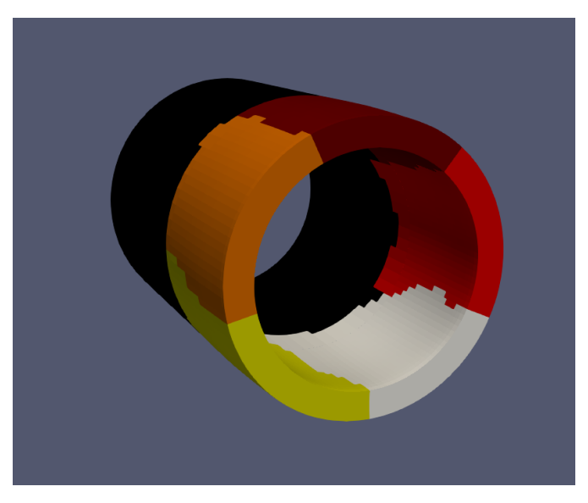


Evaluating Asynchronous Schwarz Solvers on GPUs

Pratik.Nayak @kit.edu

Motivation and Objectives

Exascale computing requires avoidance of global synchronization for efficient use of resources. **Asynchronous** methods can be an answer to this problem. We aim to study:

1. The effects of **communication patterns**;
2. The effects of **convergence detection** methods;
3. Scaling on a **multi-GPU** cluster;

on a domain decomposed Schwarz problem.

Method

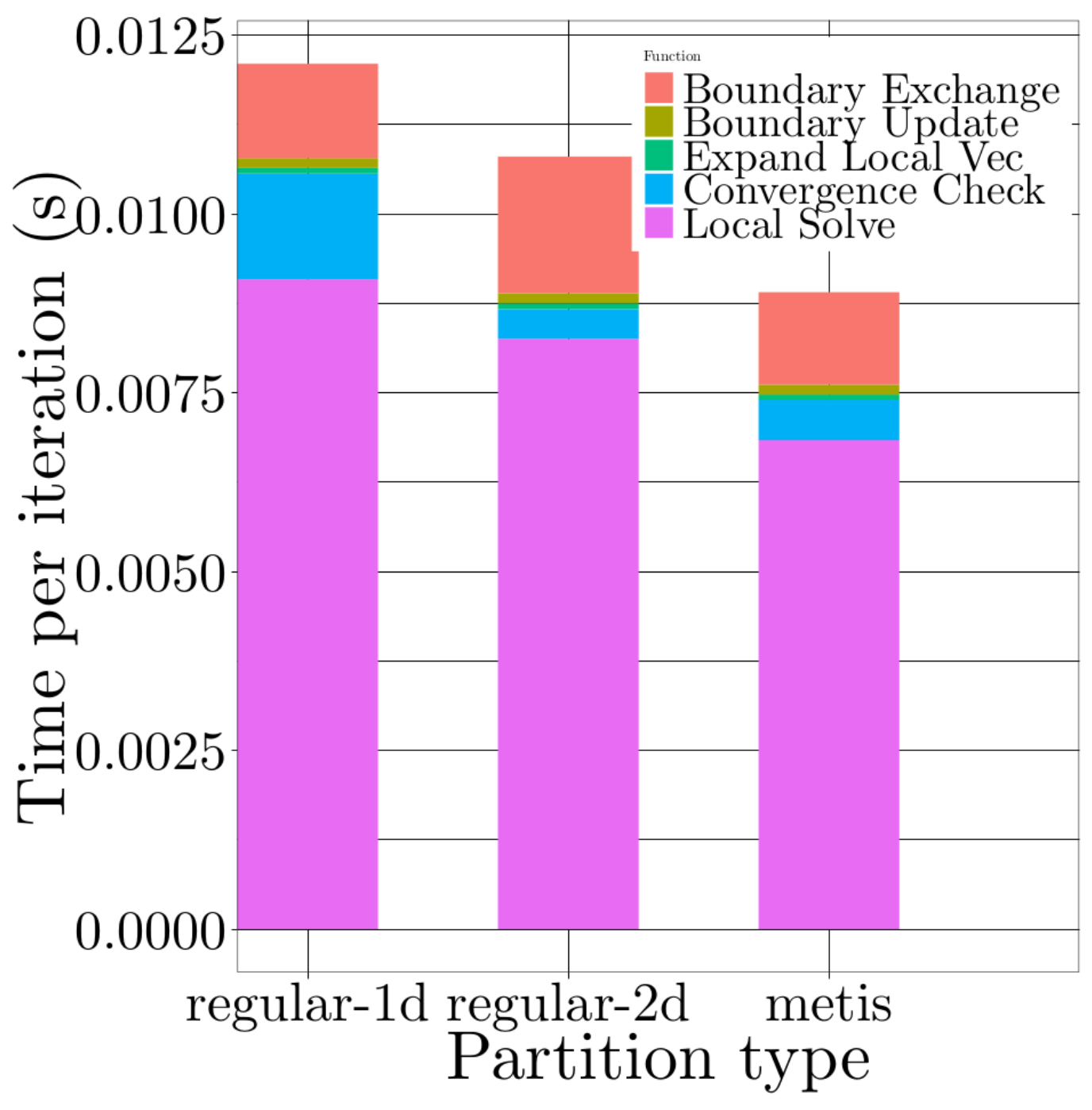
Algorithm 1 Schwarz Iterative solver

```
procedure ITERATIVE SOLUTION($A, x, b$)
    procedure INITIALIZATION AND SETUP
        Partition matrix                         ▷ Regular / objective based
        Distribute data
        Initialize data
    procedure SOLVE
        while $iter < max\_iter$ or until convergence do
            Locally solve the problem            ▷ Iterative / Direct
            Exchange boundary information
            Update boundary information
            Check for Convergence                ▷ Centralized/Decentralized
        Gather the final solution vector
```

