Fluid Dynamics Simulation on the Merrimac Streaming Supercomputer

Gurjeet Singh (CME), Jung Ho Ahn (EE), Mattan Erez (EE), Abhishek Das (EE), Prof. Eric Darve (ME)

**Background**

StreamCDP is a transport-advection equation solver which is used within the jet engine simulation routines. This partial differential equation governs two scalars in the system: the mixture fraction and the reaction progress. Reaction progress variable indicates, in percentage, the progress of the reaction. Reaction progress however is not a conserved variable. These scalars are required to determine the state variables.

**StreamCDP : Challenges - The WENO scheme**

- Following equation governs the mixture fraction:
  \[
  \frac{\partial z}{\partial t} + \frac{\partial}{\partial x} \left( \alpha z \right) = 0
  \]
- Following equation governs the reaction progress:
  \[
  \frac{\partial c}{\partial t} + \frac{\partial}{\partial x} \left( c \right) = 0
  \]

**Challenge and solution**

- The state variables are highly nonlinear functions of the scalars.
- This puts a very stringent restriction on the numerical method.
- Even small spurious oscillations can result in drastically incorrect state variables.

- WENO (Weighted Essentially Non-Oscillatory) is well-suited for this application.
- Computational region is an unstructured grid.
- Scalars are calculated at the cell volumes.
- The scheme is second order, that is, the value of the scalar at each iteration depends on the scalar values of its neighbors and neighbors of neighbors.

**StreamSPAS**

StreamSPAS implements the computation of \( Y = AX \) where \( A \) : sparse matrix, \( X \& Y \) : dense vectors. StreamSPAS also provides a test suite of sparse matrices corresponding to \( p \)-order finite element discretization using continuous Lagrange elements.

**Efficiency of the computation**

- The calculation must be organized to maximize the arithmetic operations with respect to data transfer in the kernel.
- The ratio of arithmetic operations to data transfer of the cell based computations are dependent on the number of neighbors of each cell.
- The face based calculation has a constant number of arithmetic operations with respect to data transfer.

**Simulation Results**

The kernels perform all the arithmetic operations of the calculation. The schedules below track the operations within five iterations of the kernels. The horizontal axis marks time in number of cycles. The vertical axis marks the functional units in Merrimac.

**Algorithm**

- Calculate scalar coefficients for each cell
- Calculate the gradient based on neighboring cell variables
- Calculate the residual based on scalar coefficients
- Compute flux interchange between the cells
- Update the residual calculated above
- Calculate minimum residual
- Update the variables

- For each timestep:
  - Loop over faces:
    - Gather 2 element states
    - Compute flux terms
    - Store fluxes to memory
  - Again, loop over faces and calculate the residual due to the other
  - Find the maximum residual, if it is within tolerance, quit
  - Update the scalar values and re-iterate.

**Efficiency of the computation**

- The calculation must be organized to maximize the arithmetic operations with respect to data transfer in the kernel.
- The ratio of arithmetic operations to data transfer of the cell based computations are dependent on the number of neighbors of each cell.
- The face based calculation has a constant number of arithmetic operations with respect to data transfer.

**WENO scheme**

- Following equation governs the mixture fraction:
  \[
  \frac{\partial z}{\partial t} + \frac{\partial}{\partial x} \left( \alpha z \right) = 0
  \]
- Following equation governs the reaction progress:
  \[
  \frac{\partial c}{\partial t} + \frac{\partial}{\partial x} \left( c \right) = 0
  \]

**Challenge and solution**

- The state variables are highly nonlinear functions of the scalars.
- This puts a very stringent restriction on the numerical method.
- Even small spurious oscillations can result in drastically incorrect state variables.

- WENO (Weighted Essentially Non-Oscillatory) is well-suited for this application.
- Computational region is an unstructured grid.
- Scalars are calculated at the cell volumes.
- The scheme is second order, that is, the value of the scalar at each iteration depends on the scalar values of its neighbors and neighbors of neighbors.

**StreamSPAS**

StreamSPAS implements the computation of \( Y = AX \) where \( A \) : sparse matrix, \( X \& Y \) : dense vectors. StreamSPAS also provides a test suite of sparse matrices corresponding to \( p \)-order finite element discretization using continuous Lagrange elements.

**Efficiency of the computation**

- The calculation must be organized to maximize the arithmetic operations with respect to data transfer in the kernel.
- The ratio of arithmetic operations to data transfer of the cell based computations are dependent on the number of neighbors of each cell.
- The face based calculation has a constant number of arithmetic operations with respect to data transfer.

**Algorithm**

- Calculate scalar coefficients for each cell
- Calculate the gradient based on neighboring cell variables
- Calculate the residual based on scalar coefficients
- Compute flux interchange between the cells
- Update the residual calculated above
- Calculate minimum residual
- Update the variables

- For each timestep:
  - Loop over faces:
    - Gather 2 element states
    - Compute flux terms
    - Store fluxes to memory
  - Again, loop over faces and calculate the residual due to the other
  - Find the maximum residual, if it is within tolerance, quit
  - Update the scalar values and re-iterate.