COSMA library

Marko Kabic

Motivation

Efforts to achieve communication-optimality:

Motivation

Efforts to achieve communication-optimality:

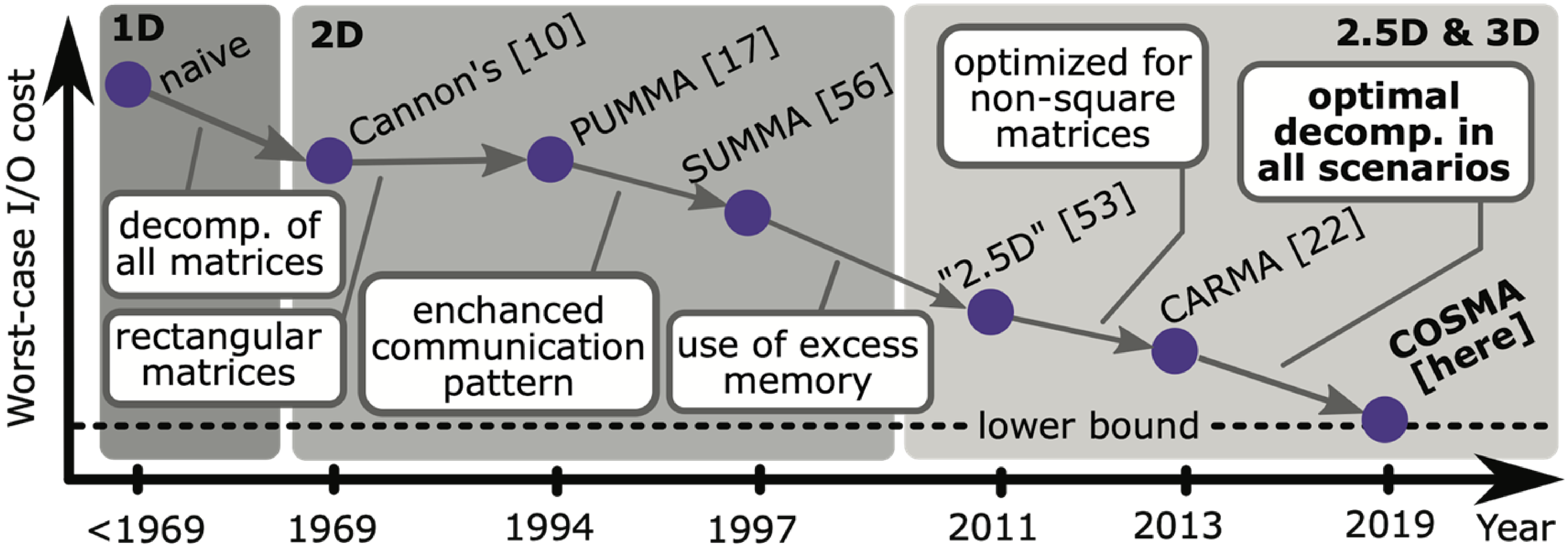
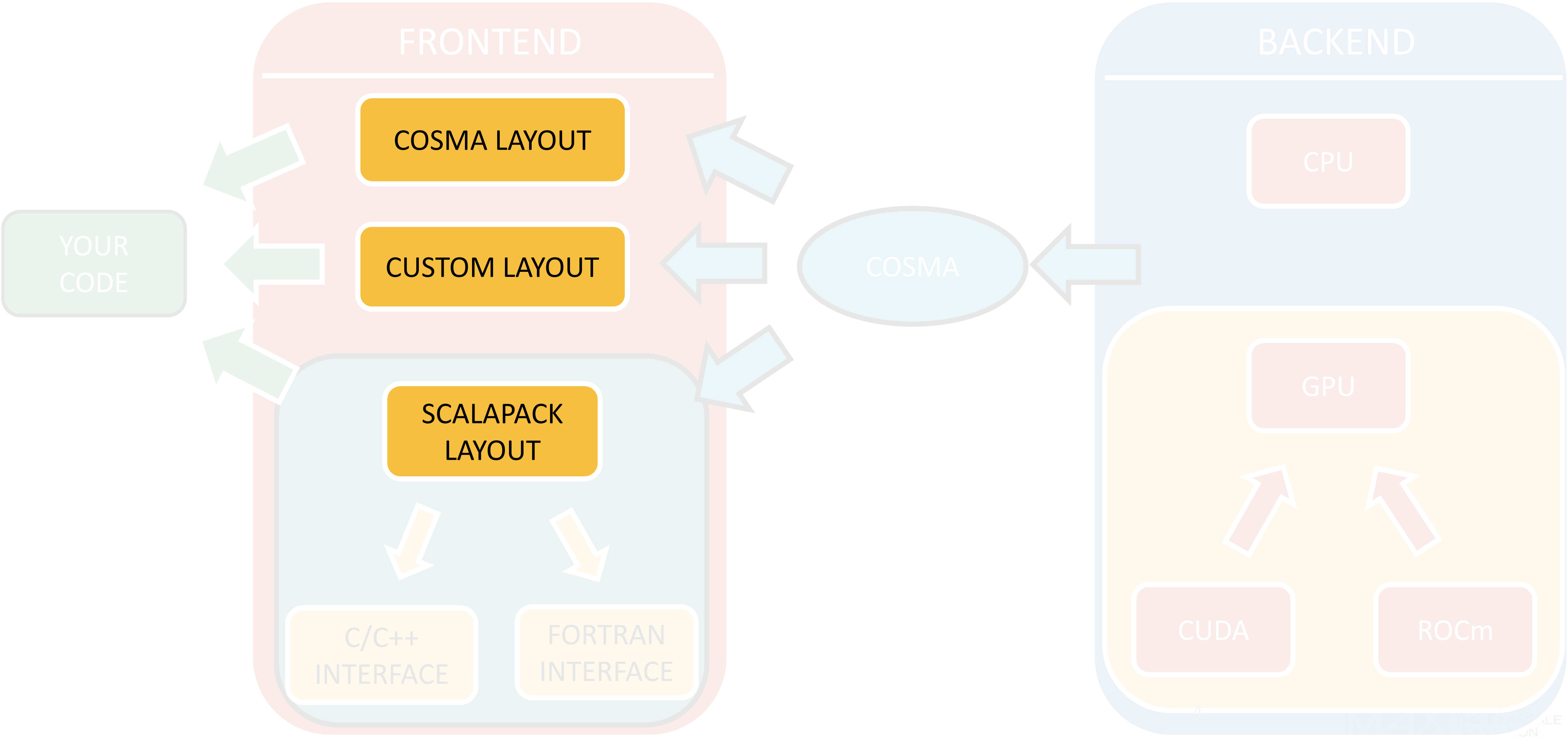
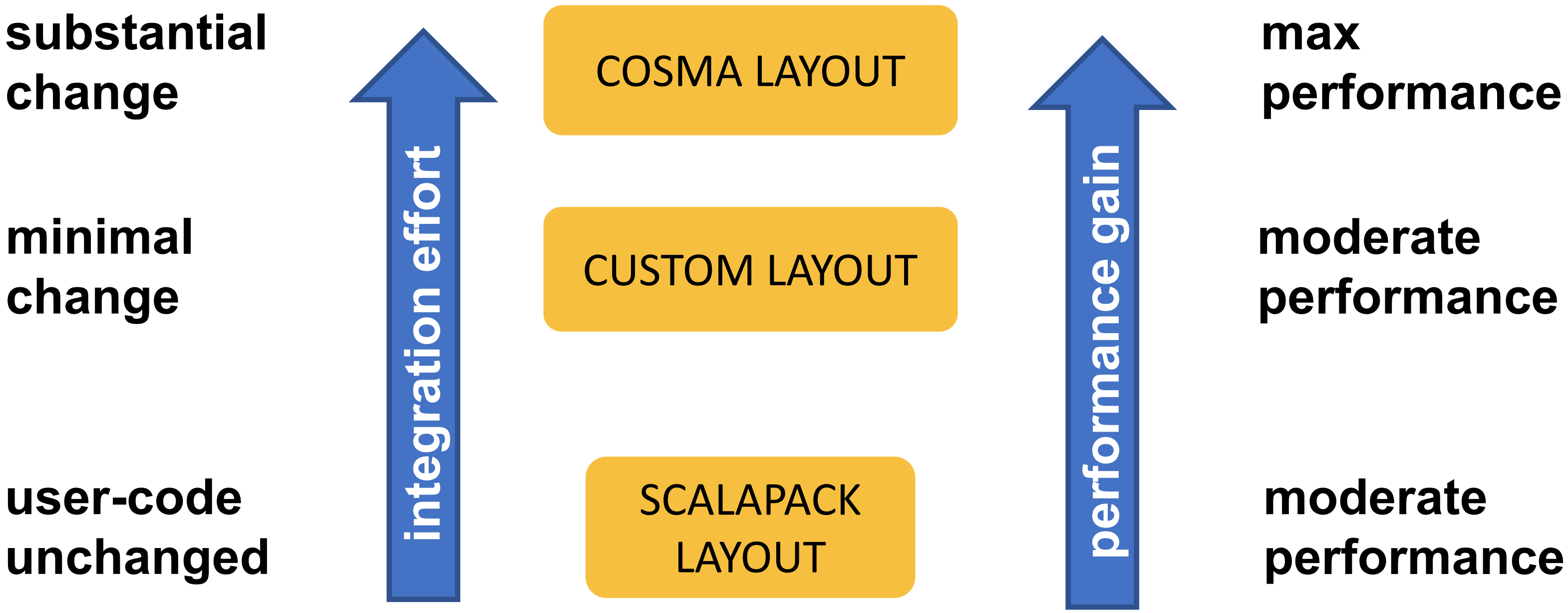


Figure 2: Illustratory evolution of MMM algorithms reaching the I/O lower bound.

