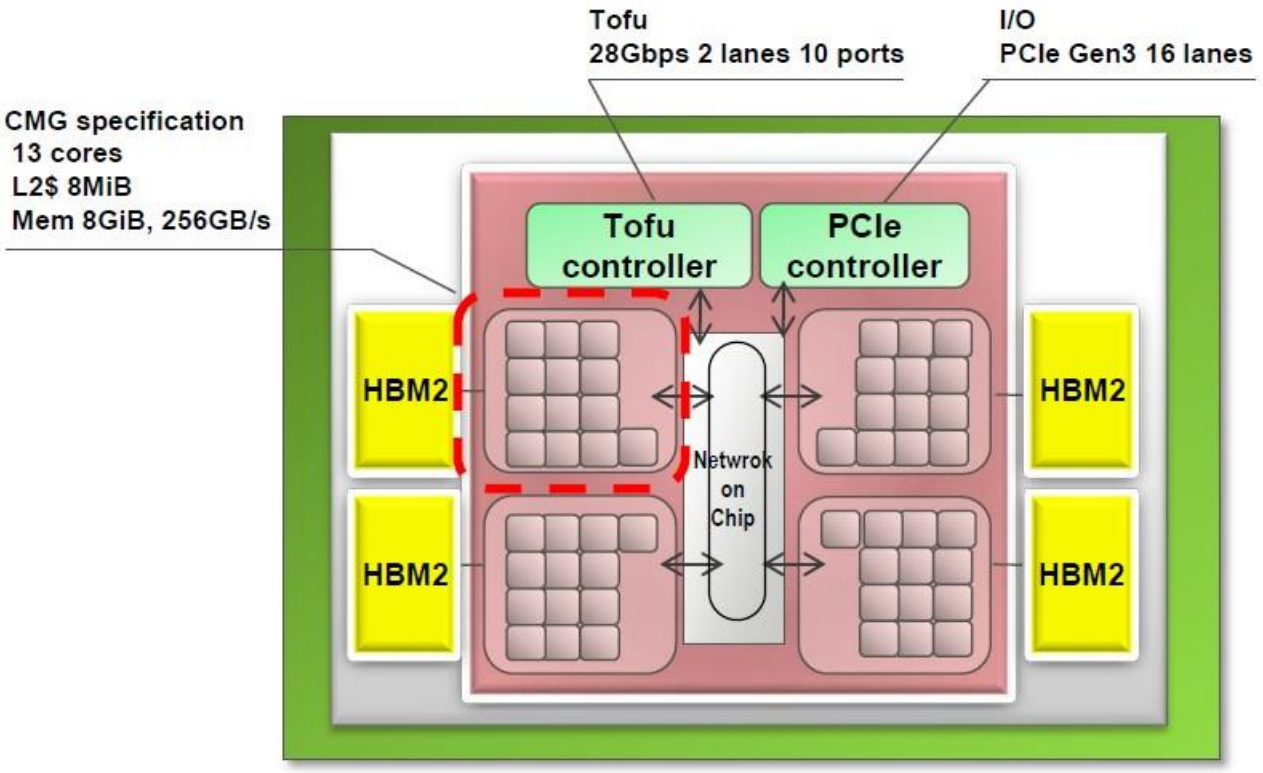
Hardware-aware graph coloring techniques for efficient sparse linear algebra

Jonas Thies (German Aerospace Center)
Christie Louis Alappat, Georg Hager and Gerhard Wellein
(University of Erlangen-Nuremberg)

Motivation: Japanese Post-K design (Fujitsu A64FX)



- ARM processor (energy efficient)
- 48 cores
- about 1 TB/s memory bandwidth
- NUMA
- 512 bit SIMD

Much friendlier than GPUs for many HPC applications

- https://www.nextplatform.com/2018/08/24/fujitsus-a64fx-arm-chip-waves-the-hpc-banner-high/

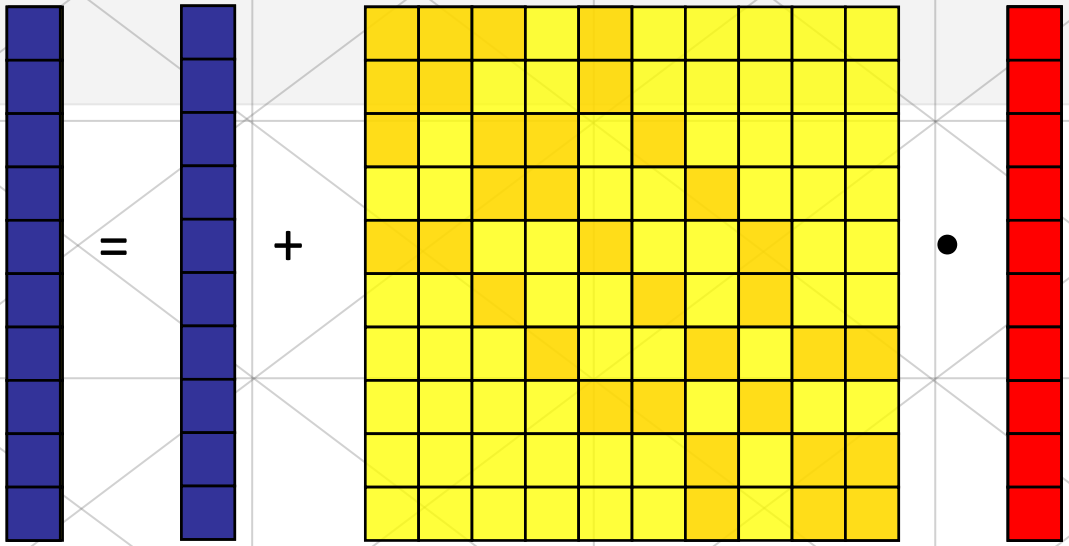
Sparse Matrix Operations – Computational Bottleneck

Sparse Matrix-Vector Multiplication (SpMV)

$$b=b+A*x$$

```
#pragma omp parallel for
for(int row=0; row<nrows; ++row)
{
   temp=0;
   for (int j=rowptr[row]; j<rowptr[row+1]; ++j)
   {
      temp += A[j] * x[col[j]];

   }
   b[row] += temp;
}
```

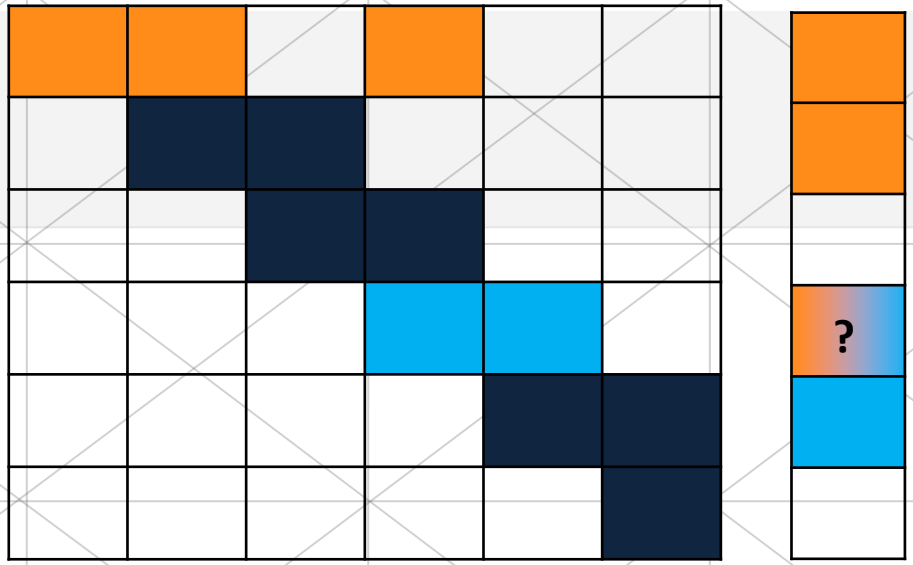


Irregular data accesses

Sparse Matrix Operations – Data Dependencies

Similar problem:
SymmSpMV

- Thread 1
- Thread 2



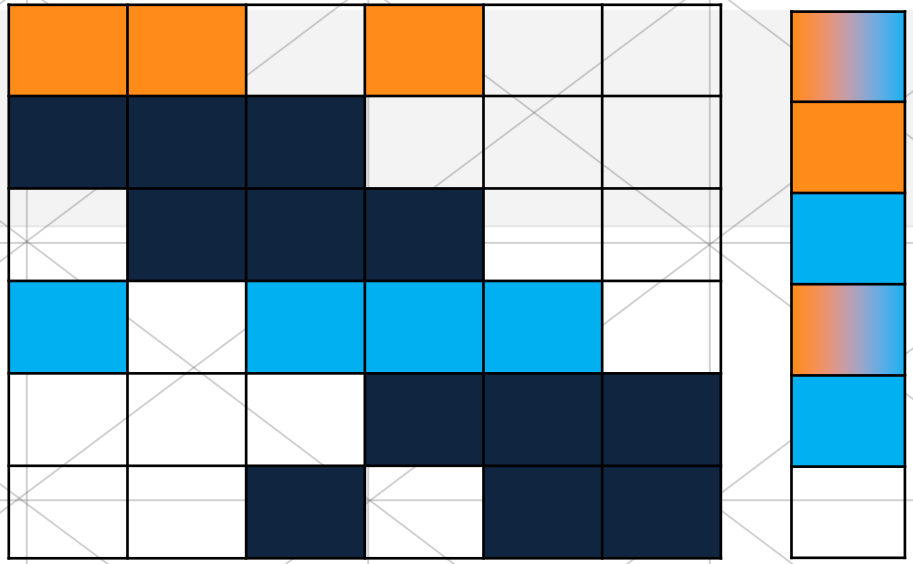

```
#pragma omp parallel for
for(int row=0; row<nrows; ++row)
{
   temp=0;
   for (int j=rowptr[row]; j<rowptr[row+1]; ++j)
   {
      temp += A[j] * x[col[j]];
      b[col[j]] += A[j] * x[row];
   }
   b[row] += temp;
}
```

D2-dependency

Sparse Matrix Operations – Data Dependencies

Different problem:
Gauß-Seidel (GS)

- Thread 1
- Thread 2



A x

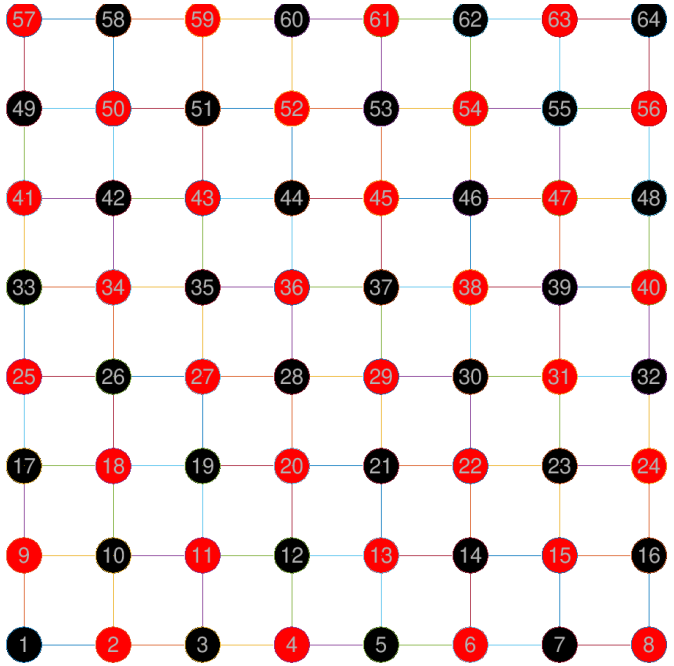
```
#pragma omp parallel for
for(int row=0; row<nrows; ++row)
{
   temp=b[row];
   for (int j=rowptr[row]+1; j<rowptr[row+1]; ++j)
   {
      temp -= A[j] * x[col[j]];
   }
    diag = A[rowptr[row]];
   x[row]  = temp/diag;
}
```

D1-dependency

How to solve this?

