Bundesministerium
für Bildung
und Forschung

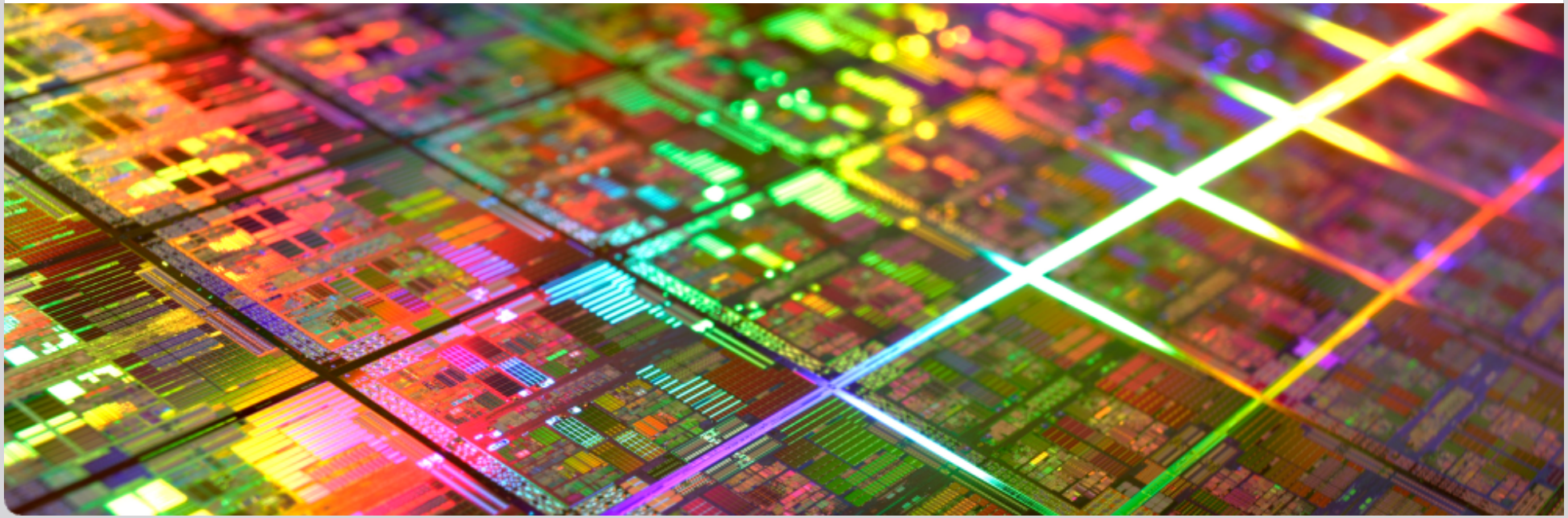
envelope

ENVELOPE - Effizienz und Zuverlässigkeit: Selbstorganisation in HPC-Systemen

Zwischenbericht

A. Brinkmann, W. Karl, S. Lankes, M. Schulz, C. Trinitis

17. Oktober 2019



Partner

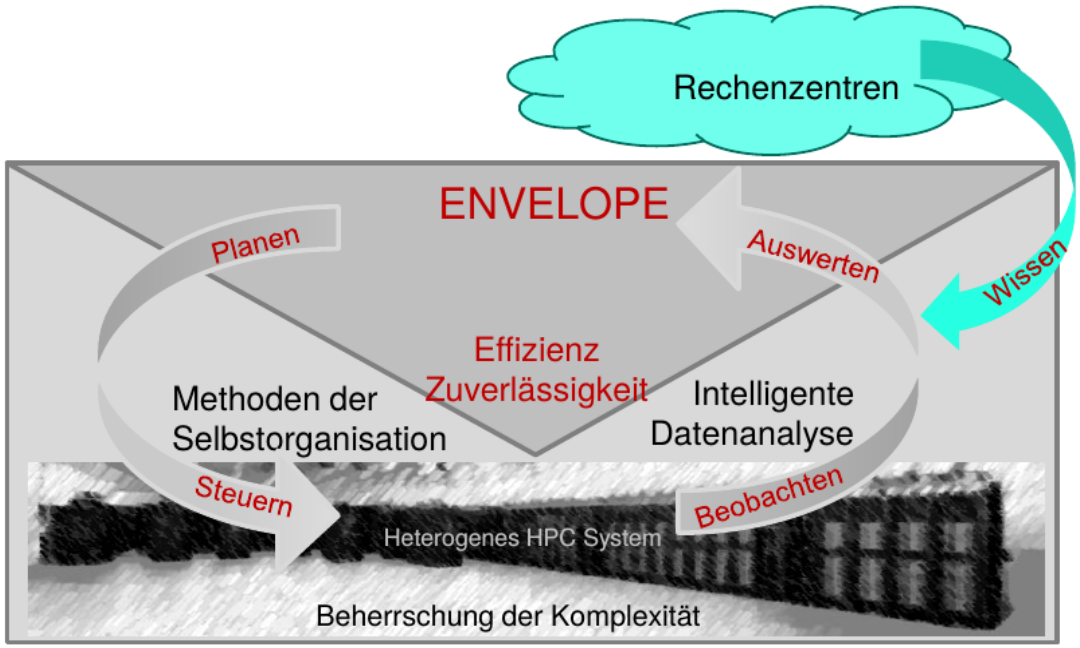
- **Karlsruher Institut für Technologie**
  - Institut für Technische Informatik
  - Prof. Dr. Wolfgang Karl (Koordination)
- **Technische Universität München**
  - Lehrstuhl für Rechnerarchitektur und Parallele Systeme
  - Prof. Dr. Martin Schulz, Dr.-Ing. Carsten Trinitis
- **Johannes Gutenberg-Universität Mainz**
  - Zentrum für Datenverarbeitung
  - Prof. Dr.-Ing. André Brinkmann
- **RWTH Aachen University**
  - Institute for Automation of Complex Power Systems
  - Prof. Antonello Monti, Ph.D., Dr. rer. nat. Stefan Lankes
- **Assoziierte Partner**
  - Leibniz Rechenzentrum
    - PD Dr. rer. nat. Josef Weidendorfer
  - MEGWARE, ParTec

Ziele

- Erhöhung der Zuverlässigkeit und Ausfallsicherheit in HPC-Systemen
