NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Spontaneous Asynchronicity: Parallel Programs out of Lockstep

Georg Hager, Ayesha Afzal
Erlangen National High Performance Computing Center (NHR@FAU)

PPAM 2022
Gdansk, Poland
September 13, 2022

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Analytic performance modeling

Goals and challanges

Starting point

Analytic (first-principles) performance modeling:

> Constructing a simplified model for the interaction between software and hardware in order to understand lowest-order performance behavior

- Basic questions addressed by analytic performance models
  - What is the bottleneck?
  - What is the next bottleneck after optimization?
  - Impact of hardware features → co-design, architectural exploration

- What if the model fails?
  - We learn something
  - We may still be able to use the model in a less predictive way

Examples for white-/gray-box models in computing

program speedup

serial fraction

$$S(N) = \frac{1}{s + \frac{1-s}{N} + c(N)}$$

Amdahl's Law with communication

latency

msg. length

bandwidth

$$T_{PtP} = T_l + \frac{L}{B}$$

Hockney model for message transmission time

time for computation

time for data transfer

$$T_{\text{exec}} = \max(T_{\text{calc}}, T_{\text{data}})$$

Roofline model for loop code execution time

non-overlapping execution

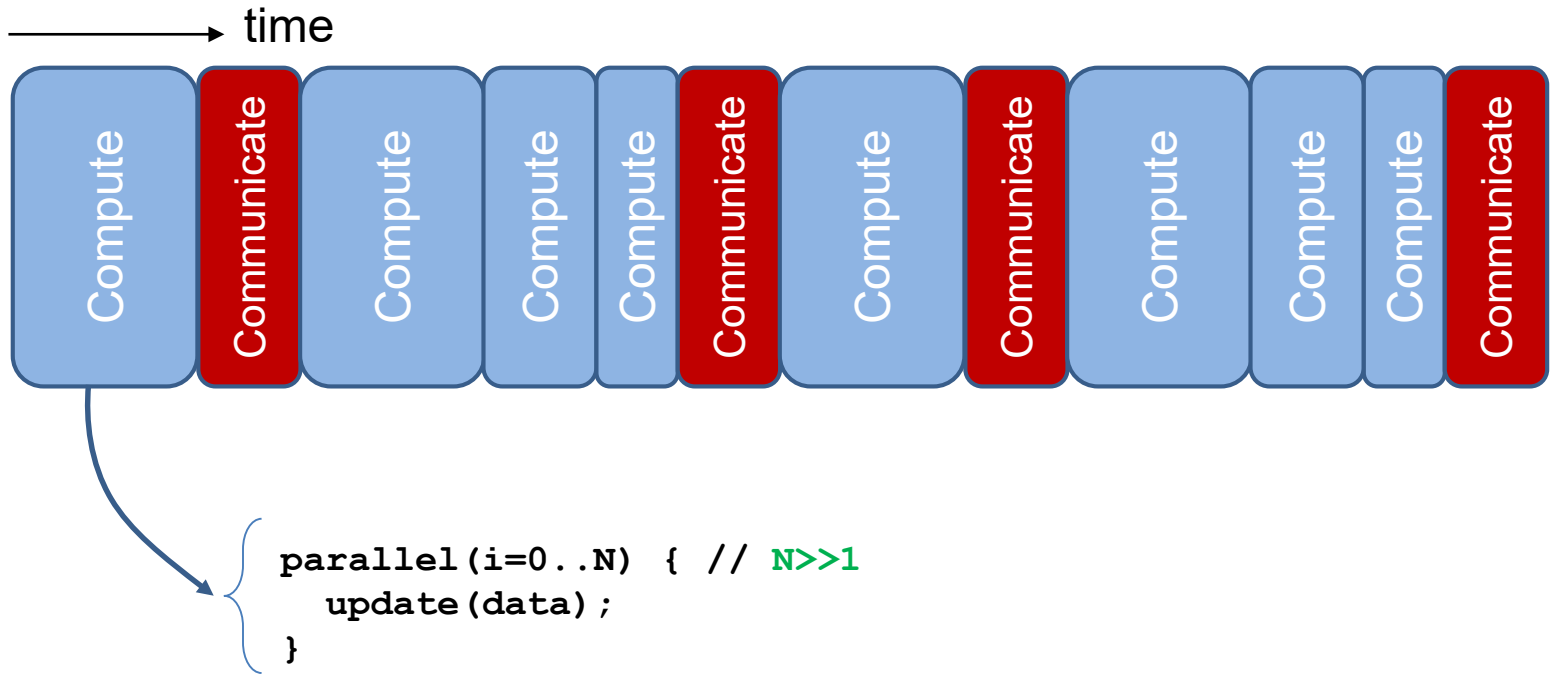
time for data transfer

overlapping execution

$$T_{\text{exec}} = f(T_{nOL}, T_{\text{data}}, T_{OL})$$

ECM model for loop code execution time

General application pattern



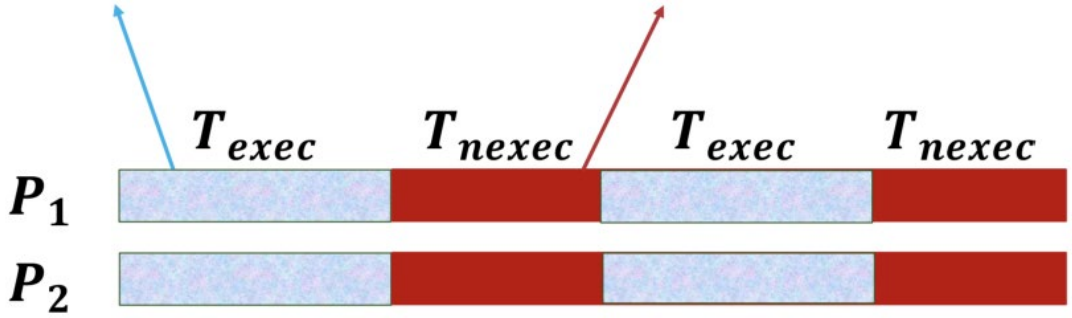
Runtime model: $T = f(\texttt{\$CODE}, \texttt{\$HARDWARE})$

Composite analytic models

Plausible assumption: $T = T_{exec} + T_{nexec}$

e.g.,
