Overview of GPU Suitability and Progress of CFD Applications


Stan Posey; sposey@nvidia.com; NVIDIA, Santa Clara, CA, USA
Agenda: GPU Suitability and Progress of CFD

- NVIDIA HPC Introduction
- CFD Suitability for GPUs
- CFD Progress and Directions
GPUs Mainstream Across Diverse HPC Domains

FY14 Segments

Supercomputing 23%
Oil & Gas 12%
Higher Ed / Research 15%
Defense/ Federal 13%
CAE / MFG 7%
Finance 4%
Consumer 6%
Consumer Web 9%

World’s Top 3 Servers are GPU-Accelerated

- Dell R720
- HP DL380
- IBM x3650

World’s Top 3 Servers are GPU-Accelerated

- Dell R720
- HP DL380
- IBM x3650
# Tesla GPU Progression During Recent Years

<table>
<thead>
<tr>
<th></th>
<th>2012 (Fermi) M2075</th>
<th>2014 (Kepler) K20X</th>
<th>2014 (Kepler) K40</th>
<th>2014 (Kepler) K80</th>
<th>Kepler / Fermi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak SP</td>
<td>1.03 TF</td>
<td>3.93 TF</td>
<td>4.29 TF</td>
<td>8.74 TF</td>
<td>4x</td>
</tr>
<tr>
<td>Peak SGEMM</td>
<td>2.95 TF</td>
<td>3.22 TF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak DP</td>
<td>.515 TF</td>
<td>1.31 TF</td>
<td>1.43 TF</td>
<td>2.90 TF</td>
<td>3x</td>
</tr>
<tr>
<td>Peak DGEMM</td>
<td>1.22 TF</td>
<td>1.33 TF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory size</td>
<td>6 GB</td>
<td>6 GB</td>
<td>12 GB</td>
<td>24 GB (12 each)</td>
<td>2x</td>
</tr>
<tr>
<td>Mem BW (ECC off)</td>
<td>150 GB/s</td>
<td>250 GB/s</td>
<td>288 GB/s</td>
<td>480 GB/s (240 each)</td>
<td>2x</td>
</tr>
<tr>
<td>Memory Clock</td>
<td>2.6 GHz</td>
<td>3.0 GHz</td>
<td>3.0 GHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCIe Gen</td>
<td>Gen 2</td>
<td>Gen 2</td>
<td>Gen 3</td>
<td>Gen 3</td>
<td>2x</td>
</tr>
<tr>
<td># of Cores</td>
<td>448</td>
<td>2688</td>
<td>2880</td>
<td>4992 (2496 each)</td>
<td>5x</td>
</tr>
<tr>
<td>Board Power</td>
<td>235W</td>
<td>235W</td>
<td>235W</td>
<td>300W</td>
<td>0% – 28%</td>
</tr>
</tbody>
</table>

**Note:** Tesla K80 specifications are shown as aggregate of two GPUs on a single board.
GPU Motivation (I): Performance Trends

