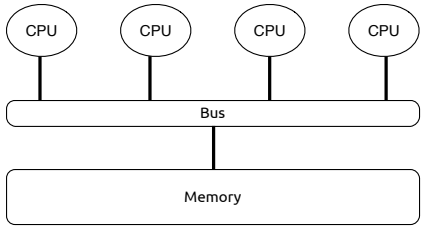
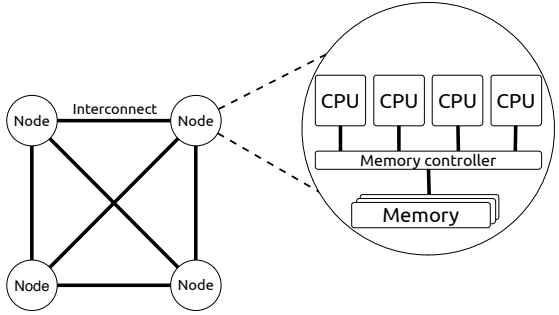
NUMA – Herausforderungen paralleler Systeme

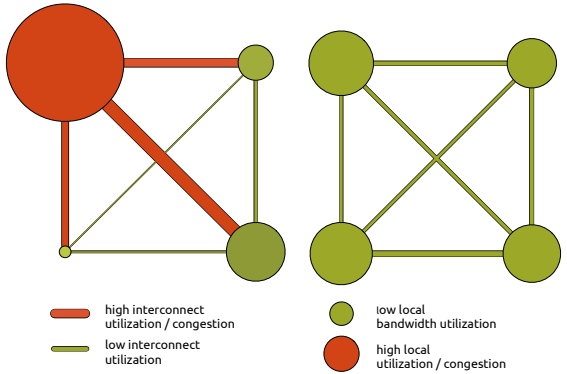
Prof. Dr. Andreas Polze
Felix Eberhardt
Frank Feinbube

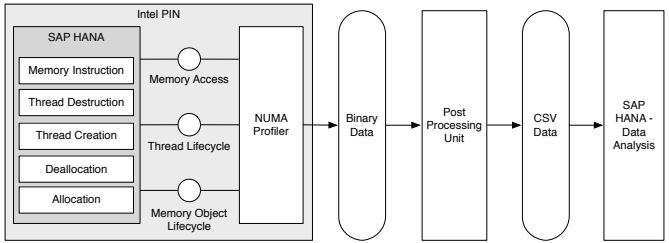
TOC

- Seminar
- Multiprocessor Machines
- NUMA Challenges
- Topics









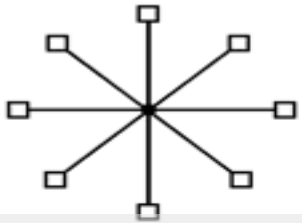
The seminar

- Extend: 2 SWS, 3 graded CP
- Dates: Wednesday 11.00 – 12.30
- Room: HS 3
- Enroll: 24.10.

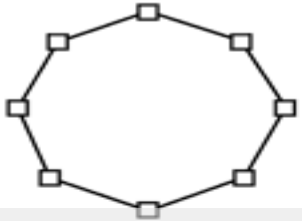
Your task

- Choose a topic
- Read related work
- Evaluate different approaches
- Implement or test a prototype/tool
- Present to fellow students (~30 min.)

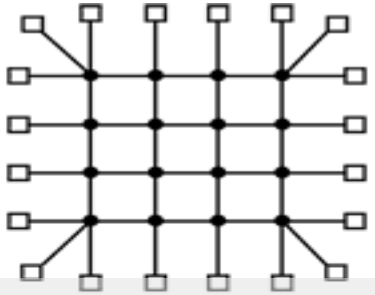
Introduction



(a)

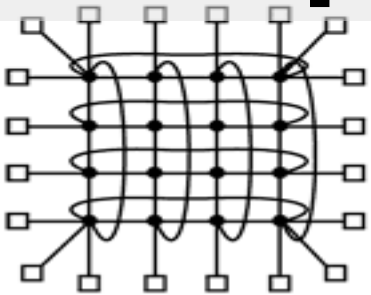


(b)

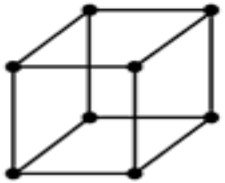


(c)

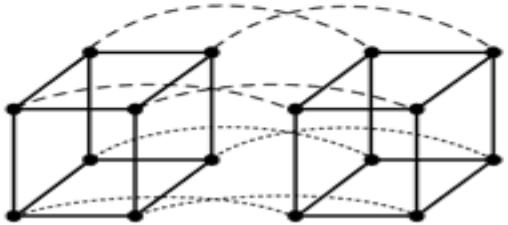
Multiprocessor Machines



(d)

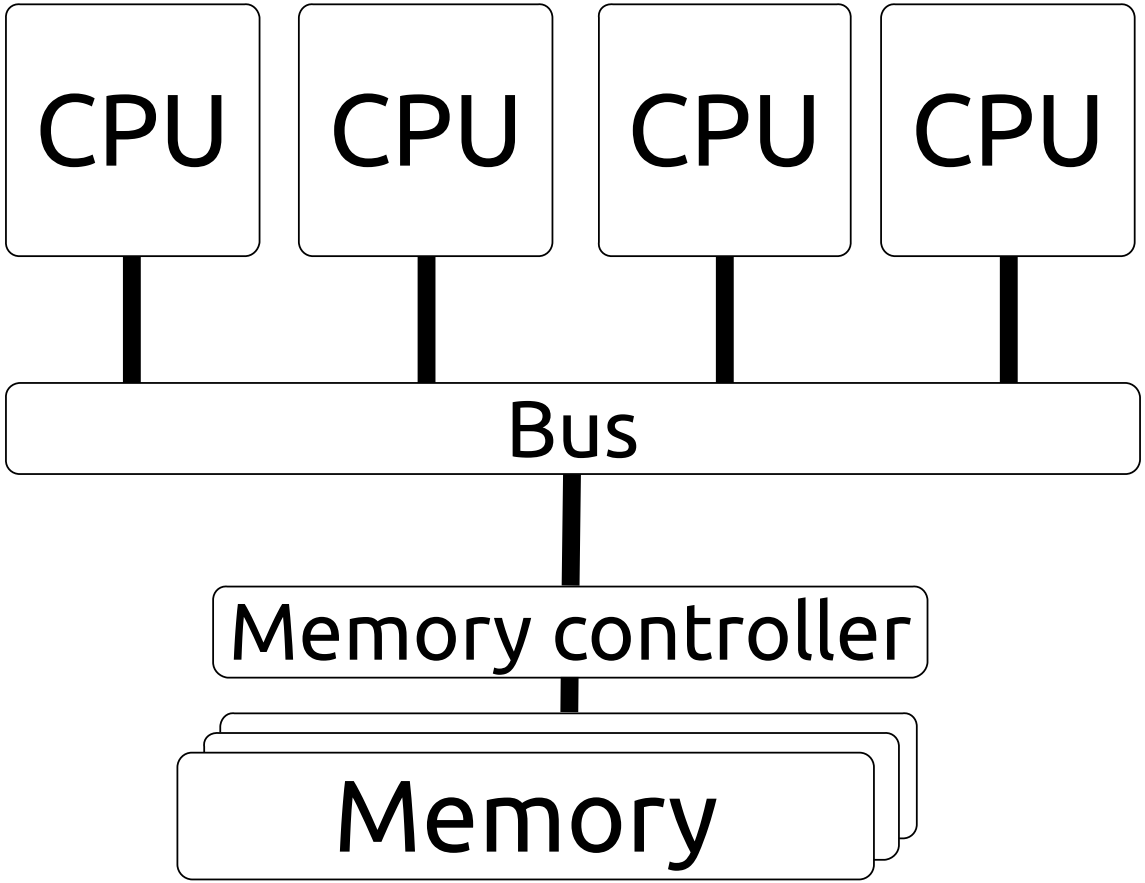


(e)