$\max(T_{BW}, T_{flops})$

e.g.,
$\lambda + \frac{V}{B}$



In practice, $T \neq T_{exec} + T_{nexec}$ and it can go in either direction

Idle wave propagation and (de)synchronization phenomena

- A. Afzal, G. Hager, and G. Wellein: *Propagation and Decay of Injected One-Off Delays on Clusters: A Case Study.* Proc. 2019 IEEE International Conference on Cluster Computing (CLUSTER), Albuquerque, NM, September 23-26, 2019. DOI: 10.1109/CLUSTER.2019.8890995
- A. Afzal, G. Hager, and G. Wellein: *Desynchronization and Wave Pattern Formation in MPI-Parallel and Hybrid Memory-Bound Programs*. In: P. Sadayappan, B. Chamberlain, G. Juckeland, H. Ltaief (eds): High Performance Computing. ISC High Performance 2020. Lecture Notes in Computer Science, vol 12151. Springer, Cham. **Available with Open Access.** DOI: 10.1007/978-3-030-50743-5_20
- A. Afzal, G. Hager, and G. Wellein: *Delay Flow Mechanisms on Clusters*. Poster at EuroMPI 2019. EuroMPI2019_AHW-Poster.pdf EuroMPI2019-AHW-Summary.pdf
- A. Afzal, G. Hager, and G. Wellein: *Analytic Modeling of Idle Waves in Parallel Programs: Communication, Cluster Topology, and Noise Impact*. ISC High Performance 2021 Digital, June 24 – July 2, 2021, Frankfurt, Germany. DOI: 10.1007/978-3-030-78713-4_19
- A. Afzal, G. Hager, and G. Wellein: *An analytic performance model for overlapping execution of memory-bound loop kernels on multicore CPUs*. Concurrency and Computation: Practice and Experience **34**(10), e6816 (2022). **Available with Open Access.** DOI: 10.1002/cpe.681

Ayesha Afzal

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

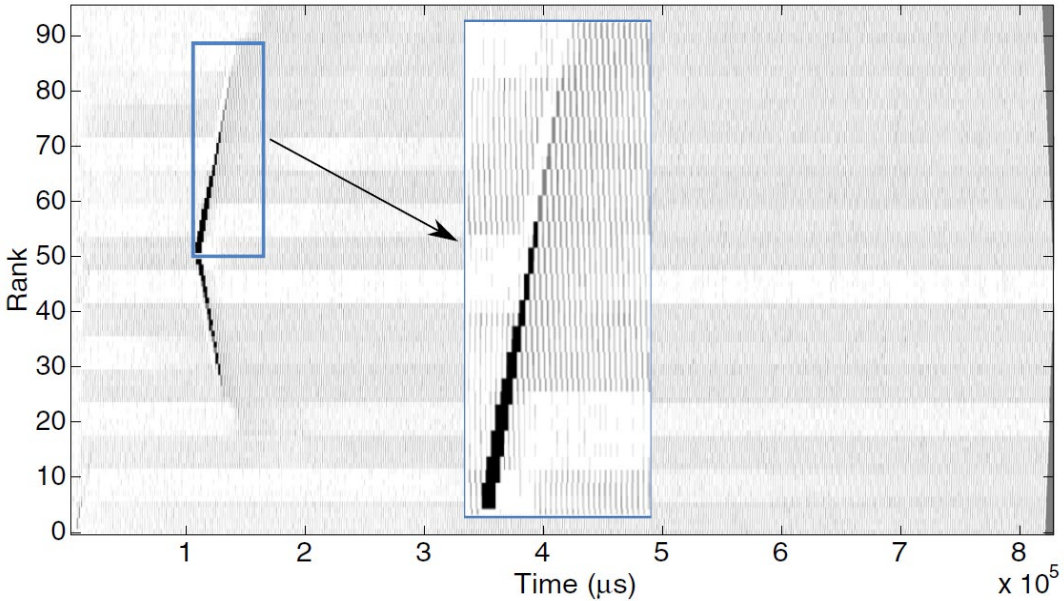
Idle wave propagation and decay

Markidis et al. (2015)

Simulator-based analysis

Idle waves perceived as “damped linear waves”

