Fine-grained consistency for geo-replicated systems

Cheng Li, Nuno Preguica, Rodrigo Rodrigues

University of Science and Technology of China
NOVA LINCS & FCT, Univ. NOVA de Lisboa
INESC-ID & Instituto Superior Técnico, Universidade de Lisboa
Unprecedented growth in Internet services

• As of June 2017, Facebook has 2 billion monthly active users.

Facebook Subscribers in the World by Regions - June 2016

Source: Internet World Stats - www.internetworldstats.com/facebook.htm
Base: 1,879,433,530 Internet users on June 30, 2018
Copyright © 2016, Minnatts Marketing Group
Geo-users demand instant responses

<table>
<thead>
<tr>
<th>Distinct Queries/User</th>
<th>Query Refinement</th>
<th>Revenue/User</th>
<th>Any Clicks</th>
<th>Satisfaction</th>
<th>Time to Click (increase in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50ms</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>200ms</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-0.3%</td>
<td>-0.4%</td>
</tr>
<tr>
<td>500ms</td>
<td>-</td>
<td>-0.6%</td>
<td>-1.2%</td>
<td>-1.0%</td>
<td>-0.9%</td>
</tr>
<tr>
<td>1000ms</td>
<td>-0.7%</td>
<td>-0.9%</td>
<td>-2.8%</td>
<td>-1.9%</td>
<td>-1.6%</td>
</tr>
<tr>
<td>2000ms</td>
<td>-1.8%</td>
<td>-2.1%</td>
<td>-4.3%</td>
<td>-4.4%</td>
<td>-3.8%</td>
</tr>
</tbody>
</table>

Google

- Strong negative impact of delay on user activities [1]
- Google counts site speed as a ranking factor [2].

Geo-Replication helps

- **Performance**: local reads
- **Availability**: data still available unless all replicas fail or become unreachable
- **Scalability**: load balance across sites for reads
Fundamental trade-offs

**Strong consistency (SC)**
- e.g., Paxos [TOCS'98]
- ✔️ State convergence
- ✔️ Invariant preservation
- ❌ High latency
- ❌ Low throughput

**Eventual consistency (EC)**
- e.g., Dynamo [SOSP'07]
- ✔️ Low latency
- ✔️ High throughput
- ❌ State divergence
- ❌ Invariant violation
Our prior work

**RedBlue Consistency** [OSDI’12, ATC’14] allows operations to be executed under either strong or eventual consistency.

- **Strong consistency (SC)**
  - e.g., Paxos [TOCS’98]
  - ✔️ State convergence
  - ✔️ Invariant preservation

- **Eventual consistency (EC)**
  - e.g., Dynamo [SOSP’07]
  - ✔️ Low latency
  - ✔️ High throughput

Coarse-grained classification may add unnecessary coordination!
Consistency spectrum

- Too many consistency models, some of which have subtle differences
- Need a unified consistency framework to capture all these semantics
Outline

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. Evaluation and results
5. Conclusion
Geo-replicated auction service

<table>
<thead>
<tr>
<th>winner</th>
<th>bidder</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bob</td>
<td>10</td>
</tr>
</tbody>
</table>

US

UK

bid($10)
Geo-replicated auction service

<table>
<thead>
<tr>
<th>winner</th>
<th>bidder</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bob</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>winner</th>
<th>bidder</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alice</td>
<td>15</td>
</tr>
</tbody>
</table>

US

UK

bid($15)
Geo-replicated auction service

<table>
<thead>
<tr>
<th>winner</th>
<th>bidder</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>Bob</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>winner</th>
<th>bidder</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

US

UK

Close()
Bob won even with a lower bid than Alice.
Fine-grained coordination

Less coordination
Visibility restrictions

• A restriction between two operations implies that one must see effects introduced by the other.

• For operation $a, b$, the restriction $r(a, b)$ implies that $a < b \lor b < a$ w.r.t any partial order $\prec$.

