TÜBİTAK

Sabancı Üniversitesi

EURO²

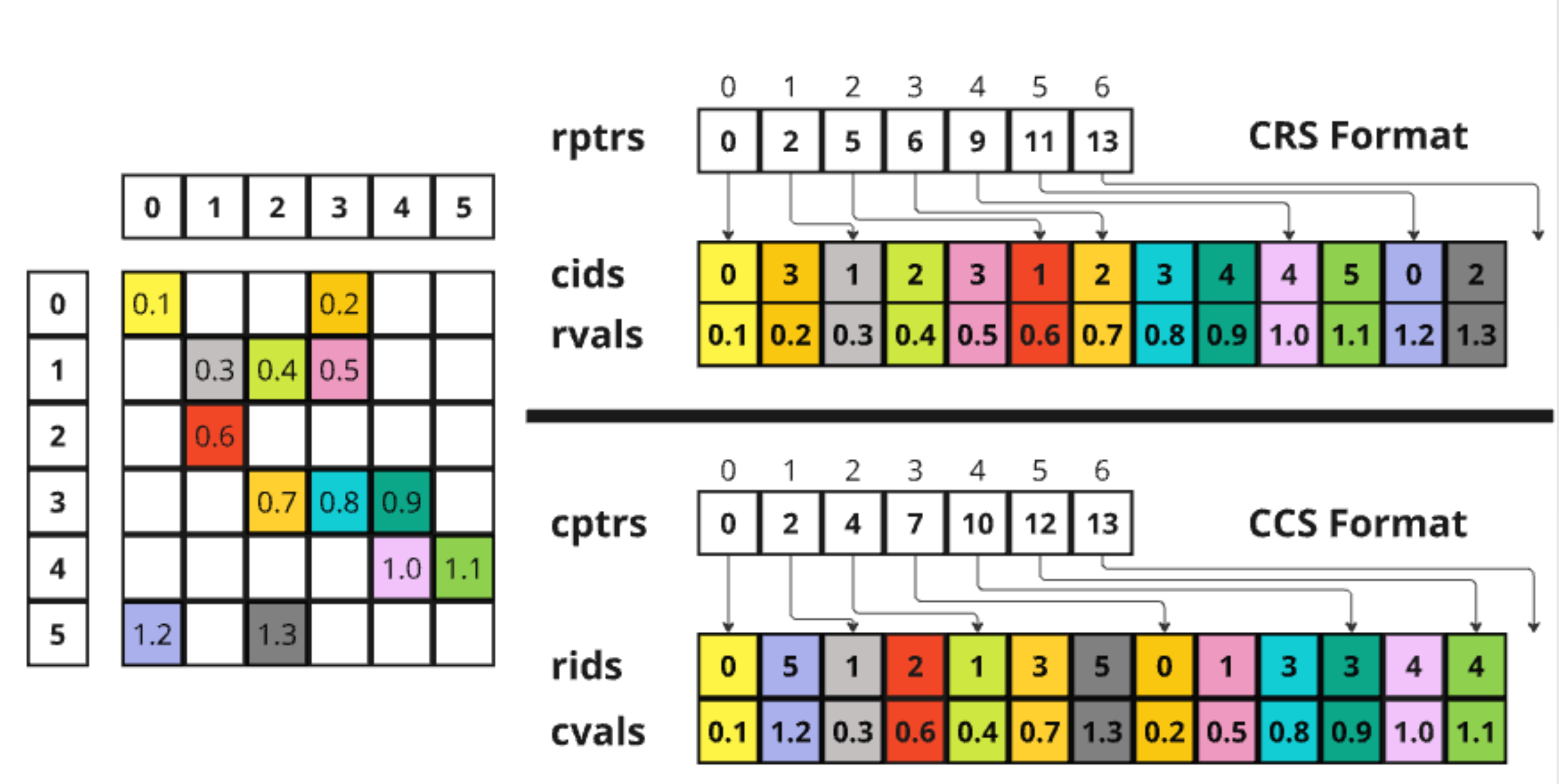
High Performance Computing with Sparse Data

