MSc Thesis Description

Sebastian Paarmann (bvs368)
sepa@di.ku.dk

December 21, 2023

1 Working Title

Implementing a WebGPU Backend for Futhark

2 Background

Futhark [4] [5] [3] is a data-parallel array programming language developed primarily at DIKU. It is a pure functional language designed to be embedded into other programs to handle particularly compute-heavy parts. The ahead-of-time compiler is capable of performing various optimization and then generating code with one of multiple backends. The different backends allow targeting a variety of hardware and systems. For example, there is a C backend to compile Futhark for running on CPUs. There are also several GPU backends, targeting APIs such as CUDA and OpenCL.

WebGPU [6] is a new standard being developed as the successor of WebGL. Both APIs allow websites to utilize GPU hardware via the browser. While WebGL is heavily based on OpenGL and focused on graphics acceleration, WebGPU more closely follows more modern APIs like Vulkan in its design and explicitly supports general-purpose compute on GPUs (GPGPU). WebGPU describes an API for interacting GPUs that is implemented by web browsers in JavaScript, but there also exist native implementations of the API, such as Dawn [2] and wgpu [7]. When using WebGPU, the code run on the GPU itself (known as *shaders*) is written in WGSL, the WebGPU Shading Language[1], a relatively simple imperative programming language with C-like syntax.

3 Problem Statement

The objective of the project is to implement a new Futhark compiler backend targeting WebGPU, allowing Futhark programs to run in a web browser with GPU acceleration. This work is split into two main components:

- A kernel code generator targeting WGSL. With the existing GPU backends such as CUDA and OpenCL, the GPU-side code is effectively written in C generated by the compiler. WGSL is sufficiently different that a separate code generator will be required.
- A CPU-side WebGPU backend. The CPU side sets up all the required resources and dispatches and coordinates the GPU-side kernels. This will need to be implemented for the new target API.

Neither WebGPU nor WGSL are fully mature yet, and do not have all the features exposed by for example CUDA. Part of the project will be to evaluate the current limitations and investigate potential workarounds. A potential outcome is that, within the scope of this thesis, only a subset of valid Futhark programs will be supported on the WebGPU backend.

More generally, the project will also include an investigation of how suitable WebGPU is as a compilation target for data-parallel languages.

4 Learning Objectives

- Explaining how to translate low level GPU programs from one paradigm to another, in the context of code generation.
- Evaluating the suitability of WebGPU and WGSL as a compilation target for high level programming languages.
- Knowledge of how to systematically performance- and correctness-test compiler backends.
- The competence to work around restrictions or limitations in a target programming model, by devising appropriate semantically equivalent formulations.

References

[1] Alan Baker, Mehmet Oguz Derin, and David Neto. *WebGPU Shading Language, W3C Working Draft*. Dec. 2023. URL: https://www.w3.org/TR/2023/WD-WGSL-20231204/.

[2] *Dawn, a WebGPU implementation*. URL: https://dawn.googlesource.com/dawn/+/refs/heads/main/README.md (visited on 12/19/2023).

[3] DIKU. *The Futhark Programming Language*. URL: https://futhark-lang.org/ (visited on 12/19/2023).

[4] Troels Henriksen. "Design and Implementation of the Futhark Programming Language". PhD thesis. Universitetsparken 5, 2100 København: University of Copenhagen, Nov. 2017.

[5] Troels Henriksen et al. "Futhark: Purely Functional GPU-programming with Nested Parallelism and In-place Array Updates". In: *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI 2017. Barcelona, Spain: ACM, 2017, pp. 556–571. ISBN: 978-1-4503-4988-8. DOI: 10.1145/3062341.3062354. URL: http://doi.acm.org/10.1145/3062341.3062354.

[6] Kai Ninomiya, Brandon Jones, and Jim Blandy. *WebGPU, W3C Working Draft*. Dec. 2023. URL: https://www.w3.org/TR/2023/WD-webgpu-20231206/.

[7] *wgpu*. URL: https://wgpu.rs/ (visited on 12/19/2023).