Consistency
Spring 2024
Consistency Models

- Strict Serializability
- Linearizability
- Sequential
- Causal+
- Eventual

Stronger to Weaker
Consistency Models

- Strict Serializability
- Linearizability
- Sequential
- Causal+
- Eventual

Stronger to Weaker
Transactions: operations that span multiple objects (e.g., keys in KV store) \textit{atomically} commit (or abort).

Total order: There exists some legal total ordering of transactions.
- Legal (intuitively defined for strict serializability): in the total ordering, read operations “see” the latest write operation.

Preserves \textit{real-time commit order}: if \textit{txn} A commits before \textit{txn} B begins, then \textit{txn} A occurs before \textit{txn} B in the total order.
- Write ops in a committed txn are visible to all future txns’ read ops.
- Intuition: once a read “sees” a txn and commits, all future reads must also “see” that txn.

**Pros:** applications can easily reason about correctness of transactions.
**Cons:** strict serializability imposes high read and write latencies on system.
Strict Serializability Example

<table>
<thead>
<tr>
<th></th>
<th>Strictly Serializable?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P1:</td>
<td>{W(x)b, W(y)b}</td>
<td>Yes</td>
</tr>
<tr>
<td>P2: {W(x)a}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3:</td>
<td>{R(x)a}</td>
<td></td>
</tr>
<tr>
<td>P4:</td>
<td>{R(x)b}</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Strictly Serializable?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P1:</td>
<td>{W(x)b, W(y)b}</td>
<td>No</td>
</tr>
<tr>
<td>P2: {W(x)a}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3:</td>
<td>{R(y)b}</td>
<td></td>
</tr>
<tr>
<td>P4:</td>
<td>{R(x)b}</td>
<td></td>
</tr>
</tbody>
</table>
Consistency Models

Strict Serializability → Linearizability → Causal+ → Eventual

Stronger → Linearizability → Causal+ → Eventual → Weaker
Linearizability

- **Total order**: There exists some legal total order of operations (not txns).
- **Difference from strict serializability?**
  - Single-object operations! No transactions!
- **Preserves real-time ordering**: if an operation $A$ completes before operation $B$ begins, then op $A$ occurs before op $B$ in the total order.
  - A completed write op is visible to all future read ops.
  - Intuition: once a read “sees” a new write, all future reads must also “see” that write.

**Pros**: Easy to reason about correctness

**Cons**: High read and write latencies
## Linearizability Example

<table>
<thead>
<tr>
<th>Linearizable?</th>
<th>No</th>
<th>Linearizable?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1:</td>
<td>W(x)a</td>
<td>P1:</td>
<td>W(x)a</td>
</tr>
<tr>
<td>P2:</td>
<td>W(x)b</td>
<td>P2:</td>
<td>W(x)b</td>
</tr>
<tr>
<td>P3:</td>
<td>R(x)b</td>
<td>R(x)a</td>
<td>P3:</td>
</tr>
<tr>
<td>P4:</td>
<td>R(x)b</td>
<td>R(x)a</td>
<td>P4:</td>
</tr>
</tbody>
</table>
Consistency Models

- Strict Serializability
- Linearizability
- Sequential
- Causal+
- Eventual

Strength Scale:
- Stronger
- Weaker
Sequential Consistency

- **Total order**: there exists some legal total order of operations.
- **Preserves process ordering**: total order respects order of each process’s operations.
- **Difference from linearizability**?
  - Order of ops across processes not determined by real-time

**Pros**: Can allow more orderings than linearizability → better performance

**Cons**: Many possible sequential executions → increased application complexity
## Sequential Consistency Example

<table>
<thead>
<tr>
<th>Sequentially Consistent?</th>
<th>Yes</th>
<th>Sequentially Consistent?</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1: W(x)a</td>
<td></td>
<td>P1: W(x)a</td>
<td></td>
</tr>
<tr>
<td>P2: W(x)b</td>
<td></td>
<td>P2: W(x)b</td>
<td></td>
</tr>
<tr>
<td>P3: R(x)b R(x)a</td>
<td></td>
<td>P3: R(x)b R(x)a</td>
<td></td>
</tr>
<tr>
<td>P4: R(x)b R(x)a</td>
<td></td>
<td>P4: R(x)a R(x)b</td>
<td></td>
</tr>
</tbody>
</table>
Consistency Models

- Strict Serializability
- Linearizability
- Sequential
- Causal+
- Eventual

- Stronger
- Weaker
Causal+ Consistency

- **Partial order**: order causally related ops the same way across all processes
- **+**: replicas’ total order eventually converges.
- **Difference from sequential consistency?**
  - Only causally related ops need to be ordered: no guaranteed total order.
  - Concurrent ops may be ordered differently across different processes.

