A Fast Selected Inversion Algorithm for Green's Function Calculation in Many-body Quantum Monte Carlo Simulations

Chengming Jiang, Zhaojun Bai, Richard Scalettar
University of California, Davis

The 30th IEEE International Parallel and Distributed Processing Symposium

Chicago, May 23-27, 2016

Outline

Outline

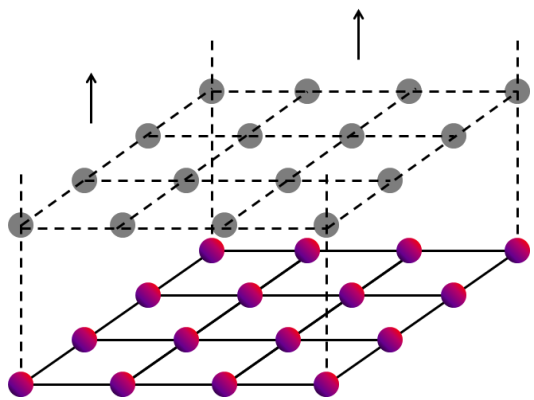
Hubbard model

- The Hubbard model[1] is defined by the Hamiltonian:

$$\mathcal{H} = \mathcal{H}_K + \mathcal{H}_\mu + \mathcal{H}_V$$

where $\mathcal{H}_K, \mathcal{H}_\mu$ and $\mathcal{H}_V$ stands for kinetic, chemical and potential energy, respectively.

- Electrons on discrete lattice sites:



[1] J. Hubbard, 1963.

DQMC, QUEST and Green's function

- Determinant Quantum Monte Carlo (DQMC) [2]
  - Simulation on Hubbard Hamiltonian
- QUantum Electron Simulation Toolbox (QUEST) [3]
  - A state-of-art implementation of DQMC simulations
- Green's function calculation
  - computational kernel of QUEST
  - inverses of **thousands** of Hubbard matrices
  - matrix dimension $NL \times NL$
  - $NL \approx 10^3 \cdot 10^2$

[2] R. Blankenbecler, D. Salapino, R. Sugar, 1981.

[3] http://quest.ucdavis.edu/

Outline

Green's function

- Green's function can be defined by the inverse of the following block **p-cyclic** matrix in the normal form

$$A = \begin{bmatrix} A_{11} & & & A_{1L} \\ A_{21} & A_{22} & & \\ & \ddots & \ddots & \\ & & A_{L,L-1} & A_{LL} \end{bmatrix},$$

where each block is $N \times N$ square and the diagonal block matrices A_{ii} for $1 \leq i \leq L$ are nonsingular.

Green's function

- Let $D = \text{diag}(A_{11}, A_{22}, \cdots, A_{LL})$, then

$$M = D^{-1}A = \begin{bmatrix} I & & & B_1 \\ -B_2 & I & & \\ & \ddots & \ddots & \\ & & -B_L & I \end{bmatrix},$$

where $B_1 = A_{11}^{-1}A_{1L}$ and $B_i = -A_{ii}^{-1}A_{i,i-1}$ for $2 \leq i \leq L$.

Green's function

- A block LU factorization of M is given by $M = LU$ where

$$L = \begin{bmatrix} I & & & & \\ -B_2 & I & & & \\ & -B_3 & I & & \\ & & \ddots & \ddots & \\ & & & -B_L & I \end{bmatrix}$$

and

$$U = \begin{bmatrix} I & & & B_1 \\ & I & & B_2 B_1 \\ & & \ddots & \vdots \\ & & I & B_{L-1} B_{L-2} \cdots B_1 \\ & & & I + B_L B_{L-1} \cdots B_1 \end{bmatrix}.$$

Green's function

- The inverses of L and U are given by

$$L^{-1} = \begin{bmatrix} I & & & \\ B_2 & I & & \\ B_3B_2 & B_3 & I & \\ \vdots & \vdots & \ddots & \ddots \\ B_L \cdots B_2 & B_L \cdots B_3 & \cdots & B_L \quad I \end{bmatrix}$$

and

$$U^{-1} = \begin{bmatrix} I & & & -B_1F \\ & I & & -B_2B_1F \\ & & \ddots & \vdots \\ & & I & -B_{L-1}B_{L-2}\cdots B_1F \\ & & & F \end{bmatrix},$$

where $F = (I + B_LB_{L-1}\cdots B_2B_1)^{-1}$.

