Geo-Replicated State Machines Using PBFT

Tolerate up to $f$ Byzantine faults

Replicated key-value store

Send request to leader

Clients and replicas distributed worldwide


Eischer et al. Resilient Cloud-based Replication with Low Latency
Geo-Replicated State Machines Using PBFT

Tolerate up to $f$ Byzantine faults

Leader runs Byzantine agreement via WAN links

Clients and replicas distributed worldwide

Replicated key-value store

Send request to leader

Reply to client

Latency depends on leader location

High latency due to many WAN communication steps


Eischer et al. Resilient Cloud-based Replication with Low Latency
Hierarchical Geo-Replicated State Machines Using Steward


Eischer et al. Resilient Cloud-based Replication with Low Latency

Generate and distribute cluster-local decisions

Request to local cluster

Generate and distribute cluster-local decisions

Request to local cluster


Eischer et al. Resilient Cloud-based Replication with Low Latency
Hierarchical Geo-Replicated State Machines Using Steward


Challenges

Need for a replication protocol that provides

- **Efficiency**: No complex protocols over wide-area links
- **Modularity**: Allow integrating with different consensus protocols
- **Adaptability**: Add and remove new locations
Our Approach: SPIDER

**SPIDER: Architecture**

- Agreement group
- Execution groups
- Inter-regional message channel
- Cloud region
- Availability zones
- Execution groups

Eischer et al. Resilient Cloud-based Replication with Low Latency
Inter-Regional Message Channel (IRMC) - Message Forwarding

Channel abstraction

Subchannel A

10 11 12 13 14 ... capacity

fr + 1 highest Can request

window move

Sender $S_1$
(Sender $S_2$
(Sender $S_3$

Distributed queue

Multiple FIFO subchannels with position indices

Receiver $R_1$

Receiver $R_2$

Receiver $R_3$

Receiver $R_4$

Eischer et al. Resilient Cloud-based Replication with Low Latency
Inter-Regional Message Channel (IRMC) - Flow Control

Channel abstraction
Subchannel A
10 11 12 13 14 15 16 17 18 19

Limited capacity
Can request window move

Eischer et al. Resilient Cloud-based Replication with Low Latency

Channel abstraction
Subchannel B
85 86 87 88 89 90 91 92 93 94

Flow-control window

Multiple FIFO subchannels with position indices

Eischer et al. Resilient Cloud-based Replication with Low Latency

Inter-Regional Message Channel (IRMC) - Message Forwarding

send(A, 17, M) f + 1 send calls

Channel abstraction
Subchannel A
10 11 12 13 14 15 16 17 18 19

Receiver R1
Receiver R2
Receiver R3
Receiver R4

Faulty message
Distributed queue

Receiver R1
Receiver R2
Receiver R3
Receiver R4

Senders S1, S2, S3

Eischer et al. Resilient Cloud-based Replication with Low Latency

Inter-Regional Message Channel (IRMC) - Flow Control

Flow-control window

Limited capacity
Can request window move

Eischer et al. Resilient Cloud-based Replication with Low Latency

Senders S1, S2, S3

Eischer et al. Resilient Cloud-based Replication with Low Latency

SPIDER - Request Processing

C

E
E

A

E
E

SPIDER - Request Processing

Preprocessing

Agreement

Weakly consistent results

Use client-specific subchannel

Only short-distance communication

Use agreement as black-box
**SPIDER - Request Processing**

- **Preprocessing**
- **Agreement**
- **Request channel**
- **Commit channel**
- **Execution**

- Use client-specific subchannel
- Use agreement as black-box
- Only short-distance communication

- Strongly consistent results

---

**SPIDER - Garbage Collection**

- **Preprocessing**
- **Request channel**
- **Commit channel**
- **Execution**

- Use client-specific subchannel
- Use agreement as black-box
- Only short-distance communication

- Create checkpoints in regular intervals

---

Eischer et al. Resilient Cloud-based Replication with Low Latency
**SPIDER - Garbage Collection**

- Create checkpoints in regular intervals
- Coordinate via channel flow-control

**SPIDER - Adding a New Region**

- Preprocessing
- Request channel
- Agreement
- Execution
- New region
- Setup channels
- Retrieve current checkpoint

