

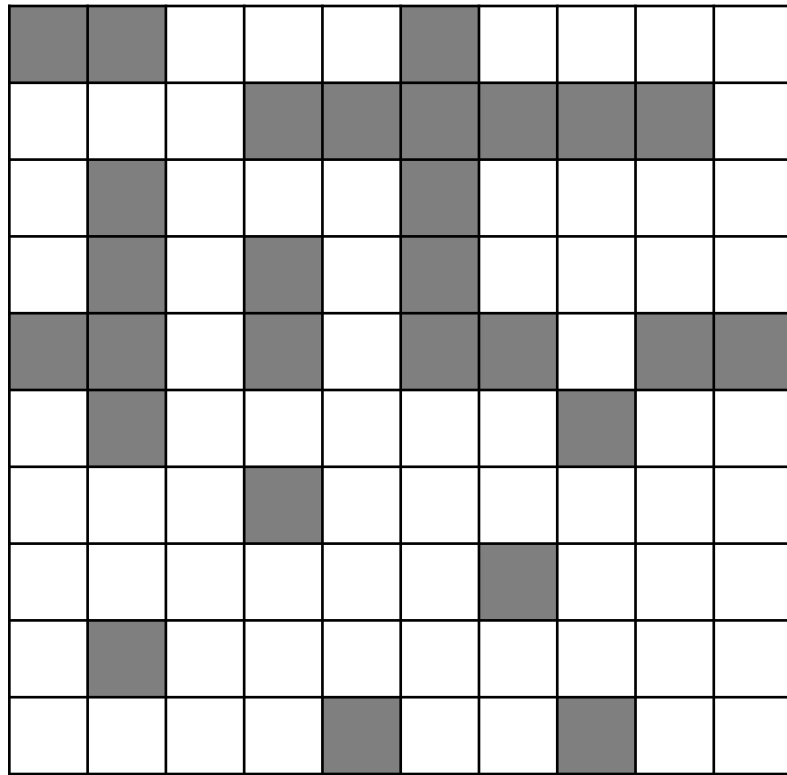
## Some advanced SpMV optimizations

Load balancing, reordering, sector cache, hardware barrier



# Load balancing

## Balancing by row



Thread 0 →  $N_{nz} = 21$

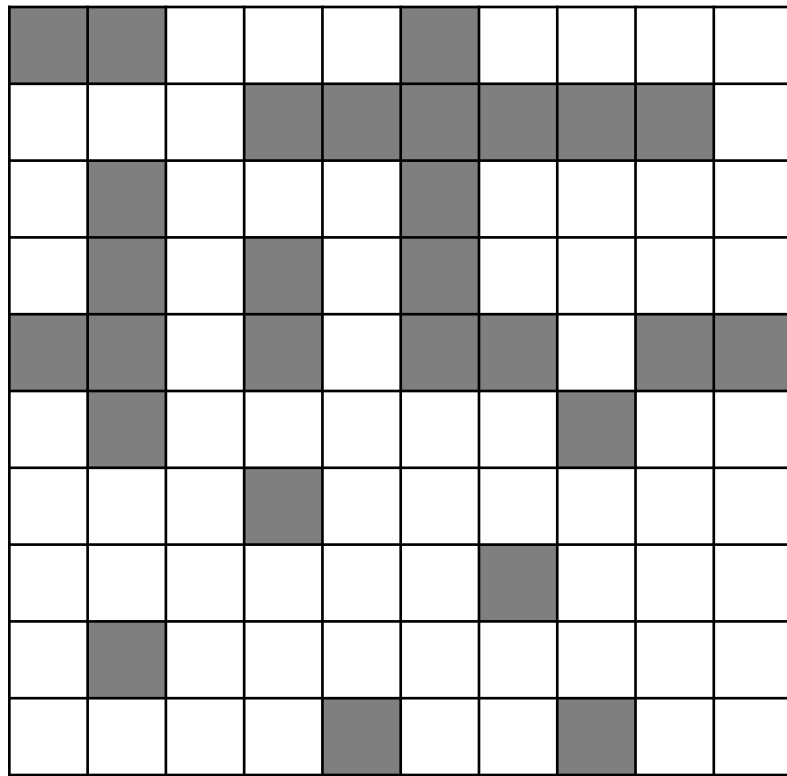
Thread 1 →  $N_{nz} = 7$



$$N_r = 10, N_{nz} = 28$$

# Load balancing

## Balancing by nnz

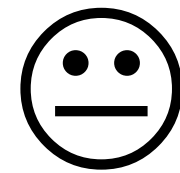


