Performance Analysis of Compressed Matrix Multiplication

Radu Ionut Barbalata - 2002688
Advanced Architecture
Sapienza University of Rome

Table of contents

Matrix Formats

- Classic Format
- ELL Format
- HYB Format
- BSR Format

Classic Format

Before

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

After

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

ELL Format

Before

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

After

1	2	*	*
3	4	*	*
5	*	*	*
6	7	8	9

Values

0	1	*	*
1	3	*	*
3	*	*	*
0	1	2	3

Columns

HYB Format

Before

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

After (ELL part)

1	2
3	4
5	*
6	7

Values

0	1
1	3
3	*
0	1

Columns

HYB Format (Part II)

Before

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

After (COO part)

Values	8	9
Rows	3	3
Columns	2	3

BSR Format

Before

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

After

1	2	0	3
0	0	0	4
0	0	6	7
0	5	8	9

Values

0
0
1
1

Rows

0
0
1
1

Columns

BSR Format

Before

1	2	0	0
0	3	0	4
0	0	0	5
6	7	8	9

After

Values			
1	2	0	3
0	0	0	4
0	0	6	7
0	5	8	9

Rows
0
0
1
1

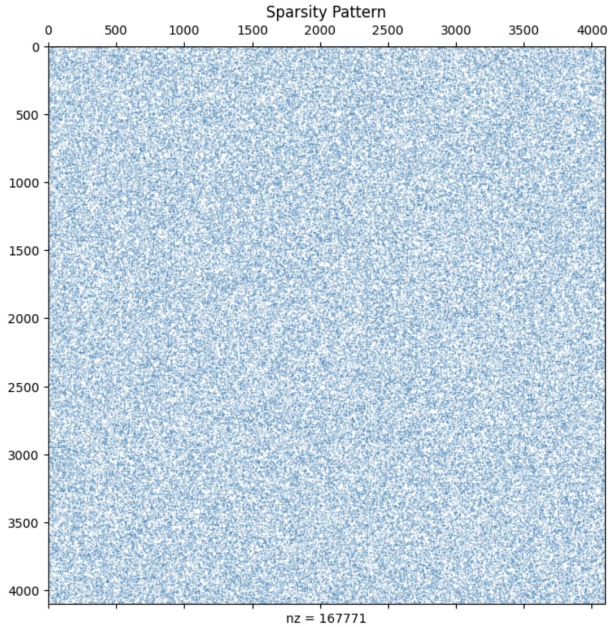
Columns
0
1
1
1

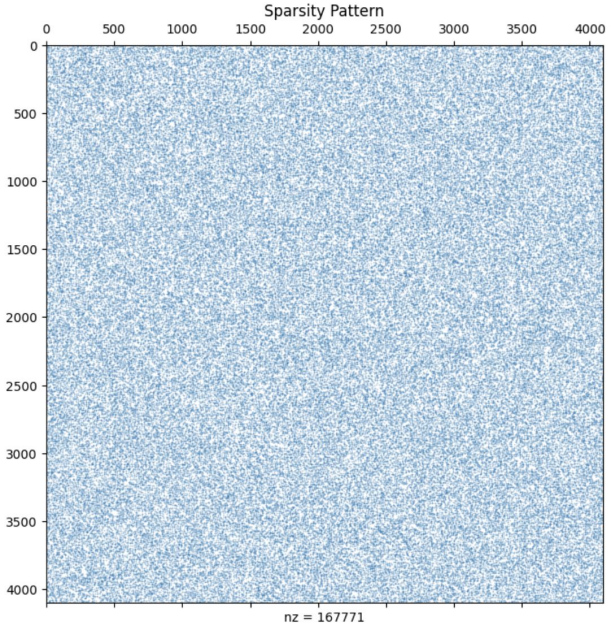
Test SetUp

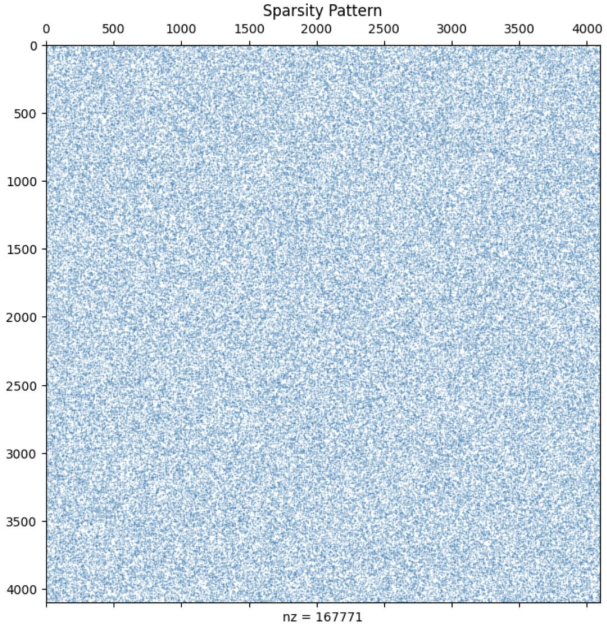
- 4 Matrices
- 16 Combinations
- 3 Iteration per Combination
- 48 Data Points for each Matrix Format
- 3 Matrices Value Patterns (Random, Banded, Block)
- 6 Matrix Compression Formats (BSR 4, BSR 8, BSR 16, CLASSIC, ELL, HYB)

