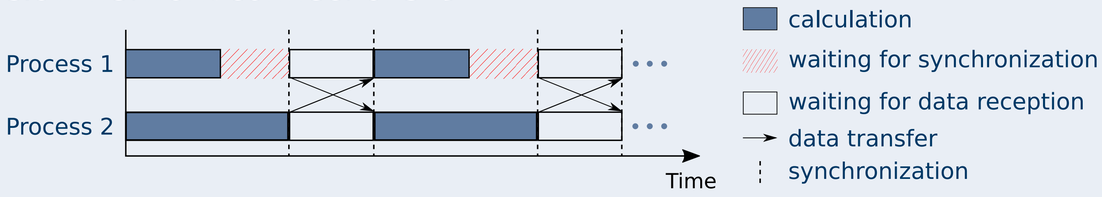
# Distributed Asynchronous Jacobi Methods

Christoph Schwarzmeier, Alexander Ditter, Dietmar Fey

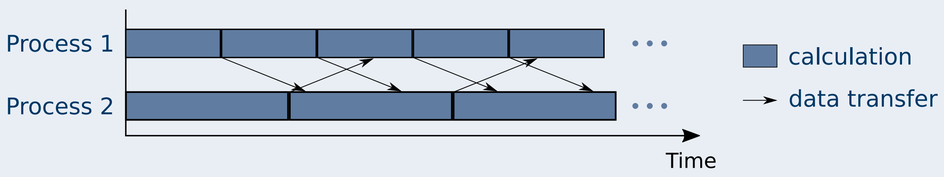
## Motivation

- Solving $Ax=b$ is a common task in parallel computing
- **Synchronous** algorithms are dominated by
  - ▶ individual slow processes
  - ▶ slow network connections

  } degrade overall performance




- **Asynchronous** algorithm




➡ avoiding synchronization decreases time-to-solution

## Iterative Methods

- **Synchronous** iterative method using $p$ processes in parallel

$$x_i^j = f_i\left(x_1^{(j-1)}, x_2^{(j-1)}, \ldots, x_n^{(j-1)}\right) \quad \text{for} \quad i = 1, 2, \ldots, p$$

  ➡ in iteration $j$, processes use other processes' data from the **preceding iteration** $j-1$

  ➡ **synchronization**

- **Asynchronous** iterative method using $p$ processes in parallel [1]

$$x_i^j = f_i\left(x_1^{(s_1(j))}, x_2^{(s_2(j))}, \ldots, x_n^{(s_n(j))}\right) \quad \text{for} \quad i = 1, 2, \ldots, p$$

  ➡ in iteration $j$, processes use other processes' data from **any iteration** $s_k(j)$

  ➡ **no synchronization**

## Implementation

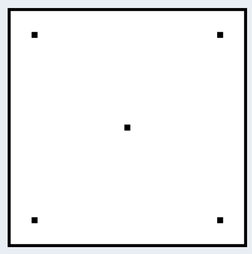
- Test problem: **Laplace's equation** $\Delta u(x, y) = 0$
  - ▶ discretized with the finite difference method
  - ▶ solved with the **Jacobi method**
- **C++**11 using Message Passing Interface (**MPI**) 3.0
- Algorithm

```
while not converged globally {
    Jacobi iteration
    asynchronous communication
}
```

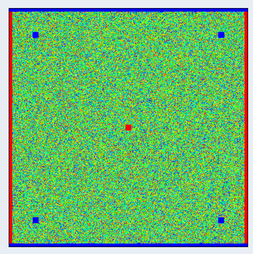
- Asynchronous **convergence detection** [2]
  - ▶ master process checks processes' individual convergence
  - ▶ individual convergence is reached if
    - a) local residual < threshold
    - b) no iterations have been performed since the last data sending
    - c) all send operations are completed
- Asynchronous **communication**
  - ▶ data is sent using `MPI_Issend()` if
    - a) data has been updated
    - b) a previously initiated sending to the same process is completed
  - ▶ data is received using `MPI_Irecv()` if a previously initiated receiving from the same process is completed

## Evaluation

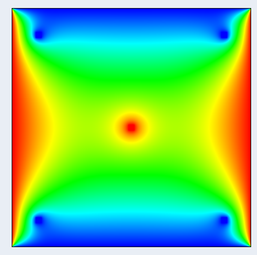
- 400x400 grid point domain, homogeneously tile-partitioned