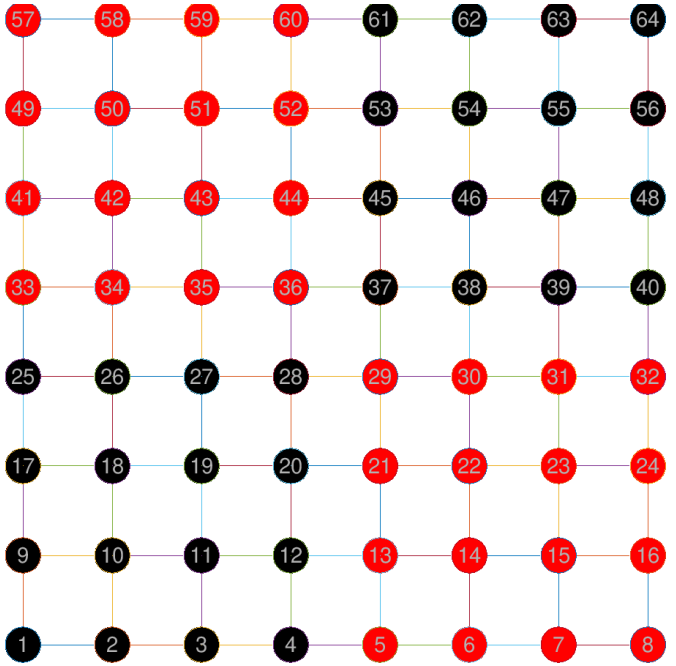
Matrix re-ordering

Multicoloring[5] (MC)



Does it have an impact on performance ?

Algebraic Block Multicoloring[6] (ABMC)



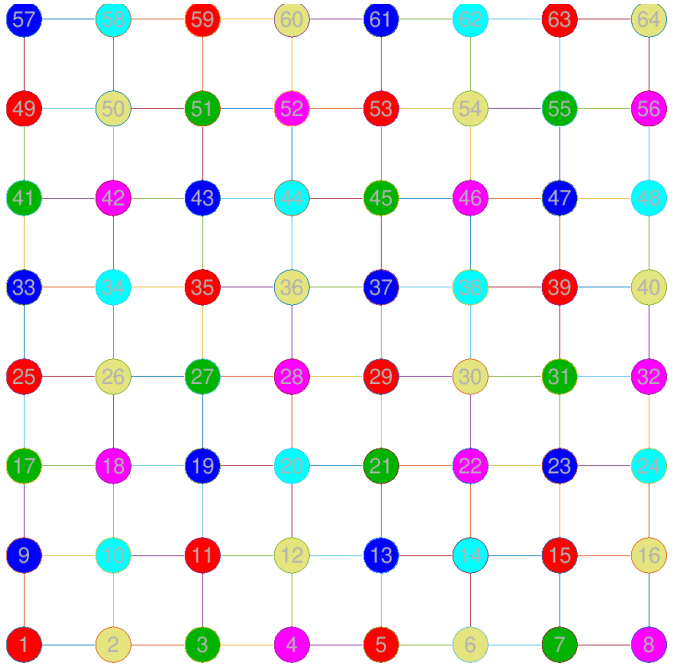
M. T. Jones and P. E. Plassmann, Scalable iterative solution of sparse linear systems URL:https://doi.org/10.1016/0167-8191(94)90004-3.

T. Iwashita, H. Nakashima, and Y. Takahashi, Algebraic block multi-color ordering method for parallel multi-threaded sparse triangular solver in iccg method URL:https://doi.org/10.1109/IPDPS.2012.51

How to solve this?

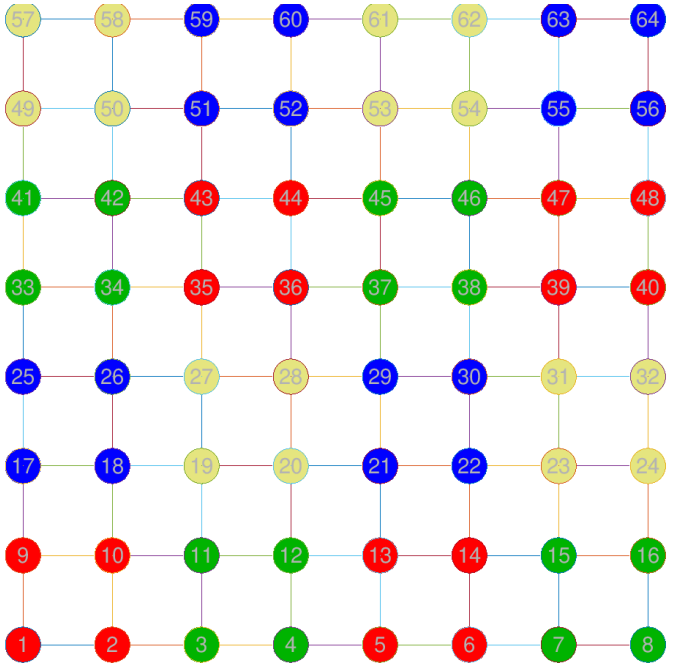
Matrix re-ordering

Multicoloring[5] (MC)



Does it have an impact on performance ?

Algebraic Block Multicoloring[6] (ABMC)



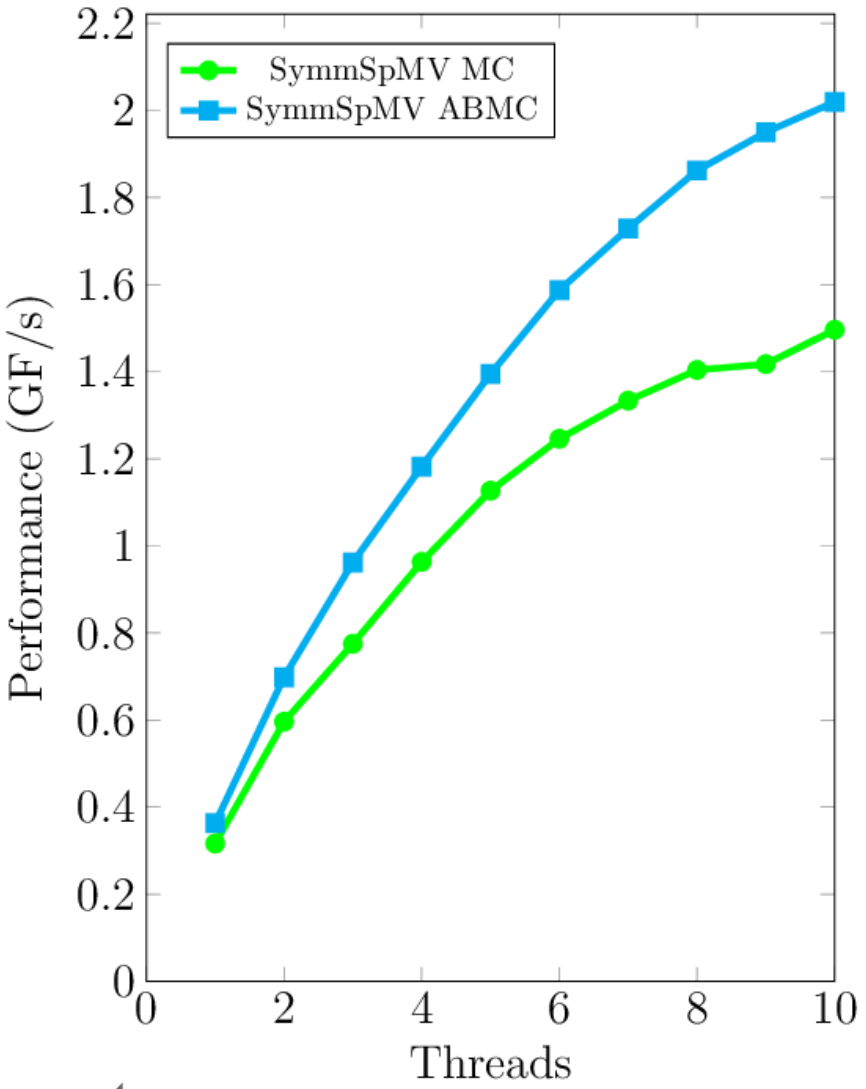
M. T. Jones and P. E. Plassmann, Scalable iterative solution of sparse linear systems URL:https://doi.org/10.1016/0167-8191(94)90004-3.

T. Iwashita, H. Nakashima, and Y. Takahashi, Algebraic block multi-color ordering method for parallel multi-threaded sparse triangular solver in iccg method URL:https://doi.org/10.1109/IPDPS.2012.51

Performance Engineering

Ivy Bridge E5-2660 v2 @ 2.2 GHz

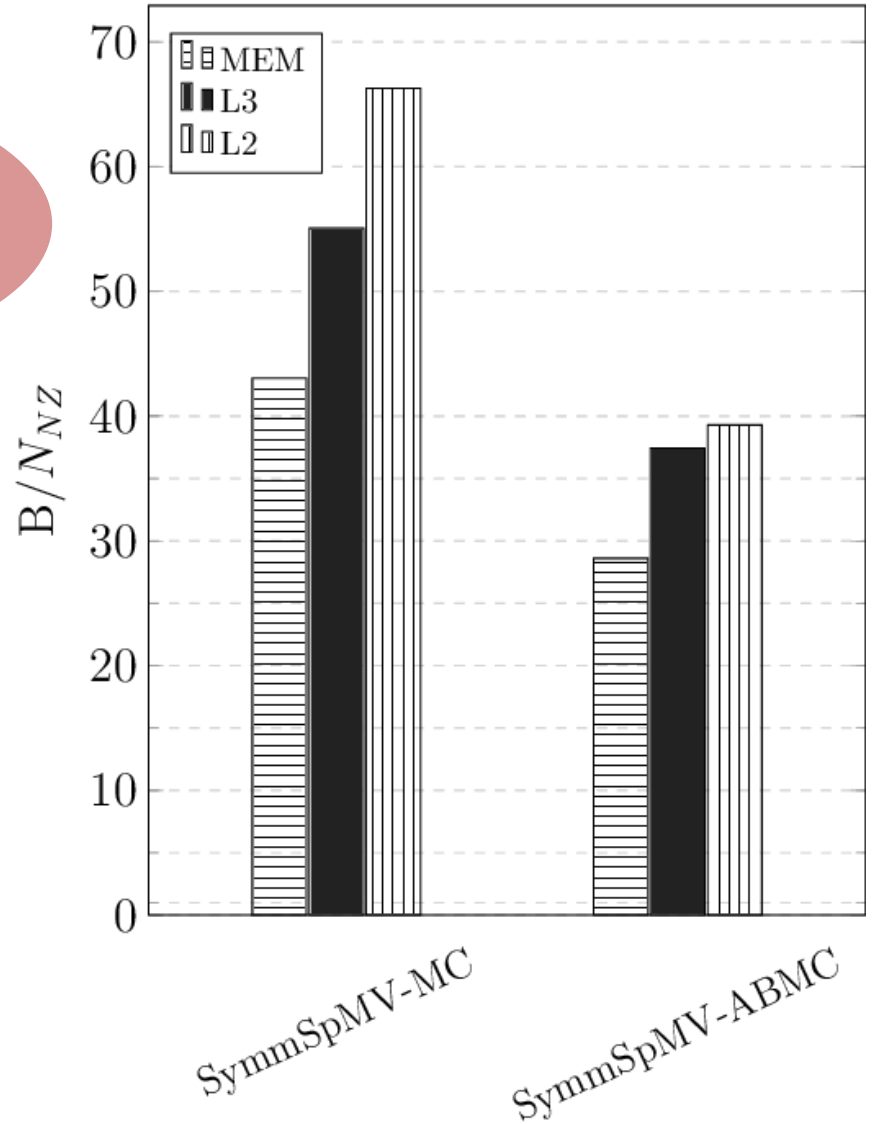
Performance



Does it have an impact on performance ?