Peak Double Precision FLOPS

Peak Memory Bandwidth

- NVIDIA GPU
- x86 CPU

GFLOPS

GPU Motivation (I): Performance Trends

Peak Double Precision FLOPS

Peak Memory Bandwidth

- NVIDIA GPU
- x86 CPU
### Green500

<table>
<thead>
<tr>
<th>Rank</th>
<th>MFLOPS/W</th>
<th>Site</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4,389.82</td>
<td>GSIC Center, Tokyo Tech</td>
</tr>
<tr>
<td>2</td>
<td>3,631.70</td>
<td>Cambridge University</td>
</tr>
<tr>
<td>3</td>
<td>3,517.84</td>
<td>University of Tsukuba</td>
</tr>
<tr>
<td>4</td>
<td>3,459.46</td>
<td>SURFsara</td>
</tr>
<tr>
<td>5</td>
<td>3,185.91</td>
<td>Swiss National Supercomputing Centre (CSCS)</td>
</tr>
<tr>
<td>6</td>
<td>3,131.06</td>
<td>ROMEO HPC Center</td>
</tr>
<tr>
<td>7</td>
<td>3,019.72</td>
<td>CSIRO</td>
</tr>
<tr>
<td>8</td>
<td>2,951.95</td>
<td>GSIC Center, Tokyo Tech</td>
</tr>
<tr>
<td>9</td>
<td>2,813.14</td>
<td>Eni</td>
</tr>
<tr>
<td>10</td>
<td>2,629.10</td>
<td>(Financial Institution)</td>
</tr>
<tr>
<td>16</td>
<td>2,495.12</td>
<td>Mississippi State (top non-NVIDIA) Intel Phi</td>
</tr>
<tr>
<td>59</td>
<td>1,226.60</td>
<td>ICHEC (top X86 cluster)</td>
</tr>
</tbody>
</table>

GPU Motivation (II): Energy Efficient HPC

### Top500

<table>
<thead>
<tr>
<th>Rank</th>
<th>TFLOPS/s</th>
<th>Site</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33,862.7</td>
<td>National Super Computer Centre Guangzhou</td>
</tr>
<tr>
<td>2</td>
<td>17,590.0</td>
<td>Oak Ridge National Lab</td>
</tr>
<tr>
<td>3</td>
<td>17,173.2</td>
<td>DOE, United States</td>
</tr>
<tr>
<td>4</td>
<td>10,510.0</td>
<td>RIKEN Advanced Institute for Computational Science</td>
</tr>
<tr>
<td>5</td>
<td>8,586.6</td>
<td>Argonne National Lab</td>
</tr>
<tr>
<td>6</td>
<td>6,271.0</td>
<td>Swiss National Supercomputing Centre (CSCS)</td>
</tr>
<tr>
<td>7</td>
<td>5,168.1</td>
<td>University of Texas</td>
</tr>
<tr>
<td>8</td>
<td>5,008.9</td>
<td>Forschungszentrum Juelich</td>
</tr>
<tr>
<td>9</td>
<td>4,293.3</td>
<td>DOE, United States</td>
</tr>
<tr>
<td>10</td>
<td>3,143.5</td>
<td>Government</td>
</tr>
</tbody>
</table>
Higher fidelity models within manageable compute and energy costs

- RANS → URANS → LES → ?
  - DNS

Increase in non-deterministic ensembles to manage/quantify uncertainty

HOMs for improved resolution of transitional and vortex-heavy flows

Accelerator technology identified as a cost-effective and practical approach to future computational challenges
VIDIA Strategy for GPU-Accelerated HPC

Strategic Alliances
- Business and technical alliances with COTS vendors
- Investment in long-term collaboration for solver-level libraries
- Development of collaborations with academic research community:

Software Development
- Libraries cuSPARSE, cuBLAS; OpenACC with PGI (acquisition) and Cray
- NVIDIA linear solver toolkit with emphasis on AMG for industry CFD

Applications Support
- Application engineering support for COTS vendors and customers
  - Implicit Schemes: Integration of libraries and solver toolkit
  - Explicit Schemes: Stencil libraries, OpenACC for directives-based
Agenda: GPU Suitability and Progress of CFD

- NVIDIA HPC Introduction
- CFD Suitability for GPUs
- CFD Progress and Directions
Programming Strategies for GPU Acceleration

Applications

- **GPU Libraries**
  Provides Fast “Drop-In” Acceleration

- **OpenACC Directives**
  GPU-acceleration in Standard Language (Fortran, C, C++)

- **Programming Languages**
  Maximum Flexibility with GPU Architecture and Software Features

Increasing Development Effort
FD Characteristics and GPU Suitability

Structured Grid FV

Unstructured FV

Unstructured FE

Explicit

Numerical operations on I,J,K stencil, no “solver”
[Flat profiles: Typical GPU strategy is directives (OpenACC)]

