Designing a Planetary-Scale IMAP Service with CRDTs

Tim Jungnickel, Lennart Oldenburg and Matthias Loibl

TU Berlin
Complex and Distributed IT-Systems

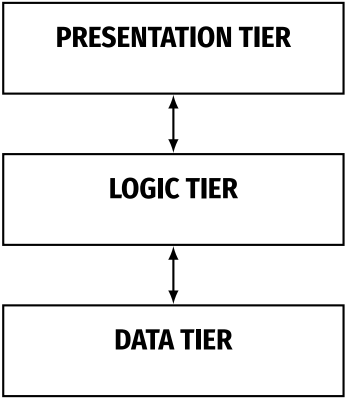
December 19, 2017

Three-Tier Software Architecture

Stateful Service
Data is stored beyond one request.

Stateless Service
All requests are treated as independent ones.

Three-Tier Software Architecture



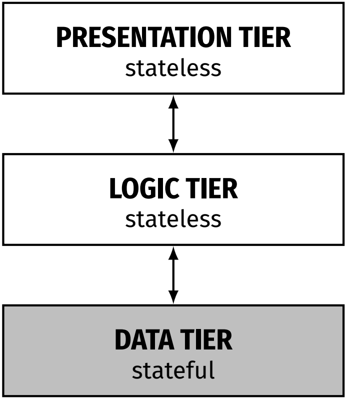
Stateful Service
Data is stored beyond one request.

Stateless Service
All requests are treated as independent ones.

Three-Tier Software Architecture

Stateful Service
Data is stored beyond one request.

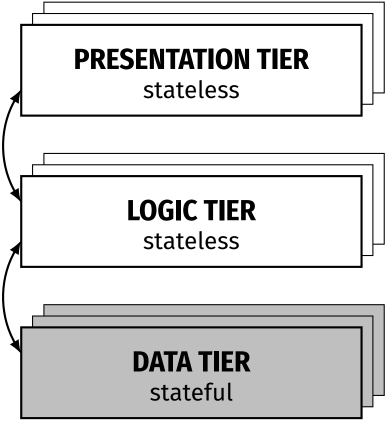
Stateless Service
All requests are treated as independent ones.



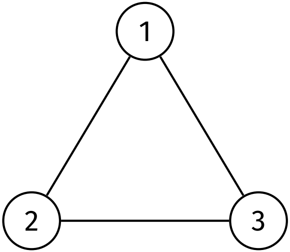
Three-Tier Software Architecture

Stateful Service
Data is stored beyond one request.

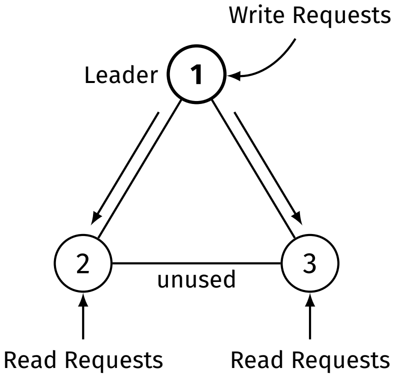
Stateless Service
All requests are treated as independent ones.



Replication



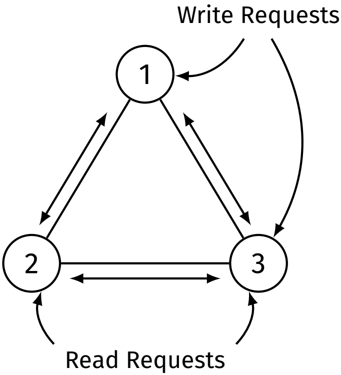
Replication



Single-Leader Replication

Only the leader answers the write requests.

Replication



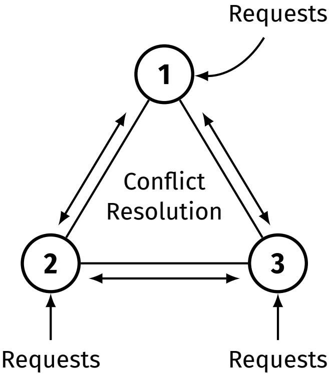
Single-Leader Replication

Only the leader answers the write requests.

Leaderless Replication

Requests are sent to multiple nodes.

Replication



Single-Leader Replication

Only the leader answers the write requests.

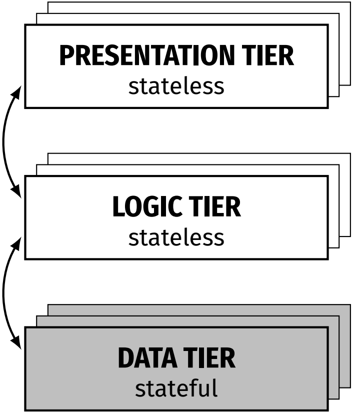
Leaderless Replication

Requests are sent to multiple nodes.

