Rendering simulations using VMD MPI on Stampede2 Supercomputer

The following tutorial introduces the use of high-performance massively parallel comput- ing to render high quality figures, from molecular dynamics (MD) trajectories, using VMD MPI. The capability of VMD to render scenes enabling shadows and ambient occlusion tech- niques is provided by the Tachyon ray tracer. As a result, figures display better description of shape and depth of molecular systems.
The example corresponds to the scene rendering of 1 μs of MD trajectory using VMD MPI. The use of parallel computing, in combination with fast multiprocessor ray tracer, enhances performance of large-scale trajectories rendering.

1 Setting up VMD display settings

The initial step to render any trajectory is setting up the preferred view and representation of the system. Since VMD builds on supercomputers contain default display settings, visual representations need to be adjusted according to the system of study.

1.1 Loading system on local machine

- Type on the *vmd* to open VMD on your local machine
- Go to Extensions → Tk Console

On the Tk console load the topology and trajectory file

```
> Main <  ( test mpi stampede 2 ) 1 % set molid [ mol new coordinates.pdb ] 0
> Main < ( test mpi stampede 2 ) 2 % animate delete all
> Main < ( test mpi stampede 2 ) 3 % mol addfile  production.dcd wait fo  a l l
```

The command mol new assigns and ID to the loaded structure. The variable *molid* contains the molecule ID, which in turns allows the user to specify the target molecule.

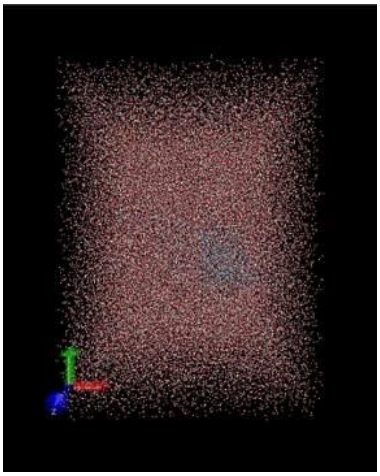

Figure 1: Default visualization on VMD.

1.2 Setting visualization commands on console

The additional changes on the display settings are going to be performed through the GUI of VMD. In order to keep track of the commands to add to the final rendering script, the Tcl commands are printed in the Tk console.

- Go to File → Log Tcl commands to console
- Set Display → Orthographic
- Set Display → Axes → Off
- Set Graphics → Colors → Categories: Display, Names: Background, Color: White
- Graphics-> Representations. . . -> Coloring method: Secondary Structure, Drawing Method: New Cartoon, Material: AOChalky, Resolution: 50

As a result, commands are printed on the Tk console.

```
Info) Logging commands to 'console'.
Info) # VMD for LINUXAMD64, version 1.9.4a7 (July 12, 2017)
Info) # Log file 'console', created by user fabiog
Info) display projection Orthographic
Info) axes location Off
Info) menu color off
Info) menu color on
Info) color Display Background white
Info) mol modselect 0 3 protein
Info) display resetview
Info) mol modcolor 0 3 Structure
Info) mol modstyle 0 3 NewCartoon 0.300000 10.000000 4.100000 0
Info) mol modmaterial 0 3 AOShiny
Info) mol modstyle 0 3 NewCartoon 0.300000 15.000000 4.100000 0
>Main< (test_mpi_stampede2) 14 %
```

Figure 2: Snapshot of commands printed on the console.

In addition, the orientation and view of the system be adjusted using the commands *rotate* and *scale*. A visual inspection of the molecule view is recommended.

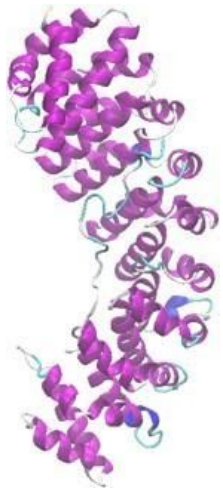

Figure 3: Visual inspection of the system.

2 Supercomputer assisted visualization

Login to Stampede2 from the Linux command console via *ssh*. In addition, copy the local directory to the work space using *scp*.

```
localhost $ ssh myusername@stampede2.tacc.utexas.edu
localhost $ scp −r render_mpi_stampede2.tar.gz myusername@stampede2.tacc.utexas.edu
```

After merging the files, the visualization settings are loaded to the submission script *render_figures_mpi.tcl*, under the *mpianalyze* procedure.

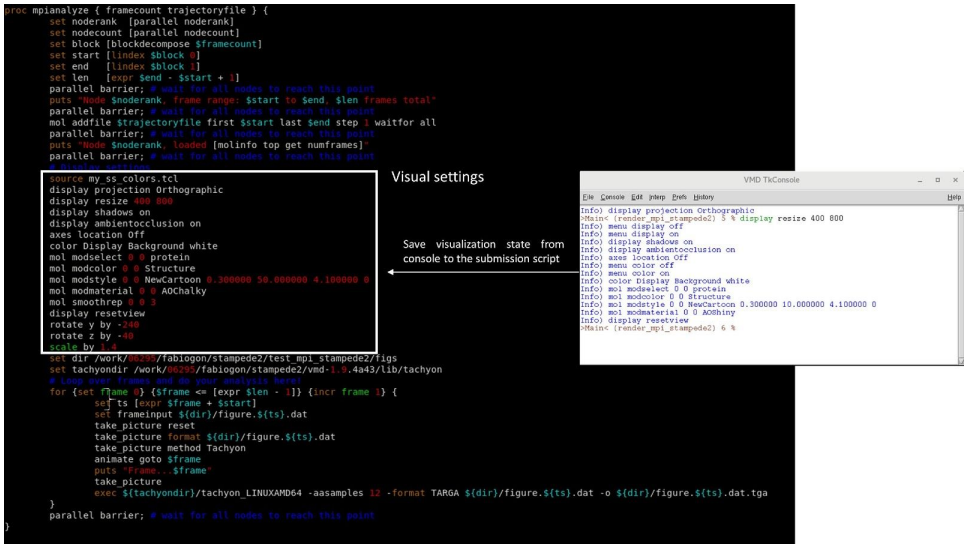

Figure 4: Addition of display settings to submission script.

After modifying the submission script, the job is then launched using the Slurm Workload Manager using the provided runbatch.sh script. The job status can be monitored using the *watch* command.

```
localhost . /runbatch_mpi.sh
localhost watch −n 20 squeue −u username
```

Figure 5: Rendered scene with Tachyon