Graphs, Matrices and Tensors

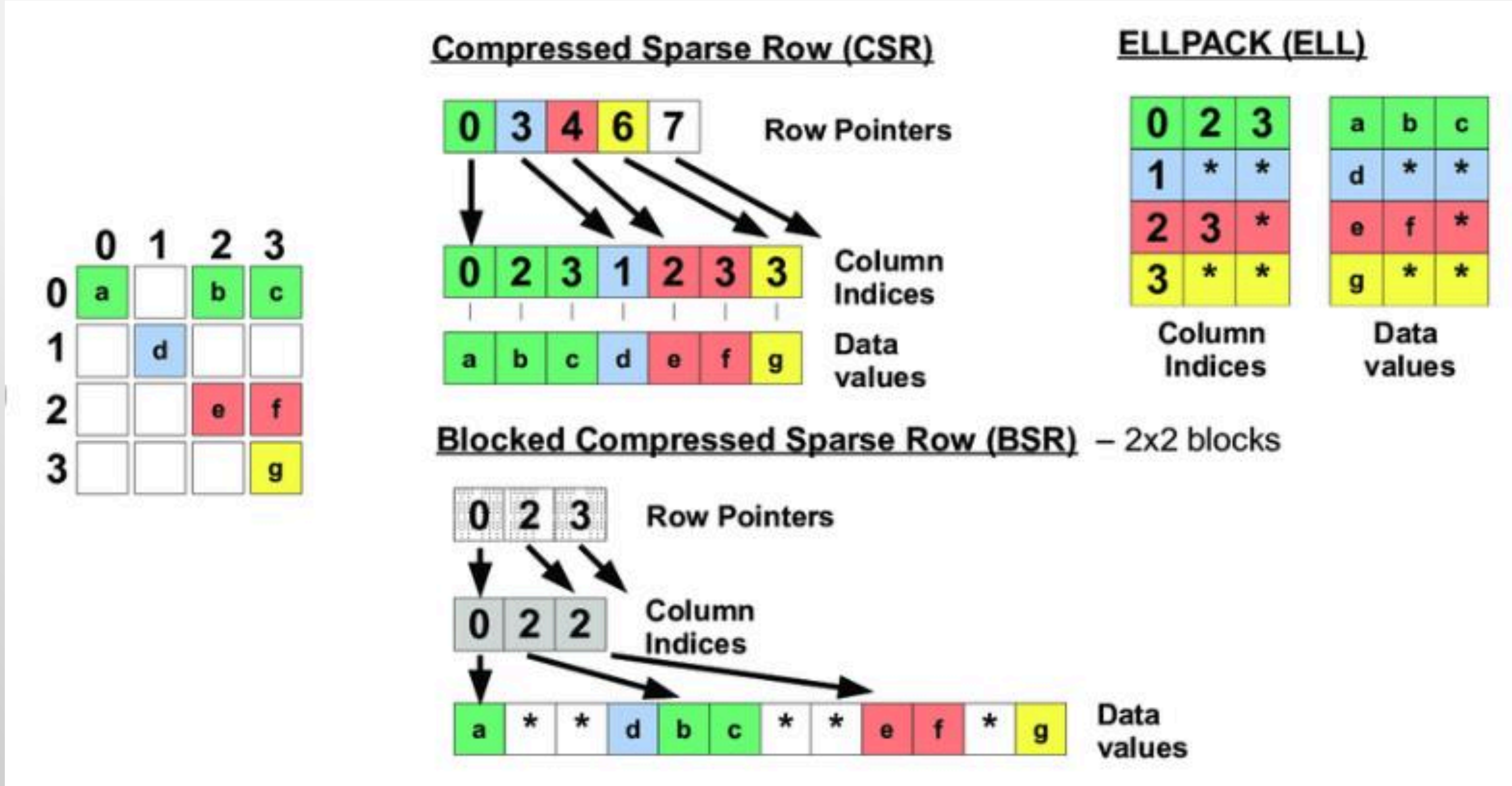
Kamer Kaya, Sabancı University

ncc@ulakbim.gov.tr

Sparse Matrix Data Structures

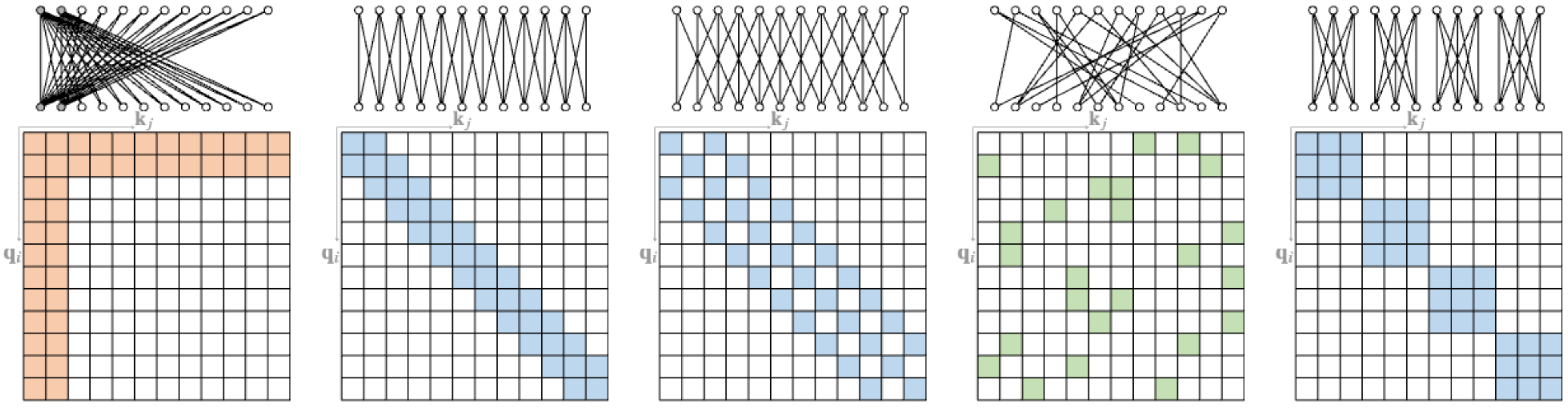


Sparse Matrix Data Structures



J. C. Pichel and B. Pateiro-López, "Sparse Matrix Classification on Imbalanced Datasets Using Convolutional Neural Networks," in *IEEE Access*, vol. 7, pp. 82377-82389, 2019, doi: 10.1109/ACCESS.2019.2924060