Green's function

- Consequently, the inverse of M, denoted by G, is then given by

$$G = M^{-1} = U^{-1}L^{-1} = (G_{k\ell})$$

where for $1 \leq k, \ell \leq L$,

$$G_{k\ell} = W_{kk}^{-1} Z_{k\ell},$$

and

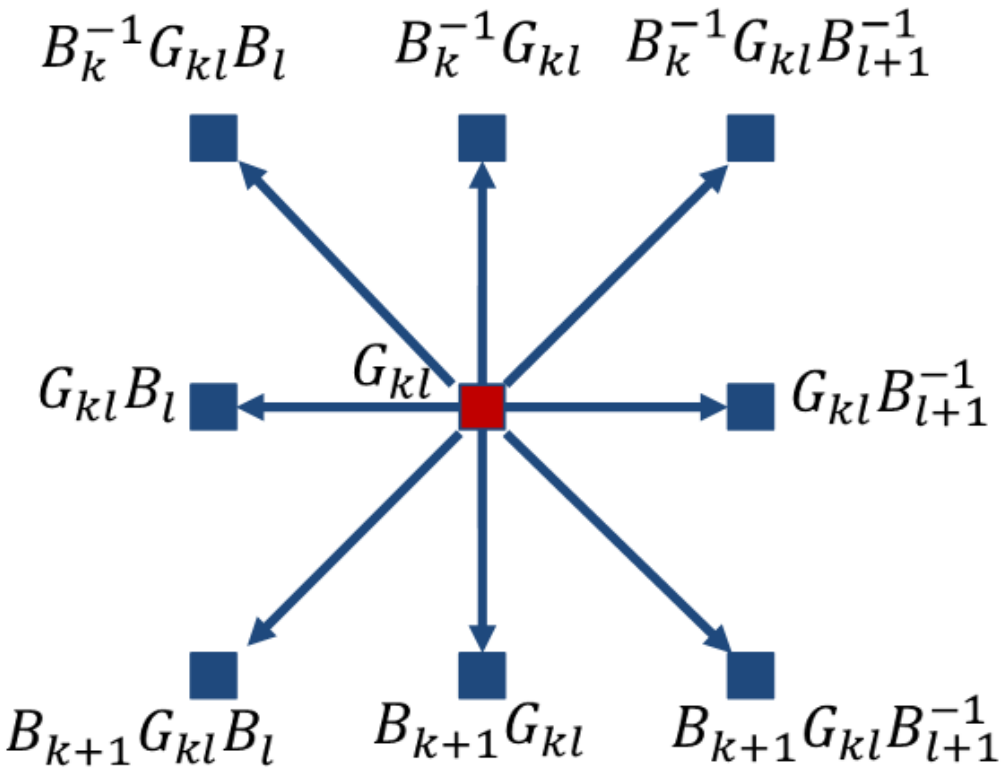
$$W_{kk} = \begin{cases} I + B_k B_{k-1} \cdots B_1 B_L \cdots B_{k+1}, & 1 \leq k \leq L-1 \\ I + B_L B_{L-1} \cdots B_1, & k = L \end{cases}$$

and

$$Z_{k\ell} = \begin{cases} -B_k B_{k-1} \cdots B_1 B_L B_{L-1} \cdots B_{\ell+1}, & k < \ell < L \\ -B_k B_{k-1} \cdots B_1, & k < \ell = L \\ I, & k = \ell \\ B_k B_{k-1} \cdots B_{\ell+1}, & k > \ell \end{cases}$$

