Making a DSM Consistency Protocol Hierarchy-Aware: an Efficient Synchronization Scheme

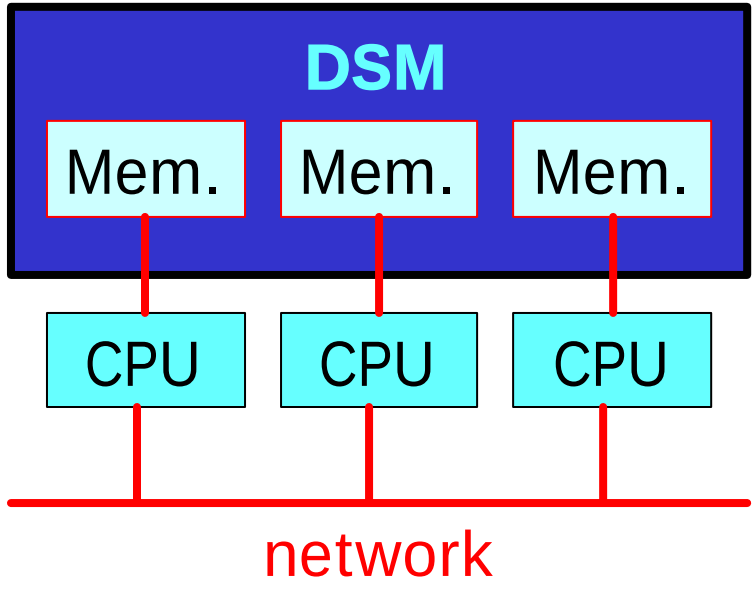
Gabriel Antoniu, Luc Bougé, Sébastien Lacour

IRISA / INRIA & ENS Cachan, France

DSM2003, Tokyo, May 13th 2003

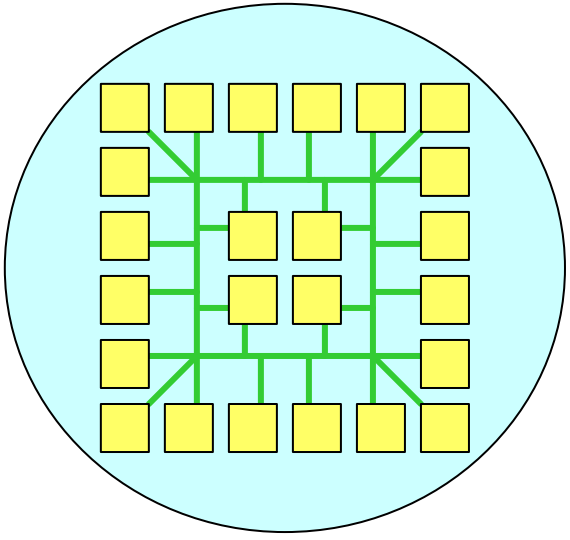
Distributed Shared Memory

- Distributed compute nodes
- Shared virtual address space

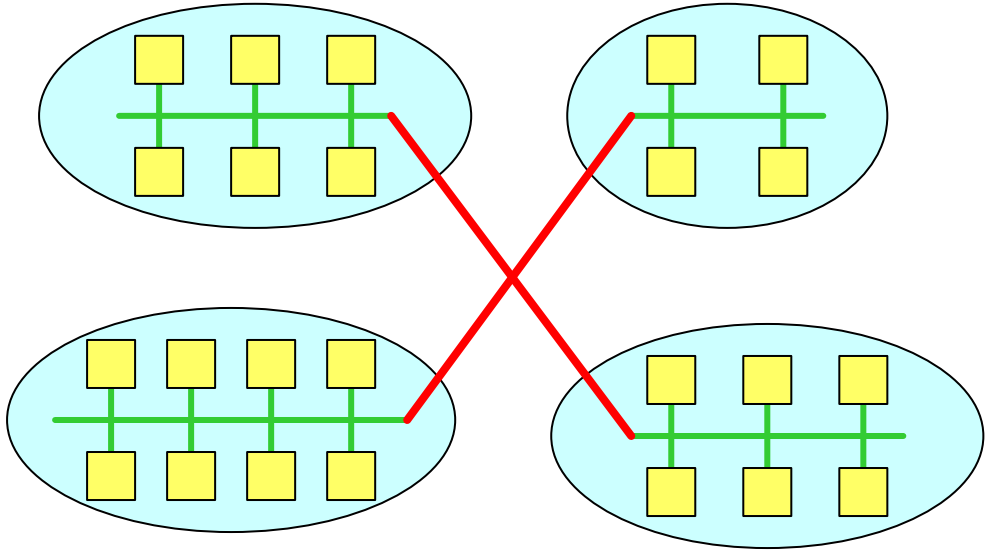


Hierarchical Network Architectures

- Flat architecture:
  - Expensive network
  - Technically difficult



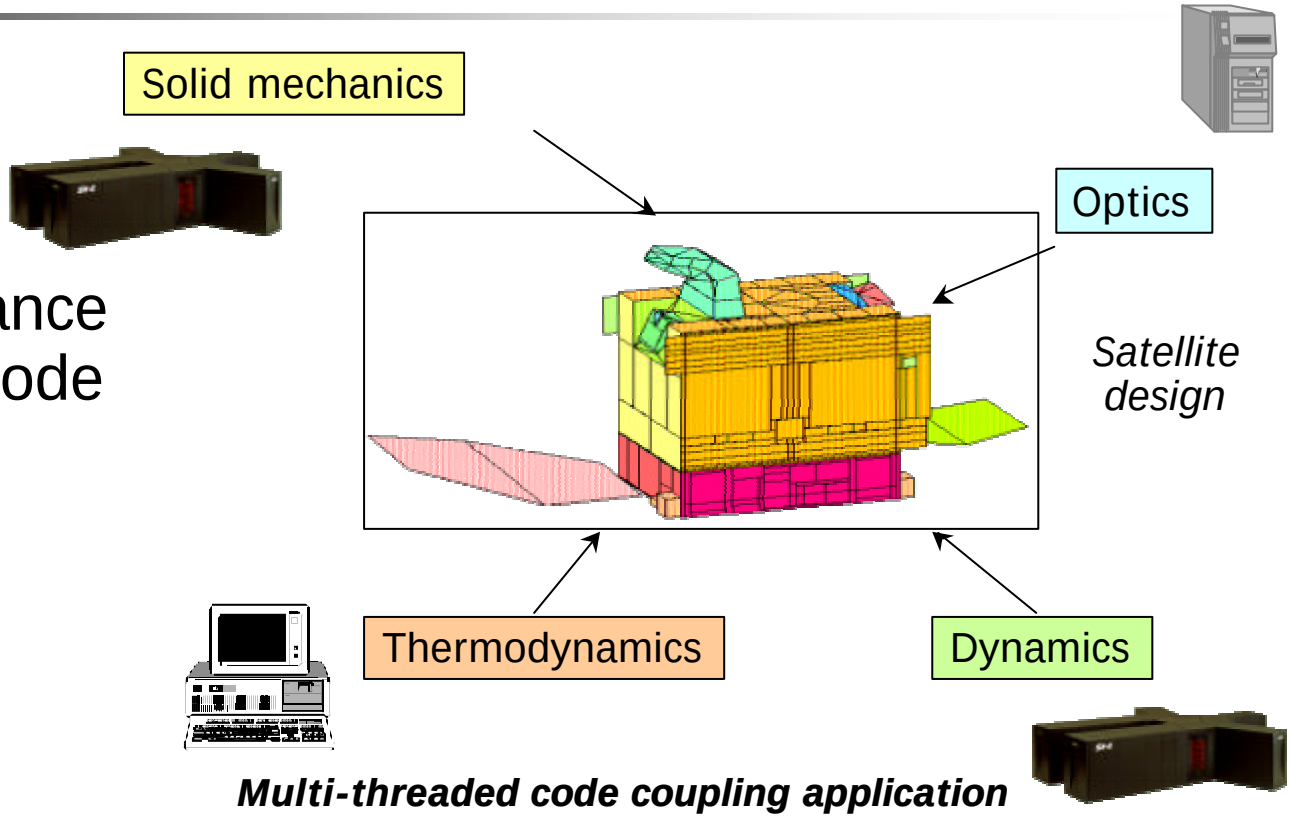
- Hierarchical architecture:
  - Cheaper network
  - Technically easier



