COSC 6360: Operating Systems
MATERIALS ON THE FOURTH FALL 2020 QUIZ
November 16, 2020

You are only responsible for the materials discussed in class as they are ***summarized*** in the handouts and discussed in the PowerPoint presentations. I expect you to understand these summaries and to be able to comment around them. Always ask yourself ***why*** a specific technique was used and ***which*** problem it addressed.

File Systems

C. Lee, D. Sim, J.-Y. Hwang, and S. Cho, F2FS: A New File System for Flash Storage, *Proc. 13th USENIX Conf. on File and Storage Technologies* (FAST '15).

- *You should understand how their node address table differs from the Sprite LFS i-node map, what differentiates internal nodes from external nodes, why they chose multi-head logging, and why they have implemented adaptive logging.*(F2FS.pptx)

R. Sandberg, D. Goldberg, S. Kleiman, Dan Walsh and Bob Lyon, "Design and implementation of the Sun network filesystem," *Proc. Summer 1985 USENIX Conference*, pp. 119–130.

- *We covered the paper in some detail. You should understand why the designers of NFS decided to select a* stateless server *and how their decision affected the overall design of the system. Do not go at a much lower level of detail than the PowerPoint slides.* (NFS.pptx)

D. Hitz and A. Watson, "The Evolution of NFS," Network Appliance, Inc. http://www.netapp.com/tech_library/evolution.html

- *You are only responsible for the discussion of* close-to-open consistency. ***Skip the rest of the paper.***

C. G. Gray and D. R. Cheriton, "Leases: An efficient fault-tolerant mechanism for distributed file cache consistency," *Proc. 12th SOSP*, pp. 202-210, Dec. 1989.

- *You are responsible for all the topics discussed in the PowerPoint slides* (Leases.pptx) *but can* skip the analytical study.

S. Ghemawat, H. Gobioff, and S.-T. Leung. "The Google File System," *Proc. 19th SOSP*, Lake George, NY, pp. 29-43 Oct. 2003.

- *You are only responsible for the contents of the PowerPoint slides. Among others, we did not cover shadow masters*

S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance, distributed object-based storage system," *Proc. 7th Symposium on Operating Systems Design and Implementation*, Nov. 2006.

- *You are responsible for all the topics discussed in the PowerPoint slides.* (Ceph.ppt)

Review questions

1. What is the main advantage of ***multi-head logging***?
2. What are the main advantage and the main disadvantage of ***stateless servers***?
3. What is a ***stale file handle***? What does NFS do to detect them?
4. How would you implement ***close-to-open consistency in NFS***?
5. What is the main advantage of ***leases***?
6. In the GFS, what are the disadvantages of selecting a chunk size that is either too small or too big?
7. In the GFS, what are ***atomic record appends***?
8. What is the purpose of the ***CRUSH function*** in the Ceph distributed file system? How does it simplify the design of the Ceph metadata server cluster?

Answers: *I. Multi-head logging allows F2FS to group together in each logs data with similar hotness. II. They are more robust than stateful servers are but cannot detect shared file accesses. III. A stale file handle refers to a file that has been deleted. NFS detects stale file handles by checking their generation numbers. IV. Close-to-open consistency requires that the client forwards to the server at close time all the blocks it has modified and checks at open time whether it has the most recent version of the file it wants to open. V. Holding a lease to a given object gives the leaseholder a guarantee that no other entity will update the object until either the leaseholder is notified or the lease period ends. (Note: leases work less well in disconnected mode as disconnected users cannot get—or renew—leases.) VI. Neither the clients nor the chunk servers cache files because it would have little benefits in a streaming environment and Linux I/O buffers already keep in RAM frequently accessed chunk. VII. Atomic record appends append data atomically, at least once, and at an offset of GFS choosing. VIII. The CRUSH function is a pseudo-random function that lets clients assign placement groups to object storage devices. It eliminates the need for storing object addresses in the cluster.*