Implicit

Finite Volume

Finite Element:

Sparse matrix linear algebra – iterative solvers
[Hot spot ~50%, few LoC: Typical GPU strategy is CUDA and libs]
## Select GPU Implementations for CFD Practice

### Structured Grid FV
- SJTU RANS
- HiPSTAR
- Turbostream
- TACOMA

### Unstructured FV
- Rolls-Royce
- HYDRA
- Flare

### Unstructured FE
- SD++
- PyFR
- Hyperflux
- JENRE, Propel

### Finite Volume
- ANSYS Fluent
- Culises for OpenFOAM

### Finite Element
- Altair
- AcuSolve
- Autodesk
- Moldflow

### Explicit
- Usually Compressible

### Implicit
- Usually Incompressible
**What is Meant by “CFD Practice”**

- These are not demonstrators, rather meaningful developments towards production use CFD
- Proven performance on large-scale engineering simulations
- Long-term maintenance and software engineering considerations
- Co-design efforts between CFD scientists and computer scientists
- In most cases, contributions from NVIDIA devtech engineering
<table>
<thead>
<tr>
<th>Explicit</th>
<th>Structured Grid FV</th>
<th>Unstructured FV</th>
<th>Unstructured FE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usually Compressible</td>
<td>SJTU RANS</td>
<td>Rolls-Royce</td>
<td>SD++</td>
</tr>
<tr>
<td></td>
<td>HiPSTAR</td>
<td>HYDRA</td>
<td>PyFR</td>
</tr>
<tr>
<td></td>
<td>Turbostream</td>
<td>BAE SYSTEMS</td>
<td>Hyperflux</td>
</tr>
<tr>
<td>Implicit</td>
<td>TACOMA</td>
<td>Flare</td>
<td>JENRE, Propel</td>
</tr>
<tr>
<td>Usually Incompressible</td>
<td></td>
<td></td>
<td>OpenFOAM</td>
</tr>
</tbody>
</table>

Structured grid explicit generally best GPU fit
Select GPU Implementations for CFD Practice

**Structured Grid FV**
- SJTU RANS
- HiPSTAR
- Turbostream
- TACOMA

**Unstructured FV**
- Rolls-Royce
- HYDRA
- Flare
- BAE SYSTEMS

**Unstructured FE**
- SD++
- PyFR
- Hyperflux
- JENRE, Propel

**Finite Volume**
- ANSYS Fluent
- Culises for OpenFOAM

**Finite Element:**
- Altair
- AcuSolve
- Autodesk
- Moldflow

Explicit
- Usually Compressible

Implicit
- Usually Incompressible

Unstructured grid usually with renumbering/coloring
<table>
<thead>
<tr>
<th>Structured Grid FV</th>
<th>Unstructured FV</th>
<th>Unstructured FE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit</td>
<td>Implicit</td>
<td>Implicit</td>
</tr>
<tr>
<td>SJTU RANS</td>
<td>BAE SYSTEMS</td>
<td>SD++</td>
</tr>
<tr>
<td>HiPSTAR</td>
<td>Rolls-Royce</td>
<td>PyFR</td>
</tr>
<tr>
<td>Turbostream</td>
<td>HYDRA</td>
<td>Hyperflux</td>
</tr>
<tr>
<td>TACOMA</td>
<td>Flare</td>
<td>JENRE, Propel</td>
</tr>
</tbody>
</table>

