ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Introduction to GPU Computing

Autumn Semester 2013

Prof. P. Koumoutsakos, G. Tauriello, C. Conti
ETH Zentrum, CLT
CH-8092 Zürich

Getting Started

Issued: 17.9.2013

Brutus Cluster

Within this course we will work with the machines in the computer rooms and remotely connect to the GPUs of the Brutus cluster.

In order to gain access to the Brutus cluster, if you do not already have it, you should request an account before the start of the course at the following address: `https://www1.ethz.ch/id/services/list/comp_zentral/cluster/brutus_acc_req_pre_EN` indicating "Engineering Tools - Introduction to GPU Computing" under project.

You can find further information on the Brutus cluster at `http://brutuswiki.ethz.ch/brutus/Brutus_wiki`.

Once you have an account, during the lectures you can request an interactive job on a GPU node with the following commands:

```
$ module load cuda
$ bsub -W 4:00 -U gpuclass -R gpu -Ip /bin/bash
```

The "-U gpuclass" flag is only valid during the class and allows to have a higher priority to run on the GPU nodes on brutus.

The output of the above commands should be:

```
GPU job.
Job <14304333> is submitted to queue <pub.8h>.
<<Waiting for dispatch ...>>
```

after which a shell should start within one minute.

For the class we will rely on CUDA 4.29 (the default version installed and running on the Brutus cluster). Most of the informations can be found at the following websites: Nvidia CUDA home page (with a lot of resources for CUDA GPU programming): `https://developer.nvidia.com/category/zone/cuda-zone` Nvidia CUDA programming guide: `http://docs.nvidia.com/cuda/index.html`

The webpage of the class can be found at the following address: `http://www.cse-lab.ethz.ch/index.php/teaching/42-teaching/classes/576-etvgpufall2013`

On brutus you can unzip our collection of test codes by executing

```
$ unzip /cluster/home/infk/gerardot/EngTools/Code.zip
```

This will create a folder `Code` at your current location containing the test files. Similarly you will be able to execute the following to get the code for the boids:

```
$ unzip /cluster/home/infk/gerardot/EngTools/boids.zip
```

In order to visualize your data you might want to transfer files from Brutus to your local machine. From your local machine, use the following command:

```
$ scp brutus.ethz.ch:/cluster/mavt/username/.../Boids.xyz .
```

where `username` is your nethz username and `...` is the path within your home folder on brutus where you stored your data. The command will take the file called Boids.xyz and copy it into the position from where it's called.

VMD

In order to visualize the moving particles, you are going to use VMD (`http://www.ks.uiuc.edu/Research/vmd/`), a visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics. VMD is installed on the ETH computers. You can find some test files on the class webpage. You can unzip the file `VMD.zip` with the command `unzip VMD.zip`.

In order to launch the program, open a terminal and enter:

```
$ vmd
```

At this point, you should see the main VMD application window and the output window (See Figure 1). In a first step, you are going to open and visualize the provided dataset `test.xyz`. Note that datasets are referred to as "molecule" in VMD.

To load the test dataset, open the *Molecular File Browser* dialog (File->New Molecule...), locate the test file on your local file system and make sure to set the file type to XYZ (See Figure 2).

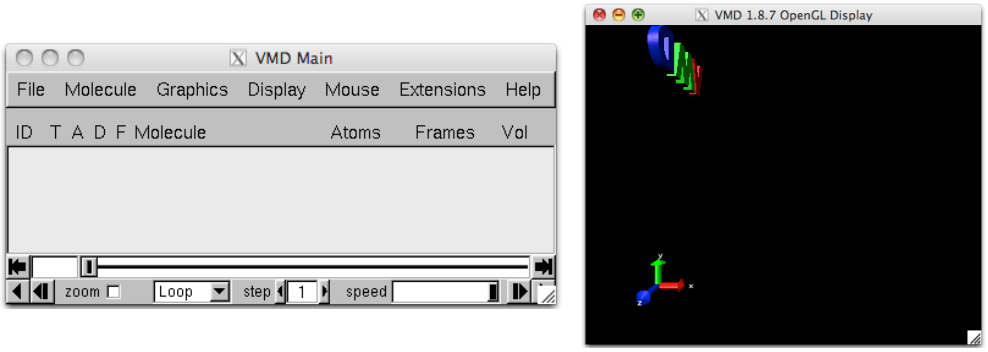

Figure 1: VMD: Main window (left) and output window (right)

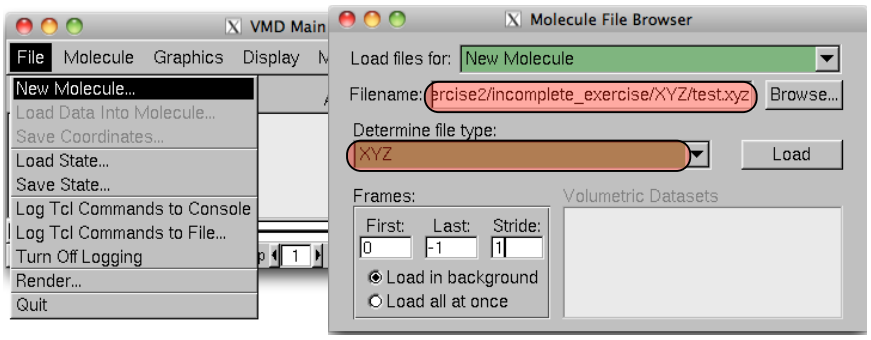

Figure 2: VMD: Loading a data set.