Latencies: FastEthernet: 50-100 µs / SCI/Myrinet: 5 µs

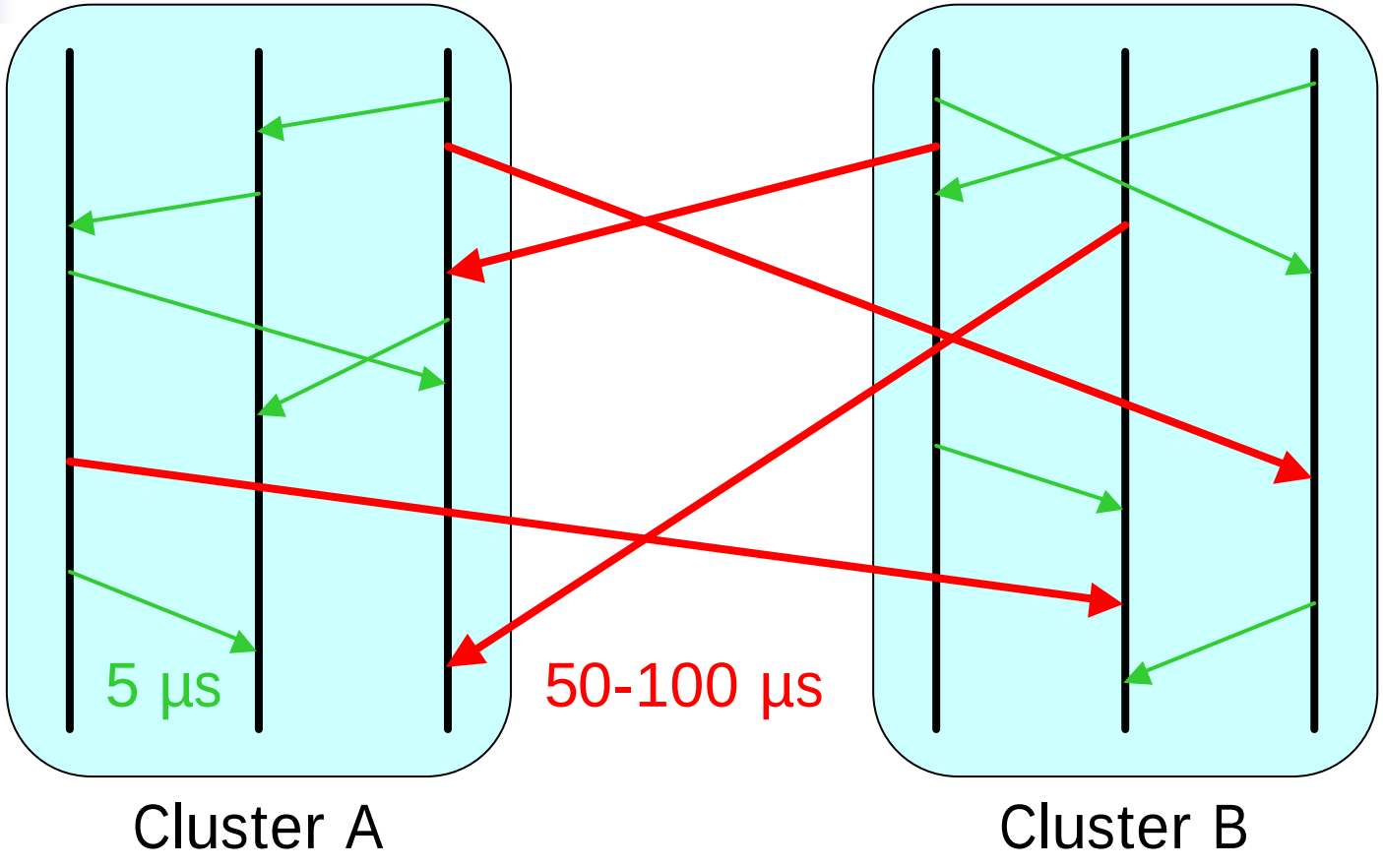
DSM on Clusters of Clusters

- Motivation:
  - high-performance computing / code coupling
- Coupling with explicit data transfer:
  - MPI, FTP, ...
- Key factor:
  - network latency

