MODULE *FastPaxos*

This is a simplified specification of *Leslie Lamport*'s Fast *Paxos* protocol. The following papers, Fast *Paxos* by *Leslie Lamport* and Fast *Paxos* Made Easy: Theory and Implementation by *Zhao Wenbing* was referenced in writing this specification.

This specification was written by *Lim Ngian Xin Terry* & *Gaurav Gandhi*.

The following assumptions are made in this simplified specification.

1. There is a unique coordinator in the system. Therefore, Phase $1a$ and $1b$ can be omitted.
2. All agents in the system can communicate with one another.
3. Agents must have some stable storage that survives failure ands restart. An agent restores its state from stable storage when it restarts, so the failure of an agent is indistinguishable from its simply pausing. There is thus no need to model failures explicitly.

EXTENDS TLC, $Naturals$, $FiniteSets$, $Integers$

CONSTANTS any, $none$, $Replicas$, $Values$, $Ballots$, $Quorums$
CONSTANTS $FastQuorums$, $FastBallots$

VARIABLES $messages$ Set of all messages sent.
VARIABLES $decision$ Decided value of an acceptor.
VARIABLES $maxBallot$ Maximum ballot an acceptor has seen.
VARIABLES $maxVBallot$ Maximum ballot an acceptor has accepted.
VARIABLES $maxValue$ Maximum value an acceptor has accepted.
VARIABLES $cValue$ Value chosen by coordinator.

INSTANCE $Paxos$

$ClassicBallots \triangleq Ballots \setminus FastBallots$ The set of ballots of classic rounds.

$$
\begin{aligned}
&FastAssume \triangleq \\
&\quad \land \forall\, q \in FastQuorums : q \subseteq Replicas \\
&\quad \land \forall\, q,\, r \in FastQuorums : q \cap r \neq \{\} \\
&\quad \land \forall\, q \in FastQuorums : (3 * Cardinality(Replicas)) \div 4 \leq Cardinality(q) \\
&\quad \land \forall\, q \in Quorums : \forall\, r,\, s \in FastQuorums : q \cap r \cap s \neq \{\}
\end{aligned}
$$

ASSUME $PaxosAssume \land FastAssume$

$$IsMajorityValue(M,\, v) \triangleq Cardinality(M) \div 2 < Cardinality(\{m \in M : m.value = v\})$$

Phase $2a$ (Fast):

The coordinator starts a fast round by sending a $P2a$ "Any" message, if no other values has been proposed before.

$$
\begin{aligned}
&FastAny \triangleq \\
&\quad \land \text{UNCHANGED } \langle decision,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle \\
&\quad \land \exists f \in FastBallots : \\
&\qquad \land SendMessage([type \mapsto \text{“P2a”}, \\
&\qquad\qquad\qquad\qquad ballot \mapsto f, \\
&\qquad\qquad\qquad\qquad value \mapsto any])
\end{aligned}
$$

Phase 2b (Fast):

Acceptors can reply to a $P2a$ "Any" message with a $P2b$ message containing their proposed value.

$$
\begin{array}{l}
\mathit{FastPropose} \triangleq \\
\quad \land \text{UNCHANGED } \langle \mathit{decision}, \mathit{cValue} \rangle \\
\quad \land \exists\, a \in \mathit{Replicas},\ m \in \mathit{p2aMessages},\ v \in \mathit{Values} : \\
\qquad \land\ m.\mathit{value} = \mathit{any} \\
\qquad \land\ \mathit{maxBallot}[a] \leq m.\mathit{ballot} \\
\qquad \land\ \mathit{maxValue}[a] = \mathit{none} \lor \mathit{maxValue}[a] = v \\
\qquad \land\ \mathit{maxBallot}' = [\mathit{maxBallot} \text{ EXCEPT } ![a] = m.\mathit{ballot}] \\
\qquad \land\ \mathit{maxVBallot}' = [\mathit{maxVBallot} \text{ EXCEPT } ![a] = m.\mathit{ballot}] \\
\qquad \land\ \mathit{maxValue}' = [\mathit{maxValue} \text{ EXCEPT } ![a] = v] \\
\qquad \land\ \forall\, n \in \mathit{p2bMessages} : \neg(n.\mathit{ballot} = m.\mathit{ballot} \land n.\mathit{acceptor} = a) \\
\qquad \land\ \mathit{SendMessage}([\mathit{type} \mapsto \text{“P2b”}, \\
\qquad\qquad\qquad\qquad \mathit{ballot} \mapsto m.\mathit{ballot}, \\
\qquad\qquad\qquad\qquad \mathit{acceptor} \mapsto a, \\
\qquad\qquad\qquad\qquad \mathit{value} \mapsto v])
\end{array}
$$

A value is chosen if a fast quorum of acceptors proposed that value in a fast round.