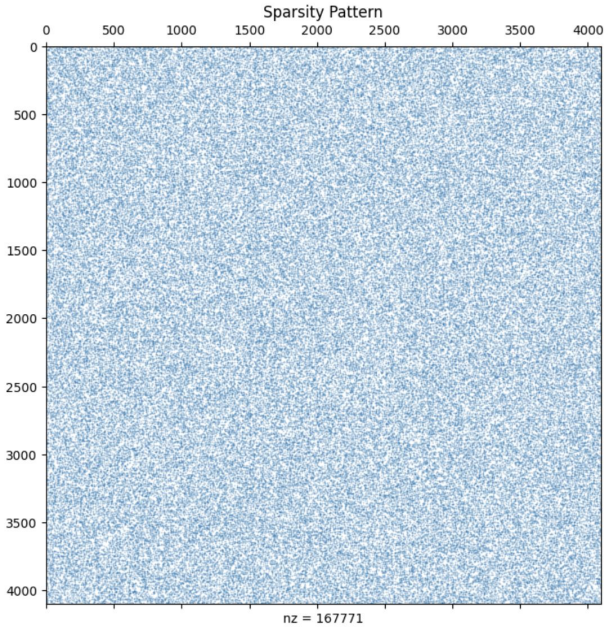
Random Matrices

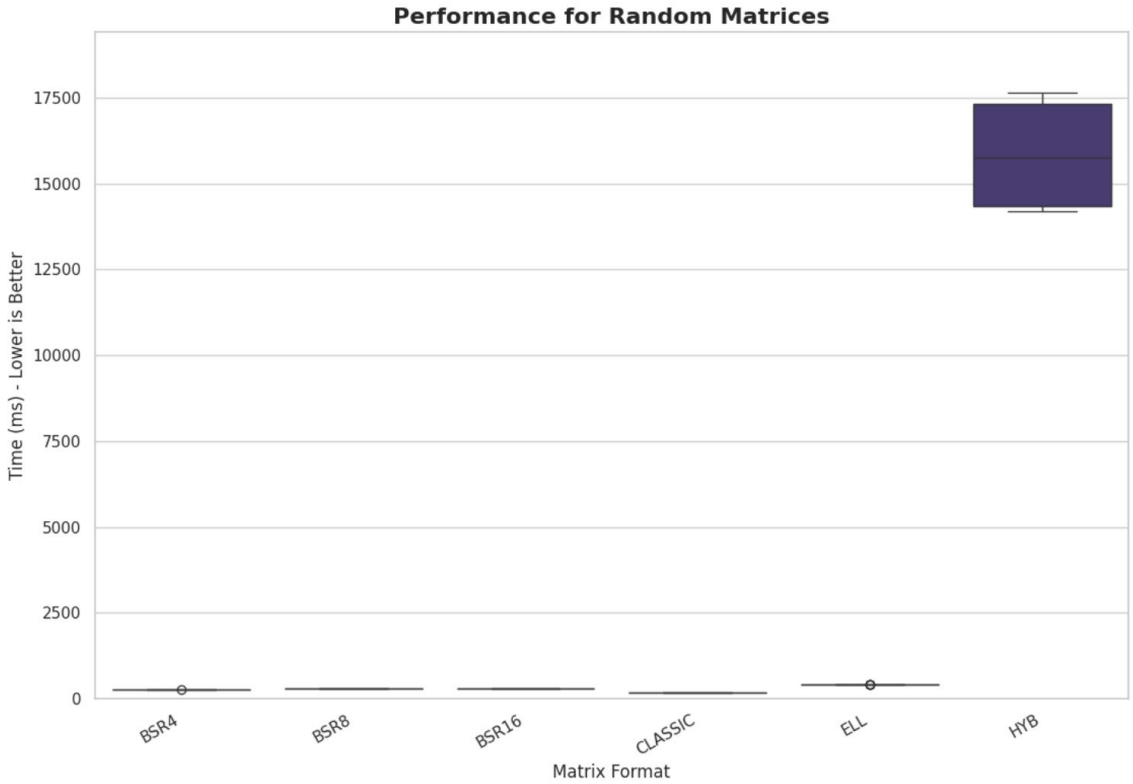
- Rows = 4096
- Columns = 4096
- percentage of NNZ = 1%

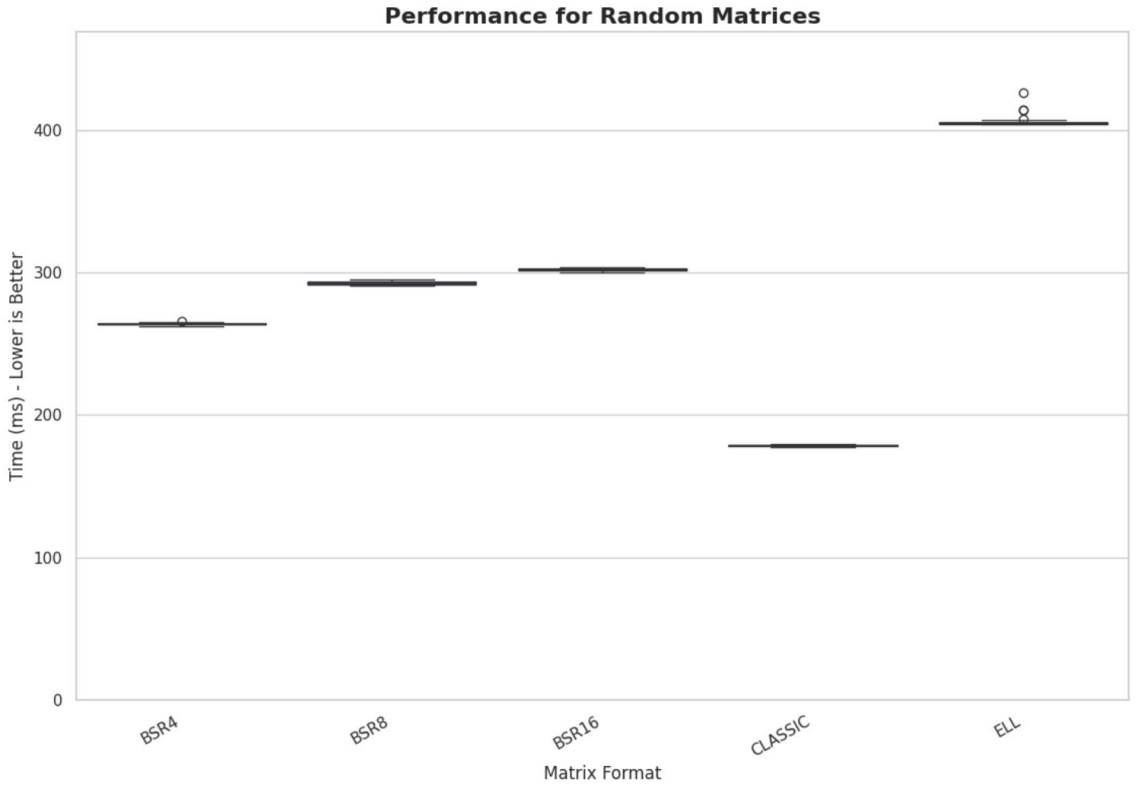

Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 167771


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 167771


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 167771


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 167771


Performance for Random Matrices
Time (ms) - Lower is Better
17500
15000
12500
10000
7500
5000
2500
0
BSR4
BSR8
BSR16
CLASSIC
ELL
HYB
Matrix Format