Portability and Usability



COSMA API


substantial change
minimal change
user-code unchanged
integration effort
COSMA LAYOUT
CUSTOM LAYOUT
SCALAPACK LAYOUT
performance gain
max performance
moderate performance
moderate performance

SCALAPACK LAYOUT

- Follow the 30 seconds tutorial: https://github.com/eth-cscs/COSMA#using-cosma-in-30-seconds

SCALAPACK LAYOUT

- Follow the 30 seconds tutorial: https://github.com/eth-cscs/COSMA#using-cosma-in-30-seconds

COMPILE COSMA

```
➔ git clone --recursive
  https://github.com/eth-cscs/COSMA cosma
  && cd cosma
➔ mkdir build && cd build
➔ cmake -DCOSMA_BLAS=CUDA
        -DCOSMA_SCALAPACK=MKL
        -DCMAKE_INSTALL_PREFIX=<install-dir>
        ..
➔ make -j 8
➔ make install
```

SCALAPACK LAYOUT

- Follow the 30 seconds tutorial: https://github.com/eth-cscs/COSMA#using-cosma-in-30-seconds

COMPILE COSMA

```
➔ git clone --recursive
  https://github.com/eth-cscs/COSMA cosma
  && cd cosma
➔ mkdir build && cd build
➔ cmake -DCOSMA_BLAS=CUDA
        -DCOSMA_SCALAPACK=MKL
        -DCMAKE_INSTALL_PREFIX=<install-dir>
        ..
➔ make -j 8
➔ make install
```

LINK TO COSMA

```
  # link to COSMA, before any SCALAPACK
➔ LIBS += -L<install-dir>/lib64
          -lcosma_pxgemm
          -lcosma -lgrid2grid
          -lTiled-MM
          -lcublas -lcudart -lrt

  # include headers
➔ INCS += -I<install-dir>/include
```

SCALAPACK LAYOUT

➔ **used in CP2K**

COMPILE COSMA

```
➔ git clone --recursive
  https://github.com/eth-cscs/COSMA cosma
  && cd cosma
➔ mkdir build && cd build
➔ cmake -DCOSMA_BLAS=CUDA
        -DCOSMA_SCALAPACK=MKL
        -DCMAKE_INSTALL_PREFIX=<install-dir>
        ..
➔ make -j 8
➔ make install
```

LINK TO COSMA

```
  # link to COSMA, before any SCALAPACK
➔ LIBS += -L<install-dir>/lib64
          -lcosma_pxgemm
          -lcosma -lgrid2grid
          -lTiled-MM
          -lcublas -lcudart -lrt

  # include headers
➔ INCS += -I<install-dir>/include
```

SCALAPACK LAYOUT

used in CP2K

user code untouched!

COMPILE COSMA

```
➔ git clone --recursive
  https://github.com/eth-cscs/COSMA cosma
  && cd cosma
➔ mkdir build && cd build
➔ cmake -DCOSMA_BLAS=CUDA
        -DCOSMA_SCALAPACK=MKL
        -DCMAKE_INSTALL_PREFIX=<install-dir>
        ..
➔ make -j 8
➔ make install
```

LINK TO COSMA

```
  # link to COSMA, before any SCALAPACK
➔ LIBS += -L<install-dir>/lib64
          -lcosma_pxgemm
          -lcosma -lgrid2grid
          -lTiled-MM
          -lcublas -lcudart -lrt

  # include headers
➔ INCS += -I<install-dir>/include
```

MAX DRIVING THE EXASCALE TRANSITION

SCALAPACK LAYOUT

COSMA
SCALAPACK LAYOUT

VS

SCALAPACK

- Create communicators
- Allocate memory
- **Solve Perfect Matching**
- **Transform Data Layout**
- **Transpose**
- Multiply
- Free communicators
- Free memory