Is it GOOD or BAD

Data Traffic



DLR

Performance Engineering – Roofline Model

SpMV

```
for(int row=0; row<nrows; ++row)
{
    temp=0;
    for (int j=rowptr[row]; j<rowptr[row+1]; ++j)
    {
        temp += A[j] * x[col[j]];
    }
    b[row] += temp;
}
```

SymmSpMV

```
for(int row=0; row<nrows; ++row)
{
    temp=0;
    for (int j=rowptr[row]; j<rowptr[row+1]; ++j)
    {
        temp += A[j] * x[col[j]];
        b[col[j]] += A[j] * x[row];
    }
    b[row] += temp;
}
```

$$I_{SpMV} = \frac{2}{8 + 4 + 8\alpha_{SpMV} + \frac{20}{N_{nzr}}} \frac{flops}{bytes}$$

$$P_{SpMV} = I_{SpMV} * bs$$

→ Measure α_{SpMV}

$$I_{SymmSpMV} = \frac{4}{8 + 4 + 24\alpha_{SymmSpMV} + \frac{4}{N_{nzr}^{symm}}} \frac{flops}{bytes}$$

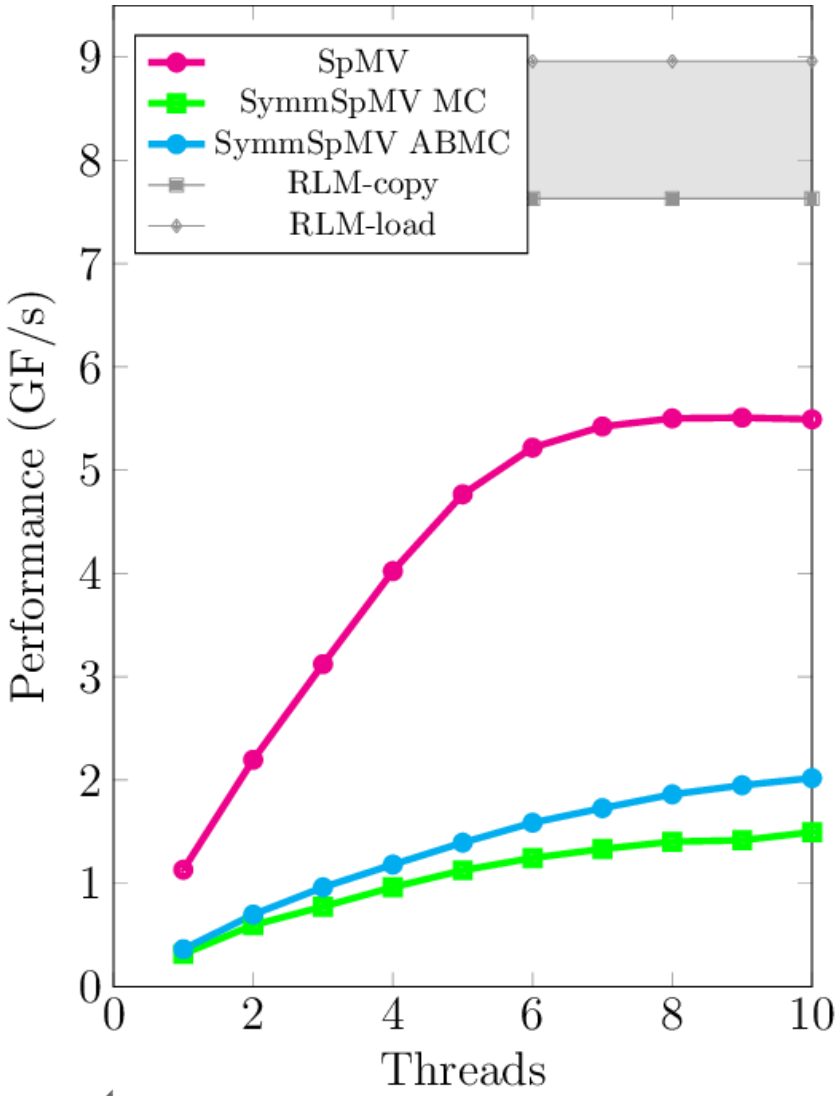
α_{SpMV} is a lower limit on $\alpha_{SymmSpMV}$

$$P_{SymmSpMV} = I_{SymmSpMV} * bs$$

Motivation

Ivy Bridge E5-2660 v2 @ 2.2 GHz

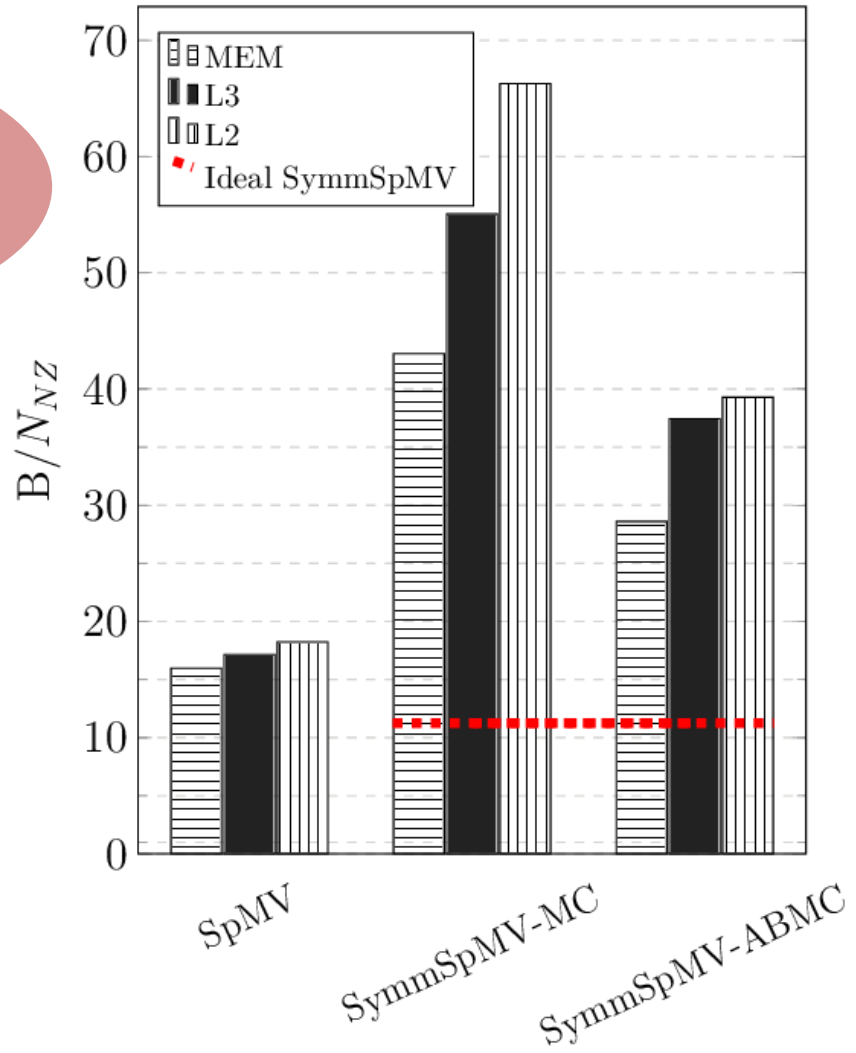
Performance



Does it have an impact on performance ?

Is it GOOD or BAD

Data Traffic

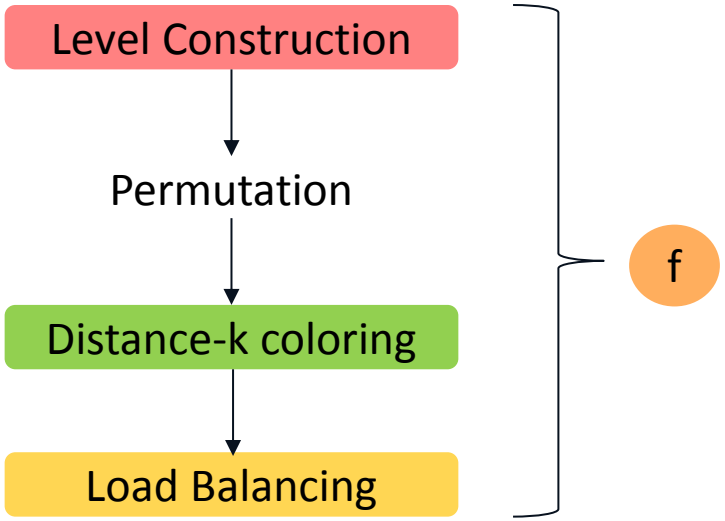


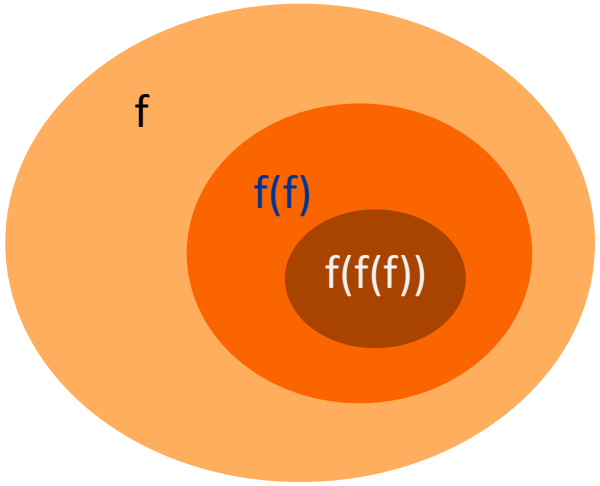
DLR

Recursive Algebraic Coloring Engine

Objectives motivated by hardware efficiency
Preserve data locality (lower α factor).
Generate sufficient parallelism to support hardware underneath.
Reduce synchronization overheads.
Use simple data format like CRS.

