Lessons-learned developing performance portable QMCPACK

IDEAS Seminar May 2023

Paul Kent (kentpr@ornl.gov)

Oak Ridge National Laboratory

Outline

- Brief introduction to Quantum Monte Carlo & QMCPACK
- Performance portability goals
- Challenges of using GPUs
- Development approach
- Summary

Aim: illustrate for other developers & code owners what has been productive for us and our ongoing pain points.

Acknowledgements

EXASCALE COMPUTING PROJECT

ECP QMCPACK team including

- Peter Doak (ORNL)
- William Godoy (ORNL)
- Ye Luo (ANL)

ECP SOLLVE project [OpenMP+LLVM]

OLCF, ALCF staff

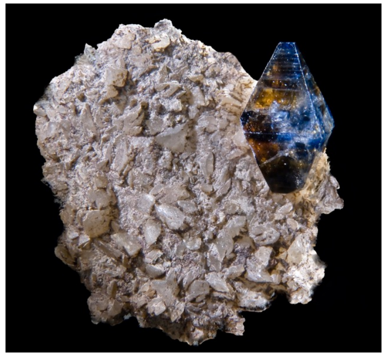
AMD, Intel, NVIDIA, HPE engineers

Quantum Monte Carlo

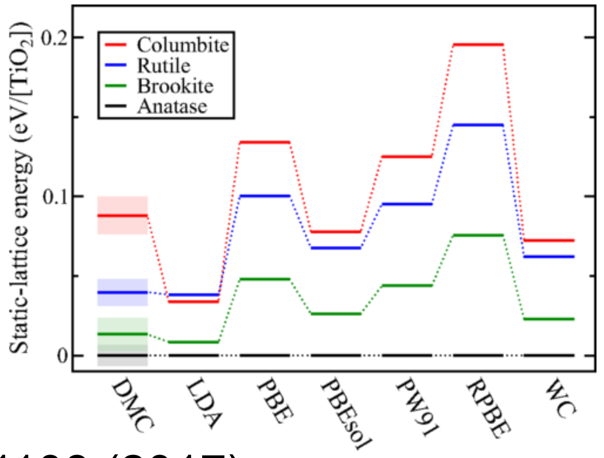
- The most accurate, general approach for solving Schrodinger's equation for "real" materials. [Foulkes RMP 2001]
- The few approximations in QMC can be tested, unlike standard methods. Nominally N^3. **Tradeoff: large computational cost.**
- Not exact, but very accurate today, can treat "strong" electron correlation, applicable to metals, insulators & molecules.
- For details and tutorials, see QMCPACK YouTube channel & https://github.com/QMCPACK/qmc_workshop_2021

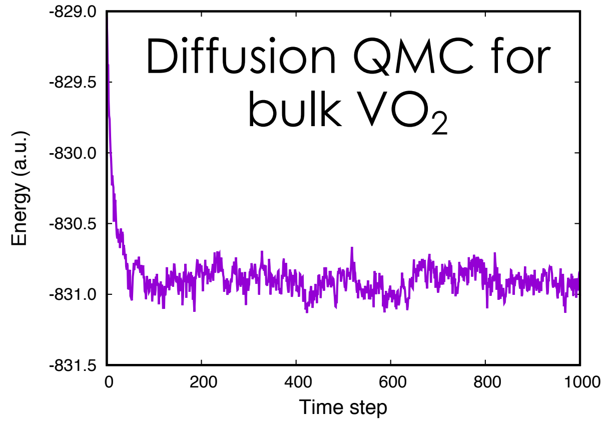
$$\frac{\partial|\psi\rangle}{\partial\tau} = -\hat{H}|\psi\rangle$$

$$|\psi(\delta\tau)\rangle = \sum_{i=0}^{\infty} c_i e^{-\epsilon_i\delta\tau}|\phi_i\rangle$$


Anatase
Didier Descouens CC-BY-SA 4.0

Rutile
Robert Lavinsky CC-BY-SA 3.0





J. Trail et al. PRB **95** 121108 (2017)
Y. Luo et al. NJP **18** 113049 (2016)

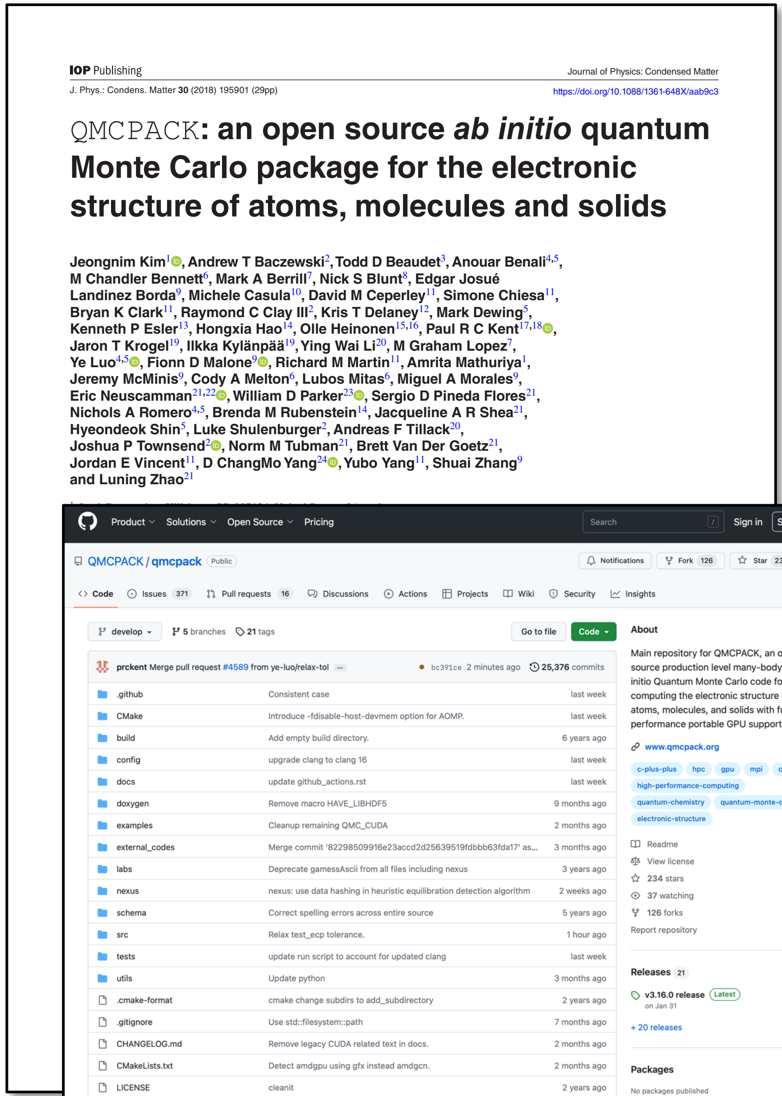
Performance Portability Goals

1. Run performantly on the full range of hardware, from laptops through to the #1 HPC machine and all 3 main vendor GPUs.
2. Use a single code path on all architectures, to the extent possible. Minimize maintenance burden, increase quality.
3. Retain ability to use specialized hardware & software, where merited.

QMCPACK

QMCPACK.org & GitHub.com/QMCPACK

- Open source, openly developed on GitHub, ~quarterly releases. Contributors credited on citation papers.
- C++17, HDF5, OpenMP+optional CUDA/HIP/SYCL+vendor dense linear algebra libraries. Highly vectorized, mixed precision supported.
- O(2x10^5) source lines.
- Science production using OpenMP target offload on NV GPUs with release versions of LLVM.
- New design has flexible dispatch, solves data movement and CPU fallback problem. Will always run unlike "legacy" GPU version.
- Code has undergone several major transitions: AoS to SoA CPU code for KNL, removal of "legacy" GPU version, ongoing removal of old CPU code paths.

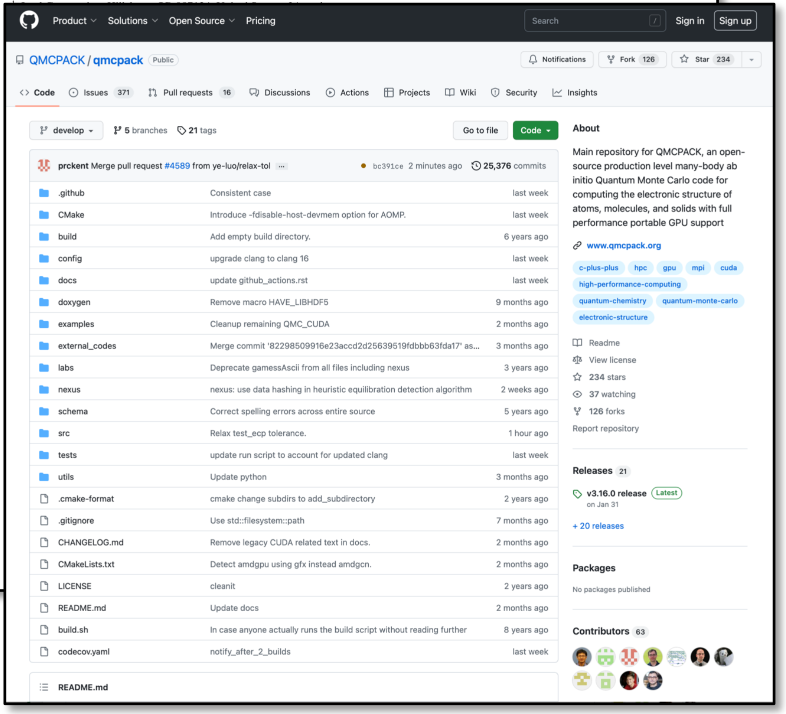


IOP Publishing — Journal of Physics: Condensed Matter

J. Phys.: Condens. Matter **30** (2018) 195901 (29pp) https://doi.org/10.1088/1361-648X/aab9c3

QMCPACK: an open source *ab initio* quantum Monte Carlo package for the electronic structure of atoms, molecules and solids