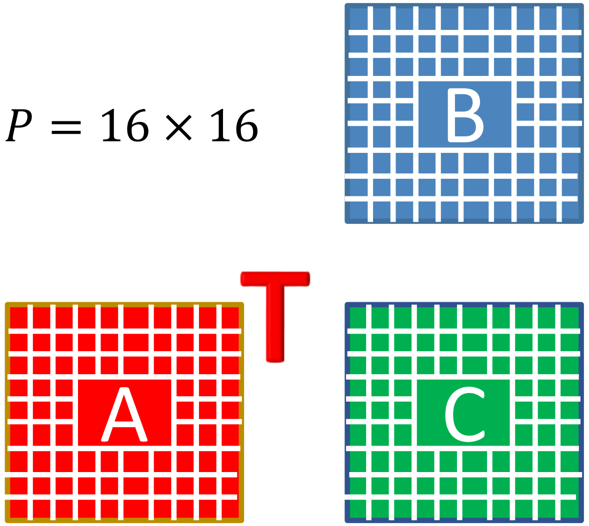
- Multiply

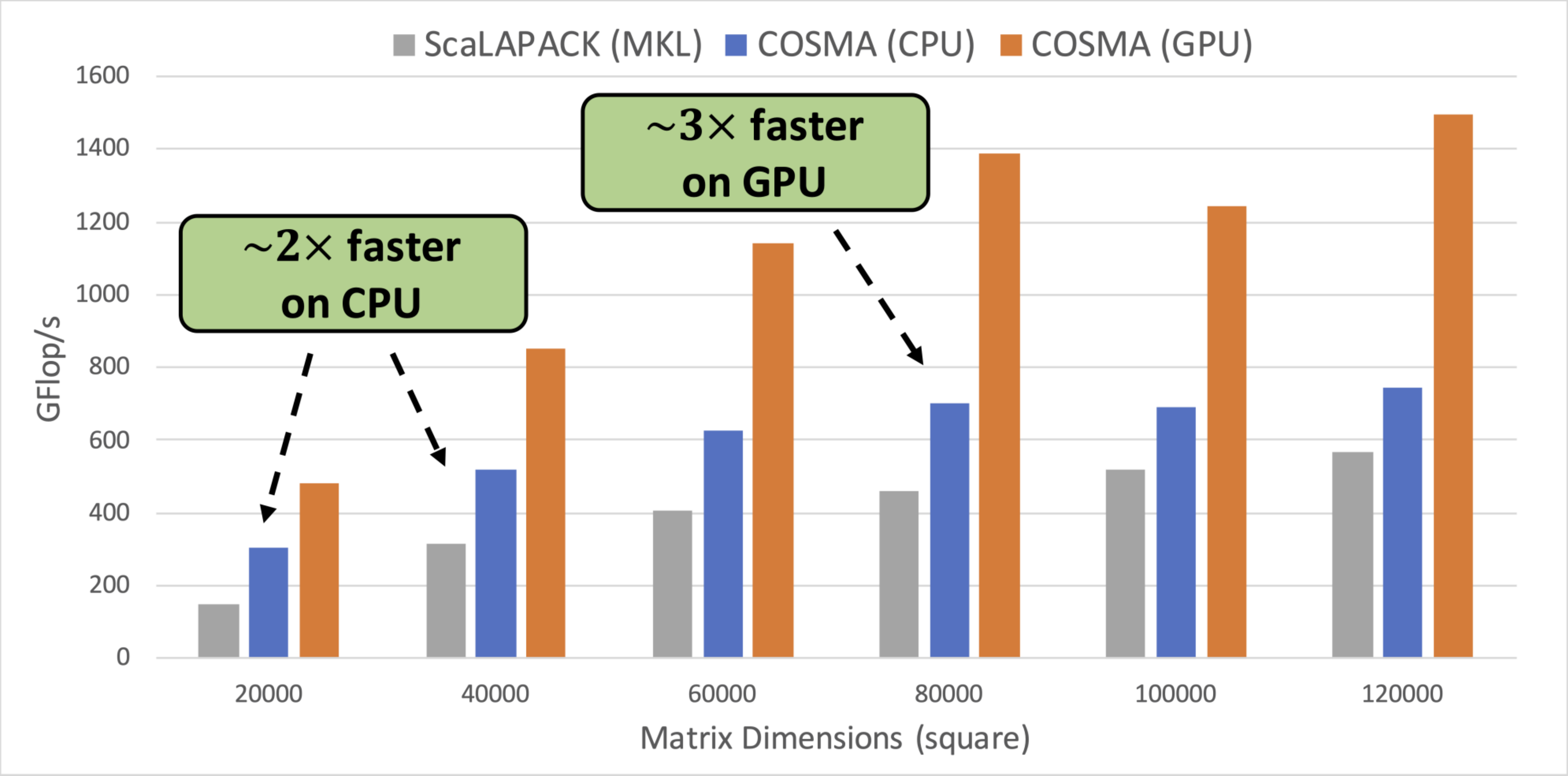
SCALAPACK LAYOUT

256 nodes, including all the overheads

↑higher = better

$P = 16 \times 16$





SCALAPACK LAYOUT

CP2K: RPA 128 water molecules

SCALAPACK LAYOUT

CP2K: RPA 128 water molecules

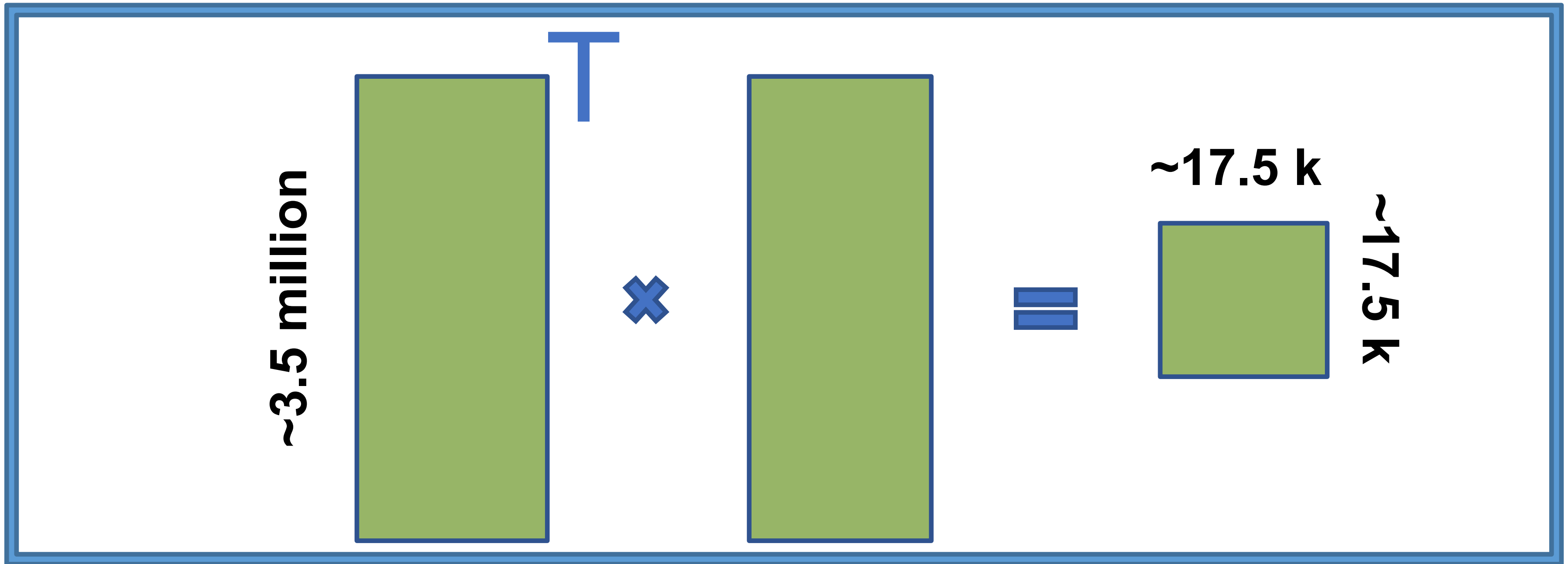
PDGEMM INFO				
DIMENSIONS			PROCESSOR GRID	
M	N	K	ROWS	COLS
17408	17408	3473408	128	1
BLOCK DIMENSIONS			TRANSPOSE FLAGS	
BLOCK M	BLOCK N	BLOCK K	MATRIX A	MATRIX B
8704	8704	13568	transposed (T)	non-transposed (N)

SCALAPACK LAYOUT

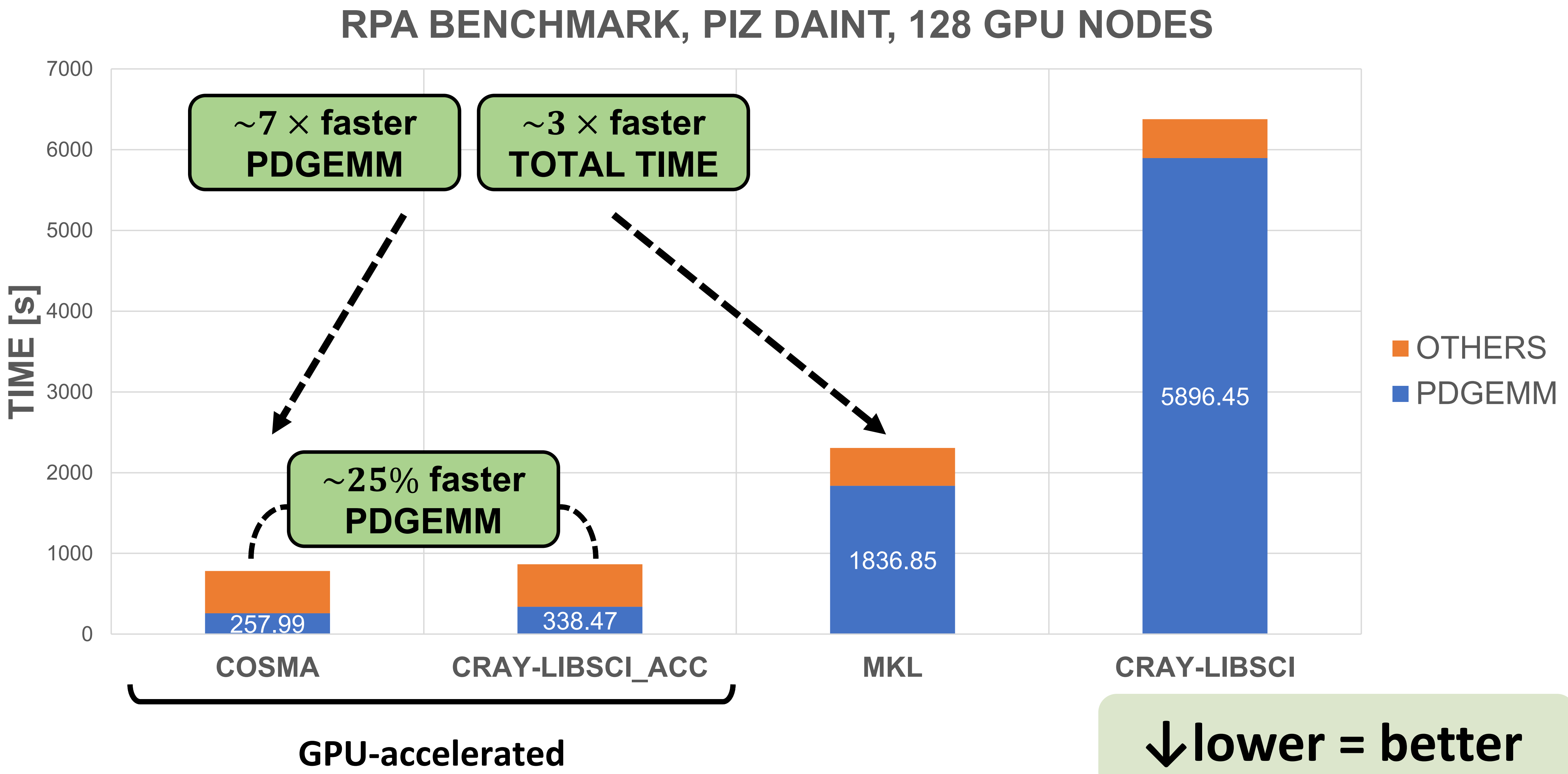
CP2K: RPA 128 water molecules

PDGEMM INFO				
DIMENSIONS			PROCESSOR GRID	
M	N	K	ROWS	COLS
17408	17408	3473408	128	1
BLOCK DIMENSIONS			TRANSPOSE FLAGS	
BLOCK M	BLOCK N	BLOCK K	MATRIX A	MATRIX B
8704	8704	13568	transposed (T)	non-transposed (N)

