Introduce Caching Technologies using Xrootd

Wei Yang

Outline

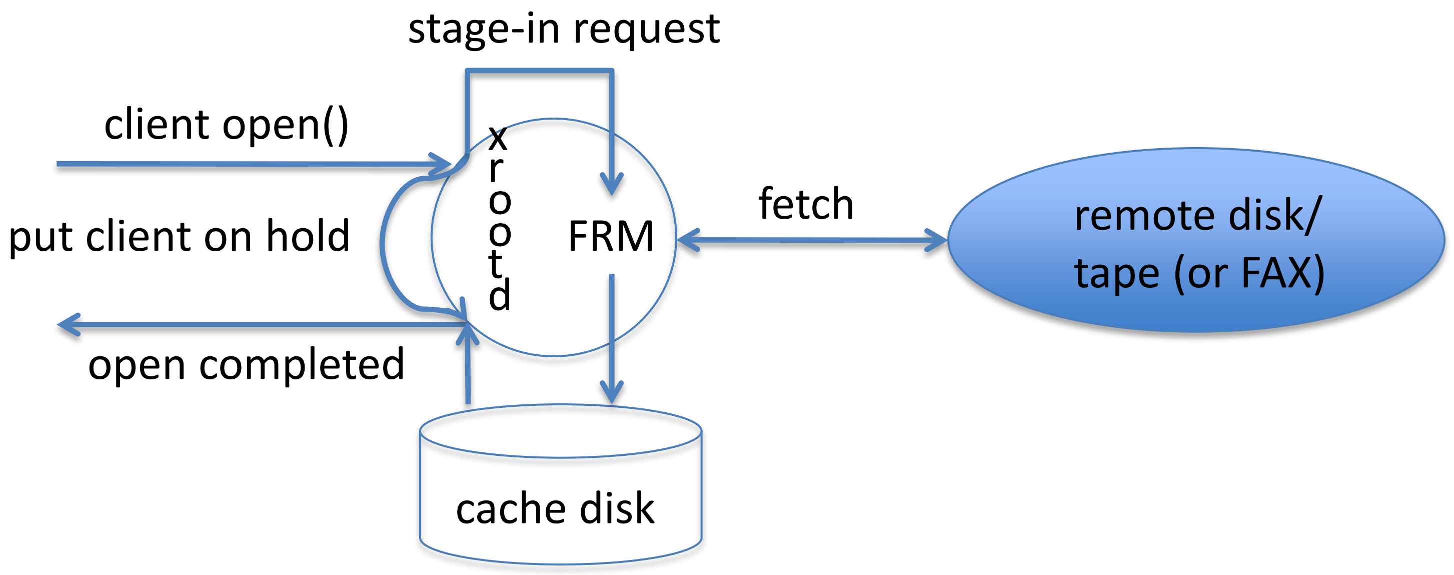
- Introduction, will not go into tech detail

- Whole file caching using FRM
  - hardened by years of real world usage
- Whole/Partial file caching using Proxy Cache
  - new, with real world usage experience from CMS
- non-Xrootd caching

File Residency Manager

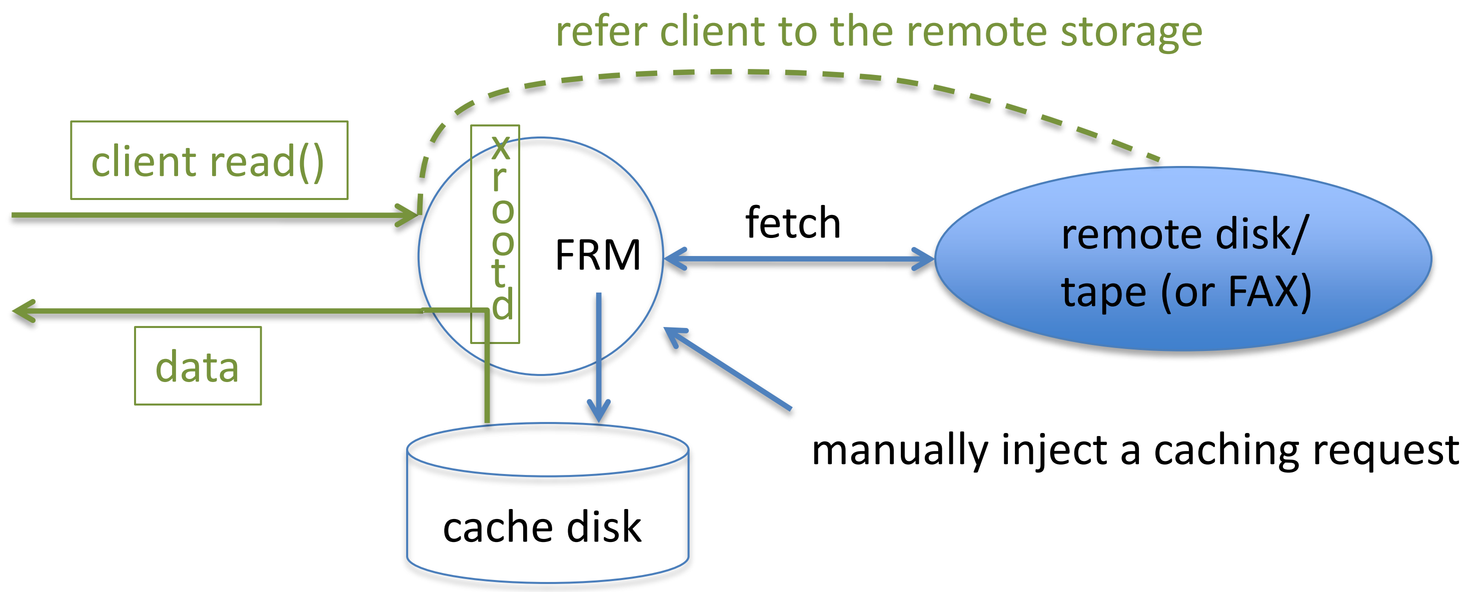
- Evolved from the xrootd interface to tape system
- Whole file caching, (external) policy based deletion
- Script interface, very flexible

