Project PBerry: FPGA Acceleration for Remote Memory

Irina Calciu, Ivan Puddu, Aasheesh Kolli

Andreas Nowatzyk, Jayneel Gandhi, Onur Mutlu

Pratap Subrahmanyam

HotOS 2019

vmware

Recent remote memory systems

- Systems: Infiniswap, HotPot, Remote regions
- Same goal: pool remote memory
- Different interfaces: swapping, dspm, file system
- Same mechanism: page faults

Use cache coherence traffic to track application memory accesses

Problem

Remote memory systems use page faults

Overheads:

1. Page faults overheads
2. Page granularity

Solution

Use cache coherence (w/ FPGA)

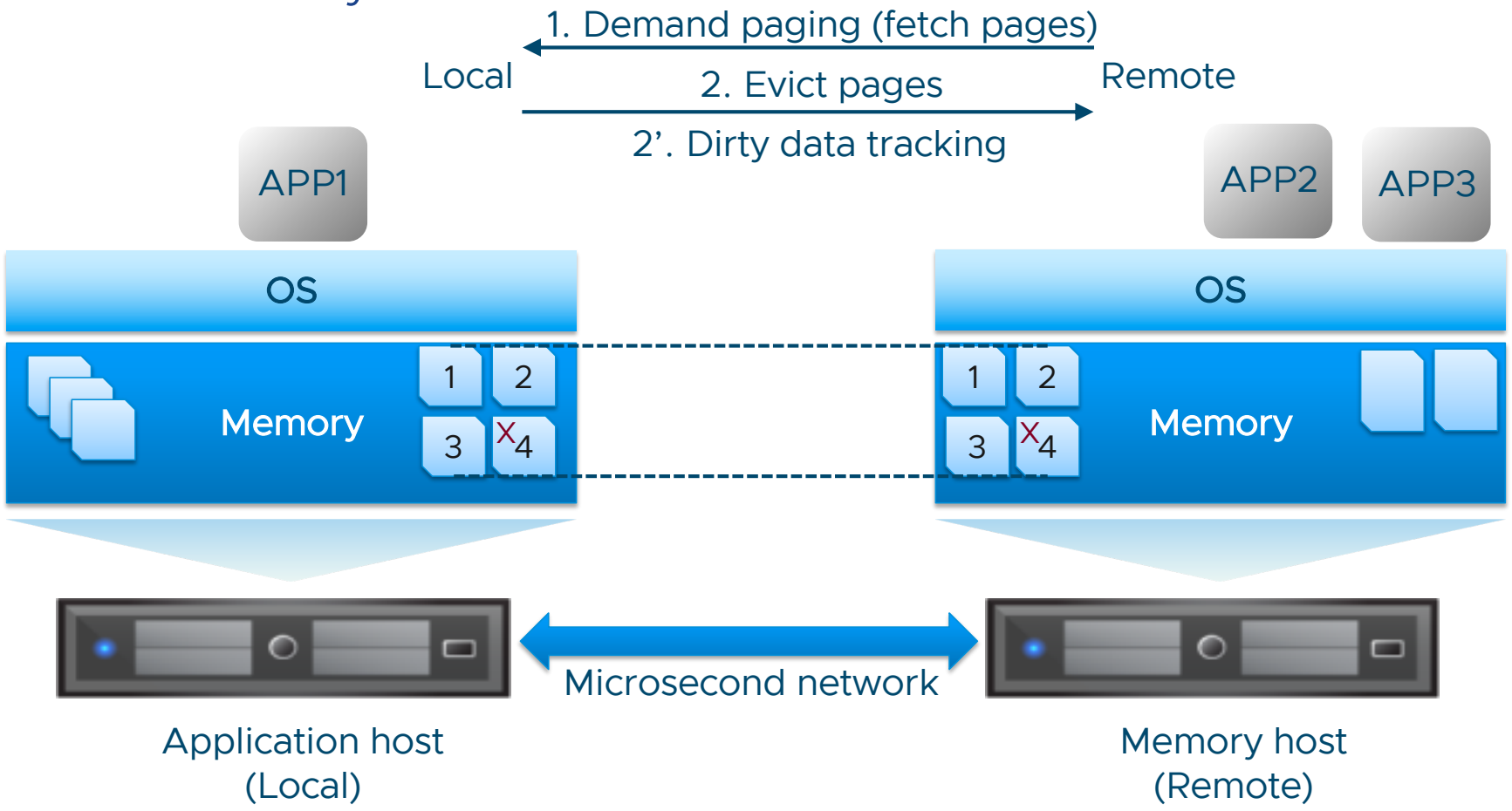
Implications:

1. Reduce page faults
2. Cache line granularity

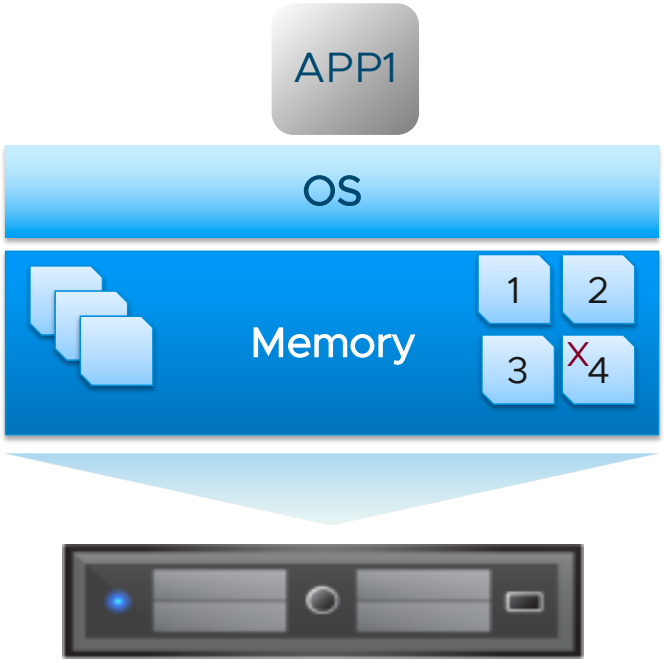
Impact

1. Remote memory
2. Live migration
3. Code analysis
4. Security
5. …

Remote memory



How to track dirty data?



Application host

Current solution:

1. Write-protect page

2. On page fault: mark page dirty

Overheads in current remote memory systems

1. Dirty data amplification (4KB, 2MB, 1GB)
   - Breaking large pages to 4KB pages [Smarduch et al., CloudOpen-Japan 2015]

Application	Amplification (4KB pages)	Amplification (2MB pages)	Amplification (64B cache-line)
Redis-random	96.82 % (3.87 KB)	99.99 % (1.99 MB)	31.70 % (20.29 B)
Redis-seq	82.91 % (3.32 KB)	99.83 % (1.99 MB)	17.18 % (10.99 B)

Overheads in current remote memory systems

1. Dirty data amplification (4KB, 2MB, 1GB)
   - Breaking large pages to 4KB pages [Smarduch et al., CloudOpen-Japan 2015]

2. Application slowdown due to (write) page faults

3. Cost of write-protecting pages

Use cache coherence traffic to track application memory accesses

Problem

Remote memory systems use page faults

Overheads:

1. Page faults overheads
2. Page granularity

Solution

Use cache coherence (w/ FPGA)

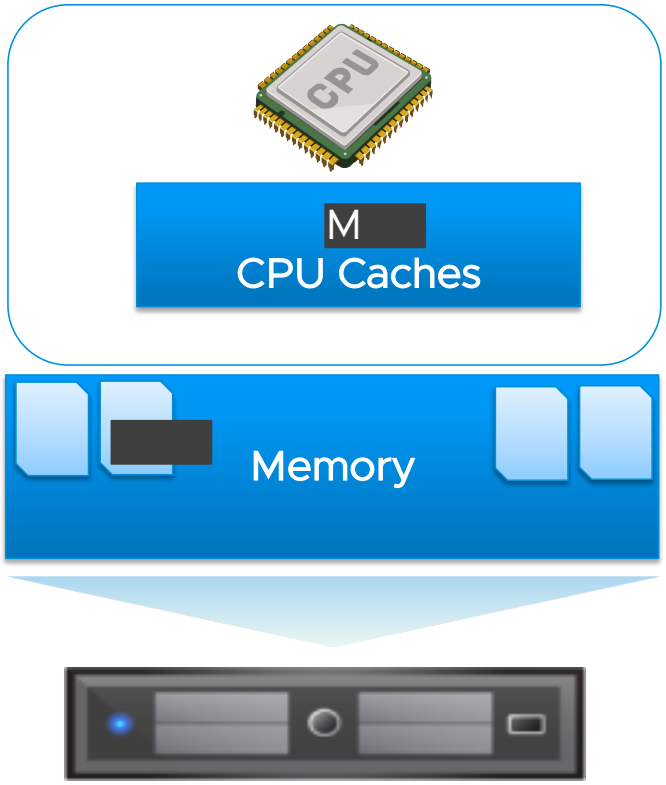
Implications:

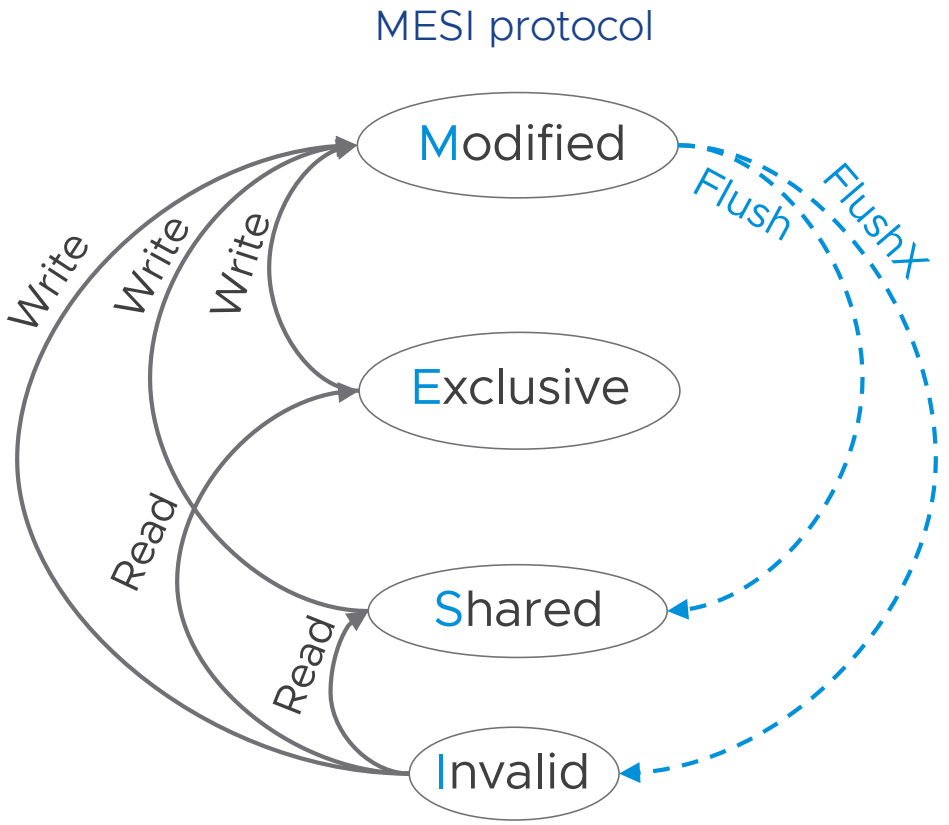
1. Reduce page faults
2. Cache line granularity

Impact

1. Remote memory
2. Live migration
3. Code analysis
4. Security
5. …

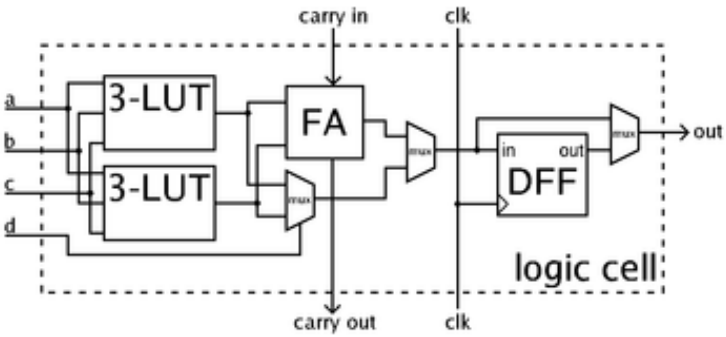
Key idea: use cache coherence for tracking memory accesses