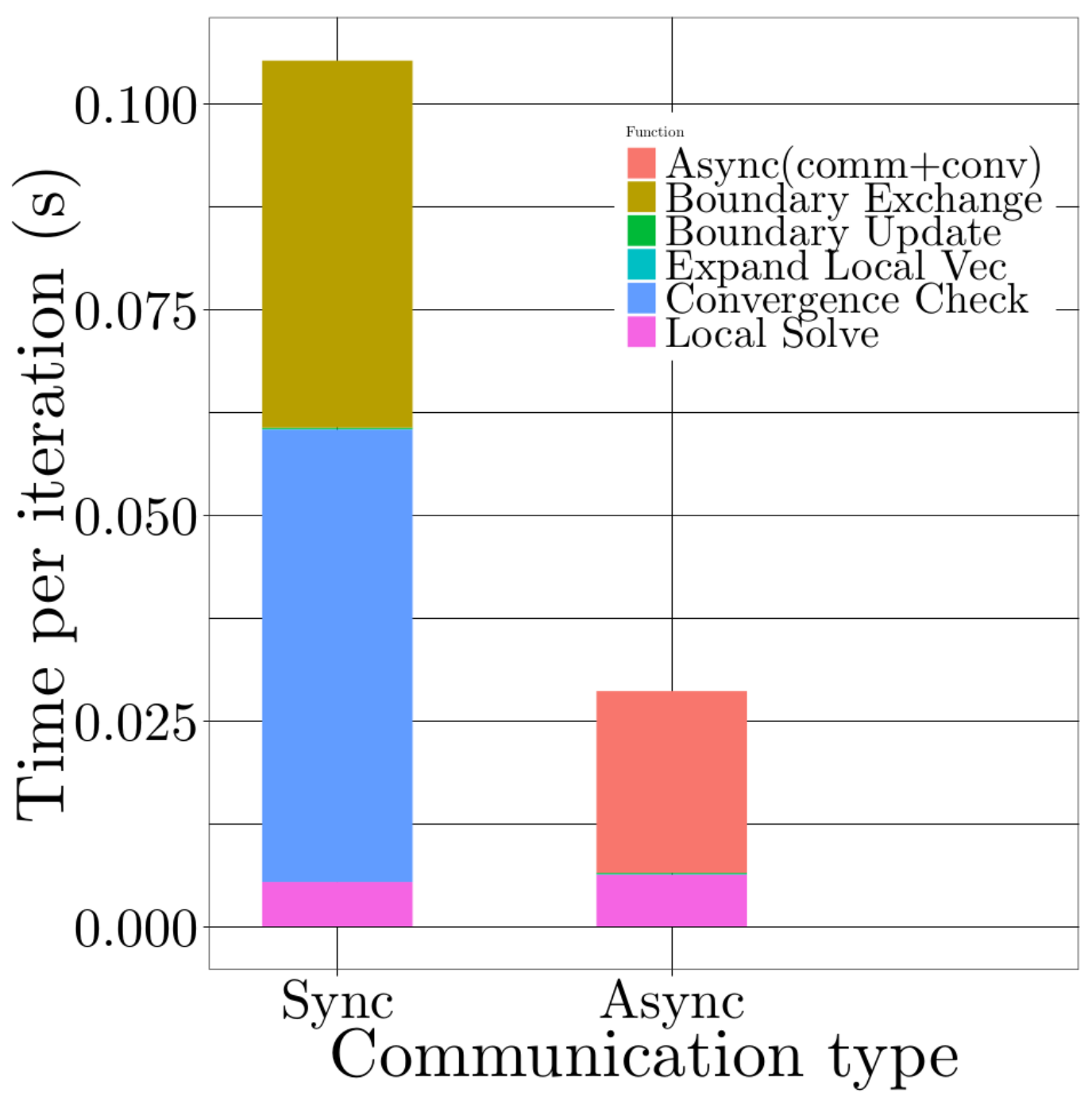
Experimental setup

1. Experiments run on Summit with one GPU (V100) for each subdomain.
2. Uses CUDA Aware version of Spectrum MPI with the gcc-7.4.0 compiler.
3. Local direct solves with CHOLMOD from the Suitesparse collection with local reordering.

