UNIVERSITY OF OREGON

Viability of a Web Based Data Visualization Software

Valerie Dagley, Professor Hank Childs

cure
CENTER for UNDERGRADUATE RESEARCH and ENGAGEMENT

Introduction

- Web applications (software accessed via the internet) are becoming central to the modern software ecosystem. They allow users to access the software from many different kinds of devices, improving accessibility and portability.
- Historically web applications have been very slow compared to native (desktop) applications, often experiencing a 5-10x slowdown or more. Modern improvements such as Web Assembly (WASM) have improved this, but web applications still struggle with speed even with such improvements.
- The introduction of WebGPU, a new web standard, allows web applications to make use of the modern computer hardware for complex calculations. This enables a massive speed up of web applications in many cases.
- Complex 3D data visualization makes use of difficult calculations which can be optimized significantly using this modern hardware. This makes it an optimal test case for the viability of the WebGPU standard, since the ability to make use of modern hardware on the web will vastly increase the usability of web-based data visualization softwares.

Background

- The modern computer is made up of 2 main processing components: the CPU and the GPU. The CPU is the main processor that runs most applications and systems. The GPU is a specialized processor that has historically been used for 3D graphics rendering, however in recent years it has been used for more varied applications. Notably, the CPU and GPU are different in that the CPU is optimized for doing sequential operations very quickly whereas the GPU is optimized for doing many similar operations in parallel (CPU = 1 complex operation at a time very quickly, GPU = many simpler operations at the same time)
- Web applications are slower than native applications for many reasons. The most relevant reason for our application is the fact that web applications are mostly unable to make use of the GPU. This means that when web applications that need to make use of parallel computations (Ex: neural networks, solving systems of equations, graphics, etc.) are extremely bottlenecked.
- Note: WebGL has existed for around 15 years, allowing web applications to make use of the GPU for graphics-specific programs on the web. However, GPU compute was not feasible.

Research Question

Does the new WebGPU standard enable web applications to utilize the GPU in a general purpose and performant manner? Furthermore, does the use of this new standard allow a real-time data visualization software to be run entirely on the web?

Method

To answer our guiding question, we chose to implement a common data visualization method that utilizes the GPU. Using this implementation, we will create a performance comparison between how this performs when served as part of a web application as compared to being served as a desktop application.

