6-9-2008

A Nano-electronics Simulator for Petascale Computing: From NEMO to OMEN

Hansang Bae
Purdue University - Main Campus

Steve Clark
Purdue University - Main Campus

Ben Haley
Purdue University - Main Campus

Ryu Hoon
Purdue University - Main Campus

Gerhard Klimeck
Purdue University - Main Campus, gekco@purdue.edu

See next page for additional authors

Authors
Hansang Bae, Steve Clark, Ben Haley, Ryu Hoon, Gerhard Klimeck, Sunhee Lee, Mathieu Luisier, and Faisal Saied

A Nano-electronics Simulator for Petascale Computing: From NEMO to OMEN

Hansang Bae[1], Steve Clark[2], Ben Haley[1], Ryu Hoon[1], Gerhard Klimeck[1, 3], Sunhee Lee[1], Mathieu Luisier[1], Faisal Saied[2, 4]

1 Introduction

The rapid progress in nanofabrication technologies has led to the emergence of new classes of nano-devices, in which the quantum nature of charge carriers dominates the device properties and performance. The need for atomistic-level modeling is particularly clear in studies of quantum dots. Quantum dots are solid-state structures capable of trapping charge carriers so that their wave functions become fully spatially localized, and their energy spectra consist of well-separated, discrete levels. Existing nanofabrication techniques make it possible to manufacture quantum dots in a variety of types and sizes [1]. Among them, semiconductor quantum dots grown by self-assembly (SADs), trapping electrons as well as holes, are of particular importance in quantum optics, since they can be used as detectors of infrared radiation [2], optical memories [3], single photon sources [4]. Arrays of quantum-mechanically coupled SADs can also be used as optically active regions in high-efficiency, room-temperature lasers [5].

The main goal of this paper is to present new capabilities that have been added to codes. We are transitioning to having in OMEN a single code that will include the functionality of the well-established NEMO 3-D code for strain and electronic structure computations, and a new capability to solve the challenging 3D quantum transport problem, and be designed to run efficiently on large systems like Ranger, the first NSF Track 2 system at TACC. We believe that OMEN will be one of the premier simulation tools for the design and analysis of realistically-sized nanoelectronic devices, and therefore to make it a valid tool for the Network for Computational Nanotechnology (NCN) community. These recent advances include algorithmic refinements, performance analysis to identify the best computational strategies, porting to state of the art HPC architectures, including the Ranger system, the BlueGene, the Cray XT3 and a Woodcrest Linux cluster. One important consequence of these recent enhancements is the ability to run 3D quantum transport computations on Ranger. 3D quantum transport for realistic devices is a very challenging computational problem, and represents a new capability for which resources of the scale of Ranger are essential. We present initial results for the transport problem on Ranger. We also present results for the electronic structure computations based on both the NEMO 3-D code, and the new implementation in OMEN. From an algorithmic point of view, a key challenge is the extraction of interior, degenerate eigenvectors at this scale. These calculations have been carried out on up to 8K cores on the BlueGene and the Cray XT3, and have been tested on several Teragrid systems. We also present initial scaling results for the electronic structure computations on Ranger.

The rest of the paper is structured as follows. In section 2 we review the physical model underlying OMEN and NEMO 3-D; in section 3, we describe the approach used for the parallelization of the computations. In section 4, we briefly describe the algorithms implemented in the packages. Section 5 has the performance results, including benchmark results for 3D transport up to 16K cores, and up to 8K cores in the electronic structure phase.

2 Modeling nano-electronic devices

Figure 1 shows the evolution from the NEMO 1-D and NEMO 3-D codes to OMEN. NEMO 1-D is a mature code that computes quantum transport for a 1D model. It can compute current-voltage curves for nano-devices. It has a natural multilevel parallelism (in voltage, momentum and energy). NEMO 3-D can compute strain and electronic structure (but not transport) for 3D devices. The new OMEN code includes the functionality of NEMO 3-D with several enhancements that will make it better tuned for petascale systems. OMEN also has the 3D analog of NEMO 1-D. This is a significant new capability that did not exist before, and requires petascale computing resources. Initial performance results for this capability are presented in section 5.