Results - Function Split



Time per iteration between different partitions.



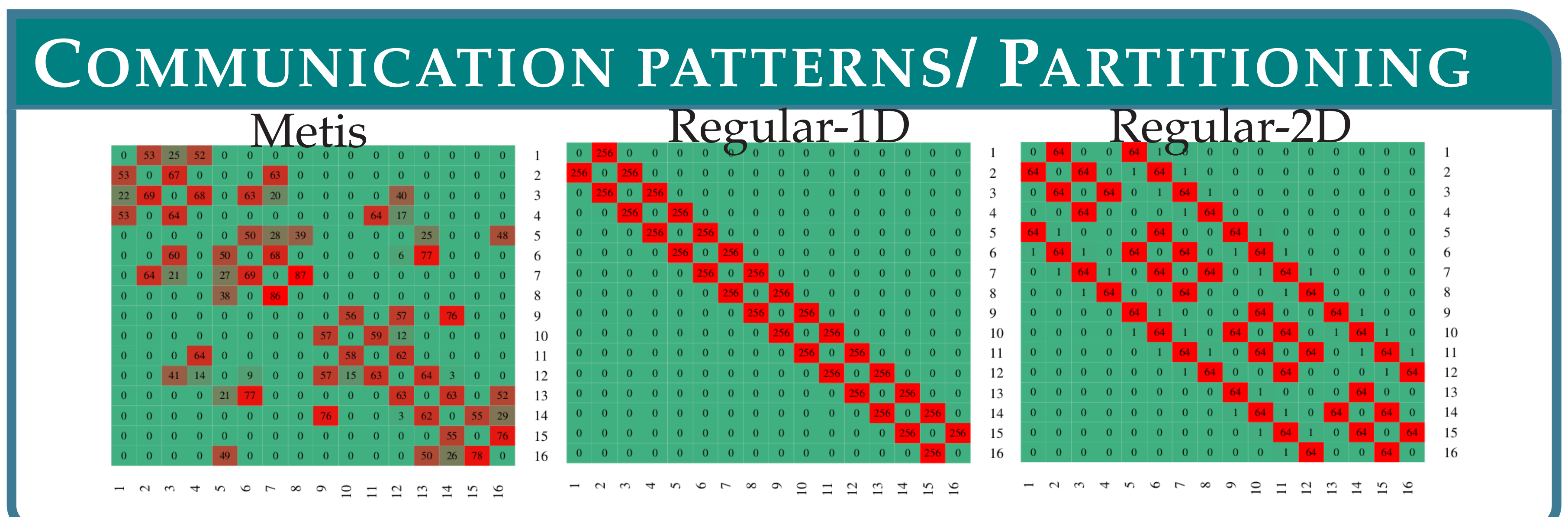
Comparison between Sync and Async version. Local solve times being the same, the async achieves much better communication and convergence timings.