- Example 1: Automatic file stage-in/caching
  - e.g. WT2's 2-tier storage



File Residency Manager, cont'd

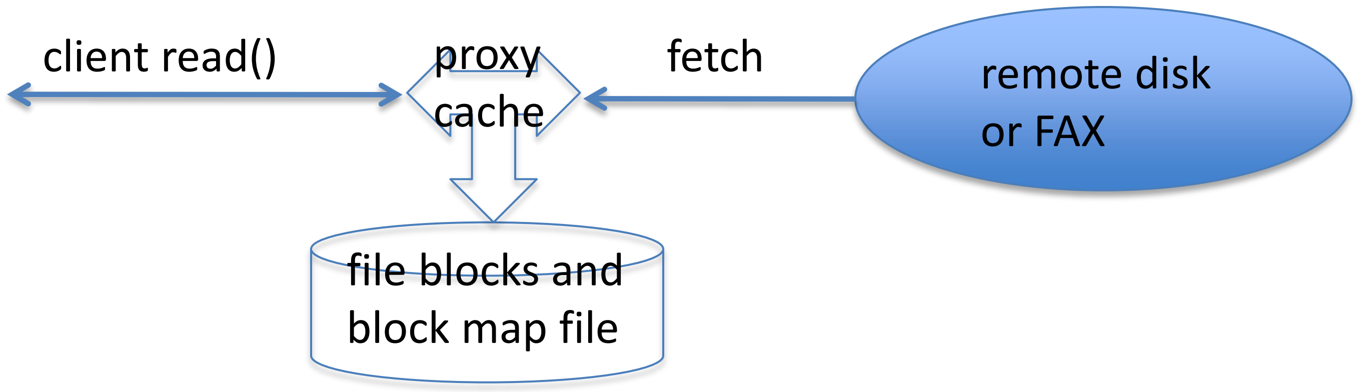
- case 2: manually inject caching request (external caching decisions)
  - xrootd and FRM operate independently
  - caching decision: NOT a cache replacement algorithm
  - caching request/decision is **independent** of the current client reading requests
  - write your own cache replacement algorithm, or use built-in LRU
  - e.g. SSD cache box at WT2. External decision come from week long usage history and Panda's job input file list



refer client to the remote storage
client read()
xrootd
FRM
fetch
remote disk/ tape (or FAX)
data
cache disk
manually inject a caching request

Proxy Cache

- Whole file or file block caching
- Serving files immediate after the required file blocks are available



- Fetch data from Xrootd storage, including FAX
  - probably will work with other types of storage via plug-ins
  - not designed to match FRM's flexibility
  - there is an interface for external caching policy
    - not sure how mature it is

Proxy Cache cont’d

- Easy to setup
  - configuration is similar to a simple proxy.
- Designed to use HDD RAID as cache
  - should also work well with a single SSD
  - need some work in order to use with SSD JBOD without burning the SSDs with unnecessary writes.

non-Xrootd caching

- Many implementations exists that using conventional caching techniques and SSDs. e.g.
  - Facebook flashcache
    - Kernel level cache, evaluated (and gave up) at WT2 in 2010)
  - SSD in storage box
    - In some cases, transparent to admins
- File block caching
  - Only cache data on the same data server
- Caching algorithms
  - LRU, ARC, etc. algorithms look at usage pattern from the most recent period
    - This is **short** period: seconds, minutes or hours.
  - They are actually cache replacement algorithms
    - The most recent block is always cached.
    - SSD caching will be burned fast
  - Maybe good for Tier 3 usage pattern (no experience)
  - Not good to capture Tier 2 usage pattern
    - WT2's SSD cache look for a week long usage pattern