**Finite Volume**

- ANSYS Fluent
- Culises for OpenFOAM

**Finite Element:**

- AcuSolve
- Moldflow

**COTS CFD mostly apply solver library (AmgX)**
<table>
<thead>
<tr>
<th>Organization</th>
<th>Location</th>
<th>Software</th>
<th>GPU Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMAC/SJTU</td>
<td>China</td>
<td>SJTU RANS</td>
<td>Fortran and CUDA</td>
</tr>
<tr>
<td>U Southhampton</td>
<td>UK</td>
<td>HiPSTAR</td>
<td>Fortran and OpenACC</td>
</tr>
<tr>
<td>Turbostream</td>
<td>UK</td>
<td>Turbostream</td>
<td>Fortran, python templates s-to-s to CUDA</td>
</tr>
<tr>
<td>GE GRC</td>
<td>US</td>
<td>TACOMA</td>
<td>Fortran and OpenACC</td>
</tr>
<tr>
<td>Rolls Royce</td>
<td>UK</td>
<td>HYDRA</td>
<td>Fortran, python DSL s-to-s to CUDA-F</td>
</tr>
<tr>
<td>BAE Systems</td>
<td>UK</td>
<td>Flare</td>
<td>C++ and s-to-s templates to CUDA</td>
</tr>
<tr>
<td>Stanford U</td>
<td>US</td>
<td>SD++</td>
<td>C++ and CUDA</td>
</tr>
<tr>
<td>PyFR</td>
<td>UK</td>
<td>PyFR</td>
<td>Python s-to-s to CUDA (C for CPU)</td>
</tr>
<tr>
<td>CFMS</td>
<td>UK</td>
<td>Hyperflux</td>
<td>Python s-to-s to CUDA (C for CPU)</td>
</tr>
<tr>
<td>JENRE, Propel</td>
<td>US</td>
<td>USA</td>
<td>C++ and Thrust templates for CUDA</td>
</tr>
<tr>
<td>ANSYS Fluent</td>
<td>US</td>
<td>Implicit FEA</td>
<td>C++ and AmgX library, OpenACC</td>
</tr>
<tr>
<td>FluiDyna</td>
<td>DE</td>
<td>Culises/OpenFOAM</td>
<td>C++ (OpenFOAM), AmgX library, CUDA</td>
</tr>
<tr>
<td>Altair</td>
<td>US</td>
<td>AcuSolve</td>
<td>Fortran and CUDA</td>
</tr>
<tr>
<td>Autodesk</td>
<td>US</td>
<td>Moldflow</td>
<td>Fortran and AmgX library</td>
</tr>
</tbody>
</table>
## Select GPU Developments at Various Stages

<table>
<thead>
<tr>
<th>Organization</th>
<th>Location</th>
<th>Software</th>
<th>GPU Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>U Wyoming / Mavriplis</td>
<td>US</td>
<td><em>Not specific</em></td>
<td>CU++ object oriented templates</td>
</tr>
<tr>
<td>GMU / Lohner</td>
<td>US</td>
<td>FEFLO</td>
<td>Python Fortran-to-CUDA translator</td>
</tr>
<tr>
<td>SpaceX</td>
<td>US</td>
<td><em>Not specific</em></td>
<td>C++ and CUDA</td>
</tr>
<tr>
<td>CPFD</td>
<td>US</td>
<td>BARRACUDA</td>
<td>Fortran and CUDA</td>
</tr>
<tr>
<td>GWU / Barba</td>
<td>US</td>
<td><em>Not specific</em></td>
<td>C++, python, pyCUDA</td>
</tr>
<tr>
<td>UTC Research</td>
<td>US</td>
<td><em>Combustion</em></td>
<td>Fortran and CUDA</td>
</tr>
<tr>
<td>Convergent Science / LLNL</td>
<td>US</td>
<td>CONVERGE</td>
<td>C++ and CUDA, cuSOLVE (NVIDIA)</td>
</tr>
<tr>
<td>Craft Tech</td>
<td>US</td>
<td>CRAFT, CRUNCH</td>
<td>Fortran and CUDA, OpenACC</td>
</tr>
<tr>
<td>Bombardier</td>
<td>CA</td>
<td><em>Not specific</em></td>
<td>C++ and CUDA</td>
</tr>
<tr>
<td>DLR</td>
<td>DE</td>
<td>TAU</td>
<td>Fortran and CUDA</td>
</tr>
<tr>
<td>ONERA</td>
<td>FR</td>
<td>elsA</td>
<td>Fortran and CUDA</td>
</tr>
<tr>
<td>Vratis</td>
<td>PL</td>
<td>Speed-IT (OFOAM)</td>
<td>C++ and CUDA</td>
</tr>
<tr>
<td>NUMECA</td>
<td>BE</td>
<td>Fine/Turbo</td>
<td>Fortran and OpenACC</td>
</tr>
<tr>
<td>Prometech</td>
<td>JP</td>
<td>Particleworks</td>
<td>C++ and CUDA</td>
</tr>
<tr>
<td>TiTech / Aoki</td>
<td>JP</td>
<td><em>Not specific</em></td>
<td>C++ and CUDA</td>
</tr>
<tr>
<td>JAXA</td>
<td>JP</td>
<td>UPACS</td>
<td>Fortran and OpenACC</td>
</tr>
<tr>
<td>KISTI / Park</td>
<td>KR</td>
<td>KFLOW</td>
<td>Fortran and CUDA, OpenACC</td>
</tr>
<tr>
<td>VSSC</td>
<td>IN</td>
<td>PARAS3D</td>
<td>Fortran and CUDA</td>
</tr>
</tbody>
</table>
NVIDIA AmgX for Iterative Implicit Methods

