

웹 브라우저의 고속 실행을 위한 뉴 메모리 기반 캡처 기법

김규식, 김태석
광운대학교 컴퓨터공학과
e-mail:tskim@kw.ac.kr

New Memory Based Capture Scheme for Fast Launching of Web Browsers

Kyu-Sik Kim, Tae-Seok Kim
Dept of Computer Engineering, Kwangwoon University

요 약

본 논문에서는 웹 브라우저의 사용자 응답성을 개선하기 위해 뉴 메모리를 활용하는 기법을 제안한다. 프레임버퍼에 저장된 기본 웹 페이지의 실행 화면을 뉴 메모리에 캡처하고, 이를 웹 브라우저 실행 시 그대로 복원함으로써 사용자가 체감하는 웹 브라우저의 응답성을 크게 개선한다. 또한 제안하는 기법을 뉴 메모리 에뮬레이션 보드에 직접 구현하여 그 성능을 검증하고자 한다.

1. 서론

최근 전원 공급이 끊어져도 데이터가 보존되는 비휘발성을 가지며 현재의 SSD나 하드디스크와 같은 저장장치보다 빠른 입·출력 속도를 갖는 차세대 비휘발성 메모리(이하 뉴 메모리)가 활발히 연구되고 있다. 여러 종류의 뉴 메모리가 연구·개발되고 있으며, 그 중 STT-RAM은 현재의 DRAM과 근접한 읽기·쓰기 성능을 내고 있다[1].

새로운 웹 표준인 HTML5에서 웹 애플리케이션을 위한 다양한 요소의 추가로 기존의 네이티브 애플리케이션을 대체할 수 있게 되었다[2]. 이에 따라 여러 플랫폼으로 파편화 된 PC 및 모바일 시장에서 웹이 주목받고 있으며, 동시에 웹 페이지를 출력해주는 소프트웨어인 웹 브라우저의 중요성도 커지고 있다.

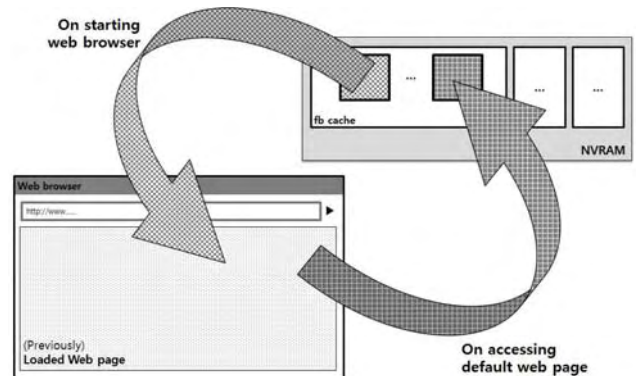
우리는 뉴 메모리를 활용하여 웹 브라우저에서 사용자가 느끼는 실행 시간을 단축하는 기법을 제안한다. 사용자가 웹 브라우저를 실행할 때 처음 접속하는 웹 페이지(이하 default web page, 기본 웹 페이지)에 접속할 때 웹 브라우저 전체를 정지 영상으로 만들어 뉴 메모리에 저장하고(이하 정지 영상 캡처), 이를 다음 웹 브라우저 실행할 때 복원(이하 정지 영상 복원)한다. 이러한 기법을 통해 사용자가 웹 브라우저 실행 시 체감하는 응답성을 개선할 것이며, 이를 뉴 메모리 에뮬레이션 보드에 직접 구현하여 확인하고자 한다.

2. 관련 연구

2.1. 뉴 메모리

뉴 메모리는 DRAM과 같은 빠른 입·출력 속도를 가지며, 저장장치와 같이 자료의 영속 저장이 가능하다는 특징이 있다[1]. 반면에 아직 자료 보존 기간이 현재의 저장장치보다 상대적으로 많이 짧거나, DRAM에 비해 수명이 짧다는 등의 문제점이 있다[3].

현재 위와 같은 문제점으로 인해 DRAM을 사용하는 메인 메모리를 전부 대체하거나, 성능 개선의 목적으로 기존 시스템에 추가하여 사용하는 것과 같은 뉴 메모리의 활용 방안에 대한 연구가 진행되고 있다[4]. 또한, 뉴 메모리 자체의 성능을 향상시키는 연구와 이를 활용하여 기존 시스템이나 소프트웨어의 성능을 향상시키고자 하는 연구 등이 계속 진행되고 있다[5,9].



