SHARING is CARING

CS 695: Virtualization and Cloud Computing

A Course Project Report

By

Prathamesh Navale
Roll No.: 23M0746

and

Varn Gupta
Roll No.: 23M0749

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

May, 2024

Contents

List of Figures

Chapter 1

Introduction

The performance and scalability of modern computing systems depend heavily on the effective management of memory resources. By merging identical memory pages, the Linux kernel's Kernel Samepage Merging (KSM) memory management mechanism seeks to minimize memory duplication. A study of KSM's performance characterization is presented in this report, emphasizing how the system behaves under various workload and configuration scenarios.

1.1 Problem Statement

The primary goal is to conduct an **empirical assessment of KSM's performance** and identify its behaviour in various situations. To be more precise, we want to answer the following study questions:

- How does KSM's performance change depending on the workload?
- What effect do various configuration options, like pages_to_scan, have on the functionality of KSM?

Chapter 2

Experiment Design and Implementation

2.1 Experiment Design

- **Workloads:** The stress command generates synthetic workloads, simulating various system stress scenarios. These include CPU stress and memory stress, allowing for a comprehensive evaluation of KSM's behavior under different workload profiles.
- **Setups:** Virtual machines (VMs) are created using VirtualBox Manager, with each VM configured to run an instance of Ubuntu Linux.
  The Host specifications are:
  - **CPU cores:** 12
  - **RAM:** 16GB
  - **OS:** Ubuntu 22.04

  The VM specifications are:
  - **CPU cores:** 2
  - **RAM:** 4GB
  - **OS:** Ubuntu 22.04
- **Performance parameters:** To assess KSM's effectiveness in varying scenarios, we track the number of performance parameters, such as pages_shared, pages_unshared, CPU utilization, etc.
- **Configuration factors:** To determine how different configuration factors affect KSM's performance, we experiment with things like scan rate, memory threshold, and KSM activation threshold.

2.2 Implementation

The experimental setup is implemented using **stress** command to generate workload, Virtual Box Manager for VM management and a Python script to monitor /sys/kernel/mm/ksm parameters. We carefully configured and varied the parameters to ensure accurate and reproducible results.

Chapter 3

Results and Analysis

To assess the behavior of Kernel Samepage Merging (KSM) with and without workload, we conducted experiments with virtual machines (VMs) running on VirtualBox Manager.

3.1 VMs with No Workload

In this section, we analyze the behavior of KSM without any workload.
Here, we first run a single VM and wait till the value becomes the constant. Then we start second VM and wait for the saturation point. During this time, we monitor **/sys/kernel/mm/ksm/pages_shared** and **/sys/kernel/mm/ksm/general_profit** using a Python script.
The experiments were conducted with different sleep_milisecs values (10 and 20) to assess their impact on KSM's behavior.

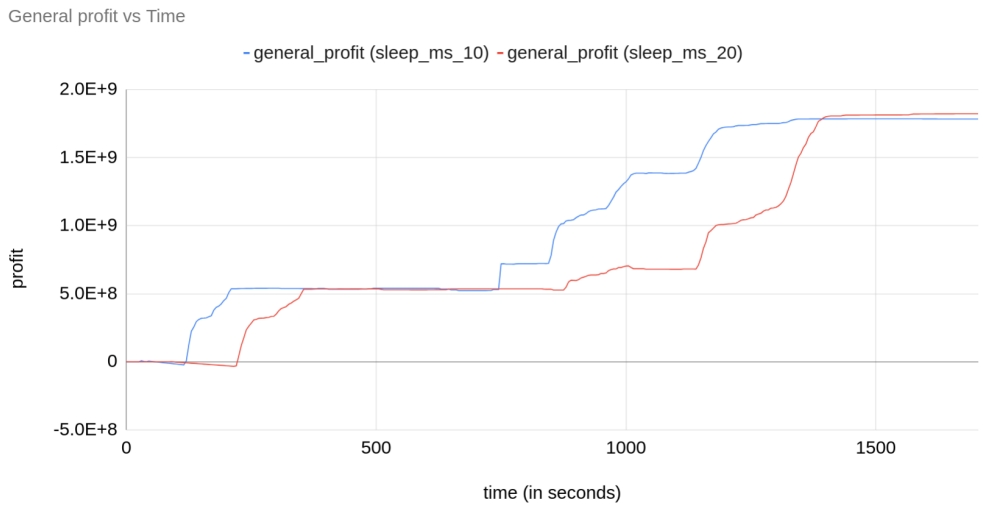

Figure 3.1: general_profit vs time

Initially, there is a decrease in the general_profit when the first virtual machine starts.

The plot slightly dips when the second virtual machine (VM) launches, for the same reason.

This is due to the fact that the new virtual machine (VM) adds memory pages to the system at startup, some of which can be duplicates of existing memory pages. KSM allocates resources to duplicate memory page identification and merging. The benefits of memory deduplication should be gradually realised when the general_profit metric rises as the first overhead is removed and KSM stabilises its operation.

