First Experiences with *ab initio* Molecular Dynamics on OpenPOWER: The Case of CPMD

Valéry Weber*, A. Cristiano I. Malossi*, Ivano Tavernelli*, Teodoro Laino*, Costas Bekas*,
Manish Modani†, Nina Wilner‡, Tom Heller‡, Alessandro Curioni*

*IBM Research - Zurich, Switzerland
Email: {vwe,acm,ita,teo,bek,cur}@zurich.ibm.com

†IBM India, Bengaluru, India
Email: mamodani@in.ibm.com

‡IBM USA
Email: {new_nina,tjheller}@us.ibm.com

Keywords—CPMD; POWER8; CUDA; NVlink; FFT; Gram–Schmidt

INTRODUCTION

Ab initio molecular dynamics (AIMD) is still one of the most commonly used approaches for calculating the time evolution of molecular and solid state systems under ambient conditions of temperature and pressure. Specifically, AIMD is particularly suited for the simulation of complex molecular systems that undergo important reorganizations of their electronic structure (bond breaking and bond forming), and for which the design of a classical force field is very cumbersome. Essential to all AIMD techniques is the calculation of the molecular potential and the corresponding forces obtained from the derivatives of the potential with respect to the nuclear coordinates. These forces are then used to solve the Newton equation of motion for calculating of the nuclear trajectories.

Within density functional theory (DFT), the molecular Hamiltonian is mapped (in principle exactly) into a system of noninteracting particles subject to a compensating local external potential (the exchange-correlation (xc) potential), for which we need approximations.

CPMD [**?**] uses a pseudopotential-based Kohn–Sham DFT description of the electronic structure in which the Kohn–Sham orbital and the electronic density are expanded in a plane-wave basis set. In addition, working with plane waves has the important advantage of simplifying the calculation of energies and forces; thus some parts of the total energy (such as the kinetic term) are efficiently computed in the Fourier (reciprocal) space, whereas other parts, like the Hartree energy and the interaction with external fields, are accurately evaluated in the real (direct) space. The limiting steps in the plane-wave implementation of AIMD codes consist in a) the forward and backward Fourier transforms (FFT) [**?**] (wavefunctions, potentials, and energy terms) and b) the orthogonalization of the wavefunctions.

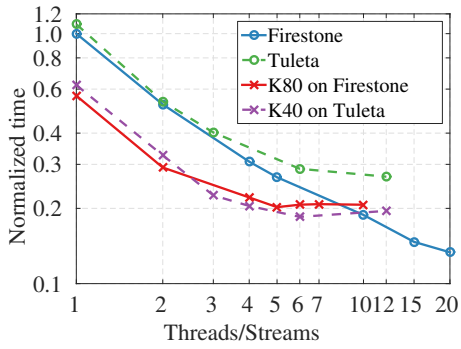
The combined use of plane waves and pseudopotential together with highly optimized algorithms for the computation of energies and forces made CPMD one of the most efficient DFT-based AIMD codes, with a documented scaling performance that extends to one million computing cores [**?**].

The advent of data-centric OpenPOWER systems based on the IBM, NVIDIA and MELLANOX collaboration offers a new potential for scalability and performance that leads to Exascale systems. Here, we present our strategy plans in migrating CPMD to the data-centric systems and summarize our progress so far. We include early results on IBM Minsky POWER system equipped with new NVIDIA Pascal G100 GPUs and NVLink.

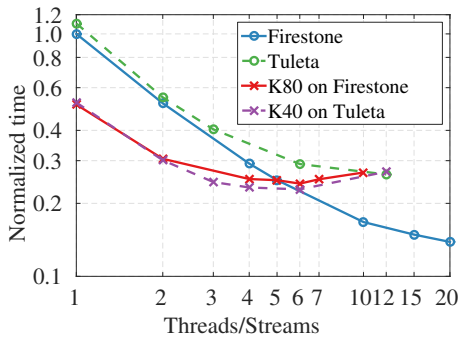
RESULTS

To illustrate the progress on porting CPMD to OpenPOWER systems, we show the strong scaling of the construction of the density, the application of the potential to the wavefunctions and the orthogonalization process for a box of 128 water molecules at normal liquid density and under periodic boundary conditions. We use the GTH pseudopotential [**?**] and plane-wave and density cutoffs of 100 Ry and 400 Ry, respectively.

