Flash Disk Emulator

Eric Aderhold and Blayne Field

May 13, 2008

Outline

- Motivation
- Methods
- Results

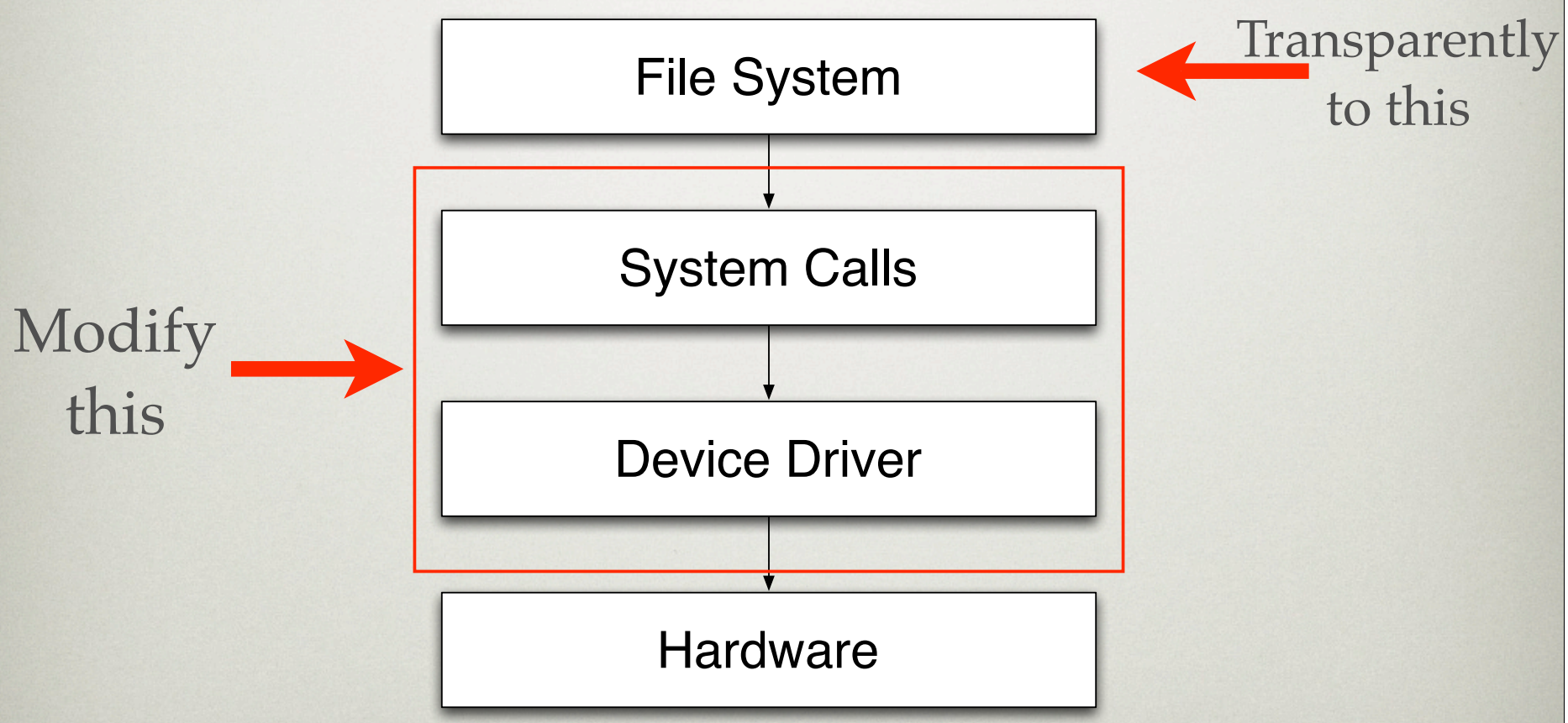
Motivation

- Flash drives are important
  - Becoming more common in the field
  - Various related research is underway
- Flash hardware can be expensive
- Flash hardware is prone to failure after heavy use

Overview

- Modified Linux RAM disk driver
  - Added timing delays to approximate flash hardware
- Emulator's performance is very similar to flash hardware

