vCache: Architectural Support for Transparent and Isolated Virtual LLCs in Virtualized Environments

Daehoon Kim*, Hwanju Kim†, Nam Sung Kim*, and Jaehyuk Huh‡

*University of Illinois at Urbana-Champaign, †University of Cambridge, ‡KAIST
Virtualized Environments

- Resource virtualization for VM consolidation
  - Providing an illusion of having dedicated physical resources to a VM
  - e.g., CPU, memory, I/O devices
  - **LLC (Last-level Cache) is not virtualized**

- **LLC virtualization**
  - **Transparency**: controllable by guest OS
    - e.g., page coloring
  - **Isolation**: isolated capacity

Providing transparent and isolated virtual LLCs to VMs
Background: Page Coloring

- Page coloring: a software-based LLC placement technique
  - OS controls placement of a page in LLC by manipulating a physical address
  - Balancing cache accesses by spreading data across the entire LLC
  - Partitioning LLC to avoid cache contention

Page coloring techniques
- [MICRO 08’][HPCA 09’][EUROSYS 09’]...
- Mainstream OS
  - e.g. Solaris, FreeBSD, WindowNT
Background: Memory Virtualization

- An additional address translation for consolidation
  - Guest-Virtual Address (GVA) to Guest-Physical Address (GPA) by guest OS
  - GPA to Host-Physical Address (HPA) by hypervisor

- Page coloring of a guest OS becomes ineffective with HPA-indexed LLC
Page Coloring in Virtualized Systems

- Example: pollute buffer mechanism [*MICRO 08’*
  - Classify cache unfriendly pages as pollute pages
  - Map pollute pages to an isolated LLC region (pollute buffer)
  - *Avoid LLC contentions by pollute pages*
Page Coloring in Virtualized Systems

- Example: pollute buffer mechanism \textit{[MICRO 08']}
  - Classify cache unfriendly pages as pollute pages
  - Map pollute pages to an isolated LLC region (pollute buffer)
  - \textit{Avoid LLC contentions by pollute pages}

\begin{center}
\begin{tikzpicture}
  \node (hypervisor) at (0,0) {Hypervisor};
  \node (guest) at (3,0) {Guest OS};
  \node (gpa) at (5,0) {GPA};
  \node (gva) at (-3,0) {GVA};
  \node (llc) at (8,0) {LLC};
  \node (hp) at (8,-2) {HPA};
  \node (pollute) at (-3,-2) {pollute pages};
  \node (buffer) at (8,-4) {pollute buffer};

  \draw (gva) -- (gpa);
  \draw (gpa) -- (hp);
  \draw (hp) -- (llc);
  \draw (llc) -- (buffer);

  \node at (2,-1) {Page coloring (pollute buffer)};
  \node at (2,-2) {Memory virtualization};
\end{tikzpicture}
\end{center}
Page Coloring in Virtualized Systems

- Example: pollute buffer mechanism \([MICRO \ 08']\)
  - Classify cache unfriendly pages as pollute pages
  - Map pollute pages to an isolated LLC region (pollute buffer)
  - Avoid LLC contentions by pollute pages

![Graph showing IPC improvement over memory ballooning occurrences]

# of occurrences of memory ballooning
Interference by VM Consolidation

- Unexpected interference by co-running VMs
  - Page color preservation alone cannot provide benefits of page coloring in consolidated environments
vCache: Transparency

- **GPA-indexed HPA-tagged virtual LLC**

- **Use GPA for indexing LLC**
  - GVA-to-GPA is managed by guest OS
  - Allow VMs to control LLC placement
  - GPA can be obtained when translating address w/o additional steps

- **Maintain Full HPA** for tag matching
  - Extend each tag to store HPA color (e.g., 7 bits for 128 colors)
vCache: Transparency

- **Use HPA** for indexing pages that cannot use GPA as LLC index
  - i.e., hypervisor pages and shared pages

- Extend TLB entry
  - **GPA color** (e.g., 7 bits for 128 colors)
  - **Page status** (1 bit)

- Cache coherence support
  - Coherence requests maintain GPA color bits
  - Extend L1/L2 cache tags to store GPA color for write-back to LLC
vCache: Isolation

- Isolated capacity: **VM-based LLC partitioning** in way granularity
  - **vLLC partition table**
    - Maintain vLLC capacity mandated by the contract with its user
    - Set by hypervisor
  - Work-conserving policy: Unreserved/unused capacity is shared
  - Modified LRU: Choose a cache line belongs to VMs with more cache lines than allocated capacity as a victim

**VM-1 and VM-3 have more cache lines than allocated capacity**

vCache chooses a cache line closer to global LRU position
Experimental Methodology

- Running Xen hypervisor on SIMICS
  - Pollute buffer mechanism [*MICRO 08’*] and ULCC [*PPoPP 11’*]
    - ULCC: User-level page coloring interface
  - 1 way = 1 MB: 4MB, 8MB, and 12MB LLC
  - Workloads
    - Mixes of SPECCPU benchmarks for pollute buffer
    - Pluto benchmarks for ULCC
  - Change GPA-to-HPA mappings with memory ballooning
Results with Single VM

- IPC improvement – after initial booting

HPA-indexed LLC shows almost same results with vCache after initial booting
Results with Single VM

- IPC improvement – after ballooning

For hmmer, vCache shows 17% IPC improvement while HPA-indexed LLC shows less than 1% improvement after memory ballooning
Results with Multiple VMs

- IPC improvement: two VMs (VM1: 4MB, VM2: 8MB)
  - Each of which runs with pollute buffer and ULCC with GPA-based indexing

GPA-based indexing alone cannot preserve effectiveness of page coloring in consolidated environment
Results with Multiple VMs

- IPC improvement: two VMs (VM1: 4MB, VM2: 8MB)
  - Each of which runs with pollute buffer and ULCC with GPA-based indexing

For soplex, sLLC further improves performance by 37% points while it degrades performance of co-running VM by 23% points
Results with Multiple VMs

- IPC improvement: two VMs (VM1: 4MB, VM2: 8MB)
  - Each of which runs with pollute buffer and ULCC with GPA-based indexing

vCache preserves the effectiveness of page coloring by guest OS with isolated capacity
vCache provides *a transparent and isolated virtual LLC* to a VM

- Transparency: GPA-indexed HPA-tagged
- Isolation: VM-based LLC partitioning in way granularity
- vCache preserves the page coloring policy deployed by each VM as non-virtualized systems