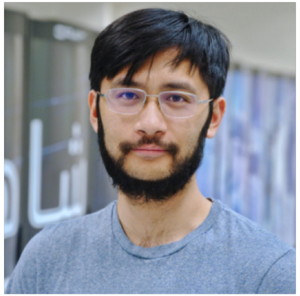
Accelerating seismic processing with Tile Low-Rank matrix approximations

M. Ravasi

Co-authors

Matteo Ravasi

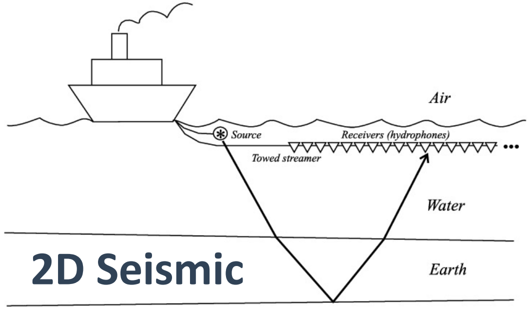


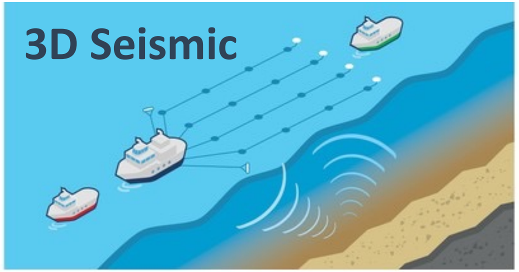
Yuxi Hong

David Keyes

Hatem Ltaief

New challenges





More data

More computations

Adjoint-based processing $m = f(d)$

(SRME, IME)

Inversion-based processing $d = g(m)$

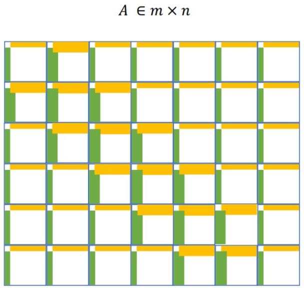
(EPSI, MDD, Mck)

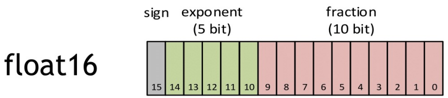
New technologies

Hardware

More *memory and compute*

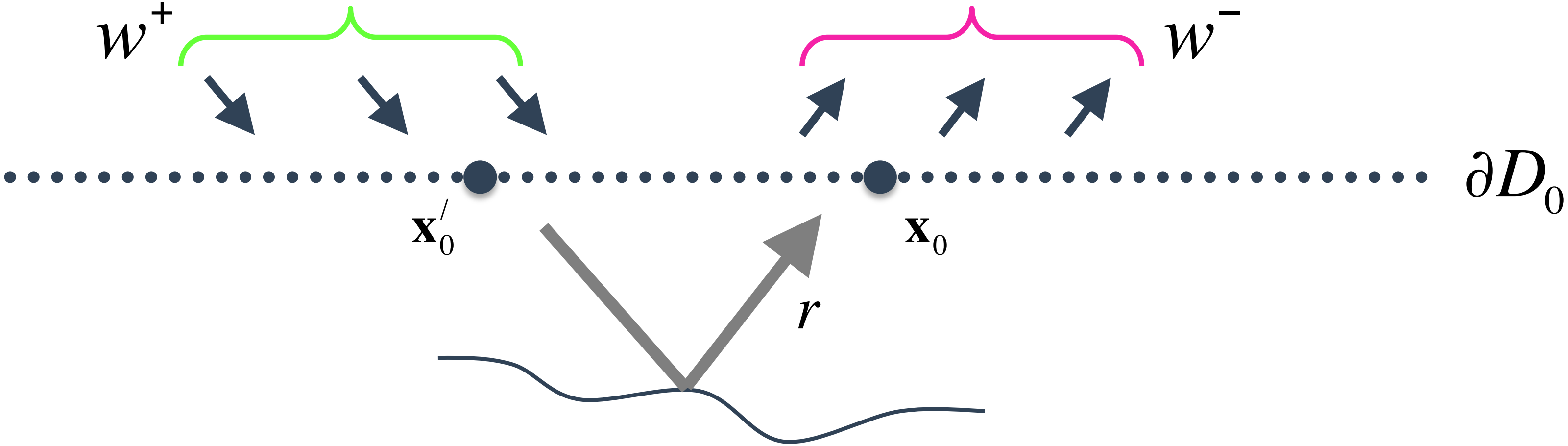
Less *memory and compute*





Algorithms & Software

A common building block



Multidimensional convolution (MDC)

$$\underline{w^-(\mathbf{x}_0,\mathbf{x}_S,t)} = \int_{\partial D_0}\int \underline{w^+(\mathbf{x}_0^{\prime},\mathbf{x}_S,t-\tau)}\,\underline{r(\mathbf{x}_0,\mathbf{x}_0^{\prime},\tau)}\,d\tau\, d\mathbf{x}_0^{\prime}$$

A common building block

$$\mathbf{w}^{-} = \mathbf{W}^{+}\mathbf{r}$$

$$\overbrace{\mathbf{F}^{\mathrm{H}} \quad \mathbf{I}^{H}_{\omega_{\max}} \quad \hat{\mathbf{W}}^{+} \quad \mathbf{I}_{\omega_{\max}} \quad \mathbf{F}}$$

Real IFFT

Padding

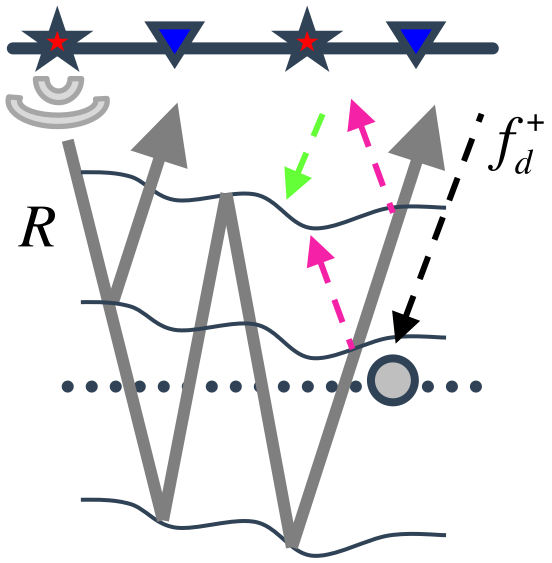
Batch matrix multiplication

Restriction

*Real FFT (**r** is real)*

Sweet spot!!

An example application



Marchenko equation

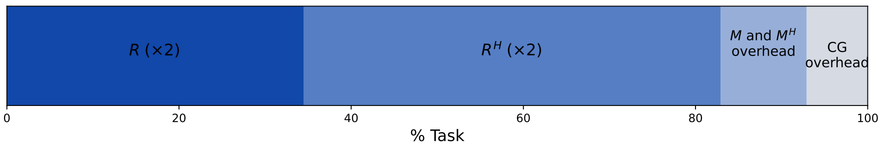
$$\begin{bmatrix} \Theta \mathbf{R} \mathbf{f}_\mathbf{d}^+ \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\Theta \mathbf{R} \\ \Theta \mathbf{R}^* & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}^- \\ \mathbf{f}_\mathbf{m}^+ \end{bmatrix}$$

