Day 11: Ordering, Snapshots
( & Some Replication)
Today

1. Ordering / Time
2. Distributed System Guarantees
3. Replication Intro
4. Global Snap
a change to one copy must be reflected at all copies!!

these servers are called Replica Managers (RM)
How are these events processed by all replicas?
Total Ordering (Passive Replication)

1. One of the RMs is elected the leader
2. All requests are redirected to the leader who gives each event a unique ID
3. The leader replicates the event with their IDs to the other RMs
Total ordering II (Active Replication)

1) Each client assigns ID to events
2) Each client broadcasts to servers

Because IDs are local. There will be duplicates. Similar to logical clock. How do you break ties?
Breaking Ties

1. IP Address
2. Time
3. Random function (with fixed seed)

<table>
<thead>
<tr>
<th>Event</th>
<th>Clock</th>
<th>Client IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1, 1</td>
<td>1.1.1.1</td>
<td></td>
</tr>
<tr>
<td>e2, 2</td>
<td>1.1.1.1</td>
<td></td>
</tr>
<tr>
<td>e3, 1</td>
<td>1.1.1.2</td>
<td></td>
</tr>
<tr>
<td>e6, 2</td>
<td>1.1.1.2</td>
<td></td>
</tr>
</tbody>
</table>
all ordering

FIFO

VC

LC

Total

VC \leq LC \leq FIFO

based on a vector of local clocks

order based on local clocks

order based on logical clocks which are local clocks with a propagation element
Total ordering is not based on a local clock. The key requirement is that all servers agree. LC, VC & FIFO can be made to be total orders by adding a deterministic tie breaker.

However, some total ordering are not LC, VC, or FIFO.
Storage System Guarantees-based on ordering constraints

Linearizability = FIFO + Total + real time

Sequential = FIFO + Total
System Background

During a restart everything in memory is deleted.
Why do we need Snapshots?

1. **Fault Tolerance**
   - Each node processes a sequence of events to build state in memory.
   - After a failure, need to reprocess all events to build state. Snapshots speed up this process. You only need to replay events after a snapshot.

[Diagram]

\[ e_1 \ldots e_{100} \ldots e_{110} \]

- **Check Point**
- **Crash**

With checkpoint you only need to replay \( e_{100} \ldots e_{110} \) to restore the system.
Why do we need Snapshots?

1. Fault tolerance
   Each node processes a sequence of events to build state in memory. After a failure, need to reprocess all events to build state. Snapshots speed up this process. You only need to replay events after a snapshot.

2. To debug systems
   During debugging, you may need to examine previous variables.
How to capture a global snapshot?

1. Coordinate between all servers to capture snapshot at a predefined time.

2. Each server periodically captures snapshots and use VC to determine a consistent snapshot across servers.