SmartKSM: A VMM-based Memory Deduplication Scanner for Virtual Machines

Shengyang Sha[1], Jianxin Li, Nan Li[1], Wuyang Ju[1], Lei Cui[1], Bo Li

School of Computer, Beihang University, {*shasy*, *lijx*, *linan*, *juwy*, *cuilei*, *libo*}@act.buaa.edu.cn

Motivations: As virtualization technology has been widely used in infrastructures such as clouds, limited main memory size has become one of the primary bottlenecks in virtualized environments. There are urgent needs to improve the utility of main memories. Content-based page sharing, a basic technique firstly proposed by VMware's ESX Server [2] and widely used in academic and industrial field, scans memory footprints in order to search equal pages (pages with equal contents) and then merges those equal pages as one copy-on-write page.

Many work and researches have focused on how to scan memory footprints to get as many equal pages as possible. ESX [2] uses hash values as hints to detect equal pages. KSM [1] maintains its pages in red-black trees using page contents as keys to search and insert. Since page contents may change, every page needs to be recalculated after a period of time. ESX scans pages in random order while KSM adopts a linear scan method. However neither of them proposes a targeted strategy to detect equal pages more efficiently. As the amount of pages increases, the average scan interval increases, leading to a decreased chance to find equal pages. In KSM this situation could become worse due to the degeneration of the unstable tree [3]. XLH [3] uses I/O-based hints to scan pages related to VM disks preferentially as these pages are good sharing candidates. This strategy is effective only to disk I/O-intensive applications while nowadays network I/O is not negligible and it's common that applications occupy a large proportion of memory footprints.

Design: Our key idea is to divide memory footprints into several sets. The higher sharing potential a page set has the higher priority the page set to be scanned. The division should meet the following two conditions: (1) pages with equal contents should be divided into the same set; (2) the distribution of sharing potentials among page sets should be regular so that we can decide the best set to be scanned.

Through theoretical analysis and experiments, we have found two effective ways of division, page-type aspect and process aspect.

The page type aspect is how a page is used in guest OS. Kloster et al. [4] have shown that pages' sharing potential, which is the possibility of pages having equal contents, varies from one page type to another. Their research shows that using content-based page sharing technique, the proportion of shared cache pages among all the shared pages is over 50% and can reach up to 93% during the kernel compilation workload while the proportions of other page types are not that significant. We conduct further analysis of page sharing patterns on VMs' page types and discover that the distribution of page types among sharing pages is highly aggregative. In other words, the sharing possibility between two pages of different page types is rather small compared with it between two pages of the same page type.

Besides the page type aspect, we also introduce the process aspect to detect pages related to a process in a VM, which extends the semantic information to application level. As we notice that similar processes in different VMs have a high possibility of having equal pages. Since memory-intensive applications, such as Memcached, could take lots of main memory while their memory footprints are application related, scanning VM memory in the process aspect will increase the possibility of finding equal pages.

Combing these two methods, we designed a VMM-based memory deduplication scanner, using semantic information of VM memory footprints to detect better sharing candidates and reduce the performance overhead. KVM kernel module is modified to maintain the information of VMs and their processes, and to provide the memory deduplication scanner with ways of obtaining semantic information of VM memory. As to the decision-making of what page set to scan, we implement a fixed policy based on the knowledge of page sharing potentials we gain from preliminary experiments. In the future, we'll take performance indicators into account to make it dynamically adaptive to multiple kinds of workloads.

Evaluation: Our evaluation focuses on the speed of detecting sharing opportunities and the overhead caused by our scanning procedure. We have conducted experiments on compiling the Linux kernel in four VMs. The result shows that we can detect 30% more sharing opportunities than KSM while the overhead is lower than KSM in about 20 seconds.

References:

[1] Arcangeli, A., et al. Increasing memory density by using KSM. In *Proceedings of the linux symposium 2009*.

[2] Waldspurger, C. A. Memory resource management in VMware ESX server. OSDI '02.

[3] Miller, K., et al., XLH: More effective memory deduplication scanners through cross-layer hints. *USENIX ATC 2013*.

[4] Kloster, J. F. et al., Determining the use of inter-domain shareable pages using kernel introspection. Technical report, Aalborg University, 2007.

[1] Students; Shengyang Sha will present. We can demo our SmartKSM system

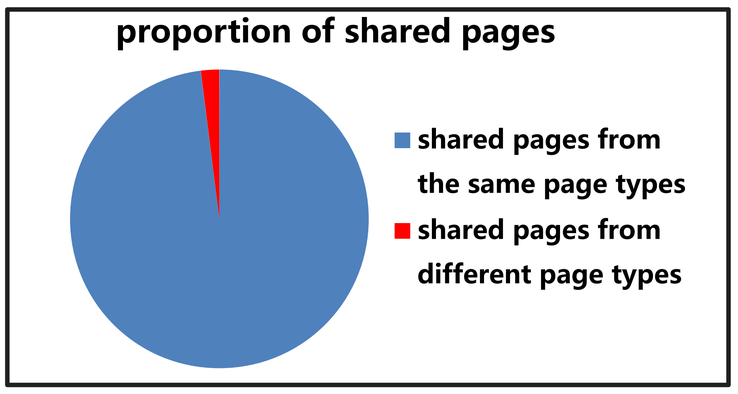
SmartKSM: A VMM-based Memory Deduplication Scanner for Virtual Machines

