
MODULE *U2PC*

EXTENDS *FiniteSets, Integers, Apache, TLC*

```

@typeAlias: key = Str;
@typeAlias: rid = Str;
@typeAlias: tid = Str;
@typeAlias: version = $tid;
@typeAlias: txn = Set($key);
@typeAlias: txnstate = $key → $version;
U2PC_ALIAS  $\triangleq$  TRUE

```

CONSTANTS

```

@type: $key → Set($rid);
Shards,
@type: $tid → $txn;
Txns

```

ASSUME $\forall k1, k2 \in \text{DOMAIN } Shards : k1 \neq k2 \Rightarrow Shards[k1] \cap Shards[k2] = \{\}$
 ASSUME "Init" $\notin \text{DOMAIN } Txns$

```

msg_read = Str;
msg_read_resp = {key : $key, ver : $version};
msg_lock = {txn : $txn, key : $key, ver : $version};
msg_lock_resp = Bool;
msg_unlock = Bool;
msg_unlock_resp = Bool;

```

VARIABLES

```

@type: $rid → {locked : Bool, version : $version, logged : $version};
Replicas,
@type: $tid → Str;
Coordinator_state,
@type: $tid → $txnstate;
Coordinator_txn_state,
@type: Set({src : $tid, key : $key});
M_read,
@type: Set({src : $rid, dst : $tid, ver : $version});
M_read_resp,
@type: Set({tid : $tid, txn : $txn, state : $txnstate});
M_lock,
@type: Set({src : $rid, dst : $tid, locked : Bool});
M_lock_resp,
@type: Set({src : $tid, apply : Bool});
M_unlock,
@type: Set({src : $rid, tid : $tid});
M_unlock_resp,

```

$\text{@type: } \$tid \rightarrow \text{Set}(\$tid);$
 $\text{Linearisability_rt}$
 $\text{Vars} \triangleq \langle \text{Replicas},$
 $\quad \text{Coordinator_state}, \text{Coordinator_txn_state},$
 $\quad \text{M_read}, \text{M_read_resp},$
 $\quad \text{M_lock}, \text{M_lock_resp},$
 $\quad \text{M_unlock}, \text{M_unlock_resp},$
 $\quad \text{Linearisability_rt} \rangle$
 $\text{Var_M_read} \triangleq \langle \text{M_read}, \text{M_read_resp} \rangle$
 $\text{Var_M_lock} \triangleq \langle \text{M_lock}, \text{M_lock_resp} \rangle$
 $\text{Var_M_unlock} \triangleq \langle \text{M_unlock}, \text{M_unlock_resp} \rangle$
 $\text{Var_Msgs} \triangleq \langle \text{Var_M_read}, \text{Var_M_lock}, \text{Var_M_unlock} \rangle$
 $\text{@type: } (a \rightarrow b) \Rightarrow \text{Set}(b);$
 $\text{Range}(F) \triangleq \{F[x] : x \in \text{DOMAIN } F\}$
 $\text{@type: } \text{Set}(\$rid);$
 $\text{RIDs} \triangleq \text{UNION } \text{Range}(\text{Shards})$
 $\text{TIDs} \triangleq \text{DOMAIN } \text{Txsns}$
 $\text{KeyLookup} \triangleq [r \in \text{RIDs} \mapsto \text{CHOOSE } k \in \text{DOMAIN } \text{Shards} : r \in \text{Shards}[k]]$
 $\text{Init} \triangleq$
 $\quad \wedge \text{Replicas} = [r \in \text{RIDs} \mapsto$
 $\quad \quad [\text{locked} \mapsto \text{FALSE}, \text{version} \mapsto \text{"Init"}, \text{logged} \mapsto \text{"NULL"}]]$
 $\quad \wedge \text{Coordinator_state} = [t \in \text{TIDs} \mapsto \text{"Start"}]$
 $\quad \wedge \text{Coordinator_txn_state} = [t \in \text{TIDs} \mapsto \text{SetAsFun}(\{\})]$
 $\quad \wedge \text{M_read} = \{\} \wedge \text{M_read_resp} = \{\}$
 $\quad \wedge \text{M_lock} = \{\} \wedge \text{M_lock_resp} = \{\}$
 $\quad \wedge \text{M_unlock} = \{\} \wedge \text{M_unlock_resp} = \{\}$
 $\quad \wedge \text{Linearisability_rt} = [t \in \text{TIDs} \mapsto \{\}]$
 $\text{RelevantReplicas}(t) \triangleq \text{UNION } \{\text{Shards}[k] : k \in \text{Txsns}[t]\}$
 $\text{CoordinatorStart}(t) \triangleq$
 $\quad \wedge \text{Coordinator_state}[t] = \text{"Start"}$
 $\quad \wedge \text{M_read}' = \text{M_read} \cup \{[src \mapsto t, key \mapsto k] : k \in \text{Txsns}[t]\}$
 $\quad \wedge \text{UNCHANGED } \langle \text{M_read_resp}, \text{Var_M_lock}, \text{Var_M_unlock} \rangle$
 $\quad \wedge \text{Coordinator_state}' = [\text{Coordinator_state} \text{ EXCEPT } ![t] = \text{"Read"}]$
 $\quad \wedge \text{UNCHANGED } \langle \text{Coordinator_txn_state}, \text{Replicas} \rangle$
 $\quad \wedge \text{Linearisability_rt}' = [\text{Linearisability_rt} \text{ EXCEPT } ![t] =$
 $\quad \quad \{t1 \in \text{TIDs} : \text{Coordinator_state}[t1] = \text{"Commit"}\}]$
 $\text{ReplicaRead}(r) \triangleq$
 $\quad \wedge \text{Replicas}[r].\text{locked} = \text{FALSE}$

