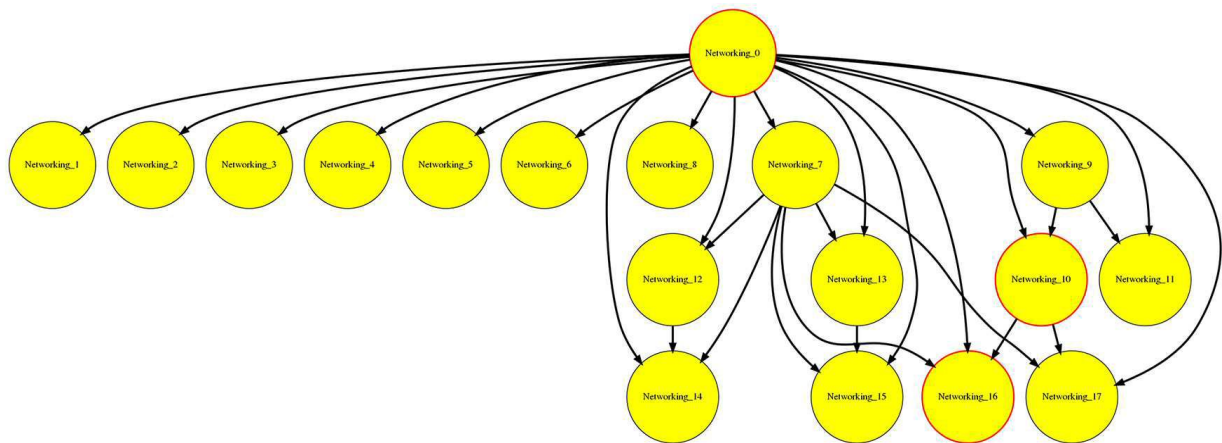
도면4



도면5



도면6



도면7

```

G = (V, E), G is a directed acyclic graph
1: function TOPSORT(G)
2:   stack ← [ ... ]
3:   Initialize visited[v] = False for all vertices v
4:   for each vertex v in V in ascending order of SCORE(v) do
5:     if visited[v] = False then
6:       RECURSIVETOPSORT(v, visited, stack)
7:     end if
8:   end for
9:   Return stack
10: end function

11: function RECURSIVETOPSORT(v, visited, stack)
12:   visited[v] = True
13:   for each adjacent vertex adjv of v in ascending order of SCORE(v) do
14:     if visited[adjv] = False then
15:       RECURSIVETOPSORT(adjv, visited, stack)
16:     end if
17:     insert v to stack
18:   end for
19: end function

20: function SCORE(v)
21:   subgraphsize ← number of vertices in subgraph of G when v is regarded
    as the root node of the subgraph
22:   if v is in critical path then
23:     Criticalpath ← 1
24:   else
25:     Criticalpath ← 0
26:   end if
27:   weight ← 10
28:   relativestarttime ← relative starting time of loading that corresponds to v
29:   score = criticalpath × 10 + subgraphsize - relativestarttime
30:   Return score
31: end function

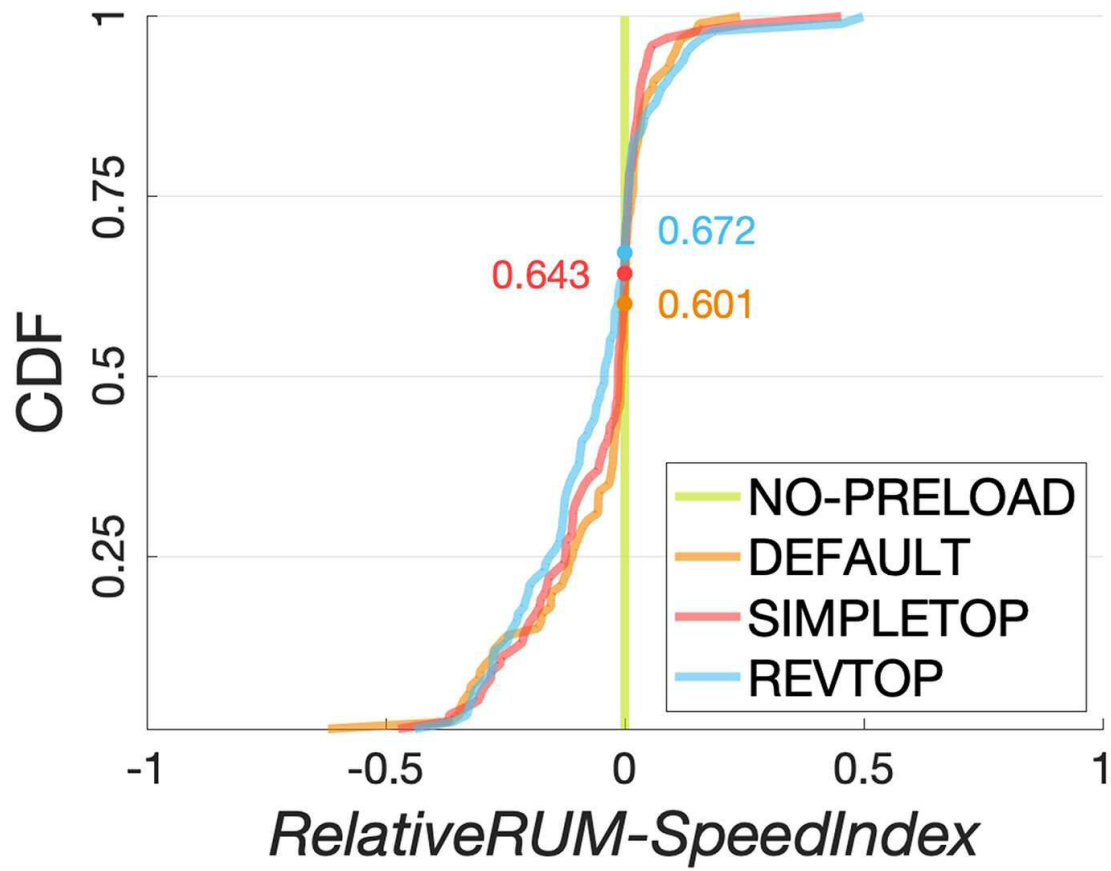
```

도면8

a	<link rel="preload" src="1.js"> </link>
b	<link rel="preload" src="2.js"> </link>
c	<link rel="preload" src="1.img"> </link>
d	<link rel="preload" src="2.img"> </link>
x	.
	.
e	<script src="1.js"> </script>
f	<script src="2.js"> </script>
	.
	.

indexB.html

도면9



도면10

