

# 大規模並列化角転送行列繰り込み群を用いた 古典スピン系の臨界現象解析

Icosahedron model: HU, Okunishi, Krcmar, Gendiar, Yunoki and Nishino, Phys. Rev. E **96**, 062112 (2017).

Dodecahedron model: HU, Okunishi, Yunoki and Nishino, in preparation.

---

上田宏 (理研R-CCS)

## 問題点と求められる仕様



### ハミルトニアンの直接的取り扱い

メリット：数值的に厳密

デメリット：計算コスト  $\mathcal{O}(\exp(N))$

「京」でも最大 48 qubits

Comput. Phys. Commun. 237, 47 (2019)

### モンテカルロ法

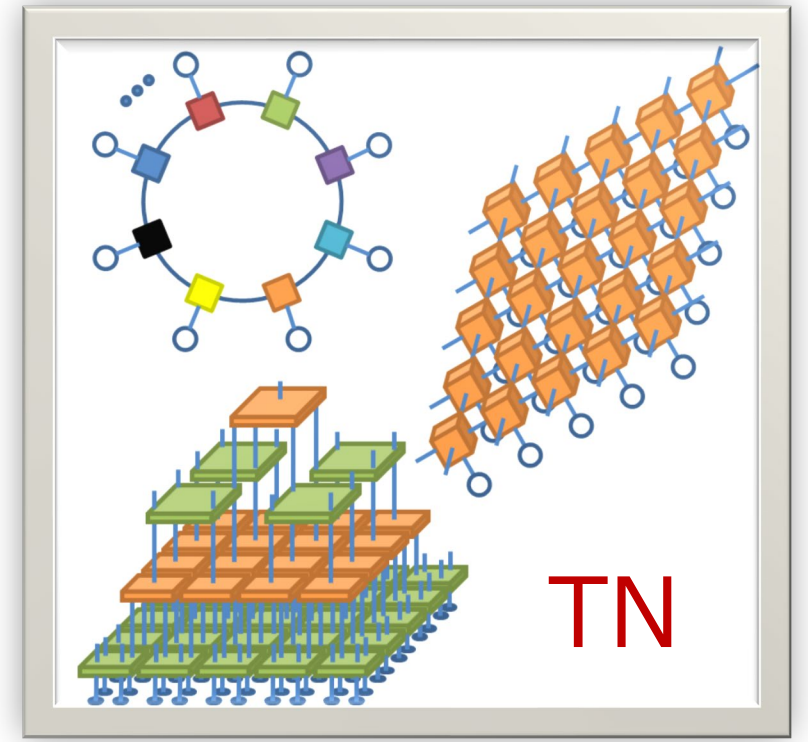
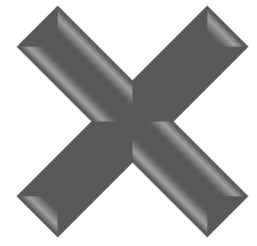
メリット：計算コスト  $\mathcal{O}(\text{poly}(N))$

デメリット：負符号問題

### 要求される仕様

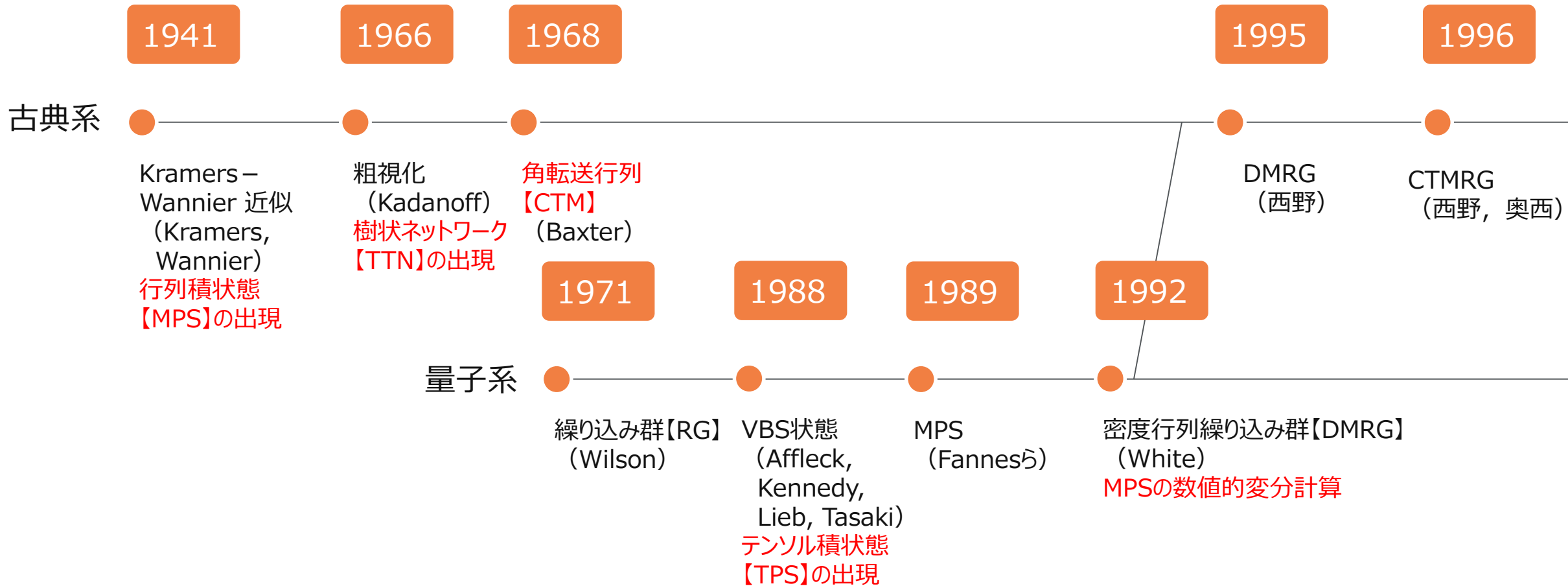
- ・ 計算コスト  $\mathcal{O}(\text{poly}(N))$  & どのような系にも適用可能
- ・ 系統的に精度を改善でき、負符号問題にさらされない

