

다중 웹 고속화 기술 지원을 위한 실행파일 포맷 설계

남현우¹, 박능수²

¹건국대학교 컴퓨터공학과 박사과정

²건국대학교 컴퓨터공학과 교수

namhw@konkuk.ac.kr, neungsoo@konkuk.ac.kr

Executable file format design for multi-web high-speed technology support

Hyunwoo Nam¹, Neungsoo Park²

¹Dept. of Computer Engineering, Konkuk University

²Dept. of Computer Engineering, Konkuk University

요 약

최근 고성능 웹 애플리케이션 개발을 위한 asm.js, WebAssembly, WebGL, WebGPU와 같은 여러 웹 고속화 기술들이 발표되며 확장되고 있다. 하지만 각 기술들은 개별적으로만 적용 가능하다는 한계점이 있었다. 따라서 본 논문에서는 웹 브라우저 환경에서 최적화된 실행 성능을 기대할 수 있는 다중 웹 고속화 기술들을 통합 한 실행파일 포맷 및 패키징 방안에 대해 제안하고자 한다.

1. 서론

최신 웹 브라우저에서는 고성능 웹 애플리케이션 구현을 위한 다양한 웹 고속화 표준 기술들이 제정되며 발전하는 중이다. 예를 들면 인터프리터 방식으로 순차 실행되어 느렸었던 Javascript 기반 코드들을 네이티브에 가까운 속도로 향상시킨 asm.js 및 WebAssembly 그리고 스레드와 같은 병렬 실행을 위한 WebWorker 표준 기술들이 있다.

이와 같은 기술들은 CPU 기반의 실행속도를 개선할 수 있으며, 추가적으로 GPU 가속을 위해 WebGL 및 WebGPU와 같은 기술들이 적용 가능하다. 하지만 이와 같은 다양한 고속화 기술들은 상호 호환이 고려되어 있지 않으며, 따라서 개별 적용으로 인하여 시너지 효과를 기대할 수 없었다. 따라서 다양한 웹 고속화 기술들을 통합하고 최적화 할 수 있는 실행 환경에 대한 필요성이 대두되고 있다.

본 논문에서는 이를 위하여 다양한 웹 고속화 기술들을 하나의 웹 애플리케이션 패키징 형태로 통합할 수 있는 실행 파일 포맷을 제안하고자 한다.

2. 관련연구

2.1 CPU 기반 고속화 기술

먼저 CPU 기반 웹 애플리케이션 고속 기술로는 기존 Javascript를 대체하기 위한 asm.js[1] 및 WebAssembly[2] 파일이 있다. 이는 동적 타입 지원

으로 인한 속도 지연이 발생하였던 Javascript 엔진을 최적화 가능한 코드 형태로 변환하거나 바이너리 포맷을 통해 속도 개선을 가능케 하였다. 이 중 WebAssembly는 웹 브라우저에서 활용할 수 있는 새로운 바이너리 표준이며, 다운로드 할 웹 파일의 크기를 줄여주고 네이티브에 가까운 빠른 실행을 가능케 하여 최신 고성능 웹 애플리케이션에서 많이 적용되고 있다.

2.2 GPU 기반 고속화 기술

WebGL[3]은 웹 기반의 3D 그래픽 라이브러리로 별도의 플러그인 사용 없이 사용 가능하며, 최신 웹 브라우저에서 대부분 사용 가능하다. 특히 WebGL2 버전의 경우 Compute Shader 사용이 가능해져 3D 그래픽 뿐 아니라 범용 컴퓨팅 계산도 가능하다.

WebGPU[4]는 “3D 그래픽 및 계산 기능”을 제공하기 위한 가속 그래픽 및 컴퓨팅을 위한 최신 웹 표준 API이다. 기존 WebCL 및 WebGL API의 대안으로 발전 중이며, WebGPU는 Javascript 및 Shader Code를 이용하여 프로그램을 작성한다.

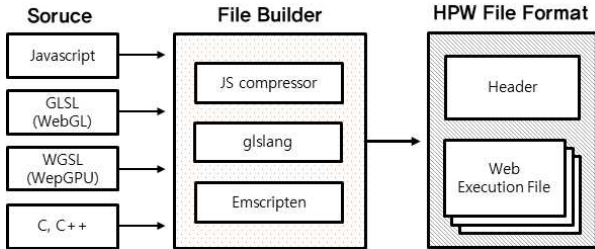
2.3 유니버설 바이너리(Universal binary)

유니버설 바이너리 파일은 PPC, x86, ARM 기반 애플사의 컴퓨터에서 에뮬레이션 없이도 다양한 아키텍처에서 실행 가능한 실행 파일 포맷을 말한다. 이는 다양한 아키텍처를 지원하기 위해 각 아키텍처별 실행 바이너리를 통합한 파일 포맷으로, 본 논문

에서는 다양한 웹 표준 기술들을 하나의 패키징 파일로 통합하기 위하여 해당 구조를 참조하였다.

