Acceleration of CFD using graphics hardware

Graham Pullan

29 January 2008

I've added some notes that weren't on the original slides to help readers of the online pdf version.

Part 1: CPUs and GPUs

Moore's Law

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue."

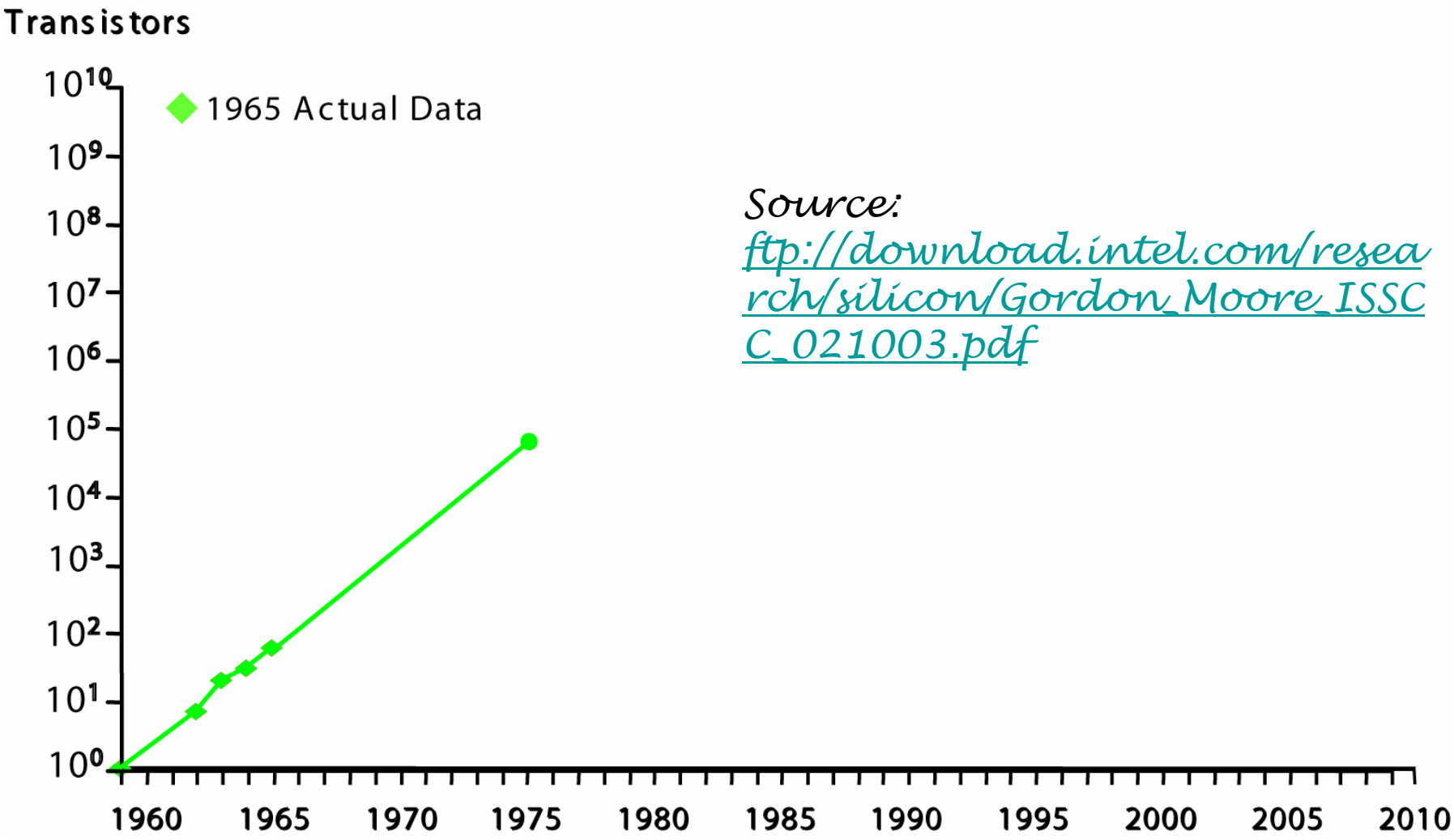
Gordon Moore (Intel), 1965

Moore's Law

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue."
Gordon Moore (Intel), 1965

"*OK, maybe a factor of two every* ***two*** *years*."
Gordon Moore (Intel), 1975 [paraphrased]

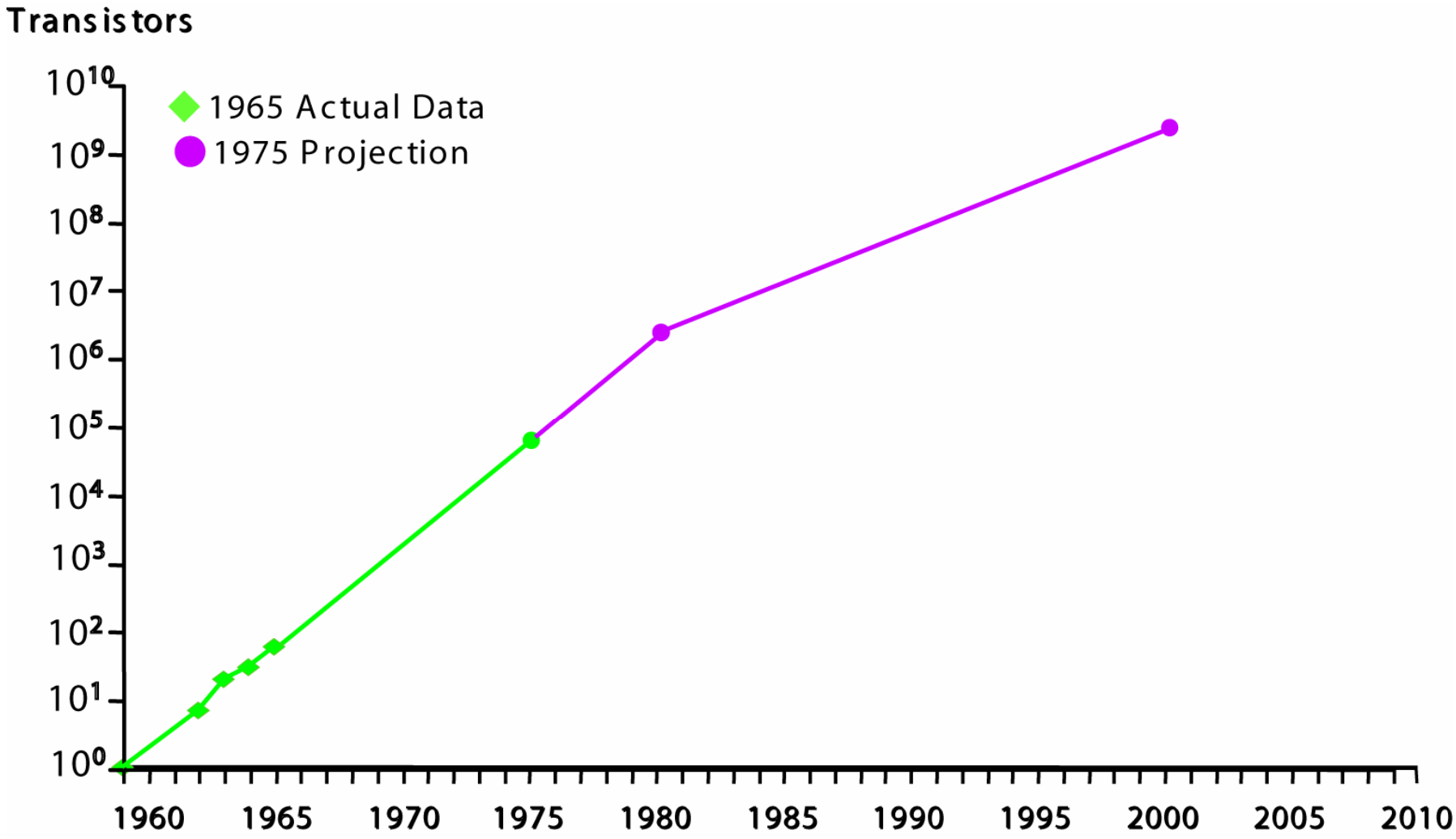
Was Moore right?



Source: *ftp://download.intel.com/research/silicon/Gordon_Moore_ISSCC_021003.pdf*

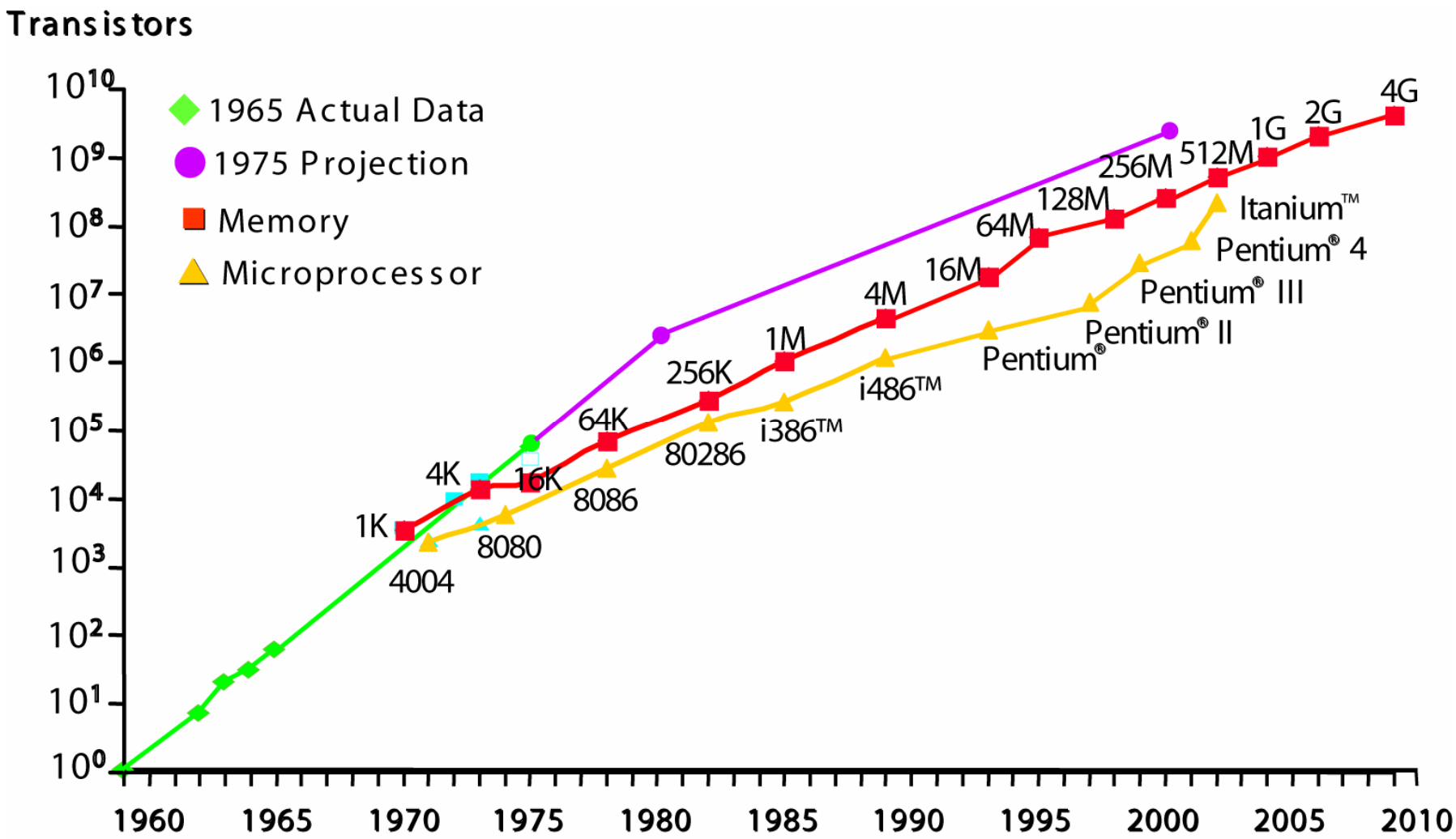
Source: Intel

Was Moore right?