[1] School of Electrical and Computer Engineering, Purdue University
[2] Rosen Center for Advanced Computing, Purdue University
[3] Network for Computational Nanotechnology, Purdue University
[4] Computing Research Institute, Purdue University

OMEN is a three-dimensional, atomistic, and full-band quantum transport simulator designed for multi-gate ultra-thin-body (UTB) and nanowire (NW) field-effect transistors (FETs). The $sp^3d^5s^*$ semi-empirical tight-binding method is chosen as band structure model. The Schrödinger equation is solved with open boundary conditions either in the Wave Function or in the Non-Equilibrium Green's Function formalism. The resulting charge and current densities are then self-consistently coupled to the Poisson equation expressed in a finite-element basis. Hence, ballistic transport, source-to-drain tunneling, alloy disorder, strain, and interface roughness can be treated on an atomistic level in n- or p-doped FETs with a cross section up to about 22 nm^2 for circular or square nanowire and a width of 8 nm for UTB. Transistor structures made of several semiconductor materials, such as Si, Ge, SiGe, GaAs, or InAs and with any channel orientation (<100>, <110>, <111>, or <112>) can be investigated. As an example a double-gate UTB FET with transport along the <100> axis is considered in Figure 2, [15], [16], [17].

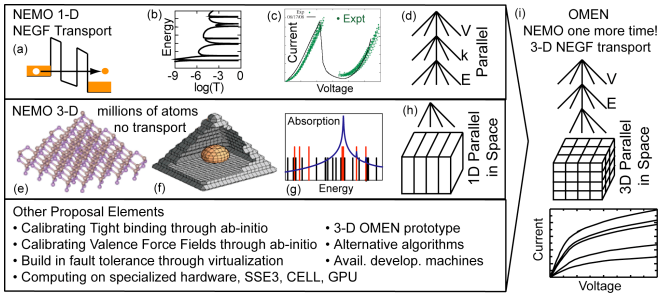

Figure 1. The NEMO 1-D code can do 1D NEGF quantum transport, while the NEMO 3-D is designed to do strain and electronic calculations. Our new code OMEN incorporates NEMO 3-D functionality and adds a 3D quantum transport capability, and includes features that make the code petascale capable.

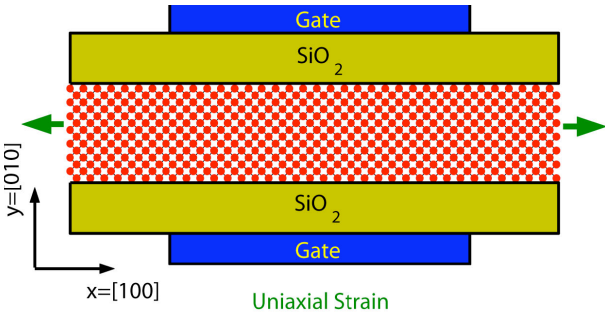

Figure 2. Schematic view of a double-gate ultra-thin body field effect transistor with transport along the <100> crystal axis and confinement along <010>.

We summarize some of the key features of the model and the NEMO 3-D package, which consists of three main parts

- The geometry constructor (setup phase)
- The strain computation
- The electronic structure computation

Geometry constructor and setup. The geometry constructor's purpose is to represent the nanostructure in atomistic detail in the memory of the computer. Each atom is its coordinates; its type (atomic number), information whether the atom is on the surface or in the interior of the sample (important later on in electronic calculations), what kind of computation it will take part of (strain only or strain and electronic), and who its nearest neighbors are. This part of NEMO 3-D occupies only a small fraction of total computational time and memory, and will not be discussed further.

The strain computation. The materials making up the quantum dot nanostructure differ in their lattice constants. This lattice mismatch leads to the appearance of strain:

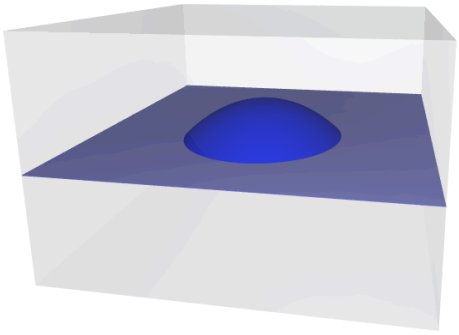

Figure 3. Schematic view of the QD nanostructure, for electronic structure calculations.

atoms throughout the sample are displaced from their bulk positions. Knowledge of equilibrium atomic positions is crucial for the subsequent calculation of the quantum dot's electronic properties, which makes the computation of strain a necessary step in realistic simulations of these nanostructures. NEMO 3-D computes strain field using an atomistic valence force field method (VFF) [13]. In this approach, the total elastic energy of the sample is computed as a sum of bond-stretching and bond-bending contributions from each atom. The equilibrium atomic positions are found by minimizing the total elastic energy of the system.

