Distributed shared memory

Computer Architecture

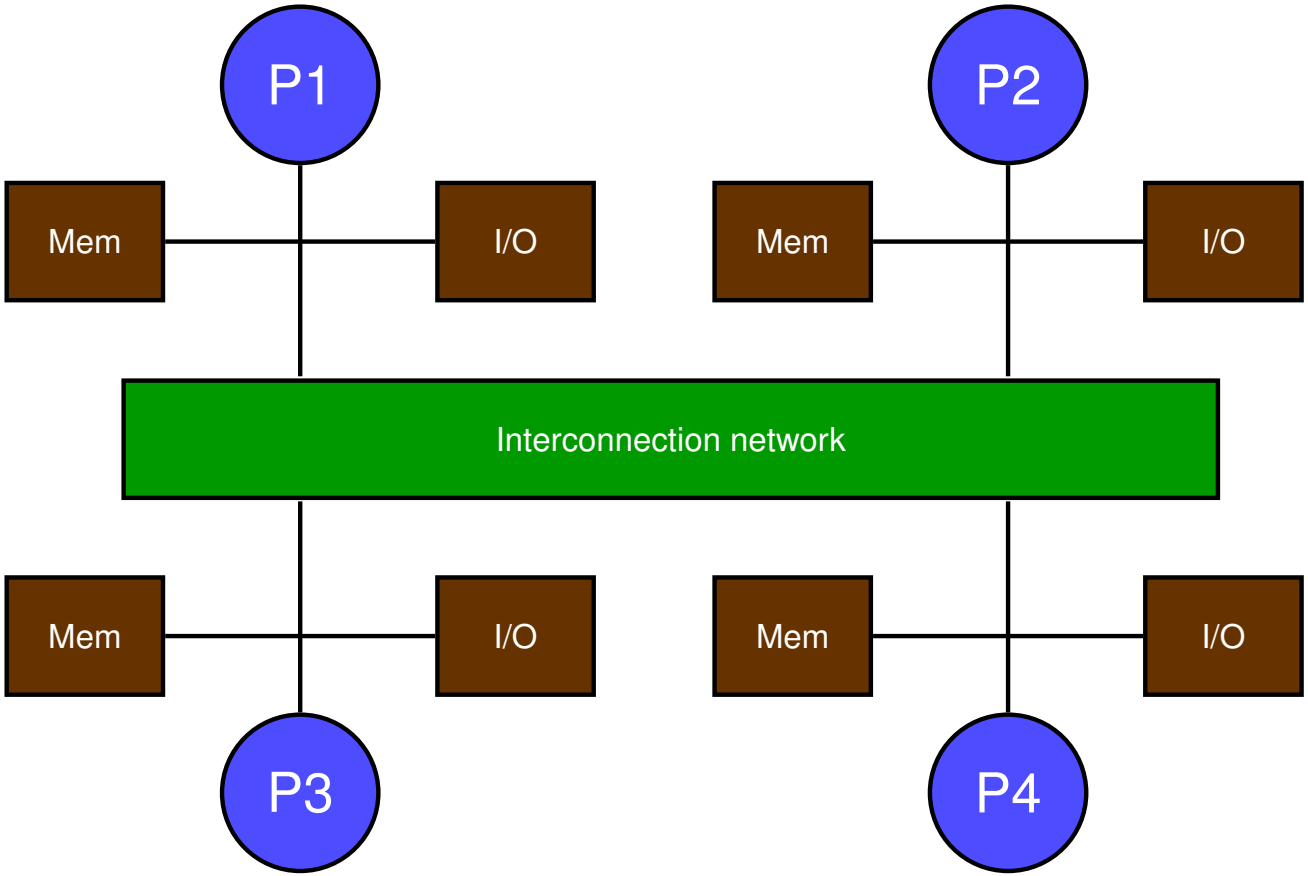
J. Daniel García Sánchez (coordinator)
David Expósito Singh
Francisco Javier García Blas

ARCOS Group
Computer Science and Engineering Department
University Carlos III of Madrid

Snooping protocols and scalability

- **Problems** with snooping protocols.
  - Requires communication with all caches:
    - For each cache miss.
    - For each shared data write.
- What is the **advantage** of snooping protocols?
  - No centralized data structure.
    - Low implementation cost.
- What is the **drawback** of snooping protocols?
  - No centralized data structure.
    - Communications limit scalability.