Classical wave equation postulated for continuum description

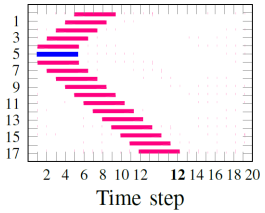


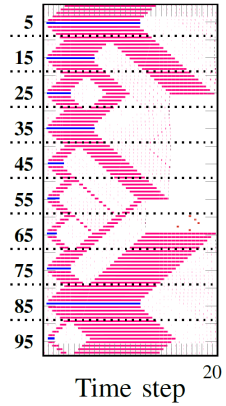
S. Markidis et al.: *Idle waves in high-performance computing*. Phys. Rev. E **91**(1), 013306 (2015).
DOI: 10.1103/PhysRevE.91.013306

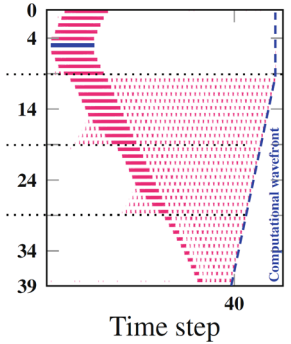
Research questions

Setting: MPI- or hybrid-parallel bulk-synchronous barrier-free programs

- How do “disturbances” propagate?
  - Injected idle periods
  - Dependence on communication characteristics
- How do idle waves interact with each other, with noise, and with the hardware?
  - Idle wave decay
    (noise-induced, bottleneck-induced, topology-induced)
- How do computational waves form? Instabilities?
  - Core-bound vs. memory-bound
  - Amplitude of the computational wave?
- Continuum description?







Idle wave propagation speed for scalable code

Assumptions:

- Scalable code (on ccNUMA domain)
- alternates between execution and communication phases
- has inter-process dependencies via point-to-point communication

Simplest case: Next-neighbor (e.g., 1-D halo) communication

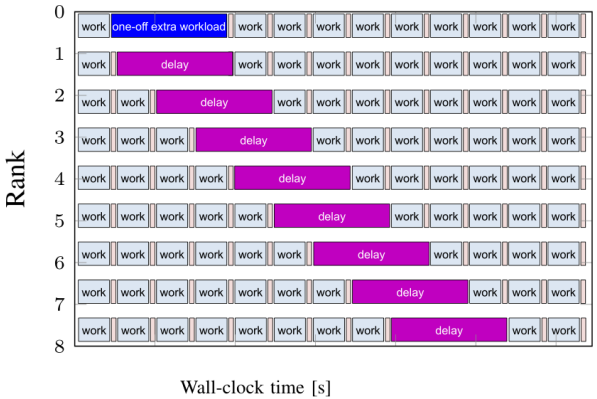
$$v_{\text{silent}}^{\min} = 1 \left[\frac{\text{ranks}}{\text{iter}}\right] \times \frac{1}{T_{\text{exec}} + T_{\text{comm}}} \left[\frac{\text{iter}}{\text{s}}\right]$$

Number of communication partners and details of communication grouping influence the speed

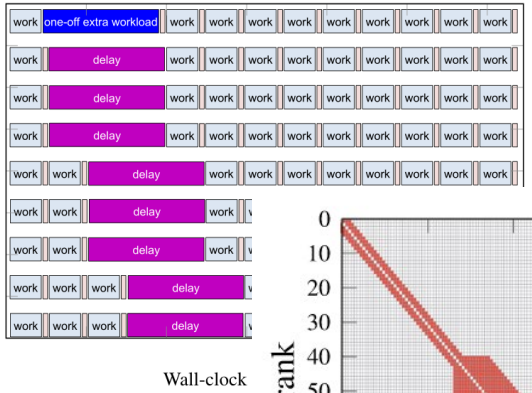
DOI: 10.1007/978-3-030-78713-4_19

Communication topology and idle wave speed

Long-distance point-to-point communication → fast idle waves

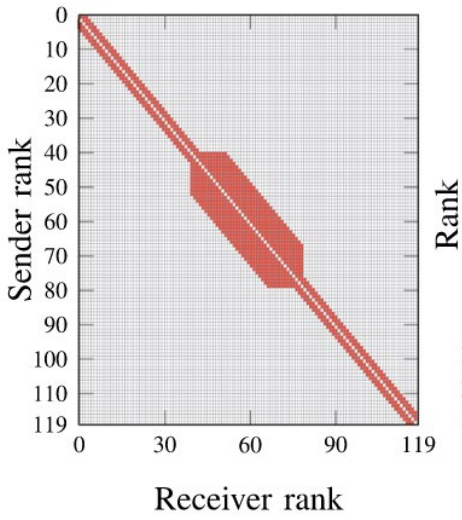


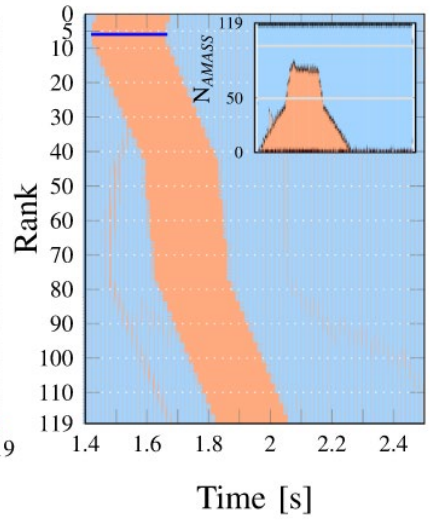
(a) $d = \pm(1)$ (slow idle wave)



