Accelerating Shallow Water Flow and Mass Transport Using Lattice Boltzmann Methods on GPUs

Kevin Tubbs

Dell | Global HPC Solutions Engineering
Introduction
Background and Overview

Role of CFD

- Improve Environmental Predictions
- Used to Improve Decision and Policy Making
- Test and Evaluate Preventative Methods
Dell Global HPC Solutions Engineering

Performance Analysis & Characterization
- Node level
- Cluster level

Power Consumption and Energy Analysis

Benchmark and Application Analysis
- Industry Standard Benchmarks & Apps: HPL, NPB, NAMD
- Focus Area Specific: CFD, Climate & Weather, Molecular Dynamics, etc.
Why Accelereyes? → Programmability

- Faster
  - Writing Kernels
  - Using Libraries

- Slower
  - Instruction Set Optimization
  - Compiler Directives

Time-Consuming vs. Easy-to-use

ArrayFire
C, C++, Fortran for CUDA & OpenCL
Lattice Boltzmann Methods
Lattice Boltzmann Method Overview:

- Modeling Shallow Water and Mass Transport

**Shallow Water Equation**

\[
\frac{\partial h}{\partial t} + \frac{\partial (hu_i)}{\partial x_i} = 0
\]

\[
\frac{\partial (hu_i)}{\partial t} + \frac{\partial (hu_iu_j)}{\partial x_i} = -g \frac{\partial}{\partial x_i} \left( \frac{h^2}{2} \right) + \nu \frac{\partial^2 (hu_i)}{\partial x_j \partial x_j} + F_i
\]

**Advection Dispersion Equation for Solute Concentration**

\[
\frac{\partial (hC)}{\partial t} + \frac{\partial (hu_jC)}{\partial x_j} = \frac{\partial}{\partial x} \left( D_{ij} h \frac{\partial C}{\partial x_j} \right) + F_C
\]

**LB Numerical Formulation**

\[
\frac{\partial f}{\partial t} + \mathbf{c} \cdot \nabla f = -\frac{1}{\tau} (f - f^{(eq)})
\]
Background: LBM

- Boltzmann equation
  \[
  \frac{\partial f}{\partial t} + \mathbf{e} \cdot \nabla f = -\frac{1}{\tau} (f - f^{(eq)})
  \]

- Particle distribution function \( f^*(\mathbf{x}, \mathbf{e}, t) \) \( g(\mathbf{x}, \mathbf{e}, t) \)

- EDF
  \[
  f^{eq}_\alpha = h \omega_\alpha \left( \frac{c_s^2}{e^2} + \frac{\mathbf{u} \cdot \mathbf{e}_\alpha}{e^2} + \frac{3}{2} \left( \frac{\mathbf{u} \cdot \mathbf{e}_\alpha}{e^2} \right)^2 - \frac{1}{2} \frac{\mathbf{u} \cdot \mathbf{u}}{e^2} \right) , \alpha > 0
  \]
  \[
  f^{eq}_0 = h - \sum_{\alpha > 0} f^{eq}_\alpha
  \]

- \( g^{eq}_\alpha = C \omega_\alpha \left( \frac{c_{sa}^2}{e^2} + \frac{h \mathbf{u} \cdot \mathbf{e}_\alpha}{e^2} + \frac{3}{2} \left( \frac{h \mathbf{u} \cdot \mathbf{e}_\alpha}{e^2} \right)^2 - \frac{1}{2} \frac{h \mathbf{u} \cdot h \mathbf{u}}{e^2} \right) , \alpha > 0
  \]
  \[
  g^{eq}_\alpha = hC - \sum_{\alpha > 0} g^{eq}_\alpha
  \]

- LBM Grid

\[\begin{align*}
\mathbf{e}_\alpha &= \begin{cases} 
(0,0), & \alpha = 0 \\
\sqrt{2}e \left[ \cos \left( \frac{\alpha - 1}{4} \right) \pi, \sin \left( \frac{\alpha - 1}{4} \right) \pi \right], & \alpha = 5,6,7,8 \\
e \left[ \cos \left( \frac{\alpha - 1}{4} \right) \pi, \sin \left( \frac{\alpha - 1}{4} \right) \pi \right], & \alpha = 1,2,3,4
\end{cases}
\end{align*}\]
### Background: LBM (cont.)

