

---

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*

*Client request/response types*  
 CONSTANTS  
     *WriteRequest,*  
     *WriteResponse,*  
     *ReadRequest,*  
     *ReadResponse*

*Server request/response types*  
 CONSTANTS  
     *ViewChangeRequest,*  
     *ViewChangeResponse,*  
     *StartViewRequest*

*Replica roles*  
 CONSTANTS  
     *NormalStatus,*  
     *ViewChangeStatus,*  
     *RecoveringStatus*

---

VARIABLE *replicas*

*globalVars*  $\triangleq$   $\langle \textit{replicas} \rangle$

VARIABLE *messages*

*messageVars*  $\triangleq$   $\langle \textit{messages} \rangle$

VARIABLE *globalTime*

VARIABLE *time*

VARIABLE *requestID*

VARIABLE *responses*

VARIABLE *writes*

VARIABLE *reads*

*clientVars*  $\triangleq \langle globalTime, time, requestID, responses, writes, reads \rangle$

VARIABLE *status*

VARIABLE *log*

VARIABLE *viewID*

VARIABLE *lastNormalView*

VARIABLE *viewChanges*

*replicaVars*  $\triangleq \langle status, log, viewID, lastNormalView, viewChanges \rangle$

VARIABLE *transitions*

*vars*  $\triangleq \langle globalVars, messageVars, clientVars, replicaVars, transitions \rangle$

---

#### Helpers

RECURSIVE *SeqFromSet*( $\_$ )

*SeqFromSet*( $S$ )  $\triangleq$

IF  $S = \{\}$  THEN  $\langle \rangle$

ELSE LET  $x \triangleq \text{CHOOSE } x \in S : \text{TRUE}$

IN  $\langle x \rangle \circ \text{SeqFromSet}(S \setminus \{x\})$

*Max*( $s$ )  $\triangleq \text{CHOOSE } x \in s : \forall y \in s : x \geq y$

*IsQuorum*( $s$ )  $\triangleq \text{Cardinality}(s) * 2 \geq \text{Cardinality}(\text{Replicas})$

*Quorums*  $\triangleq \{r \in \text{SUBSET } \text{Replicas} : \text{IsQuorum}(r)\}$

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

*IsPrimary*( $r$ )  $\triangleq \text{Primary}(\text{viewID}[r]) = r$

---

#### Messaging helpers

*Sends*( $ms$ )  $\triangleq \text{messages}' = \text{messages} \cup ms$

*Send*( $m$ )  $\triangleq \text{Sends}(\{m\})$

*Replies*( $req, resps$ )  $\triangleq \text{messages}' = (\text{messages} \cup \text{resps}) \setminus \{req\}$

$Reply(req, resp) \triangleq Replies(req, \{resp\})$

$Discard(m) \triangleq messages' = messages \setminus \{m\}$

$AdvanceTime(c) \triangleq$

$\wedge globalTime' = globalTime + 1$

$\wedge \text{IF } time[c] < globalTime \wedge globalTime - time[c] > MaxClockInterval \text{ THEN}$   
 $time' = [time \text{ EXCEPT } ![c] = globalTime' - MaxClockInterval]$

$\text{ELSE}$

$time' = [time \text{ EXCEPT } ![c] = time[c] + 1]$

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

$Write(c) \triangleq$

$\wedge AdvanceTime(c)$

$\wedge requestID' = [requestID \text{ EXCEPT } ![c] = requestID[c] + 1]$

$\wedge Sends(\{[src \mapsto c,$   
 $dest \mapsto r,$   
 $type \mapsto WriteRequest,$   
 $requestID \mapsto requestID'[c],$   
 $timestamp \mapsto CurrentTime(c)] : r \in Replicas\})$

$\wedge \text{UNCHANGED } \langle globalVars, replicaVars, responses, writes, reads \rangle$

$Read(c) \triangleq$

$\wedge requestID' = [requestID \text{ EXCEPT } ![c] = requestID[c] + 1]$

$\wedge Sends(\{[src \mapsto c,$   
 $dest \mapsto r,$   
 $type \mapsto ReadRequest,$   
 $requestID \mapsto requestID'[c]] : r \in Replicas\})$

