DATA-DRIVEN MIXED PRECISION SPARSE MATRIX VECTOR MULTIPLICATION FOR GPUS

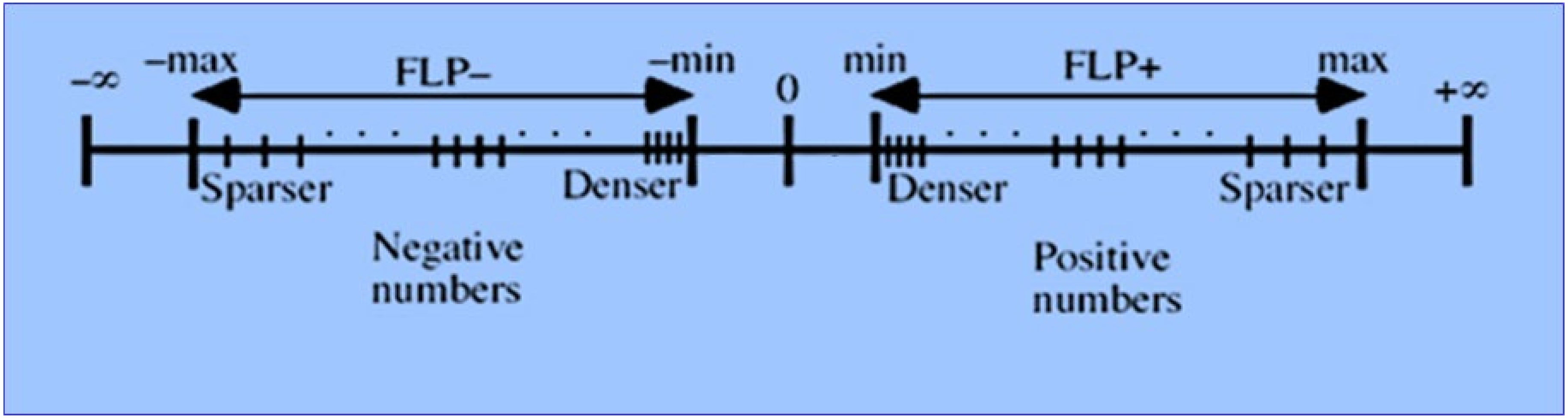
Khalid Ahmad, Hari Sundar and Mary Hall

School of Computing
University of Utah

Motivation

- Scientific applications use double precision for higher accuracy
- Downcasting precision leads to intolerable inaccuracies
- MpSpMV
  - alternative data-driven approach
  - lowers precision based on nonzero values

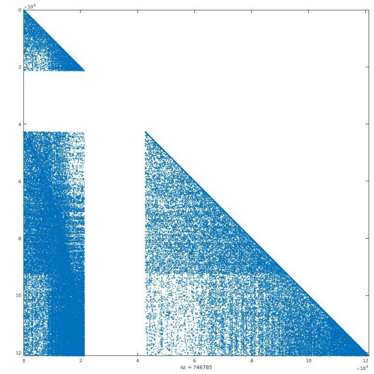
Values Closer to Zero are Well Represented



IEEE 754 floating point representation

Matrix Split

Input Matrix [cop20k_A.mtx]

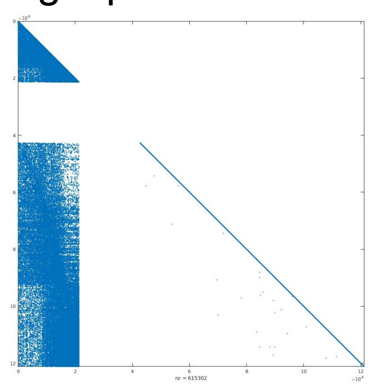


(a) NNZ=1,362,087

< |2|

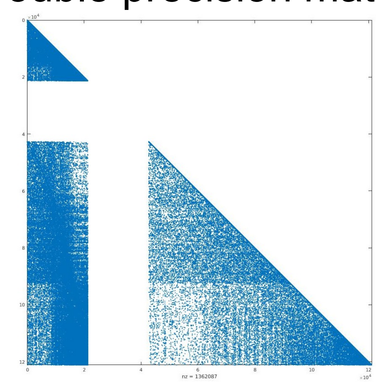
> |2|

Single precision matrix



(b) NNZ=746,785

Double precision matrix



(c) NNZ=615,302

Contributions

- ↓ execution time
- ↑ parallelism
- ↓ data movement
- ↑ accuracy compared to single precision