- Scalable linear solver library for $Ax = b$ iterative methods
- No CUDA experience required, C API: links with Fortran, C, C++
- Reads common matrix formats (CSR, COO, MM)
- Interoperates easily with MPI, OpenMP, and hybrid parallel
- Single and double precision; Supported on Linux, Win64
- Multigrid; Krylov: GMRES, PCG, BiCGStab; Preconditioned variants
- Classic Iterative: Block-Jacobi, Gauss-Seidel, ILU’s; Multi-coloring
- Flexibility: All methods as solvers, preconditioners, or smoothers
- Download AmgX library: http://developer.nvidia.com/amgx
• Poisson matrix with ~8.2B rows solved in under 13 sec (200e3 Poisson matrix per GPU)
• ORNL TITAN: NVIDIA K20X one per node; CPU 16 core AMD Opteron 6274 @2.2GHz
Agenda: GPU Suitability and Progress of CFD

- NVIDIA HPC Introduction
- CFD Suitability for GPUs
- CFD Progress and Directions
Progress Summary for GPU-Parallel CFD

- GPU progress in CFD research continues to expand
  - Growth from arithmetic intensity in high-order methods
  - Breakthroughs with Hyper-Q feature (Kepler), GPUDirect, etc.

- Strong GPU investments by commercial (COTS) vendors
  - Breakthroughs with AmgX linear solvers and preconditioners
  - Preservation of costly MPI investment: GPU 2nd-level parallelism

- Success in end-user developed CFD with OpenACC
  - Most benefits currently with legacy Fortran, C++ emerging

- GPUs behind fast growth in particle-based commercial CFD
  - New commercial developments in LBM, SPH, DEM, etc.
OpenACC Acceleration of TACOMA at GE GRC

Tri-Hybrid Computational Fluid Dynamics on DOE’s Cray XK7, Titan.

Aaron Vose†, Brian Mitchell*, and John Levesque‡.
Cray User Group, May 2014.

GE Global Research: *mitchellb@ge.com — Cray Inc.: †avose@cray.com, ‡levesque@cray.com.

Abstract — A tri-hybrid port of General Electric’s in-house, 3D, Computational Fluid Dynamics (CFD) code TACOMA is created utilizing MPI, OpenMP, and OpenACC technologies. This new port targets improved performance on NVIDIA Kepler accelerator GPUs, such as those installed in the world’s second largest supercomputer, Titan, the Department of Energy’s 27 petaFLOP Cray XK7 located at Oak Ridge National Laboratory. We demonstrate a 1.4x speed improvement on Titan when the GPU accelerators are enabled. We highlight key optimizations and techniques used to achieve these results. These optimizations enable larger and more accurate simulations than were previously possible with TACOMA, which not only improves GE’s ability to create higher performing turbomachinery blade rows, but also provides “lessons learned” which can be applied to the process of optimizing other codes to take advantage of tri-hybrid technology with MPI, OpenMP, and OpenACC.

