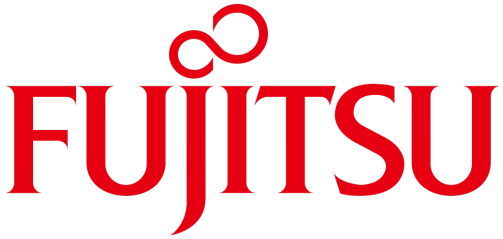
HIGH PERFORMANCE SEISMIC REDATUMING ON A64FX



Speaker: Yuxi Hong

yuxi.hong@kaust.edu.sa

Co-authors

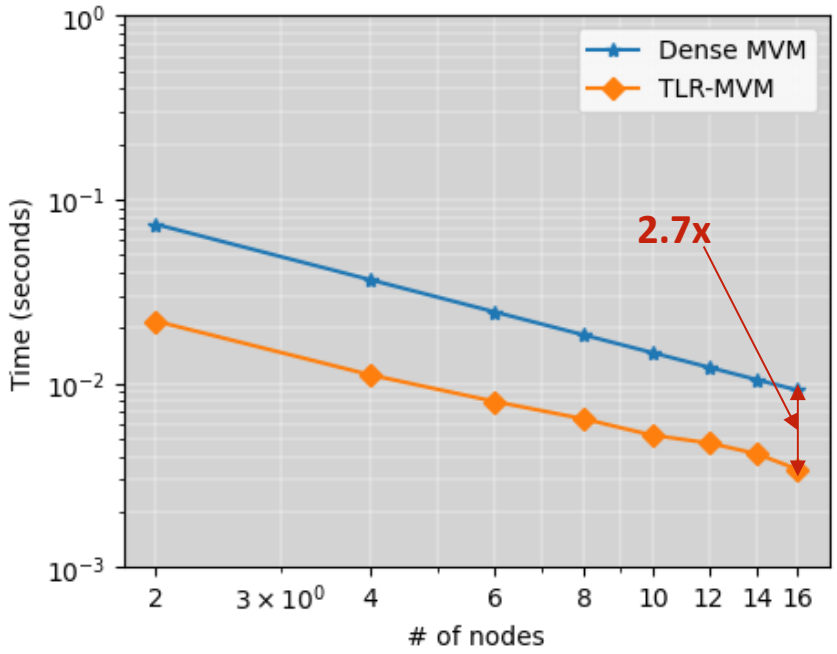
Matteo Ravasi

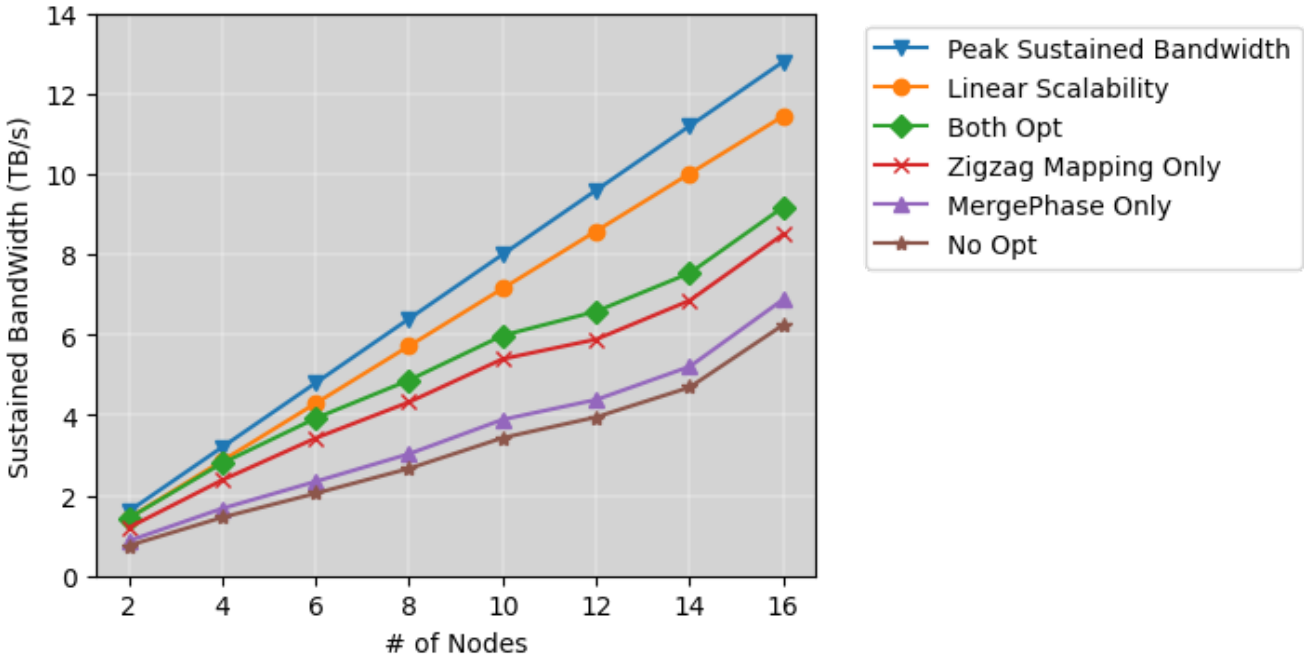
Yuxi Hong

David Keyes

Hatem Ltaief

What is this presentation about?