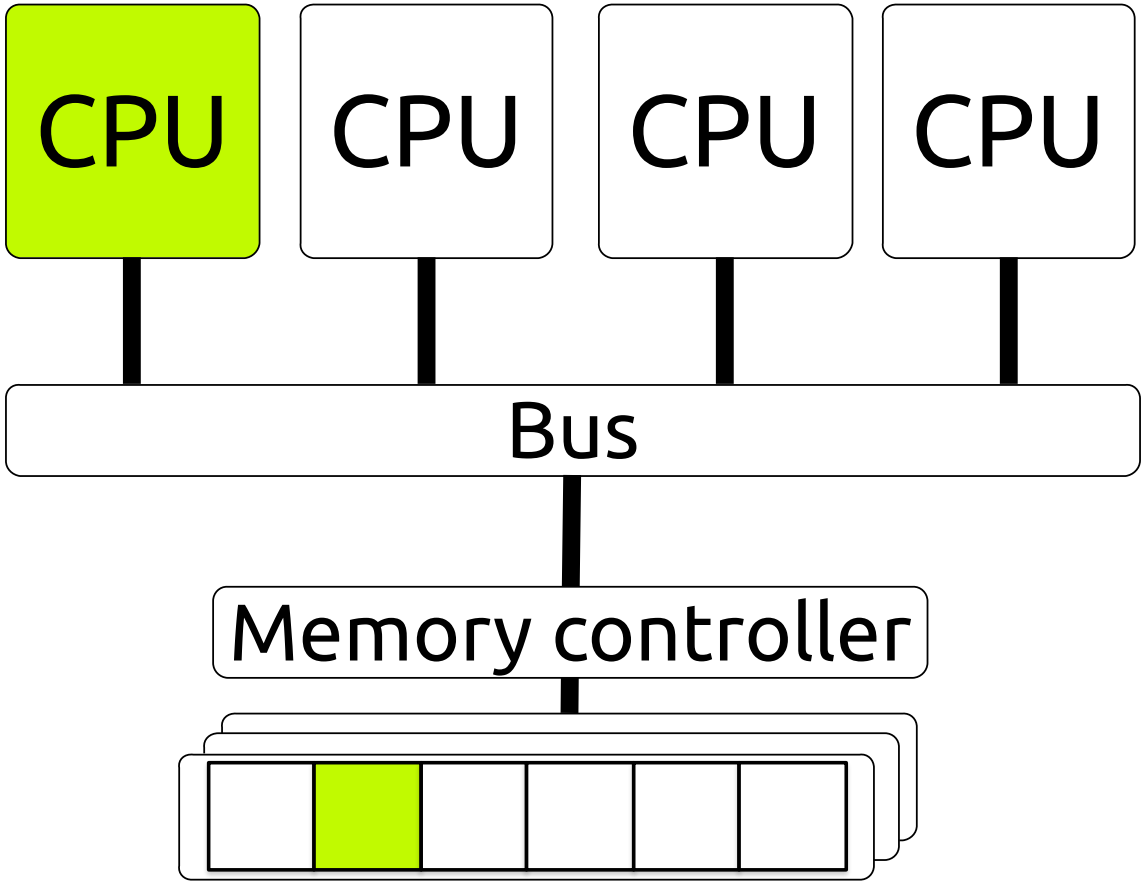


(f)