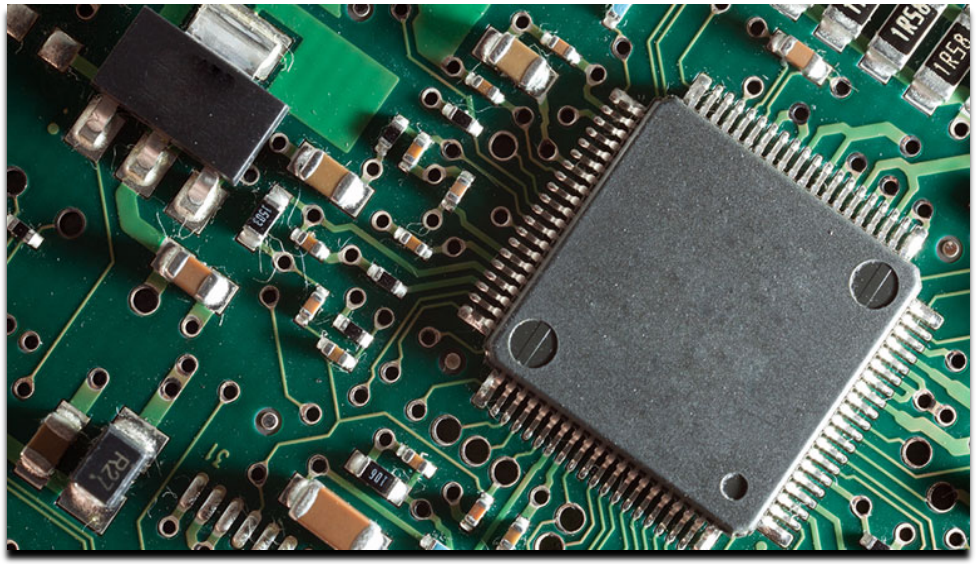




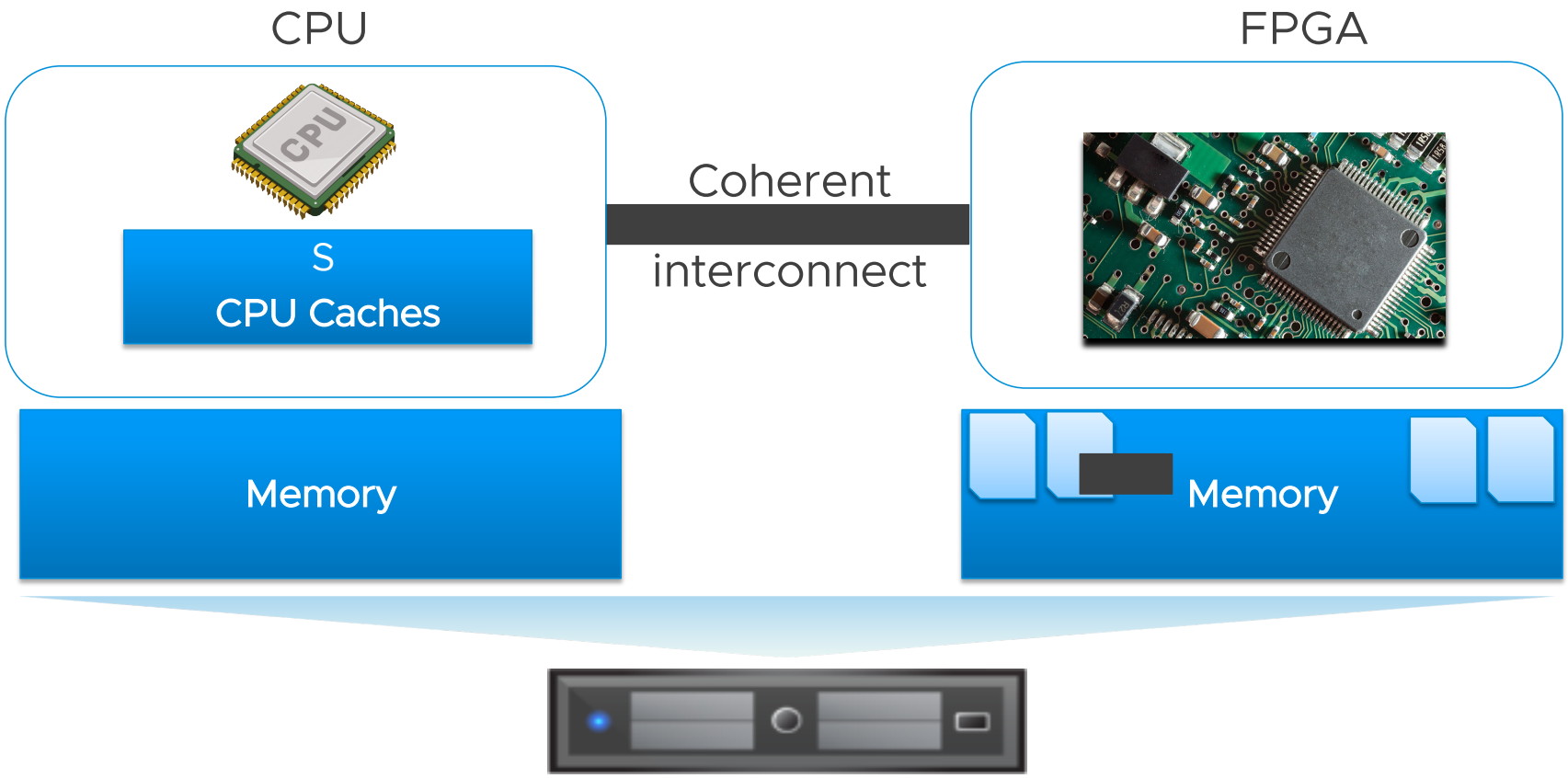
Field Programmable Gate Array (FPGA)