Sparse Matrices – Sparse Attention



(a) global (b) band (c) dilated (d) random (e) block local

Tianyang Lin, Yuxin Wang, Xiangyang Liu, Xipeng Qiu, A survey of transformers, AI Open, Volume 3, 2022

Sparse Matrices – Sparse Attention

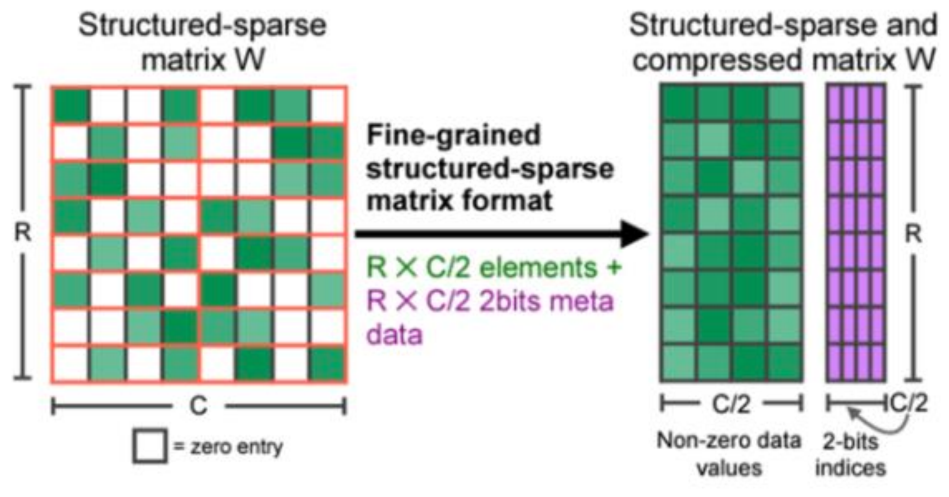


Figure 1. 2:4 structured sparsity pattern and compression

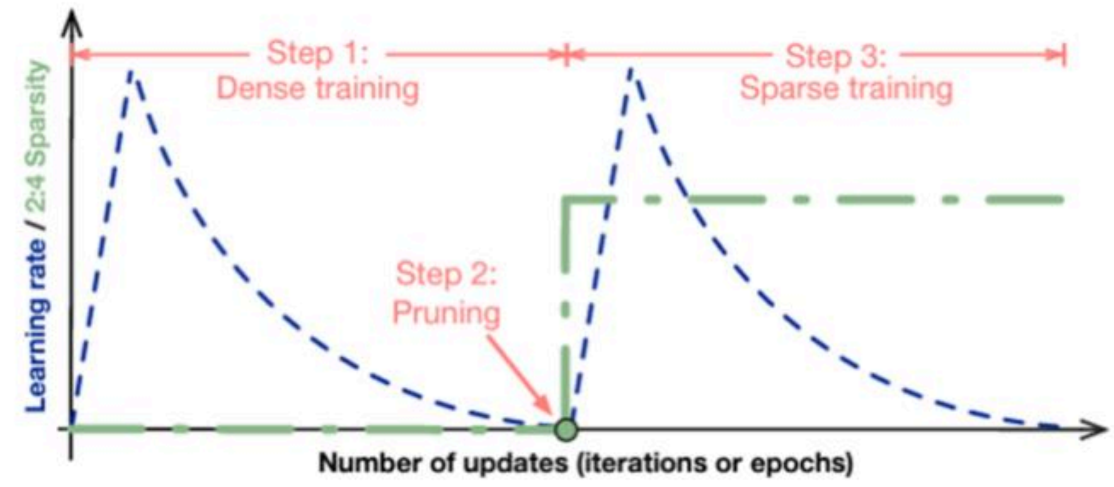


Figure 2. Basic training recipe

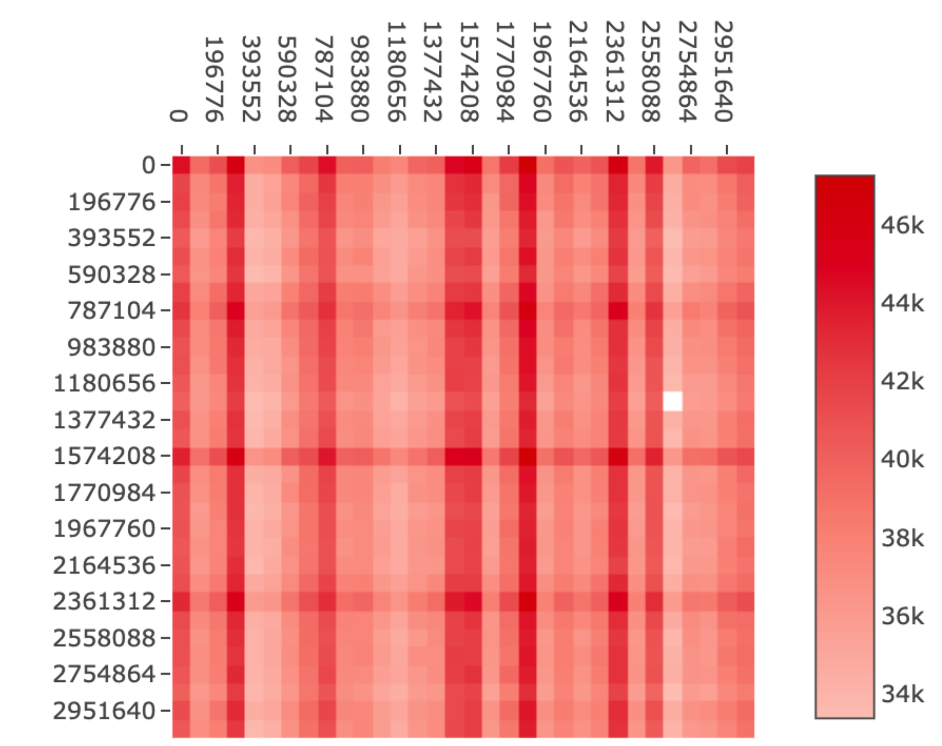
https://developer.nvidia.com/

```cpp
// Perform Sparse Matrix-Vector Multiplication (SpMV)
// A is stored in CSR format (values, row_ptr, col_idx)
// x is the input dense vector
// y is the output dense vector
void spmv_csr(const std::vector<double>& values, const std::vector<int>& row_ptr,
              const std::vector<int>& col_idx, const std::vector<double>& x,
              std::vector<double>& y) {
    // Check dimensions
    assert(row_ptr.size() > 1 && "row_ptr must have at least two elements.");
    assert(x.size() > 0 && "Input vector x must not be empty.");

    int num_rows = row_ptr.size() - 1;  // Number of rows in the sparse matrix

    // Resize the output vector to match the number of rows in the matrix
    y.resize(num_rows, 0.0);

    // Perform SpMV
    for (int i = 0; i < num_rows; ++i) {
        double sum = 0.0;
        for (int j = row_ptr[i]; j < row_ptr[i + 1]; ++j) {
            sum += values[j] * x[col_idx[j]];
        }
        y[i] = sum;
    }
}
```