4 Steps

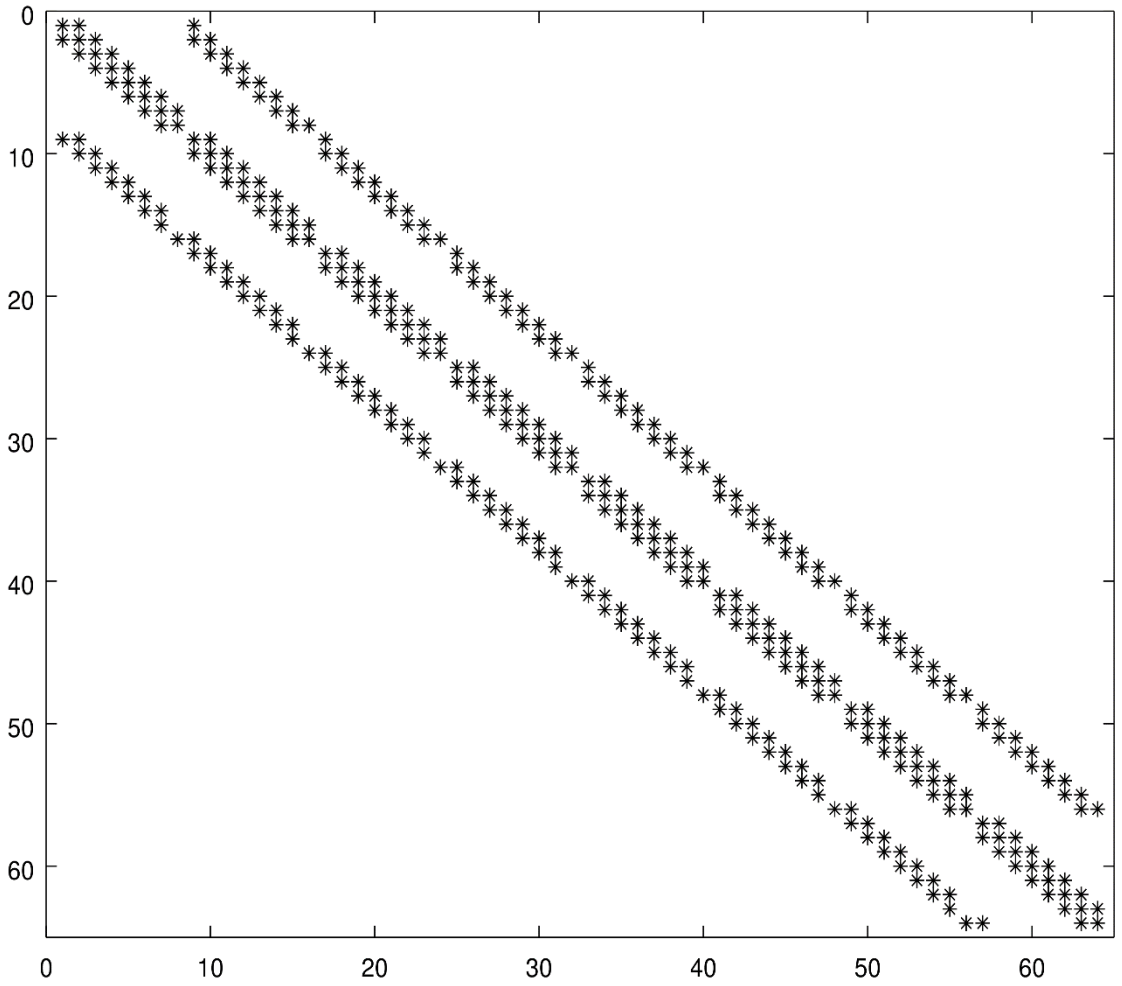




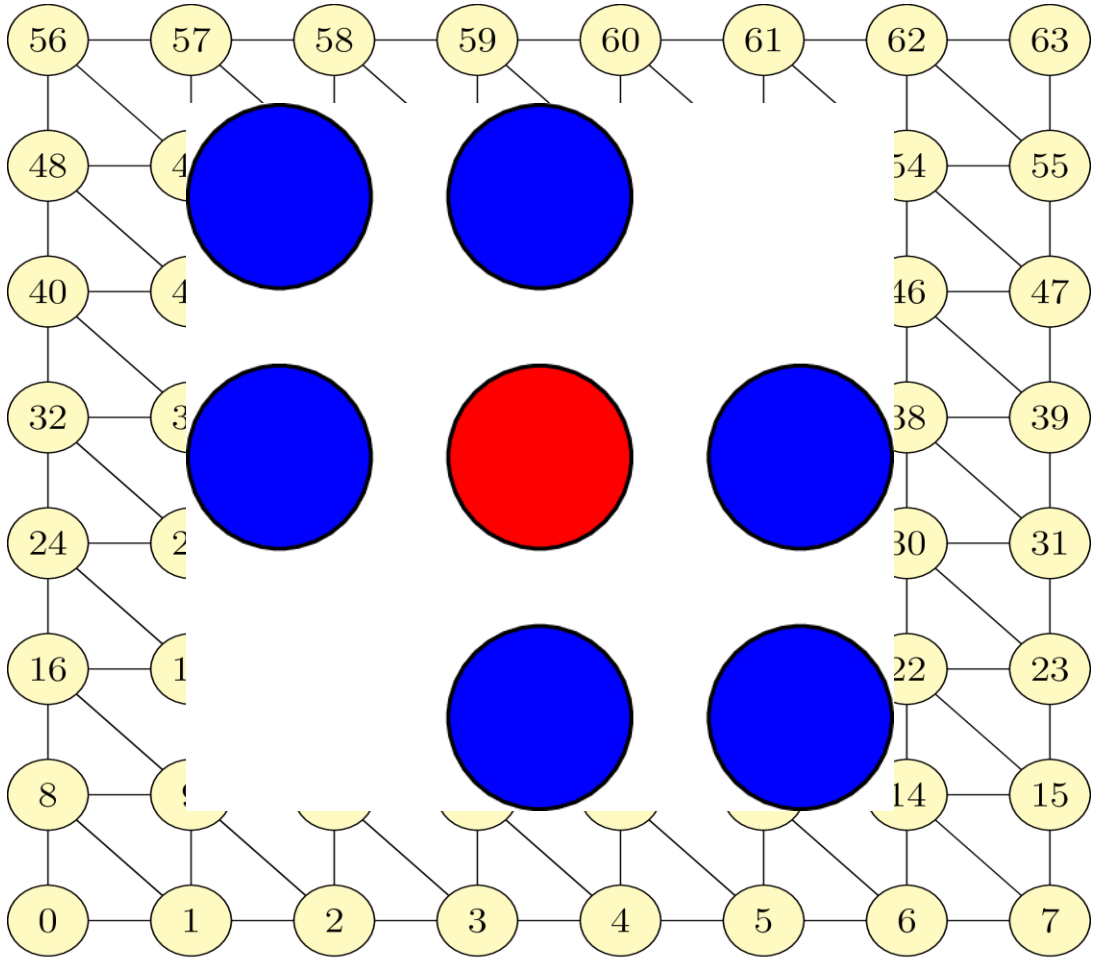
Recursive application if needed

Step 1: Level Construction – BFS

Sparse Matrix → Graph



Sparse Matrix

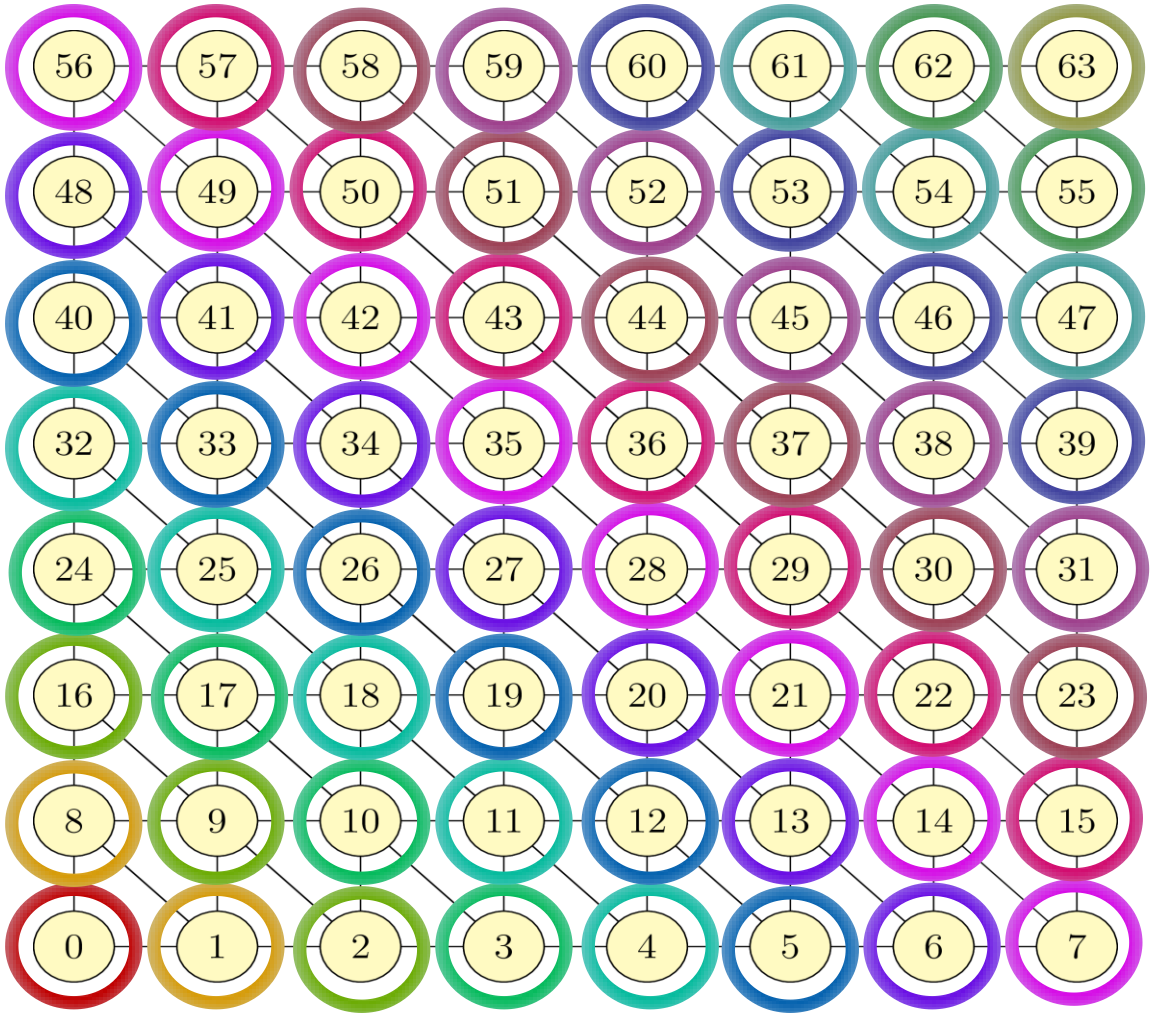


Stencil

Example of 2D-7 Point stencil

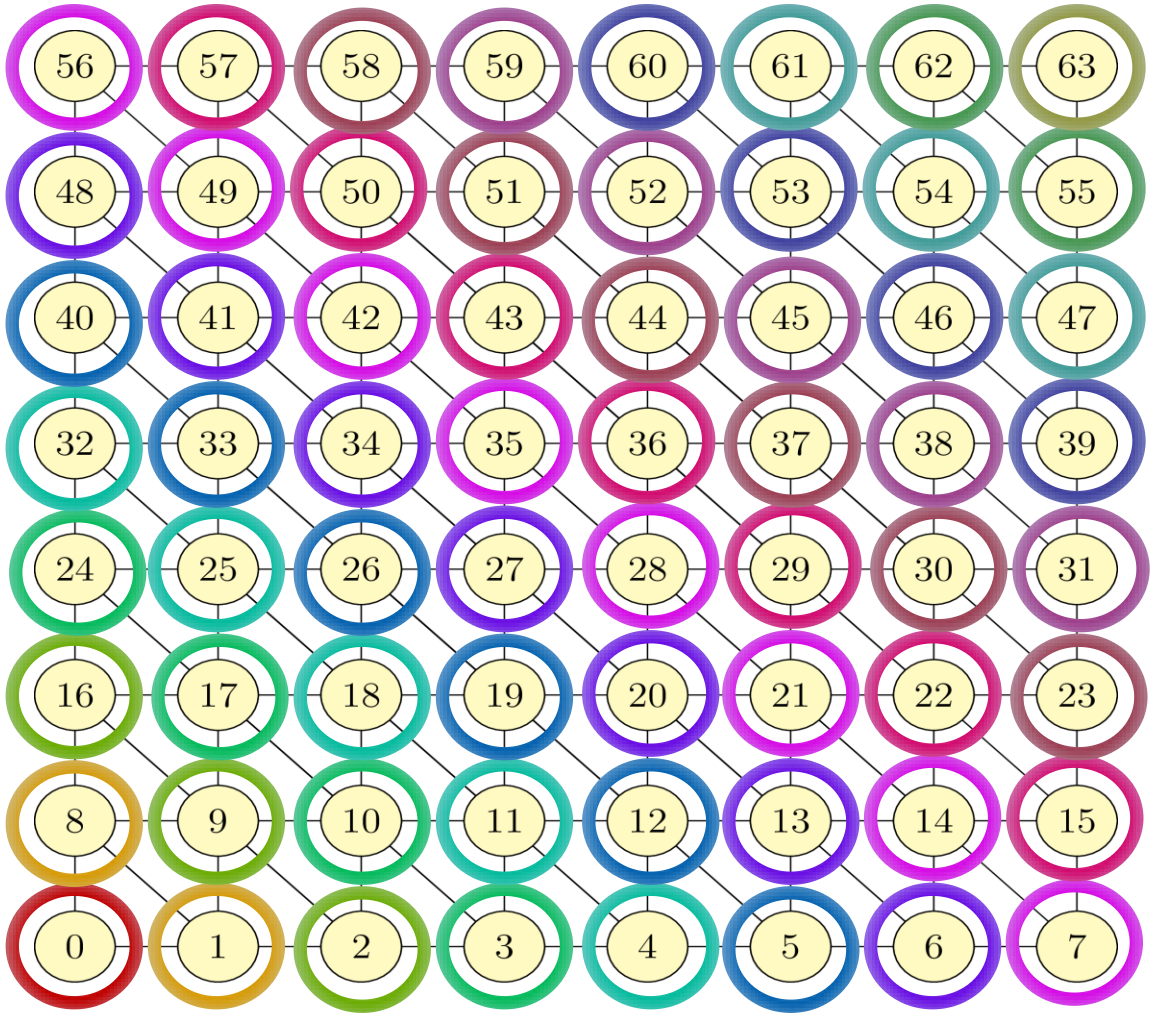
7. C. Y. Lee, An algorithm for path connections and its applications, URL:https://doi.org/10.1109/TEC.1961.5219222.

