Punch holes in your large pages for benefits (performance, flexibility) with less overhead

Paper
DOI: 10.1109/ISCA45697.2020.00079

Contact Us
chang.hyun.park@it.uu.se

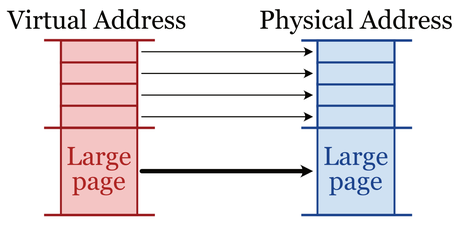
Perforated Pages: Supporting Fragmented Memory Allocation for Large Pages (ISCA 2020)

UPPSALA UNIVERSITET

Chang Hyun Park, Sanghoon Cha, Bokyeong Kim, Youngjin Kwon, David Black-Schaffer, and Jaehyuk Huh

KAIST

1 Large pages for performance

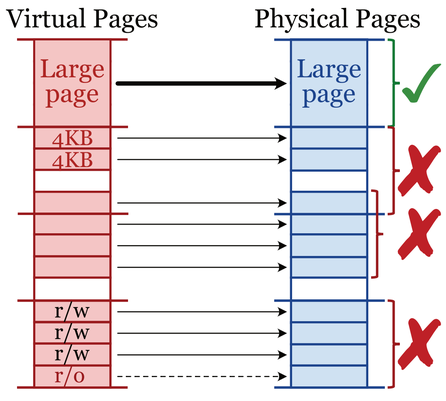


- Large page mapping:
  - 512x TLB coverage ⬆
  - ~68% performance ⬆[1]

But

2 Strict Requirements

Cause

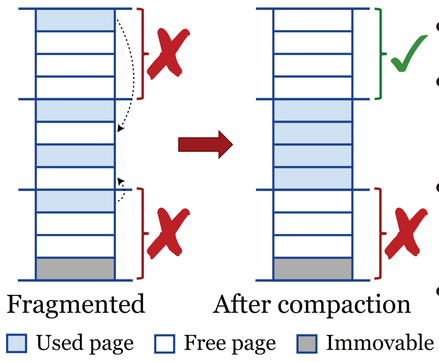


512 contiguous 4KB pages

2MB alignments

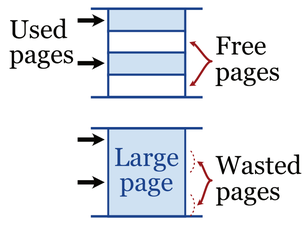
Homogeneous properties

3 Overhead 1: Data movement



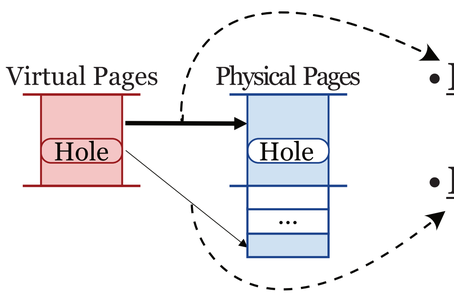
- Compact to create large pages
- Takes up to 35% of execution time[2]
- Immovable pages hinder compaction
  - Kernel allocations, I/O buffers, etc.
- Existed in 50% of 2MB regions

4 Overhead 2: Wasted Memory



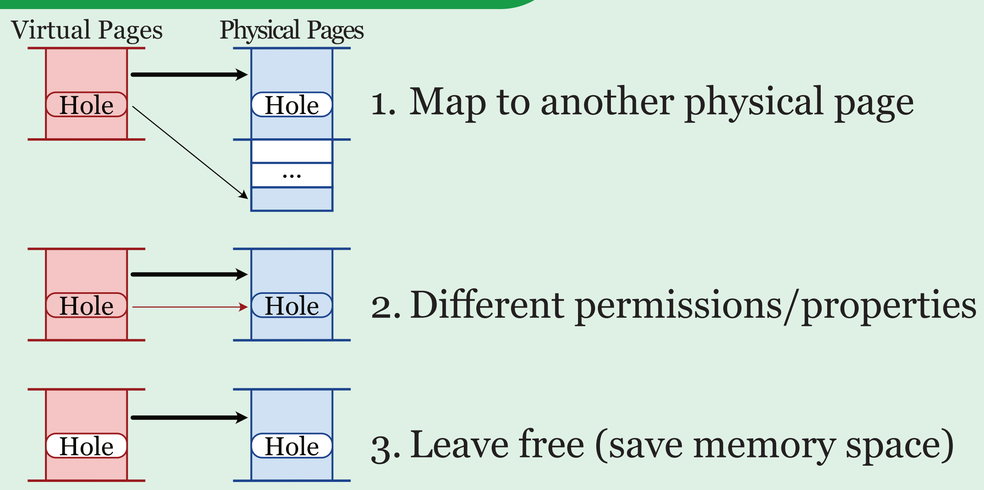
- Sparse access/use → Wasted space
- Example: Redis using large pages
  - 20% more memory consumption
  - 45% fewer TLB misses → 29% higher throughput