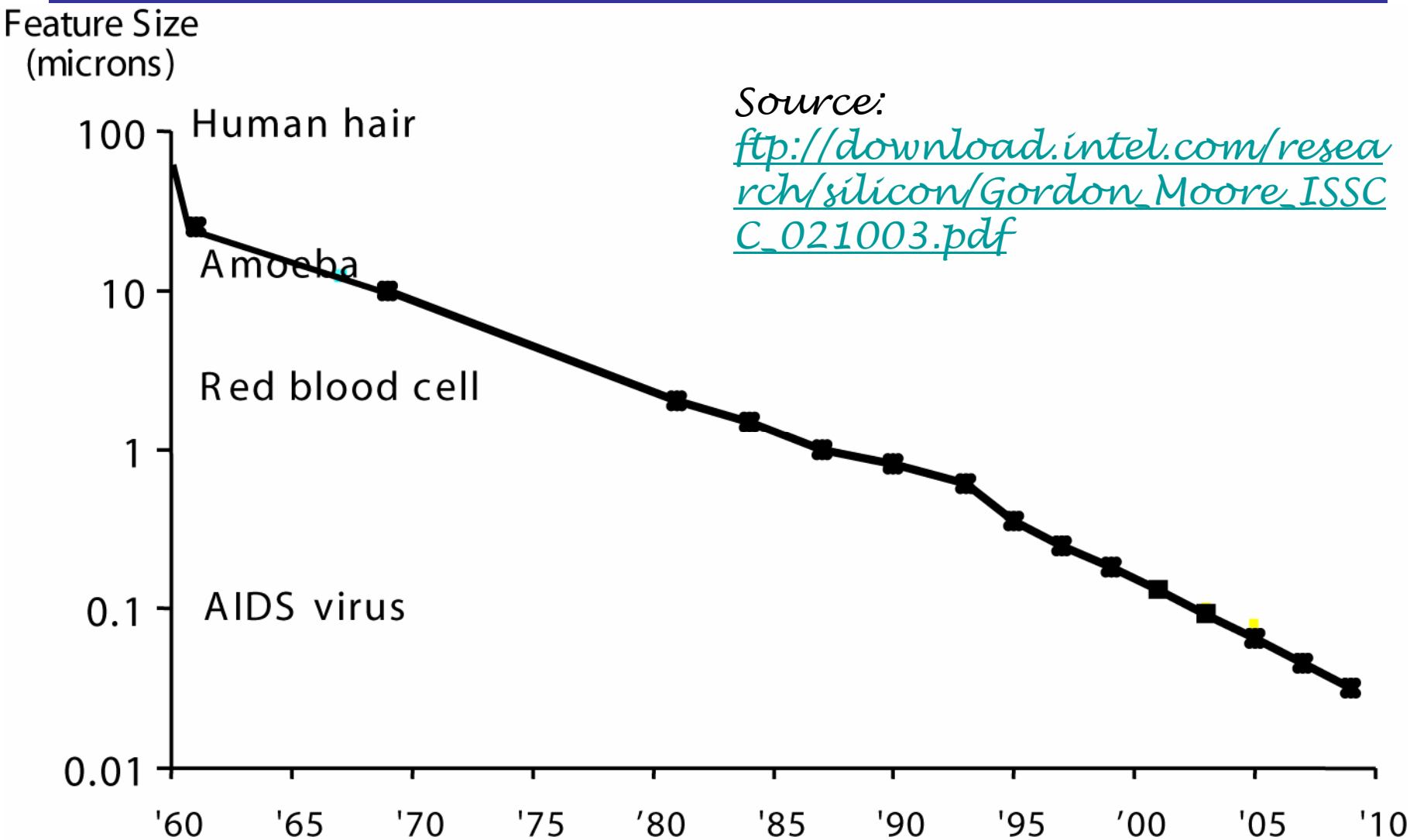
Source: Intel

Was Moore right?



Source: Intel

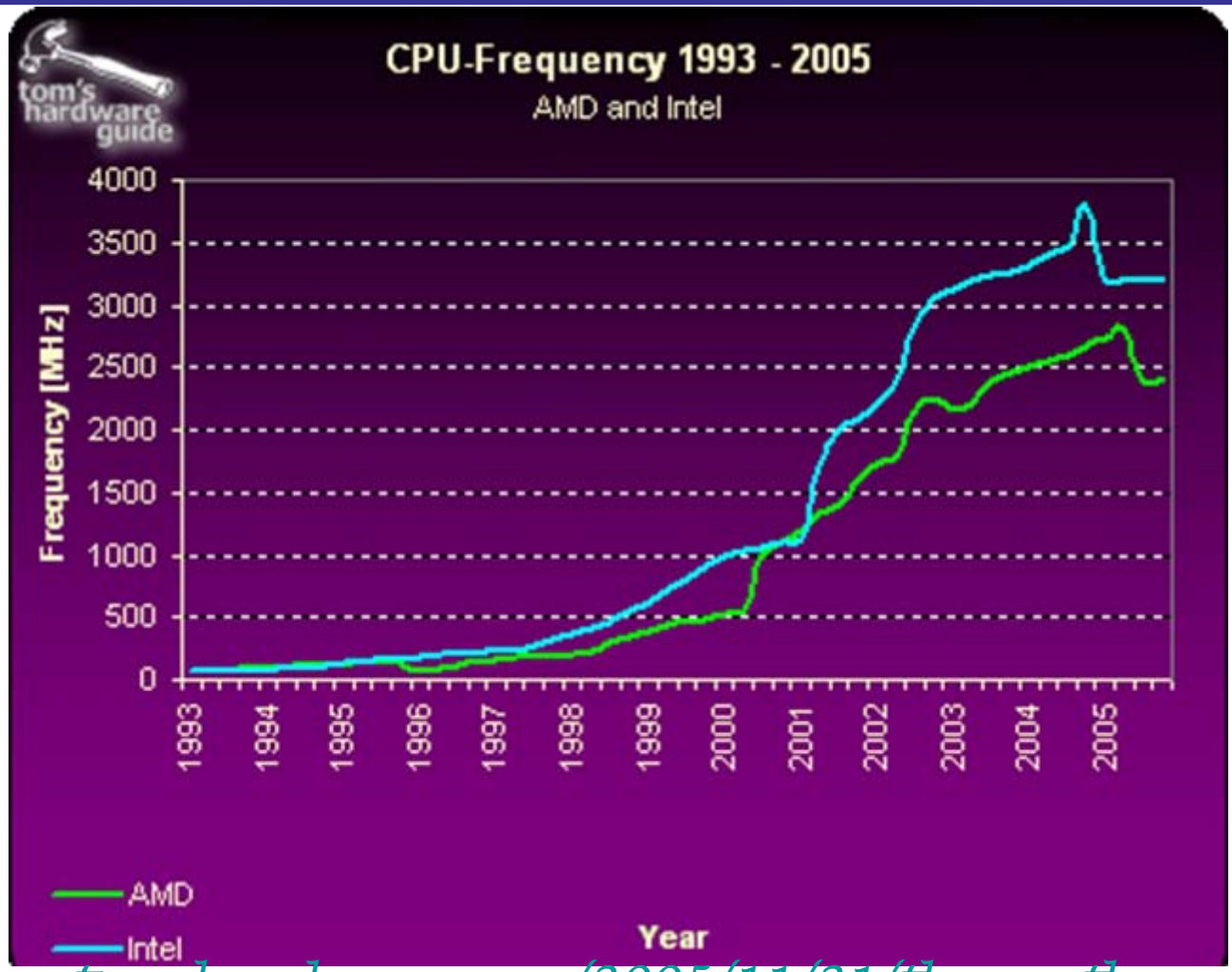
Feature size



Source:
ftp://download.intel.com/research/silicon/Gordon_Moore_ISSCC_021003.pdf

Source: Intel

Clock speed



Source: http://www.tomshardware.com/2005/11/21/the_mother_of_all_cpu_charts_2005/index.html

Power – the Clock speed limiter?

- 1 GHz CPU requires ≈ 25 W
- 3 GHz CPU requires ≈ 100 W

Power – the Clock speed limiter?

- 1 GHz CPU requires ≈ 25 W
- 3 GHz CPU requires ≈ 100 W

"*The total of electricity consumed by major search engines in 2006 approaches 5 GW*." – Wired / AMD

Source:
http://www.hotchips.org/hc19/docs/keynote2.pdf

What to do with all these transistors?

Parallel computing

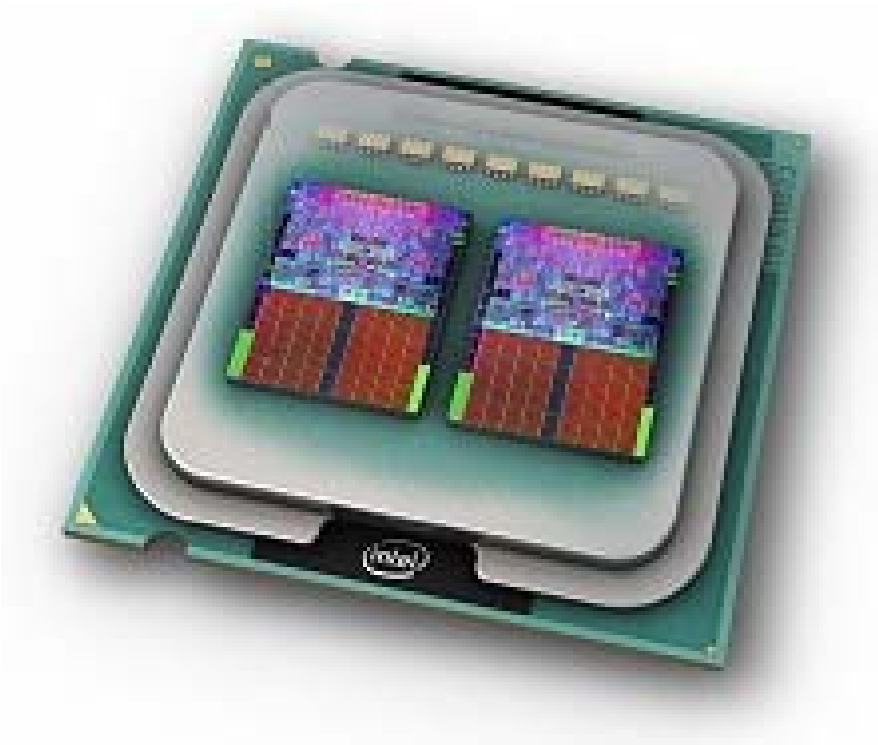
Multi-core chips are either:

- Instruction parallel
  (Multipile Instruction, Multiple Data) – MIMD

or

- Data parallel
  (Single Instruction, Multiple Data) – SIMD

Today's commodity MIMD chips: CPUs



Intel Core 2 Quad

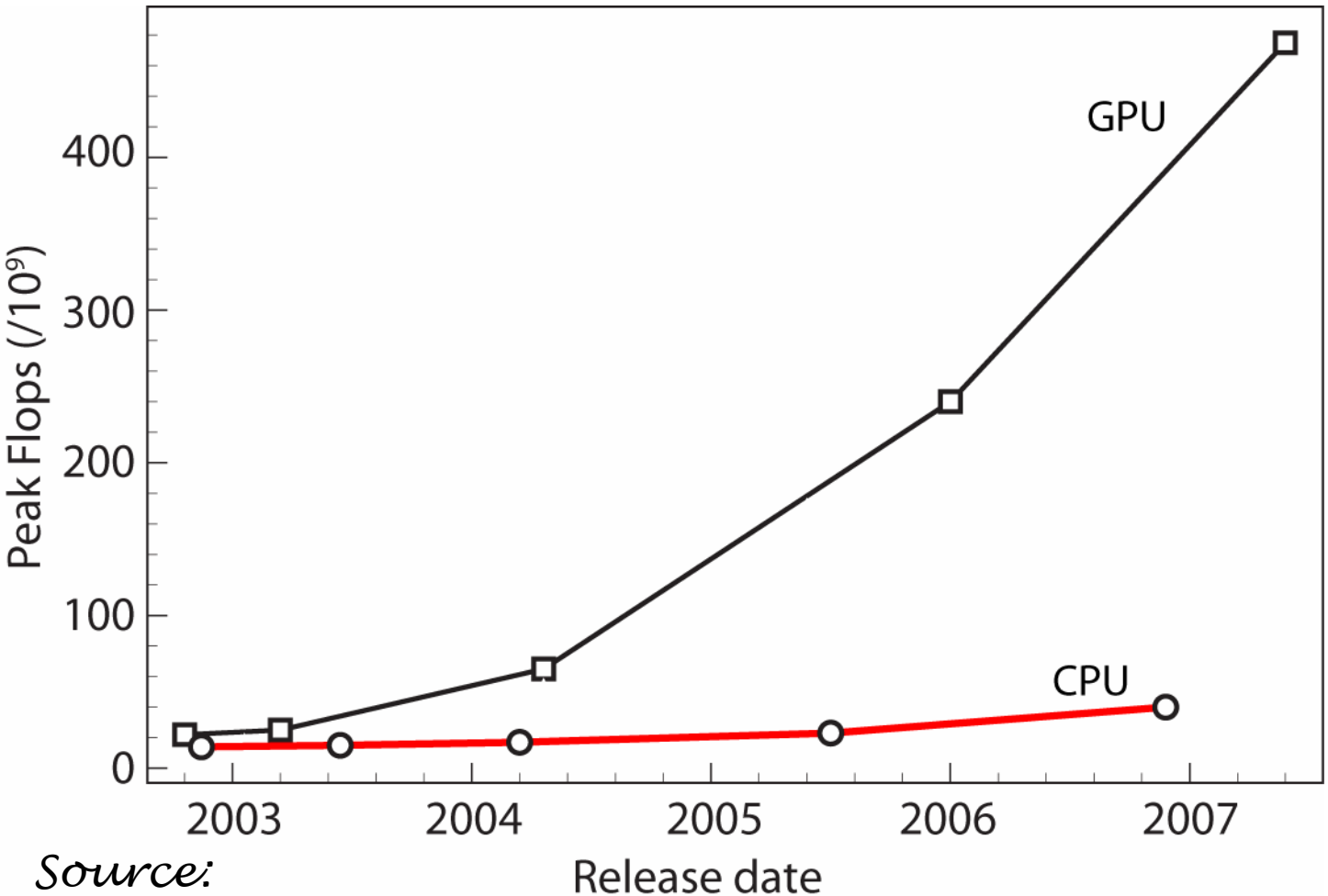
- 4 cores
- 2.4 GHz
- 65nm features
- 582 million transistors
- 8MB on chip memory