46 x

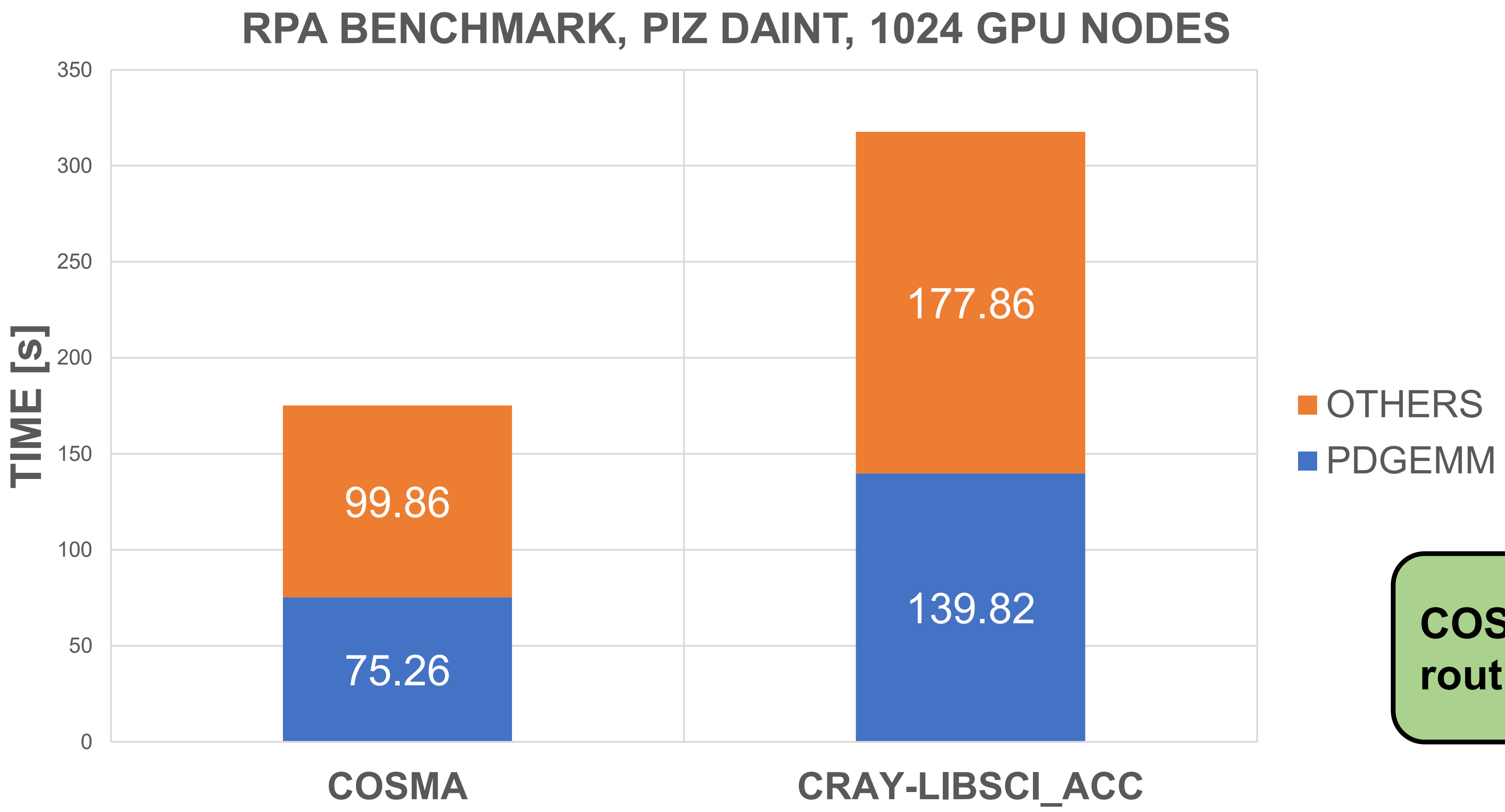


SCALAPACK LAYOUT



↓lower = better

SCALAPACK LAYOUT



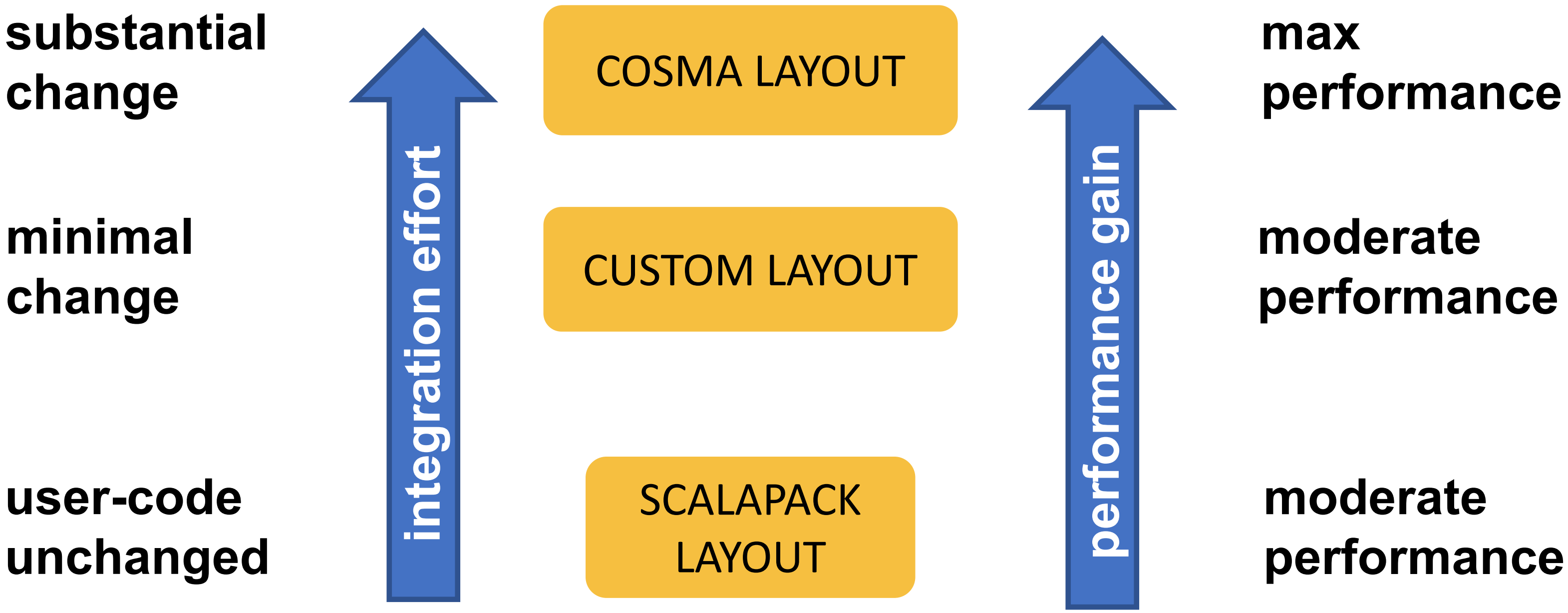
also improved

COSMA also helps in other routines like Cholesky!