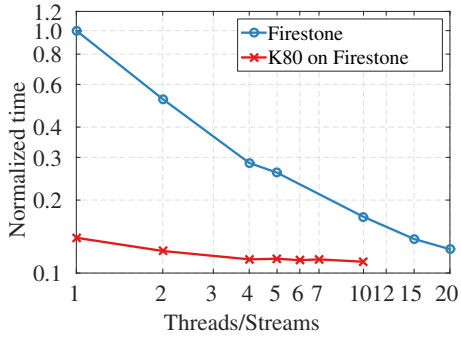
The code is compiled using the IBM XL Fortran compiler for Linux 15.1.4 with optimization flags: `-O3 -qhot -qstrict -qprefetch=aggressive:dscr=7 -qsimd=auto -qaltivec -qmaxmem=-1 -qsmp=omp`. The C-code was compiled with the GNU compiler collections 4.9.3. The runs are performed on two IBM POWER8 systems: Tuleta and Firestone. Both servers are equipped with two POWER8 processors. Each POWER8 core supports 8 hardware threads, has 64 kBytes L1 cache, 512 kBytes L2 cache, and 8 MBytes of shared L3 cache. Tuleta runs 12 cores in total at 4.2 GHz, whereas Firestone equips 20 cores at 3.42 GHz. Tuleta has one Nvidia Tesla K40, with 2880 CUDA cores; Firestone has two Nvidia Tesla K80 GPUs, each composed of two devices with 2496 CUDA cores. All computations are performed with CUDA compute capability 3.5 (on Tuleta) and 3.7 (on Firestone), both with driver version 7.5. On Firestone, our calculations use only one device of one K80, i.e., 2496 CUDA cores.



(a)



(b)



(c)

Figure 1: Log-scale performance comparison of three CPMD computational blocks run on IBM Firestone (20-cores vs. 1-device K80) and IBM Tuleta (12-cores vs. K40). Time is normalized w.r.t. one core CPU time on Firestone. All CPU runs are performed with one thread per core. All GPU runs are performed with one stream per thread per core. (a) Construction of the electronic density. (b) Applying the potential to wavefunctions. (c) Orthogonalization (results on Tuleta are not available).

The performance comparison for the three considered computational blocks is shown in Fig. 1. First, we observe that the Firestone CPU performance is better than that of the earlier Tuleta processor for the two FFT computational blocks. Concerning the GPU results, we observe a dependence on the number of streams used. The PCI-E bandwidth in the two systems is equivalent; therefore once it is saturated, the K40 tends to run slightly faster than half-K80, because of the slightly greater number of CUDA cores. The optimal number of streams varies between 4 and 6, depending on the type of computational block. Using more streams does not help, as memory bandwidth becomes the limiting factor. By analyzing the output of NVprof, we summarize, in Table I, the time percentage spent in computation and memory copies for the construction of the electronic density and applying the potential to the wavefunctions. Although all operations are performed asynchronously, the time spent in memory copies exceeds the computation time by far (about 1/3 of the total time), so that for at least 2/3 of the total time the GPU cores are idle, waiting for data.

Table I: Time percentage spent in computation and memory copy from device to host (D2H) and host to device (H2D) for constructing the electronic density and applying the potential to the wavefunctions. Computation and memory copies are performed asynchronously.

Kernel	Computation [%]	D2H [%]	H2D [%]
Electronic density	27	95	76
Applying potential	30	92	89

Future works

Our initial porting phase of CPMD to OpenPOWER architectures highlights the negative impact of a limited PCI-E bandwidth between the CPU and the GPU. To alleviate this problem, we will tackle the issue from multiple directions: at the implementation level, we will move the calculation of the electronic density and the application of the potential to the wavefunctions to the GPUs and, more generally, we will minimize data transfer whenever possible. At the architecture level, we expect a significant improvement from the NVLink high-speed interconnect equipped by next-generation Garrison POWER8' systems. NVLink will enable ultra-fast communication between the CPU and GPU, allowing data transfer at rates more than 2.5 times faster than traditional PCI-E interconnects. This should be tremendously beneficial for scenarios such as the one summarized in Table I, where multiple streams have overlapped operations, but communication time is left exposed.

References

[1] "CPMD ver. 4.1, http://www.cpmd.org," Copyright IBM Corp. 1990–2016, Copyright MPI für Festkörperforschung Stuttgart 1997–2001.

[2] S. Goedecker, "Fast radix 2, 3, 4, and 5 kernels for fast Fourier transformations on computers with overlapping multiply–add instructions," *SIAM Journal on Scientific Computing*, vol. 18, no. 6, pp. 1605–1611, 1997.

[3] V. Weber, C. Bekas, T. Laino, A. Curioni, A. Bertsch, and S. Futral, "Shedding light on lithium/air batteries using millions of threads on the BG/Q supercomputer," in *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, 2014, pp. 735–744.

[4] S. Goedecker, M. Teter, and J. Hutter, "Separable dual-space Gaussian pseudopotentials," *Physical Review B*, vol. 54, pp. 1703–1710, 1996.