Methodology



Emulating Writes

- Functions called:
  - `ramdisk_prepare_write`
  - `ramdisk_commit_write`
  - `ramdisk_set_page_dirty`
- Delays added to `ramdisk_prepare_write`

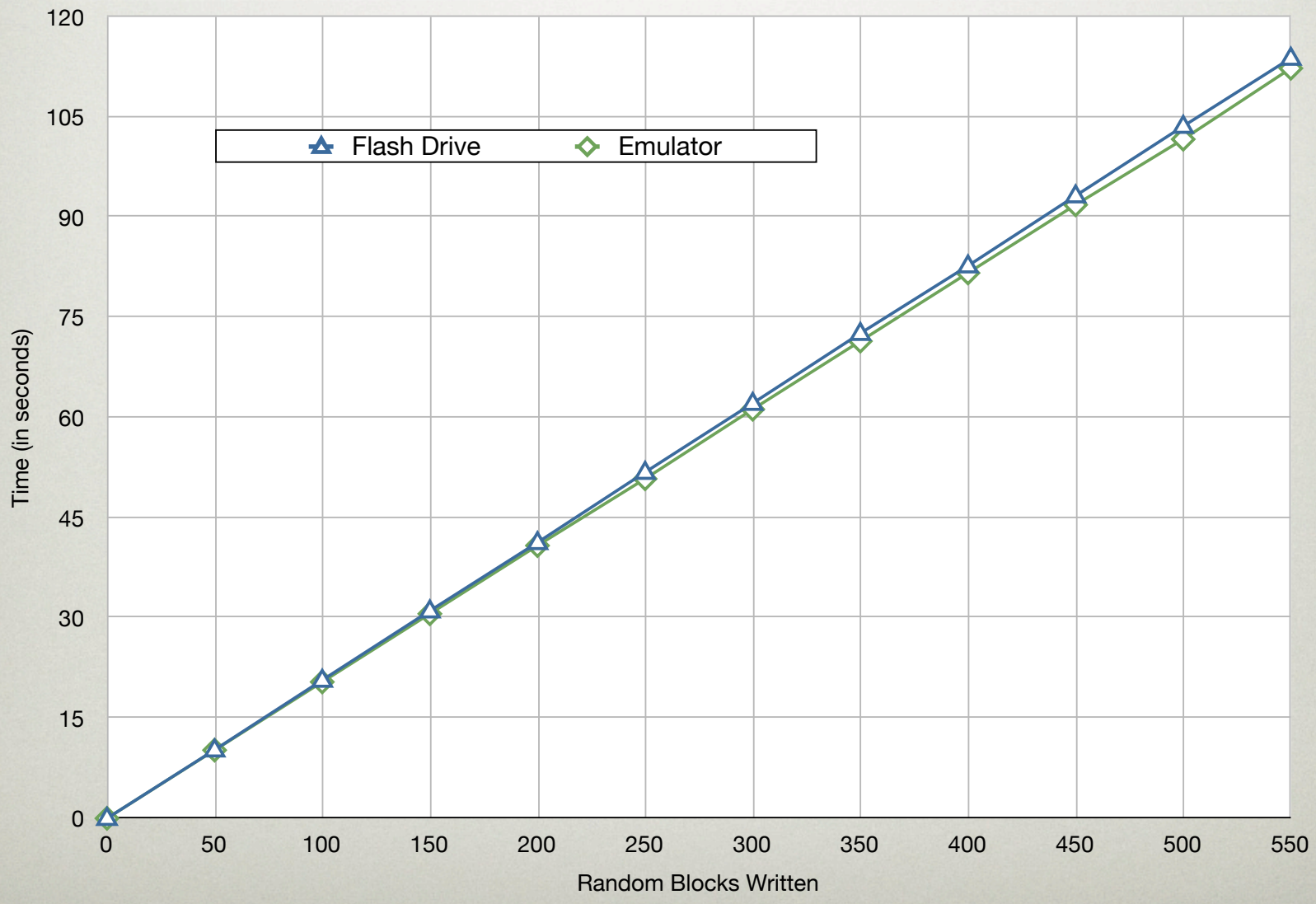
Emulating Writes

- Problem: Random vs. sequential writes
  - On a real flash disk, these take different amounts of time
  - Solution: keep track of previous request and set delay according to pattern

