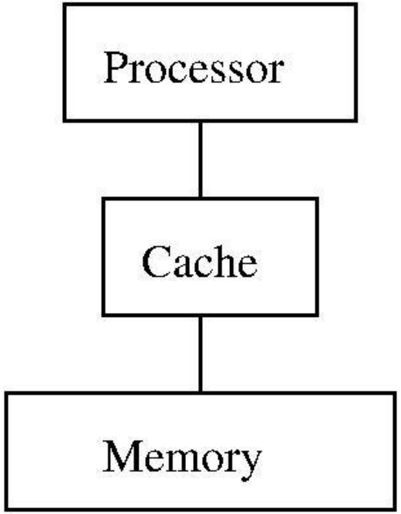
# Multiprocessor caching and false sharing

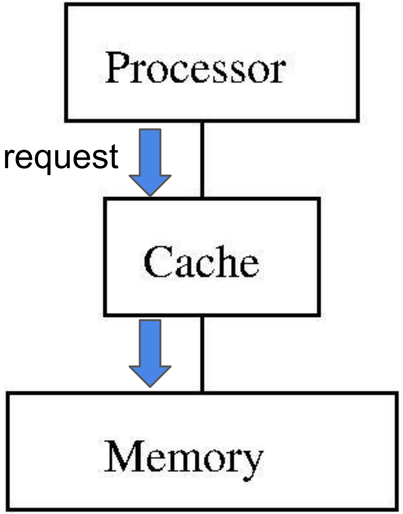
# Recall: Caching

- Cache(s) sit between processor and memory
  - Handle memory accesses w/o going all the way to memory
  - Exploit temporal locality and spatial locality



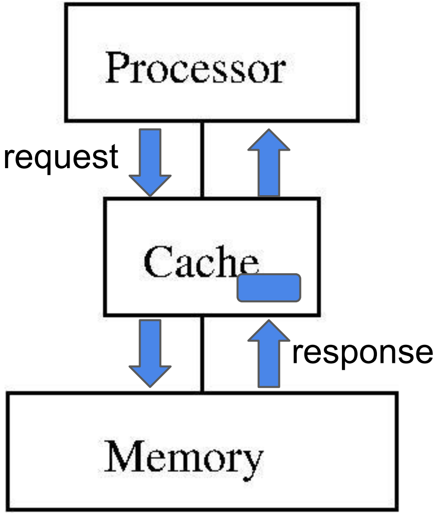
# Recall: Caching

- Cache(s) sit between processor and memory
  - Handle memory accesses w/o going all the way to memory
  - Exploit temporal locality and spatial locality



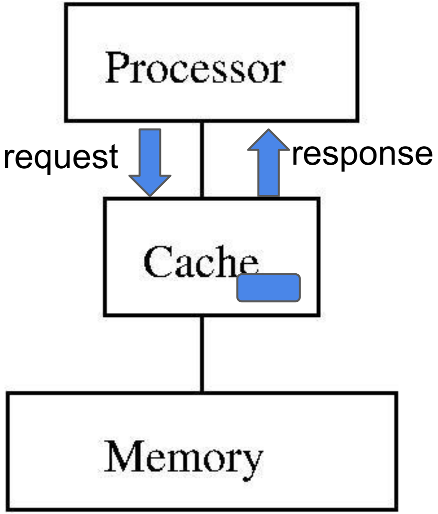
# Recall: Caching

- Cache(s) sit between processor and memory
  - Handle memory accesses w/o going all the way to memory
  - Exploit temporal locality and spatial locality



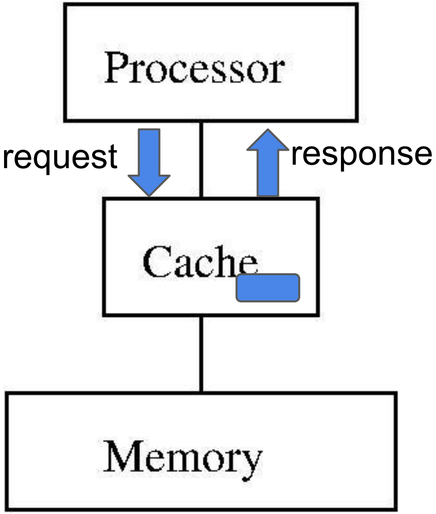
# Recall: Caching

- Cache(s) sit between processor and memory
  - Handle memory accesses w/o going all the way to memory
  - Exploit temporal locality and spatial locality



