Solutions for the Fourth Quiz

COSC 6360

Fall 2017

First question

- What is the main advantage of journaling file systems using ***asynchronous log updates*** over other journaling file systems?
  - □ ***They are faster***
- What is the main advantage of journaling file systems using ***synchronous log updates*** over other journaling file systems?
  - □ ***They guarantee the durability of updates.***

Second question

- Why does NFS require all its client requests to be both ***self-contained*** and ***idempotent***?
  - □ ***Self-contained requests are essential for the server to remain stateless.***
  - □ ***Idempotent request allows the client to repeat requests for which it did not get a reply.***

Third question

- How do ***leases*** guarantee the ***consistency*** of file updates in the presence of ***communication failures***?
  - ***The server cannot accept any update requests except from the client that holds a write lease on the file and cannot cancel the lease of a disconnected client.***
- What is the impact of the same ***leases*** on the ***server recovery process***?
  - ***None***

Fourth question

- Consider a distributed file system implementing ***close-to-open consistency***. Assuming that
  - Alice opens the file at 9:20 AM, modifies it and closes it at 10:45 AM,
  - Bob opens the file at 10:00 AM, modifies it and closes it at 10:30 AM,
  - Carol opens the file at 10:50 AM, modifies it and closes it at 11:30 AM,
- Which of these three users would see his or her changes y incorporated in the final version of the file?

Fourth question

- Consider a distributed file system implementing ***close-to-open consistency***. Assuming that
  - Alice opens the file at **9:20** AM, modifies it and closes it at **10:45** AM,
  - Bob opens the file at 10:00 AM, modifies it and closes it at ~~10:30~~ AM,
  - Carol opens the file at **10:50** AM, modifies it and closes it at **11:30** AM,

  - **Alice and Carol**

Alternate fourth question

- Consider a distributed file system implementing ***close-to-open consistency***. Assuming that
  - Alice opens the file at 9:20 AM, modifies it and closes it at 10:45 AM,
  - Bob opens the file at 10:00 AM, modifies it and closes it at 10:30 AM,
  - Carol opens the file at 10:40 AM, modifies it and closes it at 11:30 AM,
- Which of these three users would see his or her changes y incorporated in the final version of the file?

Alternate fourth question

- Consider a distributed file system implementing ***close-to-open consistency***. Assuming that
  - Alice opens the file at 9:20 AM, modifies it and closes it at 10:45 AM,
  - Bob opens the file at **10:00 AM**, modifies it and closes it at **10:30 AM**,
  - Carol opens the file at **10:40 AM**, modifies it and closes it at **11:30 AM**,

  - ***Bob and Carol***

Fifth question

- How does LBFS identify ***chunks***?
    - ***LBFS identifies chunks by their hash values***

Fifth question

- How does LBFS identify ***chunks***?
  - □ It uses the ***collision-resistant property*** of ***SHA-1 hash function***

Sixth question

- Based on what you learned from the Skylight paper, explain why log-structured file systems (LFS) would be the most ***natural*** file system organization for ***shingled disks***.

  - ***Shingled disk organizations work best in append-only mode***

Sixth question

- How would the segments of an LFS designed for shingled disks would differ from conventional LFS segments?
  - ***Segments should be separated by guard regions***
    - ***Could result in much larger segments***

Seventh question

- How does Ceph map its objects into ***placement groups***?
  - ***Using a hashing function***
    - ***(Not Crush)***

Eighth question

- Why does FAWN use a ***log-structured datastore***?
    - □ To avoid small random writes, which flash memory cannot handle very well

Seventh question

- What is the major performance penalty occurring when Nooks crosses a ***lightweight protection domain boundary***?
  - ***Crossing protection boundaries requires switching the kernel page table, which results in a flush of the current TLB (and an avalanche of TLB misses).***