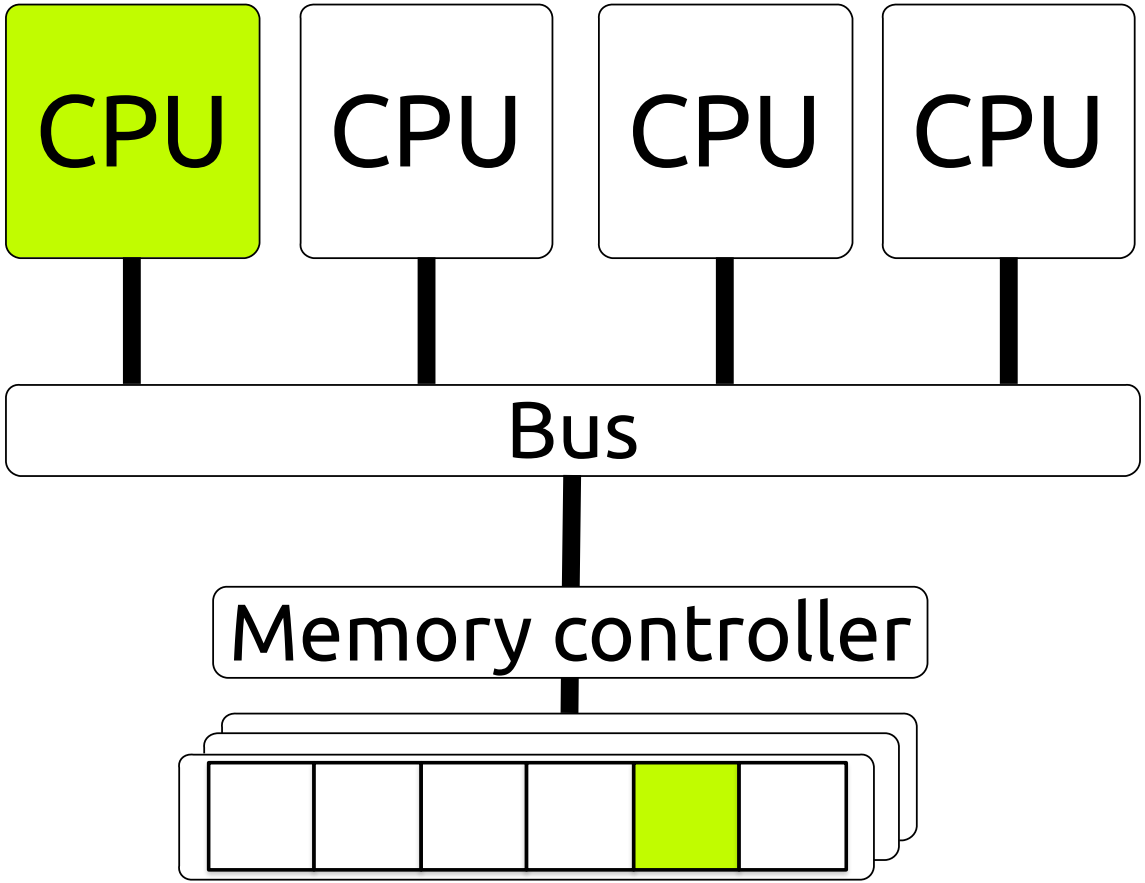
Uniform **M**emory **A**ccess



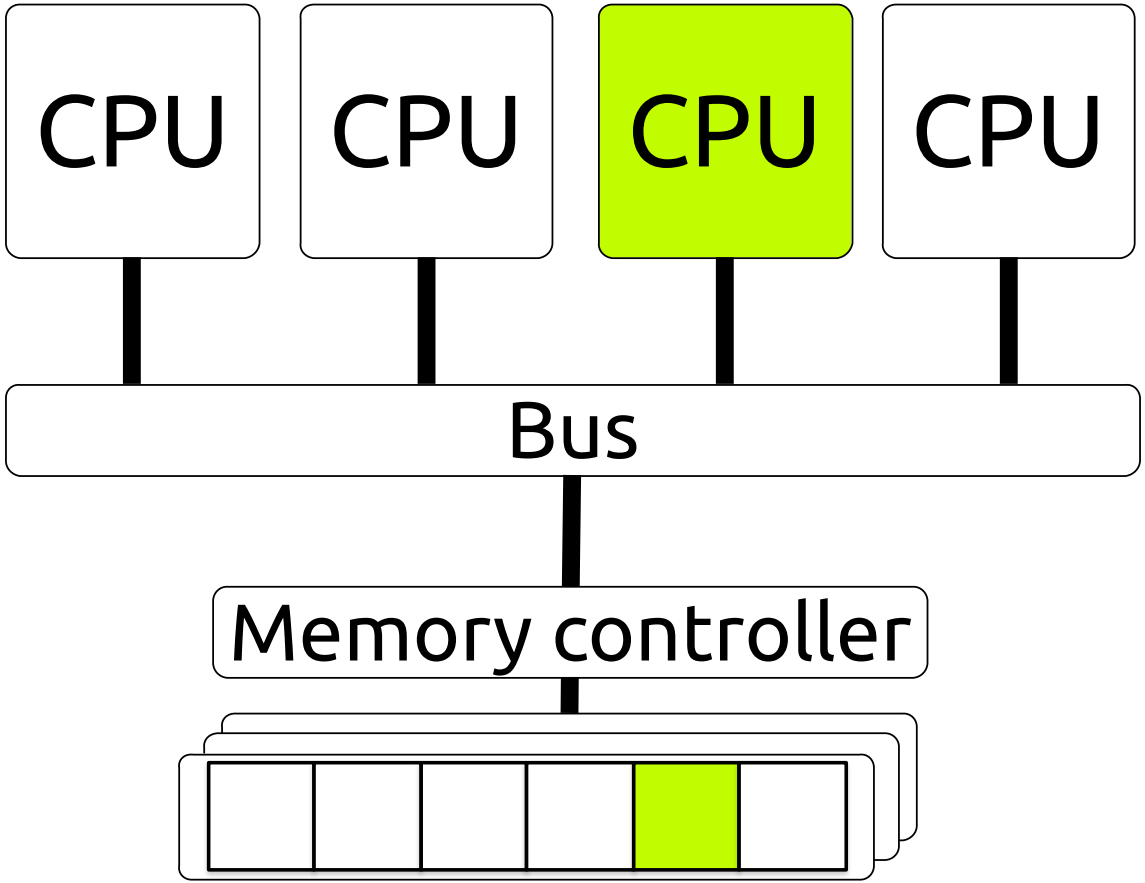
Uniform **M**emory **A**ccess



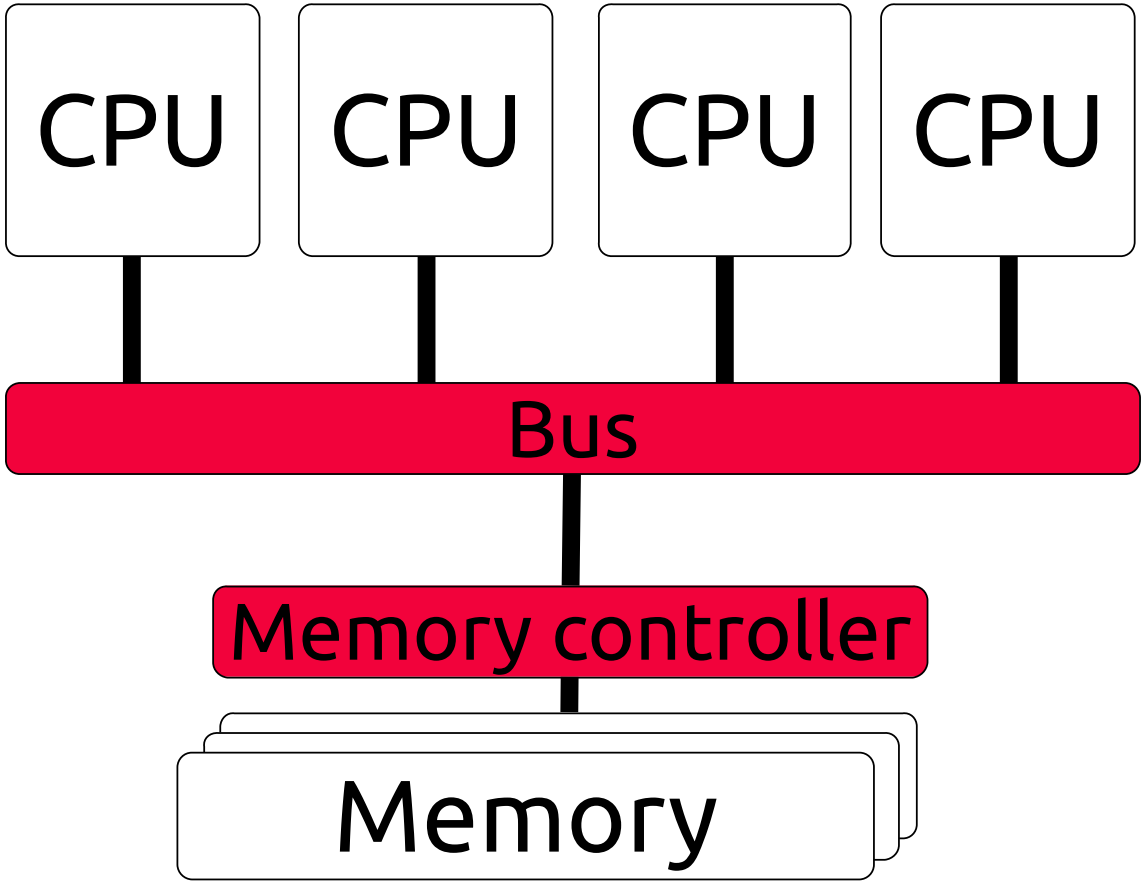
Uniform **M**emory **A**ccess



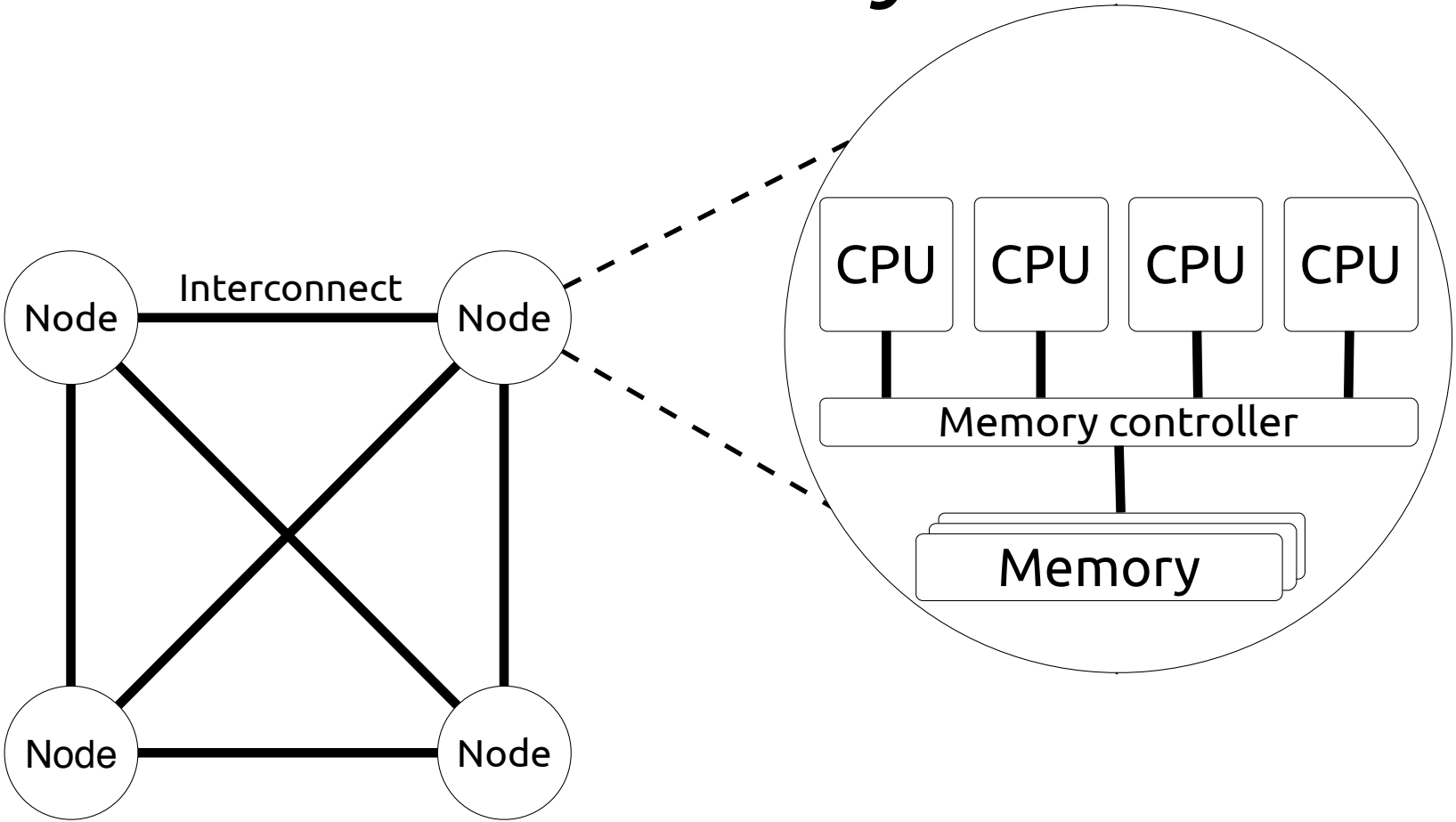
Uniform **M**emory **A**ccess



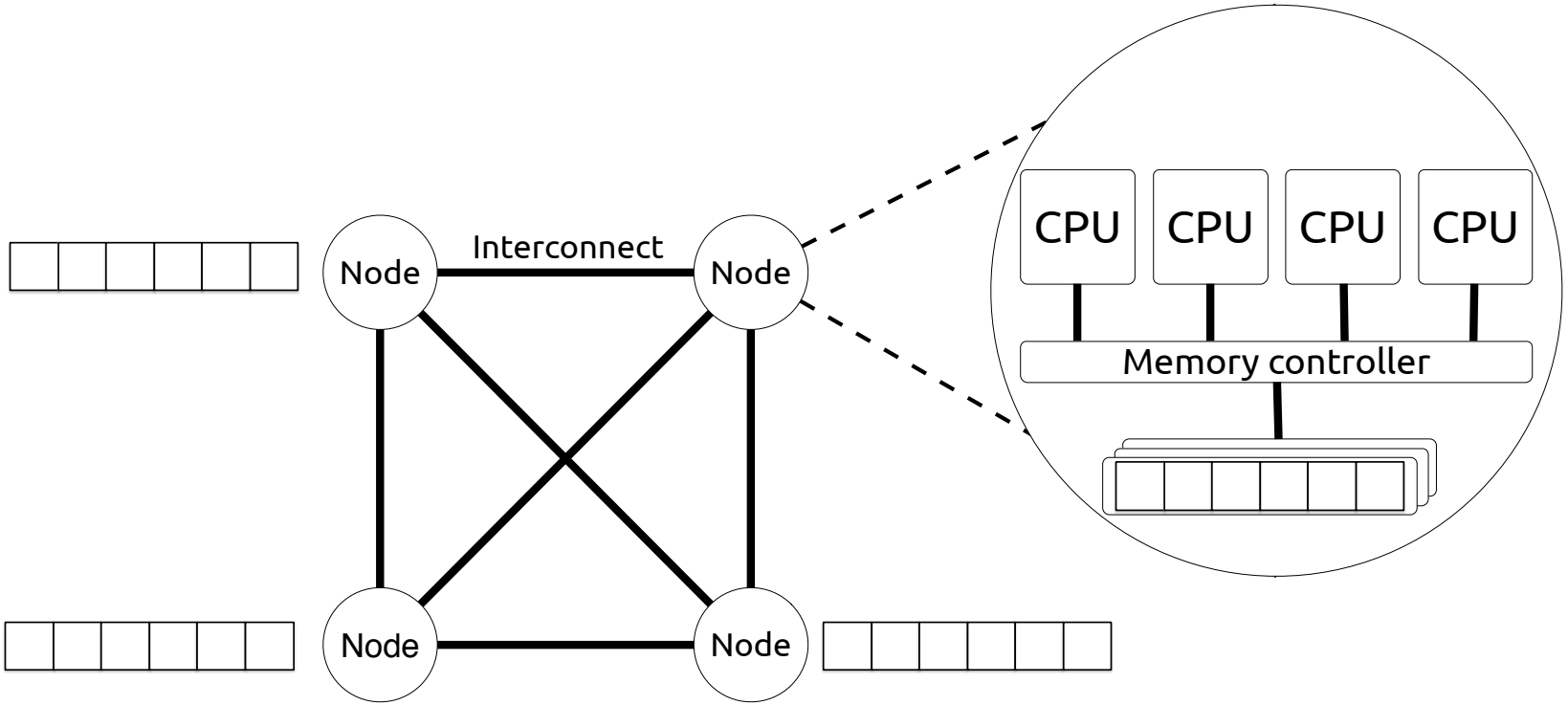
Uniform **M**emory **A**ccess



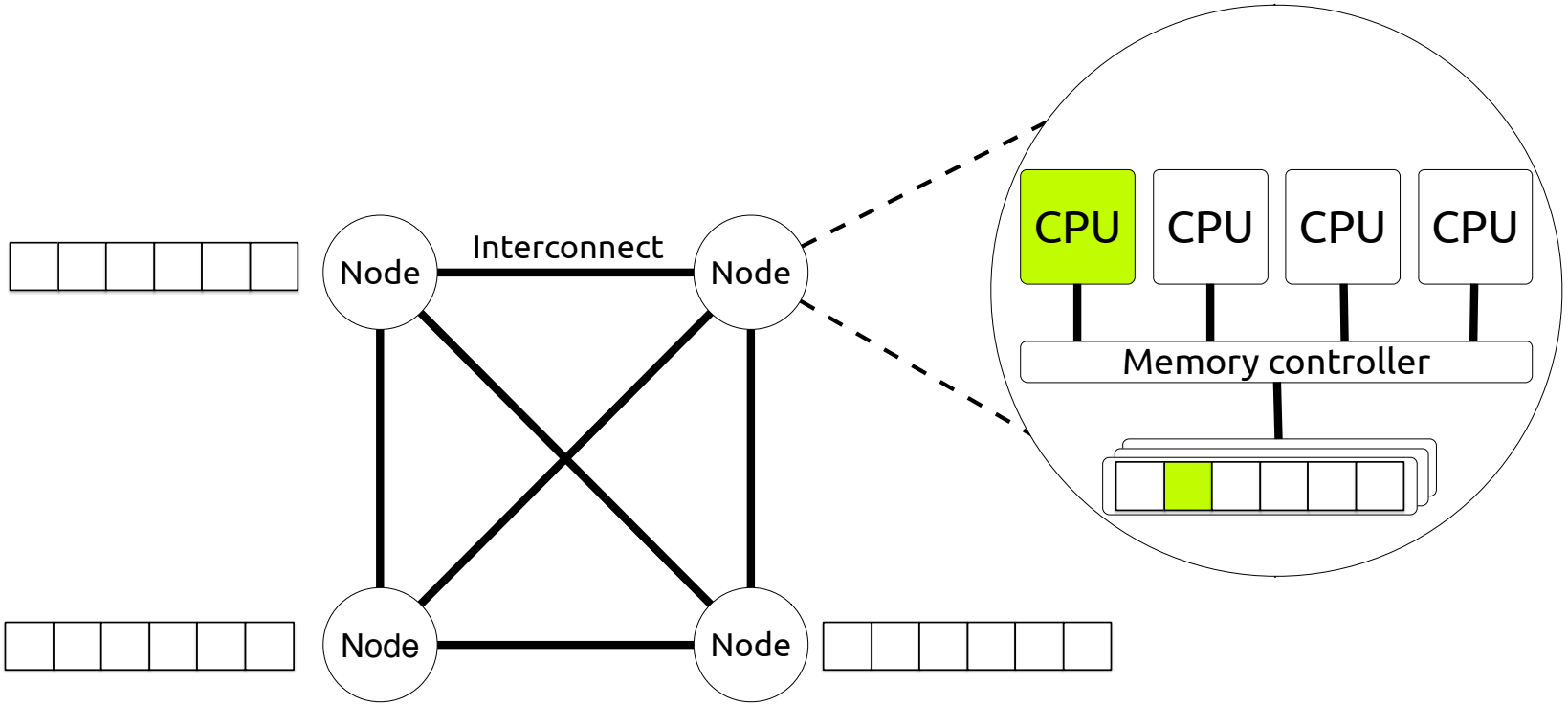
Non **U**niform **M**emory **A**ccess



