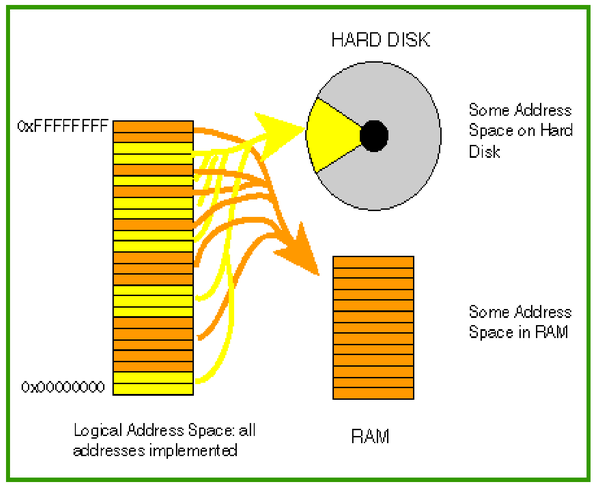
SSD-Aware Virtual Memory Management

Christopher Feener **Xuechen Zhang**
Washington State University Vancouver

Why is Virtual Memory Management Needed?

- Virtual memory can represent the memory that cannot be held in pure physical memory (DRAM). However, this memory must be mapped to and from physical memory, and that is why virtual memory management (VMM) is needed.
- Specifically for the purposes of this research, VMM is needed for deciding which physical swapping device (SSD vs disk) a virtual-memory page is mapped to.



Challenges of VMM using SSDs

Why are we using SSDs for VMM?
Incorporating SSDs into Virtual Memory Management can increase the bandwidth of the swapping system, and reduce access latency compared to disk.

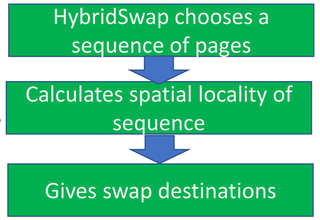
Challenges:
Unfortunately, frequent writes to SSDs could shorten the lifetime of SSDs to a thousandth of normal.

Existing Solutions

- SSDs have been used for enhancing the speed of hard disks either by being a buffer cache between DRAM and hard disk, or by being used in parallel with hard disks.
- Hard disks have also been used intentionally for reducing SSD writes.

Our Solution: HybridSwap

- Disk manages sequential reads and SSD manages random reads.
- SSD provides enhanced performance, while disk allows a longer lifetime for SSD, thus combining the advantages of both.



Integration of Temporal Locality and Spatial Locality

- Pages with the least temporal locality are the most likely to be swapped out.
- Spatial locality must also be used to make certain that swap-in latency and page-fault latency are not increased.
- Disk swap manages sequences with strong spatial locality and weak temporal locality, while SSD swap manages page sequences with weak spatial locality.
- A variant of the LRU replacement algorithm is used, similar to the 2Q replacement.

Evaluation of Spatial Locality of Page Sequences

- In order to reduce overhead and minimize time to detect page accesses, HybridSwap only records page accesses when there is a page fault.
- The lifetime of a page at swap-out is considered to be the most recent time between page accesses (swap-in and current time).
- A sequence of pages is swapped to disk only when both the difference between any two pages' access times is less than the system's current average lifetime, and if some anonymous pages in the sequence have access times.

Scheduling Page Swapping

- In the range of the last N pages in the inactive list of pages, a process's pages are replaced in each swap.
- N must be small enough to avoid swapping too many recently used pages, but large enough to add disk efficiency.

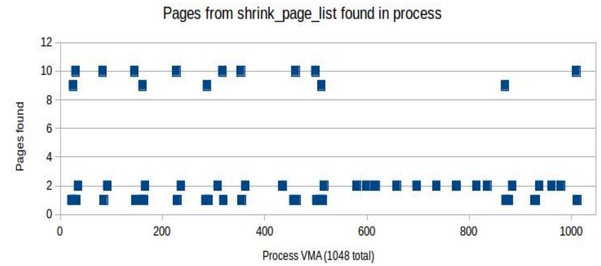
Performance of HybridSwap over SSD-Swap

Writes to SSD	16 - 40% fewer in HybridSwap
Speed	-0.8 – 7.1% faster in HybridSwap

Evaluation

- Host computer specifications: 64-bit linux, 8 GB RAM
- Qemu VM image size: 20 GB, 1 GB RAM, Linux kernel 4.11
- Type of file converted by ImageMagick: 4.3 GB PSB to JPEG, resized to 1% of original.

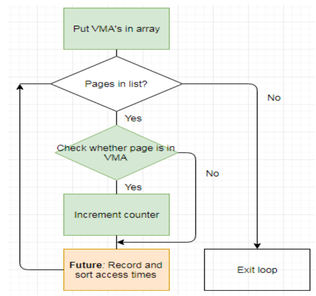
Experimental Results



Data:
This data is from a single call of shrink_page_list (SPL).

Observations:
The vast majority of VMA's had zero pages from SPL, The rest hovered around 1 or 2, sometimes peaking at 9 or 10, usually at the beginning of the graph.

Implementation



(Green denotes revisions)
Using page information:

- Different pages accessed at nearly the same time and place could be further tested.
- Then, if a page is accessed often, it is swapped to disk.
- If not, it is sent to SSD.

Conclusion and Future Work

- HybridSwap benefits from the disk's high throughput while doing sequential access, and from SSD's low latency while doing non-sequential access.
- Writes to SSDs are significantly reduced while keeping similar performance to fully-SSD swap, resulting in a lengthened lifetime for the SSDs.
- **Spatial locality** can be used in conjunction with temporal locality to determine the optimal means of swapping pages between SSDs and disks.
- **In future work**, access times may be sorted from least to greatest, and then split into two sections: The first section going to SSDs, and the second to hard disks.