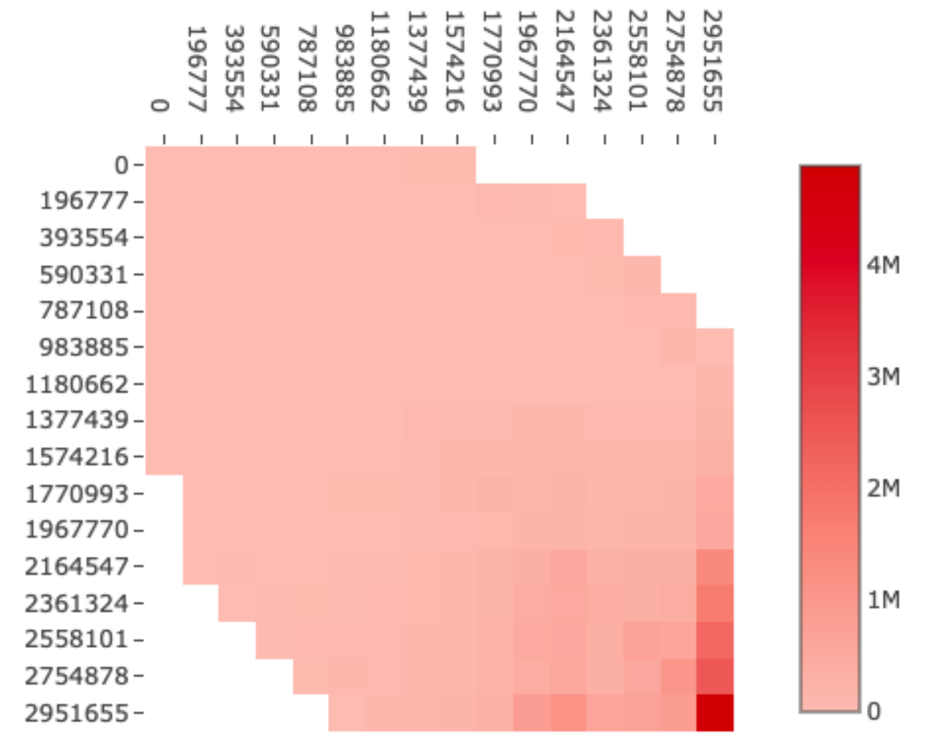
EURO²

SpMV

Sparse Matrix Ordering

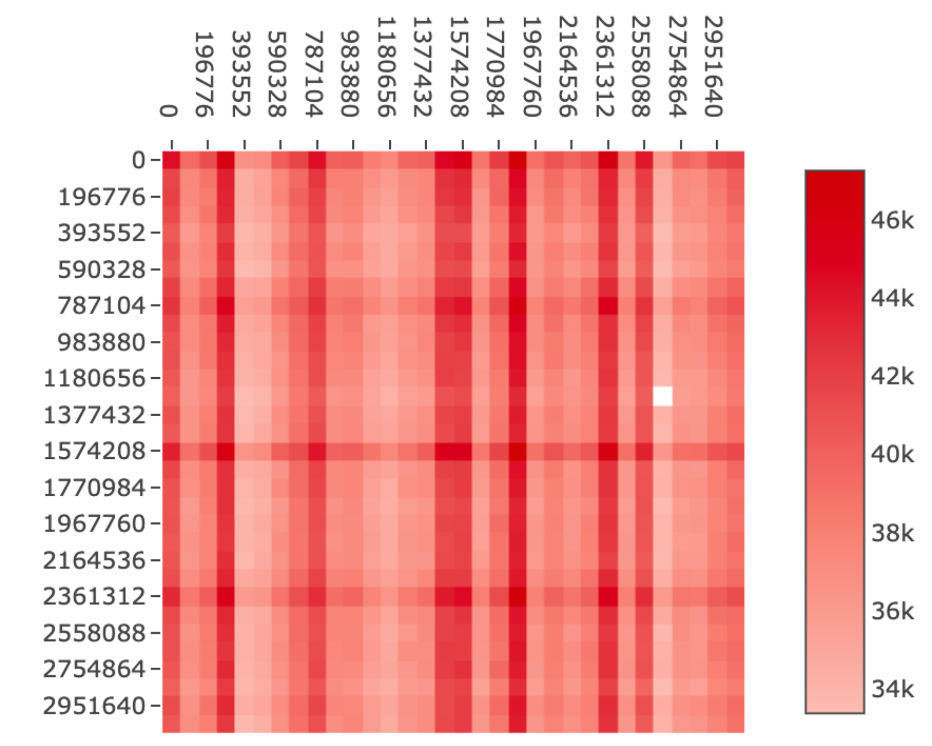


KERNEL EXECUTION TIMES						
Name	Scheduling	Chunk	1	8	15	30