Source: https://cug.org/proceedings/cug2014_proceedings/includes/files/pap113.pdf
http://on-demand-gtc.gputechconf.com/gtc-quicklink/e7FnY1
OpenACC Acceleration of TACOMA at GE GRC

TACOMA - Performance

Source: https://cug.org/proceedings/cug2014_proceedings/includes/files/pap113.pdf
http://on-demand-gtc.gputechconf.com/gtc-quicklink/e7FnY1
ASA FUN3D and 5-Point Stencil Kernel on GPUs

- **CPU**: E5-2690 @ 3Ghz, 10 cores
- **GPU**: K40c, boost clocks, ECC off
- **Case**: DPW-Wing, 1M cells
- 1 call of point_solve5 over all colors
- No data transfers in GPU results
- **1 CPU core**: 300ms
- **10 CPU cores**: 44ms (6.8x on 10)

- OpenACC1 = Unchanged code: 2.0x
- OpenACC2 = Modified code : 2.4x (same modified code runs 50% slower on CPUs)
- CUDA Fortran = Highly optimized CUDA version: 3.7x
- Compiler options (e.g. how memory is accessed) have huge influence on OpenACC results
- Possible compromise: CUDA for few hotspots, OpenACC for the rest
- Demonstrated good interoperability: CUDA can use buffers managed with OpenACC data clauses
ANSYS Fluent
NSYS Fluent Profile for **Coupled** PBNS Solver

Non-linear iterations

Assemble Linear System of Equations

Solve Linear System of Equations: \( Ax = b \)

Converged?

- No
  - Accelerate this first
- Yes
  - Stop

Runtime:
- \(~ 35\% \)
- \(~ 65\% \)
ANSYS Fluent Convergence Behavior

Coupled vs segregated solver

Coupling Momentum and Continuity Increases CFD Robustness

FLUENT technology introduces a pressure-based coupled solver to reduce computation time for low-speed compressible and incompressible flow applications.

TRUCK BODY MODEL (14 million cells)

Coupled
Stable convergence for drag coefficient at ~550 iterations

Segregated
Oscillating behavior for drag even after ~6000 iterations
Model FL5S1:
- Incompressible Flow in a Bend
- 32K Hex Cells
- Coupled Solver

Numerical Results:
Mar 2012: Test for convergence at each iteration matches precise Fluent behavior

Error Residuals vs. Iteration Number
ANSYS Fluent and NVIDIA AmgX Solver Library

ANSYS Fluent Software

Read input, matrix Set-up

AMG Operations

AMG Solver 65% of Profile time, operations on GPU

Global solution, write output

GPU

Development Strategy:
- Integration of AmgX

CPU
ANSYS Fluent 15 Performance for 111M Cells

ANSYS Fluent 15.0 Performance – Results by NVIDIA, Dec 2013

- 144 CPU cores – Amg
- 48 GPUs – AmgX

80% AMG solver time

AMG solver time per iteration (secs)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>AMG Solver Time (secs)</th>
<th>Fluent Solution Time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>144 CPU cores</td>
<td>29</td>
<td>36</td>
</tr>
<tr>
<td>144 CPU cores + 48 GPUs</td>
<td>11</td>
<td>18</td>
</tr>
</tbody>
</table>

Lower is Better

2.7 X

NOTE: AmgX is a linear solver toolkit from NVIDIA, used by ANSYS
ANSYS Fluent 16 Performance for 14M Cells

ANSYS Fluent 16.0 Performance – Results by NVIDIA, Dec 2014

Truck Body Model

- Steady RANS model
- External flow, 14M cells
- CPU: Intel Xeon E5-2697v2 @ 2.7GHz; 48 cores (2 nodes)
- GPU: 4 X Tesla K80 (2 per node)

