CISC 434: Consistency and Replication - II

- Data centric consistency models
- Client-centric consistency models
- Replica management
Client-Centric Consistency

- Concentrate on the consistency that an individual client should be offered.
- If a client modifies a replica:
  - the differences should be made transparent;
  - that replica changes are propagated with the data modified by that client earlier on this or other replicas.
Client-Centric Consistency

Various Client-Centric Consistency Models

- Monotonic reads
- Monotonic writes
- Read your writes
- Writes follow reads

Some notations

- \( X_i \) – denote the version of the data item \( x \) at local copy \( L_i \)
- \( WS(x_i) \) – series of write operations taken place at \( L_i \) since initialization that resulted \( x_i \)
- \( WS(x_i;x_j) \) – operations in \( WS(x_i) \) have also been performed at local copy \( L_j \) at a later time
Monotonic Reads

- If a process reads the value of a data item $x$, any successive read operation on $x$ by that process will always return that same value or a more recent value.

- Example – if email is opened from a different location later on, the system will also contain the previous emails.

\[
\begin{align*}
    &\text{L1: } \text{WS}(x_1) \quad \text{R}(x_1) \\
    &\text{L2: } \text{WS}(x_1;x_2) \quad \text{R}(x_2)
\end{align*}
\]

- The read operations performed by a single process $P$ at two different local copies of the same data store:
  a) A monotonic-read consistent data store.
b) A data store that does not provide monotonic reads (no guarantees are given that the read set at L2 contains all operations contained in WS(x1))
Monotonic Writes

- A write operation by a process on a data item \( x \) is completed before any successive write operation on \( x \) by the same process.

\[
\begin{align*}
L1: \quad W(x_1) & \quad W(x_2) \\
L2: \quad WS(x_1) & \quad WS(x_2)
\end{align*}
\]

- The write operations performed by a single process \( P \) at two different local copies of the same data store:
  a) A monotonic-write consistent data store.
b) A data store that does not provide monotonic-write consistency (since propagation of $W(x_1)$ to copy $L_2$ is missing, there is no guarantee that the copy of $x$ on which the second write is being performed, has the same or more recent value at the time $W(x_1)$ completed at $L_1$.
Read Your Writes

- The effect of a write operation by a process on data item x will always be seen by a successful read operation on x by the same process

L1: \[ W(x_1) \]

L2: \[ WS(x_1; x_2) \]

(a) A data store that provides read-your-writes consistency
Read Your Writes – contd.

a) A data store that does not

\[ \text{L1: } W(x_1) \quad \text{-----} \quad \text{L2: } WS(x_2) \quad \text{----} \quad R(x_2) \]

(b)
A write operation by a process on a data item $x$ following a previous read operation on $x$ by the same process, is guaranteed to take place on the same or a more recent value of $x$ that was read.

A writes-follow-reads consistent data store

\[
\begin{align*}
L1: & \quad WS(x_1) \quad R(x_1) \\
L2: & \quad WS(x_1; x_2) \quad W(x_2)
\end{align*}
\]

a) A writes-follow-reads consistent data store
A data store that does not provide writes-follow-reads consistency (no guarantees are given that the operation performed at L2 are performed on a copy that is consistent with the one just read at L1)

\[ \begin{align*}
\text{L1:} & \quad \text{WS}(x_1) \quad \text{R}(x_1) \\
\text{L2:} & \quad \text{WS}(x_2) \quad \text{W}(x_2)
\end{align*} \]
Replica Management

Where, when, and by whom replicas should be placed, and which mechanisms to use for their consistency

- Content replication and placement
- Content distribution
**Content Replication and Placement**

- **Replica-server placement**
  - Based on clients’ distance - select one server at a time so that the average distance between that server and its clients is minimal.
  - Based on topology - only place the server in each network in which all the nodes run the same routing protocol and managed by a single organization.
Content Replication and Placement

The logical organization of different kinds of copies of a data store
Content Replication and Placement

- **Permanent replicas**
  - Initial set of replicas that form a distributed data store
  - Examples: round-robin strategy of handling requests to the servers, mirroring

- **Server-initiated replicas**
  - Copying of data store to enhance performance
  - Initiated by the owner (server) of the data (store)
  - Dynamic replacement – to reduce the load on a server or to bring the files closer to the clients (request those files more)
Content Replication and Placement

- Client-initiated replicas (client caches)
  - A client locally stores a copy of the data it has just requested – the data store (provider) has nothing to do with this caching

- Which one is better?
Replica Management - Content Distribution

- **State versus operations**
  - Propagate only a notification of an update
  - Transfer data from one copy to another
  - Propagate the update operation to other copies

- **Unicasting versus multicasting**
  - Sending separate update messages to each of the N servers
  - The network sends message to a subset of receivers
  - How to combine multi-casting with a push-based approach?
Pull versus Push Protocols

- **Push-Based protocols (server-based protocols)**
  - Updates are propagated to other replicas without those replicas even asking for the updates
  - Useful for high degree of consistency

- **Pull-Based protocols (client-based protocols)**
  - Often used by client caches
  - Efficient when read-to-update ratio is low and client caches are not shared
## Pull versus Push Protocols

A comparison in the case of multiple clients, single server systems

<table>
<thead>
<tr>
<th>Issue</th>
<th>Push-based (server-based)</th>
<th>Pull-based (client-based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>State of server</td>
<td>List of client replicas and caches</td>
<td>None</td>
</tr>
<tr>
<td>Messages sent</td>
<td>Update (and possibly fetch update later)</td>
<td>Poll and update</td>
</tr>
<tr>
<td>Response time at client</td>
<td>Immediate (or fetch-update time)</td>
<td>Fetch-update time</td>
</tr>
</tbody>
</table>


Summary

Client-centric consistency
- Monotonic reads
- Monotonic writes
- Read your writes
- Writes follow reads

Replica management
- Content replication and placement
- Content distribution