SAND2021-4174C

Towards QMCPACK Performance Portability

P. Kent (kentpr@ornl.gov)[1], P. Doak[1], J. Krogel[1],
R. Clay[2], C. Melton[2], L. Shulenburger[2], A. Correa[3], M. Morales-Silva[3],
A. Benali[4], M. Dewing[4], Y. Luo[4], H. Shin[4], E. Briggs[5], W. Lu[5], J. Bernholc[5]

Accurate materials predictions

QMCPACK

Today we lack a practical quantum mechanical modeling approach applicable to general materials, including metals, as well as molecules, where all approximations are defined and systematically reducible. This is particularly problematic where today's approximate approaches are unreliable, such as in many transition metal element bearing materials. These include the metal oxides with applications ranging from rechargeable batteries to superconductors, as well as the large class of "quantum materials" currently pursued for novel phenomena.

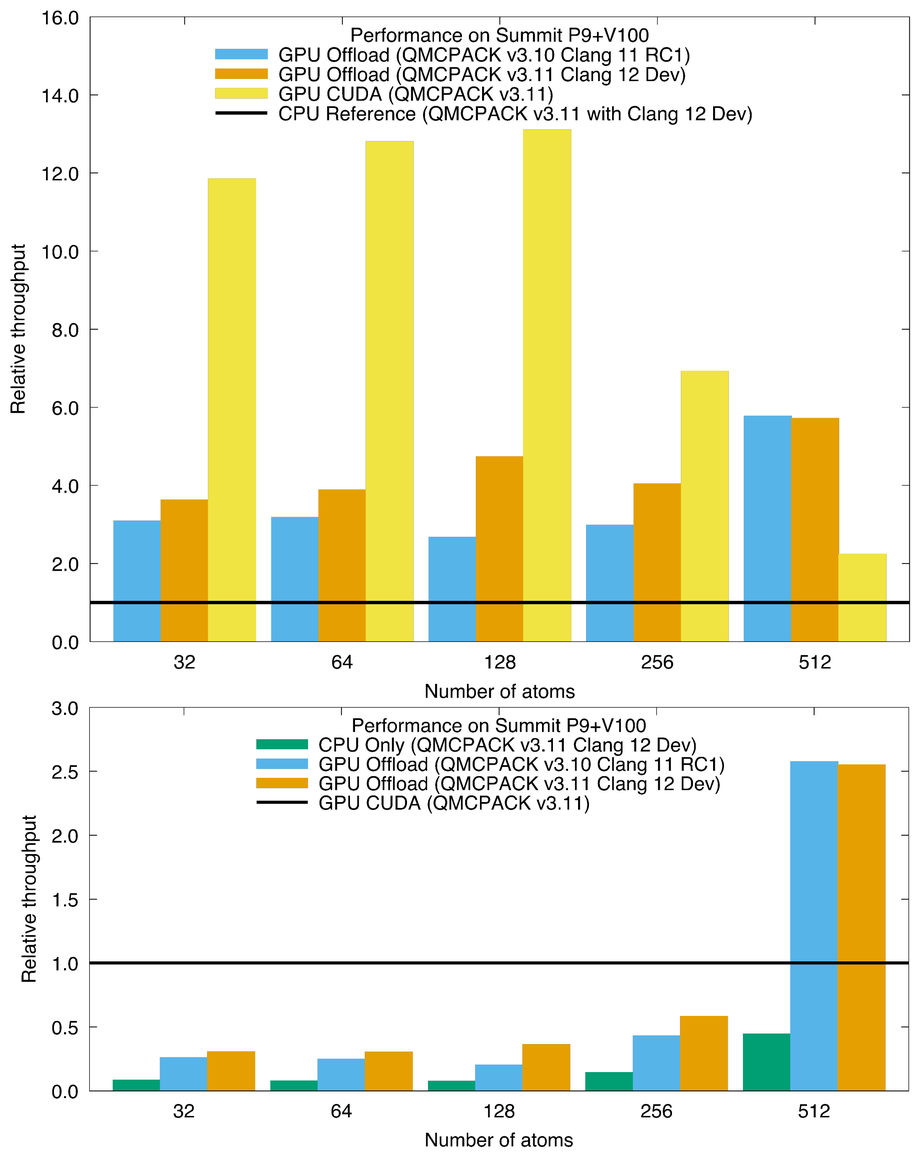
To provide such a capability, QMCPACK implements Quantum Monte Carlo (QMC) methods that solve the Schrödinger equation directly using statistical methods. Very few approximations are made and they are all potentially testable. System size scaling is favorable $O(N^2)$-$O(N^4)$, and there are ample opportunities for parallelism.

We aim to run everywhere: day 1 on Aurora, Frontier, and Perlmutter, through to workstation class CPU & GPUs - where users begin.