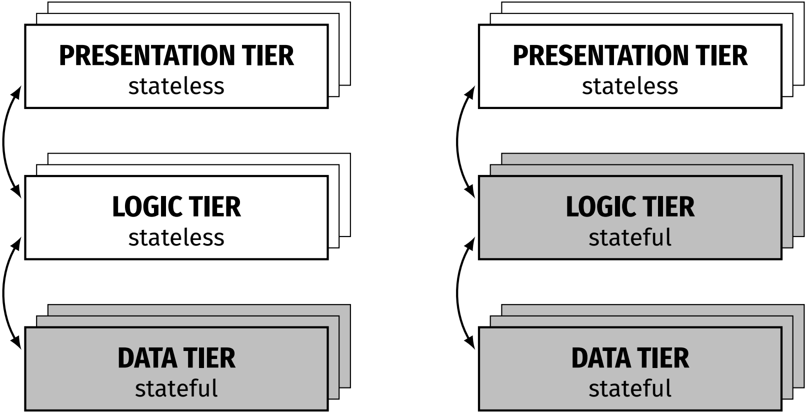
Multi-Leader Replication

All nodes answer write requests.

Towards a Stateful Logic-Layer



Towards a Stateful Logic-Layer



Internet Message Access Protocol

The standard protocol to retrieve
e-mail messages from a mail server.

Example Mail Servers

- **GMail**: 1B users
- **Deutsche Telekom**: 26M users

Control Commands

- Folders:
  `CREATE`, `DELETE`
- Messages:
  `APPEND`, `STORE`, `EXPUNGE`

Internet Message Access Protocol

The standard protocol to retrieve e-mail messages from a mail server.

Example Mail Servers

- **GMail**: 1B users
- **Deutsche Telekom**: 26M users

Control Commands

- Folders:

  `CREATE`, `DELETE`

- Messages:

  `APPEND`, `STORE`, `EXPUNGE`

```
a1 LOGIN user password
a1 OK LOGIN completed
a2 CREATE tuberlin
a2 OK tuberlin created
a3 SELECT inbox
* 18 EXSISTS
* 2 RECENT
a3 OK SELECT completed
a4 EXPUNGE
a4 OK EXPUNGE completed
a5 LOGOUT
* BYE terminating now
a5 OK LOGOUT completed.
```

Conflict-free Replicated Data Types

CRDT’s offer convergence of replicas without synchronization.

System Model

- Asynchronous network of processes
- Processes can crash and recover
- Network can partition and recover

Requirements

- Causal order delivery
- Commutativity of concurrent updates

The IMAP CRDT

Specification: The IMAP CRDT (snippet)

```
payload map $u : \mathcal{N} \rightarrow \mathcal{P}(\texttt{ID}) \times \mathcal{P}(\mathcal{M})$
    initial $(\lambda x.(\varnothing, \varnothing))$
update create (foldername $f$)
    atSource
        let $\alpha = unique()$
    downstream $(f, \alpha)$
        $u(f) \mapsto (u(f)_1 \cup \{\alpha\}, u(f)_2)$
```

The IMAP CRDT

Specification: The IMAP CRDT (snippet)

```
payload map $u : \mathcal{N} \rightarrow \mathcal{P}(\mathtt{ID}) \times \mathcal{P}(\mathcal{M})$
    initial $(\lambda x.(\varnothing, \varnothing))$
update *create* (foldername $f$)
    atSource
        let $\alpha = unique()$
    downstream $(f, \alpha)$
        $u(f) \mapsto (u(f)_1 \cup \{\alpha\}, u(f)_2)$
```

- Specified for all *consistency critical* IMAP commands.
- Fully verified in Isabelle based on a CRDT Framework:
  - Asynchronous network, crash failures, etc.
  - Commutativity of concurrent operations.
  - Convergence of replicas.

Pluto

Research Prototype

- Free Software, written in go.
- Causal order delivery + IMAP CRDT

Pluto

Research Prototype

- Free Software, written in go.
- Causal order delivery + IMAP CRDT

IMAP Benchmark

- Write intensive workload generation.
- Customizable and reusable for other IMAP servers.

