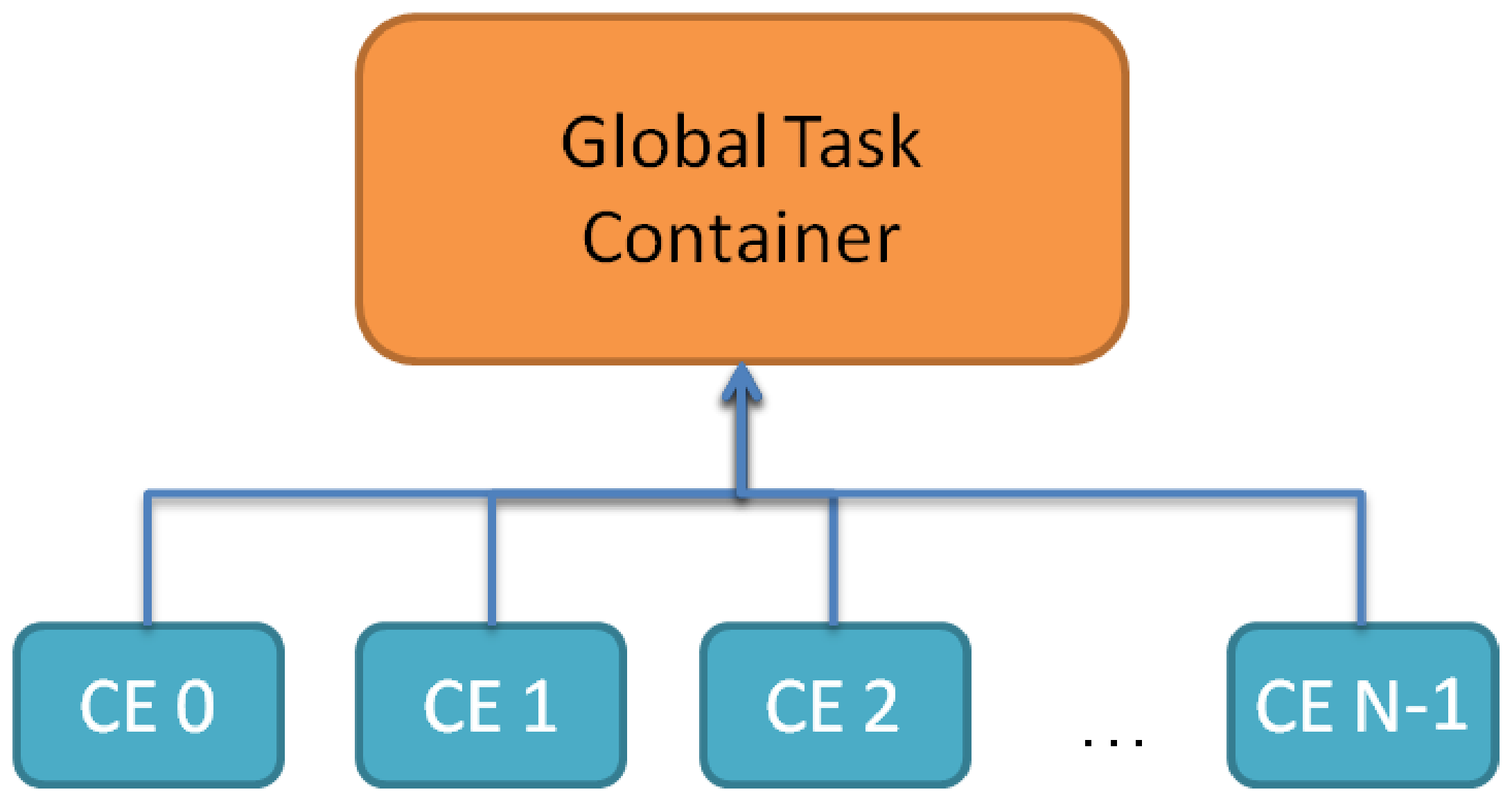
Task-based parallelization on heterogeneous HPC clusters

Jakob Siegel
Electrical and Computer Engineering, University of Delaware, Newark Delaware

Task execution on multiple nodes

- All nodes have access to a global task container (GTC).
- When a compute elements (CE) has, or soon will have computational resources available, it will requests a new task.



