

Semantics-Aware Shadow Paging: Handling Transaction Conflict in EXT4 Journaling



Joontaek Oh

Hojin Nam

Won

Kyounggho Koo

Youjip

Dept. of Electrical Engineering, KAIST

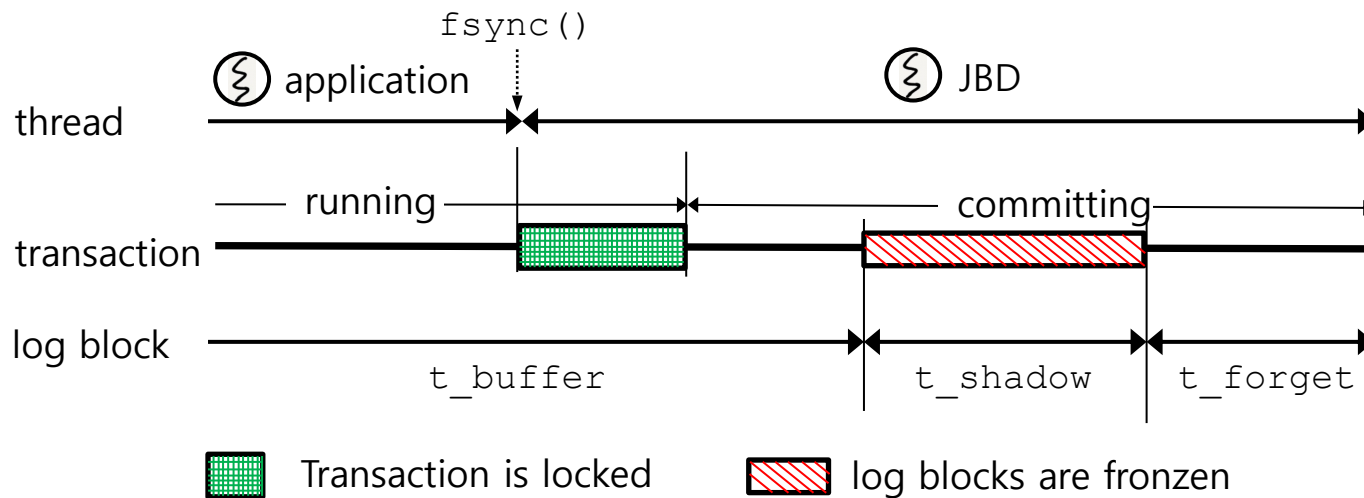
Contents

- Motivation
- Design
- Evaluation

Motivation

Transaction conflict

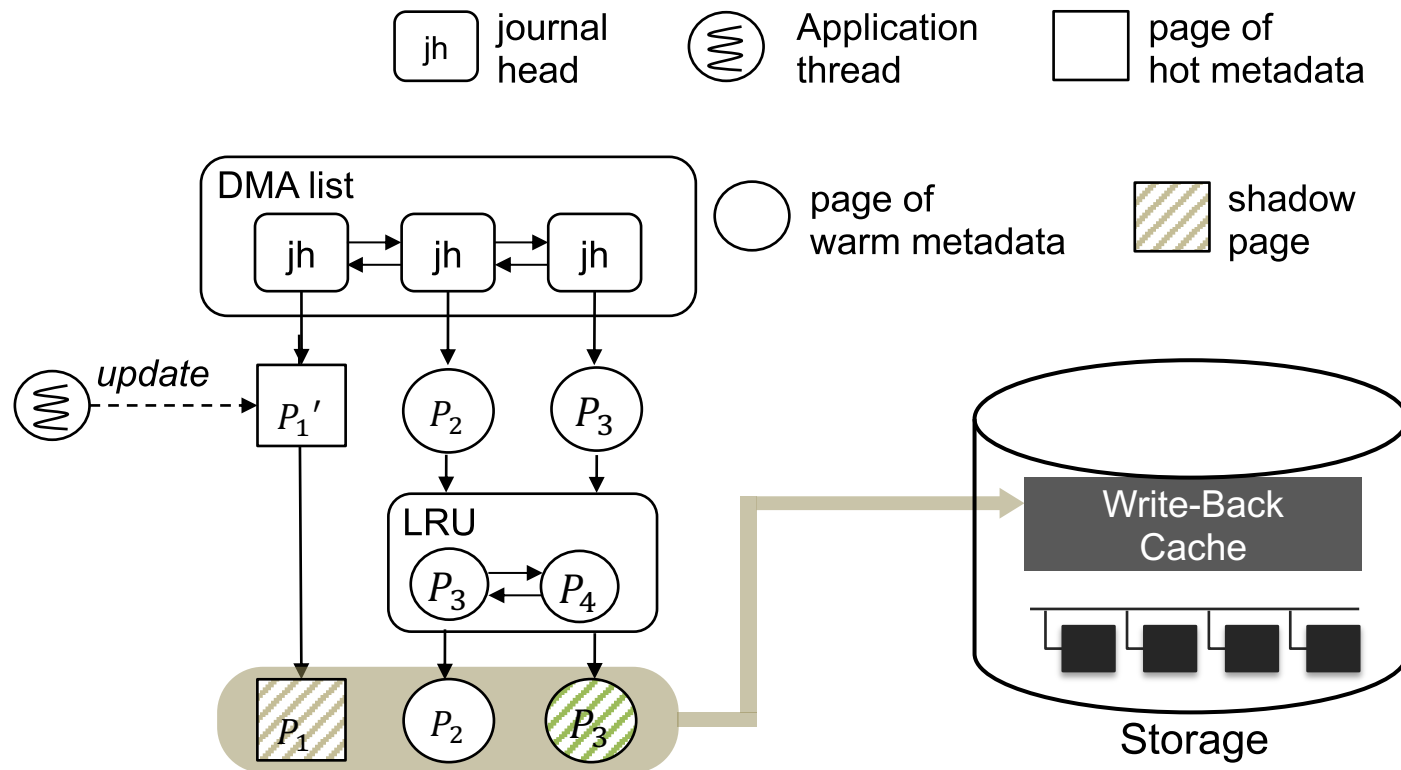
- An application may attempt to update the page that is in the committing transaction. We call it a **transaction conflict**.
- EXT4 adopts **shadow paging** to resolve this phenomenon.
- EXT4 allocates the **shadow page only** when the **conflict** occurs.



Design

Semantics-aware shadow paging (SSP)

- Modified **EXT4** with **SSP** identifies the pages causing conflict frequently and allocates the **shadow pages in advance** regardless of whether they are subject to conflicts or not.



Hot group and Warm group

- We categorize the metadata into two groups: **hot** and **warm**
- **Hot group**
 - Filesystem-wide metadata (i.e., super block, group descriptor table, inode bitmap, data bitmap)
 - Shadow page is always allocated by JBD thread!
- **Warm group**
 - Per-file metadata (i.e., inode table and directory entry)
 - Shadow page is allocated when they are in the LRU list.
 - The LRU list is updated in every completion of the commit.
 - The size of the LRU window is determined based upon MRC.

Evaluation

Server Workload: varmail

- SSP improves both throughput and scalability of the EXT4!!

