Geo-replicated services: Latency vs. Consistency

Single-site with strong consistency
- Natural semantics but high latency
- Bad user experience for remote clients

Multi-site with eventual consistency
- Low latency but undesirable behaviors
- E.g., risk selling the last car to two distinct users

Challenge: get both fast and consistent in a single system

1. **RedBlue Consistency**
   - **Blue operations**
     - Eventual consistency, partial order \(\Rightarrow\) FAST
   - **Red operations**
     - Strong consistency, total order \(\Rightarrow\) CONSISTENT

Example of admissible causal serializations:
Bob
\[ B_1 \rightarrow A_1 \rightarrow R_1 \rightarrow B_2 \rightarrow A_2 \rightarrow R_2 \rightarrow R_3 \]
Alice
\[ A_1 \rightarrow B_1 \rightarrow R_1 \rightarrow R_2 \rightarrow B_2 \rightarrow R_3 \]

2. **Challenge: increase fraction of Blue operations**
A bank example: shared account, initial balance=100, interest=0.05

Bob in EU
\[ 100 \rightarrow deposit(20) : +20 \rightarrow accrueinterest() : +5 \rightarrow 125 \]
Alice in US
\[ 100 \rightarrow deposit(20) : +20 \rightarrow accrueinterest() : +5 \rightarrow 125 \]

Solution: split operations into generator and shadow operations
Bob in EU
\[ 100 \rightarrow deposit(20) : +20 \rightarrow 120 \rightarrow accrueinterest() : +5 \rightarrow 125 \]
Alice in US
\[ 100 \rightarrow deposit(20) : +20 \rightarrow 120 \rightarrow accrueinterest() : +5 \rightarrow 125 \]

Generator decides what changes to make, shadow makes changes.

3. **What color should each operation be?**

Ensuring state convergence
- Commutes with all others?
  - Yes → Red
  - No → Blue

Ensuring invariant preservation
- a shadow operation \(u\)
  - Commutes with all others?
    - Yes → Red
    - No → Blue

4. **How does the system works?**

5. **Evaluation and results**
   - Case studies: TPC-W, RUBiS, Quoddy

<table>
<thead>
<tr>
<th>App/Workload</th>
<th>Originally</th>
<th>With shadow ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Blue (%)</td>
<td>Red (%)</td>
</tr>
<tr>
<td>TPC-W/Browsing mix</td>
<td>96.0</td>
<td>4.0</td>
</tr>
<tr>
<td>TPC-W/Shopping mix</td>
<td>85.0</td>
<td>15.0</td>
</tr>
<tr>
<td>TPC-W/Ordering mix</td>
<td>63.0</td>
<td>37.0</td>
</tr>
<tr>
<td>RUBiS/Bidding mix</td>
<td>85.0</td>
<td>15.0</td>
</tr>
<tr>
<td>Quoddy/15% updates</td>
<td>85.0</td>
<td>15.0</td>
</tr>
</tbody>
</table>

   The fraction of Red operations is dramatically reduced!
   - Experiments deployed across 5 sites in Amazon EC2

6. **Concluding remarks**
   - RedBlue consistency allows strong and eventual consistency to coexist
   - Generator/shadow split increases space of blue operations
   - Coloring rules allow for fast as possible, consistent when necessary
   - Experimental results show improved latency and throughput