- Leverage low-rank matrix computations to accelerate large seismic redatuming application.
- Achieve an average of 2.7x acceleration compared to dense MVM.
- Score around 9.5 TB/s (78 % sustained bandwidth) using 16 Fujitsu A64FX 1000 nodes.

I. Powering seismic redatuming with TLR-MVM

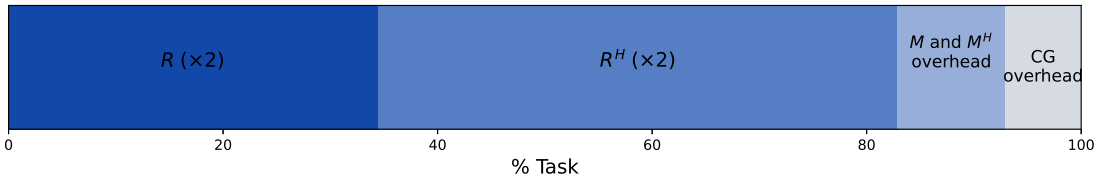
Seismic redatuming is an important technique to get insight into the Earth's subsurface.

This requires solving an inverse problem. Traditionally, due to computational challenges, only the adjoint is applied.

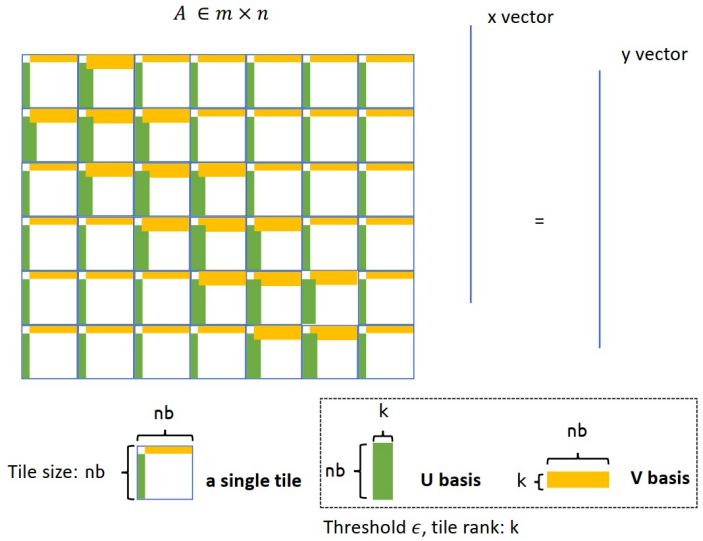
Some latest research show an alternative method to improve the solution of inverse problems by using an iterative solver, e.g., conjugate gradient iterative solver. This comes at the cost of evaluating multiple expensive MVM operations, as shown in the following equations:

$$\mathbf{x} = \mathbf{R}^H \mathbf{y}: \quad x(t, \mathbf{x}_R, \mathbf{x}_A) = \mathcal{F}^{-1}_{\omega_{max}} \left(\int_{\delta\mathbb{D}} R^*(\omega, \mathbf{x}_B, \mathbf{x}_R) \mathcal{F}_{\omega_{max}} (y(t, \mathbf{x}_B, \mathbf{x}_A)) d\mathbf{x}_B \right),$$

$$\mathbf{y} = \mathbf{R}\mathbf{x}: \quad y(t, \mathbf{x}_B, \mathbf{x}_A) = \mathcal{F}^{-1}_{\omega_{max}} \left(\int_{\delta\mathbb{D}} R(\omega, \mathbf{x}_B, \mathbf{x}_R) \mathcal{F}_{\omega_{max}} (x(t, \mathbf{x}_R, \mathbf{x}_A)) d\mathbf{x}_R \right).$$



We use tile low-rank matrix-vector multiplication (TLR-MVM) to mitigate the complexity bottleneck.



Phase 1: Multiply V bases by x vector.