~2 × faster PDGEMM &TOTAL

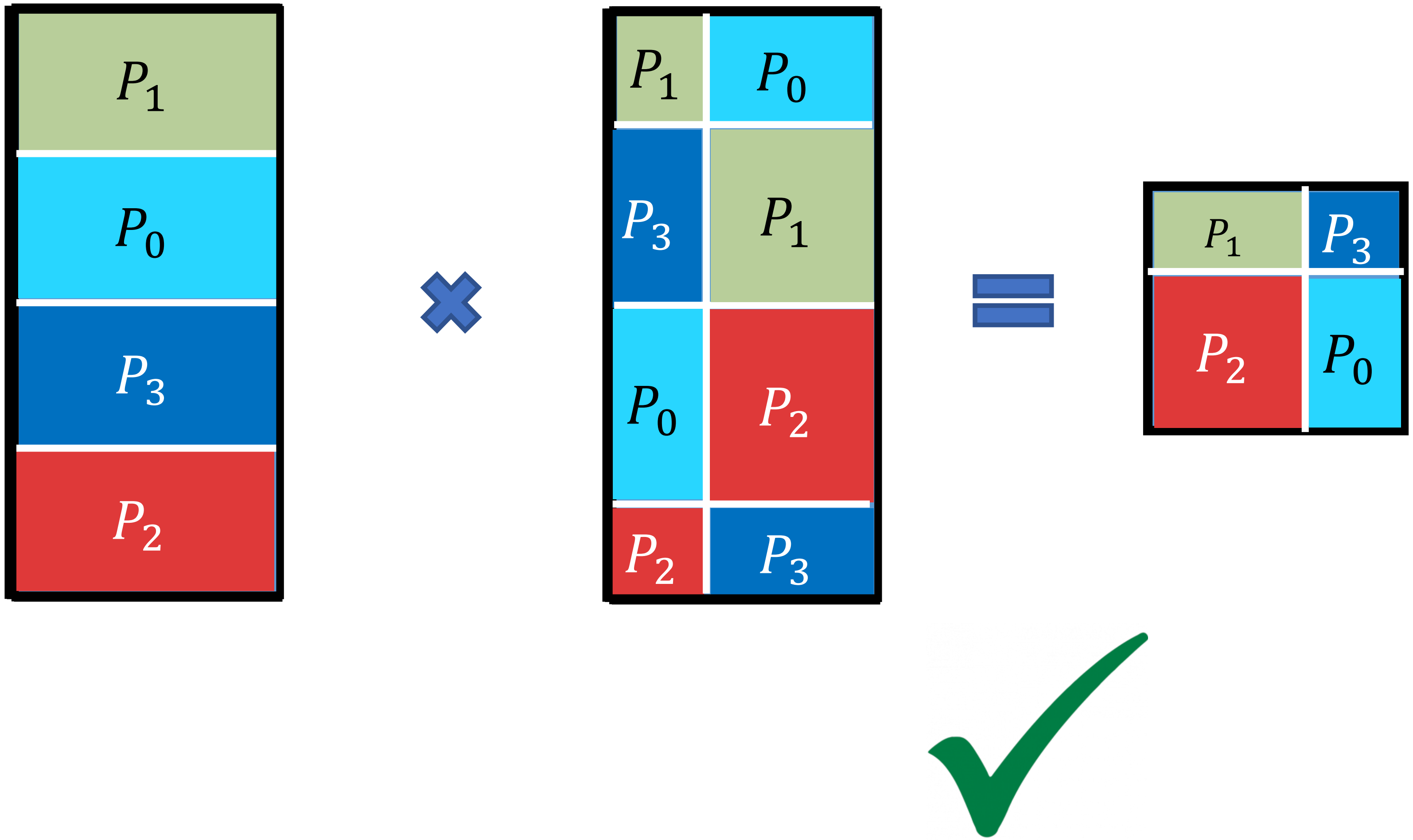
↓lower = better

COSMA API

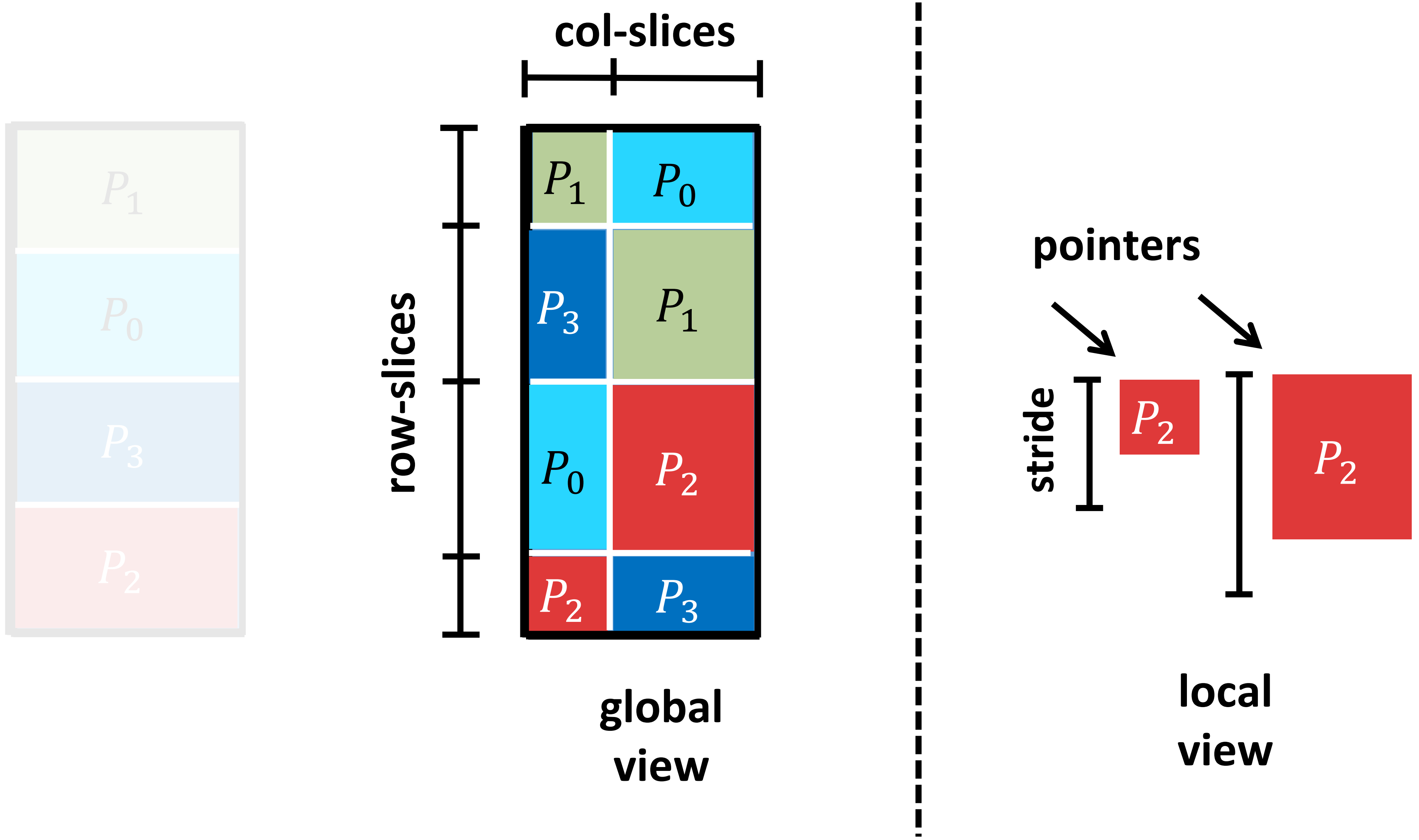

substantial change
minimal change
user-code unchanged
integration effort
COSMA LAYOUT
CUSTOM LAYOUT
SCALAPACK LAYOUT
performance gain
max performance
moderate performance
moderate performance

CUSTOM LAYOUT

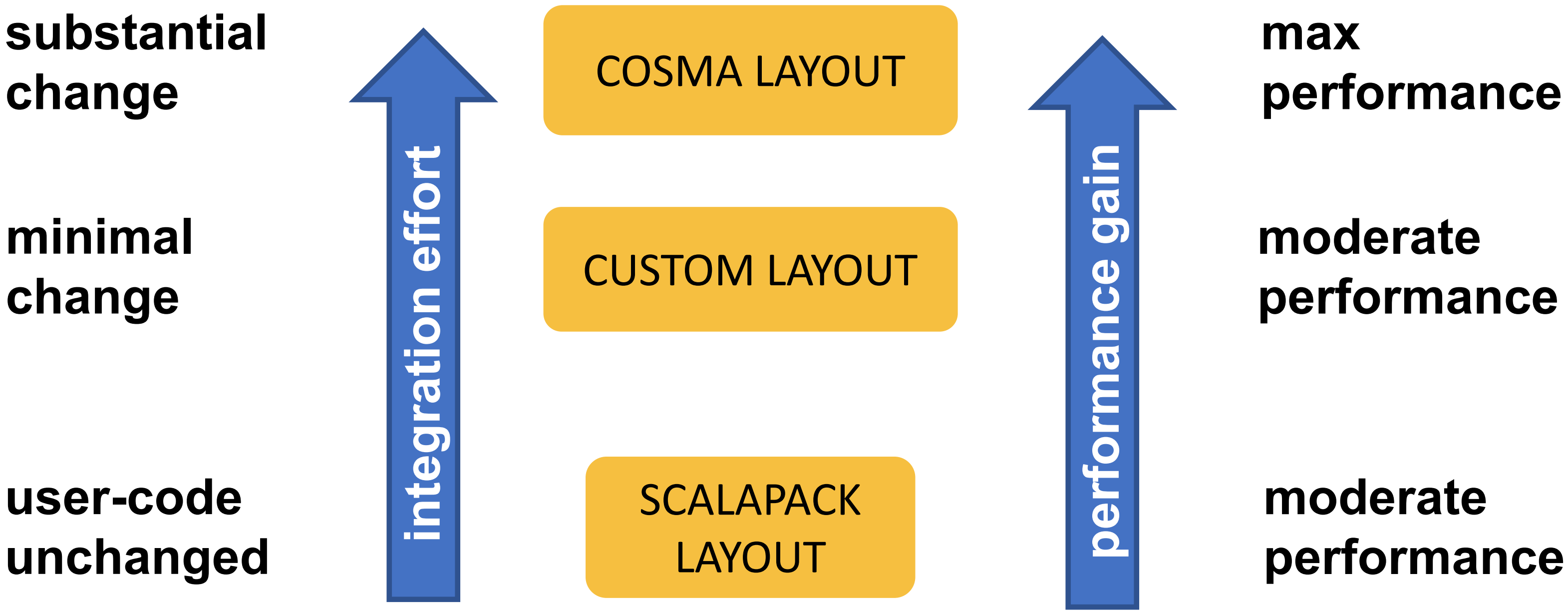
COSMA can deal with matrices that are very irregularly distributed as well



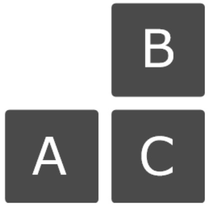
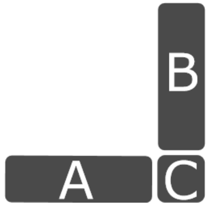
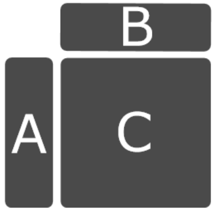
CUSTOM LAYOUT



