German Research School
for Simulation Sciences

Catching Idlers with Ease: A Lightweight Wait-State Profiler for MPI Programs

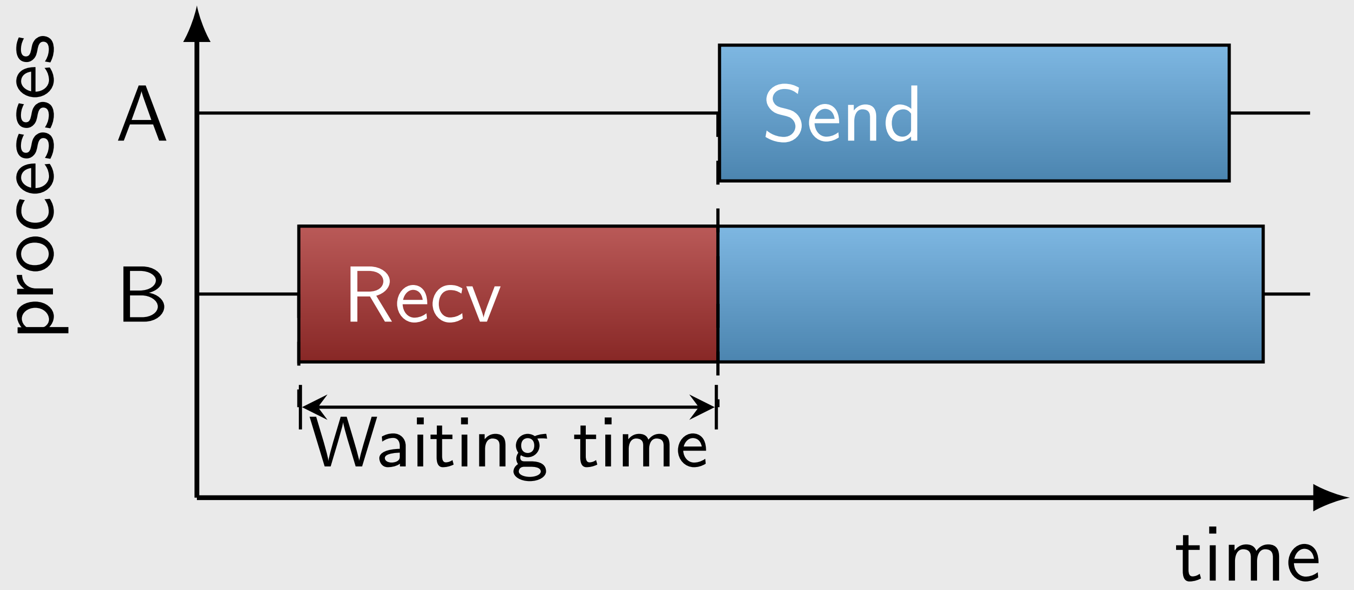
Guoyong Mao, David Böhme, Markus Geimer, Marc-André Hermanns, Daniel Lorenz and Felix Wolf

