

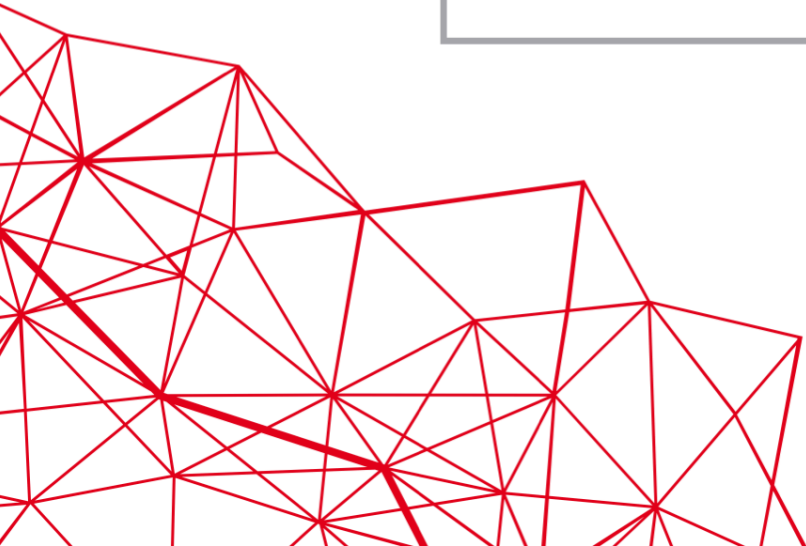
ISC

High Performance

SHAPING

TOMORROW

JUNE 24 - JULY 2, 2021 | [ISC-HPC.COM](https://isc-hpc.com)





SHAPING
TOMORROW



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

36th International Conference on High Performance Computing,
ISC High Performance 2021 Digital, June 28, 2021





SHAPING

TOMORROW



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG



AYESHA AFZAL^{1,2}, GEORG HAGER¹, GERHARD WELLEIN^{1,2}

¹ Erlangen National High Performance Computing Center

² Department of Computer Science,
University of Erlangen-Nürnberg Germany

36th International Conference on High Performance Computing,
ISC High Performance 2021 Digital, June 28, 2021





SHAPING

TOMORROW



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG



Noise-Driven Cluster-Level Performance Modelling and Engineering

AYESHA AFZAL^{1,2}, GEORG HAGER¹, GERHARD WELLEIN^{1,2}

¹ Erlangen National High Performance Computing Center

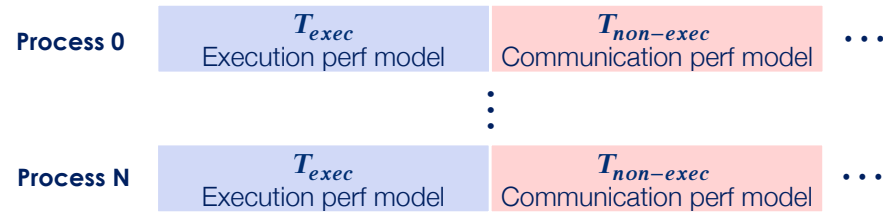
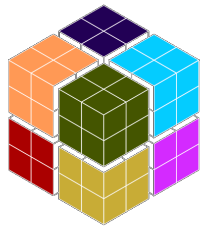
² Department of Computer Science,
University of Erlangen-Nürnberg Germany

36th International Conference on High Performance Computing,
ISC High Performance 2021 Digital, June 28, 2021

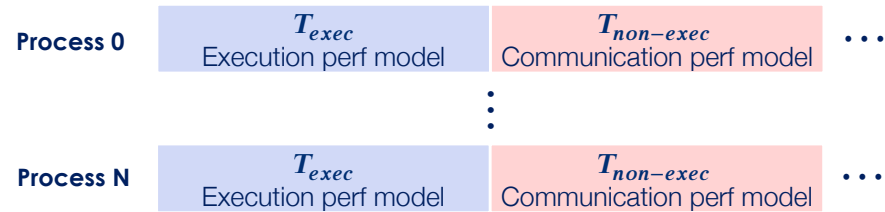
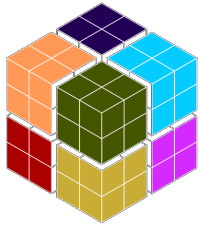


Vision: white-box first-principle performance modelling

Vision: white-box first-principle performance modelling



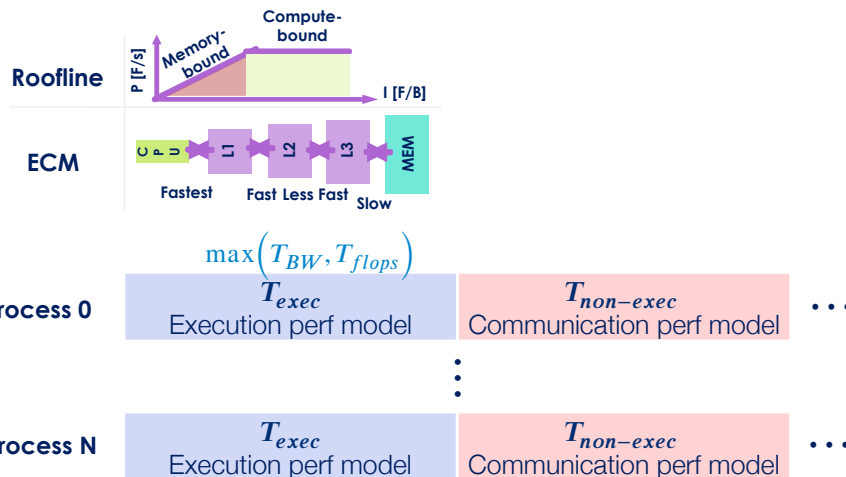
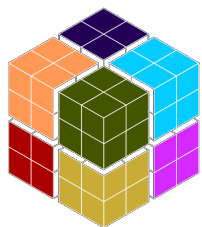
Vision: white-box first-principle performance modelling



$$T_{comp} = \text{sum}(T_{exec}, T_{non-exec})$$

Composite analytic performance model

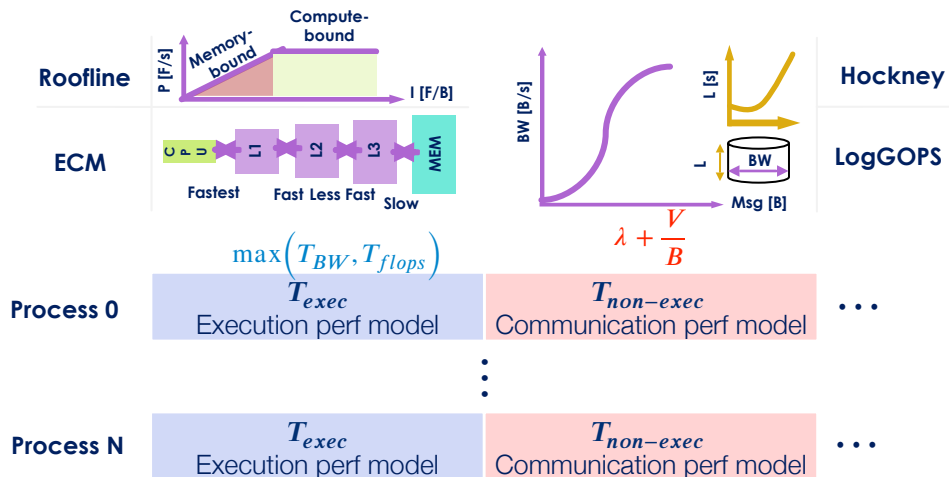
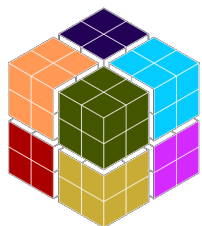
Vision: white-box first-principle performance modelling



$$T_{comp} = \text{sum}(T_{exec}, T_{non-exec})$$

Composite analytic performance model

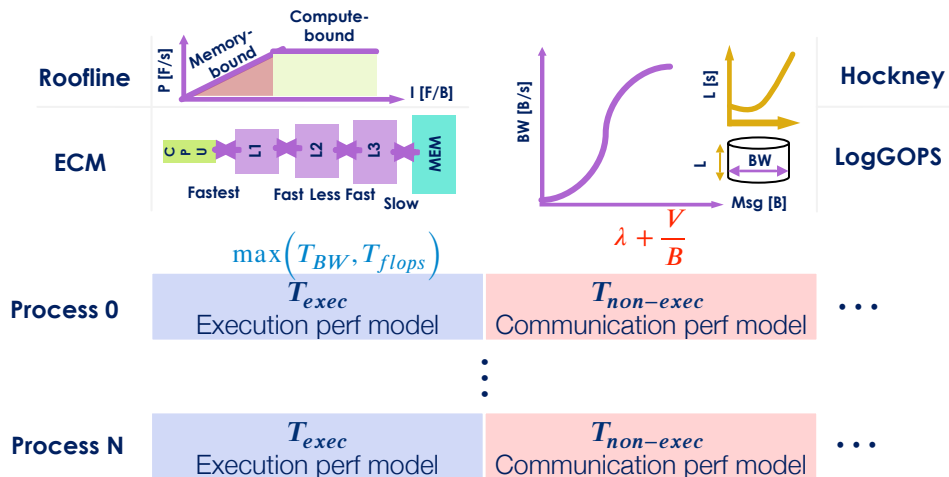
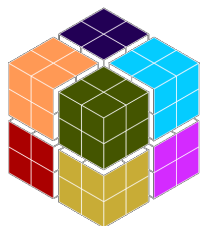
Vision: white-box first-principle performance modelling



$$T_{comp} = \text{sum}(T_{exec}, T_{non-exec})$$

Composite analytic performance model

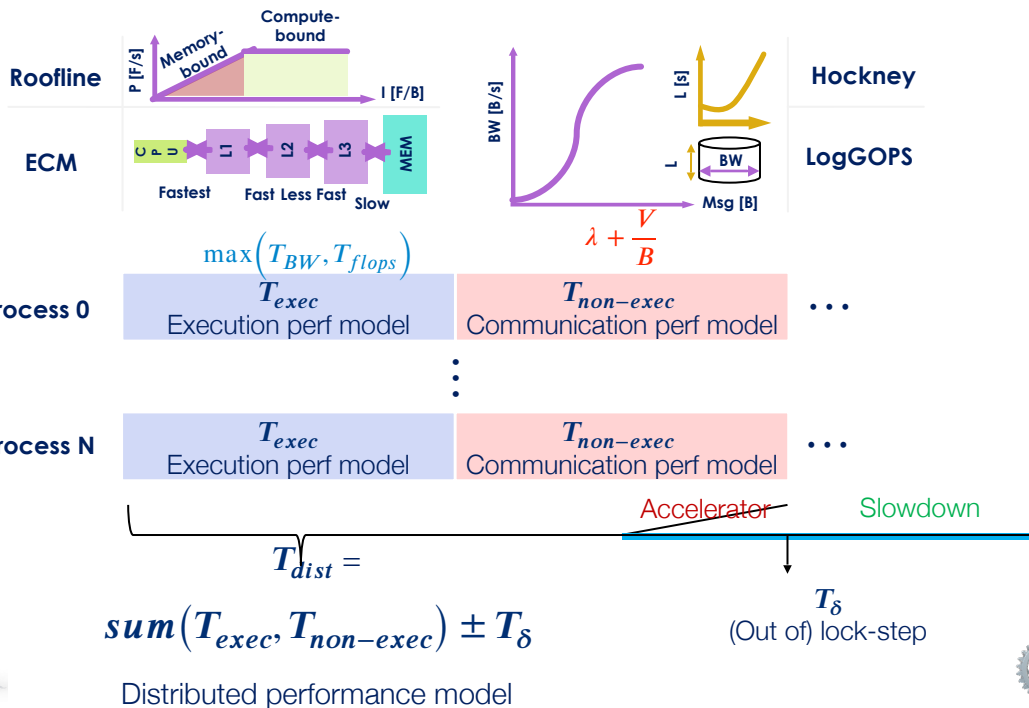
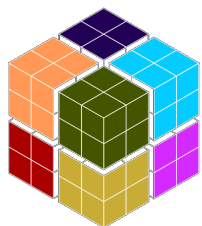
Vision: white-box first-principle performance modelling



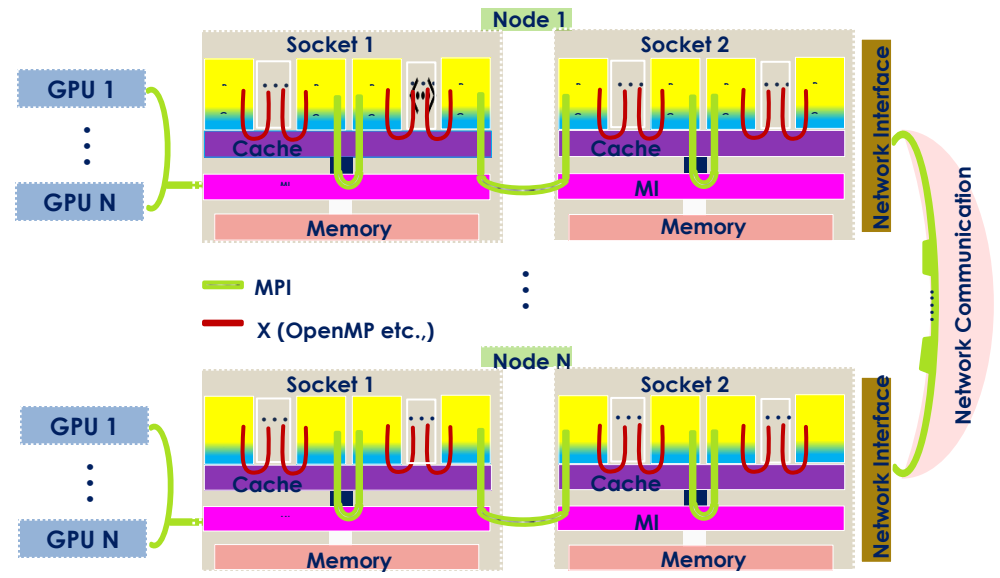
$$T_{comp} = \text{sum}(T_{exec}, T_{non-exec})$$