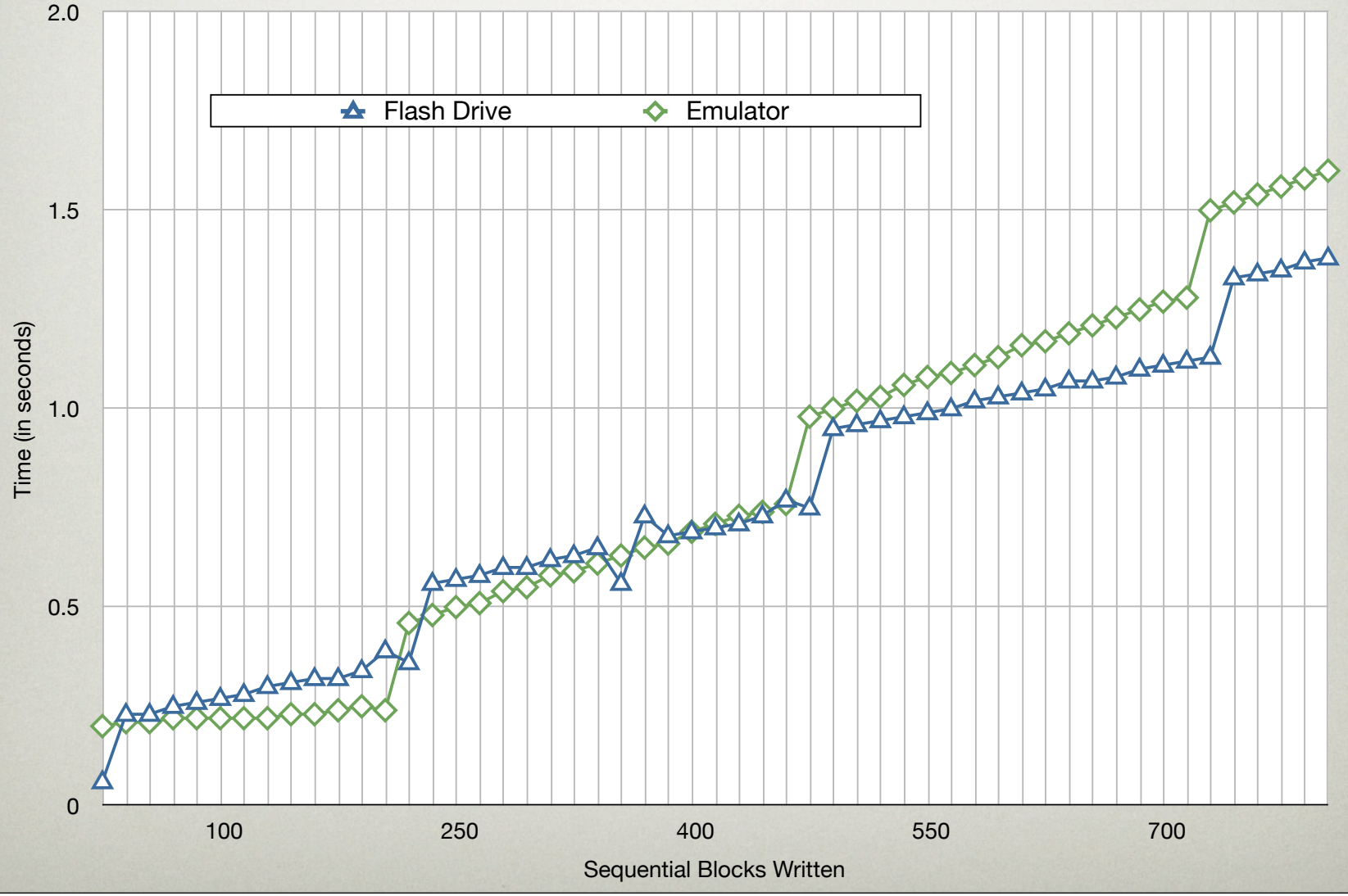
Emulating Writes

- Another problem:
  - First write always categorized as random
  - Causes a bit of overcompensation if writing sequential data
- Solution:
  - Decrease first few sequential write delays to “make up” for first write

Write Performance (Random)



Write Performance (Sequential)