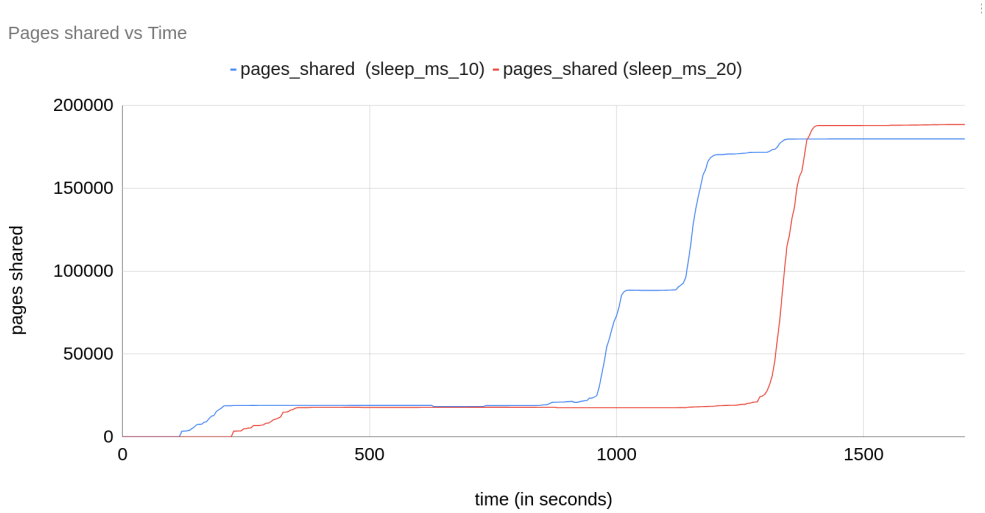

Figure 3.2: pages_shared vs time

KSM executes memory scanning and merging operations more frequently when the sleep_milisecs option is set to 10 than when it is set to 20. As a result, KSM can identify and combine redundant memory pages with shorter scan intervals.

The saturation point is the moment, given the current workload and system parameters, at which KSM has merged as many duplicate memory pages as it can. Duplicate pages can be promptly identified and merged by KSM when the sleep_milisecs option is set to 10. This is because KSM checks the memory more regularly. Because it has had more opportunity to find and integrate duplicate pages, it reaches saturation point earlier.

3.2 VMs with *stress* command running

In this section, we analyze the behaviour of KSM with VMs under stressed.
Here, we run both the VMs together and start *stress* command in both the VMs.
SYNTAX: `stress --vm <number_of_worker_threads> --vm-bytes <bytes_per_worker>`

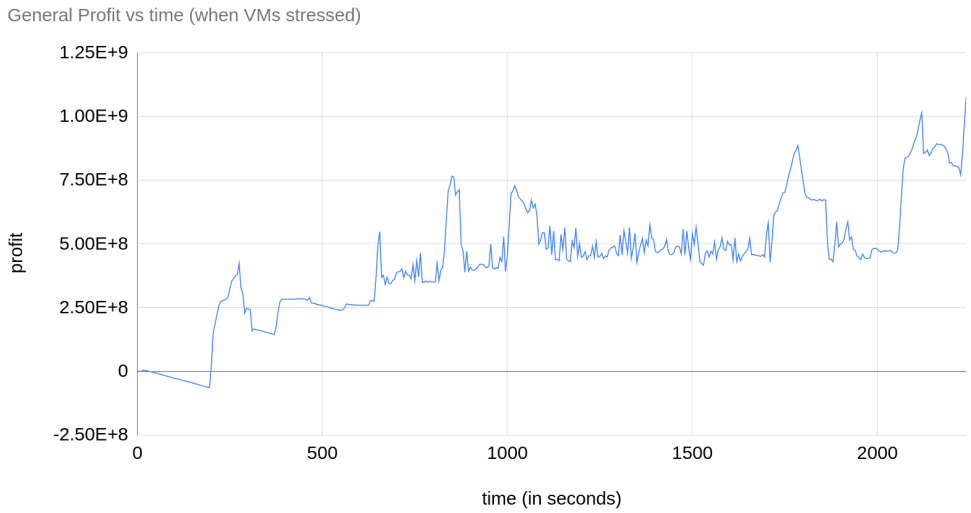

Figure 3.3: general_profit vs time

Plot fluctuations occur when variations in system activity and workload impact memory usage and KSM's effectiveness in identifying and merging duplicate pages, which in turn causes variations in overall profit. KSM's efficiency may also be impacted by how frequently it searches memory for duplicates. Variations in scanning intervals could result in variations in overall profit.