Today's commodity SIMD chips: GPUs

NVIDIA 8800 GTX

- 128 cores
- 1.35 GHz
- 90nm features
- 681 million transistors
- 128kB on chip memory
- 768MB on board memory

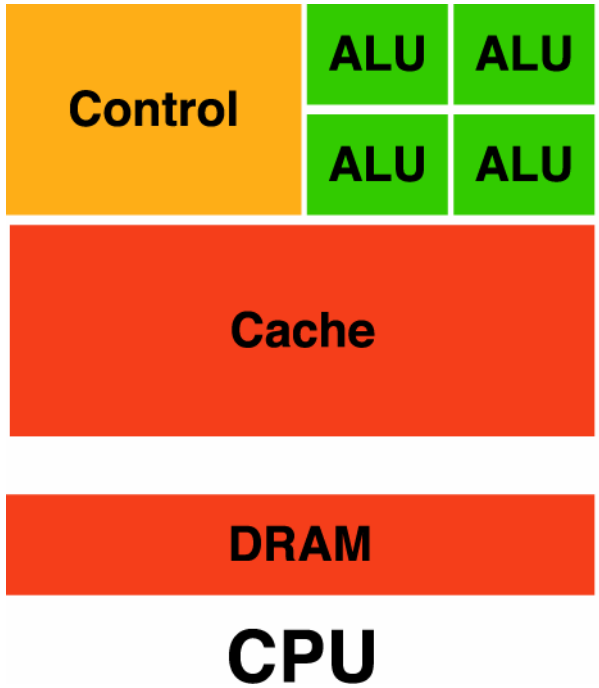
CPUs vs GPUs

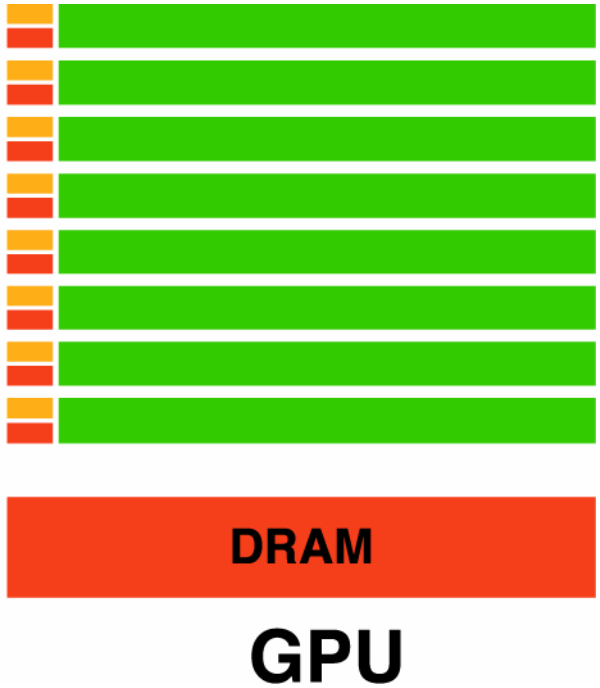


Source:
http://www.eng.cam.ac.uk/~gp10006/research/Brandvik_Pullan_2008a_DRAFT.pdf

CPUs vs GPUs

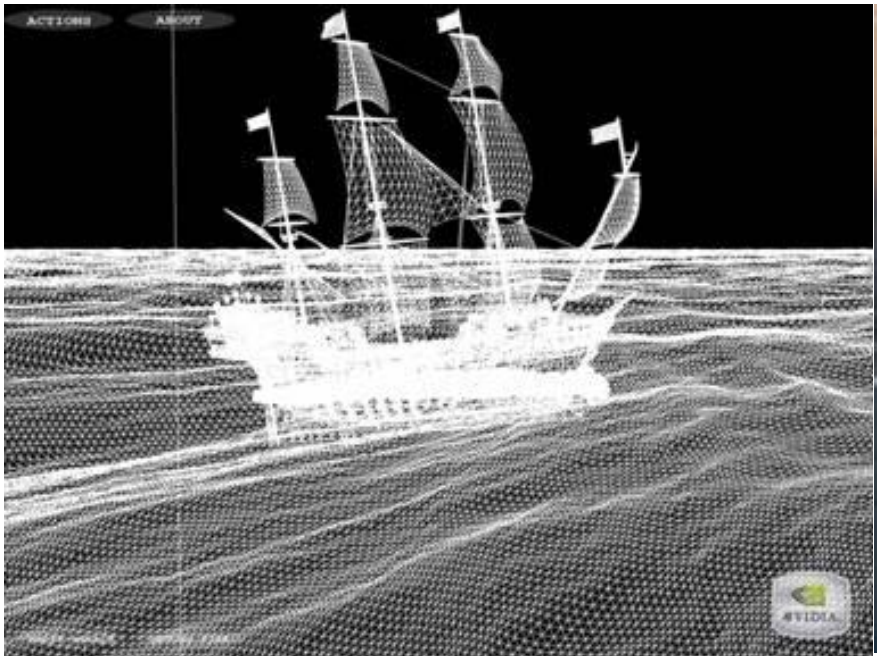
Transistor usage:





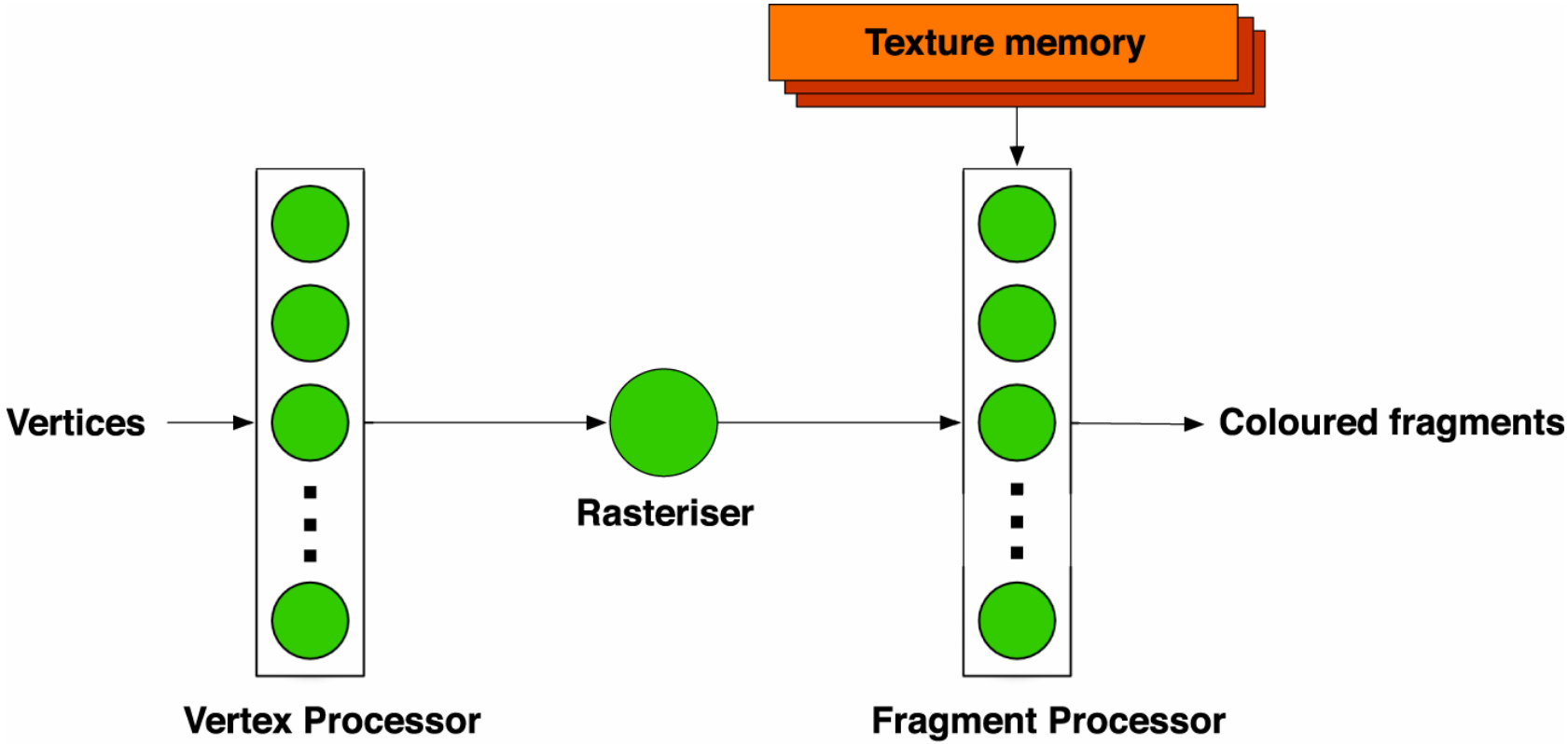
Source: NVIDIA CUDA SDK documentation

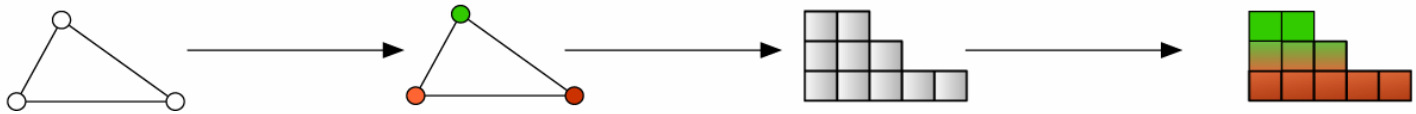
Graphics pipeline



Source:
ftp://download.nvidia.com/developer/presentations/2004/Perfect_Kitchen_Art/English_Evolution_of_GPUs.pdf

Graphics pipeline





GPUs and scientific computing

GPUs are designed to apply the
same ***shading function***
to many ***pixels*** simultaneously

GPUs and scientific computing

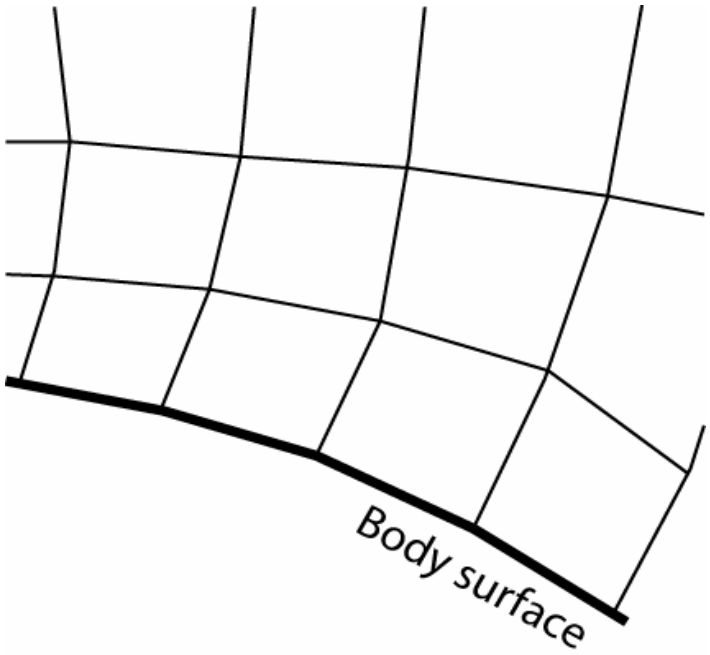
GPUs are designed to apply the
same ***function***
to many ***data*** simultaneously

This is what most scientific computing needs!