# 解決策の1つ：テンソルネットワークの導入

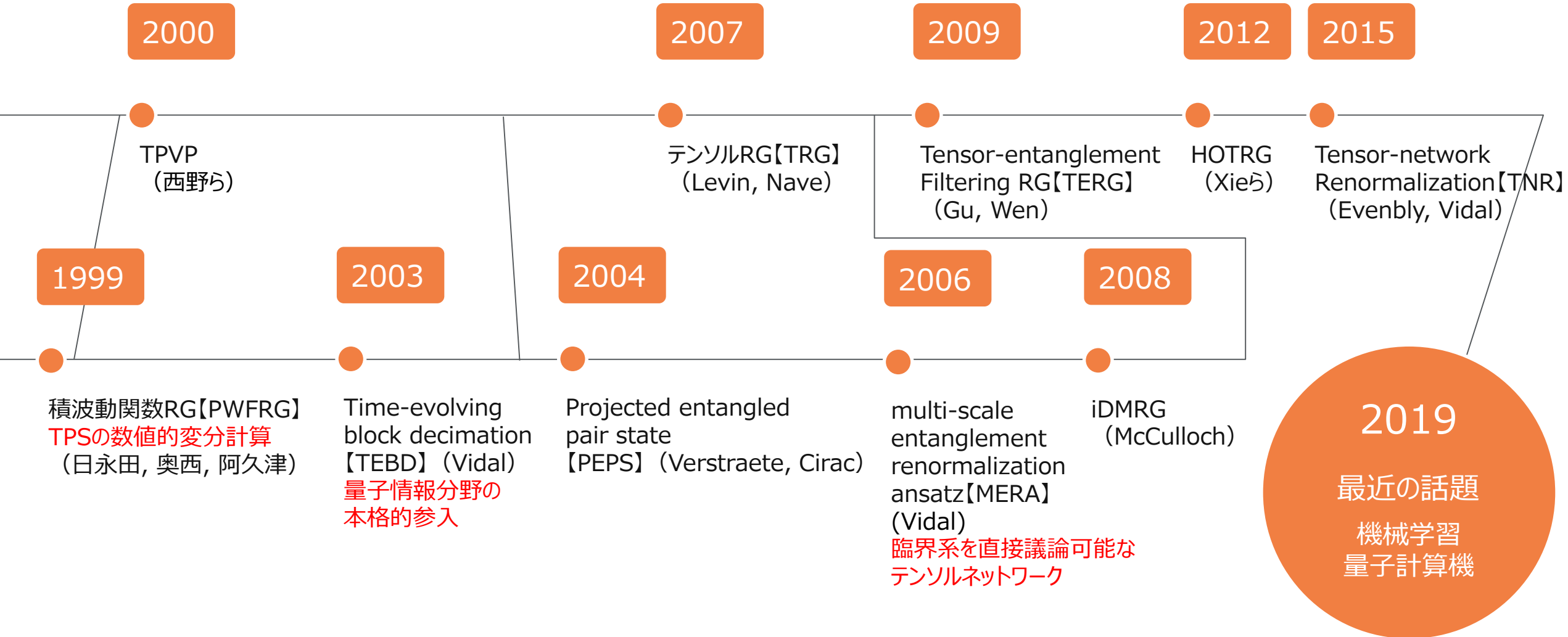


• TN (テンソルネットワーク) (物性／統計物理など)  $\equiv$  テンソルの低ランク近似 (応用数理) + エンタングルメントの考察 (量子情報)

# 行列積／テンソルネットワーク形式の歴史（その1）



# 行列積／テンソルネットワーク形式の歴史（その2）



# 古典 2 次元模型の分配関数計算

$$w_{ijkl} = l \begin{array}{c} i \\ \diamond W \\ k \\ j \end{array}$$

イジング模型の場合

$$i, j, k, l \in \{1, -1\}$$

$$E_{ijkl} = -(ij + jk + kl + li)$$

$$w_{ijkl} = \exp(-E_{ijkl}/k_B T)$$

分配関数 :

$$Z = \sum_{ijkl} w_{ijkl} =$$

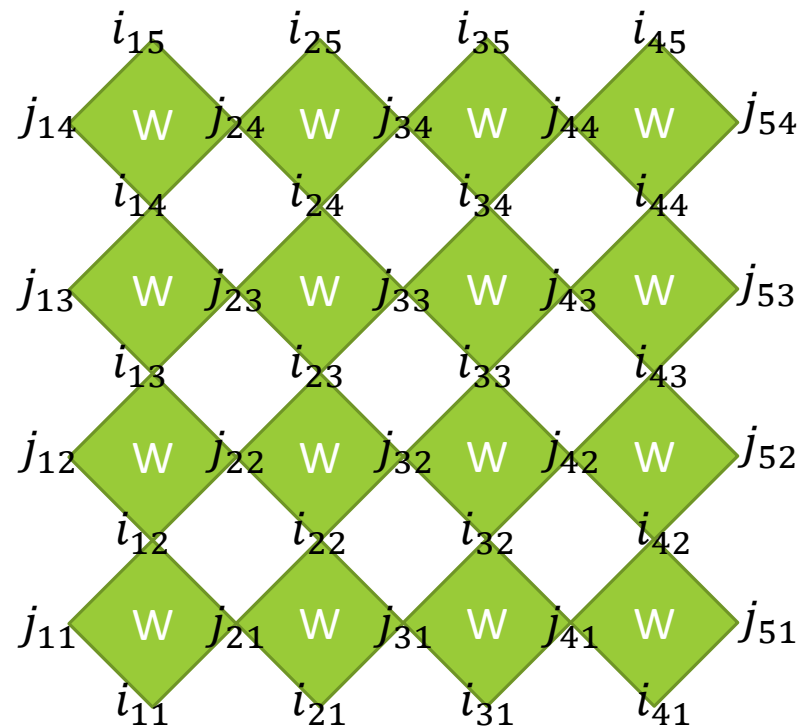


演算コスト :  $O(2^4)$

メモリコスト :  $O(2^4)$

$L = 4$  の時 :

$w_{ij} =$



$$Z = \sum_{ij} w_{ij}$$

愚直な実装 : 演算コスト :  $O(2^{2L(L+1)}) = O(2^{40})$

メモリコスト :  $O(2^4)$

# 角転送行列の導入

$$c_{ij}^{(1)} = \begin{array}{c} i \\ \diamond W \\ j \end{array}, \quad p_{ijk}^{(1)} = \begin{array}{c} i \\ \diamond W \\ k \end{array} j$$

演算コスト:  $O(2^4)$ , メモリコスト:  $O(2^4)$

$$r_{ij}^{(1)} = \begin{array}{c} i_{12} \quad i_{22} \\ \diamond W \quad \diamond W \\ j_{31} \end{array}$$

演算コスト:  $O(2^4)$ , メモリコスト:  $O(2^3)$

$$\ell_{ij}^{(1)} = \begin{array}{c} i_{13} \\ \diamond W \quad j_{22} \\ \diamond W \quad \diamond W \\ j_{31} \end{array} \quad \begin{array}{l} \text{演算コスト: } O(2^5), \\ \text{メモリコスト: } O(2^4) \end{array}$$

$$c_{ij}^{(2)} = \begin{array}{c} i_{13} \quad i_{23} \\ \diamond W \quad \diamond W \quad j_{32} \\ \diamond W \quad \diamond W \quad j_{31} \end{array}$$

演算コスト:  $O(2^6)$ , メモリコスト:  $O(2^4)$

$$\Psi_i^{(4)} = \begin{array}{c} i_{13} \quad i_{23} \quad i_{33} \quad i_{43} \\ \diamond W \quad \diamond W \quad \diamond W \quad \diamond W \\ \diamond W \quad \diamond W \quad \diamond W \quad \diamond W \end{array}$$