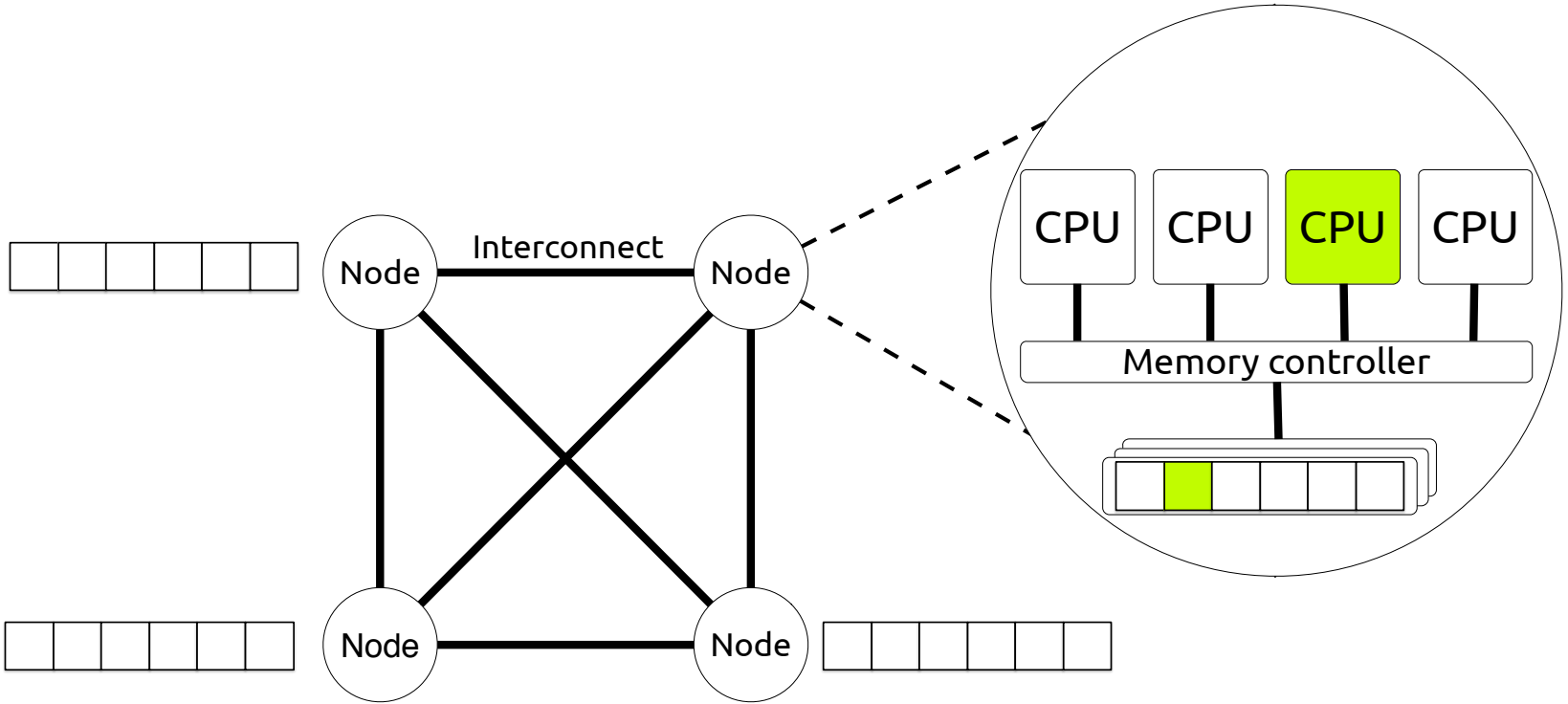
Non **U**niform **M**emory **A**ccess



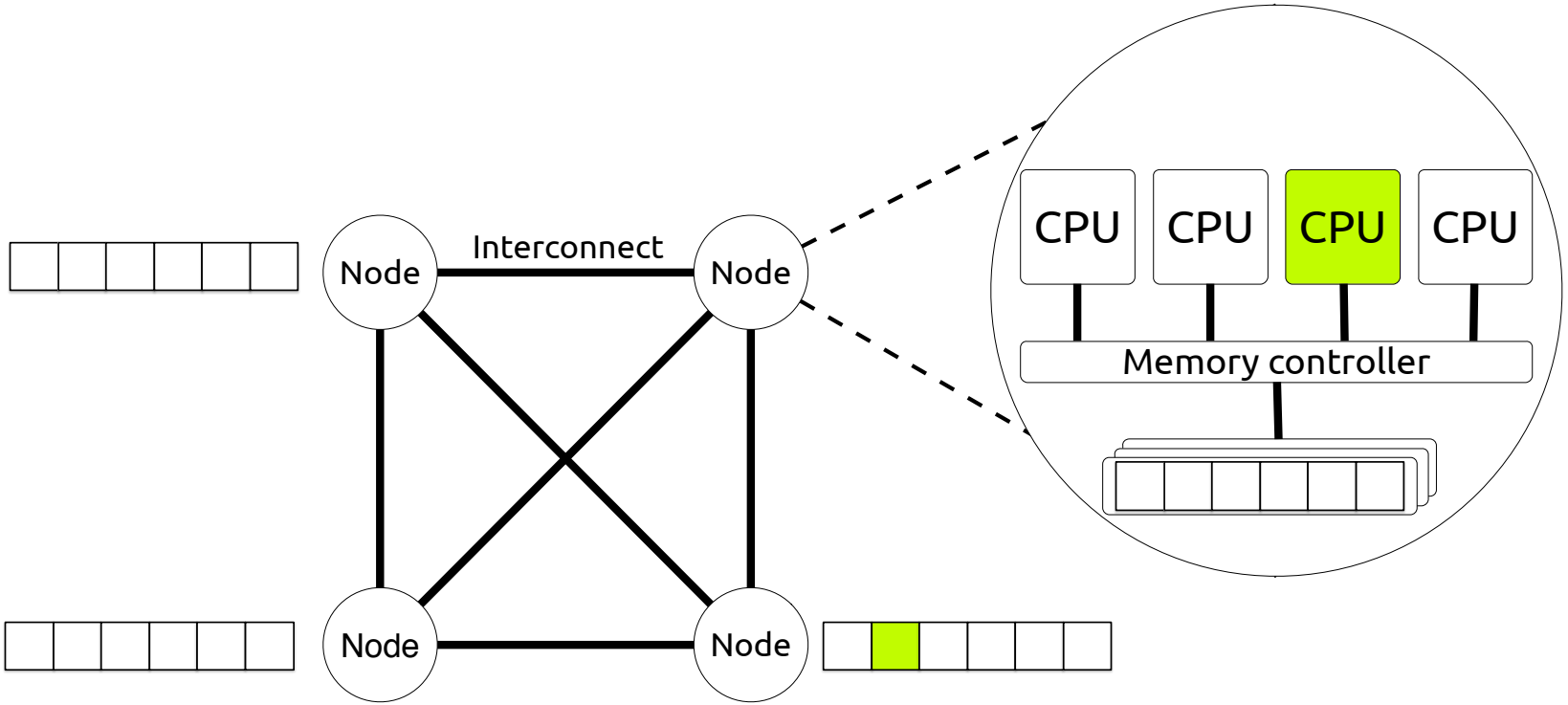
Non **U**niform **M**emory **A**ccess



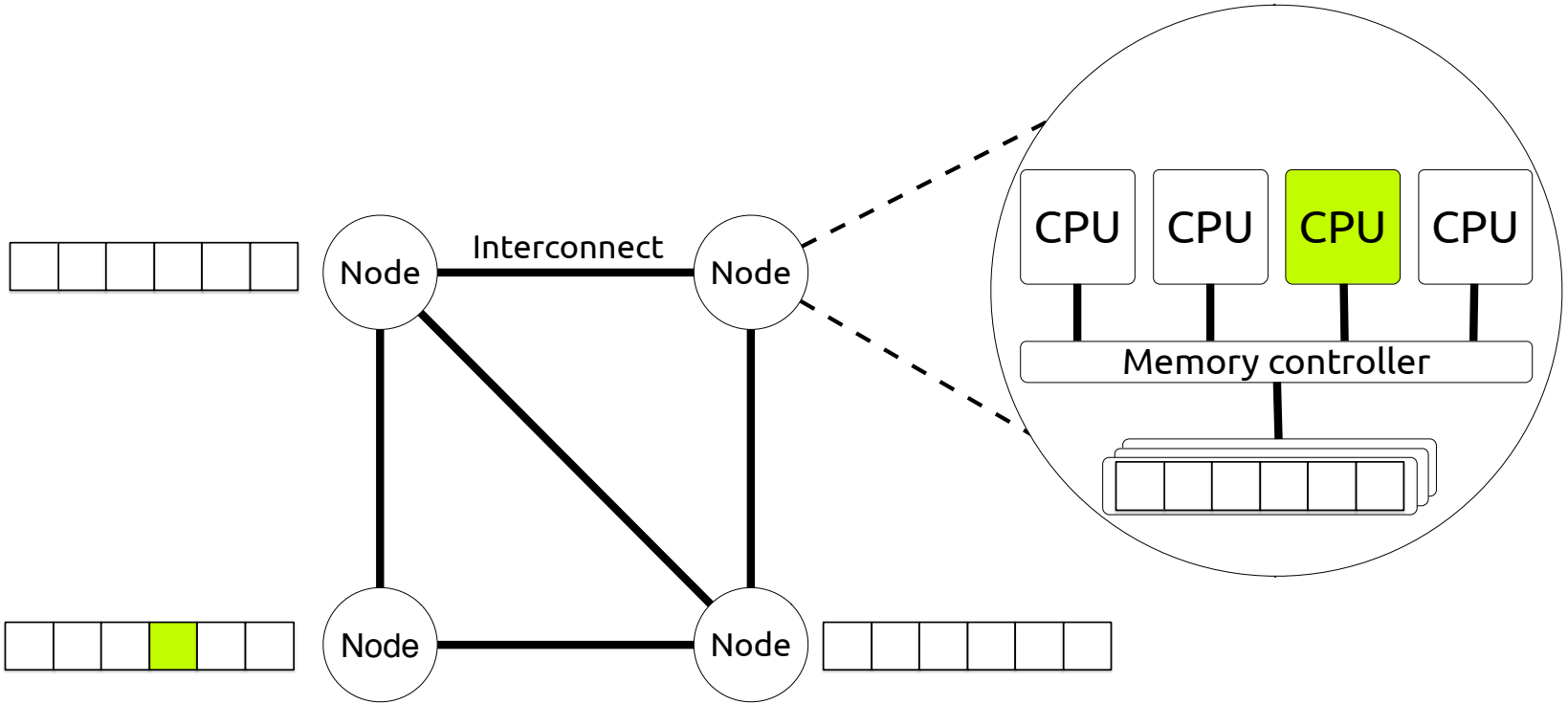
Non **U**niform **M**emory **A**ccess



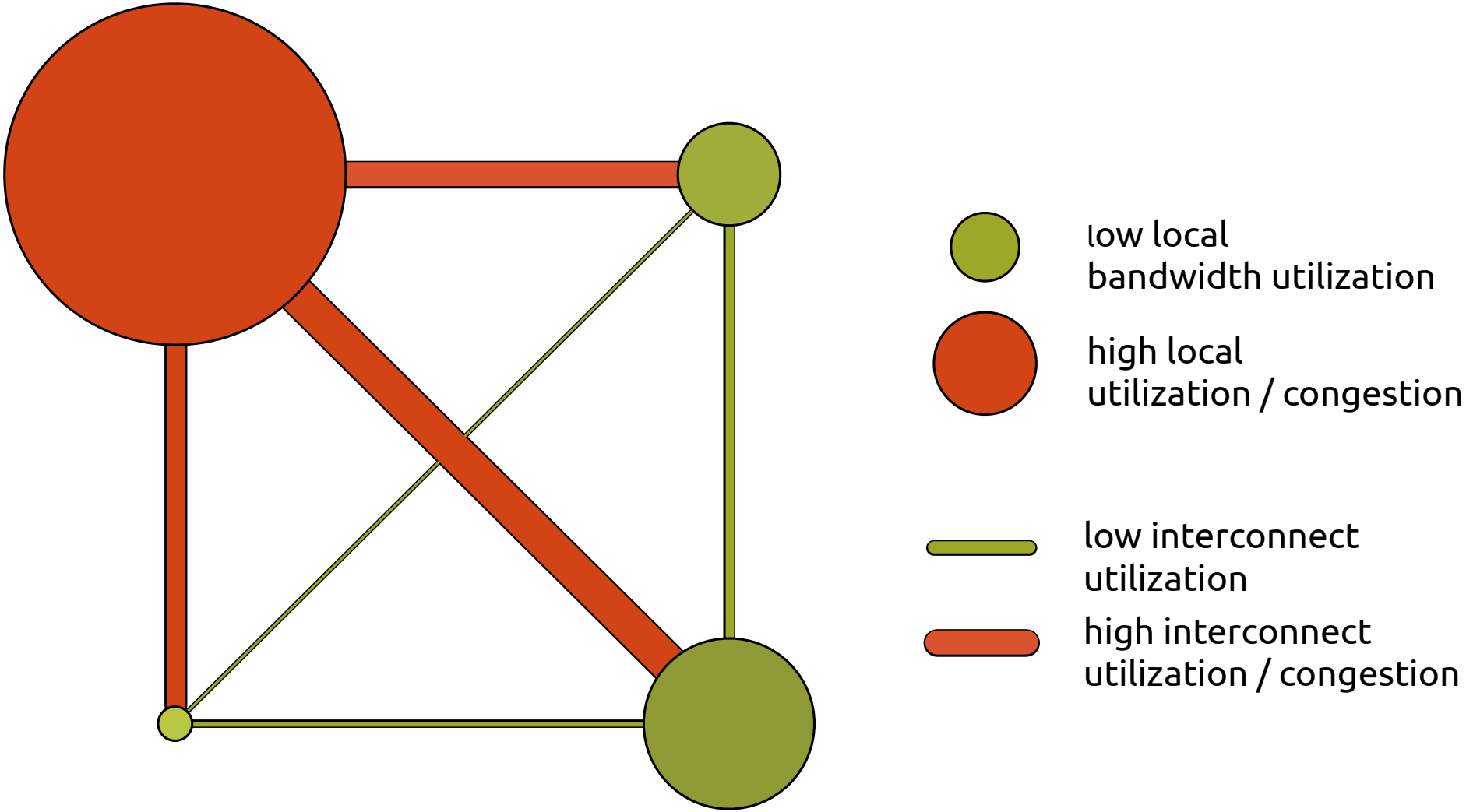
Non **U**niform **M**emory **A**ccess

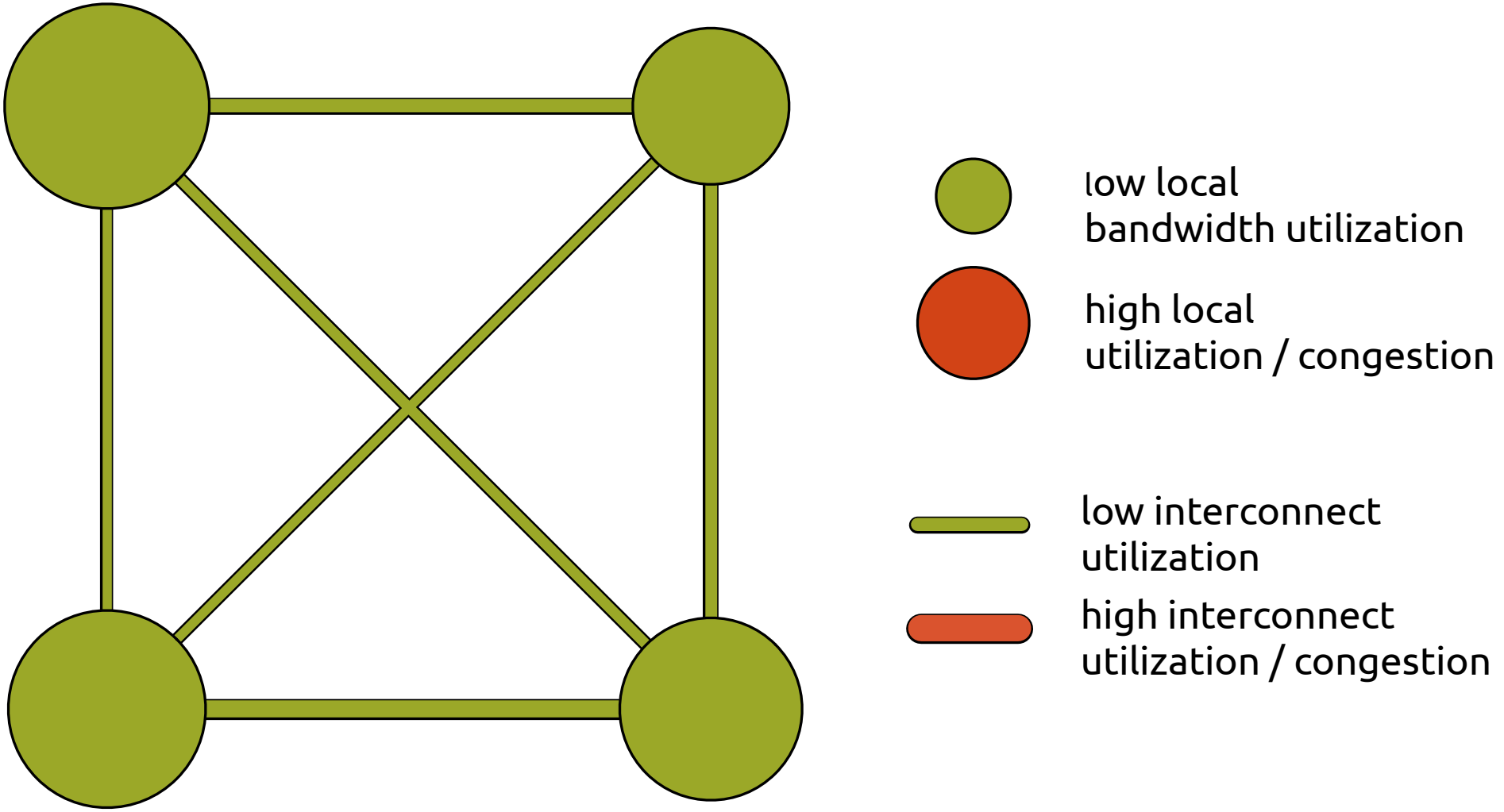


Non **U**niform **M**emory **A**ccess



NUMA Challenges


low local