Results

- Average speedup of 1.06X
- Maximum speedup over double precision of 2.61X
- On average one decimal digit more accurate than single precision

THE UNIVERSITY OF UTAH

Compressed Sparse Row (CSR)

Dense Representation

0.1	0.2	0.3	0.4	0	0
0	1.2	1.3	0	0	0
0	0	2.3	2.4	2.5	2.6
0	0	0	3.4	3.5	0
0	0	0	0	0	4.6
0	0	0	0	0	5.6

NxN

Sparse Representation

A	0.1	0.2	0.3	0.4	1.2	1.3	2.3	2.4	2.5	2.6	3.4	3.5	4.6	5.6

Auxiliary Data Structures

col	0	1	2	3	1	2	2	3	4	5	3	4	5	5

index (row pointer)	0	4	6	10	12	13	14

THE UNIVERSITY OF UTAH

Sequential SpMV CSR Computation

```
for (i=0; i<N; i++) {
    for (j=index[i]; j<index[i+1]; j++) {
        y[i] += A[j] * x[col[j]];
    }
}
```

CUSP SpMV CSR Vector Implementation

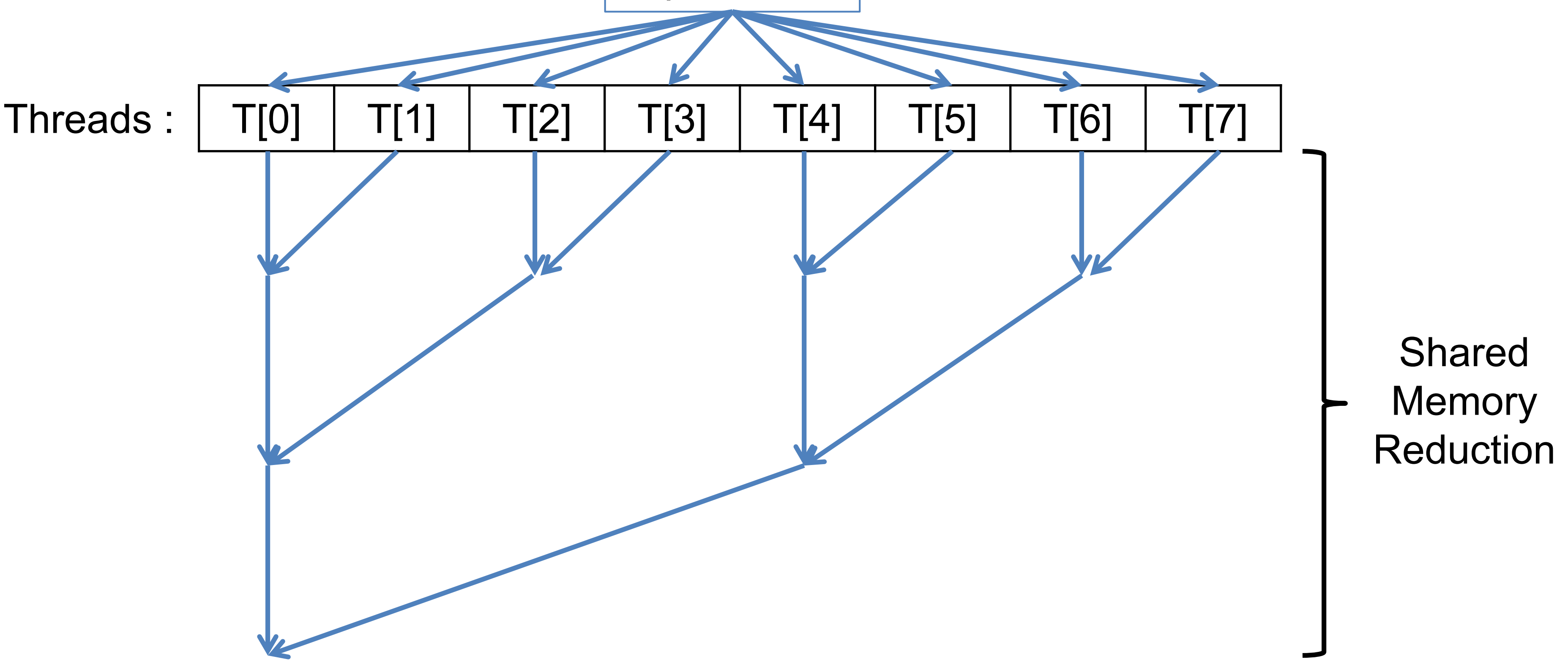
```
for (i=0; i < n; i++)
    for (j=index[i]; j<index[i+1]; j++){
        T[j] =a[j]*x[col[j]];
        y[i]  += T[j]
}
```