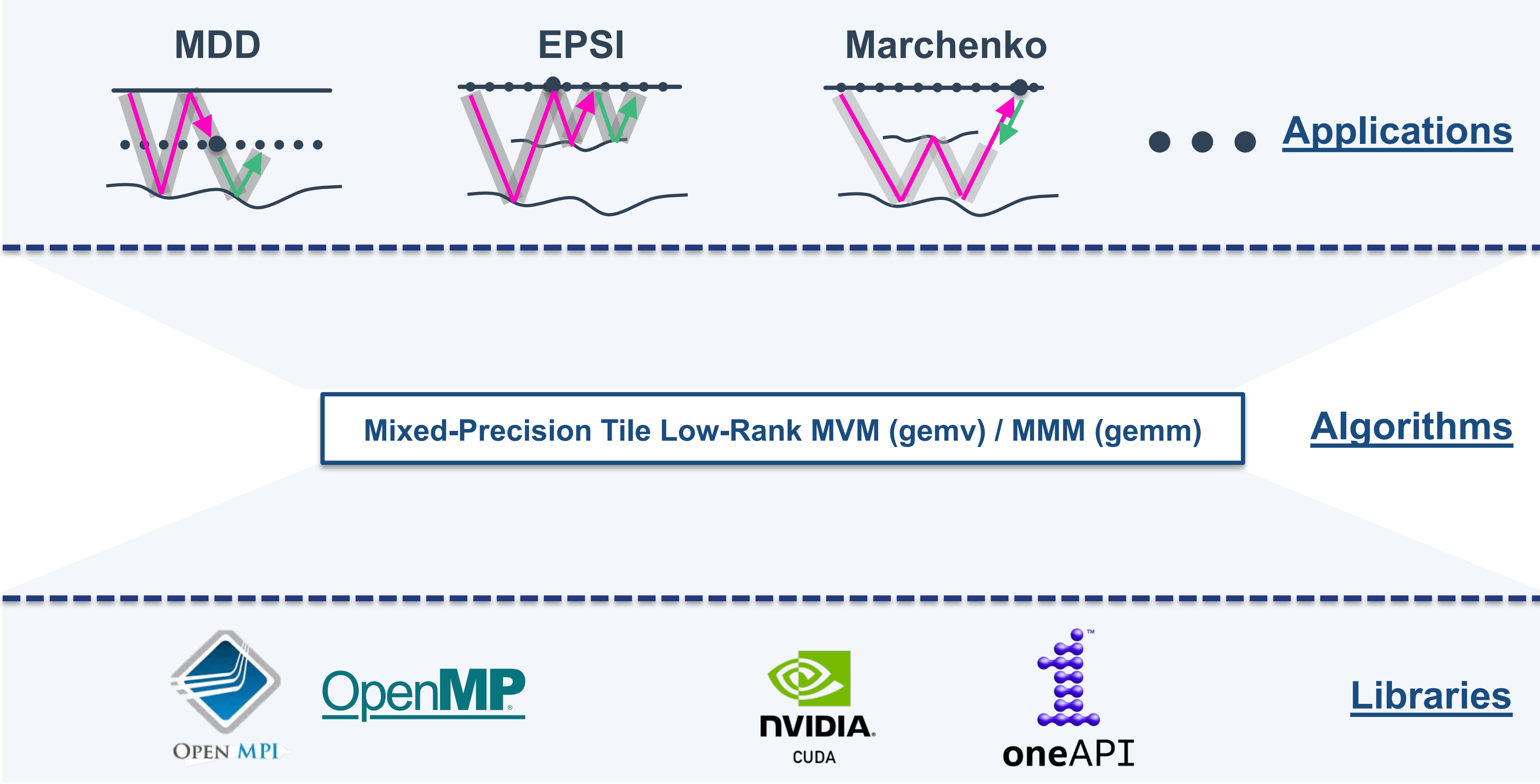
$\mathbf{R}, \mathbf{R}^*$: MDC operators

Forward: 2 MDC, Adjoint: 2 MDC

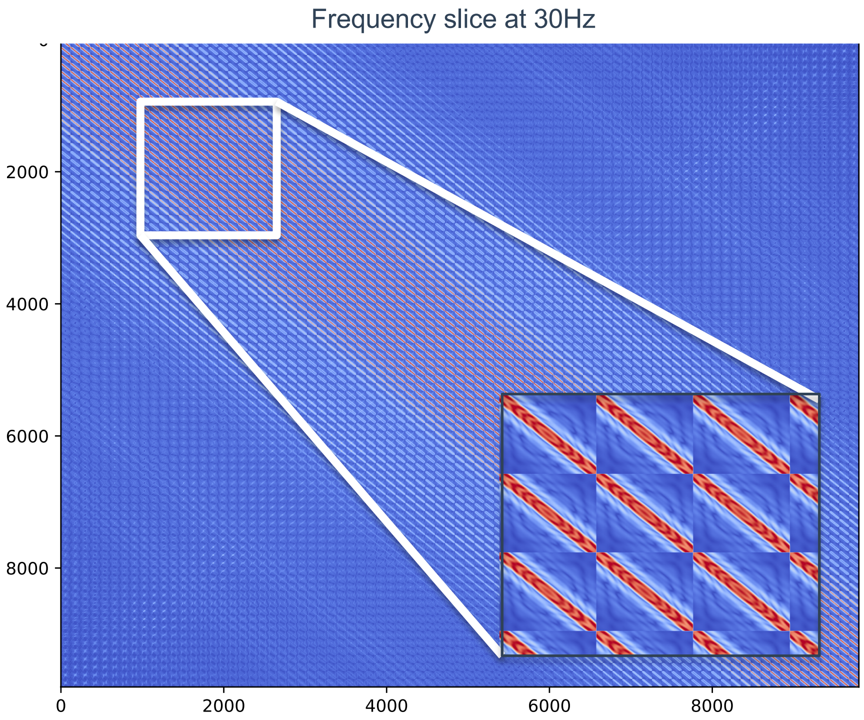
CGLS (10 iterations): 40 MDC

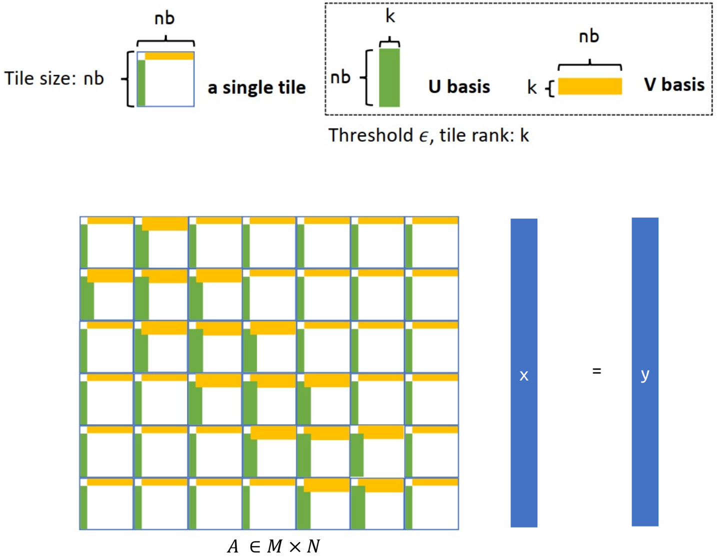
Task % for one iteration of CG:



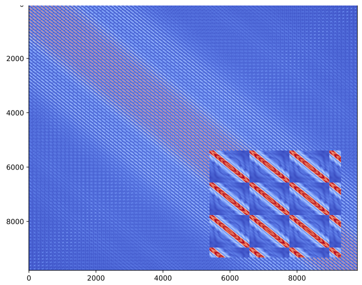

MDD
EPSI
Marchenko
Applications
Mixed-Precision Tile Low-Rank MVM (gemv) / MMM (gemm)
Algorithms
OPEN MPI
OpenMP
NVIDIA
CUDA
oneAPI
Libraries

Why Tile Low-Rank compression?

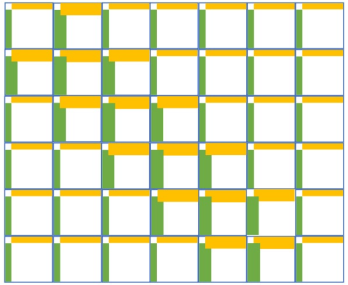




Why Tile Low-Rank compression?



