# TOWARDS AN AUTO-TUNED AND TASK-BASED SPMV (LASS LIBRARY)

SANDRA CATALÁN, TETSUZO USUI, LEONEL TOLEDO, XAVIER MARTORELL, JESÚS LABARTA, PEDRO VALERO-LARA

# SPMV

$$y := \alpha A x + \beta y$$

DENSE

SCALAR

SPARSE

# STATE OF THE ART

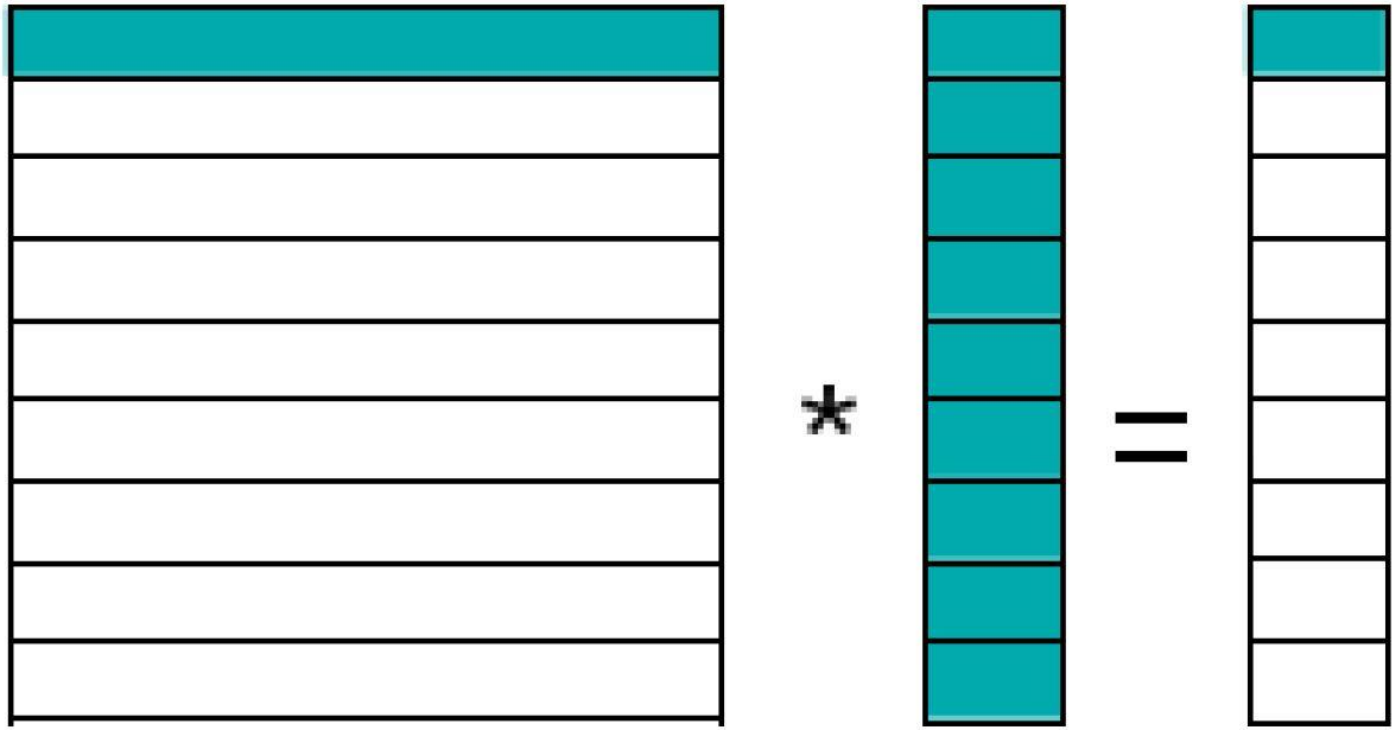
- Many libraries implement SpMV: MAGMA-Sparse, SuperLU, MUMPS -> LASs
- Many studies about best storage format -> CSR
- There are works about performance and different formats/programming models ->