Petascale Tools Workshop, Madison, WI, USA, August 4, 2014

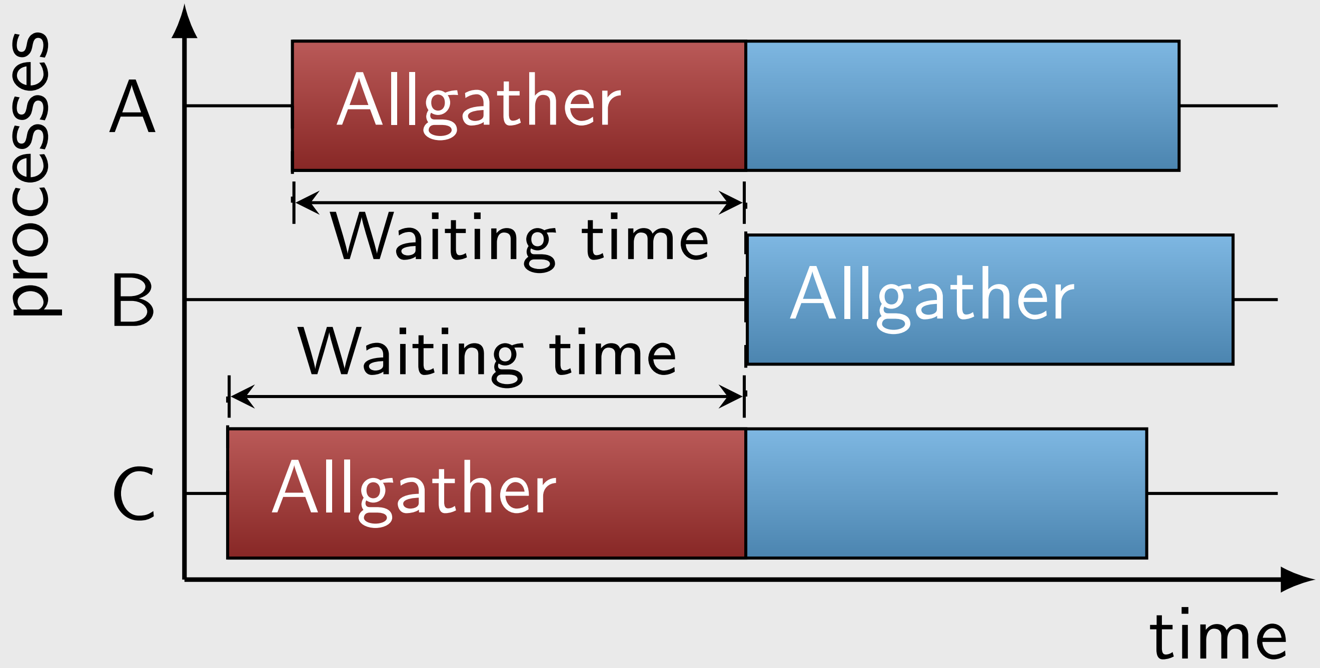
JÜLICH FORSCHUNGSZENTRUM

RWTH AACHEN UNIVERSITY

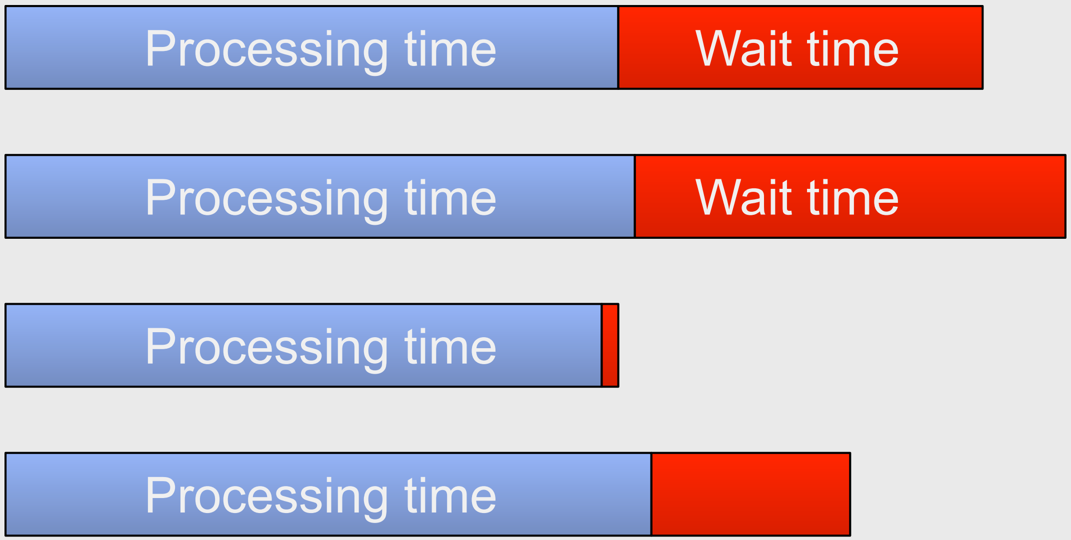
Late sender



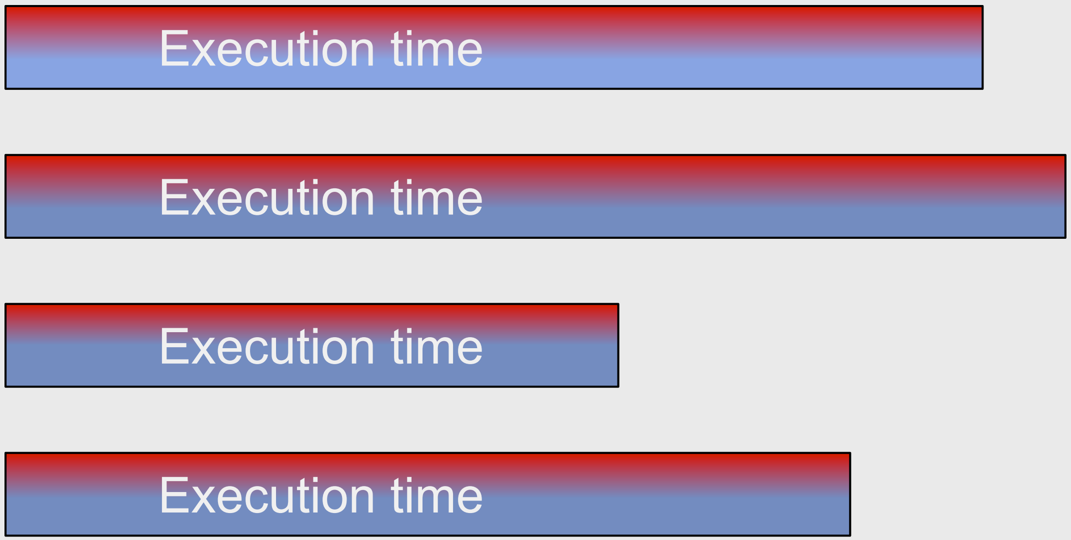
Wait an NxN


processes
A
Allgather
Waiting time
B
Allgather
Waiting time
C
Allgather
time

What we want to know

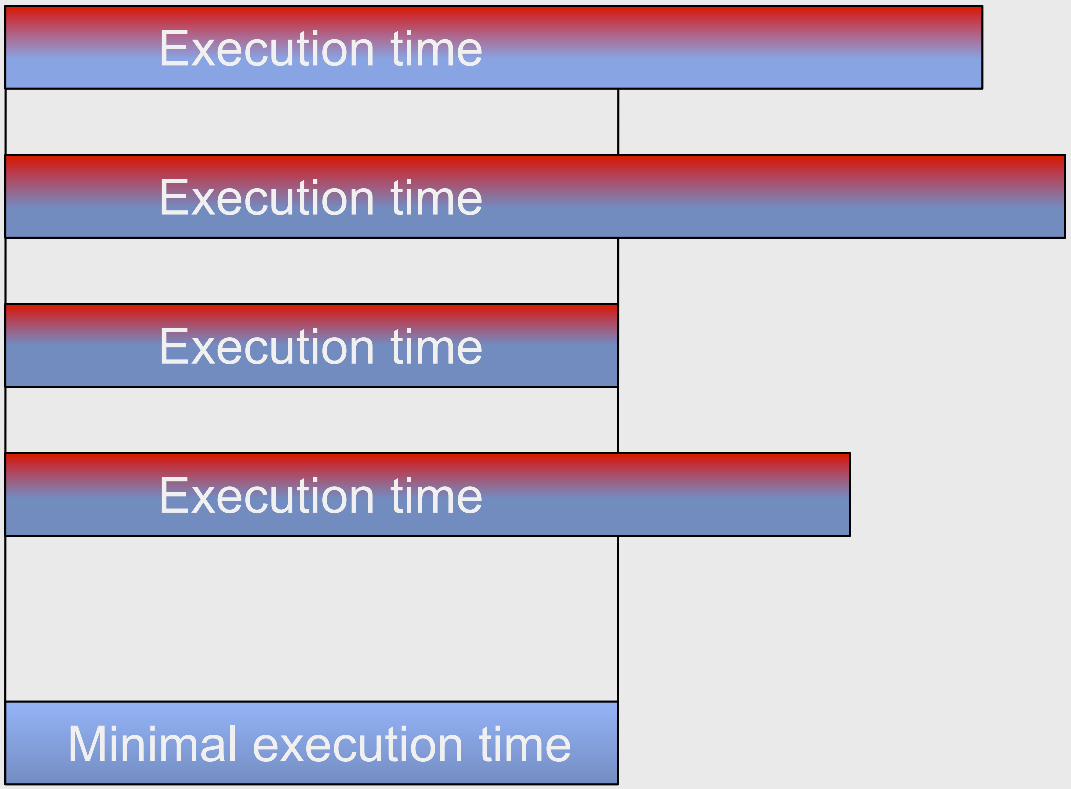


What we measure

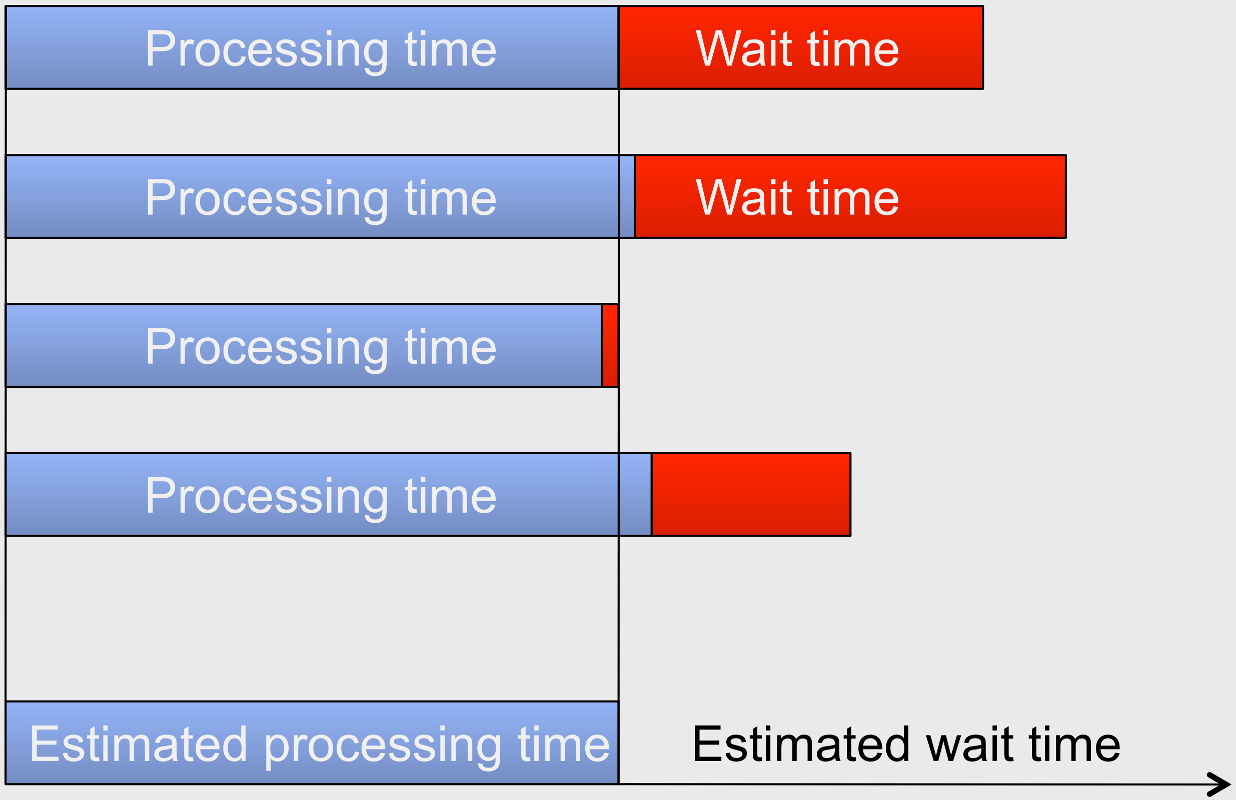


German Research School
for Simulation Sciences

The minimum idea



The minimum idea



Considered parameters

- We consider
  - MPI function
  - Message size
  - Receiver rank

- Other possible parameters
  - Sender rank
  - Data type

- Tradeoff between
  - Number of samples for a meaningful minimum and amount data
  - Parameters considered

- Need to find the relevant parameters.

