Scaling Behavior of the Tight Binding Molecular Dynamics Code with Parallel Matrix Diagonalization (ScaLAPACK): Application to Carbon Nanotube

Cem Özdoğan

Dept. of Comp. Eng., Çankaya University, Balgat, 06530, Ankara, Turkey

December 1, 2008

Abstract

One of the main objectives of this project is to reveal the scaling behavior of the our developed tight binding molecular dynamics (TBMD) code with parallel matrix diagonalization. Obtained Hamiltonian matrix is distributed among the processors for diagonalization. The effect of the block size of the block cyclic distribution of the matrix elements and the shape of the processor grid on diagonalization time are discussed. The comparison of linear scaling TBMD, traditional ($O(N^3)$) TBMD and parallel eigen solver results are given. It is found that the obtained scaling behavior is neither a linear nor cubic but bettween these.

1 Introduction

Parallel eigen solvers are one of the key kernel on the performance of the code which requires the eigenvalues and/or eigenvectors during the calculations. Many physical/chemical problems require a large Hamiltonian matrix that to be diagonalized. The size of this matrix depends on the the studied system size and results as the cubic proportionality factor for the required diagonalization time. There are many proposed linear scaling methods for the electronic structure calculations in the literature. Most of them bring additional approximations and simplifications together with the algorithm complexities to the underlying physical formalism. One of them is developed by our group [1] and our O(N) parallel TBMD code (with PVM (Parallel Virtual Machine) library) has been applied successfully to simulate SWCNT's (Single Wall Carbon Nano Tube) of various chirality for the structural and electronic properties [2, 3]. These linear scaling techniques can be used to simulate material systems, but the nontrivial point is that these methods mostly based on the notion of locality and this could be the source of the wrong results at different conditions of temperature, pressure (at extreme thermodynamical conditions). As a conclusion, these linear scalingalgorithmss are generally more sensitive to round off errors and accuracy together with the special care of introduced extra parameters/variables. Another alternative is to apply a parallel eigen solver for the diagonalization process without having any approximation or assumptions about the physics/chemistry of the system. The focus of this project can be outlined as the application of the SCALAPACK [5] library to a TBMD simulation of carbon nanotube.

2 Method

The tight binding approach (TB) serves as a valuable compromise that bridges the gap between ab initio simulations and model potentials, as far as the overall numerical efficiency and/or as far as the numerical accuracy are concerned. TB molecular dynamics (TBMD) is a computational tool designed to run finite-temperature MD simulations within a semi-empirical tight-binding scheme [4]. The electronic structure of the simulated system can be calculated by a TB Hamiltonian. Diagonalizaton process requires the utilization of the Linear Algebra PACKage [6]. The physically reasonable results require unpredicted numbers of the MD steps during the simulation and this means that the solution of a large symmetric linear eigen problem during each iteration is needed. The possible solution is to apply a parallel eigen solver for the diagonalization process instead of developing linear scaling method for electronic structure calculations; namely SCALAPACK which is parallel version of LAPACK. Within the ScaLAPACK project many LAPACK routines were ported to distributed memory computers using MPI. The basic routines of ScaLAPACK are the PBLAS.

3 Results

We used PDSYEV routine of ScaLAPACK library for all of the tests. Number of the layers of nanotubes given in the Figure 1left and -middle is increased to see the variation of the diagonalization time with respect to varying system

