MODULE *JustInTimePaxos*

EXTENDS *Naturals*, *Sequences*, *FiniteSets*, *TLC*

The set of *Paxos* replicas
CONSTANT *Replicas*

The set of *Paxos* clients
CONSTANT *Clients*

The maximum clock interval
CONSTANT *MaxClockInterval*

An empty value
CONSTANT *Nil*

The set of values to write
CONSTANT *Values*

Client request/response types+
CONSTANTS
    ClientRequest,
    ClientResponse

Server request/response types
CONSTANTS

	SlotLookup
	SlotLookupRep
	GapCommit
	GapCommitRep
ViewChangeRequest,	*ViewChangeReq*
ViewChangeResponse,	*ViewChange*
StartViewRequest,	*StartView*
SyncPrepareRequest,	*SyncPrepare*
SyncPrepareResponse,	*SyncPrepareRep*
SyncCommitRequest	*SyncCommit*

Replica roles
CONSTANTS
    Normal,
    ViewChanging,
    Recovering

VARIABLE *replicas*

VARIABLE *messages*

$messageVars \triangleq \langle messages \rangle$

VARIABLE $globalTime$

VARIABLE $time$

VARIABLE $requestID$

VARIABLE $responses$

VARIABLE $commits$

$clientVars \triangleq \langle globalTime,\ time \rangle$

VARIABLE $status$

VARIABLE $viewID$

VARIABLE log

$replicaVars \triangleq \langle status,\ viewID,\ log \rangle$

$vars \triangleq \langle messageVars,\ clientVars,\ replicaVars \rangle$

$InitMessageVars \triangleq$
$\quad \land\ messages = \{\}$

$InitClientVars \triangleq$
$\quad \land\ globalTime = 0$
$\quad \land\ time = [c \in Clients \mapsto 0]$
$\quad \land\ requestID = [c \in Clients \mapsto 0]$
$\quad \land\ responses = [c \in Clients \mapsto [r \in Replicas \mapsto [s \in \{\} \mapsto [index \mapsto 0,\ value \mapsto Nil,\ checksum \mapsto Nil]]]]$
$\quad \land\ commits = [c \in Clients \mapsto [s \in \{\} \mapsto [index \mapsto 0,\ value \mapsto Nil,\ checksum \mapsto Nil]]]$

$InitReplicaVars \triangleq$
$\quad \land\ replicas = Seq(Replicas)$
$\quad \land\ status\ \ = [r \in Replicas \mapsto Normal]$
$\quad \land\ viewID = [r \in Replicas \mapsto 0]$
$\quad \land\ log = [r \in Replicas \mapsto \langle\rangle]$

$Init \triangleq$
$\quad \land\ InitMessageVars$
$\quad \land\ InitClientVars$
$\quad \land\ InitReplicaVars$

Helpers

$Quorums \triangleq \{r \in \text{SUBSET}\ Replicas : Cardinality(r) * 2 > Cardinality(Replicas)\}$

$Primary(v) \triangleq replicas[v\%Len(replicas)] + (\text{IF } v \geq Len(replicas) \text{ THEN } 1 \text{ ELSE } 0)$

$IsPrimary(r) \triangleq Primary(viewID[r]) = r$

Messaging helpers

$SendMessage(m) \triangleq messages' = messages \cup \{m\}$

$SendMessages(ms) \triangleq messages' = messages \cup ms$

$AdvanceTime(c) \triangleq$
$\quad \land globalTime' = globalTime + 1$
$\quad \land \text{IF } time[c] < globalTime \land globalTime - time[c] > MaxClockInterval \text{ THEN}$
$\qquad time' = [time \text{ EXCEPT } ![c] = globalTime' - MaxClockInterval]$
$\quad \text{ELSE}$
$\qquad time' = [time \text{ EXCEPT } ![c] = time[c] + 1]$

$CurrentTime(c) \triangleq time'[c]$

$SendClientRequest(c, v) \triangleq$
$\quad \land AdvanceTime(c)$
$\quad \land requestID' = [requestID \text{ EXCEPT } ![c] = requestID[c] + 1]$
$\quad \land SendMessages(\{[source \mapsto c,$
$\qquad\qquad target \mapsto r,$
$\qquad\qquad type \mapsto ClientRequest,$
$\qquad\qquad requestID \mapsto requestID'[c],$
$\qquad\qquad value \mapsto v,$
$\qquad\qquad timestamp \mapsto CurrentTime(c)] : r \in Replicas\})$
$\quad \land \text{UNCHANGED } \langle\rangle$

$IsCommitted(acks) \triangleq$
$\quad \exists\, msgs \in \text{SUBSET } acks :$
$\qquad \land \{m.source : m \in msgs\} \in Quorums$
$\qquad \land \exists\, m1 \in msgs : \forall\, m2 \in msgs : m1.viewID = m2.viewID \land m1.checksum \cap m2.checksum = \{\}$
$\qquad \land \exists\, m \in msgs : m.primary$

