OAK RIDGE National Laboratory

Use case for Task Parallelism with Dynamic Resource Allocation

Arghya Chatterjee
(ORNL / Part time Ph.D. @ Georgia Tech)

Ying Wai Li and Oscar Hernandez (ORNL)

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Acknowledgements

- We would like to thank our co-authors /collaborators:

  Wael Elwasif (CSMD, ORNL)
  E. D'Azevedo (CNMS, ORNL)
  G. Alvarez (CNMS, ORNL)
  Vivek Sarkar (Georgia Tech)

- *This research used resources of the **O**ak Ridge **L**eadership **C**omputing **F**acility 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.*

- *This work was supported by the **Sci**entific **D**iscovery through **A**dvanced **C**omputing (sciDAC) program funded by U.S. DOE, Office of Science, Advanced Computing Scientific Computing Research (ASCR) and Basic Energy Sciences (BES), Division of Materials Science and Engineering.*

Introduction

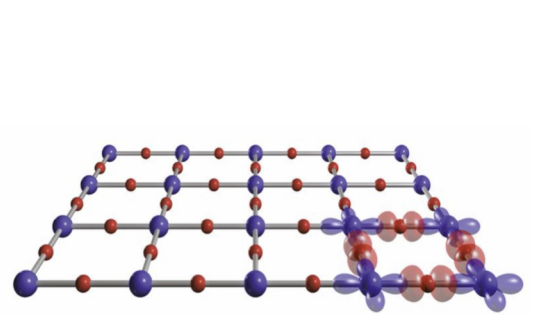
- Rapidly changing microprocessor design and heterogenous architectures
- Applications must adapt to exploit parallelism (e.g., DMRG++)
- Co—design, mini-application – serve as foundation for Exascale ready implementation
- Kronecker Product – Sparse matrix algebra (computational motif)
- Explore ways to program Summit (openPower) architecture using an incremental approach:
    - CPU Cores, followed by accelerators

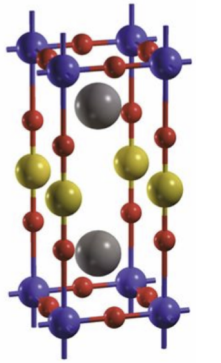
Application: Dense Matrix Renormalization Group (DMRG++)

- Algorithm: gain deeper understanding of nanoscale material properties
- Materials for superconductivity, power grid
- Ongoing effort: Material Applications Group / Computer Science Research Group

- DMRG++ today:
  - limited to 1-dimensional problems
  - uses **OpenMP** / **IBM ESSL** (for DGEMM) / **MPI**
  - Compiler: clang++ (ver. 4.0.0)

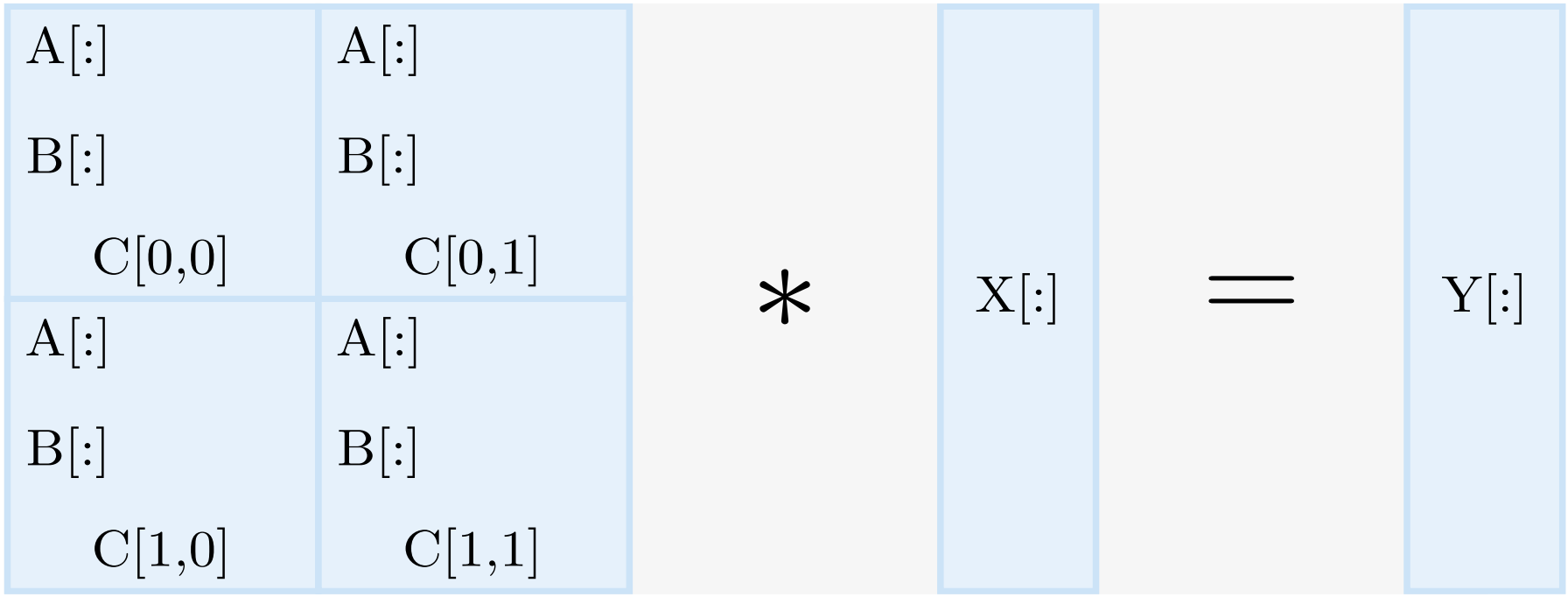
- DMRG++ future:
  - practical solutions for 2-D and 3-D problems
  - must use the GPUs (efficiently) -- most flops from GPUs