COSMA API


substantial change
minimal change
user-code unchanged
integration effort
COSMA LAYOUT
CUSTOM LAYOUT
SCALAPACK LAYOUT
performance gain
max performance
moderate performance
moderate performance

COSMA LAYOUT

shape	benchmark	total comm. volume per rank [MB]				speedup		
		ScaLAPACK	CTF	CARMA	COSMA	min	mean	max
	strong scaling	203	222	195	107	1.07	1.94	4.81
	limited memory	816	986	799	424	1.2[illegible]	[illegible]	[illegible]9
	extra memory	303	350	291	151	1.14	[illegible]	[illegible]3
	strong scaling	2636	2278	659	545	1.24	[illegible]	[illegible]5
	limited memory	368	541	128	88	1.3	[illegible]	[illegible]
	extra memory	133	152	48	[illegible]	[illegible]	2.55	6.7
	strong scaling	3507	2024	541	[illegible]	[illegible]	2.22	3.22
	limited memory	989	672	399	[illegible]	[illegible]	1.7	2.27
	extra memory	122	77	77	[illegible]	[illegible]	[illegible]76	2.8
	strong scaling	134	68	10	7	1.21	4.02	12.81
	limited memory	47	101	26	8	1.31	2.07	3.41
	extra memory	15	15	10	3	1.5	2.29	3.59
	overall					1.07	2.17	12.81

CTF:
p=128,
m=n=k=12123

ScaLAPACK:
p=17029,
m=n=113072,
k=512

CODE & MATERIALS

- Best Student Paper Award at **SC19** supercomputing conference in Denver, US
- In collaboration with Prof. Torsten Hoefler (ETH Zurich)

Denver, CO | hpc is now.

- **Code:** https://github.com/eth-cscs/COSMA
- **ACM Digital Library:** https://dl.acm.org/doi/10.1145/3295500.3356181
- **Arxiv:** https://arxiv.org/abs/1908.09606
- **YouTube Presentation:** https://www.youtube.com/watch?v=5wiZWw5ltR0
- **Press Release:** https://www.cscs.ch/science/computer-science-hpc/2019/new-matrix-multiplication-algorithm-pushes-the-performance-to-the-limits/

Questions?
marko.kabic@cscs.ch

BONUS

SCALAPACK-wrapper

128 nodes: Piz Daint Supercomputer (Cray XC50)

	CPU-ONLY			GPU ACCELERATED	
ALGORITHM	CRAY-LIBSCI	MKL	COSMA-CPU	CRAY-LIBSCI_ACC	COSMA-GPU
CONFIGURATION	1MPI x 12T	1MPI x 12T	1MPI x 12T	1MPI x 12T	1MPI x 12T
CP2K RPA-RI 128-H20 [s]	6379.14	2305.41	2238.94	865.73	781.60
46 x PDGEMM [s]	5896.45	1836.85	1723.62	338.47	257.99
NODE GFLOP/s	128.30	411.87	438.92	2235.19	2932.44
% PEAK PERF.	25.70%	82.51%	87.92%	49.67%	65.17%
NODE TYPE (128 nodes)	Intel® Xeon® E5-2690 v3 @ 2.60GHz (12 cores, 64GB RAM)			NVIDIA® Tesla® P100 16GB	
NODE PEAK PERF[GFLOP/s]	499.2			4500	
	This is only using CPU nodes on the GPU partition of Piz Daint. However, CPU node peak perf is much higher on the CPU partition.			Max peak assumes the data is already on GPU, which explains why it is not fully achieved.	

~10% faster

~25% faster

SCALAPACK-wrapper

1024 nodes: Piz Daint Supercomputer (Cray XC50)

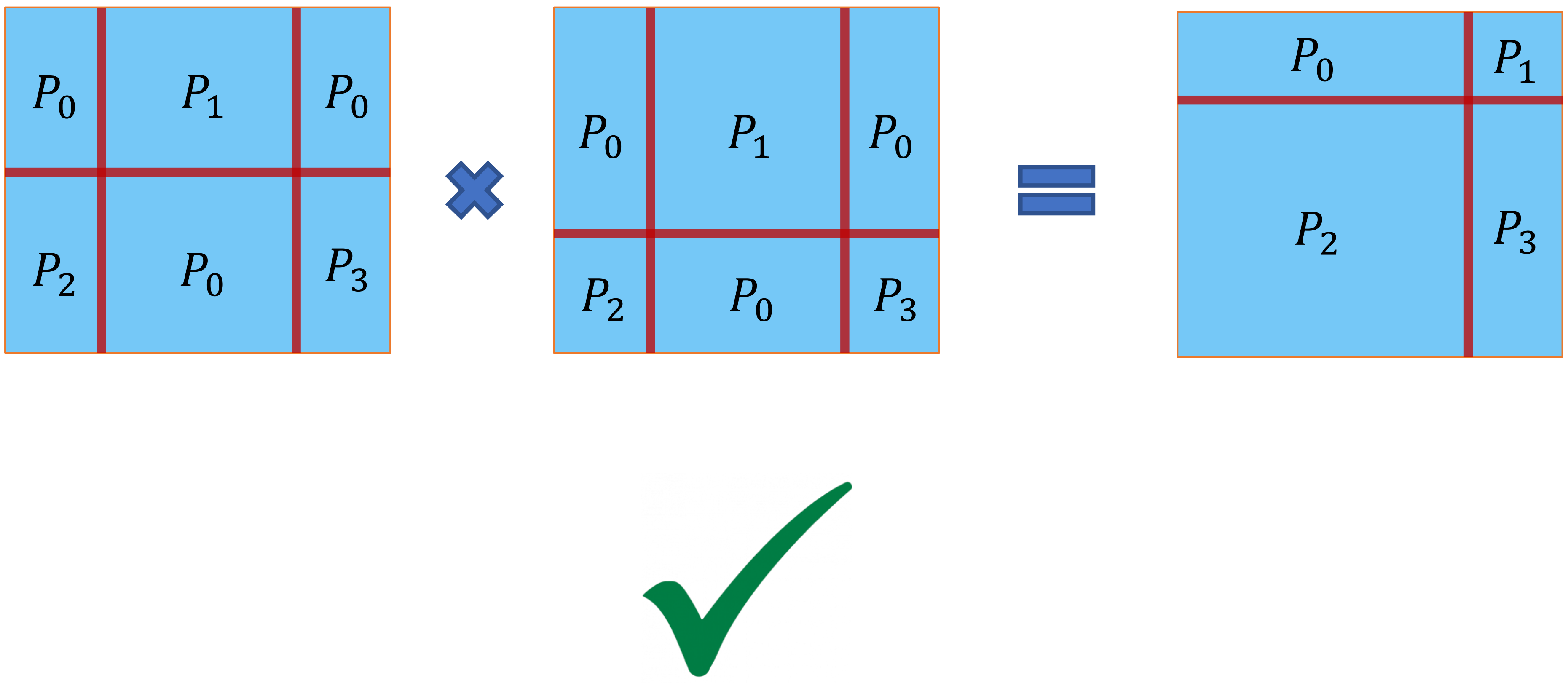
	GPU ACCELERATED	
ALGORITHM	CRAY-LIBSCI_ACC	COSMA-GPU
CONFIGURATION	1MPI x 12T	1MPI x 12T
CP2K RPA-RI 128-H20 [s]	317.68	175.12
46 x PDGEMM [s]	139.82	75.26
NODE GFLOP/s	676.37	1256.49
% PEAK PERF.	15.03%	27.92%
NODE TYPE (1024 nodes)	NVIDIA® Tesla® P100 16GB	
NODE PEAK PERF[GFLOP/s]	4500	

