

PARALLEL ENSEMBLE MONTE CARLO FORMULATION ON A CLUSTER FOR TRANSIENT CHARACTERISTICS OF GaAs

R Umar and A.P Othman

*School of Applied Physics
Science and Technology Faculty
Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor, Malaysia*

r_lan80@hotmail.com

ABSTRACT

The nature of the Ensemble Monte Carlo method where it is base on the successive and simultaneous calculation of the motions of many particles during a small time interval makes it a very suitable candidate for cluster or parallel implementation. Simulating charge transport phenomena in solid state devices is usually very costly in term of computational requirements when implemented on a single processor personal computer (PCs). This is due to the fact that to ensure numerically sound simulation results, large ensembles of particle need to be simulated, something in the order of 100 000 particles. The major problem encountered in this type of simulation using the conventional ensemble or sigle particle monte carlo method is the long computational time even on the fast 2.5 MHz PCs. An altenative and cost effective solution to this problem is the application, or a computer cluster network in a master-slave model. In managing a cluster network, we have been using the Parallel Virtual Machine (PVM) standards. In this paper we report an implementation of a parallel algorithm using parallel ensemble MC simulation for device simulation with particular emphasis on transient transport phenomena in GaAs. In this simulation we present the transient charge velocities as a function of applied field at various particle numbers and the different computational times taken when in cluster enviroment.

INTRODUCTION

In this paper, we report our progress in the developement of an Parallel Ensemble Monte Carlo (EMC) particle-base simulator for modeling semiconductor devices simulation in GaAs. The simulation tool used is an EMC. The Monte Carlo method is an approach to solve the problem the electron transport base on the numerical simulation of the motion of a large particle ensemble where the collision processes are introduced by generating appropriately distributed random numbers. With the other word The EMC method usually reffered to stochastic techniques which use random number generation. In the case of simulation involving particle transport phenomena in solid state. The EMC method is use to solving the Boltzmann Equation directly, without making assumption on the distribution function. With EMC method we create the Parallel EMC, where the code was written entirely with c/c++. Research also focused on parallel computation strategies and algorithms and learning how to implement these concepts using PVM, a set of libraries that allow programmers to write C and C++ parallel code. In addition, the research included learning how to maintain and install

nodes of the cluster. The parallel EMC program have been run by PVM software with RedHat Linux 7.3.

The Ensemble Monte Carlo Device Simulation

Monte Carlo simulation of carrier transport in semiconductor are base on folowing the time evolution of an ensemble of particle through the material in both real and momentum space. The motion of each particle in ensemble has to be simulated in turn, for the full duration of the simulation. It is assumed that this particles are effectivly independent which makes the MC simulation well suited to parallel implementation to reduce computation time. Figure 1 show step by step how the drift velocity and scattering electron mechanism in Ensemble Monte Carlo.

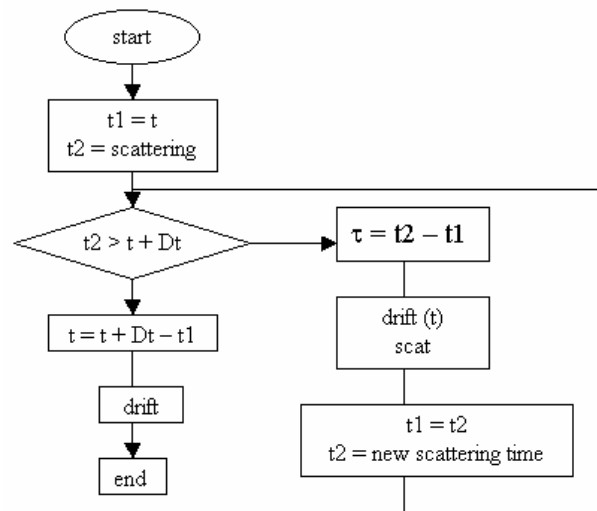


Figure 1: Flow chart for a typical Monte Carlo device simulation algorithm.

The Parallel Virtual Machine

The PVM system was originally developed at the Oak Ridge National Laboratory. The current versions of PVM are being jointly developed and maintained by a number of research teams.

Under the PVM parallel processing model, heterogeneous computer resources are made available as one large distributed memory computer. The overall computational problem is divide into a number of co-functional, possibly identical, process. Each node of the virtual machine executes one or more of these process. The cooperating process communicate via a message passing protocol defined by the pvm software. A message passing server is installed on each node and handles all communication among the nodes of the virtual machine. The client code makes use of PVM Application Program Interface calls in order to send and receive messages from other processes running on the virtual machine.