bandwidth utilization
high local
utilization / congestion
low interconnect
utilization
high interconnect
utilization / congestion


low local
bandwidth utilization
high local
utilization / congestion
low interconnect
utilization
high interconnect
utilization / congestion

Machine Topology

Decisions for placement based on:

- Hierarchy of **memory**
- **Shared** resources
- **Interconnect** network
  - Communication **costs**
  - Maximal **bandwidth**

Thread Placement

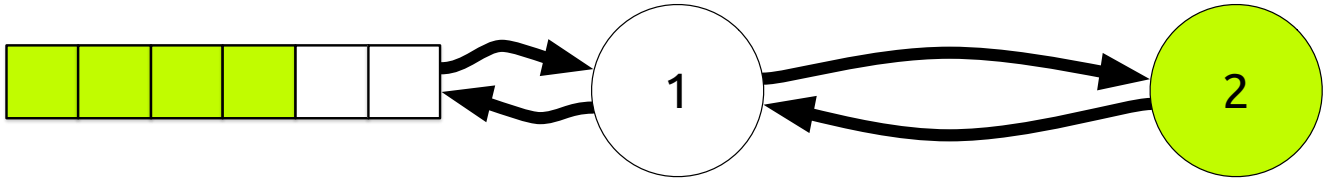
- **Tradeoff**: load balancing vs. data locality

- Affinity of threads
- Thread-pinning
- Process follows memory
  - Data access patterns

Data Placement

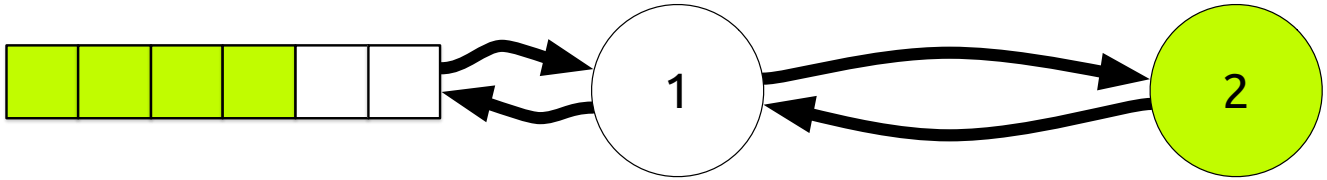
- Static (allocation time)
  - **First-touch**
  - Allocate on specific node
  - Interleaving
  - Replication
  - …
- Dynamic (throughout runtime)
  - Memory follows process strategies