  OmpSs + CSR +un/balancend matrices

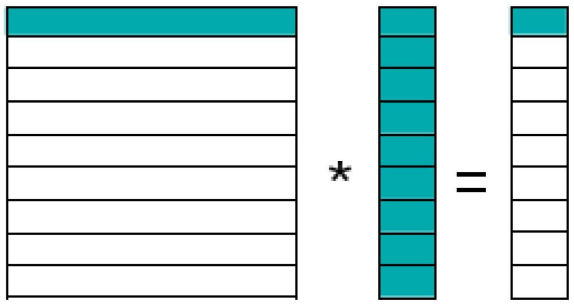
# PARALLELIZING SPMV

- One task per row
- Blocking
- Taskloop
- Grouping

# ONE TASK PER ROW



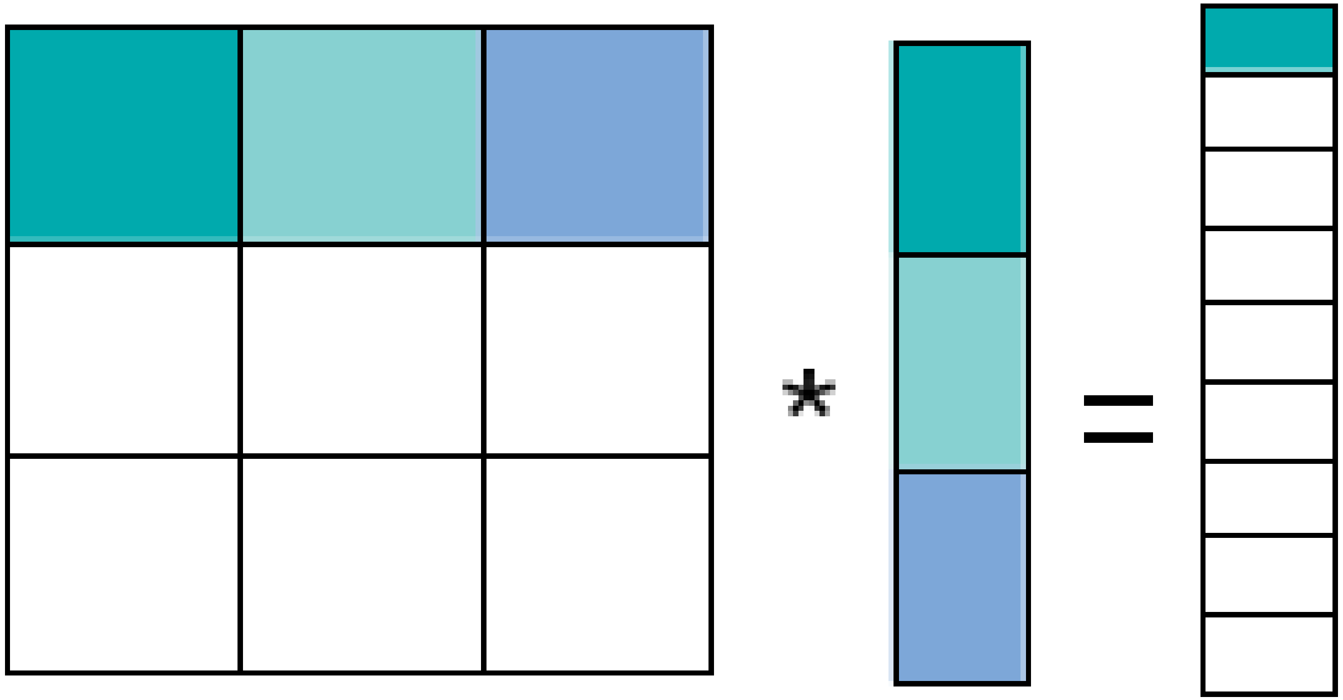
# ONE TASK PER ROW



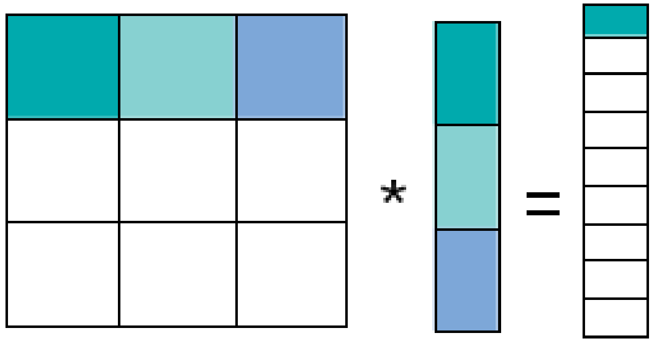

- ✓ Simple and straight-forward
- ✓ No data dependencies

- × Numerous tasks are created
- × Tasks usually very small (low number of non-zeros per row) -> overhead

