



Solutions for Fourth Quiz

COSC 6360

Fall 2015



First question

- Consider a diskless client trying to access a file named “**/usr/joe/6360/paper.doc**” that is stored on its NFS server.
- Assuming that the client already has a handle for its root directory, how many **lookup()** requests will it issue?
 - **Answer:** _____ requests
- What does NFS do to ***speed up*** these requests?



First question

- Consider a diskless client trying to access a file named “**/usr/joe/6360/paper.doc**” that is stored on its NFS server.
- Assuming that the client already has a handle for its root directory, how many **lookup()** requests will it issue?
 - **Answer: four** requests
- What does NFS do to ***speed up*** these requests?



First question

- Consider a diskless client trying to access a file named “**/usr/joe/6360/paper.doc**” that is stored on its NFS server.
 - Assuming that the client already has a handle for its root directory, how many **lookup()** requests will it issue?
 - **Answer: four** requests
 - What does NFS do to ***speed up*** these requests?
 - **NFS lets client cache file handles**



Second question

- What is a ***stale file handle***?
- What does NFS do to detect them?



Second question

- What is a *stale file handle*?
 - **A stale file handle is a file handle that points to the i-node of a file that was deleted.**
- What does NFS do to detect them?



Second question

- What is a *stale file handle*?
 - **A stale file handle is a file handle that points to the i-node of a file that was deleted.**
- What does NFS do to detect them?
 - **The NFS server compares the generation number of the file handle with the generation number of the i-node. If the numbers do not match, the file handle is stale.**



Third question

- Consider a distributed file system implementing ***close-to-open consistency***. Assuming that
 - Alice opens a file at **9:30 AM**, modifies it and closes it at **10:15 AM**,
 - Bob opens the same file at **10:00 AM**, modifies it and closes it at **10:30 AM**,
 - Carol opens the same file at **10:20 AM**, modifies it and closes it at **11:30 AM**,



Third question

- Which of these three users would see his or her changes actually incorporated in the final version of the file?

Alice Bob Carol



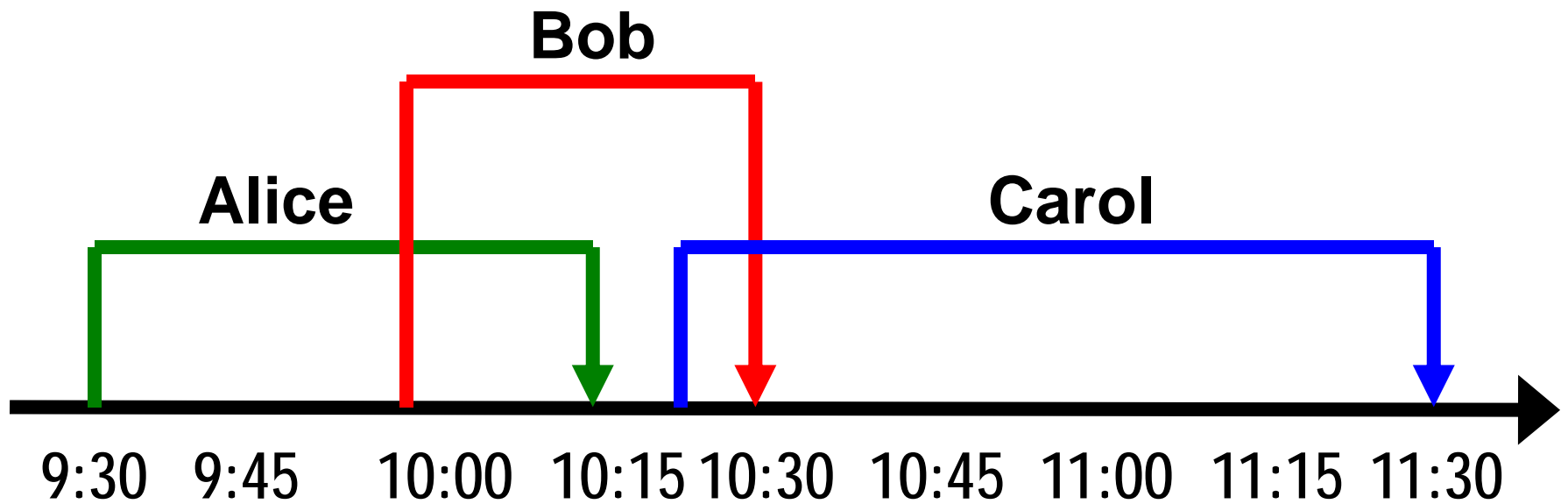
Third question

- Which of these three users would see his or her changes actually incorporated in the final version of the file?