Reduction Sum

CUSP SpMV CSR Vector Implementation

```
for (i=0; i < n; i++)
    for (j=index[i]; j<index[i+1]; j++){
        T[j] =a[j]*x[col[j]];
        y[i]  += T[j]
}
```

Product Expression is scalar expanded



SpMV Kernel Difference

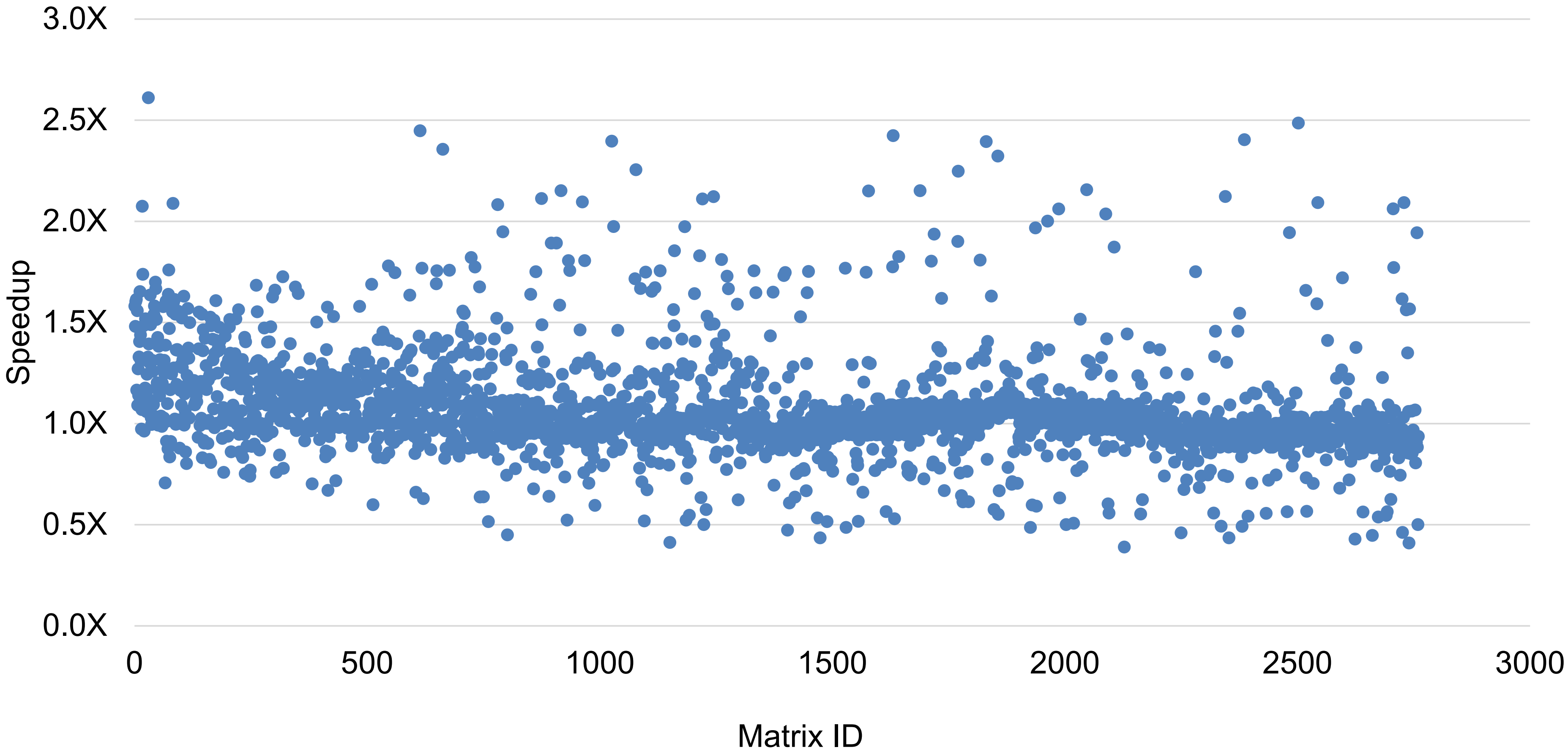
Kernel Section / Precision	Single	Mixed	Double
Prototype	A, x, y	As, x_s, Ad, x_d, y	A, x, y
Partial Products	Just one set	Two sets: single precision; double precision	Just one set
Reduction	One reduction for all		