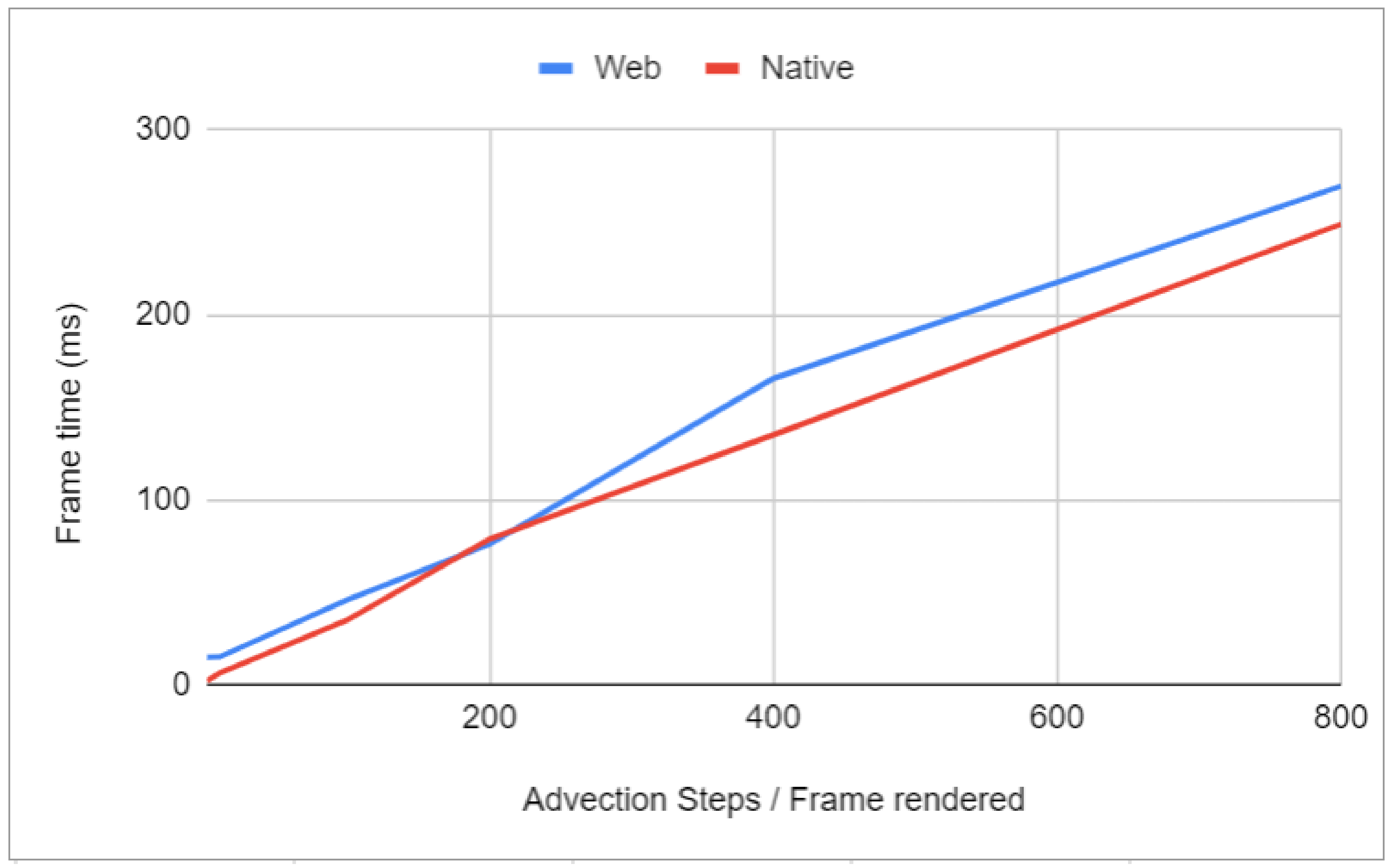
To create this application, we chose to utilize a few modern technologies. First, we used Web Assembly (WASM) which allows compiled languages to be run on the web. This allowed us to use Rust, a modern programming language built around security and performance. Using Rust and WASM, we could create an application that can utilize web technologies and native (desktop) technologies without having to do many target-specific adjustments. This keeps the performance comparison as fair as is possible between the two technologies. Finally, we used these two technologies to implement a program that utilizes WebGPU to perform mass particle advection in a 3D vector field. This data visualization method of placing particles in a vector field and following their paths is extremely parallelizable making it the perfect candidate for an application which utilizes the GPU. We then measured the amount of time it takes to compute and render each frame under increasing stress loads. Each test was run over 10 seconds, taking the average frame time for the result.

