Tango: Distributed Data Structures over a Shared Log

Group 10

- André Matos 92420
- Diogo Ribeiro 102484
- Luís Calado 103883

Index

Shared Log

Tango builds on the CORFU shared log abstraction

Log uses chain replication

CORFU operations:

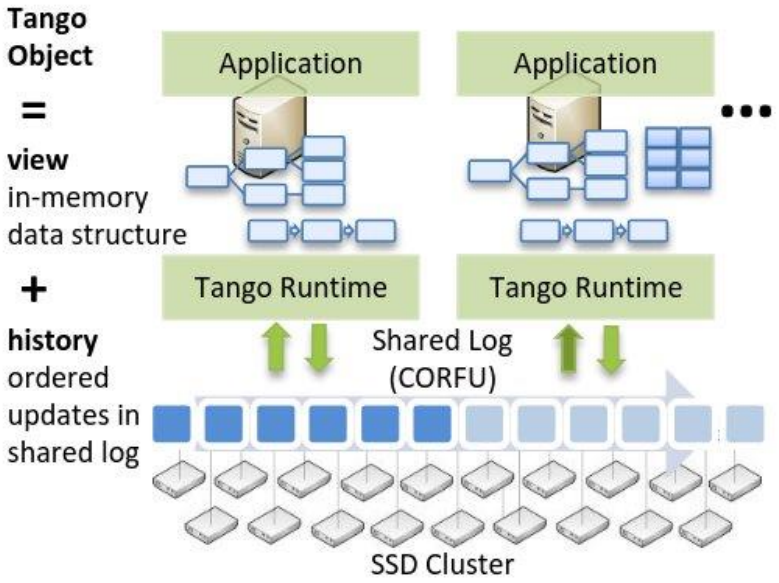
- **append(b)** -> append entry b to the shared log, obtaining an offset/position l in return
- **check()** -> check the current tail of the log
- **read(l)** -> read the entry at log position l
- **trim(l)** -> trim a particular offset in the log for garbage collection

No communication between clients.

CORFU Sequencer is used.

The Tango Object = View + History

- **Structure:** Historical log of updates + multiple views (tree, map).
- **Client Interaction:** Append updates to log, sync views with apply upcall.
- **Custom Views:** Different views (tree, set) from same log.
- **Consistency & Durability:** Linearizability, recovery via log history, checkpoints.



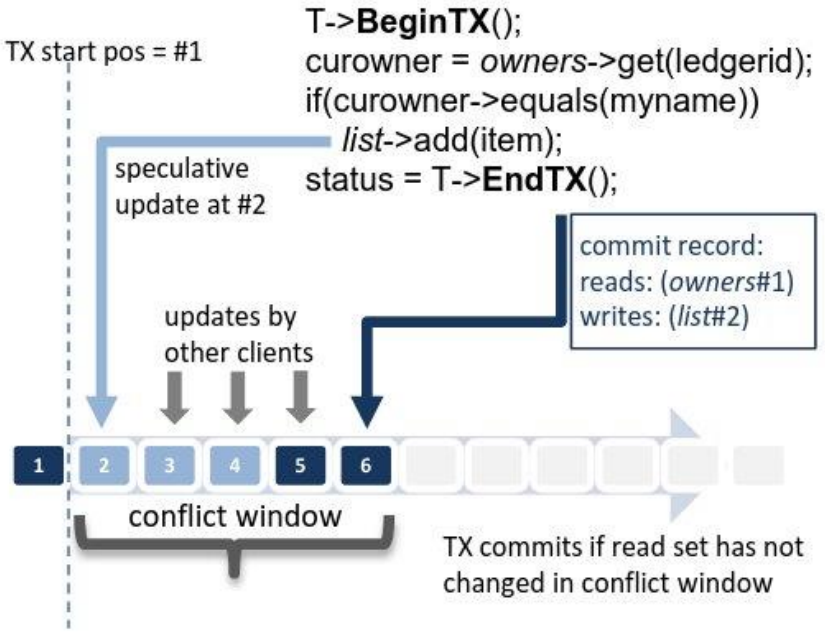
TangoRegister

- **update helper**: this accepts an opaque buffer from the object and appends it to the shared log.
- **query helper**: this reads new entries from the shared log and provides them to the object via an apply upcall.

```
class TangoRegister {
    int oid;
    TangoRuntime *T;
    int state;
    void apply(void *X) {
        state = *(int *)X;
    }
    void writeRegister(int newstate){
        T->update_helper(&newstate,
                         sizeof(int), oid);
    }
    int readRegister() {
        T->query_helper(oid);
        return state;
    }
}
```