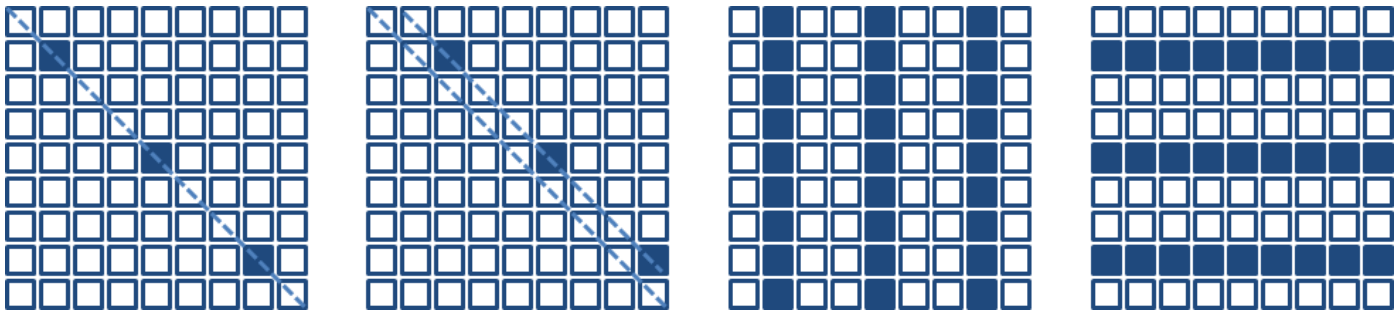
Green's function

- Relations between adjacent blocks of Green's function



Selected inversion

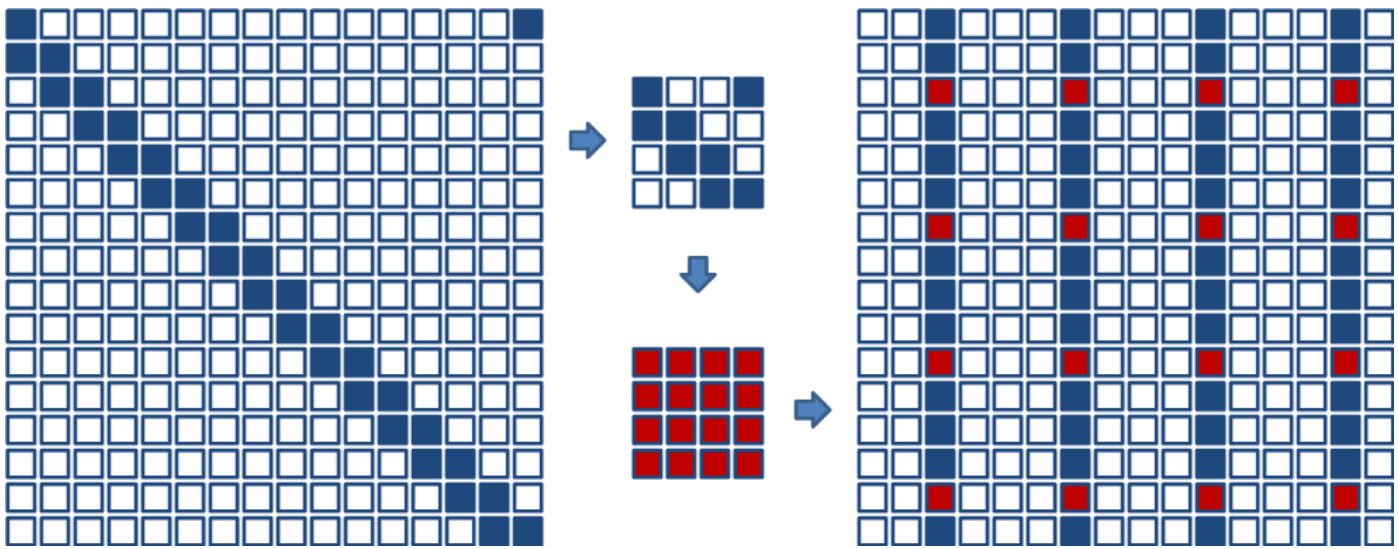
- Commonly selected patterns:



- Related works
  - Estimating trace of the matrix inverse [Stathopoulos, et al., 2013]
  - Subset of selected elements of the inverse matrix [L. Lin, et al., 2011]

FSI overview

- Fast selected inversion (FSI) algorithm:
  1. Clustering
     - block cyclic reduction (BCR)
  2. Inversion
     - block structured orthogonal factorization and inversion (BSOFI)[4]
  3. Wrapping
     - seeds + adjacency relations $\rightarrow$ selected inversion



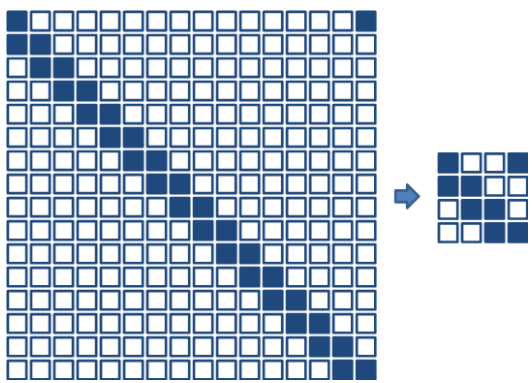
[4] S. Gogolenko, Z. Bai, R. Scalettar, 2014