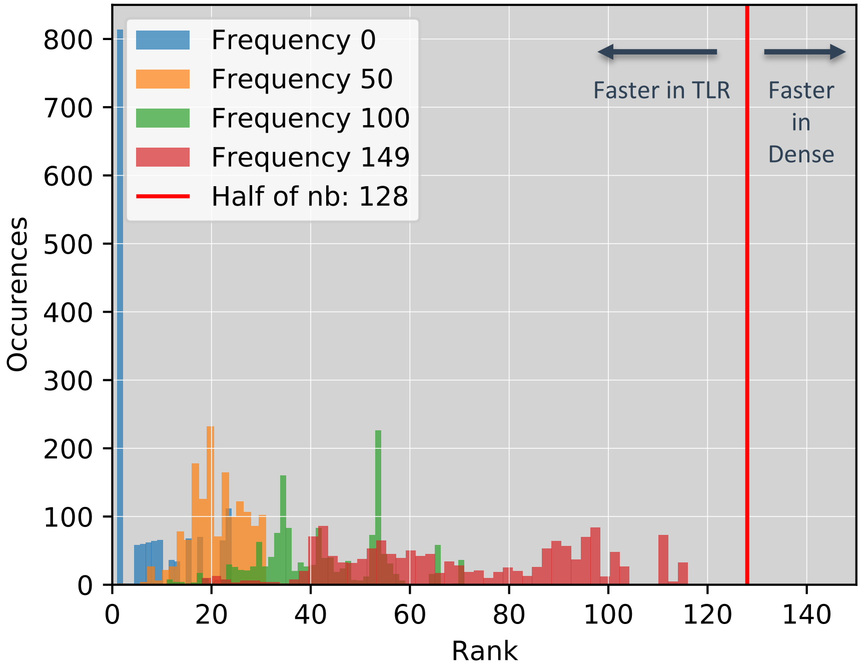
Original matrix dimension = 150 x 9801 x 9801 x 8 Bytes = 115.27 GB

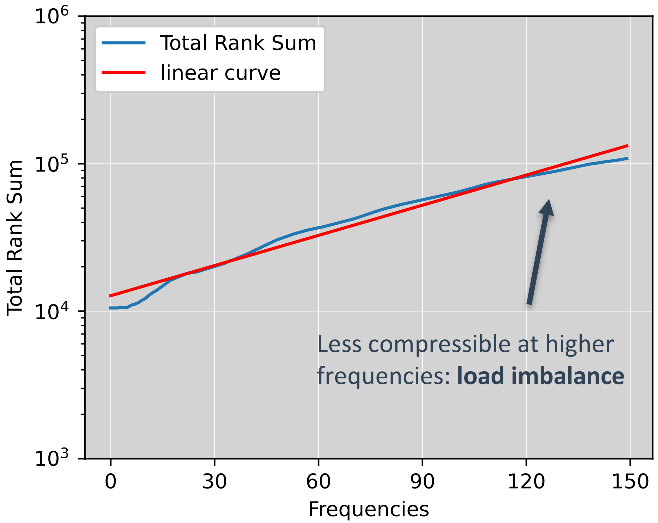


Compressed format of data (U and V for 150 frequencies) = 31 GB

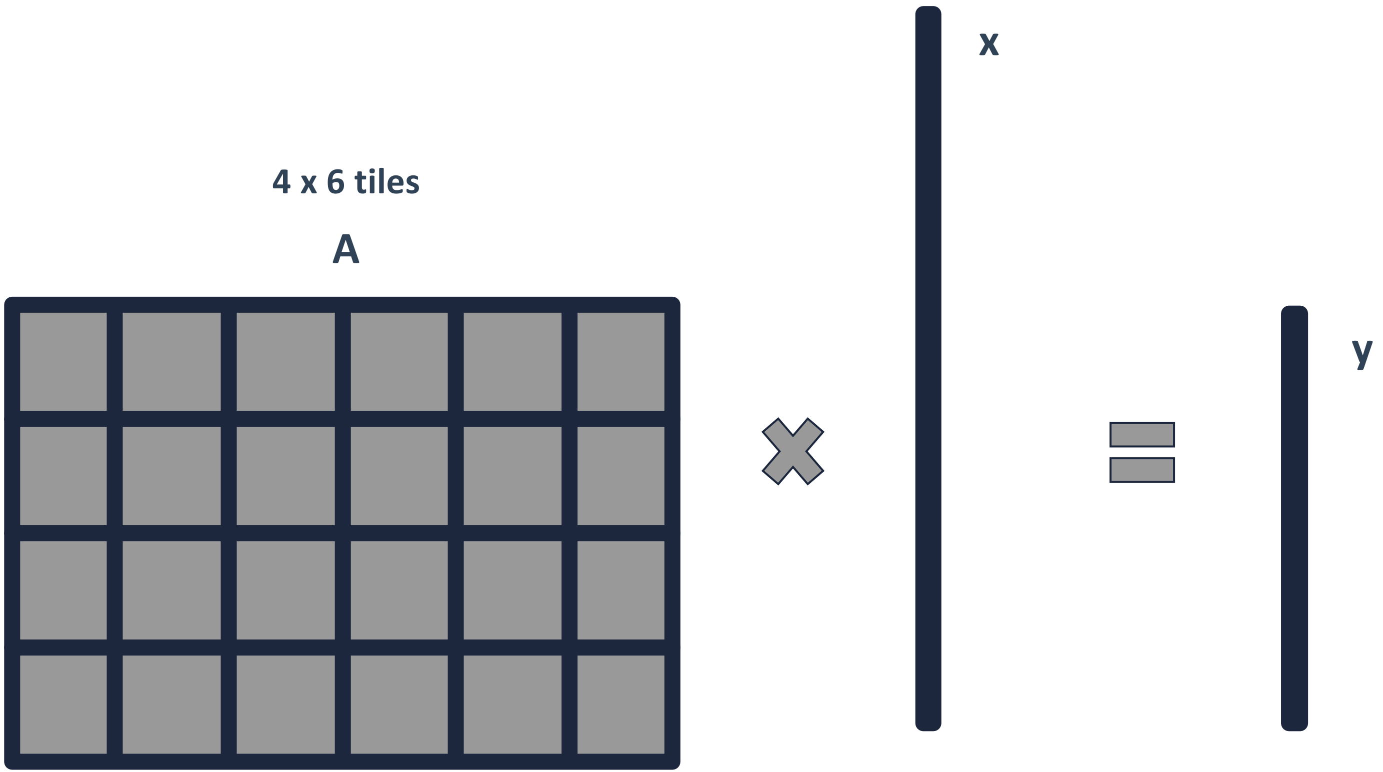
Compression factor = 3.7

Why Tile Low-Rank compression?