In order to adjust the graphical representation of the data, open the *Graphical Representation* (Graphics->Representations) dialog as indicated in Figure 3. Before you start changing the graphical representation, make sure to reset the timeline to its initial position on the very left (Figure 3, left). Set the *Coloring Method* of the particles to Position->Radial and the *Drawing Method* to VDW. To adjust the radius of the displayed spheres, play with the *Sphere Scale* parameter (Figure 3, bottom). Also feel free to play with other display settings.

In the output window, you should now observe an image similar to the one displayed in Figure 4. Use your mouse wheel on the output window to zoom on the image. By clicking and dragging the image in the output window, you can change the view angle. Use the time line (Figure 3, left) to advance the simulation in time. After successful completion of the visualization of the test dataset, repeat these steps for the simulation output data (`Boids.xyz`). Use the built in functionality of VMD to dump images that can be combined into a movie using the `makemovie.sh` script. Open the VMD Movie Generator (Extensions->Visualization->Movie Maker). In the *Movie Generator* dialog, set the *Movie Settings* to Trajectory, the *Format* to JPEG frames (ImageMagick), the working directory to `output_images` and the *Name of movie* to "boids". Click the *MakeMovie* Button to render the images. This will generate files named `final.boids.*.jpg` within the `output_images`. In order to combine these images into a movie, invoke the `makemovie.sh` script. You find the final movie inside the `final_movie/` folder.

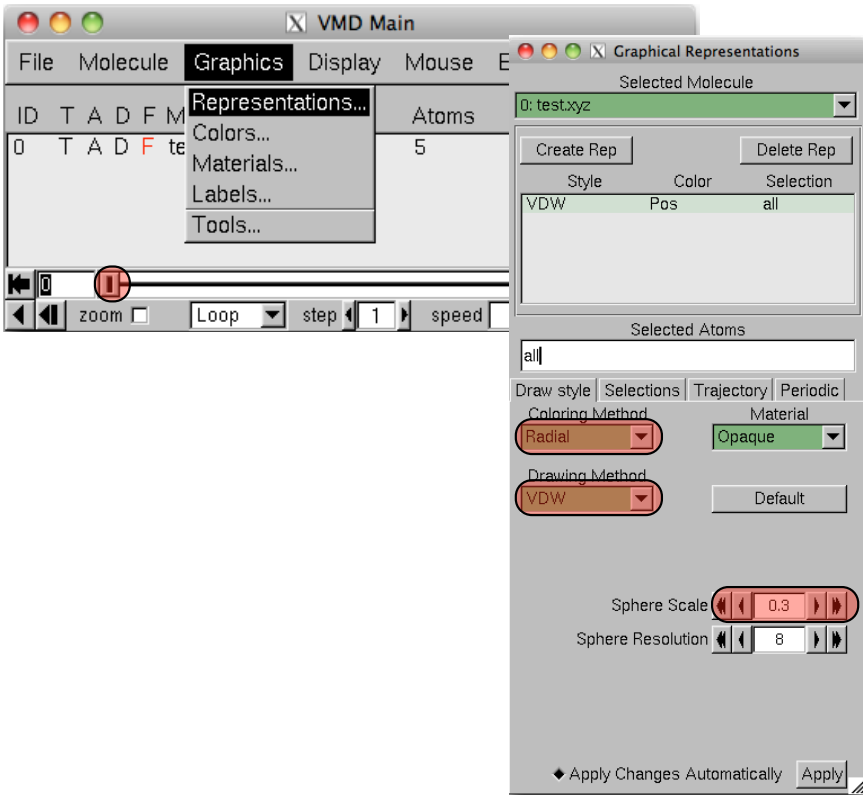

Figure 3: VMD: Main window (left) and output window (right)

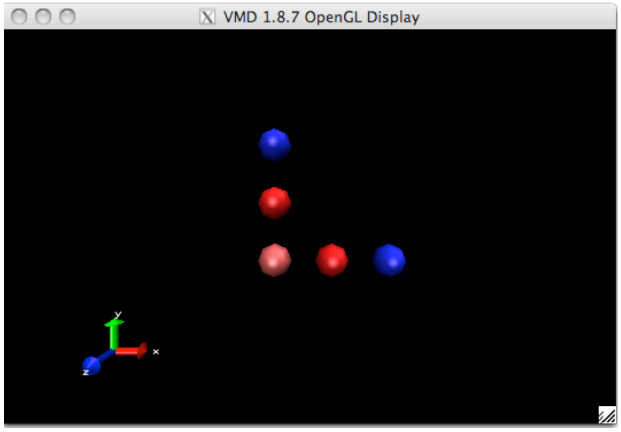

Figure 4: VMD: Main window (left) and output window (right)