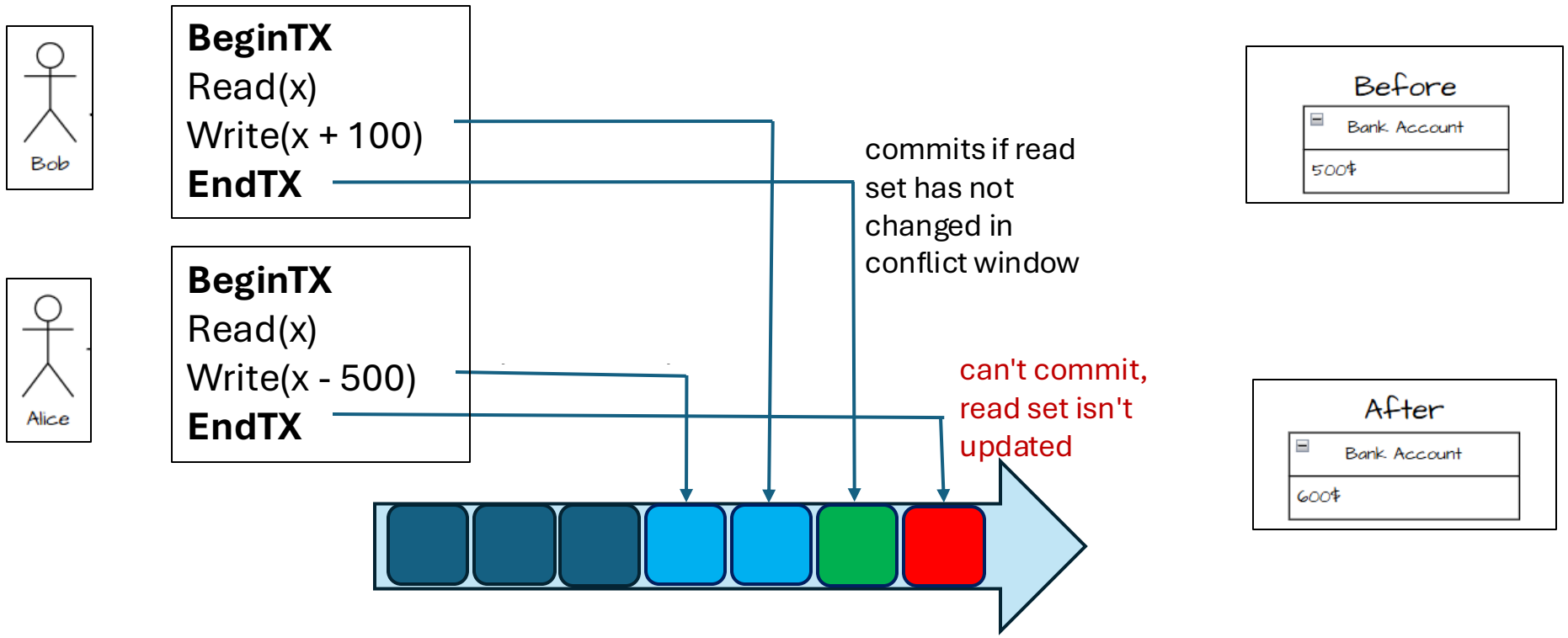
Transactions

- **Concurrency:** Speculative commit, decision after log check.
- **Commit Record:** Tracks readset (objects/versions), checks for changes.
- **Atomicity/Isolation:** Single log ensures multi-object transactions.
- **Transaction Flow:** BeginTX sets context, EndTX commits or aborts.



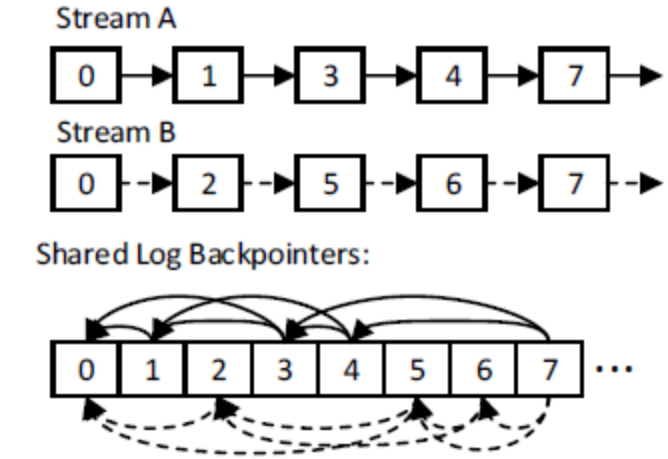
Transactions

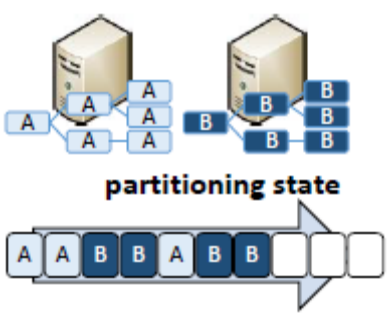
- **Read/Write:** Read-only skips commit; write-only skips log traversal.