■ **Alice** **Bob** ■ **Carol**



Third question





Fourth question

- Consider a hypothetical version of NFS that would use **leases** to control access to its files.
- Would the server of this file system still be **stateless**?



Fourth question

- Consider a hypothetical version of NFS that would use **leases** to control access to its files.
- Would the server of this file system still be **stateless**?
 - **NO, because the server will have to maintain information on leases**



Fourth question

- Consider a hypothetical version of NFS that would use **leases** to control access to its files.
- Would that make the service **less robust**?
Why?



Fourth question

- Consider a hypothetical version of NFS that would use **leases** to control access to its files.
- Would that make the service **less robust**?
Why?
 - **No, all lease information will be obsolete by the time a failed NFS server will restart**



Fifth question

- How does FARSITE store *users' secret keys*?
Why?



Fifth question

- How does FARSITE store *users' secret keys*?
Why?
- **Farsite encrypts user private keys with a symmetric key derived from user password and stores them in a globally-readable Farsite directory**
- **This frees the user from the burden of managing a fairly long non-mnemonic key.**



Sixth question

- Why is the ***CRUSH function*** in the Ceph distributed file system ***publicly available*** ?
- How does it simplify the design of the Ceph metadata server cluster?



Sixth question

- Why is the ***CRUSH function*** in the Ceph distributed file system ***publicly available***?
 - **CRUSH is publically available because Ceph clients must use it to locate object replicas**
- How does it simplify the design of the Ceph metadata server cluster?