Step 1: Level Construction – BFS

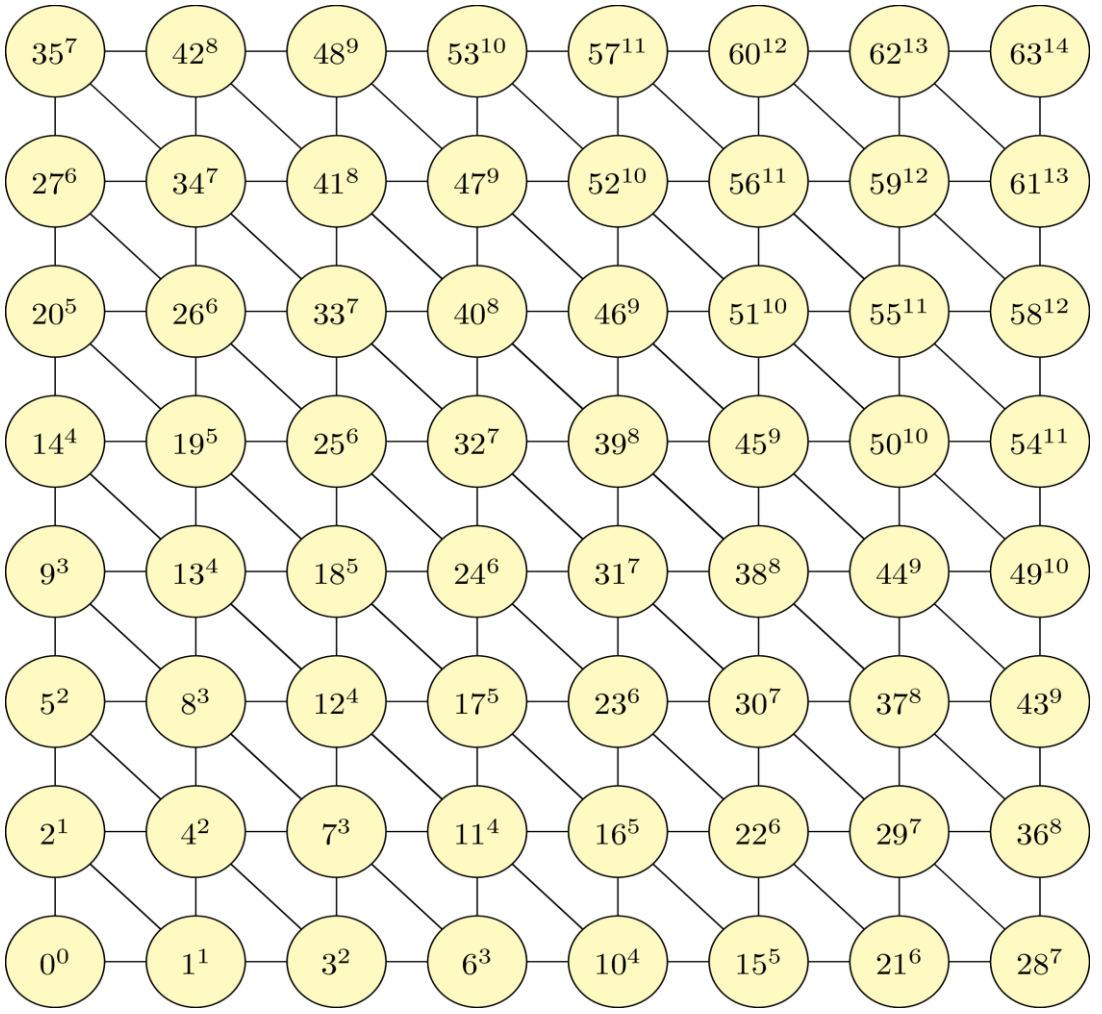


Example of 2D-7 Point stencil

Step 1: Level Construction – BFS



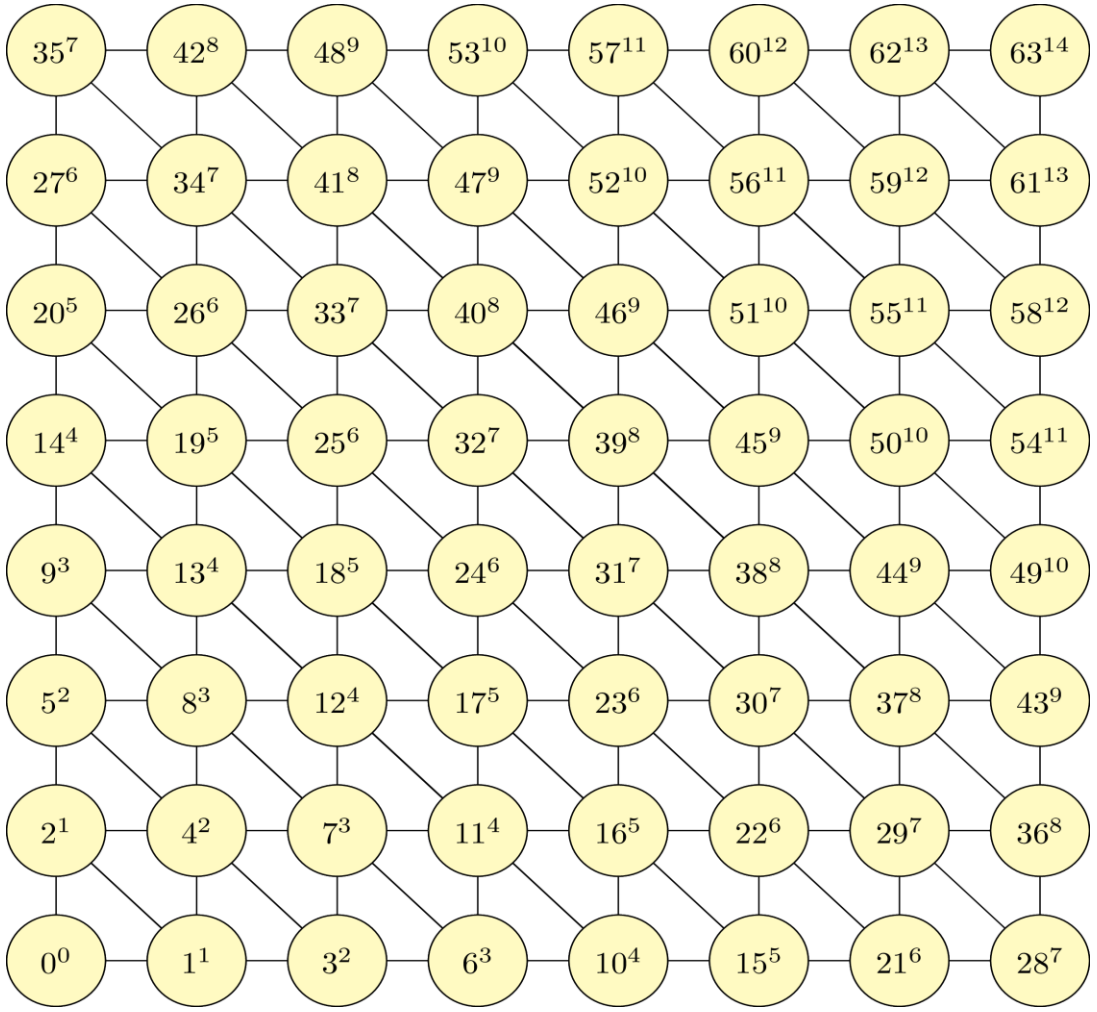
Original graph



Permuted graph

Example of 2D-7 Point stencil

Step 1: Level Construction – BFS

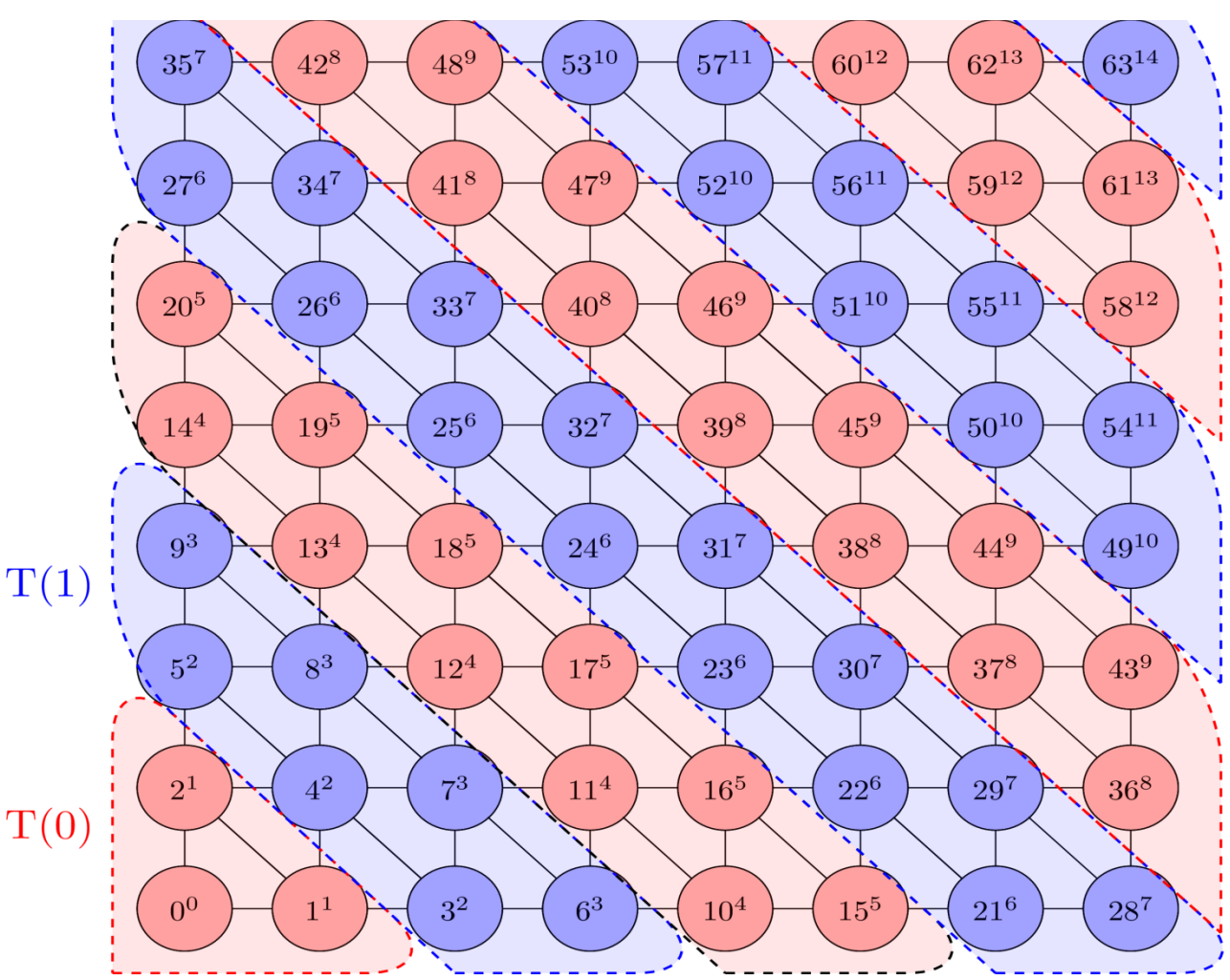


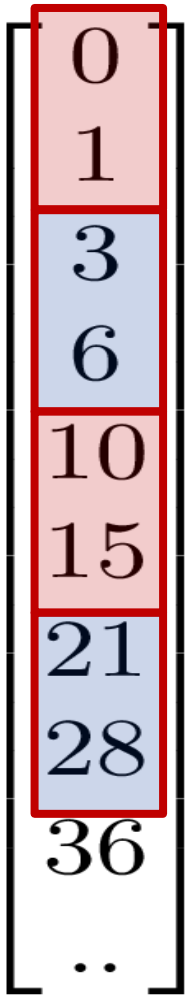
$$\begin{bmatrix} 0 \\ 1 \\ 3 \\ 6 \\ 10 \\ 15 \\ 21 \\ 28 \\ 36 \\ .. \end{bmatrix}$$