DSM basic model



- **Need to eliminate coherence traffic.**

Kinds of coherence protocols

- **Snooping**
  - Each cache keeps sharing status for each block it holds.
  - Caches accessible through broadcasting medium (bus).
  - All caches monitor if they have a copy of the block.

- **Directory based**:
  - Sharing status kept in a directory.
  - **SMP**: Centralized directory in memory or last level cache (LLC).
  - **DSM**: To avoid bottleneck a distributed directory (more complex) is used.

Directory based protocol

- **Idea**: Keep **status** for each block in cache.
  - Which caches have a block copy?
  - Block state bits.

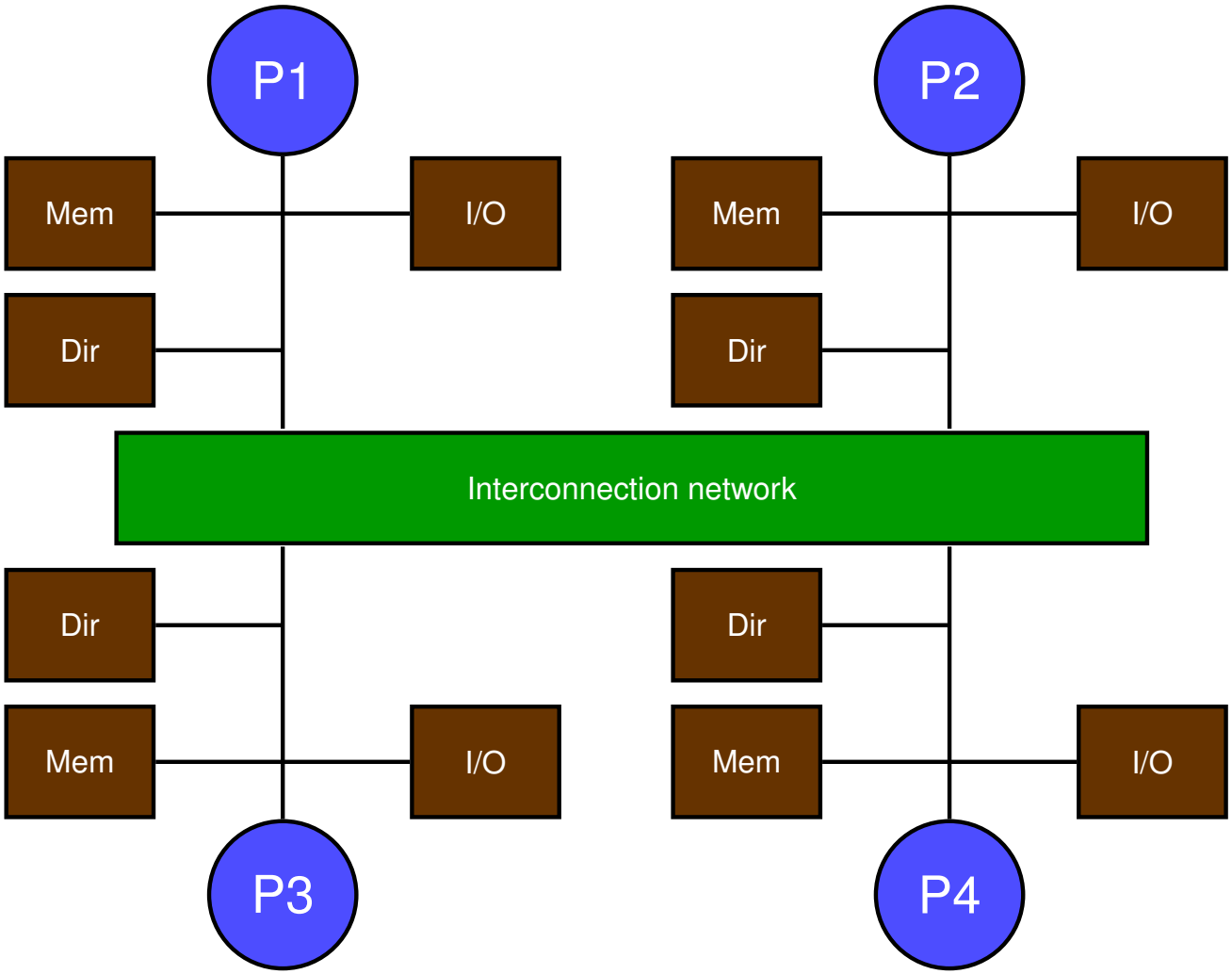
- Multi-cores with external shared cache.
  - **Bit vector** with length equal to number of cores.
    - Signals which private caches many have a copy of block.
    - Only send invalidation to caches marked in bitmap.
  - This scheme works well within a single multi-core.
  - **Example**: **Intel Core i7**.