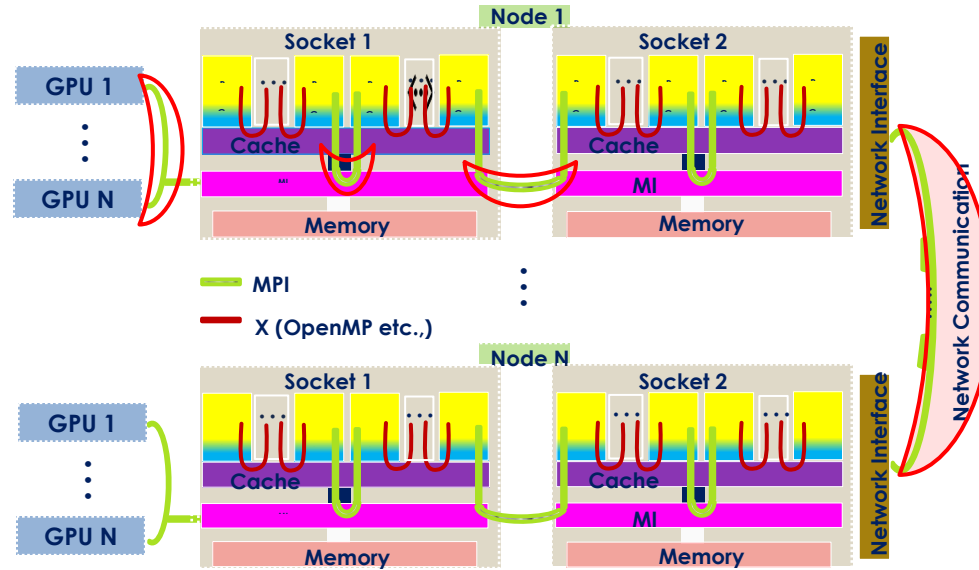
Composite analytic performance model

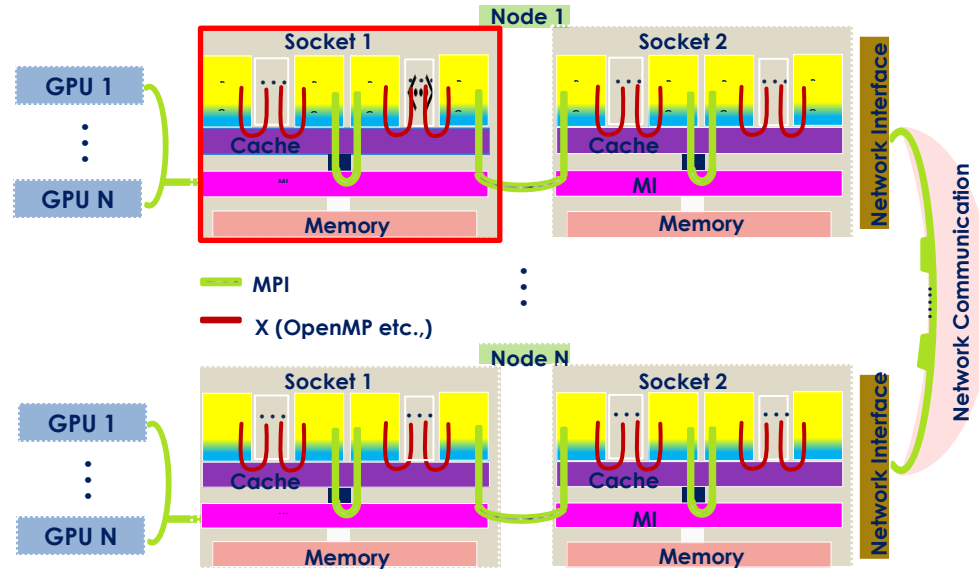
Vision: white-box first-principle performance modelling

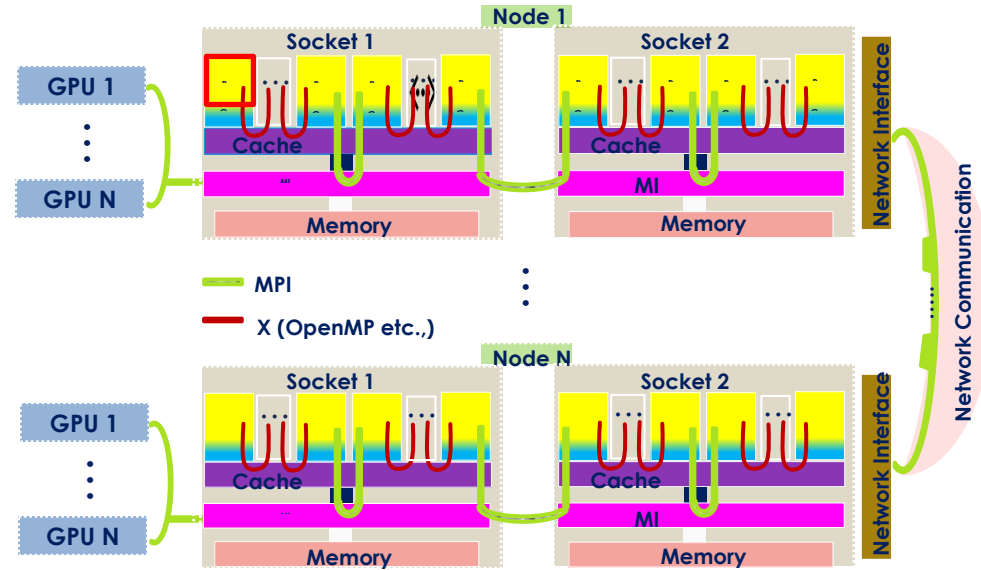


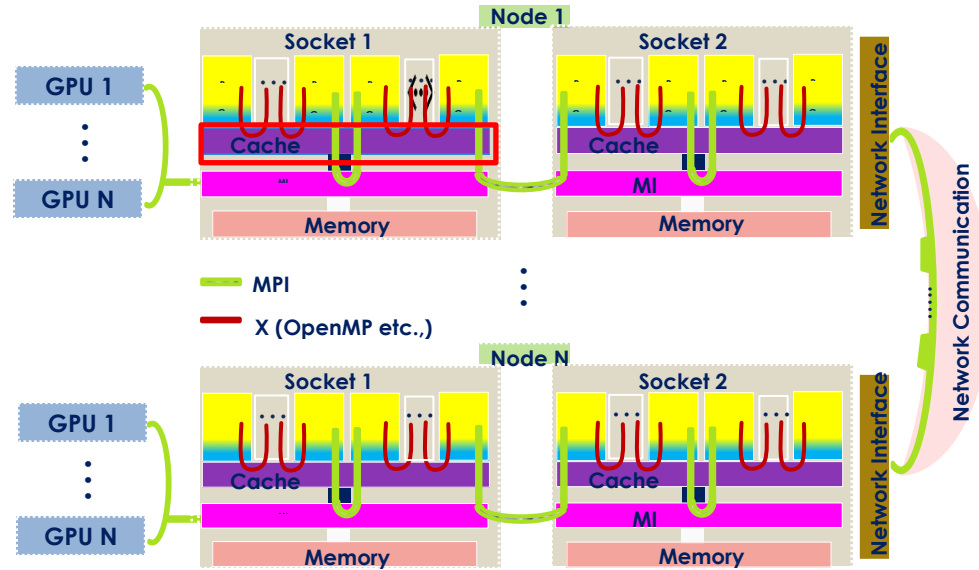
Problem: noise can cause acceleration, speed down, or neither

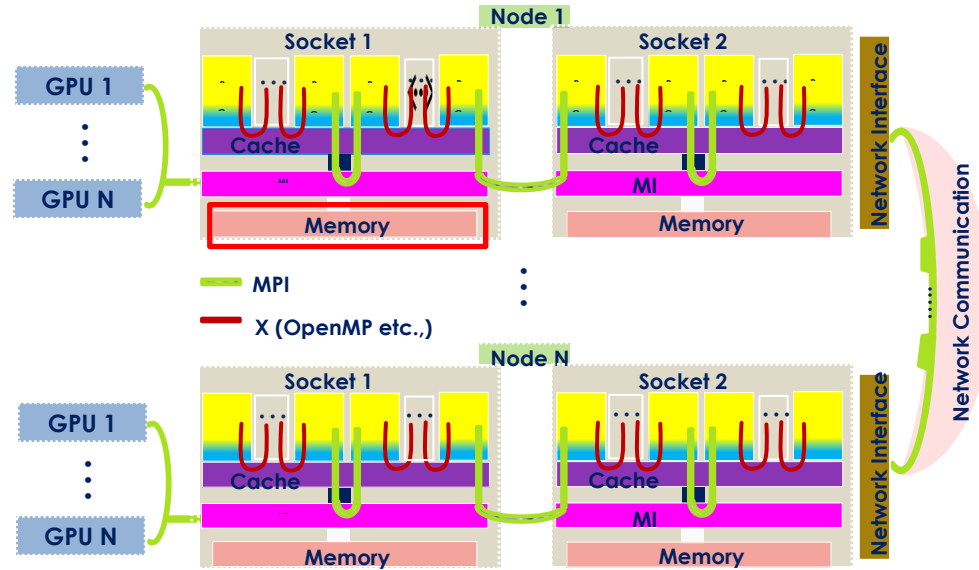




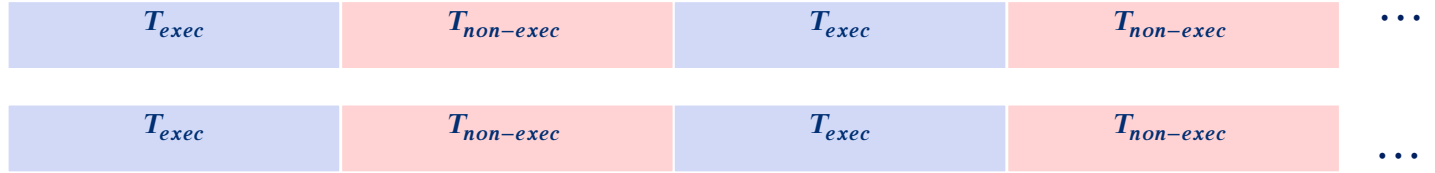




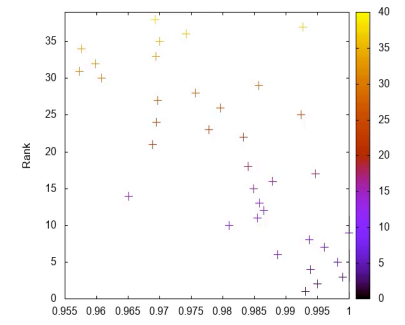




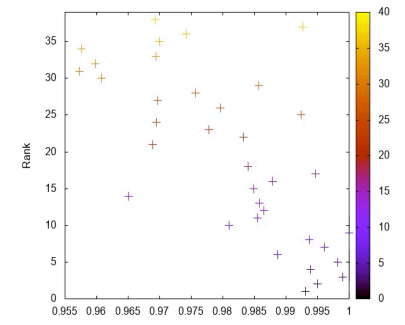
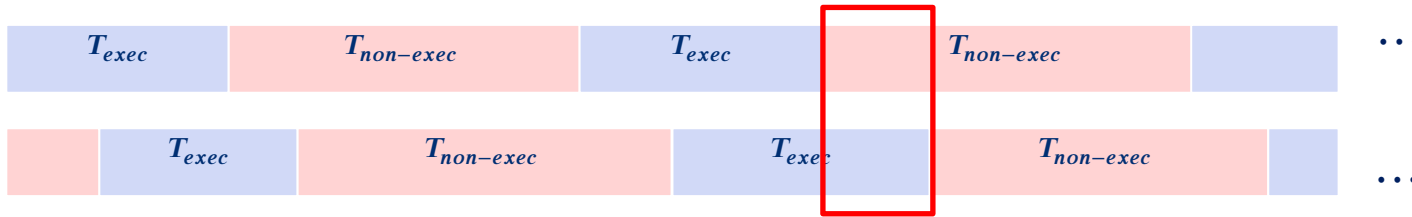
Memory-bound
parallel
programs



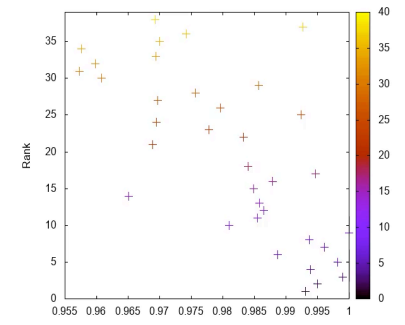
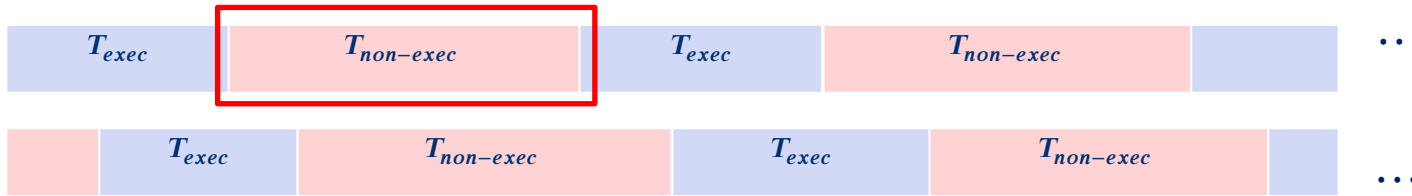
Parallel programs out of lock-step



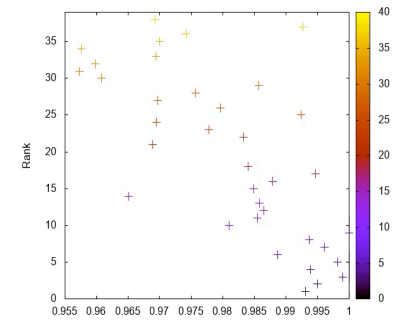
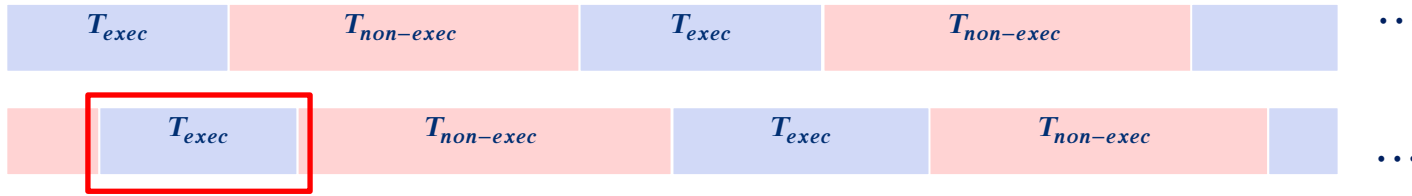
Parallel programs out of lock-step