2-D model


3-D model

Application: Dense Matrix Renormalization Group (DMRG++)

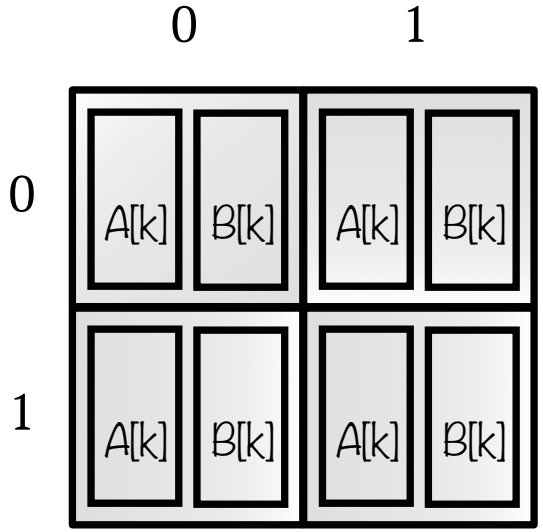
- Kronecker Product: Mini-application from DMRG++ (apply Hamiltonian)
- Y = H (Hamiltonian Matrix) * X [Computational Motif: **Dense** / **Sparse** GEMM]



- C[m][n] – vector A[k] and vector B[k]

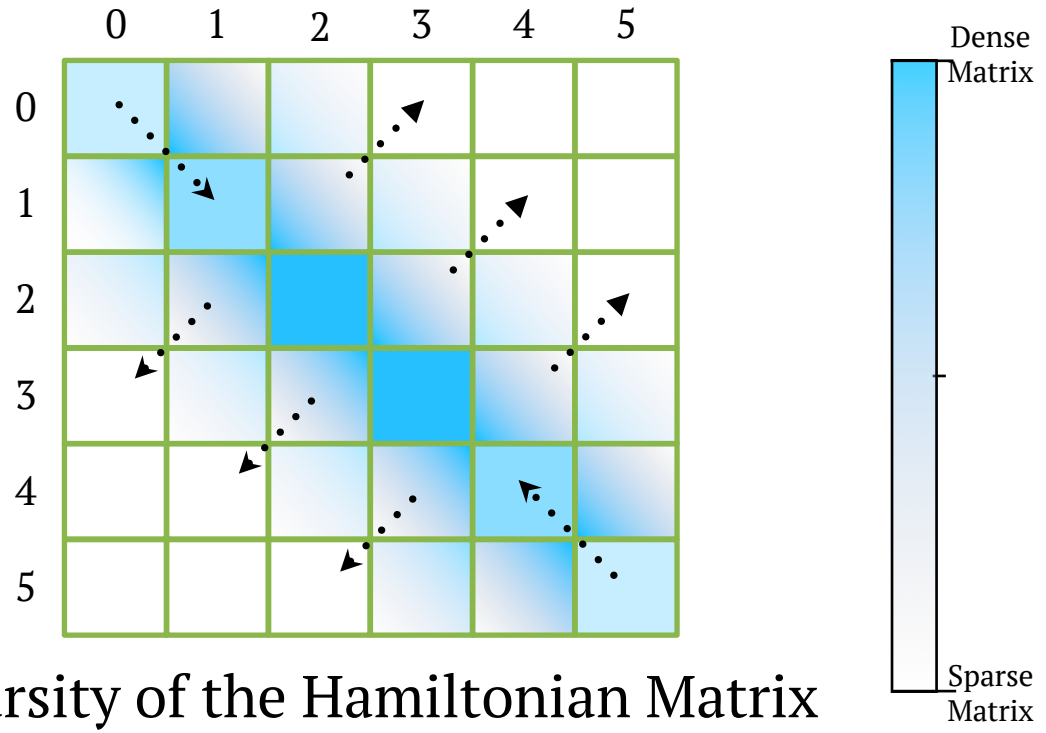
Application: Dense Matrix Renormalization Group (DMRG++)

