```
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
EXTENDS Integers, FiniteSets
Max(m, n) \stackrel{\Delta}{=} \text{ if } m > n \text{ THEN } m \text{ ELSE } n
Injective(f) \stackrel{\triangle}{=} \forall a, b \in DOMAIN \ f: (a \neq b) \Rightarrow (f[a] \neq f[b])
CONSTANTS
     Participant,
                         the set of partipants
     Value
                         the set of possible input values for Participant to propose
None \triangleq CHOOSE \ b: b \notin Value
NP \stackrel{\Delta}{=} Cardinality(Participant) number of p \in Participants
Quorum \triangleq \{Q \in \text{SUBSET } Participant : Cardinality(Q) * 2 \ge NP + 1\}
Assume QuorumAssumption \triangleq
     \land \quad \forall \ Q \in Quorum : Q \subseteq Participant
     \land \quad \forall Q1, Q2 \in Quorum : Q1 \cap Q2 \neq \{\}
Ballot \triangleq Nat
PIndex \stackrel{\triangle}{=} CHOOSE f \in [Participant \rightarrow 1 ... NP] : Injective(f)
Bals(p) \stackrel{\Delta}{=} \{b \in Ballot : b\%NP = PIndex[p] - 1\} allocate ballots for each p \in Participant
State \triangleq [maxBal : Ballot \cup \{-1\}, \\ maxVBal : Ballot \cup \{-1\}, \ maxVVal : Value \cup \{None\}]
InitState \triangleq [maxBal \mapsto -1, maxVBal \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 : SUBSET Participant, state : [Participant \rightarrow State]]
VARIABLES
                state[p][q]: the state of q \in Participant from the view of p \in Participant
     state,
     msqs
                the set of messages that have been sent
vars \stackrel{\Delta}{=} \langle state, msgs \rangle
TypeOK \triangleq
     \land state \in [Participant \rightarrow [Participant \rightarrow State]]
     \land msgs \subseteq Message
```

MODULE TPaxos

```
Send(m) \stackrel{\triangle}{=} msgs' = msgs \cup \{m\}
Init \; \stackrel{\scriptscriptstyle \Delta}{=} \;
     \land state = [p \in Participant \mapsto [q \in Participant \mapsto InitState]]
     \land msqs = \{\}
p \in Participant starts the prepare phase by issuing a ballot b \in Ballot.
Prepare(p, b) \triangleq
     \land state[p][p].maxBal < b
     \land b \in Bals(p)
     \land state' = [state \ EXCEPT \ ![p][p].maxBal = b]
     \land 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 \ EXCEPT
                 ![q][p].maxBal = Max(@, pp.maxBal),
                 ![q][p].maxVBal = Max(@, pp.maxVBal),
                 ![q][p].maxVVal = IF state[q][p].maxVBal < pp.maxVBal
                                         THEN pp.maxVVal ELSE @,
                 ![q][q].maxBal = Max(@, pp.maxBal),
                 ![q][q].maxVBal = IF state[q][q].maxBal \leq pp.maxVBal
                                         THEN pp.maxVBal ELSE @,
                                                                              make promise
                 ![q][q].maxVVal = IF state[q][q].maxBal \leq pp.maxVBal
                                         THEN pp.maxVVal ELSE @ accept
q \in Participant receives and processes a message in Message.
OnMessage(q) \stackrel{\Delta}{=}
    \exists m \in msgs:
        \land q \in m.to
        \wedge LET p \triangleq m.from
          IN UpdateState(q, p, m.state[p])
        \land IF \lor m.state[q].maxBal < state'[q][q].maxBal
              \lor m.state[q].maxVBal < state'[q][q].maxVBal
           THEN Send([from \mapsto q, to \mapsto \{m.from\}, state \mapsto state'[q]])
           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
     \land b \in Bals(p)
     \wedge state[p][p].maxBal = b corresponding the first conjunction in Voting
     \wedge state[p][p].maxVBal < b corresponding the second conjunction in Voting
     \land \exists \ Q \in Quorum: pick the value from the quorum
        \land \forall q \in Q : state[p][q].maxBal = b
        \land \lor \forall q \in Q : state[p][q].maxVBal = -1 free to pick its own value
```

```
\vee \exists q \in Q: v is the value with the highest maxVBal in the quorum
                   \wedge state[p][q].maxVVal = v
                   \land \forall r \in Q : state[p][q].maxVBal \ge state[p][r].maxVBal
       choose the value from all the local state
      \land \  \, \forall \,\, q \in \mathit{Participant} \,\,: \,\, \mathit{state}[p][q].\mathit{maxVBal} \,\,= \,\, -1 \, \backslash \,\, \ast \,\, \mathsf{free} \,\, \mathsf{to} \,\, \mathsf{pick} \,\, \mathsf{its} \,\, \mathsf{own} \,\, \mathsf{value}
      \vee \exists \ q \in \mathit{Participant}: \ \setminus *v \text{ is the value with the highest } \mathit{maxVBal}
          \wedge \; state[p][q].maxVVal = v
          \land \forall \, r \in \mathit{Participant} \colon \mathit{state}[p][q].\mathit{maxVBal} \geq \mathit{state}[p][r].\mathit{maxVBal}
      \wedge state' = [state \ EXCEPT \ ![p][p].maxVBal = b,
      ![p][p].maxVVal = v] \\ \land Send([from \mapsto p, \ to \mapsto Participant, \ state \mapsto state'[p]])
Next \triangleq \exists p \in Participant : \lor OnMessage(p)
                                          \vee \exists b \in Ballot : \vee Prepare(p, b)
                                                                  \forall \exists v \in Value : Accept(p, b, v)
Spec \stackrel{\triangle}{=} Init \wedge \Box [Next]_{vars}
ChosenP(p) \stackrel{\Delta}{=} the set of values chosen by p \in Participant
     \{v \in Value : \exists b \in Ballot : a
                            \exists Q \in Quorum : \forall q \in Q : \land state[p][q].maxVBal = b
                                                                    \land state[p][q].maxVVal = v
chosen \stackrel{\triangle}{=} UNION \{ChosenP(p) : p \in Participant\}
Consistency \triangleq Cardinality(chosen) < 1
THEOREM Spec \Rightarrow \Box Consistency
\ ∗ Modification History
\ * Last modified Sun Sep 01 12:54:07 CST 2019 by pure_
\ * Last modified Wed Jul 31 15:00:12 CST 2019 by hengxin
\ * Last modified Mon Jun 03 21:26:09 CST 2019 by stary
\ * Last modified Wed May 09 21:39:31 CST 2018 by dell
```