Parallel programs out of lock-step



Parallel programs out of lock-step

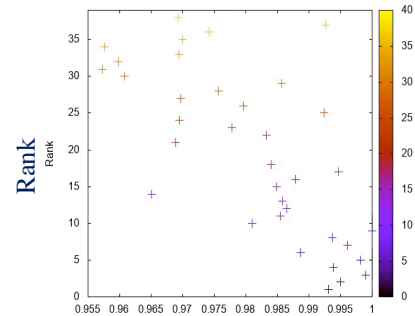


Parallel programs out of lock-step

4 Emmy sockets @2.2 GHz, non-temporal stores,
bi-dir, open chain, distance-1 communication



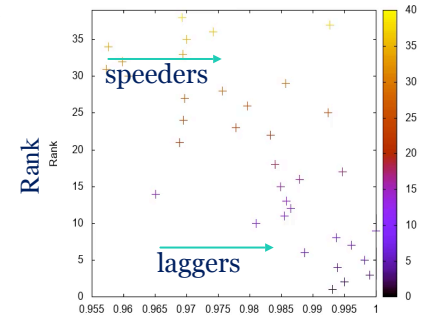
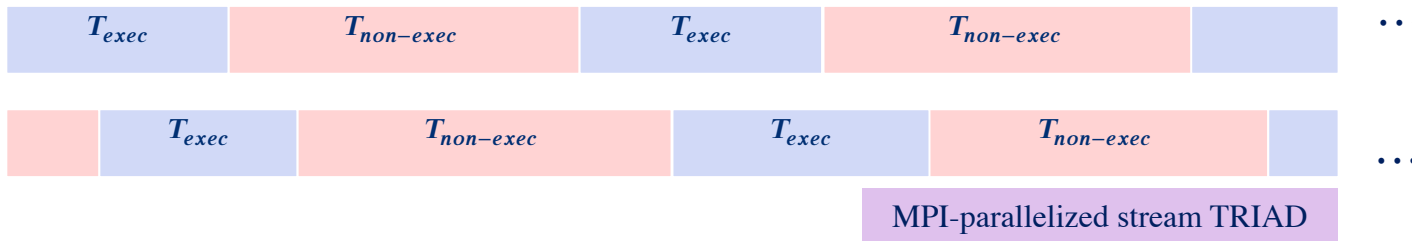
MPI-parallelized stream TRIAD



Wall-clock time normalized to slowest process

Parallel programs out of lock-step

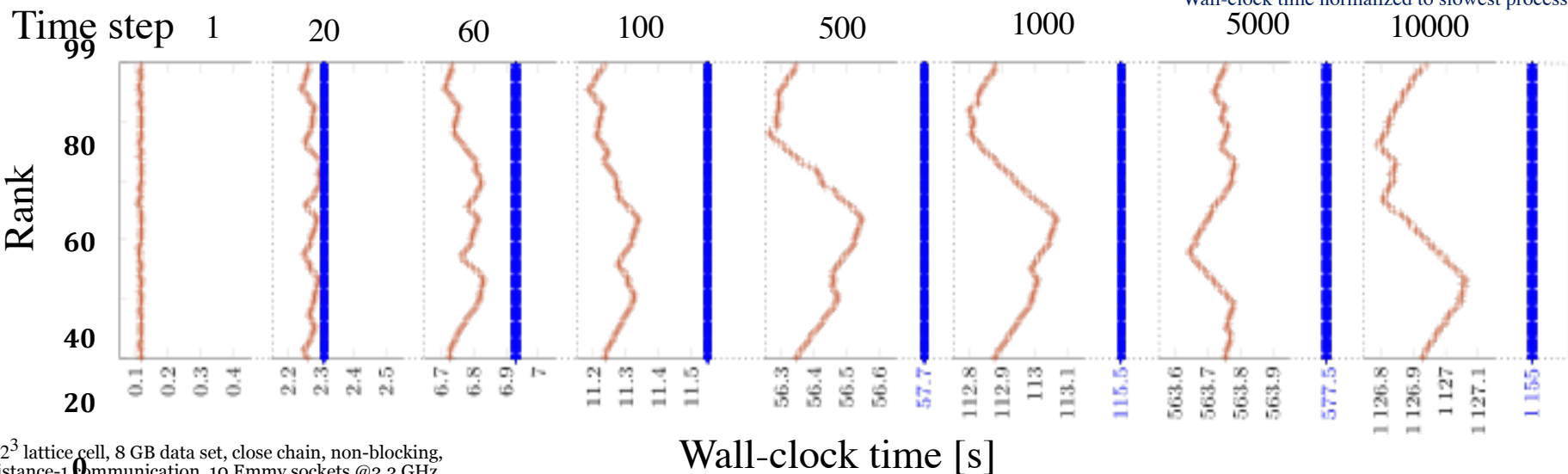
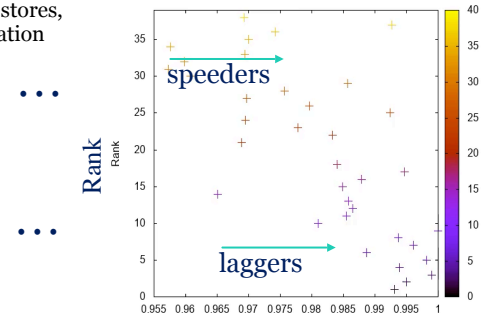
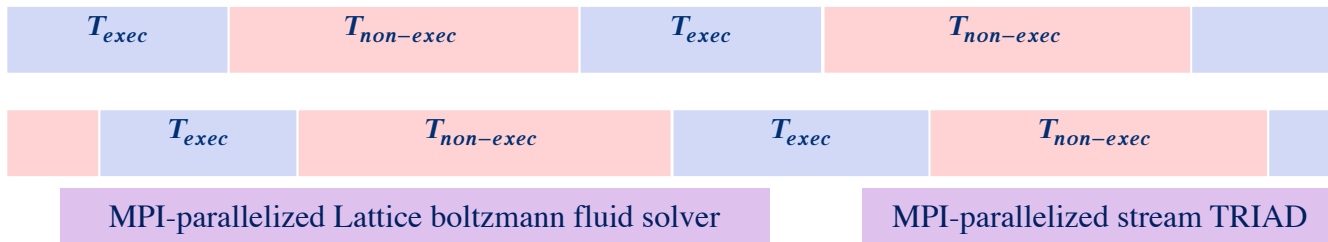
4 Emmy sockets @2.2 GHz, non-temporal stores,
bi-dir, open chain, distance-1 communication



Wall-clock time normalized to slowest process

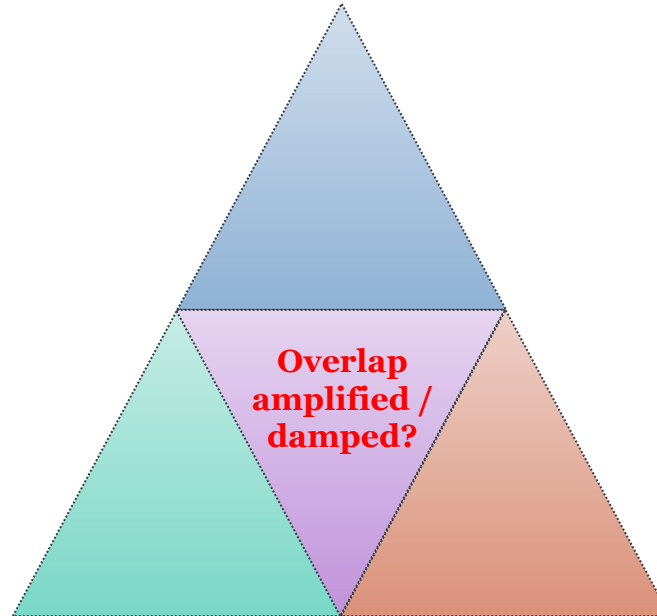
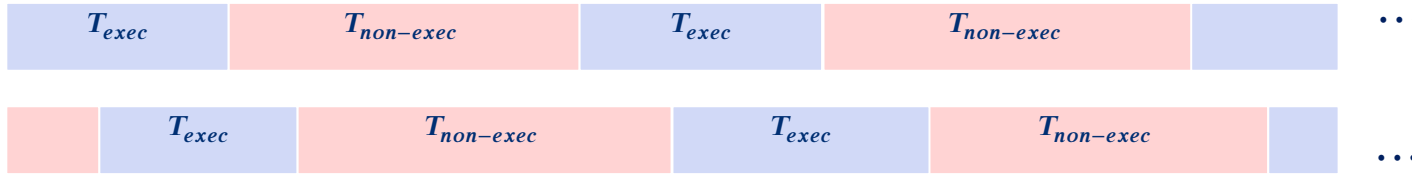
Parallel programs out of lock-step

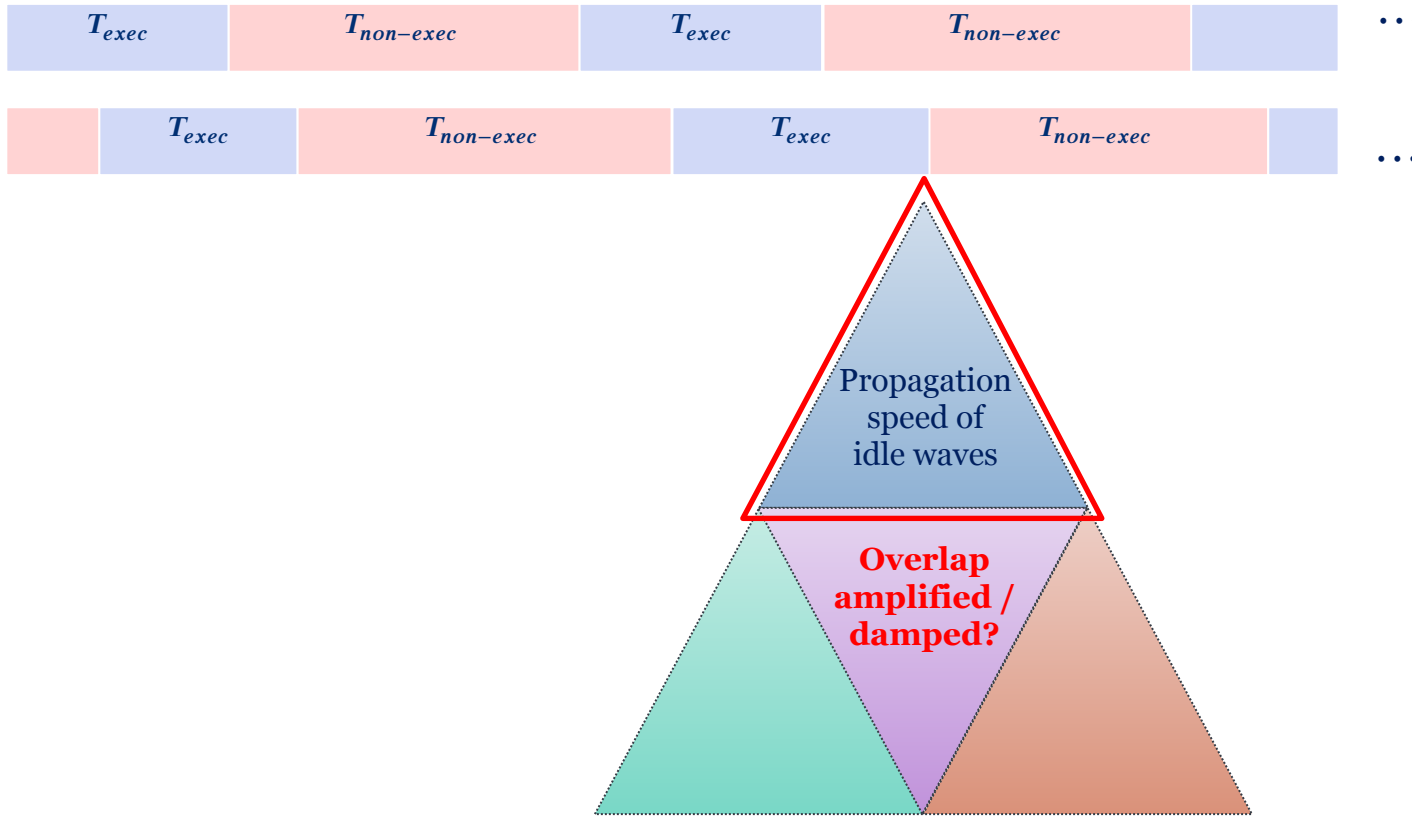
4 Emmy sockets @2.2 GHz, non-temporal stores,
bi-dir, open chain, distance-1 communication

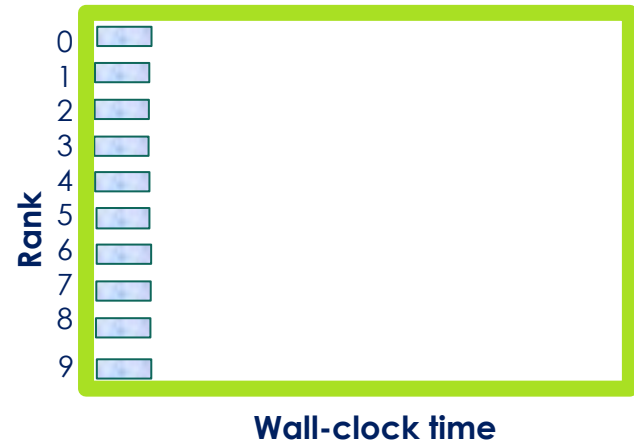


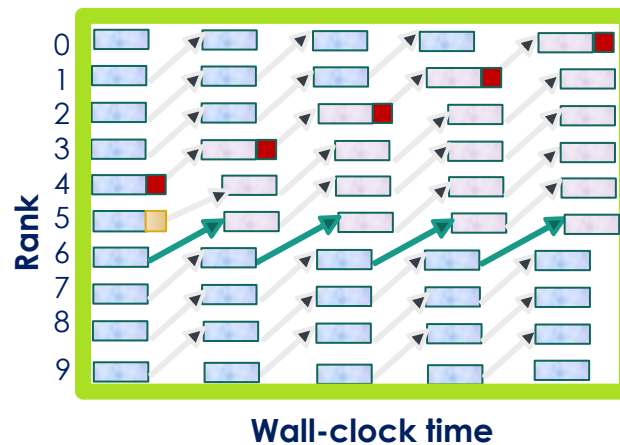
302^3 lattice cell, 8 GB data set, close chain, non-blocking,
distance-1 communication, 10 Emmy sockets @2.2 GHz