Clustering

- $\widehat{M} = \mathrm{CLS}(M, c, q)$ is for a factor-of-c BCR of M, i.e.,

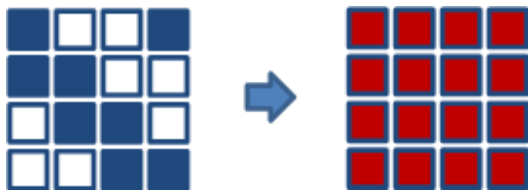
$$\widehat{M} = \begin{bmatrix} I & & & & \widehat{B}_1 \\ -\widehat{B}_2 & I & & & \\ & -\widehat{B}_3 & I & & \\ & & \ddots & \ddots & \\ & & & -\widehat{B}_b & I \end{bmatrix},$$

  where $\widehat{B}_i = B_j B_{j-1} \cdots B_{j-c+1}$ and $j = ci - q$.
- Computational cost: $2b(c-1)N^3$
- Embarrassingly parallel



Inversion

- $\widehat{G} = \widehat{M}^{-1} = (\widehat{G}_{ij})$ by BSOFI
  - Block structured orthogonal factorization and inversion
  - QR decomposition only on the $2N \times N$ dense blocks
  - Numerically stable
  - Lower computational complexity ($7b^2N^3$) than the inversion by full QR ($2b^3N^3$).

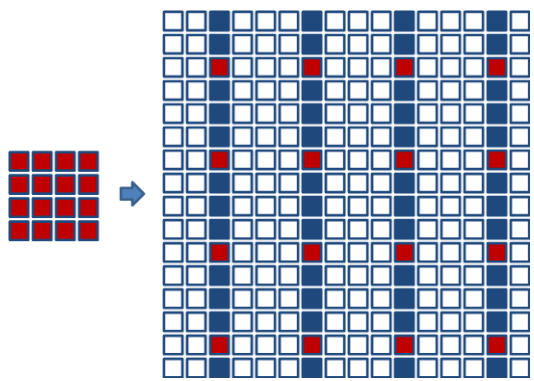


Wrapping

- A crucial observation:

$$\widehat{G}_{i,j} = G_{ci-q,cj-q} \equiv G_{k\ell} \quad \text{for } 1 \leq i, j \leq b.$$

- $G_{k\ell}$ + adjacency relations $\to \mathcal{S}$
  - Selected columns: $G_{k\ell} \rightsquigarrow G_{k+1,\ell} \rightsquigarrow G_{k+2,\ell} \rightsquigarrow \ldots$
  - Selected rows: $G_{k\ell} \rightsquigarrow G_{k,\ell+1} \rightsquigarrow G_{k,\ell+2} \rightsquigarrow \ldots$
- Computational cost: $3b(L-b)N^3$
- Embarassingly parallel



Advantages of FSI

- Computational cost (b selected block columns)

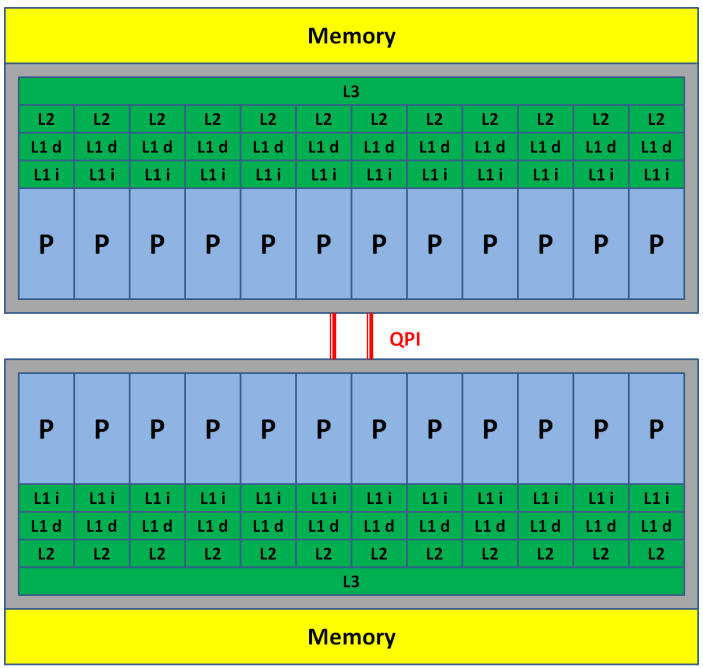
	complexity	FSI reduced factor
LU	$O(L^3N^3)$	L^2/b
Explicit form	$O(bL^2N^3)$	L
BSOFI	$O(L^2N^3)$	L/b
FSI	$O(bLN^3)$	1

- Memory requirement
  - Full inversion methods like LU or BSOFI are not feasible.
- Stability
  - FSI is more numerically stable than explicit form.
- Parallelism
  - FSI is embarrassingly parallel.

