A multi-GPU based approach for atomistic calculations of quantum energy eigenstates

Walter Rodrigues, A. Pecchia*, M. Auf der Maur, A. Di Carlo
Department of Electronics Engineering, University of Rome Tor Vergata, Rome, Italy
*CNR-ISMN, Via Salaria Km 29,600, 00017 Monterotondo, Rome, Italy
e-mail: walter.rodrigues@uniroma2.it

INTRODUCTION

In this work we show how atomistic calculations of energy eigenstates can be accelerated by exploiting modern Graphical Processor Unit (GPU) architectures. We apply our GPU approach to a tight binding (TB) model based on a $sp^3d^5s^*$ parameterization [1] to compute eigenpairs of large GaN/AlGaN quantum dots (Qdot), as shown in Fig 1. Such systems have important applications in modern nitride-based light emitting diodes (LEDs) [2].

EIGENSOLVER OPTIMIZED FOR GPU

General purpose GPUs have become very attractive to computational scientists in recent years, thanks to their massive multi-core parallelization. To our knowledge however, there exist only a limited amount of work done to develop eigensolvers optimized for GPUs.

We employ Lanczos algorithm with simple restart which is an effective method for solving Hermitian eigenvalue problems by first building an orthonormal basis and then forming approximate solutions using Rayleigh projection [3-4]. We tune the algorithm so that it occupies little memory at the expenses of more computation. This is suitable for GPU having limited memory, but large computational power.

The CPU-GPU data transfer bottleneck is overcome by implementing most of the algorithm on the GPU, leaving only the control logic to the CPU. In order to calculate more realistic nanostructures comprising of ≈1 million atoms we employ a distributed multi-GPU strategy.

RESULTS

In order to put the CUDA performance in the right perspective we compare the performance of the GPU cluster versus a distributed parallel cluster using a 20Gbit/s InfiniBand network.

The tests reveal that a single GPU attains a performance gain of a factor of 10 compared to the multi-processing implementation on a single quad core CPU. The average performance of the multi-GPU system increases by a factor of 1.5 when the number of GPUs is doubled. Benchmark calculations on a 600,000 atom Qdot shows that the multi-GPU system with 2 Tesla K20 GPUs is about 3x faster than 8 Quad-core CPUs.

A further speed up can achieved by applying a similar strategy in developing a Jacobi-Davidson based eigensolver which converges much faster with a suitable preconditioner [5].

CONCLUSIONS

A multi-GPU system can be considered as one of the best, cost effective architecture currently available for the calculation of quantum energy eigenstates for nanodevice applications.

REFERENCES

[1] J. M. Jancu, F. Bassani1, F. Della Sala, and R. Scholz., *Transferable tight-binding parametrization for the group-III nitrides*, Appl. Phys. Lett. 81, 4838 (2002).

[2] G. Penazzi, A. Pecchia, F. Sacconi and A. Di Carlo. *Calculation of optical properties of a quantum dot embedded in a GaN/AlGaN nanocolumn*. Superlattices and Microstructures v 47, Issue 1, p. 123-128, (2010).

[3] Kesheng Wu, Horst Simon. *Thick-Restart Lanczos Method for Large Symmetric Eigenvalue Problems*. SIAM Journal on Matrix Analysis and Applications, v.22 n.2, p. 602-616, (2000).

[4] D. Calvetti, L. Reichel, and D.C. Sorensen, *An Implicitly Restarted Lanczos Method for Large Symmetric Eigenvalue Problems*. *Electronic Transactions on Numerical Analysis* 2: p. 1–21 (1994).

[5] Y. M. Niquet, A. Lherbier, N. H. Quang, M. V. Fernández-Serra, X. Blase, and C. Delerue, *Electronic structure of semiconductor nanowire*, Physical Review B 73, 165319 (2006).

ISBN 978-2-9547858-0-6

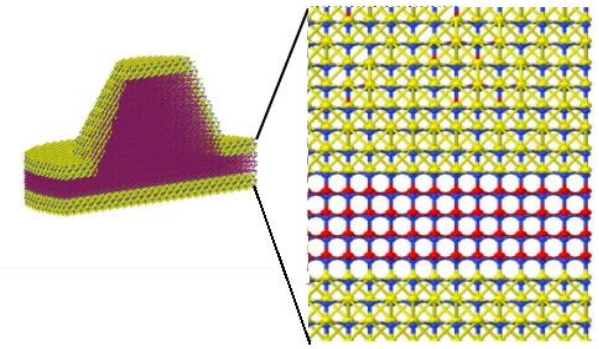

Fig. 1. Conical wurtzite GaN/AlGaN quantum dot (Qdot) with 30%Al. Atomistic description: In yellow Aluminium, in red Gallium.