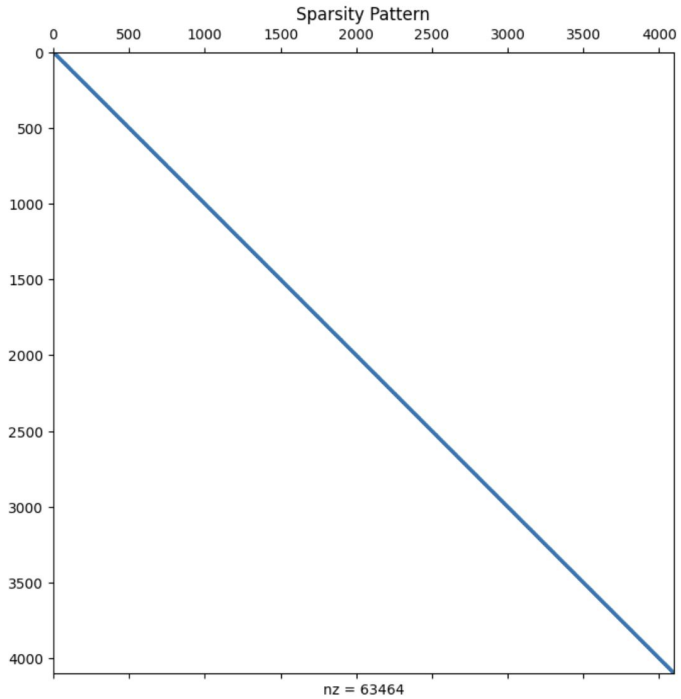

Performance for Random Matrices
Time (ms) - Lower is Better
0
100
200
300
400
BSR4
BSR8
BSR16
CLASSIC
ELL
Matrix Format

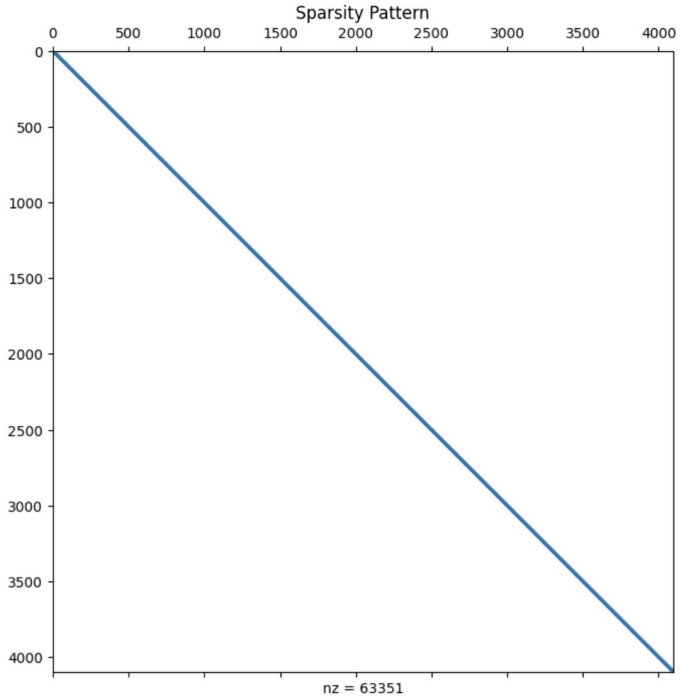
Random Matrices Results

	Total (ms)	Allocation (ms)	Mult (ms)	Free (ms)
BSR 4	390.55	111.40	263.80	15.57
BSR 8	431.37	121.07	292.59	17.11
BSR 16	453.79	134.80	302.54	17.19
CLASSIC	386.73	137.80	178.91	70.33
ELL	581.36	108.27	405.04	67.77
HYB	15890.80	106.34	15764.20	14.27

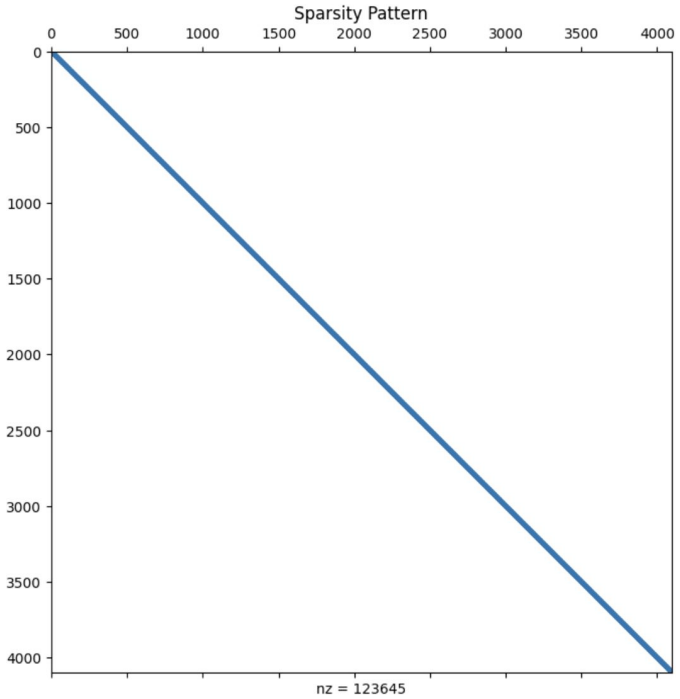
Banded Matrices

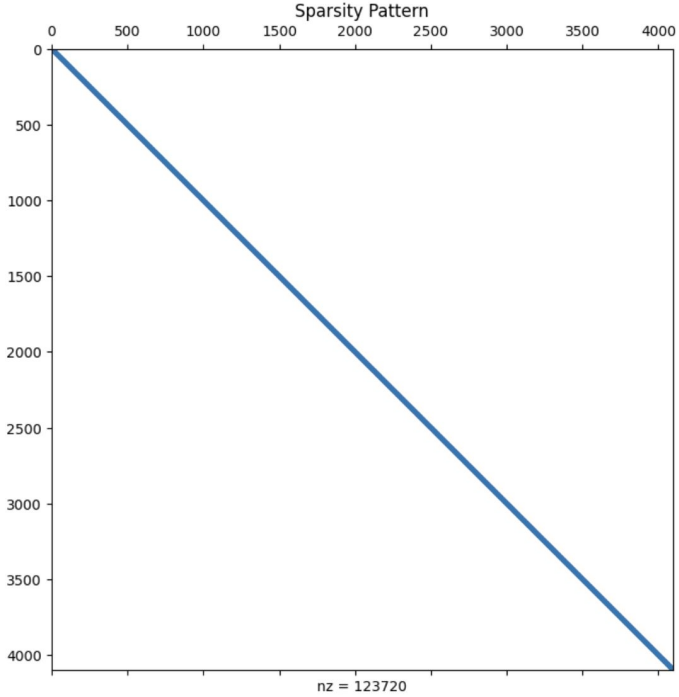
- Rows = 4096
- Columns = 4096
- Percentage NNZ = 0.4% and 0.8%
- Width = 8 and 16