Part 2: Application to CFD

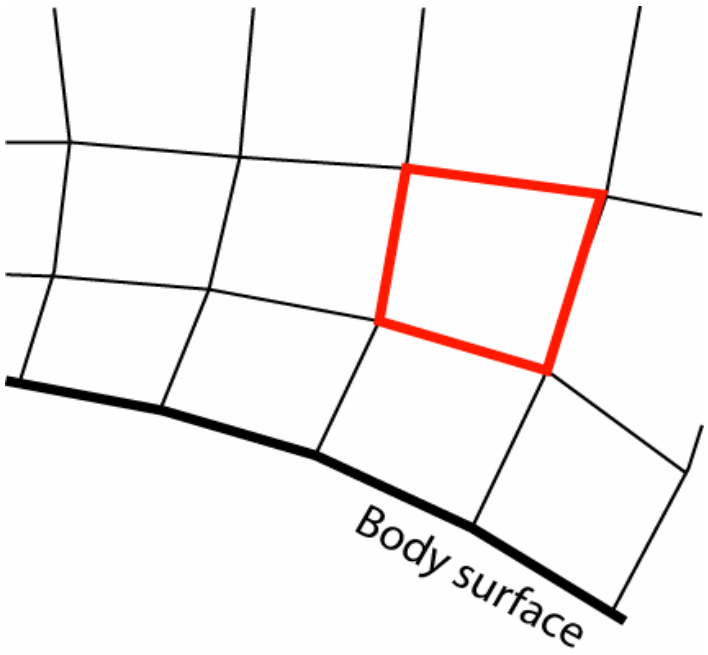
CFD basics

Body-fitted mesh



CFD basics

Body-fitted mesh



For each cell, conserve:

- mass
- momentum
- energy

and update flow properties

GPGPU programming models

Two options (of many) :

- Treat GPU as a streaming machine (Brook - Stanford)

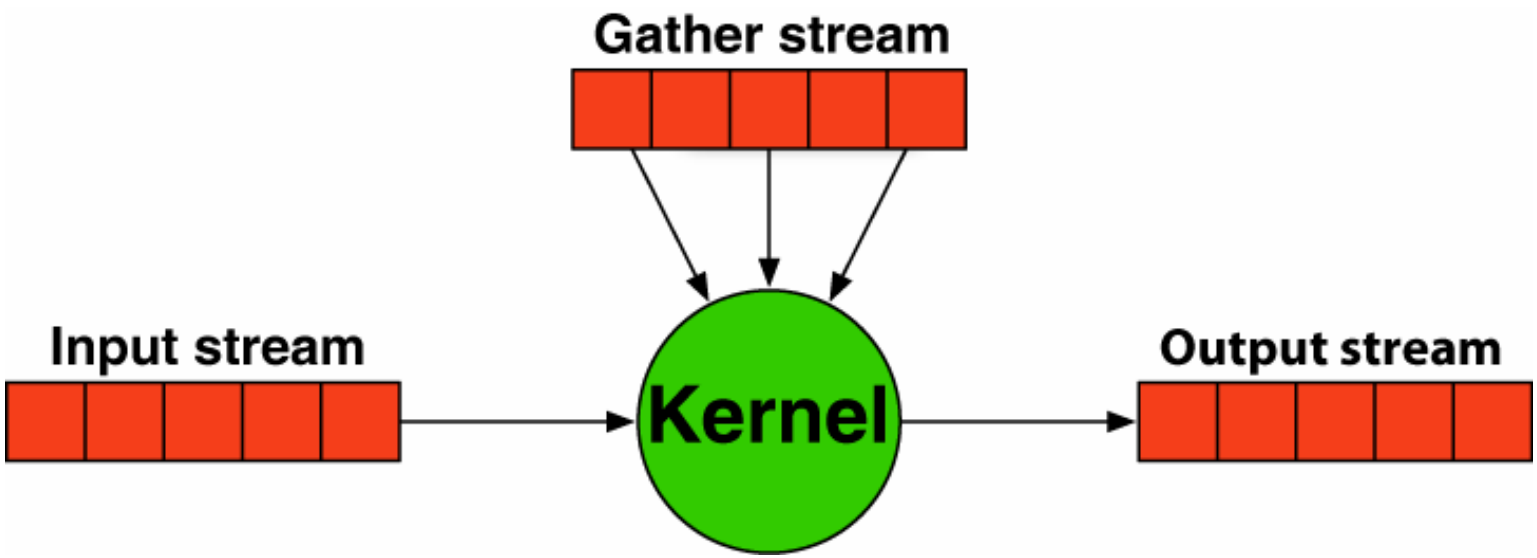
- Treat GPU as a set of multi-processors (CUDA - NVIDIA)

GPGPU programming models

Two options (of many) :

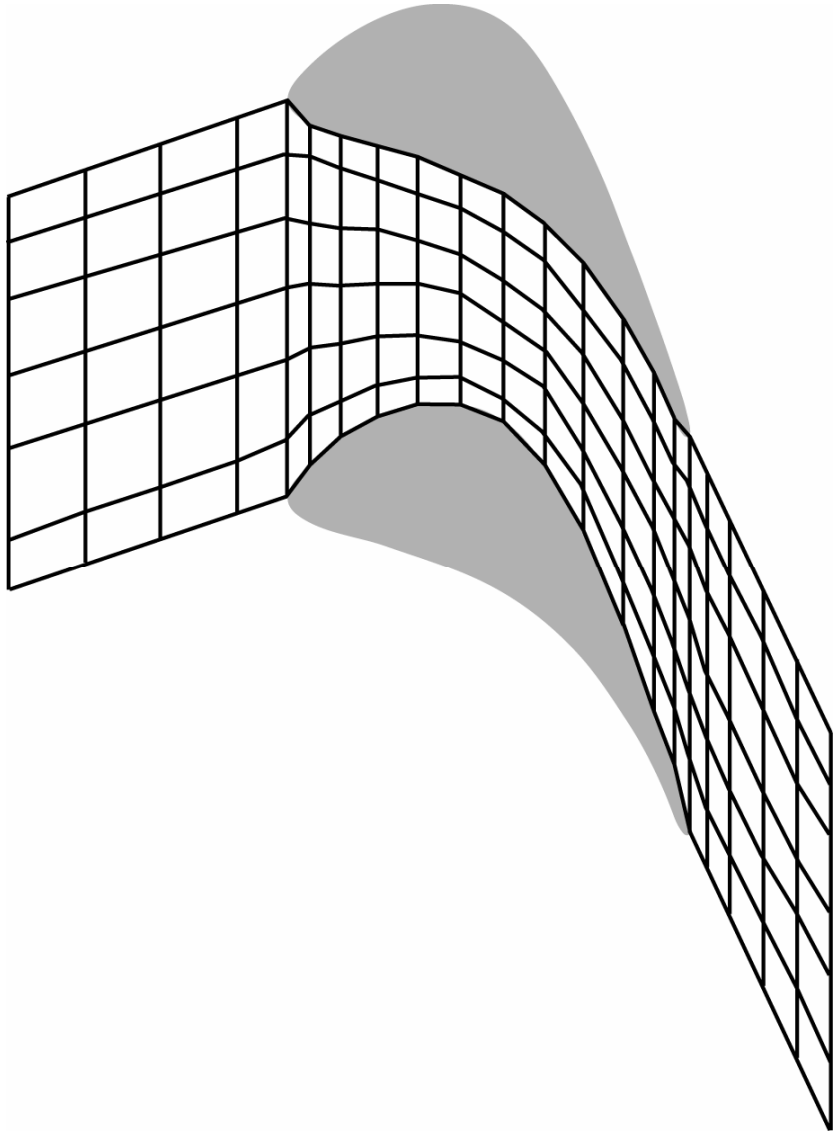
- Treat GPU as a streaming machine (Brook - Stanford)
  - Example: 2D flow solver
- Treat GPU as a set of multi-processors (CUDA - NVIDIA)
  - Example: 3D flow solver

The GPU as a streaming machine (Brook)

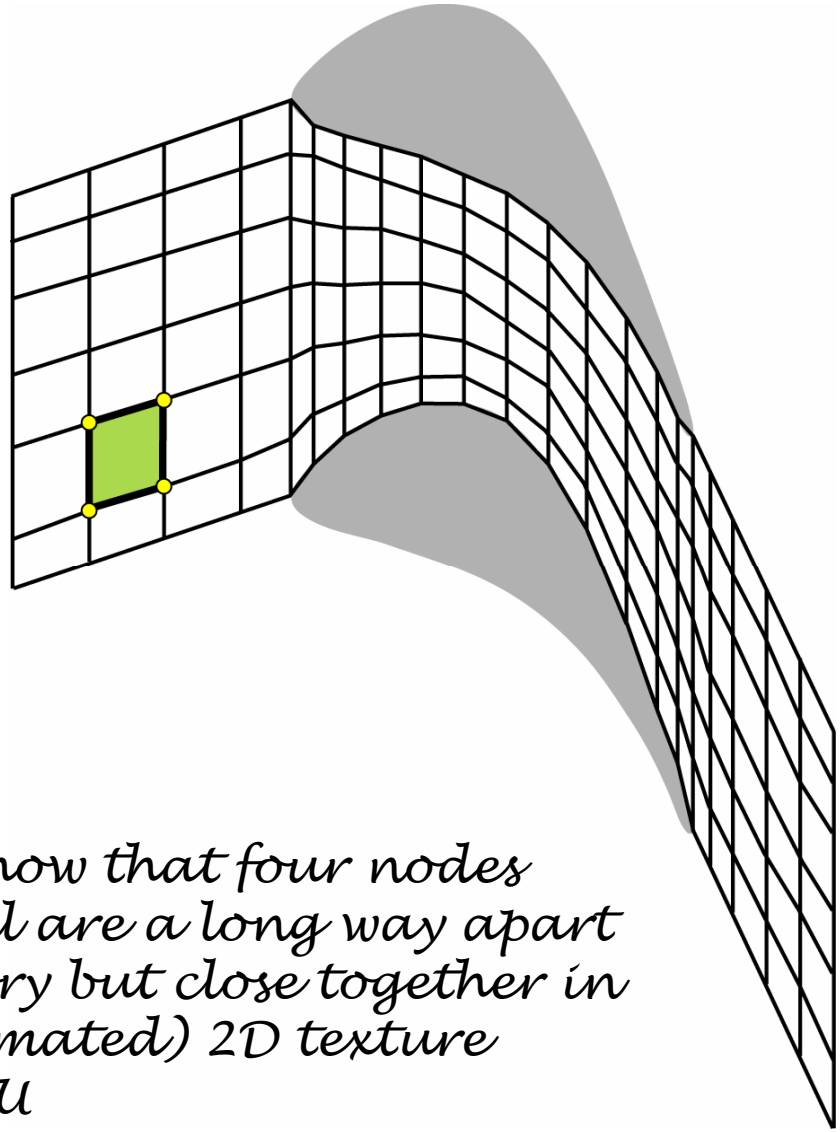


Source:
http://www.eng.cam.ac.uk/~gp10006/research/Brandvik_Pullan_2007b_DRAFT.pdf