Shengyang Sha, Jianxin Li, Nan Li, Wuyang Ju, Lei Cui, Bo Li
School of Computer Sci&Eng, Beihang University

ACT Lab

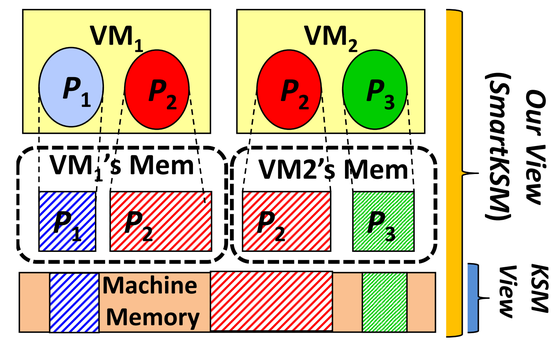
Sharing Memory Pages in Virtualized Environments

Problems Statement:
- **Low Efficiency**: overall page comparison, blind
- **Performance Degradation:** e.g. the degeneration of the unstable tree in KSM



Goal:
- Make memory deduplication scanner more efficient with a deep insight into the memory usage in guest Oses
- Improve short-lived sharing opportunities

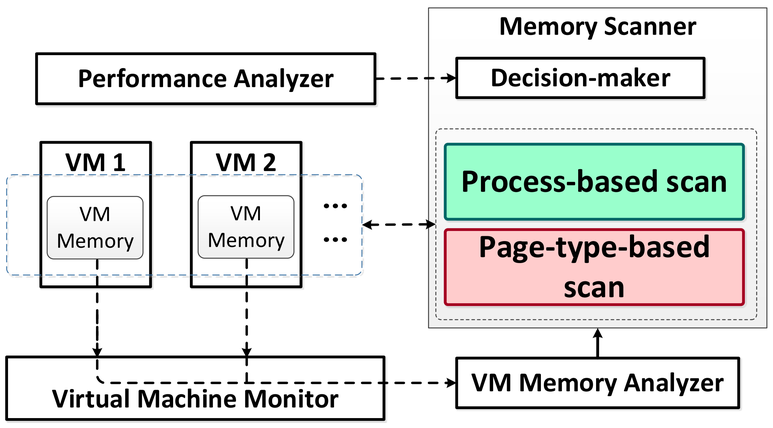
Approach:
- use **VMM-introspection** technique to divide VMs' memory based on sharing opportunities
- **Fine-grained Scanner**: Scan a specified page division in a scan pass to speed up the scanner and improve the efficiency
- **Scan strategy** based on workloads

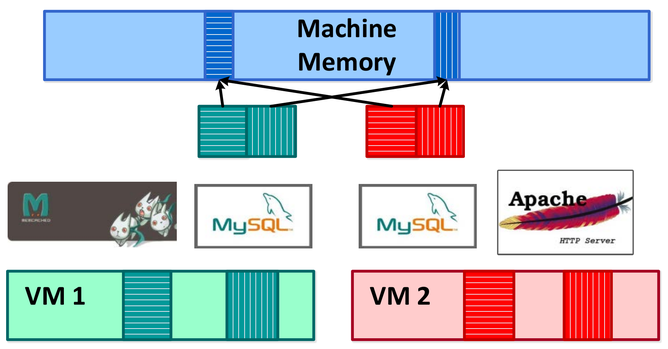


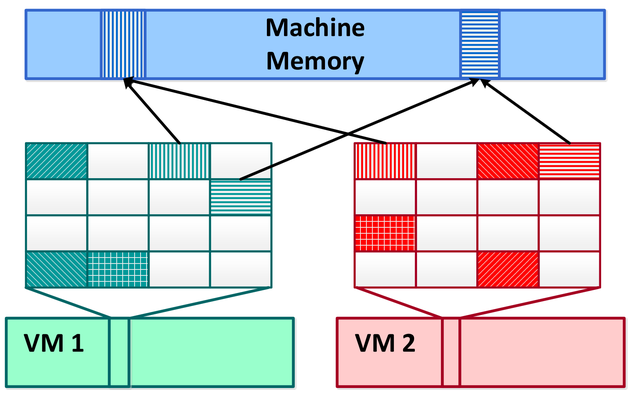
Two aspects of VMs' memory:
- Process pages: pages belong to a process
- Guest physical pages

Fine Granularity of Memory Deduplication

1. System Architecture:
- **Memory Scan Module:** Conduct memory scan procedure.
- **VMs Memory Analyzer:** Provide interfaces to analyze VMs memory
- **Scan Performance Collector:** Collect runtime data for Scan Decision Maker to decide what to do next