演算コスト:  $O(2^6)$ , メモリコスト:  $O(2^4)$

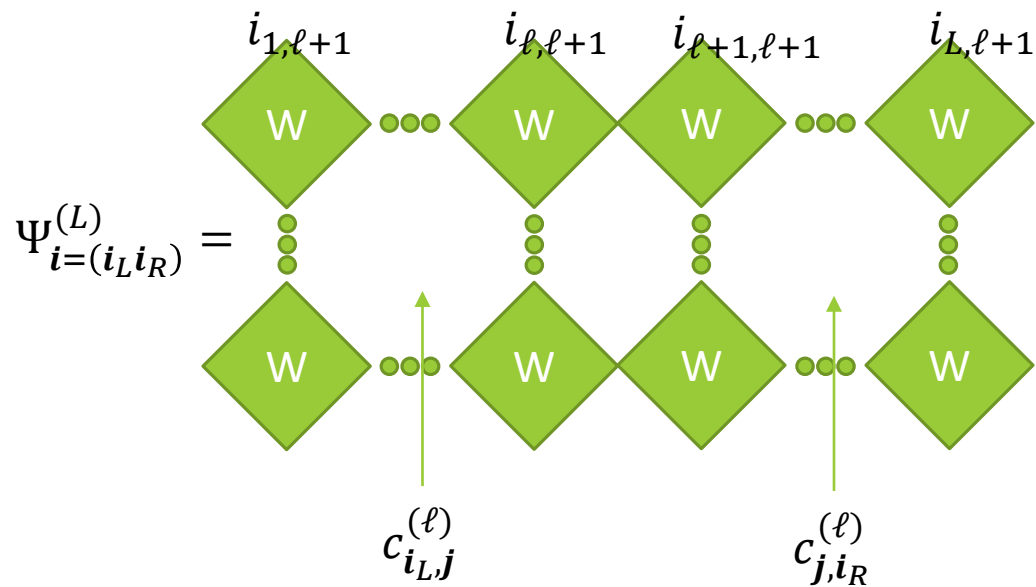
$$Z^{(4)} = \begin{array}{c} \diamond W \quad \diamond W \quad \diamond W \quad \diamond W \\ \diamond W \quad \diamond W \quad \diamond W \quad \diamond W \\ \diamond W \quad \diamond W \quad \diamond W \quad \diamond W \\ \diamond W \quad \diamond W \quad \diamond W \quad \diamond W \end{array}$$

演算コスト:  $O(2^4)$ , メモリコスト:  $O(2^4)$

演算コスト:  $O(2^{40}) \rightarrow O(2^6)$   
メモリコスト:  $O(2^4)$

# 計算のボトルネック

$L = 2\ell$ の時 :

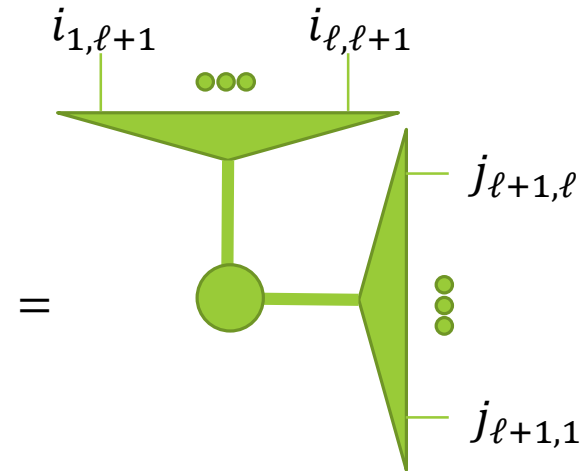


演算コスト :  $O(2^{3\ell})$ , メモリコスト :  $O(2^{2\ell})$

近似を許してより大きな  $\ell$  を取り扱う :

$$c_{i,j}^{(\ell)} \approx \sum_{\alpha=1}^m u_{i\alpha}^{(\ell)} \lambda_{\alpha}^{(\ell)} u_{j\alpha}^{(\ell)}$$

固有値問題 / 特異値分解

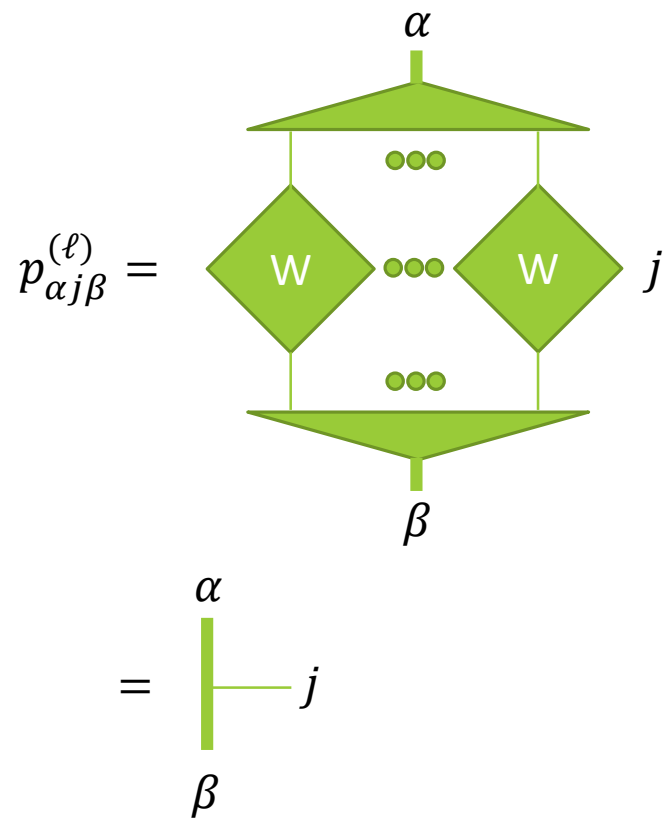


$$Z^{(L)} \approx \sum_{\alpha=1}^m (\lambda_{\alpha}^{(\ell)})^4$$

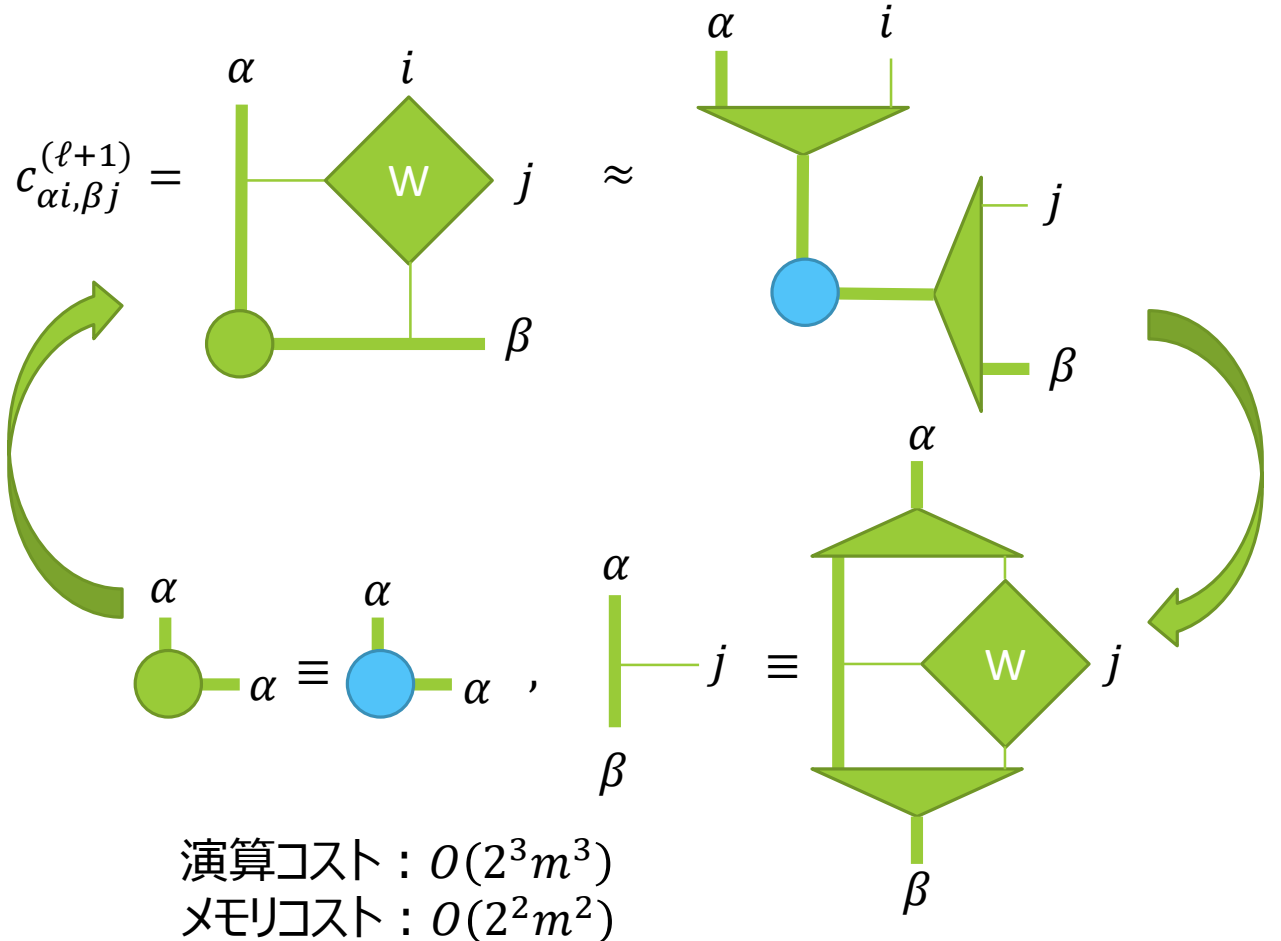
$$\text{誤差: } \epsilon = \frac{\sum_{\alpha>m}^{2\ell} (\lambda_{\alpha}^{(\ell)})^4}{Z^{(L)}}$$