* Store level_ptr

Example of 2D-7 Point stencil

Step 2: Distance-k coloring

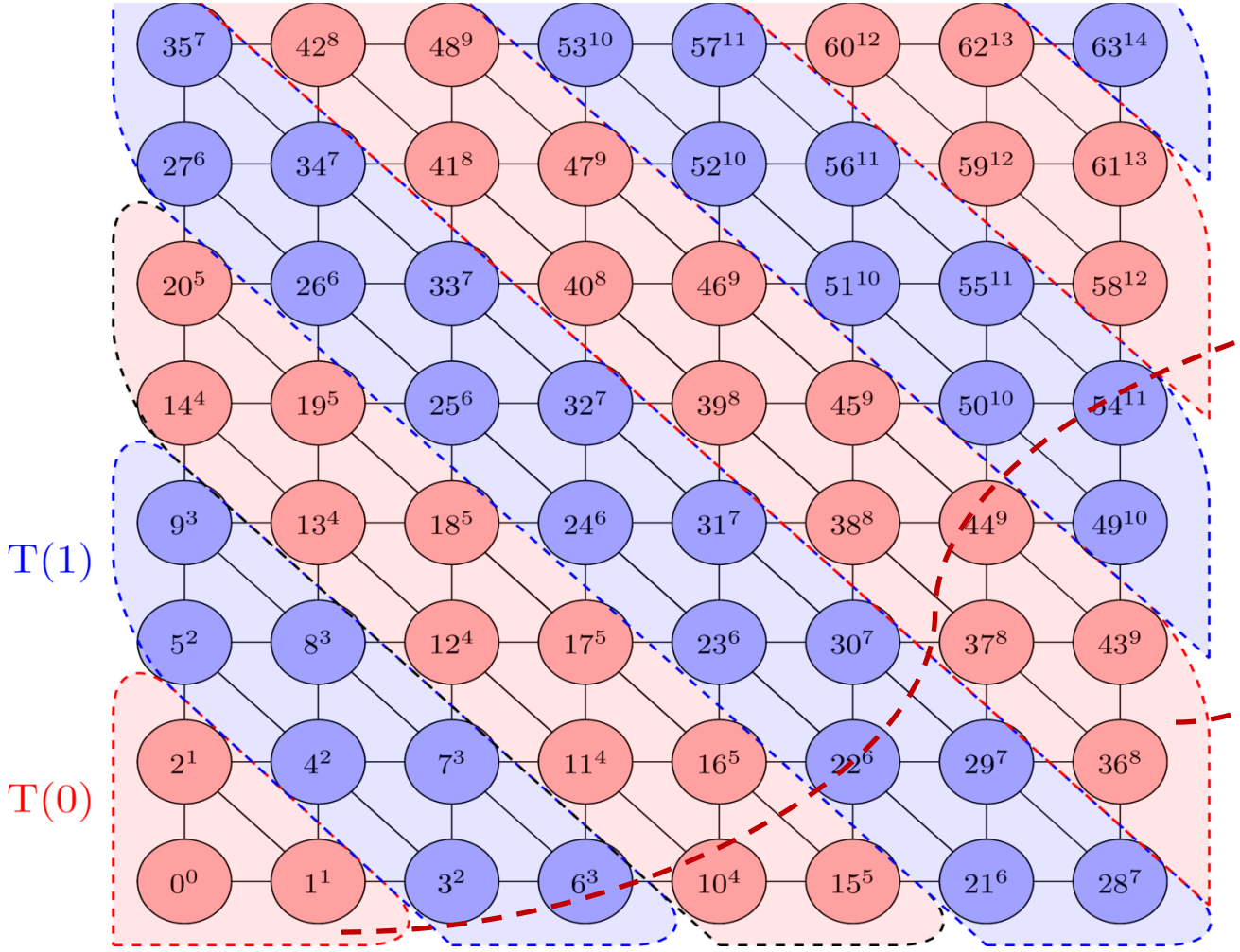




$L(n)$ *and* $L(n+k+i)$ $\forall\, i > 1$ are distance-k independent

Example of 2D-7 Point stencil for D2 coloring

But …



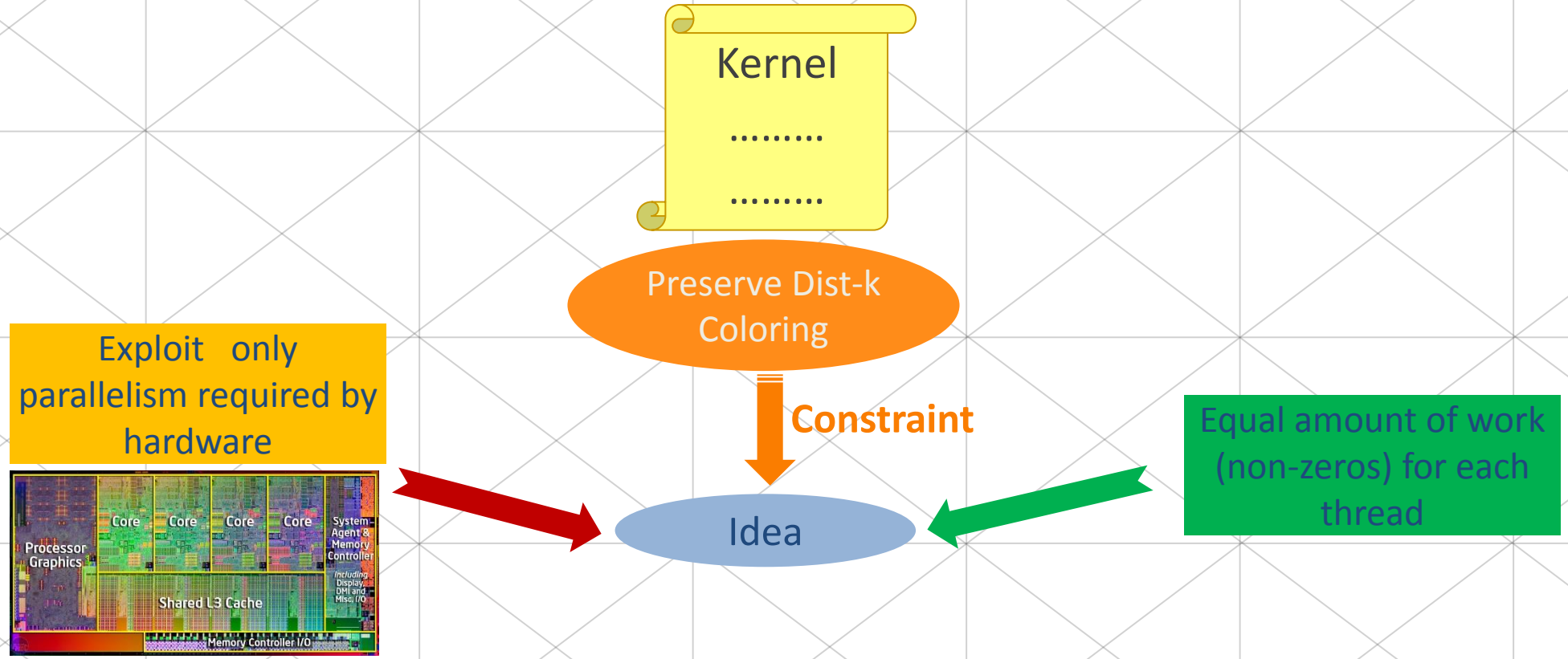
$n_r(T(0)) = 3$

$n_r(T(4)) = 13$

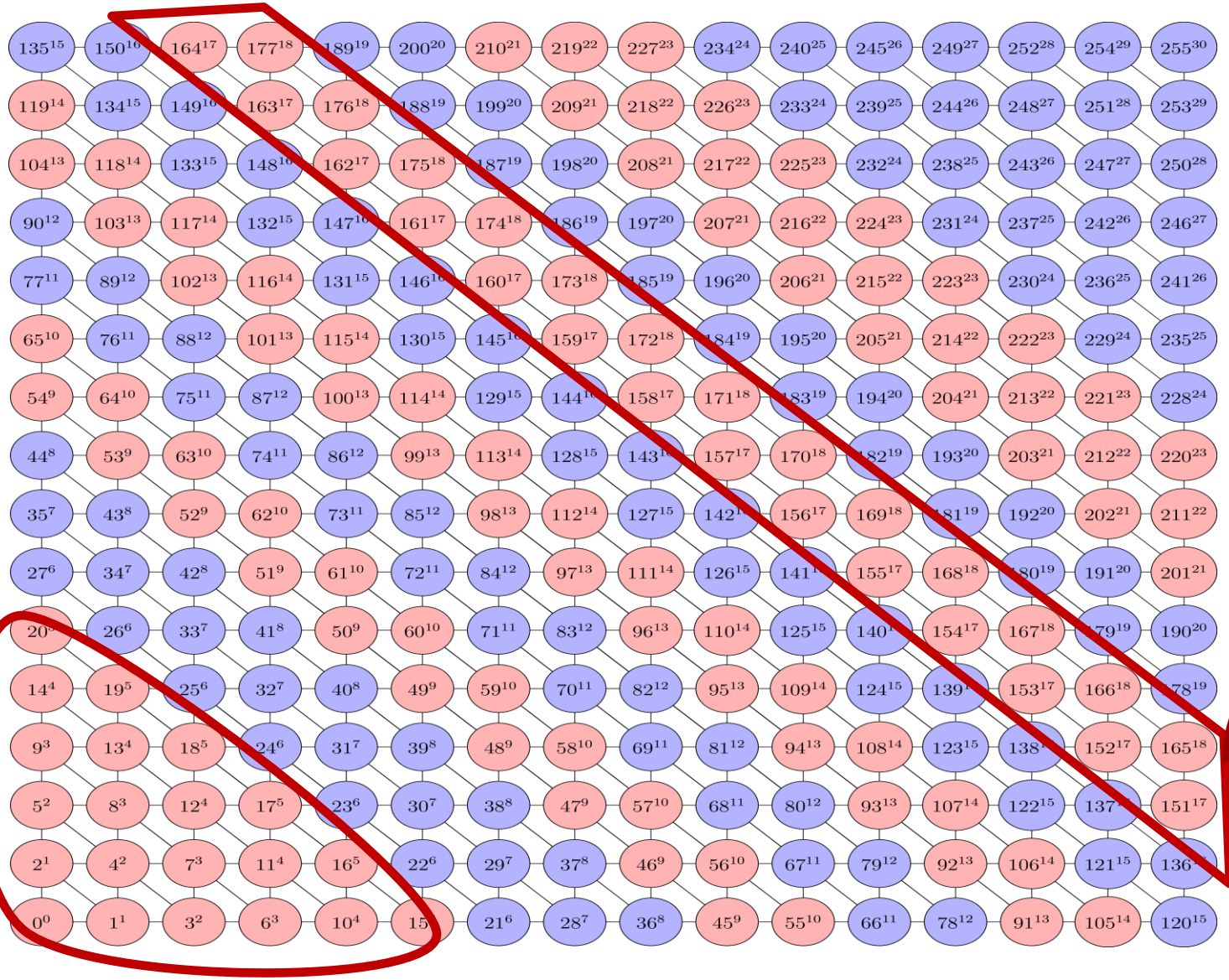
Load imbalance

Example of 2D-7 Point stencil for D2 coloring

Step 3: Load-balancing



Step 3: Load-balancing



5 threads

More levels since n_r on each levels are small

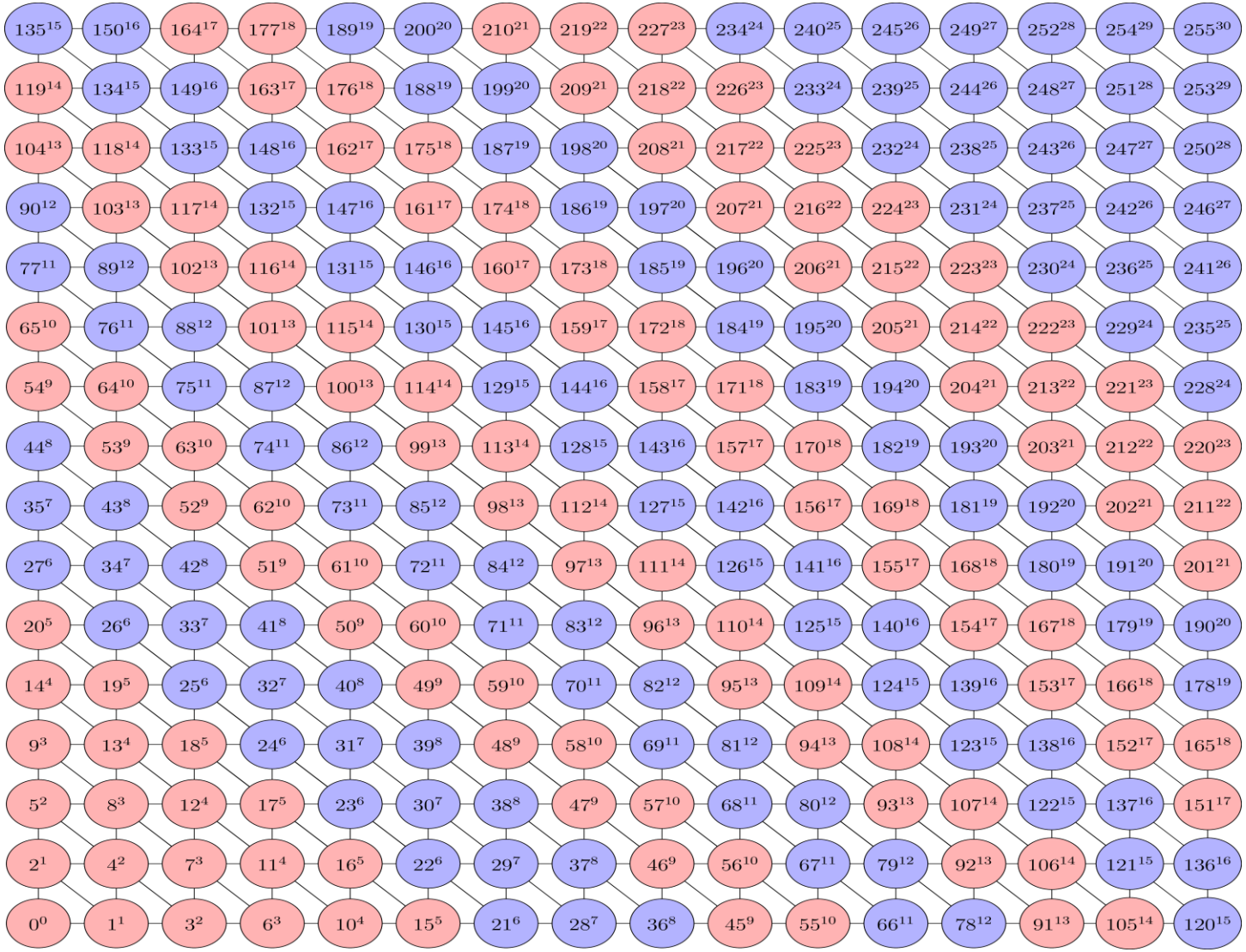
Just sufficient levels to maintain D2 independency

