웹 페이지 로딩시간 감축을 위한 HTML 5 분석

윤준수* · 박진태* · 황현서* · 표경수* · 문일영*

*한국기술교육대학교

Efficient Multicasting Mechanism for Mobile Computing Environment

Jun-soo Yun* · Jin-tae Park* · Hyun-seo Hwang* · Gyung-soo Phyo* · Il-young Moon*

*School of Computer Science and Engineering, Korea University Technology and Education

E-mail : yuntn55@koreatech.ac.kr

요 약

HTML5 기반의 웹 플랫폼이 차세대 국가 표준으로 제정되면서 웹 서비스 업체들은 스마트 미디어 기기 및 스마트 TV에서 HTML5 기반의 앱 지원 기술을 경쟁적으로 개발하고 있다. 국제 웹 표준 개발 조직인 W3C를 중심으로 Microsoft, Apple, Mozilla, Google, Opera 등 다양한 웹 브라우저 벤더가 표준화에 참여하고 있다. 이렇듯 점차 HTML5의 중요성이 강조되며 HTML5 기반의 웹 페이지는 더 많은 양의 정보를 포함하여야 되며 더 빠른 속도의 로딩시간이 필요하다. 따라서 본 논문은 웹 페이지 로딩시간을 감축하기 위한 초기 연구로써 각 브라우저별 동일한 웹 페이지를 구성하여 초기 로딩 시간을 측정한다. 뿐만 아니라 HTML5 태그, 및 CSS 속성을 하나씩 제거하면서 초기 로딩 시간에 많은 비중을 차지하는 태그 및 속성을 분석하고 그 결과를 통해 향후 웹 페이지 로딩시간을 감축할 수 있는 방안을 마련한다.

ABSTRACT

HTML5-based Web platform is established as a next-generation national standards, Web services provider, has been competitively develop support technology HTML5-based app in smart media devices and smart TV. In accordance with the W3C is an international Web standards development organization, Microsoft, Apple, Mozilla, Google, such as Opera, various Web browser vendors are participating in standardization. Gradually emphasized the importance of HTML5, HTML5 -based Web pages, it is necessary to fast load times when contained a large amount of information. Therefore, in this paper, the initial studies in order to reduce the loading time of a web page, configure each browser-specific same Web page, and measure the initial loading time. Also, one by one remove the HTML5 tags, and CSS property, to analyze the tags and attributes of the account for a large proportion to the initial load time. Through the results, it is desired to provide a process which can reduce the Web page.

키워드

HTML5, TAG, CSS, Web, Roading Time

Ⅰ. 서 론

2014년 10월 28일, W3C에서 웹 문서 및 애플리케이션 표준 기술규격으로 HTML5를 공식표준으로 확정했다. HTML4.01이 1999년 12월에 표준으로 발표된 지 15년의 시간이 흘렀다. 그동안 스마트폰, 태블릿 PC등 다양한 하드웨어의 등장으로 일상생활에서 의사소통 방식과 미디어 소비 방식을 변화시켰다. 사람들은 SNS 서비스를 통해 소식을 전하고, 애플리케이션을 통해 광고를 한다. 하지만 iOS, Android, Windows8 등 여러 스마트폰 플랫폼의 경쟁으로 인하여 스마트폰 앱 개발자 들은 어려움을 겪고 있다. 각각의 플랫폼에 맞게 앱을 개발하여야 하며 그로인한 시간 및 인력낭비가 심하게 되었다[1][2]. HTML5의 가장 큰 특징은 '호환성'이다. 검색, 지도, 메일, 동영상 등 부가 프로그램을 설치하지 않고도 서비스를 제공할 수 있다. 또한, 브라우저 사이에 표시되는 내용의 차이가 사라진다[3]. 이러한 호환성을 '크로스 플랫폼(Cross-platform)'이라 하며, HTML5에서 실현 가능하도록 만들었다. 하지만, 아직까지도 각 플랫폼들 사이에 완벽한 호환성을 보이지 못하고 있다. 크롬에서 되는 기능이 익스플로러에서는 되지 않으며, 각각의 플랫폼마다 접속 속도의 차이가 있다. 특히 웹 브라우저의 초기 로딩 속도는 여러 플랫폼마다 다르게 측정되어 사용자들 사이에는 자신이 원하는 기능에 따라 플랫폼을 설치하여야 한다. 특히 스마트폰 환경에서는 낮은 CPU와 낮은 RAM, 느린 통신 속도로 인해 웹페이지 로딩 시간이 매우 느리게 측정된다. 이러한 웹 페이지의 낮은 로딩 속도로 인하여 웹페이지의 접속을 포기하면 아마존과 구글 같은 대형 웹 페이지에서는 막대한 양의 손실을 가져올 수 있다[4][5].

