웹 실행 코드 타입에 따른 행렬 곱셈 성능 평가

남현우[1], 이명호[2], 박능수[3]
[1]건국대학교 컴퓨터공학과 연구원
[2]명지대학교 컴퓨터공학과 교수
[3]건국대학교 컴퓨터공학과 교수
namhw@konkuk.ac.kr, myunghol@mju.ac.kr, neungsoo@konkuk.ac.kr

Evaluation of matrix multiplication performance by web execution code type

Hyunwoo Nam[1], Myungho Lee[2], Neungsoo Park[3]
[1,3]Dept. of Computer Science and Engineering, Konkuk University
[2]Dept. of Computer Science and Engineering, Myongji University

요 약

웹3.0 시대의 다양한 응용 분야에서 활용되는 행렬 곱셈 연산에 대하여 최신 웹 표준 기술인 WebAssembly 및 WebGPU 표준을 적용하여 알고리즘을 구현하고 실행 성능을 평가한다. 실험 결과 작업의 크기가 커질수록 병렬화 효과로 인해 WebGPU 코드가 JS에 비해 최대 30배 빨라졌다. 또한 JS 코드에 비해서 WASM 코드의 실행 속도가 빠르며, 일부 작업의 크기가 작은 경우에는 WASM 및 WebGPU에서 초기 로딩 타임과 데이터 복사 작업에 따른 오버헤드가 있음을 확인하였다.

1. 서론

본 논문은 웹 3.0의 활용 분야인 인공지능, 블록체인, 메타버스(3D,AR,VR)와 같은 기술분야들을 지원하기 위해 웹 기반 고속화 실행 코드에 대해 연구한다. 기존 Javascript 언어의 경우 성능이 부족하였고, 이를 위해 WebAssembly[1] 및 WebGPU[2]와 같은 최신 표준들이 제안되었다. 본 논문에서는 최신 웹 표준 실행 기술들을 활용하여 인공지능 및 그래픽 렌더링 등 다양한 분야에서 활용되는 행렬 곱셈 연산을 구현한 후 성능을 측정하고 분석하였다.

2. 관련연구

2.1 웹 기반 CPU 및 GPU 실행 코드 표준 기술

CPU 기반 고속화 기술로는 asm.js 및 WASM (Web Assembly) 코드가 있다. 이는 동적 타입 지원으로 속도가 느렸던 Javascript 언어와 달리 최적화 가능한 코드 형태이거나 바이너리 포맷을 통해 실행 속도를 개선하였다. 특히 WASM은 웹의 새로운 바이너리 표준으로서 파일의 크기를 줄었으며 네이티브에 가까운 실행 속도를 보여주어 고성능이요구되는 최신 웹 앱에서 많이 활용되고 있다.

다음으로 GPU를 위한 기술로 WebGL은 별도의 플러그인 없이 최신 웹 브라우저에서 대부분 사용가능한 웹 3D 그래픽 라이브러리이다. 하지만 WebGL은 오래된 표준 기술이며 3D 그래픽 전용 라이브러리로서 컴퓨팅 목적으로는 적합하지 않다.

WebGPU는 “3D 그래픽 및 계산 기능”을 제공하기 위한 가속 그래픽 및 컴퓨팅을 위한 최신 웹 표준 API이다. 과거 WebCL 및 WebGL 표준의 대안으로 발전 중이며, 현재 WebGPU는 크롬 최신 버전에 릴리즈 되어 사용 가능하다. 또한, Rust 언어 기반 WGSL(WebGPU Shader Language)를 지원한다.

2.2 행렬 곱셈

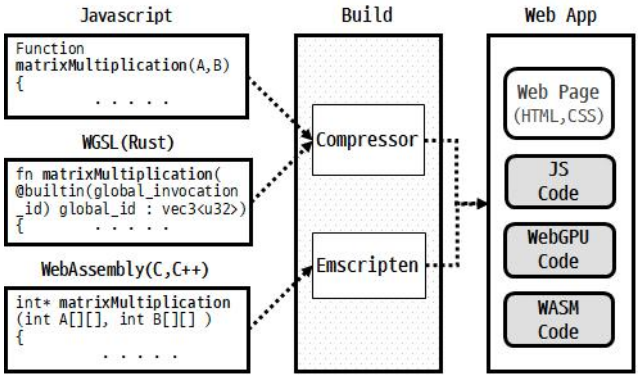
웹 브라우저는 멀티플랫폼 실행 환경으로 인공지능 및 3D 그래픽스 등 다양한 응용 분야에서 활용되고 있다. 예를 들어 웹 브라우저 기반 인공지능 애플리케이션을 구현해야 할 경우 행렬을 이용하여 다량의 데이터들을 표현하거나, 인공 신경망에서 처리되는 계산들을 행렬 연산을 활용하여 처리한다. 특히 그래픽 렌더링 작업에서는 수많은 선형 대수 혹은 행렬 곱셈 등의 연산을 필요로 한다.

