Empirical analysis of dirty tracking overhead: COW vs Dirty Bit

drunKbiTs:

Nikhil Molugu

20111415

Tabish Ahmad

21111060

Niraj Kundu

Introduction

Finding amount of memory change is a task crucial for applications like CRIU. The project aims to analyze the overheads due to dirty tracking of two well-known techniques - COW technique and PTE dirty bit technique.

Dirty Bit Method

- We check for dirty bit in all the valid PTEs of the process.

- Steps:
    - Clear dirty bits,
    - Wait some time,
    - scan all valid PTE's dirty bits.

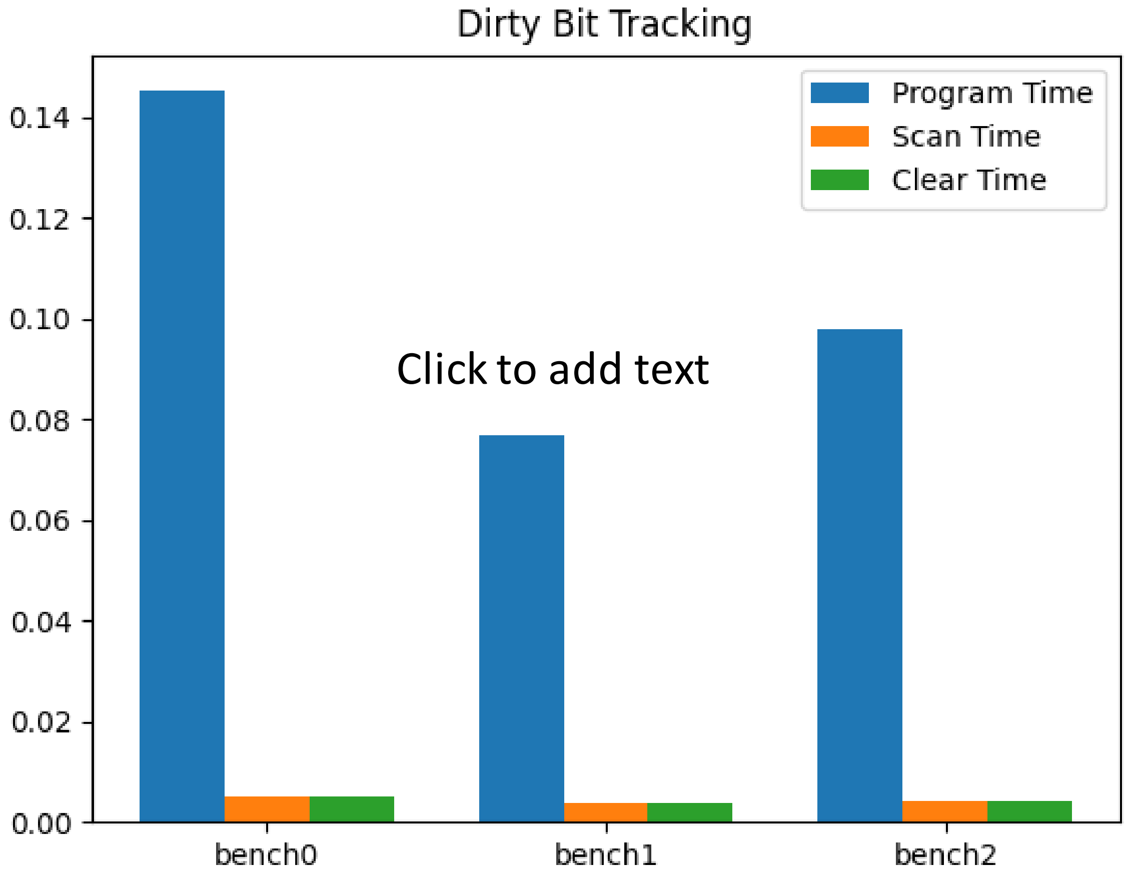
Dirty Bit - Implementation

- Kernel Module – Clears dirty bits of PTEs. Checks the process' PTEs which have their dirty bit as set.
- Creates a char device "pgdv".
- Dirty bits can be cleaned or tracked by ioctl system call of our driver.