Algorithm

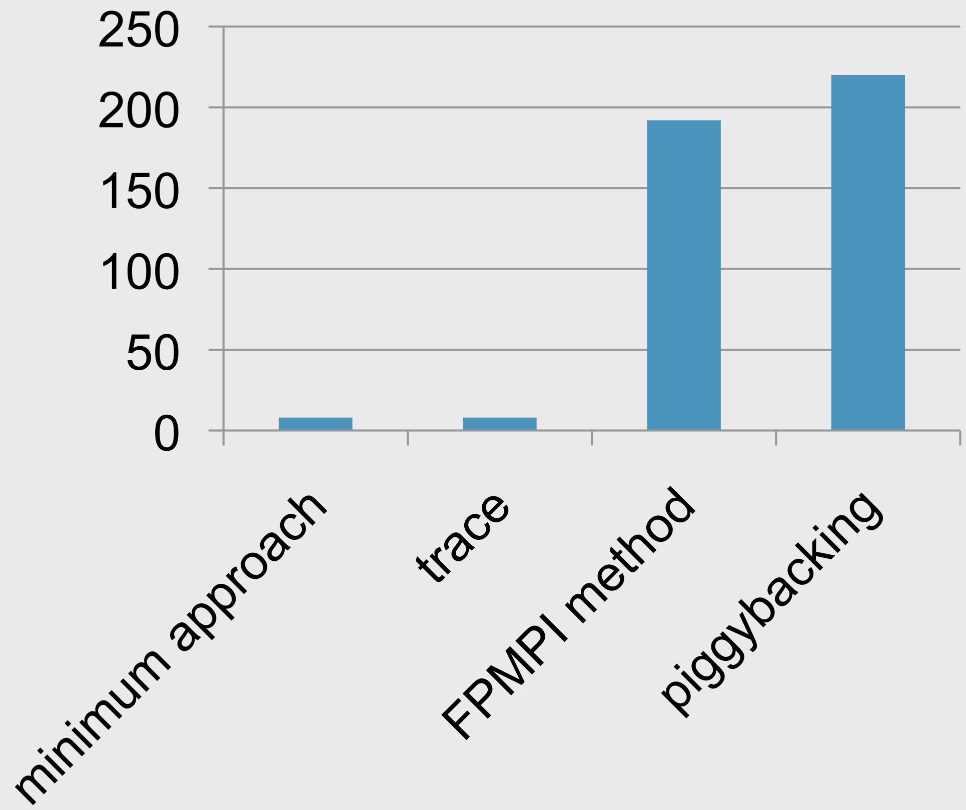
For every combination of
- MPI function
- Message size class
- Process

record the
- Minimum execution time

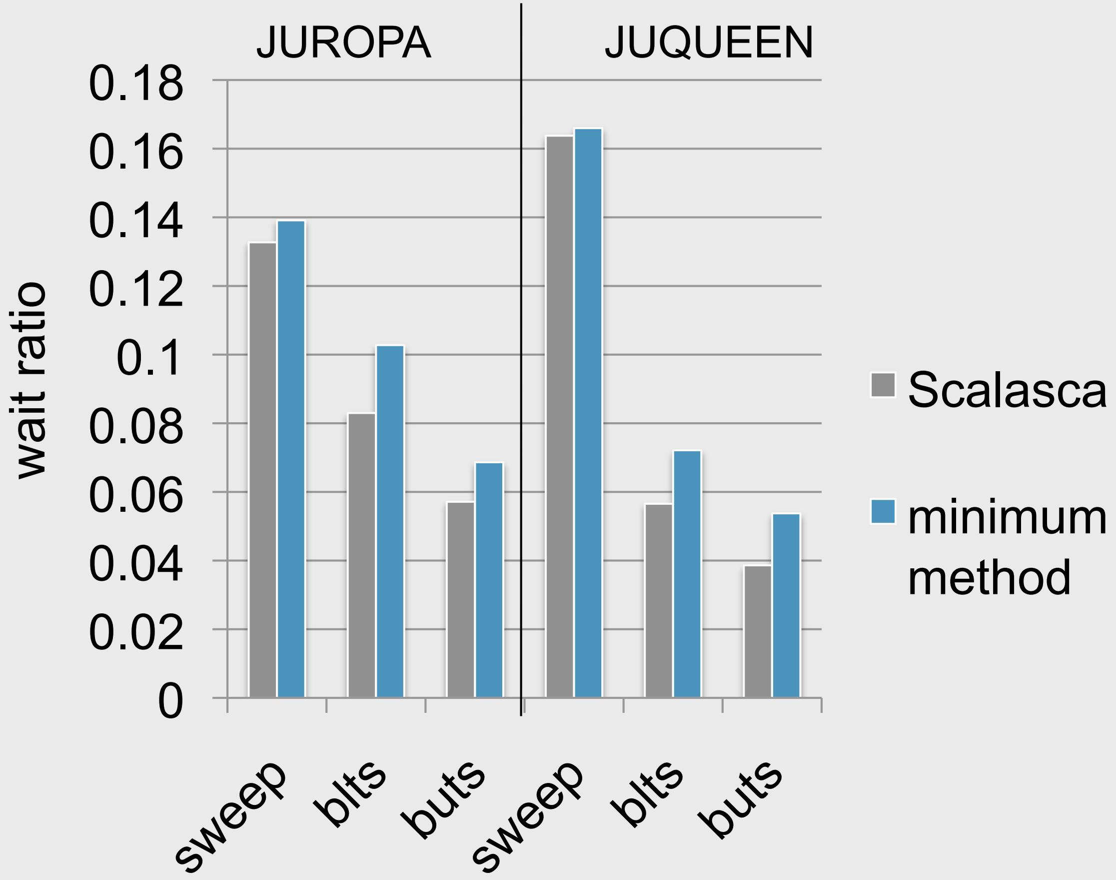
For every combination of MPI call path and message size class record the
- Number of visits
- Total execution time

At the end of the profiling run, subtract the minimum from the execution time for every visit to calculate the wait time.

