FastTrack: Foreground App-Aware I/O Management for Improving User Experience of Android Smartphones

Sangwook Shane Hahn*, Sungjin Lee†, Inhyuk Yee#, Donguk Ryu‡ and Jihong Kim*

*Seoul National University
†DGIST
#AlBrain Asia
‡Samsung Electronics

USENIX Annual Technical Conference, 2018
User Experience in Android Smartphones

App Launch → App Runtime → App Switch

App response time is crucial to the user experience
Background Apps Degrade the Quality of User Experience

App launch time increases due to **background apps**

**Background apps increase foreground app launch time by up to **2.6 times****
Background Apps Degrade the Quality of User Experience

App switch time increases due to background apps

App switch time

Nexus 5
Nexus 6
Galaxy S6
Pixel

Background apps increase foreground app switch time by up to 19.5 times
Main Cause of Performance Degradation

How background apps degrade the quality of user experience?

No rendering for BG apps

Background (storage) I/Os

Performance ↓ even in offline

CPU?

Network?

Storage?

GPU?
Background I/O Occurrence Frequency

Q: Are background I/Os (BG I/Os) occur frequently?

Average days between releases
- 3 weeks
- 2 weeks
- 1 week

Popular apps are updated in 1 to 2 weeks

App Update

Frequent BG I/O occurrences have a critical impact on the user experience
• Impact of Background I/O on User Experience

• Foreground-Background Interference Analysis

• FastTrack: Foreground App-Aware I/O Management

• Experimental Results

• Conclusions
Page cache and storage device are affected most by FG-BG interference.
Lock contentions in the page cache can significantly degrade the user experience by waiting the completion of BG I/Os frequently.
Impact on Page Cache: High Miss Rate

Existing page cache replacement policy can significantly degrade the user experience by evicting performance-critical hot FG pages.
Impact on Storage Device: Internal Priority Inversion

NAND Flash-based Storage Device

FIFO

Read Request

①

Write Request

②

FIFO

NAND Flash Memory

Switch (write-dominant)

Storage Throughput (MB/sec)

Storage Throughput

Switch (write-dominant)

BG Reads

FIFO

BG I/O

Existing device I/O scheduler’s priority policy can significantly degrade the user experience by limiting the throughput of foreground I/Os
Solution for Foreground-Background Interference

**FastTrack** is effective in improving Android smartphone user experience.
Outline

• Impact of Background I/O on User Experience
• Foreground-Background Interference Analysis
  • FastTrack: Foreground App-Aware I/O Management
  • Experimental Results
  • Conclusions
Overview of FastTrack

- **Improves foreground I/O performance** &
- **Improves the quality of smartphone user experience**

### Components

- **App Status Detector**
- **Page Allocator**
- **Page Reclaimer**
- **I/O Dispatcher**
- **Device I/O Scheduler**
- **Android Platform**
- **Page Cache**
- **Storage Device**

### Process

1. **Detect FG I/Os using UID**
2. **Preempt I/O Operations for BG I/Os**
3. **Prevent FG Data from Being Flushed**

- **Detect FG I/Os using Tagging**
- **Preempt BG I/Os’ Dispatching**
- **Deliver FG I/Os to NAND flash chip instantly**

**FG app’s UID**

**Deliver FG I/O Instantly**
App Status Detector & Page Reclaimer

Android Platform

App Status Detector

Launch

Linux Kernel

Page Reclaimer

FG App UID = 10003

Trigger

FG App (UID: 10003)
sysfs (10003)

Activity Stack

Running Activity

Activity A

Last Running Activity

Activity B

Activity C

Page Cache

Clean Pages

FG data (UID 10003)  FG data (UID 10003)
BG data (UID 10001)  BG data (UID 10001)

Dirty Pages

FG data (UID 10003)  FG data (UID 10003)
BG data (UID 10001)  BG data (UID 10001)

① Detect

② Prevent FG Data

Activity Stack

Running Activity

Activity C

Last Running Activity

Activity B

Activity A

Launch
Page Allocator

App Update

Write File B (UID 10001)

Read File A (UID 10003)

Linux Kernel

FG App UID = 10003
FG I/O Flag = true

Page Allocator

① Detect

Get_free_pages (4 pages)

② Preempt

Checkpoint

Get_free_pages (4 pages)

③ Resume
Experimental Settings for Android Smartphones

**Android Storage I/O Stack**

- **Application** → **Android Platform** → **File System** → **Page Cache** → **Block I/O Layer** → **Storage Device**

**FastTrack**

- **App Status Detector**
- **Page Allocator**
- **Page Reclaimer**
- **I/O Dispatcher**
- **Device I/O Scheduler**

**FG App Usage Scenario**

1. Launch (Read-dominant)
2. Switch (Write-dominant)

**BG App Usage Scenario**

1. Update (Write-dominant)
2. Upload (Read-dominant)
1. FastTrack can reduce gallery app launch time delay from BG I/Os by up to 87%.
2. FG reads have higher priority in storage device.
Result 2: App Switch Time Comparisons

Internal priority inversion limits the effect of FastTrack (FG write priority < BG read priority in storage device)
Experimental Settings for Emulator

FastTrack

- App Status Detector
- Page Allocator
- Page Reclaimer
- I/O Dispatcher
- Device I/O Scheduler

Android Smartphone

Application Launch/Usage

System Call Trace

Trace Replayer

Emulation at Host-level FTL + Customized SSD

FastTrack

- App Status Detector
- Page Allocator
- Page Reclaimer
- I/O Dispatcher
- Device I/O Scheduler
Result 3: Storage-Level Snapshot

Device I/O scheduler can provide a much higher throughput to FG I/Os even when FG I/Os are write and BG I/Os are read.
Result 4: Effectiveness of FastTrack over Varying BG Apps

FastTrack can provide the equivalent level of responsiveness to an FG app regardless of the number of BG apps.
Conclusion

◆ We have presented a foreground app-aware I/O management (FastTrack) for improving user experience
  • FastTrack preempts BG I/Os in the page cache
  • FastTrack prevents FG I/O’s data from being flushed
  • FastTrack immediately delivers FG I/O to the NAND flash memory with minimum interference from inflight BG I/Os
  • FastTrack reduces the user-perceived response time delay by up to 95%

◆ Future work
  • Multiple foreground app usage environment (split view, multiple windows)
  • FastTrack for desktop/server computing system