- Integrated circuit
- Reconfigured after manufacturing
- “Program the hardware”

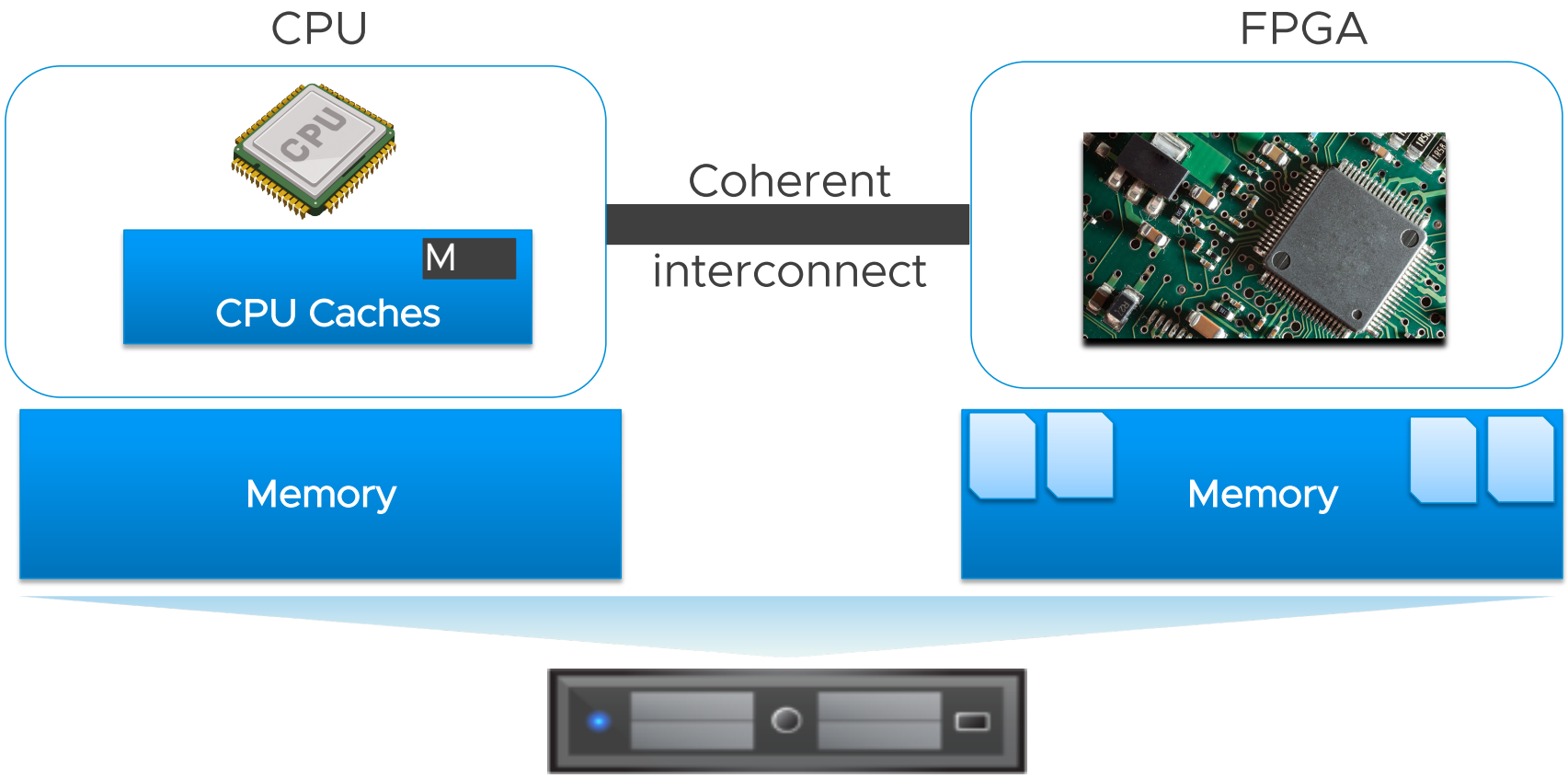