# 角転送行列法繰り込み群 [西野, 奥西, JPSJ, 1996]



演算コスト :  $O(m \times 2^{2\ell+1} + m^2 2^{\ell+1})$



# CTMRGの疑似コード

0) Set  $\mathbf{W} = \{w_{i,j}\}_{1 \leq i \leq d^2, 1 \leq j \leq d^2}, w_{s+d(\sigma-1), s'+d(\sigma'-1)} = w_{s\sigma s'\sigma'}$

Set  $\mathbf{P} = \{p_{\alpha,\beta}\}_{1 \leq \alpha \leq m, 1 \leq \beta \leq md}, p_{\alpha,\alpha'+m(s-1)} = p_{\alpha\alpha's}$

where  $p_{\alpha\alpha's} = p_{\alpha'\alpha s}$

Set  $\mathbf{\Omega} = \{\delta_{\beta,\beta'} \omega_{\beta}\}, \omega_{\beta} \geq 0 \quad [m \gg d]$

1)  $\mathbf{P}' = \{p'_{\alpha,\beta}\} := \{\omega_{\alpha} p_{\alpha,\beta}\}$

2)  $\mathbf{C} = \{c_{\beta,\beta'}\} := \text{FUNC\_CTMRG}[\mathbf{P}, \mathbf{P}', \mathbf{W}]$

3) **Diagonalization:  $\mathbf{C} = \mathbf{U}\mathbf{\Omega}\mathbf{U}^t$**   
→ Update  $\mathbf{U} = \{u_{\beta,\beta'}\}$  and  $\mathbf{\Omega}$

4)  $\mathbf{U}' = \{u'_{\alpha,\beta}\}, \{u'_{\alpha,\alpha'+m(s-1)}\} := \{u_{\alpha+m(s-1),\alpha'}\}$

5)  $\mathbf{C} := \text{FUNC\_CTMRG}(\mathbf{P}, \mathbf{U}', \mathbf{W})$

6)  $\mathbf{P} := \{(\mathbf{U}^t \mathbf{C})_{\alpha,\beta}\}$

7) go to 1)

- **EigenExa**

- **BLACS**

- **Reshape by MPI\_ALLTOALLV**

FUNC\_CTMRG[ $\mathbf{P}, \mathbf{P}', \mathbf{W}$ ]

1)  $\mathbf{L} := \mathbf{P}^t \mathbf{P}'$

2)  $\mathbf{L}' = \{\ell'_{\gamma,i}\}_{1 \leq \gamma \leq m^2, 1 \leq i \leq d^2},$

$\{\ell'_{\alpha+m(\alpha'-1), s+d(s'-1)}\} := \{\ell_{\alpha+m(s-1), \alpha'+m(s'-1)}\}$

3)  $\mathbf{C}' := \mathbf{L}' \mathbf{W}$

4)  $\{c_{\alpha+m(s-1), \alpha'+m(s'-1)}\} := \{c'_{\alpha+m(\alpha'-1), s+d(s'-1)}\}$

5) return  $\mathbf{C}$

# 行列の格納形式

1×1の2次元ブロックサイクリック分割

例)  $d = 3, m = 3$  ノード並列数 :  $3 \times 3$

11	12	13	14	15	16	17	18	19
21	22	23	24	25	26	27	28	29
31	32	33	34	35	36	37	38	39
41	42	43	44	45	46	47	48	49
51	52	53	54	55	56	57	58	59
61	62	63	64	65	66	67	68	69
71	72	73	74	75	76	77	78	79
81	82	83	84	85	86	87	88	89
91	92	93	94	95	96	97	98	99

(Global view of L)

Node:0	Node:3	Node:6
11 14 17	12 15 18	13 16 19
41 44 47	42 45 48	43 46 49
71 74 77	72 75 78	73 76 79
Node:1	Node:4	Node:7
21 24 27	22 25 28	23 26 29
51 54 57	52 55 58	53 56 59
81 84 87	82 85 88	83 86 89
Node:2	Node:5	Node:8
31 34 37	32 35 38	33 36 39
61 64 67	62 65 68	63 66 69
91 94 97	92 95 98	93 96 99

(Local view of L)

置換操作： $\{\ell'_{\alpha+m(\alpha'-1),s+d(s'-1)}\} := \{\ell_{\alpha+m(s-1),\alpha'+m(s'-1)}\}$

11	14	17	41	44	47	71	74	77
12	15	18	42	45	48	72	75	78
13	16	19	43	46	49	73	76	79
21	24	27	51	54	57	81	84	87
22	25	28	52	55	58	82	85	88
23	26	29	53	56	59	83	86	89
31	34	37	61	64	67	91	94	97
32	35	38	62	65	68	92	95	98
33	36	39	63	66	69	93	96	99



11	12	13	14	15	16	17	18	19
21	22	23	24	25	26	27	28	29
31	32	33	34	35	36	37	38	39
41	42	43	44	45	46	47	48	49
51	52	53	54	55	56	57	58	59
61	62	63	64	65	66	67	68	69
71	72	73	74	75	76	77	78	79
81	82	83	84	85	86	87	88	89
91	92	93	94	95	96	97	98	99

(Global view of L')

(Global view of L)

置換操作： $\{\ell'_{\alpha+m(\alpha'-1),s+d(s'-1)}\} := \{\ell_{\alpha+m(s-1),\alpha'+m(s'-1)}\}$