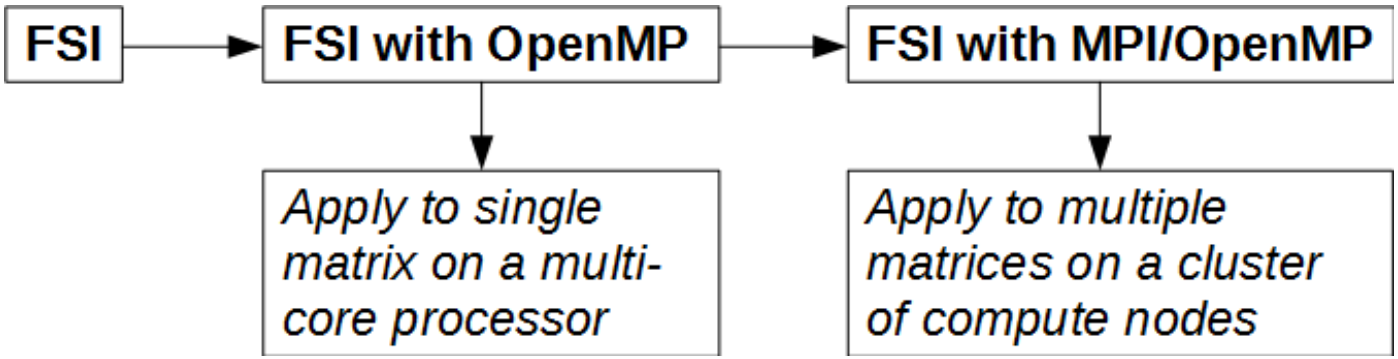
Outline

Architecture

- Hierarchical architecture
  - Multiple nodes
  - Multiple sockets
  - Multiple cores
- NERSC's supercomputer Edison
  - 5576 compute nodes
  - 2 sockets per node
  - 12 cores per socket (133824 cores in total)
- A cray X-30 dual-socket node



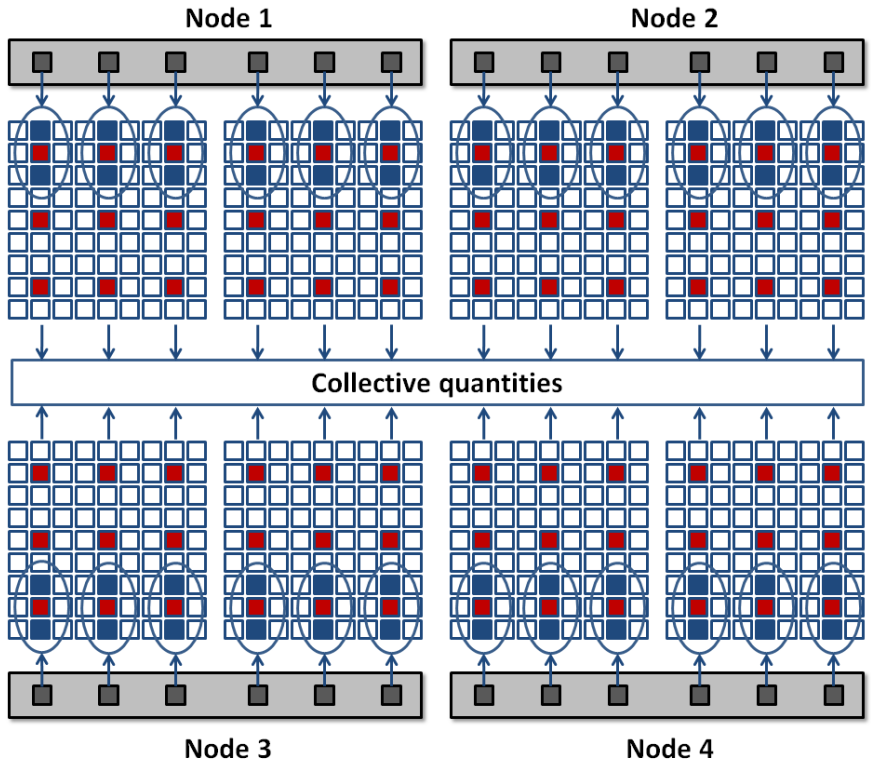
Two levels of parallelism



- OpenMP level
  - Clustering: compute $\widehat{B}_i$ for $1 \leq i \leq b$ in parallel by OpenMP;
  - Inversion: run BSOFI by using multi-threaded MKL routine;
  - Wrapping: computes the neighbors of $G_{k\ell}$ for $1 \leq k, \ell \leq b$ in parallel by OpenMP.
- MPI level
  - Generate and distribute all the Hubbard matrices by MPI processes;
  - Each MPI process runs the FSI with OpenMP.