따라서 본 논문에서는 HTML5 분석을 통해 브라우저별 초기 로딩시간의 원인을 분석하고자 한다.

Ⅱ. 관련 연구

HTML5은 크로스 플랫폼이 핵심 기술로 활용될 수 있다. 즉, HTML5는 특정 플랫폼이나 디바이스에 종속되지 않고, 편리하게 서비스 제공이 가능하다는 특징을 가지고 있기 때문에 원 소스 멀티 유스(One Source Multi Use) 실현이 가능하며, PC, 스마트폰, 태블릿 PC, 스마트 TV 등의 다양한 디바이스, Windows, Linux, iOS, Android 등의 다양한 플랫폼에서 다양한 콘텐츠를 표준화도니 방법으로 제공할 수 있다. 현재까지는 네이티브와 비교 시 느린 성능 이슈와 브라우저나 웹뷰에서 지원하는 HTML5 기능이 서로 상이한 것에 대한 이슈가 있다. 이러한 이슈들은 상당기간 유지될 것으로 보이나 Google, Microsoft, 삼성전자, Apple 등 세계의 주요 기업들의 노력으로 지속적으로 개선되고 있다.

2.1 Safari

Apple에서 만든 웹 브라우저인 Safari는 OS X와 iOS를 사용한 애플 제품들의 기본 브라우저이다. Safari는 Chrome이나 Firefox보다는 확장 프로그램이 다양하진 않지만, iCloud와 연동되는 기능으로 인해서 웹페이지 로드 속도가 빠르다. Apple은 2010년 Safari version 5.0부터 2013년 Safari version 7까지 꾸준한 발전으로 인해 웹 로딩 속도를 개선하였고, 2014년에는 OS X Yosemite 출시와 함께 Safari version 8을 공개했다. Safari 8은 JavaScript 엔진의 개량으로 브라우징의 속도가 대폭 빨라졌고, 손가락을 이용한 브라우저 컨트롤 기능도 추가하였다.

2.2 Google Chrome

Chrome은 2008년에 정식으로 구글에서 블링크 렌터링 엔진을 사용하여 개발한 웹 브라우저이다. 구글은 2013년에 렌더링 엔진을 웹키트 기반으로 만들어 크롬과 오페라에 탑재하였다. 크롬의 초기 버전은 IE에 비해 가볍고 빠른 웹 브라우저였고, 당시 열풍이었던 넷북으로 인해 점유율을 빠르게 높일 수 있었다. 하지만, 크롬은 탭마다 프로세스를 따로 사용하기 때문에 타 브라우저보다 메모리 이용량이 많았다. 이것 때문에 컴퓨터 사양에 따라서 크롬의 속도는 바뀌었다. 이후, 크롬은 구조를 간단하게 개발해 속도를 개선하였고, 렌더링 엔진은 사파리에서 사용된 웹키트를 사용하였으나 28버전부터는 웹키트를 개량하여 만든 블링크 엔진을 이용했다. 특히 JavaScript 처리 엔진으로 덴마크에서 개발한 최신 V8 JavaScript engine을 사용하여 페이지의 속도를 대폭 향상시켰다.

2.3 Mozilla Firefox

모질라 파이어폭스는 모질라 재단에서 개발한 Gecko 엔진 기반의 오픈소스 웹 브라우저이다. 파이어폭스는 3.6.x까지는 JavaScript 엔진 속도가 느렸지만, 4.0.x부터 새로운 JavaScript 엔진을 사용하고, 페이지 표시에 하드웨어 가속을 사용하여 속도를 향상시켰다. 하지만, 고질적으로 Gecko 엔진의 분산처리가 안 되어 반응속도의 문제가 있지만, 모질라도 이 문제를 인지하고 대책을 마련중이다.

Ⅲ. 웹 브라우저 초기 로딩시간 분석

웹 브라우저의 초기 로딩 시간 개선을 위하여 우선 웹 브라우저 별로 가장 많은 로딩 시간을 차지하는 속성을 찾아야 한다. 만약 가장 많은 로딩 시간을 차지하는 부분을 다른 언어로 대체하

거나 약간의 기능을 변경하여 사용하게 된다면, 초기 로딩 속도가 개선된다. 로딩 시간을 많이 차지하는 속성을 찾기 위하여 IE, Chrome, Opera, Safari, Firefox 총 5개의 브라우저를 통해 분석하였으며, 모두 동일한 인터넷 환경과 측정 방법으로 속도를 측정하였다.

3.1 실험 환경

웹 브라우저의 초기 로딩 속도를 정확하게 측정하기 위하여 아파치 HTTP 서버를 실험 PC에 설치하여 서버 PC를 만든 후 서버 PC에서 속도를 측정한다. 이렇게 하면, 인터넷 접속 속도에 따른 영향을 받지 않고 정확한 측정이 이루어진다. 로딩시간 측정은 로딩이 시작되는 시간과 로딩이 완료되는 시간을 기록한다. 로딩완료시간에서 로딩시작시간을 빼 초기 로딩 시간을 측정한다.