- Hamiltonian Matrix: CIJ[:][:] – each element:: vector A[k] and vector B[k]
  – each element in the vector A's and B's is a vector of varying length



C [npatches : npatches]

Hamiltonian Matrix



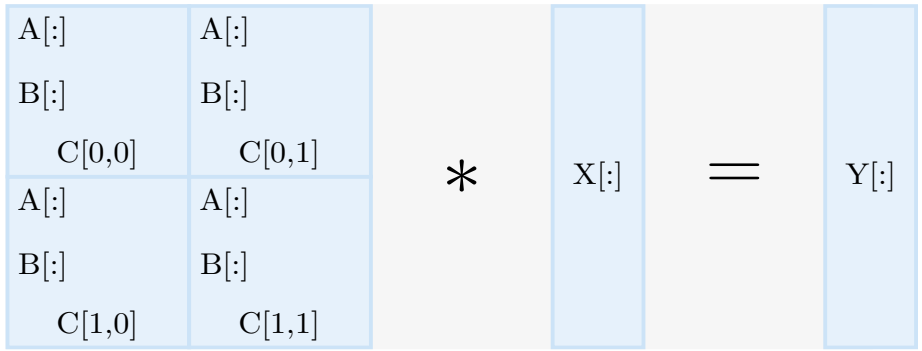
Sparsity of the Hamiltonian Matrix

- Data::
  - Clustered mostly principal diagonal
  - Density increases (towards center) / Sparsity increases (away from the diagonal)

Application: Dense Matrix Renormalization Group (DMRG++)

- Pseudo Code and Reduction::

```
1:  procedure HTarget(C[][], LPatch[], RPatch[], X[])
2:      NPatches ← Size(C)
3:      VSize ← PatchSize(LPatch, RPatch, NPatches)
4:      for i ← 1, C.rows do
5:          YI ← zeros(VSize[i])
6:          for j ← 1, C.cols do
7:              YIJ ← zeros(VSize[i])
8:              ElemInC ← Size(C[i][j])
9:              for k ← 1, ElemInC do
10:                 [MatA, MatB] ← GetMat(C[i][j], k)
11:                 YIJ[i] ← YIJ[i] + (MatA ⊗ MatB ∗ X[])
12:             end for
13:             for l ← 1, VSize[i] do
14:                 YI[l] ← YI[l] + YIJ[l]
15:             end for
16:         end for
17:         for m ← 1, VSize[i] do
18:             Y[m] ← YI[m]
19:         end for
20:     end for
21:     return Y
22: end procedure
```

