Wide-area cooperative storage with CFS

Ryan Stern

stern@cs.colostate.edu

Big Data Lab

Department of Computer Science

Colorado State University

Outline

- System Goals
- System Architecture
- File Blocks
- Safeguards
- Results

System Goals

- Decentralized control
- Scalability
- Availability
- Load balance
- Persistence
- Efficiency

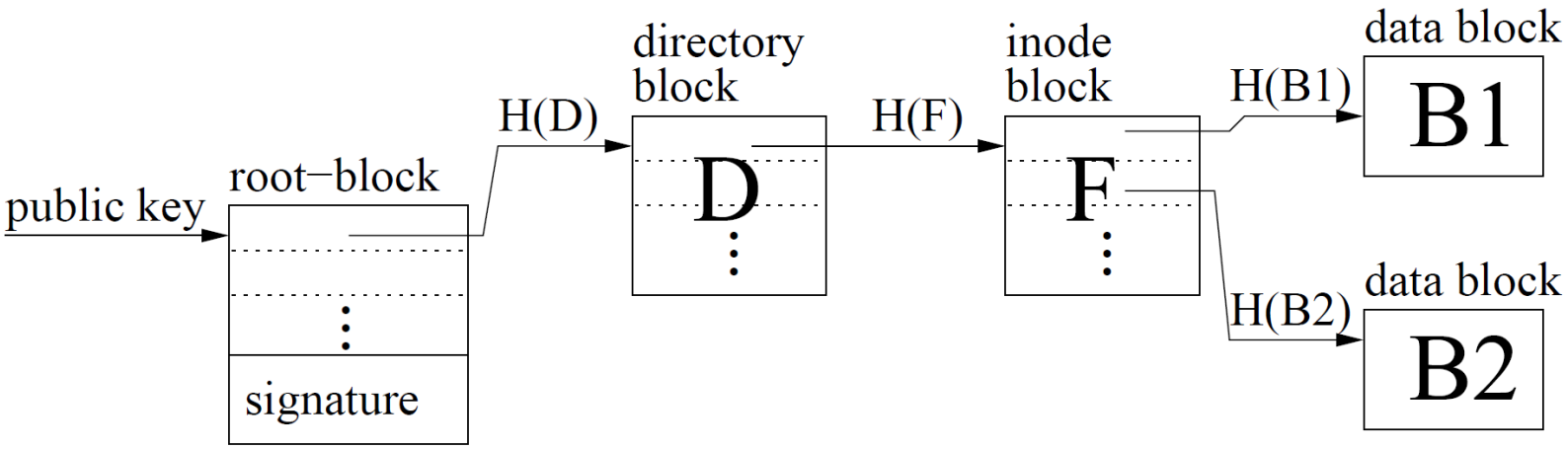
System Architecture

- 3 layers

- FS Layer:
  - The file system interface
- DHash Layer:
  - Stores and retrieves file blocks
  - Handles replication and caching
- Chord Layer:
  - Locates file blocks

File Blocks

- Each file is split into blocks
- Most blocks are stored using a hash of their content as the key
- The root block is signed and the hash of the public key is used

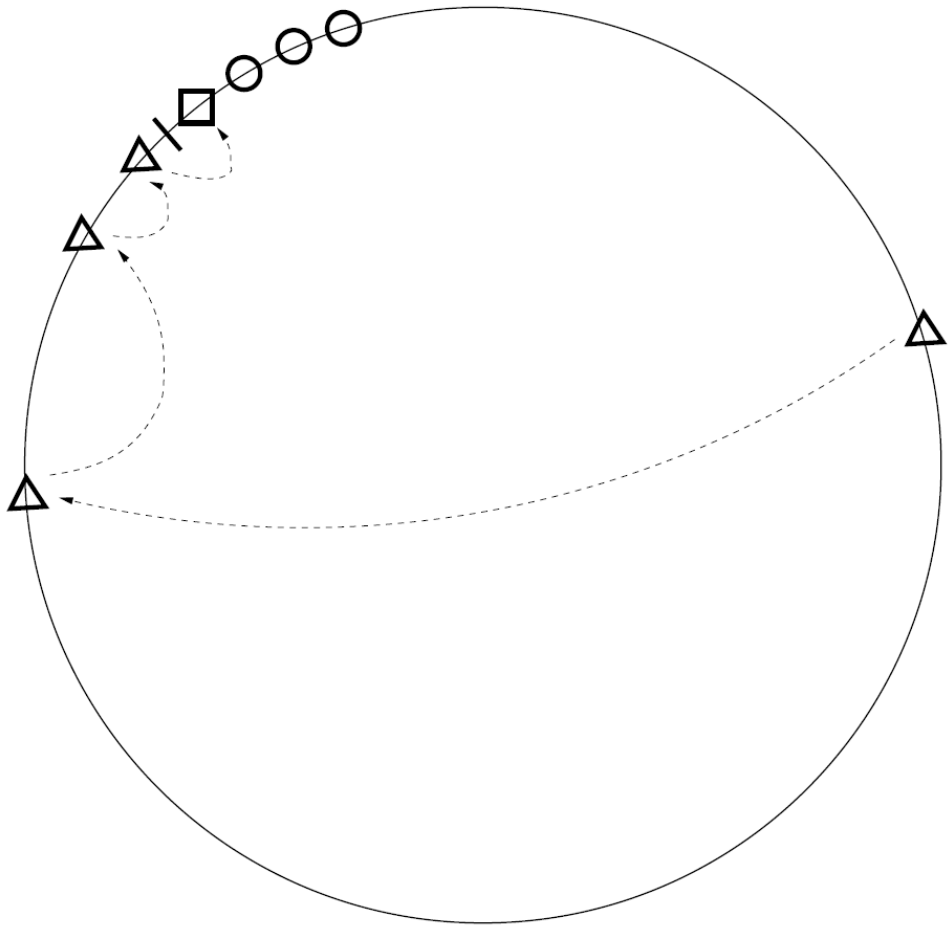


Block Replication and Caching

- Blocks are replicated on k Chord successors
  - For failure resilience
  - Clients are allowed to download from any of the servers

- Blocks are also cached
  - To help prevent servers holding popular files from becoming overloaded
  - Storage for cache is set aside by each node
  - Copies of blocks are left along the lookup path
  - Effective since the same nodes tend to be visited late in the lookup
  - Cached blocks are replaced in LRU order
  - Number of cached copies depends on popularity of the file

Block Replication and Caching



File Updates and Deletion

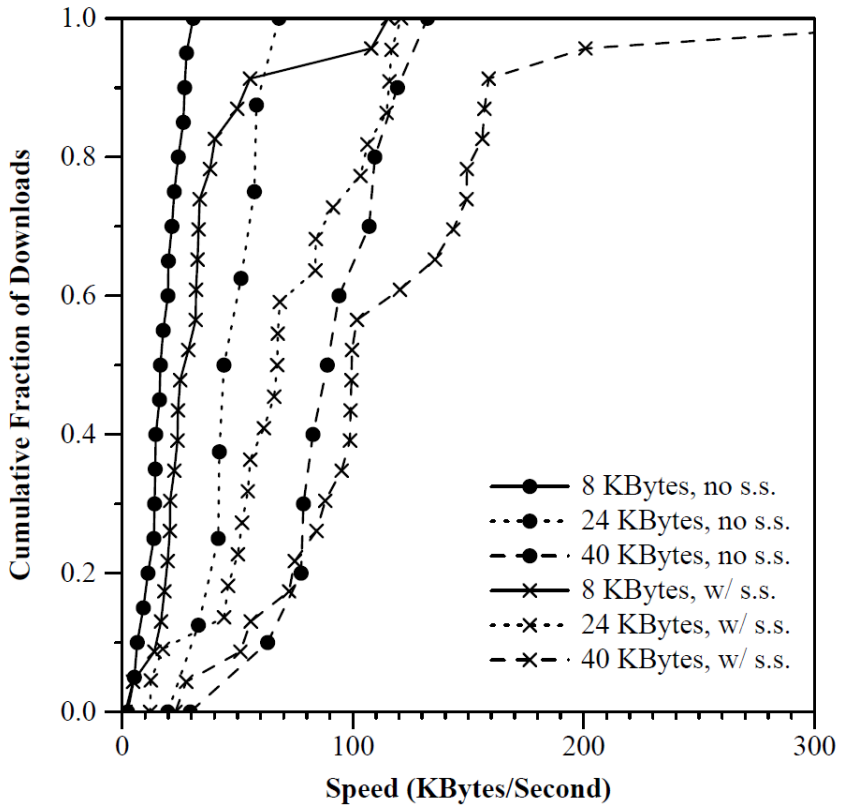
- Only the publisher of a file may modify it
  - The file system is read-only for everyone else
- No delete operation is provided
  - Publishers must periodically indicate that the blocks should continue to be stored
  - The system automatically deletes blocks that are not refreshed

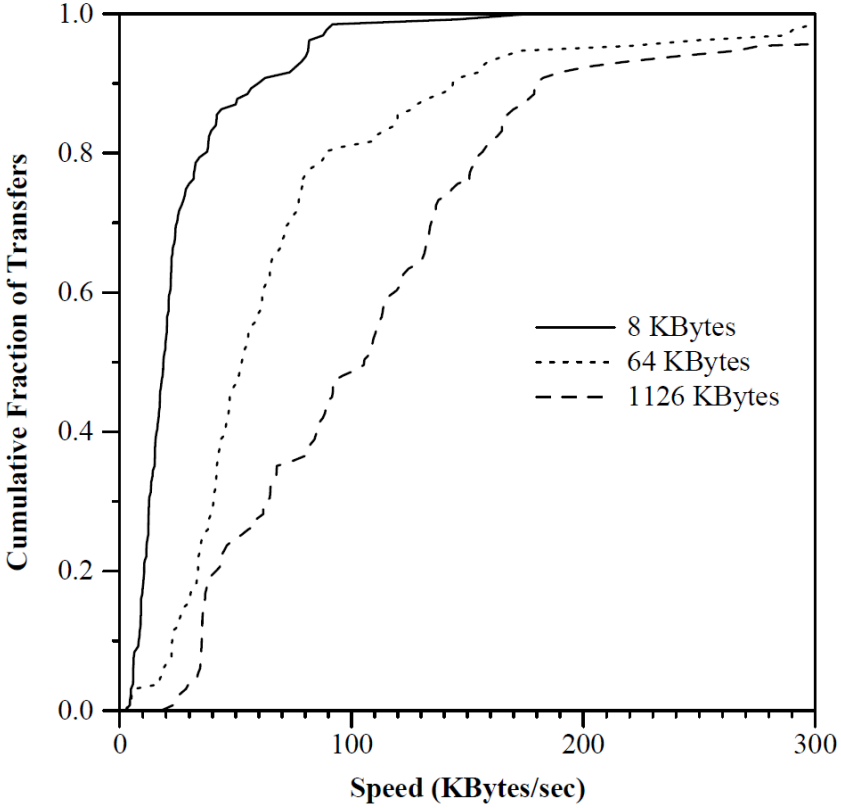
Function	Description
`put_h(block)`	Computes the block's key by hashing its contents, and sends it to the key's successor server for storage.
`put_s(block, pubkey)`	Stores or updates a signed block; used for root blocks. The block must be signed with the given public key. The block's Chord key will be the hash of `pubkey`.
`get(key)`	Fetches and returns the block associated with the specified Chord key.

Safeguards Against Rogue Participants

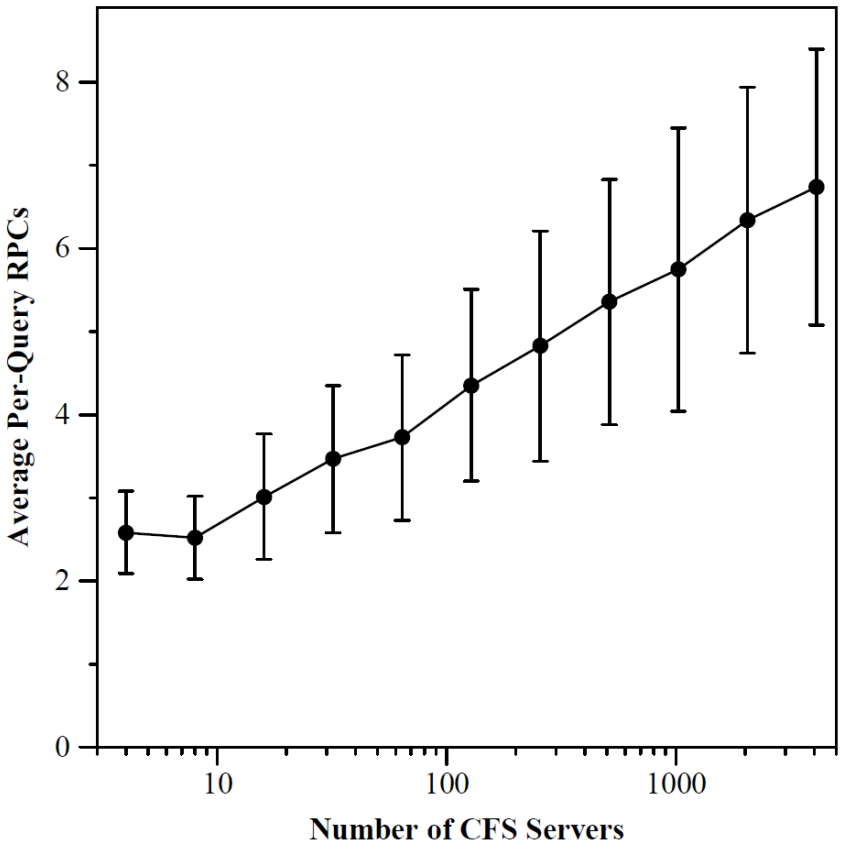
- Quota
  - Limit the amount of data a client can inject into the system
  - Based off of IP addresses
- Node IDs are authenticated
  - Nodes have very limited control over their Chord ID
  - Prevents nodes from denying access to certain files

