Recitation 19 — Consistency

Six Consistency Guarantees

- Strong Consistency: What you saw in lecture
- Eventual Consistency: Very weak. Reads can read any value that was written in the past.
- Consistent Prefix: Reads will observe an ordered sequence of writes (unlike eventual consistency, where this is not guaranteed)
- Bounded Staleness: Reads can be out of date, but not *too* out of date
- Monotonic Reads: Clients will observe increasingly up-to-date reads over time
- Read-My-Writes: Clients are guaranteed to be able to read their own writes (says nothing about whether other clients are guaranteed to read those writes)

Baseball Example: What possible scores could we read at the middle of the seventh inning?

- Strong consistency: only the current score
- Eventual consistency: All combinations of scores (including scores that never happened at the same time — e.g., the score of the game was never 1-5, and yet this is a possible read)
- Consistent prefix: All scores that actually occurred at some point, because we're guaranteed to see an ordered sequence of writes. So we'd never read, e.g., 2-0, because the home team "wrote" 3 before the visiting team wrote 2.
- Bounded Staleness: Lots of possibilities, but only going a limited amount of time in the past.
- Monotonic reads: Lots of possibilities, but nothing older than the first read (and increasing in time after that).
- Read-my-writes: Current score for the writer, all possible scores for everyone else.

Decisions

- The needs of the application dictate which consistency model(s) are appropriate.
    - Trading off consistency with performance (see Table 2)
    - Consider what causes poor performance for the stronger guarantees
- Sometimes we need to combine consistency models