Based on what you read in "All File Systems are Not Created Equal", what properties might an ideal file system embody to provide both high performance and correctness for most data-intensive applications?

Answer:

For correctness:

1. Atomicity across System Calls
2. Atomicity within System Calls(Overwrite atomicity, Directory operation atomicity)
3. Ordering between System Calls(safe rename, safe file flush)
4. Durability

   For example, we need to guarantee a write() call atomic in the FS and write() calls sent to disk in-order.

For high performance: Trading freshness for increased performance. At the meantime of decrease flush, we need other technology to maintain the correctness.

In OptFS (and more generally in any system that writes data), how can checksums be used to replace flush operations?

Answer:

Computed over data and metadata;

Checked during recovery;

Mismatch indicates blocks were lost during crash.

Paper review:

It’s difficult to balance between high-performance and correctness for file system. When we use flush operation to guarantee the atomicity and in-order of file system, these operations will damage the performance of file system. And thus, all File Systems are Not Created Equal. Optimistic File System present a way to keep the correctness while improve performance by decrease flush opeartions, which is trading freshness for increased performance.