Jeongnim Kim[1], Andrew T Baczewski[2], Todd D Beaudet[3], Anouar Benali[4,5], M Chandler Bennett[6], Mark A Berrill[7], Nick S Blunt[8], Edgar Josué Landinez Borda[9], Michele Casula[10], David M Ceperley[11], Simone Chiesa[11], Bryan K Clark[11], Raymond C Clay III[2], Kris T Delaney[12], Mark Dewing[5], Kenneth P Esler[13], Hongxia Hao[14], Olle Heinonen[15,16], Paul R C Kent[17,18], Jaron T Krogel[19], Ilkka Kylänpää[19], Ying Wai Li[20], M Graham Lopez[7], Ye Luo[4,5], Fionn D Malone[9], Richard M Martin[11], Amrita Mathuriya[1], Jeremy McMinis[9], Cody A Melton[6], Lubos Mitas[6], Miguel A Morales[9], Eric Neuscamman[21,22], William D Parker[23], Sergio D Pineda Flores[21], Nichols A Romero[4,5], Brenda M Rubenstein[14], Jacqueline A R Shea[21], Hyeondeok Shin[5], Luke Shulenburger[2], Andreas F Tillack[20], Joshua P Townsend[2], Norm M Tubman[21], Brett Van Der Goetz[21], Jordan E Vincent[11], D ChangMo Yang[24], Yubo Yang[11], Shuai Zhang[9] and Luning Zhao[21]



QMCPACK / qmcpack Public

5 branches 21 tags

prckent Merge pull request #4589 from ye-luo/relax-tol ... bc391ce 2 minutes ago 25,376 commits

.github	Consistent case	last week
CMake	Introduce -fdisable-host-devmem option for AOMP.	last week
build	Add empty build directory.	6 years ago
config	upgrade clang to clang 16	last week
docs	update github_actions.rst	last week
doxygen	Remove macro HAVE_LIBHDF5	9 months ago
examples	Cleanup remaining QMC_CUDA	2 months ago
external_codes	Merge commit '82298509916e23accd2d25639519fdbbb63fda17' as...	3 months ago
labs	Deprecate gamessAscii from all files including nexus	3 years ago
nexus	nexus: use data hashing in heuristic equilibration detection algorithm	2 weeks ago
schema	Correct spelling errors across entire source	5 years ago
src	Relax test_ecp tolerance.	1 hour ago
tests	update run script to account for updated clang	last week
utils	Update python	3 months ago
.cmake-format	cmake change subdirs to add_subdirectory	2 years ago
.gitignore	Use std::filesystem::path	7 months ago
CHANGELOG.md	Remove legacy CUDA related text in docs.	2 months ago
CMakeLists.txt	Detect amdgpu using gfx instead amdgcn.	2 months ago
LICENSE	cleanit	2 years ago
README.md	Update docs	2 months ago
build.sh	In case anyone actually runs the build script without reading further	8 years ago
codecov.yaml	notify_after_2_builds	last week

README.md

About

Main repository for QMCPACK, an open-source production level many-body ab initio Quantum Monte Carlo code for computing the electronic structure of atoms, molecules, and solids with full performance portable GPU support

www.qmcpack.org

c-plus-plus hpc gpu mpi cuda high-performance-computing quantum-chemistry quantum-monte-carlo electronic-structure

Readme
View license
234 stars
37 watching
126 forks
Report repository

Releases 21

v3.16.0 release Latest
on Jan 31

+ 20 releases

Packages

No packages published

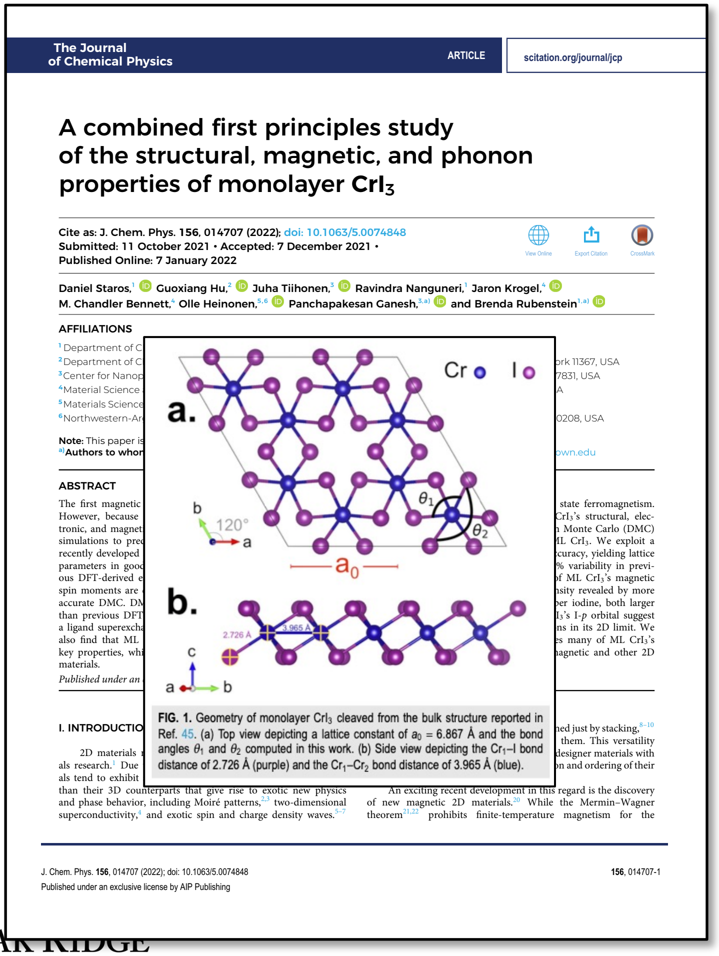
Contributors 63

OAK RIDGE National Laboratory

Recent QMC studies using QMCPACK

CrI_3 monolayers Staros JCP **156** 014707 (2022), H phase diagram Niu PRL **130** 076102 (2023), $>10^3$ molecules Huang JCTC **19** 1712 (2023). We aim to support and accelerate all of these calculations.

$O(10^3)$ electrons



The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

A combined first principles study of the structural, magnetic, and phonon properties of monolayer CrI_3

Cite as: J. Chem. Phys. 156, 014707 (2022); doi: 10.1063/5.0074848
Submitted: 11 October 2021 • Accepted: 7 December 2021 •
Published Online: 7 January 2022

View Online Export Citation CrossMark

Daniel Staros,[1] Guoxiang Hu,[2] Juha Tiihonen,[3] Ravindra Nanguneri,[1] Jaron Krogel,[4] M. Chandler Bennett,[4] Olle Heinonen,[5,6] Panchapakesan Ganesh,[3,a)] and Brenda Rubenstein[1,a)]

AFFILIATIONS

[1] Department of C … ork 11367, USA
[2] Department of C … 7831, USA
[3] Center for Nanop … A
[4] Material Science
[5] Materials Science
[6] Northwestern-Ar … 0208, USA

Note: This paper is
[a)] Authors to whom … own.edu

ABSTRACT

The first magnetic … state ferromagnetism. However, because … CrI₃'s structural, electronic, and magnet… n Monte Carlo (DMC) simulations to pred… ML CrI₃. We exploit a recently developed … curacy, yielding lattice parameters in good … % variability in previous DFT-derived e… of ML CrI₃'s magnetic spin moments are … nsity revealed by more accurate DMC. DM… per iodine, both larger than previous DFT … I₃'s I-p orbital suggest a ligand superexcha… ns in its 2D limit. We also find that ML … es many of ML CrI₃'s key properties, whi… agnetic and other 2D materials.

Published under an

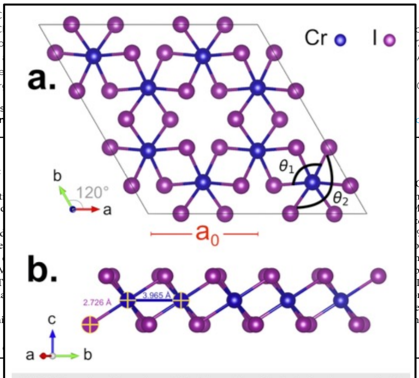


FIG. 1. Geometry of monolayer CrI_3 cleaved from the bulk structure reported in Ref. 45. (a) Top view depicting a lattice constant of a_0 = 6.867 Å and the bond angles θ_1 and θ_2 computed in this work. (b) Side view depicting the Cr_1–I bond distance of 2.726 Å (purple) and the Cr_1–Cr_2 bond distance of 3.965 Å (blue).

I. INTRODUCTIO

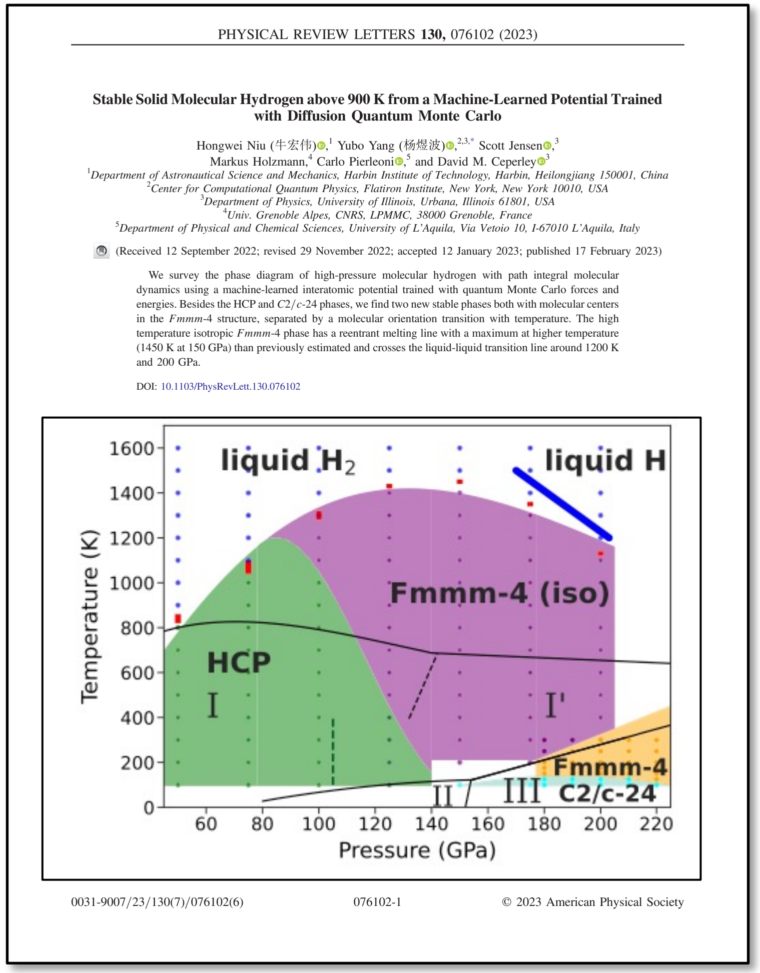
2D materials … als research.[1] Due … als tend to exhibit … than their 3D counterparts that give rise to exotic new physics and phase behavior, including Moiré patterns,[2,3] two-dimensional superconductivity,[4] and exotic spin and charge density waves.[5–7]