Example of 2D-7 Point stencil for D2 coloring

Recursion

Load imbalance

Parallelism limited by number of levels



8 threads

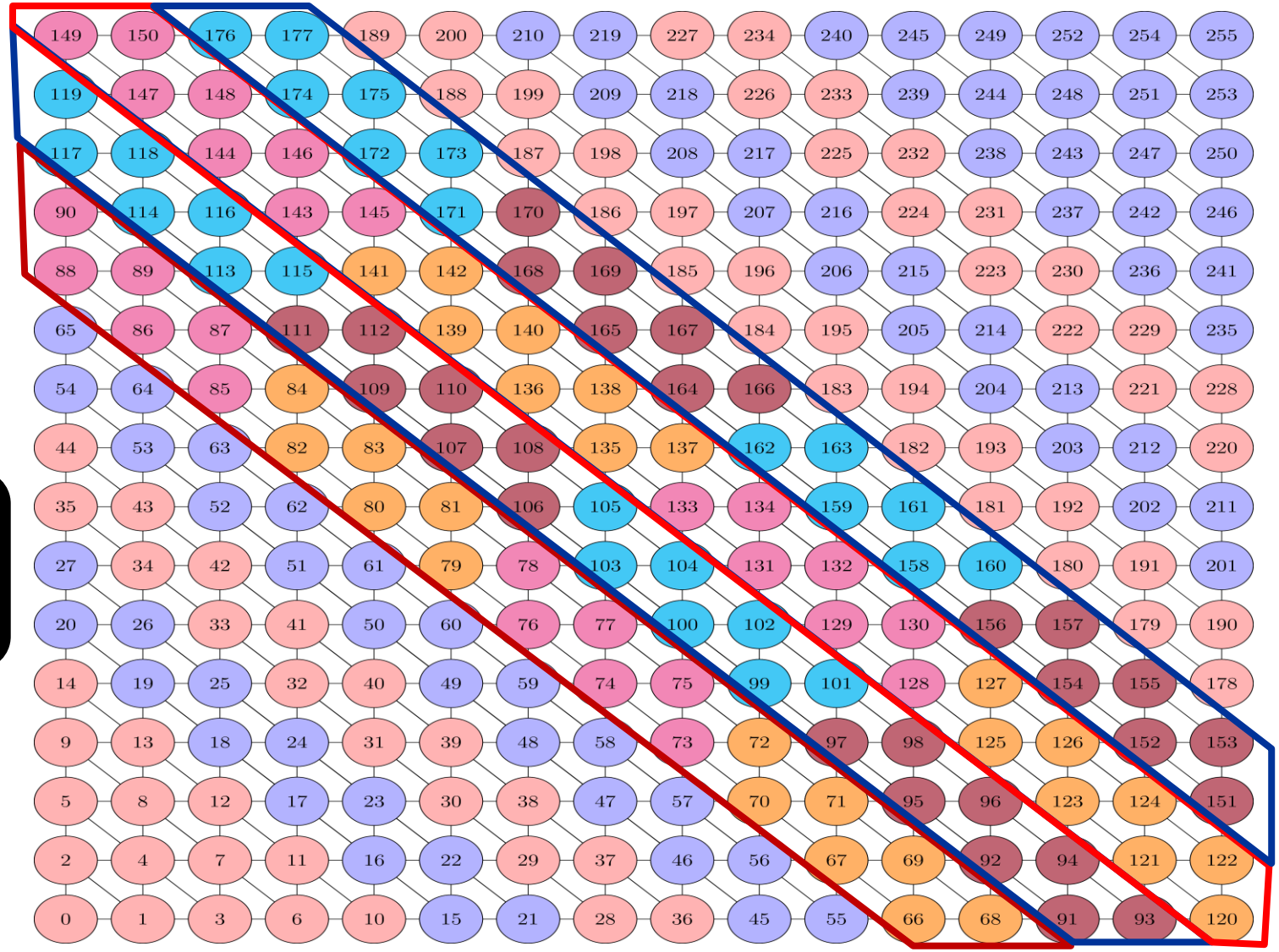
Solution: Find more parallelism using Recursion