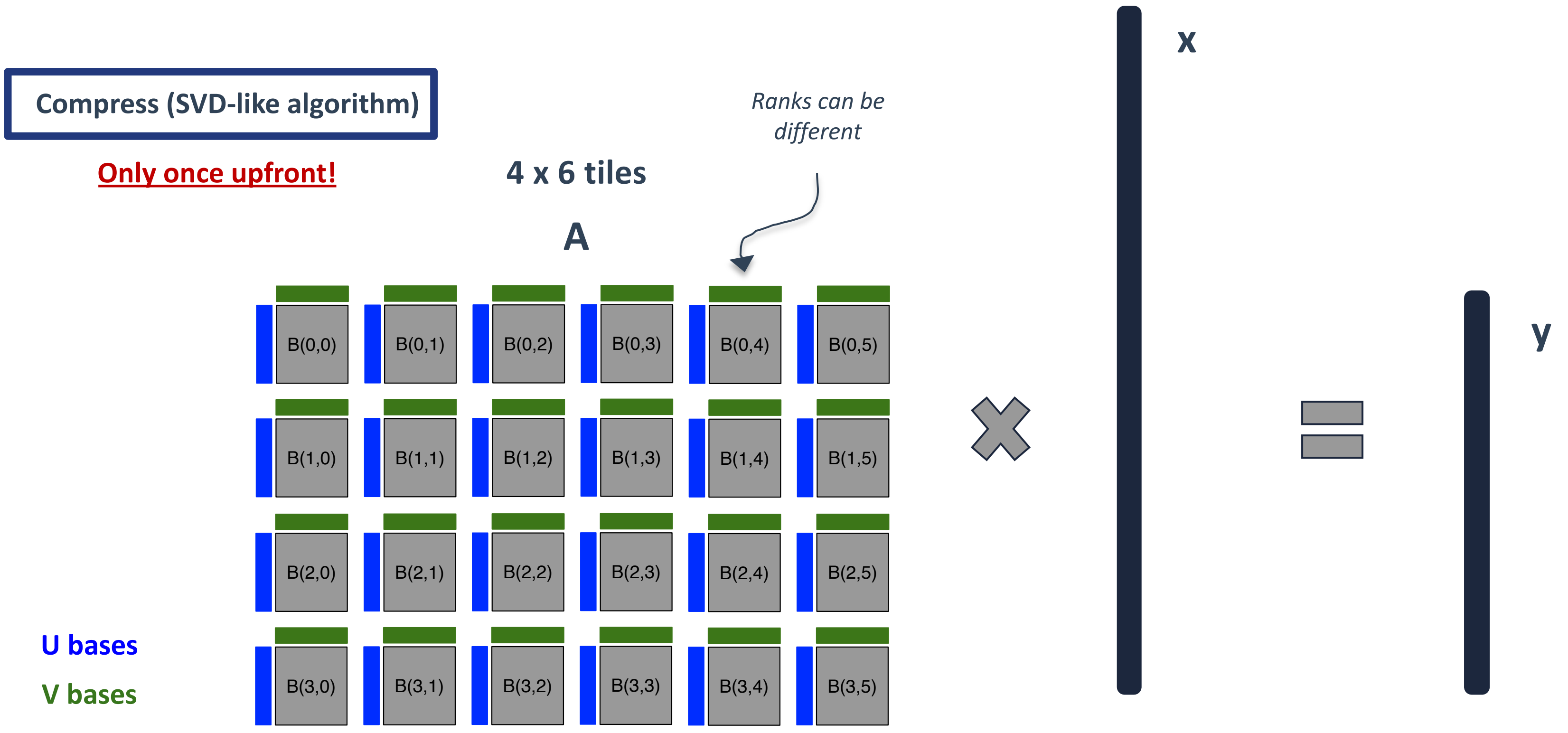




Tile Low-Rank Matrix-Vector Multiplication



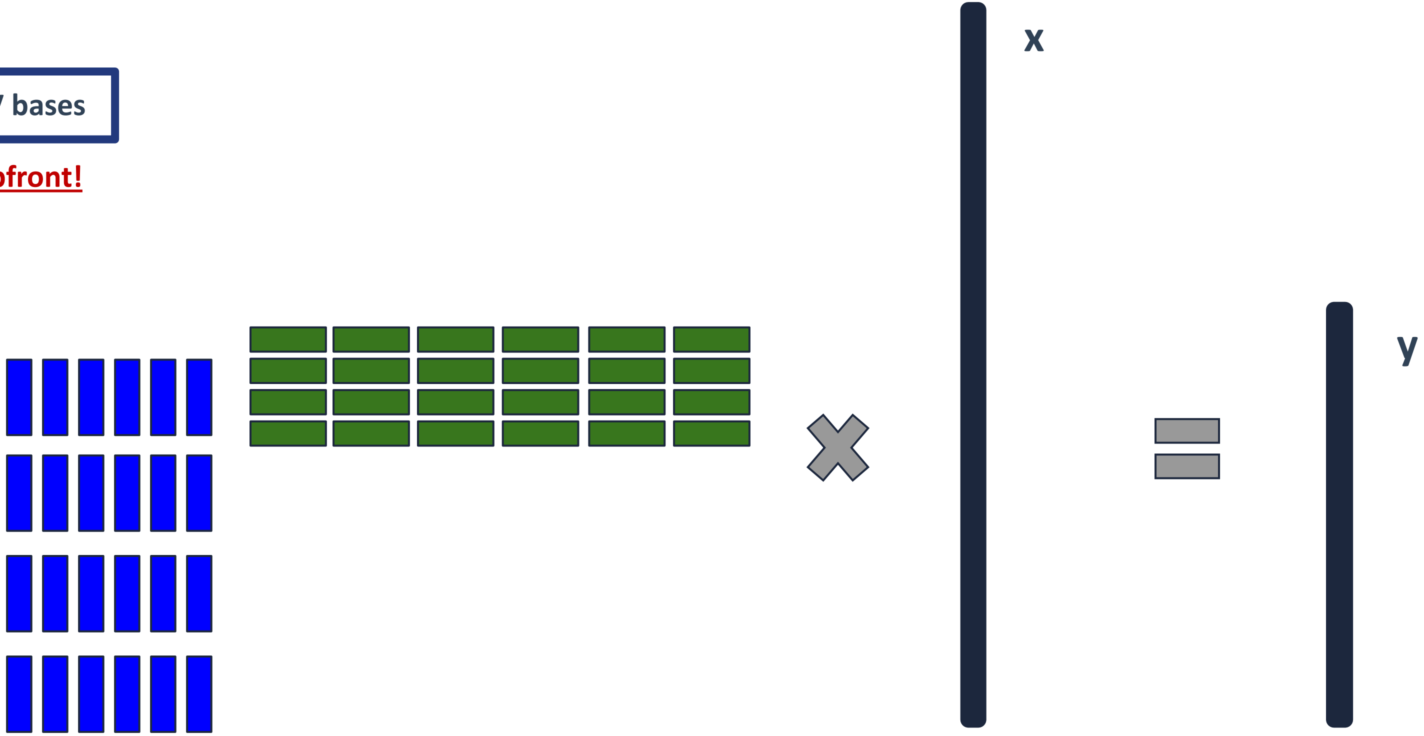
Tile Low-Rank Matrix-Vector Multiplication



Tile Low-Rank Matrix-Vector Multiplication

Stack U and V bases

Only once upfront!