The electronic structure computation. The single-particle energies and wave functions are calculated using an empirical first-nearest-neighbor tight-binding model. The underlying idea of this approach is the selection of a basis consisting of atomic orbitals (such as s, p, d, and s*) centered on each atom. These orbitals are treated as a basis set for the Hamiltonian operator and lead to a discrete Hamiltonian matrix that has a block tri-diagonal structure due to the restriction to nearest neighbor interactions.The parametrization of all bonds using analytical forms of directional cosines for various tight-binding models is given in Ref. [14]. NEMO 3-D uses the parameterization for the sp3d5s* model, in which each atom is assigned 20 spin-degenerate orbitals.

The 20-band nearest-neighbor tight-binding model is parameterized by 34 energy constants, which need to be established by fitting the computed electronic properties of materials to those measured experimentally. This is done by considering bulk semiconductor crystals (such as GaAs or InAs) under strain. Search for optimal parameterization is done using a genetic algorithm, described in detail in Refs. [8-12]. Once it is known for each material constituting the quantum dot, a full atomistic calculation of the single-particle energy spectrum is carried out on samples composed of millions of atoms. Calculating the electronic structure is the most computationally intensive phase of the code. The eigenvalue problem is solved in parallel using one of the algorithms discussed in Section 4.

3 PARALLELIZATION

The 3D transport code in OMEN has up to four levels of parallelization. These include parallelization over voltage, momentum and energy, and additional parallelization for a linear solver for a 3D equation, with spatial decomposition. The voltage/bias point level is embarrassingly parallel, while the double integral over momentum and energy that is evaluated requires adaptive quadrature and involves some communication. The linear solver is currently a sparse direct solver that is parallelized within a node.

The electronic structure code is parallelized using MPI. The most significant component in the parallelization are the matrix vector products in the strain and the electronic structure phases that are needed for the energy minimization and the eigenvalue computation, respectively. A big difference between these two phases is that in the strain phase there are three real unknowns per atom, whereas there are typically 20 complex unknowns per atom in the electronic structure phase. This difference results in a very different proportion of communication in the matrix vector products – the strain phase has more communication relative to computation and does not scale as well to large numbers of cores as the electronic structure matrix vector products, where communication a smaller fraction of the time. Figure 4 shows the data distribution used and the sparsity structure of the Hamiltonian. Only nearest neighbor communication is needed in the matrix vector product.

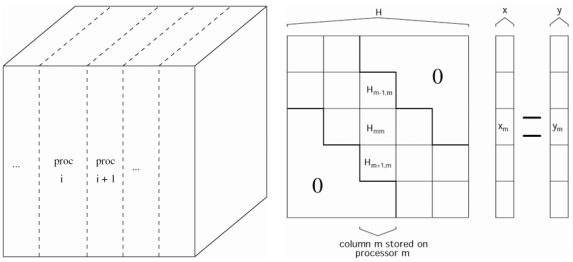

Figure 4. Data distribution scheme used in NEMO 3-D to parallelize the matrix-vector product (a), and block structure of the discrete matrix (b).

4 ALGORITHMS

To find the energy levels and wave functions of the electronic structure, e.g. quantum dot, we need to solve a large eigenvalue problem arising from the discretization of the time-independent Schrödinger equation

$$H\vec{\psi}_i = E_i\vec{\psi}_i$$

where H is the n by n Hamiltonian matrix, $\vec{\psi}_i$ is the eigenvector representing the wave function associated with the eigenvalue E_i representing the i-th energy level. We have implemented Lanczos [21] in NEMO 3-D and in OMEN. For a standard symmetric eigenvalue problem, the Lanczos algorithm builds an orthonormal basis V_k for a Krylov subspace associated with the Hamiltonian, and a k by k tridiagonal matrix T_k such that the following relationship (the Lanczos factorization) is satisfied

$$HV_k = V_kT_k + t_{k+1,k}\vec{v}_k\vec{e}_k^T$$

It has been shown [23] that the eigenvalues of T_k tend to eigenvalues of H as *k* tends to *n*. Moreover, some eigenvalues of T_k converge to eigenvalues of H very quickly. The problem of finding eigenvalues E_i of H is then reduced to the much easier problem of finding eigenvalues of a small tridiagonal matrix T_k and the eigenvectors $\vec{\psi}_i$ of H are found to be

$$[\vec{\psi}_1 \quad \ldots \quad \vec{\psi}_k] = V_kW_k$$

where W_k are the eigenvectors of T_k. The advantage of Lanczos algorithm is that it is fast. However, it cannot compute repeated eigenvalues. This is addressed by our implementation of block Lanczos, which we do not discuss here. We have also implemented Trace Minimization [18, 19] and Rayleigh-Ritz eigenvalue solvers, and interfaced the PARPACK [22] to NEMO 3-D. These eigenvalue solvers are not discussed in this paper.