# BLOCKING



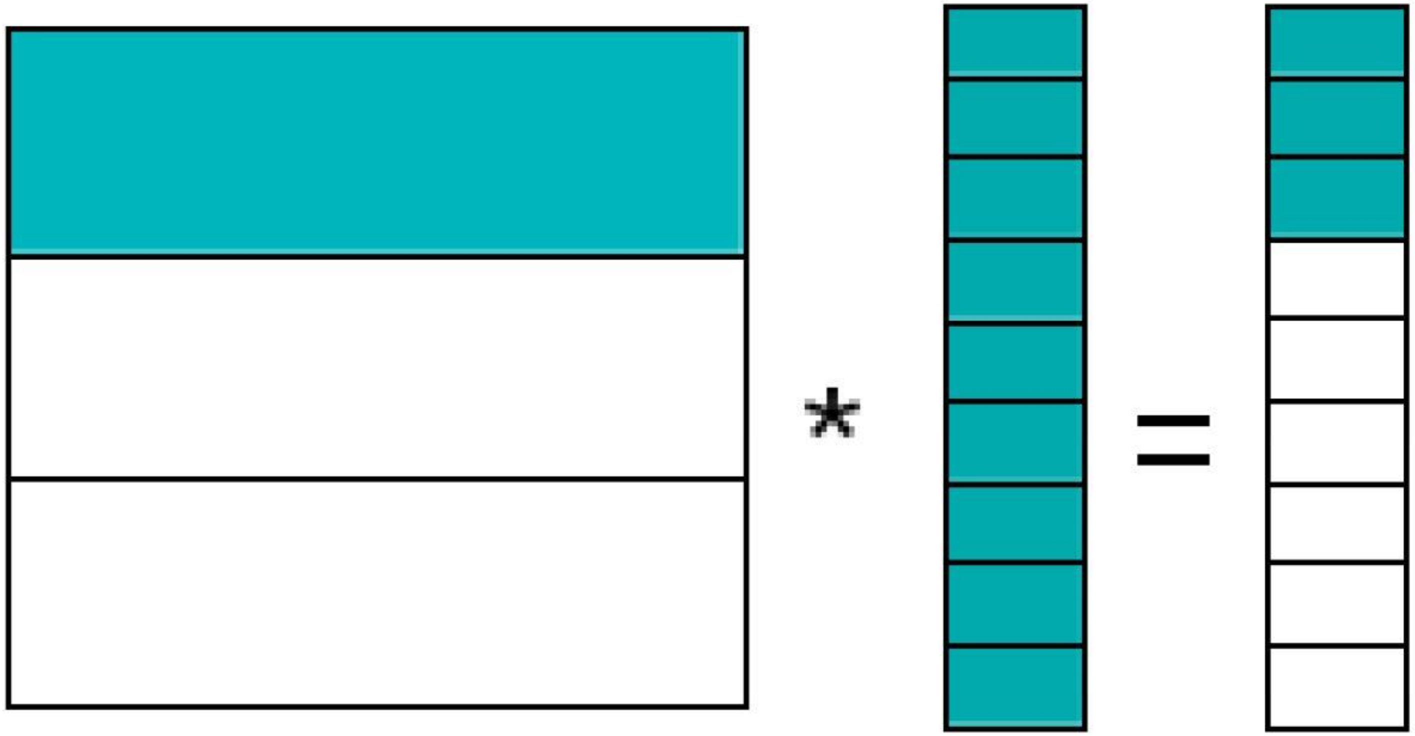
# BLOCKING



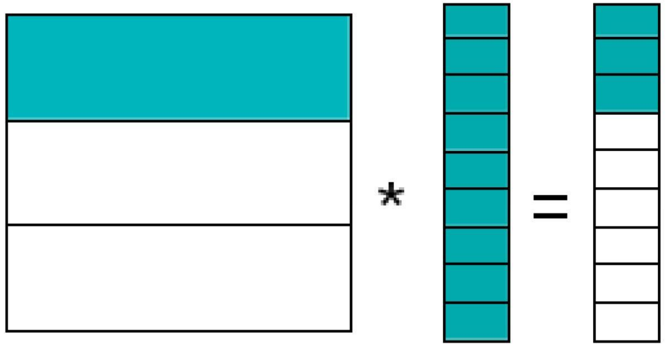

- ✓ Reuse of y entries within a task
- ✓ Successfully tested in other operations (CG)

- × Preprocess required
- × Data dependencies
- × Non-negligible changes in the code