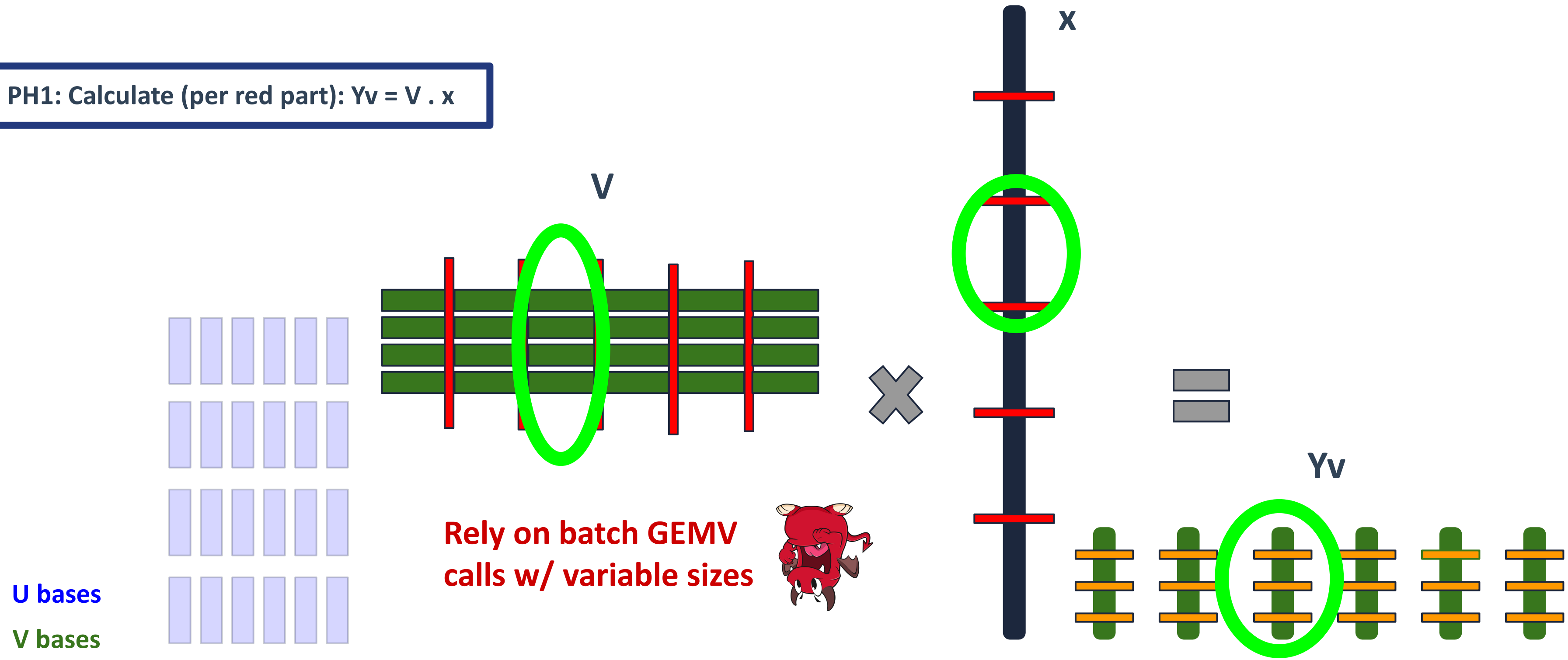


U bases

V bases

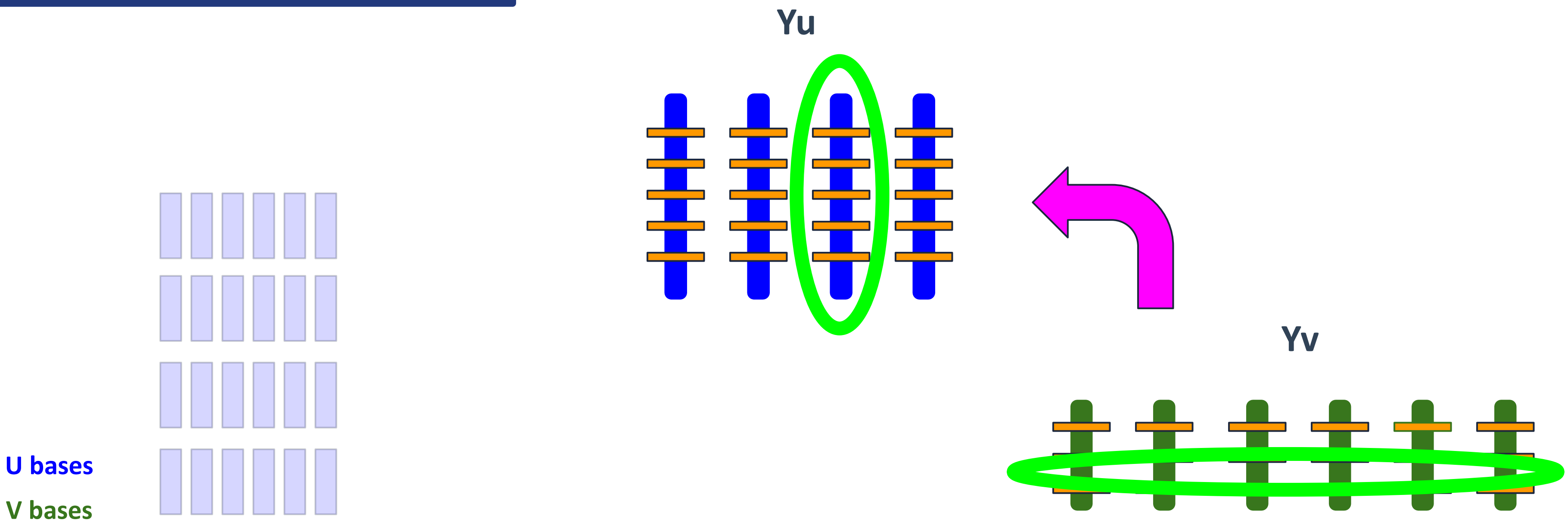
Tile Low-Rank Matrix-Vector Multiplication

PH1: Calculate (per red part): Yv = V . x



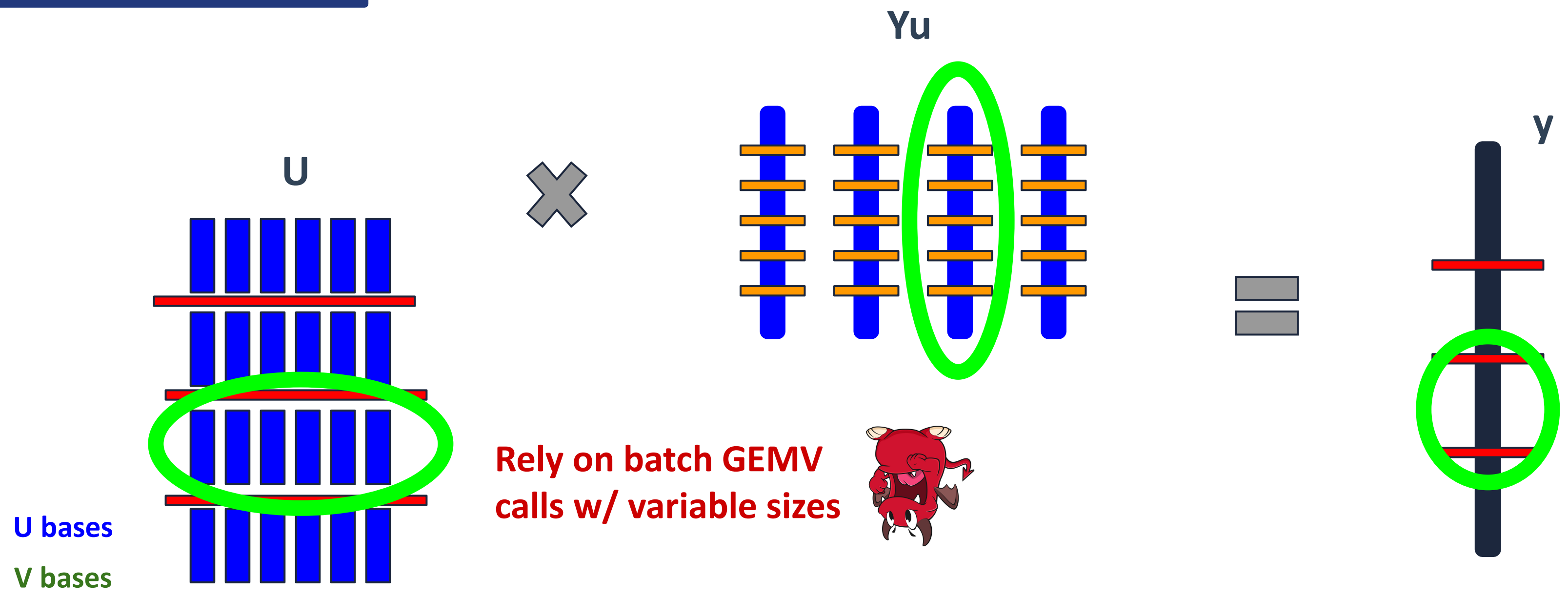
Tile Low-Rank Matrix-Vector Multiplication

PH2: Translate Yv (V bases) to Yu (U bases)

