Consistency Models

- Contract between a distributed system and the applications that run on it
- A consistency model is a set of guarantees made by the distributed system

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

Real-Time Ordering Examples

```
P_A |- w(x=1)
P_B |- w(x=2)
P_C |- w(x=3)
P_P |- w(x=4) |- w(x=5)
P_E |- w(x=6)
```
Linearizable?

\[
P_A \vdash w(x=1) \quad P_B \vdash w(x=2) \quad P_C \vdash w(x=3) \quad P_D \vdash w(x=4) \quad P_E \vdash w(x=6) \\
P_F \vdash r(x)=1 \quad r(x)=2 \quad r(x)=3 \quad r(x)=6 \quad r(x)=5 \quad \checkmark
\]

Linearizable?

\[
P_A \vdash w(x=1) \quad P_B \vdash w(x=2) \quad P_C \vdash w(x=3) \quad P_D \vdash w(x=4) \quad P_E \vdash w(x=6) \\
P_F \vdash r(x)=1 \quad r(x)=2 \quad r(x)=5 \quad r(x)=6 \quad r(x)=5 \quad \times
\]

Linearizable?

\[
P_A \vdash w(x=1) \quad P_B \vdash w(x=2) \quad P_C \vdash w(x=3) \quad P_D \vdash w(x=4) \quad P_E \vdash w(x=6) \\
P_F \vdash r(x)=1 \quad r(x)=2 \quad r(x)=3 \quad r(x)=6 \quad r(x)=5 \quad \checkmark
\]

Linearizable?

\[
P_A \vdash w(x=1) \quad P_B \vdash w(x=2) \quad P_C \vdash w(x=3) \quad P_D \vdash w(x=4) \quad P_E \vdash w(x=6) \\
P_F \vdash r(x)=1 \quad r(x)=2 \quad r(x)=5 \quad r(x)=6 \quad r(x)=3 \quad \times
\]
**Linearizability == “Appears to be a Single Machine”**

- Single machine processes requests one by one in the order it receives them
  - Will receive requests ordered by real-time in that order
  - Will receive all requests in some order

- Atomic Multicast, Viewstamped Replication, Paxos, and RAFT provide Linearizability

- Single machine processing incoming requests one at a time also provide Linearizability 😊

**Linearizability is Ideal?**

- Hides the complexity of the underlying distributed system from applications!
  - Easier to write applications
  - Easier to write correct applications

- But, performance trade-offs

**Stronger vs Weaker Consistency**

- Stronger consistency models
  + Easier to write applications
  - More guarantees for the system to ensure
    Results in performance tradeoffs

- Weaker consistency models
  - Harder to write applications
  + Fewer guarantees for the system to ensure

**Strictly Stronger Consistency**

- A consistency model $A$ is strictly stronger than $B$ if it allows a strict subset of the behaviors of $B$
  - Guarantees are strictly stronger
Sequential Consistency

- All replicas execute operations in some total order.
- That total order preserves the process ordering between operations.
  - If process P issues operation A before operation B, then A is order before B by the process order.
  - If operations A and B are done by different processes then there is no process order between them.
    - (But there must be some total order)

Linearizability is strictly stronger than Sequential Consistency

- Linearizability: \( \exists \) total order + real-time ordering
- Sequential: \( \exists \) total order + process ordering
  - Process ordering \( \subseteq \) Real-time ordering

Sequential Consistency ≈ “Appears to be a Single Machine”

- Single machine processes requests one by one in the order it receives them.
  - Will receive requests ordered by process order in that order.
  - Will receive all requests in some order.

Sequential But Not Linearizable

- Suppose Shopazon discovers a dump of passwords on the Internet from a hacking of an unrelated company.
- Shopazon notifies their users of this public release, and requires users to enter new passwords on checkout.
- Alice attempts to use it to make a large Shopazon purchase.
- She requests a password change, and the user account system recomputes the hash of her password.
- The user account system then sends a message to Alice’s account, and advises them to change their passwords.
- Alice responds to the user account storage system, and notifies Alice her password change request, writes a hash of the new password to the user account storage system, and notifies Alice that her password change request successfully completed.
- A few weeks later, Bob accesses Alice’s account.
- Bob’s read of the password for Alice’s account in Figure 1 would not return the old hash as shown, but instead return the new hash.

Definition 3.1

This is the intuition behind Lamport’s definition of strong consistency. Sequential consistency ignores real-time constraints on the allowable ordering of operations in concurrent systems. In order to ensure that the effects of operations are observed in a timely manner by all clients, we need a stronger notion of correctness, as discussed in Section 2, even the weakest of linearizable objects (shared registers) require stronger than sequential consistency, that imposes additional constraints on the allowable ordering of operations in ordered parallel systems. Programmers can more naturally reason about sequential systems, so complex logic can be implemented without reasoning about the exponentially many interleavings of concurrent operations.

Definition 3.2

Linearizability is the order specified by its program. In a linearizable model, we say that a consistency model is a way to reconcile these tradeoffs in order to provide strongly consistent operation primitives, such as read-modify-write, cannot be implemented solely by composing registers [24].