The Implemented of virtual machine used for the simultions disscused in this paper utilizes a large number of Pentium (1.8GHz) machines. The node machines are linked via a standard

10BaseT Ethernet. Groups of 9 machines are linked together via hubs. These groups are in turn connected by a 10x10Mbps switch.

Implementation on Parallel Virtual Machine

After we have done written program Ensemble Monte Carlo code in C/C++, we must coding in parallel code. Before that we must study about PVM, how to compile, run and so on. In PVM program we have two program, one program is master and second program is slave. The master will send task program to slave with way partition equally. Examaple 100 particle will be send 50% for each processor. The figure 2 show how the Master program send task to thier slaves. After the slave completely run that program, it will send resul to the master, and the master colect the result and display the running time aplication.

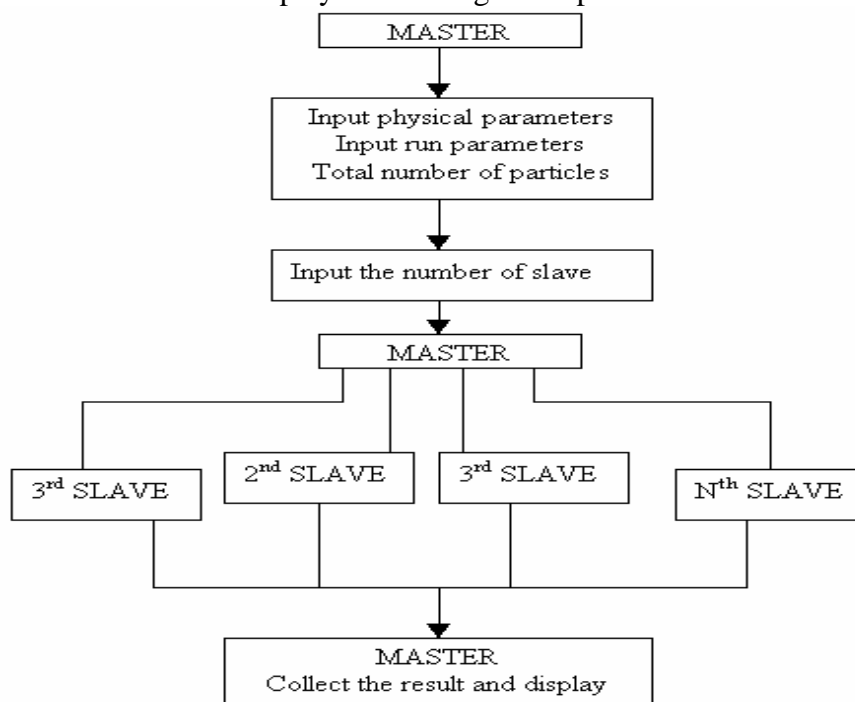


Figure 2: Flow chart for the MC-PVM device simulation algorithm.

Parallel Execution Time And Communication Time

The important factors effecting performance in message-passing on distributed memory computer systems are communication patterns and computational/communication ratios. The communication time will depend on many factors including network structure and network contention. Parallel execution time t_{para} is composed of two parts, computation time (t_{comp}) and communication time (t_{comm}). t_{comp} is the time to compute the arithmetic operations such as multiplication and addition operations of a sequential algorithms. Analysis of the t_{comp} assumes that all the processors are the same and the operating at the same speed. t_{comm} will depend upon the size of message.

$$t_{comm} = t_{startup} + m_{data} + t_{idle}$$

If the number of iterations b , and size of the message for communication m , the formulae for communication time is as follows,

$$t_{comm} = b(t_{startup} + m t_{data} + t_{idle})$$

where $t_{startup}$ is the startup time (message latency). $t_{startup}$ is time to send a message with no data. It includes time to pack the message at the source and unpack the message at the destination. The term t_{data} is the transmission time to send one data word. $t_{startup}$ and t_{data} are assumed as constants and measured in bits/ sec. t_{idle} is the time for message latency, time to wait for all the processors to complete the process. It is also a means of quantifying the degree of load imbalance in the parallel algorithm. In order to estimate the coefficients $t_{startup}$ and t_{data} , a number of experiments were conducted for different message sizes. Two linear equations can determined these two coefficients.