Example of 2D-7 Point stencil for D2 coloring

Recursion

Need more parallelism

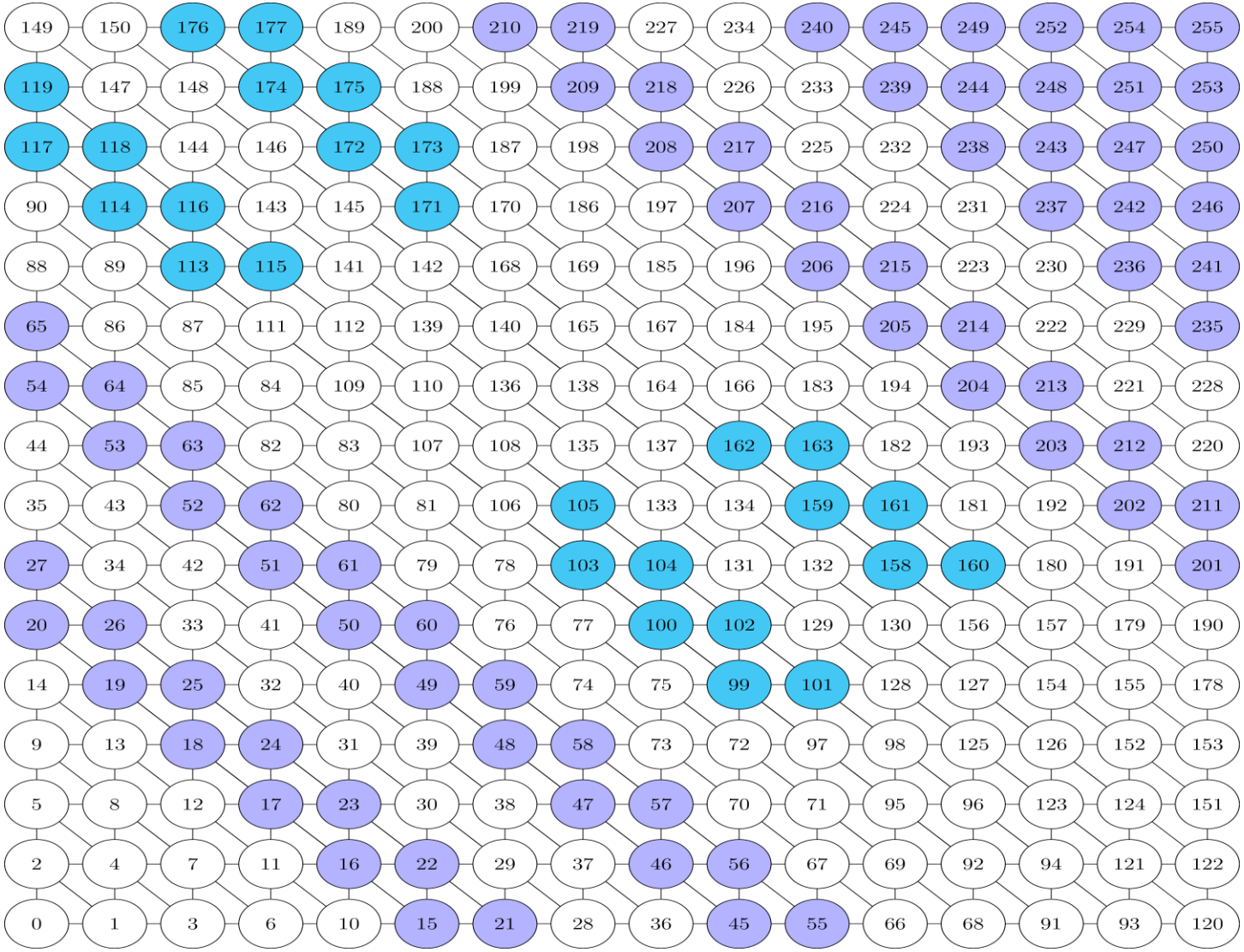
Apply recursion to selected sub-graph



Selection of sub-graph based on load-balancing

Example of 2D-7 Point stencil for D2 coloring for 8 threads

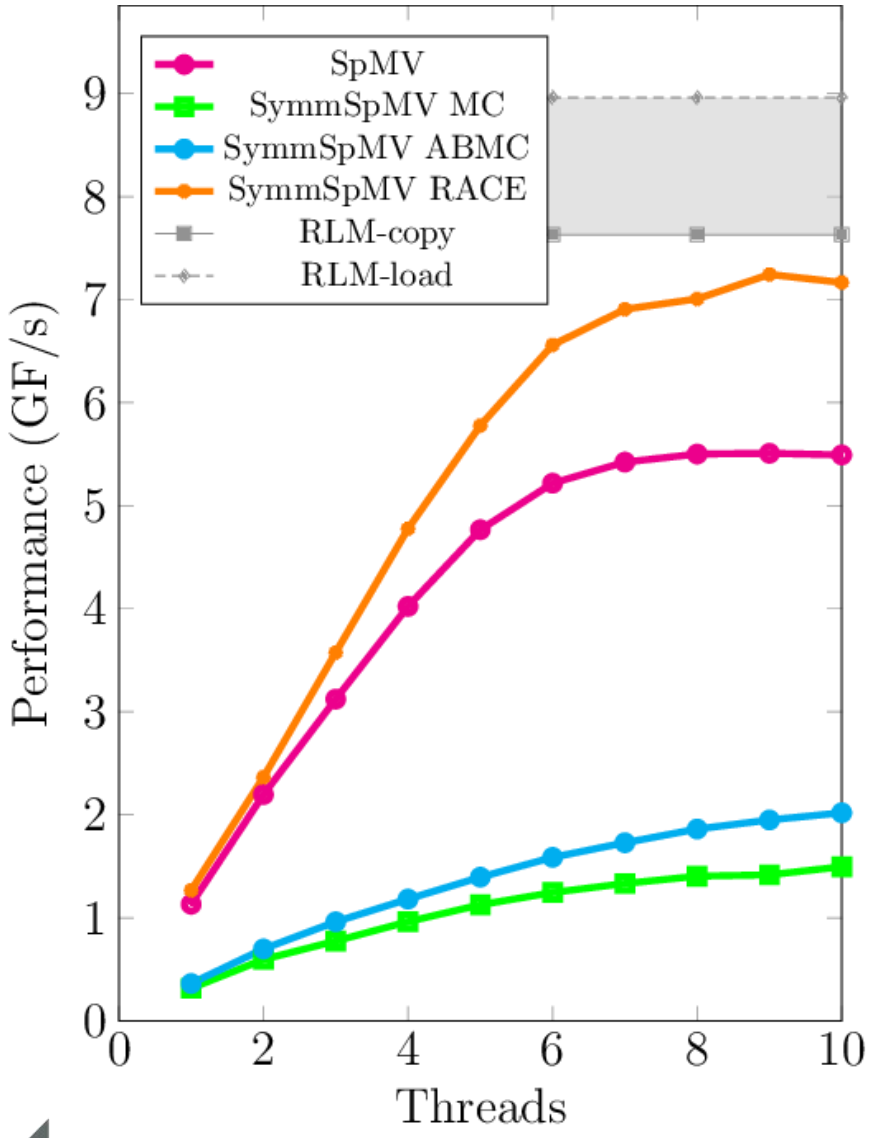
Recursion



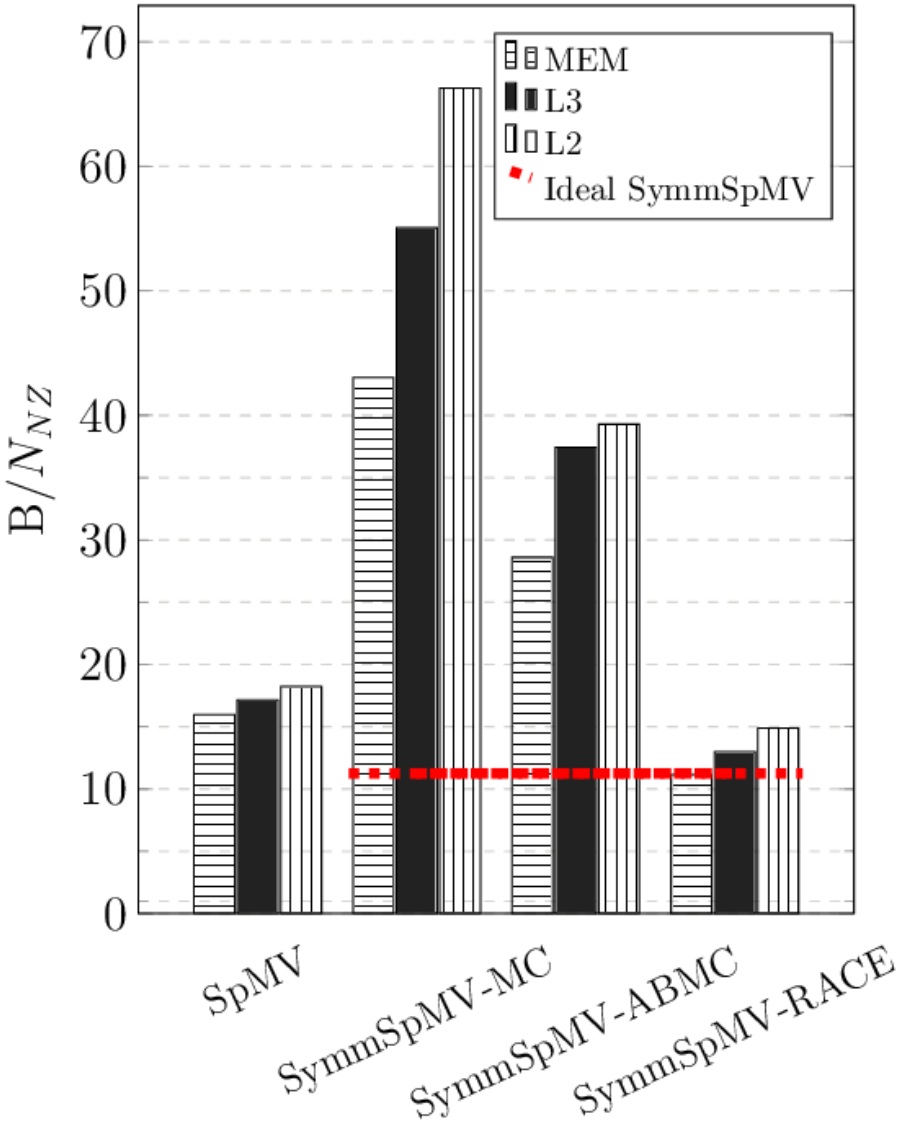
Example of 2D-7 Point stencil for D2 coloring for 8 threads

Performance

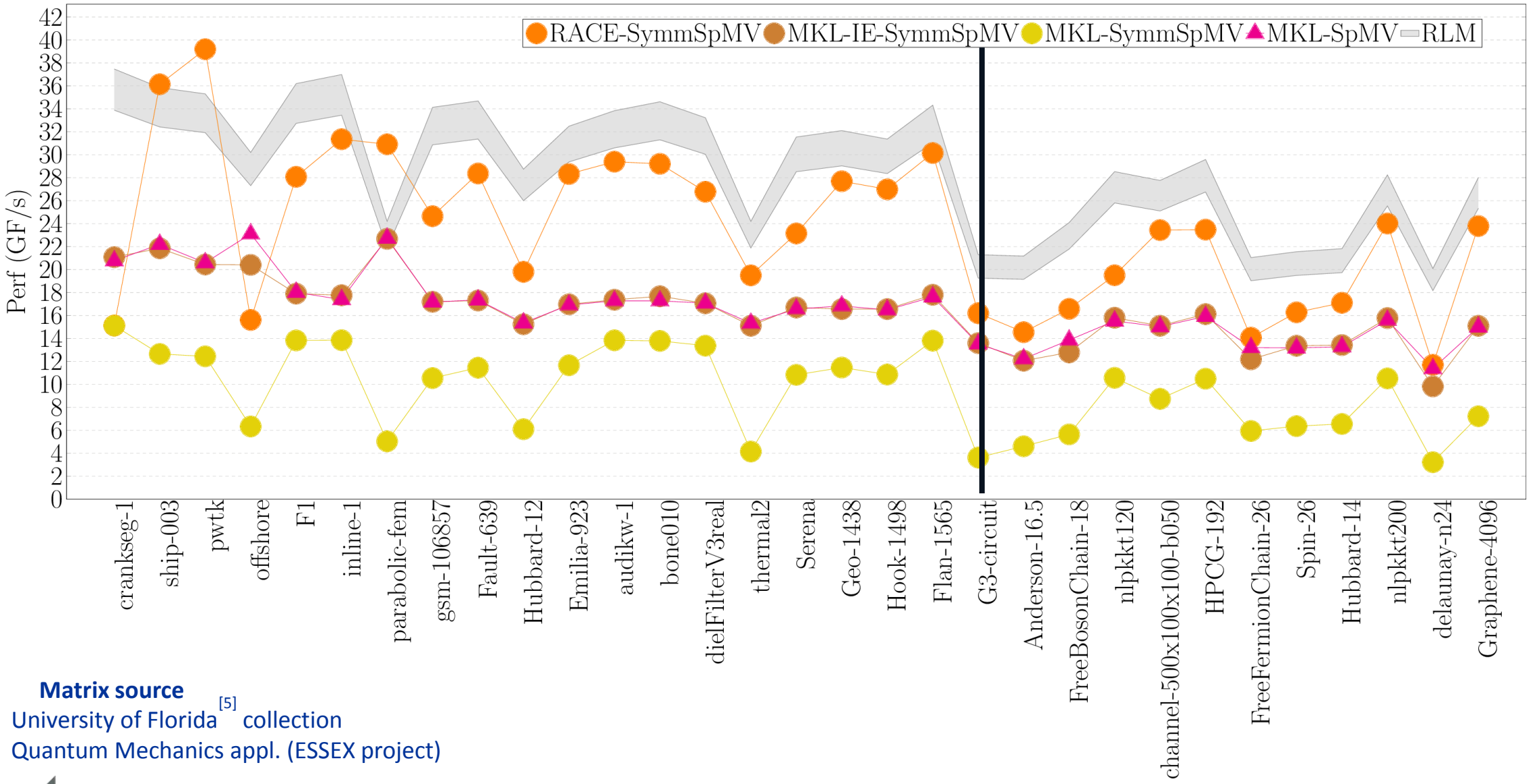
Ivy Bridge E5-2660 v2 @ 2.2 GHz



Performance



Performance Comparison with RLM and MKL– Skylake (20 threads)



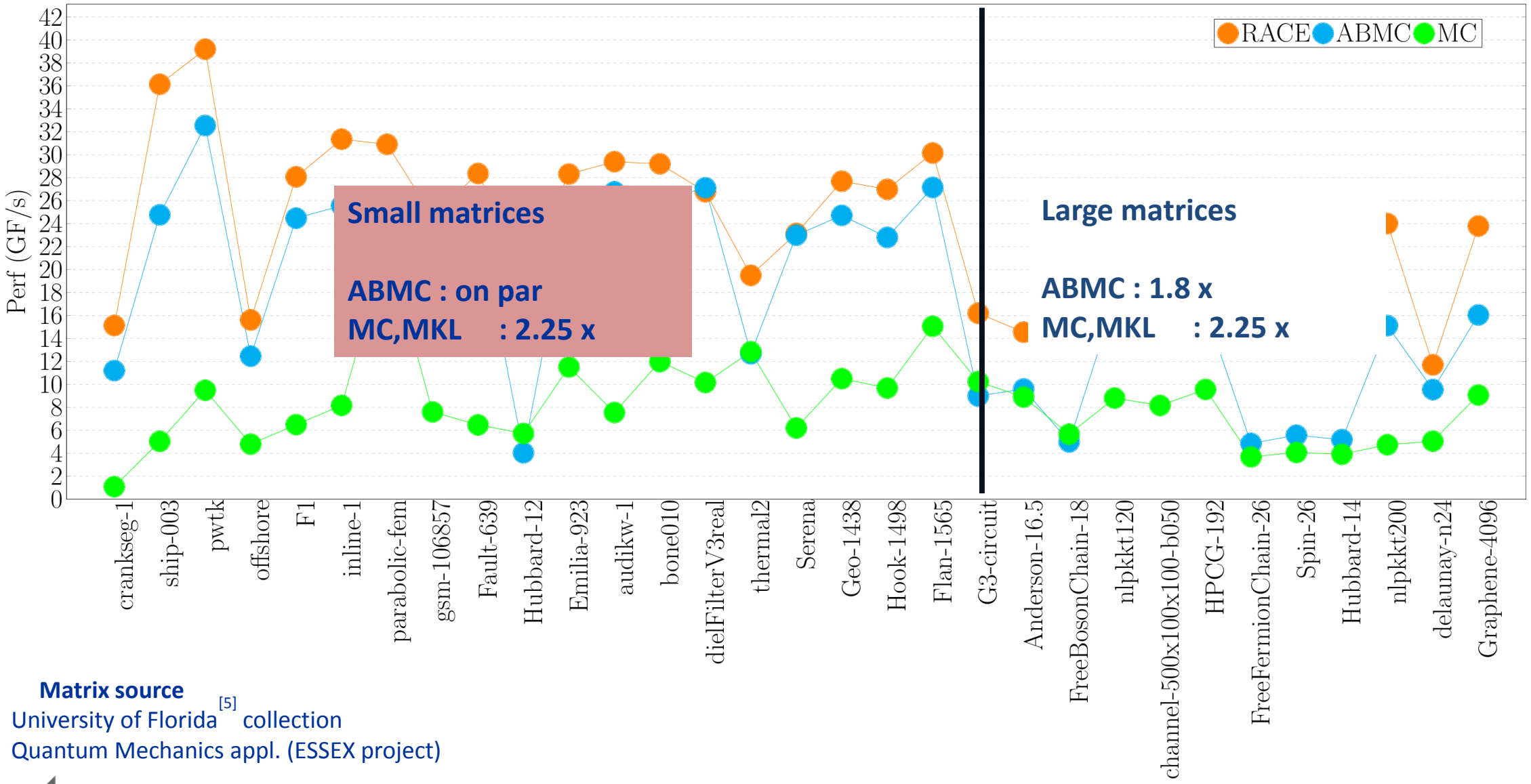
SymmSpMV

Matrix source

University of Florida[5] collection

Quantum Mechanics appl. (ESSEX project)

Performance Comparison with MC and ABMC– Skylake (20 threads)



SymmSpMV

Matrix source

University of Florida[5] collection

Quantum Mechanics appl. (ESSEX project)

Final remarks

- RACE also improves convergence of e.g. Gauß-Seidel and Kaczmarz compared to MC and ABMC (ongoing study)

- Light-weight software for setup and kernel application:
  https://bitbucket.org/ChristieAlappat/RACE-AD

- Several awards for Christie: SPPEXA best M.Sc. thesis 2017, SC18 ACM student research competition, 2nd place in 2019

- Preprint “Recursive Algebraic Coloring Technique for Hardware-Efficient Symmetric Sparse Matrix-Vector Multiplication“ (available on request)