Typical grid

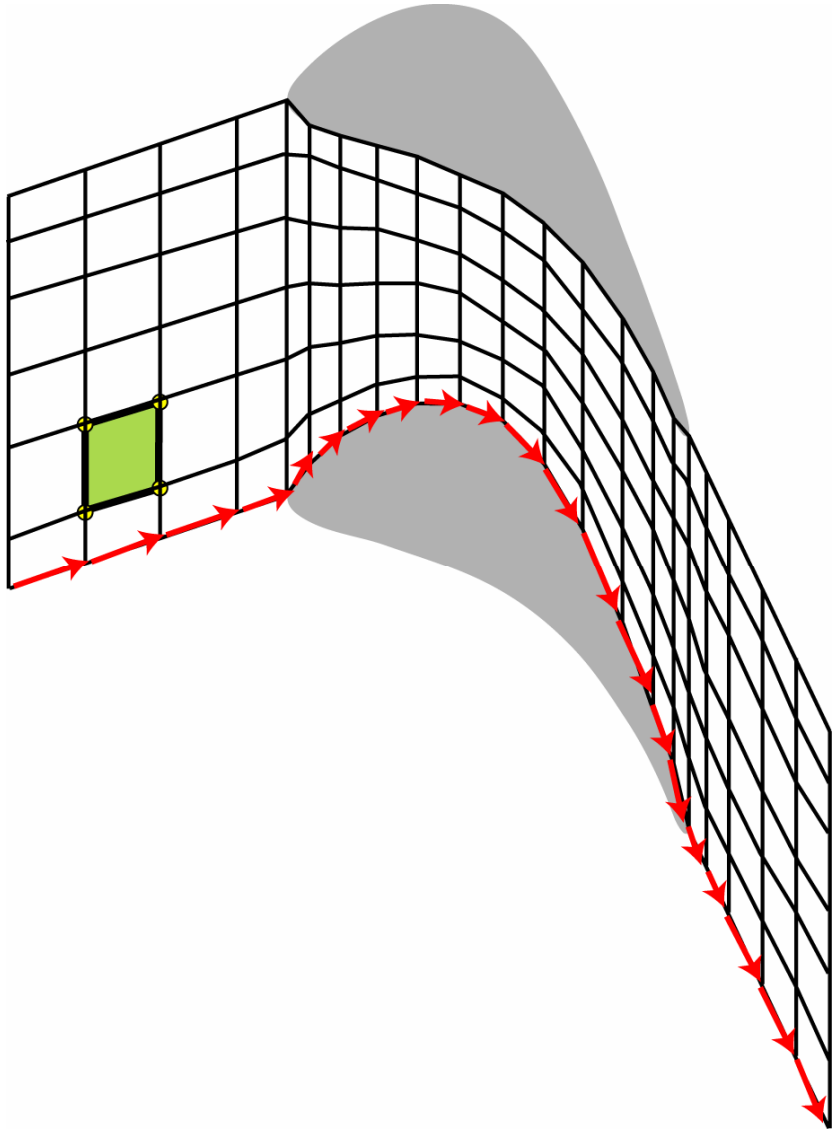


Typical grid

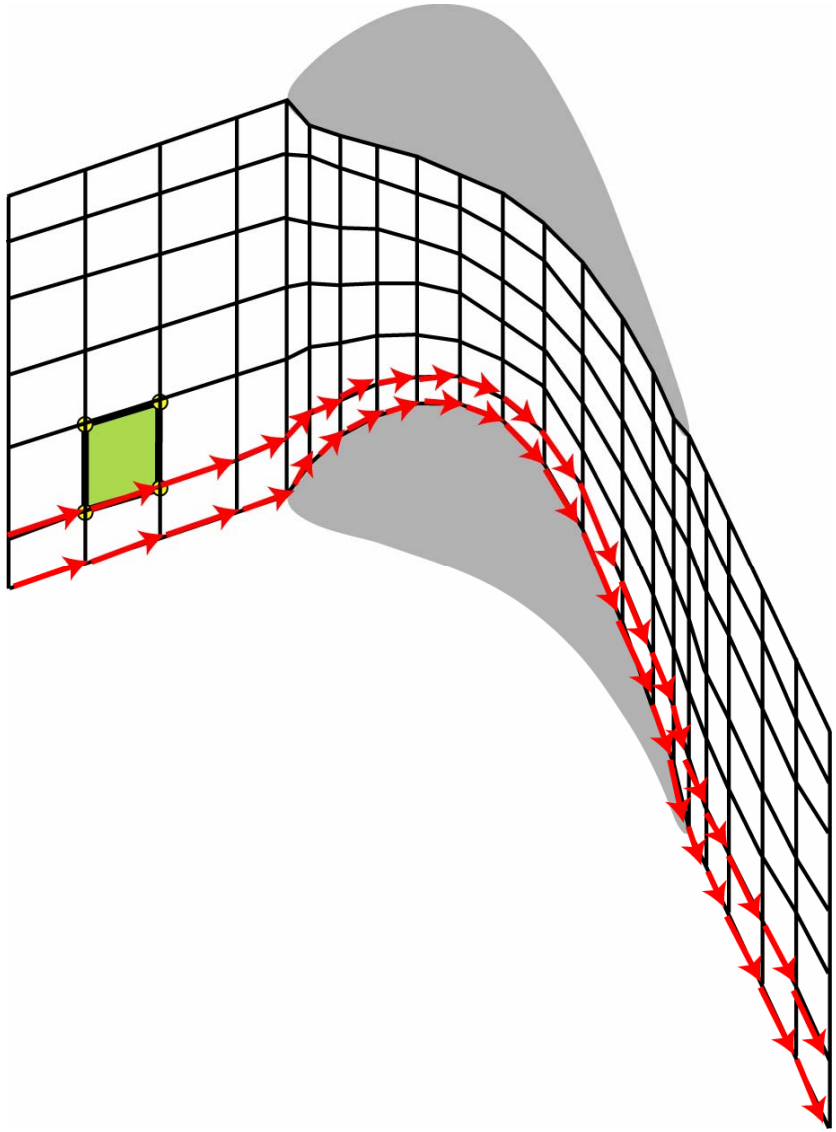


These slides show that four nodes forming a cell are a long way apart in CPU memory but close together in the (approximated) 2D texture cache of a GPU