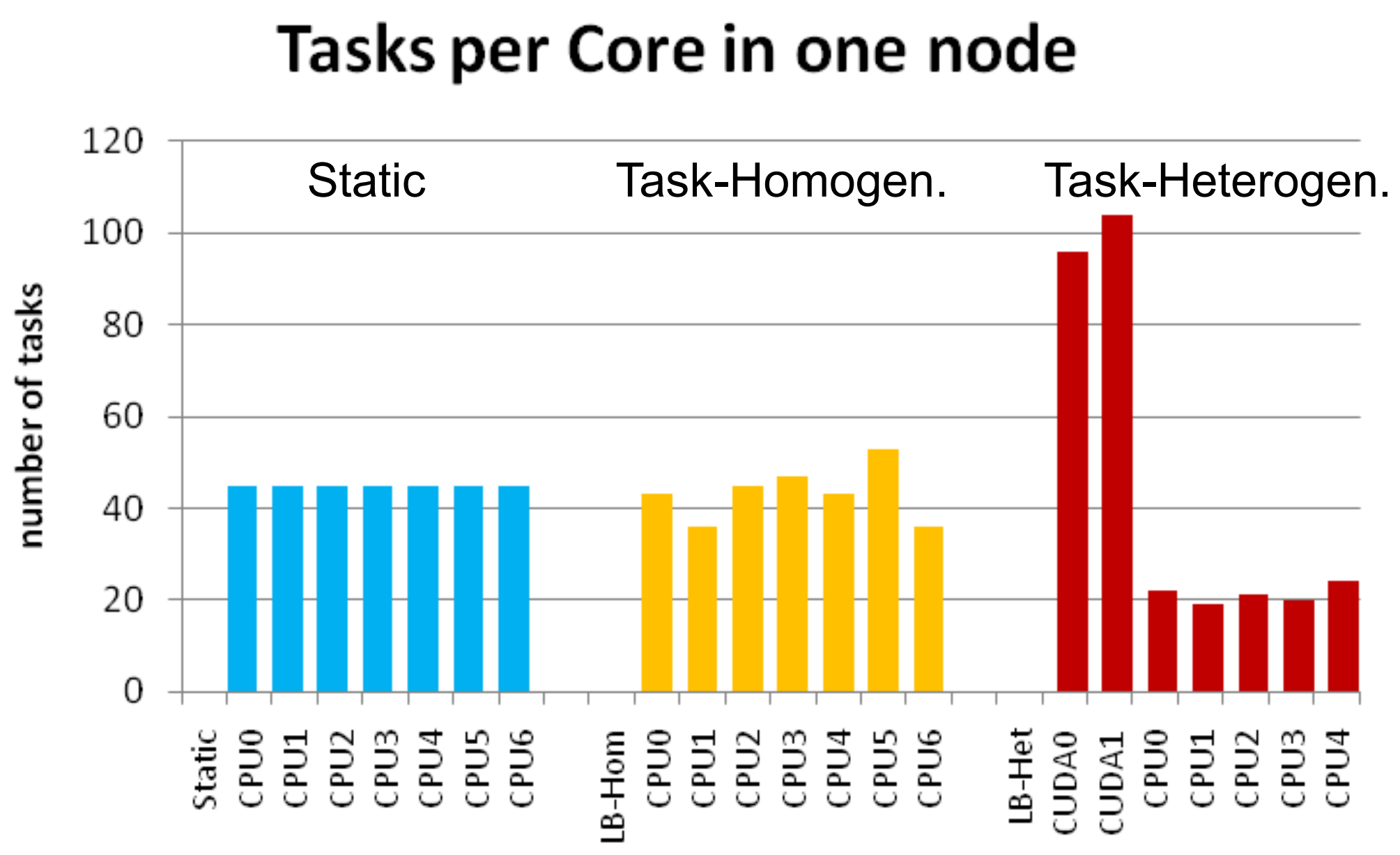
- One task corresponds to a block of computation that can be executed independently.