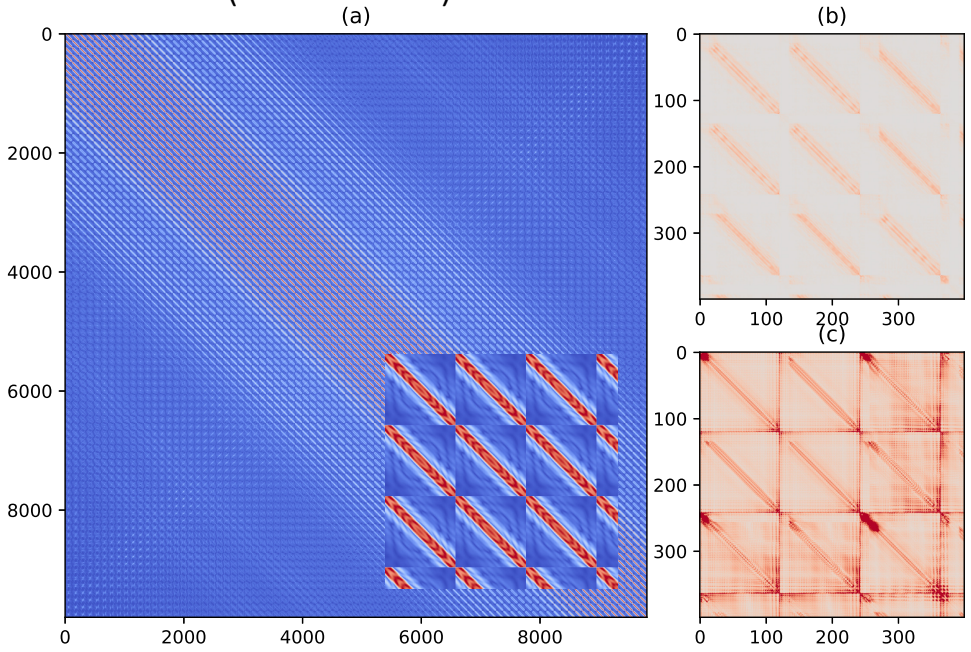
Phase 2: Shuffle results of Phase 1 to yu vector.

Phase 3: Multiply U bases to yu vector.

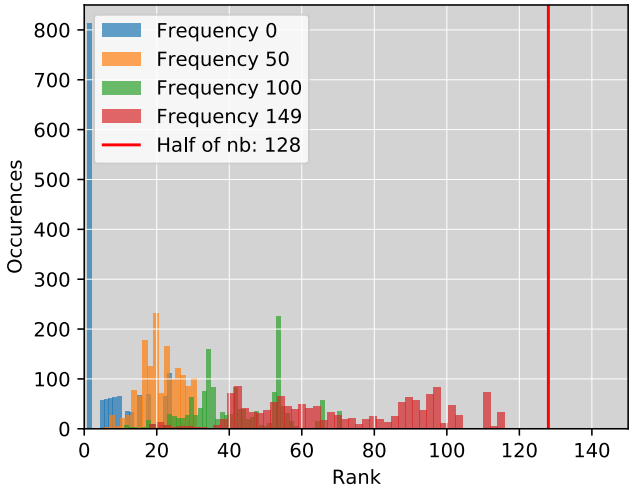
II. Dataset analysis

The workload is 150 MVMs, where each matrix size is 9801 x 9801. Each matrix corresponds to information from a single frequency.

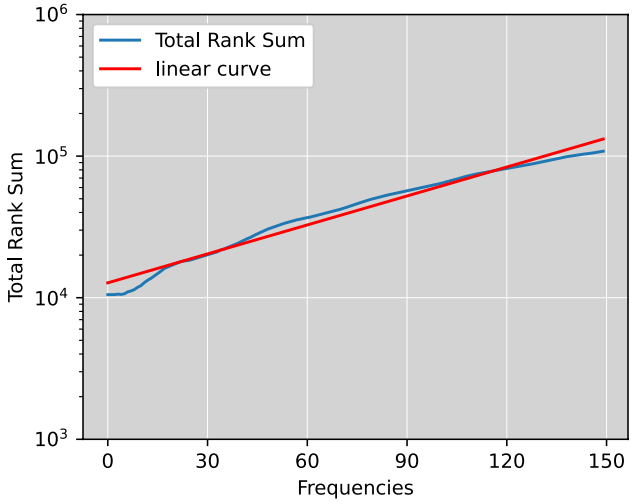
- Low rankness matrix structure of frequency 33 Hz (index=100):



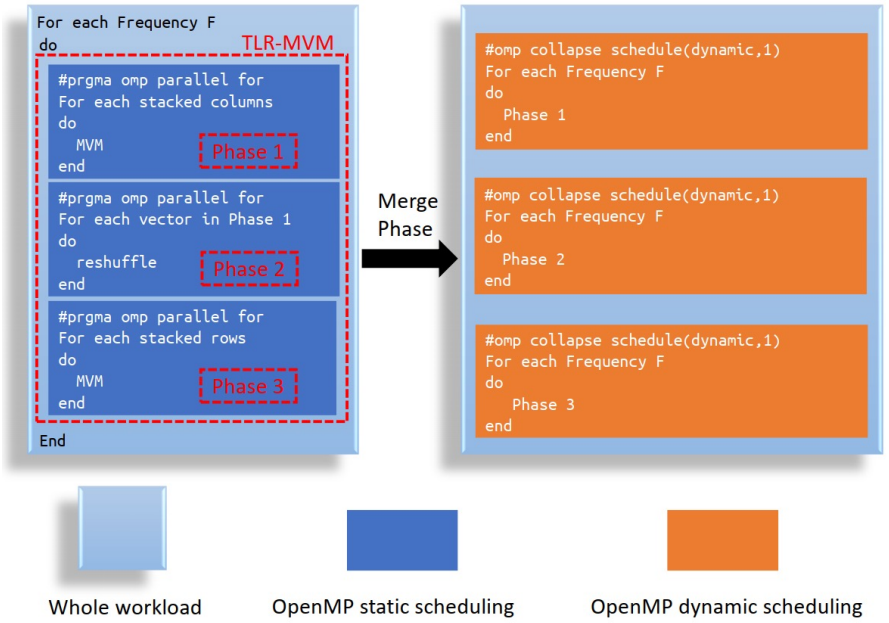
- Rank distribution of tiles. Most matrix ranks are below the critical threshold.