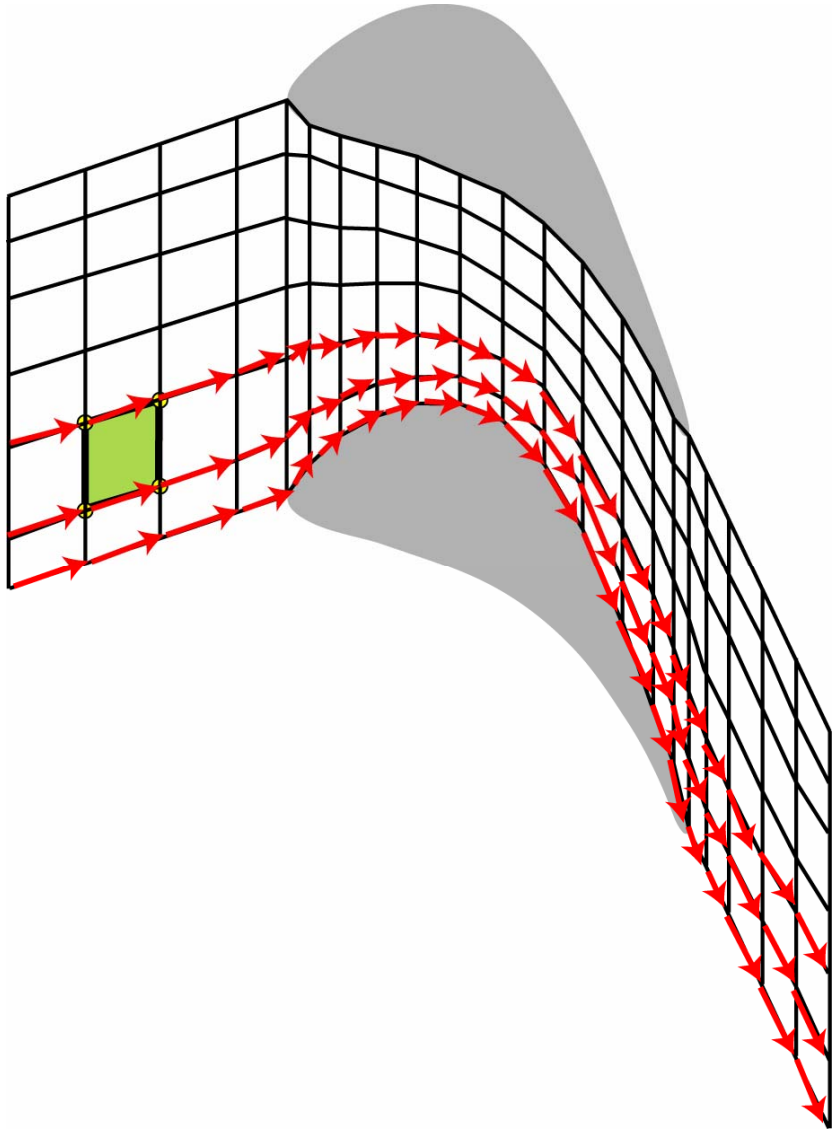
Typical grid – CPU memory



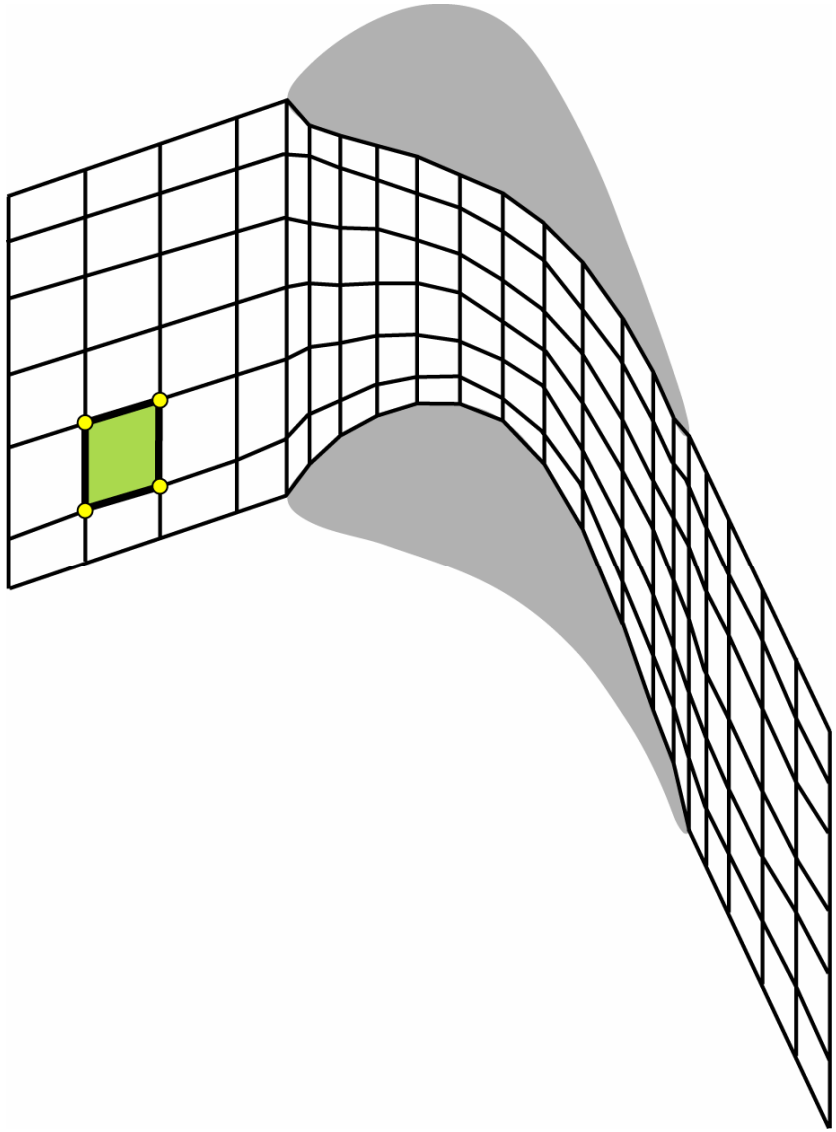
Typical grid – CPU memory



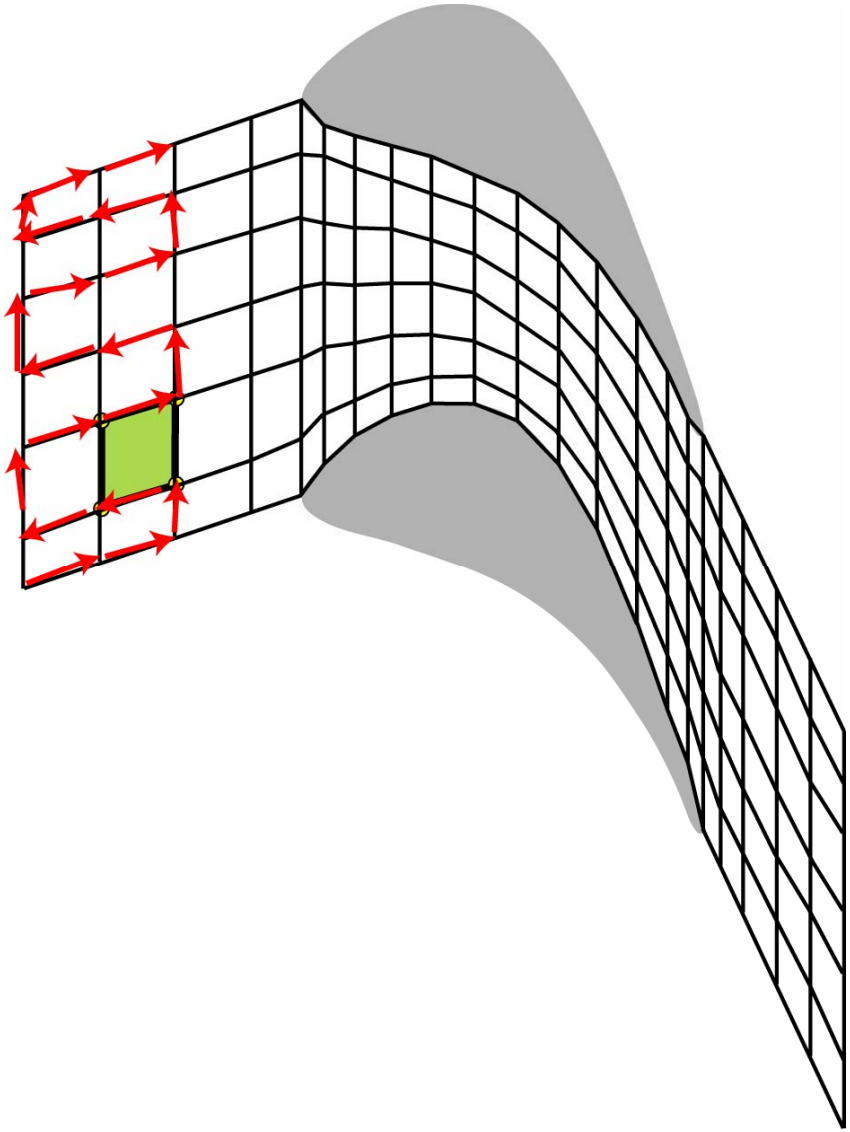
Typical grid – CPU memory



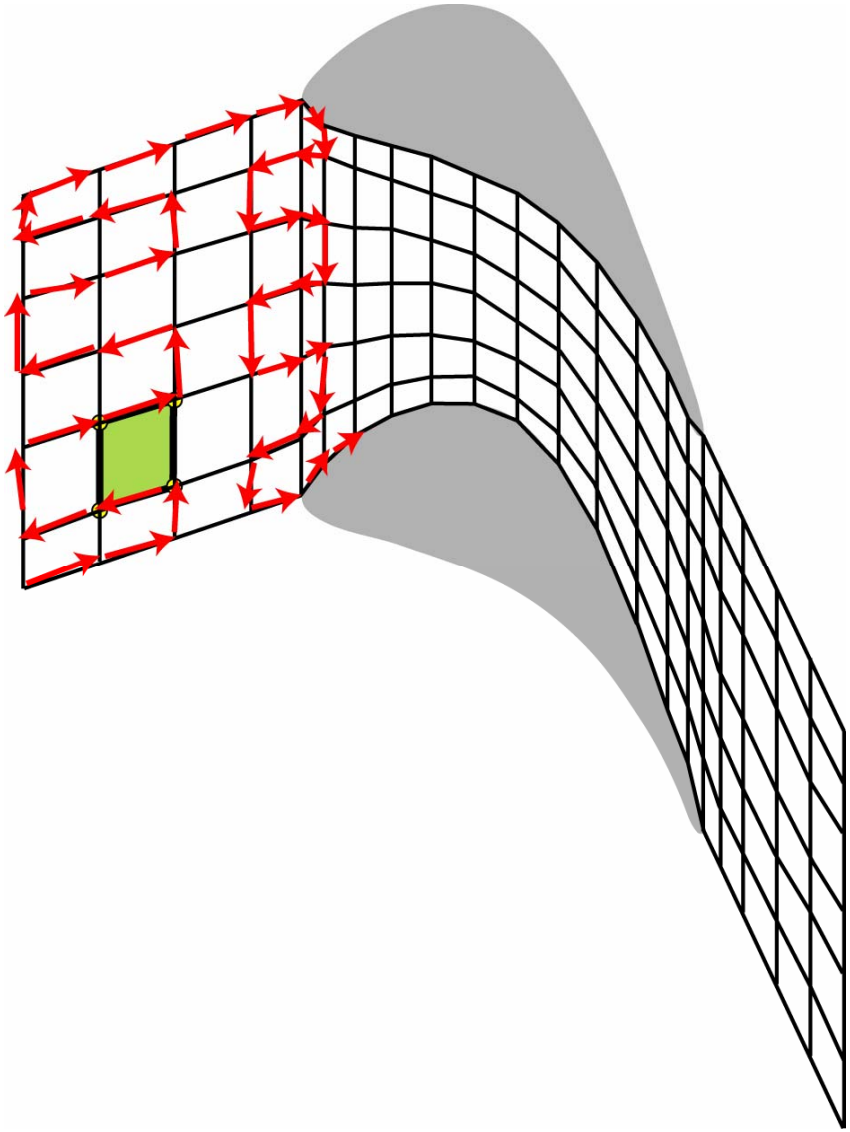
Typical grid – GPU 2D memory



Typical grid – GPU 2D memory

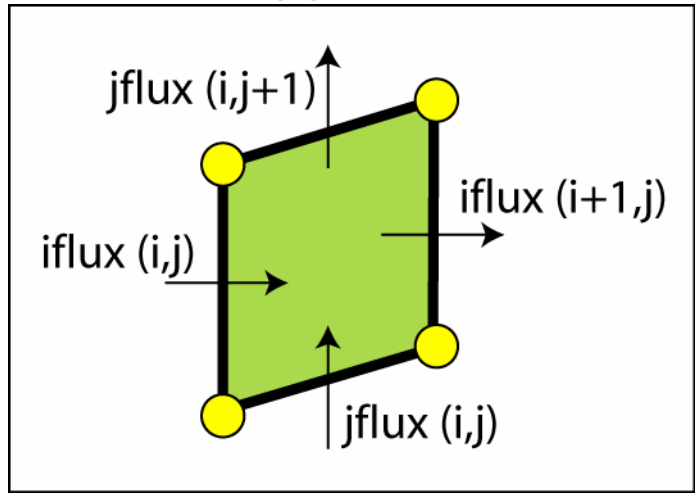


Typical grid – GPU 2D memory



Example Brook code

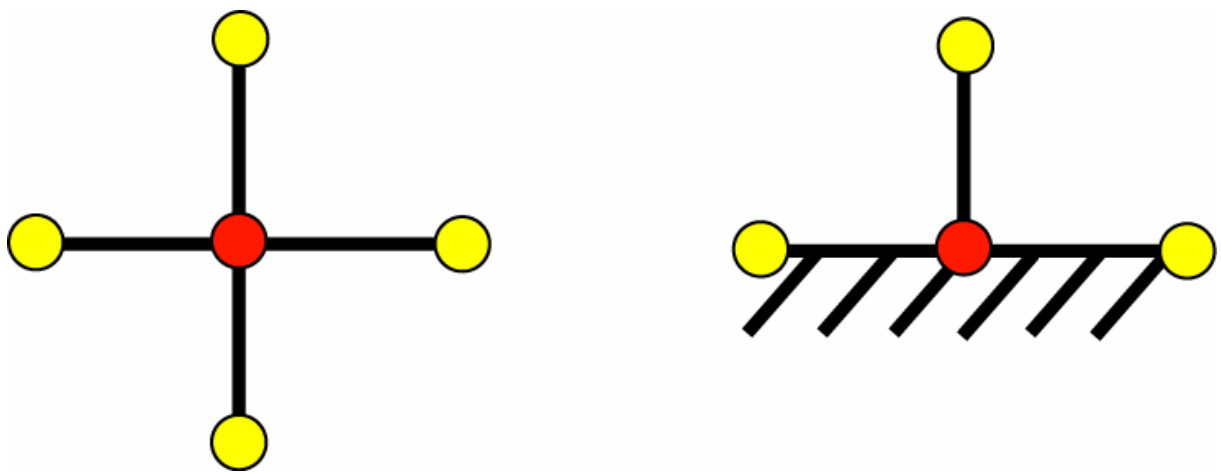
```
kernel void calc_changes(float deltat<>, float area<>,
    float4 iflux[ ][ ], float4 jflux[ ][ ], out float4 delta<>){

    float2 up ={0, 1};
    float2 right = {1, 0};
    float2 index = indexof delta.xy;
    delta =(iflux[index] - iflux[index + right]
     + jflux[index] - jflux[index + up])
          *deltat/area;
}
```



Boundary problems

- To avoid branching – apply same kernel to all nodes (including boundaries)

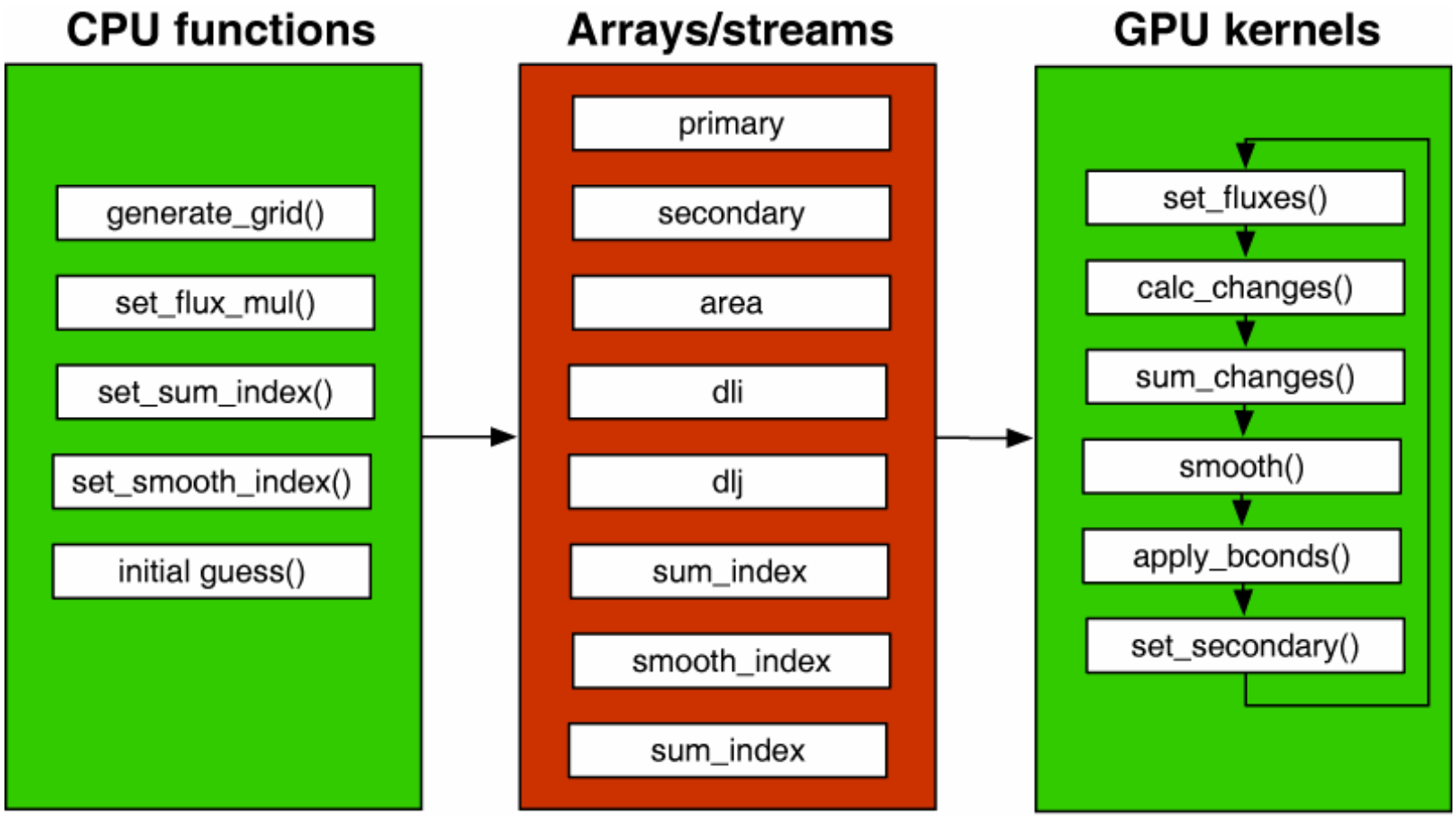
- Boundary treatment is often different:



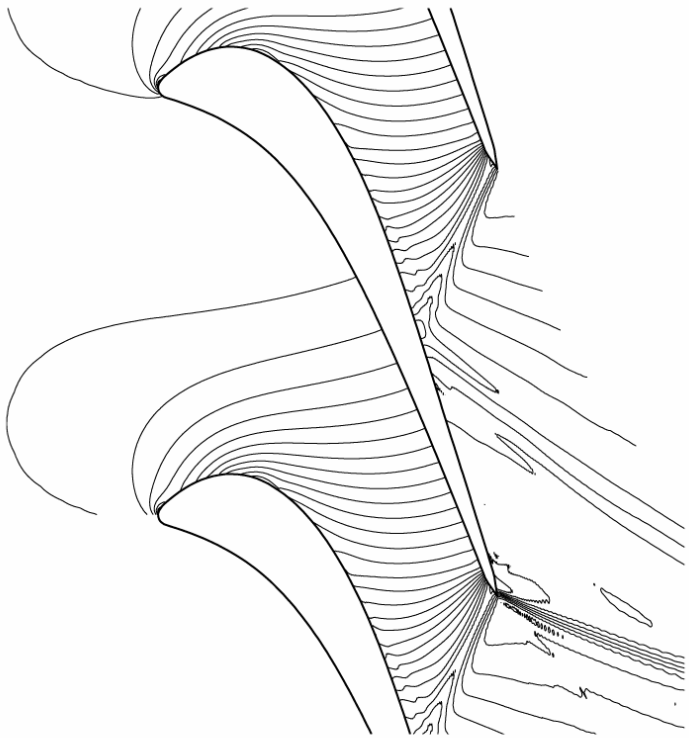
Store indices:

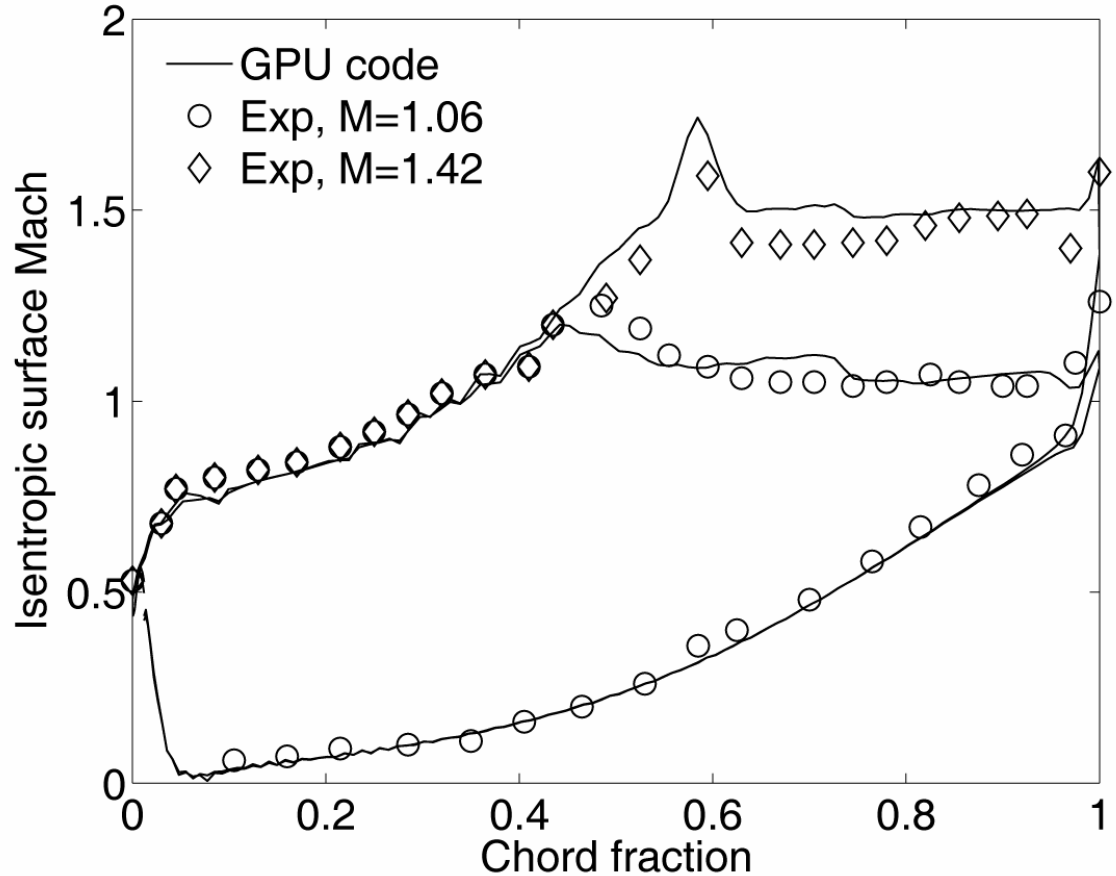
```
kernel void dist_delta(Index4 index<>, float4 primary_old<>,
    float4 delta[ ][ ], out float4 primary<>) {

    primary = primary_old +
    0.25f*(delta[index.i1] + delta[index.i2] +
            delta[index.i3] + delta[index.i4]);
}
```

CPU / GPU code split



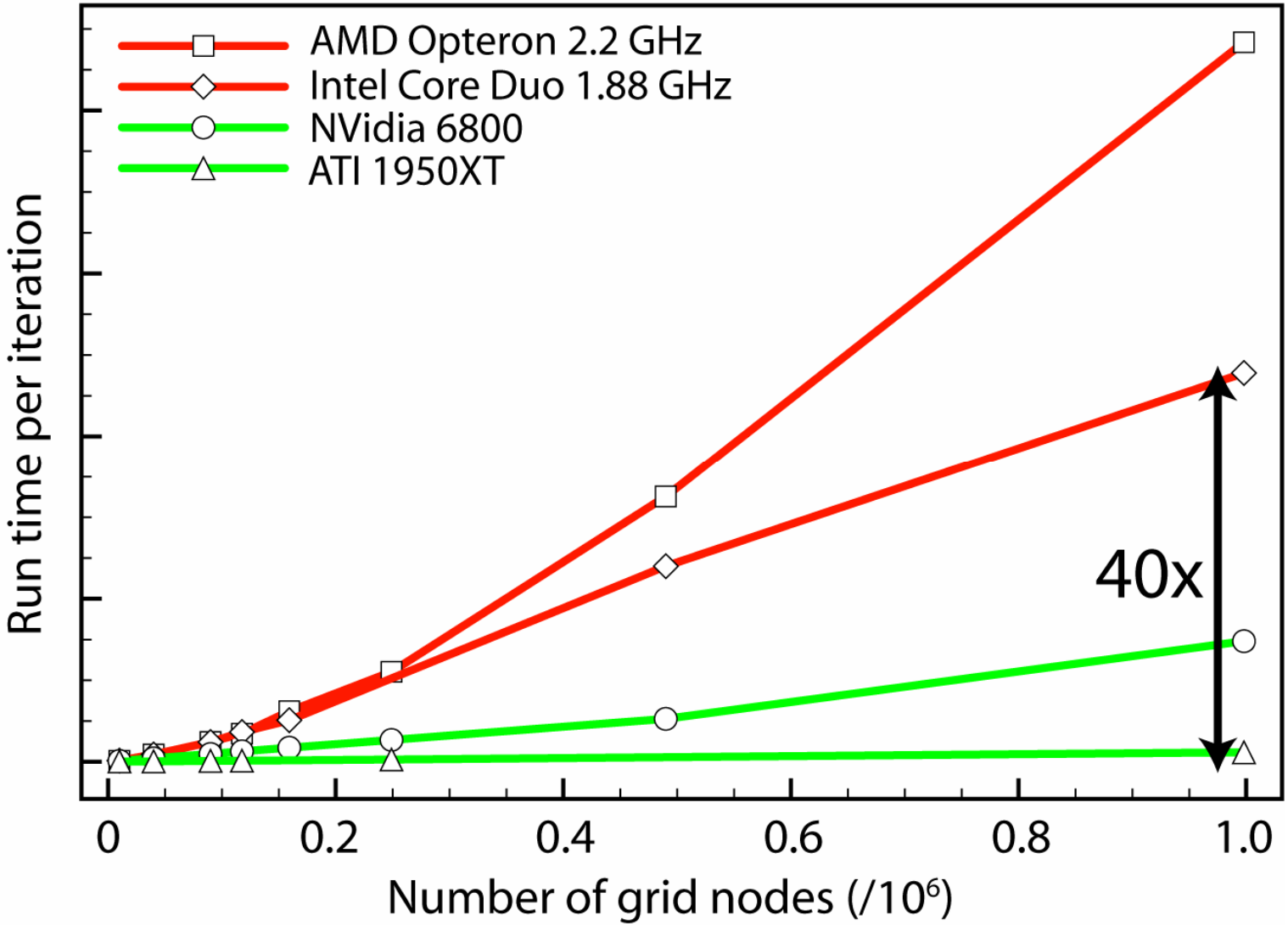
2D results



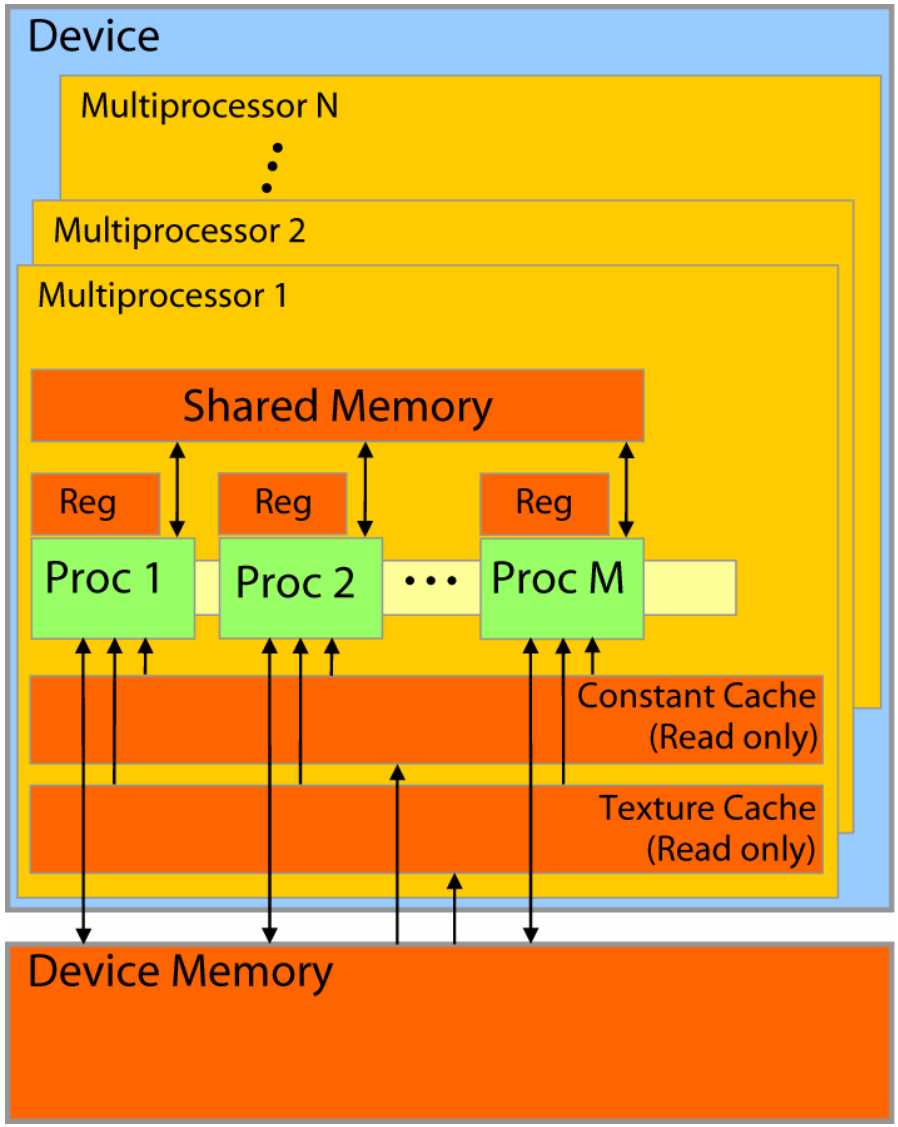


Source:
http://www.eng.cam.ac.uk/~gp10006/research/Brandvik_Pullan_2008a_DRAFT.pdf

2D performance



The GPU as a multi-processor (CUDA)