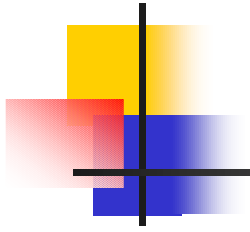


Multi-threaded code coupling application

Latency Heterogeneity and Memory Consistency Protocol



Principle: avoid communications over high-latency links

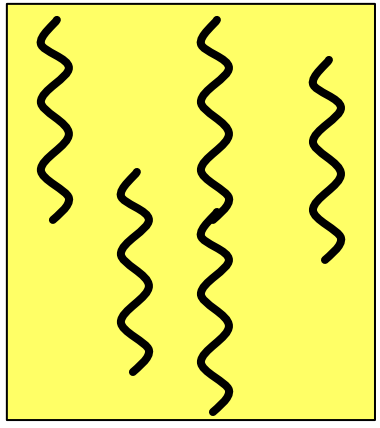


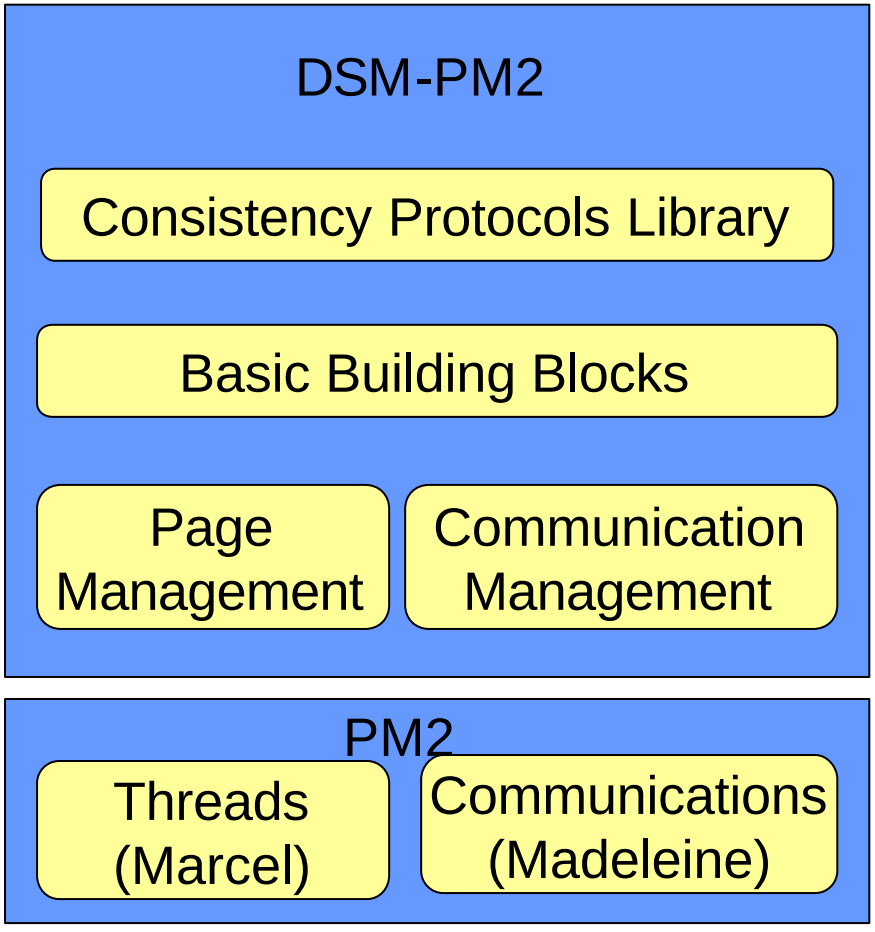
Roadmap

- Design, implement, evaluate a *hierarchical* memory consistency protocol
- Same semantics as with a flat protocol
- Well-suited for clusters of clusters
- High-performance oriented
- Few related works:
  - Clusters of SMP nodes: Cashmere-2L (1997, Rochester, NY)
  - Clusters of clusters: Clustered Lazy Release Consistency (CLRC, 2000, LIP6, Paris) $\rightarrow$ cache data locally

DSM-PM2: the Implementation Platform

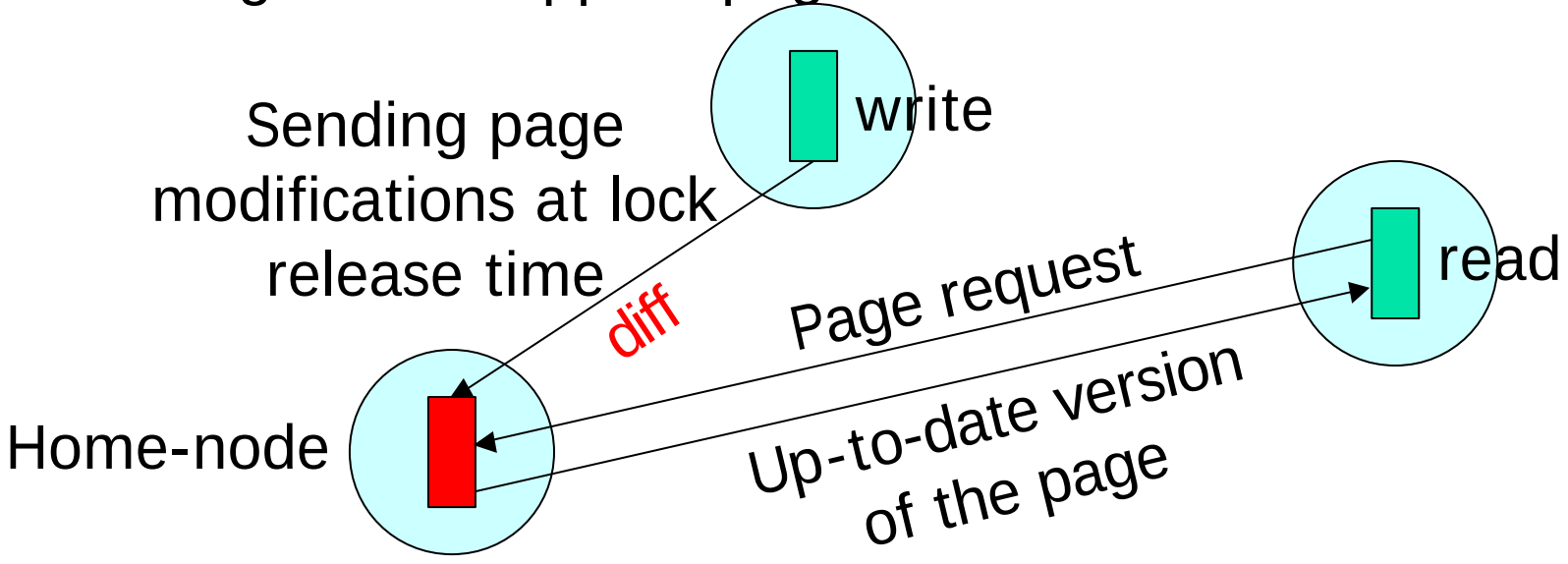
- Portability
- Multi-threading



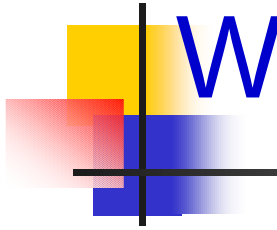


Starting Point: Home-Based Release Consistency

- Each page is attached to a Home-Node
- Multiple writers, eager version
- Home-Node:
  - holds up-to-date version of the page it hosts
  - gathers / applies page diffs



Flat Protocol & Hierarchical Architectures: Where Does Time Get Wasted?

