



Project 2: AFS

Marvin Tan, Rachit Tibrewal, Yifei Yang, Bijan Tabatabai





Server-client interface

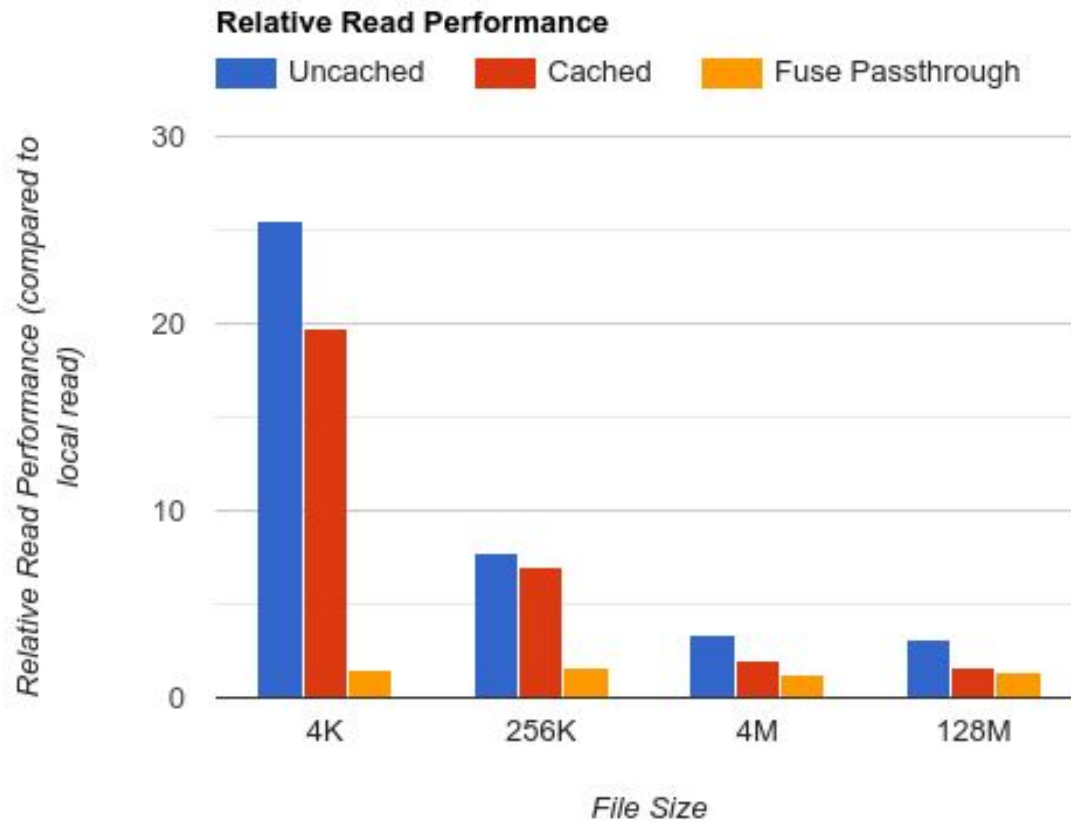
Fetch	Returns the status, data, and last modification time on the server.
Store	Stores the data and returns the status and last modification time on the server.
Remove	Delete the specified file.
Create	Create a new file.
Rename	Change the name of a file or directory.
Mkdir	Creates a new directory.
Removedir	Deletes the specified directory which must be empty.
Stat	Return status of a file or directory.
Readdir	Returns the list of files and directories in the directory.

Design Decisions

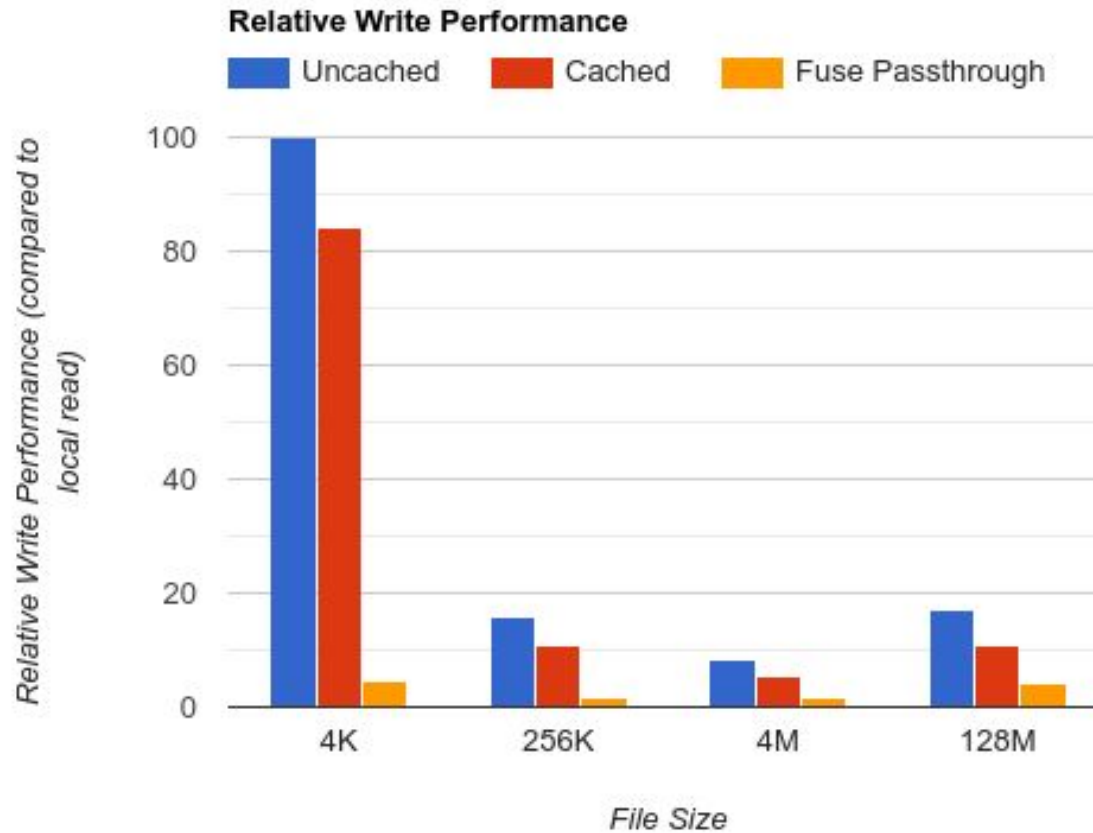
- Retrieve on open if cache is outdated, and flush on close if cache is dirty
 - A version file is created/refreshed upon file retrieval from server
 - Determine if cache outdated -> compare server file last modified time to the version file's
 - Determine if cache dirty -> compare cache file last modified time to the version file's
- Durable client side change on crash
 - Write is made to cache file stored on disk
- Prevent file corruption caused by server crash during file upload
 - Write to the file uploaded by client to a tmp file first, and rename it to the actual file name after file close