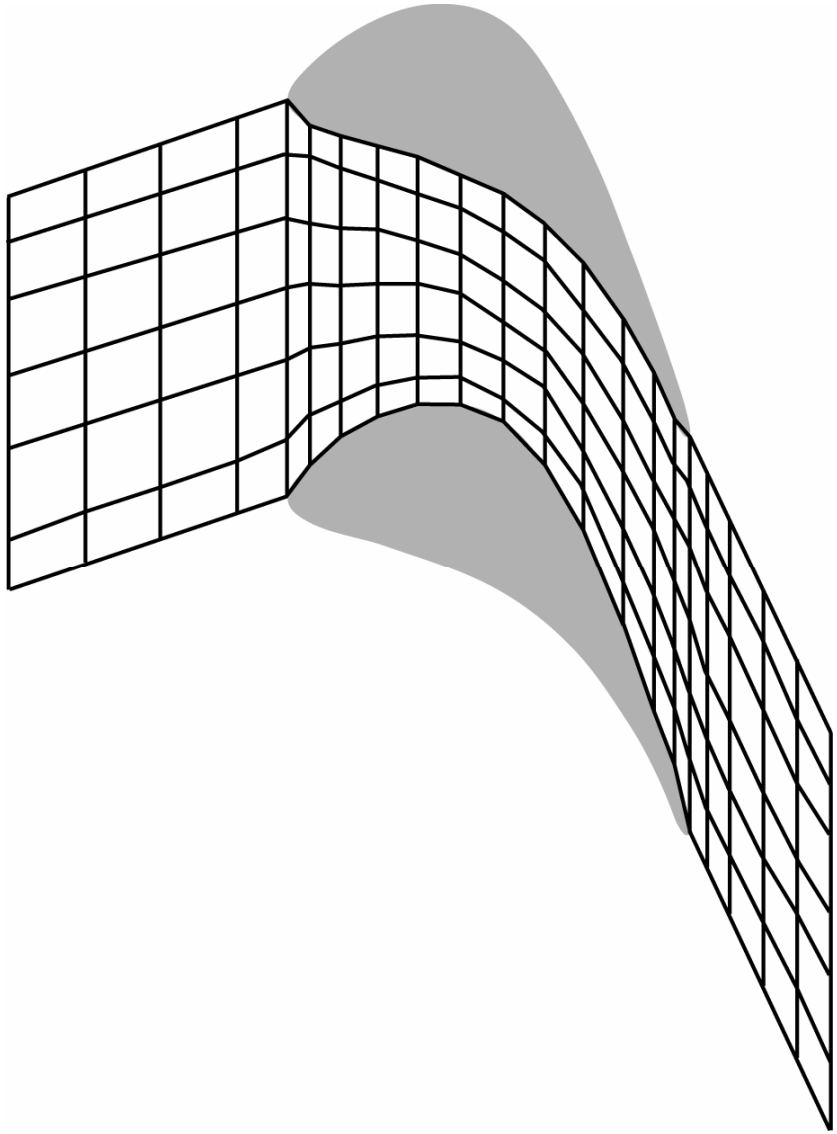
Divide 128 cores into

8 Multiprocessors (MPs)

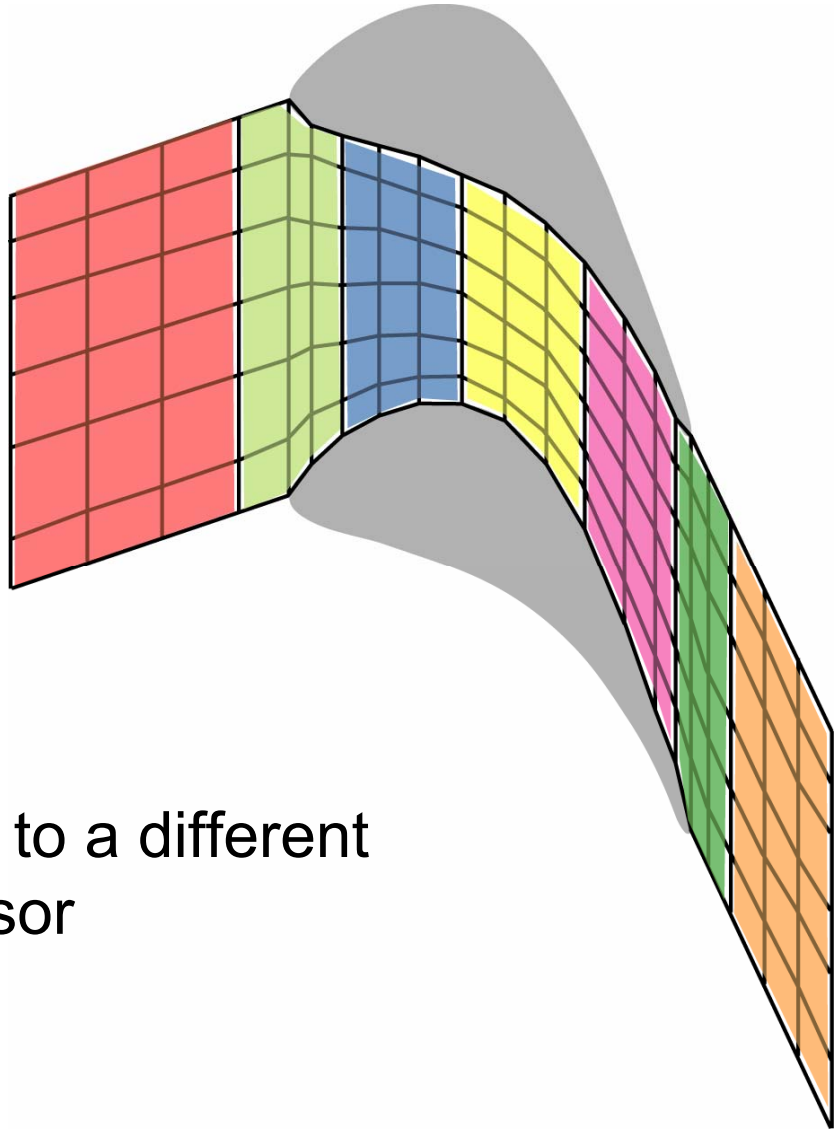
Each MP has 16kB shared memory (no caching)

Source: NVIDIA CUDA SDK documentation

Typical grid – CUDA partitioning

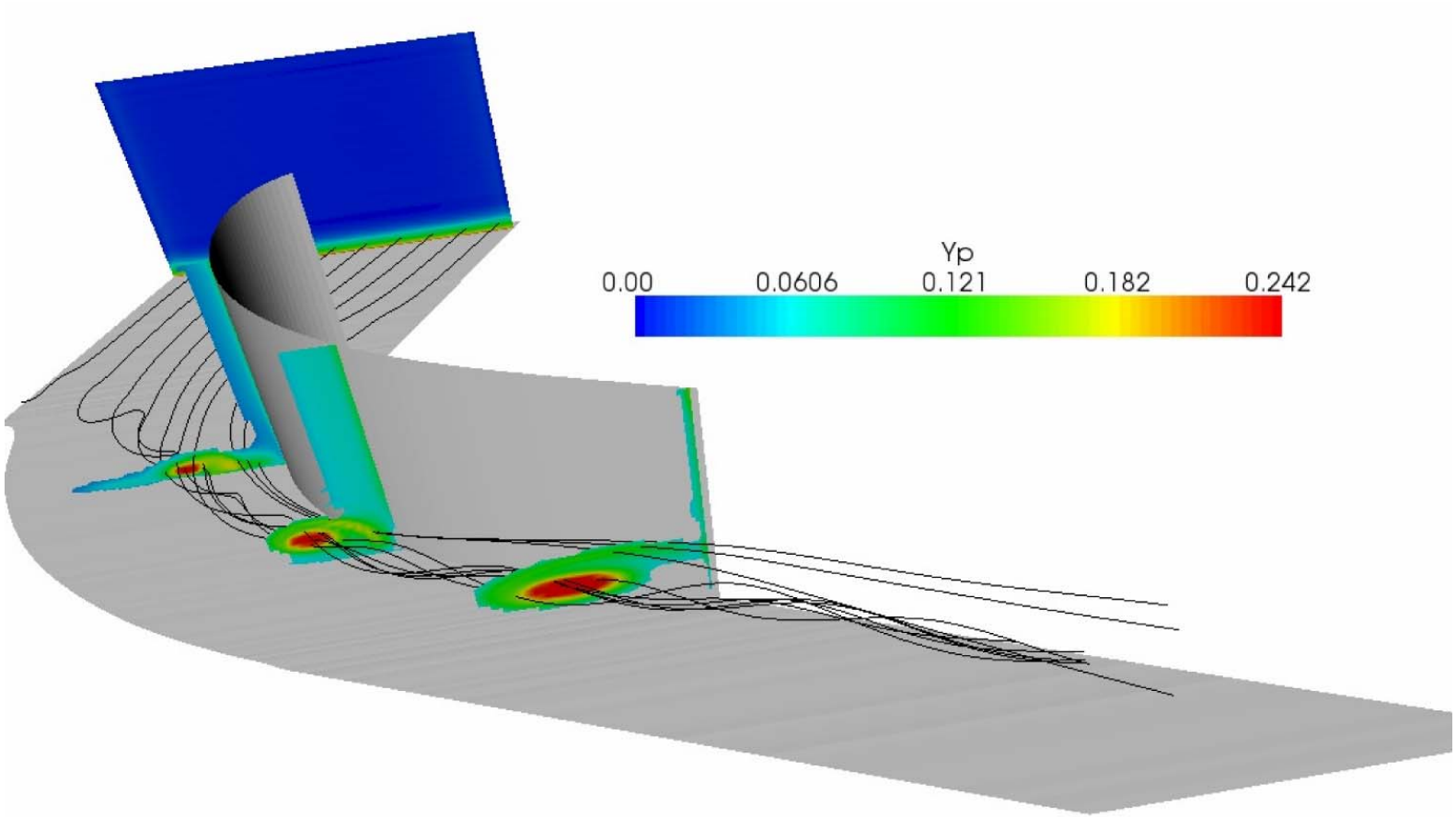


Typical grid – CUDA partitioning



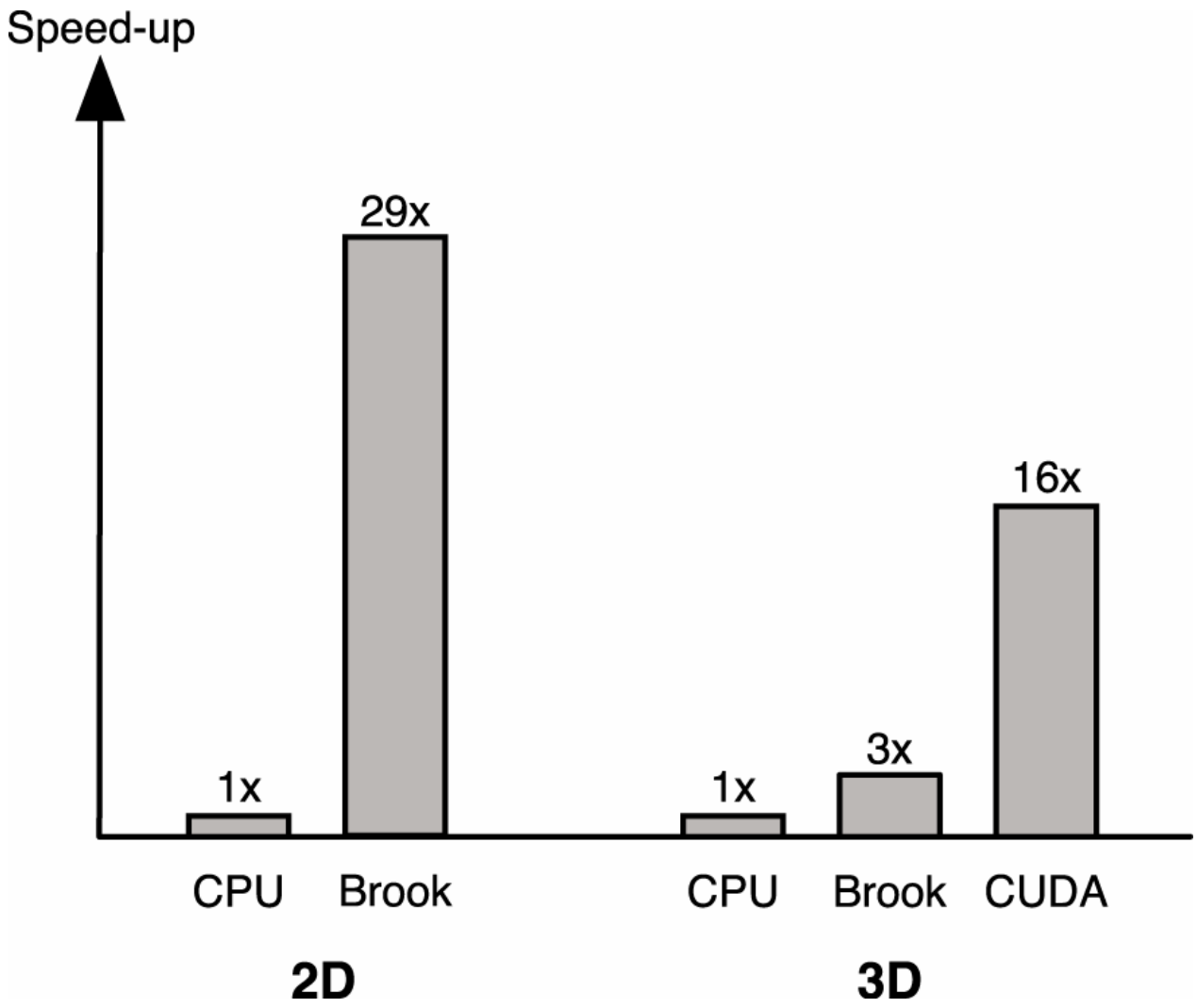
Each colour to a different multiprocessor

3D results



Source:
http://www.eng.cam.ac.uk/~gp10006/research/Brandvik_Pullan_2008a_DRAFT.pdf

2D/3D performance at 300k points



Conclusions

- Transistor counts continue to increase (exponentially)
- Current CPUs are multi-core (≈4) MIMD
- Current GPUs are many-core (≈100) SIMD

Conclusions

- Transistor counts continue to increase (exponentially)
- Current CPUs are multi-core (≈4) MIMD
- Current GPUs are many-core (≈100) SIMD

- Many-core SIMD is perfect for scientific computations
- Speed-ups for CFD of 29x (2D) and 16x (3D) obtained

Conclusions

- Transistor counts continue to increase (exponentially)
- Current CPUs are multi-core (≈4) MIMD
- Current GPUs are many-core (≈100) SIMD

- Many-core SIMD is perfect for scientific computations
- Speed-ups for CFD of 29x (2D) and 16x (3D) obtained

- Future CPUs will have heterogeneous cores
- MIMD and SIMD on one chip – the best of both worlds!

Challenges

The sands are still shifting:

- Software
  - There are many ways to program GPUs
- Hardware
  - There are several options for future hardware
    - GPUs
    - CPUs with accelerator boards / chips
    - CPU-GPU combined on a single chip

Developers need a framework to support new software

Acknowledgements

- Research student: Tobias Brandvik
- Donation of GPU hardware: NVIDIA