Experimental Verification

In order to assure that the simulation and our cluster is working properly, we performed several test case using Pi program and for simulation test we also performed several test cases using bulk Si and GaAs samples. In our theoretical model, we include all relevant scattering mechanisms for this material (intravalley acoustic and polar optical phonon scattering, and intervalley scattering between conduction band Γ and L. Figure 3 and 4 shows the velocity when applied to electric field with different temperature and different particle. Figure 3 and Figure 4 show the dependence of the temperature and particle in the drift process.

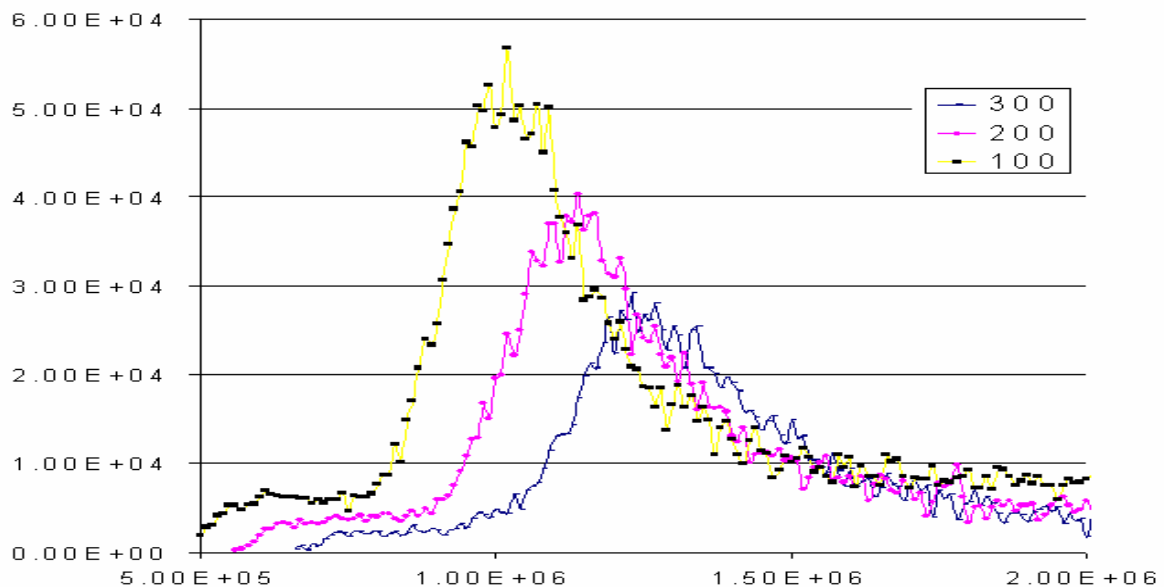


Figure 3: Velocity[cm/s] vs Electric Field [V/m] 10 000 particle

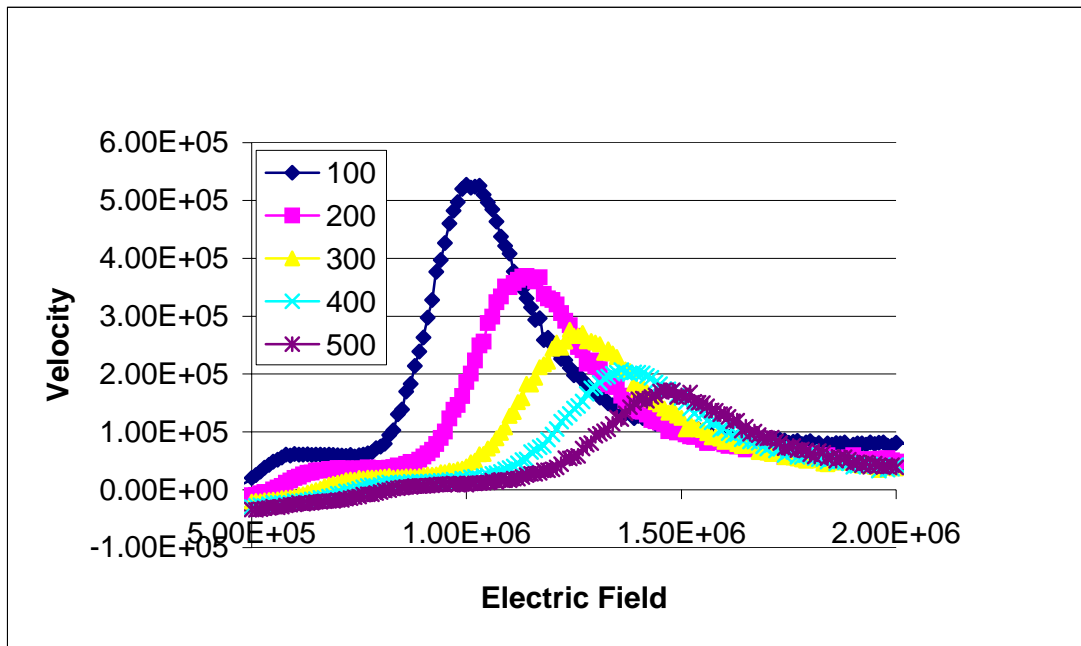


Figure 4: Velocity [cm/s] vs Electric Field [V/m] for 100 000 particle

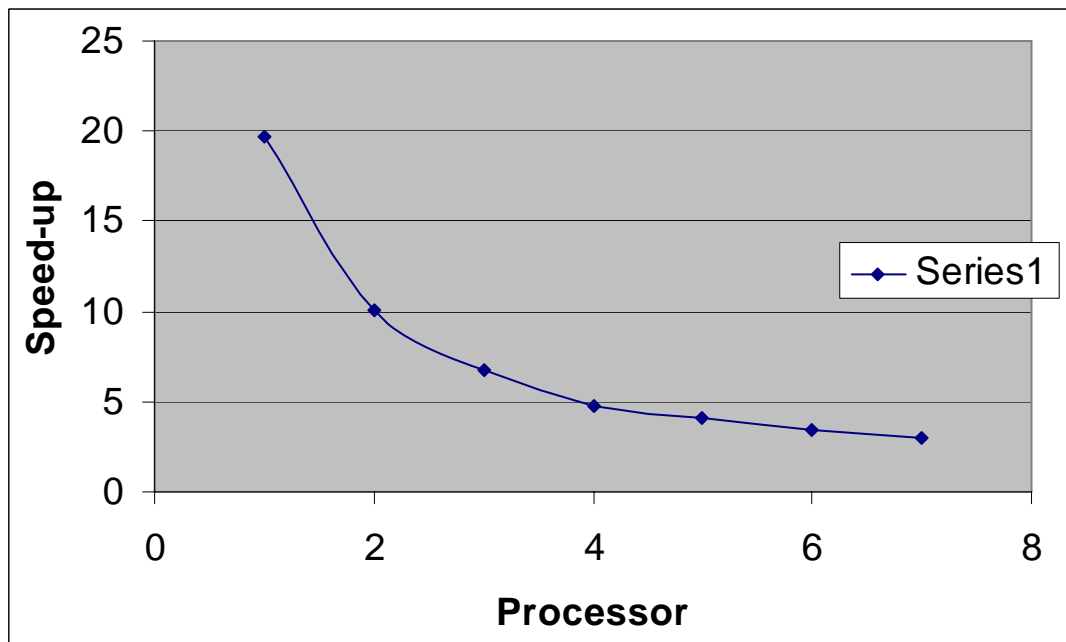


Figure 5: Speed-up vs Processor

REFERENCES

- [1] A. P. Othman, R. Umar and A. J. M Adnan. Implementing Parallel Ensemble Monte Carlo on a LINUX Cluster. Proceedings of the ATC2003 in Theoretical Sciences 2003 (CATS 2003)