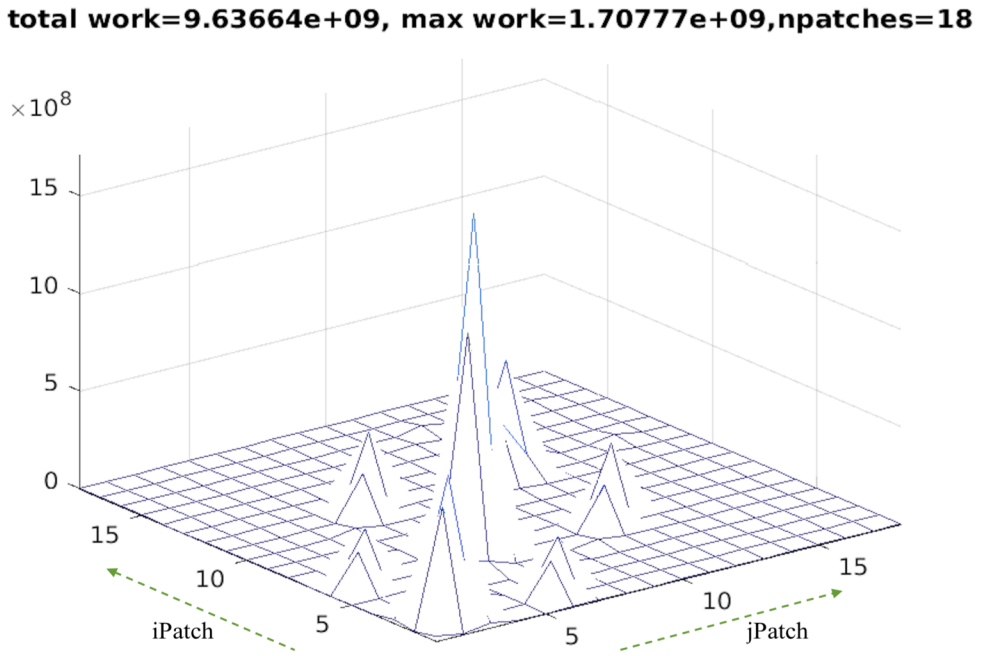


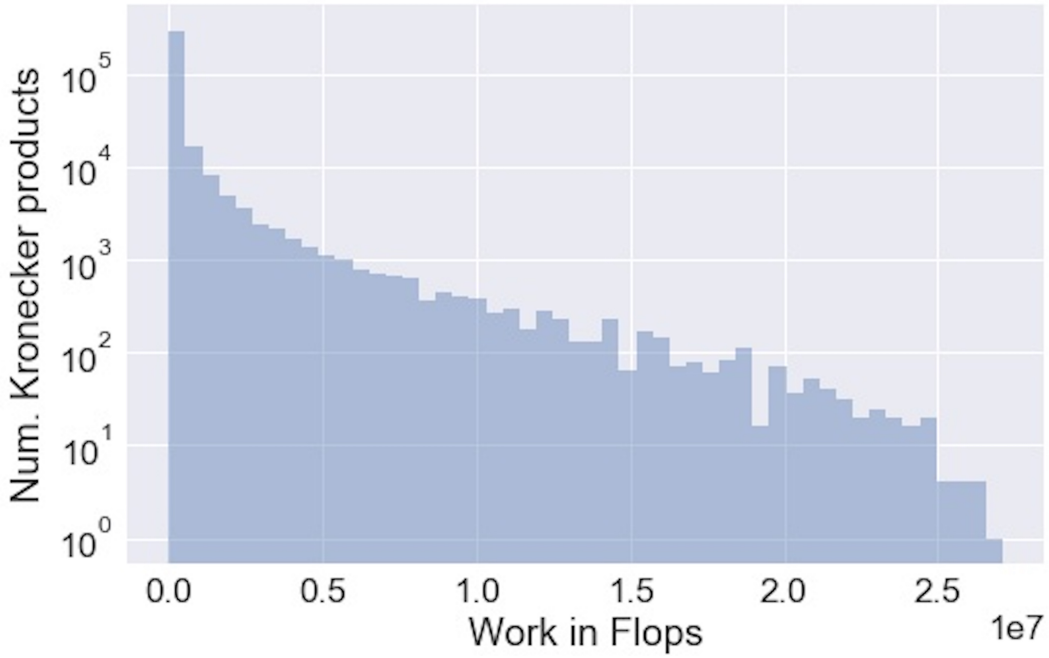
Reduction variables:: vector <double>

DGEMM operation: IBM ESSL (non-threaded)

DMRG++: Workload / Data layout (Load imbalance)

