Visualizing Memory Access in Heterogeneous Systems

Elijah Ingram & Dr. Tyler Allen (Advisor)

College of Computing and Informatics, Parallel and High Performance Computing

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Introduction

- Heterogeneous systems often include multiple different types of processors.
- Almost all modern systems are heterogeneous systems.
- Heterogeneous systems allow tasks to be performed on the processor that's best suited for them. For example, a graphical processing unit (GPU) is *much* better at processes like matrix multiplication and rendering graphics to the screen when compared to a traditional CPU.

Motivations

- In heterogeneous system, data needs to move between the different processors, which is an extremely slow operation.
- If we can find a way to speed up the data transfer between different types of processors, we can increase the overall performance of heterogeneous systems.
- Understanding *how* data moves in heterogeneous systems is a non-trivial task, so we hope to provide an accessible way for anybody to understand how data moves between processors in heterogeneous systems.

Current progress

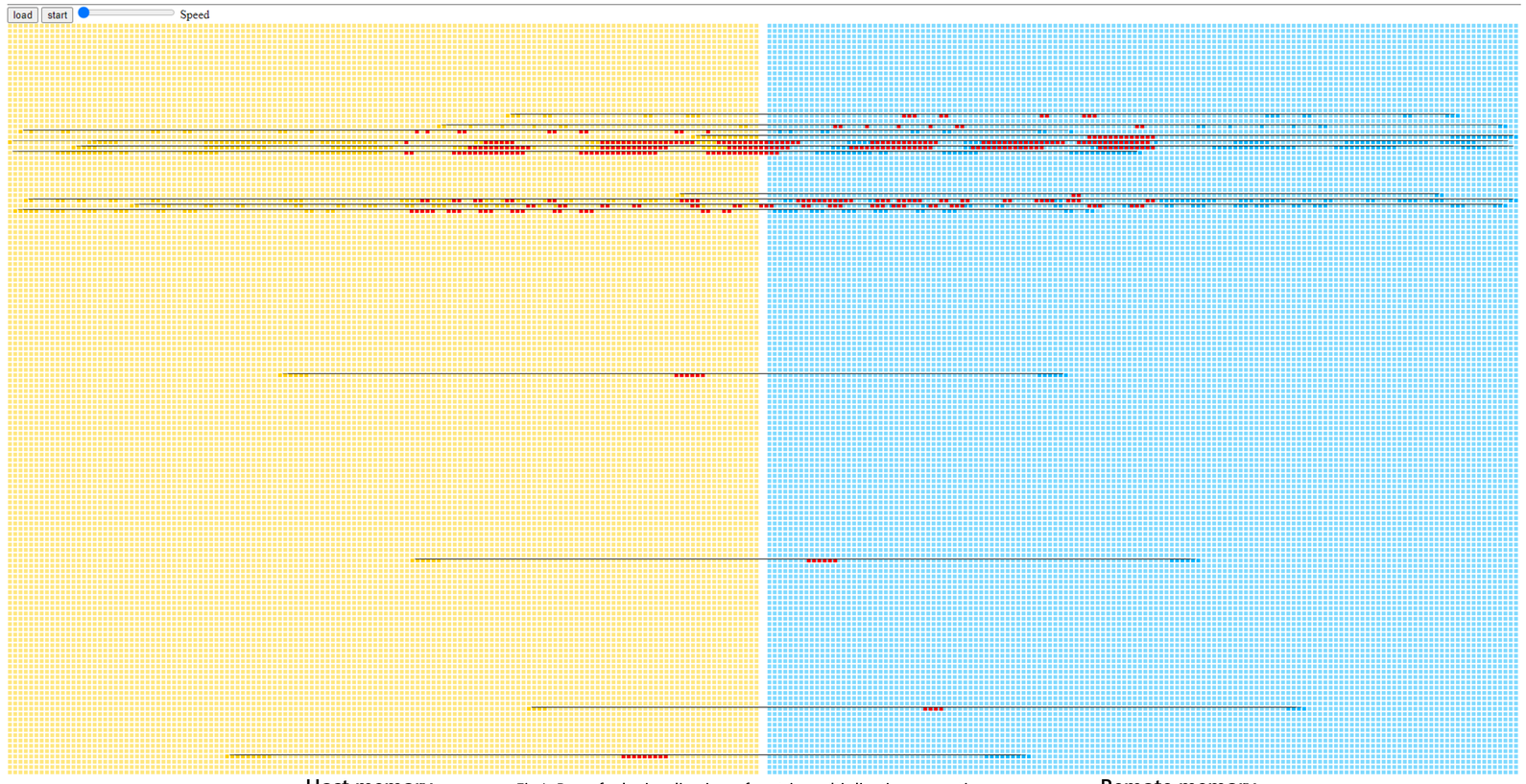

Fig1. Page fault visualization of matrix multiplication operation

Why visualizations

One way we might figure out how to make these systems faster is with visualizations. Visualizations are good for extracting patterns from data sets and providing easily understandable representations of complex processes.

We hope that creating a visualization of data movement in heterogeneous systems will:

- Reveal memory access patterns that provide insights on how optimize and improve the process of data movement across different types of processors/memory
- Provide an **accessible** visual abstraction of data access in complex heterogeneous systems

Current features

The visualization animates data movement between GPU memory and system memory. It is made up of two grids representing two descrete memory devices. The host (GPU) memory is displayed on the left (yellow), while the remote (system) memory is displayed on the right (blue).

Current visualization features include:

- **Opacity based access history:** As each page is accessed, it becomes darker, providing clear visual feedback on what pages in memory have been accessed and how frequently.
- **Visual page batching and data paths:** Memory operations in our data are performed in batches, the animation needs to keep the batching structure intact when visualizing data.
- **Adjustable animation speed:** When exploring larger sets of data, the ability to dynamically change the animation speed is useful for scrubbing through data quickly while not losing the ability to slow down and examine a specific step.
- **Page size normalization:** Some data sets are too large to effectively display on normal sized screens. To solve this, we can normalize the pages into "super pages" to decrease the total amount of cells on the screen.

Future features

Current plans to add:

- **Timeline:** Allow users to scrub through the data to specific points.
- **Interactivity:** Allow users to interactively explore the data.
- **Multi-Device support:** Support the visualization of more than two devices at once.