SPMVRowBased	dynamic	256	0.343	0.086	0.065	0.038



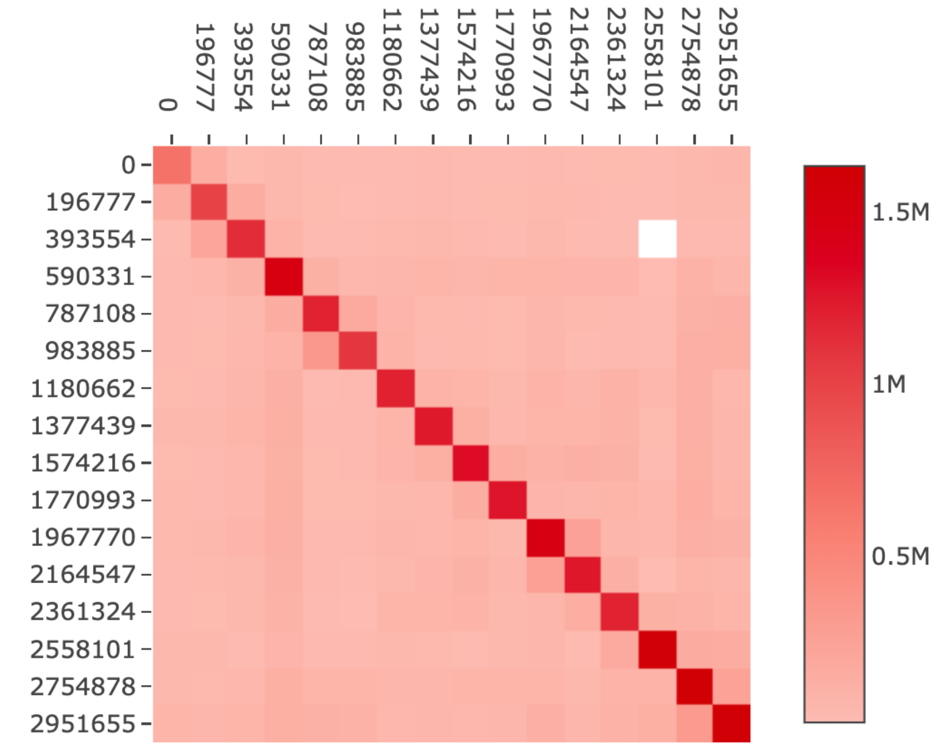
KERNEL EXECUTION TIMES						
Name	Scheduling	Chunk	1	8	15	30
SPMVRowBased	dynamic	256	0.354	0.146	0.087	0.086