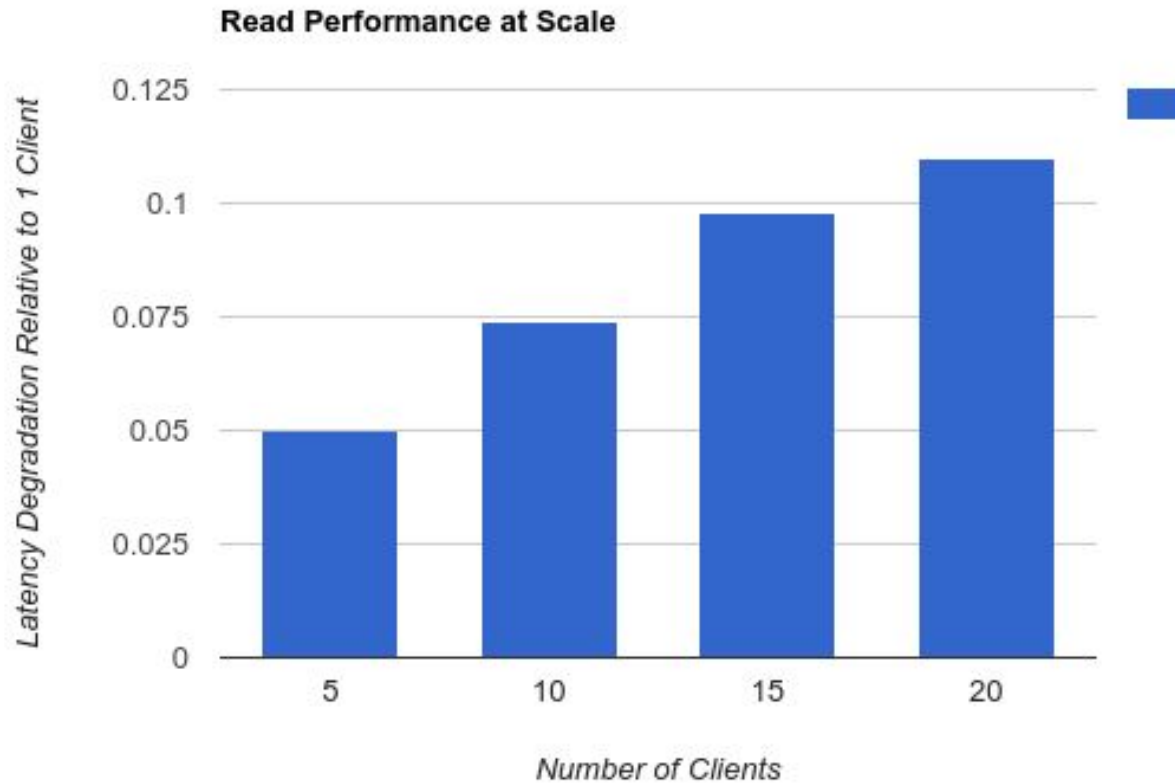
Read Performance



Write Performance



Our System at Scale



Potential performance improvements

- Don't retrieve from server for open() when flag O_TRUNC is set, since we are going to truncate the file anyway.
- Compress large files before sending over gRPC
- Maintain cache of hot file descriptors on server for reads





Reliability demo

[Link](#)

- Client crash points
 - Before/after “Stat” rpc call in “afs_open”
 - Before/after “read” cache copy in “afs_read”
 - Before/after “write” cache copy in “afs_write”
 - Before/after “Store” rpc call in “afs_flush”
- Server crash points
 - In “Stat” rpc call
 - In “Fetch” rpc call
 - Before/after “write” persistent file in “Store” rpc call
- Crash lasts for a fixed amount of time (e.g. 500ms)