REALFLOW
2013

Jumping on the GPU bandwagon with Hybrido2

Angel Tena , Head of RealFlow Technology

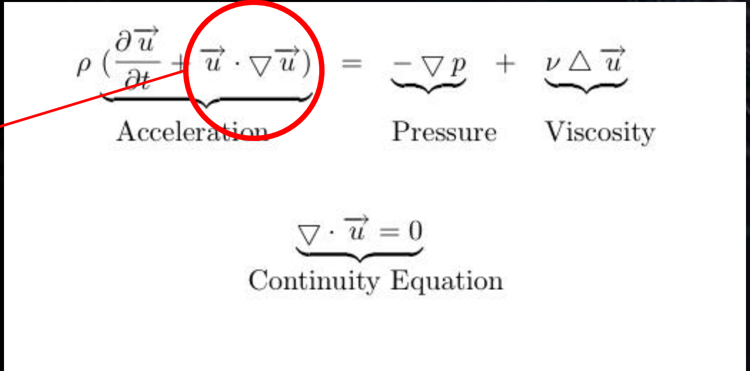
NEXT LIMIT
TECHNOLOGIES

What’s Hybrido2?

- Simulation of medium to large scale scenes
- Simulation of secondary elements (splash, foam, mist, etc)

Improved in Hybrido2?

- Hybrido1 core fluid solver was just grid-based
- Hybrido2 core fluid solver is grid-particle-based. (HY-FLIP)
  - Particles are used to transport the velocity field
  - A computational grid is used to compute the velocity field
  - Velocity rate of change transfer from the grid to the particles



$$\underbrace{\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}\right)}_{\text{Acceleration}} = \underbrace{-\nabla p}_{\text{Pressure}} + \underbrace{\nu \triangle \vec{u}}_{\text{Viscosity}}$$

$$\underbrace{\nabla\cdot\vec{u} = 0}_{\text{Continuity Equation}}$$

Particles

REALFLOW

HY-FLIP pipeline

1. Advect particles using the grid velocity field
2. Transfer the particle velocities to the grid
3. Make a copy of the grid velocity field
4. Label grid cells
5. Create fluid surface distance field
6. Sweep velocity field
7. Apply boundary conditions
8. Make the velocity field divergence free
9. Sweep velocity field
10. Transfer rate of change of grid velocity field to particles

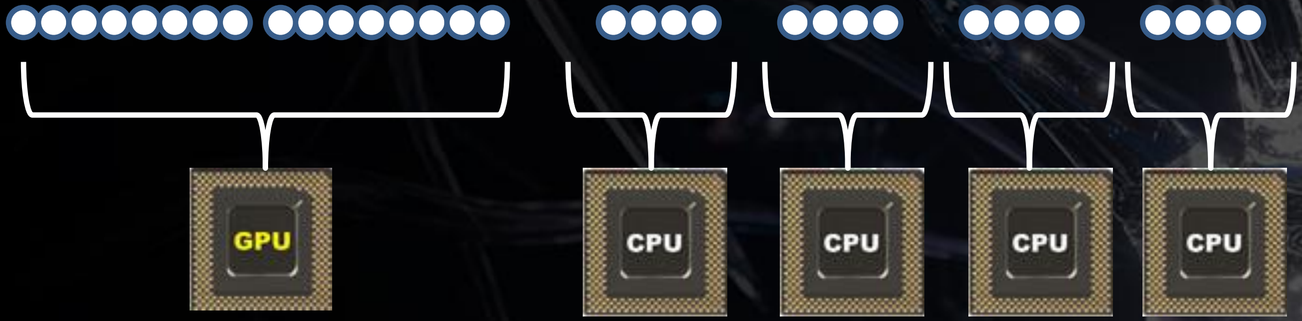
NEXT LIMIT
TECHNOLOGIES

HY-FLIP on the GPU?