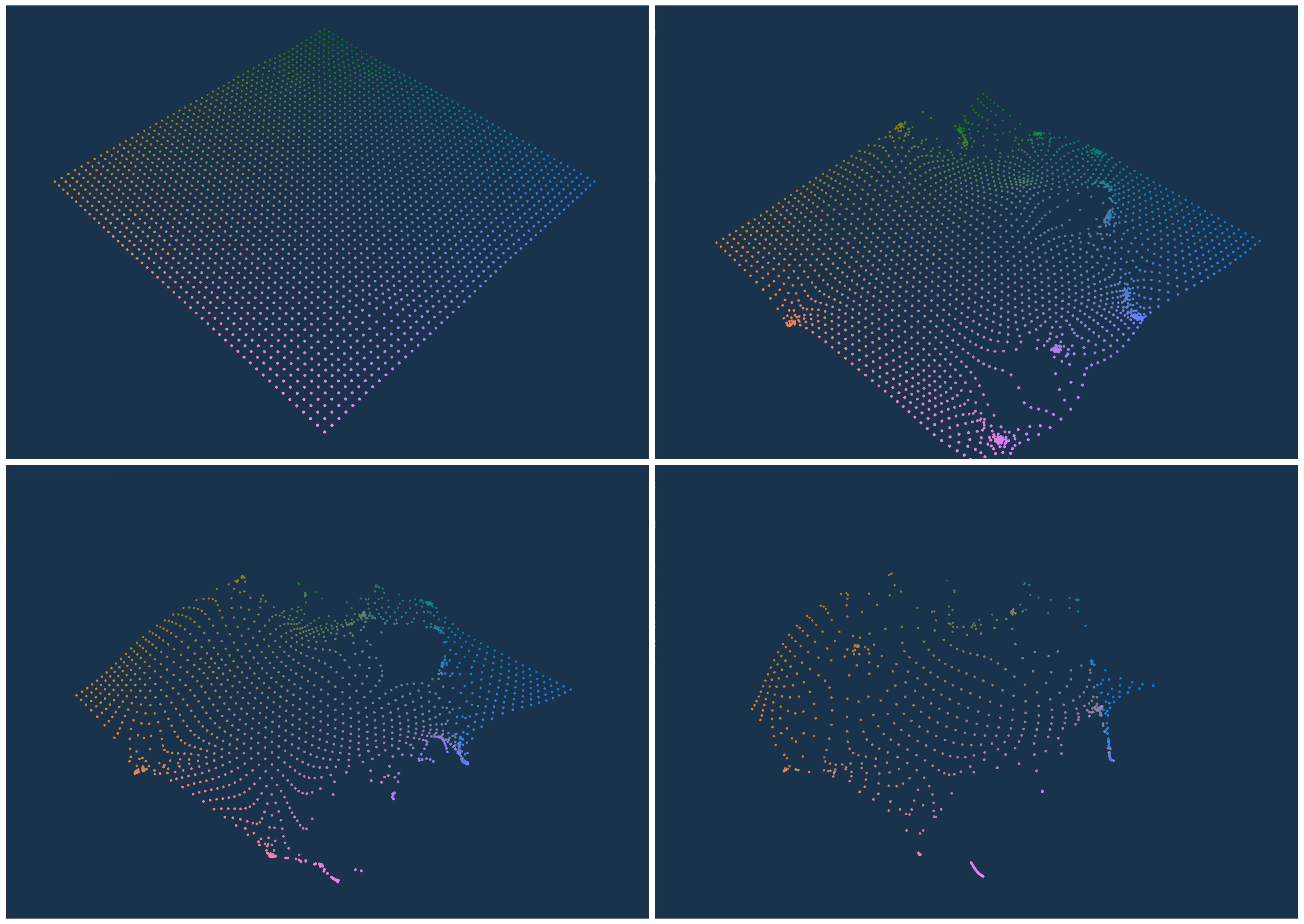
Acknowledgements

- This research was made possible by the University of Oregon Center for Undergraduate Research through their First Year Research Experience grant (FYRE).
- I would also like to thank the assistance of-
    - Professor Hank Childs, who has led me through this research throughout the summer and encouraged me along the way.
    - Ph.D students Manish Mathai and Meghanto Majumder for their interest and collaboration with this project.

Results

- Using our test data, we observed that the application served on the web scaled at an equal rate as the application served natively on the desktop.
- Slowdowns existed due to the overhead of running on the web, but are minor under large GPU loads.
- This shows extreme promise for the future of general-purpose GPU computing on the web.





visualization of a 100 x 100 particle grid as it moves through a given 3D vector field over time.

Project source code can be found at https://github.com/ValerADHD/WebGPU-Particle-Advection

Hank Childs, Manish Mathai, and Meghanto Majumder are all part off the University of Oregon CDUX research group. Information can be found at cdux.cs.uoregon.edu

Research was done through the University of Oregon Center for Undergraduate Research and Engagement (CURE). Information on CURE can be found at urds.uoregon.edu/cure