Efficient File Distribution in a Configurable, Wide-area Distributed File System

Irene Zhang, Jeremy Stribling, Frans Kaashoek

MIT CSAIL

Background

WheelFS is a wide-area distributed file system designed to be a general storage solution for wide-area applications. The goal of WheelFS is to simplify wide-area applications by helping them cope with the challenges of sharing data over the wide-area network. A wide range of distributed applications can use WheelFS as a storage layer because applications can change the behavior of WheelFS to fit their requirements using *semantic cues*.

Problem Statement

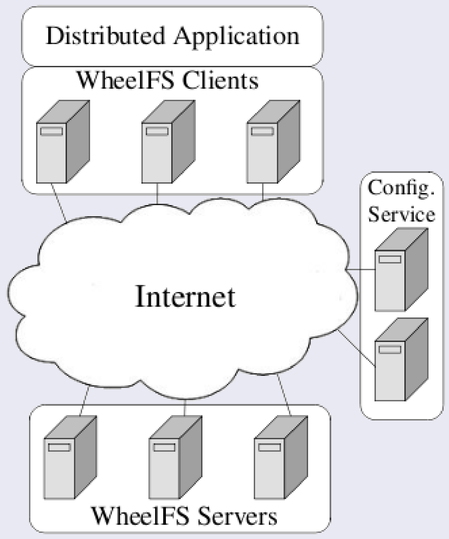
Many applications require a storage layer that can efficiently distribute files. The limited bandwidth and high latency of wide-area links make it difficult for most file systems to serve large files quickly and scale to handle many simultaneous requests for a file. Our goal is to ensure that WheelFS can provide efficient file distribution for wide-area applications.

Solution

Tread adapts prefetching, a traditional file system optimization technique, to a wide-area file system. Tread is a prefetcher for WheelFS with several features that help WheelFS improve performance:

- **File prefetching.**
  Tread performs read-ahead prefetching by default. Applications can request whole file prefetching, where the entire file is prefetched at once, using a semantic cue.
- **Directory prefetching.**
  Tread prefetches file and directory attributes to make listing directories faster.
- **Prefetching with cooperative caching.**
  Application can request cooperative caching in WheelFS for popular files. When an application requests cooperative caching and prefetching together, Tread prefetches from several peers in parallel and randomizes the order in which the file is fetched to avoid synchronization among peers.
- **Adaptive rate-limited prefetching.**
  Tread adaptively limits the rate of prefetching to avoid overloading the WheelFS servers or the network with prefetch requests.

WheelFS Overview



A WheelFS deployment consists of clients and servers scattered across the wide-area network. A single node can be both a WheelFS client and server. In addition, several sites run a configuration service that tracks which servers store each file.

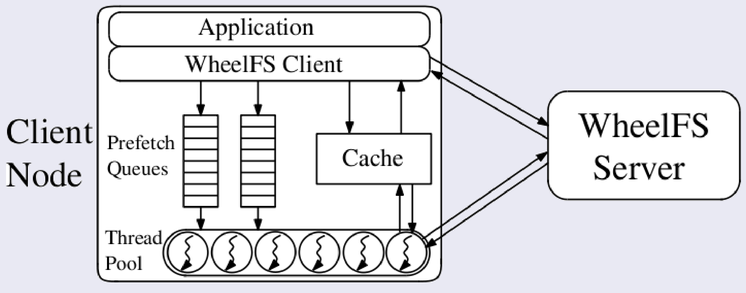
Semantic Cue Example

`/wfs/.WholeFile/.Hotspot/name`

This pathname with semantic cues is parsed as:

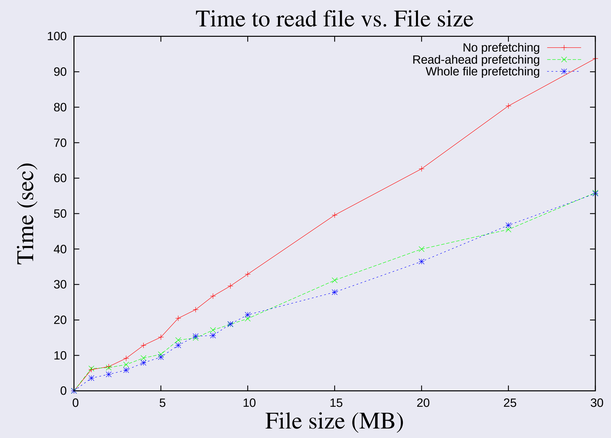
- `/wfs` – where WheelFS is mounted
- `.WholeFile` – turns on whole file prefetching
- `.Hotspot` – turns on cooperative caching
- `name` – the name of the file or directory

Tread Overview



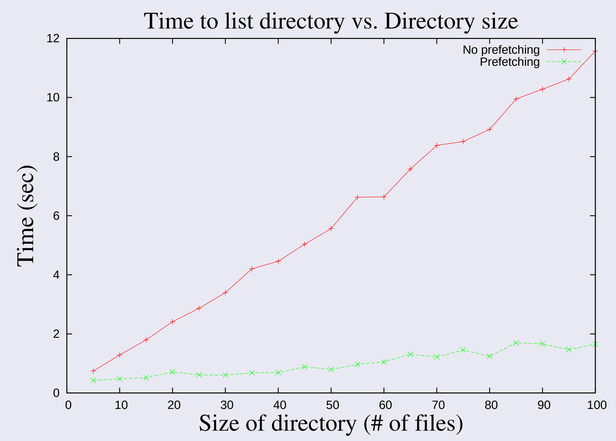
The WheelFS client makes a prefetch request by placing it in the queue. A thread from the thread pool will pick up the prefetch request and make a request to the WheelFS servers. The thread waits for a response and then stores the returned data in the WheelFS client's local cache for the client to use later.

File Prefetching



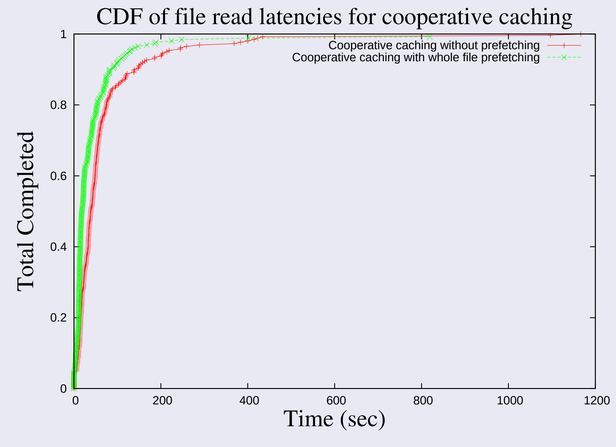
Time to read different sized files over a 100 ms RTT, 6 Mbps link with no prefetching, read-ahead prefetching and whole file prefetching.

Directory Prefetching



Time to read different sized directories over a 100 ms RTT, 6 Mbps link with and without prefetching.

Prefetching with Cooperative caching



CDF of latencies for 270 Planetlab nodes to read a 10 MB file simultaneously.