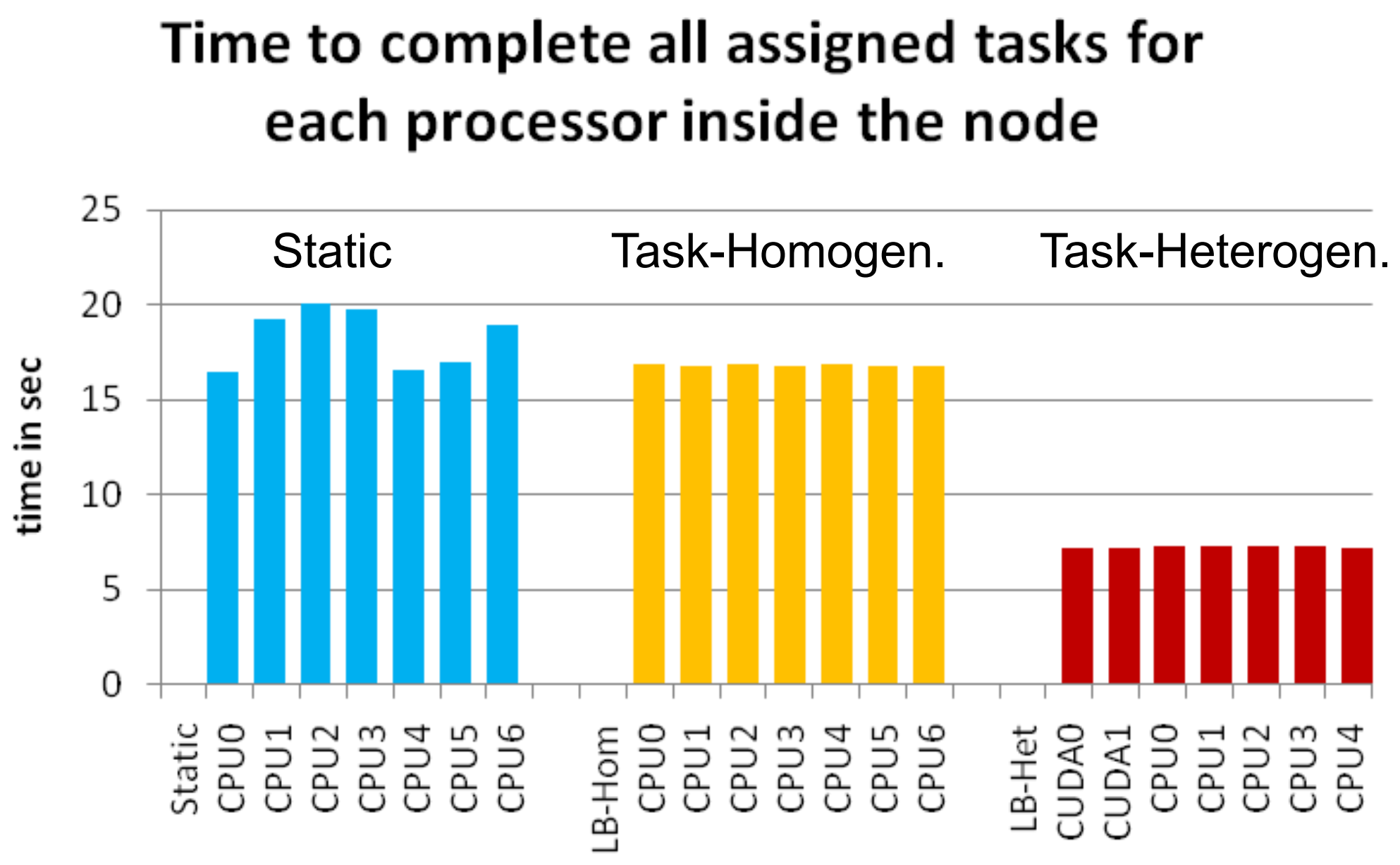
Benefits of a task based execution

- The more tasks are available the better the execution balances itself during runtime without previous need to analyze the topology.
- A task based execution model is not effected by changes to the underlying topology or the usage of different compute elements.