Key idea: use cache coherence for tracking memory accesses



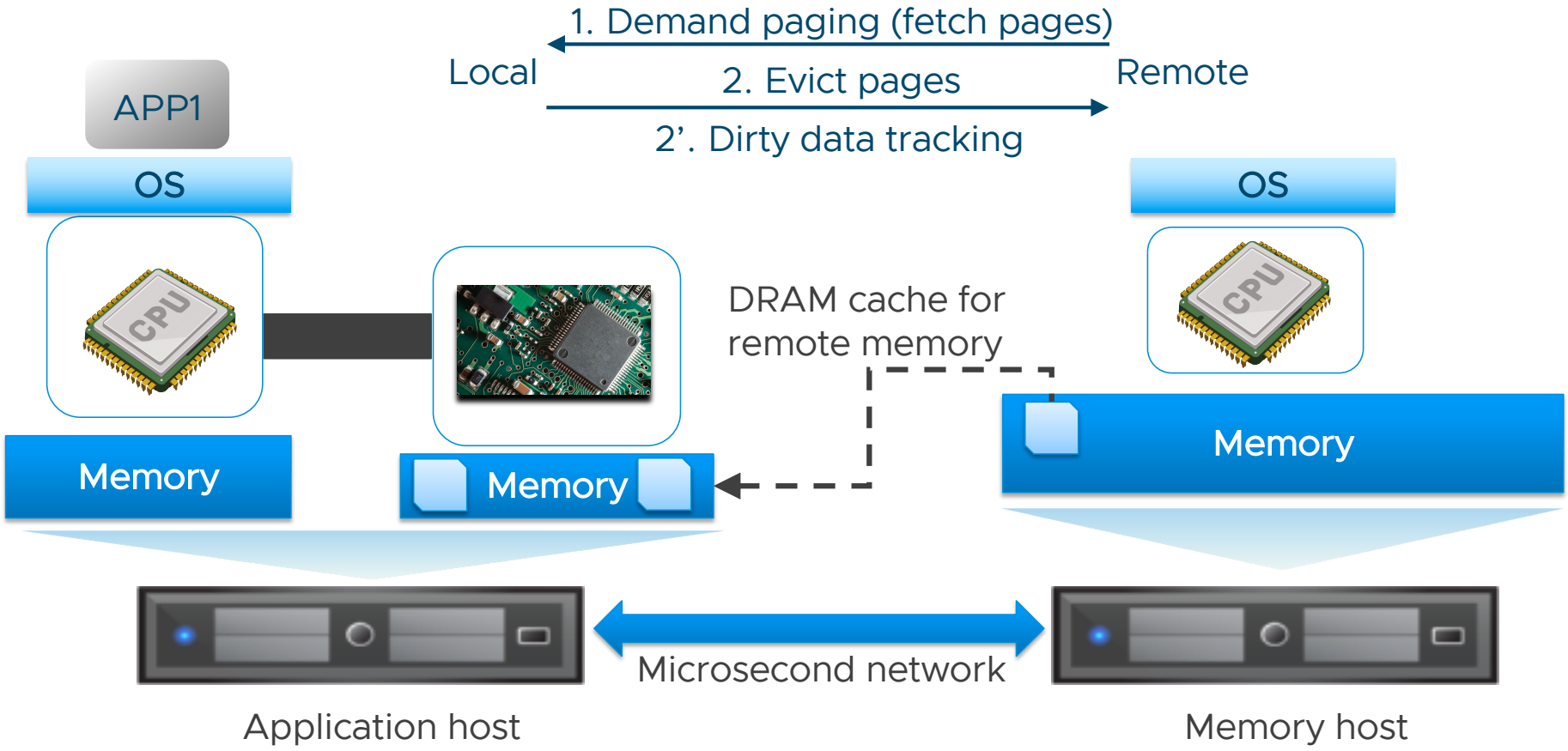
Key idea: use cache coherence for tracking memory accesses