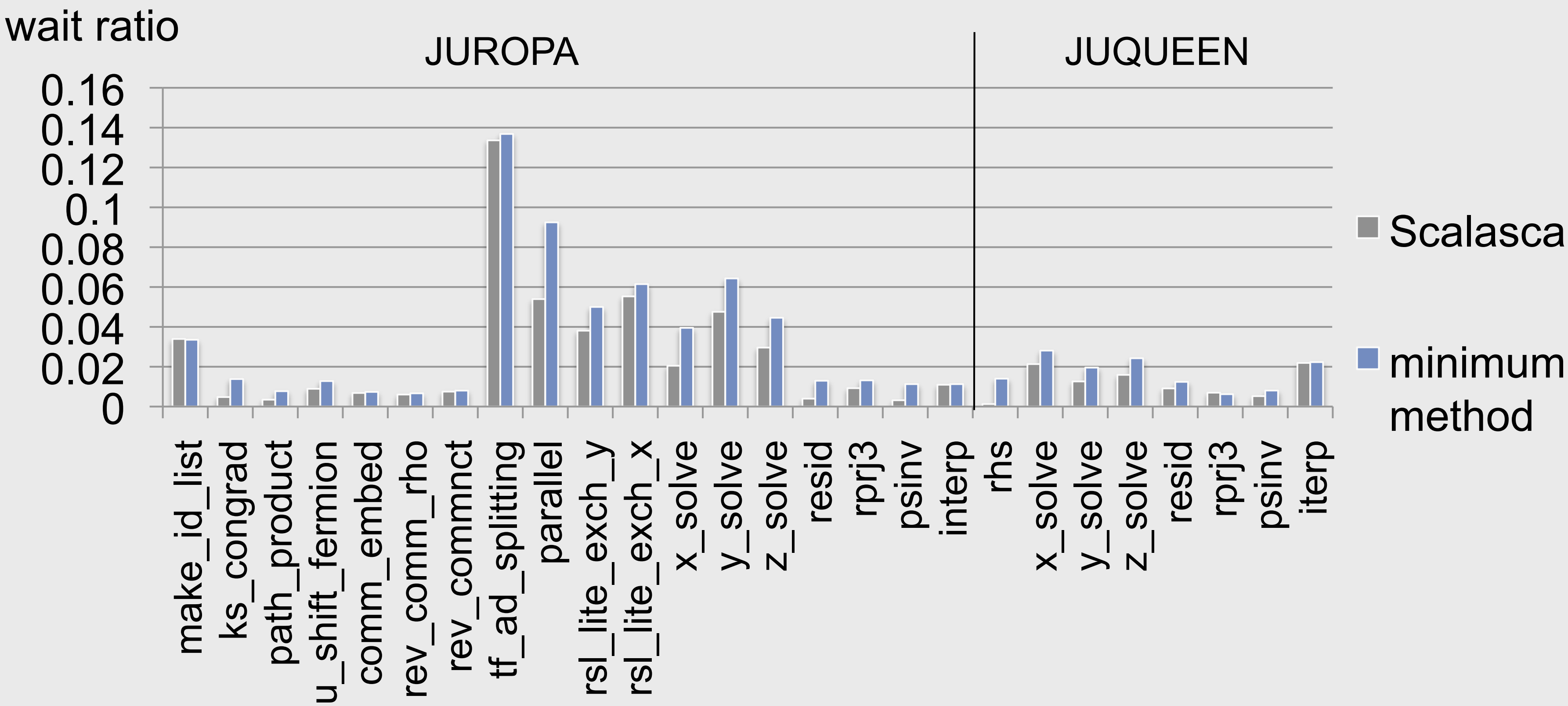
Per-call overhead increase compared to profiling overhead w/o wait state analysis (%)