3. 다중 웹 고속화 기술 적용 통합 파일 포맷

본 논문에서 제안하는 통합 파일 포맷인 HPW(High Performance Web) 실행파일을 생성하는 과정을 도식화 하면 그림 1과 같다.



(그림 1) HPW 파일 생성 프로세스

먼저 웹 애플리케이션을 구현할 수 있는 다양한 고속화 기술들에 대한 소스코드 파일들을 File Builder를 이용하여 하나의 통합된 HPW 파일로 패키징 한다. File Builder에는 기본 JS 코드를 압축하고 난독화 해주는 JS Compressor가 사용된다. 그리고 asm.js 및 WebAssembly 파일을 위한 C,C++ 소스코드는 Emscripten[5]을 사용하여 변환한다.

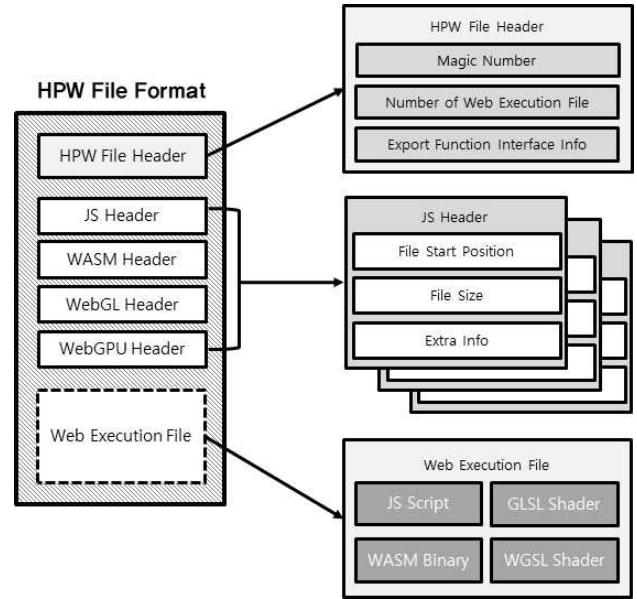
그리고 WebGPU 실행을 위한 GLSL 코드는 glslang 도구[6]를 이용하여 SPIR-V 코드로 변환된 후 패키징 된다. 다만 WebGL 실행을 위한 GLSL 코드는 추가 변환 없이 HPW 파일에 머지 된다.

최종 생성된 HPW 파일에는 각 기술 단위별로 사용되는 실행 코드들과 각종 헤더 정보들이 추가되며, 세부 구조는 그림 2와 같다. HPW 파일 포맷은 크게 메인 HPW 파일 헤더와 각 실행 파일별 필요한 헤더 정보 그리고 각 고속화 기술 타입에 따른 실행 파일(코드) 파일로 나뉜다.

HPW 파일 헤더에는 파일 구분을 위한 Magic Number와 해당 파일에서 지원하는 기술 타입들에 대한 정보 그리고 해당 모듈에서 외부 접근을 위해 정의한 API들의 Function 정보들이 포함된다.

다음으로는 각 실행 기술별 실행 파일(코드)에서 참조하기 위한 헤더 정보가 있으며, 각 헤더에는 파일의 위치, 크기, 그리고 부가 정보를 포함한다.

마지막으로 최종 실행 파일들은 각 위치에 배치되어 하나의 파일로 패키징되는 구조이다. 본 예시 처럼 실행 파일들은 하나의 파일에 통합될 수도 있으며, 구성 방법에 따라 원격 서버에 위치하고 필요 시에만 로드하여 사용도 가능할 것이다.



(그림 2) HPW 파일 포맷 구조

4. 결론 및 향후 연구

본 논문에서 제안한 다양한 웹 고속화 기술들을 지원할 수 있는 통합 실행 파일을 적용하여 향후 고성능 웹 애플리케이션의 가능성을 기대할 수 있다.

향후 연구로는 제안한 실행 파일 포맷을 로드하고 스케줄링 할 수 있는 프로그램 로더와 최적화 연구들을 지속하고자 한다.

Acknowledgements

This research was supported in part by Konkuk University's research support program for its faculty on sabbatical leave in 2021 and in part by the Supercomputer Development Leading Program of the National Research Foundation of Korea (NRF) funded by the Korean government (Ministry of Science and ICT(MSIT)) (No. 2020M3H6A1084853).

참고문헌

[1] Herman, D., L. Wagner, and A. Zakai. "asm.js: Working Draft." Available online at asmjs.org/spec/latest. Accessed June (2013).

[2] Haas, Andreas, et al. "Bringing the web up to speed with WebAssembly." ACM SIGPLAN Notices. Vol. 52. No. 6. ACM, 2017

[3] WebGL, https://developer.mozilla.org/ko/docs/Web/API/WebGL_API

[4] WebGPU W3C Working Draft, <https://www.w3.org/TR/webgpu/>

[5] Emscripten, <https://emscripten.org/index.html>

[6] Glslang, "Real-time GLSL to SPIR-V translation" <https://alexaltea.github.io/glslang.js/>