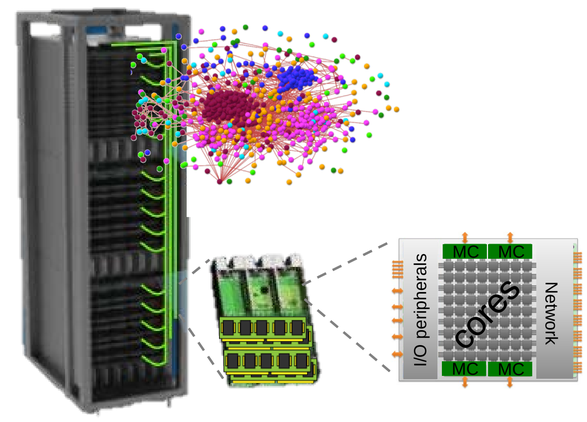
Chip Design for Rack-Scale Computing

Alexandros Daglis[†] Stanko Novakovic[†] Edouard Bugnion[†] Babak Falsafi[†] Boris Grot[‡]

[†]EcoCloud, EPFL [‡]University of Edinburgh

Application and technology trends

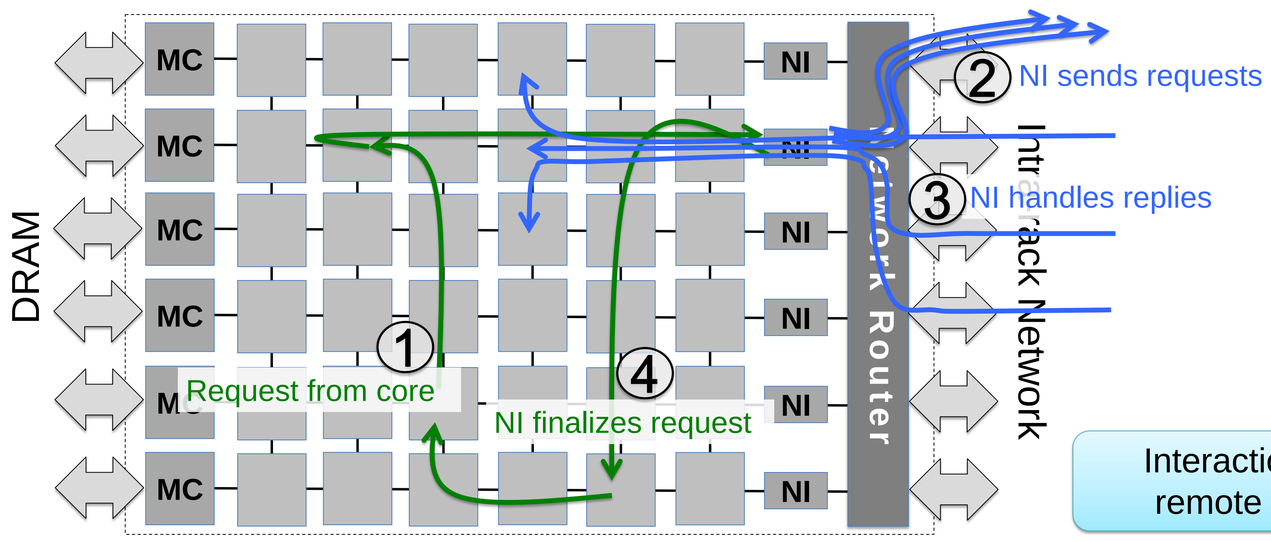
- Applications: huge distributed datasets, need low latency/high bandwidth (e.g., KV-Stores)
- Server chips: many cores for high throughput
- Rack-scale systems: vast memory pool, fast remote memory access
  - High-bandwidth interconnect
  - Fully coherent *integrated NIs* (memory-mapped core-NI interaction)