Sparse Matrix Ordering



KERNEL EXECUTION TIMES

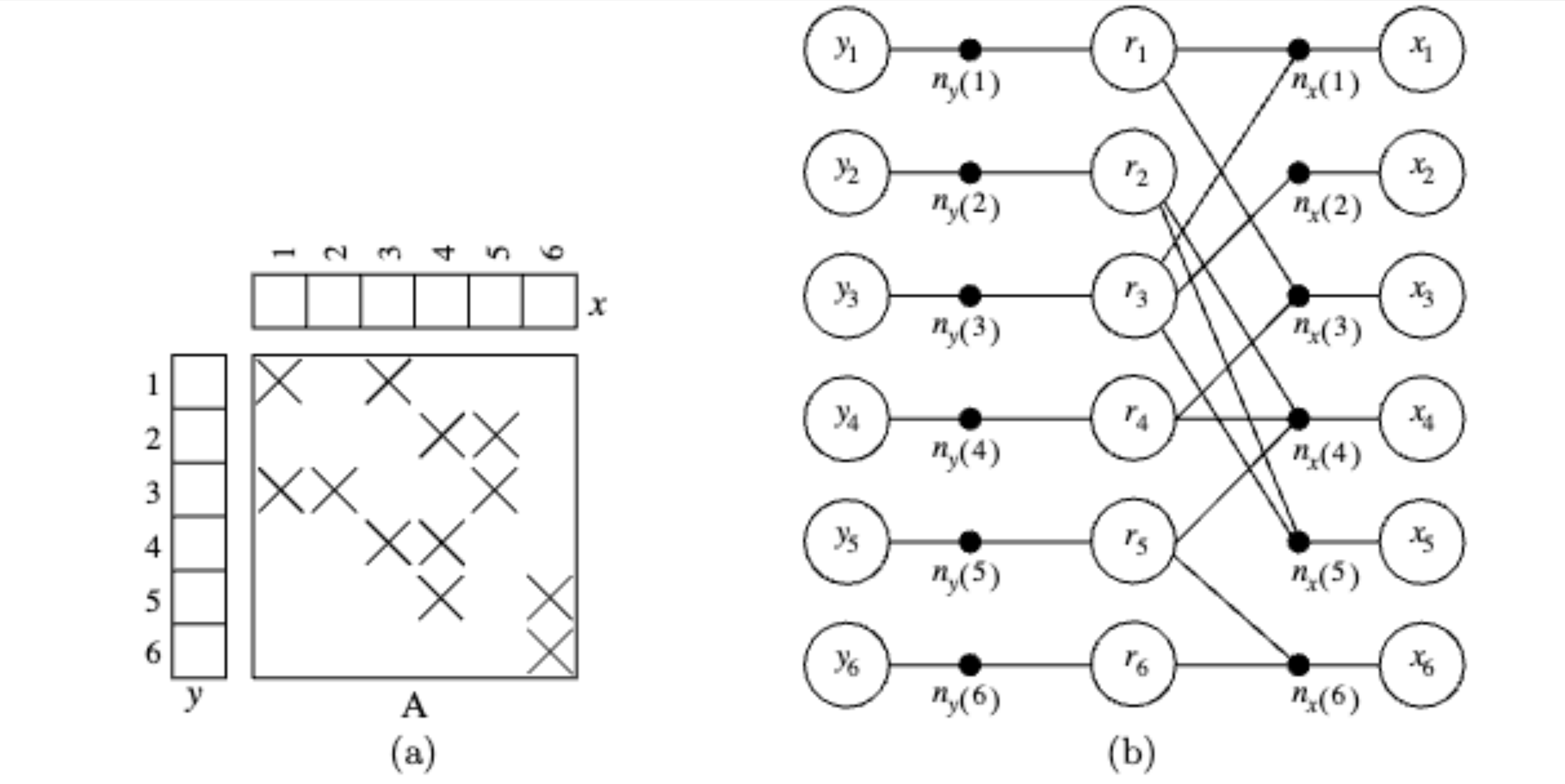
Name	Scheduling	Chunk	1	8	15	30
SPMVRowBased	dynamic	256	0.343	0.086	0.065	0.038