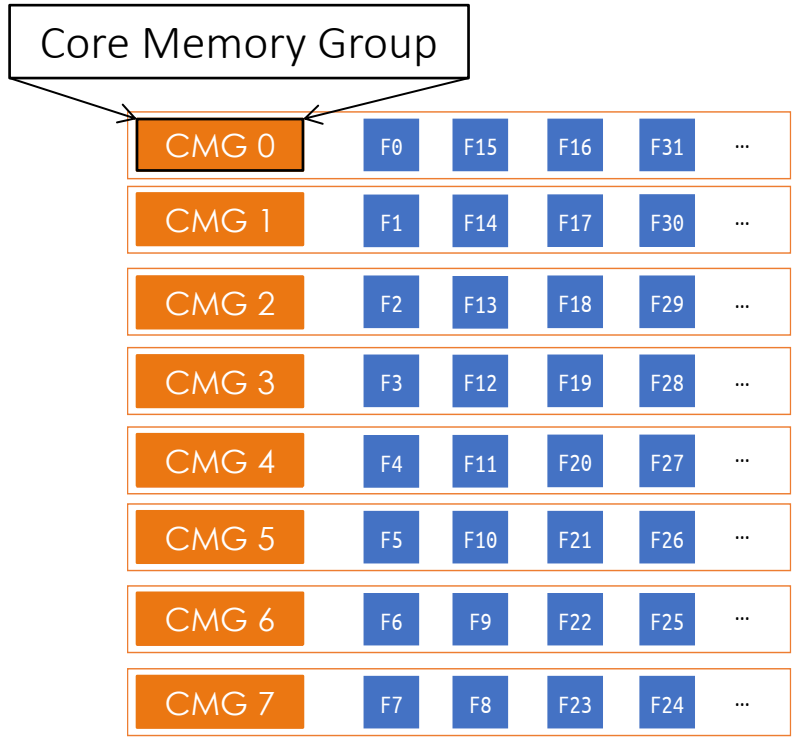


- Rank summation is larger as the matrix frequency index increases.