Dirichlet boundary conditions

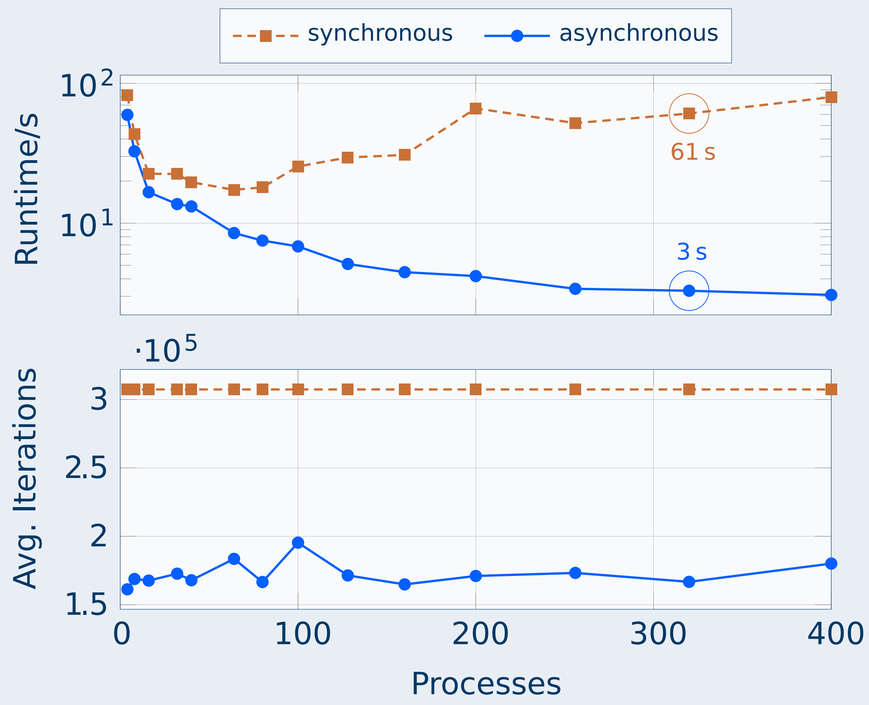


Standard uniform $U(0,1)$ initialization

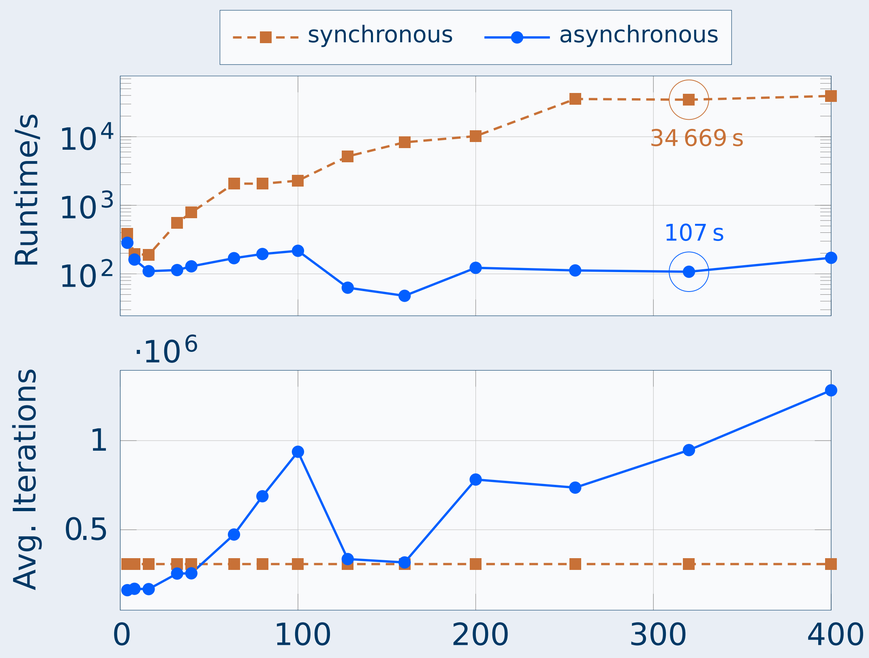


Result

- **HPC Cluster** with
  - ▶ 560 compute nodes with 2 Intel Xeon E5-2660 v2 processors
  - ▶ 10 physical cores (+SMT) per processor = 40 processes per node
  - ▶ Infiniband network with **40 GBit/s bandwidth**




- **Cloud infrastructure** with 19 virtual compute nodes based on
  - ▶ compute nodes with 2 Intel Xeon X5660 processors
  - ▶ 6 physical cores (+SMT) per processor = 24 processes per node
  - ▶ Ethernet network with **1 GBit/s bandwidth**




## References

[1] G. M. Baudet, "Asynchronous iterative methods for multiprocessors", *Journal of the Association for Computing Machinery*, vol. 25, no. 2, Apr. 1978

[2] D. P. Bertsekas and J. N. Tsitsiklis, "Convergence rate and termination of asynchronous iterative algorithms", in *Proceedings of the 3rd International Conference on Supercomputing*, ser. ICS '89, New York: ACM, 1989

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Chair of Computer Science 3
Computer Architecture
Prof. Dr.-Ing. Dietmar Fey