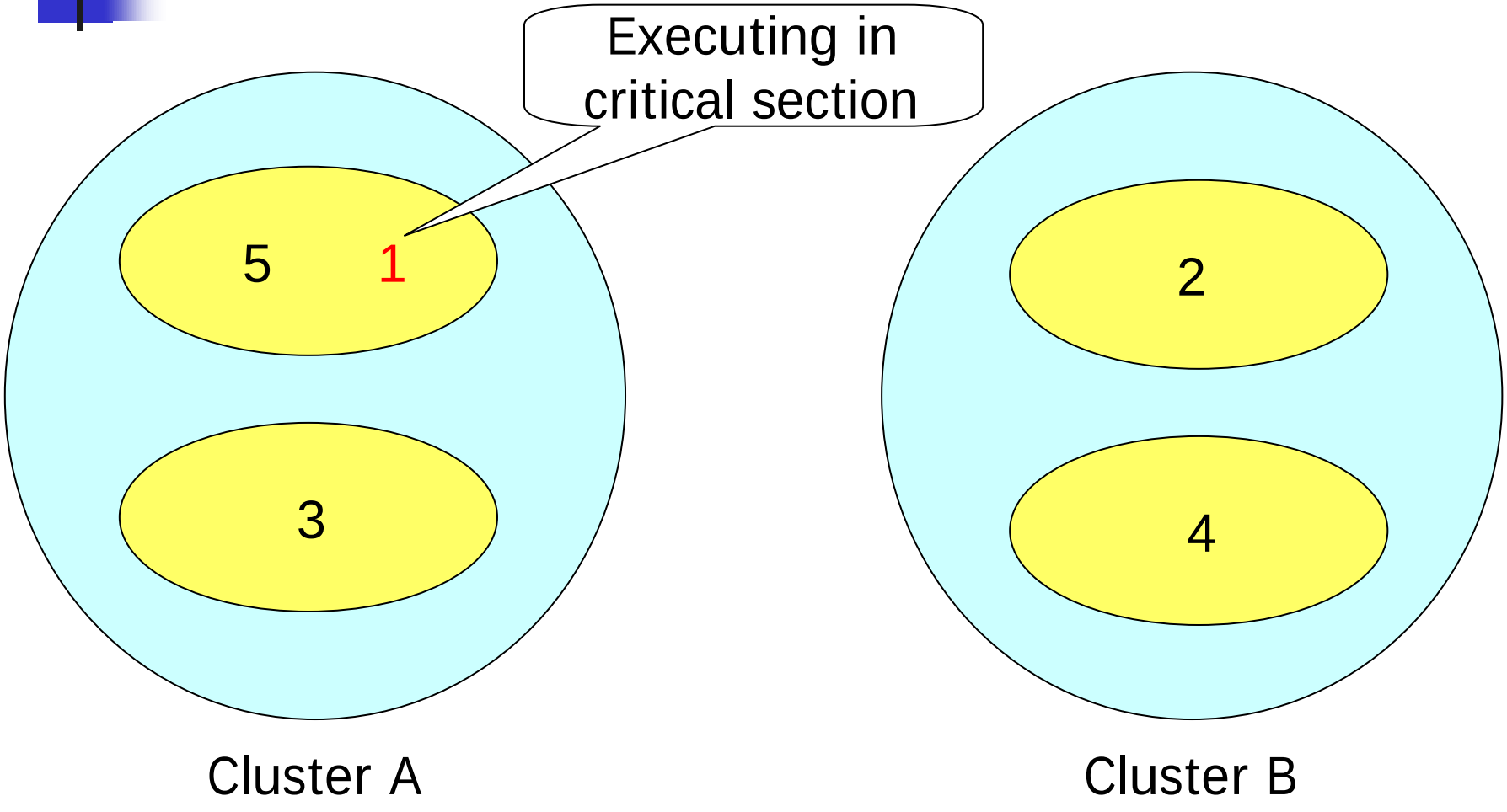


1. At synchronization operations: lock acquisition and release

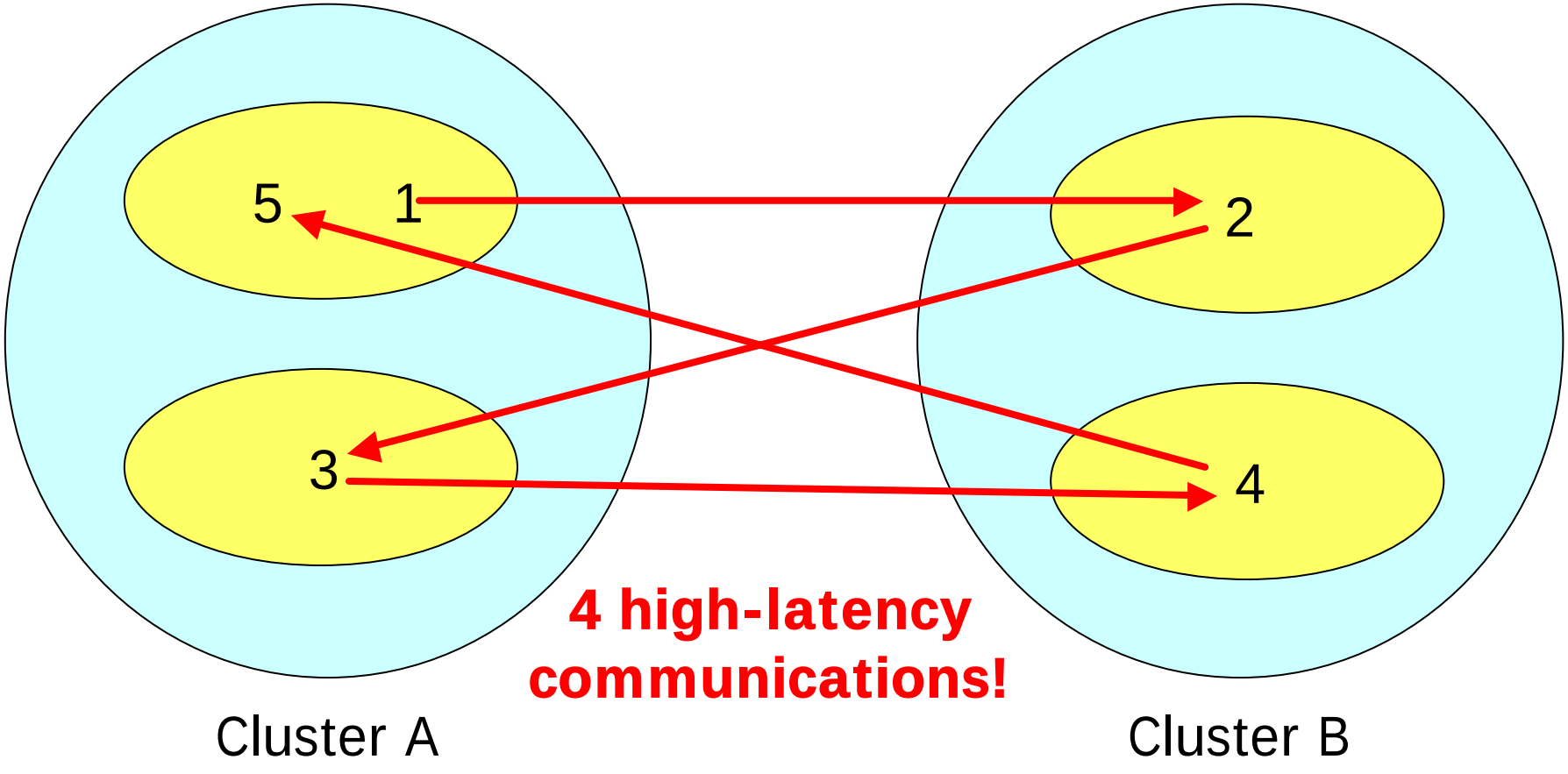
2. While waiting for message acknowledgements (consistency protocol)

- While retrieving a page from a remote node (data locality $\rightarrow$ CLRC, Paris)

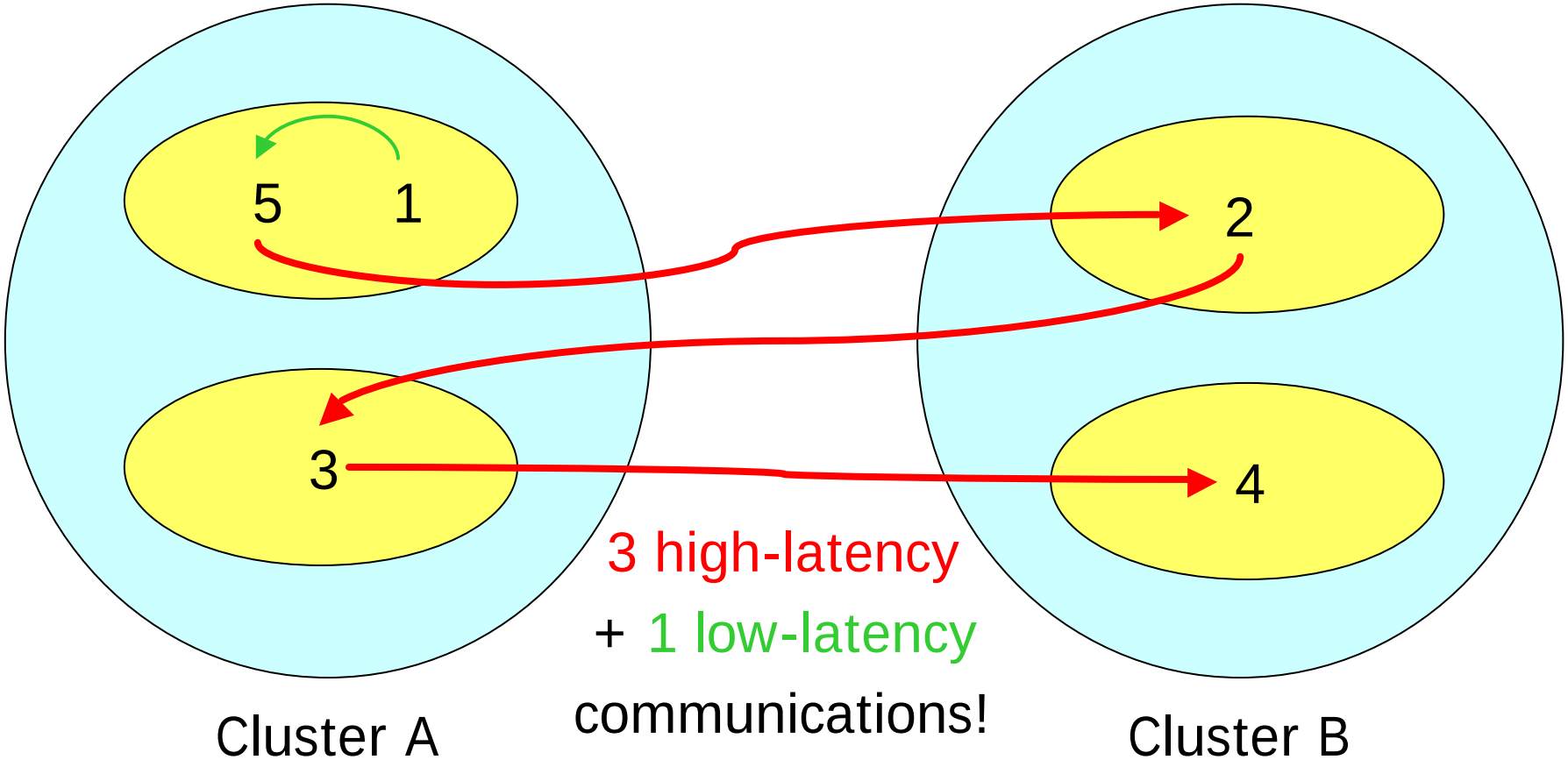
1. Improving upon Synchronization Operations