(b) $d = \pm(1, 2)$ (thr

arXiv:2205.04190

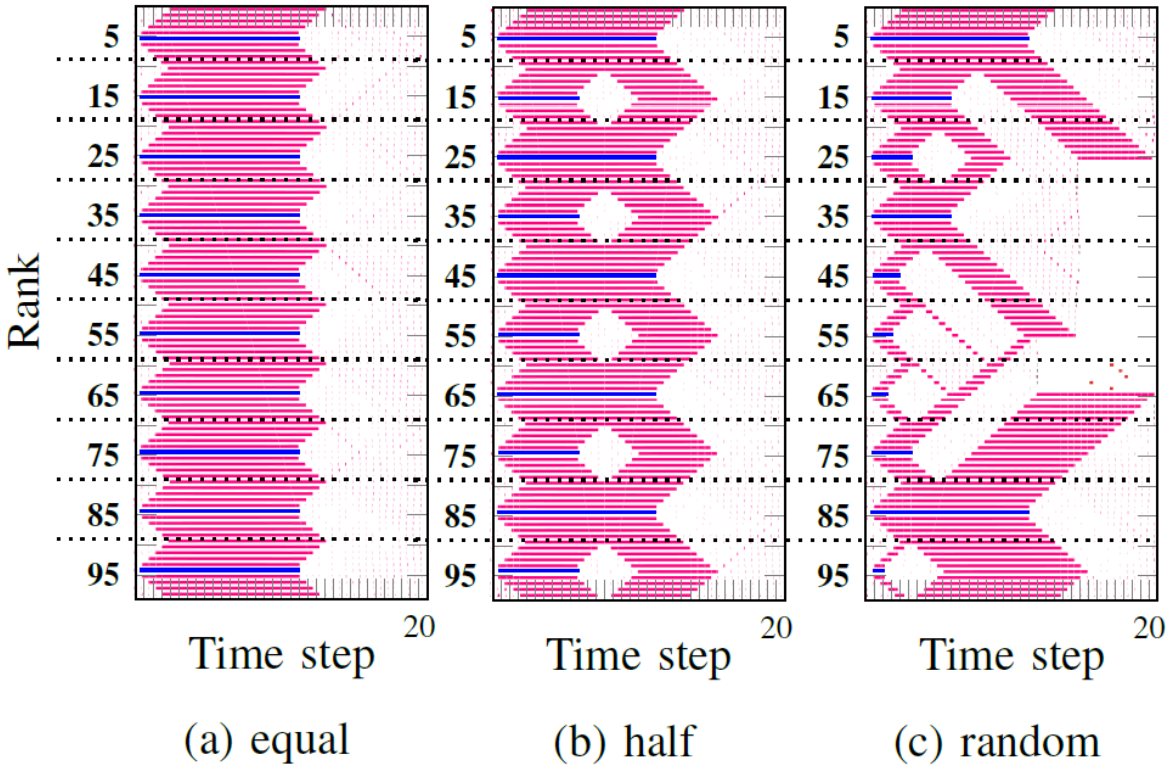




Idle waves interact nonlinearly

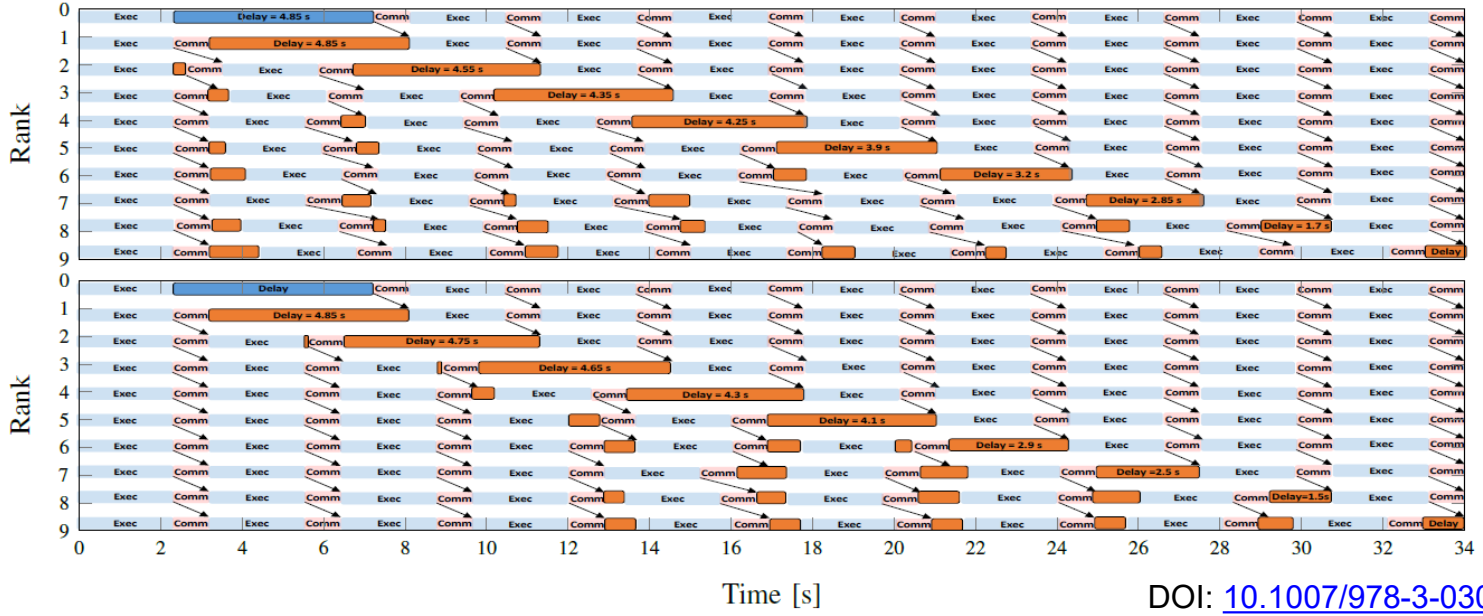
- A wave-like description cannot be based on a linear model
- Basis for noise-induced decay of idle waves