# TASKLOOP



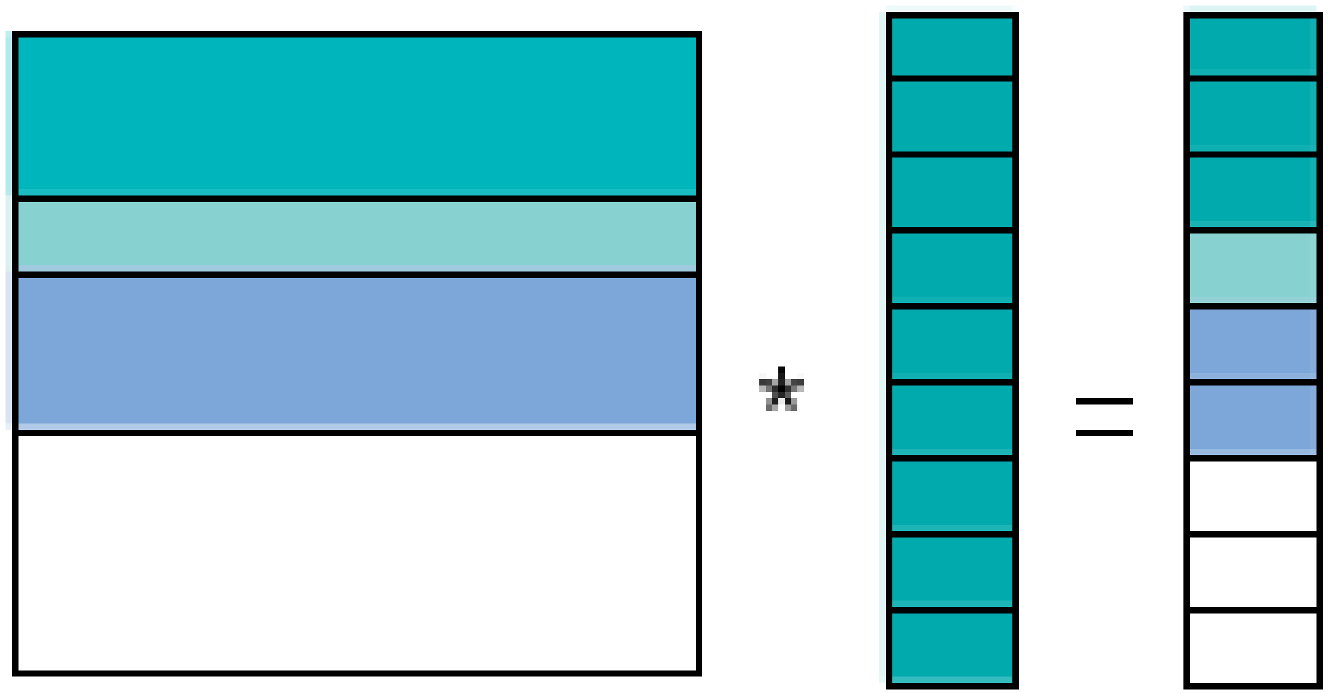
# TASKLOOP



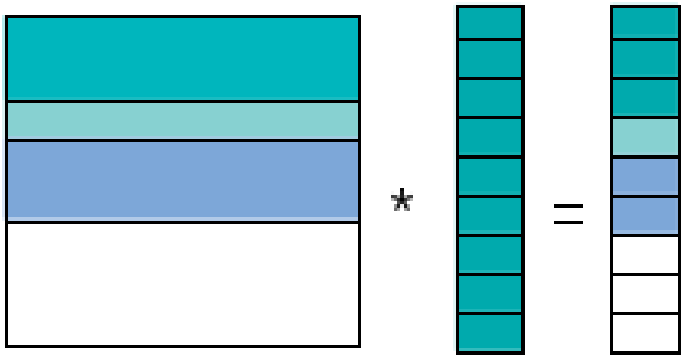

- ✓ Simple
- ✓ Allows coarser tasks

- × Grainsize needs to be computed
- × Potential unbalance

# GROUPING



# GROUPING



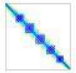
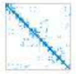

- ✓ Balanced tasks
- ✓ Successfully tested in other operations (Batched GEMM)

- × Extra calculations required
- × One core devoted to create groups

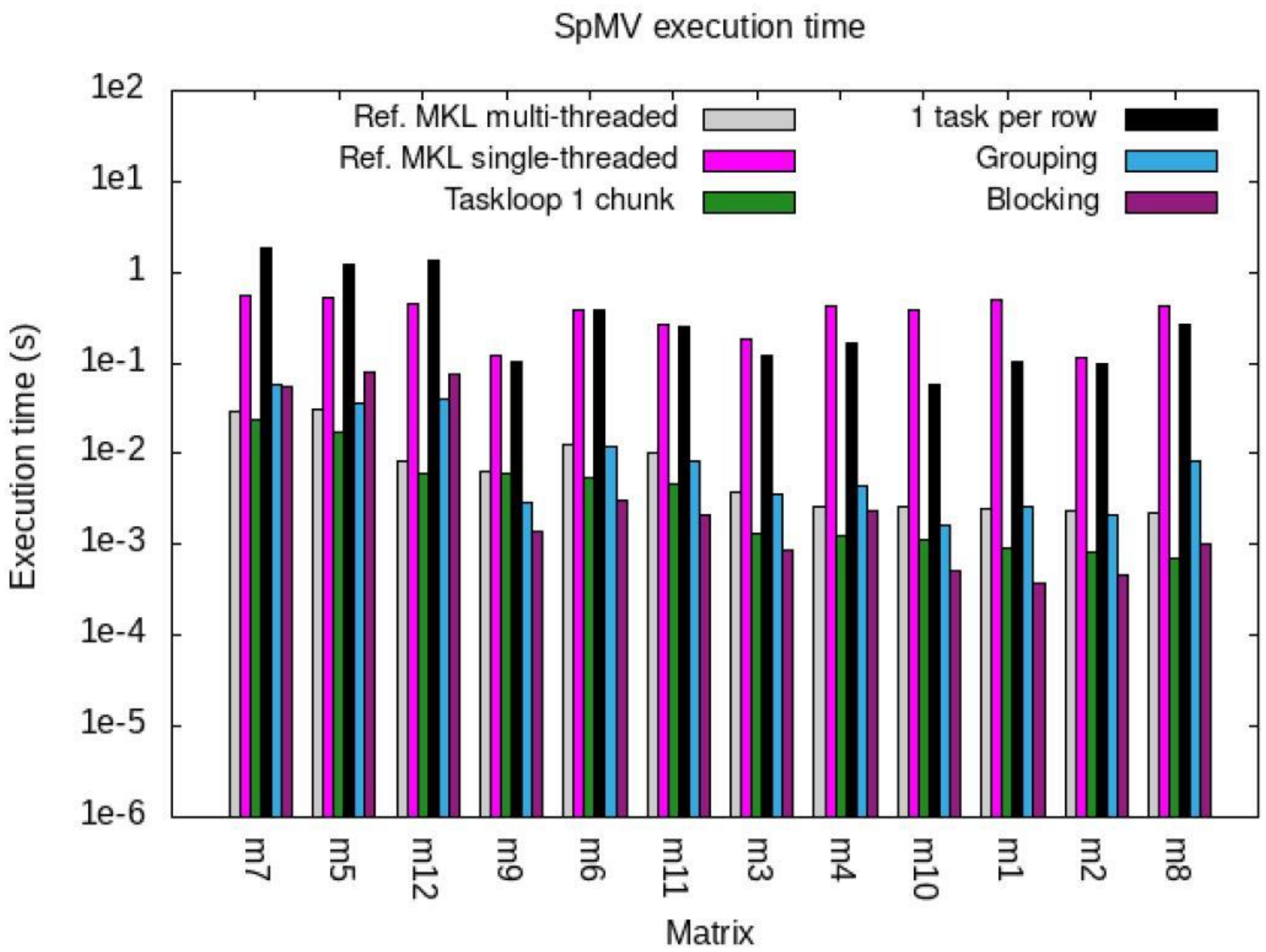
# TEST PLATFORMS

- Marenostrum
  - Single node of Marenostrum 4 (2 sockets Intel Xeon Platinum 8160)
  - 24 cores each at 2.10GHz (48 cores in total)
  - Memory hierarchy: each core has 32KB L1 and 1MB L2 caches, and 33MB L3 (cache shared)
- Dibona
  - Single node two sockets ARM Thunder X2 (ARMv8 NEON) CPU with
  - 32 cores each  at 2.0GHz (64 cores in tota)
  - Memory hierarchy: each core has 32KB of L1 cache, 256KB L2 cache, and 32MB L3 cache