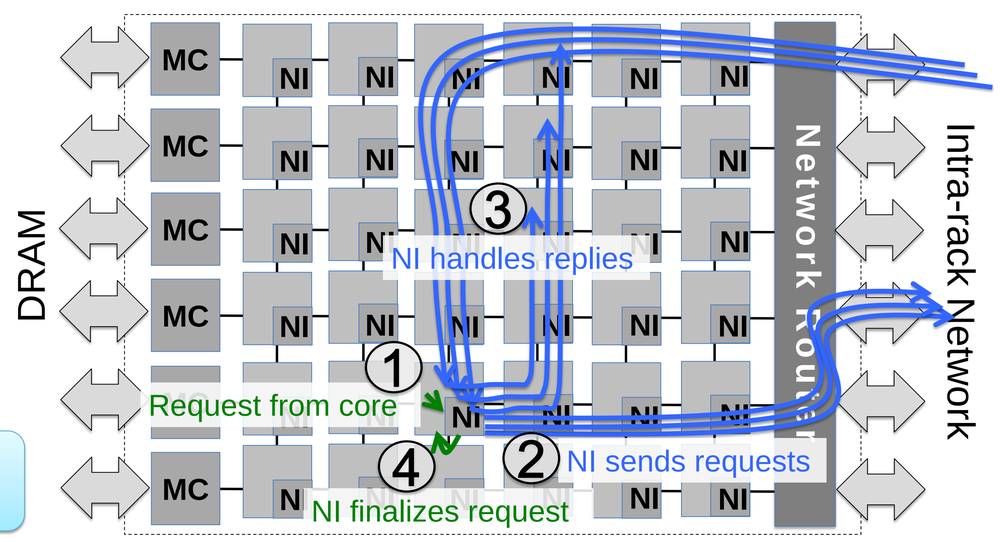
NI integration on chip critical for low-latency, high-bandwidth remote memory access

NI placement considerations

- Single, centralized NI can't meet increasing communication demands; NIs need to scale with core count
- Baseline design: Scale NIs across chip's edge (Edge NI)
- Latency-optimized design: One NI per core/tile (Per-tile NI)



Interactions for a remote read op



✗ Coherence-induced latency overhead for core-NI interactions

✓ Efficient bandwidth utilization for data transfers

Up to 50% of end-to-end latency spent on on-chip interactions

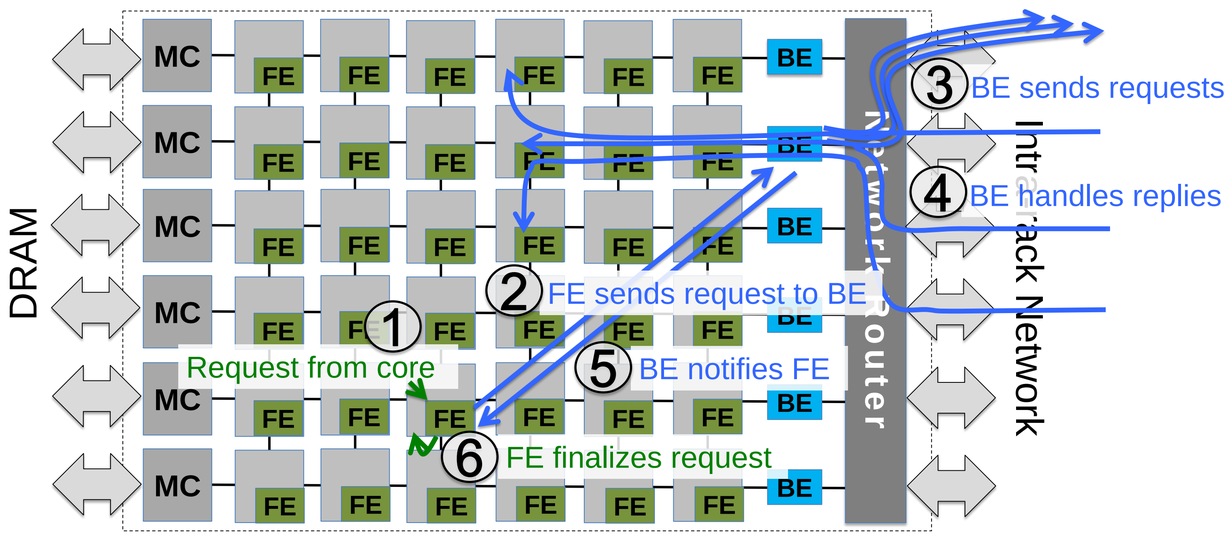
✓ Minimal latency introduced by core-NI interaction

✗ NI acts as point of indirection for data → on-chip bandwidth misuse

Peak bandwidth 75% lower than Edge NI

Our proposal: Split NI

- **Key insight:** Can split core-NI interactions from data handling
  - Split NI into Frontend (FE) and Backend (BE)



How does Split NI deliver both low latency and high bandwidth?

- Separate core-NI interaction from data handling
- All core-NI interactions localized
- All data handling at the edge – no on-chip points of indirection

Best of both worlds!

Low-latency core-NI interactions & high-bandwidth data transfers

Evaluation

- Full-system, cycle-accurate simulation on Flexus

Latency comparison

Latency component	HW NUMA	Edge NI	Per-tile NI	Split NI
Core-NI interactions	24	172	43	
Network + NI processing	70	79		80
Remote DRAM	104			
Total (ns)	**198**	**355**	**226**	**227**

Latency breakdown: single-block remote read op, single network hop, 64-core chips

Bandwidth comparison