DOI: 10.1109/CLUSTER.2019.8890995



Noise-induced idle wave decay

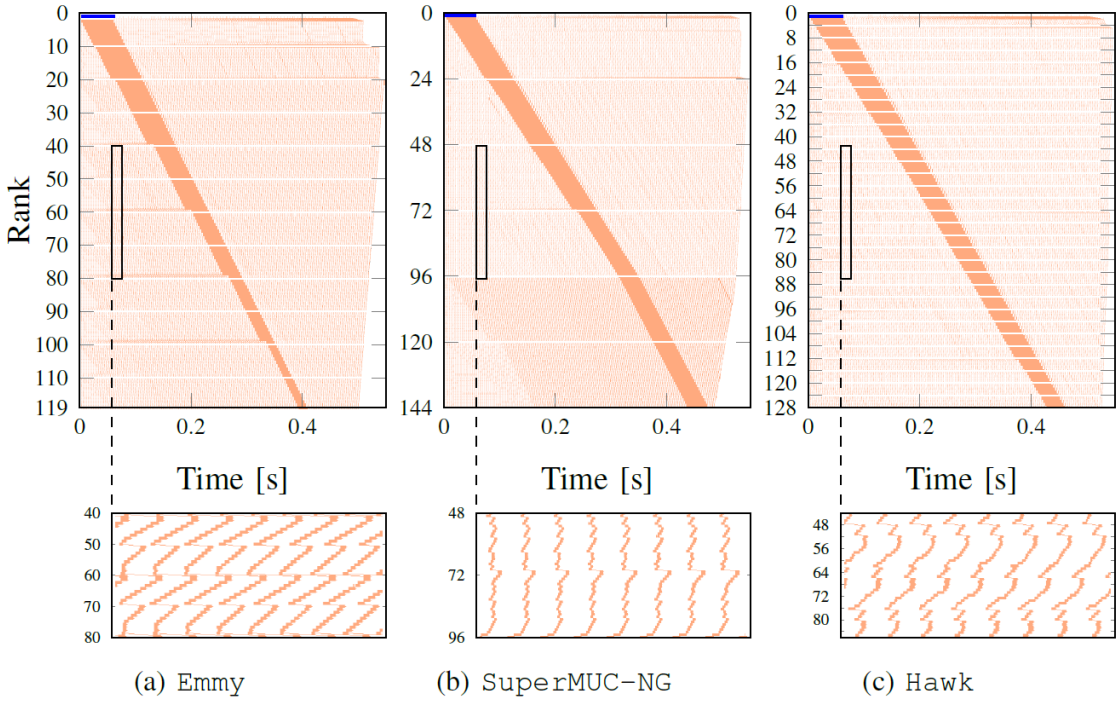
- System or application noise “eats away” on the idle wave
- Decay rate proportional to integrated noise power
- Statistical details do not matter



DOI: 10.1007/978-3-030-78713-4_19

Topological idle wave decay

- Topological boundaries (ccNUMA domains, sockets, nodes) cause fine-grained noise which dampens the idle wave
- Highly system dependent
- No decay in homogeneous situation (round-robin placement)



