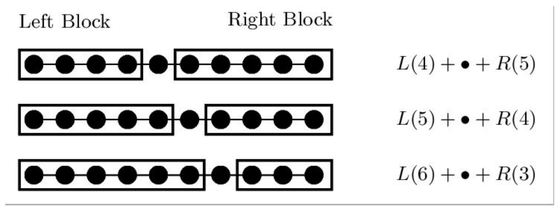
A Kronecker Product Implementation of Density Matrix Renormalization Group

W. Elwasif, A. Chatterjee, G. Alvarez, E. D'Azevedo
Oak Ridge National Laboratory

Background

- Density Matrix Renormalization Group (DMRG) is the preferred method for nanoscale modeling of strongly correlated materials such as superconductor, magnetic materials, and quantum dots
- One goal of DMRG is computation of the lowest eigen-vector of Hamiltonian (ground state) defined on a N-site lattice.
- However, Hubbard model on N-site lattice has vector space of size 4^N.
- DMRG is a systematic process to find a subspace that approximates well the eigen-vector. DMRG partitions the lattice into Left (Environment) and Right (System).
- The algorithm performs sweeps to grow Left to 4*M states and truncate Right to M states, then reverse sweep direction

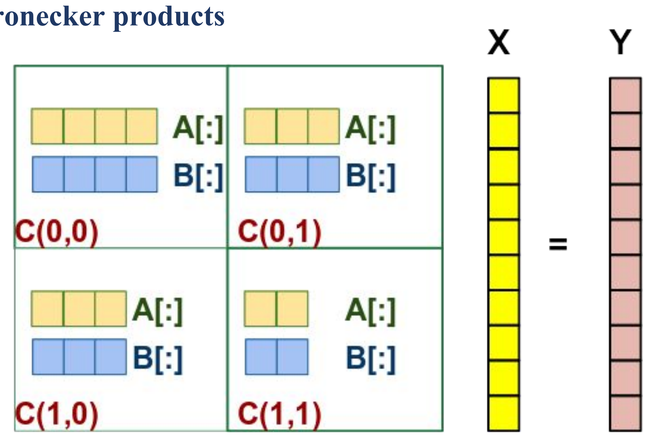


- The full Hamiltonian can be written as Kronecker products

$$H_{\text{full}} = H_L \otimes I_R + I_L \otimes H_R + \sum_{k=0}^{K} C_L^k \otimes C_R^k$$

Approach

- DMRG++ is a free and open source implementation of DMRG developed by ORNL
- One key computational kernel is matrix-vector multiplication of target Hamiltonian in Lanczos algorithm
- A mini-app of this key kernels is developed to explore different implementation approaches
- The target Hamiltonian is a large sparse matrix and computation commonly limited by memory bandwidth and memory capacity
- The admissible states can be grouped by quantum numbers to form 'patches'.
- Key idea is to organize computations by these patches so the target Hamiltonian matrix is expressed as sum of Kronecker products of smaller matrices
- **Significant savings in memory and work by exploiting Kronecker products**



Block partitioning of Hamiltonian matrix

Details

- The key computation can be viewed as computing Y = C * X
- Matrix C is block partition into Np by Np sub-matrices
- Block submatrix C[I,J] is sum of Kronecker products

$$C[I,J] = \sum_k A_{IJ}^{(k)} \otimes B_{IJ}^{(k)}$$

- Let $W_{IJ}^{(k)} = B_{IJ}^{(k)} * X[J]$

$$\begin{aligned} C[I,J] * X[J] &= \left(\sum_k^K A_{IJ}^{(k)} \otimes B_{IJ}^{(k)} \right) * X[J] \\ &= \sum_k^K (B_{IJ}^{(k)} * X[J] * (A_{IJ}^{(k)})^t) \\ &= \sum_k^K (W_{IJ}^{(k)} * (A_{IJ}^{(k)})^t) \end{aligned}$$

- Coalesce into combined batched GEMM operations

$$\begin{aligned} [W_{IJ}^1 | W_{IJ}^2 | \ldots | W_{IJ}^K] &= [B_{IJ}^1 | B_{IJ}^2 | \ldots | B_{IJ}^K] * X[J] \\ \mathcal{W}_{IJ} &= \mathcal{B}_{IJ} * X[J] \ . \end{aligned}$$