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 63464

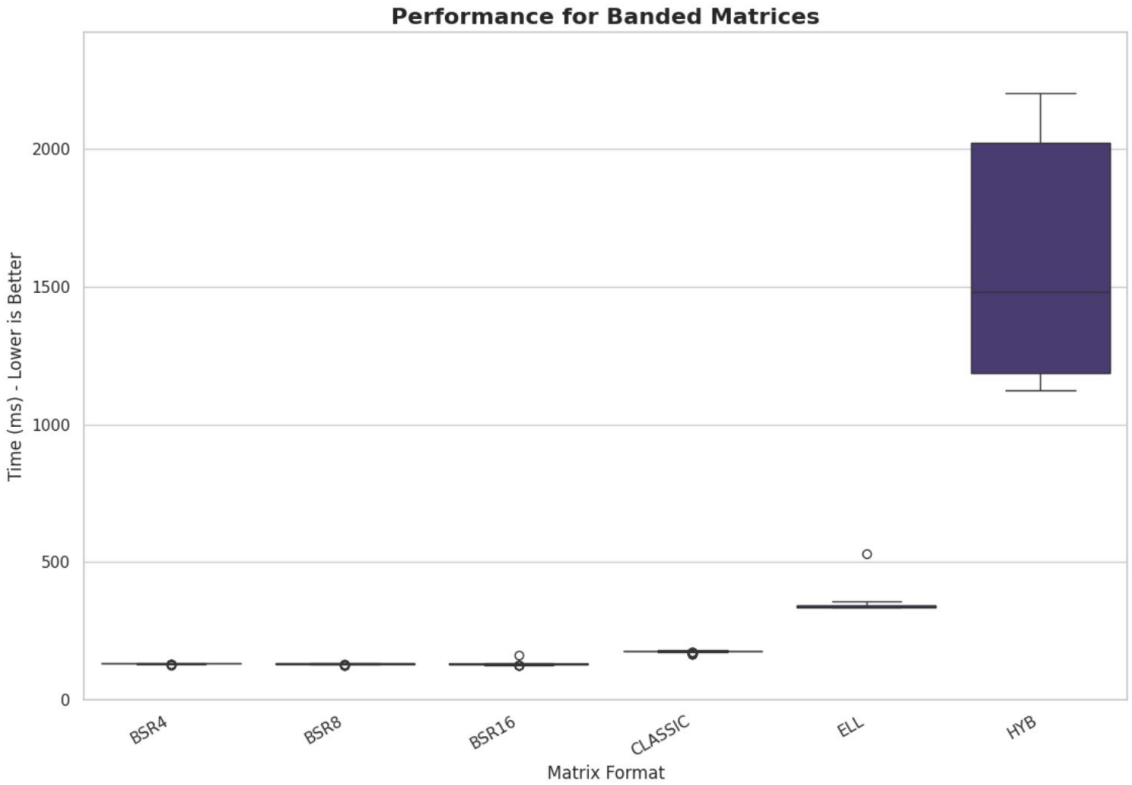

Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 63351

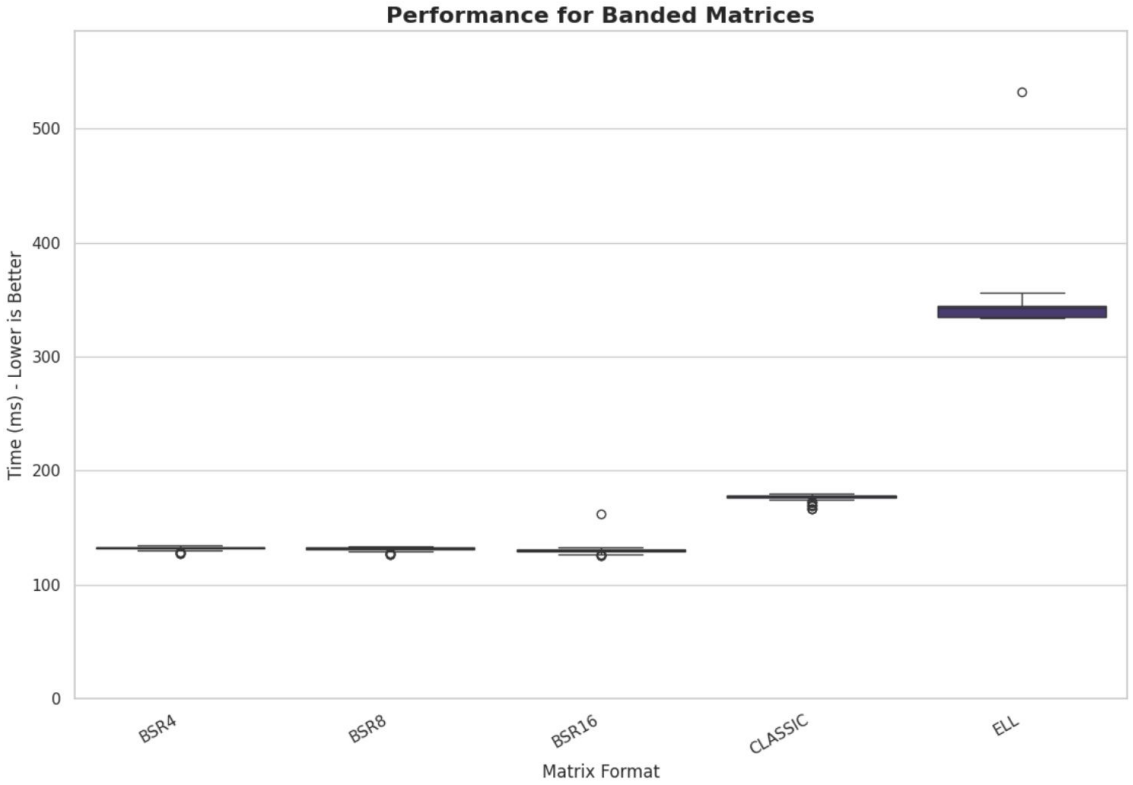
Width = 8


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 123645


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 123720

Width = 16


Performance for Banded Matrices
Time (ms) - Lower is Better
0
500
1000
1500
2000
BSR4
BSR8
BSR16
CLASSIC
ELL
HYB
Matrix Format