… ned just by stacking,[8–10] … them. This versatility … designer materials with … on and ordering of their

An exciting recent development in this regard is the discovery of new magnetic 2D materials.[20] While the Mermin–Wagner theorem[21,22] prohibits finite-temperature magnetism for the

J. Chem. Phys. **156**, 014707 (2022); doi: 10.1063/5.0074848 **156**, 014707-1
Published under an exclusive license by AIP Publishing

$O(10^2)$ electrons



PHYSICAL REVIEW LETTERS **130,** 076102 (2023)

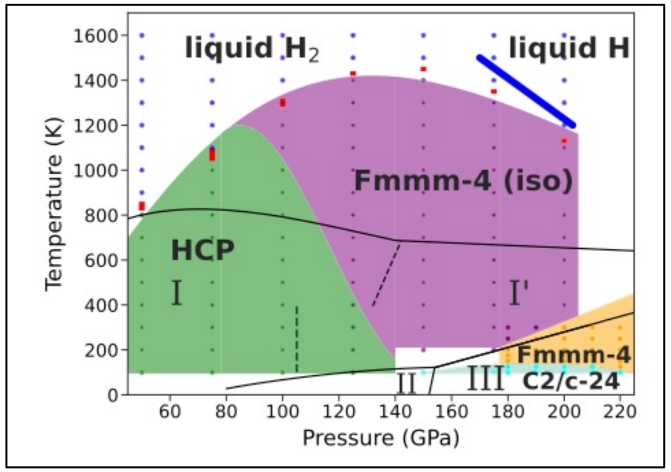
Stable Solid Molecular Hydrogen above 900 K from a Machine-Learned Potential Trained with Diffusion Quantum Monte Carlo

Hongwei Niu (牛宏伟),[1] Yubo Yang (杨煜波),[2,3,*] Scott Jensen,[3] Markus Holzmann,[4] Carlo Pierleoni,[5] and David M. Ceperley[3]
[1]Department of Astronautical Science and Mechanics, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China
[2]Center for Computational Quantum Physics, Flatiron Institute, New York, New York 10010, USA
[3]Department of Physics, University of Illinois, Urbana, Illinois 61801, USA
[4]Univ. Grenoble Alpes, CNRS, LPMMC, 38000 Grenoble, France
[5]Department of Physical and Chemical Sciences, University of L'Aquila, Via Vetoio 10, I-67010 L'Aquila, Italy

(Received 12 September 2022; revised 29 November 2022; accepted 12 January 2023; published 17 February 2023)

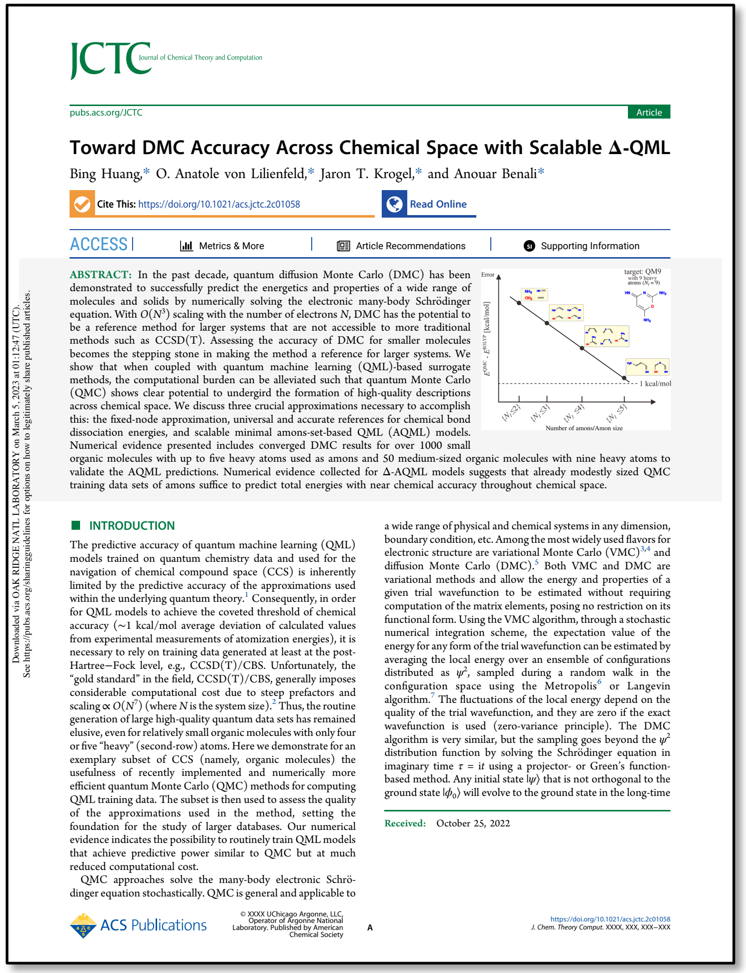
We survey the phase diagram of high-pressure molecular hydrogen with path integral molecular dynamics using a machine-learned interatomic potential trained with quantum Monte Carlo forces and energies. Besides the HCP and $C2/c$-24 phases, we find two new stable phases both with molecular centers in the $Fmmm$-4 structure, separated by a molecular orientation transition with temperature. The high temperature isotropic $Fmmm$-4 phase has a reentrant melting line with a maximum at higher temperature (1450 K at 150 GPa) than previously estimated and crosses the liquid-liquid transition line around 1200 K and 200 GPa.

DOI: 10.1103/PhysRevLett.130.076102



0031-9007/23/130(7)/076102(6) 076102-1 © 2023 American Physical Society

$O(10^{1-2})$ electrons



JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

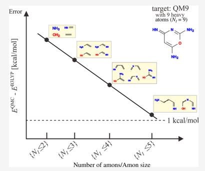
Toward DMC Accuracy Across Chemical Space with Scalable Δ-QML

Bing Huang,* O. Anatole von Lilienfeld,* Jaron T. Krogel,* and Anouar Benali*

Cite This: https://doi.org/10.1021/acs.jctc.2c01058 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: In the past decade, quantum diffusion Monte Carlo (DMC) has been demonstrated to successfully predict the energetics and properties of a wide range of molecules and solids by numerically solving the electronic many-body Schrödinger equation. With $O(N^3)$ scaling with the number of electrons N, DMC has the potential to be a reference method for larger systems that are not accessible to more traditional methods such as CCSD(T). Assessing the accuracy of DMC for smaller molecules becomes the stepping stone in making the method a reference for larger systems. We show that when coupled with quantum machine learning (QML)-based surrogate methods, the computational burden can be alleviated such that quantum Monte Carlo (QMC) shows clear potential to undergird the formation of high-quality descriptions across chemical space. We discuss three crucial approximations necessary to accomplish this: the fixed-node approximation, universal and accurate references for chemical bond dissociation energies, and scalable minimal amons-set-based QML (AQML) models. Numerical evidence presented includes converged DMC results for over 1000 small organic molecules with up to five heavy atoms used as amons and 50 medium-sized organic molecules with nine heavy atoms to validate the AQML predictions. Numerical evidence collected for Δ-AQML models suggests that already modestly sized QMC training data sets of amons suffice to predict total energies with near chemical accuracy throughout chemical space.



Downloaded via OAK RIDGE NATL LABORATORY on March 5, 2023 at 01:12:47 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

INTRODUCTION

The predictive accuracy of quantum machine learning (QML) models trained on quantum chemistry data and used for the navigation of chemical compound space (CCS) is inherently limited by the predictive accuracy of the approximations used within the underlying quantum theory.[1] Consequently, in order for QML models to achieve the coveted threshold of chemical accuracy (~1 kcal/mol average deviation of calculated values from experimental measurements of atomization energies), it is necessary to rely on training data generated at least at the post-Hartree–Fock level, e.g., CCSD(T)/CBS. Unfortunately, the "gold standard" in the field, CCSD(T)/CBS, generally imposes considerable computational cost due to steep prefactors and scaling ∝ $O(N^7)$ (where N is the system size).[2] Thus, the routine generation of large high-quality quantum data sets has remained elusive, even for relatively small organic molecules with only four or five "heavy" (second-row) atoms. Here we demonstrate for an exemplary subset of CCS (namely, organic molecules) the usefulness of recently implemented and numerically more efficient quantum Monte Carlo (QMC) methods for computing QML training data. The subset is then used to assess the quality of the approximations used in the method, setting the foundation for the study of larger databases. Our numerical evidence indicates the possibility to routinely train QML models that achieve predictive power similar to QMC but at much reduced computational cost.

