exasim

Software Track Submission

Gregor Olenik

Ginkgo

upstream CFD

Karlsruher Institut für Technologie

Technical University of Munich

Bundesministerium für Bildung und Forschung

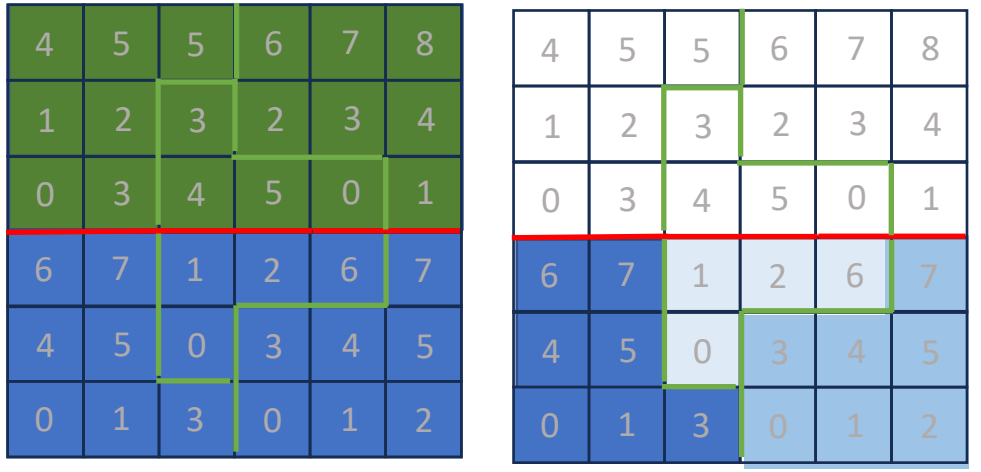
Finanziert von der Europäischen Union - NextGenerationEU

Common Theme: Heterogeneous Computing

- Our focus is offloaded linear solver
- No commonly accepted GPU OF version available that allows to offload matrix assembly
- Plugin based approaches face Over- and Undersubscription Challenge [1]
- Investigated individual partitioning for CPU and GPU workloads

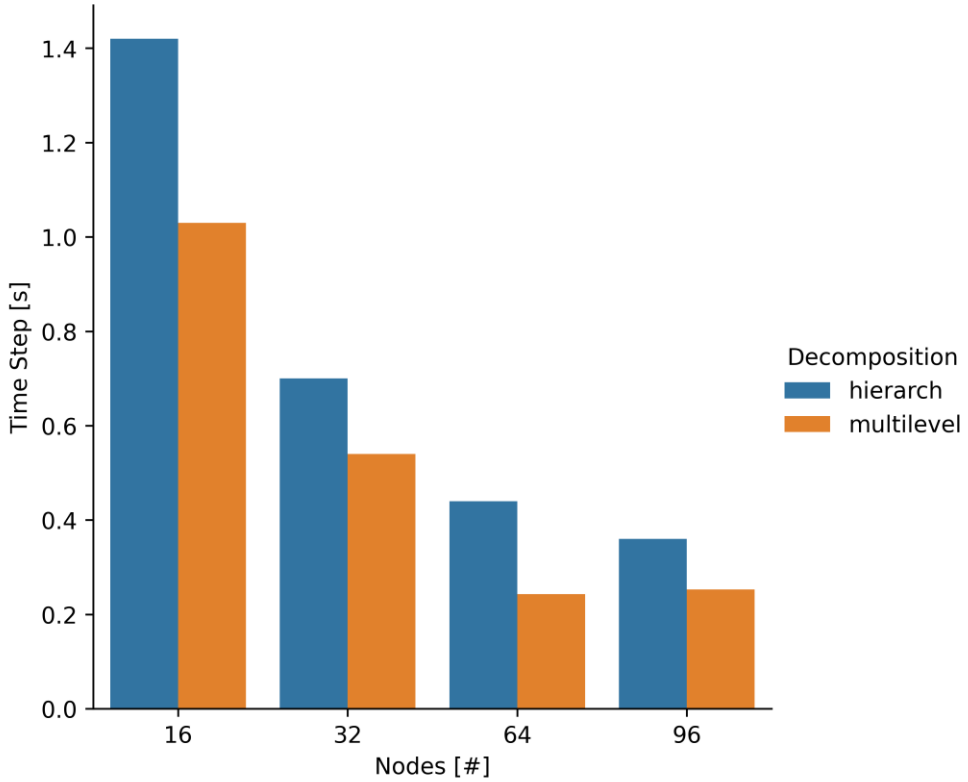
Multilevel for Repartitioning

- For efficient repartitioning we use multilevel decomposition, typical {n_nodes, n_gpu, n-cores} to avoid “holes” in partition
- Automated pre-processing procedure OBR [2], also used for baseline CPU runs typical { n_nodes, n_sockets, n-cores} -> Software Track