5 PERFORMANCE

In this section, we present very recent results for the 3D transport problem on the Ranger system at TACC, using the new OMEN code. We present some new results for the electronic structure computation in OMEN on Ranger. We also present scaling results using NEMO 3-D, on the Cray XT4 at Oak Ridge, the BlueGene/L at RPI, the Abe cluster at NCSA, the IBM JS21 jointly operated by Indiana University and Purdue University, Bigben at PSC, and a Linux cluster at Purdue.

5.1 3D Quantum Transport

In Figure 5, we show scaling results for 3D quantum transport on Ranger, using OMEN for the device shown in Figure 2. This is a double-gate field effect transistor (two-dimensional device)designed for the 22 nm technology node, i.e. according to the International Road Map for Semiconductor (ITRS). This is a large problem with 8 bias points, two Poisson iterations per bias point, 16 points for the momentum integral and about 1400 points for the energy integral. For each of these combinations, we solve the Schrödinger equation using the MUMPS code [24]. This corresponds to over 300,000 linear systems each of order about 50,000. The MUMPS code is a parallel sparse direct solver. So far we have used it with one and two processors. These initial results show that the OMEN code can exploit the power of the Ranger system to solve the transport problem for nano-devices. Given the excellent scaling in Figure 5, we feel the transport computation in the OMEN code will scale well to an even greater number of cores.

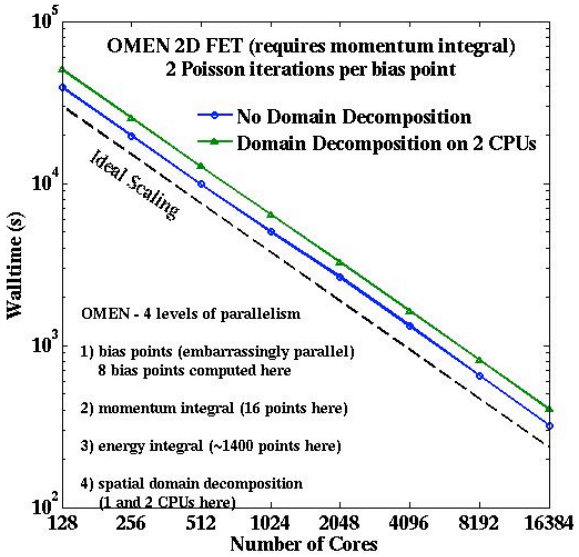

Figure 5. 3D quantum transport scaling on Ranger using OMEN for the device shown in Figure 2.

5.2 Electronic Structure

Figure 6 shows very recent results on the Ranger system with the new code for electronic structure computations in OMEN. This functionality has been available in NEMO 3-D for some time, but the OMEN code has additional features that will make it better suited for petascale computing, including a 3D domain decomposition option, and a reduced memory footprint. This will allow us to go to larger numbers of atoms, and utilize more processors. These results are a strong indication that OMEN has the potential for very high scalability, for both transport and electronic structure computations.

Figures 7, 8 and 9 shows performance data for benchmark runs with NEMO 3-D on two leading architectures, the Cray XT4 at Oak Ridge National Labs and the IBM BlueGene/L at RPI. The benchmark problems performed 500 iterations of the Lanczos eigenvalue solver, for a range of problem sizes and number of cores. The timing data can be viewed in two ways. The solid lines correspond to a fixed number of atoms per core, which is the weak scaling view. The dashed lines correspond to a fixed total problem size, the strong scaling view. Ideal scaling is a horizontal line in the weak scaling case, and the sloping green line in the strong scaling case.

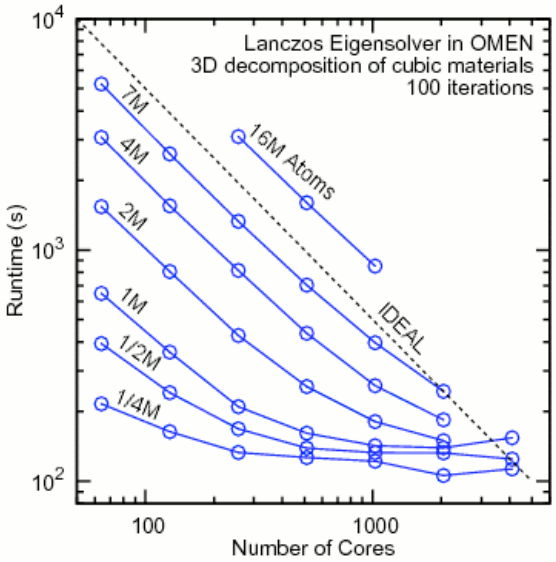

Figure 6. Scaling behavior on Ranger for the electronic structure computations in OMEN, based on a 3D spatial domain decomposition.