Contributions

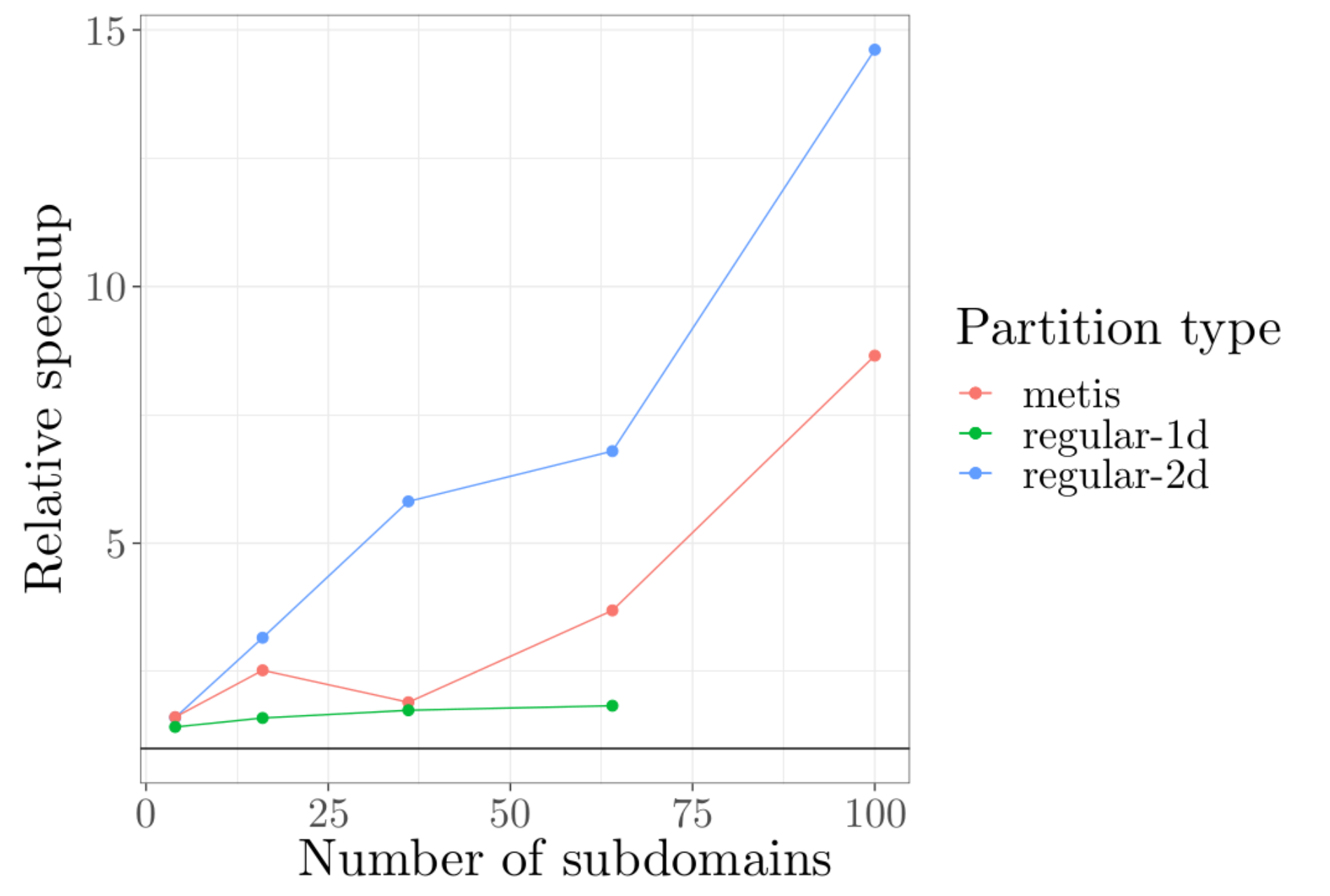
A generic framework to solve linear systems with Schwarz methods

