Soft-Dirty PTEs

The soft-dirty is a bit on a PTE which helps to track which pages a task writes to. In order to do this tracking one should

1. Clear soft-dirty bits from the task's PTEs.

   This is done by writing "4" into the `/proc/PID/clear_refs` file of the task in question.

2. Wait some time.
3. Read soft-dirty bits from the PTEs.

   This is done by reading from the `/proc/PID/pagemap`. The bit 55 of the 64-bit qword is the soft-dirty one. If set, the respective PTE was written to since step 1.

Internally, to do this tracking, the writable bit is cleared from PTEs when the soft-dirty bit is cleared. So, after this, when the task tries to modify a page at some virtual address the #PF occurs and the kernel sets the soft-dirty bit on the respective PTE.

Note, that although all the task's address space is marked as r/o after the soft-dirty bits clear, the #PF-s that occur after that are processed fast. This is so, since the pages are still mapped to physical memory, and thus all the kernel does is finds this fact out and puts both writable and soft-dirty bits on the PTE.

While in most cases tracking memory changes by #PF-s is more than enough there is still a scenario when we can lose soft dirty bits -- a task unmaps a previously mapped memory region and then maps a new one at exactly the same place. When unmap is called, the kernel internally clears PTE values including soft dirty bits. To notify user space application about such memory region renewal the kernel always marks new memory regions (and expanded regions) as soft dirty.

This feature is actively used by the checkpoint-restore project. You can find more details about it on http://criu.org

-- Pavel Emelyanov, Apr 9, 2013