Thread 0 →  $N_{nz} = 14$

Thread 1 →  $N_{nz} = 14$



But many rows with  $N_{nzc} = 1$  → big  $\alpha$

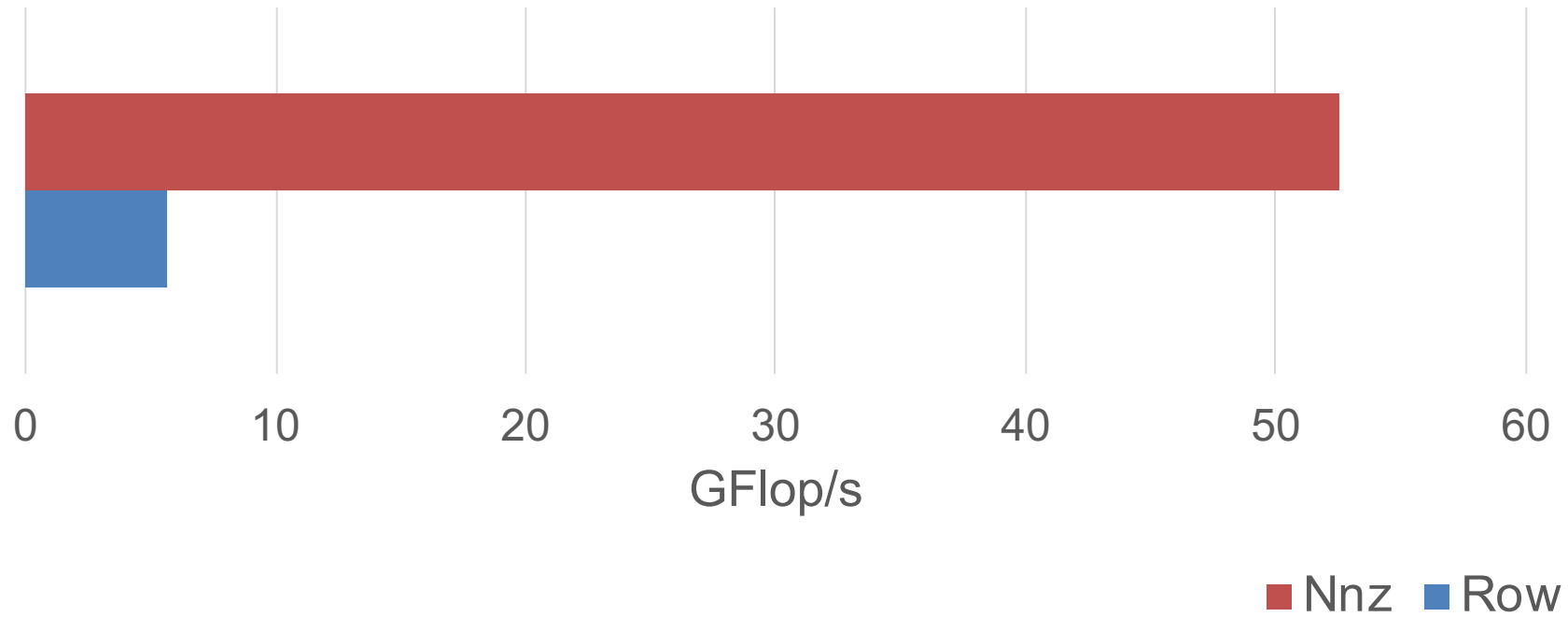


$$N_r = 10, N_{nz} = 28$$

Some form of clever guided or dynamic strategy might be good → NUMA problem

# Example (bundle\_adj matrix)

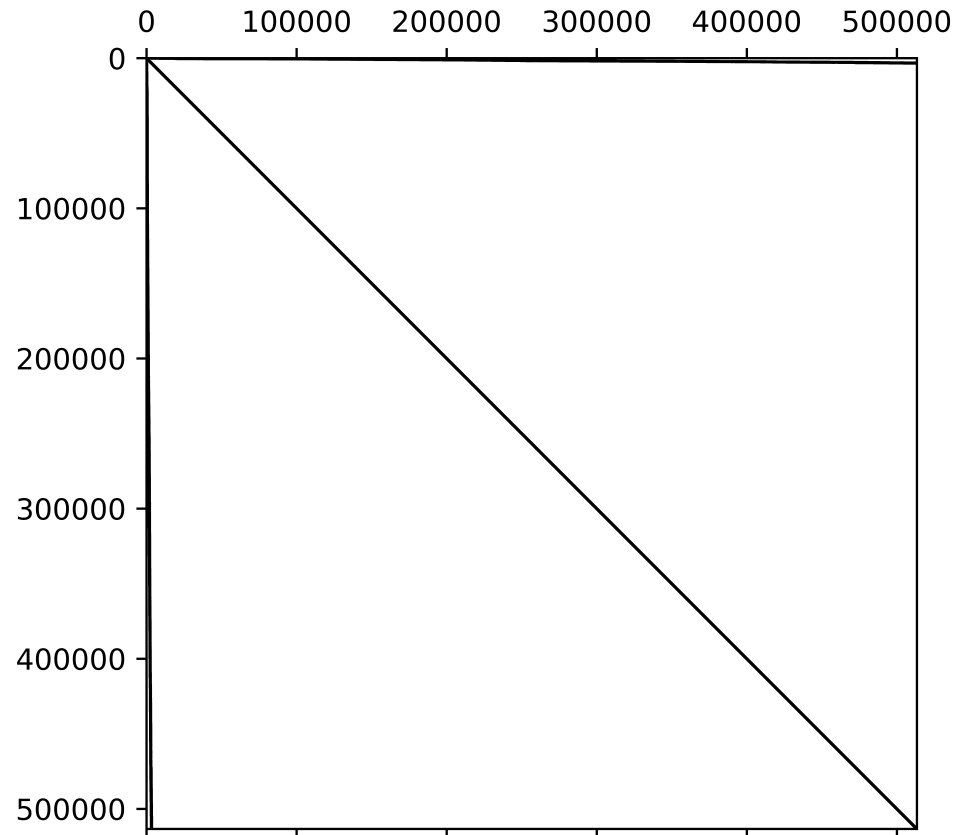
Performance on OOKAMI, bundle\_adj, C=32