DOI: 10.1007/978-3-030-78713-4_19

NHR FAU

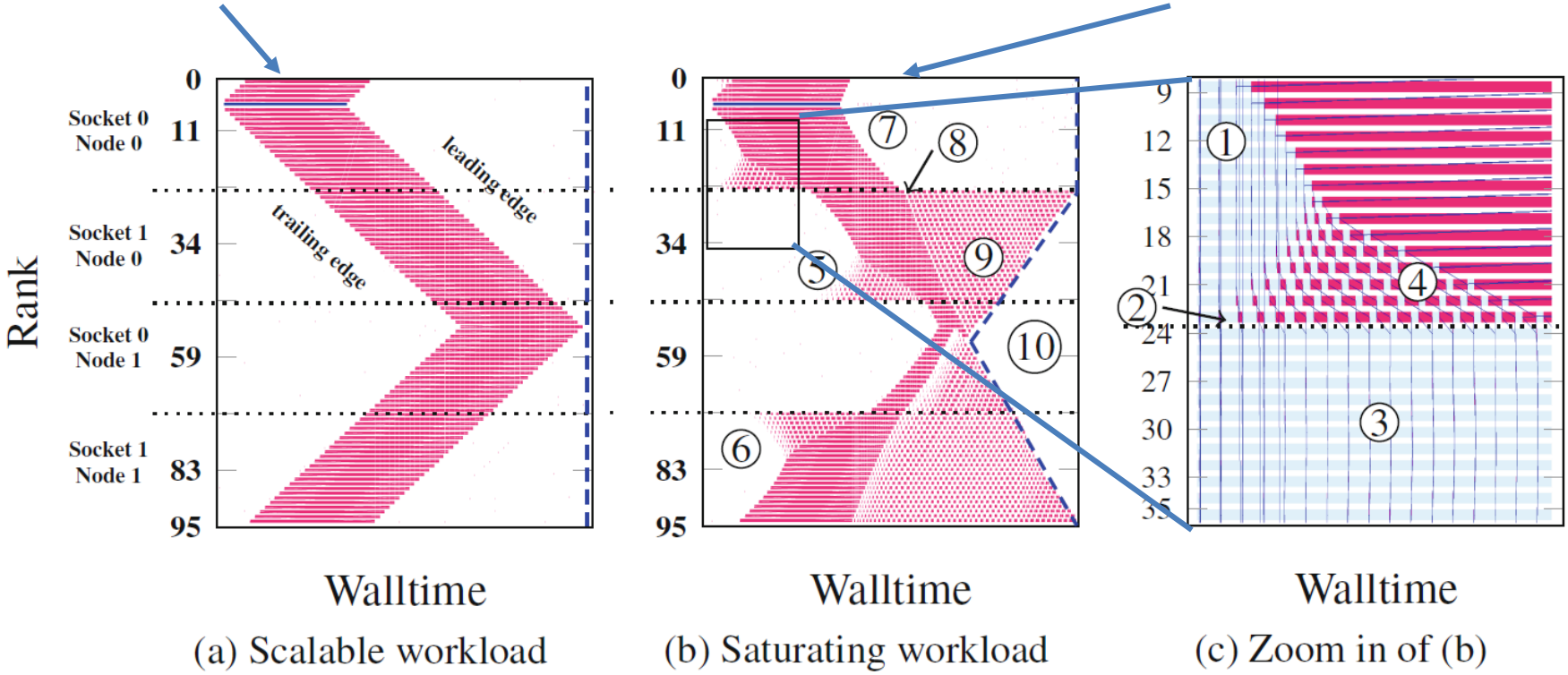
FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Bandwidth-induced idle wave decay and computational waves

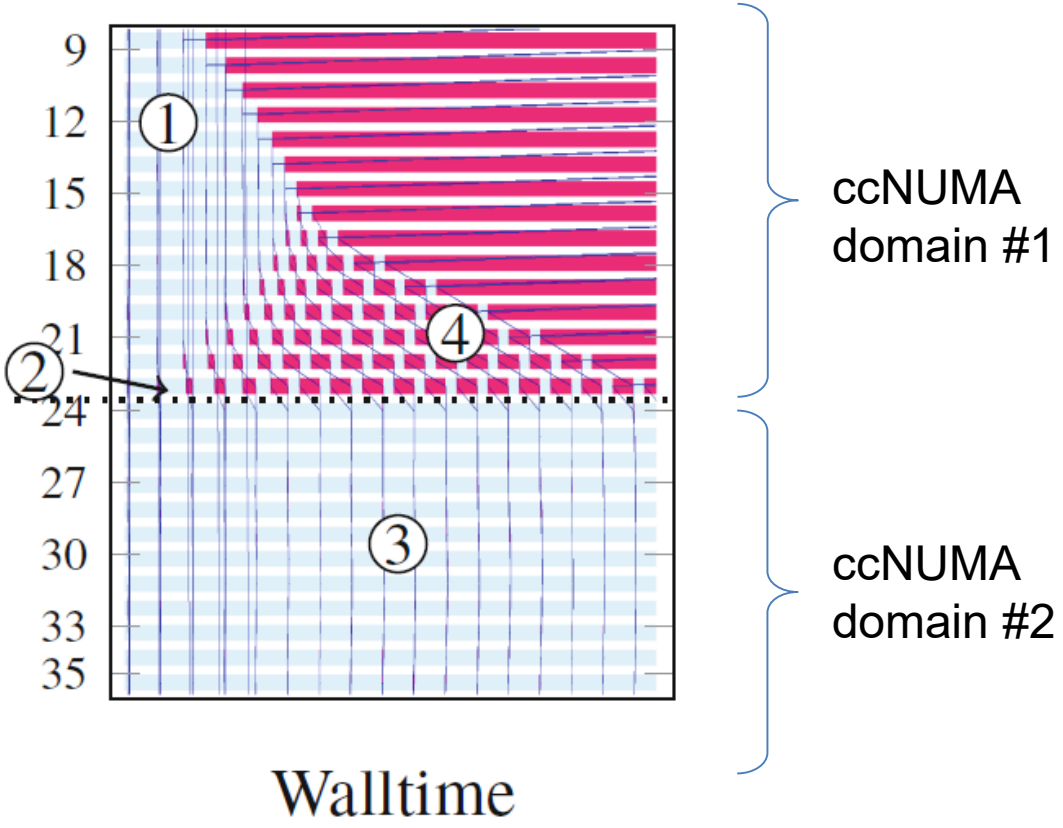
Idle wave propagation and bottleneck-induced decay