NOTE: Time until convergence
OpenFOAM
Typical OpenFOAM Use: Parameter Optimization

#1: Develop validated CFD model in ANSYS Fluent or other commercial CFD software in production

#2: Develop CFD model in OpenFOAM, validate against commercial CFD model

#3: Conduct parameter sweeps with OpenFOAM (procedure considered by many users to be cost-prohibitive using commercial CFD license models)
Culises: CFD Solver Library for OpenFOAM

Culises Easy-to-Use AMG-PCG Solver:

#1. Download and license from [http://www.FluiDyna.de](http://www.FluiDyna.de)
#2. Automatic installation with FluiDyna-provided script
#3. Activate Culises and GPUs with 2 edits to config-file

![config-file CPU-only](config-file_CPU-only.png)

![config-file CPU+GPU](config-file_CPU+GPU.png)

- solvers {
  - p
  - solver PCG
  - preconditioner DIC
  - tolerance 1e-6
  - ...
  }

- solvers {
  - p
  - solver PCG PCGGPU
  - preconditioner AMG
  - tolerance 1e-6
  - ...
  }
Culises (with AmgX) Coupling to OpenFOAM

Culises Coupling is User-Transparent:

OpenFOAM® (Version 1.7.1/2.0.1/2.1.0)

Discretization: MPI-parallel assembly of linear system remains on CPUs

Linear system \( Ax=b \)

Processor partitioning

Solution \( x \)

CUDA Memory Copy:
- `cudaMemcpy(.... cudaMemcpyHostToDevice)`
- `cudaMemcpy(.... cudaMemcpyDeviceToHost)`

Culises:
- Krylov method
- Multigrid method

(AmgX)

Culises: solves linear system(s) on multiple GPUs
**FluiDyna Culises: CFD Solver for OpenFOAM**

*Culises: A Library for Accelerated CFD on Hybrid GPU-CPU Systems*

Dr. Bjoern Landmann, FluiDyna  

---

**Solver speedup of 7x for 2 CPU + 4 GPU**

- 36M Cells (mixed type)
- GAMG on CPU
- AMGPCG on GPU

---

**DrivAer: Joint Car Body Shape by BMW and Audi**  
[http://www.aer.mw.tum.de/en/research-groups/automotive/drivaer](http://www.aer.mw.tum.de/en/research-groups/automotive/drivaer)

<table>
<thead>
<tr>
<th>Mesh Size - CPUs</th>
<th>9M - 2 CPU</th>
<th>18M - 2 CPU</th>
<th>36M - 2 CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPUs</strong></td>
<td>+1 GPU</td>
<td>+2 GPUs</td>
<td>+4 GPUs</td>
</tr>
<tr>
<td><strong>Culises</strong></td>
<td>2.5x</td>
<td>4.2x</td>
<td>6.9x</td>
</tr>
<tr>
<td><strong>OpenFOAM</strong></td>
<td>1.36x</td>
<td>1.52x</td>
<td>1.67x</td>
</tr>
</tbody>
</table>
ANSYS Fluent Investigation of DrivAer

DrivAer Validation Project: 2012 RANS & SRS Modeling

http://www.aer.mw.tum.de
Courtesy by TU Munich, Inst. of Aerodynamics

Source: ANSYS Automotive Simulation World Congress, 30 Oct 2012 - Detroit, MI
“Overview of Turbulence Modeling”
By Dr. Paul Galpin, ANSYS, Inc.