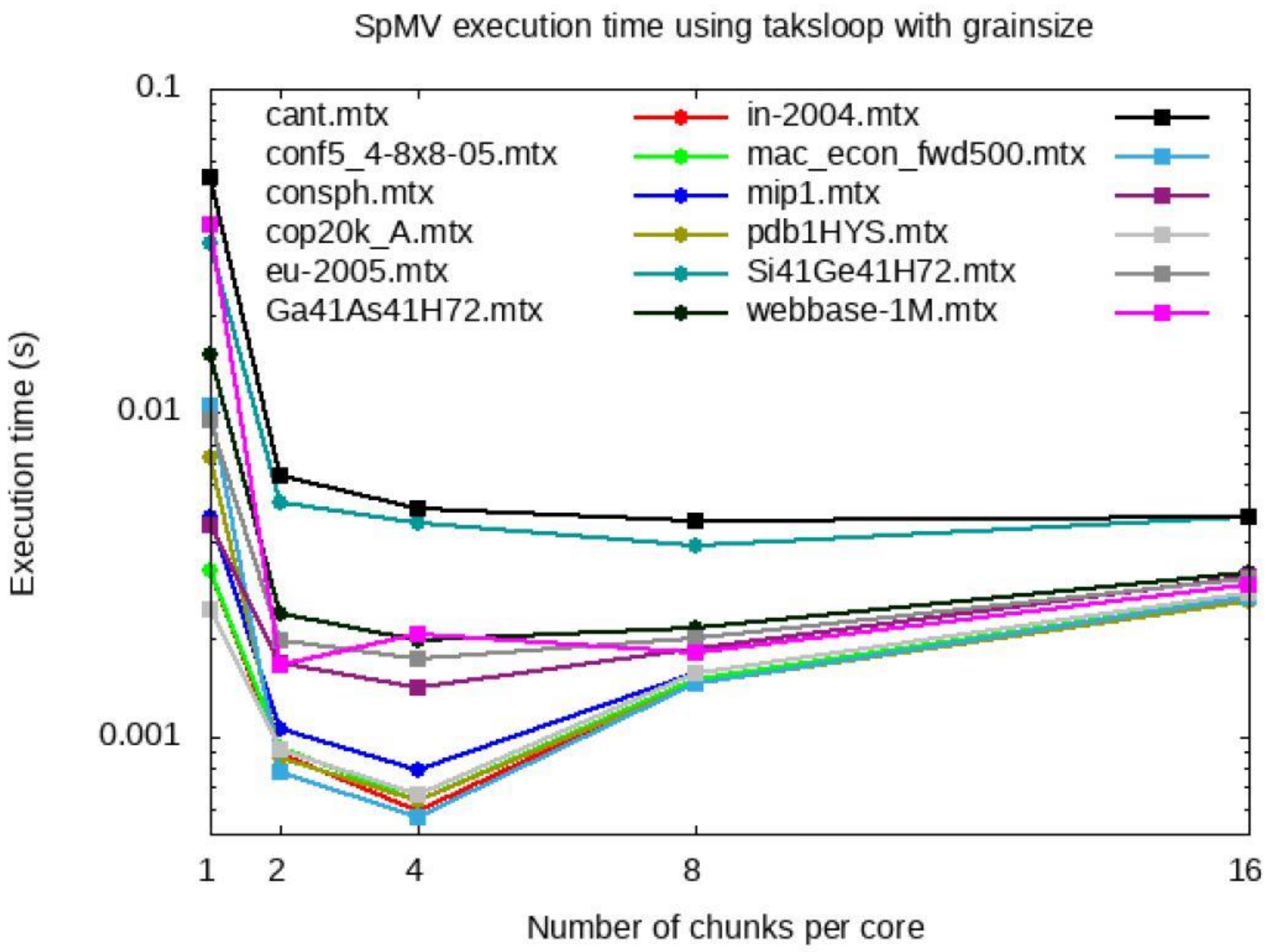
# INPUT MATRICES

| ID | Name | Domain | #rows | NNZ | Max. | Min. | Avg. | Matrix |
|---|---|---|---|---|---|---|---|---|
| m1 | cant | FEM Cantilever | 62,451 | 2,034,917 | 40 | 1 | 32 |  |
| m2 | conf5_4-8x8-05 | Quantum chromodynamics | 49,152 | 1,916,928 | 39 | 39 | 39 |  |
| m3 | consph | FEM concentric spheres | 83,334 | 3,046,907 | 66 | 1 | 36 |  |
| m4 | cop20k_A | Accelerator cavity design | 121,192 | 1,362,087 | 24 | 0 | 11 |  |
| m5 | eu-2005 | Small web crawl of .eu domain | 862,664 | 19,235,140 | 6,985 | 0 | 22 |  |
| m6 | Ga41As41H72 | Real-space pseudo potential method | 268,296 | 9,378,286 | 472 | 1 | 34 |  |
| m7 | in-2004 | Small web crawl of .in domain | 1,382,908 | 16,917,053 | 7,753 | 0 | 12 |  |
| m8 | mac_econ_fwd500 | Macroeconomic model | 206,500 | 1,273,389 | 44 | 1 | 6 |  |
| m9 | mip1 | Optimiation problem | 66,463 | 5,209,641 | 713 | 1 | 78 |  |
| m10 | pdb1HYS | Protein data bank 1HYS | 36,417 | 2,190,591 | 184 | 1 | 60 |  |
| m11 | Si41Ge41H72 | Real-space pseudo potential method | 185,639 | 7,598,452 | 531 | 1 | 40 |  |
| m12 | webbase1-M | Web connectivity matrix | 1,000,005 | 3,105,536 | 4,700 | 1 | 3 |  |