$\wedge \text{UNCHANGED } \langle globalVars, replicaVars, globalTime, time, responses, writes, reads \rangle$

$ChecksumsMatch(c1, c2) \triangleq$

$\wedge Len(c1) = Len(c2)$

$\wedge \neg \exists i \in \text{DOMAIN } c1 : c1[i] \neq c2[i]$

$IsCommitted(acks) \triangleq$

$\exists msgs \in \text{SUBSET } acks :$

$\wedge \{m.src : m \in msgs\} \in Quorums$

$\wedge \exists m1 \in msgs : \forall m2 \in msgs : m1.viewID = m2.viewID \wedge ChecksumsMatch(m1.checksum, m2.checksum)$

$\wedge \exists m \in msgs : m.primary$

$HandleWriteResponse(c, r, m) \triangleq$

$\wedge \neg \exists w \in writes[c] : w.requestID = m.requestID$

$\wedge \vee \wedge m.requestID \notin \text{DOMAIN } responses[c][r]$

$\wedge responses' = [responses \text{ EXCEPT } ![c] = [responses[c] \text{ EXCEPT } ![r] = responses[c][r] @@ (m.requestID$

$\wedge \text{UNCHANGED } \langle writes \rangle$

$$\begin{aligned}
& \vee \wedge m.requestID \in \text{DOMAIN } responses[c][r] \\
& \quad \text{Do not overwrite a response from a newer view} \\
& \wedge responses[c][r][m.requestID].viewID \leq m.viewID \\
& \wedge responses' = [responses \text{ EXCEPT } ![c] = [responses[c] \text{ EXCEPT } ![r] = [responses[c][r] \text{ EXCEPT } ![m. \\
& \wedge \text{LET } committed \triangleq IsCommitted(\{responses'[c][x][m.requestID] : x \in \{x \in Replicas : m.requestID} \\
& \quad \text{IN} \\
& \quad \vee \wedge committed \\
& \quad \quad \wedge writes' = [writes \text{ EXCEPT } ![c] = writes[c] \cup \{m\}] \\
& \quad \vee \wedge \neg committed \\
& \quad \quad \wedge \text{UNCHANGED } \langle writes \rangle \\
& \wedge Discard(m) \\
& \wedge \text{UNCHANGED } \langle globalVars, replicaVars, globalTime, time, requestID, reads \rangle \\
HandleReadResponse(c, r, m) & \triangleq \\
& \wedge \vee \wedge m.primary \\
& \quad \wedge m \notin reads[c] \\
& \quad \wedge reads' = [reads \text{ EXCEPT } ![c] = reads[c] \cup \{m\}] \\
& \vee \wedge \neg m.primary \\
& \quad \wedge \text{UNCHANGED } \langle reads \rangle \\
& \wedge Discard(m) \\
& \wedge \text{UNCHANGED } \langle globalVars, replicaVars, globalTime, time, requestID, responses, writes \rangle
\end{aligned}$$


---

Server request/response handling

$$\begin{aligned}
HandleWriteRequest(r, c, m) & \triangleq \\
& \wedge status[r] = NormalStatus \\
& \wedge \vee \wedge \vee Len(log[r]) = 0 \\
& \quad \vee \wedge Len(log[r]) \neq 0 \\
& \quad \wedge m.timestamp > log[r][Len(log[r])].timestamp \\
& \wedge \text{LET } checksum \triangleq Append([i \in \text{DOMAIN } log[r] \mapsto log[r][i].timestamp], m.timestamp) \\
& \quad entry \triangleq [client \mapsto c, \\
& \quad \quad requestID \mapsto m.requestID, \\
& \quad \quad timestamp \mapsto m.timestamp, \\
& \quad \quad checksum \mapsto checksum] \\
& \quad \text{IN} \\
& \quad \wedge log' = [log \text{ EXCEPT } ![r] = Append(log[r], entry)] \\
& \quad \wedge Reply(m, [src \mapsto r, \\
& \quad \quad dest \mapsto c, \\
& \quad \quad type \mapsto WriteResponse, \\
& \quad \quad requestID \mapsto m.requestID, \\
& \quad \quad viewID \mapsto viewID[r], \\
& \quad \quad primary \mapsto IsPrimary(r), \\
& \quad \quad index \mapsto Len(log'[r]), \\
& \quad \quad checksum \mapsto log'[r][Len(log'[r])].checksum,
\end{aligned}$$

