FlexiCache: A Flexible Interface for Customizing Linux File System Buffer Cache Replacement Policies

Pavan Konanki

Ali R. Butt

Virginia Tech

Motivation

- Many advanced replacement algorithms are available
  - E.g. ARC [FAST ‘03], PCC [OSDI ‘04], CLOCK-PRO [ATC ’05]
  - Performance depends on application access patterns

  →”Okay for all, best for none” standard approach

- Improving cache performance for all applications is difficult
  - Use application level caching – impractical
  - Tailor replacement policy to the workload – excruciating

- Designing new replacement algorithms is challenging
  - Should be studied in the kernel’s context [SIGMETRICS ’05]

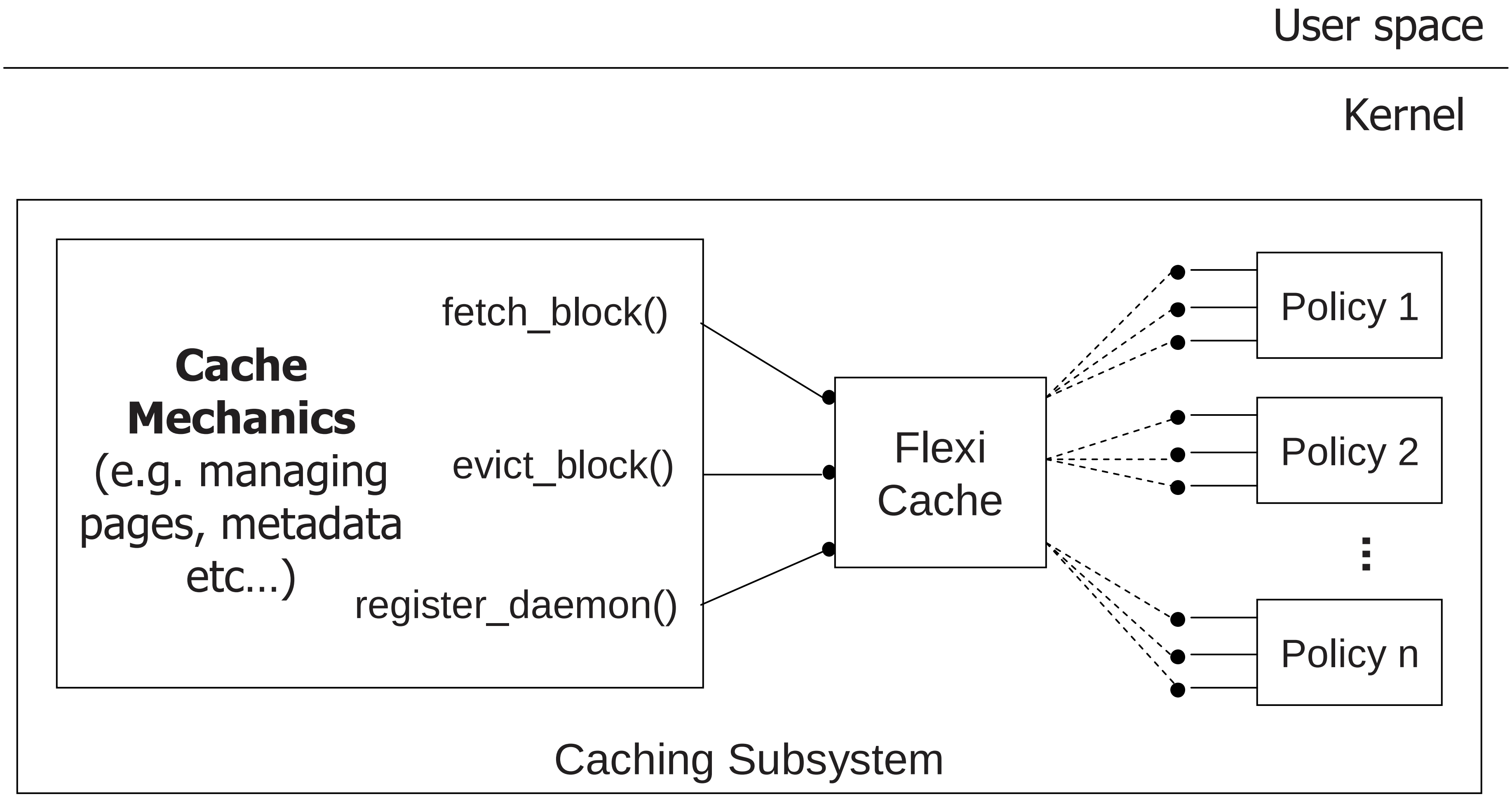
What can we do?

- Allow easy incorporation of new replacement policies
  - Design a flexible, easy to use interface

- Allow multiple replacement policies to coexist in the kernel
  - Support dynamic switching between algorithms

Our contribution: FlexiCache

A flexible interface for customizing file system buffer caching policies



User space

Kernel

Cache Mechanics
(e.g. managing pages, metadata etc…)

fetch_block()

evict_block()

register_daemon()

Flexi Cache

Policy 1

Policy 2

⋮

Policy n

Caching Subsystem

Design challenges

- Making the interface general
- Accommodating disparate replacement algorithms
- Exposing right amount of information
- Hiding the cache mechanics completely

Design stages

- Analyze kernel caching subsystem code
- Design and implement the FlexiCache interface
- Evaluate impact on performance