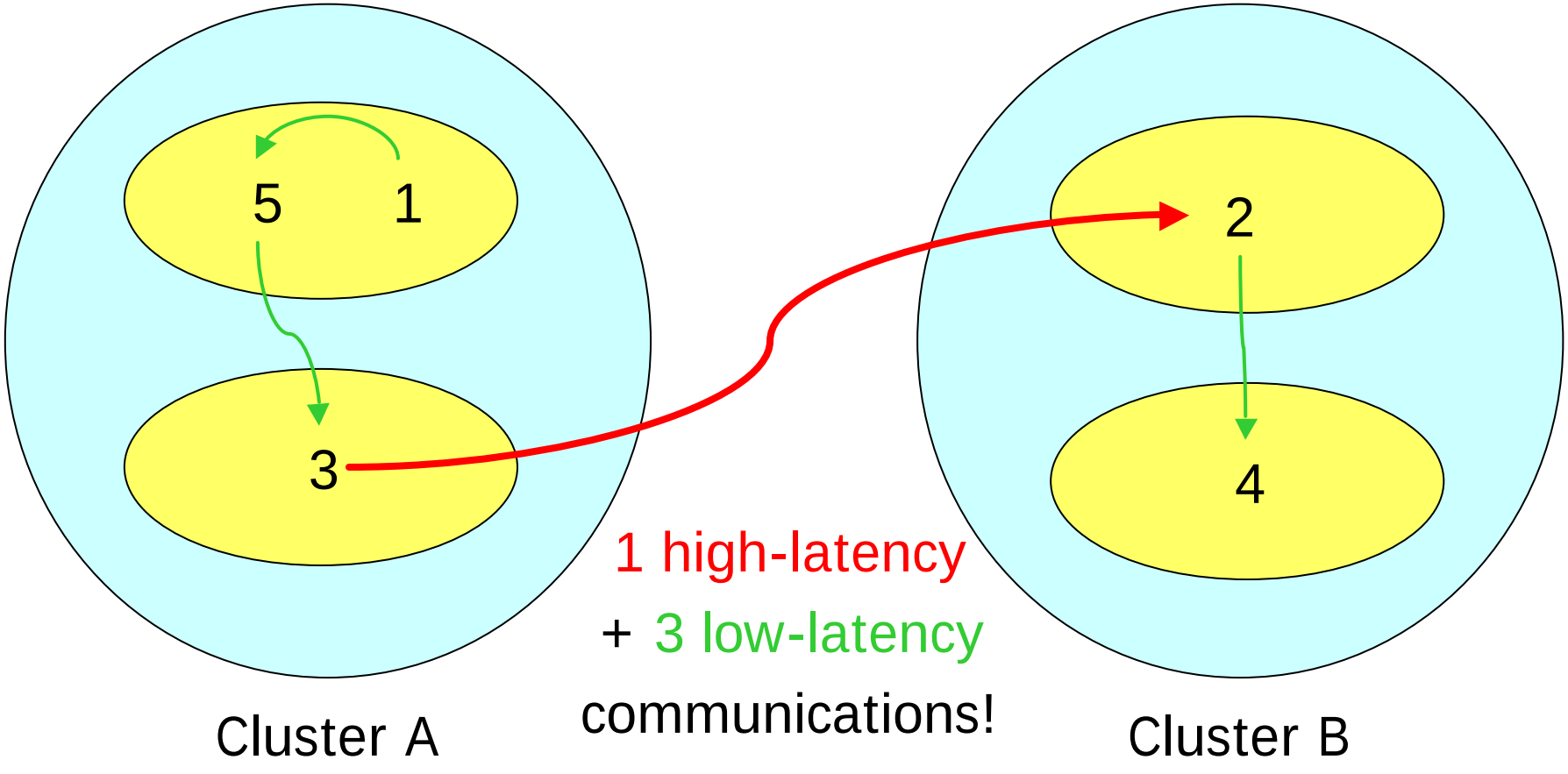
Hierarchy-Unaware Lock Acquisitions



Priority to Local Threads

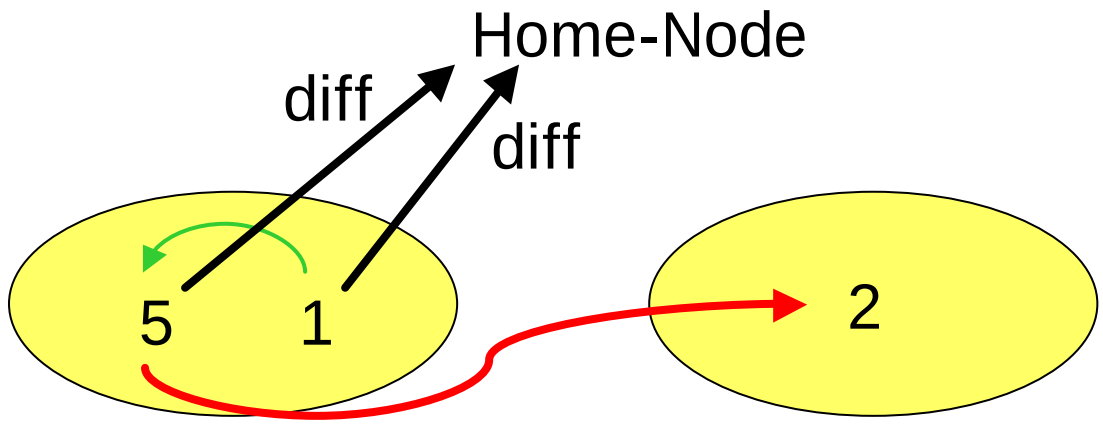


Priority to Local Nodes

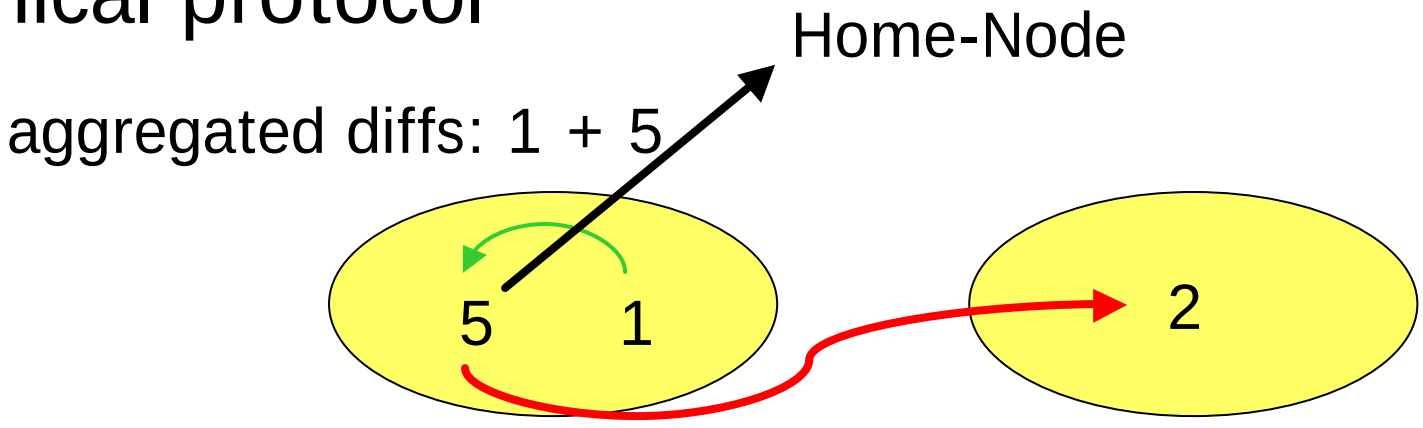


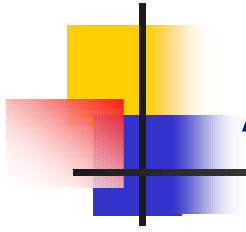
Diff Aggregation at Node Level

- Flat protocol



- Hierarchical protocol