$$\begin{aligned}
& \text{ succeeded } \mapsto \text{TRUE}] ) \\
\vee \wedge \text{ Len}(\log[r]) & \neq 0 \\
& \wedge m.\text{timestamp} \leq \log[r][\text{Len}(\log[r])].\text{timestamp} \\
& \wedge \text{Reply}(m, [\text{src} \mapsto r, \\
& \quad \text{dest} \mapsto c, \\
& \quad \text{type} \mapsto \text{WriteResponse}, \\
& \quad \text{requestID} \mapsto m.\text{requestID}, \\
& \quad \text{viewID} \mapsto \text{viewID}[r], \\
& \quad \text{primary} \mapsto \text{IsPrimary}(r), \\
& \quad \text{index} \mapsto \text{Len}(\log[r]), \\
& \quad \text{checksum} \mapsto \log[r][\text{Len}(\log[r])].\text{checksum}, \\
& \quad \text{succeeded} \mapsto \text{FALSE}]) \\
& \wedge \text{UNCHANGED } \langle \log \rangle \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{status}, \text{viewID}, \text{lastNormalView}, \text{viewChanges} \rangle \\
\text{HandleReadRequest}(r, c, m) & \triangleq \\
& \wedge \text{status}[r] = \text{NormalStatus} \\
& \wedge \text{Len}(\log[r]) > 0 \\
& \wedge \text{Reply}(m, [\text{src} \mapsto r, \\
& \quad \text{dest} \mapsto c, \\
& \quad \text{type} \mapsto \text{ReadResponse}, \\
& \quad \text{requestID} \mapsto m.\text{requestID}, \\
& \quad \text{viewID} \mapsto \text{viewID}[r], \\
& \quad \text{primary} \mapsto \text{IsPrimary}(r), \\
& \quad \text{index} \mapsto \text{Len}(\log[r]), \\
& \quad \text{checksum} \mapsto \log[r][\text{Len}(\log[r])].\text{checksum}, \\
& \quad \text{succeeded} \mapsto \text{TRUE}]) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{status}, \log, \text{viewID}, \text{lastNormalView}, \text{viewChanges} \rangle \\
\text{ChangeView}(r) & \triangleq \\
& \wedge \text{Sends}(\{[\text{src} \mapsto r, \\
& \quad \text{dest} \mapsto d, \\
& \quad \text{type} \mapsto \text{ViewChangeRequest}, \\
& \quad \text{viewID} \mapsto \text{viewID}[r] + 1] : d \in \text{Replicas}\}) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{replicaVars} \rangle \\
\text{HandleViewChangeRequest}(r, s, m) & \triangleq \\
& \wedge \text{viewID}[r] < m.\text{viewID} \\
& \wedge \text{viewID}' = [\text{viewID} \text{ EXCEPT } ![r] = m.\text{viewID}] \\
& \wedge \text{status}' = [\text{status} \text{ EXCEPT } ![r] = \text{ViewChangeStatus}] \\
& \wedge \text{viewChanges}' = [\text{viewChanges} \text{ EXCEPT } ![r] = \{\}] \\
& \wedge \text{Reply}(m, [\text{src} \mapsto r, \\
& \quad \text{dest} \mapsto \text{Primary}(m.\text{viewID}), \\
& \quad \text{type} \mapsto \text{ViewChangeResponse}, \\
& \quad \text{viewID} \mapsto m.\text{viewID}, \\
& \quad \text{lastNormal} \mapsto \text{lastNormalView}[r],
\end{aligned}$$