Analytical model for idle wave speed with scalalble workload:
DOI: 10.1007/978-3-030-78713-4_19

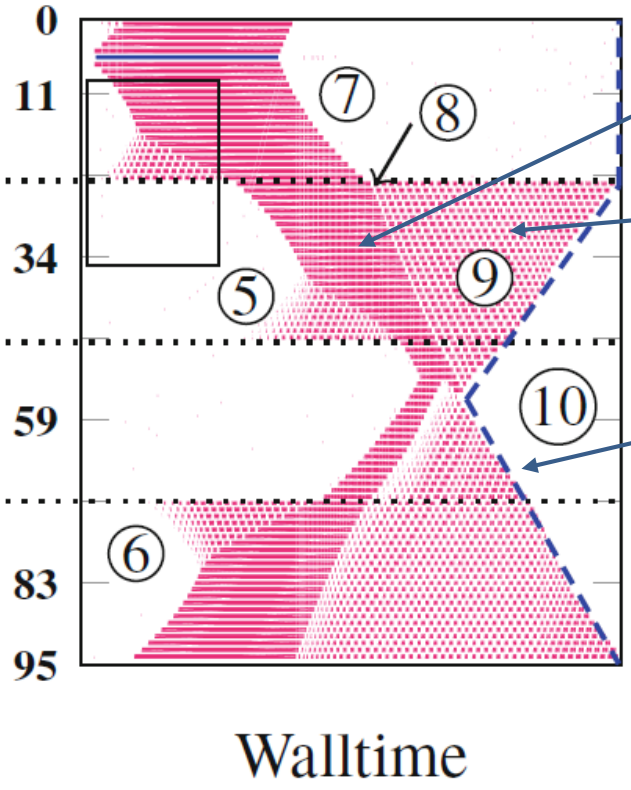
Decay even on silent system:
DOI: 10.1007/978-3-030-50743-5_20



Intricate dynamics of bottleneck-driven idle wave decay



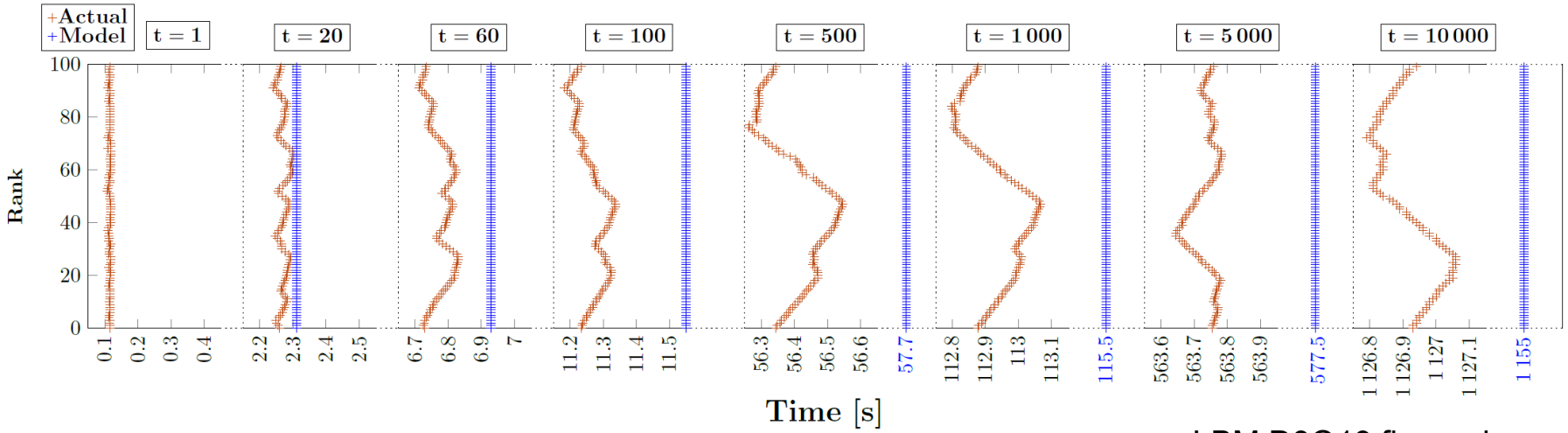
Computational wave as an echo of an idle wave



- Decaying idle wave leaves many processes desynchronized
- Inter-process skew → automatic potential communication overlap
- Computational wavefront == rank-time location of all processes at a given iteration
- Memory boundedness is a prerequisite