QMC approaches solve the many-body electronic Schrödinger equation stochastically. QMC is general and applicable to a wide range of physical and chemical systems in any dimension, boundary condition, etc. Among the most widely used flavors for electronic structure are variational Monte Carlo (VMC)[3,4] and diffusion Monte Carlo (DMC).[5] Both VMC and DMC are variational methods and allow the energy and properties of a given trial wavefunction to be estimated without requiring computation of the matrix elements, posing no restriction on its functional form. Using the VMC algorithm, through a stochastic numerical integration scheme, the expectation value of the energy for any form of the trial wavefunction can be estimated by averaging the local energy over an ensemble of configurations distributed as ψ^2, sampled during a random walk in the configuration space using the Metropolis[6] or Langevin algorithm.[7] The fluctuations of the local energy depend on the quality of the trial wavefunction, and they are zero if the exact wavefunction is used (zero-variance principle). The DMC algorithm is very similar, but the sampling goes beyond the ψ^2 distribution function by solving the Schrödinger equation in imaginary time $\tau = it$ using a projector- or Green's function-based method. Any initial state $|\psi\rangle$ that is not orthogonal to the ground state $|\phi_0\rangle$ will evolve to the ground state in the long-time

Received: October 25, 2022

ACS Publications © XXXX UChicago Argonne, LLC, Operator of Argonne National Laboratory. Published by American Chemical Society A https://doi.org/10.1021/acs.jctc.2c01058 J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

OAK RIDGE National Laboratory

Challenge of exploiting GPUs

- Exascale-generation GPUs from NVIDIA, AMD, and Intel have $>10^4$ compute elements. Need $>10^6$ similar operations in flight for optimum performance.
- If we only have 10^{2-4} electrons, there naively will not be enough work.
- Generally, no single hot kernel. Kernels are both compute & memory bound.
- Few proven designs. QMC less mature than, e.g., quantum chemistry and classical molecular dynamics where multiple performant implementations are available.

NVIDIA A100 GPUs & AMD Milan CPUs

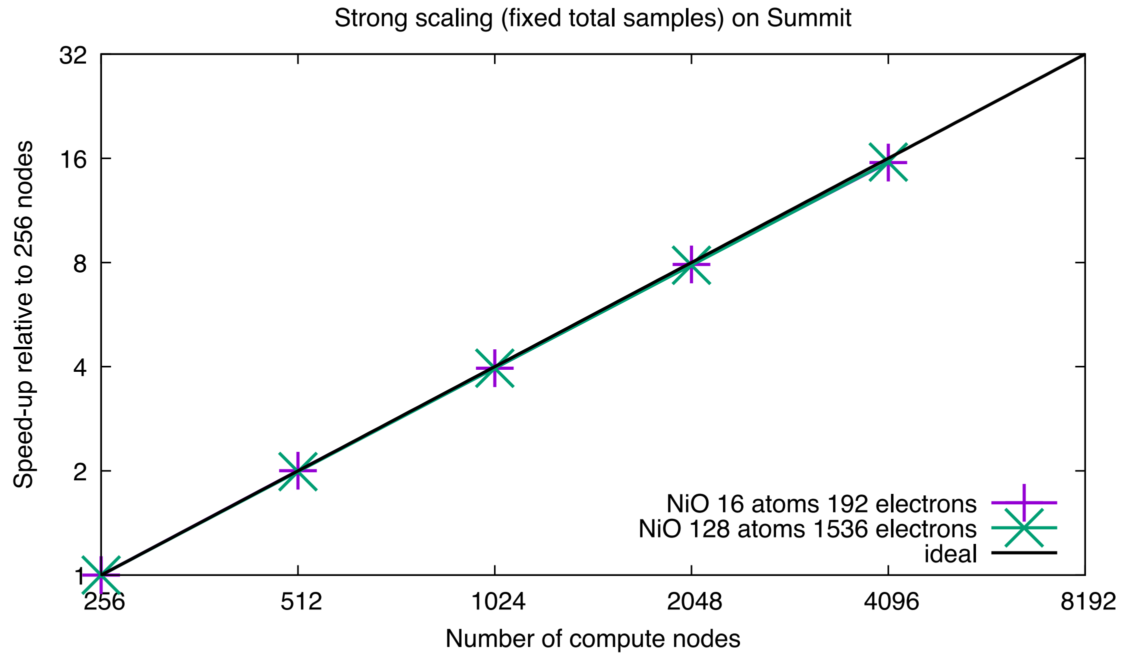
AMD GPUs and AMD CPUs

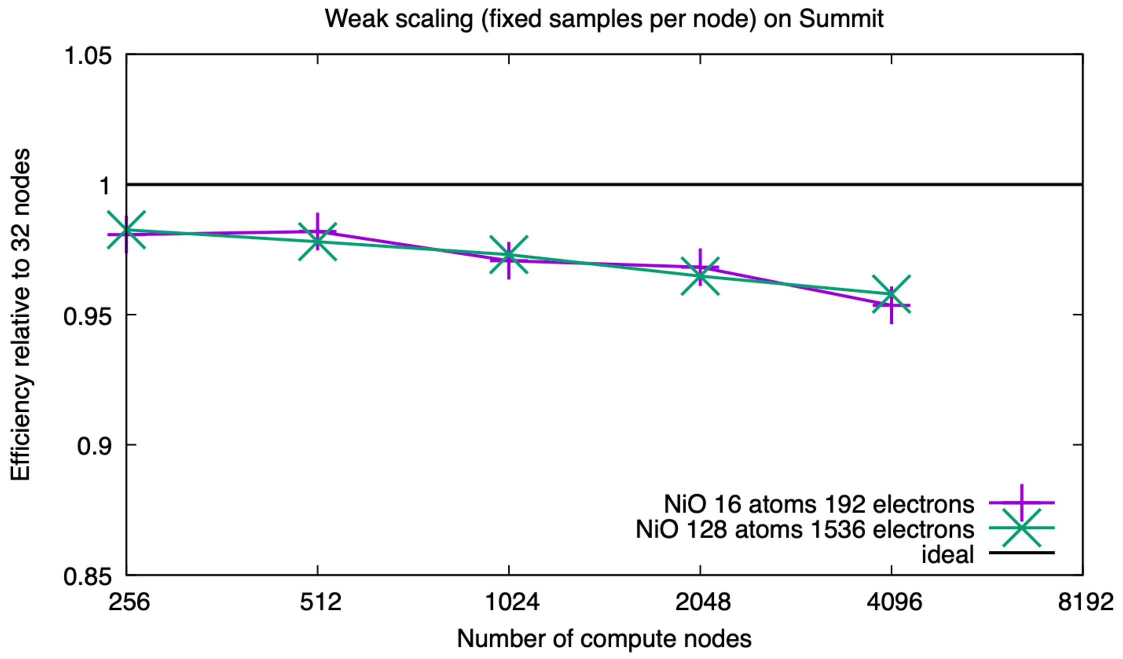
Intel Xe GPUs and Xeon CPUs

Parallel Scalability

Despite needing communications every timestep, scalability is high due to high computational cost/step, careful MPI implementation.

See Kim et al. JPCM (2018) 10.1088/1361-648X/aab9c3



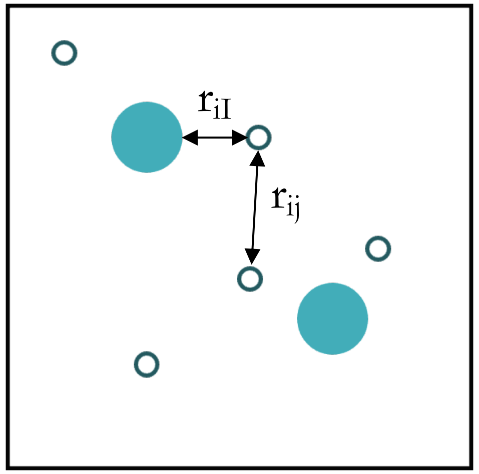


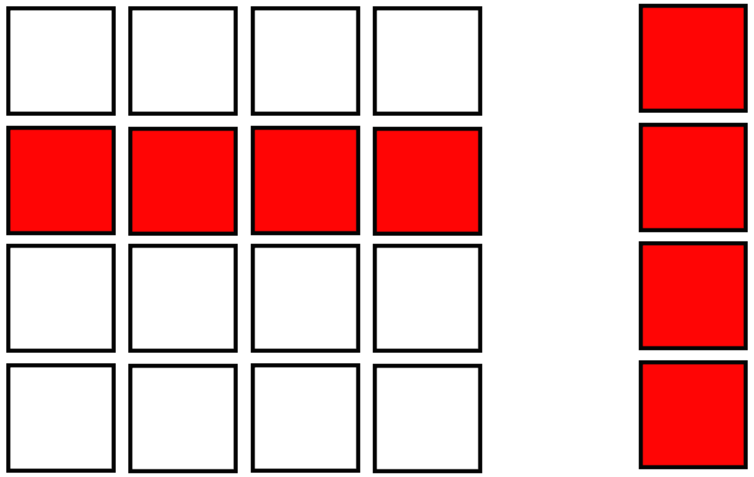
Key operations

Real space QMC uses both particle-based and dense linear algebra operations. Particle counts + matrix sizes can be small (10^2-10^4), requiring different choices to standard classical molecular dynamics or quantum chemistry techniques.

Particle operations
movement, interparticle distances, functions of position, minimum image when periodic

Dense vector, matrix operations
Spline and Gaussian basis set evaluation, determinant update, wavefunction optimization, BLAS1-3



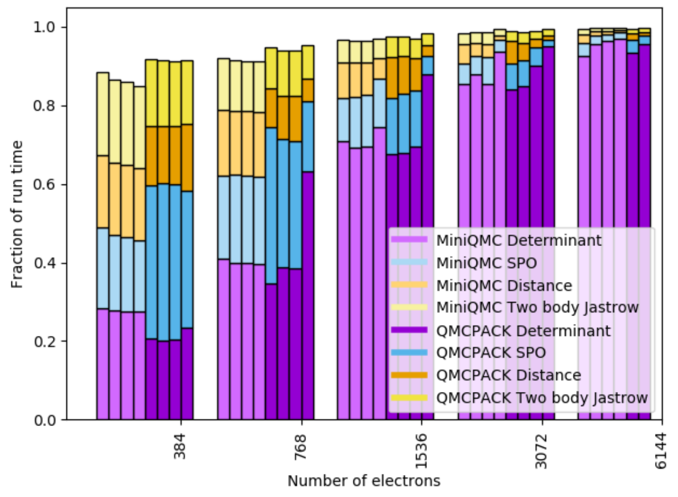


Example Qs: Is it worth maintaining neighbor lists? Benefits, tradeoffs from sparsity?

