Slogger: Scalable, Near-Zero Loss Disaster Recovery for Distributed Data Stores

Ahmed Alquraan\textsuperscript{1}, Alex Kogan\textsuperscript{2}, Virendra Marathe\textsuperscript{2}, Samer Al-Kiswany\textsuperscript{1}

\textsuperscript{1}University of Waterloo \hspace{1cm} \textsuperscript{2}Oracle Labs
Distributed systems are expected to tolerate disasters

Main techniques

• Synchronous geo-replication
• Snapshotting
Synchronous Geo-Replication

- No data loss
  - Severe performance degradation
Snapshotting

Primary site

Shard 1

Shard 2

Backup site

Snapshot
Snapshotting

Primary site

Backup site

+ Higher performance than synchronous geo-replication
- Large data loss window

Can we have a high performance DR system with small data loss window?
Slogger

A high performance disaster recovery approach that minimizes data loss

Main idea

• Asynchronous geo-replication $\rightarrow$ improves performance
• Leverages modern data center synchronized clocks $\rightarrow$ guarantee consistency
**Slogger – Basic Design**

**Primary site**

- **Leader**
- **Replica**
- **Replica**

**Backup site**

- **Leader**
- **Replica**
- **Replica**

---

**Client**

- write(X) → Leader → Replica → Replica
- ack

**Async_write(X)**

- async_write(X) → Leader → Replica → Replica

---

**Challenge:** consistency across shards?

---

**Features:**

- Higher performance
- Small data loss window
The existence of W is conditional on the existence of Z.
The existence of \( W \) is conditional on the existence of \( Z \).
The existence of W is conditional on the existence of Z.

Violates the consistency between W and Z.
Slogger

Global clock (atomic clocks, PTP)

Client

Primary site

Backup site

Shard 1

Shard 2

\[ t_2 > t_1 > t_0 \]
Slogger

Global clock (atomic clocks, PTP)

Client

Primary site

Backup site

$t_2 > t_1 > t_0$

Leader

Replica

Replica

Leader

Replica

Replica

Leader

Replica

Replica

Leader

Replica

Replica

Leader

Replica

Replica

Global clock

Watermark Service

min($t_1, t_2$)
Slogger

Global clock (atomic clocks, PTP)

Client

write(Z) -> replicate

ack

write(W) -> replicate

ack

\( t_2 > t_1 > t_0 \)

Primary site

Leader

[\( X, t_1, Z, t_1 \)]

Replica

[\( X, t_0, Z, t_1 \)]

Replica

[\( X, t_0, Z, t_1 \)]

async_write(\( Z, t_1 \))

Backup site

Leader

[\( X, t_0, Z, t_1 \)]

Replica

[\( X, t_0, Z, t_1 \)]

Replica

[\( X, t_0, Z, t_1 \)]

async_write(\( Z, t_1 \))

async_write(\( W, t_2 \))

Watermark Service

\( \min(t_1, t_2) \)
Global clock (atomic clocks, PTP)

Client

write(Z)

ack

write(W)

ack

min(t1, t2)

Watermark Service

\( t_2 > t_1 > t_0 \)
Global clock (atomic clocks, PTP)
**Slogger**

**Global clock** (atomic clocks, PTP)

Client

write(Z)

ack

write(W)

ack

\( t_2 > t_1 > t_0 \)

**Primary site**

- Leader
- Replica
- Replica

replicate

**Backup site**

- Leader
- Replica
- Replica

replicate

async_write(W, t_2)

**Shard 1**

- Leader
- Replica
- Replica

**Shard 2**

- Leader
- Replica
- Replica

Watermark Service
Global clock (atomic clocks, PTP)

Client

write(Z)

ack

write(W)

ack

\( t_2 > t_1 > t_0 \)
Evaluation

Alternatives
• Slogger on top of LogCabin
• Synchronous geo-replication
• Incremental snapshotting

Metrics
• Performance
• Backup site lag
• Fault tolerance
• Watermark service scalability

Testbed
• Two CloudLab data centers (Clemson & Wisconsin)
• 16 machines for each site
  • Dual Socket CPU – 10 cores/socket
  • Local network: 10Gbps
  • WAN: 1Gbps
  • RTT: 26 milliseconds
Performance Comparison

Slogger achieves optimal performance with a small data loss window

**Backup lag:** 29ms

**Median Latency (ms)**

Throughput (ops/sec) X 1000

- LogCabin (without geo-replication)
- Slogger
- Snapshot-32KB
- Snapshot-2MB

19
Conclusion

Slogger

• Exploits synchronized clocks within a data center
• Preserves consistency
• Achieves optimal performance with milliseconds data loss window
Thank you!