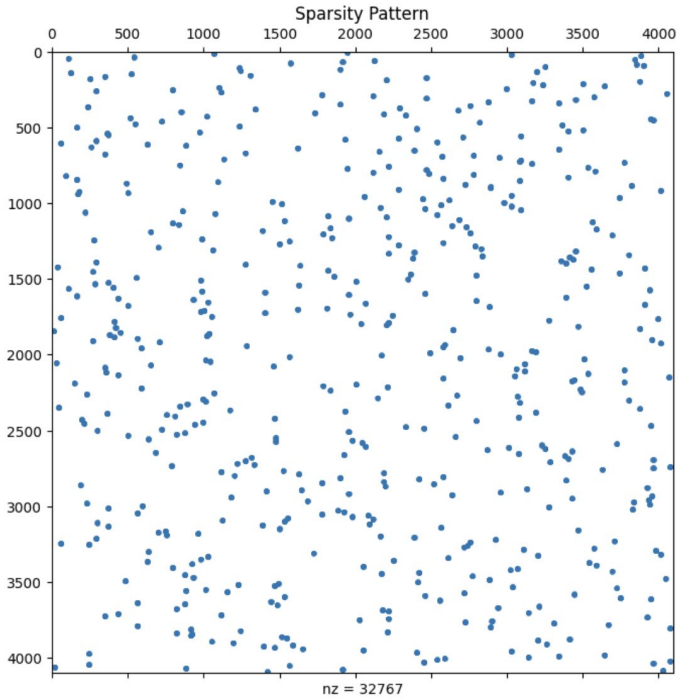

Performance for Banded Matrices
Time (ms) - Lower is Better
0
100
200
300
400
500
BSR4
BSR8
BSR16
CLASSIC
ELL
Matrix Format

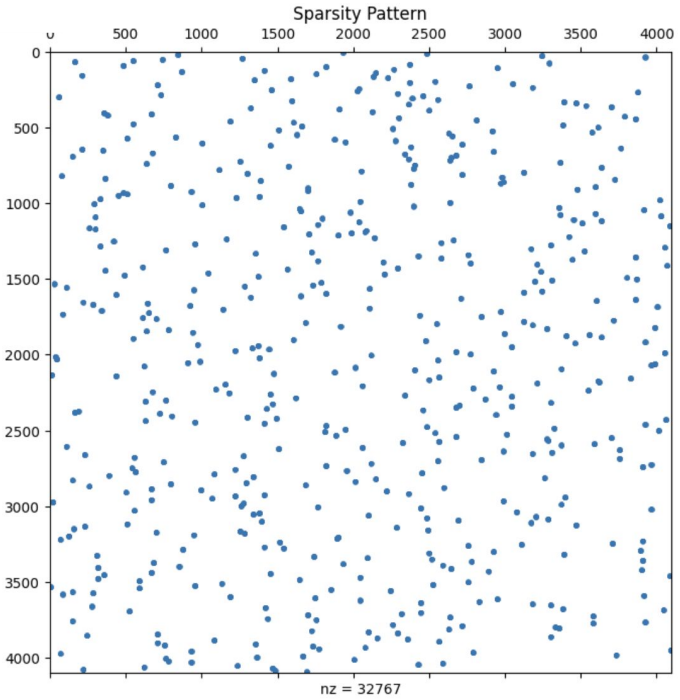
Banded Matrices Results

	Total (ms)	Allocation (ms)	Mult (ms)	Free (ms)
BSR 4	256.88	108.52	132.37	15.86
BSR 8	252.79	105.72	131.81	15.03
BSR 16	307.48	108.97	130.40	67.37
CLASSIC	385.35	137.27	177.85	70.31
ELL	514.14	105.64	342.49	66.78
HYB	1618.49	105.90	1479.66	14.76

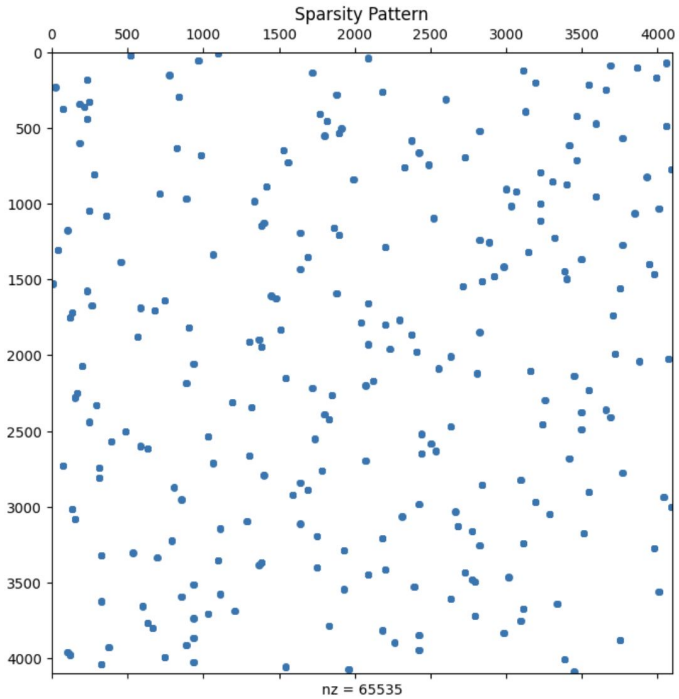
Block Matrices

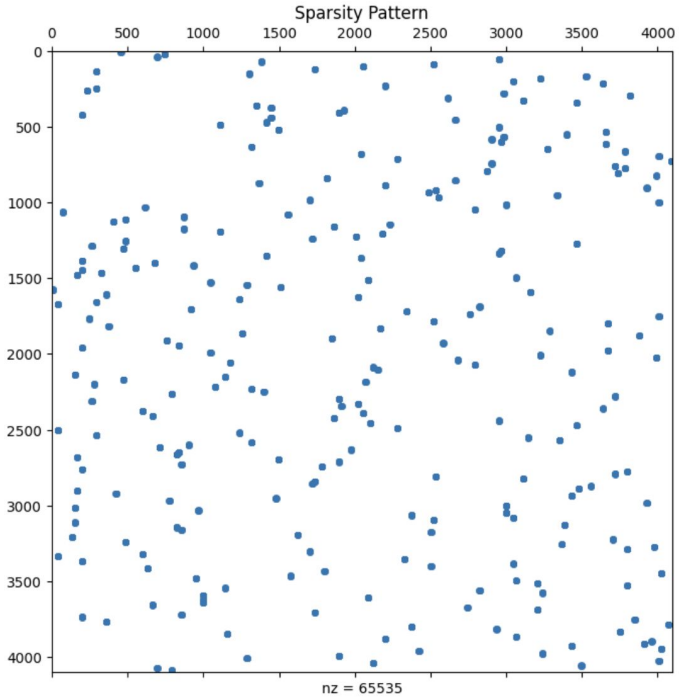
- Rows = 4096
- Columns = 4096
- Percentage NNZ = 0.3% and 0.6%
- Blocks Size = 8 or 16