If $a < b \lor b < a$, then $r(a, b)$ is met in $\prec$. 
Partial order-restrictions (PoR) Consistency

• A geo-replicated system $S$ is associated with a set of restrictions $Rs$.

• $S$ is **PoR Consistent** if, for any its executions, there exists an admissible partial order, where all restrictions in $Rs$ are met.
Partial order-restrictions (PoR) Consistency

- A geo-replicated system $S$ is associated with a set of restrictions $Rs$.
- $S$ is PoR Consistent if, for any of its executions, there exists an admissible partial order, where all restrictions in $Rs$ are met.

- Fewer restrictions → Weaker consistency

Tunable (parameterized) consistency model
Partial order-restrictions (PoR) Consistency

- A geo-replicated system $S$ is associated with a set of restrictions $Rs$.
- $S$ is **PoR Consistent** if, for any its executions, there exists an admissible partial order, where all restrictions in $Rs$ are met.

### Tunable (parameterized) consistency model

- **Causal consistency**
  $Rs = \{\}$

- **RedBlue consistency**
  $Rs = \{r(a,b) \mid a, b \text{ are red operations}\}$

- **Serializable**
  $Rs = \{r(a,b) \mid \text{for any pair of operations } a, b\}$
Challenges of adopting PoR

• What are the set of restrictions to be added?
  • They must ensure relevant properties, e.g., state convergence, invariant preservation.

• Is the set of added restrictions minimal?
  • i.e., no unnecessary coordination
State convergence

• If all replicas execute the same set of operations then they reach the same state
• Must place a restriction over any pair of non-commuting operations
• Consider a geo-replicated bank example

```c
deposit(float m) {
    balance = balance + m;
}

accrueinterest() {
    float delta = balance * interest;
    balance = balance + delta;
}
```
Invariant preservation

- Insight: for any violation, add restrictions among a minimal set of concurrent conflicting operations
  - i.e., removing any conflicting op, violation disappears
  - named as “I-conflict set”
Invariant preservation

Invariant: \( \exists \text{ winner} \rightarrow \text{winner. bid is highest in bidTable} \)

\[
\begin{align*}
\text{bid}(\text{uId}, \text{bid}) & \quad \text{close}(\text{wId})
\end{align*}
\]
Invariant preservation

Invariant: \( \exists \) winner \( \rightarrow \) winner. bid is highest in bidTable

\[
\{ \text{auction is Open} \} \quad \text{bid}(uId, bid) \quad \{ \text{bidTable} = \text{bidTable} \cup \{< uId, bid >\} \}
\]

\[
\{ \exists w \in \text{bidTable}.w.uId = wId \land w.\text{bid is highest} \} \quad \text{close}(wId) \quad \{ \text{winner} = wId \land \text{auction is Closed} \}
\]

Weakest precondition

Postcondition
Invariant preservation

Invariant: $\exists \text{winner} \rightarrow \text{winner. bid is highest in bidTable}$

\[
\begin{align*}
\{\text{auction is Open}\} & \quad \{\exists w \in \text{bidTable}. w.\text{ulId} = \text{wId} \land w.\text{bid is highest}\} \\
\text{bid(ulId, bid)} & \quad \text{close(wId)} \\
\{\text{bidTable} = \text{bidTable} \cup \{\text{ulId, bid}\}\} & \quad \{\text{winner} = \text{wId} \land \text{auction is Closed}\}
\end{align*}
\]

- \{close, bid\} is an “I-conflict set”.
- The restriction $r\{close, bid\}$ must be enforced!
Outline

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. Evaluation and results
5. Conclusion
Olisipo - Design rationale

Give a restriction $r(a, b)$

- Workload 1: $a$ and $b$ have the same prevalence

- Workload 2: $a$ occurs more often than $b$
Olisipo - Design rationale

Give a restriction $r(a, b)$

- Workload 1: $a$ and $b$ have the same prevalence

Symmetry protocol: Every $a$ ($b$) instance acquires a permission from a centralized server w.r.t all concurrent $b$ ($a$) instances.