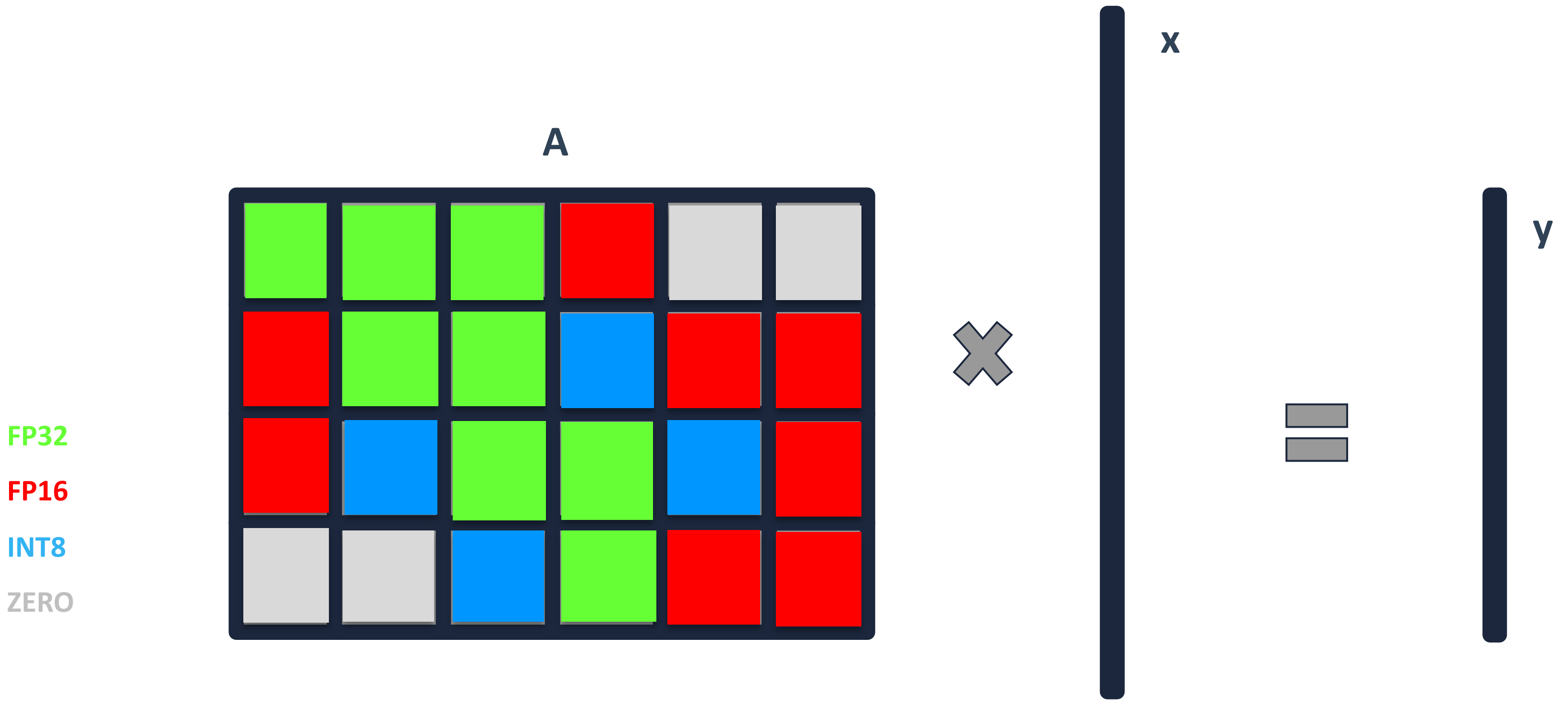


Tile Low-Rank Matrix-Vector Multiplication

PH3: Calculate y = U . Yu



Mixed precisions

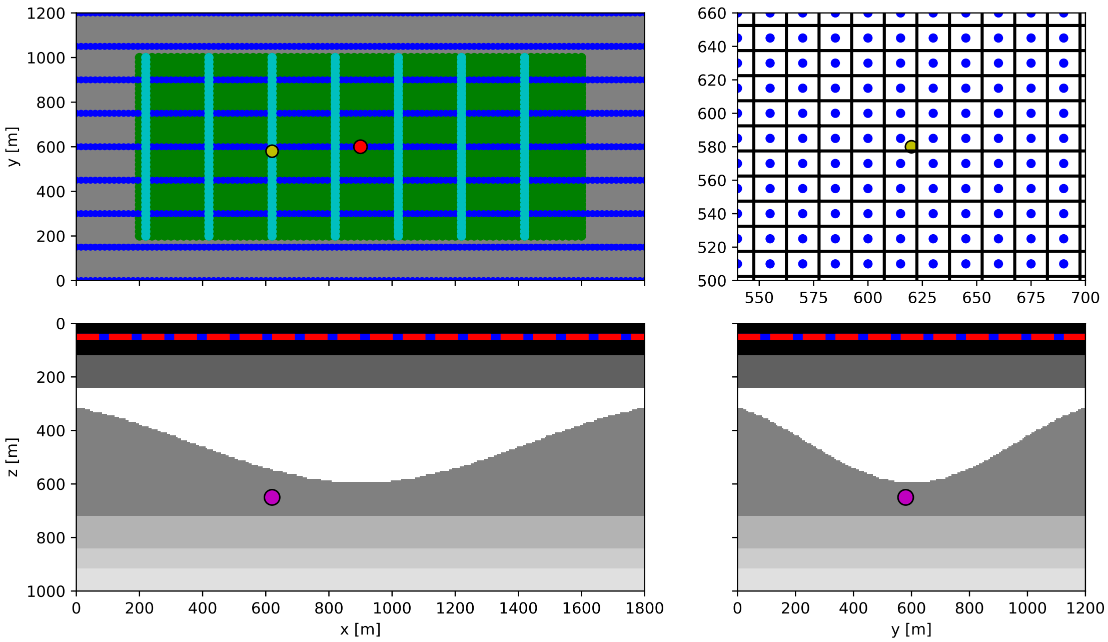


Mixed-precision hardware support

GEMV/GEMM capabilities

Hardware	FP32	FP16	INT8
intel	✓	X	X
AMD	✓	✓	✓
NEC	✓	X	X
FUJITSU	✓	✓	✓
NVIDIA	✓	✓	✓

Numerical Example



- Constant velocity, variable density model of size 1.8km x 1.2km x 1km.

- Regular carpet of sources and receivers

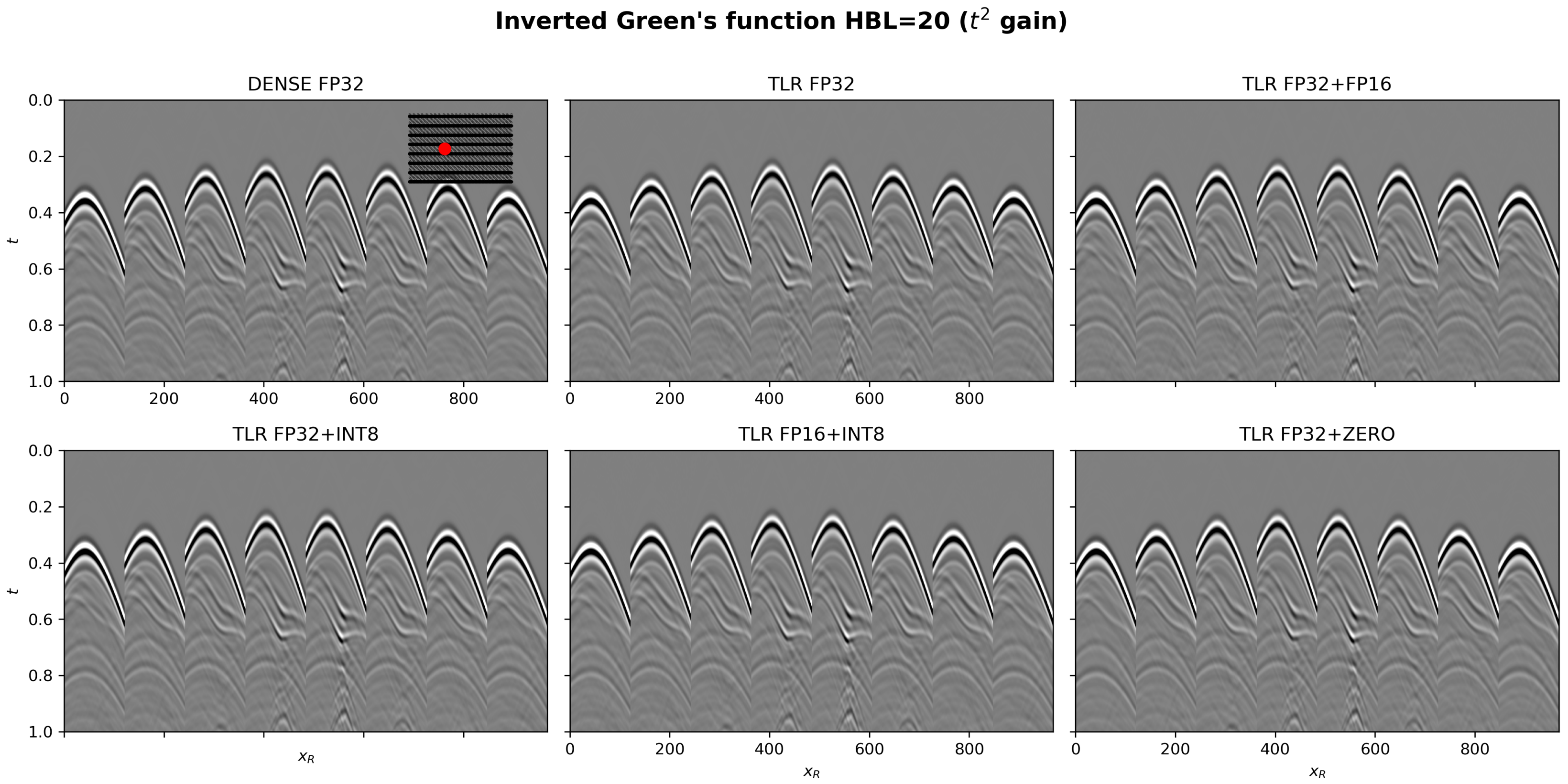
$N_S = N_R = 9801, \quad dx_S = dx_R = 15m$