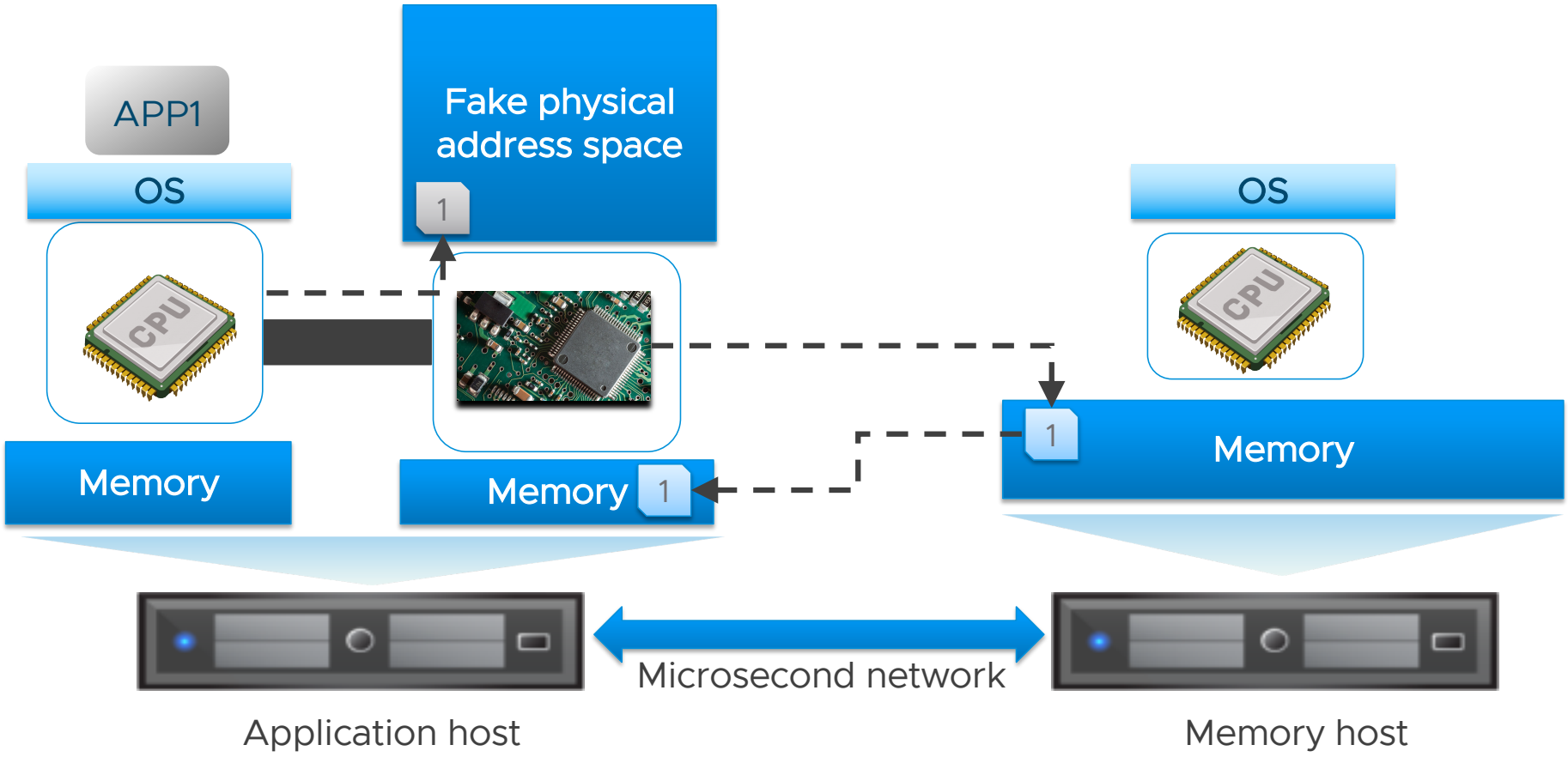
FPGA primitives

- Read and write tracking using cache coherence
  - Data pages
  - Code pages
  - OS pages (page tables)
- Local and remote copy
- In-place transform