# PERFORMANCE RESULTS

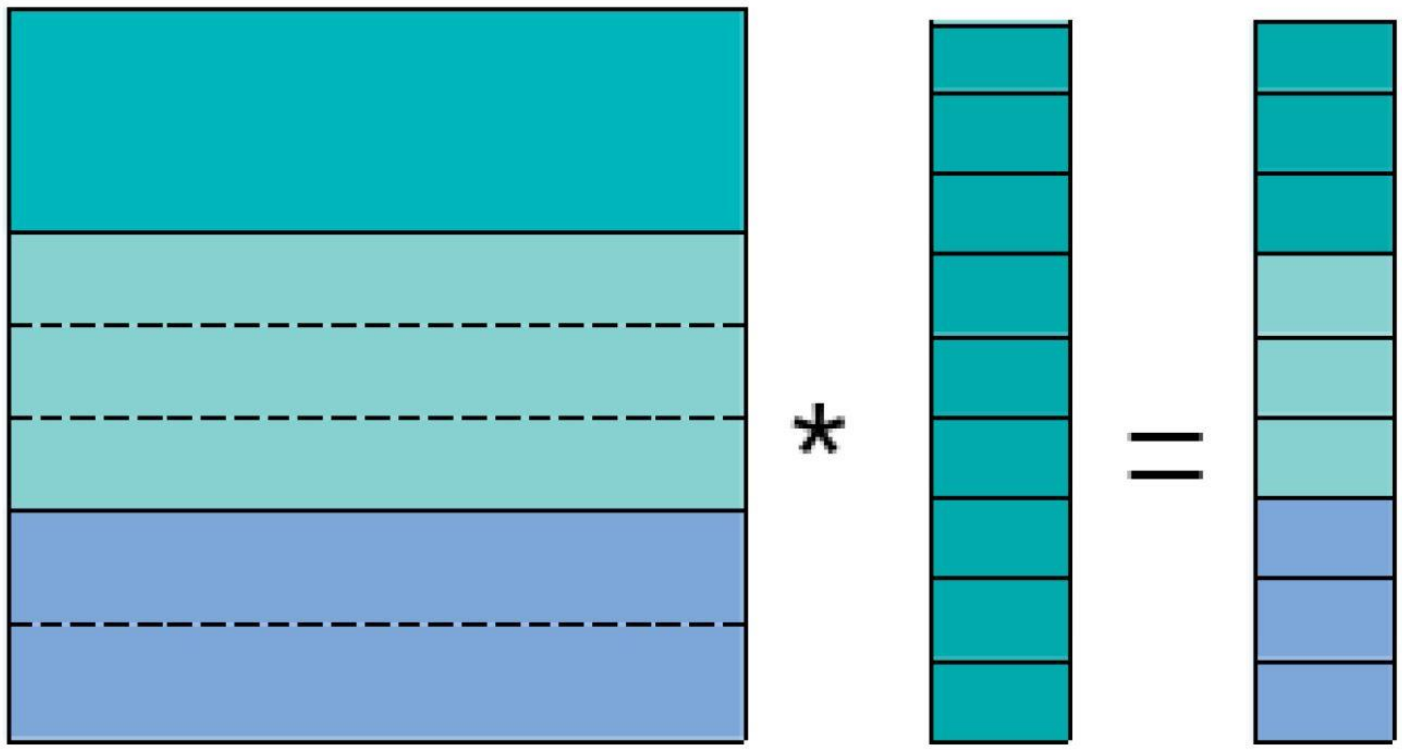


# THE RIGHT GRAINSIZE IN TASKLOOP

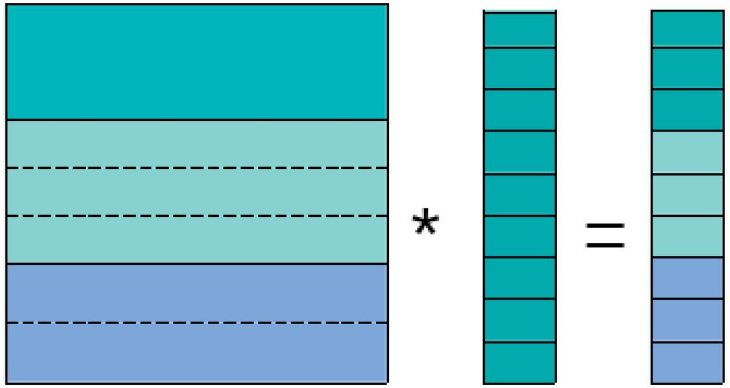


# HOW GRAINSIZE AFFECTS TASK DISTRIBUTION

# ADAPTED NUMBER OF