$HandleClientResponse(c, r, m) \triangleq$
$\quad \land m.requestID \notin commits[c]$
$\quad \land \lor \land m.requestID \notin responses[c][r]$
$\qquad\quad \land responses' = [responses \text{ EXCEPT } ![c] = [responses[c] \text{ EXCEPT } ![r] = responses[c][r] \,@@\, (m.requestI$
$\qquad \lor \land m.requestID \in responses[c][r]$
$\qquad\quad$ Do not overwrite a response from a newer view
$\qquad\quad \land responses[c][r][m.requestID].viewID \leq m.viewID$
$\qquad\quad \land responses' = [responses \text{ EXCEPT } ![c] = [responses[c] \text{ EXCEPT } ![r] = [responses[c][r] \text{ EXCEPT } ![m.r$

$\land$ LET $committed \triangleq IsCommitted(\{responses'[c][x] : x \in Replicas\})$
IN
$\lor \land committed$
$\land commits' = [commits$ EXCEPT $![c] = commits[c] \cup \{m\}]$
$\lor \land \neg committed$
$\land$ UNCHANGED $\langle commits \rangle$
$\land$ UNCHANGED $\langle\rangle$

Server request/response handling

$HandleClientRequest(r, c, m) \triangleq$
$\land status[r] = Normal$
$\land \lor \land \lor Len(log[r]) = 0$
$\lor \land Len(log[r]) \neq 0$
$\land m.timestamp > log[r][Len(log[r])].timestamp$
$\land log' = [log$ EXCEPT $![r] = Append(log[r], m)]$
$\land SendMessage([source \mapsto r,$
$target \mapsto c,$
$type \mapsto ClientResponse,$
$requestID \mapsto m.requestID,$
$viewID \mapsto viewID[r],$
$primary \mapsto IsPrimary(r),$
$index \mapsto Len(log'[r]),$
$checksum \mapsto \{log'[r][i].timestamp : i \in$ DOMAIN $log'[r]\},$
$succeeded \mapsto$ TRUE$])$
$\lor \land Len(log[r]) \neq 0$
$\land m.timestamp \leq log[r][Len(log[r])].timestamp$
$\land SendMessage([source \mapsto r,$
$target \mapsto c,$
$type \mapsto ClientResponse,$
$requestID \mapsto m.requestID,$
$view \mapsto viewID[r],$
$primary \mapsto IsPrimary(r),$
$index \mapsto Len(log[r]),$
$checksum \mapsto \{log[r][i].timestamp : i \in$ DOMAIN $log[r]\},$
$succeeded \mapsto$ FALSE$])$
$\land$ UNCHANGED $\langle\rangle$

$HandleViewChangeRequest(r, s, m) \triangleq$
$\land$ UNCHANGED $\langle\rangle$

$HandleViewChangeResponse(r, s, m) \triangleq$
$\land$ UNCHANGED $\langle\rangle$

$HandleStartViewRequest(r, s, m) \triangleq$

$\land$ UNCHANGED $\langle\rangle$

$HandleSyncPrepareRequest(r,\ s,\ m) \triangleq$
$\land$ UNCHANGED $\langle\rangle$

$HandleSyncPrepareResponse(r,\ s,\ m) \triangleq$
$\land$ UNCHANGED $\langle\rangle$

$HandleSyncCommitRequest(r,\ s,\ m) \triangleq$
$\land$ UNCHANGED $\langle\rangle$

$ReceiveMessage(m) \triangleq$
$\lor \land m.type = ClientRequest$
$\land HandleClientRequest(m.target,\ m.source,\ m)$
$\lor \land m.type = ClientResponse$
$\land HandleClientResponse(m.target,\ m.source,\ m)$
$\lor \land m.type = ViewChangeRequest$
$\land HandleViewChangeRequest(m.target,\ m.source,\ m)$
$\lor \land m.type = ViewChangeResponse$
$\land HandleViewChangeResponse(m.target,\ m.source,\ m)$
$\lor \land m.type = StartViewRequest$
$\land HandleStartViewRequest(m.target,\ m.source,\ m)$
$\lor \land m.type = SyncPrepareRequest$
$\land HandleSyncPrepareRequest(m.target,\ m.source,\ m)$
$\lor \land m.type = SyncPrepareResponse$
$\land HandleSyncPrepareResponse(m.target,\ m.source,\ m)$
$\lor \land m.type = SyncCommitRequest$
$\land HandleSyncCommitRequest(m.target,\ m.source,\ m)$

$Next \triangleq$
$\lor \exists\, c \in Clients :$
$\exists\, v \in Values :$
$SendClientRequest(c,\ v)$
$\lor \exists\, m \in messages : ReceiveMessage(m)$

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

---

* Modification History
* Last modified Sun *Sep* 20 16:51:08 *PDT* 2020 by *jordanhalterman*
* Created *Fri Sep* 18 22:45:21 *PDT* 2020 by *jordanhalterman*