실험을 진행할 웹 페이지는 최근에 가장 많이 사용되는 사이트들과 같은 기능의 웹페이지를 구성하여 로딩시간을 측정하고, 점차 HTML5의 각 태그, JavaScript, CSS를 하나씩 제거하면서 가장 많이 시간이 감축되는 요소를 찾는다. 측정횟수는 각 100회씩 실시하여 평균값을 이용한다.

3.2 HTML5 요소 분석

3.2.1 HTML 속성, JavaScript, CSS 분석

HTML5의 속성은 <p>, <ul>, <link>, <input>, <label>, <img> 등으로 가장 많이 사용하는 태그 요소들을 제거하였다. 표1은 5개의 브라우별 각 요소를 제거하였을 시 감소율을 나타낸 것으로 oprigin의 경우 원본 홈페이지로써 요소를 제거하지 않고, 측정한 결과이다. 그림 2는 5개의 브라우저별 각 요소를 제거하였을 시 측정된 시간을 나타낸 그래프이다.

표 1. 브라우저 별 HTML 각 요소 제거 시 시간 감소율

요소	감소율				
	익스플로러	크롬	파이어폭스	사파리	오페라
origin	0%	0%	0%	0%	0%
Script	94.94%	80.11%	81.92%	69.49%	87.51%
css	9.37%	3.09%	3.48%	35.76%	5.87%
p	8.54%	1.26%	5.87%	9.54%	3.21%
ul	10.04%	1.36%	5.33%	18.33%	0.77%
link	6.64%	5.36%	8.77%	42.97%	7.14%
input	2.81%	4.34%	8.9%	17.7%	7.86%
label	0.15%	3.33%	8.5%	17.11%	2.71%
section	1.92%	4.73%	6.2%	5.26%	3.32%
img	6.72%	5.38%	11.1%	9.09%	4.26%
video	4.85%	4.03%	9.57%	20.78%	7.19%
div	29.25%	3.82%	17.75%	48.19%	9.8%

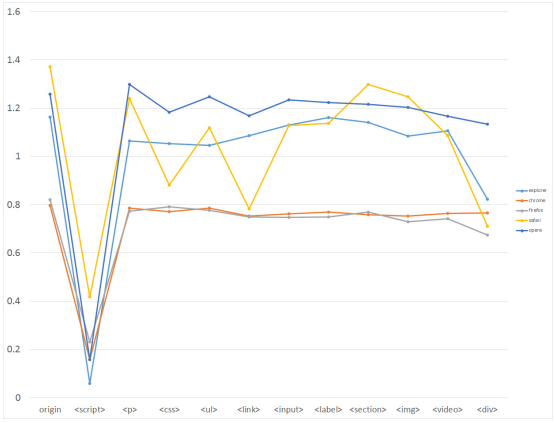

그림 1. 각 브라우저별 HTML 요소 제거 시 로딩시간

각 브라우저마다 차이는 있지만 JavaScript를 제거하고 측정을 하였을 시 평균 82.79% 라는 감소율을 보였다. 이는 자바 스크립트가 인터프리터 언어이기도 하는 동시에 많은 계산량을 가지고 있기 때문인 것으로 보인다. 또한 제거 할 때에도 많은 시간 감축을 보였으나 를 제거하게 된다면 원활한 UI를 제공해 주기 어렵기 때문에 를 되도록 제거하지 않는 것이 좋다. CSS 역시 많은 시간 감축을 보였는데 CSS가 제거 되었을 경우도 UI 부분에 많은 영향을 미친다.

3.2.2 CSS의 각 속성 분석

웹 브라우저의 초기 로딩속도 저하에 많은 영향을 미치는 요소는 JavaScript, , CSS 이다. JavaScript의 경우 제거를 하게 되면 웹페이지의 기능상에 문제가 생기며 의 경우는 기본적인 태그 속성이라 더 이상 분석을 할 수 없다. 따라서 CSS의 각 속성들을 제거하여 추가적인 실험을 하였다. 실험 환경은 전과 동일하며 CSS의 제거 요소로는 background, margin, height, width, padding, position, border, outline, overflow, webkit-keyframes, webkit-annimation, transition, float, display, color, text, box, z-indx, font, cursor로 모두 웹페이지의 UI만 관여하는 속성들이다. 이 속성들을 제거한다고 하여도 웹페이지 기능에는 크게 문제되지 않는다. 표 2는 CSS의 각 요소를 제거 하였을 때의 시간 감소율을 나타낸 것이고 그림 3 는 요소 제거 시 로딩시간을 그래프로 나타낸 것이다.