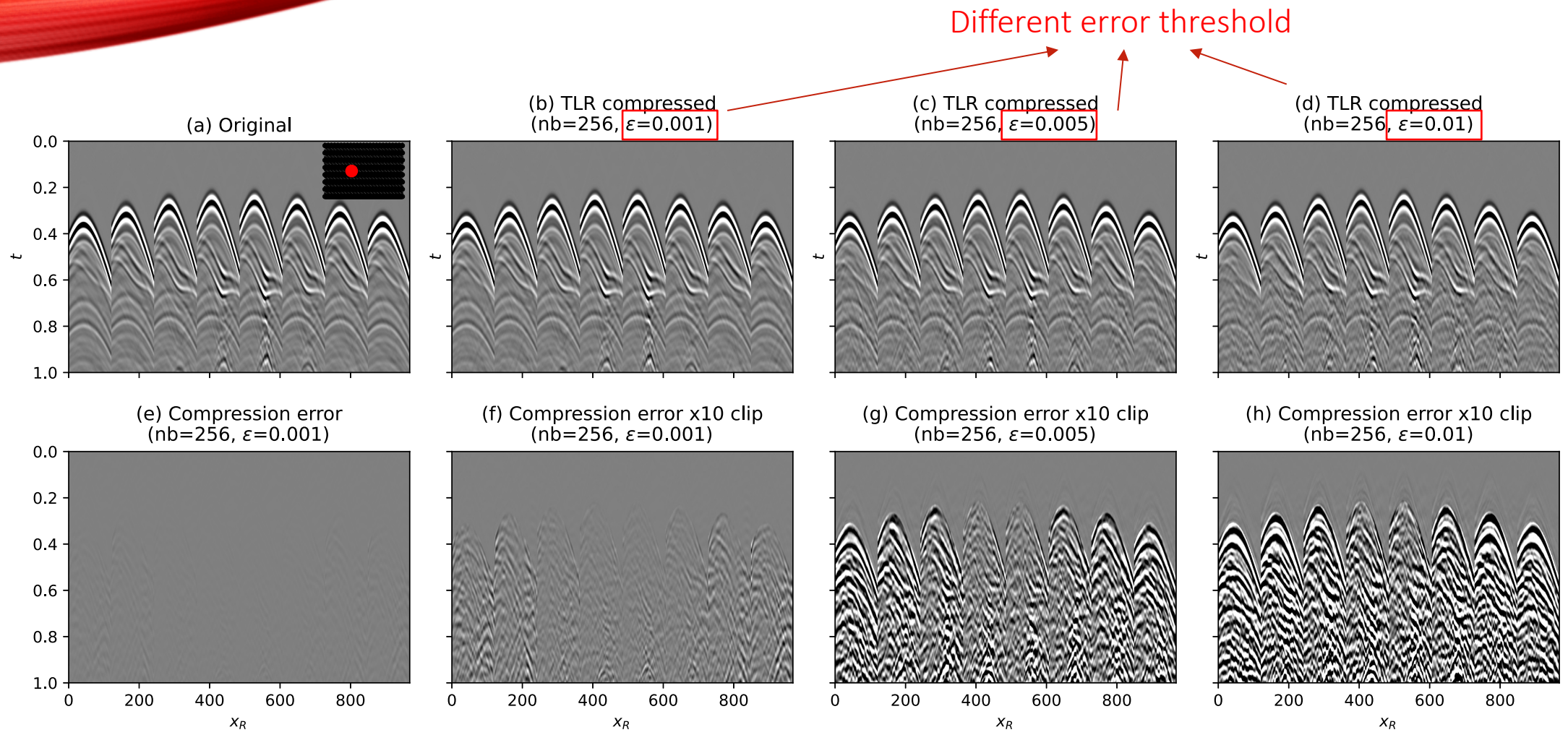
III. Load balancing optimizations





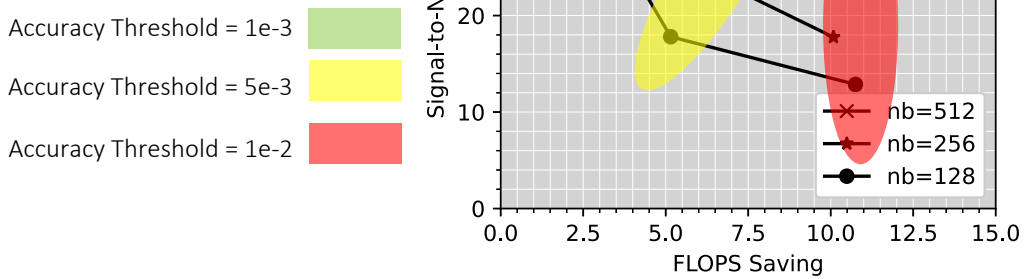
- *Phase Merge Strategy* and switch static scheduling to dynamic scheduling for intra-node load balancing.
- *Zigzag Mapping strategy* for inter-node load balancing.

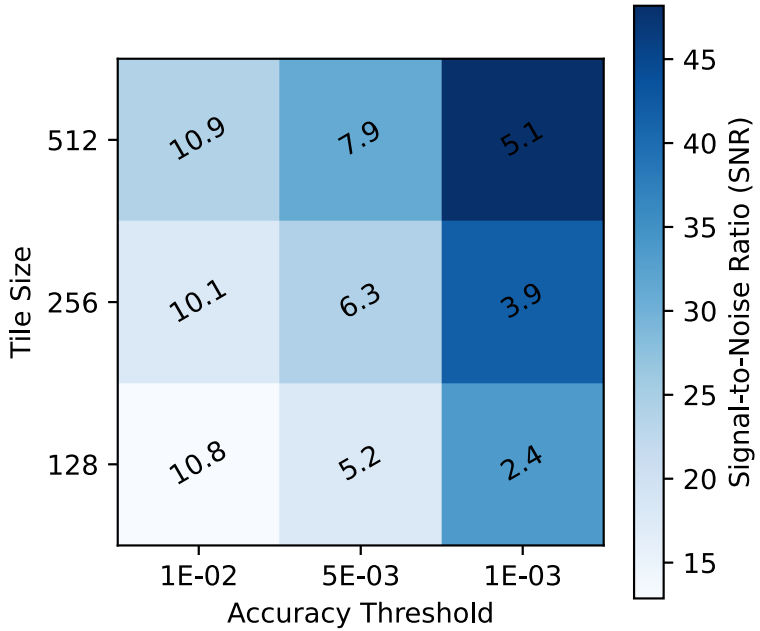
IV. Tile-Low Rank MVM approximation



- TLR-MVM introduces negligible error to the final images.

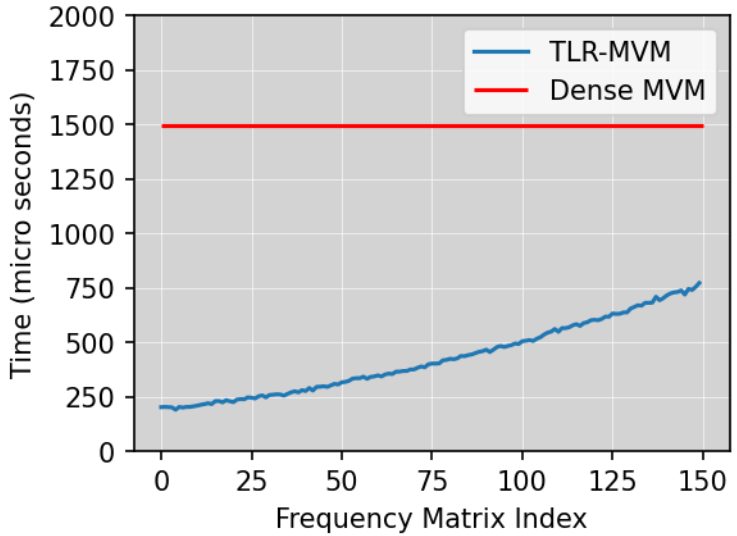
V. Application SNR vs different algorithmic configurations