The code has two different options for handling the Hamiltonian matrix, both of which were used in the experiments

shown in Figures 7, 8 and 9. The "stored" option corresponds to the case where the matrix is computed once, and stored. The "recompute" option corresponds to not storing the matrix, but re-computing it, each time it is needed (once for each Lanczos iteration). The stored option uses more memory but does fewer floating point operations. The recomputed option is the opposite: the memory footprint is reduced, but the number of operations increases. The recomputed option tends to show better scalability, because communication costs are a lower fraction of the total time. The stored option leads to lower running times. The recompute option will become more interesting if the new generation of petascale architectures has smaller memories per core. Note that for the BlueGene/L, the stored option was not feasible, because of the small amount of memory per processor.

One way to characterize scalability for an atomistic model like NEMO 3-D is how many atoms are needed per core to get close to ideal weak scaling (fewer is better). Algorithmically, the recompute option shows good weak scaling for a smaller number of atoms per core (64 atoms for the BG/L and 1024 to 2048 atoms for the XT4 compared to 4096 atoms for the stored option on the XT4)), while architecturally, the BlueGene/L shows better scalability, because of its different system balance (slower processor with a fast interconnect).

The data in these three figures is up to 8192 cores. This represents a very clear ability to scale to a large number of processors. Based on the scaling curves, we see that the code in its current form will scale to an even larger number of processors, for larger problems, as access to these large resources becomes more common.

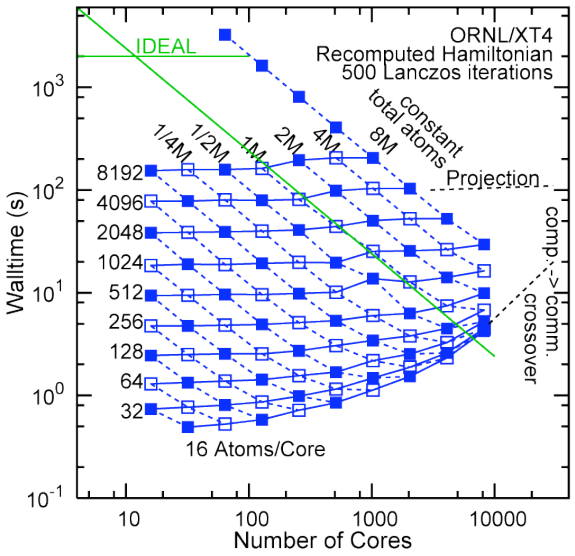

Figure 7. Wall clock time vs. number of cores for 500 Lanczos iterations in the electronic structure computation. Strong and weak scaling of NEMO 3-D on Cray XT4, with the stored-matrix option.

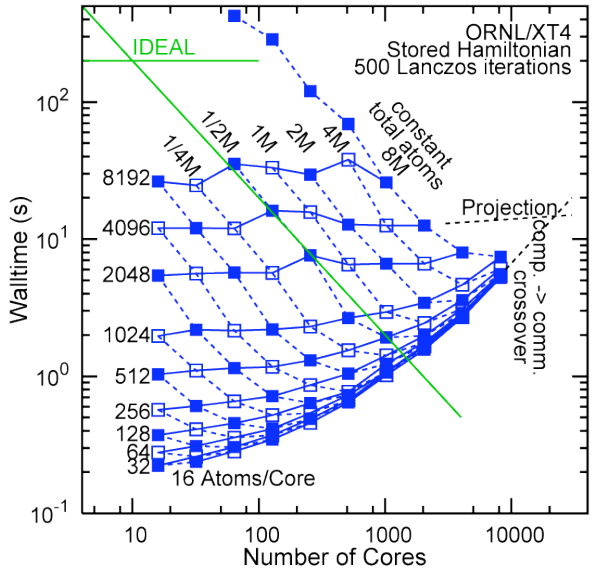

Figure 8. Wall clock time vs. number of cores for 500 Lanczos iterations in the electronic structure computation. Strong and weak scaling of NEMO 3-D on Cray XT4, with the recomputed-matrix option.