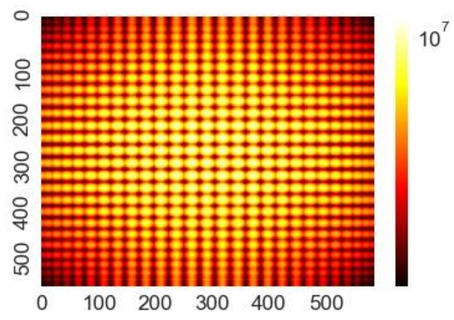
$$\begin{aligned} Z[I,J] &= C[I,J] * X[J] \\ &= [W_{IJ}^1 | W_{IJ}^2 | \ldots | W_{IJ}^K] * [A_{IJ}^1 | A_{IJ}^2 | \ldots | A_{IJ}^K]^t \\ &= \mathcal{W}_{IJ} * (\mathcal{A}_{IJ})^t \end{aligned}$$

- Similarly in batched GEMM in computing block rows Y[I]

$$\begin{aligned} Y[I] &= \sum_J Z[I,J] \\ &= \sum_J (\mathcal{W}_{IJ} * \mathcal{A}_{IJ}^t) \\ &= [\mathcal{W}_{I1} | \ldots | \mathcal{W}_{I,N_p}] * [\mathcal{A}_{I1} | \ldots | \mathcal{A}_{I,N_p}]^t \end{aligned}$$

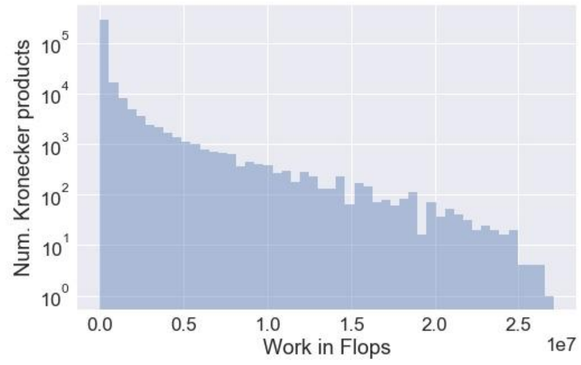
Performance

- Unified Managed Memory with MAGMA batched GEMM
- Titan V (V100) GPU with 12 Gbytes of device memory
- Intel Xeon E5-4640 (2.1 GHz) with 512 Gbytes
- 1st call to MAGMA has high overhead for data transfer of matrices to GPU
- Subsequent calls to MAGMA has higher performance
- A few large patches can cause significant imbalances in computational work load

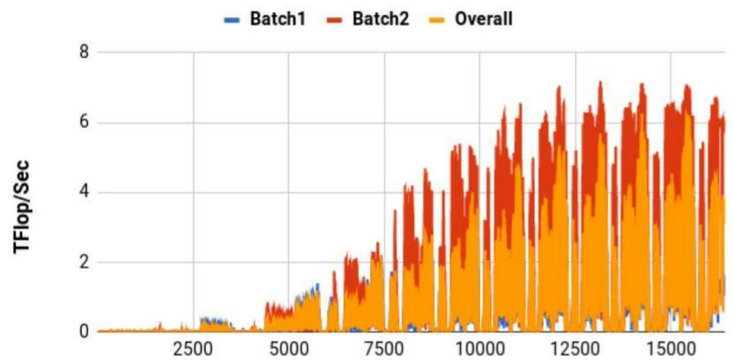


Heat map of distribution of work in blocked Hamiltonian matrix

Performance



Histogram of work intensity in Hamiltonian matrix



Batched GEMM performance (FP32) in DMRG++ can reach 6 Tflops/sec on GPU

Summary

- Matrix-vector multiplication of target Hamiltonian matrix in Lanczos algorithm is a key computational kernel
- Kronecker product formulation significantly faster compared to sparse matrix multiply
- Significant variations in work load across patches
- Batched GEMM achieves high performance on GPU

Acknowledgements

This material is based upon work supported by the U.S. DOE, Office of Science, BES, ASCR, SciDAC program. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Early development of this research effort was sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.