**Pros**: preserves causality while improving efficiency.

**Cons**: harder to reason about concurrency.
<table>
<thead>
<tr>
<th>Ops</th>
<th>Concurrent</th>
</tr>
</thead>
<tbody>
<tr>
<td>a,b</td>
<td>No</td>
</tr>
<tr>
<td>a,e</td>
<td>Yes</td>
</tr>
<tr>
<td>a,g</td>
<td>No</td>
</tr>
<tr>
<td>c,e</td>
<td>No</td>
</tr>
<tr>
<td>c,d</td>
<td>No</td>
</tr>
<tr>
<td>d,g</td>
<td>No</td>
</tr>
<tr>
<td>d,f</td>
<td>No</td>
</tr>
<tr>
<td>e,g</td>
<td>No</td>
</tr>
<tr>
<td>a,d</td>
<td>No</td>
</tr>
</tbody>
</table>

The diagram shows two processes, P1 and P2, with operations a, b, c, d, e, f, and g. The operations are connected with arrows indicating their order and dependencies. The table on the left shows whether certain operations can be executed concurrently (Yes) or not (No).
Causality Consistency Example

Causally Consistent? Yes
P1: \( W(x)a \)
P2: \( W(x)b \)
P3: \( R(x)b \) \( R(x)a \)
P4: \( R(x)a \)

Causally Consistent? No
P1: \( W(x)a \)
P2: \( R(x)a \) \( W(x)b \)
P3: \( R(x)b \) \( R(x)a \)
P4: \( R(x)a \)
Consistency Models

- Strict Serializability
- Linearizability
- Sequential
- Causal
- Eventual

Consistency models range from Stronger to Weaker.
Eventual Consistency

- **Eventual convergence**: If no more writes, all replicas eventually agree.
- **Difference from causal consistency?**
  - Does not preserve causal relationships
  - Is the “+” in causal+.
- Frequently used with application conflict resolution, anti-entropy

**Pros**: highly available; think Bayou.

**Cons**: no safety guarantees, need conflict resolution.
In a nutshell...

**Strict Serializability**: total order + real time guarantees over *transactions*

**Linearizability**: total order + real time guarantees over *operations*

**Sequential consistency**: total order + process order

**Causal+ consistency**: causally ordered + replicas eventually converge

**Eventual consistency**: eventually, everyone should agree on state
Exercise 1:

Consistency Model:

- Strictly Serializable: Yes
- Linearizable: Yes
- Sequential: Yes
- Causal+: Yes
- Eventual: Yes

P1: \{W(x) 1, W(y) 2\} \{R(y) 4\}

P2: \{W(x) 1, R(y) 4\}

P3: \{W(x) 0, W(y) 4\}

P4: \{R(x) 0\} \{R(x) 1\}
Exercise 2:

Consistency Model:
- Linearizable: Yes
- Sequential: Yes
- Causal+: Yes
- Eventual: Yes

<table>
<thead>
<tr>
<th>P1</th>
<th>W(x) 1</th>
<th>R(y) 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>R(x) 1</td>
<td>R(y) 4</td>
</tr>
<tr>
<td>P3</td>
<td>R(x) 1</td>
<td>W(y) 4</td>
</tr>
<tr>
<td>P4</td>
<td>R(x) 1</td>
<td>R(y) 4</td>
</tr>
</tbody>
</table>
# Exercise 3:

<table>
<thead>
<tr>
<th></th>
<th>W(x)</th>
<th>W(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1:</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>P2:</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>P3:</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>P4:</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>P5:</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

## Consistency Model:

<table>
<thead>
<tr>
<th></th>
<th>Linearizable</th>
<th>Sequential</th>
<th>Causal+</th>
<th>Eventual</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Exercise 4:

<table>
<thead>
<tr>
<th></th>
<th>W(x)</th>
<th>W(y)</th>
<th>R(x)</th>
<th>R(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>3</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>P5</td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Consistency Model:

- Linearizable: No
- Sequential: No
- Causal+: Yes
- Eventual: Yes
Exercise 5:

Consistency Model:
- Linearizable: No
- Sequential: No
- Causal+: Yes
- Eventual: Yes

P1: W(x) 1
P2: W(x) 3
P3: W(x) 7
P4: R(x) 3  R(x) 7  R(x) 1
P5: R(x) 3  R(x) 1  R(x) 7
Exercise 6:

Consistency Model:
- Linearizable: No
- Sequential: No
- Causal+: Yes
- Eventual: Yes

P1: \( W(x) \ 1 \)
P2: \( W(x) \ 3 \)
P3: \( R(x) \ 3 \quad W(x) \ 7 \) 
P4: \( R(x) \ 3 \quad R(x) \ 7 \quad R(x) \ 1 \) 
P5: \( R(x) \ 3 \quad R(x) \ 1 \quad R(x) \ 7 \)
Exercise 7:

Consistency Model:

- Linearizable: No
- Sequential: No
- Causal+: No
- Eventual: Yes

P1: \( W(x) \ 1 \)

P2: \( R(x) \ 1 \quad W(x) \ 3 \)

P3: \( R(x) \ 3 \quad W(x) \ 7 \)

P4: \( R(x) \ 3 \quad R(x) \ 7 \quad R(x) \ 1 \)

P5: \( R(x) \ 3 \quad R(x) \ 1 \quad R(x) \ 7 \)