

```

1  |----- MODULE OneVotePaxosStore -----|
   | 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 “UniversalPaxosStore.tla”):
   | - Use OneVote and IntersectingQuorum together to replace the Client-restricted config for Ballot
   |   allocation; that is, no Bals(p) in this version.
   | - Still no message types or state flags.
16 | EXTENDS Integers, FiniteSets
17 |-----|
18 |  $Max(m, n) \triangleq \text{IF } m > n \text{ THEN } m \text{ ELSE } n$ 
19 |-----|
20 | CONSTANTS
21 |   Participant, the set of participants
22 |   Value         the set of possible input values for Participant to propose
24 |  $None \triangleq \text{CHOOSE } b : b \notin Value$ 
26 |  $Quorum \triangleq \{Q \in \text{SUBSET } Participant : \text{Cardinality}(Q) * 2 = \text{Cardinality}(Participant) + 1\}$ 
27 |
28 | ASSUME  $QuorumAssumption \triangleq$ 
29 |    $\wedge \forall Q \in Quorum : Q \subseteq Participant$ 
30 |    $\wedge \forall Q1, Q2 \in Quorum : Q1 \cap Q2 \neq \{\}$ 
32 |  $Ballot \triangleq Nat$ 
33 |-----|
34 |  $State \triangleq [maxBal : Ballot \cup \{-1\},$ 
35 |    $maxVVal : Ballot \cup \{-1\}, maxVVal : Value \cup \{None\}]$ 
37 |  $InitState \triangleq [maxBal \mapsto -1, maxVVal \mapsto -1, maxVVal \mapsto None]$ 
   | 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.
43 |  $Message \triangleq [from : Participant, to : \text{SUBSET } Participant, state : [Participant \rightarrow State]]$ 
44 |-----|
45 | VARIABLES
46 |   state,  $state[p][q]$ : the state of  $q \in Participant$  from the view of  $p \in Participant$ 
47 |   msgs   the set of messages that have been sent
49 |  $vars \triangleq \langle state, msgs \rangle$ 
51 |  $TypeOK \triangleq$ 
52 |    $\wedge state \in [Participant \rightarrow [Participant \rightarrow State]]$ 
53 |    $\wedge msgs \subseteq Message$ 
55 |  $Send(m) \triangleq msgs' = msgs \cup \{m\}$ 
56 |-----|
57 |  $Init \triangleq$ 

```

58 $\wedge state = [p \in Participant \mapsto [q \in Participant \mapsto InitState]]$
 59 $\wedge msgs = \{\}$
 $p \in Participant$ starts the prepare phase by issuing a ballot $b \in Ballot$.
 63 $Prepare(p, b) \triangleq$
 64 $\wedge state[p][p].maxBal < b$
 65 $\wedge state' = [state \text{ EXCEPT } ![p][p].maxBal = b]$
 66 $\wedge Send([from \mapsto p, to \mapsto Participant, state \mapsto state'[p]])$
 $q \in Participant$ 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.
 75 $UpdateState(q, p, pp) \triangleq$
 76 $state' = [state \text{ EXCEPT}$
 77 $![q][p].maxBal = Max(@, pp.maxBal),$
 78 $![q][p].maxVVal = Max(@, pp.maxVVal),$
 79 $![q][p].maxVVal = \text{IF } state[q][p].maxVVal < pp.maxVVal$
 80 $\quad \text{THEN } pp.maxVVal \text{ ELSE } @,$
 81 $![q][q].maxBal = Max(@, pp.maxBal),$
 82 $![q][q].maxVVal = \text{IF } state[q][q].maxBal \leq pp.maxVVal$
 83 $\quad \text{THEN } pp.maxVVal \text{ ELSE } @, \text{ make promise}$
 84 $![q][q].maxVVal = \text{IF } \vee state[q][q].maxBal < pp.maxVVal$
 85 $\quad \text{OneVote}$
 86 $\quad \vee state[q][q].maxBal = pp.maxVVal \wedge @ = None$
 87 $\quad \text{THEN } pp.maxVVal \text{ ELSE } @] \text{ accept}$
 $q \in Participant$ receives and processes a message in $Message$.
 91 $OnMessage(q) \triangleq$
 92 $\exists m \in msgs :$
 93 $\wedge q \in m.to$
 94 $\wedge \text{LET } p \triangleq m.from$
 95 $\text{IN } UpdateState(q, p, m.state[p])$
 96 $\wedge \text{IF } \vee m.state[q].maxBal < state'[q][q].maxBal$
 97 $\quad \vee m.state[q].maxVVal < state'[q][q].maxVVal$
 98 $\quad \text{THEN } Send([from \mapsto q, to \mapsto \{m.from\}, state \mapsto state'[q]])$
 99 $\text{ELSE UNCHANGED } msgs$
 $p \in Participant$ starts the accept phase by issuing the ballot $b \in Ballot$ with value $v \in Value$.
 104 $Accept(p, b, v) \triangleq$
 105 $\wedge state[p][p].maxVVal \neq b \text{ for OneVote; TODO: too strong?}$
 106 $(i.e., state[p][p].maxVVal = b \Rightarrow v = state[p][p].maxVVal)$
 107 $(it \text{ ensures } \forall p \in Participant, b \in Ballot : Accept(p, b, _) \text{ only once})$
 108 $\wedge \exists Q \in Quorum : \forall q \in Q : state[p][q].maxBal = b$
 109 $\wedge \forall q \in Participant : state[p][q].maxVVal = -1 \text{ free to pick its own value}$
 110 $\vee \exists q \in Participant : v \text{ is the value with the highest } maxVVal$
 111 $\wedge state[p][q].maxVVal = v$
 112 $\wedge \forall r \in Participant : state[p][q].maxVVal \geq state[p][r].maxVVal$

```

113       $\wedge state' = [state \text{ EXCEPT } ![p][p].maxVBal = b, ![p][p].maxVVal = v]$ 
114       $\wedge Send([from \mapsto p, to \mapsto Participant, state \mapsto state'[p]])$ 
115  |-----|
116   $Next \triangleq \exists p \in Participant : \vee OnMessage(p)$ 
117                                      $\vee \exists b \in Ballot : \vee Prepare(p, b)$ 
118                                      $\vee \exists v \in Value : Accept(p, b, v)$ 
119   $Spec \triangleq Init \wedge \Box[Next]_{vars}$ 
120  |-----|
121   $ChosenP(p) \triangleq$  the set of values chosen by  $p \in Participant$ 
122       $\{v \in Value : \exists b \in Ballot :$ 
123           $\exists Q \in Quorum : \forall q \in Q : \wedge state[p][q].maxVBal = b$ 
124           $\wedge state[p][q].maxVVal = v\}$ 
125   $chosen \triangleq \text{UNION } \{ChosenP(p) : p \in Participant\}$ 
127   $Consistency \triangleq Cardinality(chosen) \leq 1$ 
128  THEOREM  $Spec \Rightarrow \Box Consistency$ 
129  |-----|
    \ * Modification History
    \ * Last modified Wed Jul 31 16:36:13 CST 2019 by hengxin
    \ * Last modified Mon Jul 22 13:59:15 CST 2019 by pure_
    \ * Last modified Mon Jun 03 21:26:09 CST 2019 by stary
    \ * Last modified Wed May 09 21:39:31 CST 2018 by dell
    \ * Created Mon Apr 23 15:47:52 GMT + 08:00 2018 by pure_

```