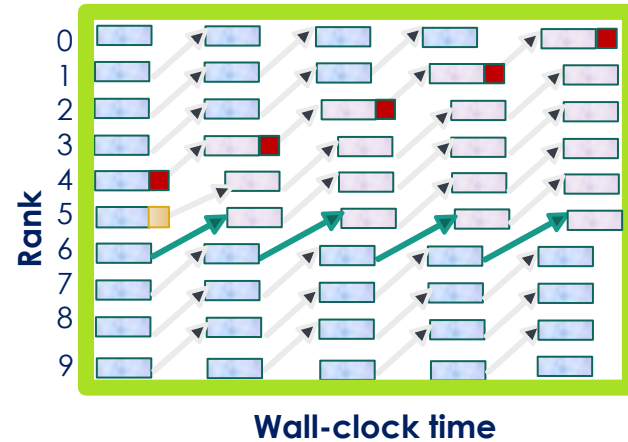
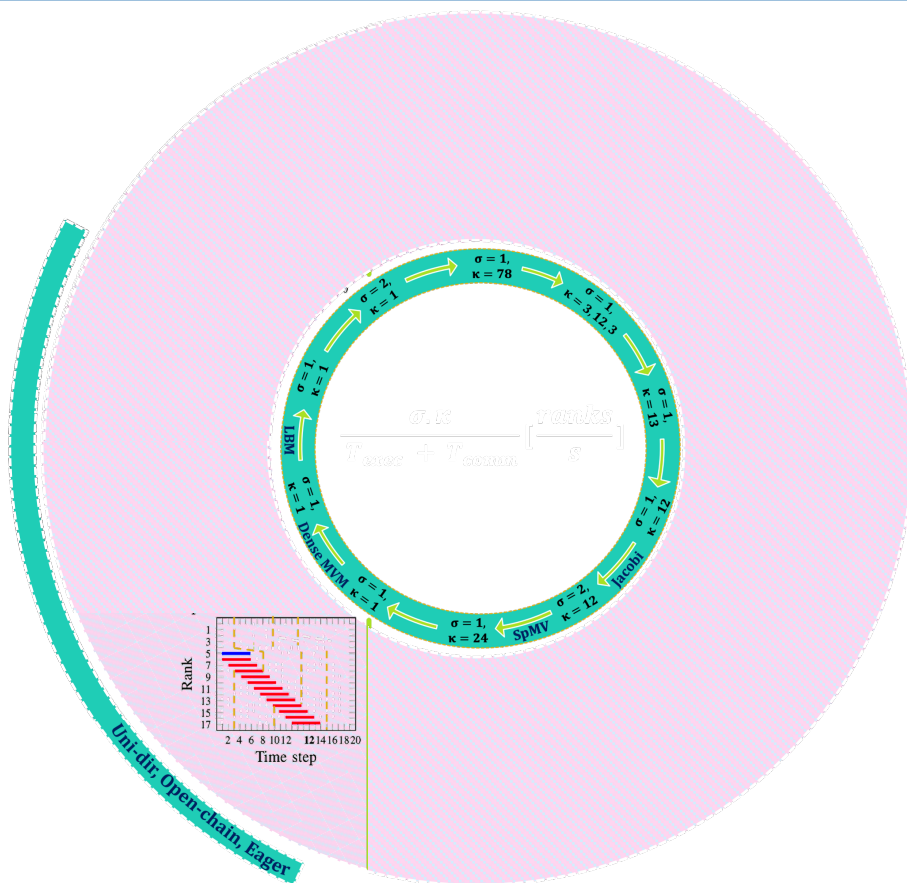
Parallel programs out of lock-step

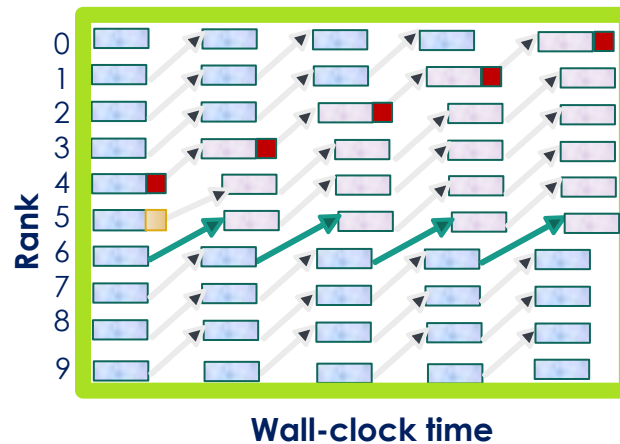
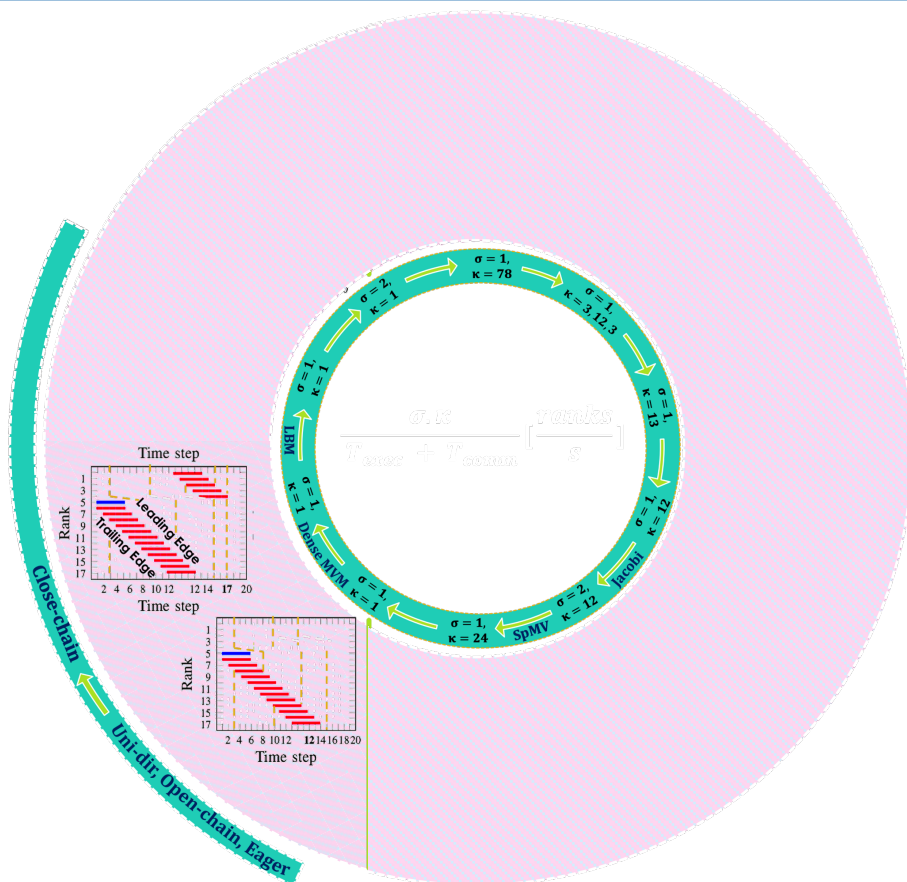


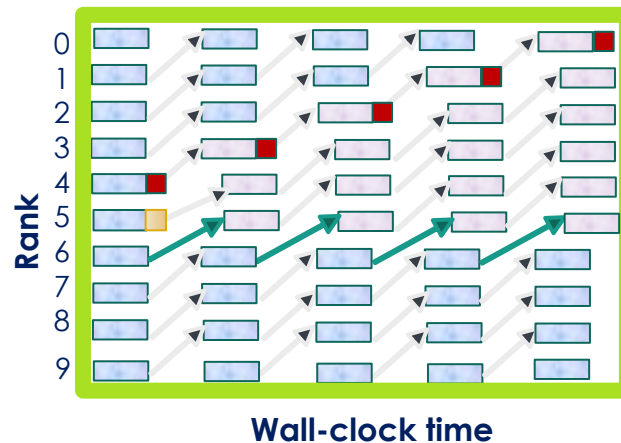
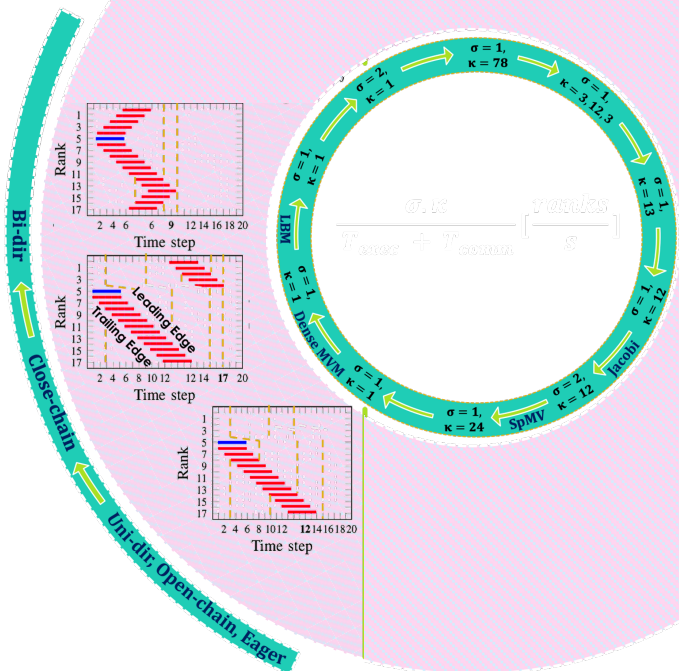




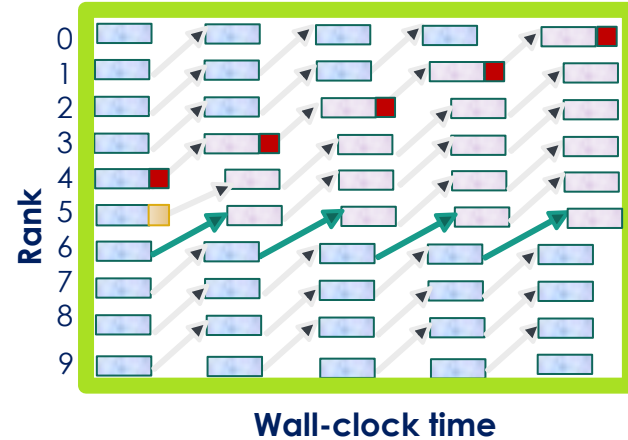
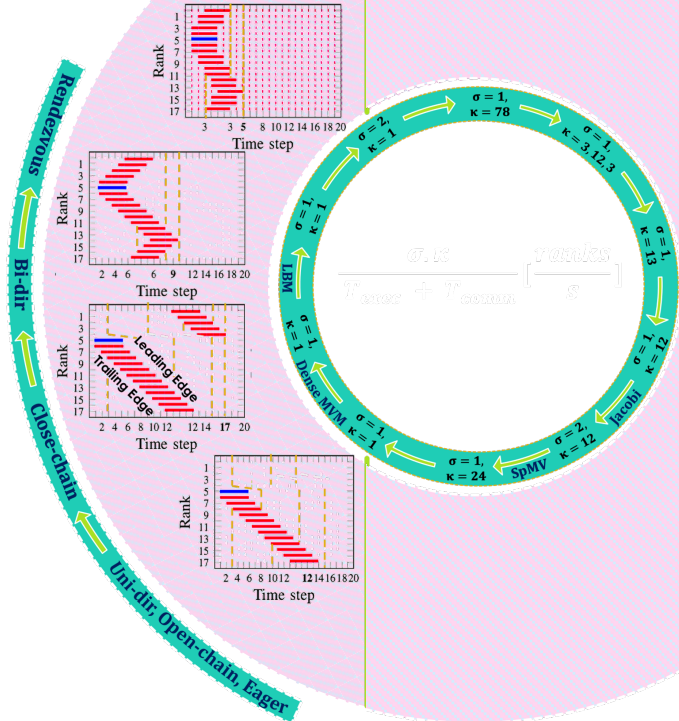