Centralized directory and scalability

- A centralized directory avoids broadcasting but
  - Becomes a bottleneck.
  - Scalability problem with increasing number of processors.

- **Solution**: **Distributed directory**.
  - Distribute directory with memory.
  - Each directory has information about the associated local memory.
    - Target directory is always known.
  - Different coherence requests go to different directories.

Distributed directory



Directory

- Basic operations.
  - Handling a read miss.
  - Handling a clean (not modified) shared block write.
- Directory must keep **status for each block**:
  - **Shared**: One or more nodes have a block in cache and its value in memory is updated.
  - **Non cached**: No node has a copy of the block in cache.
  - **Modified**: Only a node has a copy of the block in cache and it has written on it.
    - The block value in memory is not updated.
- **Besides**:
  - Bit map with information of which nodes have a copy of block.

Messages

Message	Source	Target	Content	Function
Read miss	Local cache	Local directory	P,A	P has read miss in A. Request data and P is sharer.
Write miss	Local cache	Local directory	P,A	P has write miss in A. Request data and P is owner.
Invalidation	Local cache	Local directory	A	Invalidate A in all caches.
Invalidation	Local cache	Remote directory	A	Invalidate shared copy.
Fetch	Local directory	Remote cache	A	Fetch block. State to shared.
Fetch/ Invalidation	Local directory	Remote cache	A	Fetch block. Invalidate block.
Response data value	Local directory	Local cache	D	Return value to directory.
Write-back data	Remote cache	Local directory	A,D	Write-back data.