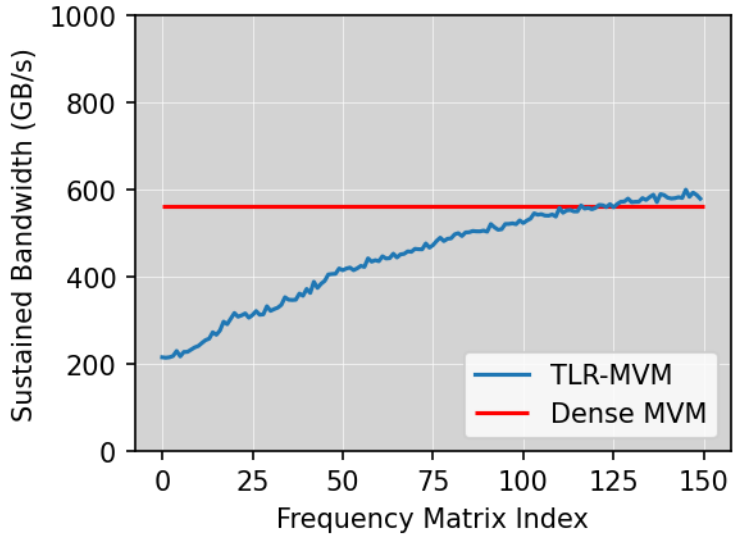




- There are 2 parameters for user to tune the algorithms to trade off FLOPS saving and application accuracy, i.e., error (or accuracy) threshold and tile size.
- In seismic application, signal-to-noise ratio (SNR) is used to quantify the quality of the results. We test on different tile sizes (nb) and accuracy thresholds. In the left figure, from left to right the error threshold is 1e-3, 5e-3, and 1e-2.
- We set 40 as the SNR threshold and find two eligible configurations. We conduct subsequent experiments using nb = 256 and error accuracy 1e-3.

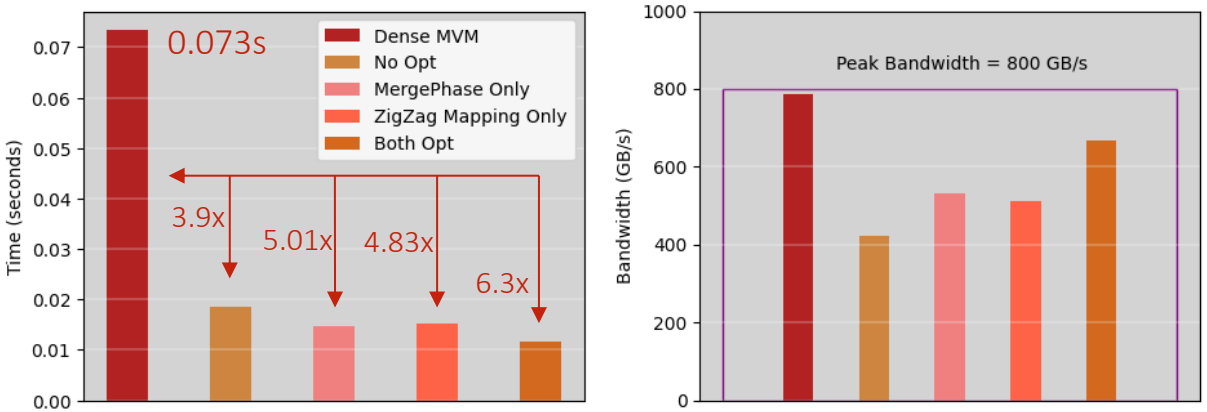
VI. Time analysis of each single frequency matrix