$\sim 2\times$ faster

MAX DRIVING THE EXASCALE TRANSITION

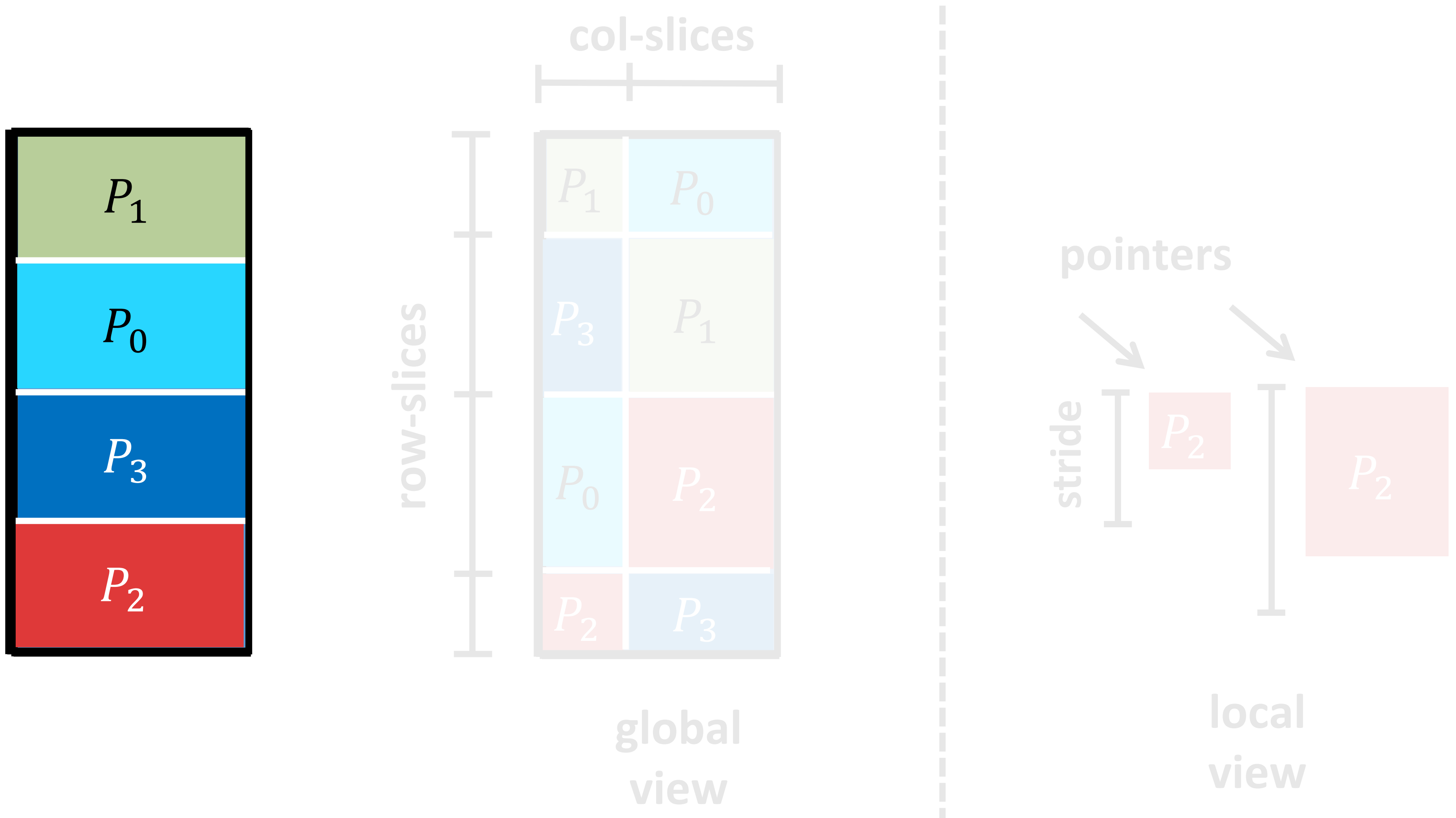
COSMA API

COSMA can deal with matrices that are very irregularly distributed as well

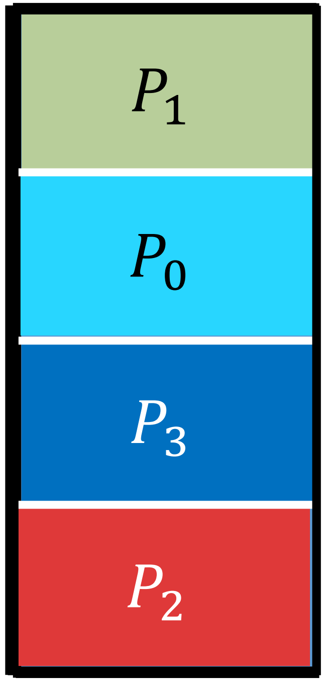


CUSTOM LAYOUT

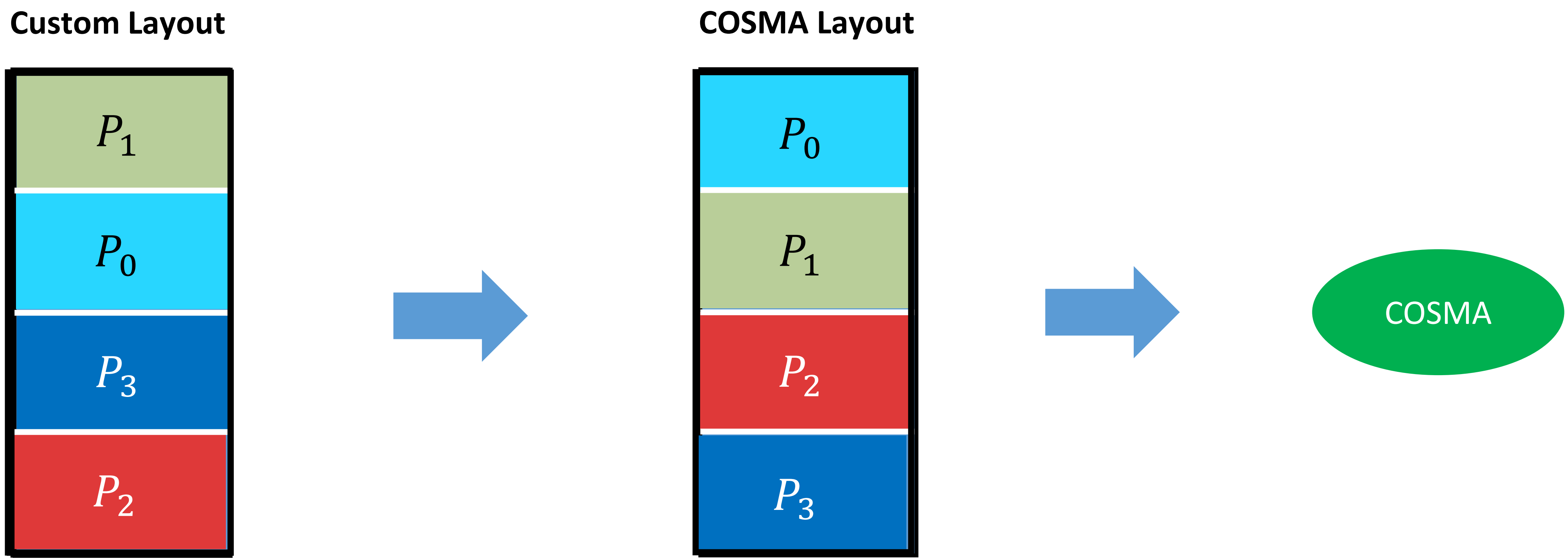
CUSTOM LAYOUT



CUSTOM LAYOUT

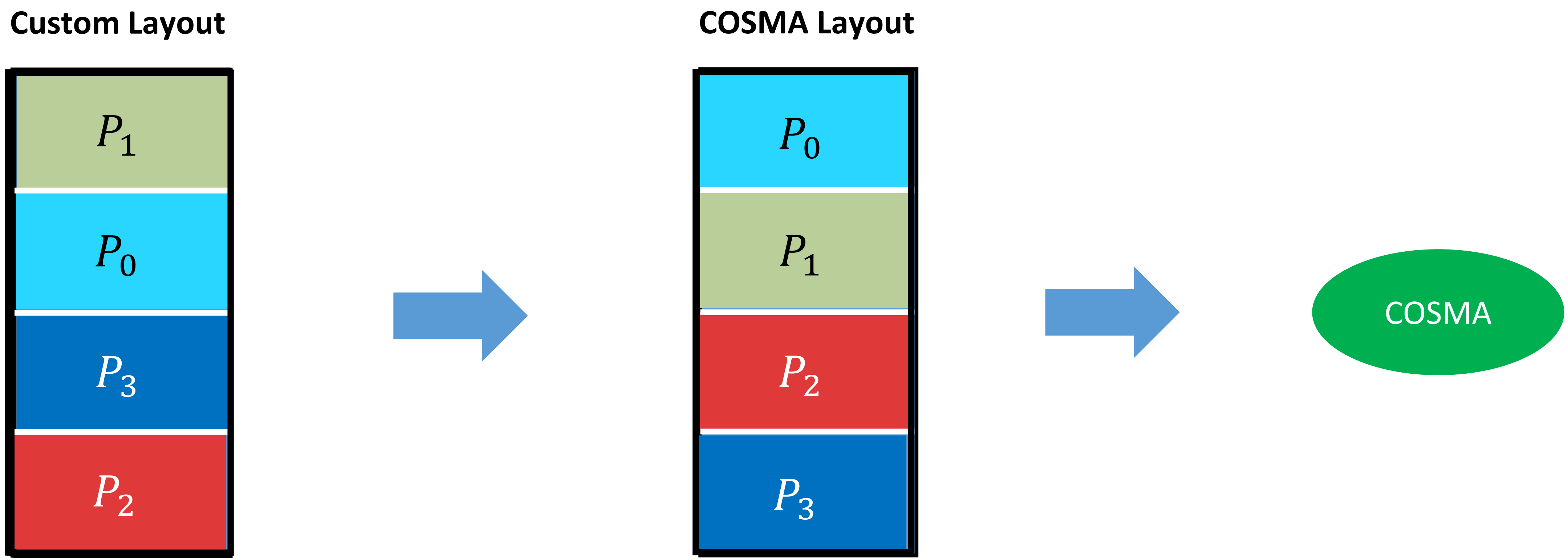


CUSTOM LAYOUT



IDEA: relabel the ranks to minimize the communication cost

CUSTOM LAYOUT

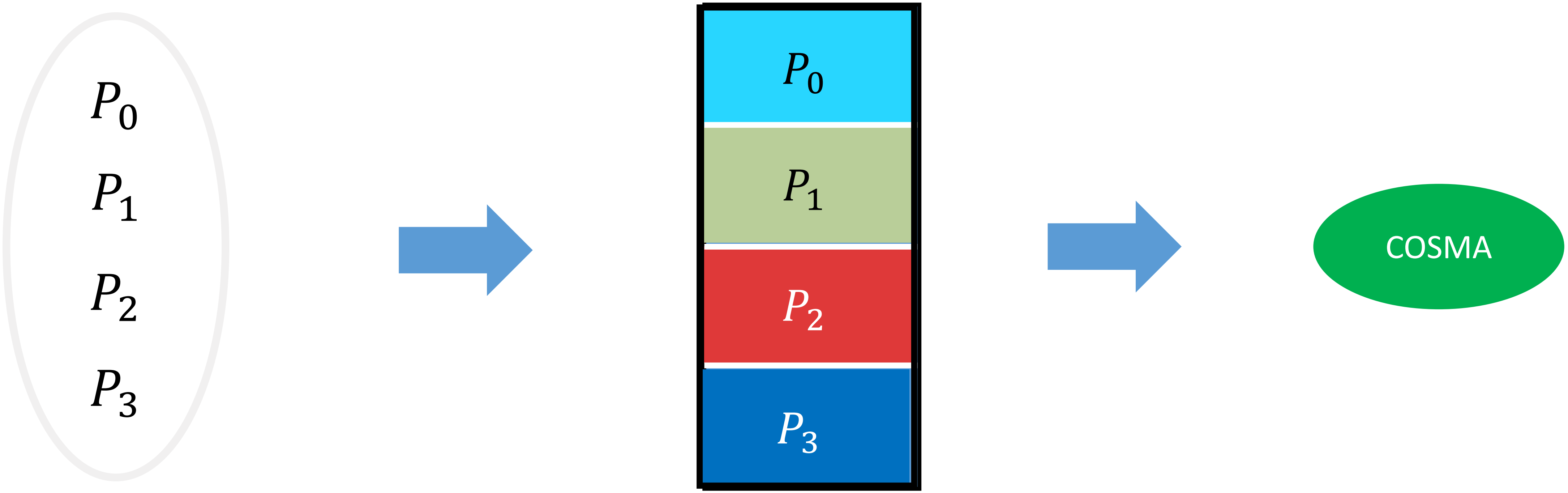