$$\begin{aligned}
& \wedge \exists m \in M_read : \\
& \quad \wedge \neg \exists m1 \in M_read_resp : m1.src = r \wedge m1.dst = m.src \\
& \quad \wedge M_read_resp' = M_read_resp \cup \\
& \quad \quad \{[src \mapsto r, dst \mapsto m.src, ver \mapsto Replicas[r].version]\} \\
& \quad \wedge \text{UNCHANGED } \langle M_read, Var_M_lock, Var_M_unlock \rangle \\
& \wedge \text{UNCHANGED } \langle Replicas \rangle \\
& \wedge \text{UNCHANGED } \langle Coordinator_state, Coordinator_txn_state, Linearisability_rt \rangle
\end{aligned}$$

$$\begin{aligned}
CoordinatorRead(t) & \triangleq \\
& \wedge Coordinator_state[t] = \text{"Read"} \\
& \wedge \forall k \in Txns[t] : \exists m \in M_read_resp : KeyLookup[m.src] = k \\
& \wedge \exists F \in [Txns[t] \rightarrow RIDs] : \\
& \quad \wedge \forall k \in Txns[t] : \wedge k = KeyLookup[F[k]] \\
& \quad \quad \wedge \exists m \in M_read_resp : m.dst = t \wedge m.src = F[k] \\
& \quad \wedge Coordinator_txn_state' = [Coordinator_txn_state \text{ EXCEPT } ![t] = [\\
& \quad \quad k \in Txns[t] \mapsto \\
& \quad \quad \quad (\text{CHOOSE } m \in M_read_resp : m.dst = t \wedge m.src = F[k]).ver \\
& \quad \quad \quad] \\
& \quad \wedge Coordinator_state' = [Coordinator_state \text{ EXCEPT } ![t] = \text{"Lock"}] \\
& \wedge \text{UNCHANGED } \langle Replicas, Msgs, Linearisability_rt \rangle
\end{aligned}$$

$$\begin{aligned}
CoordinatorLock(t) & \triangleq \\
& \wedge Coordinator_state[t] = \text{"Lock"} \\
& \wedge M_lock' = M_lock \cup \\
& \quad \{[tid \mapsto t, txn \mapsto Txns[t], state \mapsto Coordinator_txn_state[t]]\} \\
& \wedge \text{UNCHANGED } \langle M_lock_resp, Var_M_read, Var_M_lock \rangle \\
& \wedge Coordinator_state' = [Coordinator_state \text{ EXCEPT } ![t] = \text{"Decide"}] \\
& \wedge \text{UNCHANGED } \langle Coordinator_txn_state, Replicas, Linearisability_rt \rangle
\end{aligned}$$