Redatuming carpet @ 650m

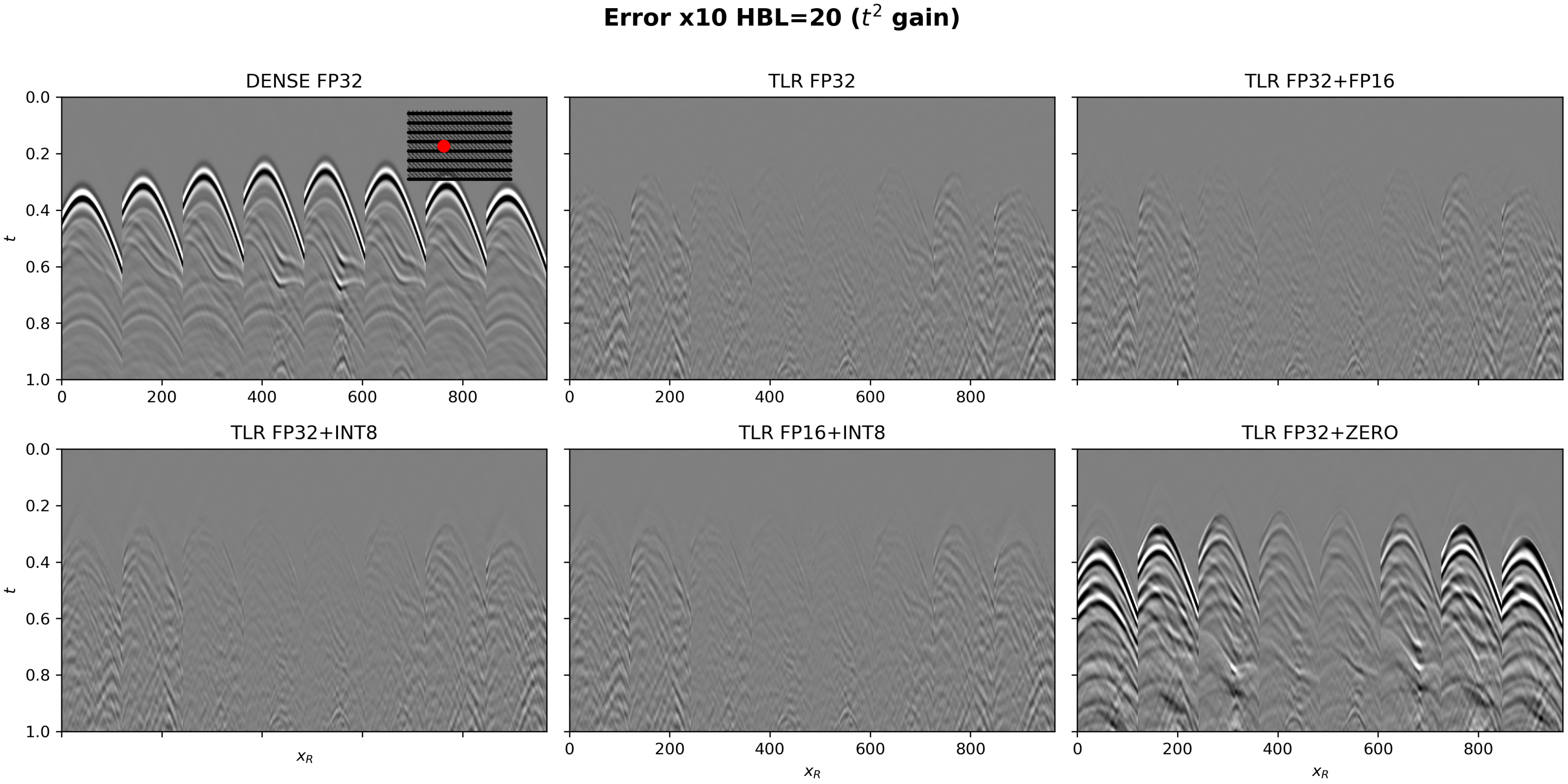
Marchenko lines in displays

More details in Ravasi and Vasconcelos, 2021 - GPY

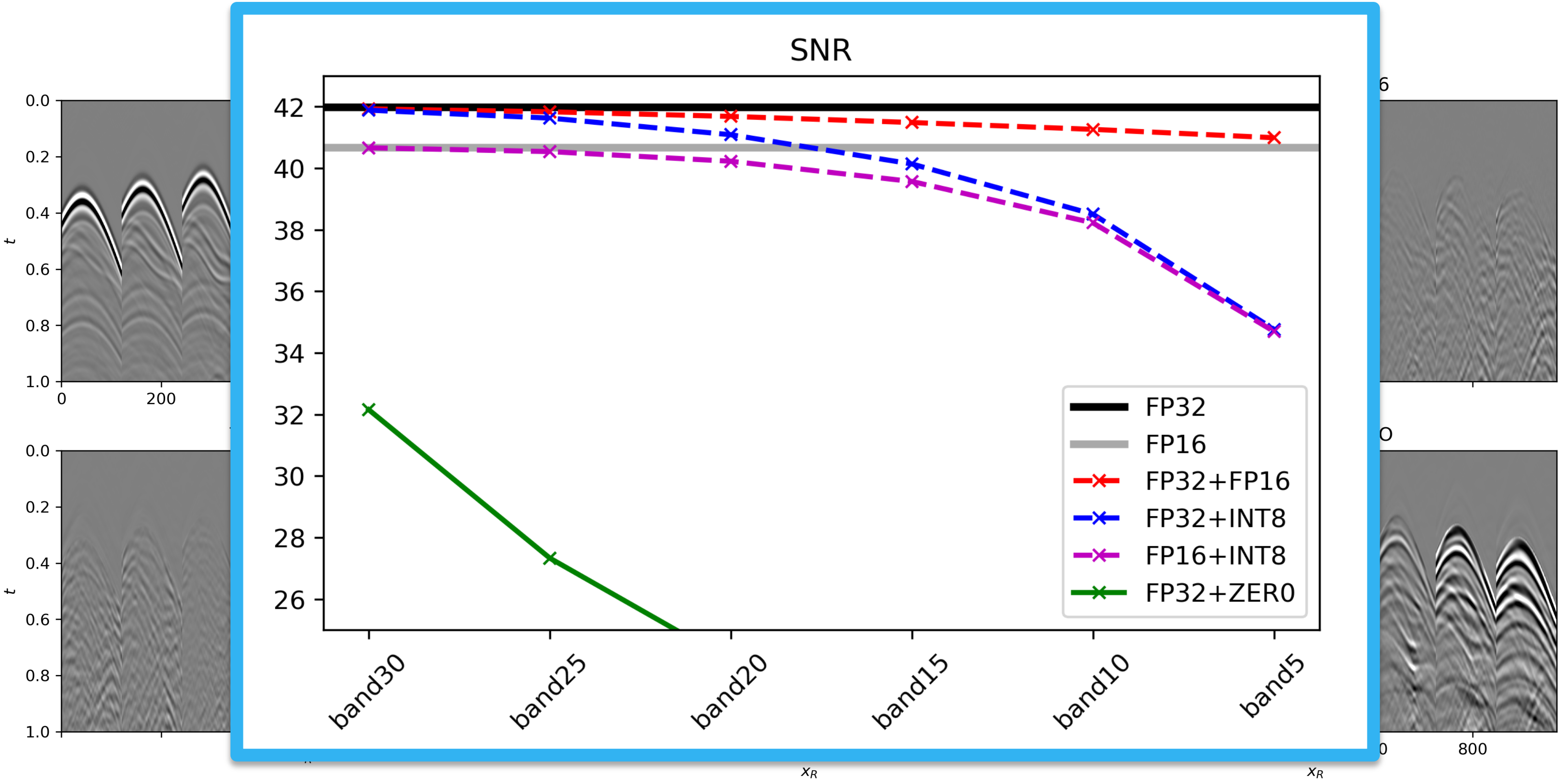
Numerical Example



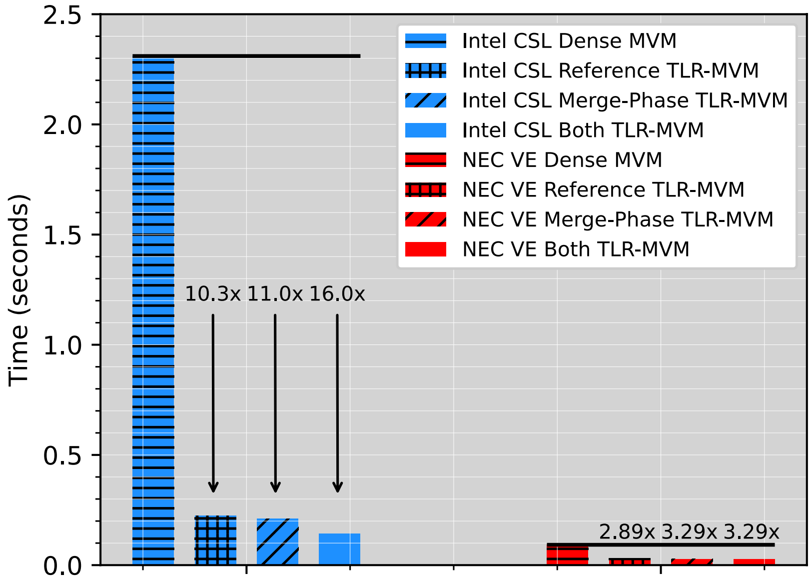
Numerical Example



Numerical Example



TLR-MVM Performance tests



CascadeLake

SX-Aurora TSUBASA

EPYC Rome

A64FX

A100

Coming soon!

More details in Hong at al., 2021 - SFI

And the oneAPI dream

```
// phase 1
#pragma omp parallel for
for(int i=0; i<tlrmvmptr->Ntg; i++){
    if(tlrmvmptr->colsum[i] != 0){
        cblas_cgemv(CblasColMajor, CblasNoTrans, tlrmvmptr->AvMs[i],
        tlrmvmptr->AvKs[i], &alpha, tlrmvmptr->h_Avbp[i],
        tlrmvmptr->AvMs[i], tlrmvmptr->h_xbp[i],
        1, &beta, tlrmvmptr->h_yvbp[i], 1);
    }
}
```

OpenMP

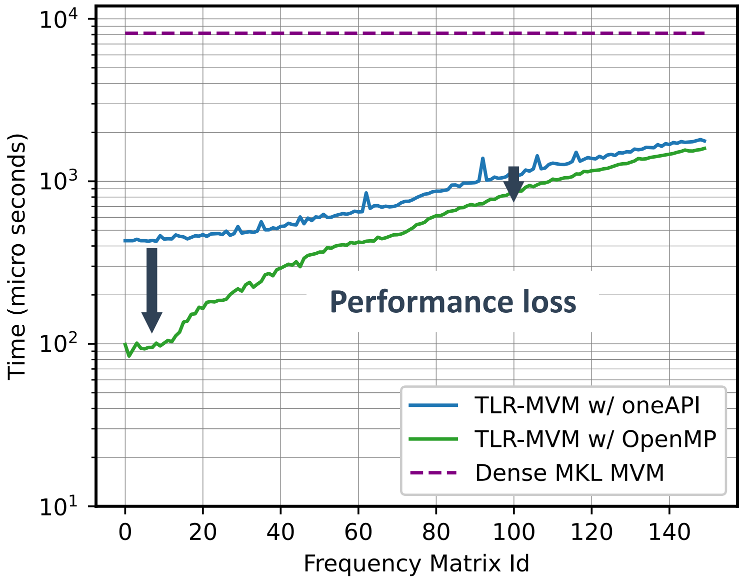
Simplified code for Phase 1

```
// phase 1
for(int i=0; i<tlrmvmptr->Ntg; i++){
    if(tlrmvmptr->colsum[i] != 0){
        oneapi::mkl::blas::gemv(tlrmvmptr->device_queue,
        transA, tlrmvmptr->AvMs[i],
        tlrmvmptr->AvKs[i], alpha, tlrmvmptr->h_Avbp[i],