IDEA: relabel the ranks to minimize the communication cost

CUSTOM LAYOUT

Custom Layout

COSMA Layout



IDEA: relabel the ranks to minimize the communication cost

CUSTOM LAYOUT

Custom Layout

P_0

P_1

P_2

P_3

COSMA Layout

P_0

P_1

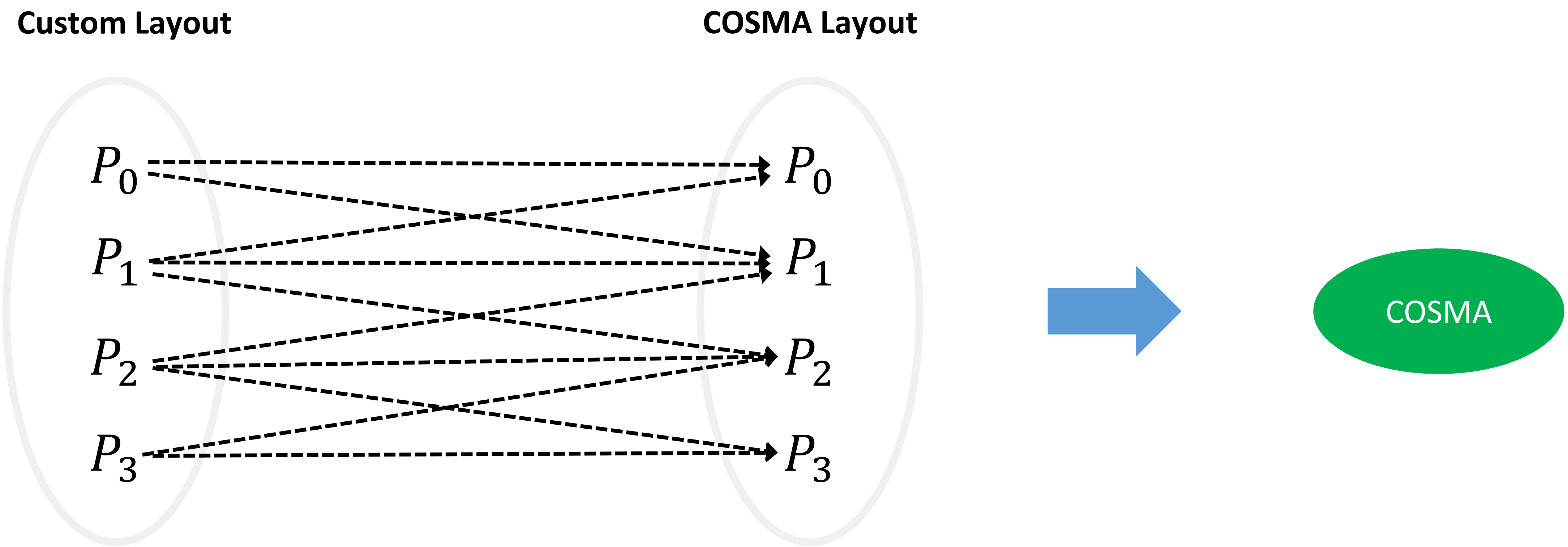
P_2

P_3

COSMA

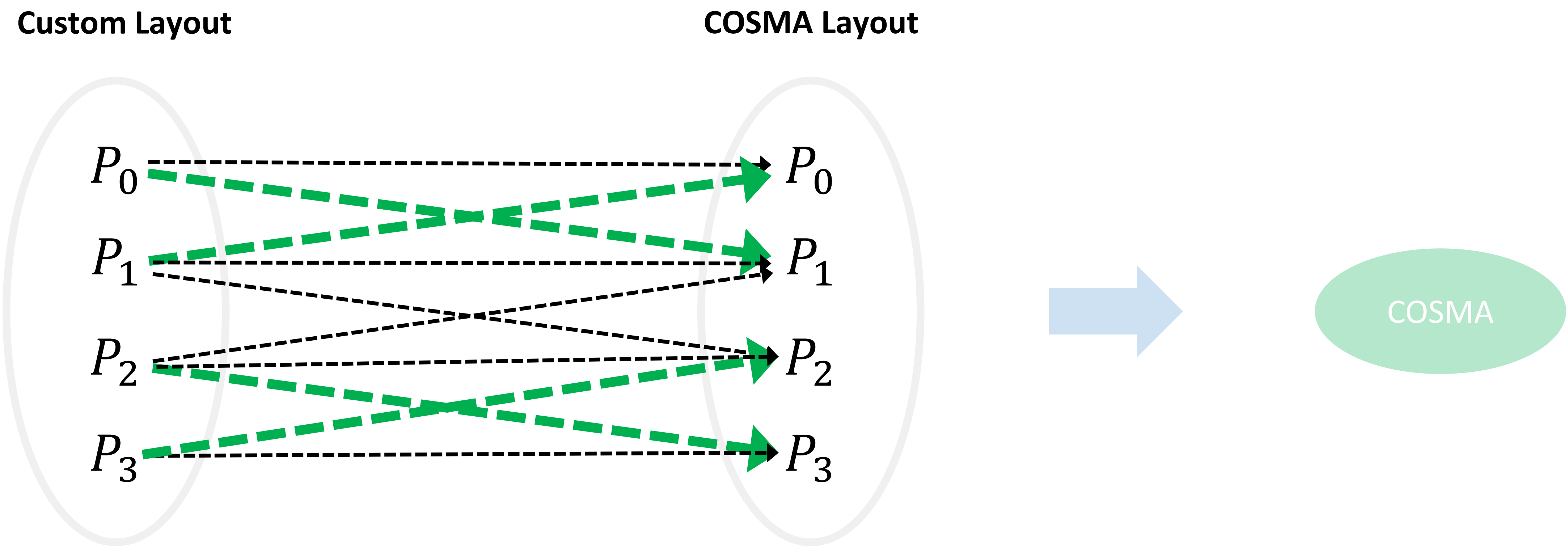
IDEA: relabel the ranks to minimize the communication cost

CUSTOM LAYOUT



IDEA: relabel the ranks to minimize the communication cost

CUSTOM LAYOUT



Optimal relabeling = **maximum weighted perfect matching.**