- Effiziente Nutzung der zur Verfügung stehenden Ressourcen im Hinblick auf
  - Ausführungszeit
  - Energieeffizienz
- Verbergen der Komplexität heterogener HPC-Systeme vor dem Anwender

Ansatz

- Betrachtung des Systems auf mehreren Ebenen
  - Lokale Betrachtung der Rechenknoten
  - Globale Sichtweise des Systems
  - Techniken auf Anwendungsebene
- Einsatz von Methoden der Selbstorganisation
  - Vergleich von system- und anwendungsbasierten Strategien



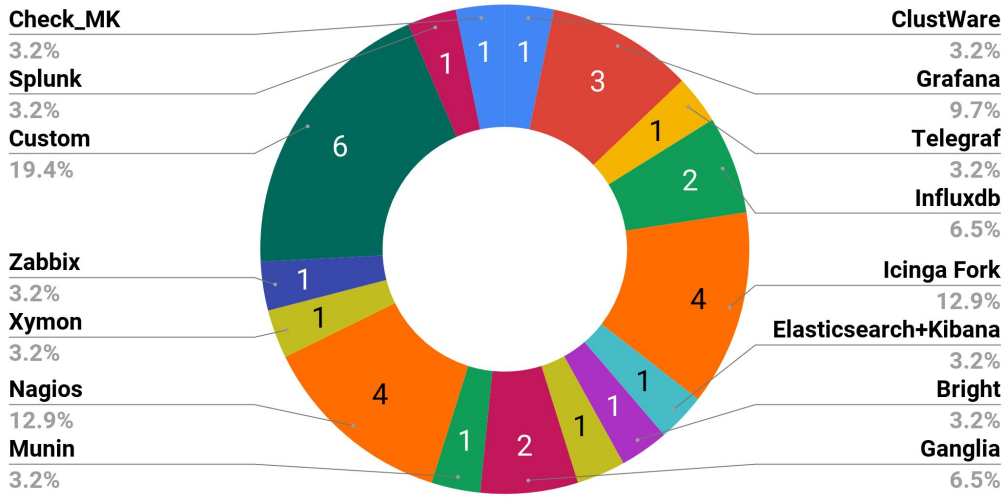
Ansatz

- Erhöhung der Zuverlässigkeit und Ausfallsicherheit
  - Proaktive Vorhersage von Knotenausfällen mittels Machine-Learning Modellen
  - Datenhaltungskomponenten zur Migration und Checkpointing
    - Anwendungsgesteuert
    - Containerbasiert
    - Unikernels
- Effiziente Nutzung der vorhandenen Ressourcen
  - Dynamische Abbildungsentscheidung mit unterschiedlichen Optimierungszielen
- Verbergen der Komplexität heterogener Systeme
  - Task-basiertes Laufzeitsystem mit Bibliotheksansatz