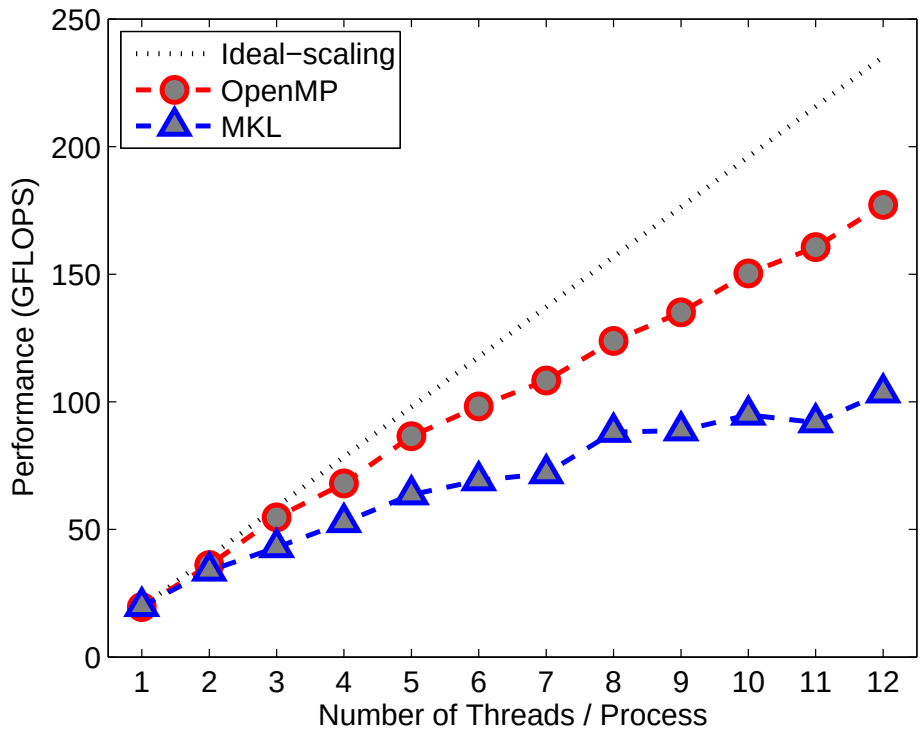
Parallel application of FSI

- A pictorial illustration:



FSI performance

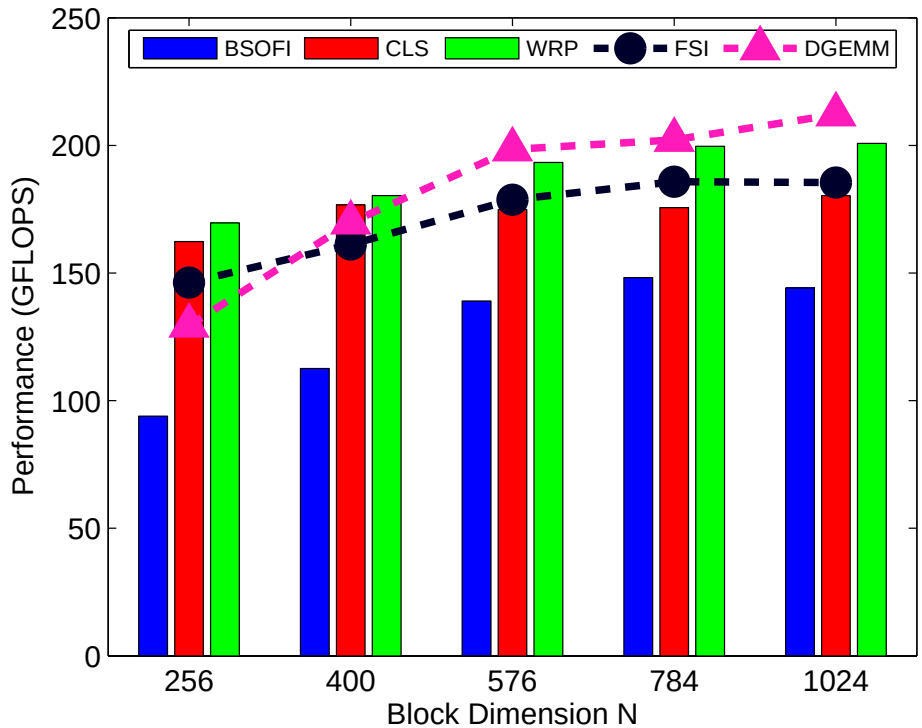
- FSI with OpenMP on a single 12-core Intel "Ivy Bridge" processor;
- $(N, L, c) = (576, 100, 10)$;
- Selected inversion of $b = L/c = 10$ block columns;



- 80% improvement.