Data Access Patterns

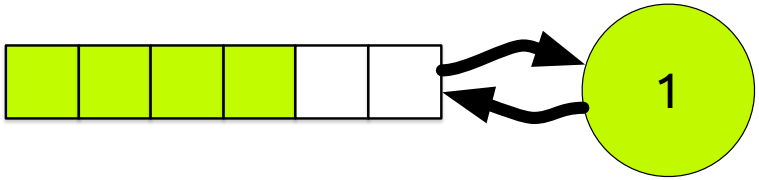


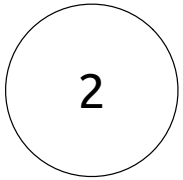
Data Access Patterns

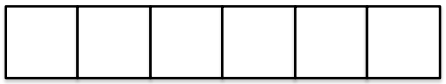
Data Access Patterns

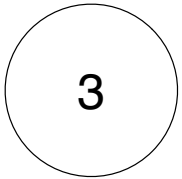


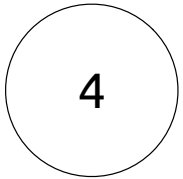
Data Access Patterns



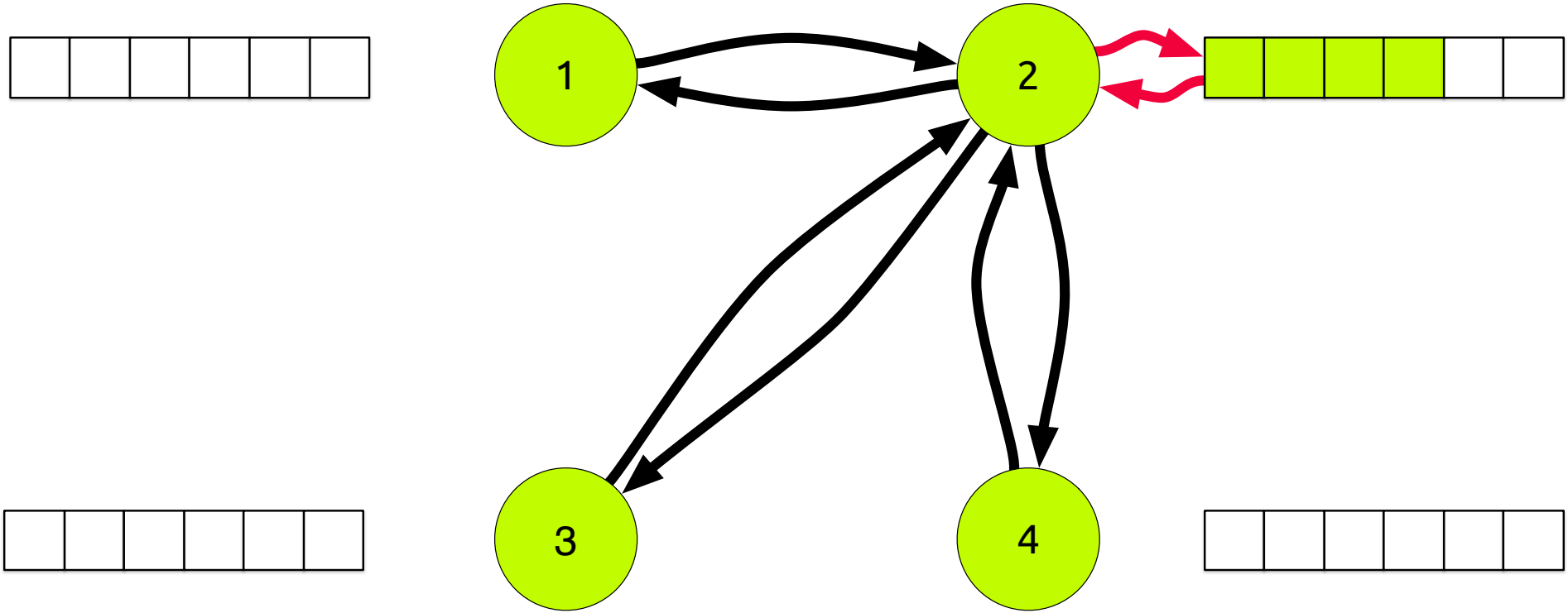




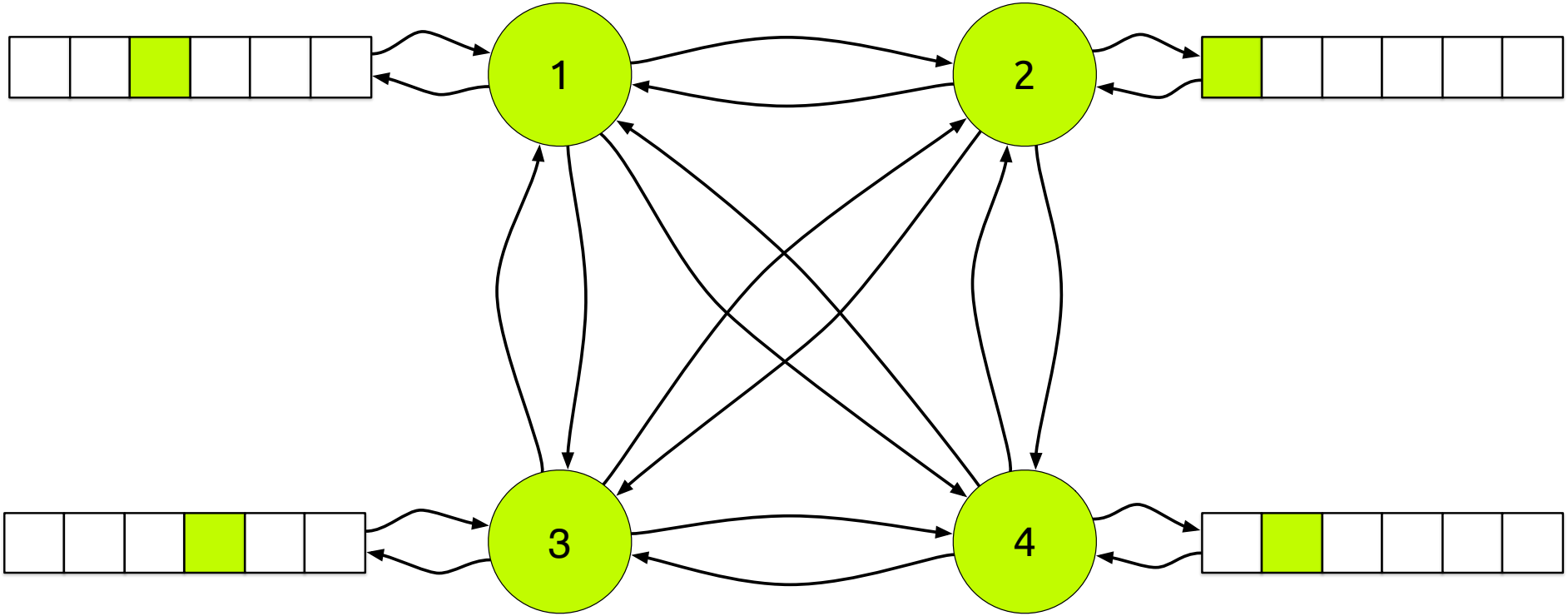




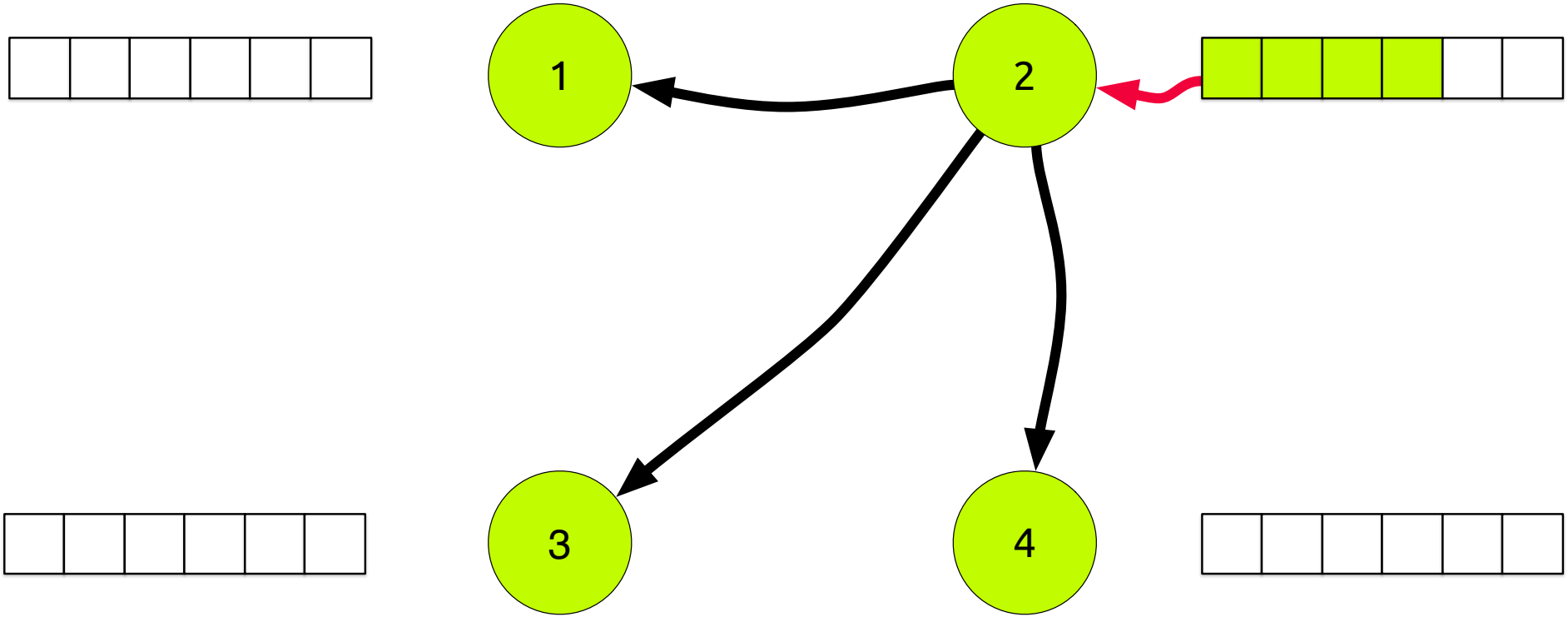
Data Access Patterns



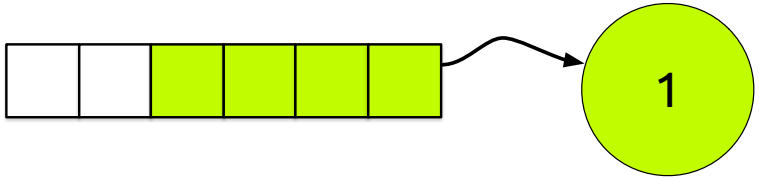
Data Access Patterns

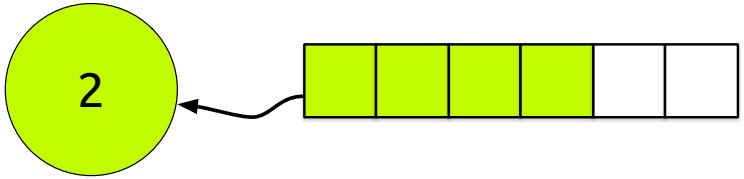


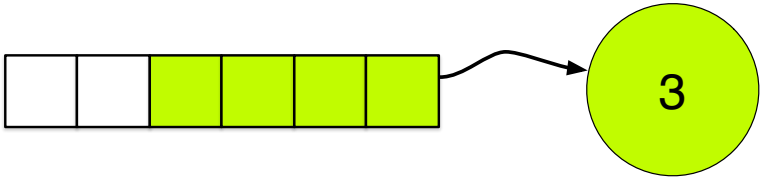
Data Access Patterns

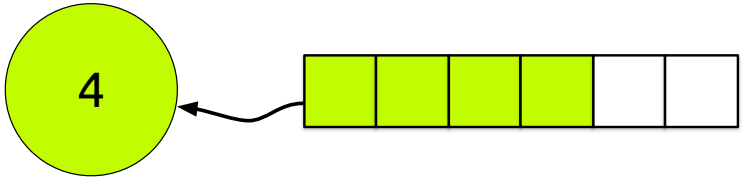


Data Access Patterns









Analyzing Runtime Behavior on NUMA Systems

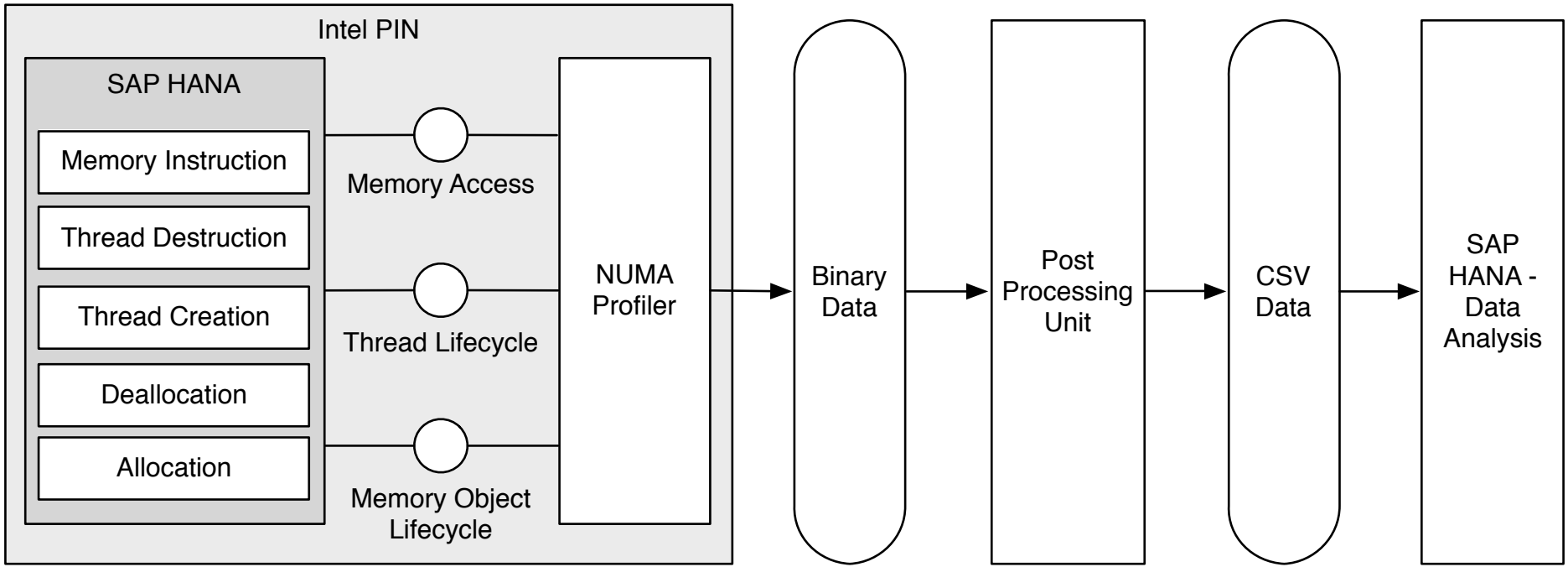
NUMA4HANA Analyzer

- Detecting **data access patterns**
- **Current profilers** cannot analyze data placement sufficiently for HANA
  - No knowledge about data allocations
  - No knowledge which threads access which C++ object, array, table, …

Data Accesses

- Instrument every **load and store** operation
- Pass **effective address** to analysis routine
- Analysis routine:
  - Call move_pages() to obtain NUMA node

Architecture



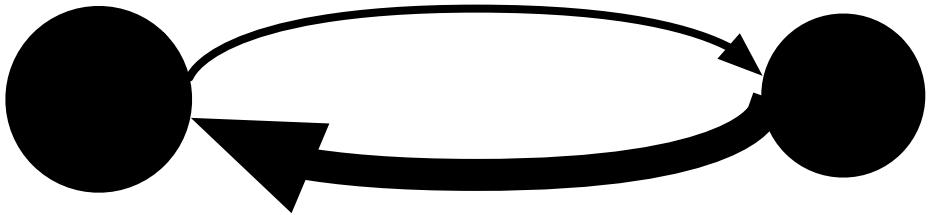
Thread Access Patterns

```
select sum(read) / sum(read + write) * 100 as read,
       sum(write) / sum(read + write) * 100 as write
from memory_object_access
```

Read	Write
66,37%	33,63%

Read and Write Distribution

Node 0 local	Node 0 to Node 1	Node 1 to Node 0	Node 1 local