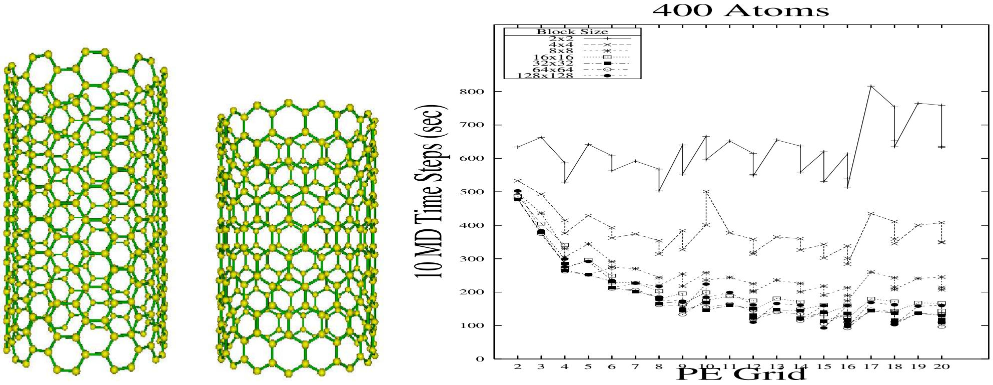

Figure 1: Carbon nanotubes; 10 × 10 chirality (left) and 17 × 0 chirality (middle) and simulation time vs processor elements grid for 20 layers (400 atoms) length 10 × 10 nanotube at different block sizes (right).

size. Several parameters affecting the diagonalization time are investigated, such as block size of the block cyclic distribution and processors grid shape. Obtained scaling behavior and parallel performance metrics such speed-up and efficiency are reported.

3.1 Block size of the block cyclic distribution and processors grid shape

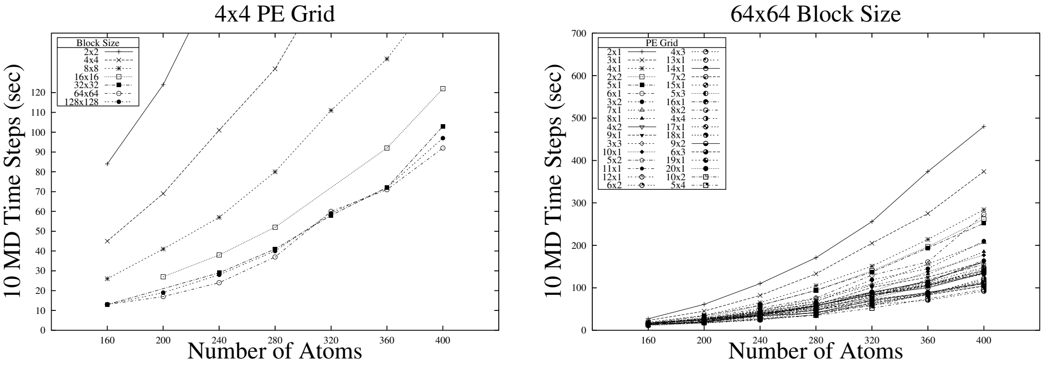

Figure 2: The effect of the block size on simulation time for varying number of atoms at processor elements grid 4 × 4 (left) and the simulation time vs number of atoms for different PE grids at 64 × 64 block size (right).

3.2 Scaling behavior

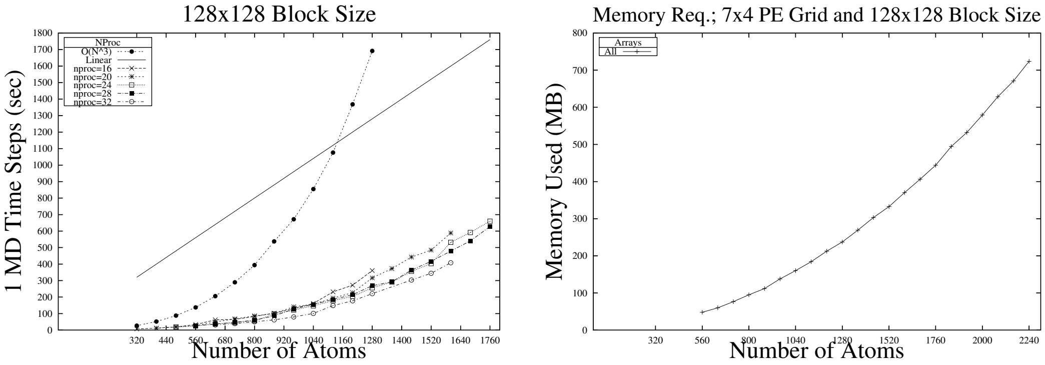

Figure 3: Comparison of methods for scaling (left). The total memory requirements of the code with PDSYEV routine (right).