FSI performance

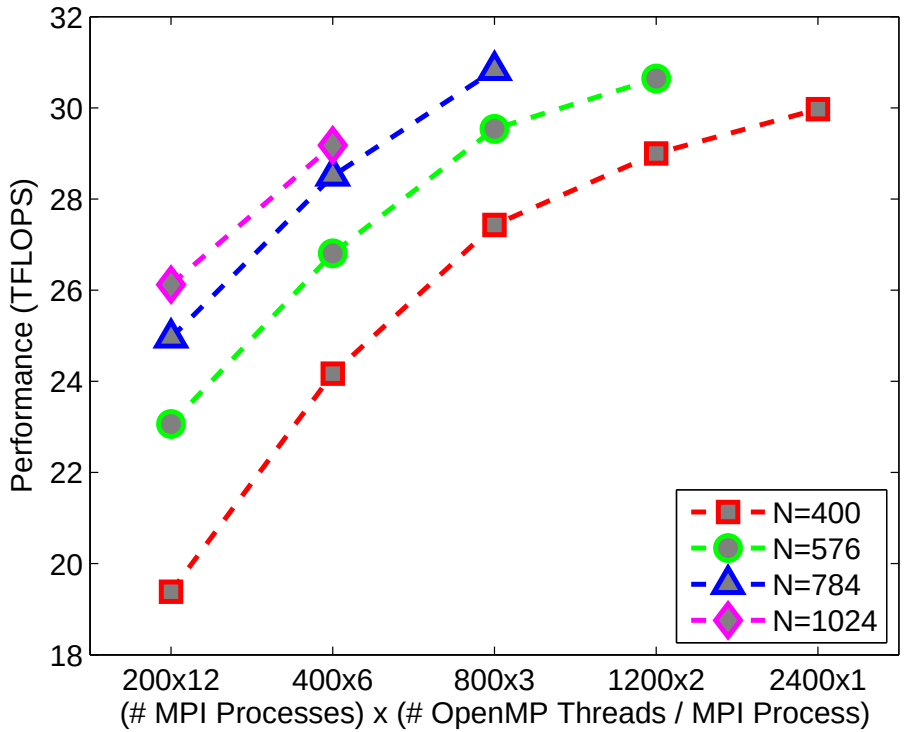
- FSI with OpenMP on a single 12-core Intel "Ivy Bridge" processor;
- $(L, c) = (100, 10)$;
- Selected inversion of $b = L/c = 10$ block columns;



- Close to MKL BLAS-3 `DGEMM`.

FSI performance

- FSI with OpenMP/MPI on 100 dual-socket Edison compute nodes (2400 cores):
- $(L, c) = (100, 10)$;



- Pure MPI execution is restricted due to the memory capacity;
- Hybrid implementation achieves best performance for large scale matrices.

Outline

DQMC overview

Algorithm 1 DQMC simulation

initialize HS configuration $h_0 = (h_{\ell i}) = (\pm 1)$
% Warmup stage
for i=1,...,w **do**
  DQMC sweep
end for
% Measurement stage
for i=1,...,m **do**
  DQMC sweep
  compute Green's function and physical measurements
end for

DQMC sweep

Algorithm 2 DQMC sweep

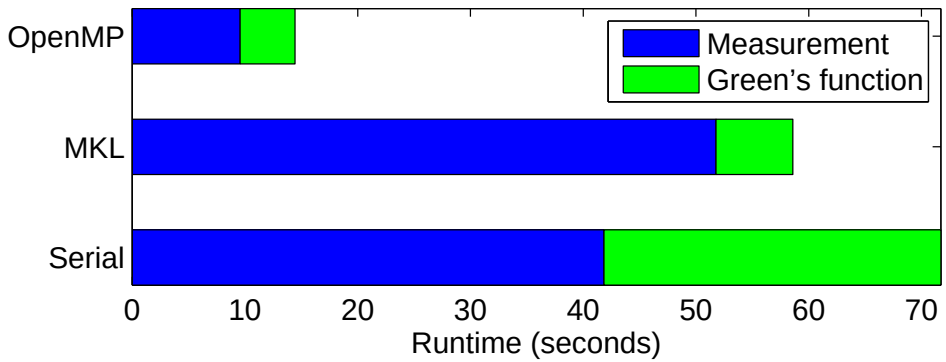
for $\ell = 1, 2, ..., L$ **do**
  for $i = 1, 2, ..., N$ **do**
    (1) Propose a new configuration: $h'_{\ell i} = -h_{\ell i}$;
    (2) Compute the Metropolis ratio:

$$r_{\ell i} = \frac{\det[M_+(h')]\det[M_-(h')]}{\det[M_+(h)]\det[M_-(h)]};$$

    (3) Apply Metropolis acceptance-rejection:
    randomize $r \sim$ uniform$[0, 1]$,
    if $r \leq \min\{1, r_{\ell i}\}$ **then**
      $h = h'$.
    end if
  end for
end for

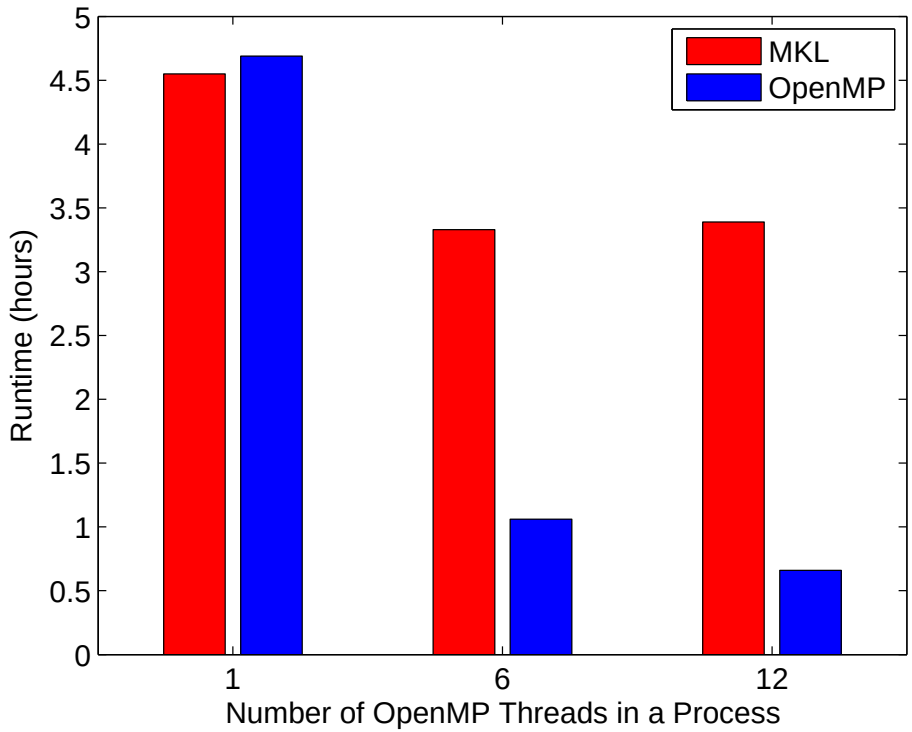
FSI in DQMC

- Runtime profile on a single Hubbard matrix with $(L, N, c) = (100, 400, 10)$;
- All the diagonal blocks, b block rows and b block columns of each G are computed.



FSI in DQMC

- Runtime of a full DQMC simulation on an "Ivy Bridge" processor of Edision;
- $(w, m) = (100, 200)$;
- $(L, N, c) = (100, 400, 10)$;



- FSI with MKL only gains a factor of 1.3 speedup;
- FSI with OpenMP gains a factor of 6.9 speedup.

Outline

Concluding Remarks

Conclusion:

- Parallel FSI enhances QMC capabilities.
- Solutions of problems that require larger number of electrons will be allowed.
- More complicated types of interactions can be studied.

Future work:

- Extension of FSI to other types of structured matrices.
- GPU implementation of FSI.
- Hybrid massive parallelization of the full DQMC simulation.

Acknowledgment

- Sergiy Gogolenko (Donetsk National Technical University), Chia-Chen Chang (University of California, Davis).
- This research used resources of the NERSC, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.
- CJ and ZB were supported in part by NSF grant CCF-1527091.
- RTS was supported in part by DOE grant DE-NA0002908.