KERNEL EXECUTION TIMES

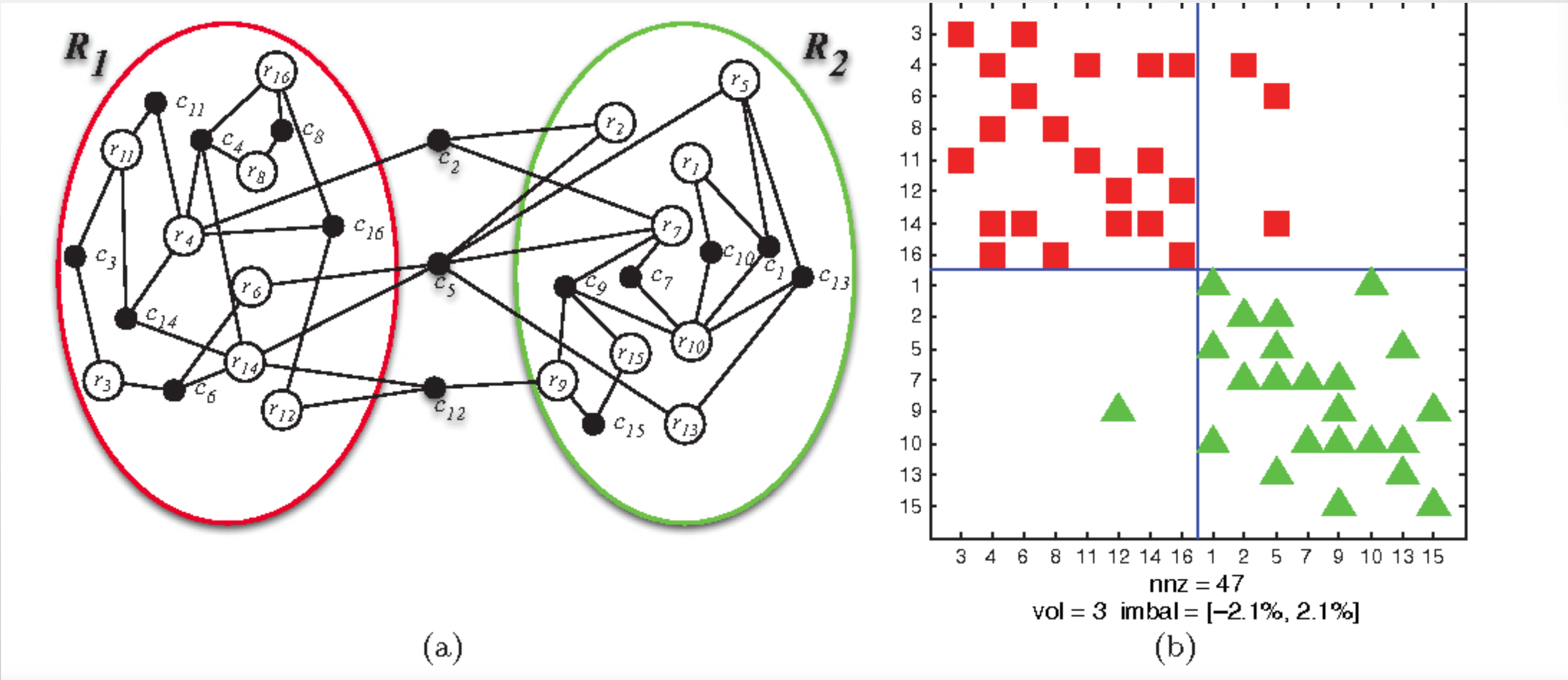
Name	Scheduling	Chunk	1	8	15	30
SPMVRowBased	dynamic	256	0.361	0.057	0.030	0.022

Sparse Matrix Partitioning



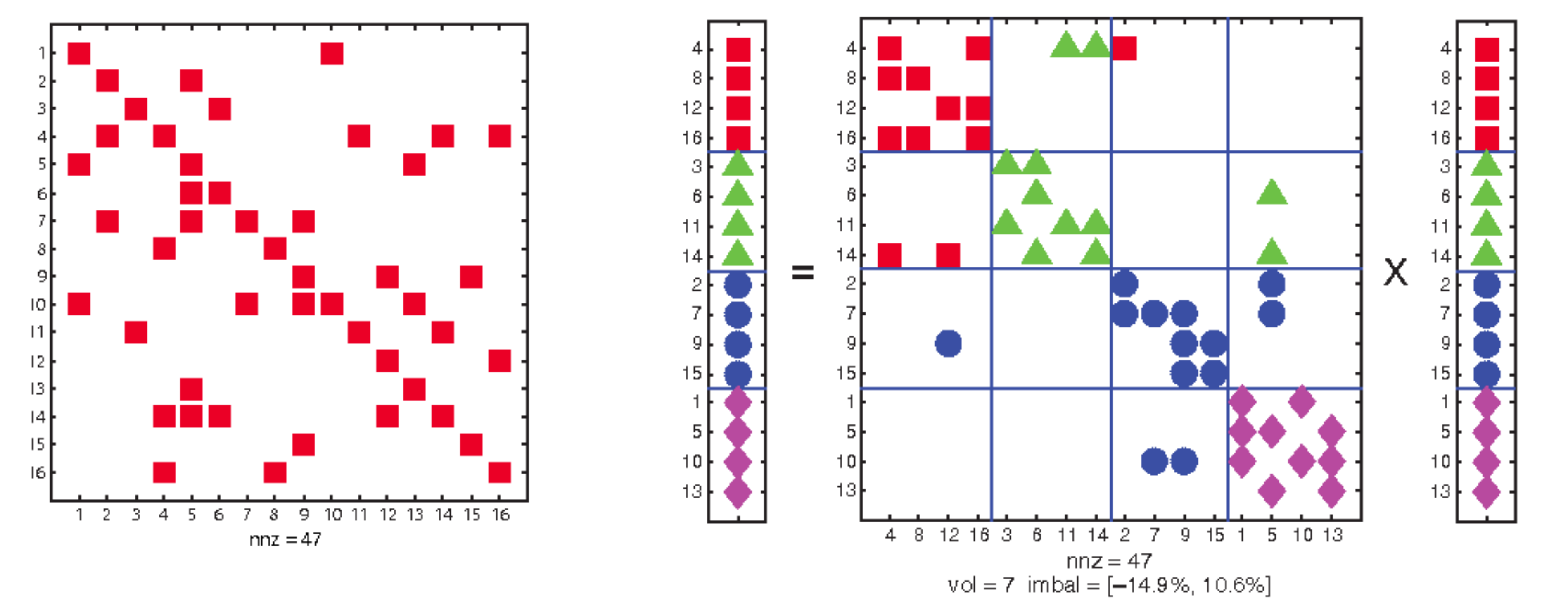
Bora Uçar, Cevdet Aykanat. Revisiting Hypergraph Models for Sparse Matrix Partitioning. SIAM Review, 2007, 49 (4), pp.595–603. 10.1137/060662459. hal-00803507