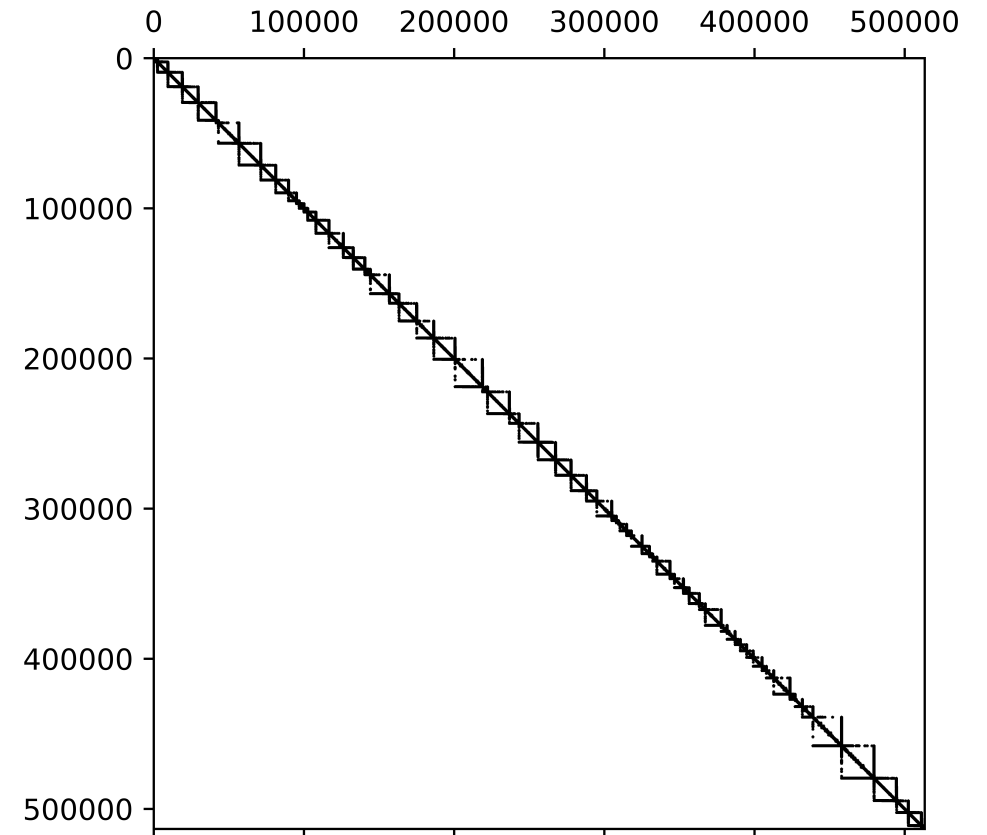
# Reordering



# Reordering



Original

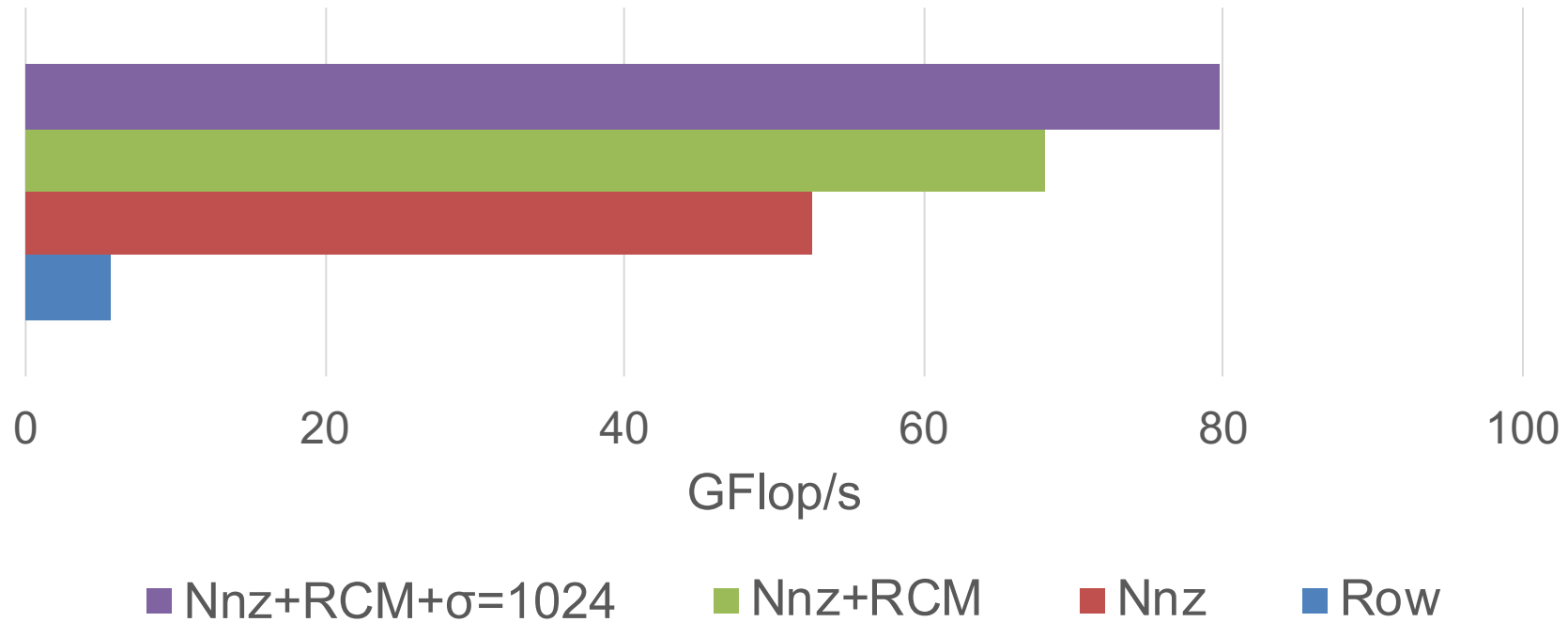


After RCM

Bundle\_adj matrix

# Example (bundle\_adj matrix)

Performance on OOKAMI, bundle\_adj, C=32

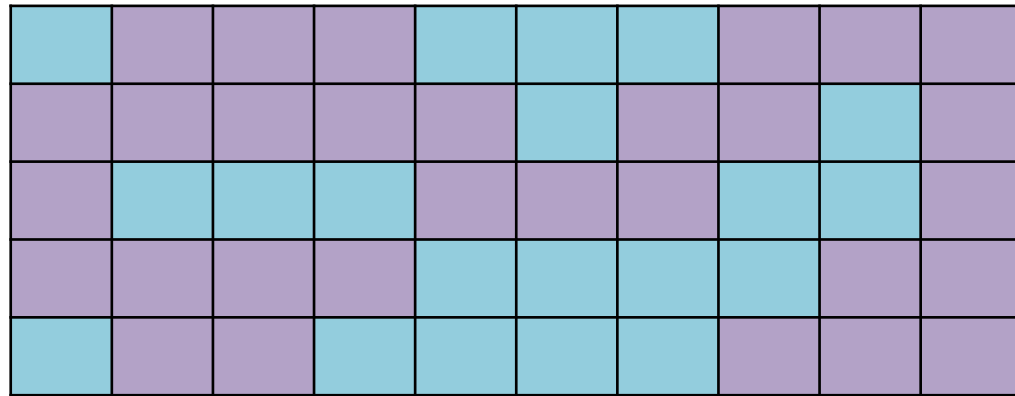


