# Proof-of-Concept for Heterogeneous GPU Computing:
## Lattice-Boltzmann Fluid Simulations with Image Capture Analysis for User-drawn Boundary Conditions

Andrew Dawson*, Shingo Lavine+, Xavier Quinn‡
*: Rensselaer Polytechnic Institute (Troy, NY) +: Brown University (Providence, RI) ‡: Union College (Schenectady, NY)

## Abstract

This project demonstrates a Lattice-Boltzmann Method fluid simulation running on a heterogeneous GPU architecture using captured graphical input from the user. Components for the software stack were identified and deployed on the heterogeneous processor, and python scripts are used to translate information from image analysis to simulation.

## Introduction

What is Heterogeneous Computing?

- Utilizing more than one kind of processor, e.g. CPU and GPU, to solve a problem
- Can have significant performance and power usage advantages over homogeneous systems

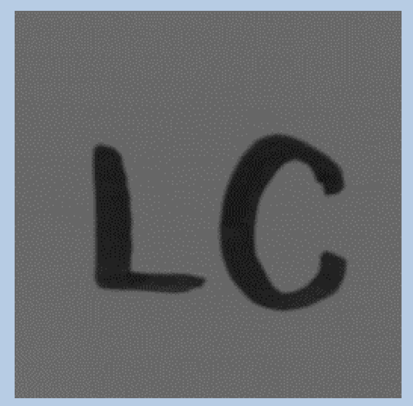
The new Sierra system to be deployed at LLNL as part of the CORAL procurement is an example of a heterogeneous system where IBM Power processors are coupled with Nvidia GPU accelerators. The objective of this project is to explore a heterogeneous system while creating an interesting interactive simulation. The data flow involves the user drawing a shape while an edge-detection algorithm interprets the drawing to define boundary conditions for a fluid simulation.

## Methods and Results

The Jetson TX1 device from Nvidia combines a multi-core ARM CPU with 256 CUDA-capable Maxwell GPU cores. While the board is commonly used for deep learning and computer vision embedded applications such as autonomous vehicles, here the platform is being used as a proxy for a cluster compute node. The Jetson TX1 board was able to support the LBM simulation along with the image capture analysis. Multiple boards were clustered allowing problems to be solved with a distributed parallel algorithm leading to computational speedups.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

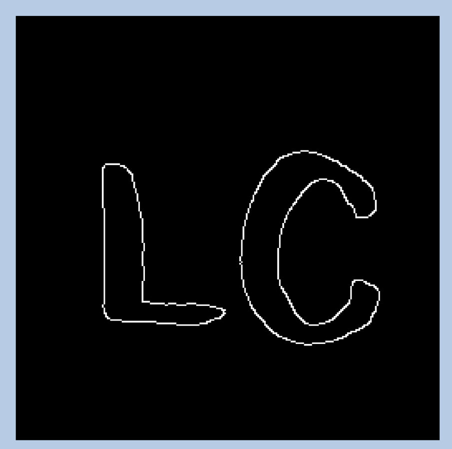
## Simulation Workflow



*User Drawn Image*

### Image Capture Input

Gstreamer[1], a framework for creating various media-based applications, is used to capture an image.



*OpenCV Edge Detected Image*

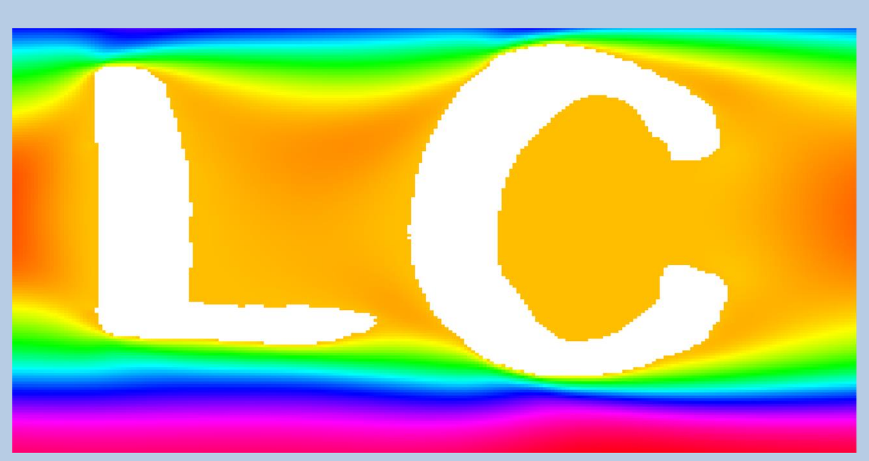
### Edge Detection

Algorithms within OpenCV[2], an open-source computer vision and image recognition library, identify edges within the image

*Python Script Cropped Image and Filled In*

### Image Processing Script

A python script translates the edges generated by OpenCV into boundary conditions readable by sailfish. The python script enters the boundary values and executes the sailfish simulation.

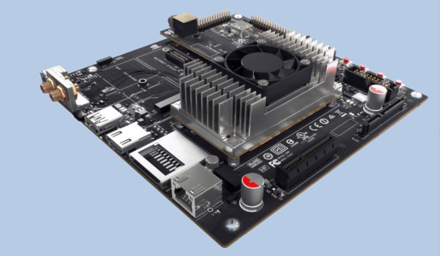


*Sailfish Fluid Simulation*

### Fluid Simulation

Sailfish[3] solves the fluid dynamics equation using a Lattice-Boltzmann Method (LBM). In contrast to more traditional methods (e.g. finite volume), LBM could be more parallelizable on GPU architectures.

User Iteration



*Jetson TX1*

## Discussion

One motivation for this project was to create an environment in which a user could quickly demonstrate fluid flow over rudimentary airfoils. The program allows for simulation flexibility by allowing the user to draw or photograph real objects and could theoretically be expanded to allow for 3D spatial capture as input for simulations. Using image capture input enables users to demonstrate a proof-of-concept prototype with minimal programming.

Furthermore, this project demonstrates the viability of using LBM physics simulations on a CPU/GPU architecture. Future high performance computing resources will have heterogeneous architectures and LBM algorithms may enable better utilization of the capabilities.

Next Steps:
- CPU vs GPU/CPU Performance Benchmarking
- Developing a GUI
- Clustering additional nodes
- Strong and weak scaling studies
- 3D Spatial Capture

***Acknowledgements:***
Joseph Han (Mentor), Thomas Bennett, Ian Lee, David Fox, Pam Hamilton, Kim Cupps, Giuseppe Di Natale: Lawrence Livermore National Laboratory (Livermore Computing)

***References:***
1: https://gstreamer.freedesktop.org/
2: http://opencv.org/
3: http://dx.doi.org/10.1016/j.cpc.2014.04.018

***Review and Release Number:***
***LLNL-POST-699462***