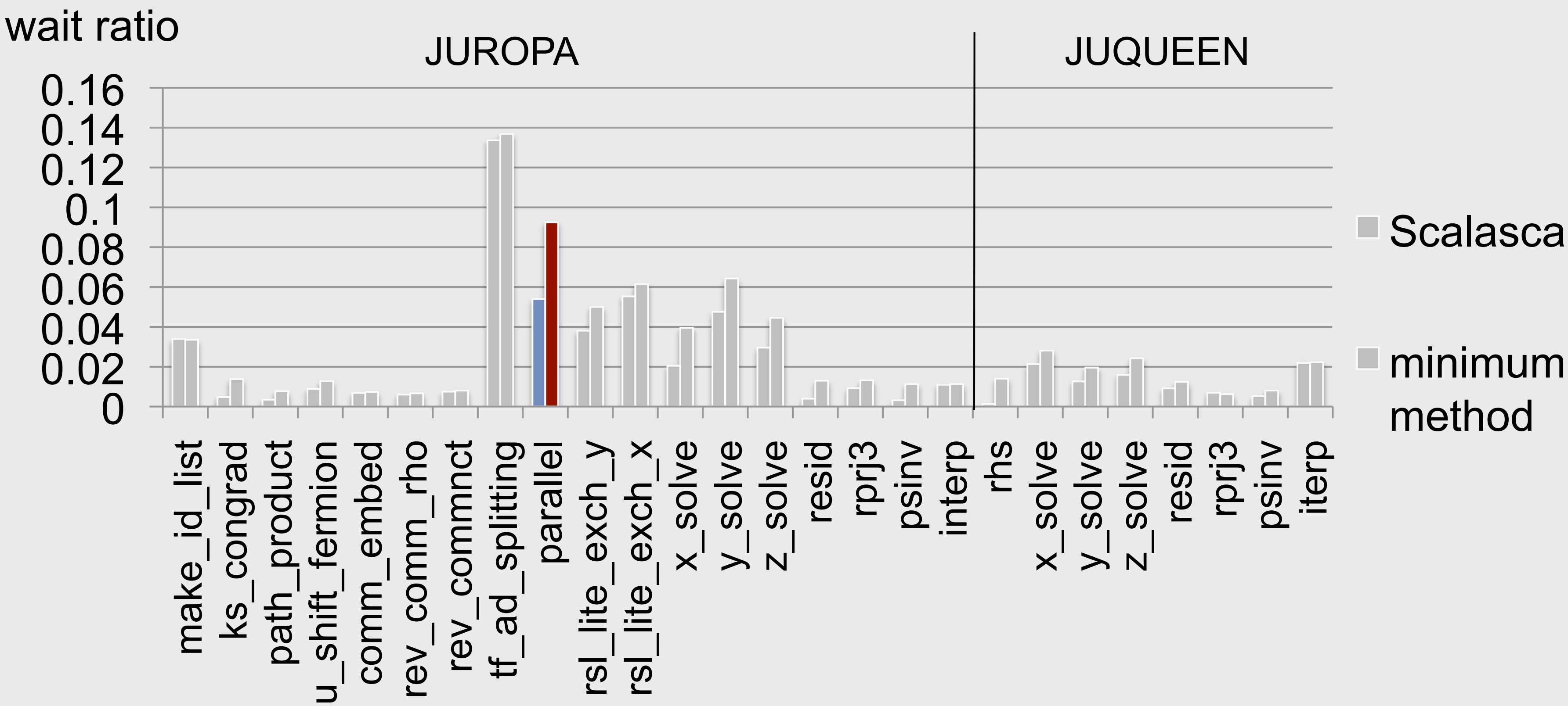
Accuracy MPI_Recv



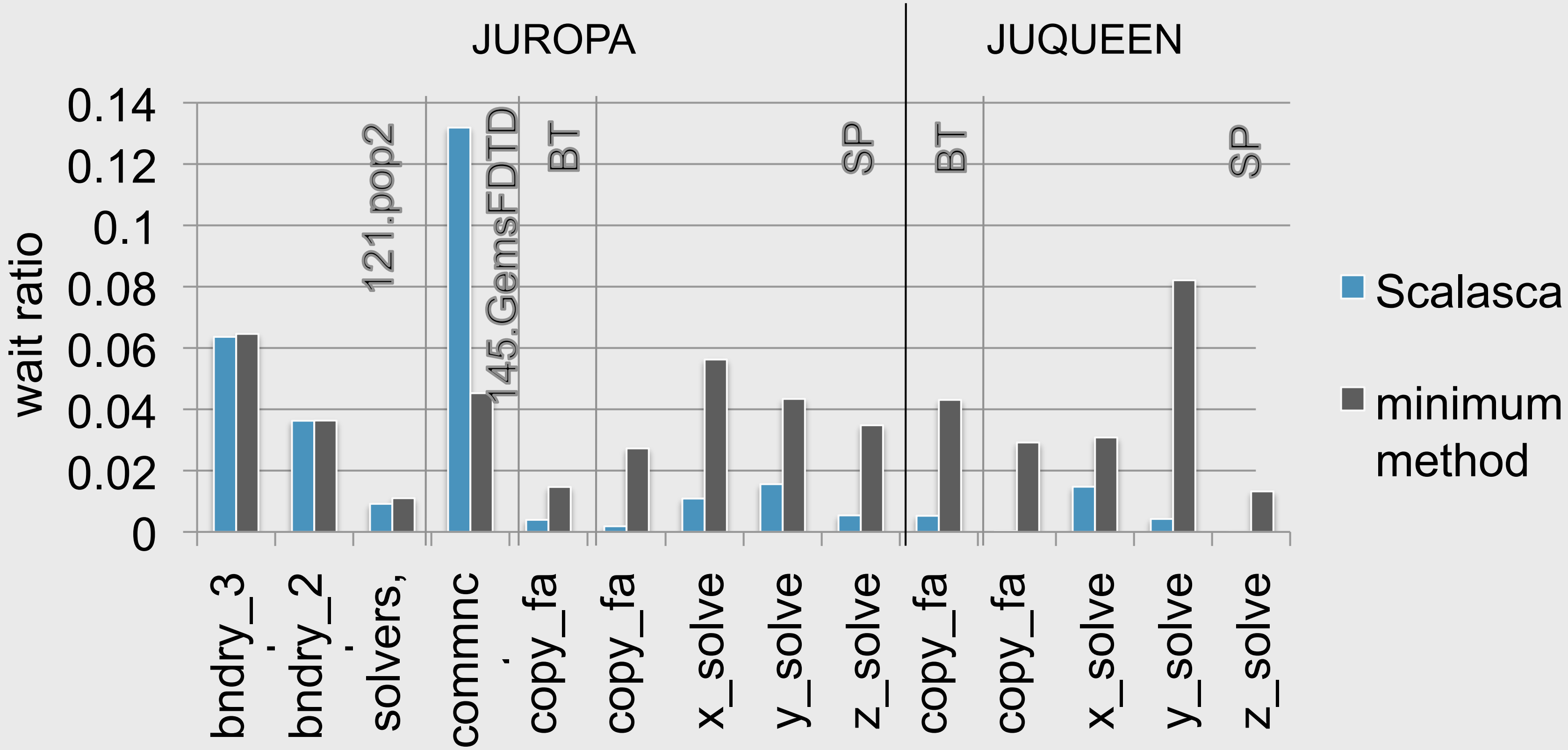
Accuracy MPI_Wait



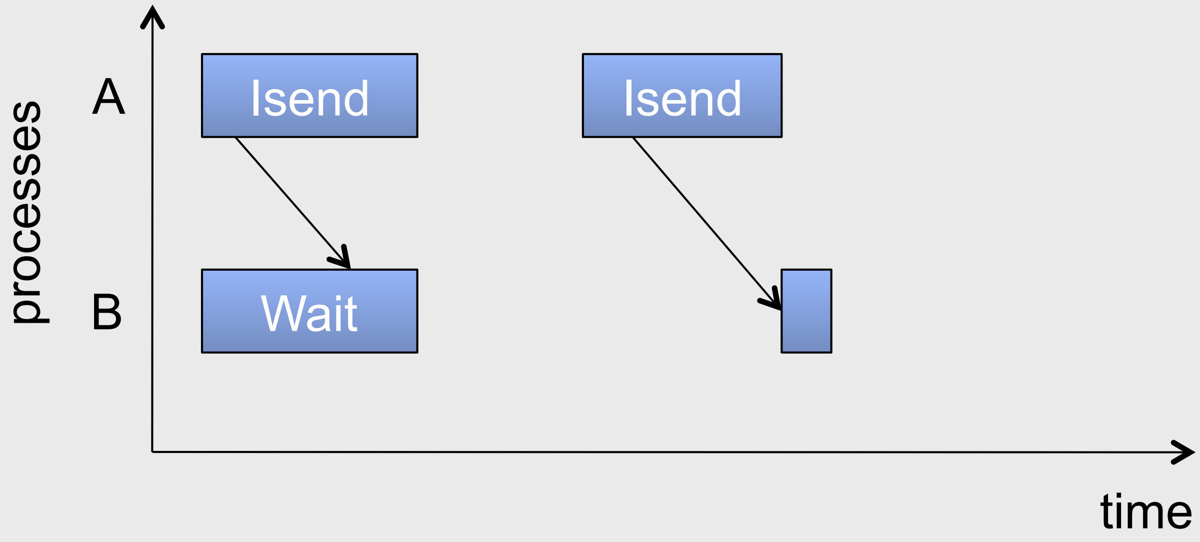
Accuracy MPI_Wait



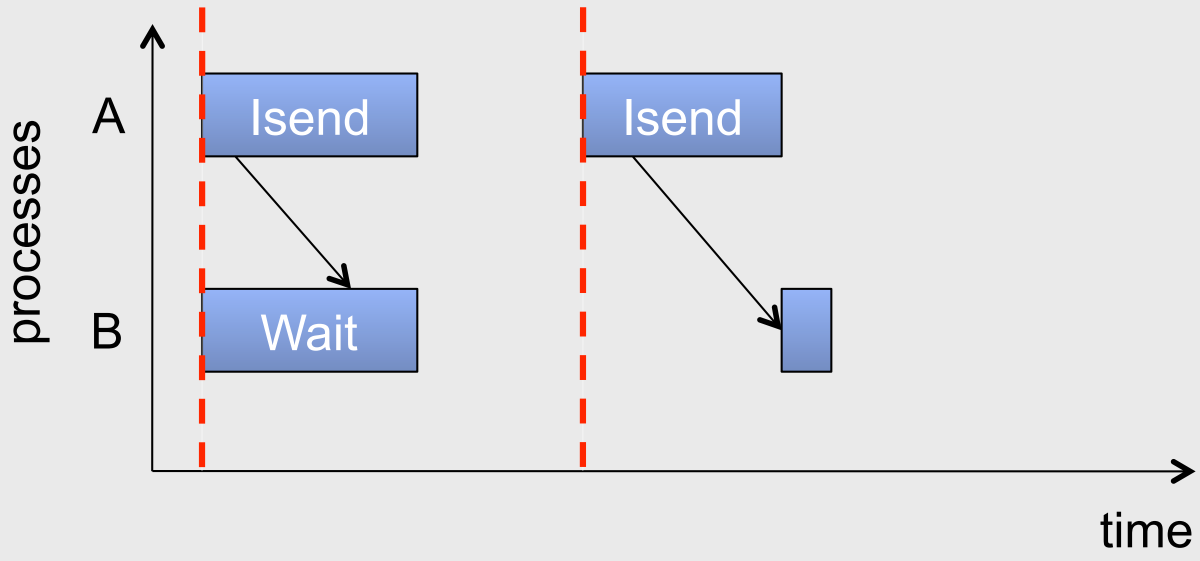
Accuracy MPI_Waitall



Non-blocking communication

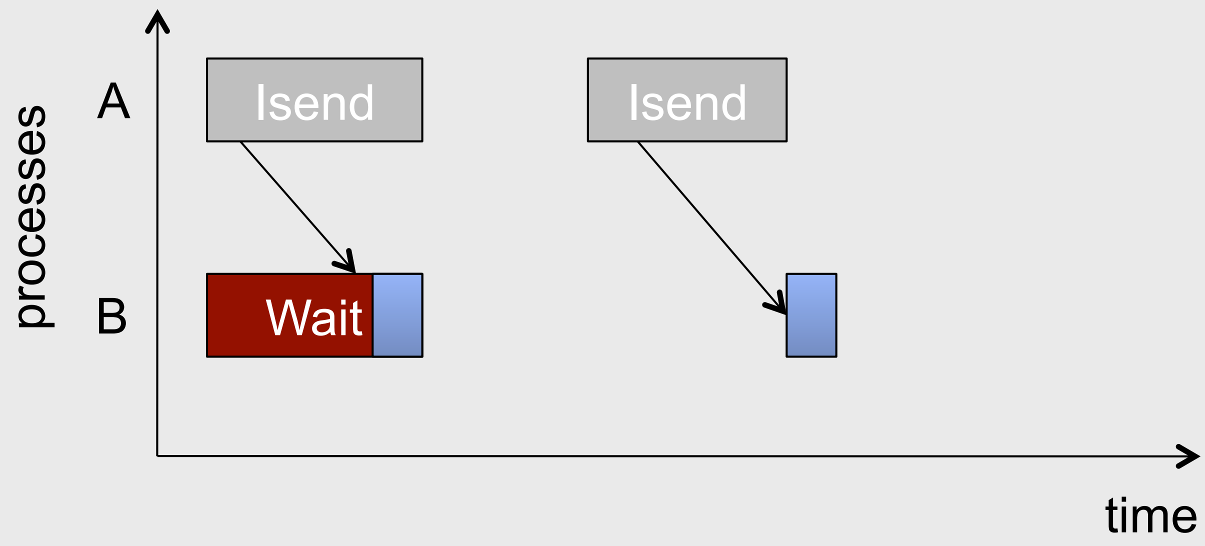