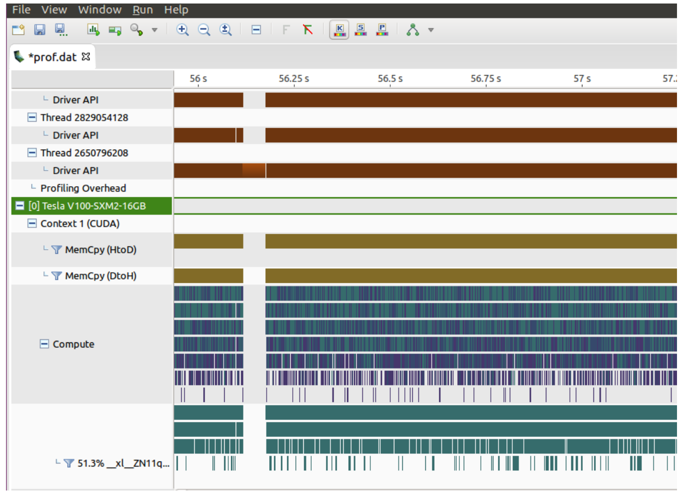
Oak Ridge National Laboratory

Miniqmc miniapp for design & performance experiments

- https://github.com/QMCPACK/miniqmc
- Order of magnitude smaller than QMCPACK
- Resulted in new design of QMCPACK with revised internal APIs and flexible runtime dispatch.
- Picked OpenMP target offload as default implementation route, supplemented if needed by vendor specific optimized code.
- Miniapp requires ongoing effort to maintain and update to keep synced with main application. => Unfortunately, miniqmc is currently out of date…



Profile validated vs main app ("proxy not imposter")



Easier profiling of miniapp aids in testing multithreaded offload strategy

Algorithmic Challenge – How to map QMC to GPUs?

Canonical QMC Algorithm

```
do time step i   [ 1K-100K ]
    do walker j  [ M walkers, ~1 per core, OpenMP ]
        do electron k [ N=$10^2$-$10^4$ ]
            propose new position $\underline{\mathbf{r}}_k$'  [ O(~1-$N^2$) cost ]
            evaluate $\boldsymbol{\Psi}(\underline{\mathbf{r}}_k')$  [ O(~N-$N^2$) cost ]
            accept/reject using ~$|\boldsymbol{\Psi}'|^2/|\boldsymbol{\Psi}|^2$
            if (accept) update $\boldsymbol{\Psi}$  [ O($N^2$) cost ]
        end k
        evaluate Hamiltonian, Observables
    end j
    spawn/kill walkers, load balance
end i
```

- Works well on CPUs. Usually, 1 CPU thread per Monte Carlo walker.
- For GPUs, simply offloading the compute for each walker is not efficient since there is not enough numerical work.

Previous GPU approach: Batching many independent walker moves

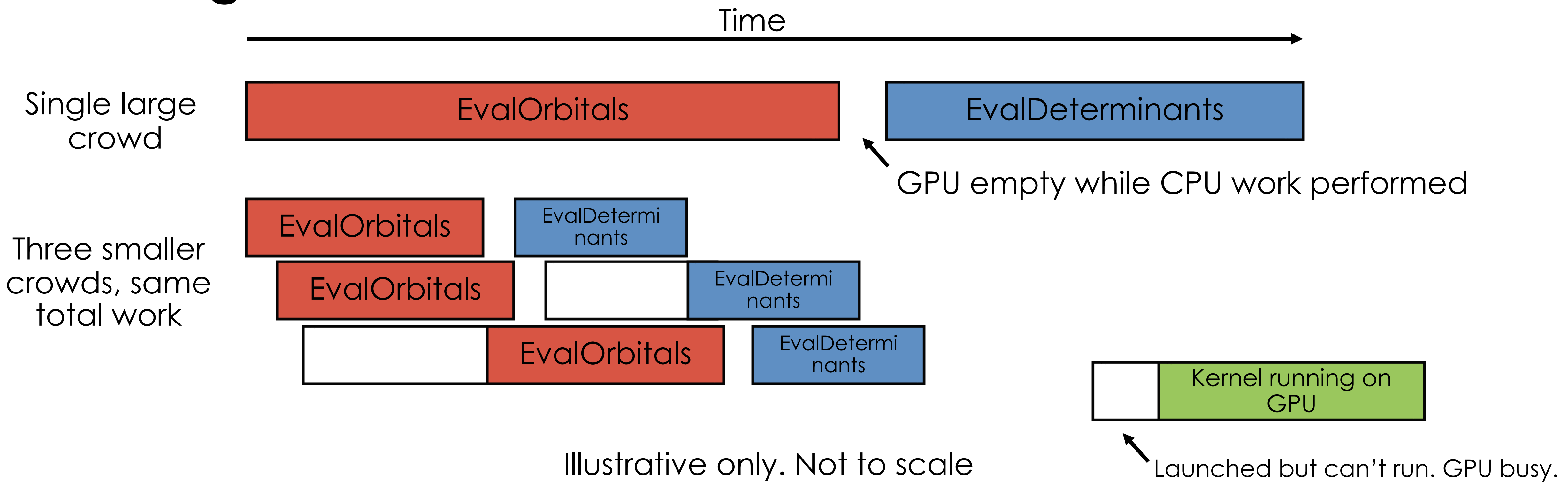
```
Batched Metropolis QMC Algorithm,  M walkers/node
do time step i   [ 1K-100K ]

        do electron k [ N=$10^2$-$10^4$ ]
                propose M new positions $\{\underline{\mathbf{r}}_k'\}_M$ [ O(~M-MN$^2$) cost ]
                evaluate $\{\Psi(\underline{\mathbf{r}}_k')\}_M$  [ O(~MN-MN$^2$) cost ]
                accept/reject using ~$|\Psi'|^2/|\Psi|^2$
                if (accept) update $\{\Psi\}_M$  [ O(MN$^2$) cost ]
        end k
        evaluate Hamiltonian, Observables for all M walkers

        spawn/kill walkers, load balance
end i
```

- Batch (group) all operations over M walkers (Markov chains), now operations are O(MxN) or O(MxN^2). Choose M large enough to saturate GPU, typically 10-1000.
- CUDA version runs very efficiently, Esler et al. CISE **14** 40 (2012)

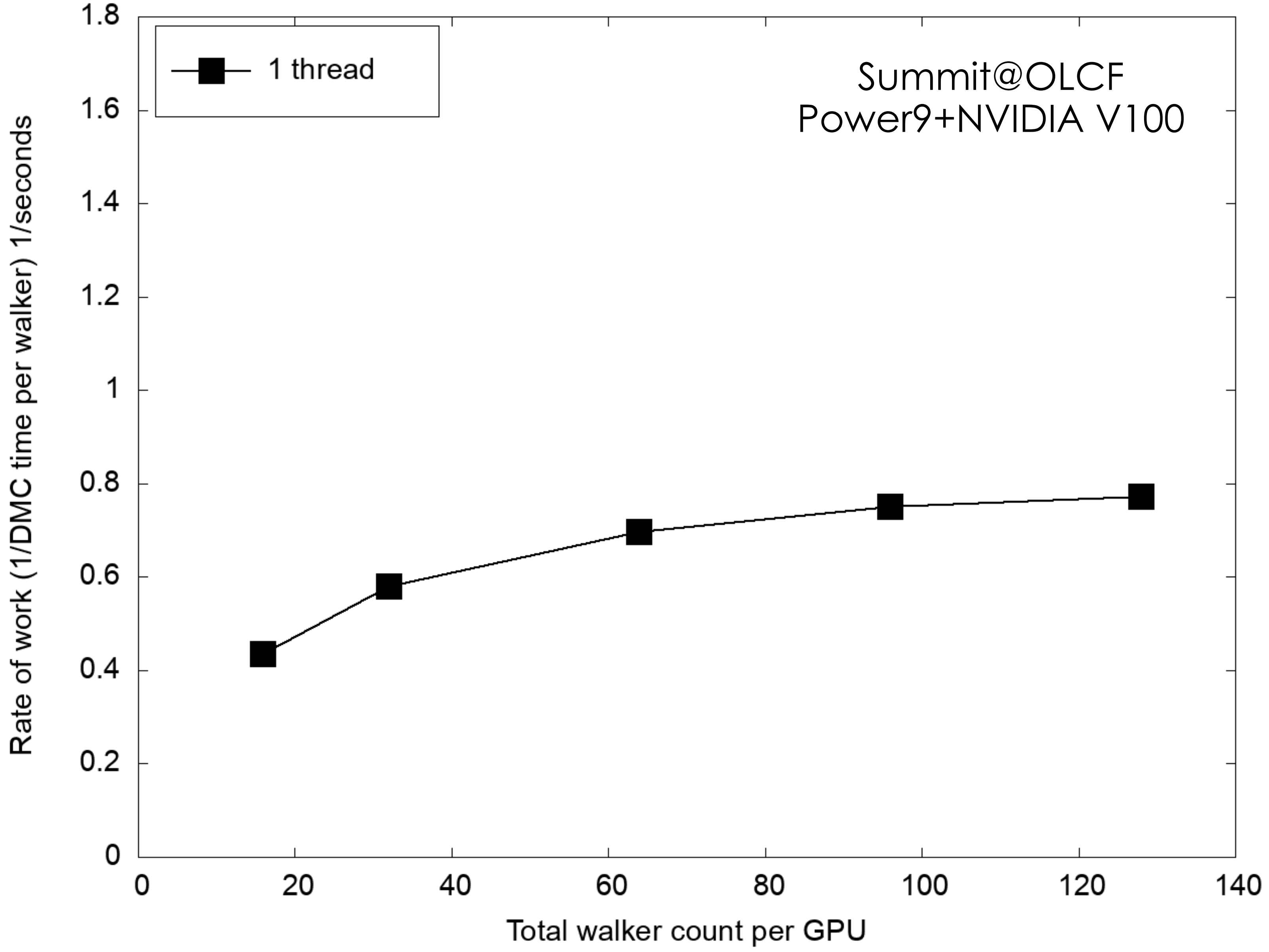
New approach: multithreaded offload Batching smaller "crowds" of walkers



Use multiple smaller batches ("crowds") launched from different host threads, not a single large batch.