1. Extensible Open-source implementation.
2. Test-bed with Ginkgo [1], `deal.ii`[2] and MPI+CUDA.
3. Different partitioning and convergence detection algorithms.

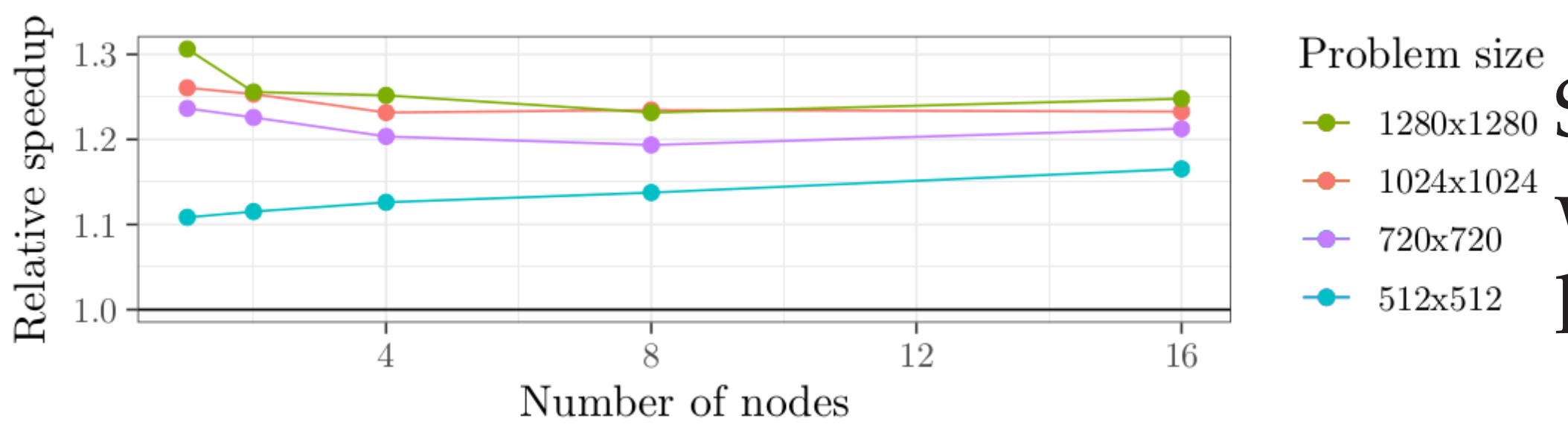
Communication patterns/ Partitioning



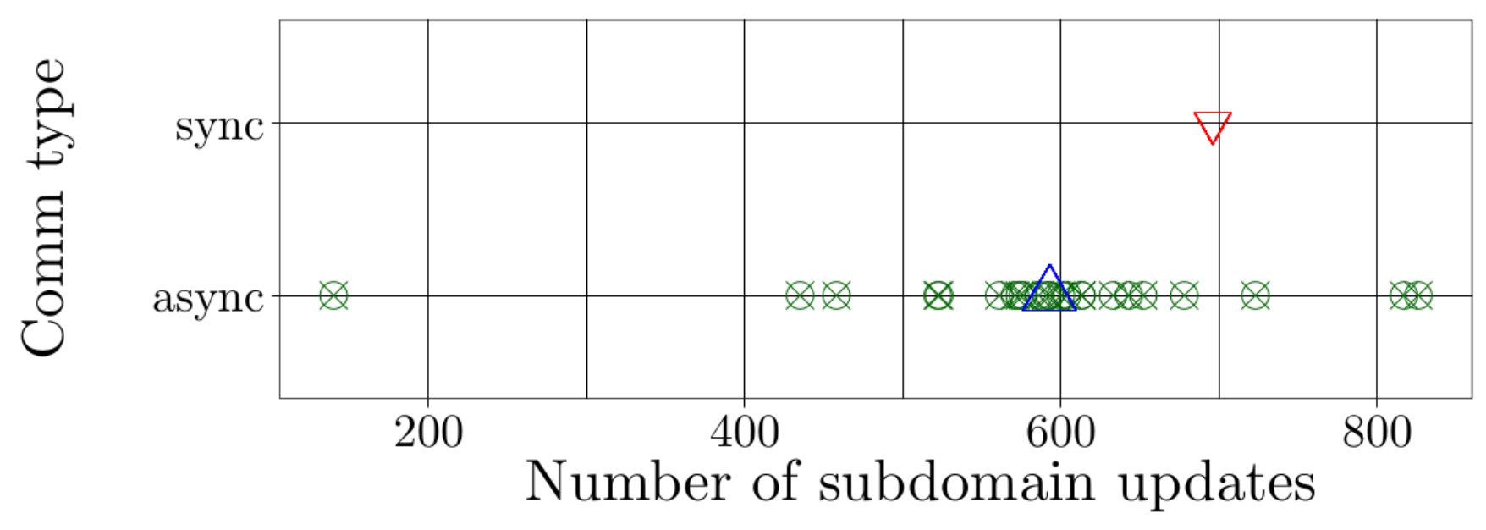
Results - Asynchronous Speedup



Speedup of async over sync for different partition schemes.



Speedup of async over sync with increasing global problem size. Strong scaling.



Number of subdomain updates for different subdomains. Sync: Same for all subd's. Async: Varies; median lesser than sync.

Conclusions

1. A regular partitioning has a slow rate of information propagation.
2. The asynchronous version achieves speedup due to the cheaper cost per iteration and on average lower iterations per subdomain.
3. The Restricted Schwarz method has some limitations on scaling due to the lack of global information exchange.
4. Removing the bulk-synchronous nature can be beneficial even for well-balanced problems.
5. More details and analysis in Preprint paper [4].

References

[1] The Ginkgo library `https://github.com/ginkgo-project/ginkgo`
[2] G. Alzetta et.al, The `deal.ii` library: Version 9.0 In *J. Num. Math, 2018*
[3] M. J. Gander, Optimized Schwarz methods In *SIAM J. Numer. Anal./ 44(2), 699-731*
[4] P. Nayak, T. Cojean, H. Anzt, Evaluating Abstract Asynchronous Schwarz Solvers In *arXiv e-prints, arXiv:2003.05361*

Future Directions

1. Optimized Schwarz methods for better information exchange [3]
2. A two level preconditioner with global information propagation.
3. Investigation of heterogenous, GPU+CPU solver.

`https://github.com/pratikvn/schwarz-lib`