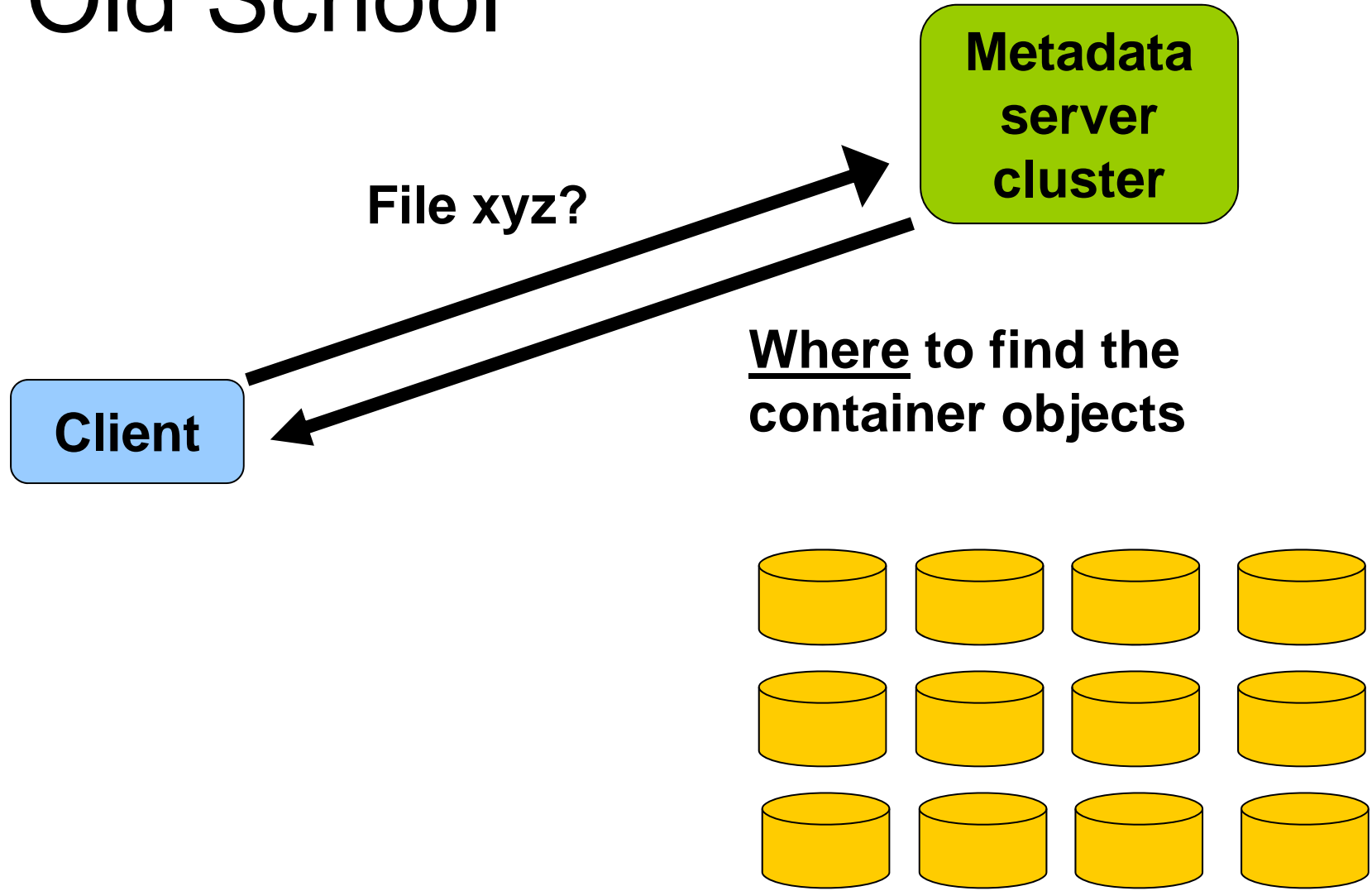


Sixth question

- Why is the ***CRUSH function*** in the Ceph distributed file system ***publicly available*** ?
 - **CRUSH is publically available because Ceph clients must use it to locate object replicas**
- How does it simplify the design of the Ceph metadata server cluster?
 - **Thanks to CRUSH, the Ceph metadata server does not have to maintain object lists.**

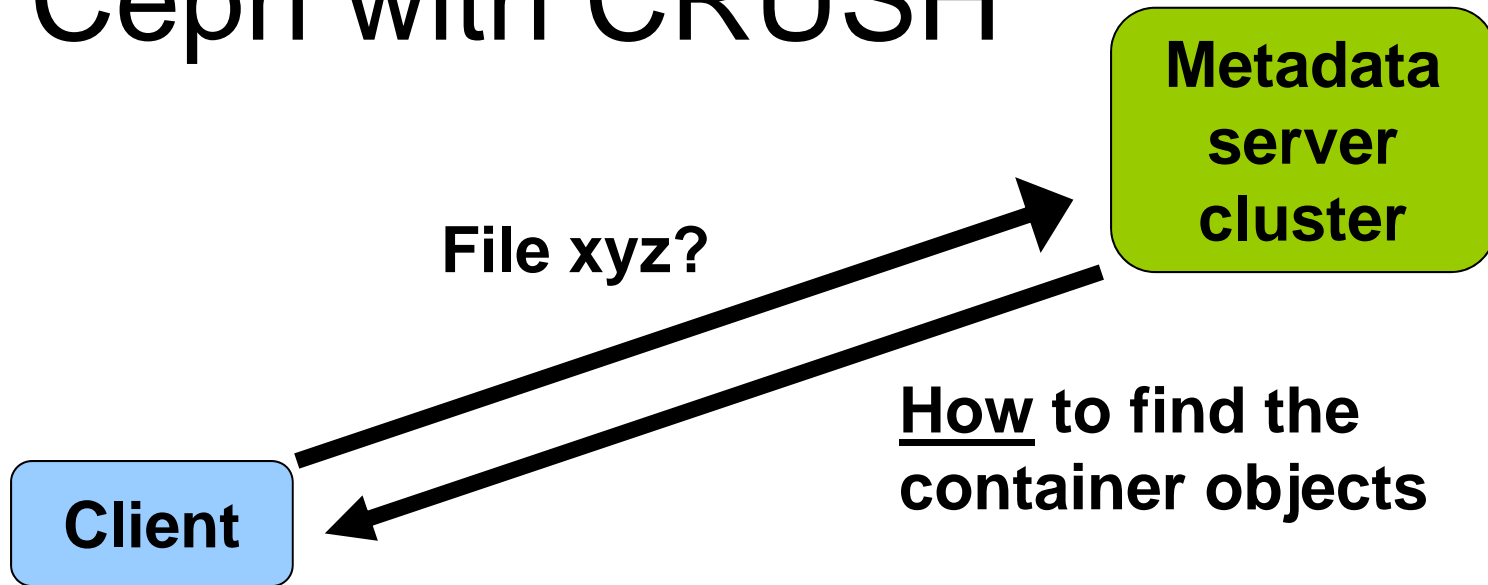


Old School





Ceph with CRUSH



Client uses CRUSH and data provided by MDS cluster to find the file