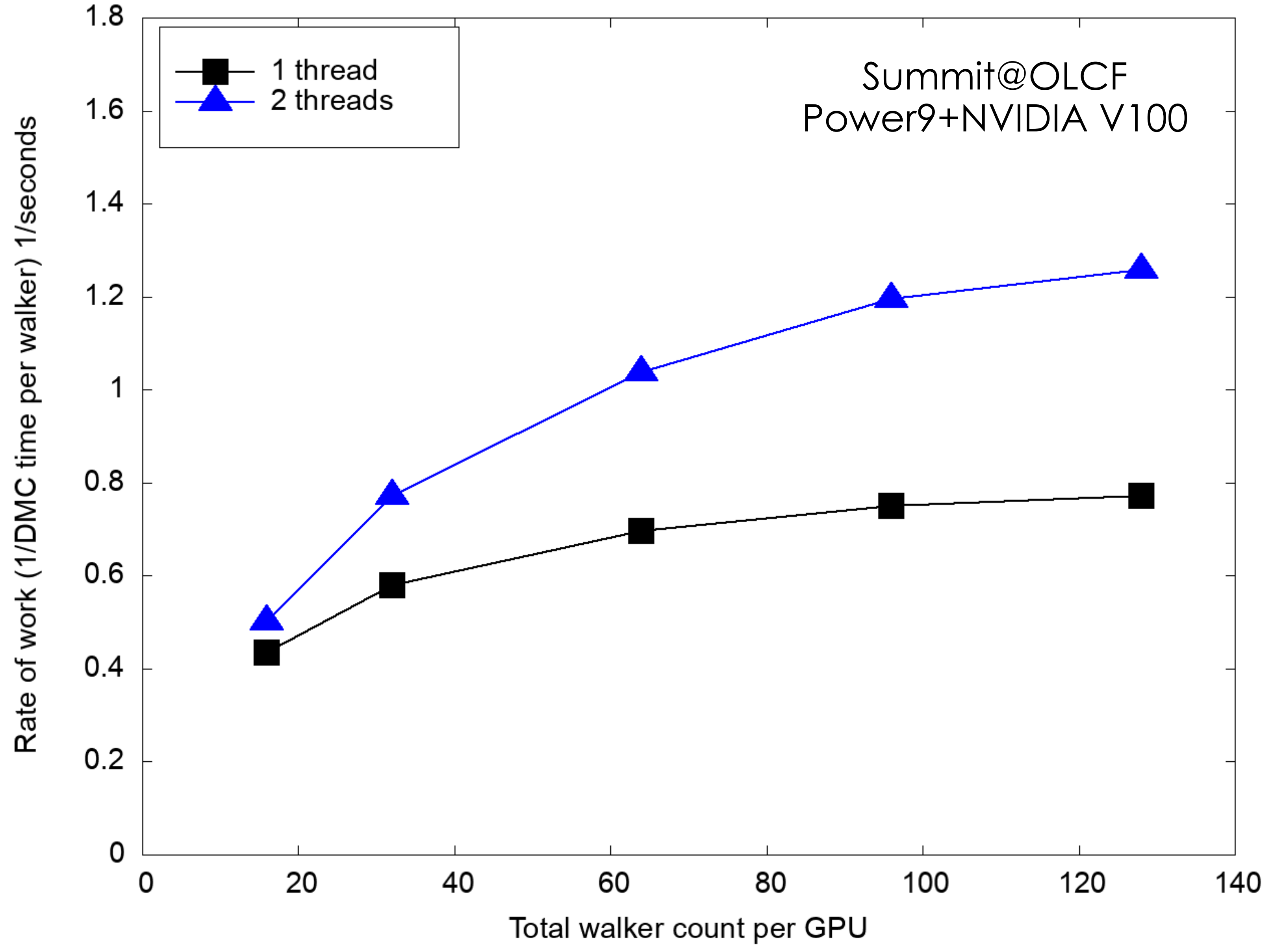
- Trades some kernel efficiency for more asynchronous work and potentially greater throughput. Highly dependent on problem, hw+sw stack.
- Can recover original GPU algorithm with 1 crowd/thread.
- Highly beneficial if there is any significant CPU work remaining

Results: 128 atoms NiO / 1536 electrons



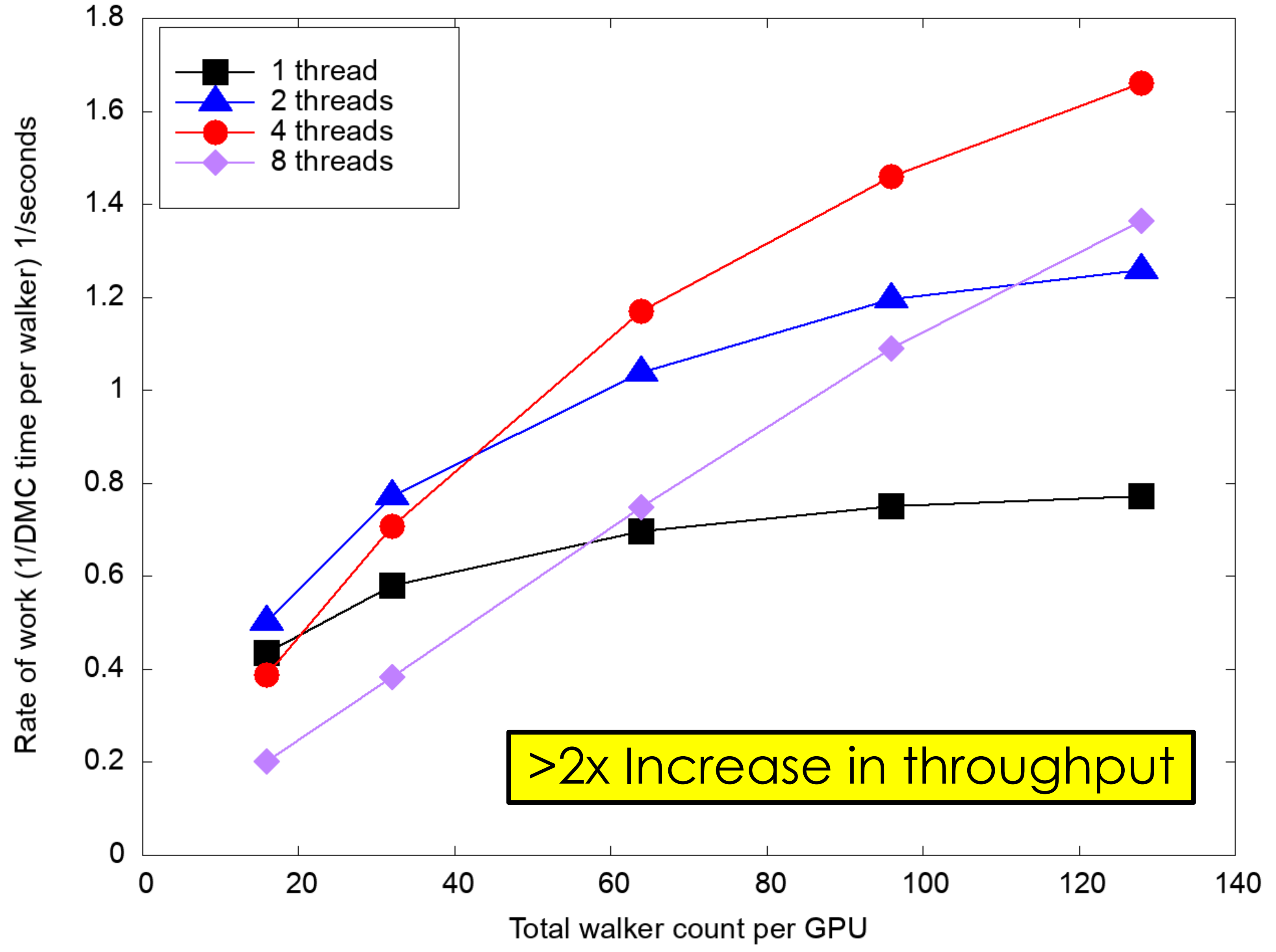
OAK RIDGE National Laboratory

Results: 128 atoms NiO / 1536 electrons



Oak Ridge National Laboratory

Results: 128 atoms NiO / 1536 electrons

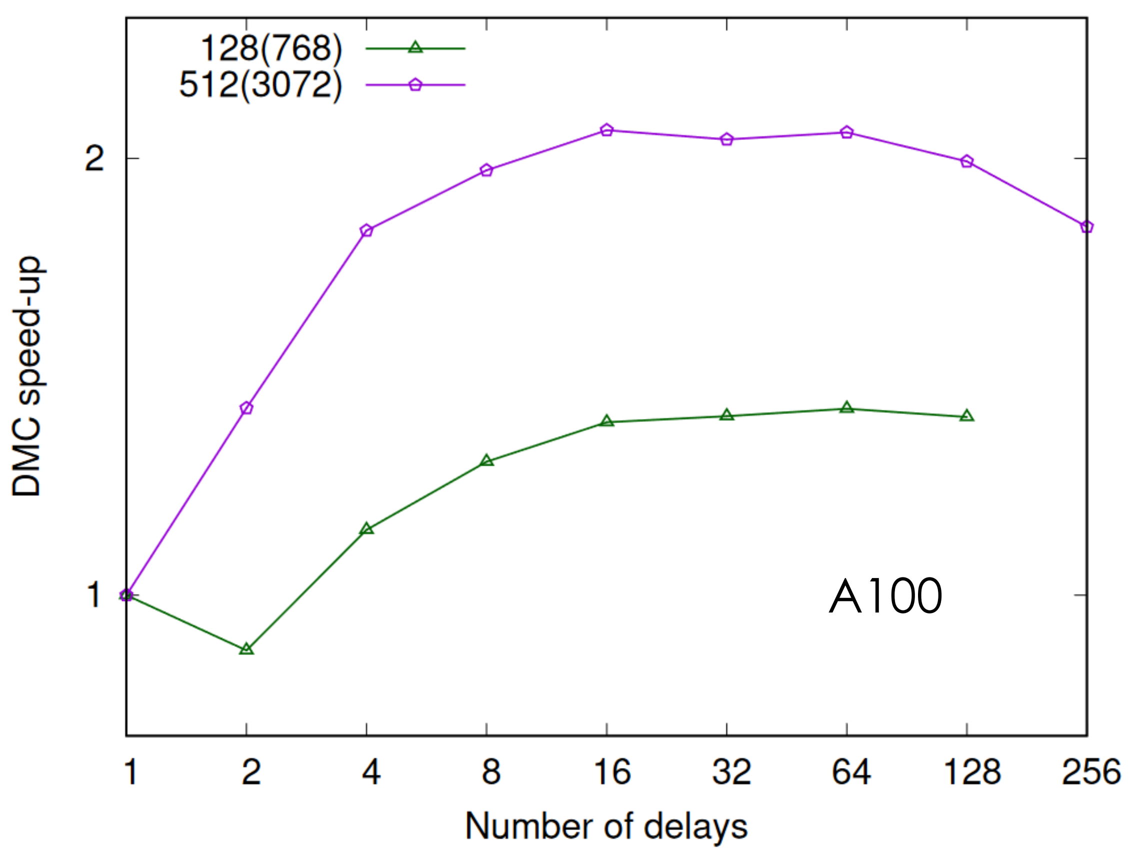


OAK RIDGE National Laboratory

Revisiting key algorithms

- We have replaced older algorithms with more compute bound/less memory bound algorithms.
- **New delayed update algorithm counterintuitively increases the operation count for higher performance.**
- Matrix multiply rich but extra work per step. Uses Sherman-Morrison-Woodbury formula to obtain wavefunction ratios during a delay period *n*, then update inverse. Avoid recalculating intermediates. Improves on our earlier algorithm (McDaniel 2017).
- ~2x faster on GPUs, ~10x faster on CPUs.