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 32767

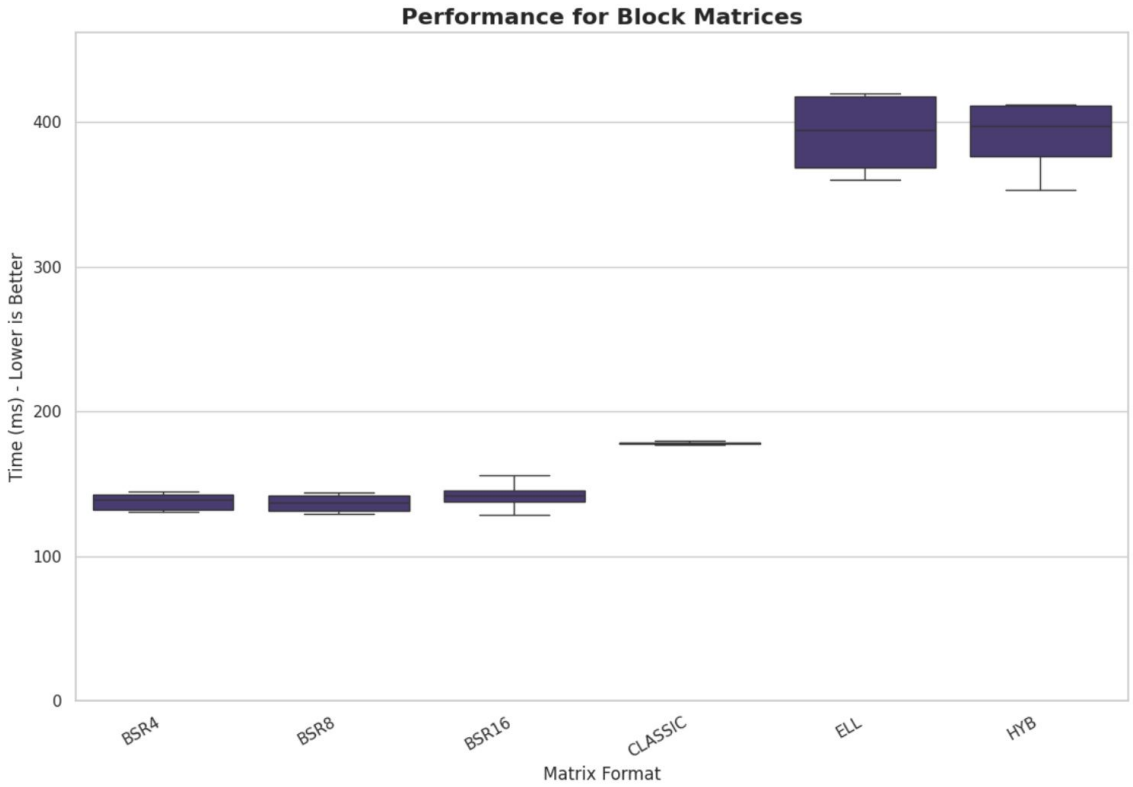

Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 32767

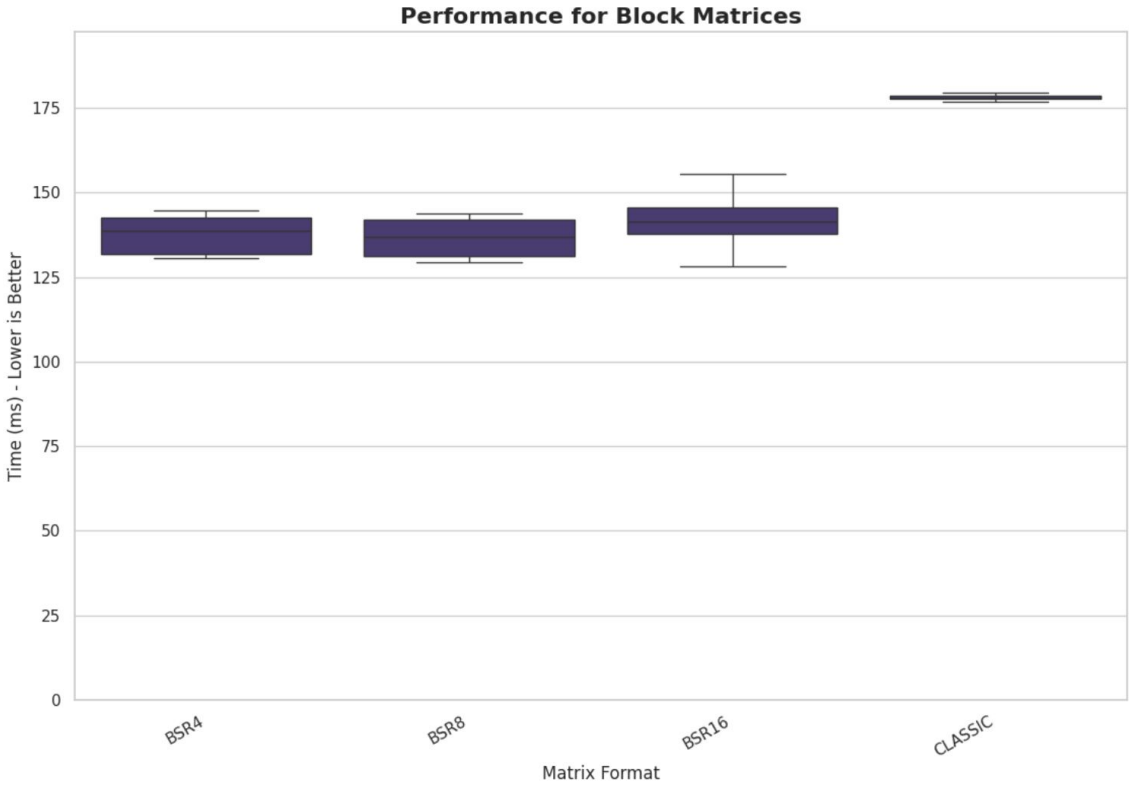
Block Size = 8





Block Size = 16


Performance for Block Matrices
Time (ms) - Lower is Better
0
100
200
300
400
BSR4
BSR8
BSR16
CLASSIC
ELL
HYB
Matrix Format