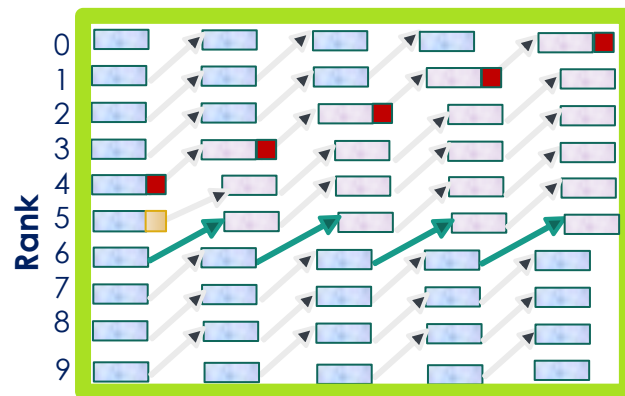
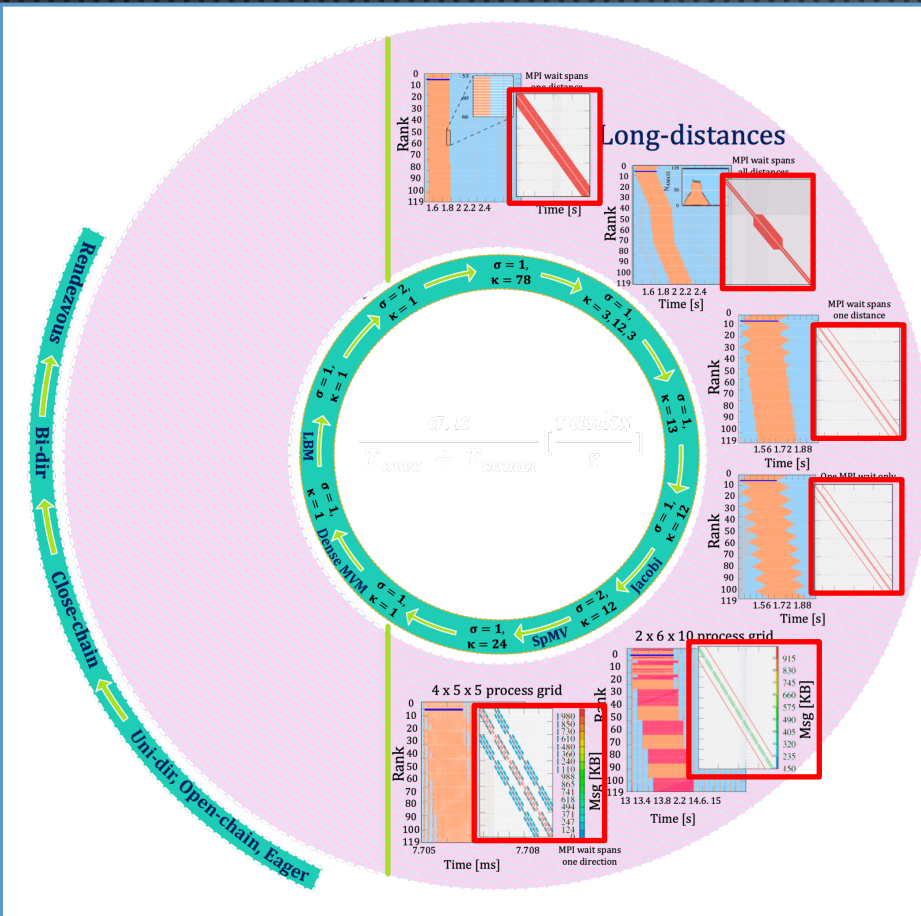




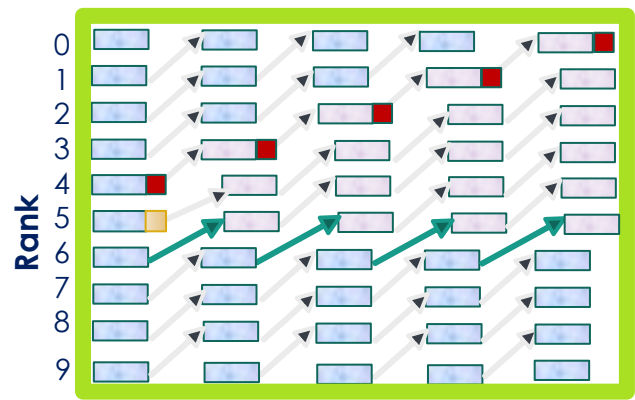
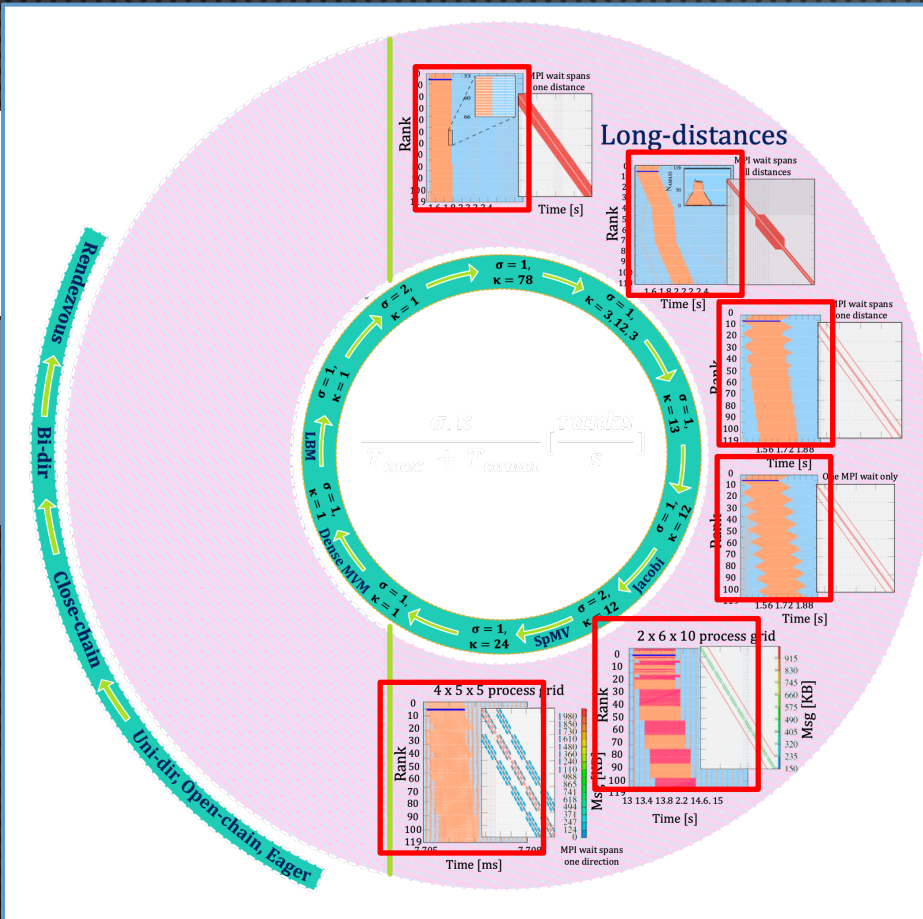


Distance-1

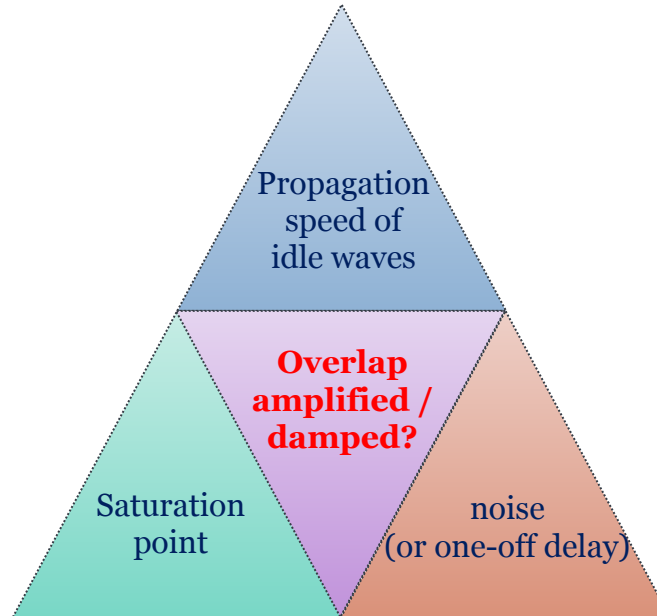
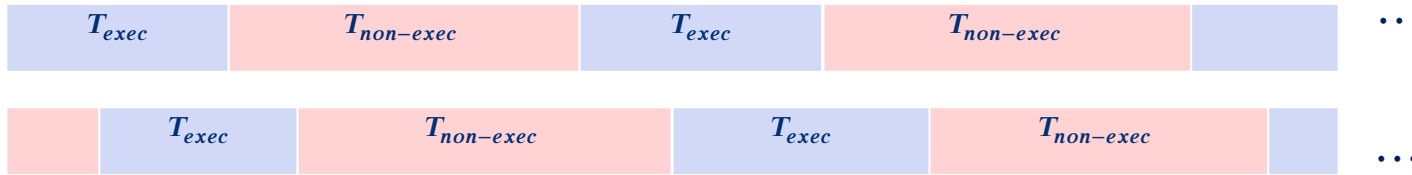


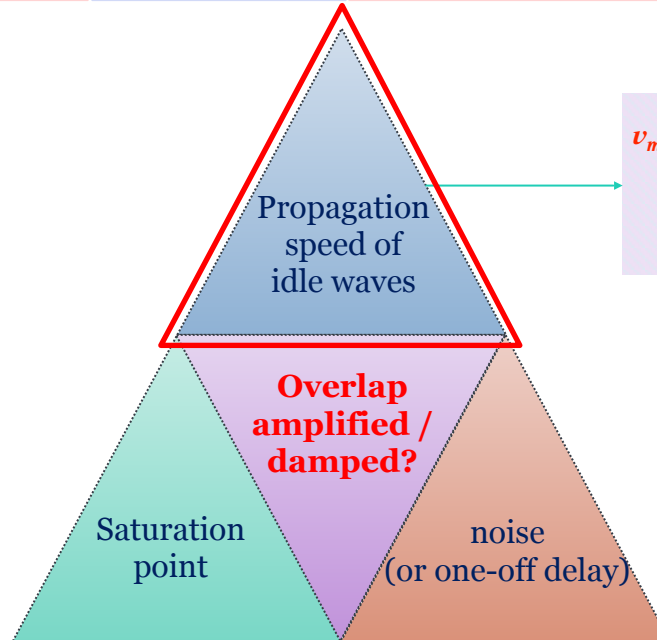
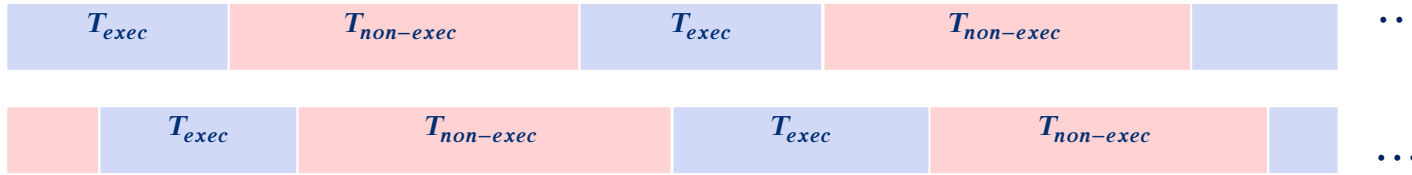


Wall-clock time



Wall-clock time

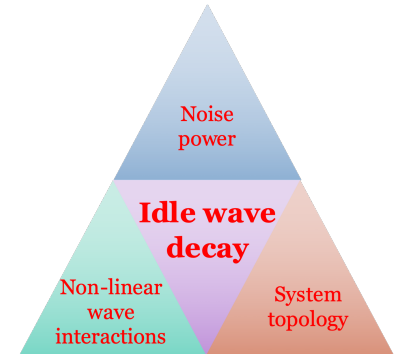


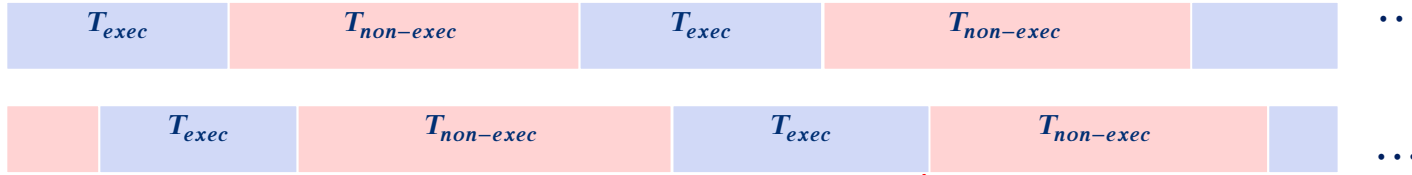


$$v_{min} = \kappa \times \sigma \times 1 \left[\frac{rank}{iter} \right] \times \frac{1}{T_{comp} + T_{comm}} \left[\frac{iter}{s} \right]$$

$$\kappa = \frac{j(j+1)}{2} \text{ or } (j + [i]) \text{ or } j$$

j/i: longest /shorter-distance partner





Afzal et al.

IEEE Cluster 2019

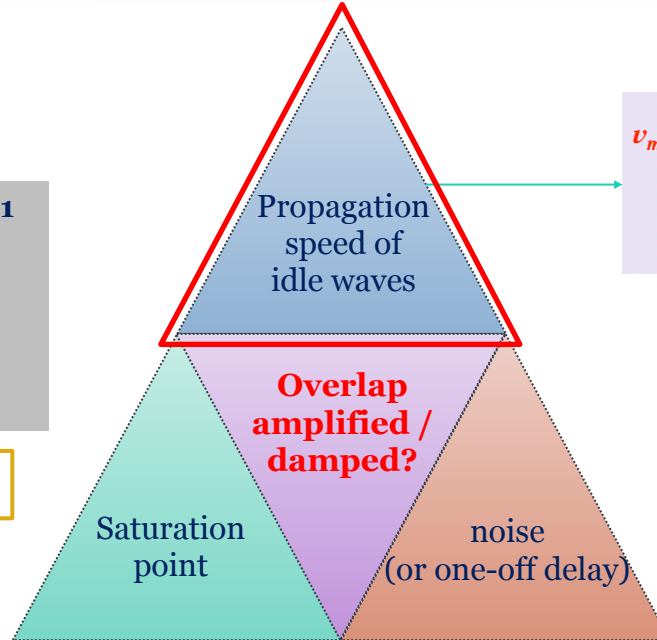


DOI:10.1109/
CLUSTER.
2019.8890995

ISC HPC 2021



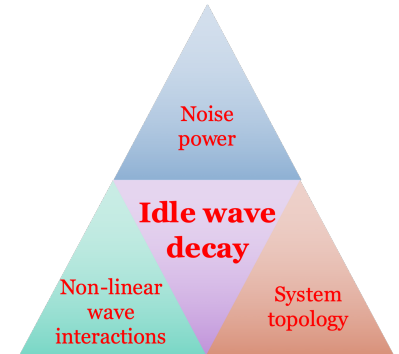
arXiv: 2103.03175
cs.DC]



$$v_{min} = \kappa \times \sigma \times 1 \left[\frac{rank}{iter} \right] \times \frac{1}{T_{comp} + T_{comm}} \left[\frac{iter}{s} \right]$$

$$\kappa = \frac{j(j+1)}{2} \text{ or } (j + [i]) \text{ or } j$$

j/i : longest /shorter-distance partner



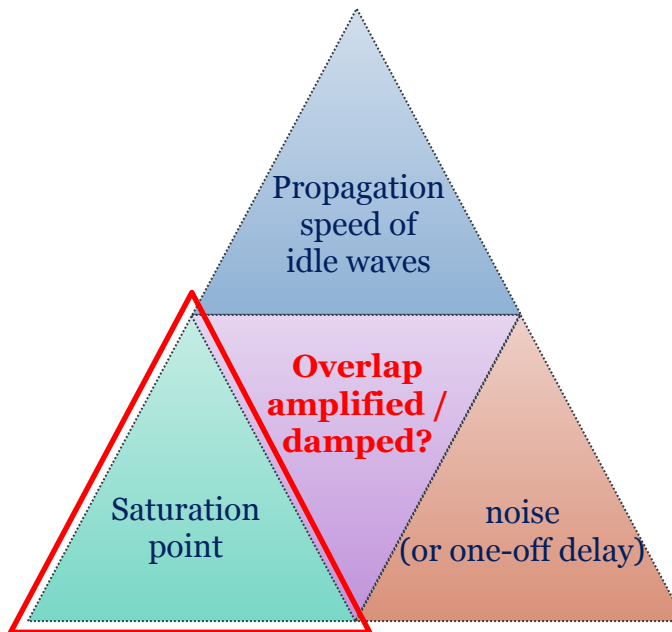


Afzal et al.