Linearizability is slow and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consensus protocols in that strong synchronizability and write performance, shared registers are fundamentally weaker than consistency models, strictly stronger than sequential consistency, that imposes additional constraints on the allowable ordering of operations in ordered parallel systems. Programmers can more naturally reason about sequential systems, so complex logic can be implemented without reasoning about the exponentially many interleavings of concurrent operations.
Consistency Hierarchy

- **Linearizability**
  - e.g., RAFT
- **Sequential Consistency**
- **Causal+ Consistency**
  - e.g., Bayou
- **Eventual Consistency**
  - e.g., Dynamo

Causal+ Consistency

- Partially orders all operations, does not totally order them
  - Does not look like a single machine
- ** Guarantees**
  - For each process, \( \exists \) an order of all writes + that process's reads
  - Order respects the happens-before (\( \rightarrow \)) ordering of operations
  - + replicas converge to the same state
    - Skip details, makes it stronger than eventual consistency

Causal Consistency

1. Writes that are potentially causally related must be seen by all processes in same order.
2. Concurrent writes may be seen in a different order on different processes.
   - Concurrent: Ops not causally related

Causal Consistency

1. Writes that are potentially causally related must be seen by all processes in same order.
2. Concurrent writes may be seen in a different order on different processes.
   - Concurrent: Ops not causally related
Causal Consistency

<table>
<thead>
<tr>
<th>Operations</th>
<th>Concurrent?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, b</td>
<td></td>
</tr>
<tr>
<td>b, f</td>
<td></td>
</tr>
<tr>
<td>c, f</td>
<td></td>
</tr>
<tr>
<td>e, f</td>
<td></td>
</tr>
<tr>
<td>e, g</td>
<td></td>
</tr>
<tr>
<td>a, c</td>
<td></td>
</tr>
<tr>
<td>a, e</td>
<td></td>
</tr>
</tbody>
</table>

Physical time ↓

Causal Consistency

<table>
<thead>
<tr>
<th>Operations</th>
<th>Concurrent?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, b</td>
<td>N</td>
</tr>
<tr>
<td>b, f</td>
<td>Y</td>
</tr>
<tr>
<td>c, f</td>
<td>Y</td>
</tr>
<tr>
<td>e, f</td>
<td>Y</td>
</tr>
<tr>
<td>e, g</td>
<td>N</td>
</tr>
<tr>
<td>a, c</td>
<td>Y</td>
</tr>
<tr>
<td>a, e</td>
<td>N</td>
</tr>
</tbody>
</table>

Physical time ↓

Causal+ But Not Sequential

| PA | w(x=1) | r(y)=0 |
| PB | w(y=1) | r(x)=0 |

√ Casual+

Happens Before Order

w(x=1) → r(y)=0
w(y=1) → r(x)=0

PA Order: w(x=1), r(y)=0, w(y=1)
PB Order: w(y=1), r(x)=0, w(x=1)

X Sequential

Process Ordering

w(x=1) → r(y)=0
w(y=1) → r(x)=0

No Total Order

w(x=1) → r(y)=0
w(y=1) → r(x)=0

Eventual But Not Causal+

| PA | w(x=1) | w(y)=1 |
| PB |       |       |

√ Eventual

As long as PB eventually would see r(x)=1 this is fine

Happens Before Ordering

w(x=1) → w(y)=1
r(y)=1 → r(x)=0

No Order for PA

w(x=1) → w(y)=1
r(y)=1 → r(x)=0

X Causal+
**Consistency Hierarchy**

- Linearizability: e.g., RAFT
- Sequential Consistency
- Causal+ Consistency: e.g., Bayou
- Eventual Consistency: e.g., Dynamo

---

**Causal Consistency: Quiz**

- $P_A$  $\vdash$  $w(x=1)$
- $P_A$  $\vdash$  $w(x=3)$
- $P_B$  $\vdash$  $r(x)=1$ $\leftarrow$  $r(x)=2$
- $P_B$  $\vdash$  $w(x=2)$
- $P_C$  $\vdash$  $r(x)=3$
- $P_C$  $\vdash$  $r(x)=2$
- $P_D$  $\vdash$  $r(x)=3$
- $P_D$  $\vdash$  $r(x)=2$

- Valid under causal consistency
- **Why?** $x=3$ and $x=2$ are concurrent
  - So all processes don’t (need to) see them in same order
- $P_C$ and $P_D$ read the values ‘1’ and ‘2’ in order as potentially causally related. No ‘causality’ for ‘3’.

---

**Sequential Consistency: Quiz**

- $P_A$  $\vdash$  $w(x=1)$
- $P_B$  $\vdash$  $r(x)=1$ $\leftarrow$  $w(x=2)$
- $P_B$  $\vdash$  $w(x=3)$
- $P_C$  $\vdash$  $r(x)=3$
- $P_C$  $\vdash$  $r(x)=2$
- $P_D$  $\vdash$  $r(x)=3$
- $P_D$  $\vdash$  $r(x)=2$

- Invalid under sequential consistency
- **Why?** $P_C$ and $P_D$ see 2 and 3 in different order
- But fine for causal consistency
  - 2 and 3 are not causally related
Causal Consistency

\[ \begin{align*}
P_A & \models w(x=1) \\
P_B & \models r(x)=1 \models w(x=2) \\
P_C & \models r(x)=2 \\
P_D & \models r(x)=1 \\
\end{align*} \]

\[ \begin{align*}
X \quad & x=2 \text{ happens after } x=1 \\
\end{align*} \]

Causal Consistency

\[ \begin{align*}
P_A & \models w(x=a) \\
P_B & \models w(x=b) \\
P_C & \models r(x)=b \\
P_D & \models r(x)=a \\
\end{align*} \]

\[ \begin{align*}
\checkmark \quad & P_B \text{ doesn’t read value of 1 before writing 2} \\
\end{align*} \]