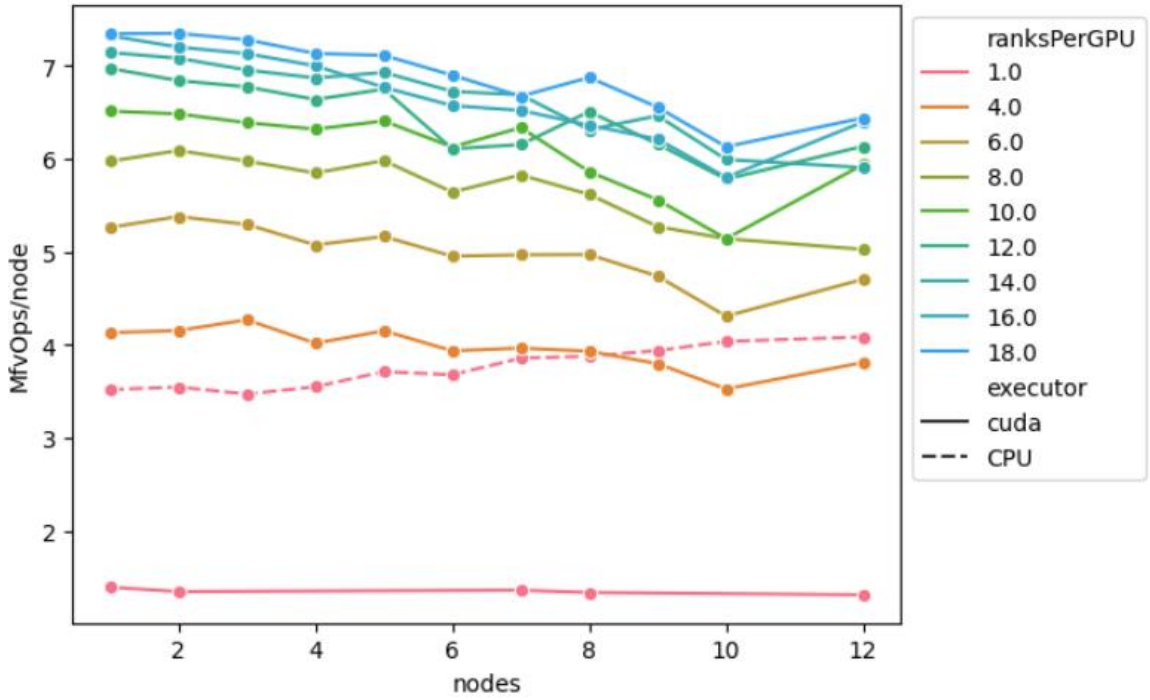
Results Multilevel vs Hierarchical

- Changing partitioning from hierarchical to multilevel alone improved performance by approximately 40%
- This be potentially improved further by (at least for GPUs) by considering mapping of subdomains to devices to maximize bandwidth