Spontaneous asynchronicity

There are always idle waves, and they have consequences…



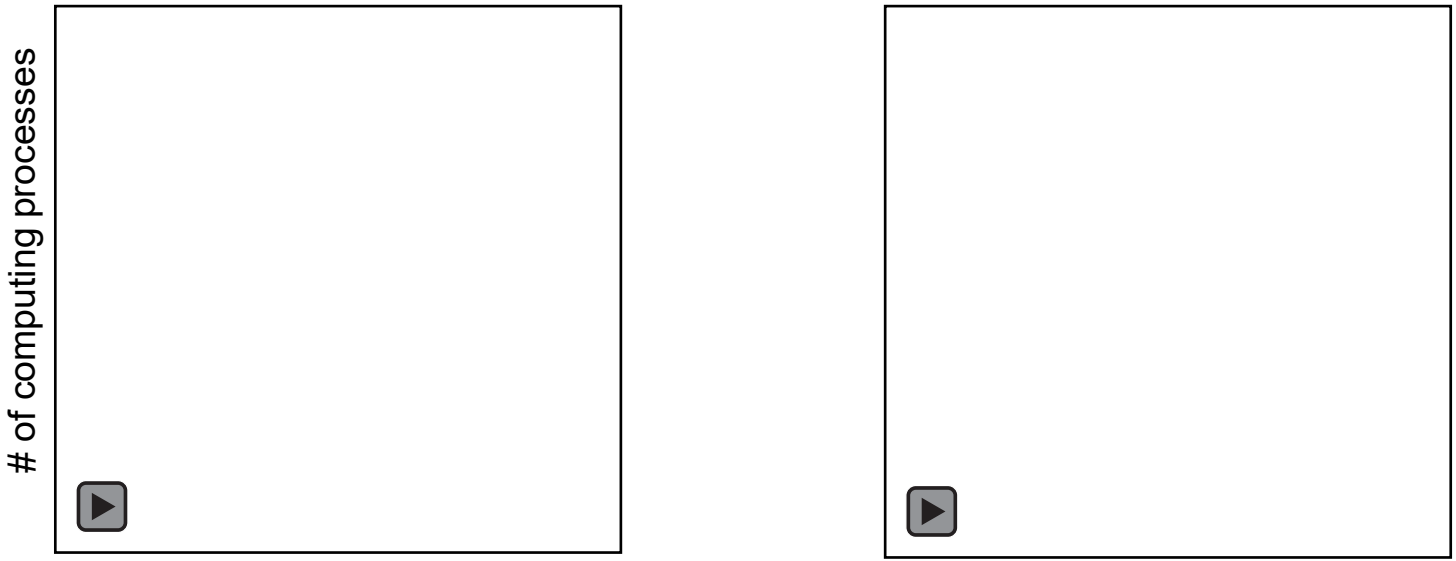
LBM D3Q19 flow solver,
RRZE Emmy cluster,
10 cores per NUMA domain

→ Opportunity for automatic communication overlap

→ "Spontaneous symmetry breaking"?

Computational wave settles at the saturation point (sometimes)

DOI: 10.1007/978-3-030-50743-5_20



NHR FAU

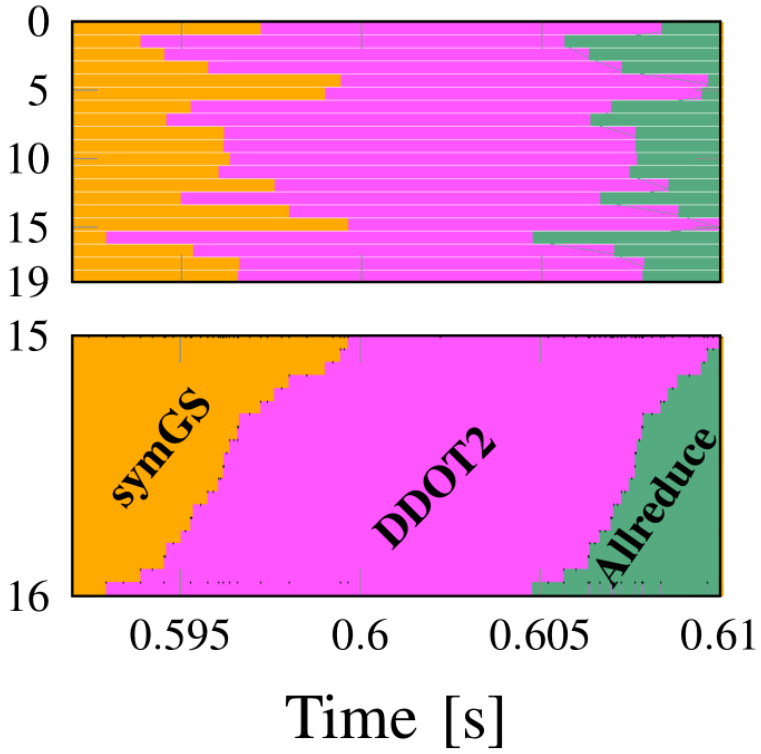
FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Further questions

Further questions about desynchronized execution

- Performance of different kernels running concurrently on one ccNUMA domain? → DOI: 10.1002/cpe.6816

- Decisive parameter: per-kernel single-thread memory bandwidth fraction ("memory pressure")



Further questions about desynchronized execution

- Can we inject a delay to make a program faster?
  - Yes we can
- Do we always have to strictly balance the load?
  - No, some variation may be good for performance
- Is there a more favorable pattern than compute-communicate-repeat?
  - Yes: memory-bound compute – scalable compute – repeat
- Can the system (application,computer) be described by coupled oscillators?

$$\dot{\theta}_i = \omega_i + \alpha \sum_j \sin(\theta_j - \theta_i) \quad \longrightarrow \quad \dot{\theta}_i = \omega_i + \zeta_i(t) + \alpha \sum_j T_{ij}\, V(\theta_j, \theta_i, \tau_{ij}(t))$$

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Thank You.

KONWIHR