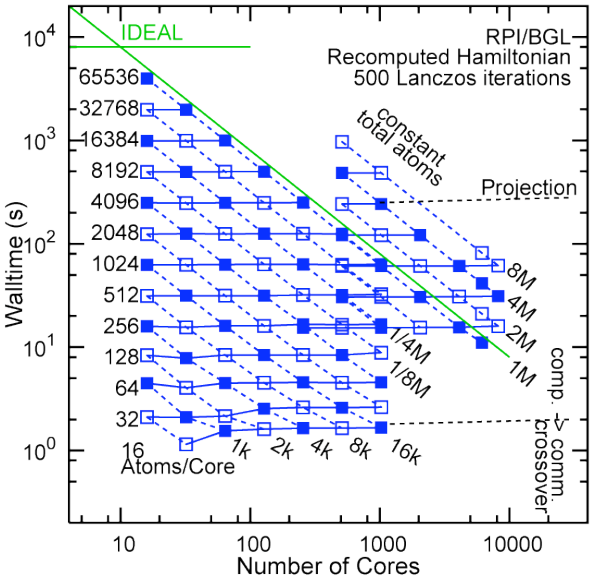

Figure 9. Wall clock time vs. number of cores for 500 Lanczos iterations in the electronic structure computation. Strong and weak scaling of NEMO 3-D on RPI/BGL, with the recomputed-matrix option.

Figures 10 and 11 show scaling data for NEMO 3-D, for a benchmark problem with 8 million atoms and 500 Lanczos iterations (strong scaling). Figure 10 shows two architectures (Cray XT4 and IBM BG/L) for clarity, while Figure 11 includes four additional platforms in the comparison. The solid lines correspond to the stored option for the Hamiltonian while the dashed lines correspond to using the recompute option. Referring to Figure 10, for 1024 cores, the recompute option on the XT4 takes about 5 times longer than the stored option on the same platform. The BG/L takes roughly 3 times longer than the XT4, when both are used with the recompute option. On the BlueGene, NEMO 3-D can only be used in recompute mode because of the small memory per node, and the XT4 has the advantage of faster processors and more memory per core, and consequently we see that the stored option on the XT4 is about 15 times faster than the recompute option on the BG/L. In Figure ref-8B, the additional four platforms were only available up to 1-24 cores. We see that NEMO 3-D scales well on all the platforms used. The NCSA Abe cluster is the fastest, up to 1024 cores, with the stored option.

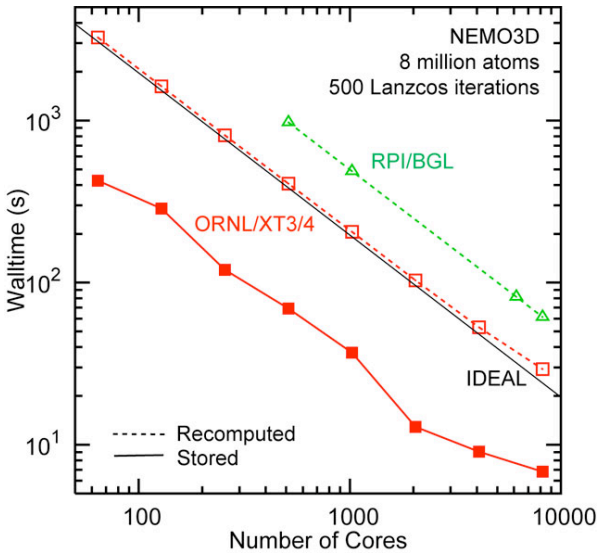

Figure 10. Wall clock time vs. number of cores for 500 Lanczos iterations with NEMO 3-D in the electronic structure computation with 8 million atoms. Results for the Cray XT4 at Oak Ridge and the BlueGene/L at RPI. Solid red line represents the stored Hamiltonian case, the dashed lines represent the recomputed Hamiltonian case.