TASKS: TASKLOOP + NESTING



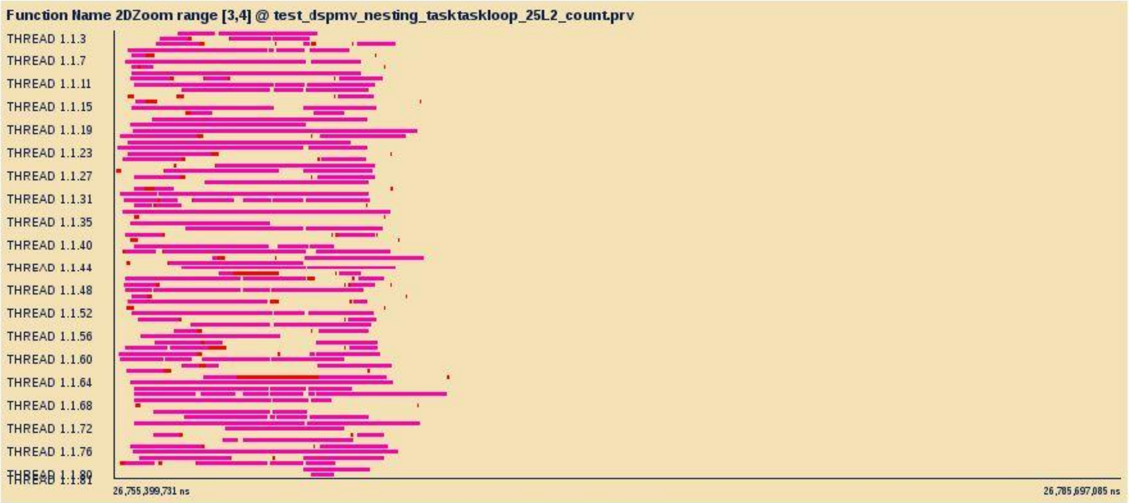
# ADAPTED NUMBER OF TASKS: TASKLOOP + NESTING



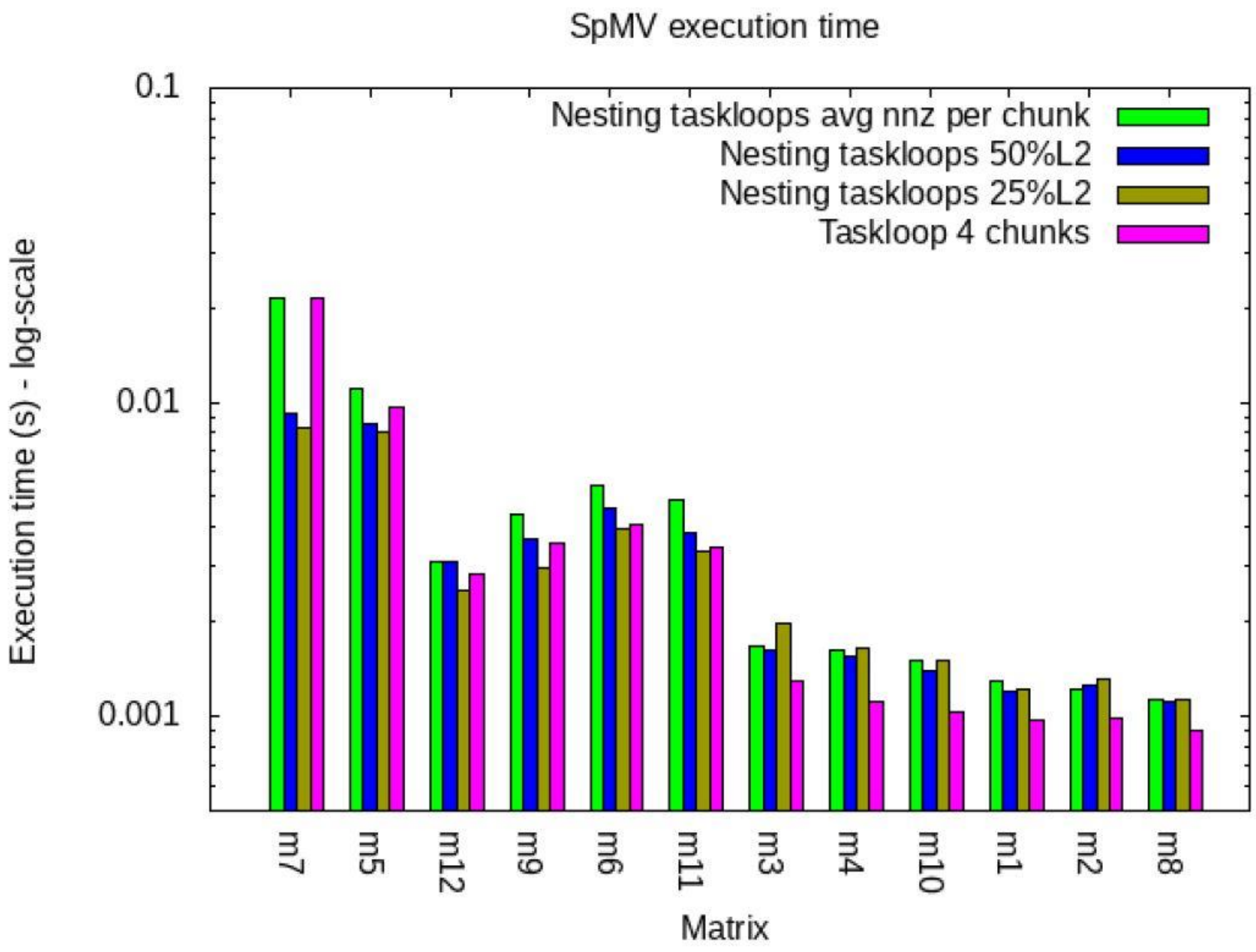
✓ No data dependencias
✓ Balanced nnz per task