Avoiding Starvation

- Control the lack of fairness
- Bounds to limit the number of consecutive acquisitions of a lock:
  - By the threads of a node
  - By the nodes of a cluster
  - Can be tuned at run-time

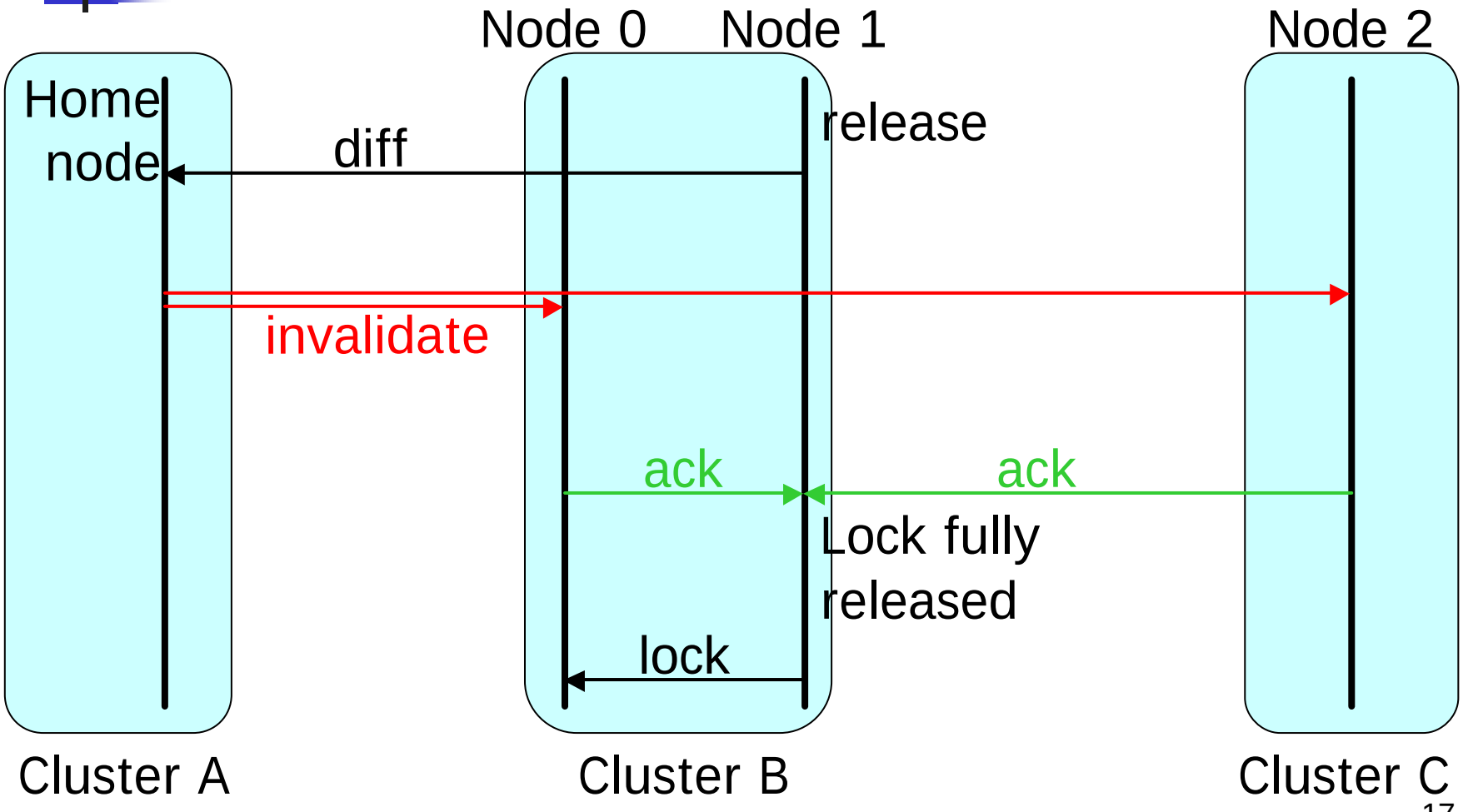
Flat Protocol & Hierarchical Architectures: Where Does Time Get Wasted?