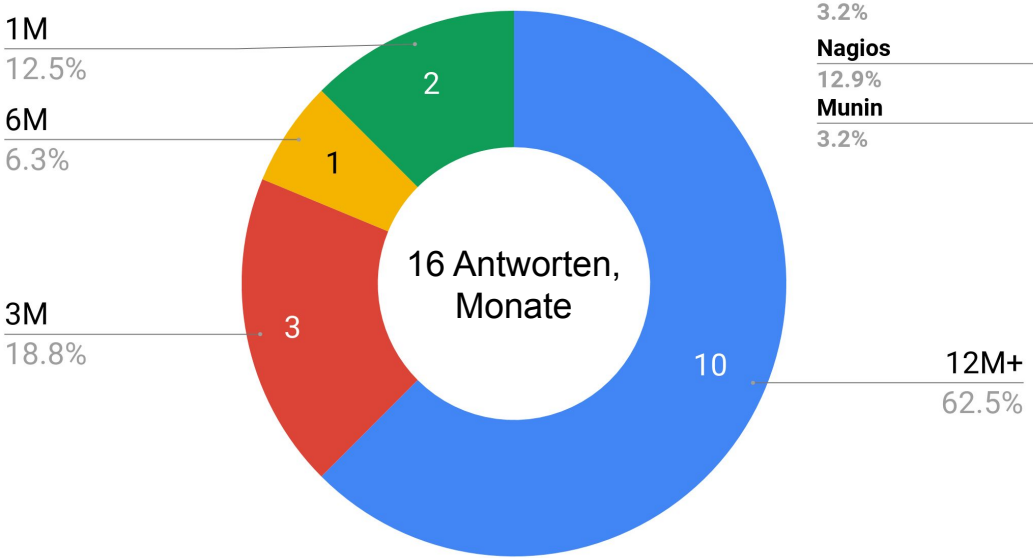
Rechenzentrumsumfrage

envelope

- Welche Monitoringsoftware sind eingesetzt?



- Wie lange werden Monitoringdaten gespeichert?



