Hardware Acceleration of the 3D Discrete Element Method

Depeng Yang, Akila Gothandaraman, Junqing Sun, Gregory D. Peterson
Electrical and Computer Engineering, The University of Tennessee

Abstract

The discrete element method (DEM) is used to accurately predict the motion over time of a large number of particles such as molecules or particles of soil. The computational demands for DEM simulations are quite significant, so improved performance could address a spectrum of scientific and engineering applications. This paper proposes an accelerated implementation for the DEM using FPGAs.

1. Introduction

The discrete element method is a popular technique for simulating large numbers of particles and their interactions. The approach is a generalization of the finite element method, and is used for particles ranging from molecules for biological or chemical applications to granules for civil engineering and soil mechanics. With DEM, a time evolution of the collection of particles is simulated by computing the forces on each particle at each time step, computing the associated acceleration and velocity, and then updating the positions of the particles. The particles simulated can be idealized spheres or have irregularly shaped structure. The particles can be homogeneous or heterogeneous with respect to size, shape, or behavior. The DEM simulation also may include boundary conditions that can be simple (e.g., a wall or surface) or complex (e.g., soil strata or flows through structures). The forces that act on the particles can include forces from a distance related to gravitational or Coulombic/electrostatic forces, van der Waals forces, or Pauli repulsion. When particles are in contact with each other or a boundary, forces can include friction, damping, recoil, cohesion, and adhesion. Depending on the complexity of the system and willingness to invest the computational resources, some or all of the forces listed above may be included with a DEM simulation at different levels of fidelity.

In this paper we explore a feasible, scalable way of using FPGAs to implement a 3D DEM model with simple boundary. This dedicated hardware architecture is parallel, pipelined, and can be scaled up. The hardware circuit performs calculations for 1024 particles in a closed cube while considering gravity and damping forces from contact. The corresponding software running in PC is modified to validate the results from hardware and to make performance comparisons. The speedup is 18 at 90MHz for this small problem. With more particles in this model, the speed up will greatly increase.

2. The discrete element method (DEM)

The open source 3-D code YADE (Galizzi and Kozicki 2005) [1] and commercial 2-D code PFC2D are popular simulation software codes using DEM. In spite of subtle differences between the two codes, the same basic approach is used. The steps in DEM are summarized below [4]:

(1) Initialize the processing, which includes determining the time step, initial position, velocity, density, and radius of particles.

(2) Fetch the next two particles. Whether contact occurs is detected by calculating the distance between the two particles.

(3) If the particles are in contact or touch a boundary, the composite force is calculated and accumulated for the particles. If not in contact, return to step (2) to fetch the next pair of particles.

(4) Return to step (2) to fetch the next particles.

(5) When all particle pairs have been considered for contact, update position and velocity for each particle. By applying Newton’s Second Law (F=ma), new velocities and positions are computed for the next iteration.

The forces considered in DEM are caused by conflict and boundaries, gravity, and damping force. In this case, local damping is used, which represented as fellow:

$$F_{damping} = -\beta \,|\, F_{ub} \,|\, \mathrm{sgn}(v) \qquad (1)$$

Where β is the local damping constant; F_{ub} is the composite force, which is the vector sum of all applied forces on a single object (either particle or boundary); the term $\mathrm{sgn}(v)$ is +1, 0, or -1, according to the value of v as positive, zero, or negative, respectively. This damping hardware architecture is described next.

3. FPGA Implementation of the DEM

The hardware architecture is fully pipelined and divided into three stages: i) parallel composite force calculation, ii) force accumulation, and iii) position and velocity update. Parallel composite force calculation includes a contact detect unit, a contact force and damping unit, and a boundary process unit. Because the force on each particle may come from collisions/conflicts, damping, and interactions with boundaries, these forces are computed in parallel in the first stage and accumulated in the next stage. The third stage is responsible for updating and saving the new position and velocity for each particle for the next time iteration. The architecture is shown in Figure 1.