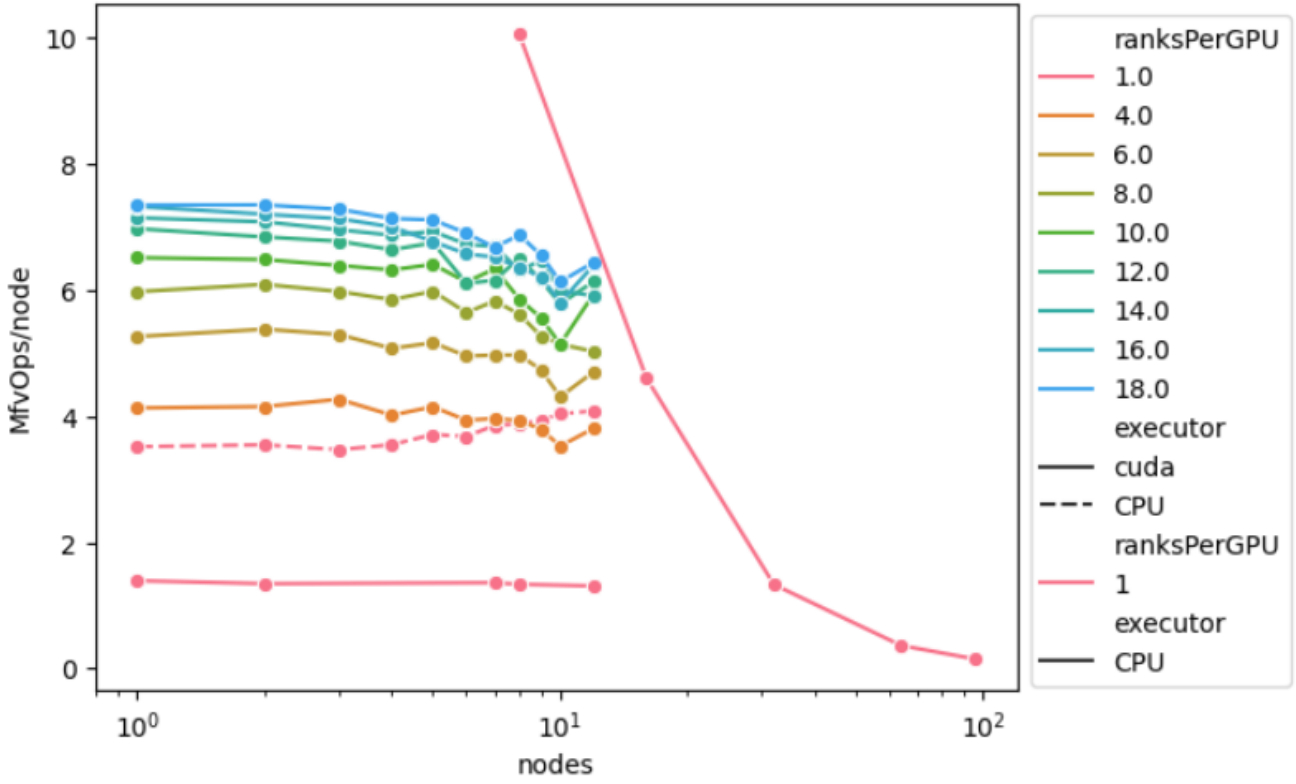
Results Offloading GPU CG solver

- OpenMPI is susceptible to performance degradation when oversubscribing (more than 1 MPI rank per Device) GPUs

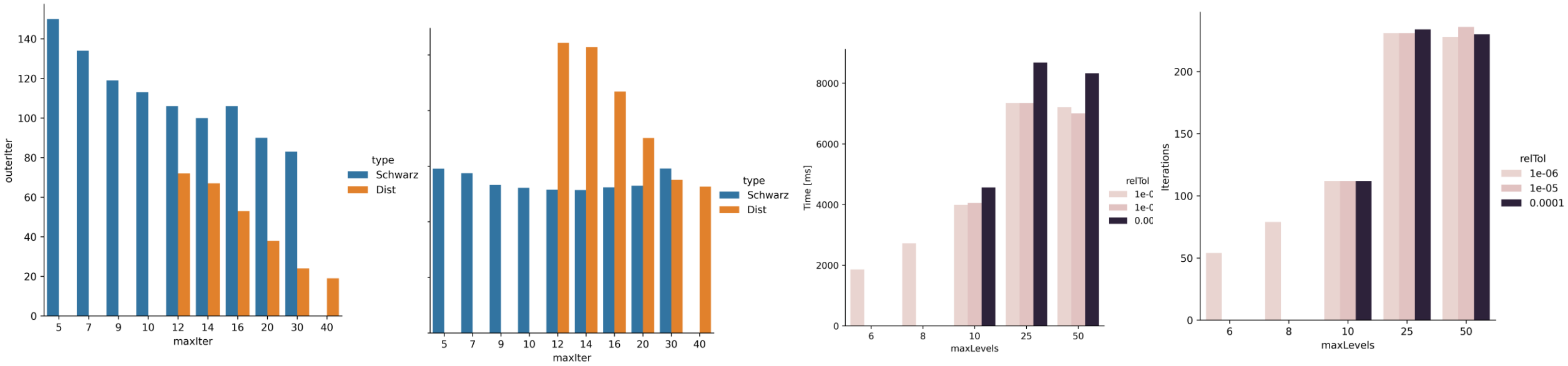


CG vs Multigrid

- Multigrid (GAMG) outperforms any CG implementation (CPU/GPU)
- There might be a range between 16 - 32 nodes, where CG is faster. But in general GPUs profit from more DoF per device.
- Thus the obvious choice would be to use a GPU Multigrid solver/



WIP Tuning Multigrid



- Ginkgo now features different flavours of (algebraic) Multigrid
  - Solver, preconditioner, distributed, local as RAS
- However, the multigrid parameter space is large. CPU based setting wont necessarily work well for GPUs (fewer levels)
- Reuse of geometric restriction and prolongation