× More complex code
× A threshold needs to be choosen

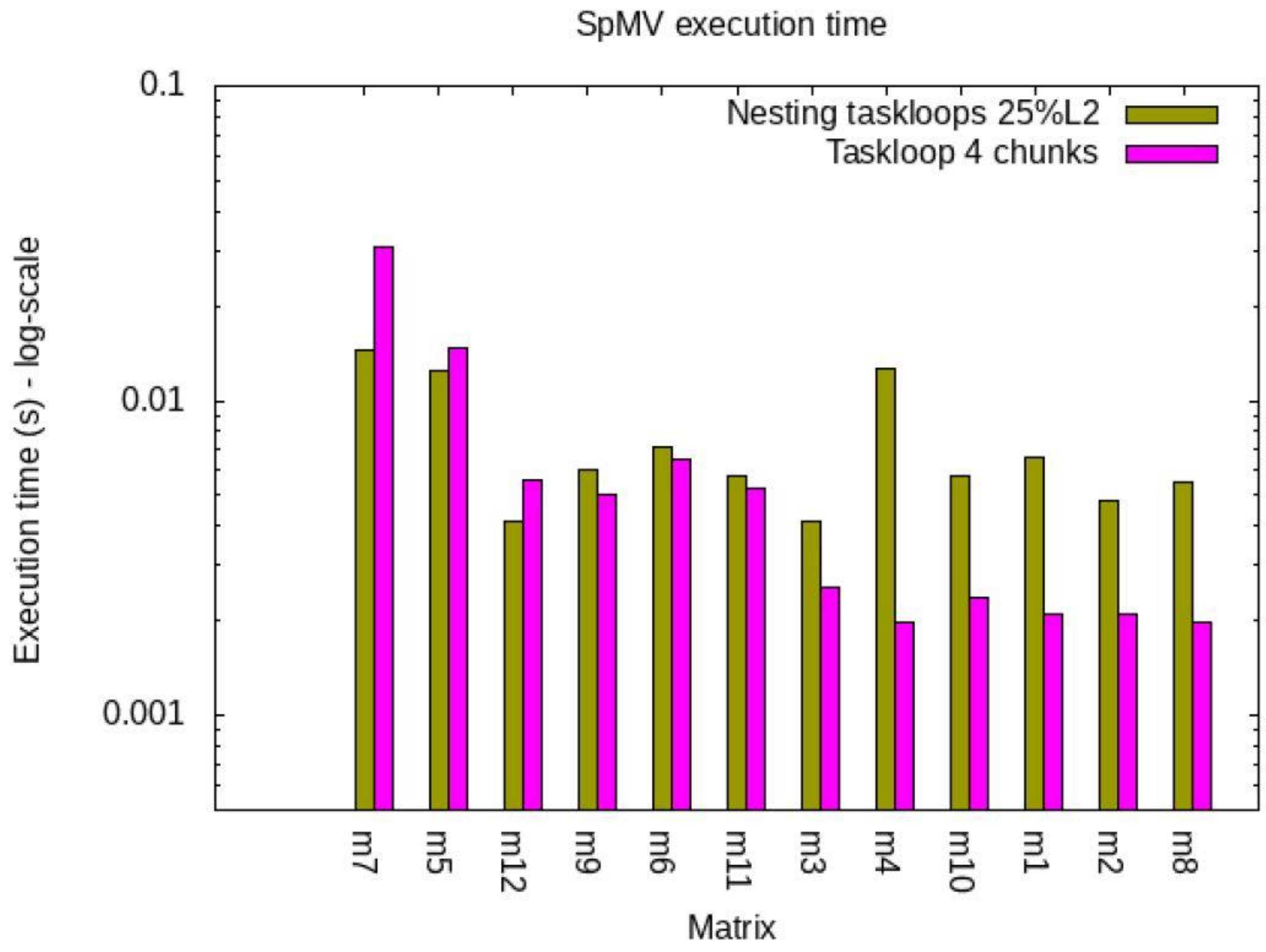
# HOW NESTING AFFECTS TASKS DISTRIBUTION




# RESULTS ON INTEL



# RESULTS ON ARM



# CONCLUSIONS

- Taskloop worked well for balanced sparse matrices
- Taskloop+nesting can do better in unbalanced sparse matrices
- Detecting unbalance is important

# FUTURE WORK

- Combining both strategies
- Include the result in LASs

# THANKS!!

Any question?