1. At synchronization operations: lock acquisition and release

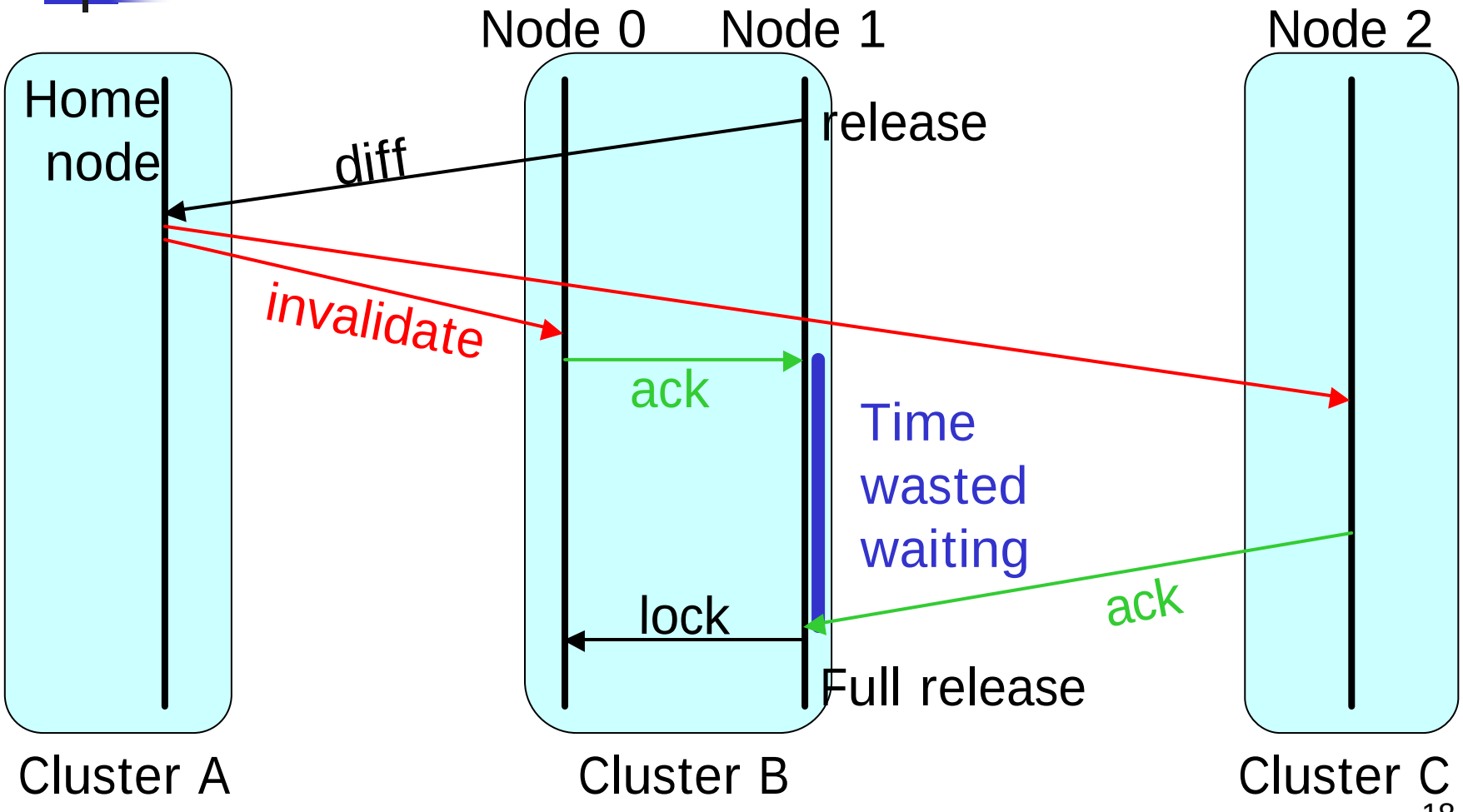
2. While waiting for message acknowledgements (consistency protocol)

- While retrieving a page from a remote node (data locality $\rightarrow$ CLRC, Paris)

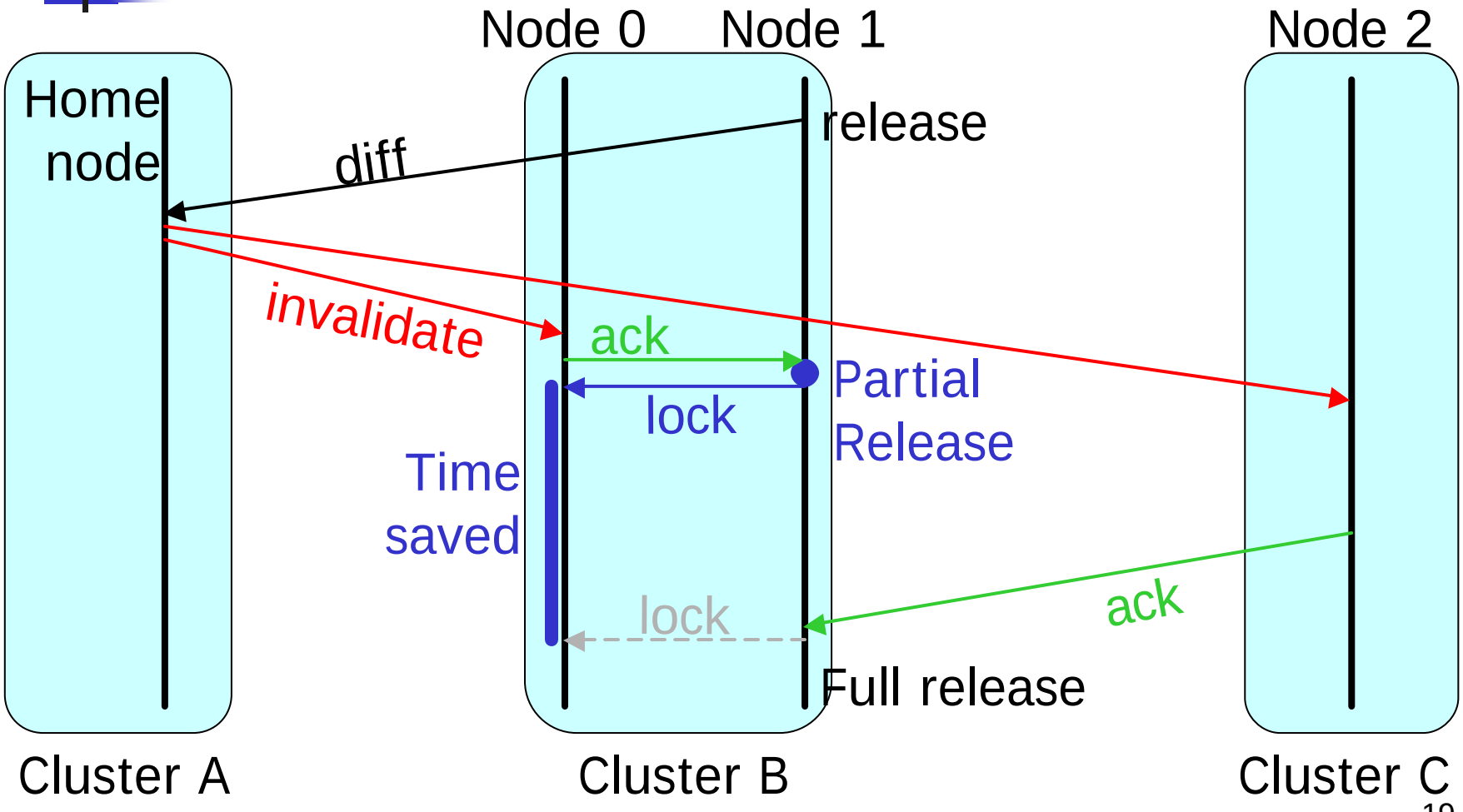
Lock Release in a Flat Protocol



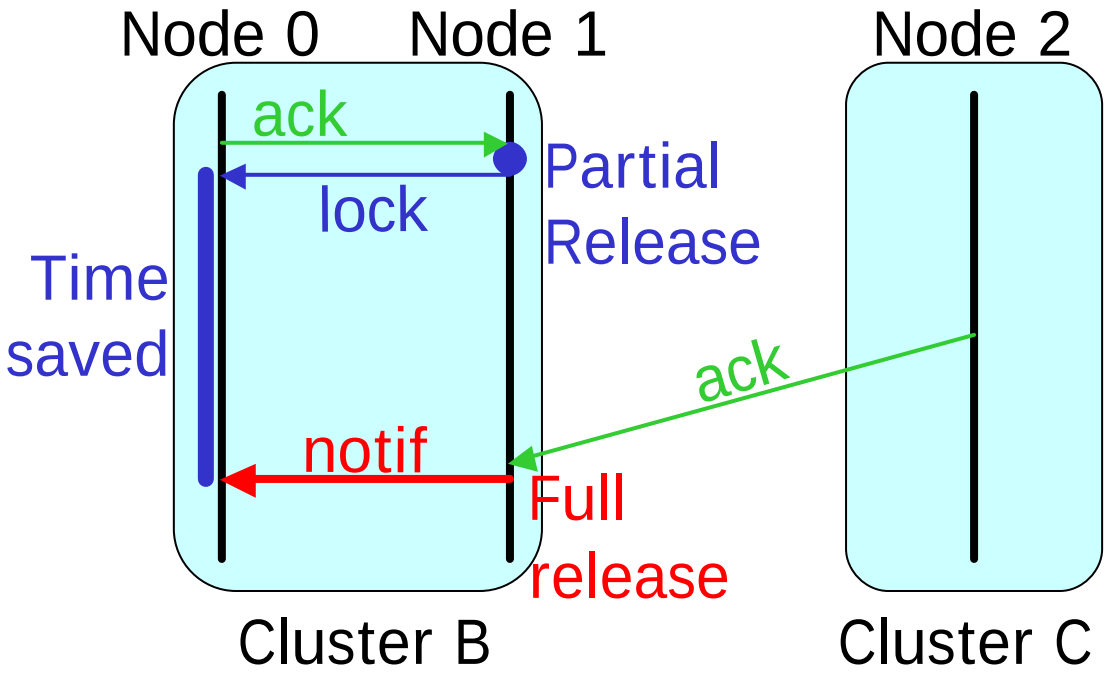
Lock Release in a Hierarchical Architecture