3.3 Parallel performance metrics such speed-up and efficiency

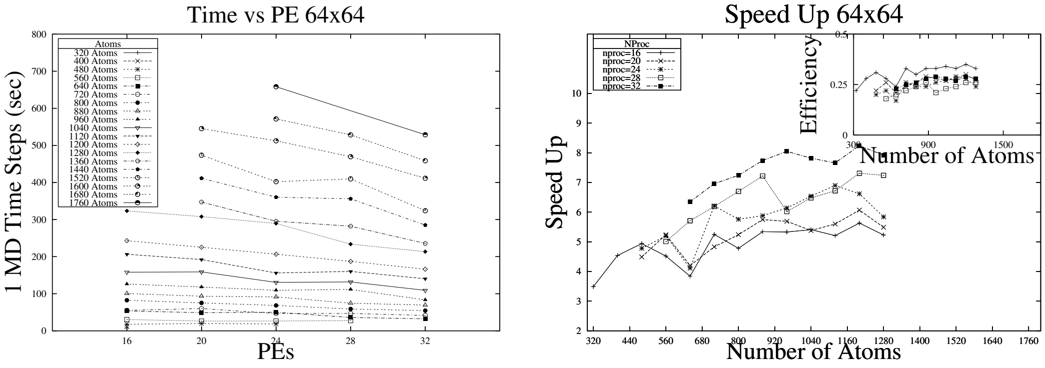

Figure 4: Simulation time vs processors for different system sizes at block size of 64 × 64 (left). Obtained speed-up and efficiency results for varying number of atoms at different number oprocessorsrs (right).

4 Conclusion

Our expectation is to obtain a linear scaling behavior without introducing any approximation and simplifications to the traditional TBMD approach. The effect of the block size of the block cyclic distribution of the matrix elements on diagonalization time is investigated for several block sizes; 2×2, 4×4, 8×8,16×16, 32×32, 64×64 and 182×128. The first finding to be reported is that the increase in the PE decreases time with the exceptions of 2×2 and 4×4 on higher number of PEs (see Figure 1right). It is found that the block sizes of 32×32, 64×64 and 128×128 are the proper choices and the most appropriate one is the size of 64×64. This behavior can be seen from the Figs. 1right and 2left. The simulation time vs number of atoms for different PE grids at 64×64 block size is given at Figure 2(right). Better results are obtained for the equal row and column numbers of the processor grid; e.g. 4×4 produces better times than 8×2 and 16×1 processor grids.

The comparison of linear scaling TBMD [2], traditional ($O(N^3)$) TBMD [4] and parallel eigen solver results for increasing number of processors are depicted in the Figure 3(left). It is seen that the obtained scaling behavior is neither a linear nor cubic but bettween these. One of the possible explanation of thabehaviorur could be the increasing memory requirements. similarar curve to simulation time is also obtained for the memory usage with increasing number of atoms (the 7×4 (28 processors) grid size at the block size of 128×128 see Figure 3right). The highest possible system size with the available hardware is found as 2240 atoms (112 layers) with 725 MB memory usage. The matrix size at that number of atoms is 8961×8961 (with $4 * N + 1 \times 4 * N + 1$). Although increasing number of PEs leads to a decrease in the simulation time (see Figure 4left), the obtained speed-up and efficiency results are not as good as expected (see Figure 4right).

This results can be concluded as the scaling behavior of the PDSYEV routine of SCALAPACK library is not linear but promising for real physical simulation. The PDSYEVD and PDSYEVX subroutines should also be studied to compare the performance metrics for futher improvement of the study.

5 Acknowledgment

This work was carried out under the HPC-EUROPA project (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action under the FP6 “Structuring the European Research Area” Programme.

References

[1] G. Dereli, C. Özdoğan, Phys. Rev. B. 67, 035415, 2003.

[2] C. Özdoğan, G. Dereli, T. an, Comput. Phy. Comm. 148, 2002.

[3] G. Dereli, C. Özdoğan, Phys. Rev. B. 67, 035416, 2003.

[4] L. Colombo, *Compt. Mater. Sci.* **12**, 278 (1998).

[5] http://www.netlib.org/scalapack/

[6] http://www.netlib.org/lapack/