(그림 1) 기본 웹 페이지의 저장 및 복원 기법

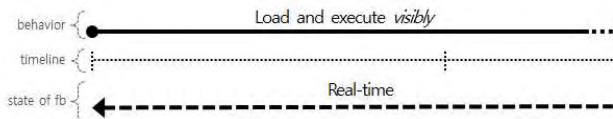
2.2. 프레임버퍼

디스플레이 장치에 출력할 영상 데이터를 담고 있는 그래픽 카드의 메모리 또는 메인 메모리의 일부를 지칭하는 말이다[6]. 리눅스에서는 디스플레이 장치에 영상을 출력하기 위해 하드웨어와 소프트웨어가 데이터를 교환할 때 사용되는 인터페이스를 의미한다[7]. 본 연구에서 제안하는 기법인 웹 브라우저의 정지 영상 캡처 및 복원 시 프레임버퍼를 이용하고자 한다.

3. 제안하는 기법

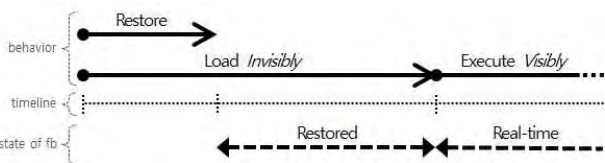
본 연구에서는 사용자가 인식하는 웹 브라우저 실행 속도의 단축을 위해 그림 1과 같이 웹 브라우저의 화면을 캡처하고 복원하는 기법을 제안한다.

3.1. 정지 영상 복원



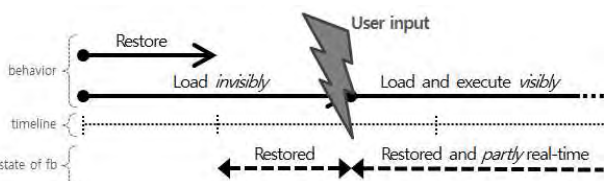
(그림 2) 복원할 영상이 없어서 일반적인 방법으로 웹 브라우저를 실행하는 경우

사용자가 웹 브라우저를 실행하면 웹 브라우저는 뉴 메모리의 정지 영상 저장 영역(이하 fb cache)에서 기본 웹 페이지에 해당하는 정지 영상을 찾는다. Fb cache에 해당하는 데이터가 없는 경우, 그림 2와 같이 기존 웹 브라우저와 동일하게 동작한다.



(그림 3) 정지 영상 복원 후 웹 브라우저를 로딩하는 동안 사용자의 입력이 없는 경우

Fb cache에 해당하는 캡처 영상이 있는 경우, 그 영상을 프레임버퍼에 복원한다. 이 시점에서 사용자는 로딩이 완료된 화면을 볼 수 있다. 이 이후에 백그라운드에서 웹 브라우저 로딩 과정을 진행하고, 로딩이 완료된 이후로 웹 브라우저의 화면을 실시간으로 출력한다. (그림 3)



(그림 4) 정지 영상 복원 후 웹 브라우저를 로딩하는 동안 사용자의 입력이 있는 경우

다만, 그림 4와 같이 백그라운드로 웹 브라우저 로딩을 수행하는 중간에 사용자의 입력이 들어올 때, 다음과 같은 추가적인 과정이 필요하다.

우선, 해당 시점 이전에 로딩이 완료된 요소들은 프레임버퍼에 즉각적으로 반영하고, 그 이후에 로딩이 완료된 요소들은 이전에 복원한 영상 위에 중복 기록하여 사용자가 보고 있는 영상이 변경되지 않도록 한다.

또한, 사용자로부터의 입력은 처리되기 전까지 저장한다. 그 중, 로딩이 완료된 요소에 대한 입력은 즉각적으로 처리하고, 그 이외의 입력은 대상 요소의 로딩이 완료되는 시점에 수행한다. 이와 같은 방법을 통해 사용자의 반응에 가능한 한 빨리 응답하여 사용자의 입력에 대한 반응성을 향상시키고자 한다.