A contact detect unit in the 3D DEM model can be expressed as below:

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - (R_1 + R_2)^2 > 0$$

where the particles have radii of R_1 and R_2, respectively. If two particles are in contact, the contact and damping force is computed and forwarded to the force accumulator unit. Otherwise, this force is still computed but omitted.

The boundary unit checks whether any particle touches a boundary and this corresponding force is calculated. We consider an abstract boundary modeled as many particles at fixed location with special properties for certain boundary shapes. In this way, forces resulting from a boundary can be viewed as from normal particles and processed in the same pipeline without a need to build up a special architecture for boundary.

The conflict/contact force and damping unit handles the computation to realize equation 1. We represent data in the FPGA implementation using 52 bit fixed point to provide precision as good as in 3D DEM simulation software for the civil engineering/soil engineering applications of interest.

4. Performance

The hardware design was implemented to target the Xilinx Virtex 2 Pro FPGA contained in a XUP board (XC2VP30) and also tested with a Xilinx Virtex4 development board (XC4VLX60). The initial position, velocity and boundary information was transferred through RS232 and saved in block RAM in FPGA to initialize the DEM hardware simulation. Concurrently a version of DEM software was modified so that fixed-point hardware could be emulated. The software running time was measured on the Unix platform using system timing functions to make a comparison with hardware. The speedup for simulating 1024 particles with the Virtex 4 is below:

Software Performance		**Hardware Performance**	
Computer	**Time**	**System Clock**	**Speedup**
PC	**0.1884s ~0.3749s**	**100Mhz**	**17~34**

The PC test system is a dual core, 2.80GHz P4 with 512KB cache and 1GB memory. This PC execution time heavily depends on initial positions. The circuit was tested at 100MHz due to platform constraints, but the Xilinx tools report more than 200MHz for a working frequency. Algorithms can be employed for reducing contact detection time in software [2] by up to 30 times. But in most cases, the hardware performance is still better than software version. The hardware circuit uses 15365 4 input LUTs, which can be integrated into one FPGA. With more particles calculated in system, we can achieve higher speedup in hardware DEM simulator.

5. Conclusions

This paper presents a hardware accelerated implementation of the 3D discrete element method that achieves a speedup of 18 over software. The core for the design will be ported to the Cray XD1 as future work.

References

[1] O. Galizzi and J. Kozicki, YADE-Yet Another Dynamic Engine, 2005, http://yade.berlios.de

[2] E.G. Nezami, Y.M.A. Hashash, D. Zhao, J. Ghaboussi "A fast contact detection algorithm for 3-D discrete element method", Computers and Geotechnics, 2004

[3] N. Azizi, I. Kuon, A. Egier, A. Darabiha, and P. Chow, "Reconfigurable Molecular Dynamics Simulator", *IEEE FCCM*, 2002.

[4] B. Carrión Schäfer1, S.F. Quigley, A.H.C. Chan. "Analysis and Implementation of the Discrete Element Method using a Dedicated Highly Parallel Architecture in Reconfigurable Computing", *IEEE FCCM*, 2002.

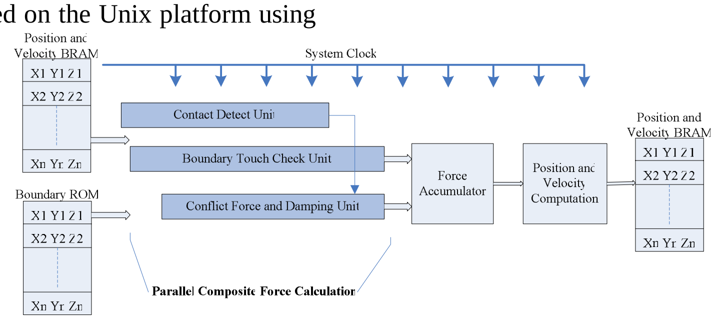

Figure 1 Hardware architecture for DEM simulation