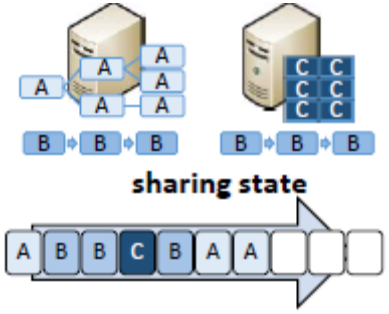


Layered Partitions

- **Partitioning:** Objects mapped to streams; clients read relevant streams.
- **Streams:** Linked lists built through backpointers.
- **Append:** Stream IDs sent to sequencer, respond with backpointers.
- **Failure Handling:** Reverse log traversal, junk values fill gaps.

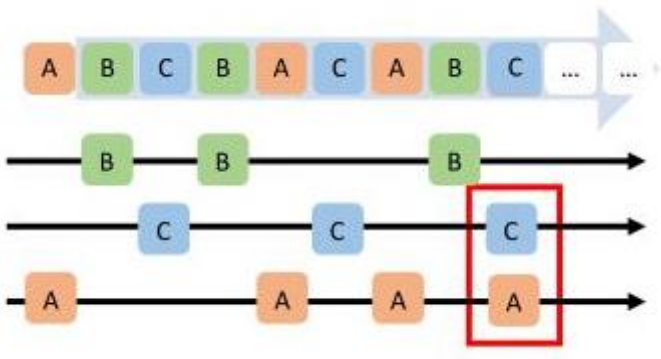


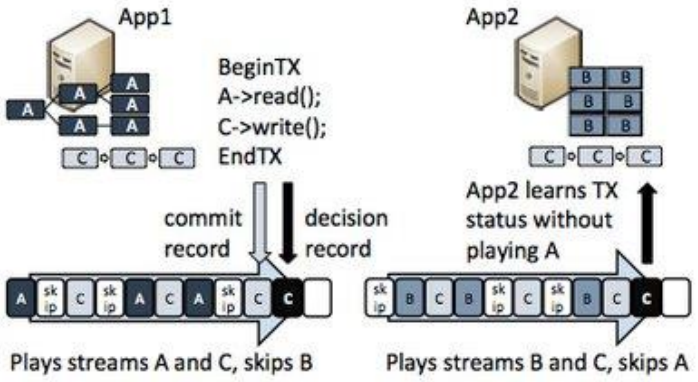




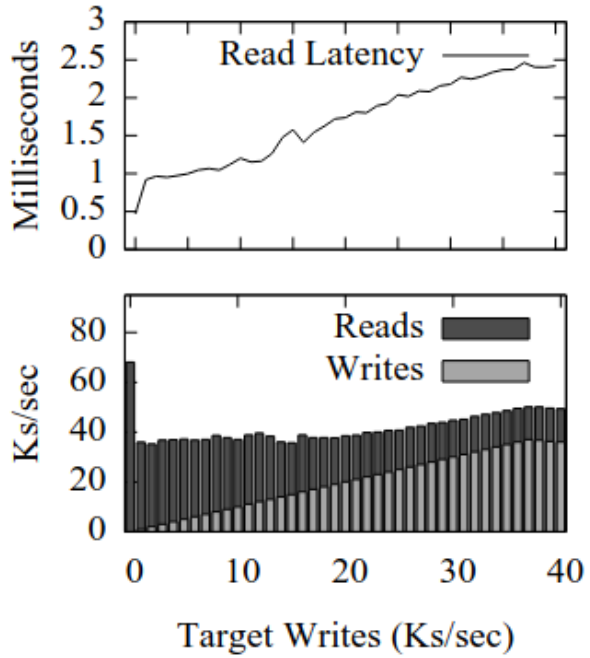
Transactions Over Streams

- **Multiappend:** Transactions affect multiple streams, single commit in shared log.
- **Limitation:** Can't involve reads from objects without local copies.
- **Commit Decision:** Client adds commit decision to log; other clients can add if needed.
- **Failure Handling:** If commit fails, any client can reconstruct objects from the log after a timeout.