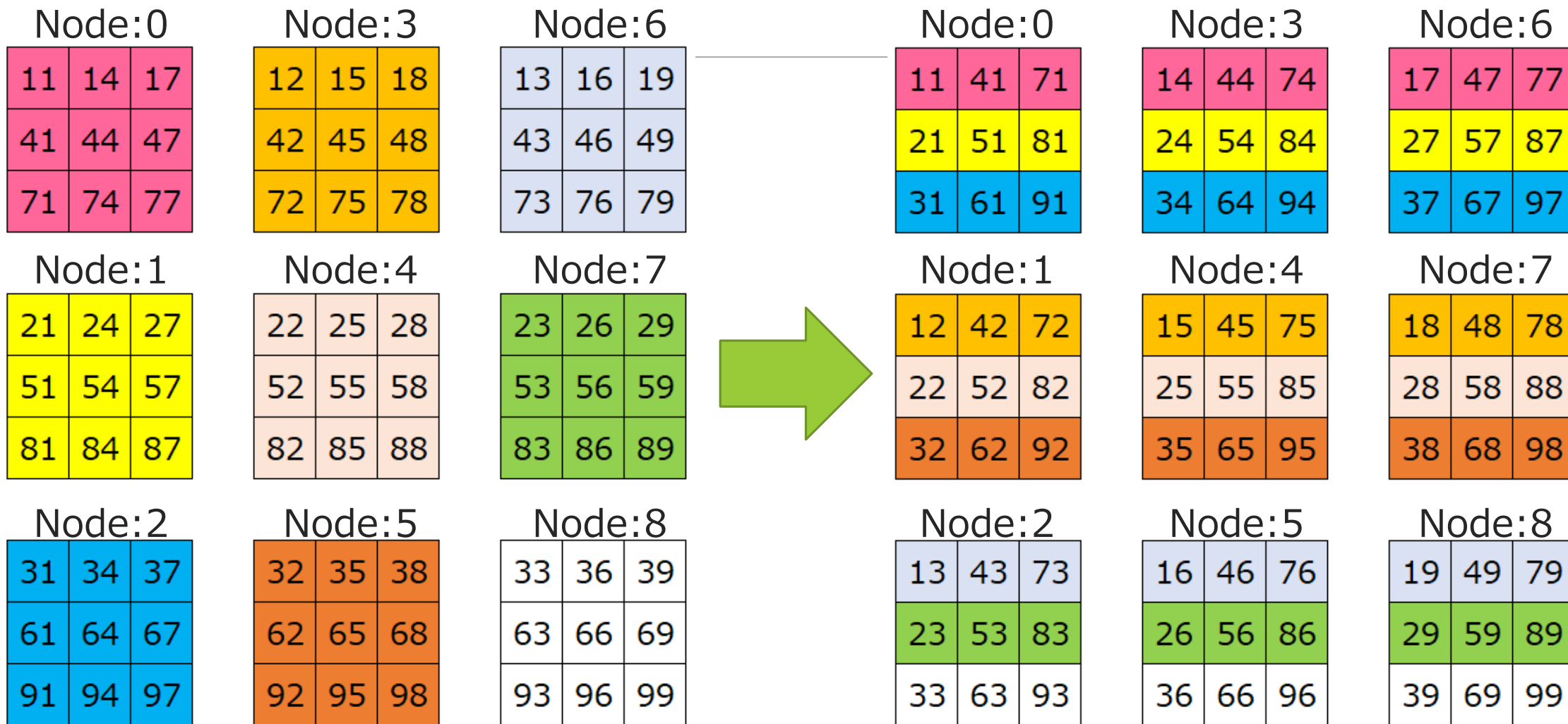
11	14	17	41	44	47	71	74	77
12	15	18	42	45	48	72	75	78
13	16	19	43	46	49	73	76	79
21	24	27	51	54	57	81	84	87
22	25	28	52	55	58	82	85	88
23	26	29	53	56	59	83	86	89
31	34	37	61	64	67	91	94	97
32	35	38	62	65	68	92	95	98
33	36	39	63	66	69	93	96	99

Node:0 11 41 71 21 51 81 31 61 91	Node:3 14 44 74 24 54 84 34 64 94	Node:6 17 47 77 27 57 87 37 67 97
Node:1 12 42 72 22 52 82 32 62 92	Node:4 15 45 75 25 55 85 35 65 95	Node:7 18 48 78 28 58 88 38 68 98
Node:2 13 43 73 23 53 83 33 63 93	Node:5 16 46 76 26 56 86 36 66 96	Node:8 19 49 79 29 59 89 39 69 99

(Global view of L')

(Global view of L)

# 実現したい操作



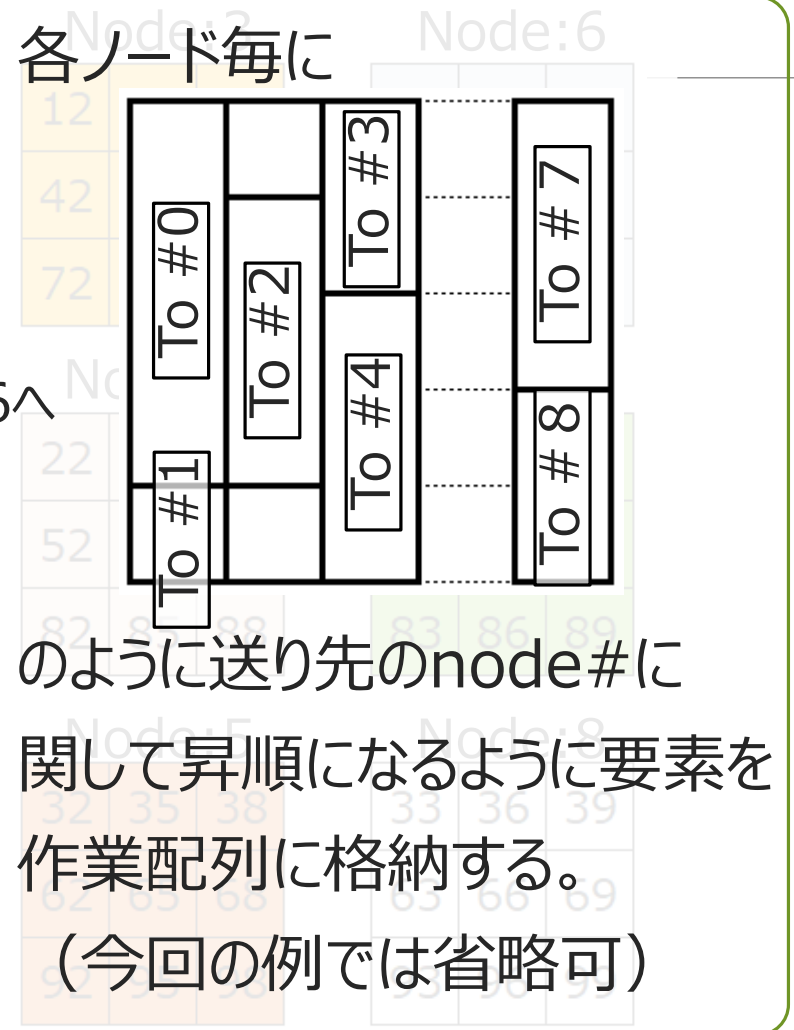
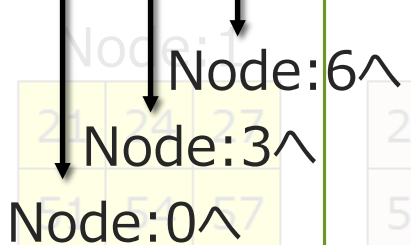
(Local view of L)

(Local view of L')