$$\begin{aligned}
ReplicaLock(r) & \triangleq \\
& \wedge \exists m \in M_lock : \\
& \quad \wedge KeyLookup[r] \in m.txn \\
& \quad \wedge \neg \exists m1 \in M_lock_resp : m1.src = r \wedge m1.dst = m.tid \\
& \quad \wedge \text{IF } (\neg Replicas[r].locked) \wedge Replicas[r].version = m.state[KeyLookup[r]] \\
& \quad \quad \text{THEN} \\
& \quad \quad \wedge Replicas' = [Replicas \text{ EXCEPT } ![r] = [\\
& \quad \quad \quad locked \mapsto \text{TRUE}, version \mapsto Replicas[r].version, logged \mapsto m.tid]] \\
& \quad \quad \wedge M_lock_resp' = M_lock_resp \cup \\
& \quad \quad \quad \{[src \mapsto r, dst \mapsto m.tid, locked \mapsto \text{TRUE}]\} \\
& \quad \quad \text{ELSE} \\
& \quad \quad \wedge M_lock_resp' = M_lock_resp \cup \\
& \quad \quad \quad \{[src \mapsto r, dst \mapsto m.tid, locked \mapsto \text{FALSE}]\} \\
& \quad \quad \wedge \text{UNCHANGED } Replicas \\
& \wedge \text{UNCHANGED } \langle M_lock, Var_M_read, Var_M_unlock \rangle \\
& \wedge \text{UNCHANGED } \langle Coordinator_state, Coordinator_txn_state, Linearisability_rt \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{CoordinatorCommit}(t) \triangleq \\
& \quad \wedge \text{Coordinator_state}[t] = \text{"Decide"} \\
& \quad \wedge \forall k \in \text{Txns}[t] : \forall r \in \text{Shards}[k] : \exists m \in M_lock_resp : \\
& \quad \quad \wedge m.src = r \wedge m.dst = t \\
& \quad \quad \wedge m.locked \\
& \quad \wedge \text{Coordinator_state}' = [\text{Coordinator_state} \text{ EXCEPT } ![t] = \text{"Commit"}] \\
& \quad \wedge M_unlock' = M_unlock \cup \{[src \mapsto t, apply \mapsto \text{TRUE}]\} \\
& \quad \wedge \text{UNCHANGED } \langle \text{Var_M_read}, \text{Var_M_lock}, M_unlock_resp \rangle \\
& \quad \wedge \text{UNCHANGED } \langle \text{Replicas}, \text{Coordinator_txn_state}, \text{Linearisability_rt} \rangle \\
\\
& \text{CoordinatorStartAbort}(t) \triangleq \\
& \quad \wedge \text{Coordinator_state}[t] = \text{"Decide"} \\
& \quad \wedge \exists k \in \text{Txns}[t] : \exists r \in \text{Shards}[k] : \exists m \in M_lock_resp : \\
& \quad \quad \wedge m.src = r \wedge m.dst = t \\
& \quad \quad \wedge \neg m.locked \\
& \quad \wedge \text{Coordinator_state}' = [\text{Coordinator_state} \text{ EXCEPT } ![t] = \text{"TryAbort"}] \\
& \quad \wedge M_unlock' = M_unlock \cup \{[src \mapsto t, apply \mapsto \text{FALSE}]\} \\
& \quad \wedge \text{UNCHANGED } \langle \text{Var_M_read}, \text{Var_M_lock}, M_unlock_resp \rangle \\
& \quad \wedge \text{UNCHANGED } \langle \text{Replicas}, \text{Coordinator_txn_state}, \text{Linearisability_rt} \rangle \\
\\
& \text{ReplicaUnlock}(r) \triangleq \\
& \quad \exists m \in M_unlock : \\
& \quad \wedge \text{Replicas}[r].locked \\
& \quad \wedge m.src = \text{Replicas}[r].logged \\
& \quad \wedge \text{IF } m.apply \\
& \quad \quad \text{THEN } \text{Replicas}' = [\text{Replicas} \text{ EXCEPT } ![r] = [\\
& \quad \quad \quad locked \mapsto \text{FALSE}, version \mapsto \text{Replicas}[r].logged, logged \mapsto \text{"NULL"}]] \\
& \quad \quad \text{ELSE } \text{Replicas}' = [\text{Replicas} \text{ EXCEPT } ![r] = [\\
& \quad \quad \quad locked \mapsto \text{FALSE}, version \mapsto \text{Replicas}[r].version, logged \mapsto \text{"NULL"}]] \\
& \quad \wedge M_unlock_resp' = M_unlock_resp \cup \{[src \mapsto r, tid \mapsto \text{Replicas}[r].logged]\} \\
& \quad \wedge \text{UNCHANGED } \langle \text{Var_M_read}, \text{Var_M_lock}, M_unlock \rangle \\
& \quad \wedge \text{UNCHANGED } \langle \text{Coordinator_state}, \text{Coordinator_txn_state}, \text{Linearisability_rt} \rangle \\
\\
& \text{CoordinatorAbort}(t) \triangleq \\
& \quad \wedge \text{Coordinator_state}[t] = \text{"TryAbort"} \\
& \quad \wedge \exists k \in \text{Txns}[t] : \forall r \in \text{Shards}[k] : \\
& \quad \quad \vee \exists m \in M_unlock_resp : m.src = r \wedge m.tid = t \\
& \quad \quad \vee \exists m \in M_lock_resp : m.src = r \wedge m.dst = t \wedge \neg m.locked \\
& \quad \wedge \text{Coordinator_state}' = [\text{Coordinator_state} \text{ EXCEPT } ![t] = \text{"Abort"}] \\
& \quad \wedge \text{UNCHANGED } \langle \text{Replicas}, \text{Msgs}, \text{Coordinator_txn_state}, \text{Linearisability_rt} \rangle \\
\\
& \text{Next} \triangleq \\
& \quad \vee \exists r \in \text{RIDs} : \vee \text{ReplicaRead}(r) \\
& \quad \quad \vee \text{ReplicaLock}(r) \\
& \quad \quad \vee \text{ReplicaUnlock}(r)
\end{aligned}$$