$$\begin{aligned}
& \log \quad \mapsto \log[r]) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \log, \text{lastNormalView} \rangle \\
\text{HandleViewChangeResponse}(r, s, m) & \triangleq \\
& \wedge \text{IsPrimary}(r) \\
& \wedge \text{viewID}[r] = m.\text{viewID} \\
& \wedge \text{status}[r] = \text{ViewChangeStatus} \\
& \wedge \text{viewChanges}' = [\text{viewChanges} \text{ EXCEPT } ![r] = \text{viewChanges}[r] \cup \{m\}] \\
& \wedge \text{LET} \\
& \quad \text{isViewQuorum}(vs) \triangleq \text{IsQuorum}(vs) \wedge \exists v \in vs : v.\text{src} = r \\
& \quad \text{newViewChanges} \triangleq \{v \in \text{viewChanges}'[r] : v.\text{viewID} = \text{viewID}[r]\} \\
& \quad \text{normalViews} \triangleq \{v.\text{lastNormal} : v \in \text{newViewChanges}\} \\
& \quad \text{lastNormal} \triangleq \text{CHOOSE } v \in \text{normalViews} : \forall v2 \in \text{normalViews} : v2 \leq v \\
& \quad \text{goodLogs} \triangleq \{n.\log : n \in \{v \in \text{newViewChanges} : v.\text{lastNormal} = \text{lastNormal}\}\} \\
& \quad \text{combineLogs}(ls) \triangleq \\
& \quad \quad \text{LET} \\
& \quad \quad \quad \text{logsWith}(i) \triangleq \{l \in ls : \text{Len}(l) \geq i\} \\
& \quad \quad \quad \text{entries}(i) \triangleq \{l[i] : l \in \text{logsWith}(i)\} \\
& \quad \quad \quad \text{quorums}(i) \triangleq \{l \in \text{SUBSET } \text{logsWith}(i) : \text{IsQuorum}(l)\} \\
& \quad \quad \quad \text{checksums}(l, i) \triangleq \{e.\text{checksum} : e \in l[i]\} \\
& \quad \quad \quad \text{isCommitted}(i) \triangleq \exists e \in \text{entries}(i) : \forall l \in \text{quorums}(i) : e.\text{checksum} \in \text{checksums}(l, i) \\
& \quad \quad \quad \text{entry}(i) \triangleq \text{CHOOSE } e \in \text{entries}(i) : \forall l \in \text{quorums}(i) : e.\text{checksum} \in \text{checksums}(l, i) \\
& \quad \quad \quad \text{maxIndex} \triangleq \text{Max}(\{\text{Len}(l) : l \in ls\}) \\
& \quad \quad \quad \text{commits} \triangleq \{i \in 1 \dots \text{maxIndex} : \text{isCommitted}(i)\} \\
& \quad \quad \quad \text{maxCommit} \triangleq \text{IF } \text{Cardinality}(\text{commits}) > 0 \text{ THEN } \text{Max}(\text{commits}) \text{ ELSE } 0 \\
& \quad \quad \text{IN} \\
& \quad \quad \quad [i \in 1 \dots \text{maxCommit} \mapsto \text{entry}(i)] \\
& \quad \text{IN} \\
& \quad \vee \wedge \text{isViewQuorum}(\text{newViewChanges}) \\
& \quad \quad \wedge \text{Replies}(m, \{[\text{src} \mapsto r, \\
& \quad \quad \quad \text{dest} \mapsto d, \\
& \quad \quad \quad \text{type} \mapsto \text{StartViewRequest}, \\
& \quad \quad \quad \text{viewID} \mapsto \text{viewID}[r], \\
& \quad \quad \quad \log \mapsto \text{combineLogs}(\text{goodLogs})] : d \in \text{Replicas}\}) \\
& \quad \vee \wedge \neg \text{isViewQuorum}(\text{newViewChanges}) \\
& \quad \quad \wedge \text{Discard}(m) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{status}, \text{viewID}, \log, \text{lastNormalView} \rangle \\
\text{HandleStartViewRequest}(r, s, m) & \triangleq \\
& \wedge \vee \text{viewID}[r] < m.\text{viewID} \\
& \quad \vee \wedge \text{viewID}[r] = m.\text{viewID} \\
& \quad \quad \wedge \text{status}[r] = \text{ViewChangeStatus} \\
& \wedge \log' = [\log \text{ EXCEPT } ![r] = m.\log] \\
& \wedge \text{status}' = [\text{status} \text{ EXCEPT } ![r] = \text{NormalStatus}] \\
& \wedge \text{viewID}' = [\text{viewID} \text{ EXCEPT } ![r] = m.\text{viewID}]
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{lastNormalView}' = [\text{lastNormalView} \text{ EXCEPT } ![r] = m.\text{viewID}] \\
& \wedge \text{Discard}(m) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{viewChanges} \rangle
\end{aligned}$$