Example: Sparse Matrix calculations

Results for imbalanced computations e.g. Sparse Matrix-Matrix Multiply or various n-body problems show that a tasking approach can easily balance even great differences.





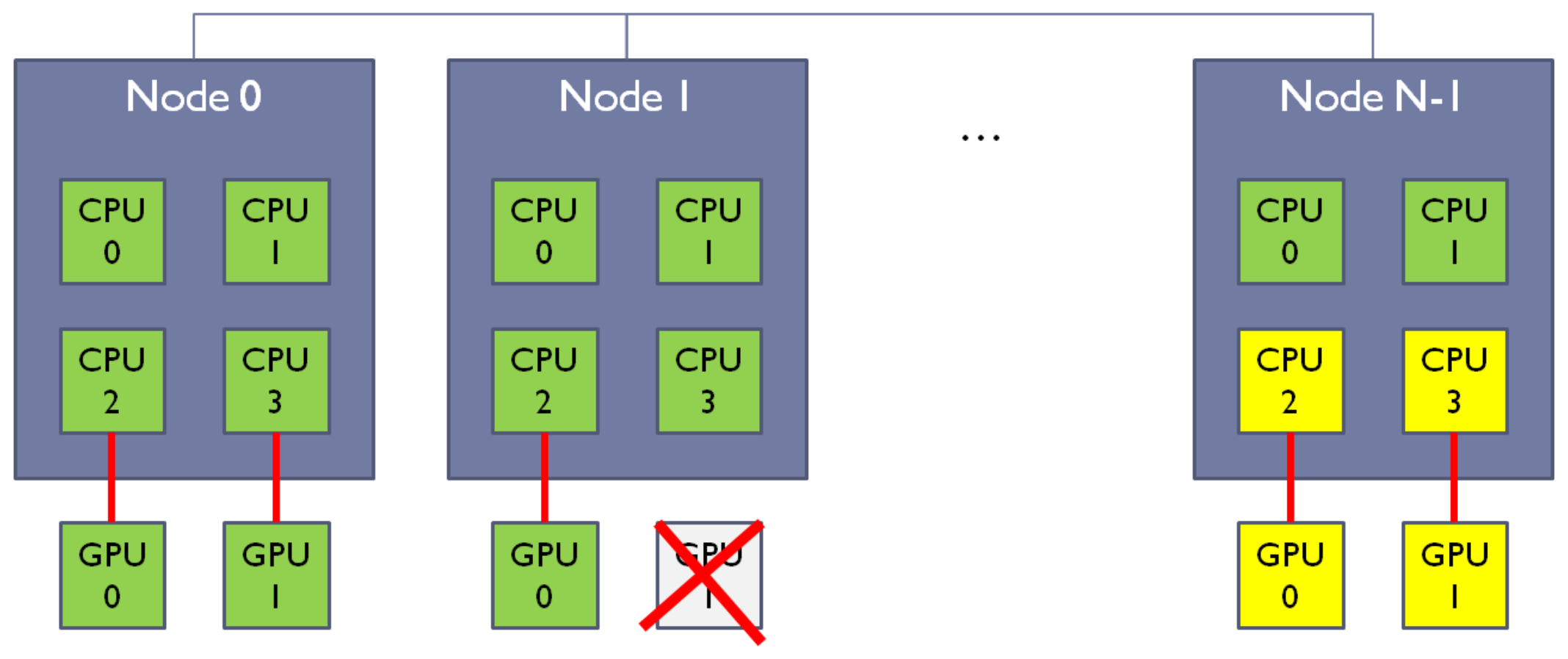
Selected results are shown for a 72,000 × 72,000 sparse matrix, containing 28,715,634 (0.5%) nonzero elements when multiplied with itself. One task represents a 1,024 × 1,024 block of the result matrix. The computation was run in three modes: **Static**: each CPU handles a fixed number of tasks. **Task-Homogeneous**: execution on CPUs in the cluster using tasking. **Task-Heterogeneous**: execution on CPUs and available accelerators using tasking. The figures show results for one of the 8 nodes used on a HPC cluster where each node is a dual quad core Intel Nehalem running at 2.8GHz with 48 GB of memory. Every two nodes share one S1070s Tesla box with 4 GPUs, which gives each node access to 2 GPUs.