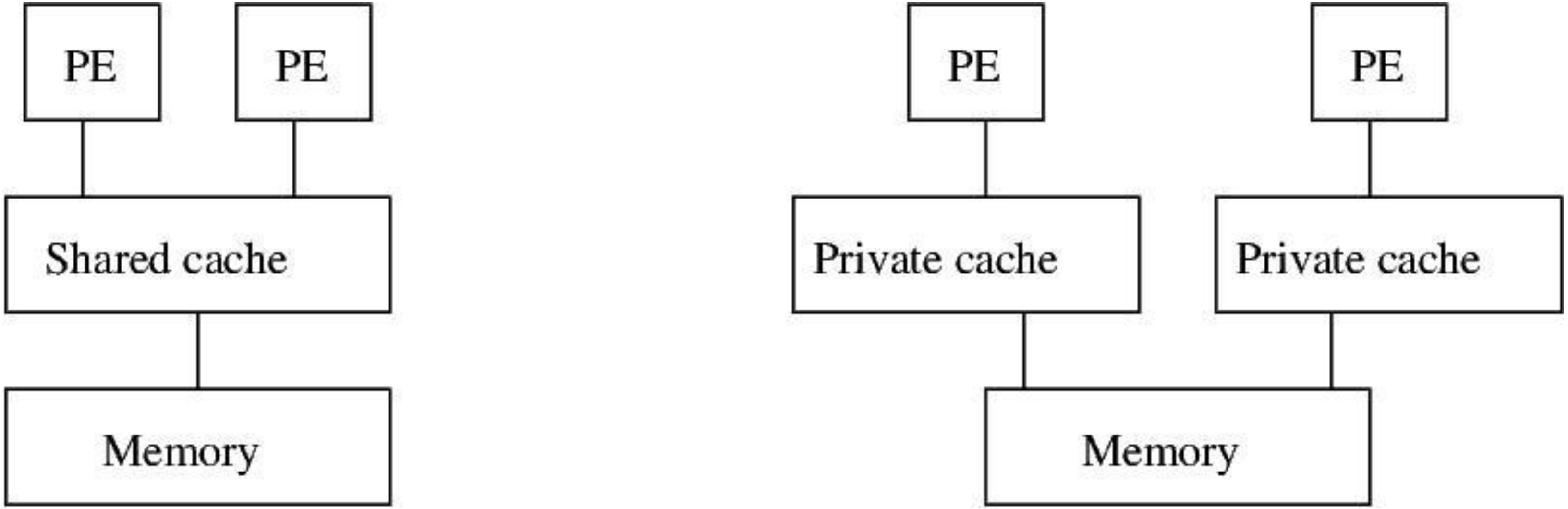
# Recall: Caching

- Cache(s) sit between processor and memory
    - Handle memory accesses w/o going all the way to memory
    - Exploit temporal locality and spatial locality



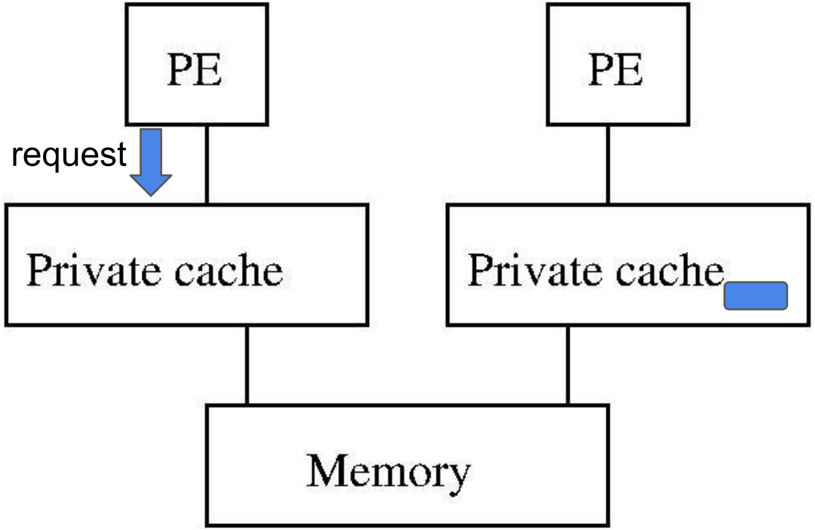
- Store lines that with recently accessed memory locations and other locations nearby