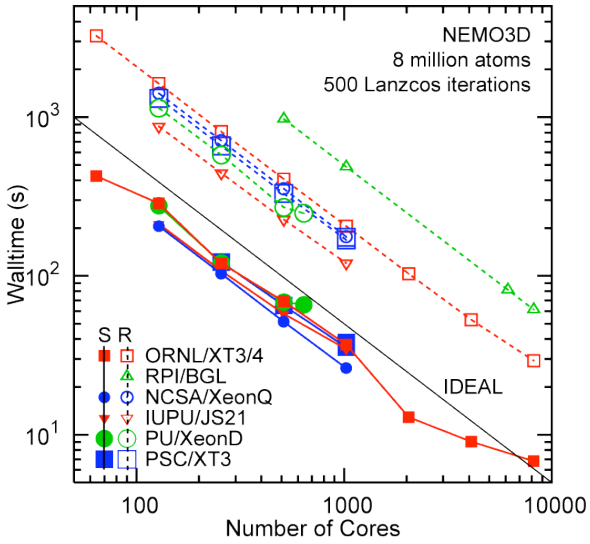

Figure 11. Same as Figure 10, but with more platforms included. NCSA/XeonQ is the Abe cluster at NCSA, IUPU/JS21 refers to the IBM JS21 jointly operated by Indiana University and Purdue University, PU/XeonD is a Linux cluster at Purdue and PSC/XT3 is the Bigben system at PSC.

In addition to the performance for the NEMO 3-D benchmark cases, with 100 iterations in the strain and electronic structure cases, we carried out end-to-end runs on the PU/Woodcrest cluster. This involves iterating to convergence and computing the eigenstates in the desired range (conduction band and valence band). For each problem size, measured in millions of atoms, we ran the end-to-end case to completion, for one choice of number of cores. The iteration counts for the Lanczos computation are given in Table 1. We then combined this with the measured average time per iteration for a range of values of number of cores, to compute the predicted performance displayed in Figure 12.

0.89	1.99	3.92	6.80	16.18	21.07	52.57
5,061	5,121	6,141	7,921	9,621	10,401	14,691

Table 1. The iteration counts for the Lanczos computation for a range of problem sizes. The first row gives the number of atoms in millions, and the second row gives the number of Lanczos iterations.

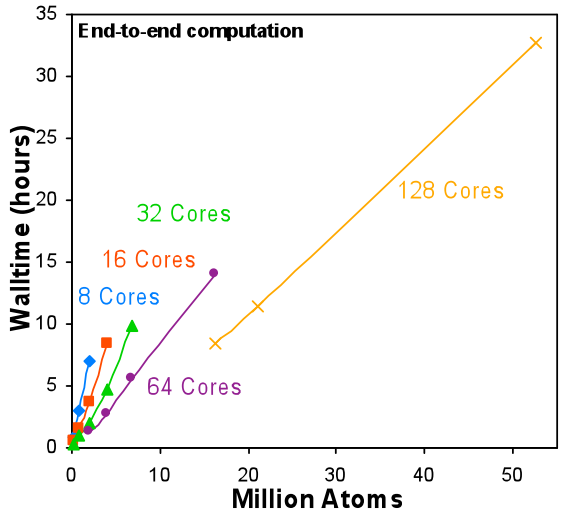

Figure 12. Wall clock time vs. number of atoms for end-to-end computations of the electronic structure of a quantum dot, for various numbers of cores on the PU/Woodcrest cluster.

6 Conclusions

We have presented performance results for large scale simulations of nano-electronic devices, based on on-going work to enhance both the capability and scalability of our codes. Our initial results for the transport problem show that resources on the scale of the Ranger system can make this very challenging computation tractable. In particular, we have shown excellent scalability on Ranger up to 16K cores for quantum transport on nano-devices. Our results for the electronic structure computations show that the Lanczos solver can scale up to 8K cores. Taken together, our results show that problems in computational nano-electronics that had seemed beyond reach are now becoming feasible.

Acknowledgment

This work was supported in part by NSF grant EEC-0228390 that funds the Network for Computational Nanotechnology and in part by NSF PetaApps grant number 0749140. We are grateful to the Texas Advanced Computing Center for providing us access to the Ranger system, to Oak Ridge National Labs for providing us with access to their Cray XT3/XT4, and to Rensselaer Polytechnic Institute's (RPI) Computational Center for Nanotechnology Innovations (CCNI) for providing us with access to their BlueGene/L. Some of the studies reported here were done on the Abe cluster at the National Center for Supercomputing Applications, on the Cray XT3 at the Pittsburg Supercomputing Center, and on the IBM JS21 at Indiana University. We also acknowledge the support of Purdue's Rosen Center for Advanced Computing.