Bulk NiO supercells. Key: Atoms(Electrons per spin)



OAK RIDGE National Laboratory

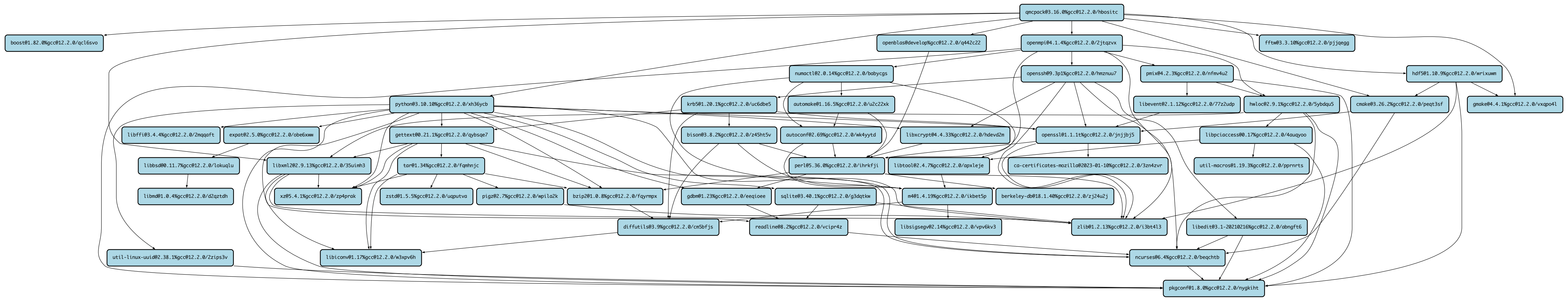
Development approach

General approach

- Focus on making the best use of our time
- Pragmatically adopt “best practices”. Refine based on actual data from code review, CI, tools experience.
- Keep barrier for new developers, open-source contributions low.
- Limit required dependencies. Define a support policy for compilers, libraries etc.

E.g. Transitioned documentation to use sphinx & readthedocs. Minimal barrier to doc edits, plus full CI on changes.

Ecosystem challenges: version control of dependencies

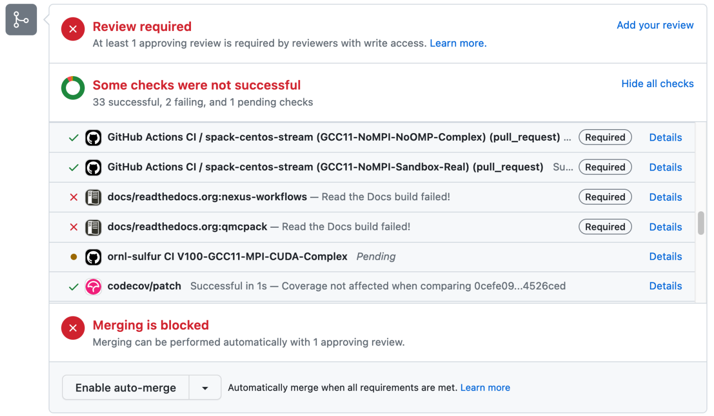

qmcpack@3.16.0%gcc@12.2.0/hbositc
boost@1.82.0%gcc@12.2.0/qcl6svo
openblas@develop%gcc@12.2.0/q442c22
openmpi@4.1.4%gcc@12.2.0/2jtqzvx
fftw@3.3.10%gcc@12.2.0/pjjqegg
numactl@2.0.14%gcc@12.2.0/babycgs
openssh@9.3p1%gcc@12.2.0/hmznuu7
pmix@4.2.3%gcc@12.2.0/nfmv4u2
hdf5@1.10.9%gcc@12.2.0/wrixuwm
python@3.10.10%gcc@12.2.0/xh36ycb
krb5@1.20.1%gcc@12.2.0/uc6dbe5
automake@1.16.5%gcc@12.2.0/u2c22xk
libevent@2.1.12%gcc@12.2.0/77z2udp
hwloc@2.9.1%gcc@12.2.0/5ybdqu5
cmake@3.26.2%gcc@12.2.0/peqt3sf
gmake@4.4.1%gcc@12.2.0/vxqpo4l
libffi@3.4.4%gcc@12.2.0/2mqqaft
expat@2.5.0%gcc@12.2.0/obe6xww
gettext@0.21.1%gcc@12.2.0/qybsqe7
bison@3.8.2%gcc@12.2.0/z45ht5v
autoconf@2.69%gcc@12.2.0/wk4yytd
libxcrypt@4.4.33%gcc@12.2.0/hdevd2m
openssl@1.1.1t%gcc@12.2.0/jnjjbj5
libpciaccess@0.17%gcc@12.2.0/4auqyoo
libbsd@0.11.7%gcc@12.2.0/lokuqlu
libxml2@2.9.13%gcc@12.2.0/35uimh3
tar@1.34%gcc@12.2.0/fqmhnjc
perl@5.36.0%gcc@12.2.0/ihrkfji
libtool@2.4.7%gcc@12.2.0/apxleje
ca-certificates-mozilla@2023-01-10%gcc@12.2.0/3zn4zvr
util-macros@1.19.3%gcc@12.2.0/pprnrts
libmd@1.0.4%gcc@12.2.0/d2qztdh
xz@5.4.1%gcc@12.2.0/zp4prak
zstd@1.5.5%gcc@12.2.0/uqputva
pigz@2.7%gcc@12.2.0/wpila2k
bzip2@1.0.8%gcc@12.2.0/fqyrmpx
gdbm@1.23%gcc@12.2.0/eeqioee
sqlite@3.40.1%gcc@12.2.0/g3dqtkw
m4@1.4.19%gcc@12.2.0/ikbet5p
berkeley-db@18.1.40%gcc@12.2.0/zj24u2j
diffutils@3.9%gcc@12.2.0/cm5bfjs
readline@8.2%gcc@12.2.0/vcipr4z
libsigsegv@2.14%gcc@12.2.0/vpv6kv3
zlib@1.2.13%gcc@12.2.0/i3bt4l3
libedit@3.1-20210216%gcc@12.2.0/abngft6
util-linux-uuid@2.38.1%gcc@12.2.0/2zips3v
libiconv@1.17%gcc@12.2.0/w3xpv6h
ncurses@6.4%gcc@12.2.0/beqchtb
pkgconf@1.8.0%gcc@12.2.0/nygkiht

Minimal QMCPACK dependencies excluding compilers, many optional python dependencies (spack graph output)

- Even "low dependency" apps have many dependencies. These are all undergoing development… changing GPU support, python module changes, HDF5 API updates etc. can all lead to breakage.
- For nightly testing, "version control" achived via spack package manager (https://spack.io/) to cover a sparse matrix of older/newer software. However, users use ~any combination of versions…

Testing and Continuous Integration

- Large set of unit, deterministic, & stochastic integration tests built up. Test subset run in CI, plus more extensive sets nightly and weekly for many different compiler, CPU, GPU, library combinations.
- Helps us make large changes to the code, onboard new developers, engage with vendors, contributors.
- Pull requests undergo review, testing, coverage reporting, sanitizer tests. Procedures for reviews developed, e.g., merger can not be at same institution as PR.
- CI uses GitHub actions, plus our own hardware, needed to test GPUs not available in cloud. Aim for O(1h) turnaround. Window for input to vendors for fixes in their next release is small.
- OpenMP target offload "GPU" code partly tested via offload to host CPUs with LLVM. Huge time, $ savings.