# Sector cache





# Sector cache




Normal cache



Sector cache

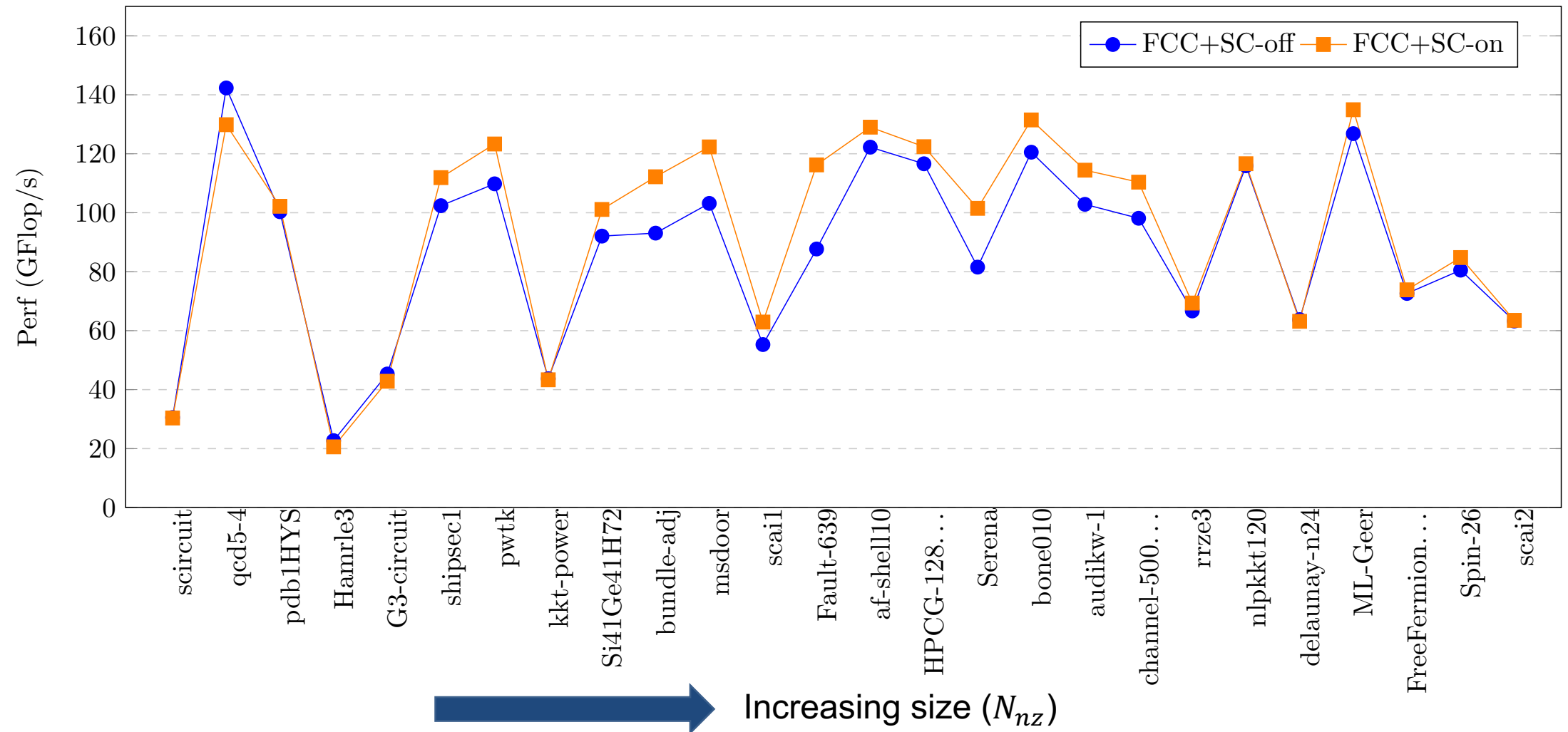


 Matrix (no-reuse)