# Caching for multiple processing elements (PEs)



# Cache coherence

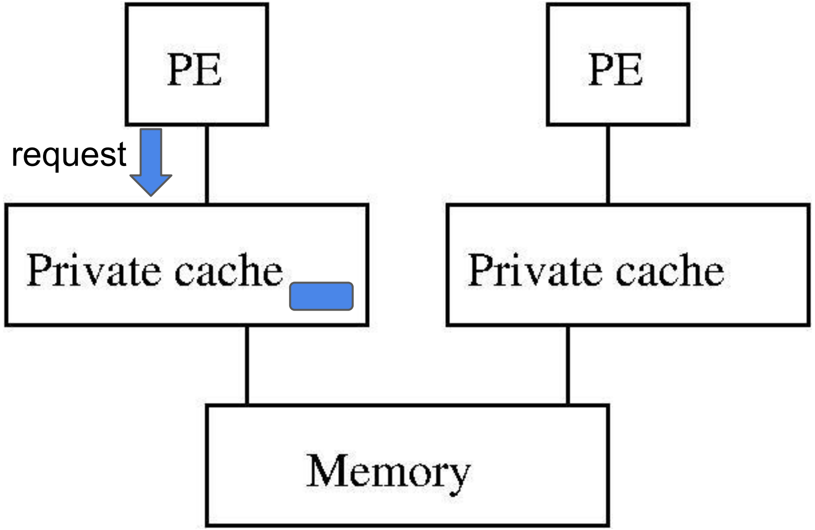
- Potential issue when cache line is in one cache and other PE requests it



# Cache coherence

- Potential issue when cache line is in one cache and other PE requests it

- Cache of requesting PE needs way to steal that line

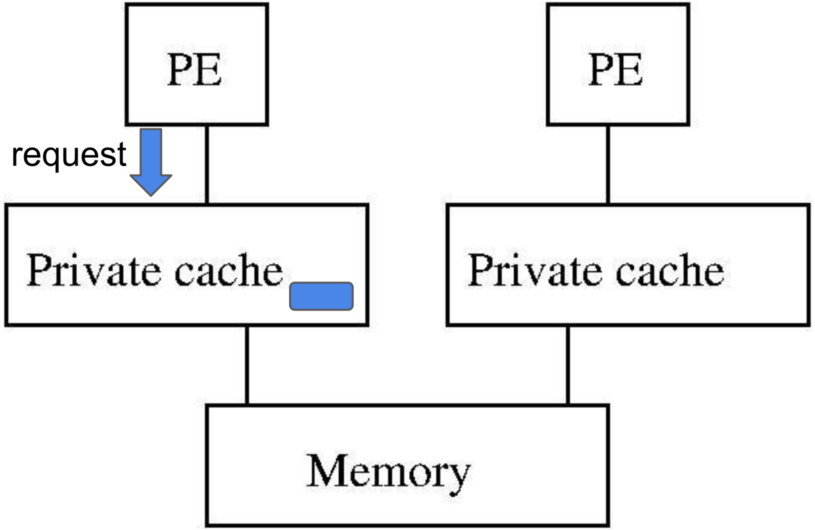


# Cache coherence

- Potential issue when cache line is in one cache and other PE requests it

- Cache of requesting PE needs way to steal that line

- Slower than normal cache hit because of need to steal

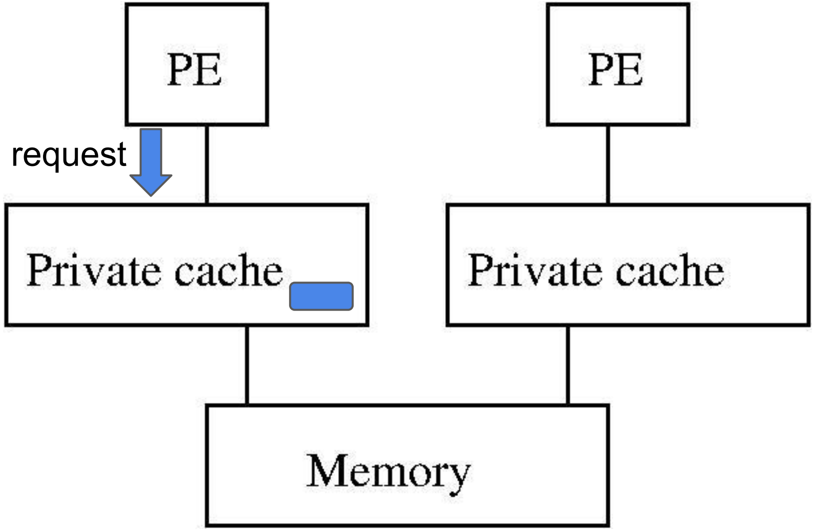


# Cache coherence

- Potential issue when cache line is in one cache and other PE requests it

- Cache of requesting PE needs way to steal that line

- Slower than normal cache hit because of need to steal



- False sharing: When a memory access causes a cache line to be stolen even though no data is actually being shared