Sparse Matrix Partitioning



SIAM Journal on Scientific Computing 2010 On Two-Dimensional Sparse Matrix Partitioning: Models, Methods, and a Recipe Ümit V. Çatalyürek, C. Aykanat, B. Uçar

Sparse Matrix Partitioning



SIAM Journal on Scientific Computing 2010 On Two-Dimensional Sparse Matrix Partitioning: Models, Methods, and a Recipe Ümit V. Çatalyürek, C. Aykanat, B. Uçar

Sparse Matrix Partitioning

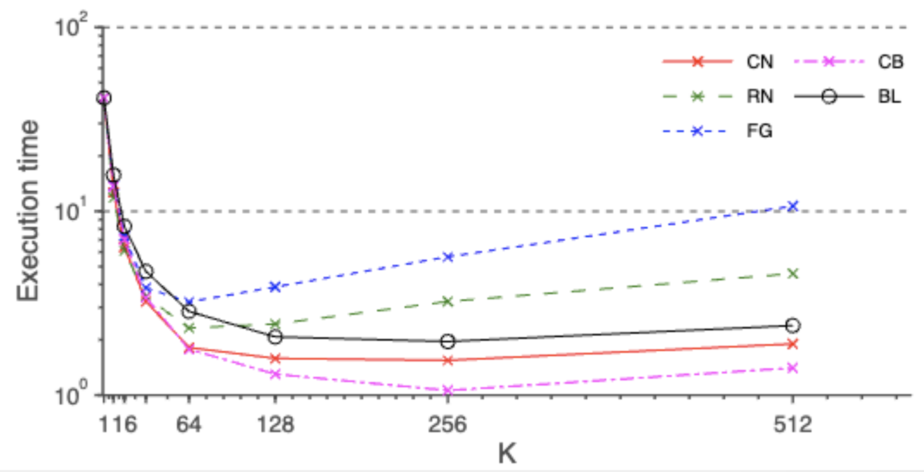
Metric	$8 \le K \le 64$ SpMV	PETSc	Trilinos	$128 \le K \le 512$ SpMV	PETSc	Trilinos
MaxNnz	8.02	7.81	6.80	0.49	0.44	0.83
TotVol	0.18	0.38	1.00	0.39	0.36	1.06
MaxSV	1.66	1.53	2.20	0.00	0.00	0.11
TotMsg	0.15	0.28	0.00	7.90	8.03	4.51
MaxSM	0.00	0.00	0.00	1.22	1.18	3.49

Table 2: Regression analysis of SpMV, PETSc and Trilinos with all matrices and models CN and BL.

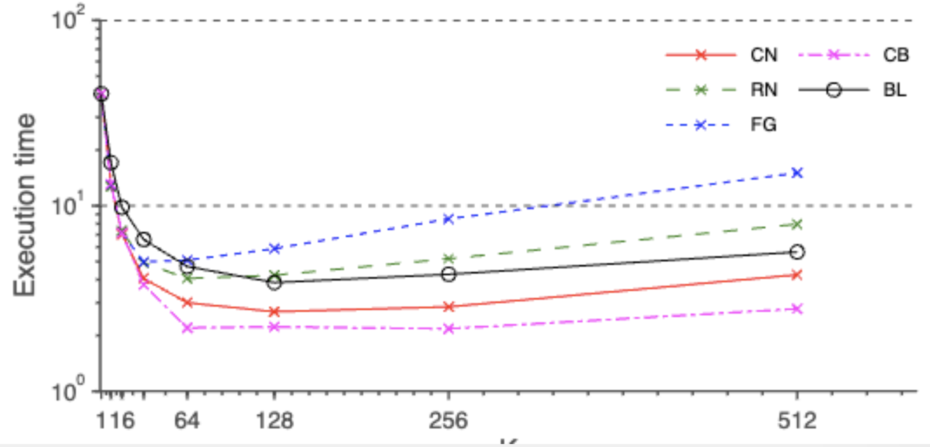
Kamer Kaya, Bora Uçar, Umit Catalyurek. Analysis of Partitioning Models and Metrics in Parallel Sparse Matrix-Vector Multiplication, 10th PPAM - Parallel Processing and Applied Mathematics, Sep 2013, Varsovie, Poland. Springer, pp.174–184, 2014

Sparse Matrix Partitioning

SpMV



Trilinos



Thanks

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia