Hydro-dynamics in CUDA
Integration with FORTRAN
Contents

- Context
- What is CUDA?
- Example: matrix multiplication
- Example: Thomas in CUDA
- Correctness Thomas
- Achieved results & future prospects
- Conclusion
- Questions
Me
- Jonathan van der Wielen
- Bachelor Technical Computer Science
  - High performance

Goal:
- Use CUDA for performance gains in MOHID
Context

- Maretec, IST
  - Marine Environment & Technology Center
  - Numerical models for coastal, ocean and land areas (MOHID)
- FORTRAN
Contents

- Context
- What is CUDA?
- Example: matrix multiplication
- Example: Thomas in CUDA
- Correctness Thomas
- Achieved results & future prospects
- Conclusion
- Questions
What is CUDA?

- NVIDIA technology
- Run general purpose applications on GPU
- Massively parallel data processing
- Program in C++ or FORTRAN
What is CUDA?
What is CUDA?

- Streaming Multiprocessors (SM’s)
- Several types of memory
- Grids, blocks, threads
- All threads execute the same code, but with different data
What is CUDA?
Contents

- Context
- What is CUDA?
- Example: matrix multiplication
- Example: Thomas in CUDA
- Correctness Thomas
- Achieved results & future prospects
- Conclusion
- Questions
// CPU implementation (C++)
void MatrixMulHost(float* a, float* b, float* c, int width)
{
    for (int i = 0; i < width; i++)
    {
        for (int j = 0; j < width; j++)
        {
            float temp = 0;
            for (int k = 0; k < width; k++)
            {
                temp +=
                a[i * width + k] * b[k * width + j];
            }
            c[i * width + j] = temp;
        }
    }
}
// CUDA implementation (host) (1200x faster than CPU)
void MatrixMul(float *a, float *b, float *c, int width) {
    float *devA = 0, *devB = 0, *devC = 0;
    int size = sizeof(float) * width * width;
    cudaMalloc((void**)&devA, size);
    cudaMalloc((void**)&devB, size);
    cudaMalloc((void**)&devC, size);
    cudaMemcpy(devA, a, size, cudaMemcpyHostToDevice);
    cudaMemcpy(devB, b, size, cudaMemcpyHostToDevice);
    cudaMemcpy((void**)&devA, a, size, cudaMemcpyHostToDevice);
    cudaMemcpy((void**)&devB, b, size, cudaMemcpyHostToDevice);

    dim3 block(16, 16);
    dim3 grid(ceil((float)width / block.x), ceil((float)width / block.y));
    KernelMatrixMul<<<grid, block>>>(devA, devB, devC, width);
    cudaMemcpy(c, devC, size, cudaMemcpyDeviceToHost);
}
// CUDA implementation (device)
__global__ void KernelMatrixMul(float *a, float *b, float *c, int width) {
    int i = blockIdx.y * blockDim.y + threadIdx.y;
    int j = blockIdx.x * blockDim.x + threadIdx.x;

    if(i < width && j < width)
    {
        float temp = 0;

        for(int k = 0; k < width; k++)
        {
            temp += a[i * width + k] * b[k * width + j];
        }

        c[i * width + j] = temp;
    }
}
Contents

- Context
- What is CUDA?
- Example: matrix multiplication
- Example: Thomas in CUDA
- Correctness Thomas
- Achieved results & future prospects
- Conclusion
- Questions
Example: Thomas in CUDA

- Tridiagonal solver for implicit schemes
- Sweep back and forth over a matrix:

\[
\begin{bmatrix}
  b_1 & c_1 & 0 \\
  a_2 & b_2 & c_2 \\
  & a_3 & b_3 & \ddots \\
  & & \ddots & a_{n-1} & c_{n-1} \\
  0 & & \cdots & a_n & b_n
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_{n-1} \\
  x_n
\end{bmatrix}
=
\begin{bmatrix}
  d_1 \\
  d_2 \\
  \vdots \\
  d_{n-1} \\
  d_n
\end{bmatrix}.
\]

- Top left to bottom right is a dimension
Example: Thomas in CUDA

- How to parallelize?
  - Algorithm itself cannot be parallelized
  - Algorithm is executed for every \([I,J]\) cell for \([K]\)
    - These operations are independent
Example: Thomas in CUDA

- My approach
  - Write Thomas in CUDA / C++, independent of MOHID
  - Realize binding between FORTRAN and C
  - Integrate CUDA code into MOHID
  - Run performance and correctness benchmarks
Example: Thomas in CUDA

- Program flow
Correctness Thomas

- Some deviation expected due to rounding errors
- Maximum deviation in water level after one hour simulation: 0.0000000618%
- Higher deviation in velocity; influence on end results is negligible
Correctness Thomas

Water level at point [25, 31]

Water level (m)

Simulation time (s)

CUDA
FORTRAN opt.
Achieved results Thomas

Tejo test case - Total execution time

Execution time (s)

- FORTRAN
- CUDA

- Thomas code
- Other code
Achieved results Thomas

- Most overhead: data transfer
- How to minimize overhead?

![Bar chart showing execution time for X, Y, and Z dimensions with different execution modes: Copy, Transpose, Thomas.]
Future prospects

Rough speed up estimation for ModuleHydroDynamic

Execution time

- Parallelizable
- Estimated overhead

FORTTRAN

CUDA Z-based
Contents

- Context
- What is CUDA?
- Example: matrix multiplication
- Example: Thomas in CUDA
- Correctness Thomas
- Achieved results & future prospects
- Conclusion
- Questions
CUDA gives interesting performance gains for 2D or 3D numerical models

- Reusable binding between FORTRAN and CUDA
- Switching to CUDA can be done in phases
- Using CUDA brings extra complexity
- Getting the best performance requires running a whole application in CUDA
Contents

- Context
- What is CUDA?
- Example: matrix multiplication
- Example: Thomas in CUDA
- Correctness Thomas
- Achieved results & future prospects
- Conclusion
- Questions
References

- MOHID:
  - http://mohid.codeplex.com
  - http://mohid.wordpress.com

- CUDA:
  - David B. Kirk & Wen-mei W. Hwu, 2010, *Programming Massively Parallel Processors*
  - Jason Sanders & Edward Kandrot, 2011, *CUDA by Example*
Questions