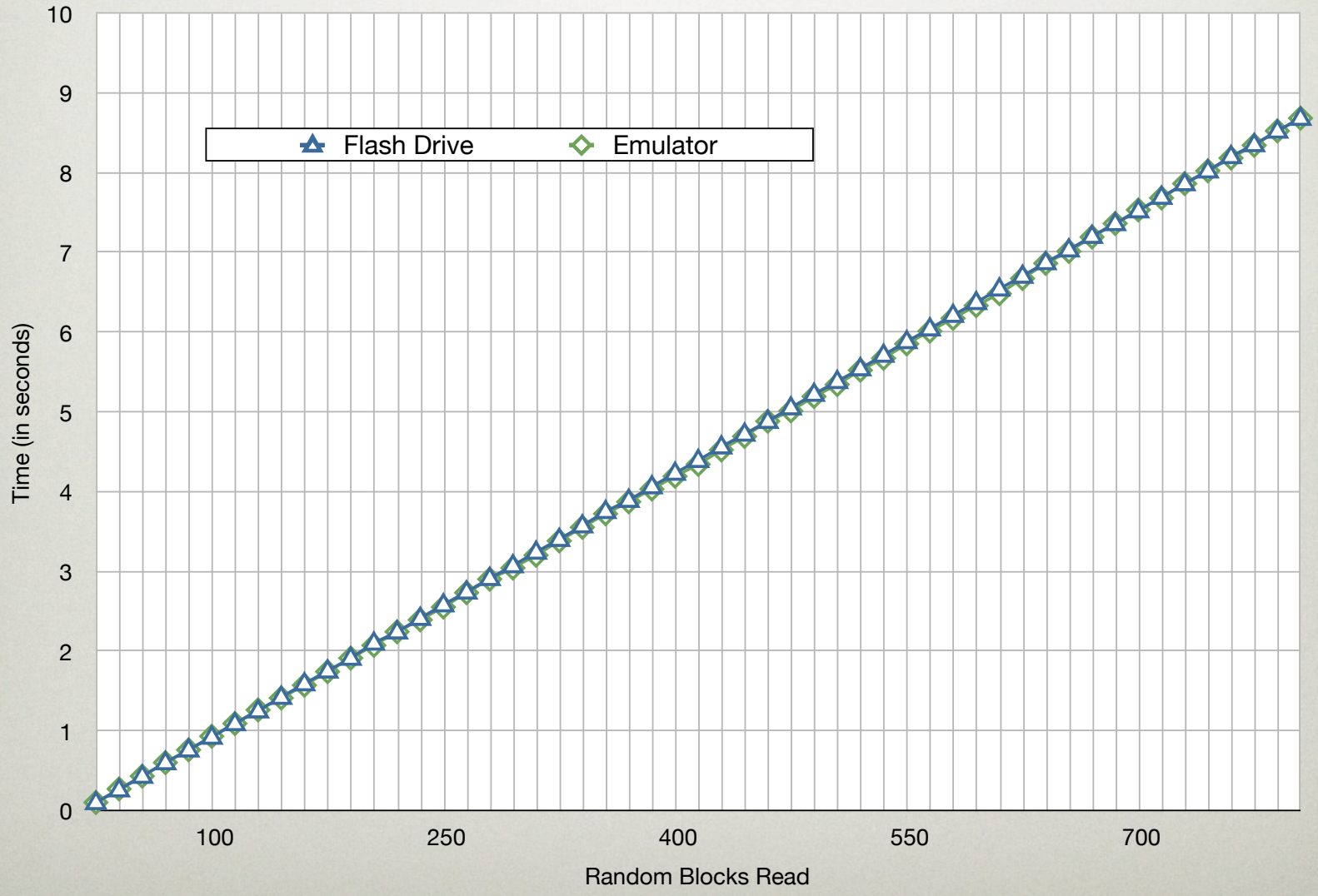
Emulating Reads

- Problem: reads to a RAM disk do not call any of the RAM disk code
  - Solution: add delays at the VFS layer instead of the RAM disk layer
    - Only apply them to the RAM disk/ Flash emulation device

