Parallel implementation of machine learning-based many-body potentials on CPU and GPU

Yaoguang Zhai
University of California, San Diego
La Jolla, California
yazhai@ucsd.edu

Nathaniel Danandeh
University of California, San Diego
La Jolla, California
ndanande@ucsd.edu

Zhenye Tan
University of California San Diego
La Jolla, California
tanzhenyennn@gmail.com

Sicun Gao
University of California, San Diego
La Jolla, California
sicung@ucsd.edu

Francesco Paesani
University of California, San Diego
La Jolla, California
fpaesani@ucsd.edu

Andreas W. Götz
University of California San Diego
La Jolla, California
agoetz@sdsc.edu

ABSTRACT

Machine learning models can be used to develop highly accurate and efficient many-body potentials for molecular simulations based on the many-body expansion of the total energy. A prominent example is the MB-pol water model that employs permutationally invariant polynomials (PIPs) to represent the 2-body and 3-body short-range energy terms. We have recently shown that the PIPs can be replaced by Behler-Parinello neural networks (BP-NN). We present OpenMP parallel implementations of both PIP and BP-NN models as well as a CUDA implementation of the BP-NN model for GPUs. The OpenMP implementations achieve linear speedup with respect to the optimized single threaded code. The BP-NN GPU implementation outperforms the CPU implementation by a factor of over 4. This opens the door for routine molecular dynamics simulations with highly accurate many-body potentials on a diverse set of hardware.

KEYWORDS

Many-body potential, Permutationally invariant polynomial, Neural network, OpenMP, CUDA

ACM Reference Format:
Yaoguang Zhai, Nathaniel Danandeh, Zhenye Tan, Sicun Gao, Francesco Paesani, and Andreas W. Götz. 2018. Parallel implementation of machine learning-based many-body potentials on CPU and GPU. In *Proceedings of ACM Conference (SC'18).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

INTRODUCTION

Computer simulations are a powerful tool for molecular sciences but are still limited by a tradeoff between accuracy of the molecular models and the associated computational cost. Recently, machine learning approaches have been employed to develop many-body potentials for molecular simulations that are both highly accurate and computationally efficient. In particular, the MB-pol potential[1, 2] correctly predicts the properties of water across different phases, from energetics and infrared spectra of small water clusters to structural, thermodynamic and dynamical properties of liquid water and ice[6, 13], comparing favorably to other water potentials[12].

In MB-pol, the potential energy of water is represented as a sum of explicit 1-body (1B), 2-body (2B), 3-body (3B), and N-body energies. The short-range parts of the 2B and 3B terms account for the complex quantum mechanical many-body effects that cannot be accurately represented by simple classical expressions as used in conventional molecular simulations. MB-pol therefore employs permutationally invariant polynomials (PIPs)[5], taking water dimer (two water molecules) or trimer (three water molecules) atomic distances as the input, trained against large data sets of highly accurate electronic structure calculations. In typical condensed phase simulations, tens of thousands of 2B and 3B PIPs have to be computed for each energy and force calculation. An efficient implementation is thus critical. Unfortunately, due to the large number of parameters and terms, GPUs are not well suited for PIPs.

We have recently developed a new approach that uses Behler-Parinello neural networks (BP-NN)[3, 4] in place of PIPs[9]. This approach represents dimer/trimer coordinates in terms of atomic symmetry functions that are fed into fully-connected sigmoid neural networks to calculate the 2B and 3B short-range energies and forces. Here we present parallel implementations of 2B/3B short-range terms both with PIPs and BP-NNs using OpenMP as well as BP-NNs using CUDA for GPUs. Benchmarks on a variety of different hardware platforms demonstrates the good parallel scaling and GPU speedup of our code.

APPROACH

The energy of a system containing N (water or other) molecules can be expressed as a sum of terms corresponding to the number of involved molecules,

$$E_N = \sum_{i=1}^{N} V_i^{1\mathrm{B}} + V^{N\mathrm{B}}(1, 2, ..., N) + \sum_{i<j}^{N} V_{\text{short range}}^{2\mathrm{B}}(i, j) + \sum_{i<j}^{N} V_{\text{long range}}^{2\mathrm{B}}(i, j)$$

SC'18, Nov 2018, Dallas, Texas, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

$$+ \sum_{i<j<k}^{N} V^{\mathrm{3B}}_{\text{short range}}(i,j,k) + \sum_{i<j<k}^{N} V^{\mathrm{3B}}_{\text{long range}}(i,j,k),$$

where $V^{n\mathrm{B}}$ represents the n-body (interaction) energies and $V_{\text{short range}}$ and $V_{\text{long range}}$ denote energies at short- and long-range, respectively.