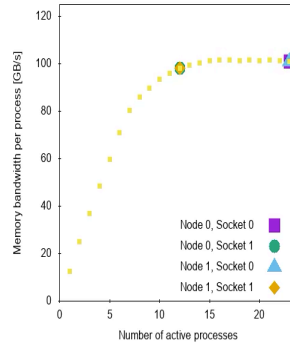
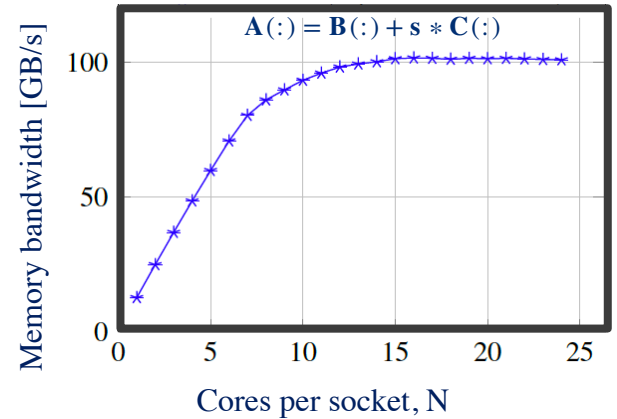
ISC HPC 2020



DOI:
10.1007/978-3-030-5
0743-5_20



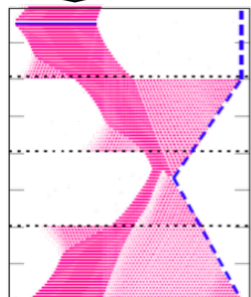
SuperMUC-NG @2.3 GHz, non-temporal stores,
bi-dir, 1024 B, close chain, distance-1 communication



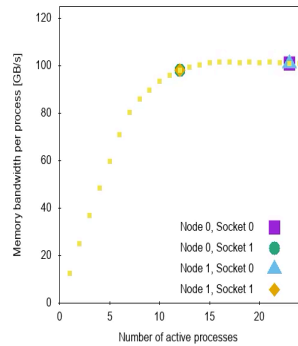
Rank
11
34
59
83

N=24
(4 sockets)

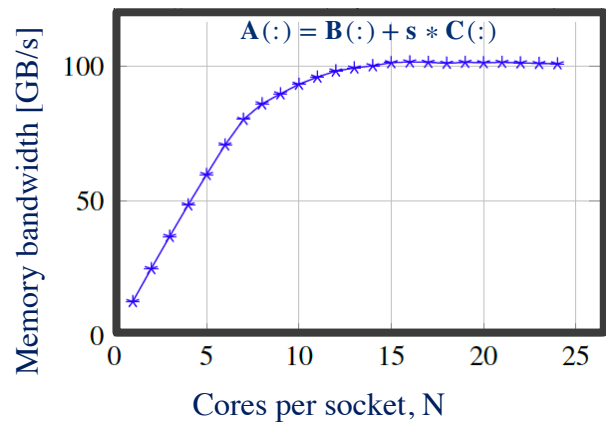
$$A(:) = B(:) + s * C(:)$$



Time step 50



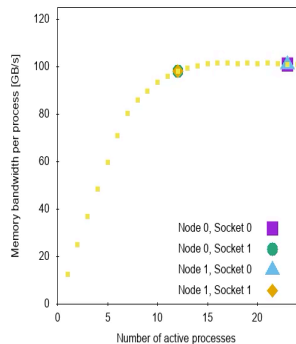
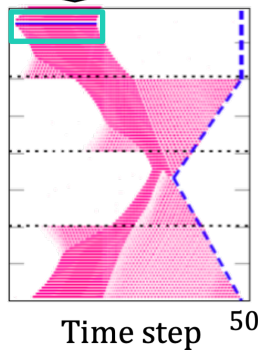
SuperMUC-NG @2.3 GHz, non-temporal stores,
bi-dir, 1024 B, close chain, distance-1 communication



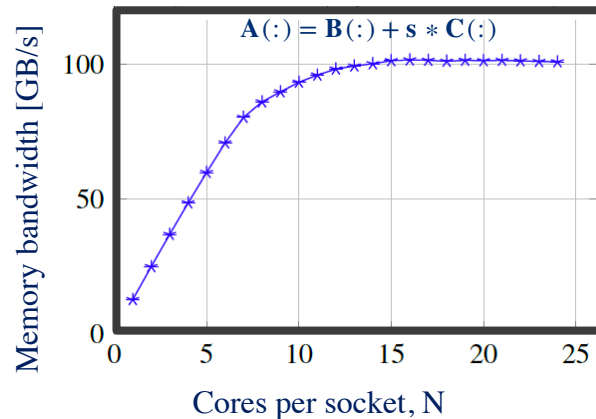
Rank
11
34
59
83

N=24
(4 sockets)

$$A(:) = B(:) + s * C(:)$$



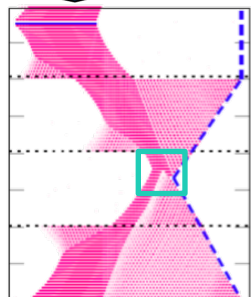
SuperMUC-NG @2.3 GHz, non-temporal stores,
bi-dir, 1024 B, close chain, distance-1 communication



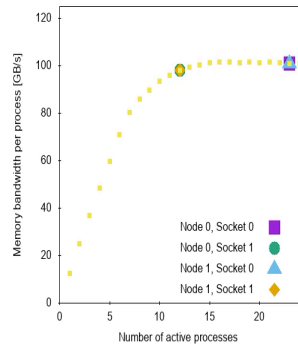
Rank
11
34
59
83

N=24
(4 sockets)

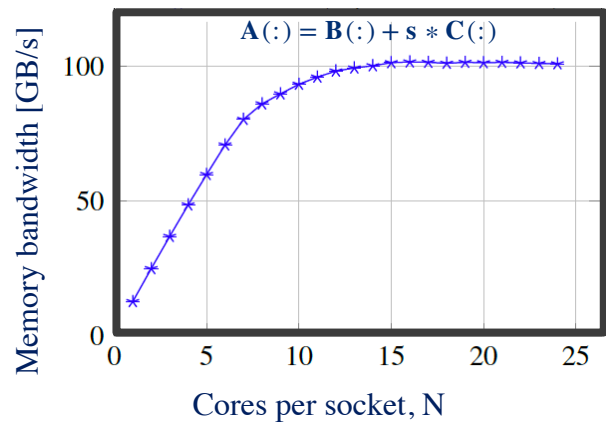
$$A(:) = B(:) + s * C(:)$$



Time step 50



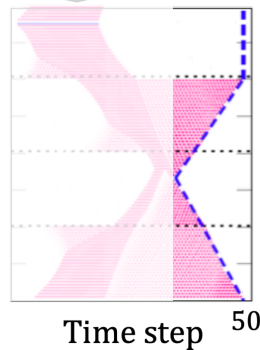
SuperMUC-NG @2.3 GHz, non-temporal stores, bi-dir, 1024 B, close chain, distance-1 communication



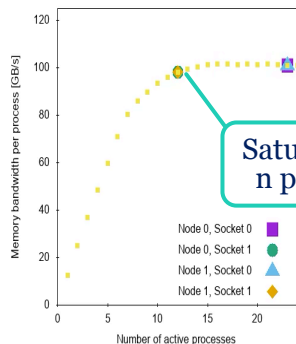
Rank
11
34
59
83

N=24
(4 sockets)

$$A(:) = B(:) + s * C(:)$$

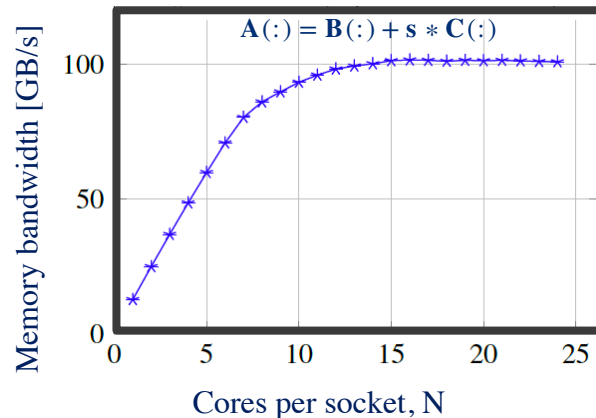


Memory bandwidth [GB/s]



Number of concurrently active processes

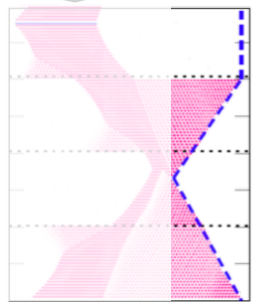
SuperMUC-NG @2.3 GHz, non-temporal stores,
bi-dir, 1024 B, close chain, distance-1 communication



Rank
11
34
59
83

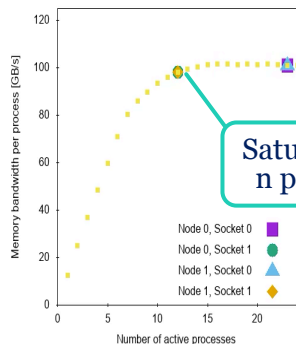
N=24
(4 sockets)

$$A(\cdot) = B(\cdot) + s * C(\cdot)$$



Time step 50

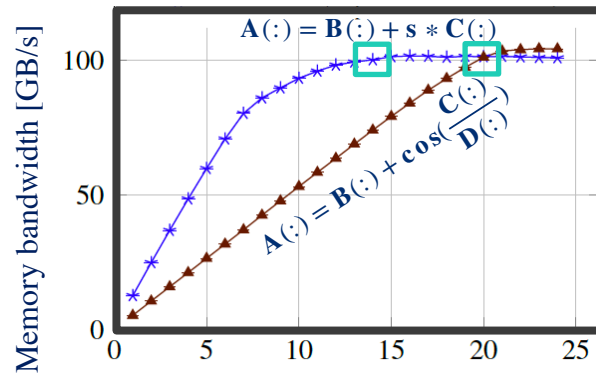
Memory bandwidth [GB/s]



Saturation point

Number of concurrently active processes

SuperMUC-NG @2.3 GHz, non-temporal stores, bi-dir, 1024 B, close chain, distance-1 communication

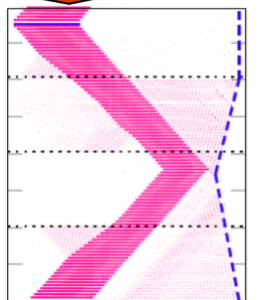


Cores per socket, N

Rank
11
34
59
83

N=24
(4 sockets)

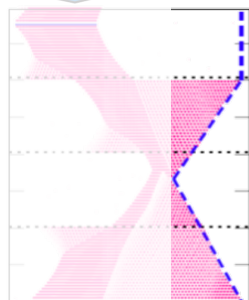
$$A(\cdot) = B(\cdot) + \cos\left(\frac{C(\cdot)}{D(\cdot)}\right)$$



Time step 50

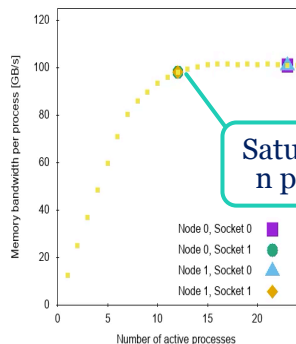
N=24
(4 sockets)

$$A(\cdot) = B(\cdot) + s * C(\cdot)$$



Time step 50

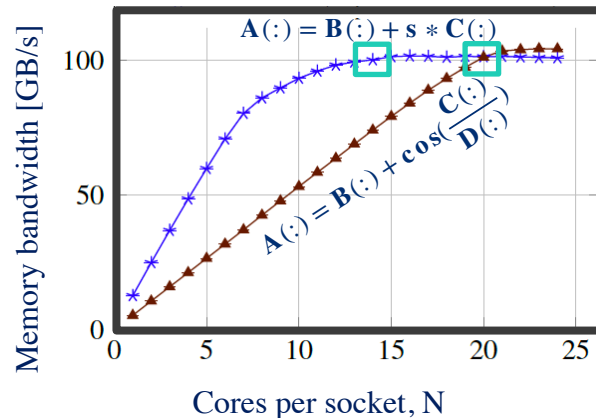
Memory bandwidth [GB/s]

