How to use the Kernel Samepage Merging feature

KSM is a memory-saving de-duplication feature, enabled by CONFIG_KSM=y, added to the Linux kernel in 2.6.32. See mm/ksm.c for its implementation, and http://lwn.net/Articles/306704/ and http://lwn.net/Articles/330589/

The KSM daemon ksmd periodically scans those areas of user memory which have been registered with it, looking for pages of identical content which can be replaced by a single write-protected page (which is automatically copied if a process later wants to update its content).

KSM was originally developed for use with KVM (where it was known as Kernel Shared Memory), to fit more virtual machines into physical memory, by sharing the data common between them. But it can be useful to any application which generates many instances of the same data.

KSM only merges anonymous (private) pages, never pagecache (file) pages. KSM's merged pages were originally locked into kernel memory, but can now be swapped out just like other user pages (but sharing is broken when they are swapped back in: ksmd must rediscover their identity and merge again).

KSM only operates on those areas of address space which an application has advised to be likely candidates for merging, by using the madvise(2) system call: int madvise(addr, length, MADV_MERGEABLE).

The app may call int madvise(addr, length, MADV_UNMERGEABLE) to cancel that advice and restore unshared pages: whereupon KSM unmerges whatever it merged in that range. Note: this unmerging call may suddenly require more memory than is available - possibly failing with EAGAIN, but more probably arousing the Out-Of-Memory killer.

If KSM is not configured into the running kernel, madvise MADV_MERGEABLE and MADV_UNMERGEABLE simply fail with EINVAL. If the running kernel was built with CONFIG_KSM=y, those calls will normally succeed: even if the the KSM daemon is not currently running, MADV_MERGEABLE still registers the range for whenever the KSM daemon is started; even if the range cannot contain any pages which KSM could actually merge; even if MADV_UNMERGEABLE is applied to a range which was never MADV_MERGEABLE.

Like other madvise calls, they are intended for use on mapped areas of the user address space: they will report ENOMEM if the specified range includes unmapped gaps (though working on the intervening mapped areas), and might fail with EAGAIN if not enough memory for internal structures.

Applications should be considerate in their use of MADV_MERGEABLE, restricting its use to areas likely to benefit. KSM's scans may use a lot of processing power: some installations will disable KSM for that reason.

The KSM daemon is controlled by sysfs files in /sys/kernel/mm/ksm/, readable by all but writable only by root:

pages_to_scan - how many present pages to scan before ksmd goes to sleep
e.g. "echo 100 > /sys/kernel/mm/ksm/pages_to_scan"
Default: 100 (chosen for demonstration purposes)

```
sleep_millisecs  - how many milliseconds ksmd should sleep before next scan
                   e.g. "echo 20 > /sys/kernel/mm/ksm/sleep_millisecs"
                   Default: 20 (chosen for demonstration purposes)

run              - set 0 to stop ksmd from running but keep merged pages,
                   set 1 to run ksmd e.g. "echo 1 > /sys/kernel/mm/ksm/run",
                   set 2 to stop ksmd and unmerge all pages currently merged,
                         but leave mergeable areas registered for next run
                   Default: 0 (must be changed to 1 to activate KSM,
                               except if CONFIG_SYSFS is disabled)
```

The effectiveness of KSM and MADV_MERGEABLE is shown in /sys/kernel/mm/ksm/:

```
pages_shared     - how many shared pages are being used
pages_sharing    - how many more sites are sharing them i.e. how much saved
pages_unshared   - how many pages unique but repeatedly checked for merging
pages_volatile   - how many pages changing too fast to be placed in a tree
full_scans       - how many times all mergeable areas have been scanned
```

A high ratio of pages_sharing to pages_shared indicates good sharing, but
a high ratio of pages_unshared to pages_sharing indicates wasted effort.
pages_volatile embraces several different kinds of activity, but a high
proportion there would also indicate poor use of madvise MADV_MERGEABLE.

Izik Eidus,
Hugh Dickins, 17 Nov 2009