Karlsruhe Institute of Technology

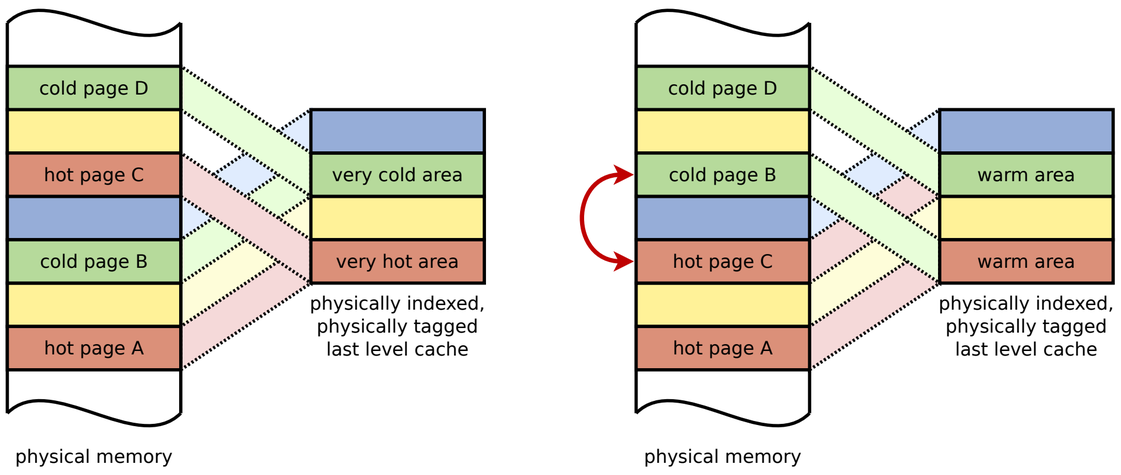
System Architecture Group
Department of Informatics
http://os.ibds.kit.edu/

Hardware-Assisted Virtual Memory Management

Improving page replacement and migration with on-line memory access information

Raphael Neider and Frank Bellosa

1. Motivation

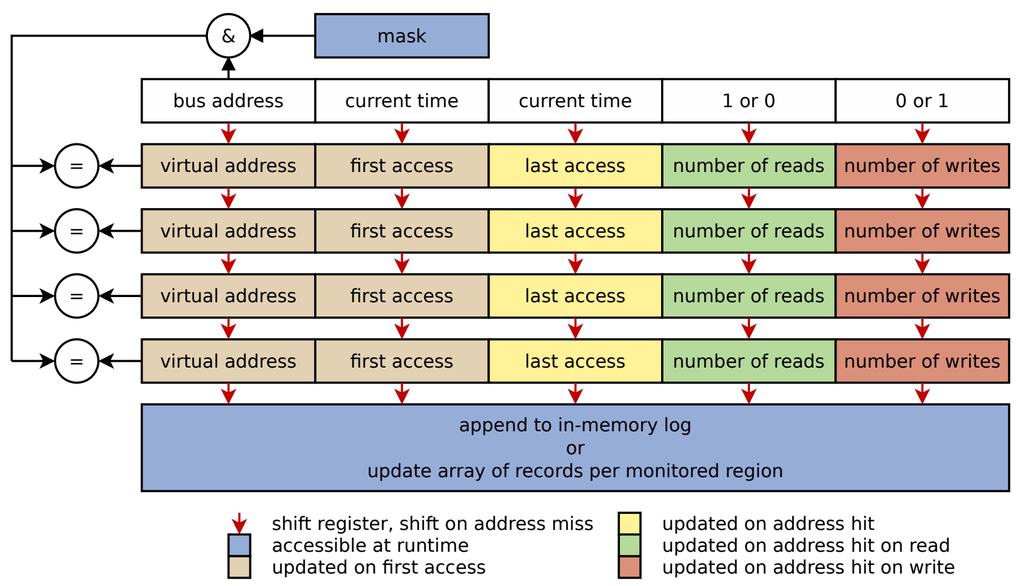


- Operating systems with virtual memory support are common
- Page placement (and migration) policy required
  - Aware of caches, NUMA, memory technologies
- Page replacement policy required
  - Optimal, least frequently used (LFU), least recently used (LRU)
- Only *referenced* and *dirty* bits available
  - No access frequency or count ➜ no LFU
  - No time of last access ➜ no LRU
  - No type of use (read-only/-mostly vs. write-often)
  - No data on *physical* memory accesses
- Memory traces for off-line analysis desired
  - Only available from simulations ➜ short time frame
- Thesis
  - More information on memory usage helps virtual memory management perform better!

2. Requirements

- Support variety of policies
  ➜ Record timestamps, reads, and writes
- Tracing every memory access is too costly
  ➜ Find shortcuts
- Cache hits are irrelevant
  ➜ Monitor activity after caches / at memory controller
- 100 % accuracy is not required
  ➜ “Batch” memory access information
  ➜ Access records per page are (usually) sufficient
- Live feedback to OS and software is required
  ➜ Provide efficient interface
- Address ranges and granularities should be configurable
  ➜ Allow different policies per memory technology
  ➜ Allow fine-grained examination of cache line utilization

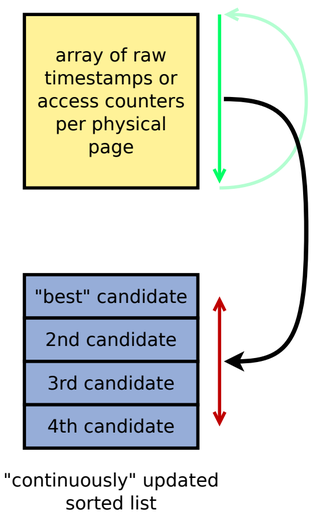
3. Memory Profiling Unit (MPU)



- Record timestamp of first and last access per page
- Record number of reads and writes per page
- Keep n such records in associative memory (e.g., 16 ways)
- Replace entries via FIFO
- Write oldest entry to log on removal
  ➜ Data in the log will never be too old
- Scan/consolidate/write-back log in software

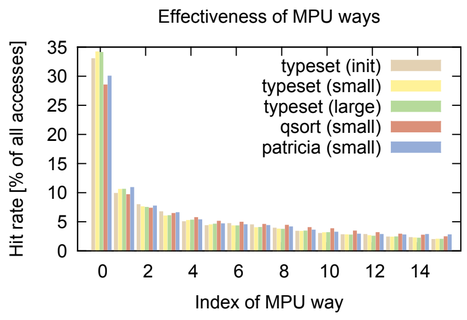
4. Hardware-Assisted Candidate Selection

- Hardware remembers m (e.g., 4) best candidates
  - Candidates are the pages with
    - smallest timestamp (LRU)
    - least accesses (LFU)
    - most accesses (migration)
  - Remember largest entries on updates
    - Requires aging policy to prevent overflows
  - Search smallest entries on
    - update of largest remembered entry
    - reset of other records
- One unit per physical memory region (memory technology, NUMA node)
- One unit to record misses per “cache page color”
  ➜ Place new data in uncontended cache areas
  ➜ Migrate heavily used pages from contended areas



5. First Results of an FPGA-Based Prototype

- “Real” hardware is inaccessible
- Implemented on the OpenProcessor platform
  - SoC on FPGA devel. board
  - RISC CPU @ 50 MHz
  - 64 MiB DDR SDRAM



- > 98 % hit rate with 16 MPU ways
- Median candidate selection cost
  - 2-handed clock: 13 770 µs
  - Hardware LRU: 211 µs
- Up to 90 % less swap-ins