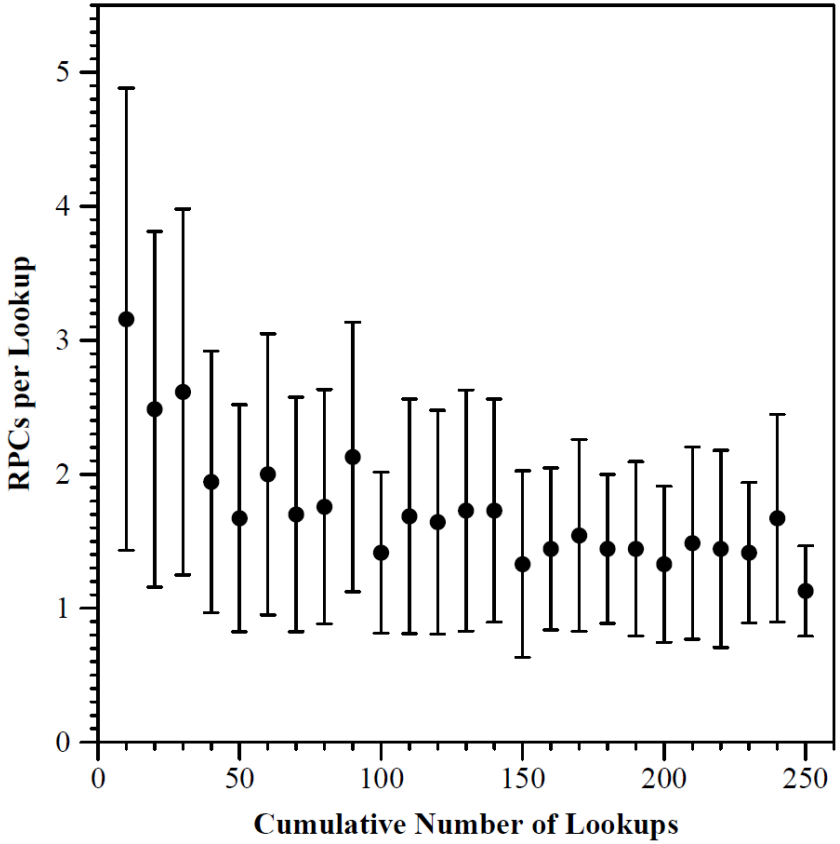
Results: Download Speeds



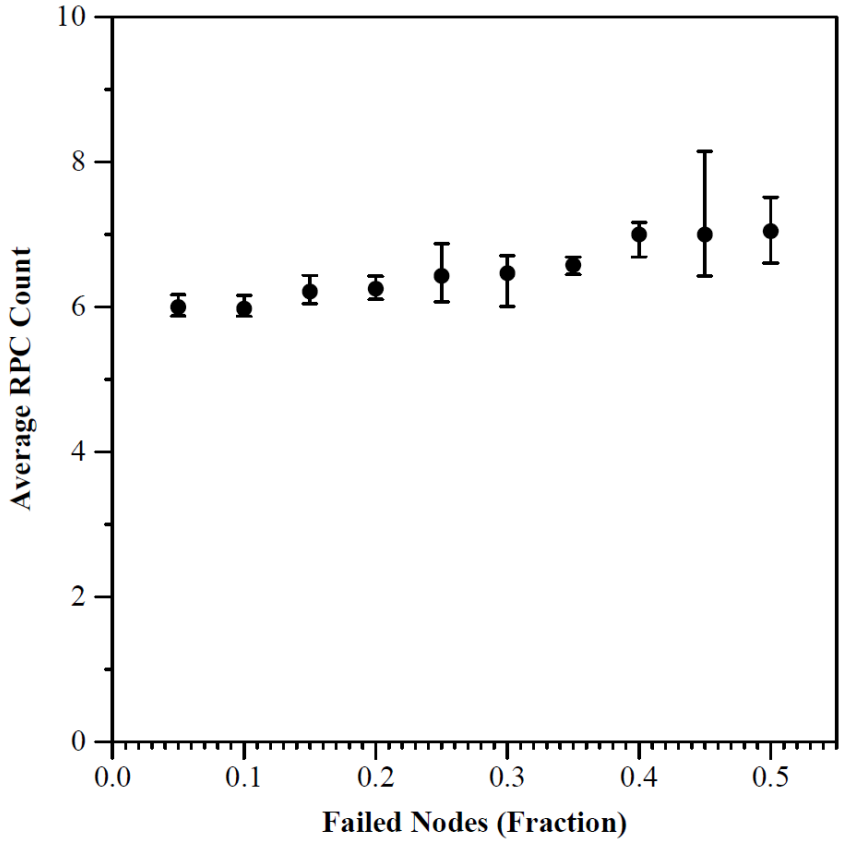


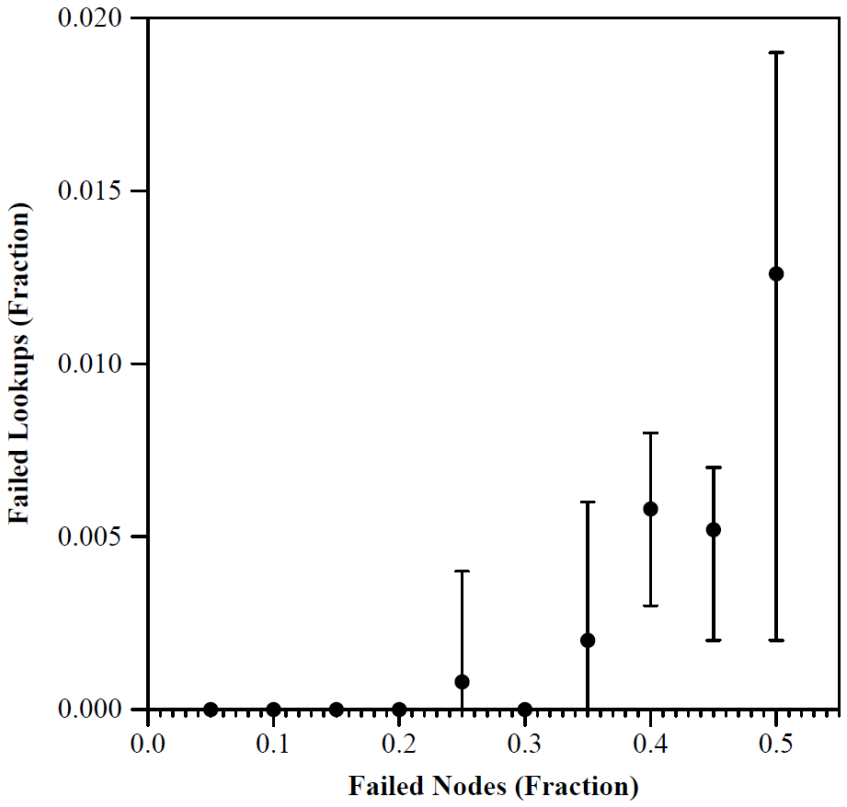
Results: Number of RPCs to obtain a Block





Results: Failures





Questions?