- Workload 2: $a$ occurs more often than $b$
Olisipo - Design rationale

Give a restriction $r(a, b)$

- Workload 1: $a$ and $b$ have the same prevalence

Symmetry protocol: Every $a$ ($b$) instance acquires a permission from a centralized server w.r.t all concurrent $b$ ($a$) instances.

- Workload 2: $a$ occurs more often than $b$

Asymmetry protocol: Every $b$ instance acts as a global barrier w.r.t all concurrent $a$ instances.
Olisipo - Overview

<table>
<thead>
<tr>
<th>Op</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
</tr>
</tbody>
</table>

Geo-replicated storage system
Olisipo - Overview

Which operations that a must see?

\[ B = \{b_1, b_2, b_3\} \]

Execute \( a \) until \( B \) locally replicated

<table>
<thead>
<tr>
<th>Op</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( b )</td>
</tr>
<tr>
<td>( b )</td>
<td>( a )</td>
</tr>
</tbody>
</table>

Geo-replicated storage system
Olisipo - Overview

Which operations that \( a \) must see?

\[ B = \{b1, b2, b3\} \]

Replicate \( a \) among \( n \) replicas

Execute \( a \) until \( B \) locally replicated

<table>
<thead>
<tr>
<th>Op</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( b )</td>
</tr>
<tr>
<td>( b )</td>
<td>( a )</td>
</tr>
</tbody>
</table>

Replicate \( a \)

Geo-replicated storage system
Olisipo - Overview

Which operations that $a$ must see?

$$B = \{b_1, b_2, b_3\}$$

Replicate $a$ among $n$ replicas

The effects of $a$ is persistent!

Execute $a$ until $B$ locally replicated

<table>
<thead>
<tr>
<th>Op</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$b$</td>
</tr>
<tr>
<td>$b$</td>
<td>$a$</td>
</tr>
</tbody>
</table>

Replicate $a$

Geo-replicated storage system
Outline

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. Evaluation and results
5. Conclusion
Case study

RUBiS

- An e-commerce benchmark that emulates an auction site
- 3 invariants corresponding to 3 I-conflict sets
  - \{\text{registerUser}', \text{registerUser}'\}
  - \{\text{storeBuyNow}', \text{storeBuyNow}'\}
  - \{\text{placeBid}', \text{closeAuction}'\}

<table>
<thead>
<tr>
<th>RedBlue consistency</th>
<th>PoR consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 restrictions</td>
<td>3 restrictions</td>
</tr>
</tbody>
</table>

PoR consistency places fewer restrictions than RedBlue!
Experimental setup

• Replicating RUBiS across three regions in EC2 platform
  • EU-FRA, US-EAST, US-WEST

• Baselines:
  • Unreplicated RUBiS offering strong consistency
  • Three-region RUBiS replication under RedBlue consistency

• Questions to answer:
  • User observed latency improvement
  • Peak throughput improvement
  • Performance impact when choosing different coordination policy
Latency and throughput improvement

User observed latency (lower is better)

Peak throughput (higher is better)
Choosing different coordination policies

Improper choice leads to performance penalty

Proper choice makes latency for requests demanding coordination as local access
Outline

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. Evaluation and results
5. Conclusion
Conclusion

• Fundamental tension between performance and consistency

• PoR consistency maps consistency semantics to a minimal set of visibility restrictions over a pair of operations.

• Olisipo enforces all restrictions throughout all executions of a geo-replicated system.

• Results show that PoR consistency places fewer restrictions and achieves better performance than RedBlue consistency.
Fine-grained consistency for geo-replicated systems

Cheng Li, Nuno Preguica, Rodrigo Rodrigues

Thanks for your attention!