How tasking helps with parallel 3D FFT

- To parallelize large 3D FFT in a static, balanced way we usually divide the calculations it into equal portions and assign each portion to a core.
- It can be calculated as 3 1D FFTs for each direction or a combination of a 1D and a 2D FFT, depending on the size and the system.

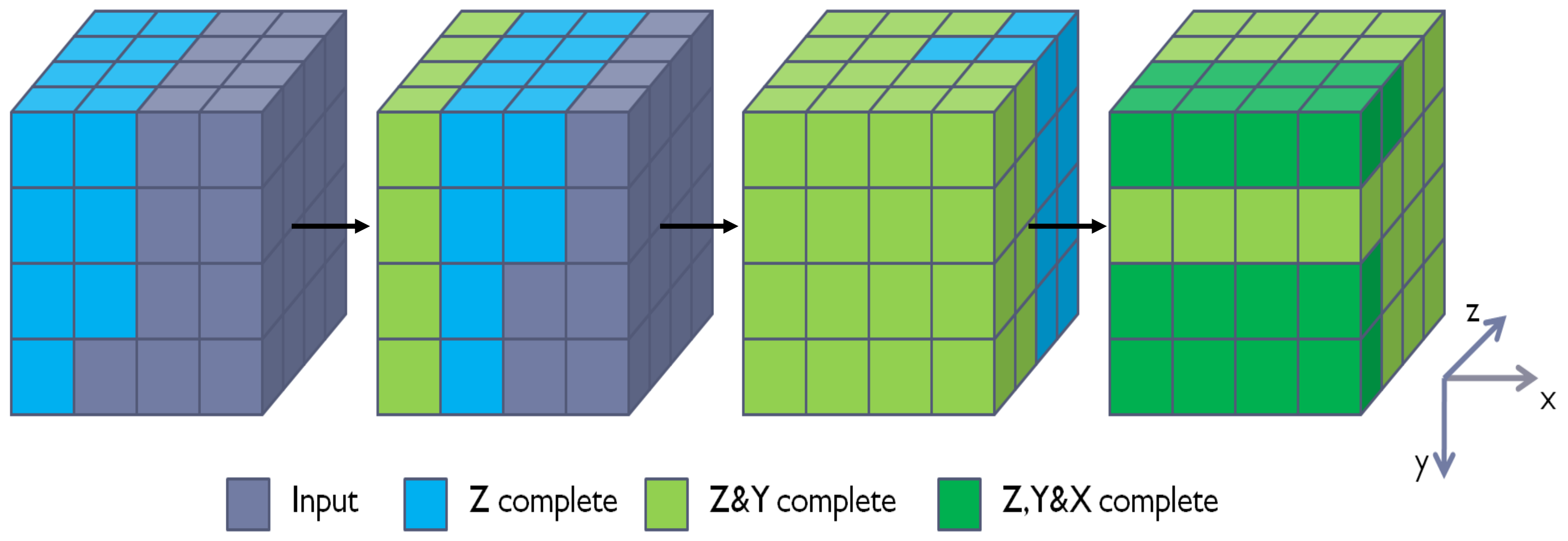
But what if an imbalance is introduced by:

- Working on a heterogeneous cluster (different CEs)?
- No guarantee for exclusive usage of the cluster?
- Resource outages?



Idea: 3D FFT follow up task generation

- Using a tasking approach with fine grain tasks to overlap calculations with communication for the 3 1D FFT approach.
- When a plane for one direction is complete tasks for the next direction can be inserted into the GTC.



- By not having all needed communication taking place at once after a global synchronization or after all equal length computations are completed the interconnect bandwidth will not be saturated.
- Overlapping different data access patterns for different types of tasks furthermore might reduce communication bottlenecks

This work has been made possible by the generous support of the NSF through research grants
OCI-0904534 and CCF-0925863

Jakob Siegel, Xiaoming Li