2. Partial Lock Release in a Hierarchical Architecture



Partial Lock Release in a Hierarchical Architecture



- Partially released locks can travel within a cluster
- Fully released locks (acks received from all clusters) can travel to remote clusters

Performance Evaluation: Thread Level

Local Thread Priority

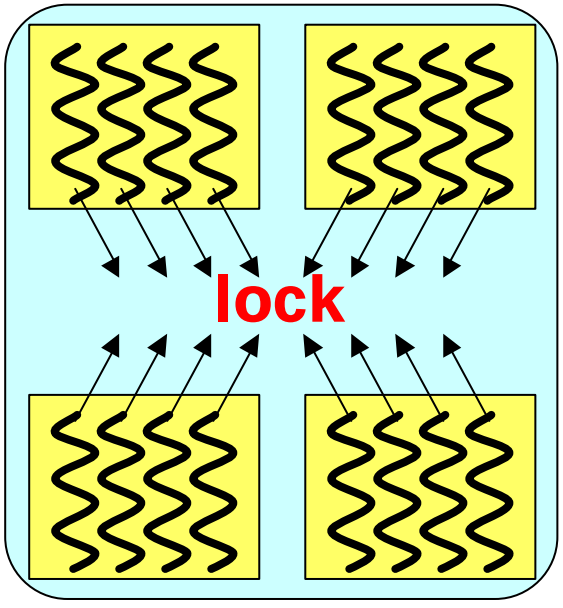
max_tp	1	5	15	25	infinity
speed-up	1	3.4	5.8	6.9	~60 (unfair)

Inter-node message → Intra-node message

Modification Aggregation

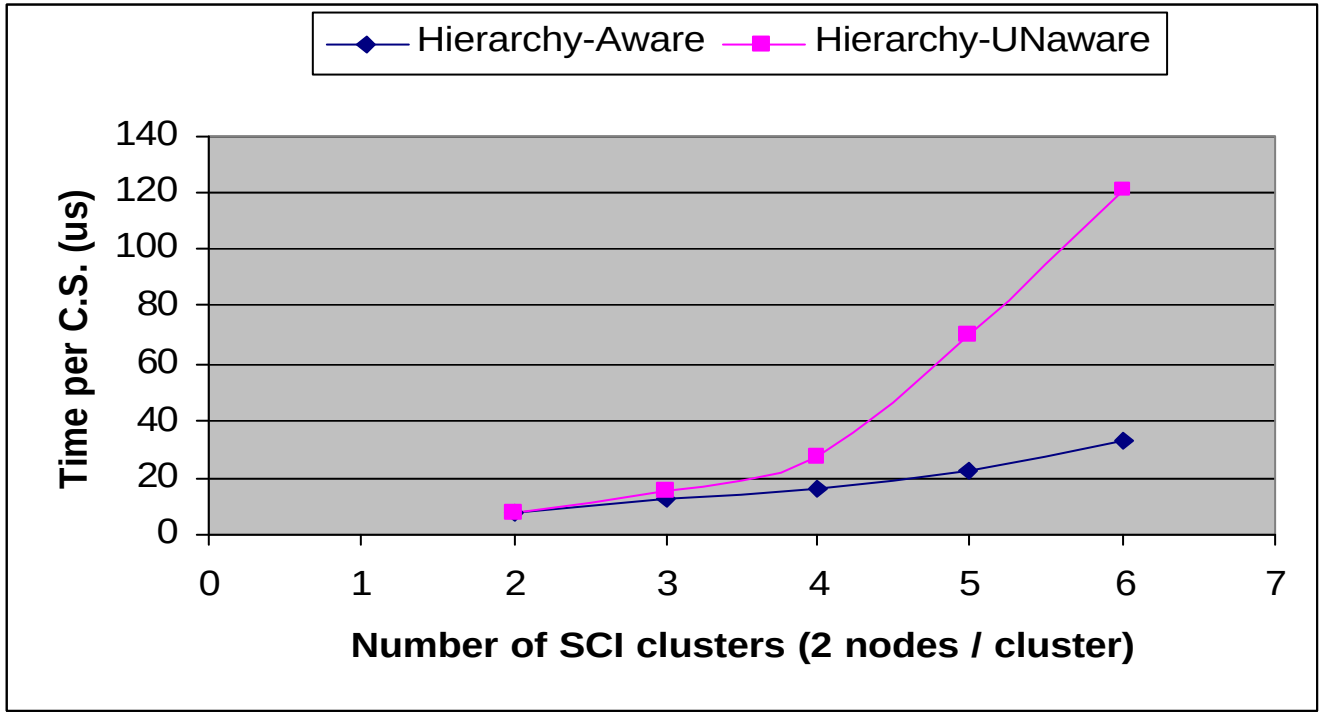
max_tp	1	5	15	25	infinity
speed-up	1	2.1	4.7	7.3	unfair

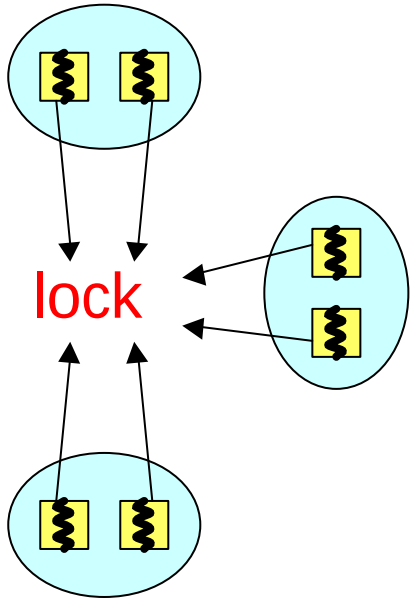
Less inter-node messages



SCI Cluster (10,000 C.S. per thread)

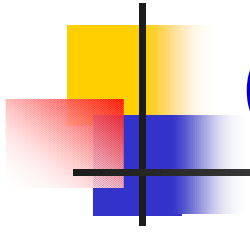
Partial Lock Release (node level): Performance Gain





Inter-cluster message receipt overlapped with intra-cluster computation

(10,000 C.S. per thread)



Conclusion

- Hierarchy-aware approach to distributed synchronization
  - Complementary with local data caches (CLRC)
- New concept of "Partial Lock Release":
  - applicable to other synchronization objects (semaphores, monitors), except for barriers
  - applicable to other eager release consistency protocols
- More hierarchy levels, greater latency ratios:

```
PING paraplapla.irisa.fr (131.254.12.8) from 131.254.12.68 : 56(84) bytes of data.
64 bytes from paraplapla.irisa.fr (131.254.12.8): icmp_seq=9 ttl=255 time=385 usec
PING ccgrid2003.apgrid.org (192.50.75.123) from 131.254.12.68 : 56(84) bytes of data.
64 bytes from sf280.hpcc.jp (192.50.75.123): icmp_seq=9 ttl=235 time=322.780 msec
```