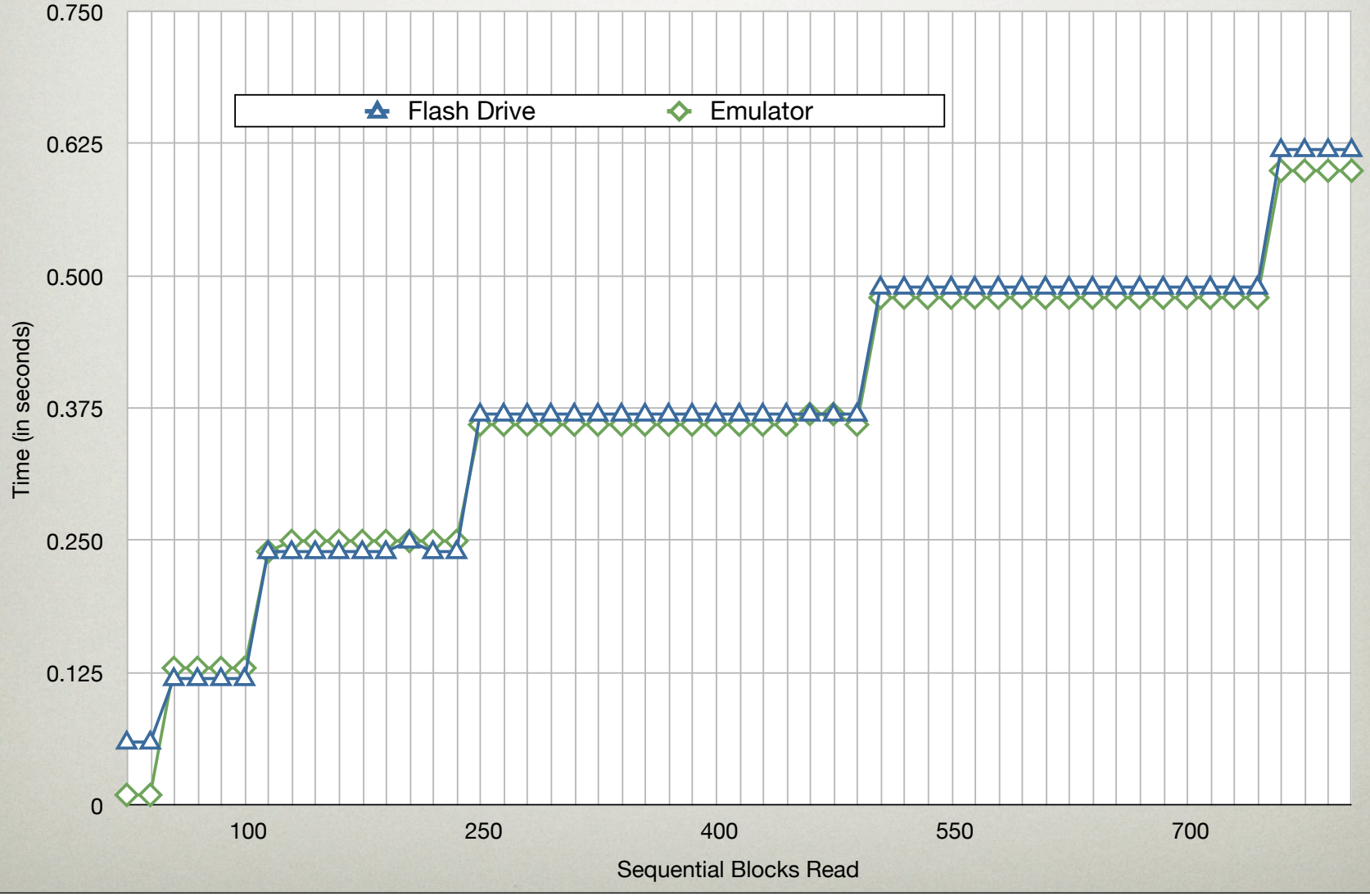
Emulating Reads

- Random vs. sequential reads
  - Very similar approach
- "Jumps" in sequential performance
  - Very similar approach

Read Performance (Random)



Read Performance (Sequential)



Next Steps

- Features that would make the emulator more useful:
  - Ability to create a separate disk from existing RAM disks
  - Ability to emulate more various flash hardware devices

Questions?