이를 위해 본 논문의 실험에서는 행렬 A, B를 곱하여 D 행렬을 구하는 기본 행렬곱 알고리즘을 적용하였으며, 이를 정리하면 아래의 수식과 같다.

$$D_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj} = A_{i1} B_{1j} + \cdots + A_{ip} B_{pj}$$

3. 웹 기반 행렬 곱셈 알고리즘 구현

본 논문에서는 행렬 연산을 3가지 소스코드 타입으로 작성하였다. CPU에서 실행 가능한 Javascript 및 WASM 코드와 그리고 GPU에서 실행 가능한 WGSL 코드를 작성하였으며, 세부 과정을 도식화하면 그림 1과 같다.



(그림 1) 실행 코드에 따른 코드 작성 및 패키징 과정

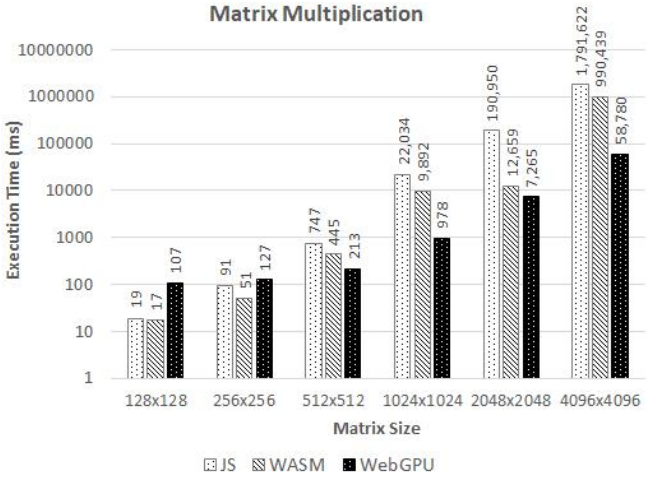
먼저 각 코드 타입별로 소스코드를 작성하며 WebGPU는 Rust 언어 기반의 WGSL 셰이더 언어로 작성하는데 Javascript 코드와 같이 텍스트 파일 포맷으로 Compressor를 통해 압축/난독화 된 후 웹 앱에 패키징 된다. 그리고 WebAssembly는 C, C++ 언어로 작성되어 Emscripten 컴파일러를 이용하여 WASM 코드로 빌드된 후 패키징 된다. 각 코드들은 동일한 함수 이름과 매개변수 형태를 가지며 ,각 언어별 Host API를 사용하여 호출하고 실행된다.

4. 실험 평가

세부 실험 환경은 표 1과 같으며, 다양한 크기의 행렬 사이즈를 대상으로 행렬 곱셈 연산에 소요된 총 소요 시간을 ms 단위로 측정하였다. 측정 결과는 그림 2와 같으며, 작업 크기가 작은 128x128, 256x256 크기에서는 WASM 코드의 실행 속도가 가장 빠르고 WebGPU 코드의 경우 오히려 JS보다도 느리다는 것을 확인하였다.

<표 1> 실험 환경

Type	Description
CPU	Intel i5-10210U @ 1.60Ghz
GPU	Intel(R) UHD Graphics (8Gb)
Memory	16Gb RAM
OS	Windows 10
Browser	Chrome 116.0.5845.188



(그림 2) 행렬 곱셈 실행속도 측정 결과

이는 초기화 시점에 WebGPU 객체를 생성하고 커널 코드에 인자값 전달 및 계산 결과를 받아올 때 CPU와 GPU 사이의 데이터를 복사하는 과정에서 오버헤드가 발생하기 때문이다.

다음으로 작업 크기가 1024x1024, 2048x2048, 4096x4096와 같이 커지는 경우 WebGPU 코드의 실행 속도가 JS 코드에 비해 최대 30배 빠르다. 이는 작업의 크기가 커질수록 연산 수행시 초기화나 데이터 송/수신 작업에 소요되는 시간이 전체 실행 시간에서 그 비중이 줄어들기 때문이다.

최종 실험 결과 작업 크기가 작은 경우에는 Javascript 코드가 일부 빠르나, 실제 고성능 처리가 필요한 큰 작업의 크기에서는 CPU에 최적화된 WASM 코드 또는 GPU에서 병렬 실행 가능한 WebGPU 코드를 활용하는 것이 실행 속도 개선에 가장 효과적이었다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00321688)

참고문헌

[1] Haas, A et al. “Bringing the web up to speed with WebAssembly”. In Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, Barcelona, Spain, 2017, 185-200.

[2] Kenwright, Benjamin. "Introduction to the webgpu api." ACM SIGGRAPH 2022 courses. 2022. 1-184.