Review required
At least 1 approving review is required by reviewers with write access. Learn more.
Add your review

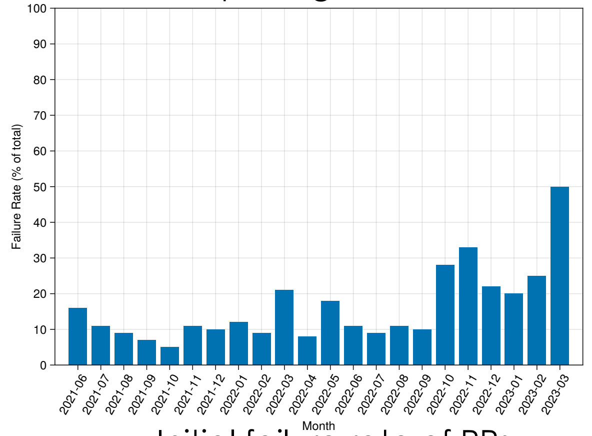
Some checks were not successful
33 successful, 2 failing, and 1 pending checks
Hide all checks

✓	GitHub Actions CI / spack-centos-stream (GCC11-NoMPI-NoOMP-Complex) (pull_request) ...	Required	Details
✓	GitHub Actions CI / spack-centos-stream (GCC11-NoMPI-Sandbox-Real) (pull_request) Su...	Required	Details
✗	docs/readthedocs.org:nexus-workflows — Read the Docs build failed!	Required	Details
✗	docs/readthedocs.org:qmcpack — Read the Docs build failed!	Required	Details
●	ornl-sulfur CI V100-GCC11-MPI-CUDA-Complex *Pending*		Details
✓	codecov/patch Successful in 1s — Coverage not affected when comparing 0cefe09...4526ced		Details

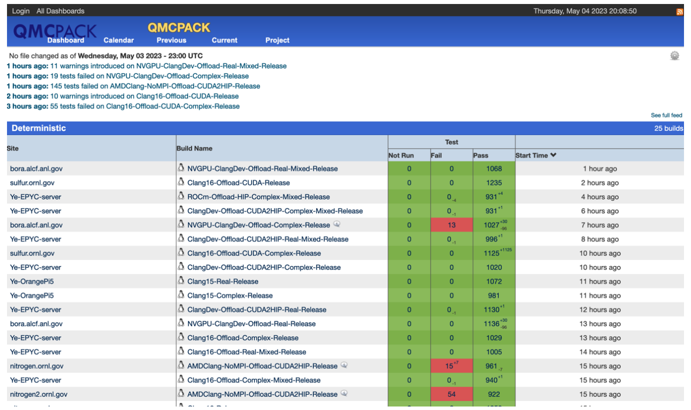
Merging is blocked
Merging can be performed automatically with 1 approving review.

Automatically merge when all requirements are met. Learn more

CI reporting on PRs



Initial failure rate of PRs



Thursday, May 04 2023 20:08:50

No file changed as of **Wednesday, May 03 2023 - 23:00 UTC**
1 hours ago: 11 warnings introduced on NVGPU-ClangDev-Offload-Real-Mixed-Release
1 hours ago: 19 tests failed on NVGPU-ClangDev-Offload-Complex-Release
1 hours ago: 145 tests failed on AMDClang-NoMPI-Offload-CUDA2HIP-Release
2 hours ago: 10 warnings introduced on Clang16-Offload-CUDA-Release
3 hours ago: 55 tests failed on Clang16-Offload-CUDA-Complex-Release

See full feed

Deterministic 25 builds

Site	Build Name	Test			
		Not Run	Fail	Pass	Start Time
bora.alcf.anl.gov	NVGPU-ClangDev-Offload-Real-Mixed-Release	0	0	1068	1 hour ago
sulfur.ornl.gov	Clang16-Offload-CUDA-Release	0	0	1235	2 hours ago
Ye-EPYC-server	ROCm-Offload-HIP-Complex-Mixed-Release	0	0_{-4}	931^{+4}	4 hours ago
Ye-EPYC-server	ClangDev-Offload-CUDA2HIP-Complex-Mixed-Release	0	0_{-1}	931^{+1}	6 hours ago
bora.alcf.anl.gov	NVGPU-ClangDev-Offload-Complex-Release	0	13	1027^{+30}_{-96}	7 hours ago
Ye-EPYC-server	ClangDev-Offload-CUDA2HIP-Real-Mixed-Release	0	0_{-1}	996^{+1}	8 hours ago
sulfur.ornl.gov	Clang16-Offload-CUDA-Complex-Release	0	0	1125^{+1125}	10 hours ago
Ye-EPYC-server	ClangDev-Offload-CUDA2HIP-Complex-Release	0	0	1020	10 hours ago
Ye-OrangePi5	Clang15-Real-Release	0	0	1072	11 hours ago
Ye-OrangePi5	Clang15-Complex-Release	0	0	981	11 hours ago
Ye-EPYC-server	ClangDev-Offload-CUDA2HIP-Real-Release	0	0_{-1}	1130^{+1}	12 hours ago
bora.alcf.anl.gov	NVGPU-ClangDev-Offload-Real-Release	0	0	1136^{+30}_{-96}	13 hours ago
Ye-EPYC-server	Clang16-Offload-Complex-Release	0	0	1029	13 hours ago
Ye-EPYC-server	Clang16-Offload-Real-Mixed-Release	0	0	1005	14 hours ago
nitrogen.ornl.gov	AMDClang-NoMPI-Offload-CUDA2HIP-Release	0	15^{+7}	961_{-7}	15 hours ago
Ye-EPYC-server	Clang16-Offload-Complex-Mixed-Release	0	0_{-1}	940^{+1}	15 hours ago
nitrogen2.ornl.gov	AMDClang-NoMPI-Offload-CUDA2HIP-Release	0	54	922	15 hours ago

QMCPACK cdash

Many edge case bugs found & resolved

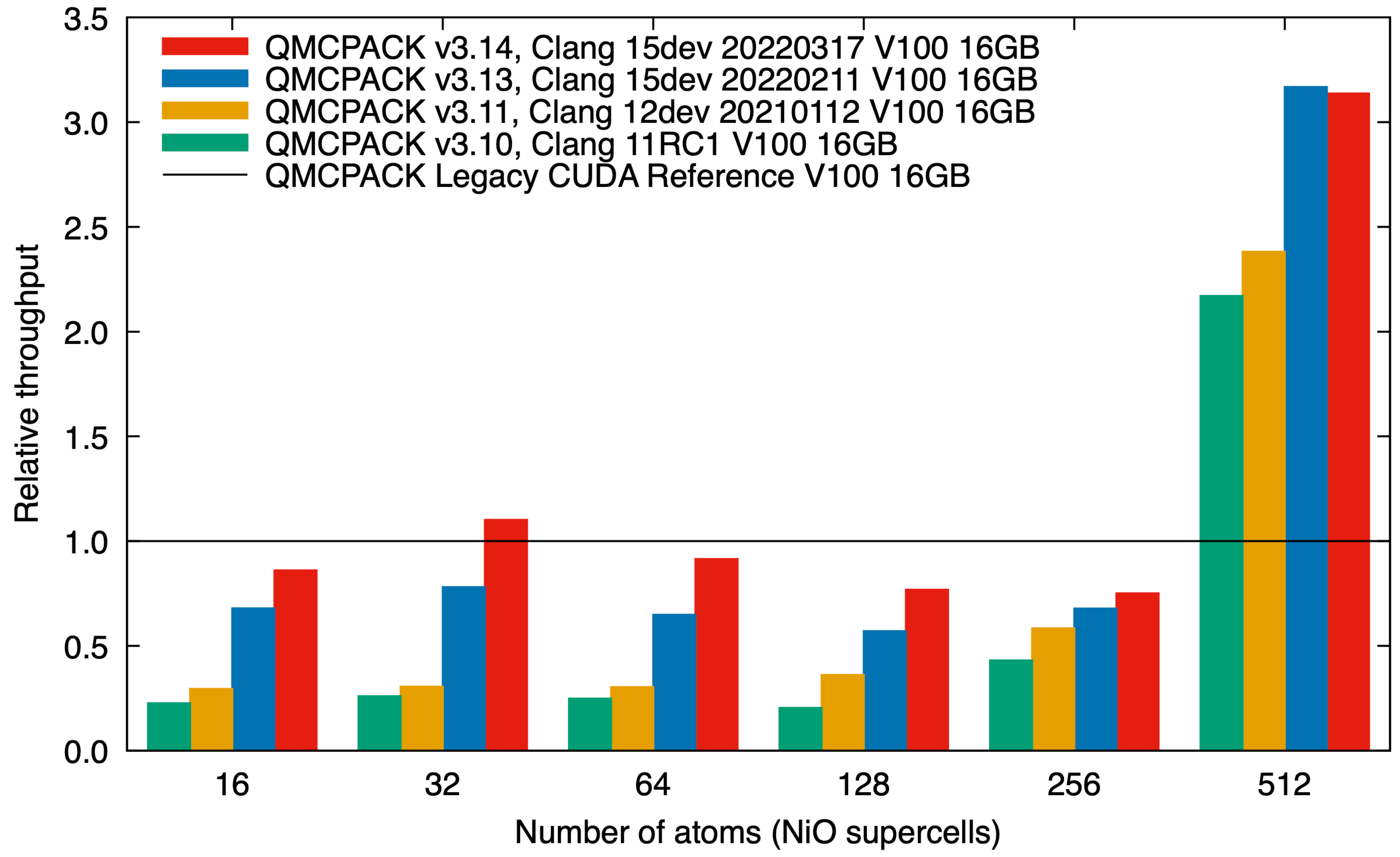
For QMCPACK problems, testing regimen minimizes chance of recurrence. Statistics let us (re)focus testing effort on most critical areas. For external problems, comprehensive testing **allows us to give prompt feedback to the relevant developers.**

Examples

- QMCPACK:
  - State machines associated with efficient Monte Carlo.
  - Handling of optional features via legacy #ifdefs requiring separate builds.
- Wider software stack:
  - Compilers, particularly OpenMP offload. Problems with complex reductions, multithreading... Latest releases of LLVM in production on NV.
  - Libraries. E.g. Numerically incorrect results from GPU dense linear algebra libraries, threading problem in CPU OpenBLAS (fixed promptly).
  - Many transient packaging and compatibility issues associated with specific versions of libraries, compilers, tools. Important to have latest versions promptly in spack, in addition to older versions.

Status

As of LLVM 15.0 (released 9/2022), performance portable version sufficiently close in performance to limited feature legacy GPU implementation for science production, but with full capabilities available.



OAK RIDGE National Laboratory

Ongoing challenges, open questions

- Further maturation of the ecosystem is necessary. Having Frontier, Aurora, Polaris (etc.) in production will help.
- Helpful to have “stable” and “leading edge” machines available for CI. It is not practical for every app team to run their own CI. Lack of access slows development velocity.
- Automated testing at facilities would help catch issues with vendor provided software, MPI, their unique environment etc.
- Can we obtain ~full performance with only OpenMP using newer/revised versions of the standard, or will some CUDA/HIP/SYCL still be required? What is needed in future C++ standards?

Conclusions

- Performance portable QMC is a challenge!
- Performance portable QMCPACK is in science production.
- Next challenges: taming memory usage, science features.
- Modern development practices, particularly testing, has improved code quality, enabled large changes, and increased our efficiency overall. These practices require dedicated resourcing and staff.

Questions, comments? kentpr@ornl.gov