- Problems
  - Several parts of the pipeline are very difficult to implement on the GPU efficiently
  - RealFlow scenes are usually over 12GB of RAM
  - RealFlow user can interact with the solver data via scripting
  - RealFlow needs to cache at least once every simulation frame

Yes, HY-FLIP on the GPU

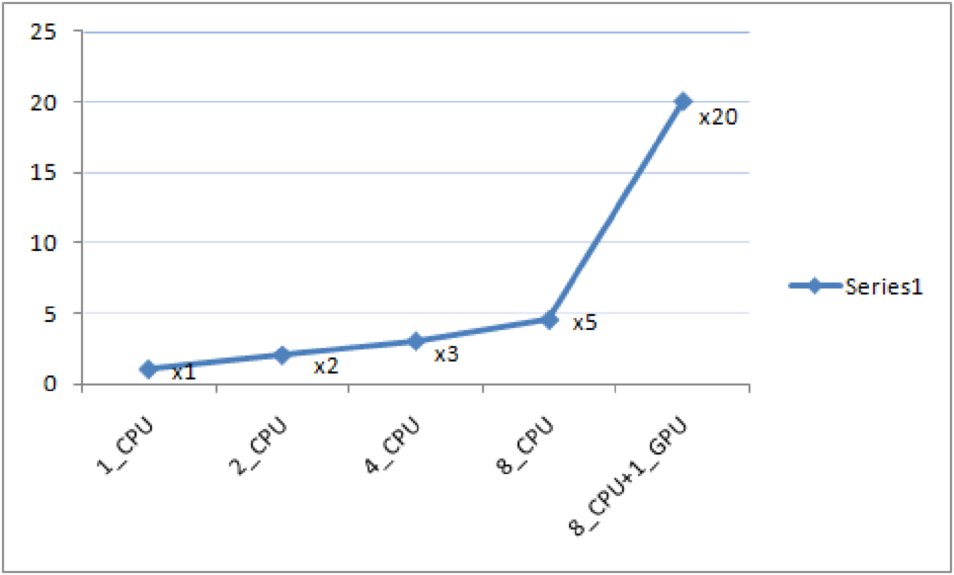
- Only the stages using particles on the GPU
- Particle slicing



REALFLOW

Speed-up example

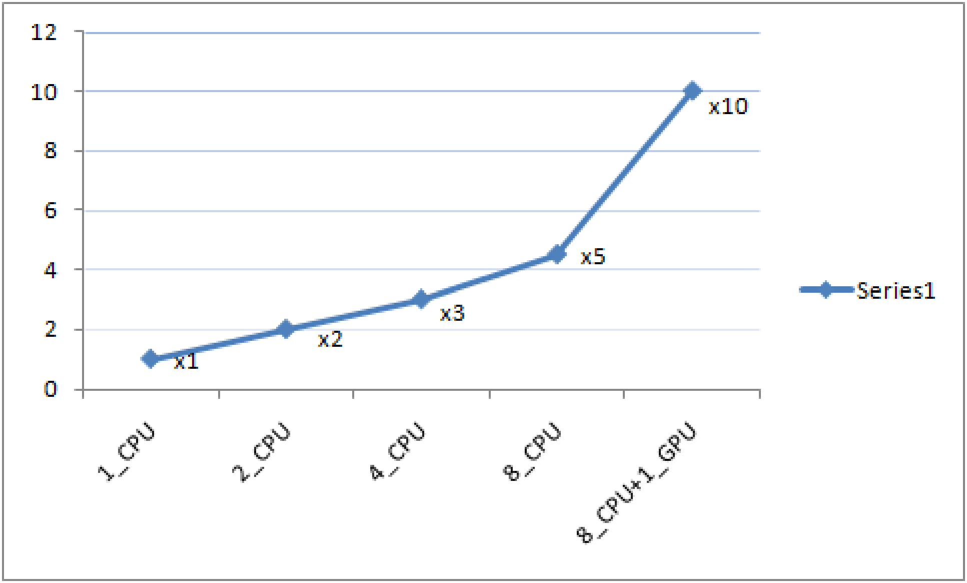
- Advecting particles using MacCormark
  - 65 million of particles
  - Grid 200x200x200
  - 5 steps
  - Intel® Core™ i7-2600K
  - NVIDIA® Quadro 6000



NEXT LIMIT TECHNOLOGIES

Overall speed

- **Moving stages (1), (2) and (10) to the GPU**
  - 65 million of particles
  - Grid 200x200x200
  - 5 steps
  - Intel® Core™ i7-2600K
  - NVIDIA® Quadro 6000



Some implementation bits

- OpenCL 1.1 (CUDA is in the roadmap)
- Works with both dense and sparse grids
- Loss of performance when using sparse grids
- Dynamic load balancing
- Dynamic scheduling when a job can’t be completed on the GPU due to memory overflow

Hybrido2 coming soon

REALFLOW 2013

Thank you!!!

NEXT LIMIT TECHNOLOGIES