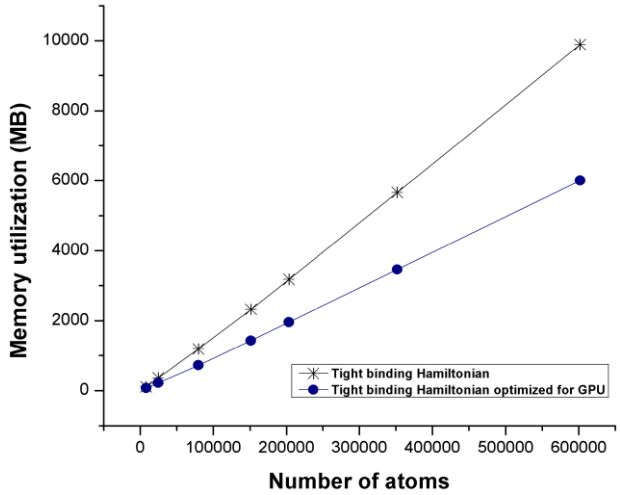

Fig. 2. Memory utilization by TB Hamiltonian when no optimization is performed vs optimization by using the splitting technique that leads to 35-40% memory saving.

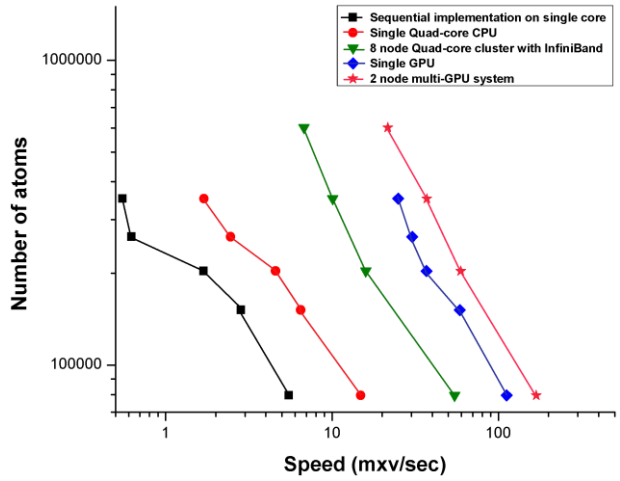

Fig. 3. Speed comparison for matrix-vector multiplication on a single Intel Xeon W3530 core vs Quad-core Intel Xeon W3530 CPU using OpenMP vs MPI/OpenMP implementation on 8 Quad-core Intel Xeon X5560 CPU connected through 20 Gbps InfiniBand vs CUDA implementation on a single Tesla K20 GPU vs MPI/CUDA implementation on a multi-GPU system having two Tesla K20 GPUs.

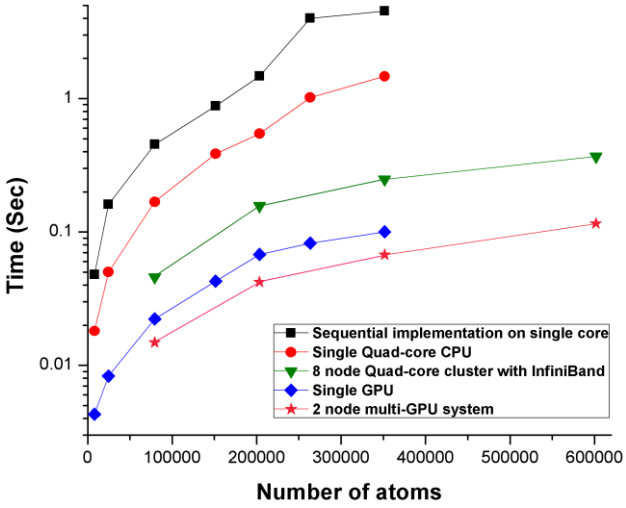

Fig. 4. Time per Lanczos iteration on a single Intel Xeon W3530 core vs Quad-core Intel Xeon W3530 CPU using OpenMP vs MPI/OpenMP implementation on 8 Quad-core Intel Xeon X5560 CPU connected through 20 Gbps InfiniBand vs CUDA implementation on a single Tesla K20 GPU vs MPI/CUDA implementation on a multi-GPU system having two Tesla K20 GPUs.

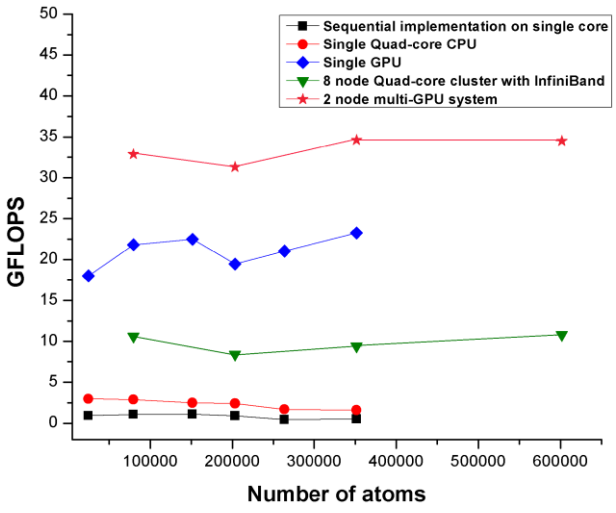

Fig. 5. Performance comparison on a single Intel Xeon W3530 core vs Quad-core Intel Xeon W3530 CPU using OpenMP vs MPI/OpenMP implementation on 8 Quad-core Intel Xeon X5560 CPU connected through 20 Gbps InfiniBand vs CUDA implementation on a single Tesla K20 GPU vs MPI/CUDA implementation on a multi-GPU system having two Tesla K20 GPUs.