5 Our Solution: Perforated Pages



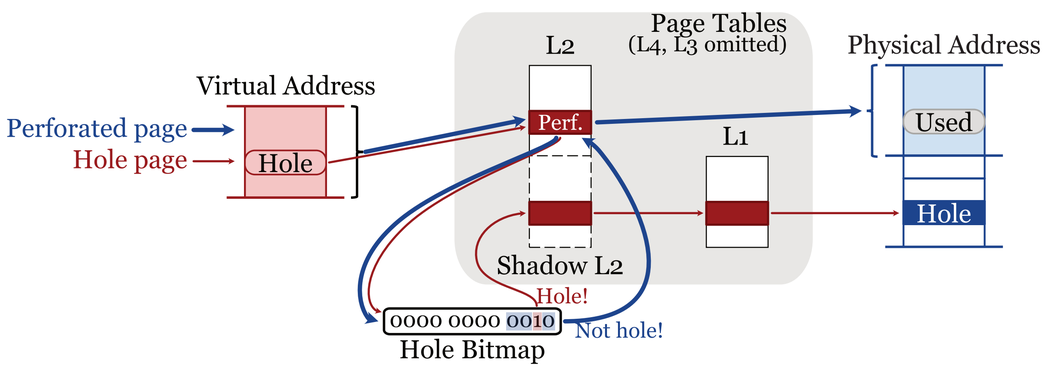
- Perforated page (2MB): maps common case
  - Large page-like **coverage** and **mapping**
- Hole Page (4KB): maps special case
  - **Flexibility** for specific mappings

Example Usages ⬇



1. Map to another physical page
2. Different permissions/properties
3. Leave free (save memory space)

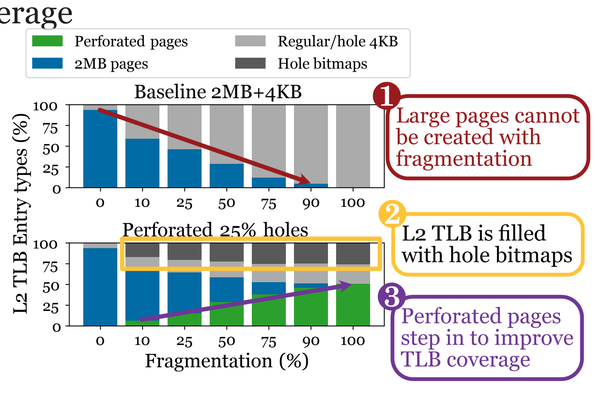
6 How it works - Page table



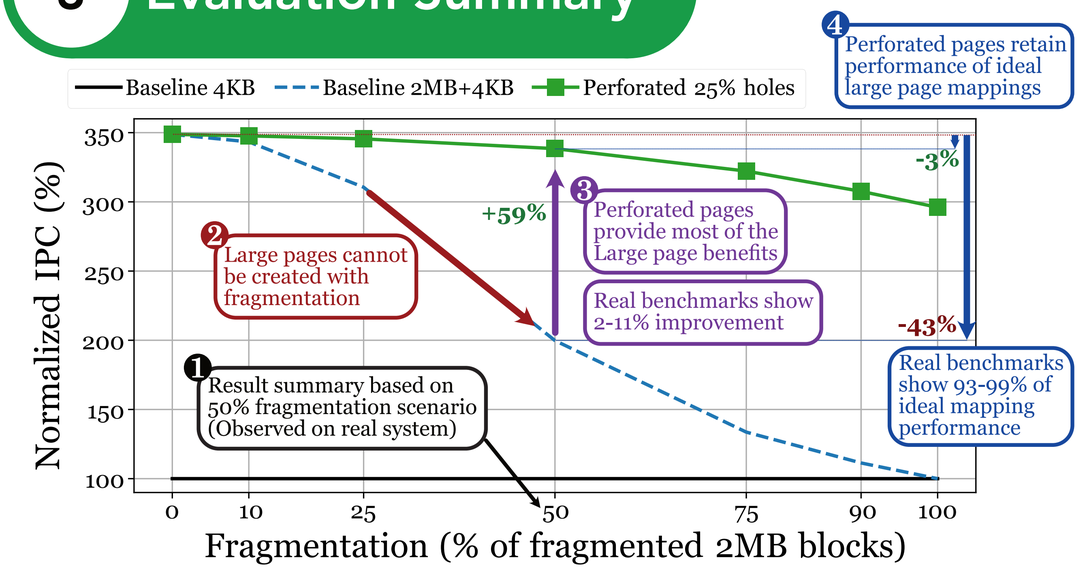
- Hole Bitmap: Identifies hole pages - stored in DRAM
- Shadow L2: provides an additional pointer in the page table
  - Main L2 PTE points to perforated page mapping (2MB)
  - Shadow L2 PTE points to L1 PT that holds hole page mappings (4KB)

7 How it works - L2 TLB

- Cache and translate perforated/hole pages at L2 TLB
- L2 TLB caches additional entries:
  - Perforated mappings: 2MB page-like coverage
  - Hole mappings: 4KB flexible mappings
  - Hole bitmaps: cached in L2 TLB
- Optimization: bitmap filter
  - Identify non-hole regions
  - Stored in perforated mappings



8 Evaluation Summary



[1] Margaritov et al., MICRO '19
[2] Parwar et al., ASPLOS '18