Scalasca detects no wait state



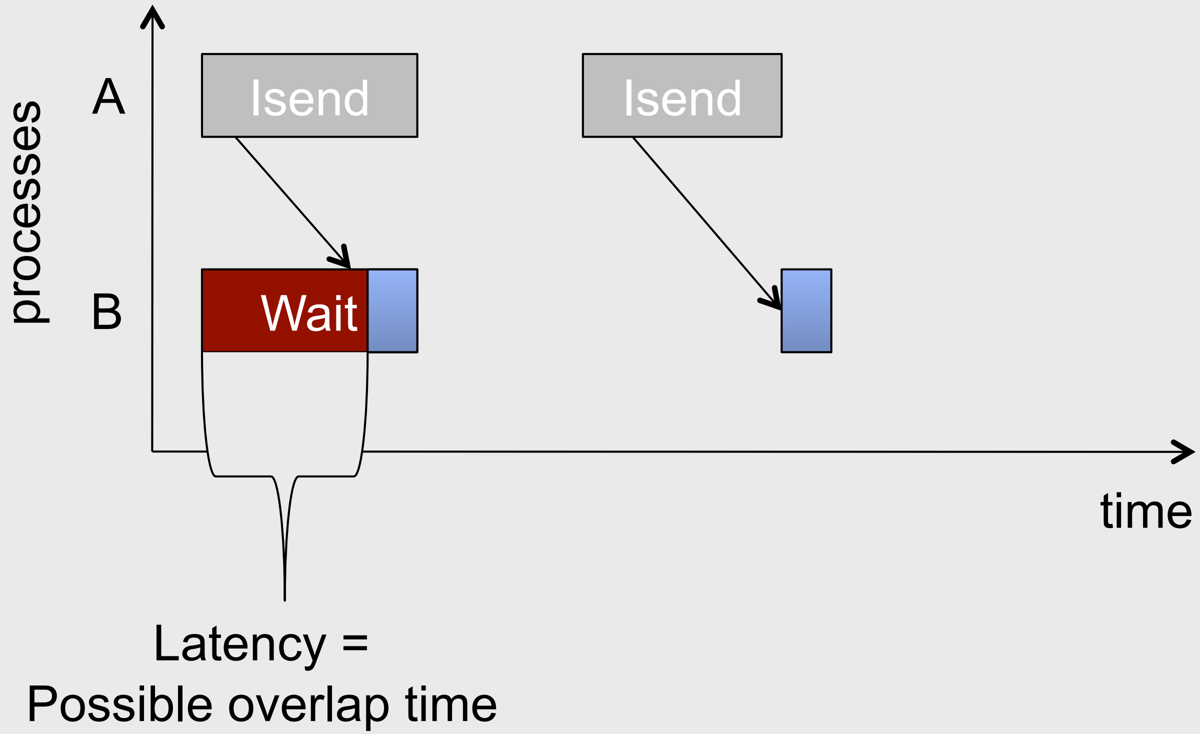
German Research School for Simulation Sciences

Minimum approach does calculate wait states

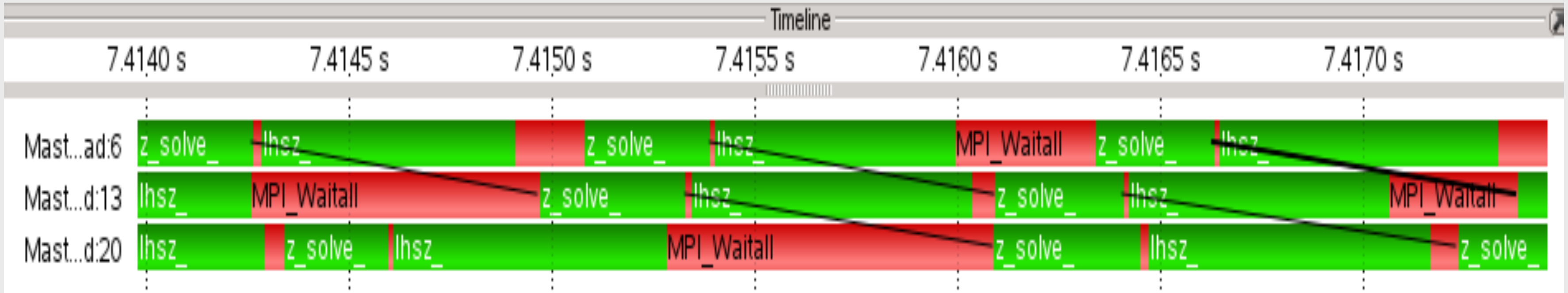


JÜLICH FORSCHUNGSZENTRUM RWTH AACHEN UNIVERSITY

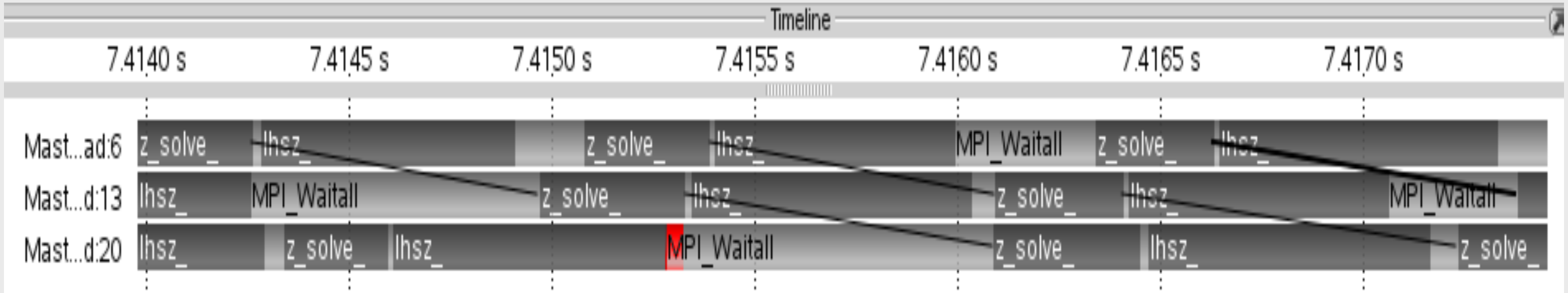
But is this wrong for performance analysis?