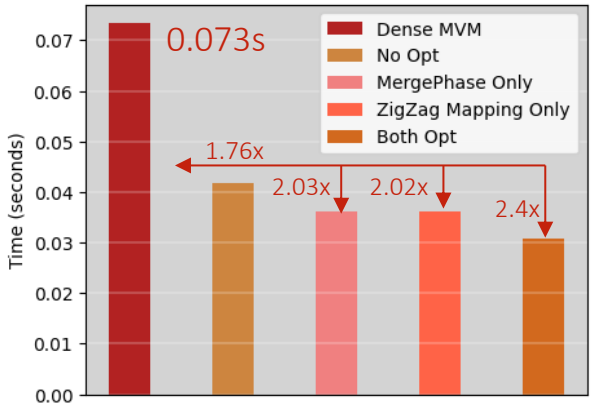


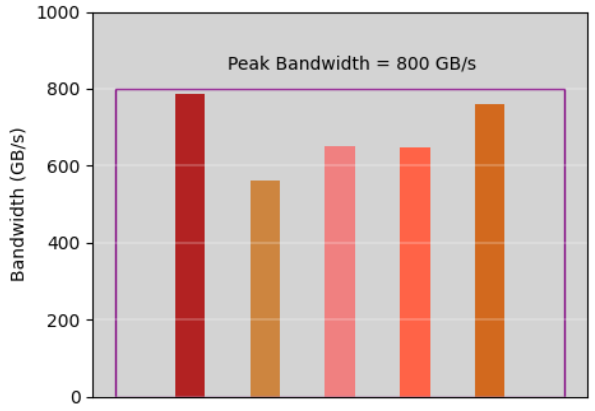
- Higher frequencies take more time to process.
- For each single frequency, we are better than dense MVM.
- For Dense MVM we use SCALAPACK Parallel MVM and map each mpi process to one CMG.

VII. Single-node TLR-MVM performance



Single node with frequency 0 to frequency 74



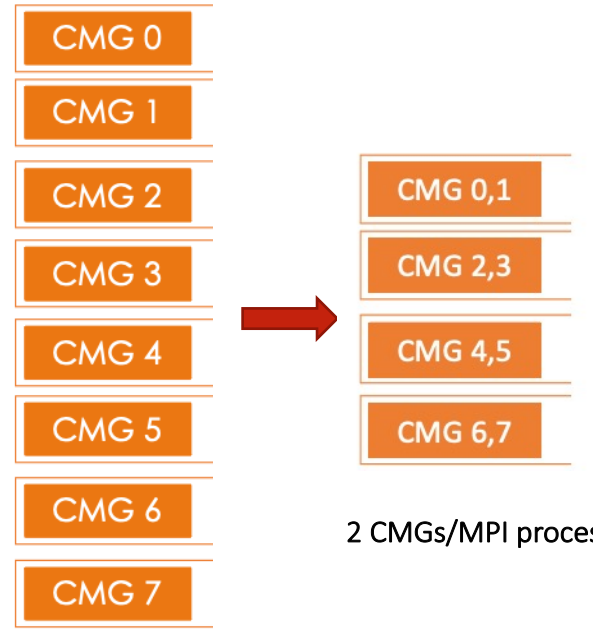


Single node with frequency 75 to frequency 149

- We split the 150 frequency matrices in ascending order into two parts. We use the two sets to test two optimization strategies for load balancing on a single Fujitsu A64FX 1000 node.
- The second set is larger in terms of memory footprint and thus can achieve higher bandwidth compared to the first set.
- When we turn on both optimization strategies, we further saturate the sustained memory bandwidth.
- For Dense MVM, we feed 4 matrices into 1 node at the same time, which gives us best performance.

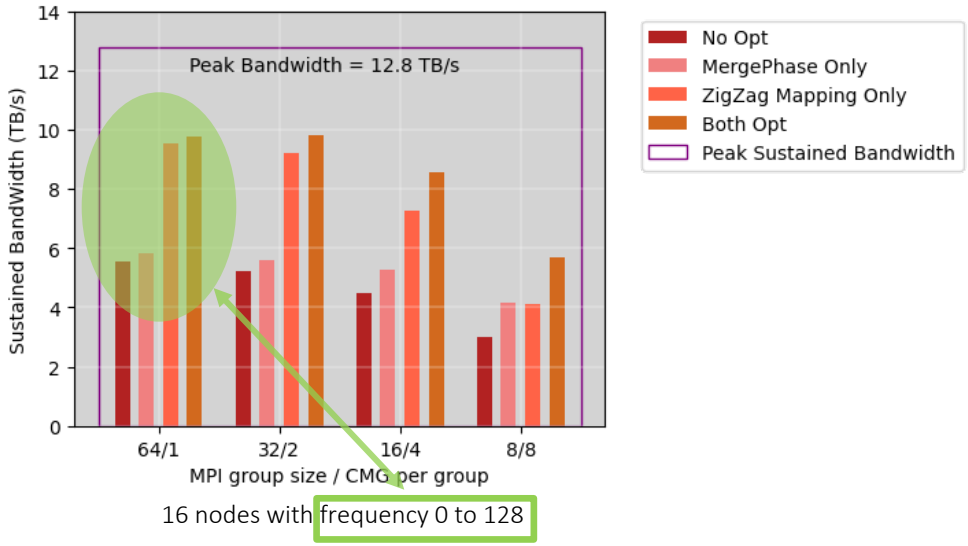
VIII. Enhance data locality with CMG-aware mapping

- We have 16 nodes in total. When the input matrices can not be evenly distributed across the nodes, it may lower performance.
- To further deal with load imbalance of input data, we propose to group several CMGs inside 1 MPI process.
- We find 2 CMGs per group gives us best performance.