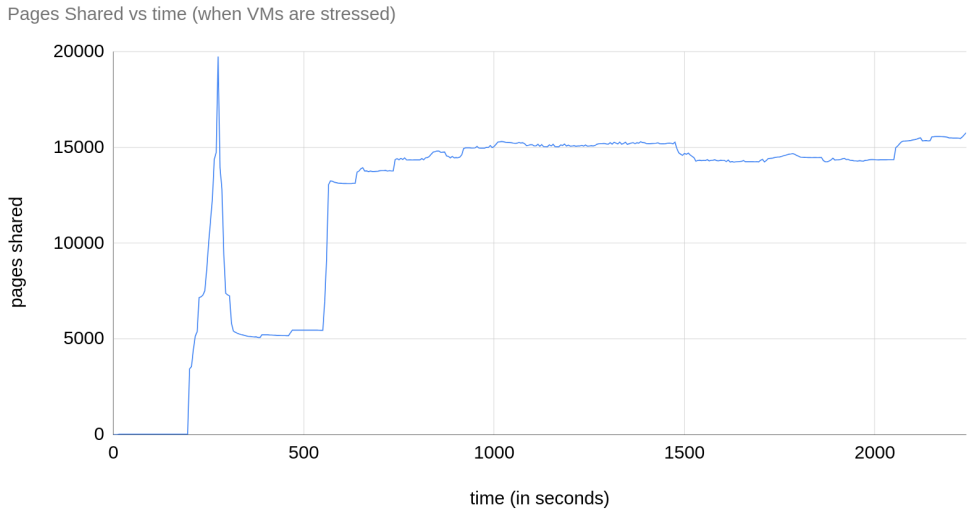

Figure 3.4: pages_shared vs time

In the plot, we see a spike as the KSM gets to its work and duplicate memory pages are identified and merged. But when we start *stress* command in both the VMs, we see a significant fall in the shared pages as there's an increase in memory demand and altered memory usage patterns are induced by the stress workload.

The stress command increases system activity and memory utilisation when it is applied to both virtual machines. There can be fewer identical memory pages available for Kernel Samepage Merging (KSM) to detect and merge as a result of this increasing memory consumption. This causes a brief drop in the rate of merging identical memory pages, which lowers the pages_shared plot.

3.3 Without KSM enabled

Here, we first ran our script without KSM enabled. When VMs were booted, free memory left is noted down. After some time, when the value became consistent, we enabled KSM and saw a significant gain in free memory gradually as KSM started identifying and merging duplicate memory pages. We can see a decrease in free memory as we started *stress* command in the VMs. Thus, a decrease in free memory.

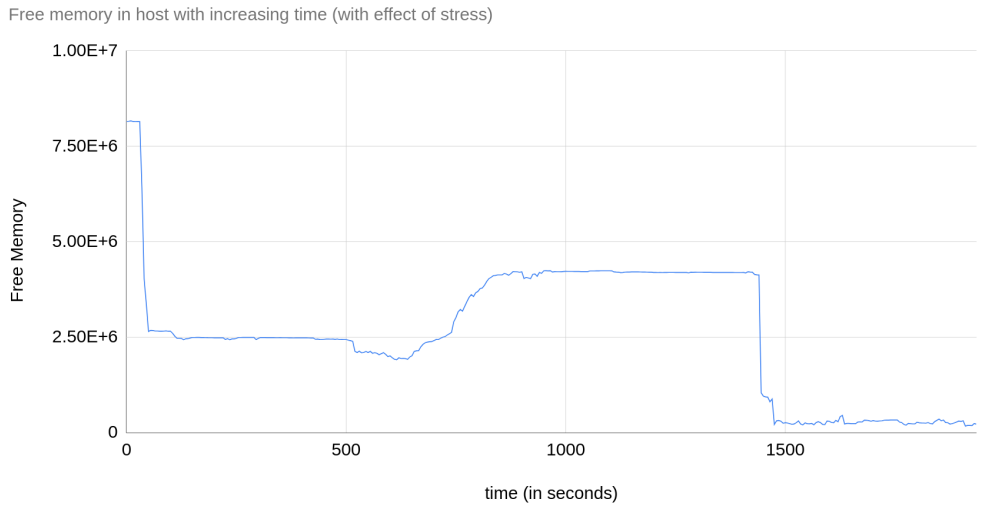

Figure 3.5: With KSM disabled

Chapter 4

Conclusion

Based on a performance characterization research, it has been demonstrated that Kernel Samepage Merging (KSM) effectively reduces memory duplication and maximises memory utilisation in virtualized systems. Workload fluctuation and dynamic memory utilisation do not affect system performance since KSM removes redundant memory pages shared by virtual machines.

The reduction in pages_shared during stress loading on both virtual machines demonstrates how increased memory demands affect KSM's efficiency. This emphasises how crucial it is to understand workload factors in order to properly manage memory.

All things considered, KSM is essential for increasing the effectiveness of memory management and the scalability of the system in virtualized environments. It also offers helpful information on how to best optimise memory use and boost system performance. In order to handle changing needs, more study could examine sophisticated optimisation strategies and interaction with other memory management systems.