In the legacy MB-pol PIP model, the 2B short-range term is formulated as the sum of 1153 monomials each based on an exponential function of interatomic distance; the 3B short range term is the sum of 1163 exponential based monomials[1, 2].

In the BP-NN approach, the $V^{2\mathrm{B}}$ and $V^{3\mathrm{B}}$ short-range terms are formulated as the sum of atomic energies from BP-NNs[9],

$$V^{\mathrm{2B/3B}}_{\text{short range}} = \sum_{p}^{M} V^{\mathrm{2B/3B}}_{\mathrm{NN}}(G_p),$$

in which $G_{ij,p}$ and $G_{ijk,p}$ are abstracted dimer/trimer feature vectors, a.k.a. symmetry functions, for atom p in the corresponding dimer (i,j) and trimer (i,j,k) that consists of M atoms. $V^{\mathrm{2B/3B}}_{\mathrm{NN}}$ are deep neural networks for dimers and trimers, respectively.

Elements of the symmetry function vector G_p (for both 2B and 3B) encode atomic environments by radial and angular functions,

$$G^{\text{radial}}_{p} = \sum_{p\neq q}^{M} \exp\left(-\eta(R_{pq} - R_{\text{ref}})^2\right)$$

$$G^{\text{angular}}_{p} = \sum_{p\neq q\neq r}^{M} 2^{1-\zeta}(1+\lambda\cos(\theta_{pqr}))\exp\left(-\eta'(R_{pq}+R_{pr}+R_{qr})^2\right)$$

where q, r are the indices of other atoms in the same dimer or trimer, R_{pq} is the distance between atoms p and q and θ_{pqr} is the angle between the vector $\vec{pq}$ and $\vec{pr}$. R_{ref}, η, η', ζ, and λ are parameters.

IMPLEMENTATION

Both the MB-pol legacy PIP model and the BP-NN model are implemented in C++. The PIP expressions are automatically generated and factored using Maple[8] to minimize the number of required floating point operations. We have parallelized the calculation of PIPs over batches of dimers and trimers using OpenMP[11].

The BP-NN approach is computationally demanding because it involves many exponential functions. In addition, the permutation of atoms in calculating symmetry functions leads to irregular memory access patterns. Benchmarking results of an initial implementation showed the angular components of the symmetry function to be a computational bottleneck (around 50 % of total run time). In our BP-NN implementation we use OpenMP[11] and CUDA[10] to parallelize the symmetry function computations, and the Intel MKL[7] cBLAS library and cuBLAS/cuDNN[10] libraries to speed up the neural network flows on CPU and GPU, respectively.

In the CPU BP-NN code, the parallelization of the symmetry functions is carried out such that different dimers and trimers are dealing with the same permuted term synchronously. The data arrays are organized such that the same variables from different dimers and trimers are allocated in contiguous memory to enable vectorization. Workflows of neural networks employ relevant cBLAS functions for optimized performance. Both forward and backward flows (to compute forces) in symmetry functions and in neural networks are parallelized and optimized.

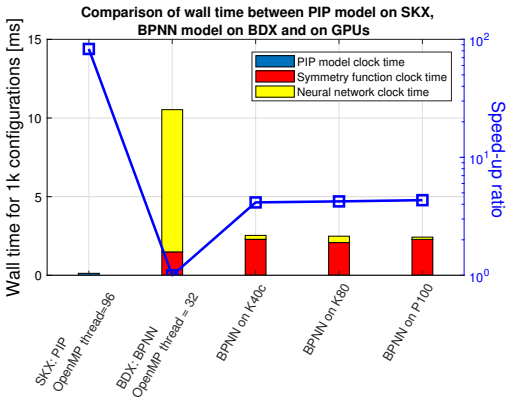

Figure 1: Comparison of wall clock time and speed-up ratio for energies of 1,000 water dimers/trimers with the PIP model on SKX, BPNN model on BDX and GPUs based many-body potential.

In the GPU BP-NN code, the parallelization of the symmetry functions is implemented such that the calculation of identical terms over all dimers and trimers are distributed over CUDA blocks. The block size is set to maximize the use of shared memory on the available GPU hadrware. Neural networks on the GPU launch functions from cuBLAS and cuDNN for batches of dimers and trimers to optimize the GPU utilization.