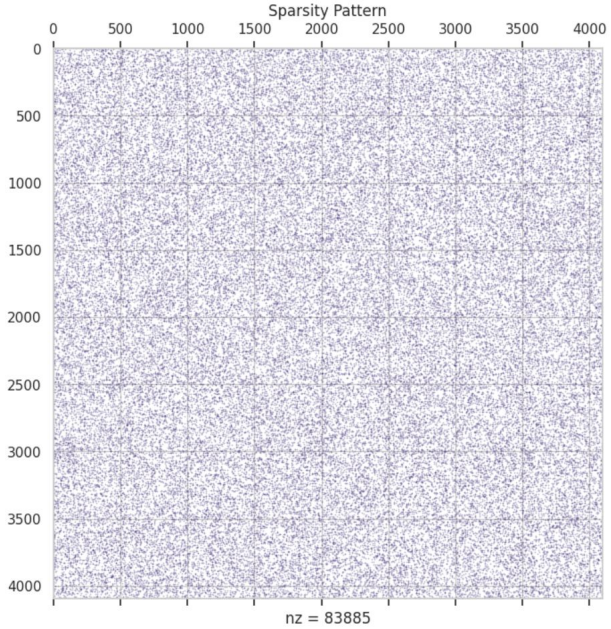

Performance for Block Matrices
Time (ms) - Lower is Better
0
25
50
75
100
125
150
175
BSR4
BSR8
BSR16
CLASSIC
Matrix Format

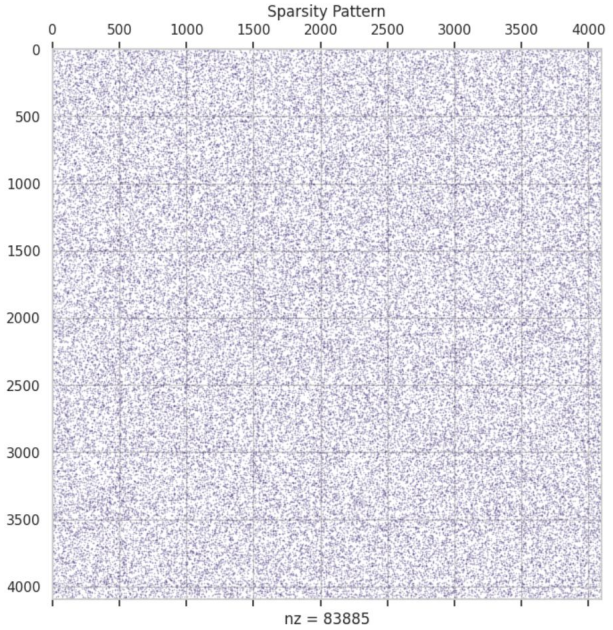
Block Matrices Results

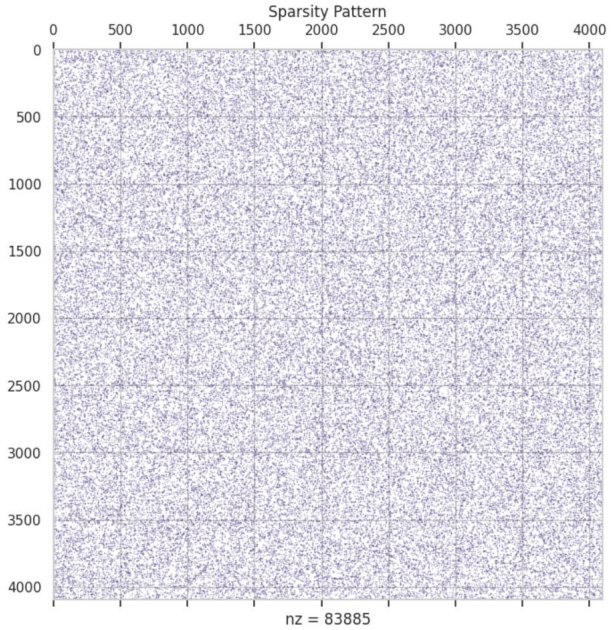
	Total (ms)	Allocation (ms)	Mult (ms)	Free (ms)
BSR 4	261.81	105.13	138.65	14.73
BSR 8	260.23	105.60	136.79	15.63
BSR 16	315.67	107.67	141.52	67.23
CLASSIC	385.95	136.17	178.04	70.36
ELL	584.64	108.388	394.36	67.50
HYB	526.75	102.22	397.45	14.85

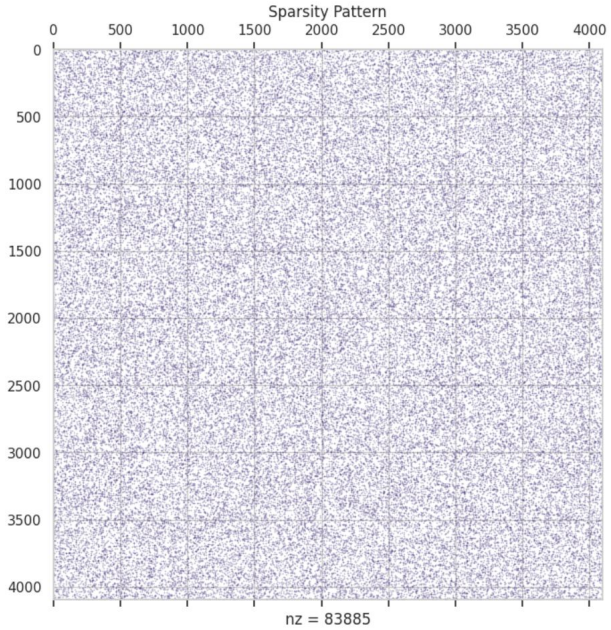
Random Matrices (0.5%)