# 1) 局所要素置換

Node:0

11	14	17
41	44	47
71	74	77



Node:2

31	34	37
61	64	67
91	94	97

Node:0

11	41	71
21	51	81
31	61	91

Node:1

12	42	72
22	52	82
32	62	92

Node:2

13	43	73
23	53	83
33	63	93

Node:3

14	44	74
24	54	84
34	64	94

Node:4

15	45	75
25	55	85
35	65	95

Node:5

16	46	76
26	56	86
36	66	96

Node:6

17	47	77
27	57	87
37	67	97

Node:7

18	48	78
28	58	88
38	68	98

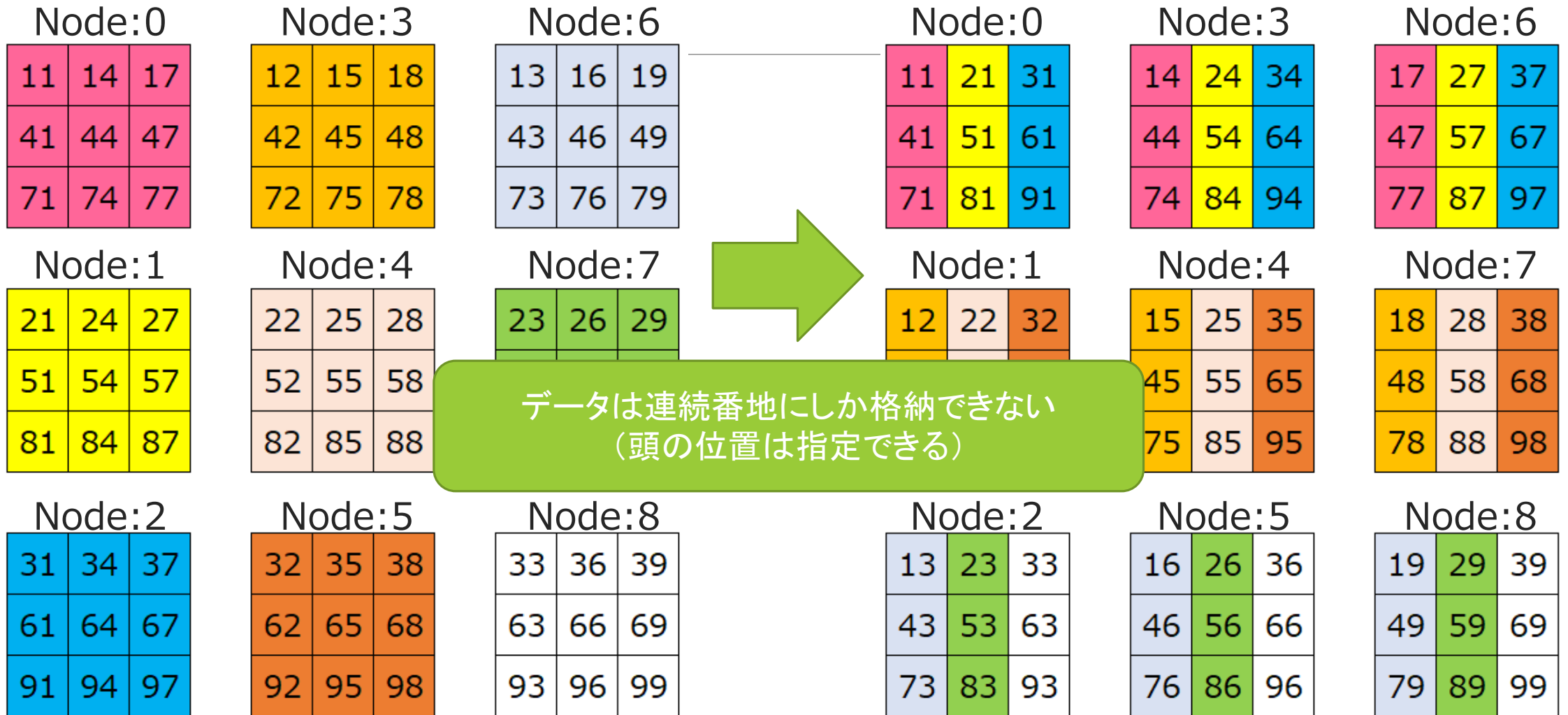
Node:8

19	49	79
29	59	89
39	69	99

(Local view of L)

(Local view of L')

# 2) MPI\_ALLTOALLV

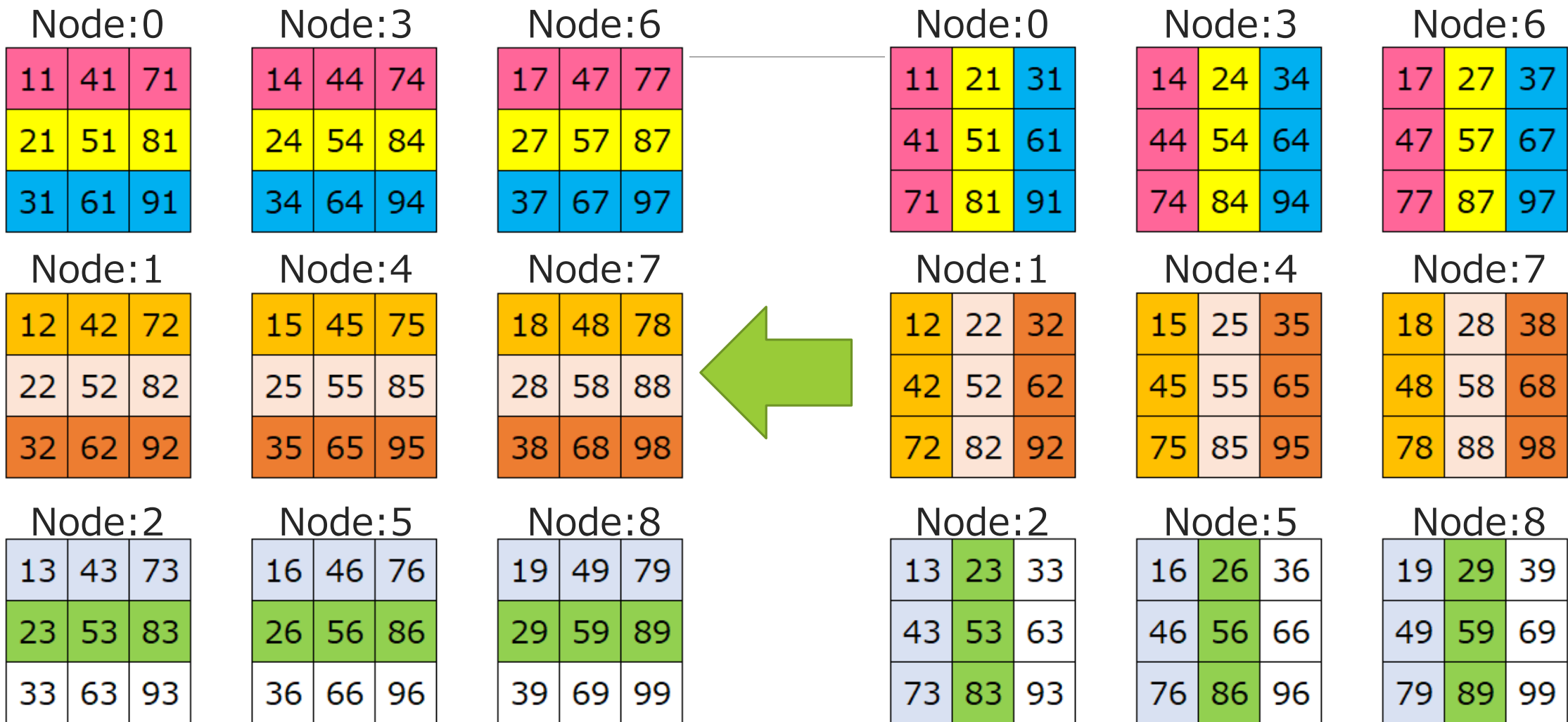


(Local view of work Mat.)

