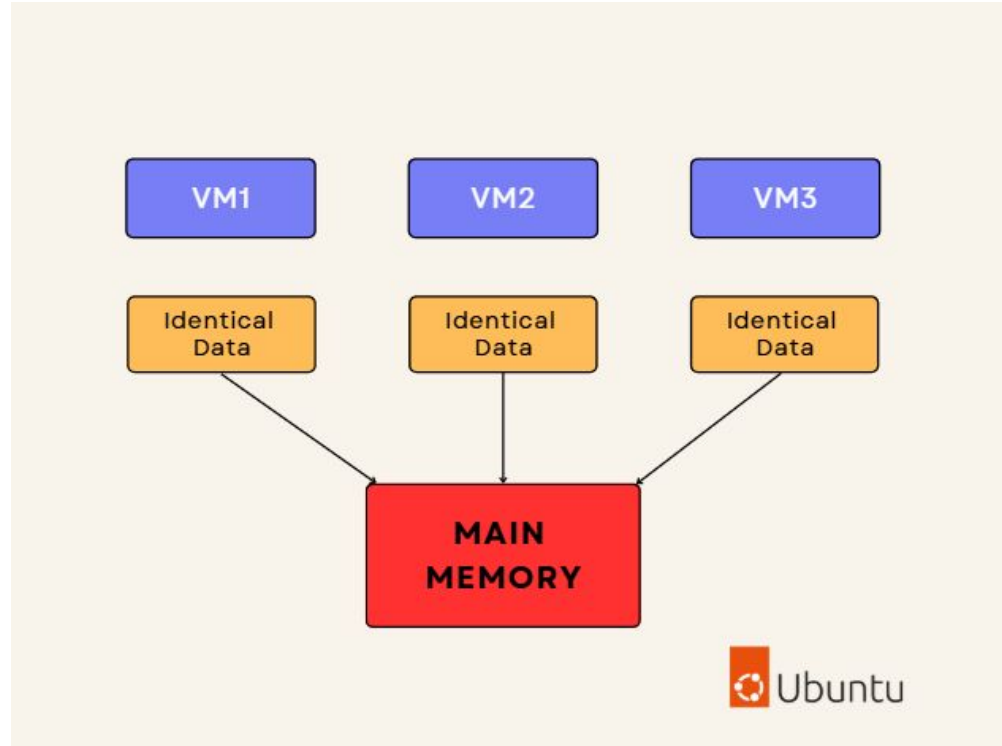


KSM++:
Using I/O-based hints to make
memory-deduplication scanners more efficient

RESolve, 2012

Arohan Hazarika, 22B3948

Virtual Machines and Content Redundancy



Too Slow to Catch: The Pitfall of Traditional Deduplication

Motivation and Problem Statement

Traditional memory deduplication like KSM misses short-lived redundant pages due to slow scans and lack of semantic insight into VM memory usage

Challenges

Finding short-lived duplicate pages, limited visibility into VM memory, scan speed vs. system load, unstable data structures, and sudden spikes in I/O hints

How to Solve this problem?

Main Idea

Use I/O-based hints to guide KSM's scanner to short-lived, high-potential duplicate pages

Solution Elements

- Instrument VFS to tag memory pages during disk I/O
- Insert hinted pages into a bounded, circular hint stack
- Process hints in interleaved spurts alongside regular KSM scans
- Immediately insert hinted pages into the unstable tree
- Map hinted pages read-only to detect post-insertion modifications

Deduplication effectiveness with varying stack sizes

How does the size of the hint stack affect deduplication effectiveness over time?

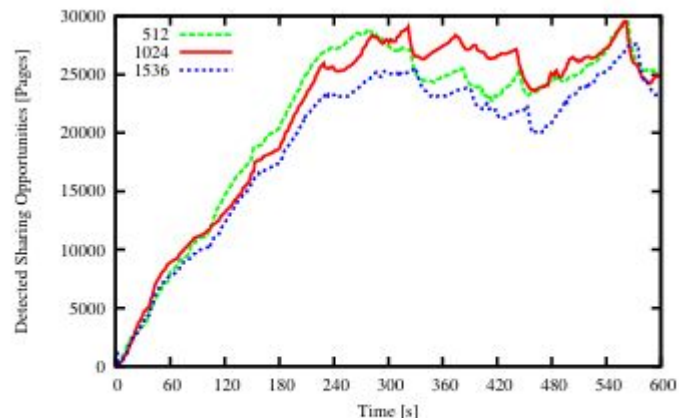


Figure 5. Deduplication effectiveness with varying stack sizes.

A stack size of 1024 balances between losing hints (512) and processing stale ones (1536), making it optimal for bursty I/O workloads.

Anything else we can do?

- Add a **per-page hint score** that increases with repeated hints and decays over time, prioritize high-score pages for merging
- Avoids wasting scans on one-time hints, rather better to focus on pages with frequent hints
- **Question:** Does scoring-based hint prioritization improve merge efficiency compared to LRU stack based method?
- **X-axis:** Hint prioritization method (LRU vs scoring)
- **Y-axis:** Merge success rate, time-to-merge
- **Hypothesis:** Scoring helps merge more useful pages faster, with lower overhead

Summary:)



**SCAN
EVERYTHING, ALWAYS**

**ONLY SCAN
ANONYMOUS PAGES
MARKED MERGEABLE**

**PRIORITIZE
PAGES THAT HAVEN'T
CHANGED RECENTLY**

**USE I/O ACTIVITY
TO HINT SHORT-LIVED
SHARING
OPPORTUNITIES AND AVOID
SCANNING IRRELEVANT PAGES**