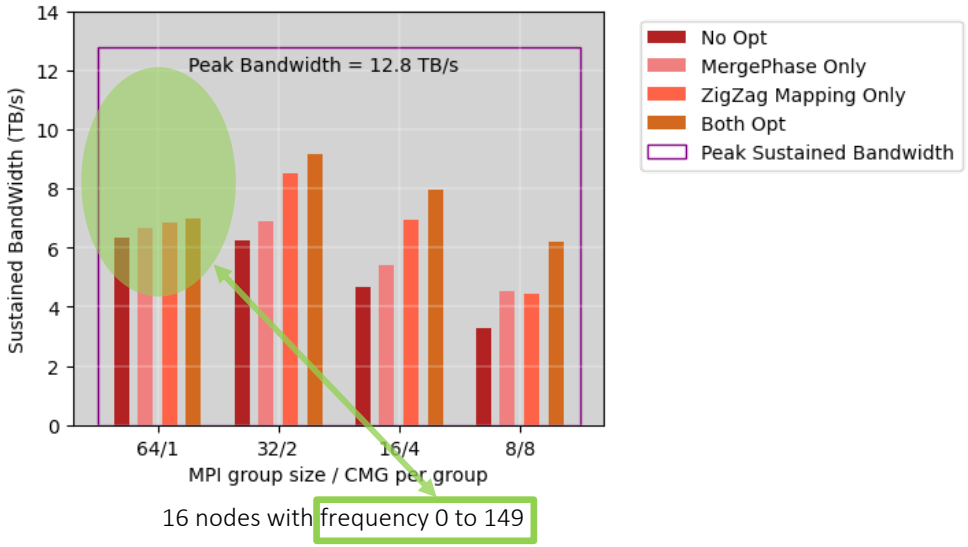


1 CMG/MPI process

2 CMGs/MPI process

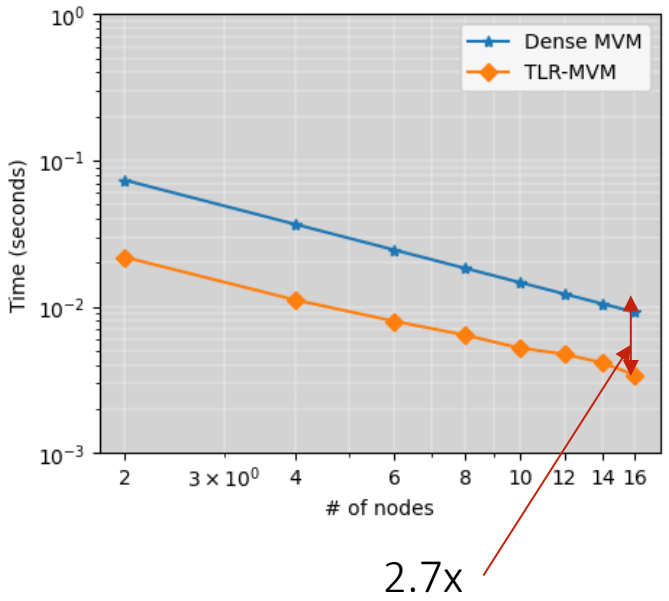


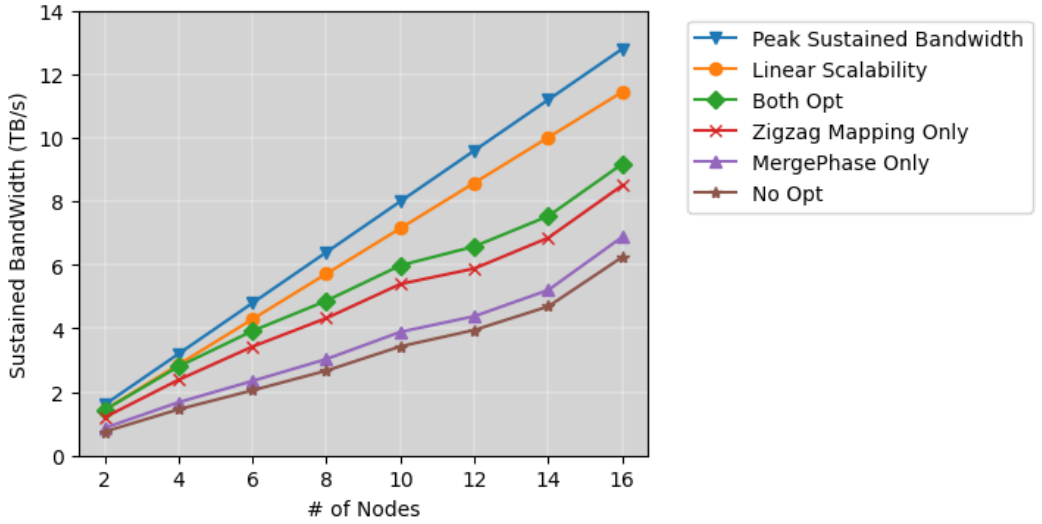
16 nodes with frequency 0 to 128



16 nodes with frequency 0 to 149

IX. Performance scalability results





- Compare optimized TLR-MVM against dense MVM.
- Achieve performance 2.7x faster than dense MVM.
- Obtain 78% sustained bandwidth using 16 Fujitsu A64FX 1000 nodes.

FUJITSU

KAUST

THANK YOU!

High Performance Seismic Redatuming on A64FX

We Thank Fujitsu Corp for remote hardware access.