- Work distribution for the Hamiltonian Matrix



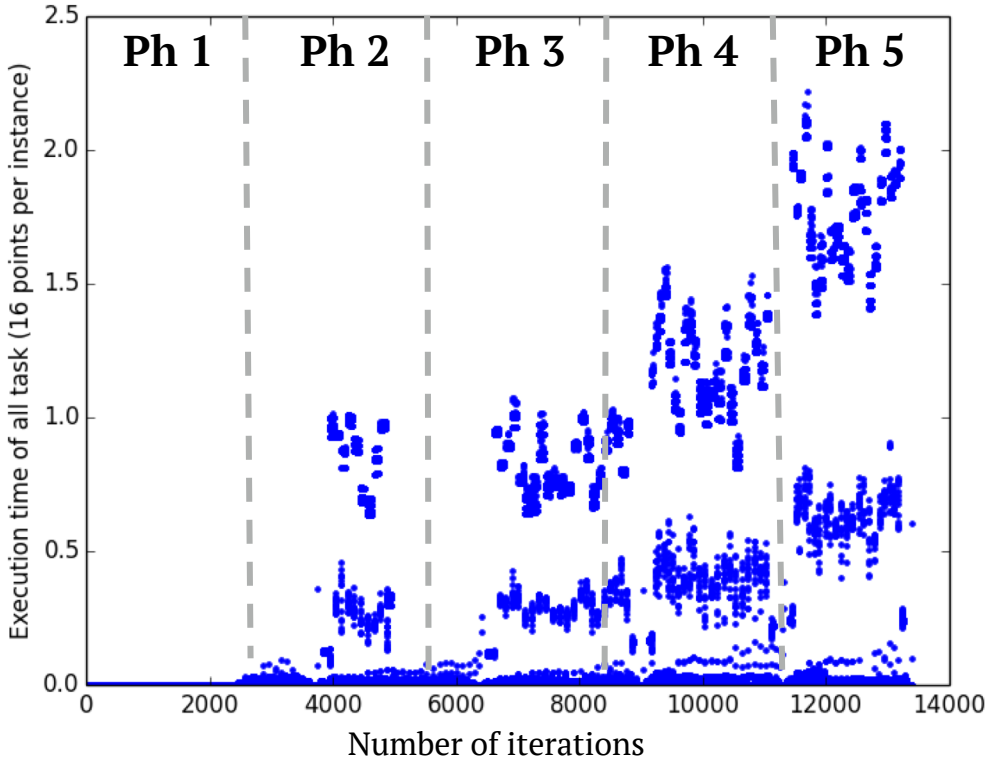


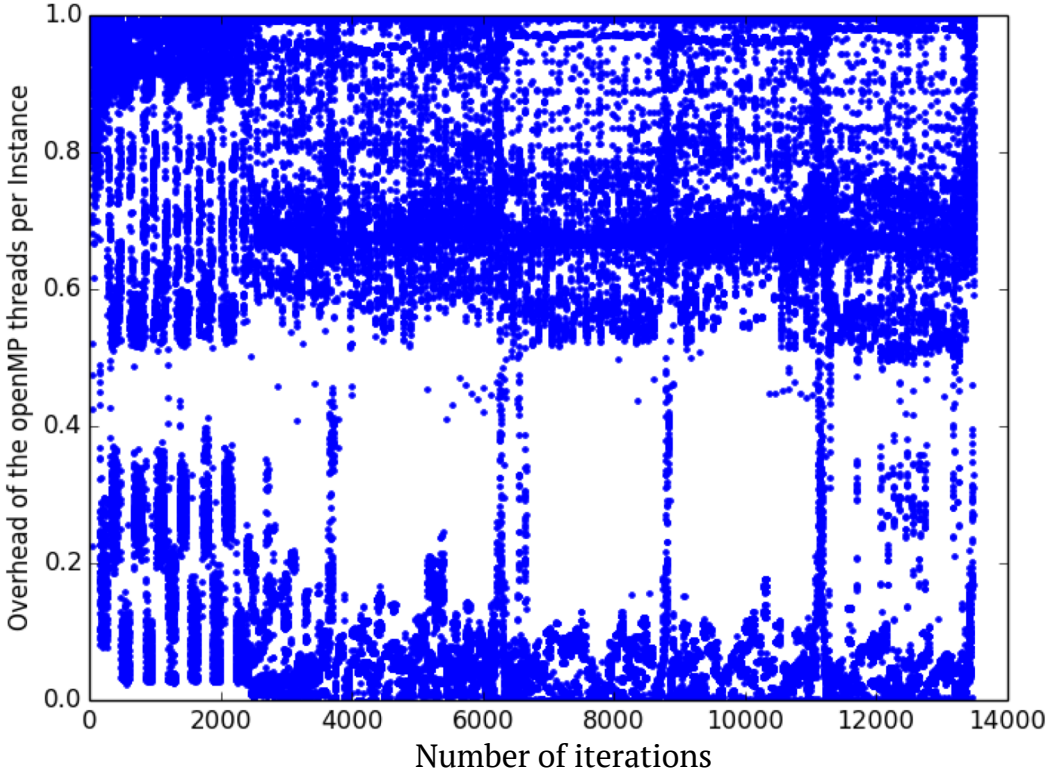
- 3-d line plot of the Hamiltonian Matrix
- x- and y-axes: matrix cells(C[i][j])
- z- unit of work available in each cell