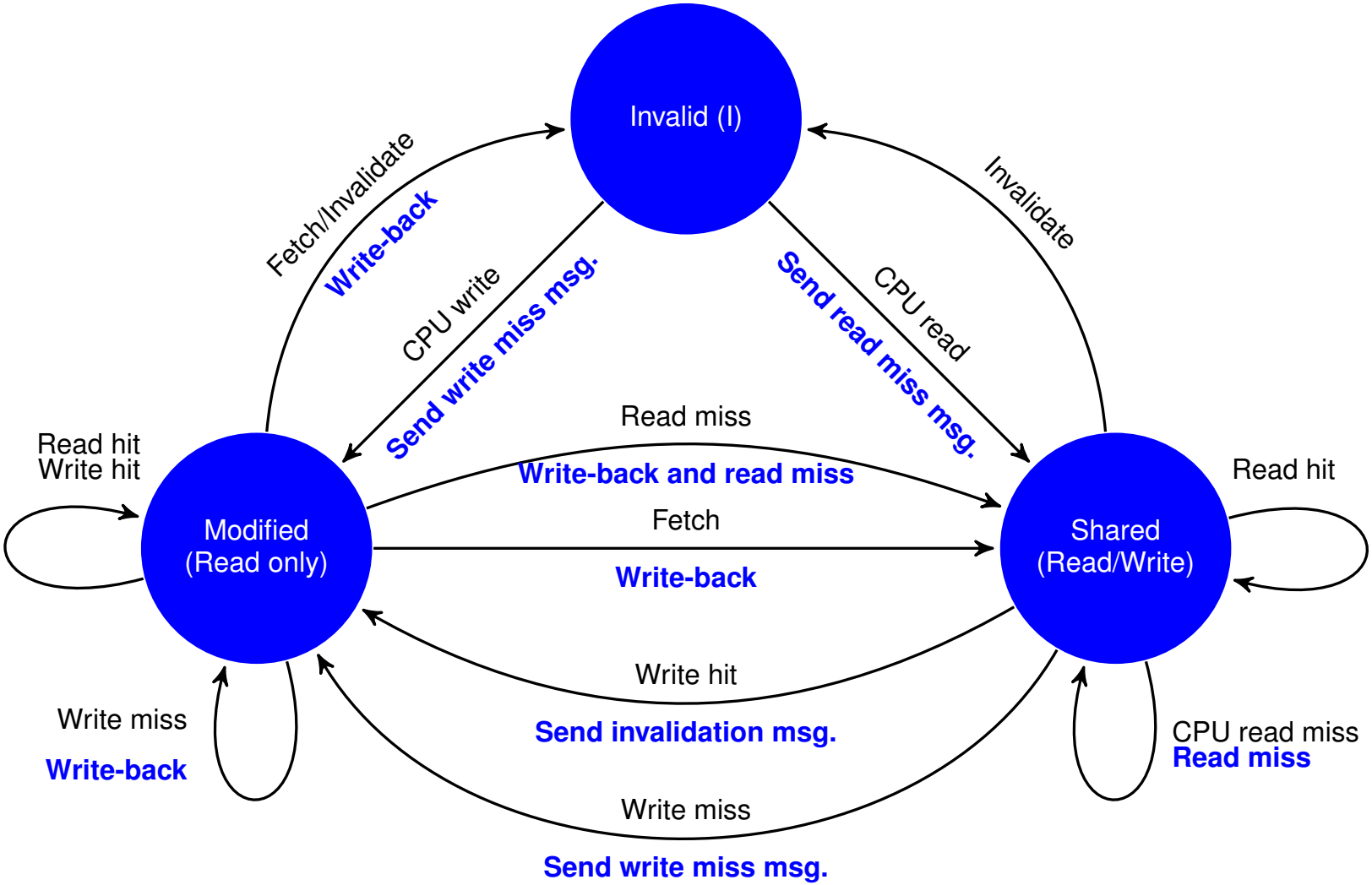
P → **Node**, **A** → **Address**, **D**→ **Data**

State transition

- **In multi-core chips**:
  - **Internal coherence** is kept through **centralized directory**.
  - **Same directory** may act as **local directory** in **DSM**.

- **Protocol implementation**:
  - State transition in **local cache**.
    - Send requests to **local directory**.
  - State transition in **directory**.

State transition in individual cache



Non cached entry

- Value in memory is updated.
- **Requests**:
  - **Read miss**:
    - Send data in memory to requesting node.
    - Requesting node is the only one in shared state.
    - State transitions to shared.
  - **Write miss**:
    - Send data in memory to requesting node.
    - Block transitions to exclusive state.
    - Requesting node is the owner.

Shared entry

- Value in memory is up to date.
- **Requests**:
  - **Read miss**:
    - Send data from memory to requesting node.
    - Requesting node added to entry nodes set.
  - **Write miss**:
    - Send data from memory to requesting node.
    - Send invalidation messages to entry nodes set.
    - Enable in set only the requesting node.
    - Transition state to exclusive.