Pluto

Research Prototype

- Free Software, written in go.
- Causal order delivery + IMAP CRDT

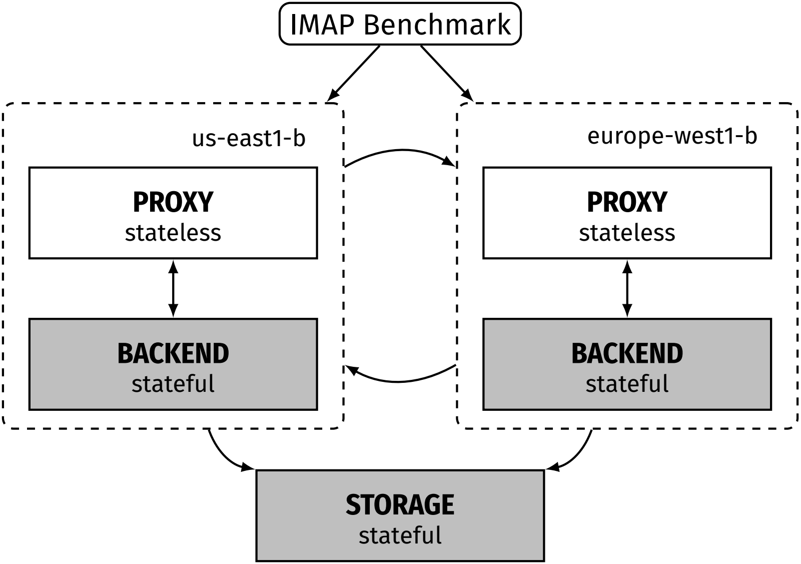
IMAP Benchmark

- Write intensive workload generation.
- Customizable and reusable for other IMAP servers.

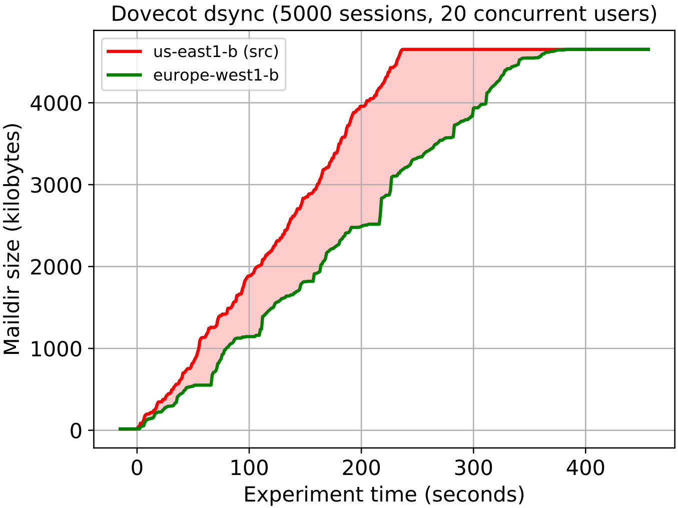
Cloud Deployment

- Kubernetes based deployment in the Google Cloud.
- Monitoring with *Prometheus* and *Styx* [PromCon 2017].

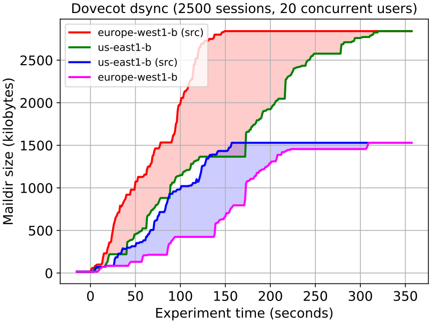
Experiment Setup

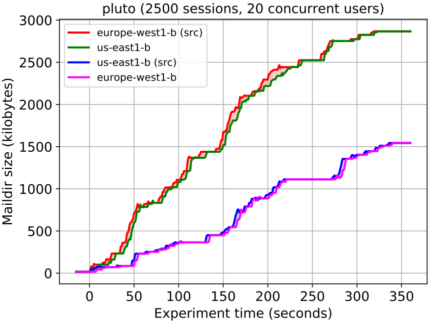


Replication Lag Diagrams

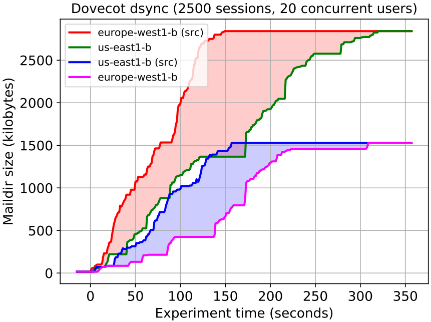


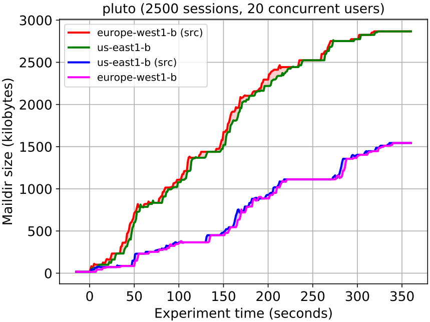
Multi-Datacenter Replication Lag





Multi-Datacenter Replication Lag





209.83 MB*s

97.92 MB*s

14.32 MB*s

5.83 MB*s

Conclusion

IMAP Server based on CRDTs

- We provide a verified IMAP-CRDT that guarantees convergence among replicas.
- We were able to reduce the replication lag.
- The response time needs improvement in order to compete with industry software.

Conclusion

IMAP Server based on CRDTs

- We provide a verified IMAP-CRDT that guarantees convergence among replicas.
- We were able to reduce the replication lag.
- The response time needs improvement in order to compete with industry software.

Takeaways

- Implementing multi-leader replication on the logic-layer is a challenging but manageable task.
- CRDTs offer the necessary tools to build software at planetary scale.

fin