Linearizability
- A Quick Overview

Ashvin Goel

Electrical and Computer Engineering
University of Toronto

ECE1724

These slides are modified versions of slides from Michael Freedman & Wyatt Lloyd’s course on Distributed Systems
Data Consistency Models

• Contract (or a set of guarantees) that a system provides to applications about expected behavior when data is accessed (read, written, updated, etc.)
Linearizability

- Assumption: Each operation accesses one data item

- All operations are performed in some total order

- The total order preserves the real-time ordering between operations
  - If operation A completes before operation B begins in real-time, then A is ordered before B
  - If neither A nor B completes before the other begins, then there is no real-time order
    - But there must be some total order
Understanding Linearizability

• Writes are ordered
  • Writes appear to occur instantaneously

• Reads read latest data
  • After a write completes, a later read (in real-time order) returns the value of the write (or later write)
  • Once a read returns a value, all later reads return that value or the value of a later write
Real-Time Ordering Examples

\[ P_A \vdash w(x=1) \]

\[ P_B \vdash w(x=2) \]

\[ P_C \vdash w(x=3) \]

\[ P_D \vdash w(x=4) \quad \vdash w(x=5) \]

\[ P_E \vdash w(x=6) \]
Linearizable?

\[ P_A \vdash w(x=1) \]

\[ P_B \vdash w(x=2) \]

\[ P_C \vdash w(x=3) \]

\[ P_D \vdash w(x=4) \vdash w(x=5) \]

\[ P_E \vdash w(x=6) \]

\[ P_F \vdash r(x)=1 \vdash r(x)=2 \vdash r(x)=3 \vdash r(x)=6 \vdash r(x)=5 \]
Linearizable: Yes

\[ P_A \quad \text{w}(x=1) \quad \]

\[ P_B \quad \text{w}(x=2) \quad \]

\[ P_C \quad \text{w}(x=3) \quad \]

\[ P_D \quad \text{w}(x=4) \quad \text{w}(x=5) \quad \]

\[ P_E \quad \text{w}(x=6) \quad \]

\[ P_F \quad r(x)=1 \quad r(x)=2 \quad r(x)=3 \quad r(x)=6 \quad r(x)=5 \quad \checkmark \]
Linearizable?

\[ P_A \quad \text{w}(x=1) \]

\[ P_B \quad \text{w}(x=2) \]

\[ P_C \quad \text{w}(x=3) \]

\[ P_D \quad \text{w}(x=4) \quad \text{w}(x=5) \]

\[ P_E \quad \text{w}(x=6) \]

\[ P_F \quad r(x)=1 \quad r(x)=2 \quad r(x)=3 \quad r(x)=6 \quad r(x)=3 \]
Linearizable: No

\[ P_A \mid w(x=1) \mid \]
\[ P_B \mid w(x=2) \mid \]
\[ P_C \mid w(x=3) \mid \]
\[ P_D \mid w(x=4) \mid w(x=5) \mid \]
\[ P_E \mid w(x=6) \mid \]
\[ P_F \mid r(x)=1 \mid r(x)=2 \mid r(x)=3 \mid r(x)=6 \mid r(x)=3 \]
\[ \text{stale read} \]
Why Linearizability?

• Behavior is like single machine processing one request at a time
  • Hides the complexity of distributed and replicated systems from applications
  • Hides complexity associated with failures
  • Easier to write correct applications

• Atomic broadcast (Zab protocol used by ZooKeeper), RAFT, PAXOS, etc., provide linearizability for replicated data stores

• However, linearizability is a strong consistency guarantee that can limit performance
  • We will discuss this issue today