3.2. 정지 영상 캡처

사용자가 기본 웹 페이지로 설정한 웹 페이지에 접속할 경우 그림 1과 같이 영상 정보를 프레임버퍼로부터 캡처하여 fb cache에 저장한다. 다만, 이전의 fb cache 내에 존재하는 영상 데이터가 충분히 최신의 상태이거나, 기본 웹 페이지 로딩이 끝나기 전에 사용자의 입력을 통해 다른 화면으로 바뀌는 경우에는 캡처하지 않는다.

4. 개발 환경

<표 1> 하드웨어 및 소프트웨어 개발 환경

종류	내용
테스트 보드	TUNA-NVM H2 (이하 TUNA 보드)
FPGA	XC7Z045-3FFG900 Zynq-7000
메인 메모리	1GB DDR3
뉴 메모리 에몰레이션	최대 2GB 지원
운영체제	Linux Kernel 3.18
웹 브라우저	qtdemobrowser (Qt 4.7.4, Webkit 기반)

본 연구에서는 표 1과 같은 하드웨어 및 소프트웨어 환경을 구성하여 제안하는 기법을 구현하고자 한다. 우선, 뉴 메모리 에몰레이션을 위해 TUNA 보드[8]를 사용한다. 이 보드는 메인 메모리 외에 추가로 뉴 메모리 에몰레이션을 위한 DRAM을 장착하고, 이에 단독 전원을 추가하여 뉴 메모리 에몰레이션을 구현하였다.

제안하는 기법을 추가할 웹 브라우저로 Qt 4.7.4 내에 구현되어 있는 qtdemobrowser를 사용한다. 이는 Webkit을 웹 브라우저 렌더러로 사용하며, qt 4.7.4를 이용하여 프로그램의 GUI, network 등을 구현하였다.

5. 결론 및 향후 연구 방향

본 논문에서는 뉴 메모리를 활용하여 웹 브라우저에서 자체 영상을 캡처하고 복원하는 방법을 통해 구동 시간을 줄이는 방법을 제안하였다. 이후에 본 내용을 실제 웹 브라우저에 반영하고, 실험을 통해 사용자 입장에서 웹 브라우저 실행 속도가 실제로 향상되는지 확인할 것이다. 또한, 이 기법을 기본 웹 페이지 뿐만 아니라 사용자가 방문하는 모든 웹 페이지를 대상으로 하여 캡처 및 복원함으로써 성능 향상을 얻을 수 있는지 연구하고자 한다.

6. 감사의 글

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천 기술개발사업(정보통신)의 일환으로 수행하였음. [No.10041608, 차세대 메모리 기반의 스마트 디바이스용 임베디드 시스템 소프트웨어]

참고문헌

- [1] Chen, Feng, Michael P. Mesnier, and Scott Hahn. "A protected block device for Persistent Memory." Mass Storage Systems and Technologies (MSST), 2014 30th Symposium on. IEEE, 2014.
- [2] W3C, "A vocabulary and associated APIs for HTML and XHTML", <http://www.w3.org/TR/html5/>.
- [3] Martin Fink, "Beyond DRAM and Flash, Part 2: New Memory Technology for the Data Deluge", <http://www8.hp.com/hpnext/posts/beyond-dram-and-flash-part-2-new-memory-technology-data-deluge>.
- [4] Dulloor, Subramanya R., et al. "System software for persistent memory." Proceedings of the Ninth European Conference on Computer Systems. ACM, 2014.
- [5] 전자신문, "D램·낸드 보완·대체할 차세대 메모리", <http://www.etnews.com/20150309000062>
- [6] Wikipedia, "Framebuffer", <http://en.wikipedia.org/wiki/Framebuffer>
- [7] "The Frame Buffer Device", <https://www.kernel.org/doc/Documentation/fb/framebuffer.txt>
- [8] "TUNA NVM-H2 Platform", <http://www.mangoboard.com/sub.html?ptype=view&code=notice&idx=8162>
- [9] 남태호, 김태석. "NVRAM 을 이용한 웹 브라우저의 메모리 캐싱 기법." 한국정보과학회 학술발표논문집 (2014): 1486-1488.