        tlrmvmptr->AvMs[i], tlrmvmptr->h_xbp[i],
        1, beta, tlrmvmptr->h_yvbp[i], 1);
    }
}
tlrmvmptr->device_queue.wait();
```

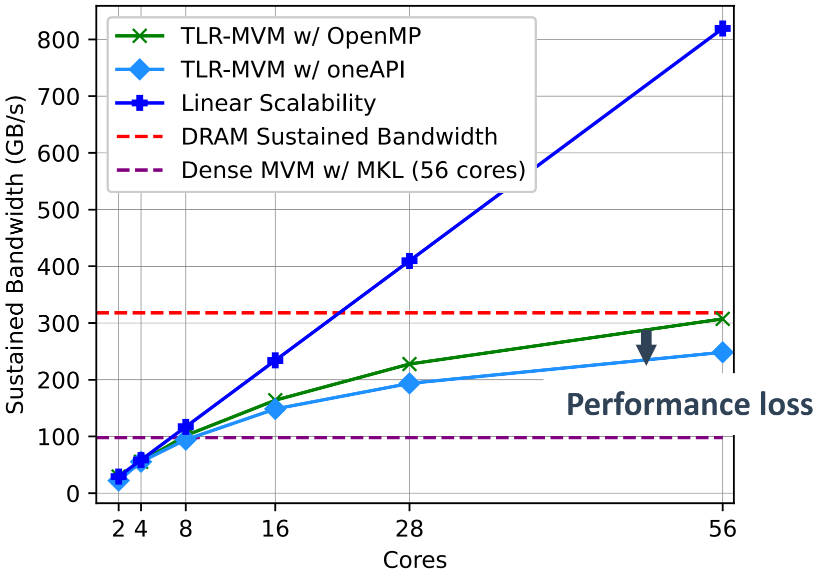
DPC++ Queue

And the oneAPI dream

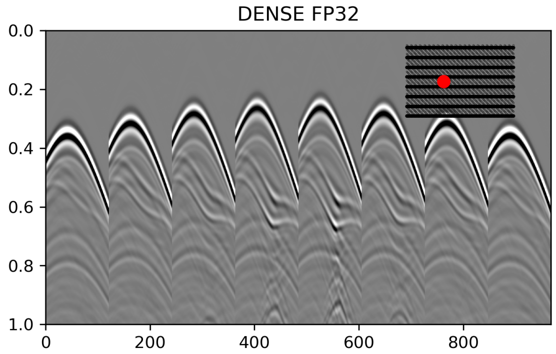
Time-to-solution (56 cores)

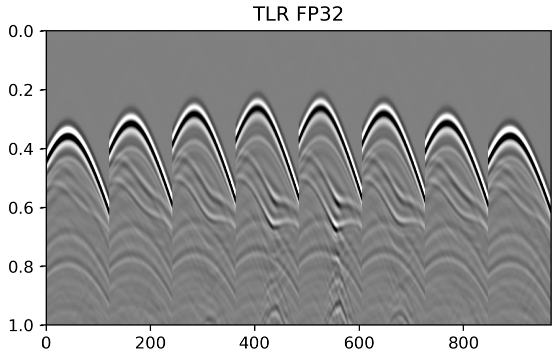


Sustained bandwidth (Freq Id=100)



Conclusion





Can you spot the difference?
… *your energy bill will!*

<u>TLR-based MDC (NEC 128VEs)</u>

Size reduction: x**3.7**
Time reduction: **x3**
Sustained bandwidth: **110 TB/s**

References

- M Ravasi, I Vasconcelos , An open-source framework for the implementation of large-scale integral operators with flexible, modern high-performance computing solutions: Enabling 3D Marchenko imaging by least-squares inversion, Geophysics 86 (5), WC177-WC194.

- Y Hong, H Ltaief, M Ravasi, L Gatineau, DE Keyes, Accelerating Seismic Redatuming Using Tile Low-Rank Approximations on NEC SX-Aurora TSUBASA, Supercomputing Frontiers and Innovations.

Thank you for listening!

07. 12. 2021

Deep imaging group.

King Abdullah University of Science and Technology