- [2] Norma Alias, Dr.Mohd Salleh Sahimi Mohamed, Prof.Abdul Rahman Abdullah. Communication Issues For Parallel Alternating Group Explicit Methods On A Distributed Memory Architecture. Proceedings of the ATC2003 in Theoretical Sciences 2003 (CATS 2003)
- [3] C. Jacoboni, and L. Reggiani. (1983) The Monte carlo method for solution of charge transport in semiconductors with application to covalent materials Rev. Mod.Phys. 55 (3), 645
- [4] T. Kurosawa (1966) . Monte carlo Simulation of hot electron problems, J. Phys. Soc. Suppl, 21, 527
- [5] C. Jacoboni and P. Lugli (1989), The Monte carlo method for Semiconductor Device Simulation, Springer-Verlag, New York
- [6] M.V. Fischetti, and S.E. Laux (1988) Monte carlo Analysis of Electron transport in small semiconductor devices including band structure and space-charge effects Phys. Rev. B, 38 (19), 9721
- [7] Zhou Xing (1995) Workshop on High Performance Computing Activities in Singapore (Unpublished)
- [8] K.Tomizawa, Numerical Simulation of Submicron Devices, Artech House, Boston, 1993.
- [9] http://www.epm.ornl.gov/pvm/pvm_home.html.