- Distribution of work per patch (cell)
- Majority of patches: negligible work
- Few patches: bulk of the work

DMRG++: Load Imbalance over time

- DMRG++ runs in phases (5 phase till convergence)



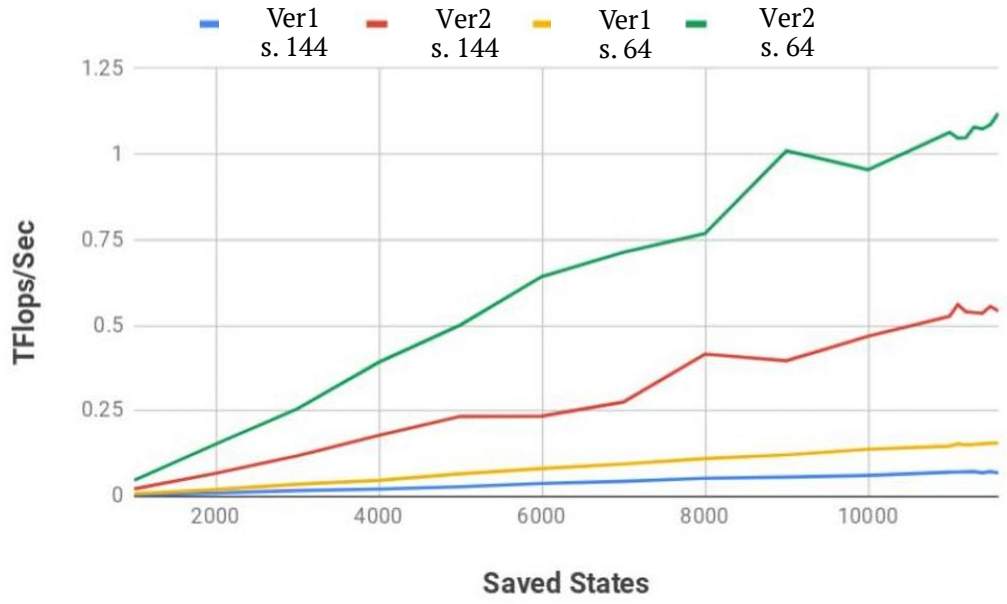


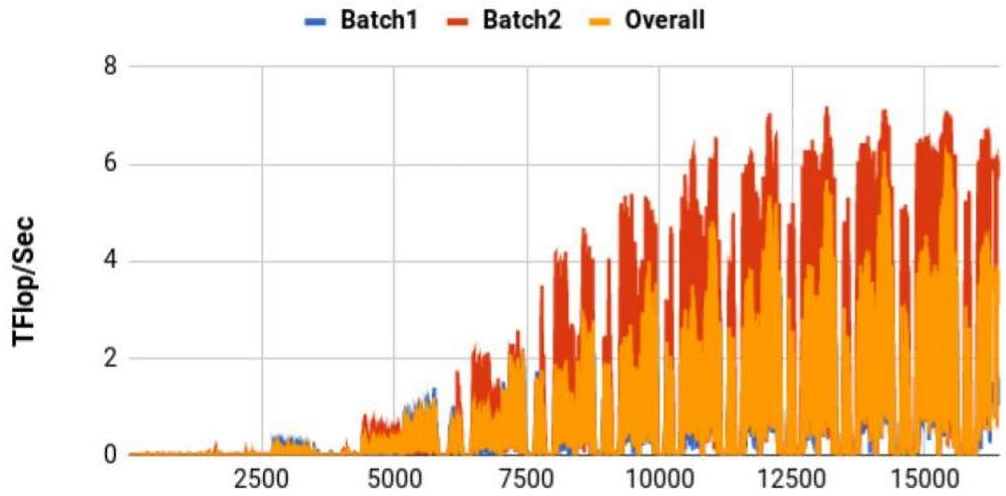
- Each phase: about ~2700 iterations
- Matrix size increases – after each phase
- 16 worker threads (on Titan)