Mixed precision is profitable:
2N + 1 < NNZ_s

* Kernel code available in the paper

THE UNIVERSITY OF UTAH

Sparse Matrix Collection



1359 out of the 2202 matrices showed speedup using mixed precision

THE UNIVERSITY OF UTAH

Experimental Evaluation

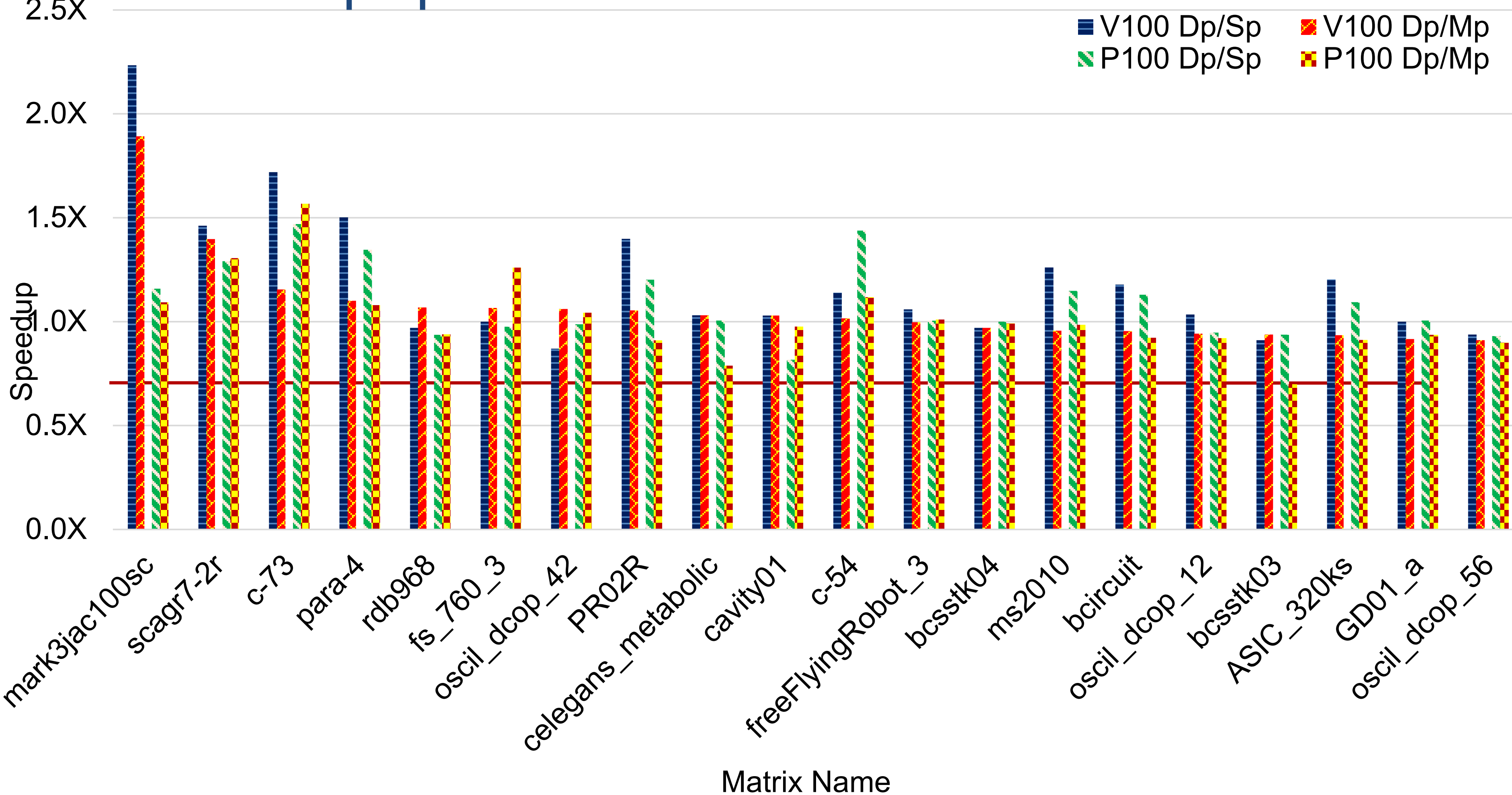
- Methodology
  - Select representative subset of matrices
  - Nvprof metrics collected
- Results
  - V100 and P100
  - Average of 500 SpMV runs recorded