- **Zeroth Moment**
  
  \[
  h = \sum_{\alpha} f_\alpha \\
  hC = \sum_{\alpha} g_\alpha
  \]

- **First Moment**
  
  \[
  hu_i = \sum_{\alpha} e_{\alpha i} f_\alpha \\
  Chu_i = \sum_{\alpha} e_{\alpha i} g_\alpha
  \]

- **LBM evolves in a two-step process**
  
  - **Step 1. Collision**
    
    \[
    f_\alpha (x, t) = f_\alpha (x, t) - \frac{1}{\tau} \left( f_\alpha (x, t) - f_{\alpha eq} (x, t) \right) \\
    + \frac{\Delta t}{6e^2} e_{\alpha i} F_i (x, t)
    \]

  - **Step 2. Streaming**
    
    \[
    f_\alpha (x + e_\alpha \Delta t, t + \Delta t) = f_\alpha (x, t)
    \]
Results:

1) Model Validation
2. Simulation Results
2D – Dam Break: Backward Facing Step

- **Domain:** 12 m x 12 m
- **Dam Break Initial Conditions**
  - Upstream: water depth 5 m
  - Downstream: water depth 1m
- **BC:** Free – Slip at walls for flow

- **t = 0.5 s**
2D Partial Dam Break

- Domain: 200 m x 200 m
- Dam Break Initial Conditions
  - Upstream: water depth 10m
  - Downstream: water depth 5m
- BC: Free – Slip at walls for flow, Impermeable for concentration

Water Depth Surface  \( t = 7.2 \) s

Velocity Vector Field  \( t = 7.2 \) s
2D – Continuous Release Mass Transport

Problem Setup
- Domain: Infinite Shallow Water Pool
- Constant Water Depth 1 m

\[ \tau_s = \frac{1}{2} + \left( 12 \left( \tau_A - \frac{1}{2} \right) \right)^{-1} \]

\[ c_{sij}^2 = \frac{hD_{ij}}{\Delta t \left( \tau_A - \frac{1}{2} \right)} \]

Velocity Anisotropic Dispersion

\[ D_{xx} = h|u|\kappa_L \quad D_{yy} = h|u|\kappa_T \quad \kappa_L / \kappa_T = 10 \]
2D – Dam Break with Mass Transport

- Domain: 800m x 200m
- Dam Break with Transport Initial Conditions
  - Upstream: water depth 10m
  - Downstream: water depth 5m
  - BC: Free – Slip at walls for flow
- Time Snapshots
  - t = 10 s
  - t = 30 s
  - t = 60 s
  - t = 90 s
2D – Dam Break with Mass Transport

- Domain: 800 m x 200 m
- Dam Break with Transport Initial Conditions
  - Upstream: water depth 10m, Concentration 0.7
  - Downstream: water depth 5m, Concentration 0.2
  - BC: Free – Slip at walls for flow, Impermeable for concentration

- Time Snap shots
  - t = 10 s
  - t = 30 s
  - t = 60 s
  - t = 90 s

\[ D_{ij} = \left( k_L |\mathbf{u}| \delta_{ij} + (k_L - k_T) \frac{u_i u_j}{|\mathbf{u}|} \right) \frac{h \sqrt{g}}{C_z} \]

- \( k_L = 5.93 \)
- \( k_T = 0.23 \)
2D – Dam Break: Backward Facing Step

Multiple-relaxation-time Lattice Boltzmann Model

Execution Time (s)

Grid Size

- Jacket GPU
- MATLAB® CPU

Jacket: MATLAB® on the GPU

Great Science

Great Power
2D – Continuous Release Mass Transport

Two-relaxation-time Lattice Boltzmann Model

Execution Time (s)

Grid Size

- 451x151
- 901x301
- 1801x601
- 2401x801
- 3001x1001
- 3601x1201

- Jacket GPU
- MATLAB® CPU

18x

Jacket: MATLAB® on the GPU

Great Science
Great Power
2D – Dam Break with Mass Transport

MRT-LBM for shallow water and TRT-LBM for transport

Execution Time (s)

- 40x101
- 80x101
- 160x401
- 240x601
- 320x801
- 360x101

Grid Size

Jacket GPU
MATLAB® CPU

22x
Summary

• LBM is a flexible algorithm for various CFD simulations

• Tools from Accelereyes allow rapid development and validation of LBM based CFD algorithms

• The GPU accelerated LBM algorithm achieved maximum of 24X speedup on GPU based architectures.

• The GPU parallel performance depends on size of simulation.
Resources

- Blogs
- Whitepapers
- HPC Advisor Online

- www.dell.com/gpu
- www.dell.com/hpc
- www.hpcatdell.com
- www.DellHPCSolutions.com
Related Work

Publications, Proceedings & Presentations:

**Publications**
http://blog.accelereyes.com/blog/2010/12/01/lattice_boltzmann_model/

**Presentations**

**Proceedings**
Questions