Results

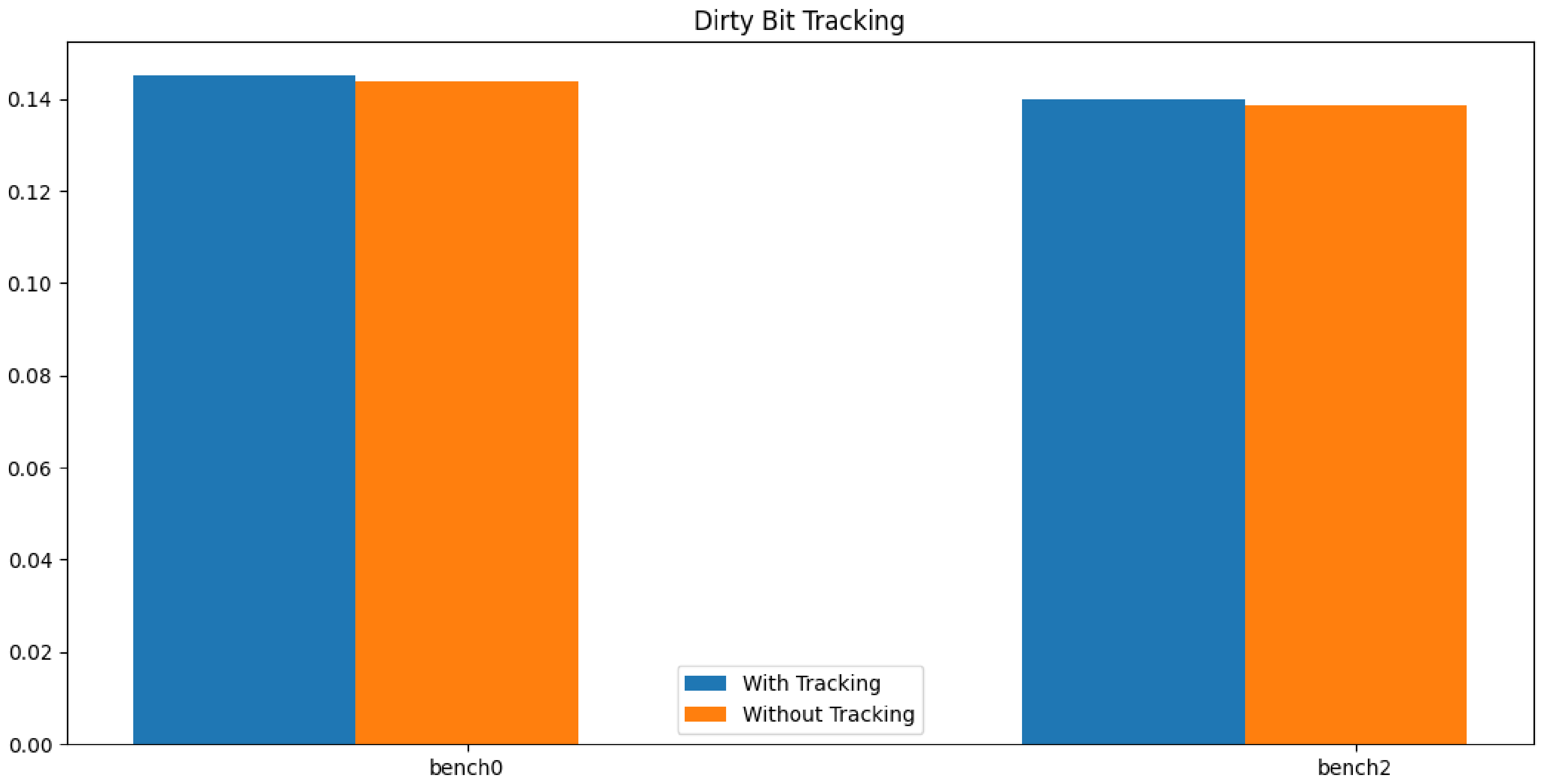
- Benchmark Program:
  - Allocate some memory,
  - clear dirty bits
  - make changes to the memory,
  - scan dirty bits

Results



Click to add text

Results


Dirty Bit Tracking
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
With Tracking
Without Tracking
bench0
bench2

Conclusion

- We were able to implement dirty bit tracking method.
- We show the overheads introduced by dirty bit clearing and checking.

Thank You