---

**Resilient Cloud-based Replication with Low Latency**

Eischer et al.
Evaluation

- Replicas in 4 AWS EC2 regions: Virginia, Oregon, Ireland, Tokyo
- 50 clients per region
- BFT: PBFT with 1 replica per region
- HFT: Steward with 4 replicas as cluster in each region
- SPIDER: 4 agreement replicas in Virginia, 3 replicas per execution group per region

Latency depends on client location. Short roundtrip times to Ohio and Ireland in agreement group region. Best latency for clients in Ohio 177ms and 132ms. SPIDER only has to wait for single WAN roundtrip to agreement. Latency varies with leader location. Stable response times.

Eischer et al. Resilient Cloud-based Replication with Low Latency
Eischer et al. Resilient Cloud-based Replication with Low Latency

- Replicas in 4 AWS EC2 regions: Virginia, Oregon, Ireland, Tokyo
- 50 clients per region
- BFT: PBFT with 1 replica per region
- HFT: Steward with 4 replicas as cluster in each region
- SPIDER: 4 agreement replicas in Virginia, 3 replicas per execution group per region

**Response time [ms]**

<table>
<thead>
<tr>
<th>Leader site</th>
<th>Best Latency for clients in Ohio</th>
<th>Latency varies with leader location</th>
<th>Stable response times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leader in V</td>
<td>177ms</td>
<td>132ms</td>
<td>89ms</td>
</tr>
<tr>
<td>Leader in O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Eicher et al. Resilient Cloud-based Replication with Low Latency

- Replicas in 4 AWS EC2 regions: Virginia, Oregon, Ireland, Tokyo
- 50 clients per region
- BFT: PBFT with 1 replica per region
- HFT: Steward with 4 replicas as cluster in each region
- SPIDER: 4 agreement replicas in Virginia, 3 replicas per execution group per region

**Response time [ms]**

<table>
<thead>
<tr>
<th>Leader site</th>
<th>Best Latency for clients in Ohio</th>
<th>Latency varies with leader location</th>
<th>Stable response times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leader in V</td>
<td>177ms</td>
<td>132ms</td>
<td>89ms</td>
</tr>
<tr>
<td>Leader in O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Eischer et al. Resilient Cloud-based Replication with Low Latency

- Replicas in 4 AWS EC2 regions: Virginia, Oregon, Ireland, Tokyo
- 50 clients per region
- BFT: PBFT with 1 replica per region
- HFT: Steward with 4 replicas as cluster in each region
- SPIDER: 4 agreement replicas in Virginia, 3 replicas per execution group per region

**Response time [ms]**

<table>
<thead>
<tr>
<th>Leader site</th>
<th>Best Latency for clients in Ohio</th>
<th>Latency varies with leader location</th>
<th>Stable response times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leader in V</td>
<td>177ms</td>
<td>132ms</td>
<td>89ms</td>
</tr>
<tr>
<td>Leader in O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Eischer et al. Resilient Cloud-based Replication with Low Latency

- Replicas in 4 AWS EC2 regions: Virginia, Oregon, Ireland, Tokyo
- 50 clients per region
- BFT: PBFT with 1 replica per region
- HFT: Steward with 4 replicas as cluster in each region
- SPIDER: 4 agreement replicas in Virginia, 3 replicas per execution group per region

**Response time [ms]**

<table>
<thead>
<tr>
<th>Leader site</th>
<th>Best Latency for clients in Ohio</th>
<th>Latency varies with leader location</th>
<th>Stable response times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leader in V</td>
<td>177ms</td>
<td>132ms</td>
<td>89ms</td>
</tr>
<tr>
<td>Leader in O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leader in T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Summary

### Problem
- Performance depends on leader location
- Either high latency or high complexity
- Best replica locations depend on client locations

### SPIDER
- Efficient: IRMCs forward group decisions
- Modular: Decoupled agreement and execution groups
- Adaptable: Add or remove execution groups at runtime

### More details in the paper
- Different possible Inter-Regional Message Channel (IRMC) implementations
- Handling malicious clients and other attacks

---

Eischer et al.  Resilient Cloud-based Replication with Low Latency