(Local view of work Mat.2)



# 3) 局所要素置換



(Local view of  $L'$ )

(Local view of work Mat.2)

# CTMRGの疑似コード

0) Set  $\mathbf{W} = \{w_{i,j}\}_{1 \leq i \leq d^2, 1 \leq j \leq d^2}, w_{s+d(\sigma-1), s'+d(\sigma'-1)} = w_{s\sigma s'\sigma'}$

Set  $\mathbf{P} = \{p_{\alpha,\beta}\}_{1 \leq \alpha \leq m, 1 \leq \beta \leq md}, p_{\alpha, \alpha'+m(s-1)} = p_{\alpha\alpha's}$

where  $p_{\alpha\alpha's} = p_{\alpha's\alpha}$

Set  $\mathbf{\Omega} = \{\delta_{\beta,\beta'} \omega_{\beta}\}, \omega_{\beta} \geq 0 \quad [m \gg d]$

1)  $\mathbf{P}' = \{p'_{\alpha,\beta}\} := \{\omega_{\alpha} p_{\alpha,\beta}\}$

2)  $\mathbf{C} = \{c_{\beta,\beta'}\} := \text{FUNC\_CTMRG}[\mathbf{P}, \mathbf{P}', \mathbf{W}]$

3) **Diagonalization:  $\mathbf{C} = \mathbf{U}\mathbf{\Omega}\mathbf{U}^t$**   
 → Update  $\mathbf{U} = \{u_{\beta,\beta'}\}$  and  $\mathbf{\Omega}$

4)  $\mathbf{U}' = \{u'_{\alpha,\beta}\}, \{u'_{\alpha, \alpha'+m(s-1)}\} := \{u_{\alpha+m(s-1), \alpha'}\}$

5)  $\mathbf{C} := \text{FUNC\_CTMRG}(\mathbf{P}, \mathbf{U}', \mathbf{W})$

6)  $\mathbf{P} := \{(\mathbf{U}^t \mathbf{C})_{\alpha,\beta}\}$

7) go to 1)

- **EigenExa**
- **BLACS**
- **Reshape by MPI\_ALLTOALLV**

MPI\_ALLTOALLVを利用したテンソルの Reshape の際に必要なリストを作成

FUNC

1)  $\mathbf{L}$ : ※テンソルのサイズが動的に変わる場合はその都度リスト作成が必要

2)  $\mathbf{L}' = \{l'_{\alpha+m(s-1), \alpha'}\}_{1 \leq \alpha \leq d^2}$

$\{l'_{\alpha+m(\alpha'-1), s+d(s'-1)}\} := \{l_{\alpha+m(s-1), \alpha'+m(s'-1)}\}$

3)  $\mathbf{C}' := \mathbf{L}'\mathbf{W}$

4)  $\{c_{\alpha+m(s-1), \alpha'+m(s'-1)}\} := \{c'_{\alpha+m(\alpha'-1), s+d(s'-1)}\}$

5) return  $\mathbf{C}$

# CTMRGの疑似コード

0) Set  $\mathbf{W} = \{w_{i,j}\}_{1 \leq i \leq d^2, 1 \leq j \leq d^2}, w_{s+d(\sigma-1), s'+d(\sigma'-1)} = w_{s\sigma s'\sigma'}$

Set  $\mathbf{P} = \{p_{\alpha,\beta}\}_{1 \leq \alpha \leq m, 1 \leq \beta \leq md}, p_{\alpha, \alpha'+m(s-1)} = p_{\alpha\alpha's}$

where  $p_{\alpha\alpha's} = p_{\alpha's\alpha}$

Set  $\mathbf{\Omega} = \{\delta_{\beta,\beta'} \omega_{\beta}\}, \omega_{\beta} \geq 0 \quad [m \gg d]$

1)  $\mathbf{P}' = \{p'_{\alpha,\beta}\} := \{\omega_{\alpha} p_{\alpha,\beta}\}$

2)  $\mathbf{C} = \{c_{\beta,\beta'}\} := \text{FUNC\_CTMRG}[\mathbf{P}, \mathbf{P}', \mathbf{W}]$

3) **Diagonalization:  $\mathbf{C} = \mathbf{U}\mathbf{\Omega}\mathbf{U}^t$**   
→ Update  $\mathbf{U} = \{u_{\beta,\beta'}\}$  and  $\mathbf{\Omega}$

4)  $\mathbf{U}' = \{u'_{\alpha,\beta}\}, \{u'_{\alpha, \alpha'+m(s-1)}\} := \{u_{\alpha+m(s-1), \alpha'}\}$

5)  $\mathbf{C} := \text{FUNC\_CTMRG}(\mathbf{P}, \mathbf{U}', \mathbf{W})$

6)  $\mathbf{P} := \{(\mathbf{U}^t \mathbf{C})_{\alpha,\beta}\}$

7) go to 1)

- **EigenExa**

- **BLACS**

EigenExa, BLACSともに1×1の  
2次元ブロックサイクリック分割で実行

FUNC\_CTMRG[ $\mathbf{P}, \mathbf{P}', \mathbf{W}$ ]

1)  $\mathbf{L} := \mathbf{P}^t \mathbf{P}'$

2)  $\mathbf{L}' = \{\ell'_{\gamma,i}\}_{1 \leq \gamma \leq m^2, 1 \leq i \leq d^2}$ ,

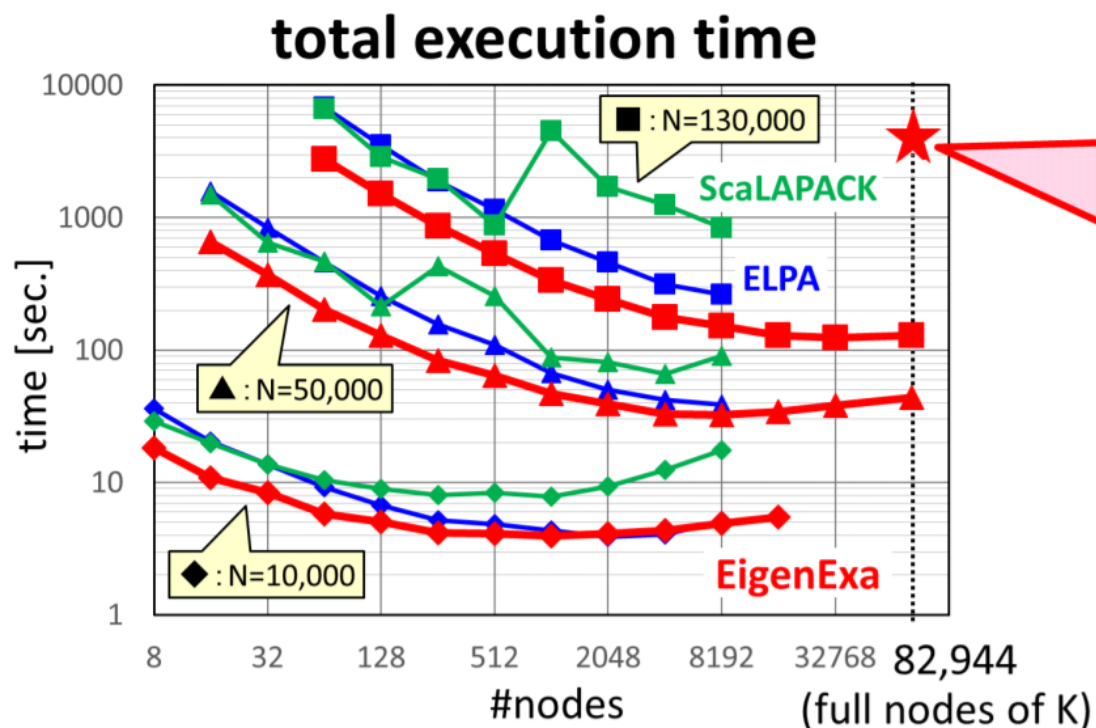
$\{\ell'_{\alpha+m(\alpha'-1), s+d(s'-1)}\} := \{\ell_{\alpha+m(s-1), \alpha'+m(s'-1)}\}$