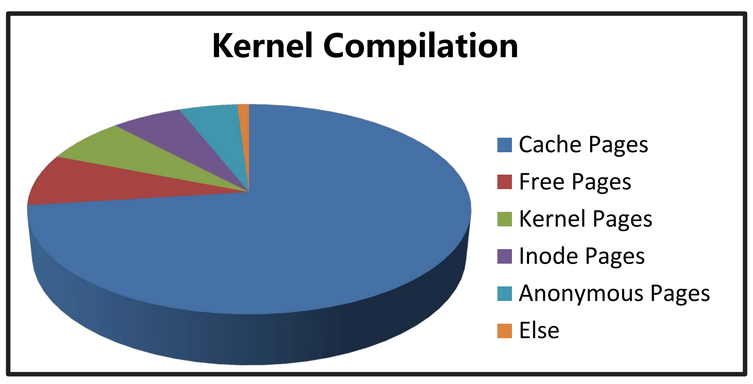


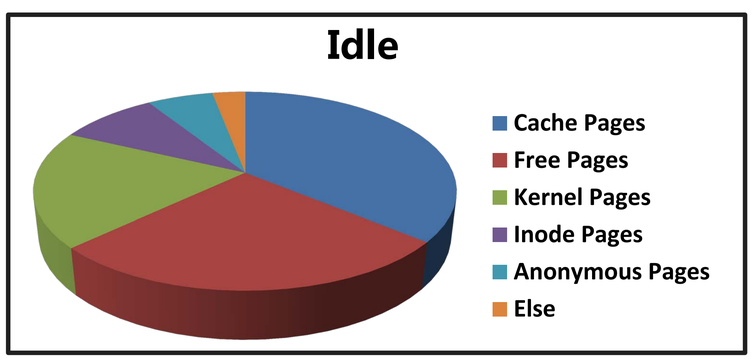
2. Process-based Scan:
- Process identification and analysis
  - CR3, task_struct, mm_struct, vm_area_struct
- Classify similar processes to a process group
- Scan and merge pages of the same process group

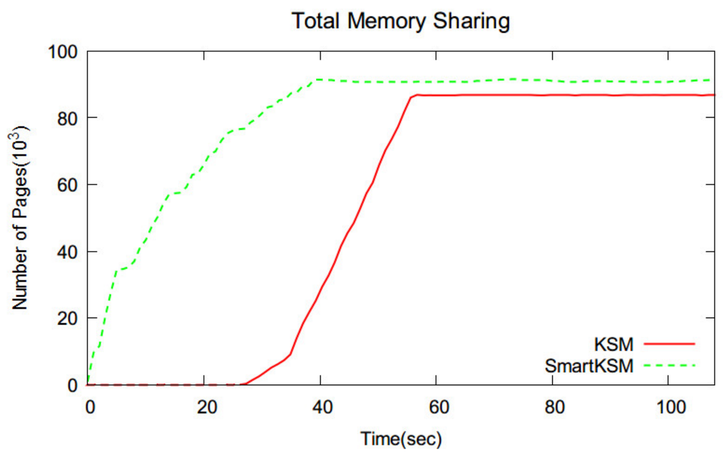
3. Page-type based Scan:
- pages in guest physical memory is classified into five types according to their usages:
  - **Free pages**
  - **Kernel pages**
  - **Cache pages**
  - **Anonymous pages**
  - **Inode pages**
- Scan specific page type(s) in each scan round based on various strategies

Evaluation

Page type proportion in Sharing pages

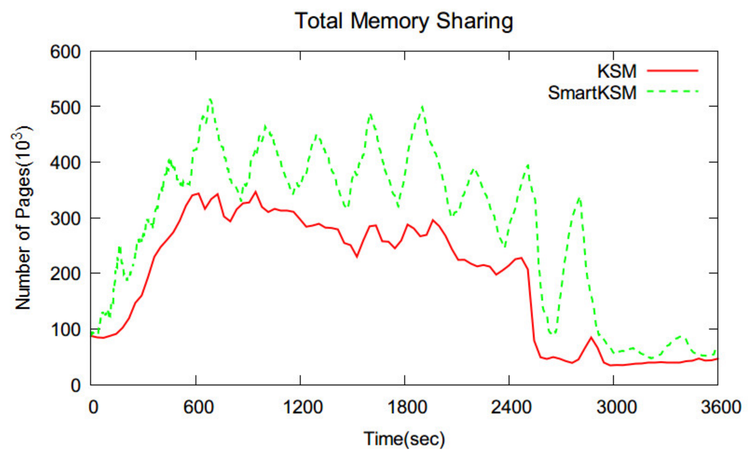






(a) The Idle Workload

In idle workload, smartKSM can merge pages faster than KSM.



(b) The Kernel Compilation Workload

In kernel compilation workload, SmartKSM shares 49.4% more than KSM. This shows that SmartKSM can detect short-lived sharing opportunities better than KSM.