

Specification of the consensus protocol in *PaxosStore*.

See [*PaxosStore@VLDB2017*](<https://www.vldb.org/pvldb/vol10/p1730-lin.pdf>) by Tencent.

In this version (adopted from “*PaxosStore.tla*”):

- Client-restricted config (Ballot)
 - *Message* types (*i.e.*, “Prepare”, “Accept”, “ACK”) are deleted. No state flags (such as “Prepare”, “Wait-Prepare”, “Accept”, “Wait-Accept”) are needed. – Choose value from a quorum in *Accept*.

EXTENDS *Integers, FiniteSets*

$Max(m, n) \triangleq \text{IF } m > n \text{ THEN } m \text{ ELSE } n$
 $Injective(f) \triangleq \forall a, b \in \text{DOMAIN } f : (a \neq b) \Rightarrow (f[a] \neq f[b])$

CONSTANTS

Participant, the set of participants
Value the set of possible input values for *Participant* to propose

$None \triangleq \text{CHOOSE } b : b \notin \text{Value}$
 $NP \triangleq \text{Cardinality}(\text{Participant})$ number of $p \in \text{Participants}$

$Quorum \triangleq \{Q \in \text{SUBSET } Participant : \text{Cardinality}(Q) * 2 \geq NP + 1\}$

ASSUME $QuorumAssumption \triangleq$
 $\wedge \forall Q \in Quorum : Q \subseteq Participant$
 $\wedge \forall Q1, Q2 \in Quorum : Q1 \cap Q2 \neq \{\}$

$Ballot \triangleq Nat$

$PIndex \triangleq \text{CHOOSE } f \in [Participant \rightarrow 1..NP] : Injective(f)$
 $Bals(p) \triangleq \{b \in Ballot : b \% NP = PIndex[p] - 1\}$ allocate ballots for each $p \in Participant$

$State \triangleq [maxBal : Ballot \cup \{-1\},$
 $maxVVal : Value \cup \{None\}]$

$InitState \triangleq [maxBal \mapsto -1, maxVVal \mapsto -1, maxVVal \mapsto None]$

For simplicity, in this specification, we choose to send the complete state of a participant each time. When receiving such a message, the participant processes only the “partial” state it needs.

$Message \triangleq [from : Participant, to : \text{SUBSET } Participant, state : [Participant \rightarrow State]]$

VARIABLES

state, $state[p][q]$: the state of $q \in Participant$ from the view of $p \in Participant$
msgs the set of messages that have been sent

$vars \triangleq \langle state, msgs \rangle$

$TypeOK \triangleq$
 $\wedge state \in [Participant \rightarrow [Participant \rightarrow State]]$
 $\wedge msgs \subseteq Message$

$Send(m) \triangleq msgs' = msgs \cup \{m\}$

$Init \triangleq$

$\wedge state = [p \in Participant \mapsto [q \in Participant \mapsto InitState]]$
 $\wedge msgs = \{\}$

$p \in Participant$ starts the prepare phase by issuing a ballot $b \in Ballot$.

$Prepare(p, b) \triangleq$

$\wedge state[p][p].maxBal < b$
 $\wedge b \in Bals(p)$
 $\wedge state' = [state \text{ EXCEPT } ![p][p].maxBal = b]$
 $\wedge Send([from \mapsto p, to \mapsto Participant, state \mapsto state'[p]])$

$q \in Participant$ updates its own state $state[q]$ according to the actual state pp of $p \in Participant$ extracted from a message $m \in Message$ it receives. This is called by $OnMessage(q)$.

Note: pp is $m.state[p]$; it may not be equal to $state[p][p]$ at the time $UpdateState$ is called.

$UpdateState(q, p, pp) \triangleq$

$state' = [state \text{ EXCEPT}$
 $![q][p].maxBal = Max(@, pp.maxBal),$
 $![q][p].maxVVal = Max(@, pp.maxVVal),$
 $![q][p].maxVVal = \text{IF } state[q][p].maxVVal < pp.maxVVal$
 $\text{THEN } pp.maxVVal \text{ ELSE } @,$
 $![q][q].maxBal = Max(@, pp.maxBal),$
 $![q][q].maxVVal = \text{IF } state[q][q].maxVVal \leq pp.maxVVal$
 $\text{THEN } pp.maxVVal \text{ ELSE } @, \text{ make promise}$
 $![q][q].maxVVal = \text{IF } state[q][q].maxVVal \leq pp.maxVVal$
 $\text{THEN } pp.maxVVal \text{ ELSE } @] \text{ accept}$

$q \in Participant$ receives and processes a message in $Message$.

$OnMessage(q) \triangleq$

$\exists m \in msgs :$
 $\wedge q \in m.to$
 $\wedge \text{LET } p \triangleq m.from$
 $\text{IN } UpdateState(q, p, m.state[p])$
 $\wedge \text{IF } \vee m.state[q].maxBal < state'[q][q].maxBal$
 $\vee m.state[q].maxVVal < state'[q][q].maxVVal$
 $\text{THEN } Send([from \mapsto q, to \mapsto \{m.from\}, state \mapsto state'[q]])$
 $\text{ELSE UNCHANGED } msgs$

$p \in Participant$ starts the accept phase by issuing the ballot $b \in Ballot$ with value $v \in Value$.

$Accept(p, b, v) \triangleq$

$\wedge b \in Bals(p)$
 $\wedge state[p][p].maxBal = b$ corresponding the first conjunction in Voting
 $\wedge state[p][p].maxVVal < b$ corresponding the second conjunction in Voting
 $\wedge \exists Q \in Quorum : \text{pick the value from the quorum}$
 $\wedge \forall q \in Q : state[p][q].maxBal = b$
 $\wedge \forall q \in Q : state[p][q].maxVVal = -1$ free to pick its own value