References

[1] For reviews and references see, e.g., Jacak, L., Hawrylak, P., and Wojs, A, "Quantum dots", Springer-Verlag, Berlin, 1998.
[2] Aslan, B., Liu, H.C., Korkusinski, M., Cheng, S.-J., and Hawrylak, P., Appl. Phys. Lett., 82, 630, 2003.
[3] Petroff, P.M., in "Single Quantum Dots: Fundamentals, Applications, and New Concepts", Peter Michler, Ed., Springer, Berlin, 2003.
[4] Michler, P., et al., Science, 290, 2282, 2000; Moreau, E., et al., Phys. Rev. Lett., 87, 183601, 2001.
[5] Arakawa, Y., and Sasaki, H., Appl. Phys. Lett., 40, 939, 1982; Fafard, S., et al., Science, 22, 1350, 1996; Maximov, M.V., et al., J. Appl. Phys., 83, 5561, 1998.
[6] Petroff, P.M. and DenBaars, S.P., Superlatt. Microstruct. 15, 15, 1994.
[7] For a review and references see, e.g., Tadic, M., et al., J. Appl. Phys. 92, 5819, 2002.
[8] Klimeck, G., Oyafuso, F., Boykin, T.B., Bowen, R.C., and von Allmen, P., "Development of a Nanoelectronic 3-D (NEMO 3-D) Simulator for Multimillion Atom Simulations and Its Application to Alloyed Quantum Dots", Computer Modeling in Engineering and Science, 3, 601, 2002.
[9] G. Klimeck, F. Oyafuso, R. C. Bowen, T. B. Boykin, T. A. Cwik, E. Huang, E. S. Vinyard .3-D atomistic nanoelectronic modeling on high performance clusters: multimillion atom simulations., Superlattices and Microstructures, Vol. 31, Nos 2–4, 2002.
[10] G. Klimeck, F. Oyafuso, T. B. Boykin, R. C. Bowen, P. von Allmen. Development of a Nanoelectronic 3-D (NEMO3-D) Simulator for Multimillion Atom Simulations and Its Application to Alloyed Quantum Dots., CMES, vol.3, no.5, pp.601-642, 2002.

[11] F. Oyafuso, G. Klimeck, P. von Allmen, T. Boykin, and R. C. Bowen, "Strain Effects in large-scale atomistic quantum dot simulations", *Phys. Stat. Sol*. (b), Vol. 239, p 71-79 (2003).
[12] F. Oyafuso, G. Klimeck, R. C. Bowen, T. B. Boykin, and P. von Allmen. "Disorder Induced Broadening in Multimillion Atom Alloyed Quantum Dot Systems", *Phys. Stat. Sol*. (c), vol 0004, pg 1149-1152 (2003).
[13] L.-W. Wang and A. Zunger. Solving Schrödinger's equation around a desired energy: Application to silicon quantum dots. *J. Chem. Phys.*, 100(3):2394–2397, 1994
[14] A. R. Tackett and M. Di Ventra. Targeting Specific Eigenvectors and Eigenvalues of a Given Hamiltonian Using Arbitrary Selection Criteria. *Physical Review B*, 66:245104, 2002.
[15] M. Luisier, A. Schenk, and W. Fichtner, "Three-Dimensional Full-Band Simulations of Si Nanowire Transistors", IEDM Tech. Digest 2006, 811 (2006).
[16] M. Luisier, A. Schenk, and W. Fichtner, "Atomistic Treatment of interface roughness in Si nanowire transistors with different channel orientations", Appl. Phys. Lett. 90, 102103 (2007).
[17] M. Luisier, G. Klimeck, A. Schenk, and W. Fichtner, "Atomistic Simulation of Nanowires in the sp3d5s* Tight-Binding Formalism: from Boundary Conditions to Strain Calculations", Phys. Rev. B 74, 205323 (2006).
[18] A. H. Sameh and J. Wisniewski. A Trace Minimization Algorithm for the Generalized Eigenvalue Problem . *SIAM Journal on Numerical Analysis* , Vol. 19, No. 6, pp. 1243-1259, 1982.
[19] A. Sameh and Z. Tong, The trace minimization method for the symmetric generalized eigenvalue problem. *J. Comput. Appl. Math.* 123, 155-175, 2000.
[20] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK USERS GUIDE: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, PA, 1998.
[21] C. Lanczos, "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators", Journal of Research of the National Bureau of Standards, Vol. 45, No 4, 1950
[22] K. Maschhoff and D. Sorensen, "A portable implementation of ARPACK for distributed memory parallel architectures", Copper Mountain Conference on Iterative Methods, 1996
[23] R.R. Underwood, "An iterative block Lanczos method for the solution of large sparse symmetric eigenproblems", Ph.D. Dissertation, Stanford University, Stanford, CA, 1975
[24] MUMPS http://graal.ens-lyon.fr/MUMPS/