BENCHMARK RESULTS

We benchmarked our PIP and BP-NN implementations for data sets containing more than 10,000 dimer and trimer configurations, which is representative of each time step of a water bulk phase simulation. Benchmarks were performed on SDSC Comet, TACC Stampede, and an Nvidia development cluster using Intel HSX, BDX, and SKX processors with different numbers of OpenMP threads and on Nvidia Tesla K80, P100, and K40c GPUs. We investigated the performance of both energy and force (gradient) calculations.

Figure 1 shows the wall clock time and speed-up ratio for energies with the PIP implementation running on SKX, as well as BPNN model on BDX and GPUs. Due to large number of FLOPS BPNN on CPU is around 40 times slower than the PIP model that is run with similar configuration. The deep neural network in BP-NN GPU implementation benefits from its supporting libraries gaining an improvement factor of 60 on a single P100 compared with its CPU version. The symmetry function based on CUDA, however, does not achieve improvement on GPU. In most cases it is slightly slower than working on CPU leading to a total performance increase of 4.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation through grant ACI-1642336. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant ACI-1548562. We also thank Nvidia for access to a GPU development cluster for this work.

REFERENCES

[1] Volodymyr Babin, Claude Leforestier, and Francesco Paesani. 2013. Development of a "First principles" water potential with flexible monomers: dimer potential energy surface, VRT spectrum, and second virial coefficient. *J. Chem. Theory Comput.* 9, 12 (2013), 5395–5403. https://doi.org/10.1021/ct400863t

[2] Volodymyr Babin, Gregory R. Medders, and Francesco Paesani. 2014. Development of a "First principles" water potential with flexible monomers. II: Trimer potential energy surface, third virial coefficient, and small clusters. *J. Chem. Theory Comput.* 10, 4 (2014), 1599–1607. https://doi.org/10.1021/ct500079y

[3] Jörg Behler. 2015. Constructing high-dimensional neural network potentials: A tutorial review. *Int. J. Quant. Chem.* 115, 16 (March 2015), 1032–1050. https://doi.org/10.1002/qua.24890

[4] Jörg Behler and Michele Parrinello. 2007. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* 98, 14, Article 146401 (April 2007), 4 pages. https://doi.org/10.1103/PhysRevLett.98.146401

[5] Bastiaan J Braams and Joel M Bowman. 2009. Permutationally invariant potential energy surfaces in high dimensionality. *Int. Rev. Phys. Chem.* 28, 4 (2009), 577–606.

[6] Gerardo Andrés Cisneros et al. 2016. Modeling molecular interactions in water: From pairwise to many-body potential energy functions. *Chem. Rev.* 116, 13 (2016), 7501–7528. https://doi.org/10.1021/acs.chemrev.5b00644 PMID: 27186804.

[7] Intel Corporation. 2018. Intel(R) Math Kernel Library 2018 Update 1 for Linux. https://software.intel.com/en-us/mkl-windows-developer-guide

[8] Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. 2016. Maple (2016).

[9] Thuong T. Nguyen, Eszter Székely, Giulio Imbalzano, Jörg Behler, Gábor Csányi, Michele Ceriotti, Andreas W. Götz, and Francesco Paesani. 2018. Comparison of permutationally invariant polynomials, neural networks, and Gaussian approximation potentials in representing water interactions through many-body expansions. *J. Chem. Phys.* 148, Article 241725 (2018), 11 pages. https://doi.org/10.1063/1.5024577

[10] NVIDIA Corporation. 2017. CUDA Toolkit Documentation v8.0. https://docs.nvidia.com/cuda/archive/8.0

[11] OpenMP Architecture Review Board. 2015. OpenMP Application Program Interface Version 4.5. https://www.openmp.org/wp-content/uploads/openmp-4.5.pdf

[12] Francesco Paesani. 2016. Getting the right answers for the right reasons: Toward predictive molecular simulations of water with many-body potential energy functions. *Acc. Chem. Res.* 49, 9 (2016), 1844–1851. https://doi.org/10.1021/acs.accounts.6b00285 PMID: 27548325.

[13] Sandeep K Reddy, Shelby C Straight, Pushp Bajaj, C Huy Pham, Marc Riera, Daniel R Moberg, Miguel A Morales, Chris Knight, Andreas W Götz, and Francesco Paesani. 2016. On the accuracy of the MB-pol many-body potential for water: Interaction energies, vibrational frequencies, and classical thermodynamic and dynamical properties from clusters to liquid water and ice. *J. Chem. Phys.* 145, 19, Article 194504 (2016), 13 pages.