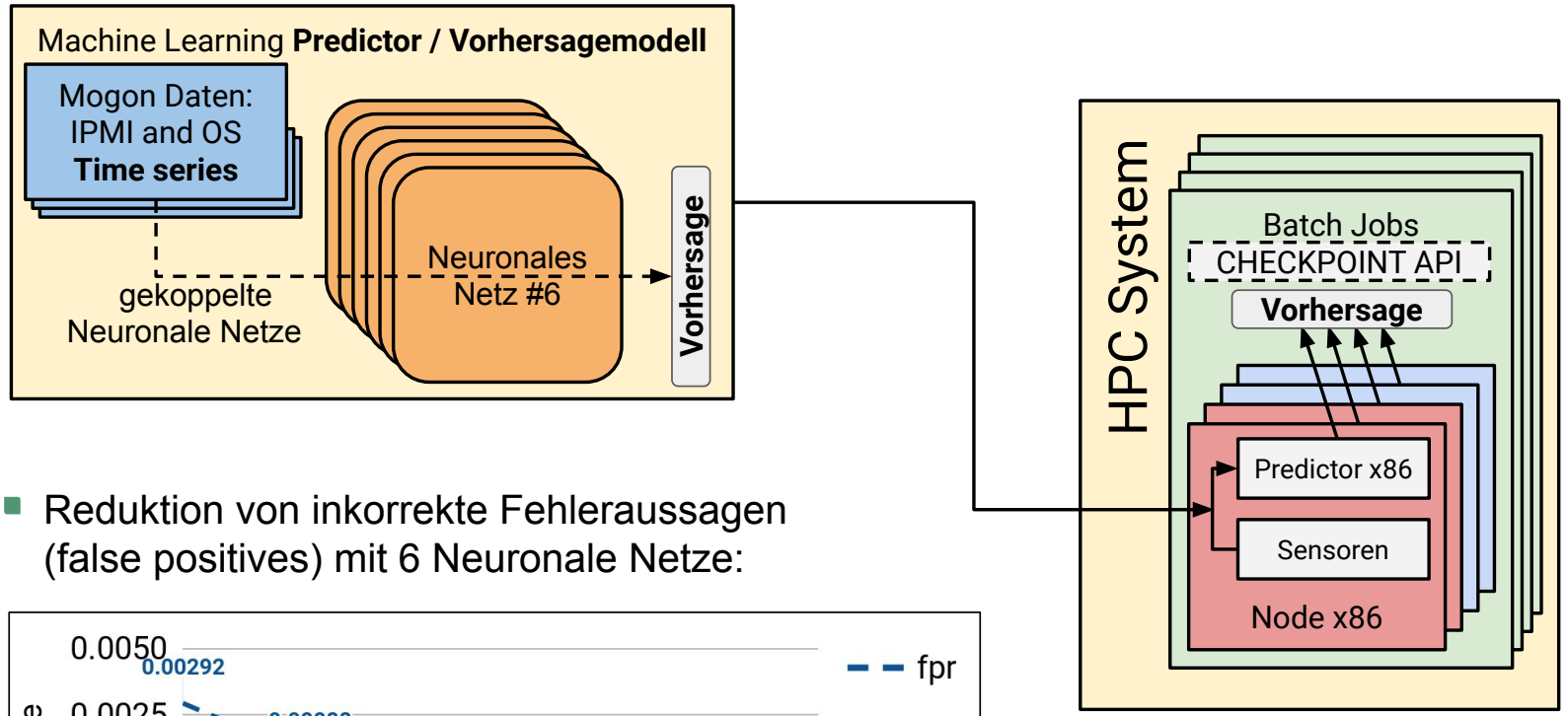
Prädiktor

Ziel: Erstellung eines Prädiktors zur proaktiven Vorhersage von Ausfällen der Rechenknoten

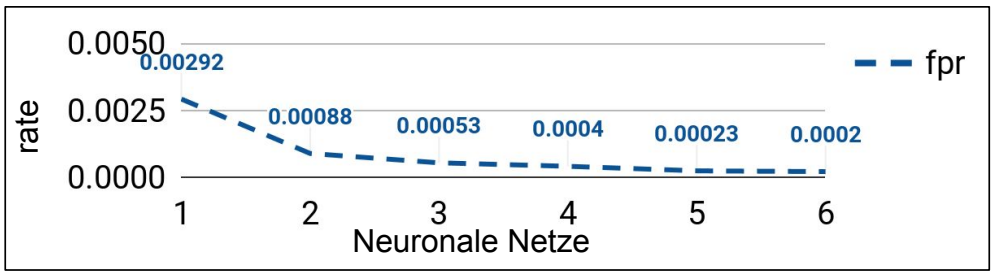
- Datensammlung aus MOGON I & MOGON II an der JGU Mainz
  - IPMI (Spannung, Temperaturen), OS (CPU, Last, Speicher), Job-/Knotenzustände
- Komprimierte Daten: ~2 TB für 1 Jahr und über 500 Knoten
- Tagliche Daten: ~5 GB für 500 Knoten
- MOGON I & II benutzen Ganglia, RRD Datenbasis und Custom Daemons für die Datensammlung

Prädiktor

Ziel: Erstellung eines Prädiktors zur proaktiven Vorhersage von Ausfällen



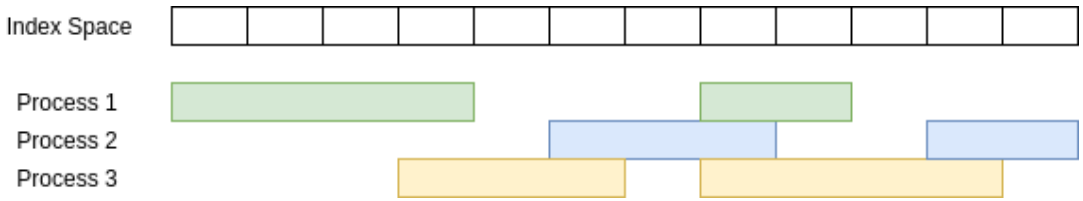
- Reduktion von inkorrekte Fehleraussagen (false positives) mit 6 Neuronale Netze:



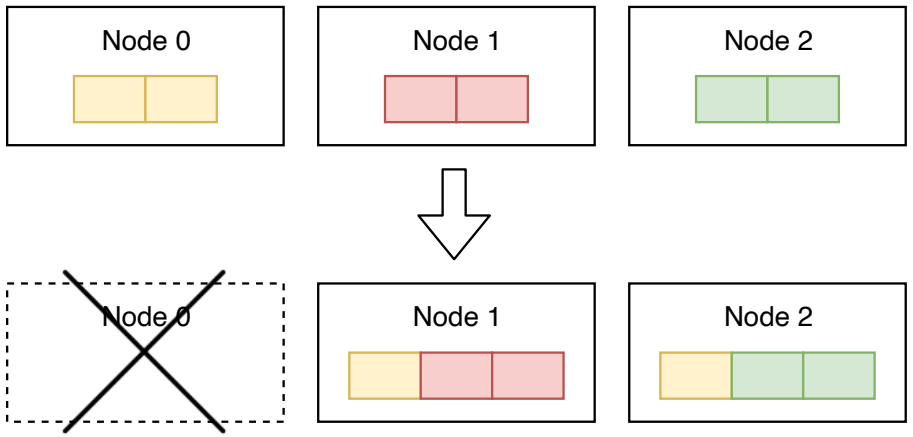
LAIK: Anwendungsintegr. Fehlertoleranz

envelope

- Indexraum-Abstraktion mittels Datenpartitionierung



- Fehlertoleranz durch Umpartitionierung

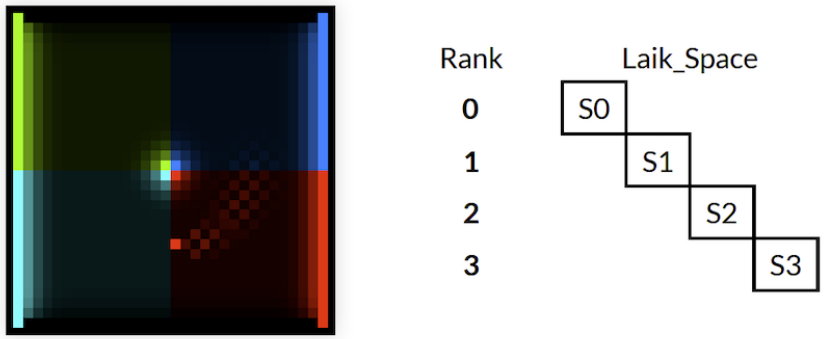


Checkpointing im Arbeitsspeicher

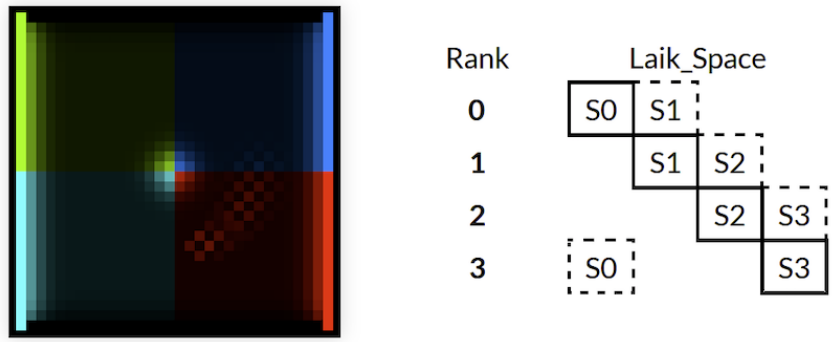
envelope

- In-Memory Checkpoint auf Nachbarknoten
- Beispiel: Jacobi 2D

App Data

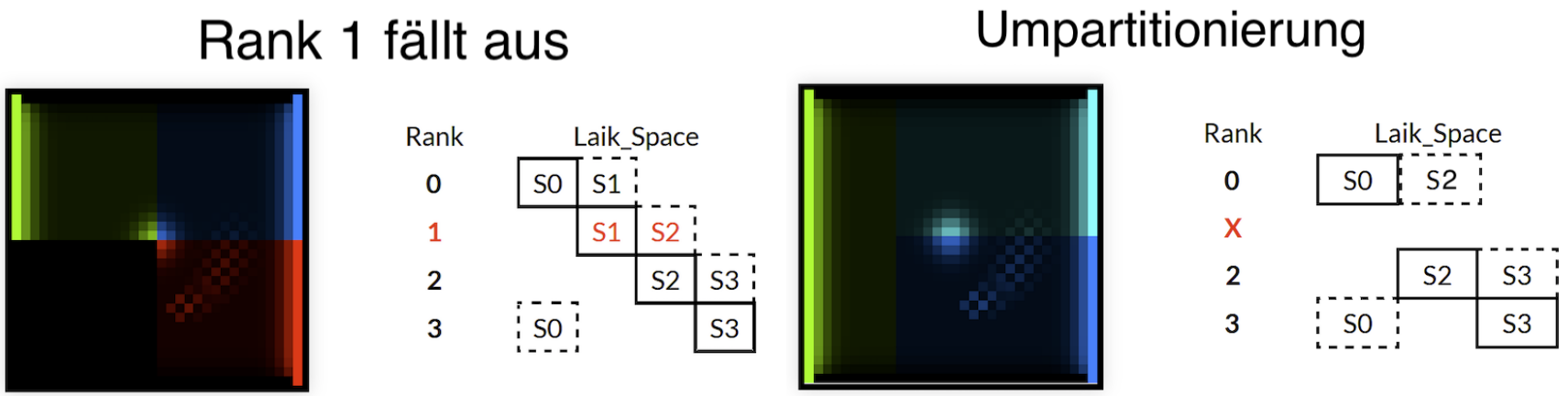


Checkpoint



Rekonstruktion der Datenpartitionen

- Ausgefallene Datenpartition wird bei einer Umpartitionierung rekonstruiert
- Datenpartitionen werden neu balanziert

