Bringing GPU capabilities in Scilab

June 16th, 2010

Presentation

Sylvestre Ledru

- In charge of R&D projects
- Responsible for GNU/Linux, Mac OS X and Unix version
- Developer
- Community management
- ...

GPGPU in Scilab

What is GPGPU?

- Use graphic cards for numerical computing purposes
- Much more powerful than CPU on some algorithms
- But hard to use...

Two main solutions

- CUDA by Nvidia for Nvidia
  Focused on GPU computing
- OpenCL by the Khronos group
  More general

Why GPU in Scilab?

- To improve computation time
- Perfect to massively parallel algorithms
- To simplify its uses

Advantages compared to ad-hoc devs

- Take advantages of all Scilab features (high-level, easy to use...)
- Hide a part of the complexity
- Integration in a current software solution

Alpha version of Scilab GPU module

Description

- One goal: leverage GPU capabilities from Scilab
- Started in the context of the Google Summer of Code 2009
- Continued within the OpenGPU context (System@tic and Cap Digital)

Features

- Handle both CUDA & OpenCL technologies
- Similar profiles and usages
- GPU functions (kernel) are compiled and transfered by the module
- Explicit transfert of variables
- Available on the Scilab forge : http://forge.scilab.org/index.php/p/sciCuda/

Features

- CUDA & OpenCL managed transparently with same profiles:
  - cudaAlloc vs openclAlloc
    cudaToGpu vs openclToGpu
    buildCuda vs buildOpenCL
    cudaApplyFunction vs openclApplyFunction
    ...

Code example - kernel

```
__global__ void someSimpleKernel(double* src,double*
dst, int numberOfElement)
{
  int idx=threadIdx.x+blockIdx.x*blockDim.x;
  if(idx<numberOfElement)
  {
       dst[idx]=src[idx];
  }
}
```

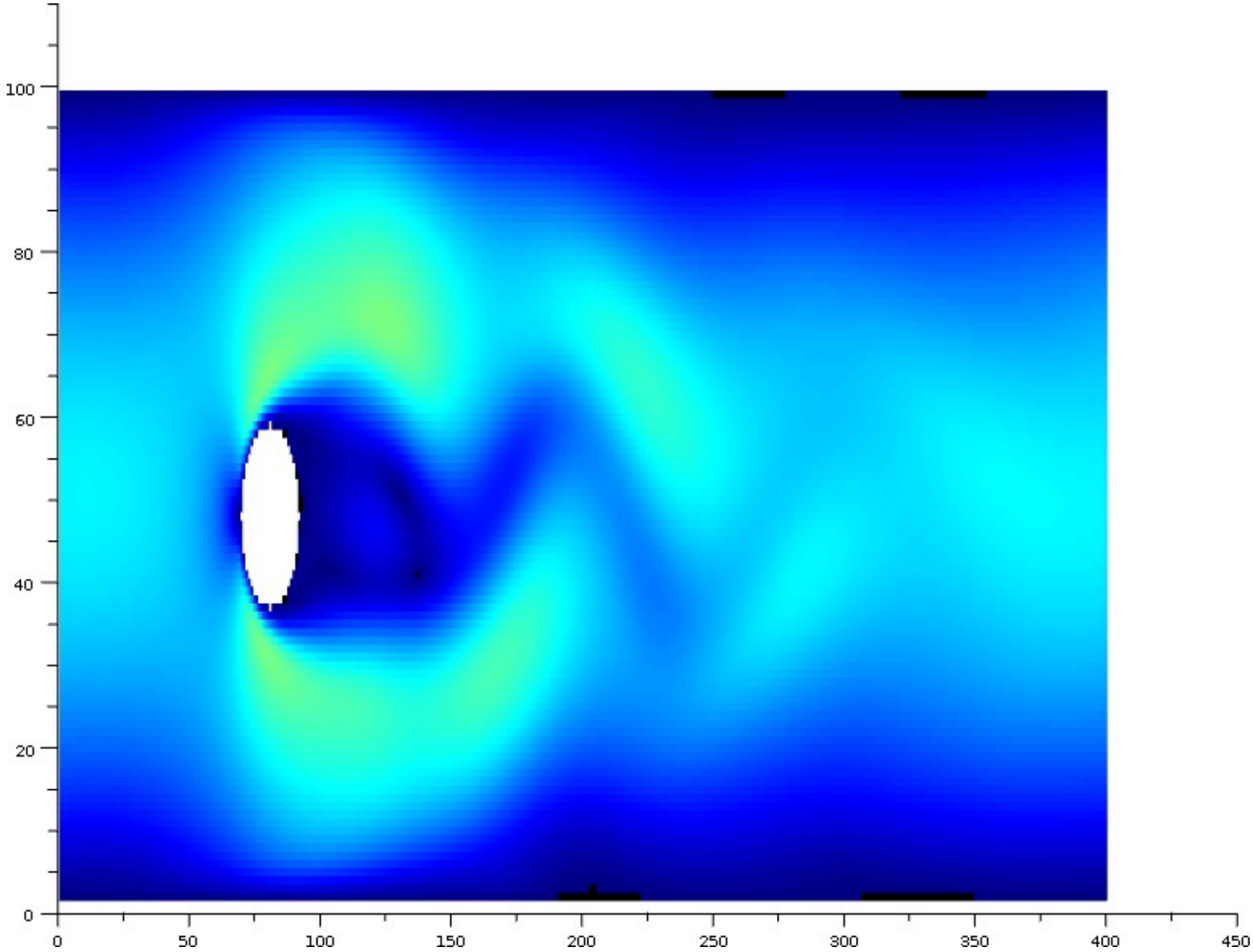
Code example - Scilab code

```
buildCuda(abs_path+"simple.cu");

A_host=rand(256,256);
A=cudaToGpu(A_host);
B=cudaAlloc(256,256);

kernel=cudaLoadFunction("simple.ptx","someSimpleKernel");
lst=list(A,B,int32(256*256));
cudaApplyFunction(kernel,lst,128,1,256*256/128,1);
B_host=cudaFromGpu(B);
```

Example: Channel flow past a cylinderical obstacle



Future

- A first tagged release
- More transparent uses
- Code generation to GPU
- Xcos integration
- Linear algebra features based on cublas

Thanks for your attention

www.scilab.org

Scilab