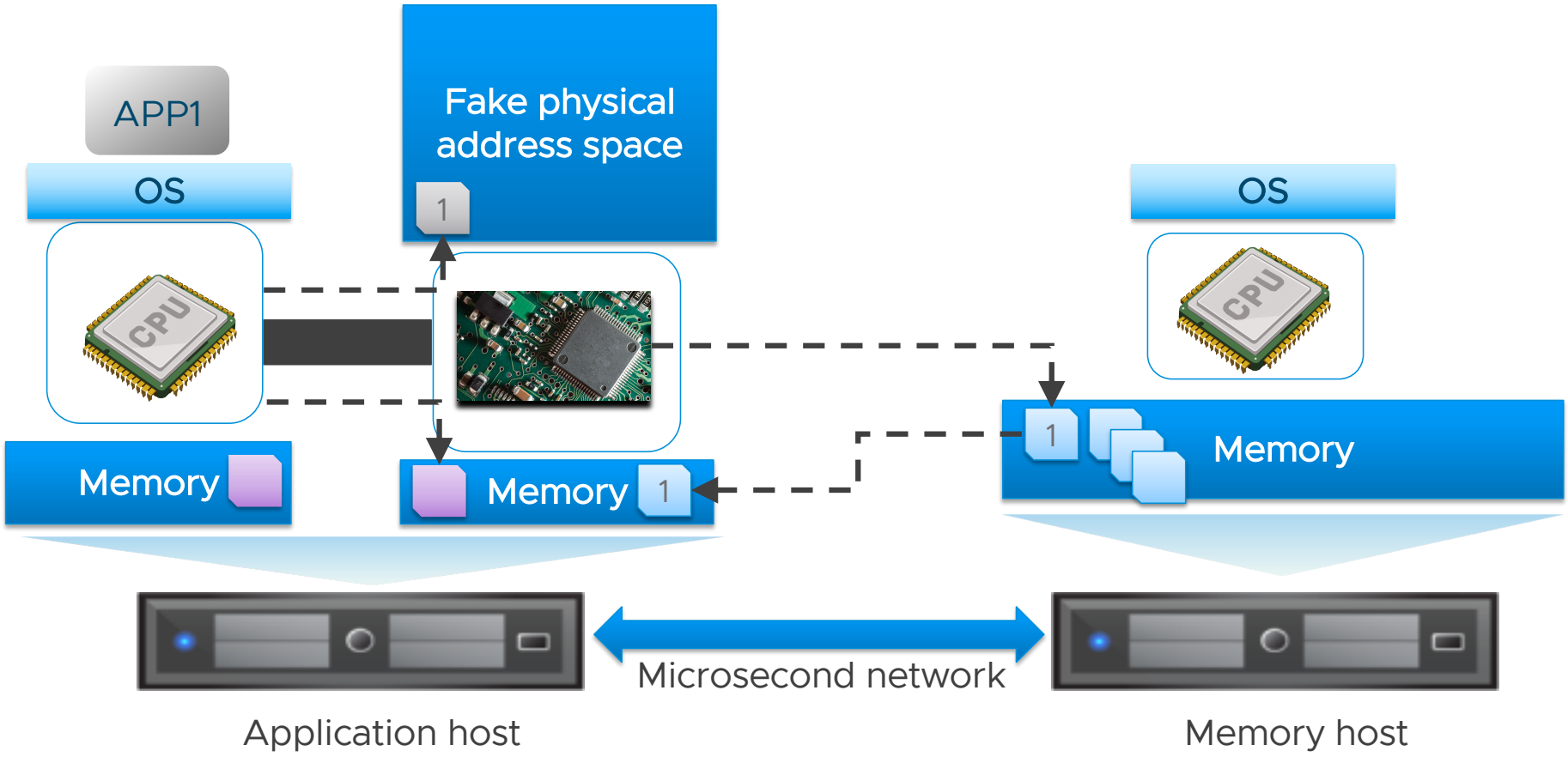
Remote memory using a cache coherent FPGA



1. “Demand paging” using a cache coherent FPGA



1. “Demand paging” optimization: translation triggered prefetching



Use cache coherence traffic to track application memory accesses

Problem

Remote memory systems use page faults

Overheads:

1. Page faults overheads
2. Page granularity

Solution

Use cache coherence (w/ FPGA)

Implications:

1. Reduce page faults
2. Cache line granularity

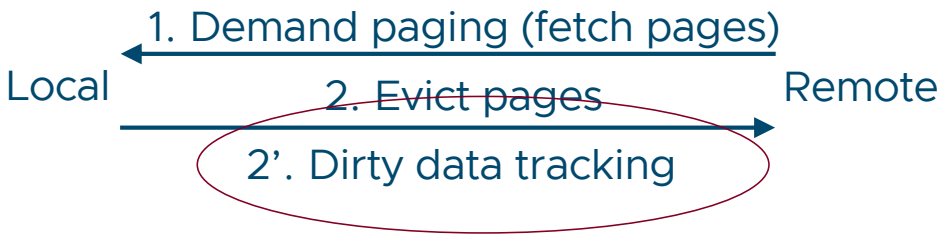
Impact

1. Remote memory
2. Live migration
3. Code analysis
4. Security
5. …

Benefits

- Reduce (write) page faults
- Reduce dirty data amplification (cache-line)
- Determine access pattern, smart prefetching
- Downside: FPGA memory slower than CPU memory

Preliminary results (emulated)



- Redis-random:
  - 89% less dirty data (cache-line vs. page granularity)
  - 27% faster than WP
- Redis-seq:
  - 50% less dirty data (cache-line vs. page granularity)
  - 4% faster than WP

Other use cases

Use case	Page type	Primitive
Live migration	Data, page tables	Read/write tracking, remote copy
Compression, Encryption	Data	Read/write tracking, in-place transform
Replication, Checkpointing	Data	Write tracking, local/remote copy
Near-memory compute	Data	Read/write tracking, in-place transform
Code analysis	Code	Read tracking
Security	Data	Read/write tracking

Use cache coherence traffic to track application memory accesses

Problem	Solution	Impact
Remote memory systems use page faults	Use cache coherence (w/ FPGA)	Other use cases

Thank you!

Contact: icalciu@vmware.com