Exclusive entry

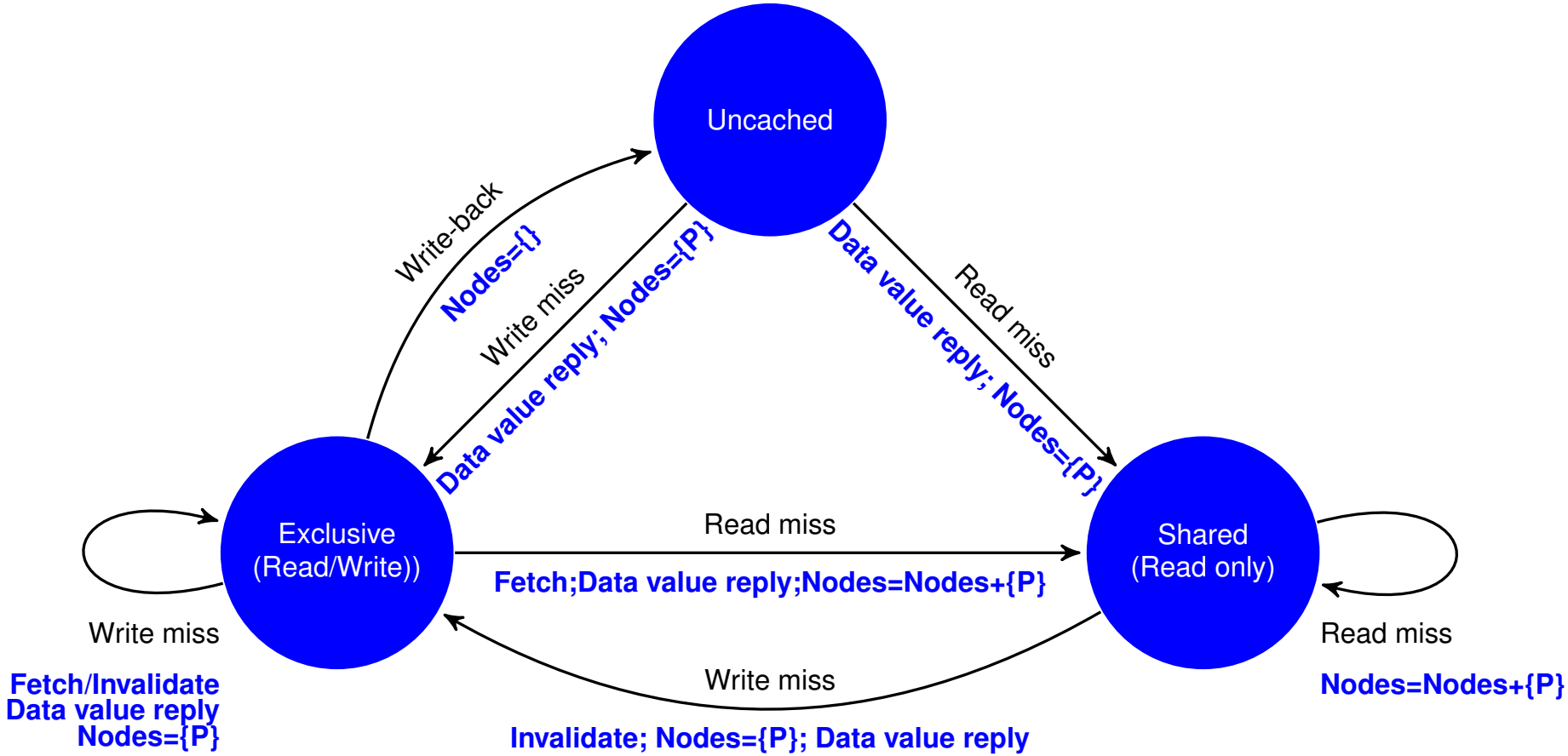
- Block value is in cache in the node identified by the set (owner node).

- **Requests**:
  - **Read miss**:
    - Send fetch message to owner.
    - Write data in memory.
    - Send data to requesting node.
    - Add requesting node to node set.

Exclusive entry

- **Requests**:
  - **Write back**:
    - Happens when owner performs write-back of block.
    - Block transitions to uncached state.
    - Clean entry set.
  - **Write miss**:
    - Block has new owner.
    - Invalidate block in old owner and get value.
    - Send value to requesting node.
    - Enable in set only the new requesting.

Directory state transition



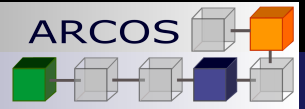
Summary

- Scalability problems in snooping protocols.
- Alternatives for directory based protocols:
  - Centralized directory in SMP.
  - Distributed directory in DSM.
- In multi-core chips:
  - Internal coherence through centralized directory.
  - Used as local directory in DSM.

References

- **Computer Architecture. A Quantitative Approach**
  5th Ed.
  Hennessy and Patterson.
  Sections: 5.4.

- **Recommended exercises**:
  - 5.9, 5.10, 5.11, 5.12.



Distributed shared memory

Computer Architecture

J. Daniel García Sánchez (coordinator)
David Expósito Singh
Francisco Javier García Blas

ARCOS Group
Computer Science and Engineering Department
University Carlos III of Madrid