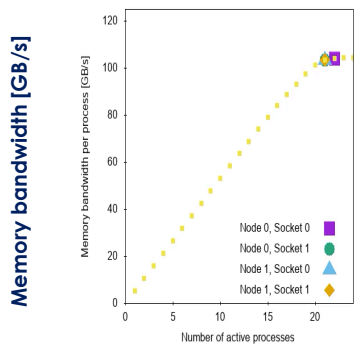
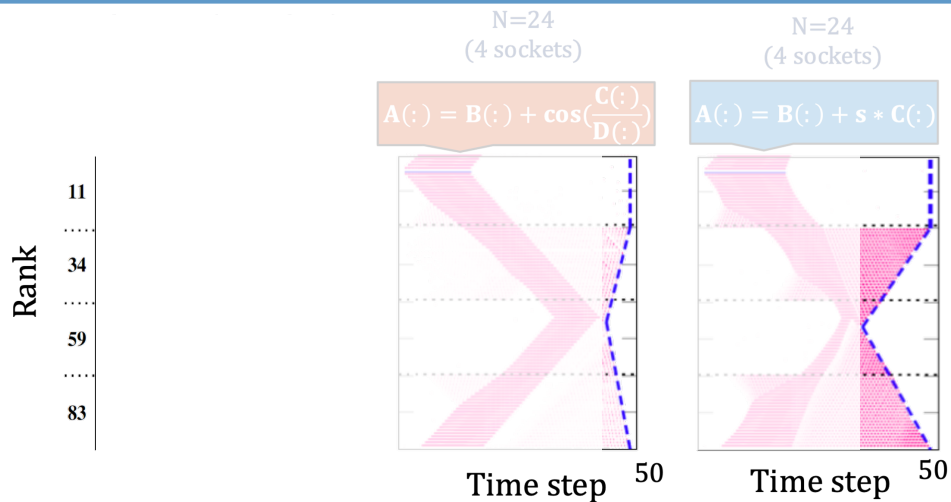


Saturation point

Number of concurrently active processes

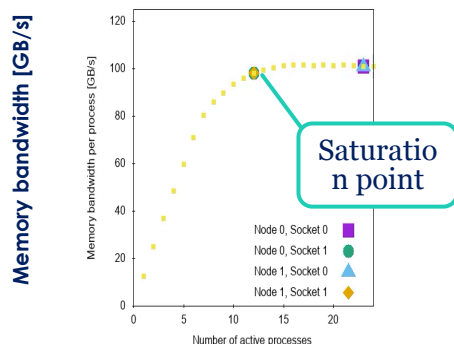
SuperMUC-NG @2.3 GHz, non-temporal stores, bi-dir, 1024 B, close chain, distance-1 communication





Saturation point

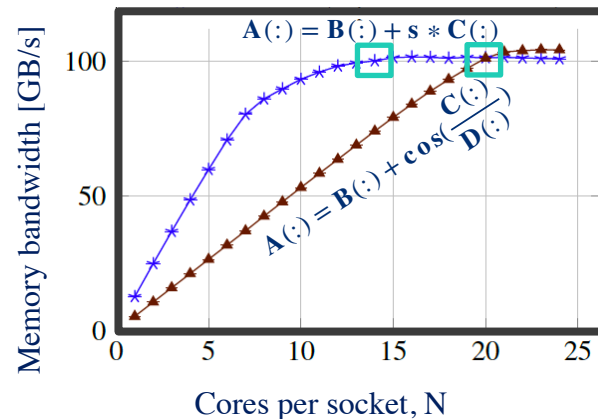
Number of concurrently active processes

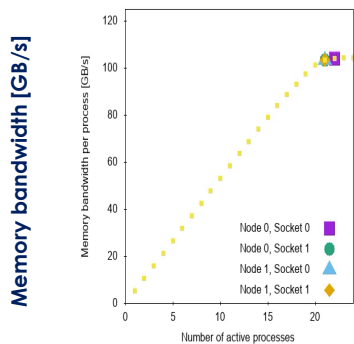
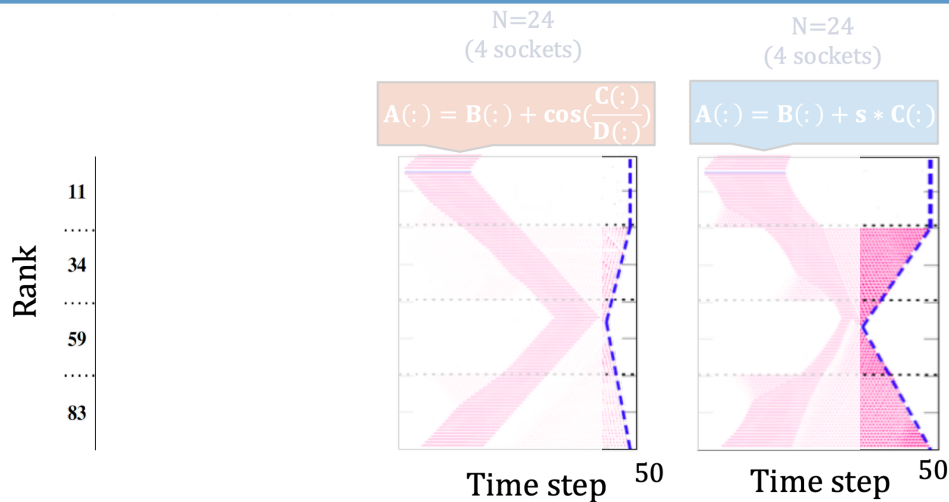


Saturation point

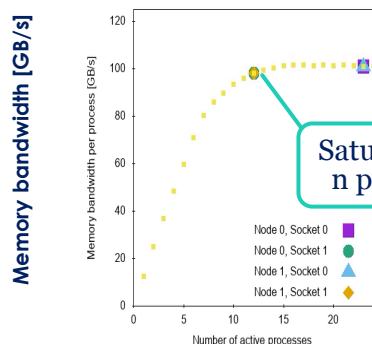
Number of concurrently active processes

SuperMUC-NG @2.3 GHz, non-temporal stores, bi-dir, 1024 B, close chain, distance-1 communication





Saturation point

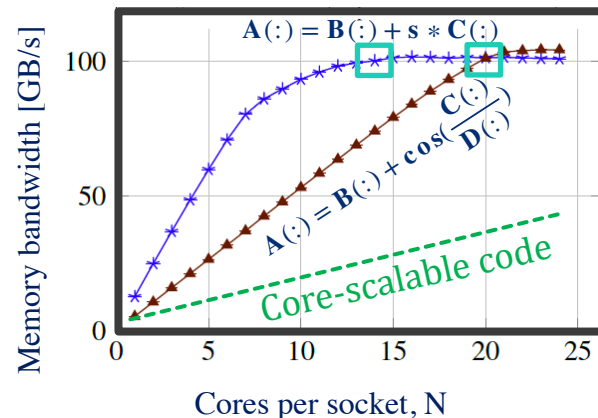


Saturation point

Number of concurrently active processes

Number of concurrently active processes

SuperMUC-NG @2.3 GHz, non-temporal stores, bi-dir, 1024 B, close chain, distance-1 communication



Scalable versus contented processes

Any code @N=1 (96 sockets)
Core-scalable code@ any N

N=24
(4 sockets)

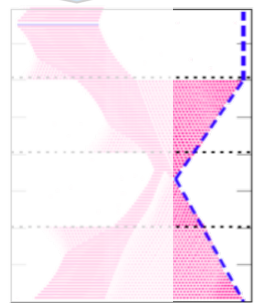
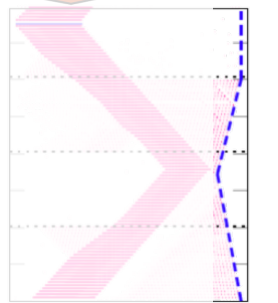
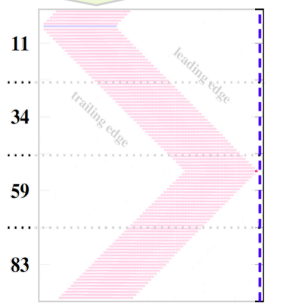
N=24
(4 sockets)

Process-scalable code

$$A(\cdot) = B(\cdot) + \cos\left(\frac{C(\cdot)}{D(\cdot)}\right)$$

$$A(\cdot) = B(\cdot) + s * C(\cdot)$$

Rank

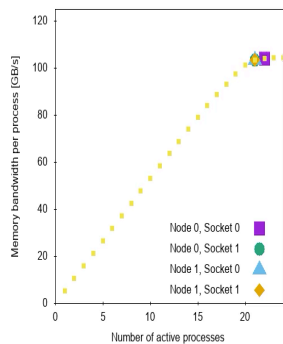


Time step 50

Time step 50

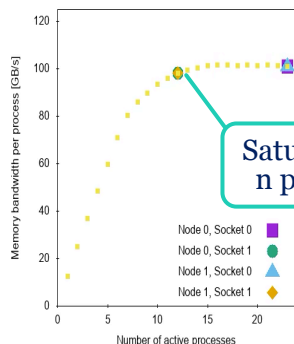
Time step 50

Memory bandwidth [GB/s]



Saturation point

Memory bandwidth [GB/s]

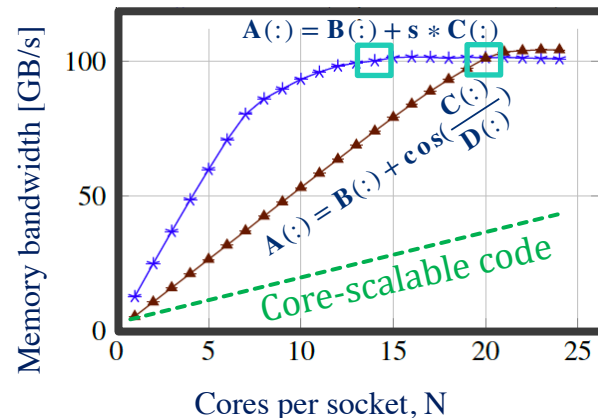


Saturation point

Number of concurrently active processes

Number of concurrently active processes

SuperMUC-NG @2.3 GHz, non-temporal stores,
bi-dir, 1024 B, close chain, distance-1 communication



Scalable versus contented processes

Any code @N=1 (96 sockets)
Core-scalable code@ any N

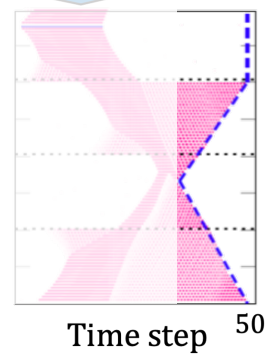
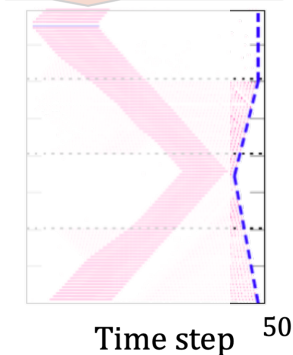
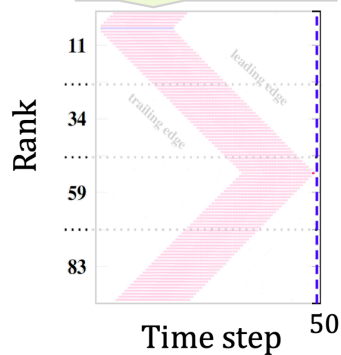
N=24
(4 sockets)

N=24
(4 sockets)

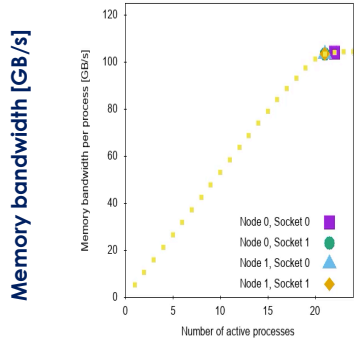
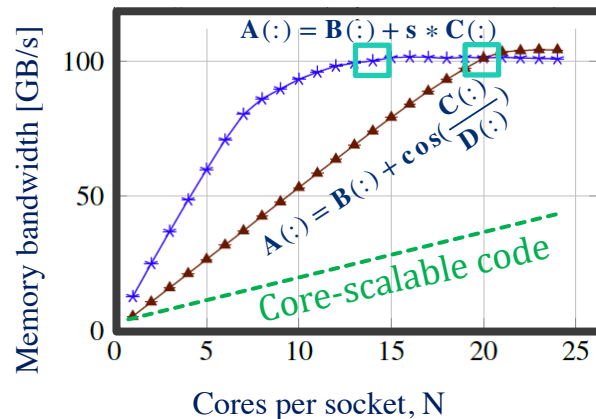
Process-scalable code

$$A(\cdot) = B(\cdot) + \cos\left(\frac{C(\cdot)}{D(\cdot)}\right)$$

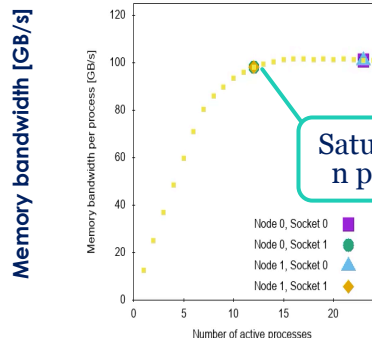
$$A(\cdot) = B(\cdot) + s * C(\cdot)$$



SuperMUC-NG @2.3 GHz, non-temporal stores,
bi-dir, 1024 B, close chain, distance-1 communication



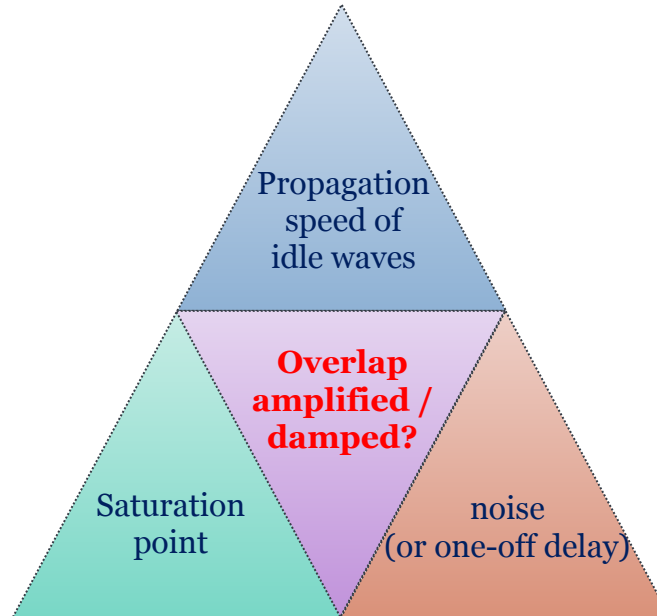
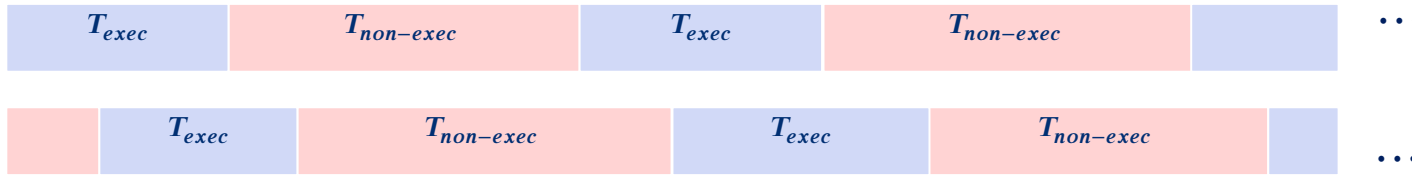
Saturatio
n point