Selecting Representative Matrices

- K Dimension Algorithm
- We used 4 dimensions to select a representative subset of matrices:
  - MpSpMV speedup over DpSpMV
  - Number of non zeros in a sparse matrix
  - Density of the sparse matrix
  - Ratio of nonzero values inside the range

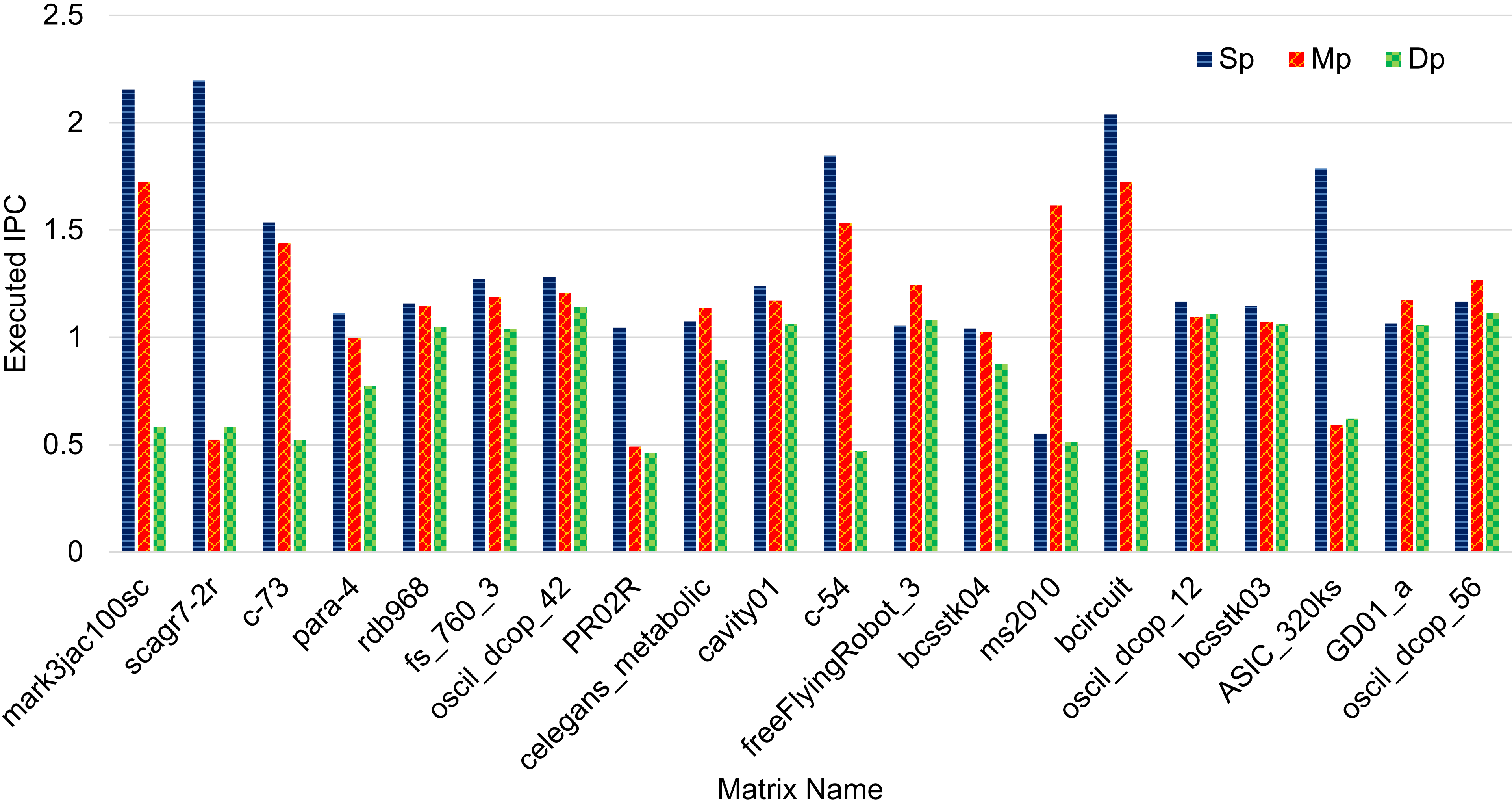
THE UNIVERSITY OF UTAH

Speedup Comparison of MpSpMV and SpSpMV over DpSpMV on V100 and P100 GPUs



11 matrices show speedup using mixed precision.
Performance for rest of matrices → mixed precision ≈ double Precision.