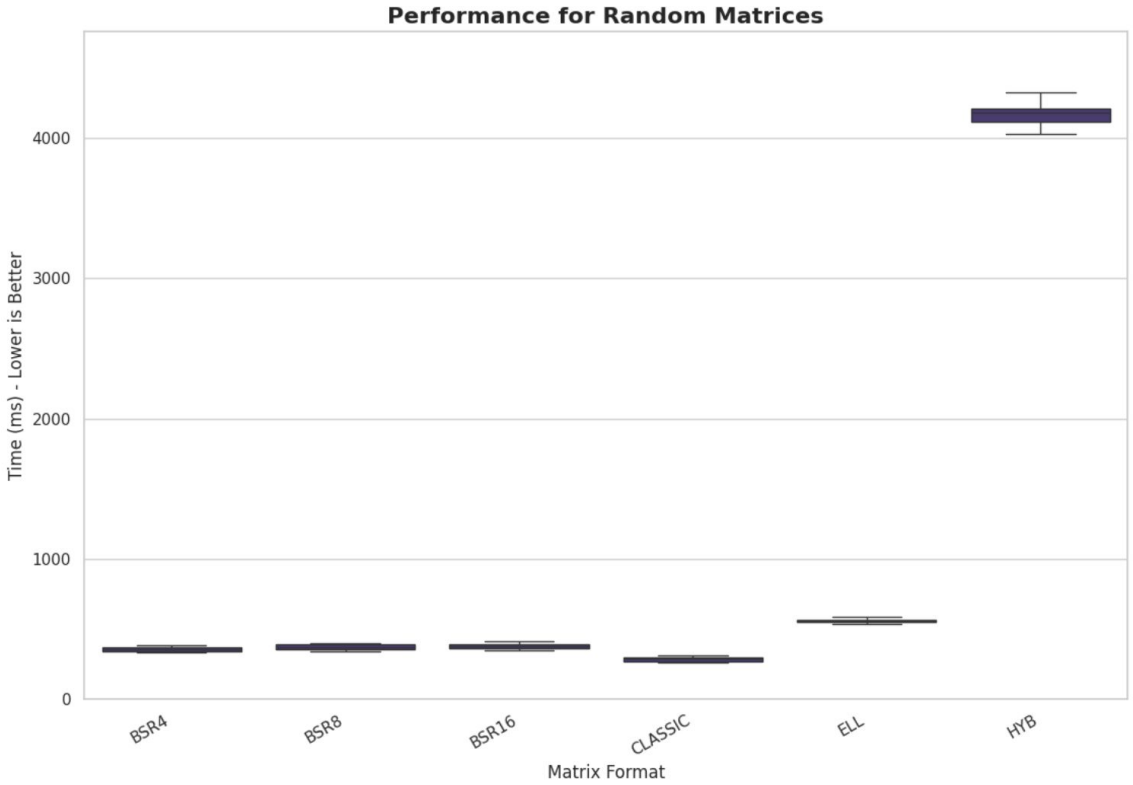
- Rows = 4096
- Columns = 4096
- percentage of NNZ = 0.5%

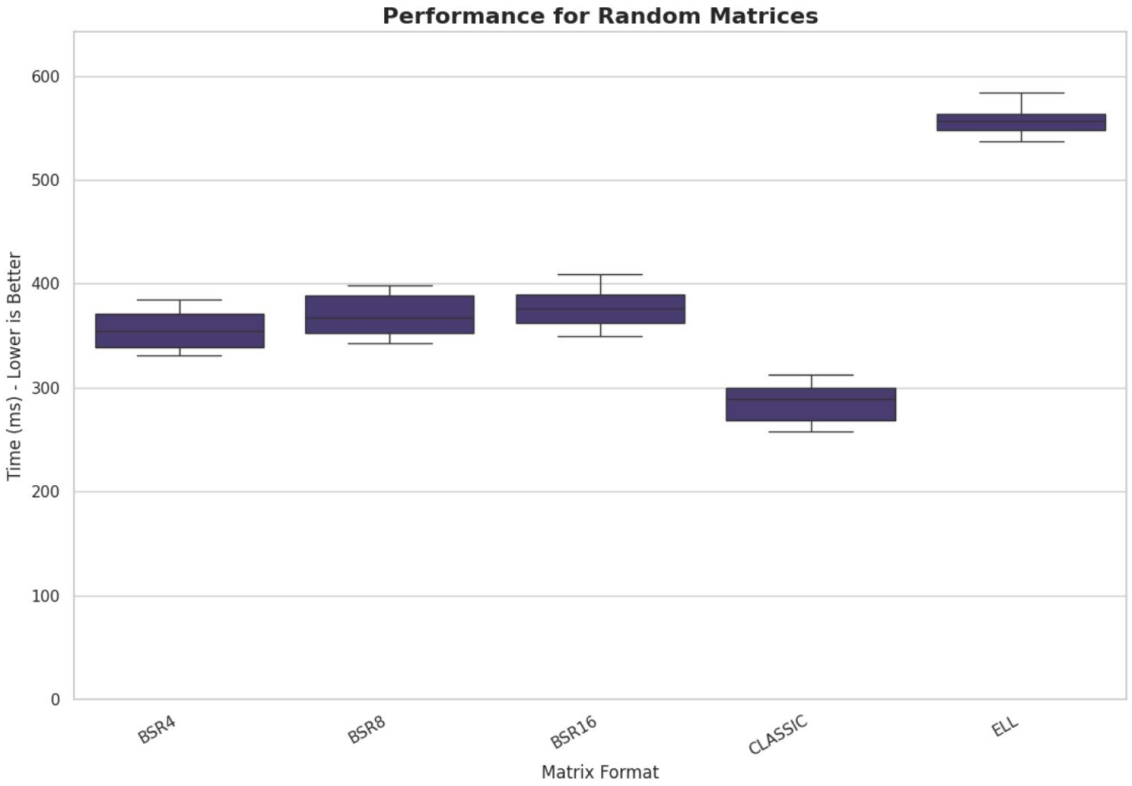

Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 83885


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 83885


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 83885


Sparsity Pattern
0
500
1000
1500
2000
2500
3000
3500
4000
nz = 83885


Performance for Random Matrices
Time (ms) - Lower is Better
0
1000
2000
3000
4000
BSR4
BSR8
BSR16
CLASSIC
ELL
HYB
Matrix Format


Performance for Random Matrices
Time (ms) - Lower is Better
0
100
200
300
400
500
600
BSR4
BSR8
BSR16
CLASSIC
ELL
Matrix Format

Random Matrices Results (0.5% NNZ)

	Total (ms)	Allocation (ms)	Mult (ms)	Free (ms)
BSR 4	350.55	110.40	230.80	13.57
BSR 8	371.37	121.07	241.59	14.11
BSR 16	389.79	134.80	247.91	14.19
CLASSIC	290.73	130.80	100.91	65.33
ELL	552.36	106.27	382.04	63.77
HYB	4302.80	105.34	4100.20	13.27

Random Matrices Results (1% NNZ)

	Total (ms)	Allocation (ms)	Mult (ms)	Free (ms)
BSR 4	390.55	111.40	263.80	15.57
BSR 8	431.37	121.07	292.59	17.11
BSR 16	453.79	134.80	178.91	17.19
CLASSIC	386.73	137.80	178.91	70.33
ELL	581.36	108.27	405.04	67.77
HYB	15890.80	106.34	15764.20	14.27

Conclusion

CLASSIC

- Good In General
- Easy Memory Access
- Waste of Memory

BSR

- Block size based on Matrices
- Good in Matrices with patterns
- Low Branch Divergence

ELL

- High Branch Divergence
- Low cache usage (w.r.t. BSR)

HYB

- Excellent Compression
- Prev of ELL
- Sync for Coo Part

Thank you for the attention!