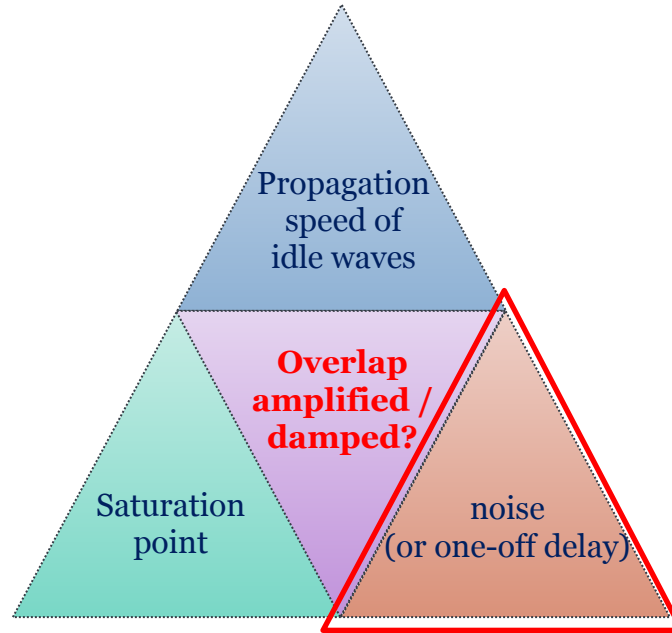


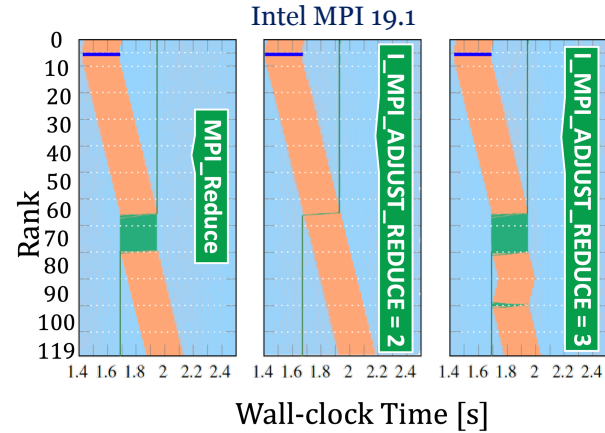
Saturatio
n point

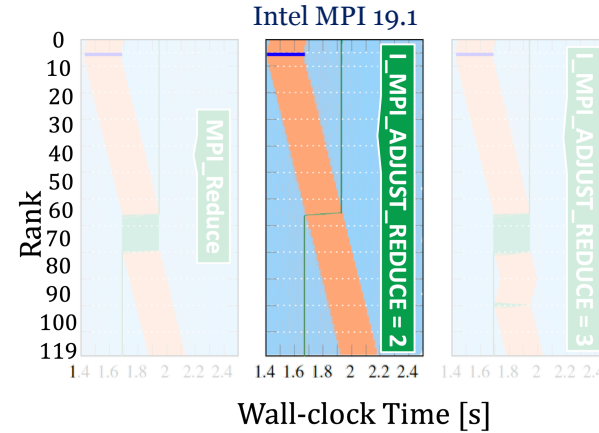
Number of concurrently active processes

Number of concurrently active processes

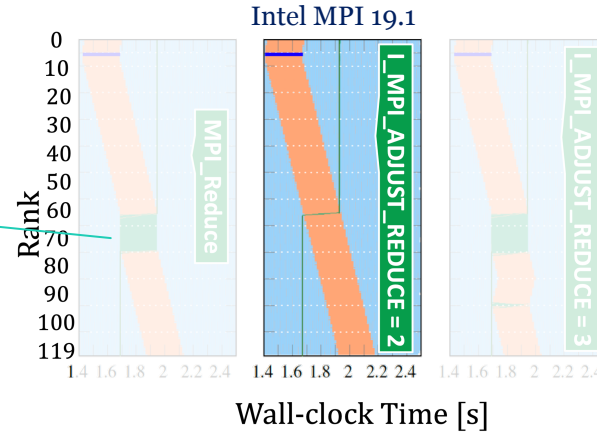


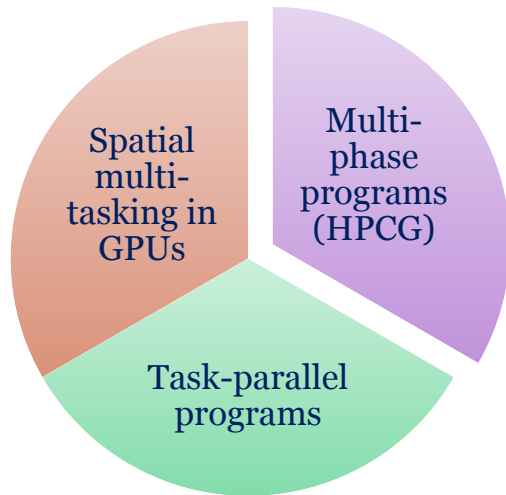


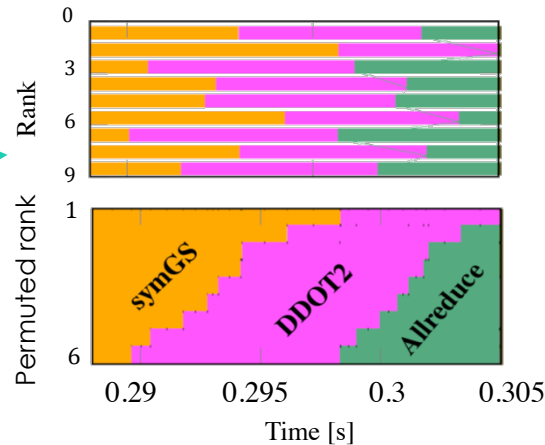
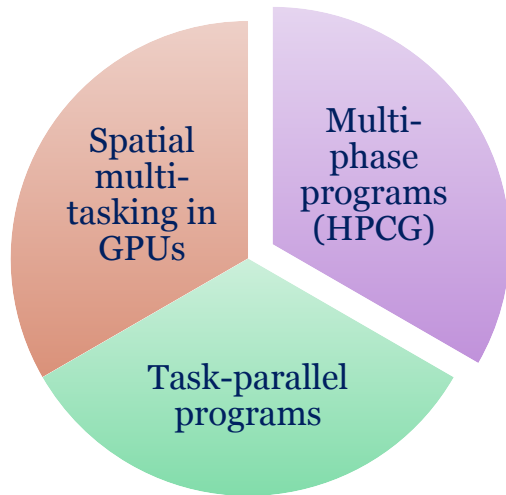




Algorithmic dependency







Afzal et al.



arXiv: 2011.00243
[cs.DC]

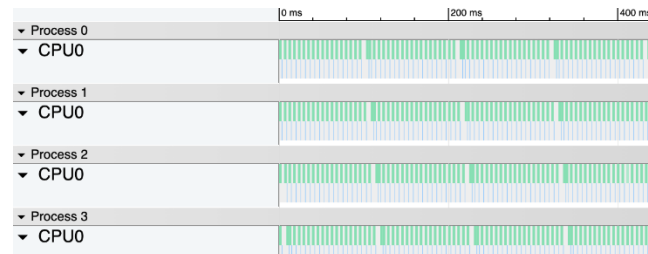
Machine + application knowledge



Parallel simulator



Graphical trace visualization



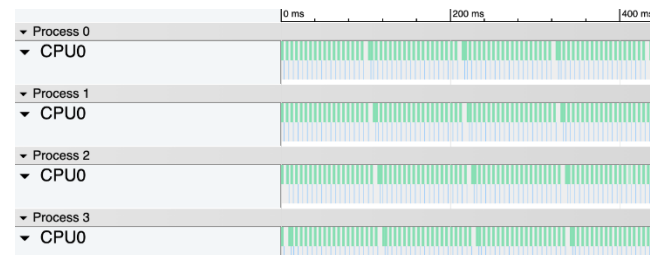
Machine + application knowledge

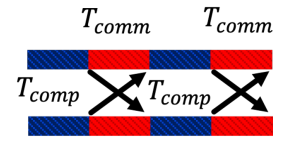


Parallel simulator

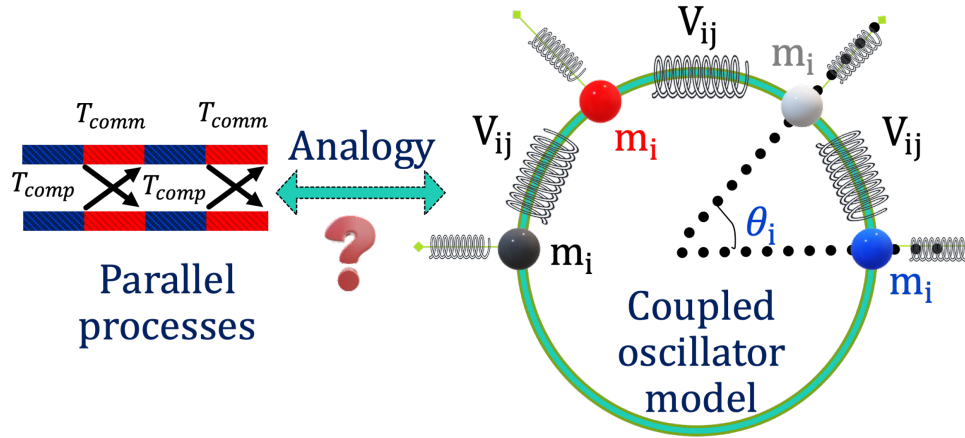


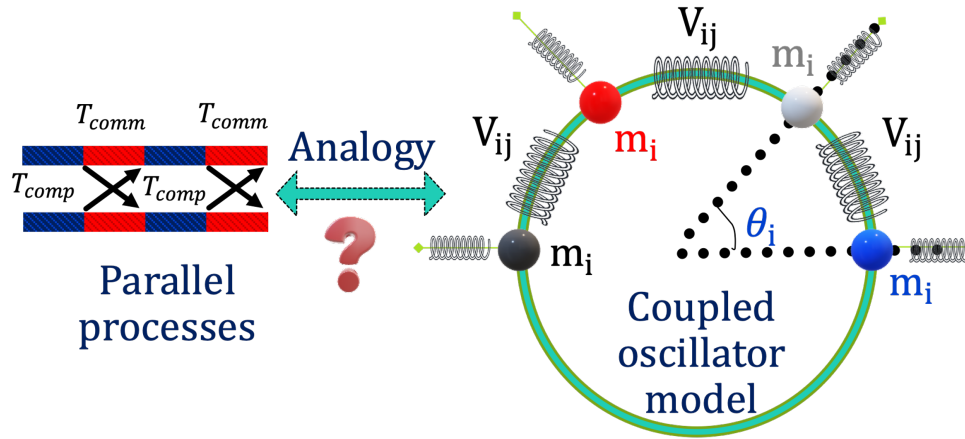
Graphical trace visualization





Parallel
processes





ISC HPC 2021
research poster



Noise-Driven Cluster-Level Performance Modelling and Engineering

Ayesha Afzal
ayesha.afzal@fau.de

POSTER
ARTIFACTS



[https://tiny.cc/
AR_PhDForum_ISC21](https://tiny.cc/AR_PhDForum_ISC21)

POSTER



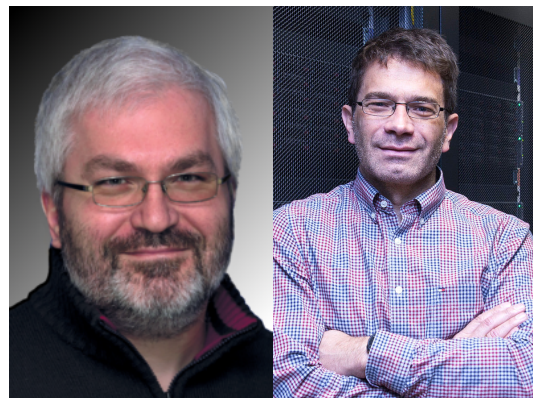
[https://tiny.cc/
PhDForum_ISC21](https://tiny.cc/PhDForum_ISC21)

1



Title Noise-Driven Cluster-Level
Performance Modelling and Engineering

Contact Ayesha Afzal
ayesha.afzal@fau.de



Acknowledgement Advisors
Georg Hager, Gerhard Wellein

POSTER
ARTIFACTS



[https://tiny.cc/
AR_PhDForum_ISC21](https://tiny.cc/AR_PhDForum_ISC21)

POSTER



[https://tiny.cc/
PhDForum_ISC21](https://tiny.cc/PhDForum_ISC21)

1



Title Noise-Driven Cluster-Level
Performance Modelling and Engineering

Contact Ayesha Afzal
ayesha.afzal@fau.de

Acknowledgement Advisors
Georg Hager, Gerhard Wellein

OMI4papps



POSTER
ARTIFACTS



[https://tiny.cc/
AR_PhDForum_ISC21](https://tiny.cc/AR_PhDForum_ISC21)

POSTER



[https://tiny.cc/
PhDForum_ISC2](https://tiny.cc/PhDForum_ISC2)

1



Title Noise-Driven Cluster-Level
Performance Modelling and Engineering

Contact Ayesha Afzal
ayesha.afzal@fau.de

Acknowledgement Advisors
Georg Hager, Gerhard Wellein

OMI4papps



POSTER
ARTIFACTS



[https://tiny.cc/
AR_PhDForum_ISC21](https://tiny.cc/AR_PhDForum_ISC21)

POSTER



[https://tiny.cc/
PhDForum_ISC21](https://tiny.cc/PhDForum_ISC21)

1

*Thank you
for Listening!*

KONWIHR