24,71%	10,57%	43,05%	21,67%



Top 3 most access heavy threads

TID	N0 to N0	N0 to N1	N1 to N0	N1 to N1
59441	36,54	3,69	52,83	6,76
59891	15,35	31,67	8,13	44,64
61462	30,71	4,95	55,01	9,10

The Topics

1. NUMA system architecture

- Multiprocessor architectures: AMD, Intel, IBM, Sparc
- Interconnection technologies
- Cache coherency protocols

2. Operating Systems

- Thread and data placement approaches (papers)
- Topology discovery
- Kernel APIs

3. Programming models

- NUMA-aware algorithms
- NUMA support in high level programming languages
- OpenMP, OpenMPI
- PGAS (Unified Parallel C, Coarray Fortran, Fortress, Chapel, X10, and Global Arrays)
- C++11 Memory consistency protocol
- NUMA-aware hybrid computing: OpenCL/CUDA

4. Profiling

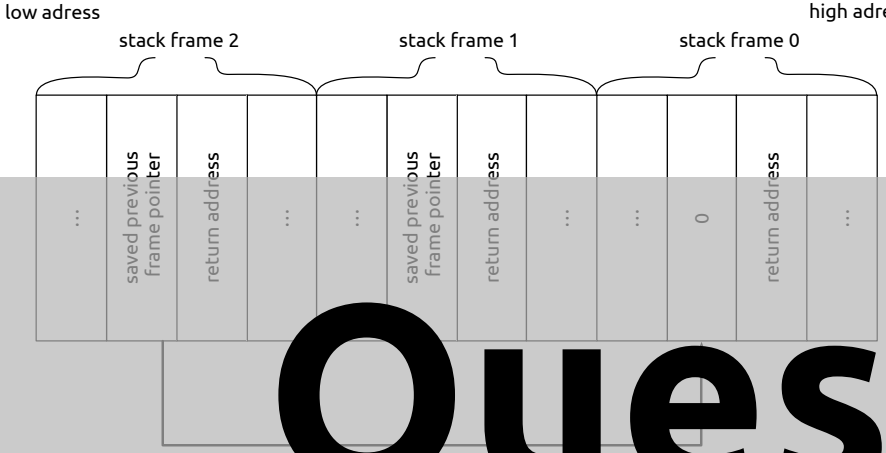
- Performance Counter
- Paper Profiler
- Tools for x64 (Linux, Windows): Intel Vtune, …
- Tools for IBM Power
- Tools for Sparc (Solaris)

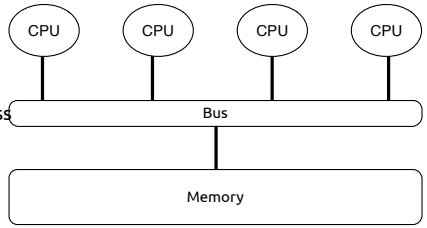
5. Case study: Linux NUMA balance

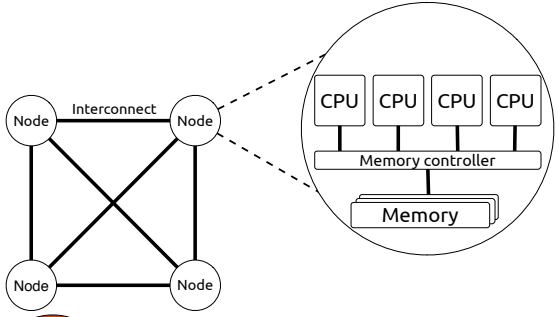
- Evolution of the NUMA-awareness of the kernel
- Mailing lists
- Git repositories
- Different approaches

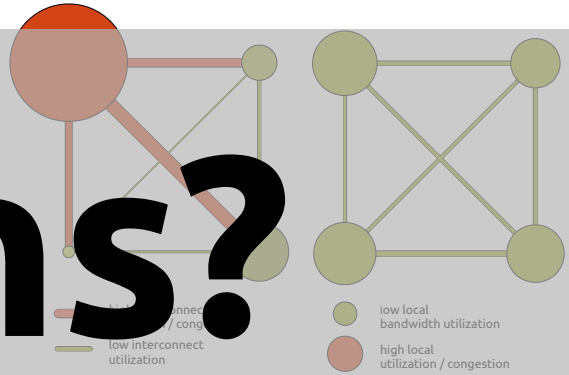
Summary

- The task
- NUMA challenges
- The topics









Questions?

```
select sum(read) / sum(read + write) * 100 as read,
sum(write) / sum(read + write) * 100 as write
from memory_object_access
```

Local Read	Local Write	Remote Read	Remote Write
31,23%	14,67%	35,15%	18,95%
Local		**Remote**	
45,9%		54,1%