$$
\begin{array}{l}
\mathit{FastDecide} \triangleq \\
\quad \land \text{UNCHANGED } \langle \mathit{messages}, \mathit{maxBallot}, \mathit{maxVBallot}, \mathit{maxValue}, \mathit{cValue} \rangle \\
\quad \land \exists\, b \in \mathit{FastBallots},\ q \in \mathit{FastQuorums} : \\
\qquad \text{LET } M \triangleq \{m \in \mathit{p2bMessages} : m.\mathit{ballot} = b \land m.\mathit{acceptor} \in q\} \\
\qquad\qquad\ V \triangleq \{w \in \mathit{Values} : \exists\, m \in M : w = m.\mathit{value}\} \\
\qquad \text{IN } \land \forall\, a \in q : \exists\, m \in M : m.\mathit{acceptor} = a \\
\qquad\qquad\ \land 1 = \mathit{Cardinality}(V) \\
\qquad\qquad\ \land \exists\, m \in M : \mathit{decision}' = m.\mathit{value}
\end{array}
$$

Phase 2a (Classic)

If more than one value has been proposed, the collision is resolved using the following rules:

1. If the proposals contain different values, a value must be selected if the majority of acceptors in the fast quorum have casted a vote for that value.

2. Otherwise, the coordinator is free to select any value.

$$
\begin{array}{l}
\mathit{ClassicAccept} \triangleq \\
\quad \land \text{UNCHANGED } \langle \mathit{decision}, \mathit{maxBallot}, \mathit{maxVBallot}, \mathit{maxValue} \rangle \\
\quad \land \exists\, b \in \mathit{ClassicBallots},\ f \in \mathit{FastBallots},\ q \in \mathit{FastQuorums},\ v \in \mathit{Values} : \\
\qquad \land\ f < b \quad \text{There was a fast round before this classic round.} \\
\qquad \land\ \mathit{cValue} = \mathit{none} \lor \mathit{cValue} = v \\
\qquad \land\ \mathit{cValue}' = v \\
\qquad \land\ \forall\, m \in \mathit{p2aMessages} : m.\mathit{ballot} \neq b \\
\qquad \land \text{LET } M \triangleq \{m \in \mathit{p2bMessages} : m.\mathit{ballot} = f \land m.\mathit{acceptor} \in q\} \\
\qquad\qquad\ V \triangleq \{w \in \mathit{Values} : \exists\, m \in M : w = m.\mathit{value}\} \\
\qquad\quad \text{IN } \land \forall\, a \in q : \exists\, m \in M : m.\mathit{acceptor} = a \\
\qquad\qquad\ \land 1 < \mathit{Cardinality}(V) \quad \text{Collision occured.}
\end{array}
$$

$\land$ IF $\exists\, w \in V : IsMajorityValue(M,\, w)$
  THEN $IsMajorityValue(M,\, v)$ Choose majority in quorum.
  ELSE $v \in V$ Choose any.
$\land\, SendMessage([type \mapsto$ "P2a",
$ballot \mapsto b,$
$value \mapsto v])$

Phase 2b (Classic)

Same as in *Paxos*.

$ClassicAccepted \triangleq$
  $\land$ UNCHANGED $\langle cValue \rangle$
  $\land\, PaxosAccepted$

Same as in *Paxos*.

$ClassicDecide \triangleq$
  $\land$ UNCHANGED $\langle messages,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle$
  $\land\, \exists\, b \in ClassicBallots,\, q \in Quorums :$
    LET $M \triangleq \{m \in p2bMessages : m.ballot = b \land m.acceptor \in q\}$
    IN $\land\, \forall\, a \in q : \exists\, m \in M : m.acceptor = a$
       $\land\, \exists\, m \in M : decision' = m.value$

$FastTypeOK \triangleq \land\, PaxosTypeOK$
  $\land\, cValue \in Values \cup \{none\}$

$FastInit \triangleq \land\, PaxosInit$
  $\land\, cValue = none$

$FastNext \triangleq \lor\, FastAny$
  $\lor\, FastPropose$
  $\lor\, FastDecide$
  $\lor\, ClassicAccept$
  $\lor\, ClassicAccepted$
  $\lor\, ClassicDecide$

$FastSpec \triangleq \land\, FastInit$
  $\land\, \Box[FastNext]_{\langle messages,\, decision,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle}$
  $\land\, \mathrm{SF}_{\langle messages,\, decision,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle}(FastDecide)$
  $\land\, \mathrm{SF}_{\langle messages,\, decision,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle}(ClassicDecide)$

Only proposed values can be learnt.

$FastNontriviality \triangleq \lor\, decision = none$
  $\lor\, \exists\, m \in p2bMessages : m.value = decision \land m.ballot \in FastBallots$

$FastSafetyProperty \triangleq \land\, \Box[FastNontriviality]_{\langle messages,\, decision,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle}$
  $\land\, \Box[PaxosConsistency]_{\langle messages,\, decision,\, maxBallot,\, maxVBallot,\, maxValue,\, cValue \rangle}$

$FastSymmetry \triangleq PaxosSymmetry$

THEOREM $FastSpec \Rightarrow PaxosSpec$