QMCPACK is openly developed at https://github.com/QMCPACK/ . Miniapps and key kernels have also been written or extracted to facilitate easier study and prototyping.

QMCPACK Performance with OpenMP Offload

Performance of our new batched offload implementation now delivers **sufficient acceleration for production science**, but it not yet optimal. To catch the legacy CUDA implementation, some additional functionality must be moved to the GPUs and data movement optimized. Once implemented we will broaden the feature set and optimize the performance on AMD, which already passes ~all tests.



MiniQMC OpenMP Target Offload Compiler Experience

Thanks to the ECP SOLLVE team, LLVM developers, and vendor compiler developers for major improvements since ECP Houston 2020!
E.g. Greater functional correctness and improved asynchronous execution support enabling usable performance.

Compiler	Clang 12.0.0rc3	AOMP 11.12-0	XL 16.1.1-5	OneAPI 2021.2.0	Cray 11.0.2	GCC 11dev 20210315	NVHPC 21.02
device	NVIDIA	AMD	NVIDIA	Intel	NVIDIA	NVIDIA	NVIDIA
math header conflict	Pass	Pass	Pass	Pass	Pass	Pass	Pass
complex arithmetic	Pass	Pass	Pass	Pass	Fail	Pass	Pass
math linker error	Pass	Pass	Pass	Pass	Pass	Pass	Fail
static linking	Fail	Pass	Pass	Pass	Pass	Pass	Pass
Async tasking	Pass	FC	Pass	FC	FC	FC	Fail
multiple streams	Pass	Pass	Pass	FC	FC	FC	Pass
check_spo	Pass	Pass	Pass	Pass(R)	Pass	Pass	Fail
check_spo_batched	Pass	Pass	Pass	Pass(R)	Pass	Pass	Fail
miniqmc_sync_move	Pass	Pass	Pass	Pass	Pass	Pass	Pass

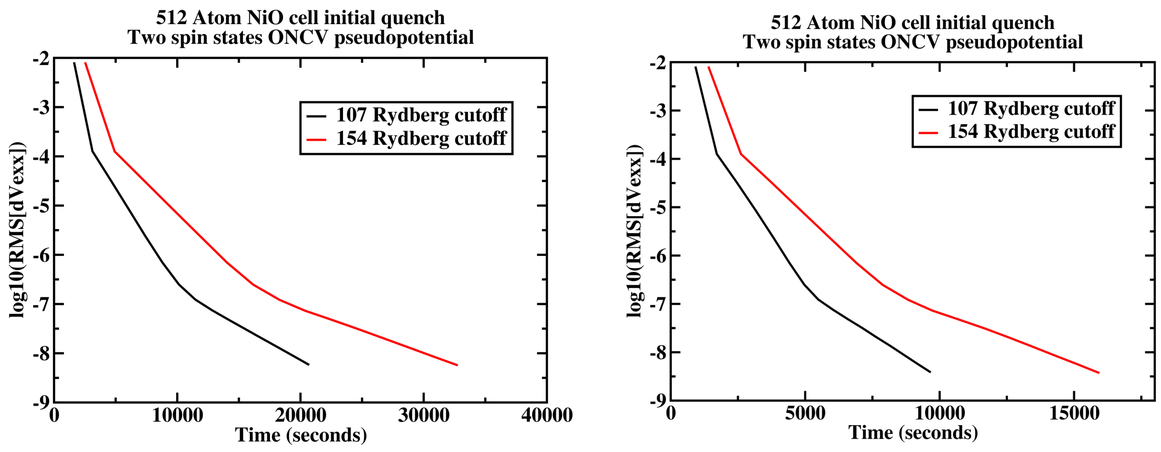
Fail: Crash in compilation, linking, execution, or incorrect results
FC: Functionally correct but implementation suboptimal. E.g. Synchronous execution allowed by standard but asynchronous execution required for sensible performance.
(R): Regression in latest release, but previously OK.
Pass: Compilation through execution, performance reasonable.

Data obtained from available public versions in March 2021. Additional details with links, build options etc. at https://github.com/ye-luo/qmcpack/wiki/OpenMP-offload

Please contact us with questions & suggestions!

RMG DFT

The Real-space MultiGrid Density Functional Theory code is a source of input trial wavefunctions for QMCPACK. We implemented explicit GPU memory management (vs relying on coherent memory) and made a HIP port of the existing CUDA implementation to AMD GPUs. Allowing for additional optimization, performance is significantly increased since 2020. The HIP version delivers significant acceleration even on older consumer Radeon VII cards and will be further optimized for Frontier.



Convergence of RMG for 512-atom NiO cell, as used in QMCPACK FOM, with hybrid-exchange functional (Left) with 2020 version with managed GPU memory and (Right) current version with explicit memory management and additional optimizations.

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.