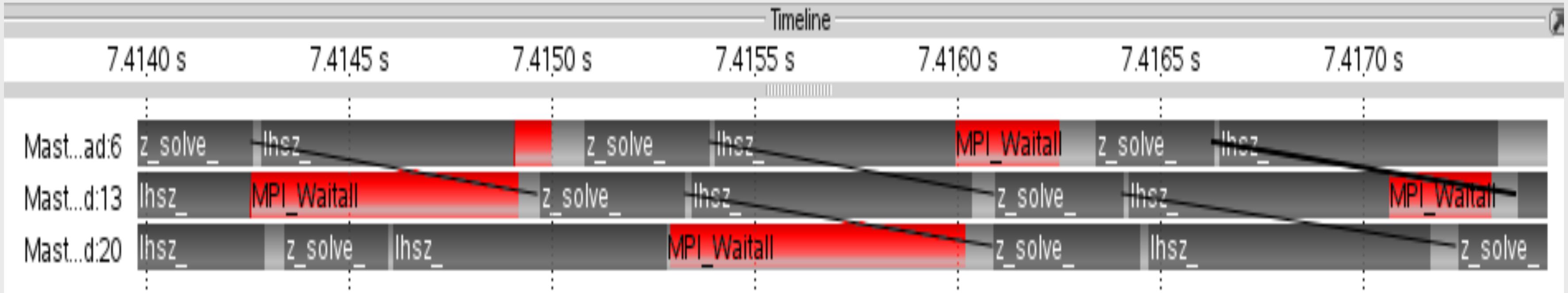
Detailed example from SP



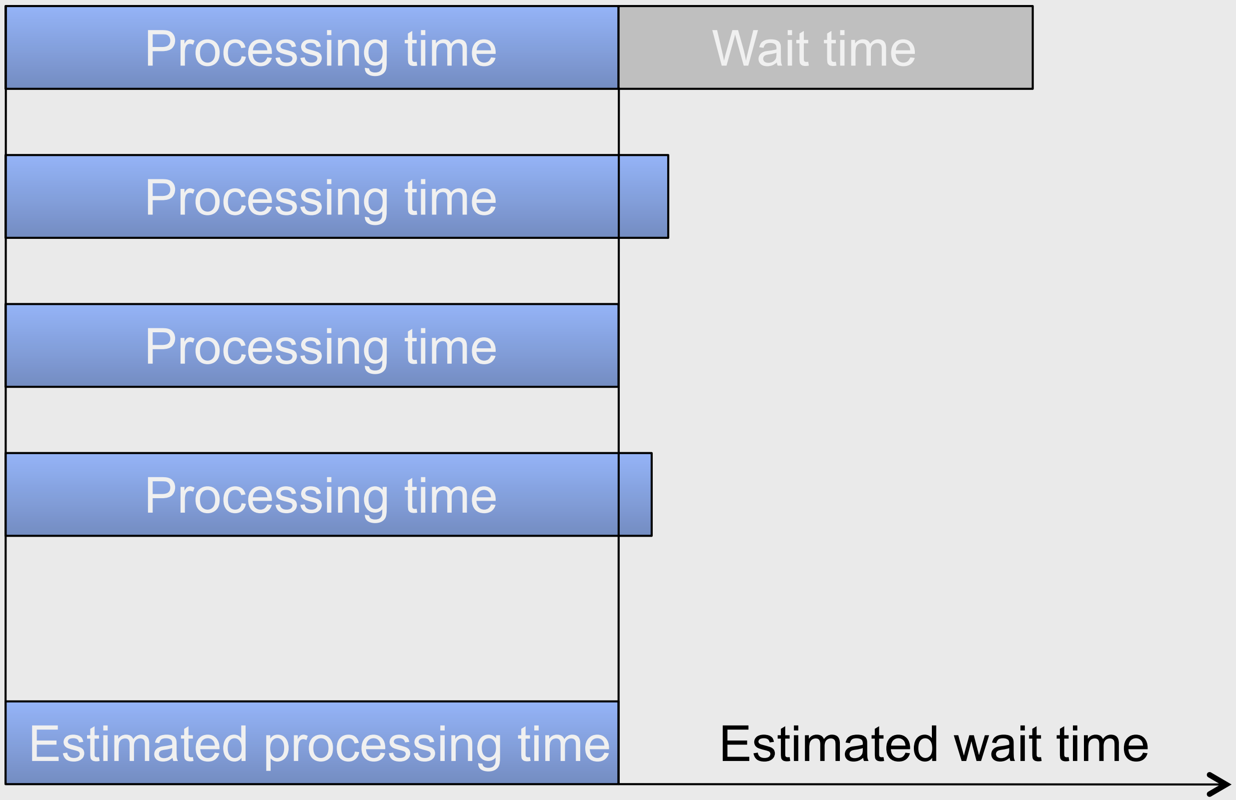
Wait time according to Scalasca



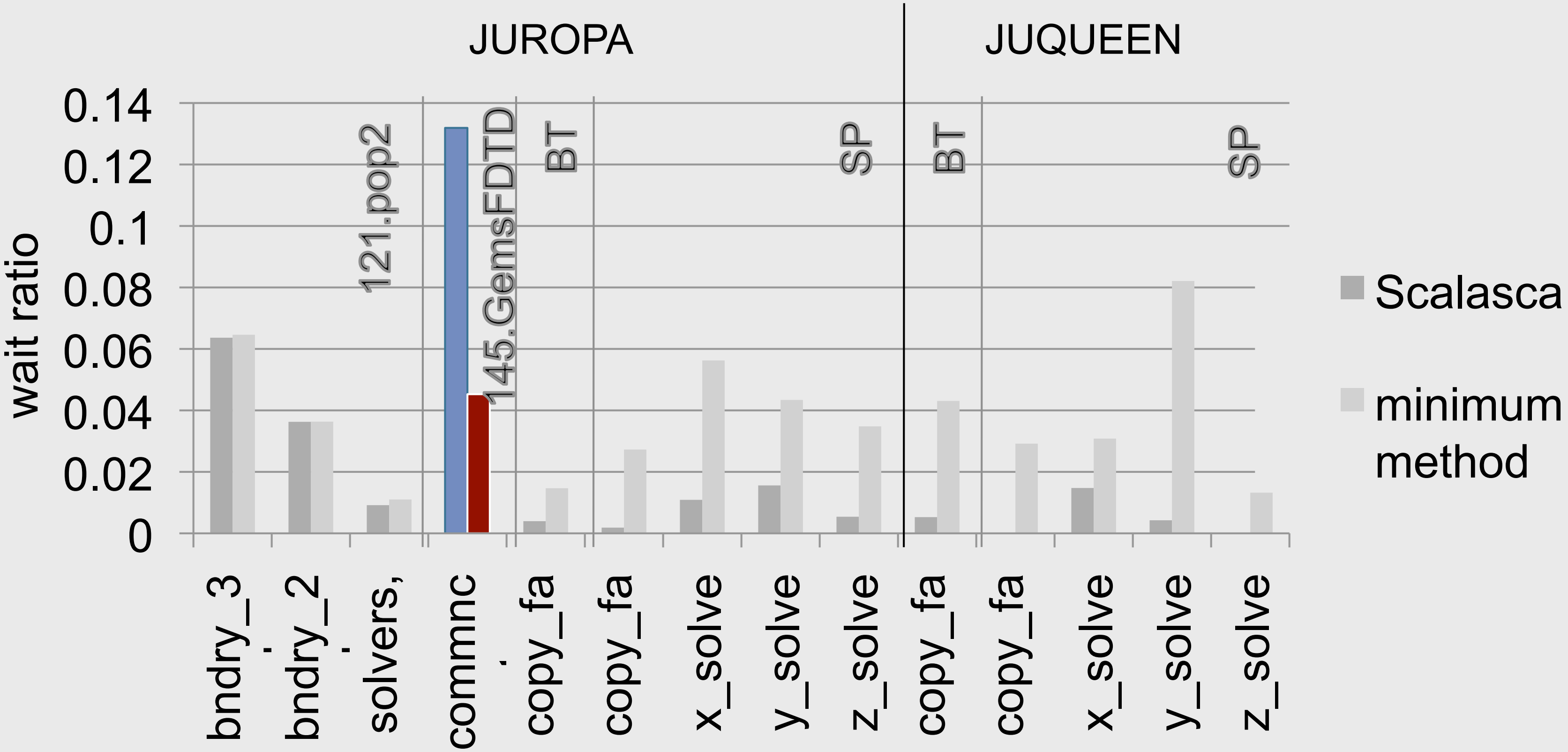
Wait time according to minimum method



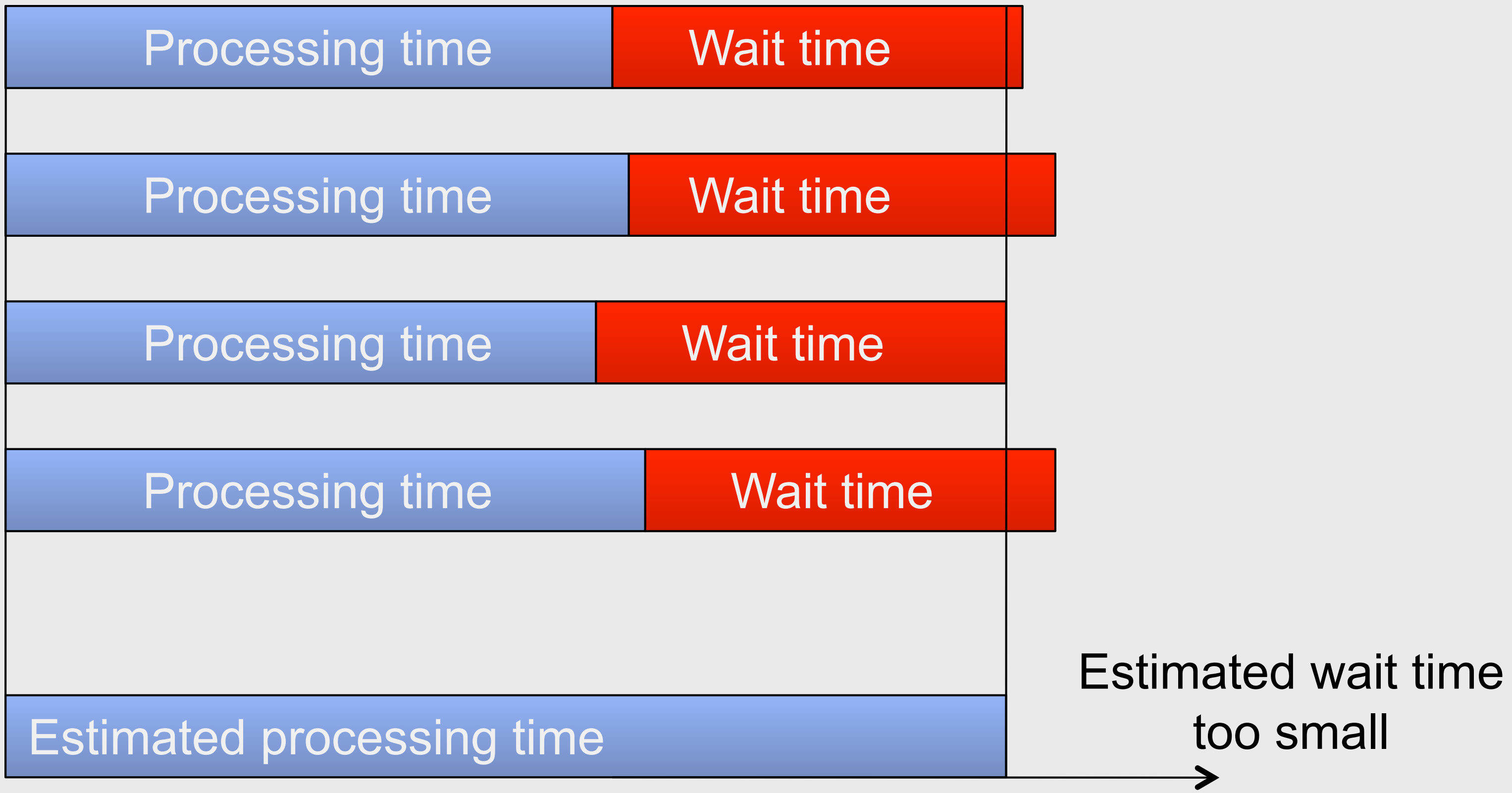
Jitter may cause a little higher wait time



Accuracy MPI_Waitall



Static imbalance



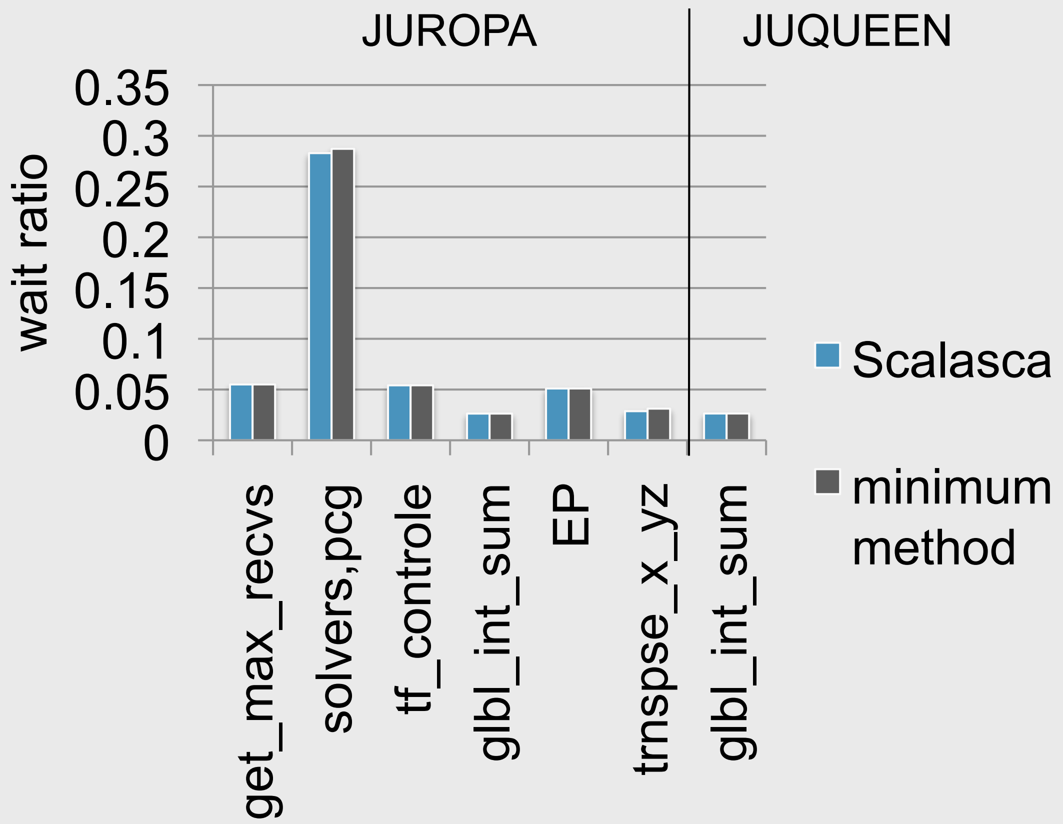
German Research School for Simulation Sciences

Static imbalance

- Calculating global minima could resolve process local static imbalances
  - Reduction operation after measurement
  - No dilation at measurement time

- Loose sender/receiver parameterization of minima
- For collective operations, global minima were better

Accuracy for Wait at NxN



Conclusion (1)

- Minimum method works for the estimation of blocking and non-blocking communication
  - For blocking communication results similar to Scalasca
  - For non-blocking communication, in Waitall wait time do not match the Scalasca analysis.

- Low runtime overhead
- No trace recording or piggybacking

- May not produce 100% accurate numbers, but
  - Sufficient accuracy to locate performance problems
  - Point to places where we might want to investigate further with trace analysis

German Research School
for Simulation Sciences

Conclusion (2)

- Detection of good minimum crucial
  - Static imbalance
  - Tradeoff between number of parameters and number of samples
- Jitter may lead to minor increase of measured wait state
- For non-blocking communication
  - Count possible overlap time
  - Might be larger than pure Late Sender time
  - Isn’t this even more accurate to estimate the optimization potential?

JÜLICH FORSCHUNGSZENTRUM RWTH AACHEN UNIVERSITY

German Research School for Simulation Sciences

Reference

Guoyong Mao, David Böhme, Marc-André Hermanns, Markus Geimer, Daniel Lorenz, Felix Wolf: *Catching Idlers With Ease: A Lightweight Wait-State Profiler for MPI Programs*. In: EuroMPI '14: Proc. Of the 21st European MPI Users' Group Meeting, Tokyo, Japan, Sep. 9-12, 2014