- Overhead of using OpenMP: intra-node
- Overhead : [fork / join + barrier] – work
- High parallelism overhead – many small workloads

DMRG++: Kernel modified to use batched DGEMM

- Performance results on : Nvidia VOLTA Titan V GPGPU card with 12GB of memory connected via PCIe 3.0 bus.





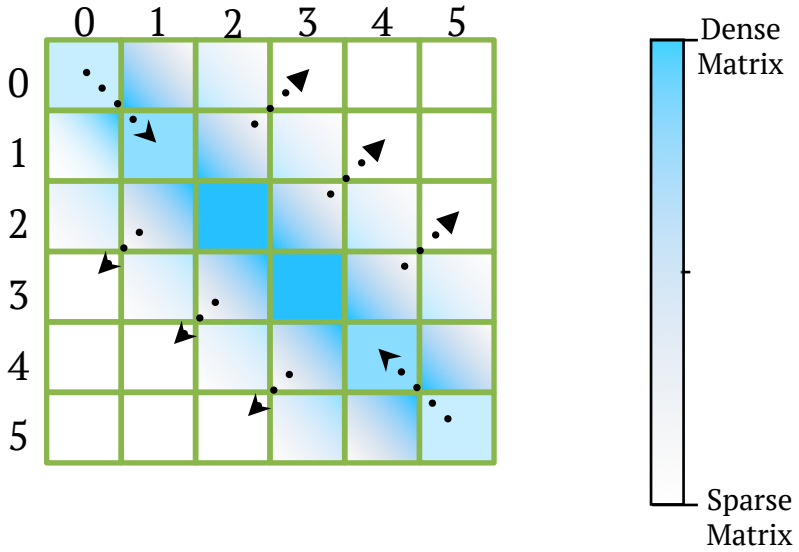
Number of iterations

- System size: 144 sites ; 64 sites
- Ver1: data movement from CPU to GPU
  Ver2: data already exist on the GPU
- Smaller system – **lower** sparsity ; **higher** throughput
- Larger system – **higher** sparsity ; **lower** throughput

- Kronecker multiplication with 100 sites
- More saved matrices over iteration space
- Two batches of GEMM Blas calls (use of CPU & GPGPU)
- Achieved up to 7.2 Tflops / sec (single precision)
  - 48% of peak performance 15 Tflop / sec

Use case for dynamic resource allocation / graphs

- Popular Pattern in a lot of Scientific Applications
  - Density Matrix Renormalization Group:: **Vector** Reductions



Load Imbalance (increases over execution time)

Task synchronizations (and task reductions)

Dynamic Resource allocation (scheduling)

Batched DGEMM operations (alternative, high productivity)

Use of HiHat & Cuda Graphs (CPU – GPU co-ordination)

Multiple GPU Reductions (same Vectors)

Inter GPU (on node) communication, without through CPU

DMRG: *1)* Wael Elwasif, E DÁzevedo, **Chatterjee, Arghya**, Gonzalo Alvarez, Oscar Hernandez, and Vivek Sarkar. ***Mini App For Density Matrix Renormalization Group Hamiltonian Application Kernel***. WRAp 2018: in conjunction with IEEE Cluster 2018.

2) **Chatterjee, Arghya**, Gonzalo Alvarez, E DÁzevedo, Wael Elwasif, Oscar Hernandez, and Vivek Sarkar. ***Porting DMRG++ Scientific Application to OpenPOWER***. IWOPH at ISC '18, 2018.

Use case for Task Parallelism with dynamic resource allocation

Arghya “Ronnie” Chatterjee

Post Master’s Research Associate, CSMD, ORNL

Part-time Ph.D. Student, Georgia Tech

chatterjeea@ornl.gov

Tuesday 21st, 2018