$$\begin{aligned}
& \vee \exists t \in TIDs : \vee \text{CoordinatorStart}(t) \\
& \quad \vee \text{CoordinatorRead}(t) \\
& \quad \vee \text{CoordinatorLock}(t) \\
& \quad \vee \text{CoordinatorCommit}(t) \\
& \quad \vee \text{CoordinatorStartAbort}(t) \\
& \quad \vee \text{CoordinatorAbort}(t)
\end{aligned}$$

$$Spec \triangleq Init \wedge \Box[Next]_{Vars}$$

$$\begin{aligned}
& \text{Serialisability}(C) \triangleq \\
& \quad \vee \text{Cardinality}(C) < 2 \\
& \quad \vee \exists R \in \text{SUBSET}(C \times C) : \\
& \quad \quad \text{Irreflexive} \\
& \quad \quad \wedge \forall t1 \in C : \langle t1, t1 \rangle \notin R \\
& \quad \quad \text{Transitive} \\
& \quad \quad \wedge \forall t1, t2, t3 \in C : (\langle t1, t2 \rangle \in R \wedge \langle t2, t3 \rangle \in R) \Rightarrow \langle t1, t3 \rangle \in R \\
& \quad \quad \text{Above 2 ensure there are no cycles} \\
& \quad \quad R \text{ respects observed order} \\
& \quad \quad \wedge \forall t1, t2 \in C : \\
& \quad \quad \quad (\exists k \in \text{Txns}[t2] : \text{Coordinator_txn_state}[t2][k] = t1) \Rightarrow \langle t1, t2 \rangle \in R \\
& \quad \quad \text{If two transactions interfere, there is an order} \\
& \quad \quad \wedge \forall t1, t2 \in C : \\
& \quad \quad \quad (t1 \neq t2 \wedge \text{Txns}[t1] \cap \text{Txns}[t2] \neq \{\}) \Rightarrow \langle t1, t2 \rangle \in R \vee \langle t2, t1 \rangle \in R \\
& \quad \quad \text{Strict serialisability / Linearisability check} \\
& \quad \quad \wedge \forall t1, t2 \in C : \\
& \quad \quad \quad (t1 \in \text{Linearisability_rt}[t2]) \Rightarrow \langle t1, t2 \rangle \in R
\end{aligned}$$

$$\text{CommittedTIDs} \triangleq \{t \in TIDs : \text{Coordinator_state}[t] = \text{"Commit"}\}$$

$$\text{AbortedTIDs} \triangleq \{t \in TIDs : \text{Coordinator_state}[t] = \text{"Abort"}\}$$

$$\text{Safety_non_recovery} \triangleq \text{Serialisability}(\text{CommittedTIDs})$$