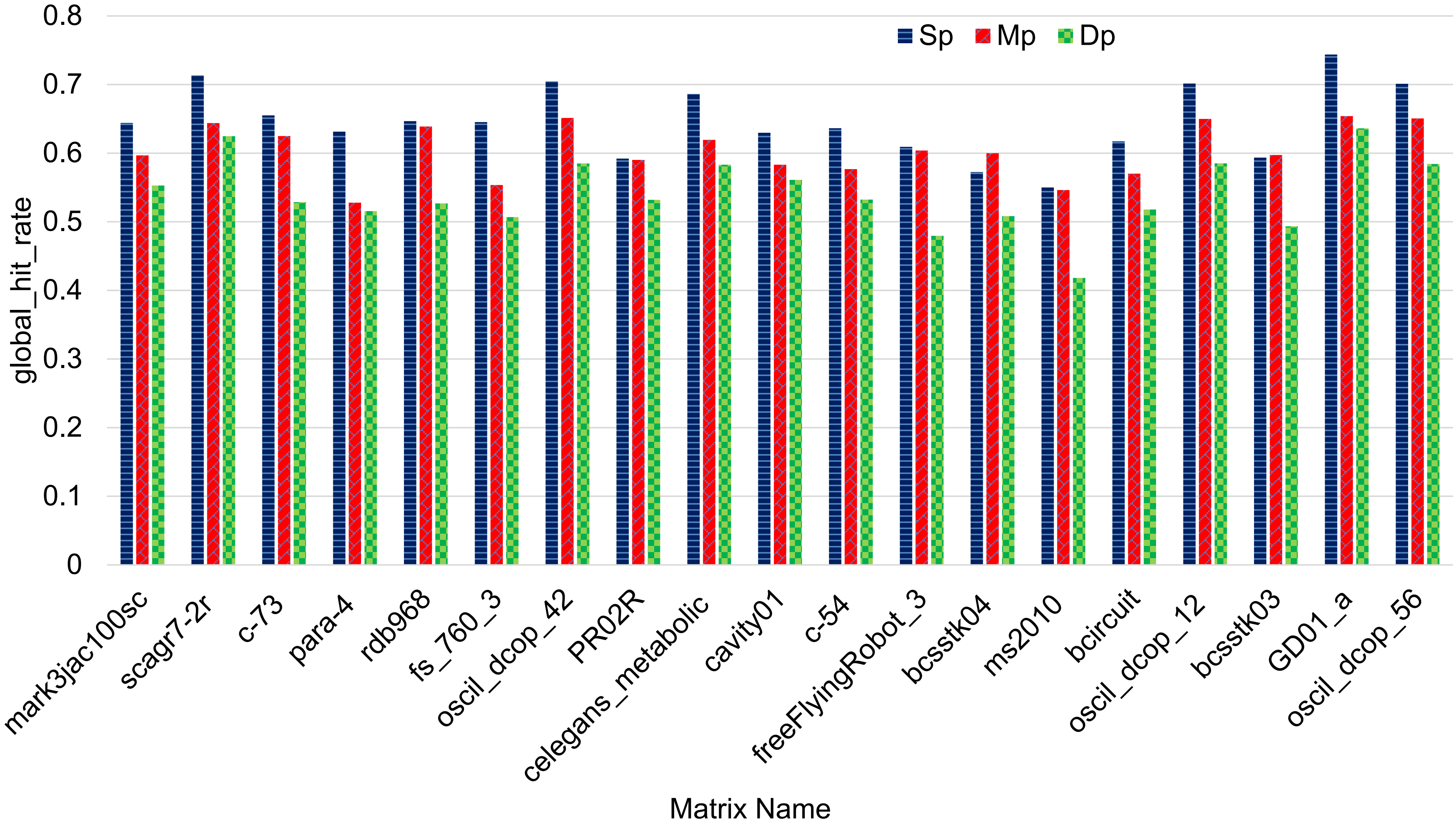
Executed IPC



Higher instructions per cycle than double precision

THE UNIVERSITY OF UTAH

Global Hit Rate



Higher hit rates in L1 cache compared to double precision

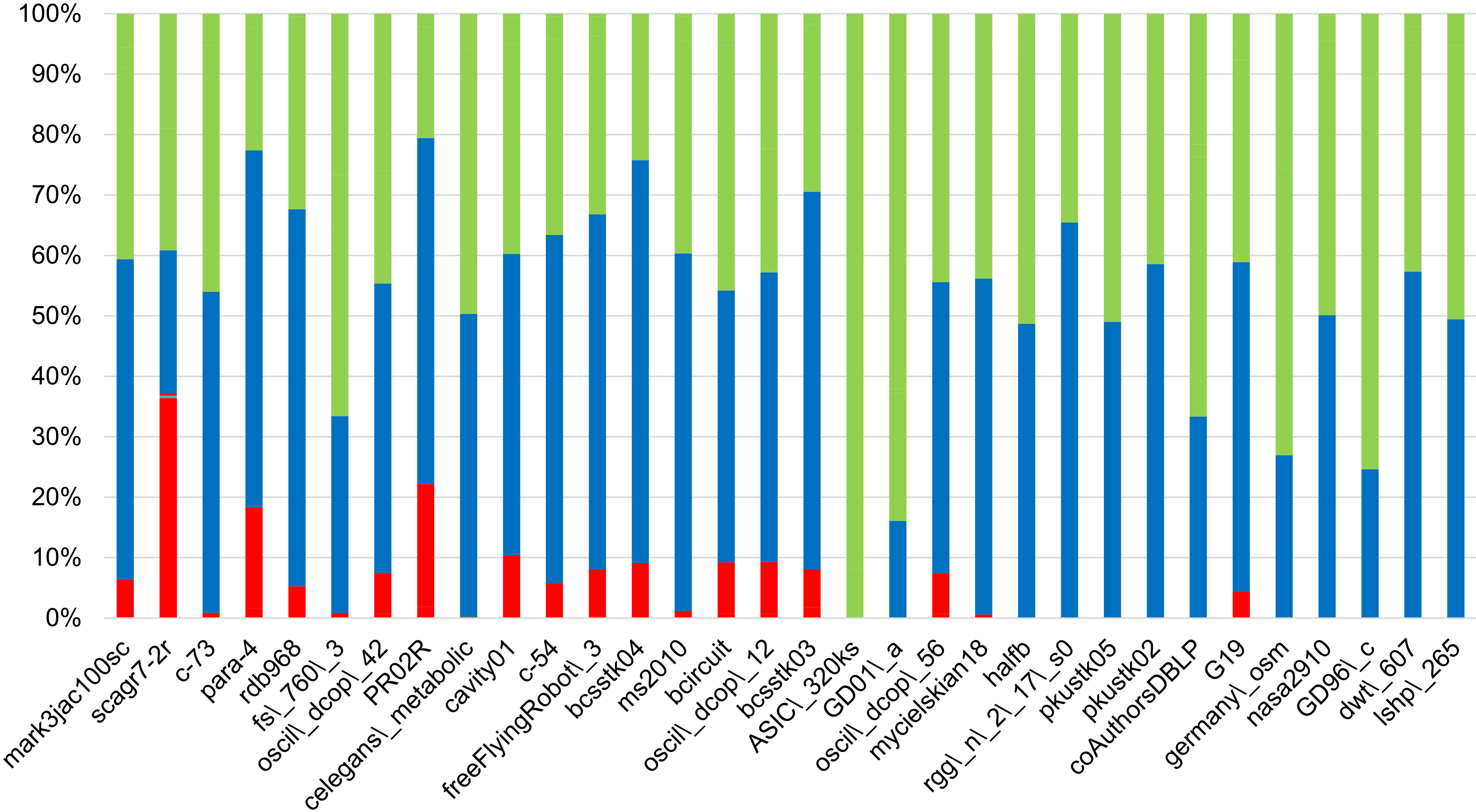
Accuracy

$$X_\tau = \frac{1}{9}$$

$$X_A = 0.111$$