표 2. 브라우저별 CSS 각 요소 제거 시 시간 감소율

요소	감소율				
	익스플로러	크롬	파이어폭스	사라피	오페라
origin	0%	0%	0%	0%	0%
background	1.95%	3.27%	26.69%	44.72%	10.38%
margin	6.29%	6.3%	19.63%	6.79%	12.19%
height	10.9%	10.9%	17.27%	7.85%	4.51%
width	8.11%	11.51%	18.71%	19.09%	23.2%
padding	7.13%	6.18%	22.64%	7.31%	16.35%
position	7.27%	1.57%	79.9%	57.67%	6.86%
border	9.79%	7.75%	16.47%	11.42%	14.15%
outline	8.95%	9.45%	14.51%	7.82%	9.75%
overflow	11.88%	13.21%	16.64%	4.31%	19.69%
webkit -keyframes	5.45%	1.21%	14.95%	26.72%	22.76%
webkit -animation	7.27%	8.24%	17.93%	12.09%	16.98%
transition	3.49%	9.33%	19.96%	6.57%	6.86%
float	12.16%	9.57%	14.97%	7.86%	2.71%
display	8.11%	10.9%	14.05%	4.79%	6.41%
color	14.54%	1.57%	21.66%	11.75%	5.51%
text	7.13%	0.12%	20.42%	4.46%	12.37%
box	1.25%	12.72%	9.08%	3.26%	8.67%
z-lindex	6.43%	16.6%	16.03%	3.79%	23.12%
font	1.11%	0%	18.5%	1.72%	18.51%
cursor	4.05%	2.66%	19.19%	4.89%	18.51%

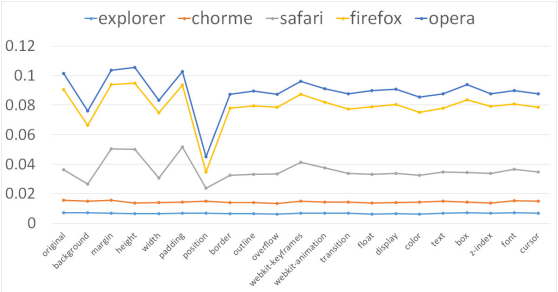

그림 2. 각 브라우저별 CSS 요소 제거 시 로딩시간

CSS 속성들의 시간 감축 역시 각 브라우저 마다 차이는 있지만 HTML5 속성처럼 큰 감축을 보이는 속성은 없다. 다만 CSS의 webkit 속성을 제거 시 많은 감축을 보이고 있었다. webkit의 경우 웹 브라우저의 렌더링을 하는 레이아웃 엔진으로 주로 웹페이지의 움직임을 담당하는 부분이다. webkit을 이용하여 JavaScript의 움직임을 제어하게 된다면 웹페이지의 초기 로딩속도를 감축할 수 있을 것이다.

Ⅳ. 결 론

스마트폰의 획기적인 기능과 편리성으로 인해 스마트폰 애플리케이션 시장은 폭발적으로 증가하였지만, 파편화된 플랫폼으로 인해 스마트폰 애플리케이션 개발은 힘들어졌으며, 새로운 대안이 필요했다. 따라서 모든 플랫폼을 지원할 수 있는 HTML5가 각광받고 있다. HTML5는 널리 사용되고 있고, 웹페이지 초기 로딩 시간 단축에 관한 연구도 세계 곳곳에서 활발히 진행 중이다. 본 논문에서는 HTML5 분석을 통해 브라우저별 초기 로딩시간의 원인을 분석하였다. 이를 통해서 웹페이지 초기 로딩 시간에는 JavaScript가 가장 많은 영향을 미친다는 것을 알아냈다. JavaScript의 경우에는 HTML5에서 없어서는 안 되는 중요한 연산기능을 수행하기 때문에 JavaScript를 제거하기 보다는 대체할 수 있는 방안을 연구해야 할 것이다.

참고문헌

[1] B. H. Ann, B. J. Kim, HTML5: The Current Status of Standards and the Case Studies, Korea Information Science Society, 30(5), 10-15, 2012

[2] J. C. Lee, All driving a new paradigm of Internet HTML5, The Society Of Air-Conditioning And Refrigerating Engineers Of Korea, 44(1), 88-90, 2015

[3] H. K. Cha, W. S. Lee, S. Y. Lee, W3C HTML5 Standards Trends, TTA Journal, Vol.159, 94-101, 2015

[4] S. H. Kim, S. J. Koh, Loading Speed Improvement of Web Browsers on Mobile Devices, Telecommunications Review, 22(1), 139-152, 2012

[5] H. S. Ko, T. K. Lee, K. S. Kim, T. S. Son, A Study of the Agent-based Web App Execution Time Improvement In Mobile Office Environment, Korean Society For Internet Information, 14(2), 177-178, 2013