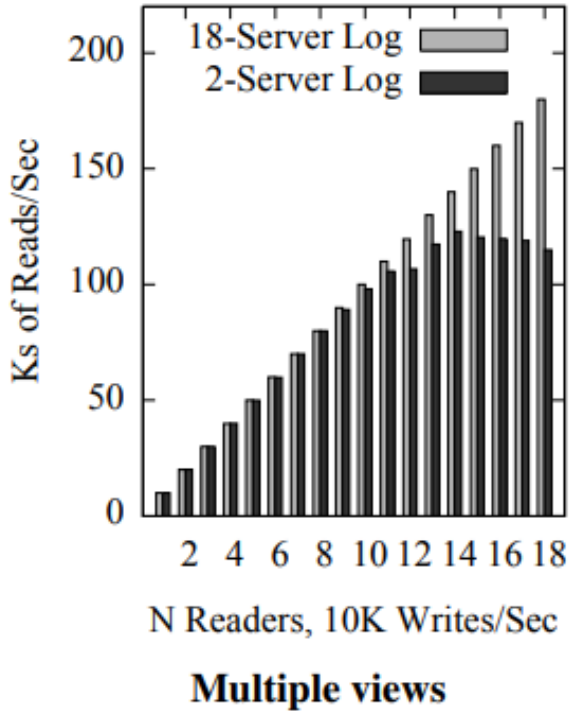


Evaluation

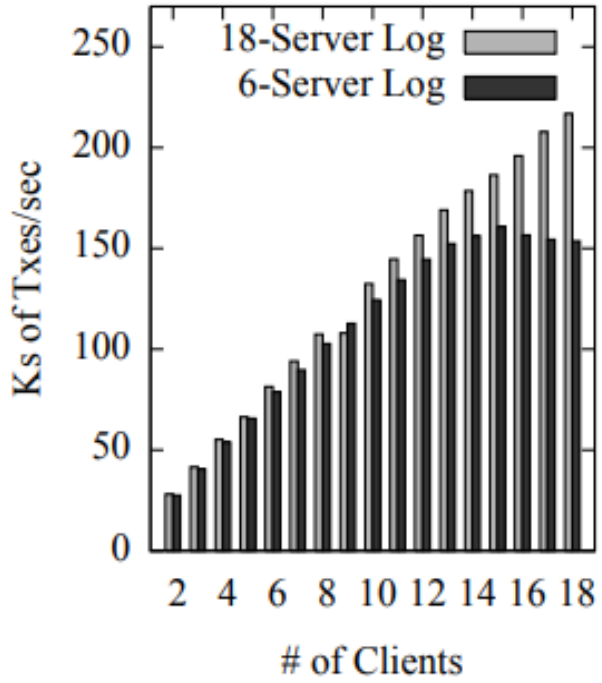


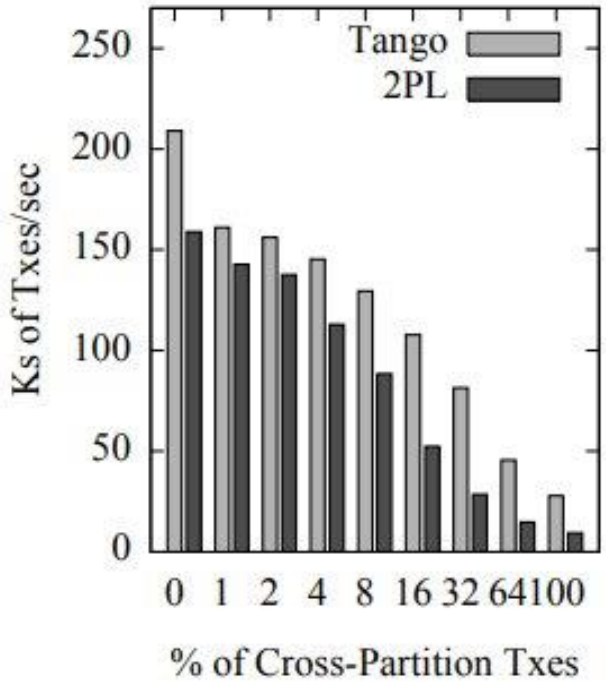
Two views

Evaluation


18-Server Log
2-Server Log
200
150
100
50
0
Ks of Reads/Sec
2 4 6 8 10 12 14 16 18
N Readers, 10K Writes/Sec
Multiple views

Evaluation


18-Server Log
6-Server Log
Ks of Txes/sec
250
200
150
100
50
0
2 4 6 8 10 12 14 16 18
# of Clients


Tango
2PL
Ks of Txes/sec
250
200
150
100
50
0
0 1 2 4 8 16 32 64100
% of Cross-Partition Txes

Conclusion

Main Contributions:

- Different structures, same log
- Clients with different views
- Clients choose views