$$|X_\tau - X_A| = 0.\underbrace{000}_{\text{3 significant decimal digits}}11$$

SpSpMV Accuracy

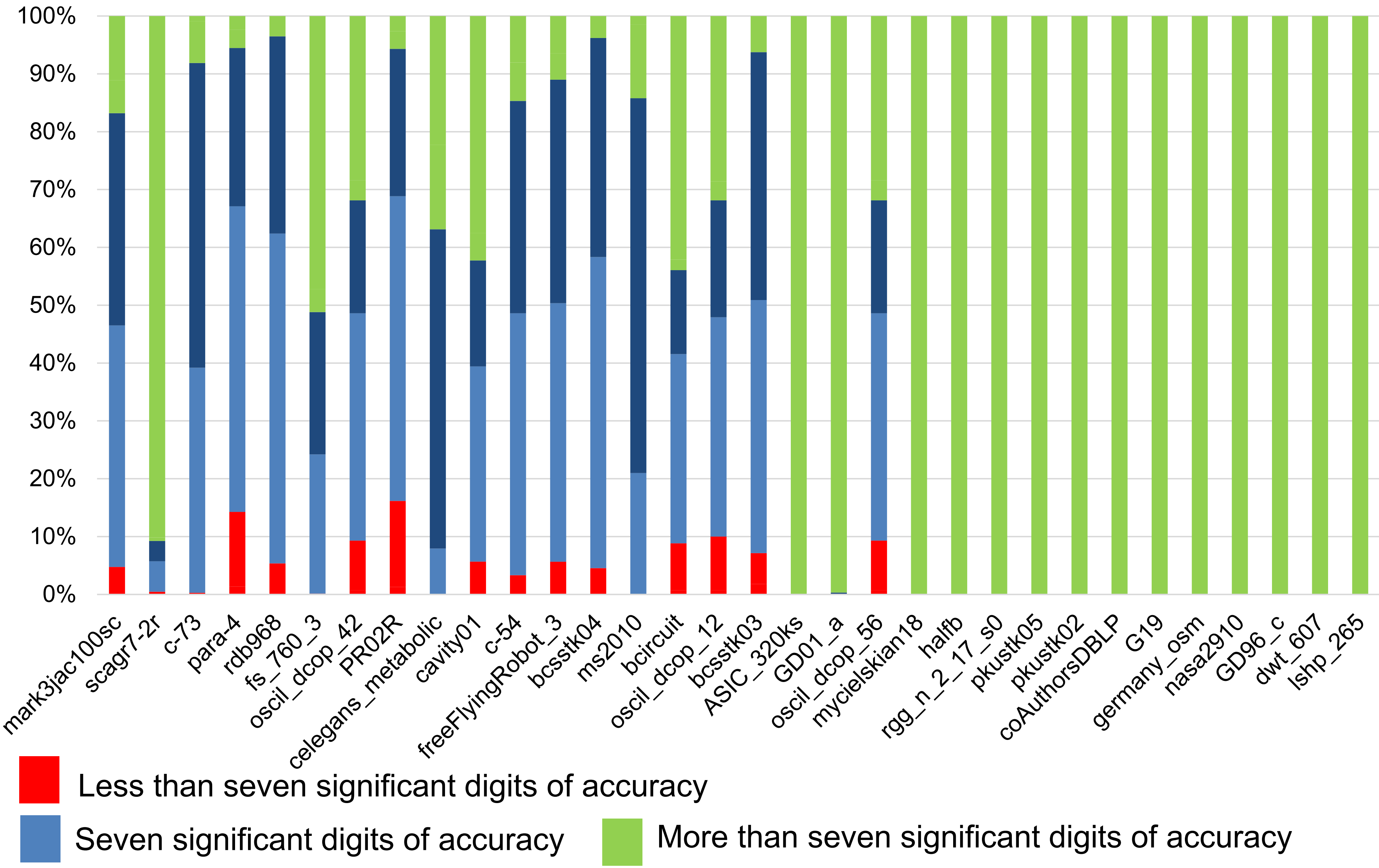


Less than six significant digits of accuracy

Six significant digits of accuracy

More than six significant digits of accuracy

MpSpMV Accuracy



Conclusion

New data driven mixed precision implementation

- ↓ execution time
- ↑ parallelism
- ↓ data movement
- ↑ accuracy compared to single precision

Results

- Average speedup of 1.06X
- Maximum speedup over double precision of 2.61X
- On average one decimal digit more accurate than single precision