Available ANSYS models

<table>
<thead>
<tr>
<th></th>
<th>Mesh 1</th>
<th>Mesh 2</th>
<th>Mesh 2 Full Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements</td>
<td>17,493,930</td>
<td>56,569,437</td>
<td>113,136,874</td>
</tr>
<tr>
<td>Nodes</td>
<td>6,778,624</td>
<td>18,992,636</td>
<td>37,901,816</td>
</tr>
<tr>
<td>Max. Aspect Ratio</td>
<td>30,062.5</td>
<td>27,493.5</td>
<td>27,493.5</td>
</tr>
<tr>
<td>Avg. Y* (Avg over car-surface)</td>
<td>0.958305</td>
<td>0.968906</td>
<td>0.952016</td>
</tr>
</tbody>
</table>
Particle-Based CFD for GPUs
## Particle-Based Commercial CFD Software Growing

<table>
<thead>
<tr>
<th>ISV</th>
<th>Software</th>
<th>Application</th>
<th>Method</th>
<th>GPU Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exa</td>
<td>PowerFLOW</td>
<td>Aerodynamics</td>
<td>LBM</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Fluidyna</td>
<td>LBultra</td>
<td>Aerodynamics</td>
<td>LBM</td>
<td>Available v2.0</td>
</tr>
<tr>
<td>Next Limit</td>
<td>XFlow</td>
<td>Aerodynamics</td>
<td>LBM</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Autodesk</td>
<td>Project Falcon</td>
<td>Aerodynamics</td>
<td>LBM</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Prometech</td>
<td>Particleworks</td>
<td>Multiphase/FS</td>
<td>MPS (~SPH)</td>
<td>Available v3.5</td>
</tr>
<tr>
<td>BARRACUDA</td>
<td>BARRACUDA</td>
<td>Multiphase/FS</td>
<td>MP-PIC</td>
<td>In development</td>
</tr>
<tr>
<td>EDEM</td>
<td>EDEM</td>
<td>Discrete phase</td>
<td>DEM</td>
<td>In development</td>
</tr>
<tr>
<td>ANSYS</td>
<td>ANSYS Fluent–DDPM</td>
<td>Multiphase/FS</td>
<td>DEM</td>
<td>In development</td>
</tr>
<tr>
<td>STAR-CCM+</td>
<td>STAR-CCM+</td>
<td>Multiphase/FS</td>
<td>DEM</td>
<td>Evaluation</td>
</tr>
<tr>
<td>AFEA</td>
<td>AFEA</td>
<td>High impact</td>
<td>SPH</td>
<td>Available v2.0</td>
</tr>
<tr>
<td>ESI</td>
<td>ESI</td>
<td>High impact</td>
<td>SPH, ALE</td>
<td>In development</td>
</tr>
<tr>
<td>LSTC</td>
<td>LSTC</td>
<td>High impact</td>
<td>SPH, ALE</td>
<td>Evaluation</td>
</tr>
<tr>
<td>Altair</td>
<td>Altair</td>
<td>High impact</td>
<td>SPH, ALE</td>
<td>Evaluation</td>
</tr>
</tbody>
</table>
FluiDyna Lattice Boltzmann Solver ultraFluidX

http://www.fluidyna.com/content/ultrafluidx

Spin-Off in 2006 from TU Munich

CFD solver using Lattice Boltzmann method (LBM)

Demonstrated 25x speedup single GPU

Multi-GPU ready

Contact FluiDyna for license details
Aoki CFD solver using Lattice Boltzmann method (LBM) with Large Eddy Simulation (LES)
NVIDIA observes strong CFD community interest in GPU acceleration
- New technologies in 2016: Pascal, NVLink, more CPU platform choices
- NVIDIA business and engineering collaborations in many CFD projects
- Investments in OpenACC: PGI release of 15.3; Continued Cray collaborations

GPU progress for several CFD applications – we examined a few of these
- NVIDIA AmgX linear solver library for iterative implicit solvers
- OpenACC for legacy Fortran-based CFD
- Novel use of DSLs, templates, Thrust, source-to-source translation

Check for updates on continued collaboration with NASA (and SGI)
- Further developments for FUN3D undergoing review at NASA LaRC
- Collaborations with NASA GSFC ongoing with climate model and other
Thank you and Questions?

Stan Posey; sposey@nvidia.com; NVIDIA, Santa Clara, CA, USA