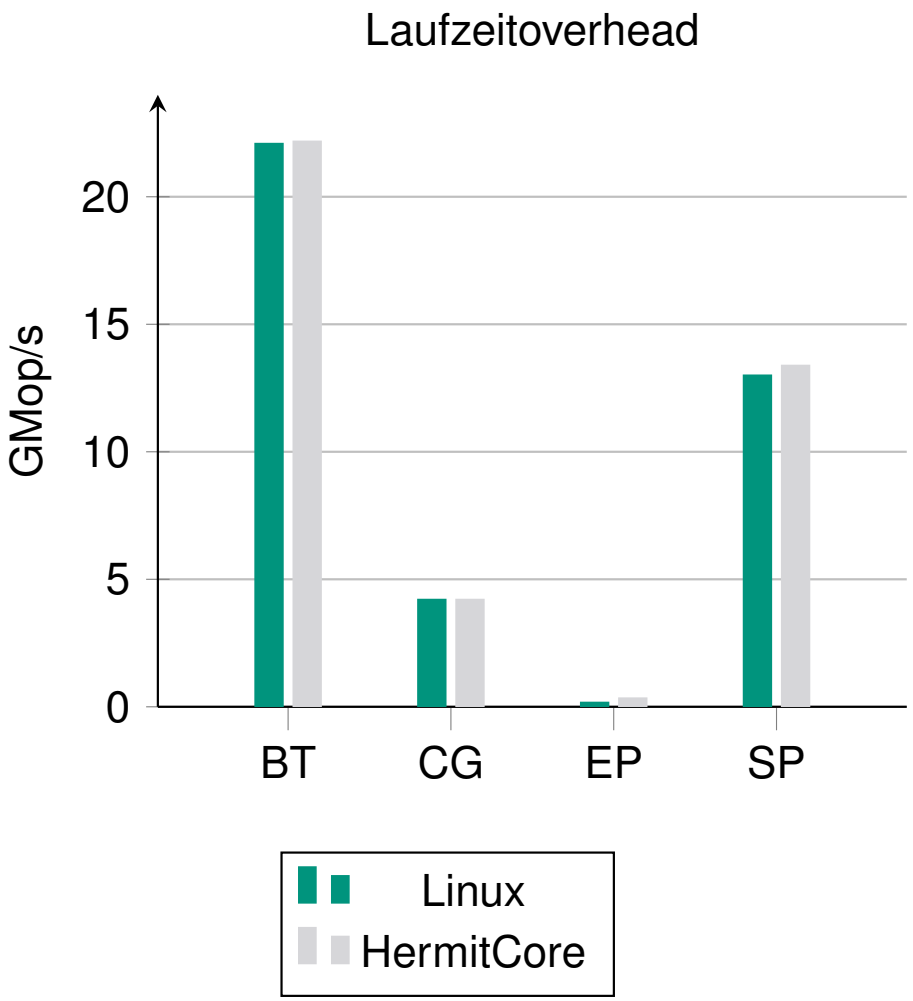


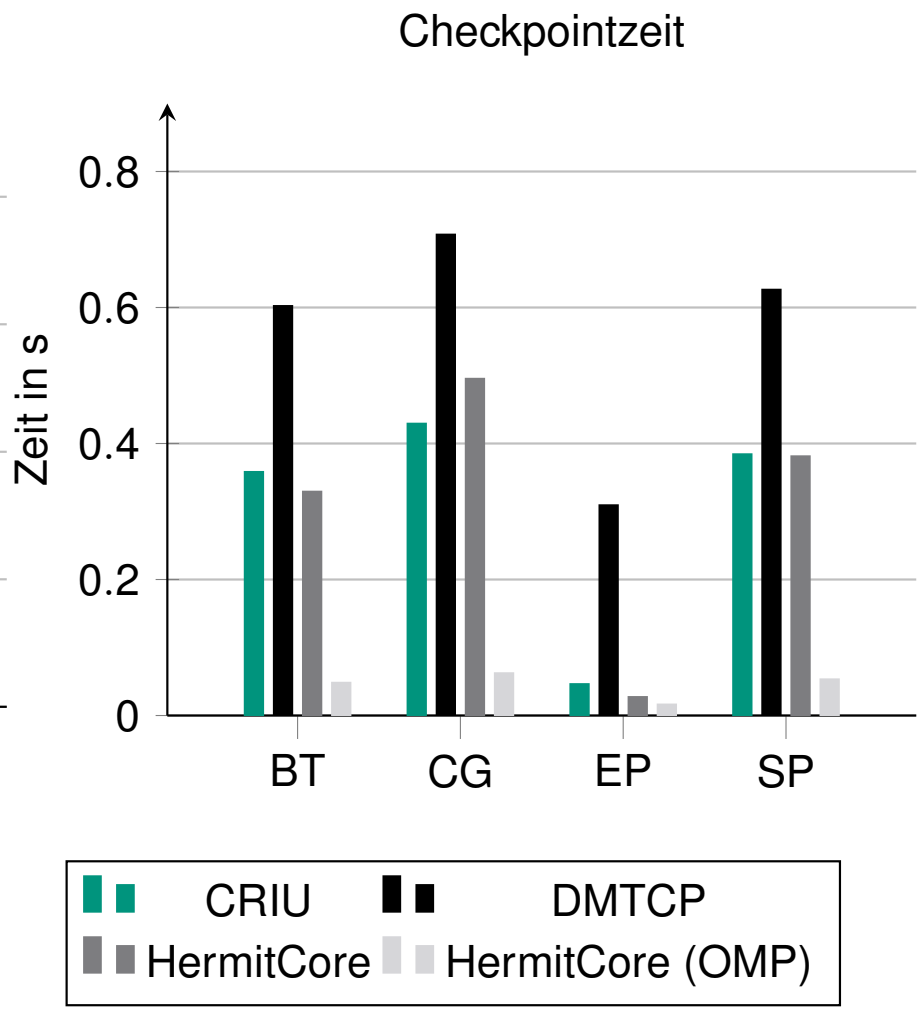
- Demo: `https://youtu.be/-pR2Gl1nOto`

Migration der Anwendung

- LAIK-Demo basiert auf einem fehlertoleranten Laufzeitsystem
- Fehlt dieses, muss die MPI-Anwendung zu einem neuen Knoten migriert werden
- Annahme: Die (MPI-)Anwendung läuft innerhalb einer VM / eines Containers
- Migration der VM / des Containers (z.B. HermitCore)
- Vertikale Integration der Laufzeitsysteme, um die Neuausrichtung der Kommunikation zu ermöglichen
  - U. a. werden die Kommunikatoren (z. B. `MPI_COM_WORLD`) nach der Migration angepasst