3)  $\mathbf{C}' := \mathbf{L}' \mathbf{W}$

4)  $\{c_{\alpha+m(s-1), \alpha'+m(s'-1)}\} := \{c'_{\alpha+m(\alpha'-1), s+d(s'-1)}\}$

5) return  $\mathbf{C}$

# EigenExa



## one million dimensional dense eigenvalue computing

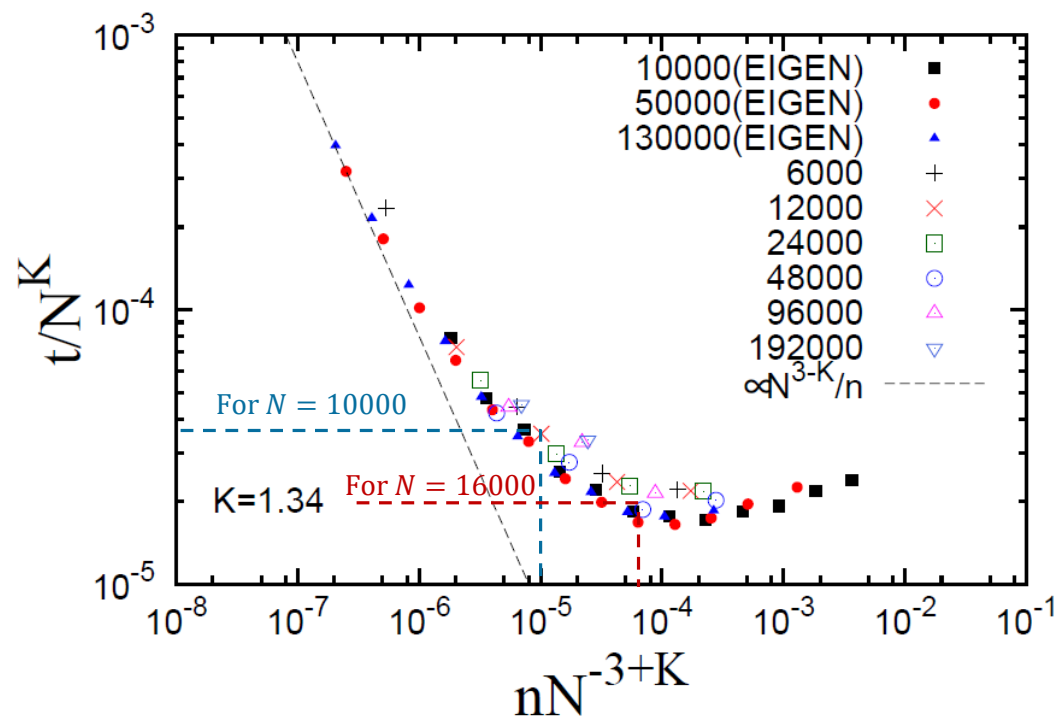
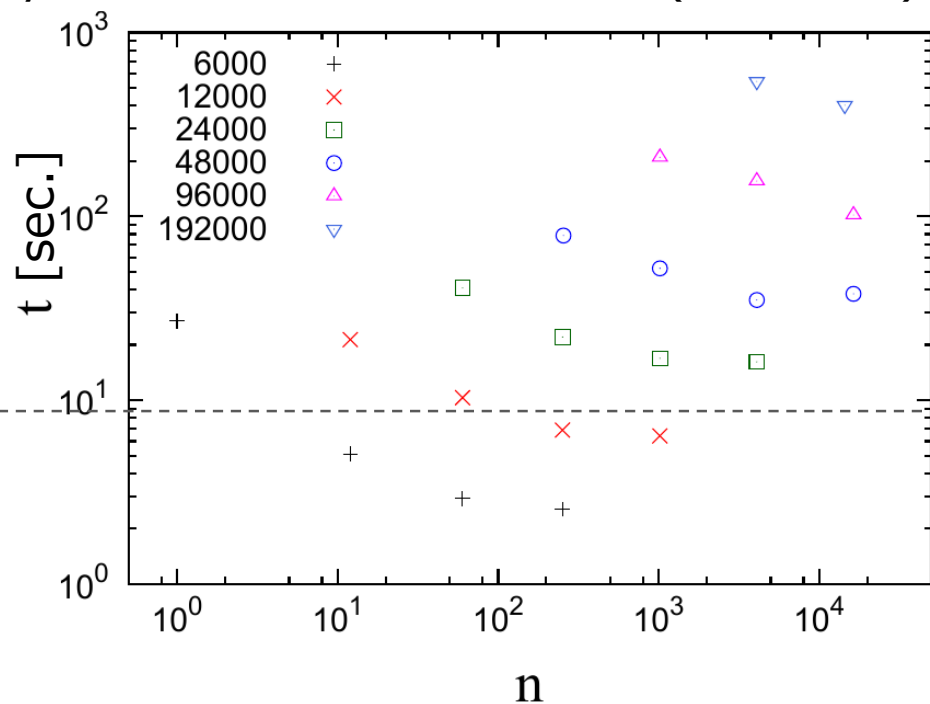
- using full nodes of K (82,944 procs., 663,552 cores)
- 3,464 sec. (penta: 1,968s, D&C: 922s, back: 574s)
- 1.7 PFLOPS (16% of peak)
- Residual :  $\max_i \|Av_i - \lambda_i v_i\|_2 / \|A\|_F = 3.1 \times 10^{-13}$
- Orthogonality :  $\|V^T V - I\|_F = 2.1 \times 10^{-10}$

## evaluation conditions

- Calculations : all eigenpairs of N x N random matrix
- Libraries : BLAS / MPI provided from Fujitsu on K
- Assignments : 1 MPI process with 8 threads / 1 node

# 並列化CTMRGのベンチマーク計算@京

System: Icosahedron model ( $N = 12m$ )



# まとめ

---

- ✓ 対称的な古典2次元模型に対するCTMRGを完全にMPI並列化
  - テンソル要素の入れ替え : MPI\_ALLTOALLV
  - 行列積 : BLACS
  - 対角化 : EigenExa
- ✓ 計算時間  $t$ , 行列次元  $N$ , 並列ノード数  $n$  の間のスケーリング関係式の調査

$$t = N^K F[n/N^{3-K}] \sim \begin{cases} N^3/n & (n/N^{3-K} \ll 1) \\ N^K & (n/N^{3-K} \gg 1) \end{cases}$$

京の場合 :  $K = 1.34$

# まとめ

---

- ✓ 計算全体のボトルネック：CTMの固有値問題

内部自由度 $d$ が大きいとき：Partial eigensolver のほうがお得になる

理由その1) 必要な固有値固有ベクトルの個数は行列次元の $1/d$

理由その2) 反復解法の上手い初期ベクトルを与えられる可能性が高い

PWFRG【西野, 奥西, 1995】やiDMRG【McCulloch, 2008】といった  
波動関数予測と同様の概念

- ✓ 非対称な模型の場合：固有値問題→特異値分解