MODULE *TPaxos*

Specification of the consensus protocol in *PaxosStore*.

See [*PaxosStore*@*VLDB*2017](*https* : *//www.vldb.org/pvldb/vol*10/*p*1730 − *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*

$Max(m,\, n) \triangleq$ IF $m > n$ THEN m ELSE n
$Injective(f) \triangleq \forall\, a,\, b \in$ 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 $b : b \notin Value$
$NP \triangleq Cardinality(Participant)$ number of $p \in$ Participants

$Quorum \triangleq \{Q \in$ SUBSET $Participant : Cardinality(Q) * 2 \geq NP + 1\}$
ASSUME $QuorumAssumption \triangleq$
$\wedge \quad \forall\, Q \in Quorum : Q \subseteq Participant$
$\wedge \quad \forall\, Q1,\, Q2 \in Quorum : Q1 \cap Q2 \neq \{\}$

$Ballot \triangleq Nat$

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

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

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

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 $Participant,\ state : [Participant \to State]]$

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

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

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

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

$Init \triangleq$
$\quad \land state = [p \in Participant \mapsto [q \in Participant \mapsto InitState]]$
$\quad \land msgs = \{\}$

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

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

$q \in Participant$ 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.

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

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

$OnMessage(q) \triangleq$
$\quad \exists\, m \in msgs :$
$\qquad \land \ q \in m.to$
$\qquad \land \text{LET } p \triangleq m.from$
$\qquad\quad \text{IN } UpdateState(q, p, m.state[p])$
$\qquad \land \text{IF } \lor m.state[q].maxBal < state'[q][q].maxBal$
$\qquad\qquad \lor m.state[q].maxVBal < state'[q][q].maxVBal$
$\qquad\quad \text{THEN } Send([from \mapsto q, to \mapsto \{m.from\}, state \mapsto state'[q]])$
$\qquad\quad \text{ELSE UNCHANGED } msgs$

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

$Accept(p, b, v) \triangleq$
$\quad \land \ b \in Bals(p)$
$\quad \land \ state[p][p].maxVBal \neq b$
$\quad \land \exists\, Q \in Quorum :$ pick the value from the quorum
$\qquad \land \forall\, q \in Q : state[p][q].maxBal = b$
$\qquad \land \ \lor \forall\, q \in Q : state[p][q].maxVBal = -1$ free to pick its own value
$\qquad\quad \lor \exists\, q \in Q :$ v is the value with the highest $maxVBal$ in the quorum

$\land\ state[p][q].maxVVal = v$
$\land\ \forall\, r \in Q : state[p][q].maxVBal \geq state[p][r].maxVBal$

choose the value from all the local state

$\land\ \lor\ \forall\, q \in Participant : state[p][q].maxVBal = -1 \setminus *$ free to pick its own value
$\lor\ \exists\, q \in Participant : \setminus * v$ is the value with the highest $maxVBal$
$\land\ state[p][q].maxVVal = v$
$\land\ \forall\, r \in Participant$: $state[p][q].maxVBal \geq state[p][r].maxVBal$

$\land\ state' = [state \text{ EXCEPT } ![p][p].maxVBal = b,$
$\qquad ![p][p].maxVVal = v]$
$\land\ Send([from \mapsto p,\ to \mapsto Participant,\ state \mapsto state'[p]])$

$Next \triangleq \exists\, p \in Participant : \lor\ OnMessage(p)$
$\qquad \lor\ \exists\, b \in Ballot : \lor\ Prepare(p,\ b)$
$\qquad\qquad \lor\ \exists\, v \in Value : Accept(p,\ b,\ v)$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$ChosenP(p) \triangleq$ the set of values chosen by $p \in Participant$
$\{v \in Value : \exists\, b \in Ballot :$
$\qquad \exists\, Q \in Quorum : \forall\, q \in Q : \land\ state[p][q].maxVBal = b$
$\qquad\qquad \land\ state[p][q].maxVVal = v\}$

$chosen \triangleq \text{UNION } \{ChosenP(p) : p \in Participant\}$

$Consistency \triangleq Cardinality(chosen) \leq 1$

THEOREM $Spec \Rightarrow \Box Consistency$

$\setminus *$ Modification History
$\setminus *$ Last modified *Wed Aug* 28 10:34:30 *CST* 2019 by *pure_*
$\setminus *$ Last modified *Wed Jul* 31 15:00:12 *CST* 2019 by *hengxin*
$\setminus *$ Last modified *Mon Jun* 03 21:26:09 *CST* 2019 by *stary*
$\setminus *$ Last modified *Wed* May 09 21:39:31 *CST* 2018 by dell
$\setminus *$ Created *Mon Apr* 23 15:47:52 *GMT* + 08:00 2018 by *pure_*