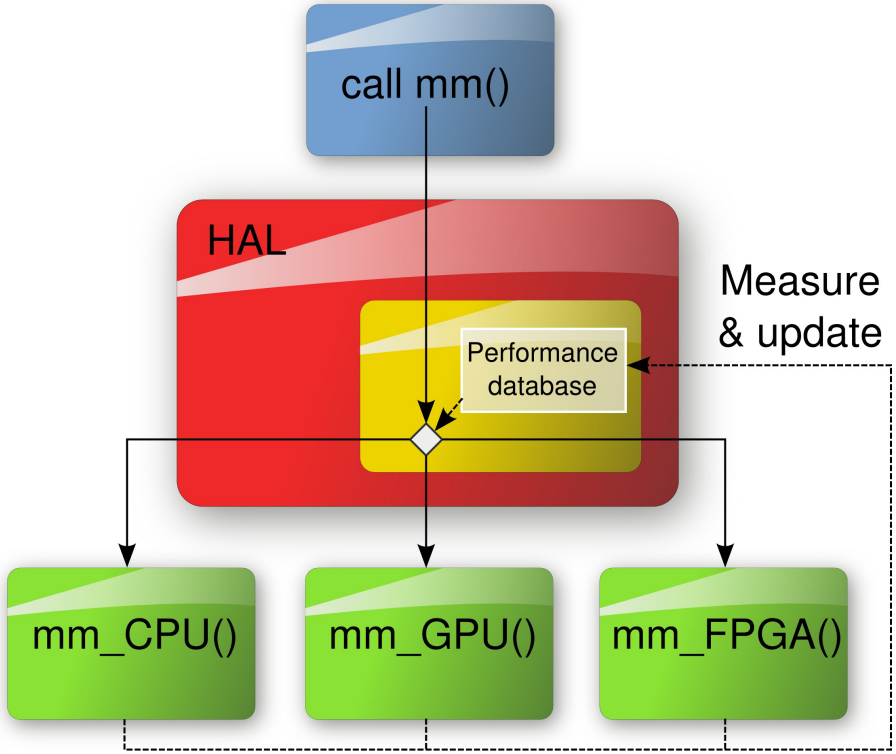
Optimierungen von Checkpoint-Techniken





Task-basiertes Laufzeitsystem

- Knotenlokale Abbildungsentscheidung mittels Ausführungsdatenbank
  - Laufzeit
  - Thread-Anzahl
  - Energieverbrauch
  - Temperatur
  - Heuristische Fehlerrate
- Vorhersage unbekannter Problemgrößen durch Interpolation



Publikationen

- insgesamt 17 Veröffentlichungen (Auswahl)
  - D. Jauk, D. Yang, and M. Schulz.: *Predicting Faults in High Performance Computing Systems: An In-Depth Survey of the State-of-the-Practice*, International Conference for High Performance Computing, Networking, Storage, and Analysis, 2019
  - S. Lankes, S. Pickartz, and J. Breitbart, *HermitCore*, Operating Systems for Supercomputers and High Performance Computing, B. Gerofi, Y. Ishikawa, R. Riesen, and R. W. Wisniewski, Eds. Springer International Publishing, November 2019
  - A. Frank, T. Süß, A. Brinkmann: *Effects and benefits of node sharing strategies in HPC batch systems*, 33rd IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2019
  - T. Becker, N. Rudolf, D. Yang and W. Karl: *Symptom-based Fault Detection in Modern Computer Systems*, PARS Workshop 2019

Fazit

- Entwicklung eines Vorhersagemodells zur Aktivierung von Migration und Checkpointing
- Vertikale Integration:
  - Lokales Laufzeitsystem HALadapt zur Abbildungsverteilung innerhalb eines heterogenen Knotens
  - Globales Laufzeitsystem LAIK zur anwendungsgesteuerten Daten-Migration und -Partitionierung
  - Anwendungstransparentes Migrations-Framework zur Migration mittels VMs

$\Rightarrow$ ENVELOPE: Der Weg zu zuverlässigem Exascale-Computing

. . . damit Sie auch morgen noch kraftvoll rechnen können!