 Vector (potential reuse)

# Performance

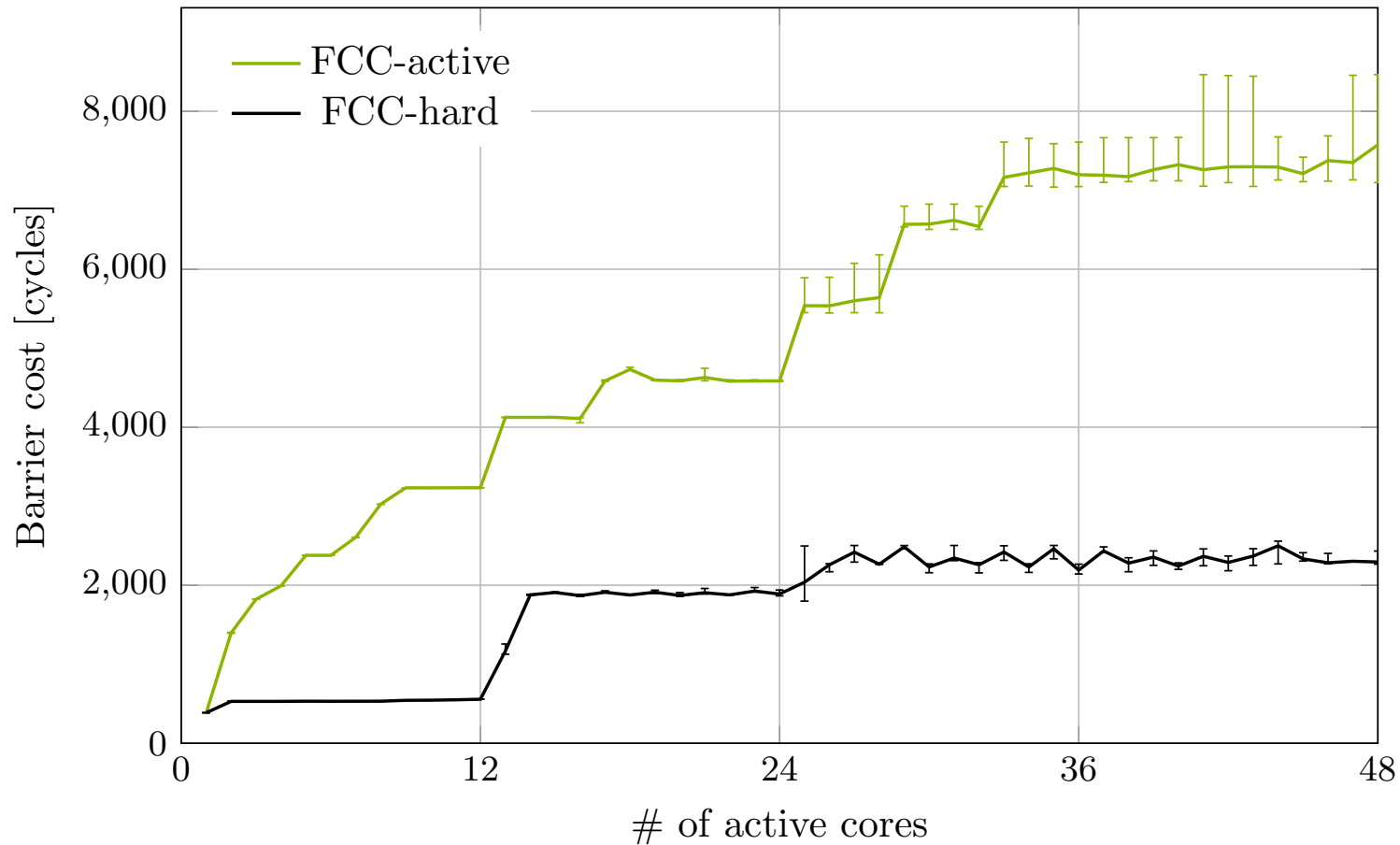
## SpMV performance on Fugaku (FX1000)



# Hardware barrier



# Hardware barrier



Worth a try for tiny matrices

For more details: ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX, accepted for publication in Concurrency and Computation: Practice and Experience, <https://arxiv.org/abs/2103.03013>