---

$$\begin{aligned}
\text{InitMessageVars} & \triangleq \\
& \wedge \text{messages} = \{\}
\end{aligned}$$

$$\begin{aligned}
\text{InitClientVars} & \triangleq \\
& \wedge \text{globalTime} = 0 \\
& \wedge \text{time} = [c \in \text{Clients} \mapsto 0] \\
& \wedge \text{requestID} = [c \in \text{Clients} \mapsto 0] \\
& \wedge \text{responses} = [c \in \text{Clients} \mapsto [r \in \text{Replicas} \mapsto [s \in \{\} \mapsto [\text{index} \mapsto 0, \text{checksum} \mapsto \text{Nil}]]]] \\
& \wedge \text{writes} = [c \in \text{Clients} \mapsto \{\}] \\
& \wedge \text{reads} = [c \in \text{Clients} \mapsto \{\}]
\end{aligned}$$

$$\begin{aligned}
\text{InitReplicaVars} & \triangleq \\
& \wedge \text{replicas} = \text{SeqFromSet}(\text{Replicas}) \\
& \wedge \text{status} = [r \in \text{Replicas} \mapsto \text{NormalStatus}] \\
& \wedge \text{log} = [r \in \text{Replicas} \mapsto \langle \rangle] \\
& \wedge \text{viewID} = [r \in \text{Replicas} \mapsto 1] \\
& \wedge \text{lastNormalView} = [r \in \text{Replicas} \mapsto 1] \\
& \wedge \text{viewChanges} = [r \in \text{Replicas} \mapsto \{\}]
\end{aligned}$$

$$\begin{aligned}
\text{Init} & \triangleq \\
& \wedge \text{InitMessageVars} \\
& \wedge \text{InitClientVars} \\
& \wedge \text{InitReplicaVars} \\
& \wedge \text{transitions} = 0
\end{aligned}$$


---

The type invariant checks that no read ever reads a different value than a previous write

$$\begin{aligned}
\text{Inv} & \triangleq \\
& \wedge \forall c1, c2 \in \text{Clients} : \\
& \quad \neg \exists r \in \text{reads}[c1] : \\
& \quad \quad \exists w \in \text{writes}[c2] : \\
& \quad \quad \quad \wedge r.\text{index} = w.\text{index} \\
& \quad \quad \quad \wedge \neg \text{ChecksumsMatch}(r.\text{checksum}, w.\text{checksum}) \\
& \wedge \forall c1, c2 \in \text{Clients} : \\
& \quad \neg \exists r1 \in \text{reads}[c1] : \\
& \quad \quad \exists r2 \in \text{reads}[c2] : \\
& \quad \quad \quad \wedge r1.\text{index} = r2.\text{index} \\
& \quad \quad \quad \wedge \neg \text{ChecksumsMatch}(r1.\text{checksum}, r2.\text{checksum})
\end{aligned}$$

$$\text{Transition} \triangleq \text{transitions}' = \text{transitions} + 1$$

$$\begin{aligned}
Next &\triangleq \\
&\vee \exists c \in Clients : \\
&\quad \wedge Write(c) \\
&\quad \wedge Transition \\
&\vee \exists c \in Clients : \\
&\quad \wedge Read(c) \\
&\quad \wedge Transition \\
&\vee \exists r \in Replicas : \\
&\quad \wedge ChangeView(r) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = WriteRequest \\
&\quad \wedge HandleWriteRequest(m.dest, m.src, m) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = WriteResponse \\
&\quad \wedge HandleWriteResponse(m.dest, m.src, m) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = ReadRequest \\
&\quad \wedge HandleReadRequest(m.dest, m.src, m) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = ReadResponse \\
&\quad \wedge HandleReadResponse(m.dest, m.src, m) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = ViewChangeRequest \\
&\quad \wedge HandleViewChangeRequest(m.dest, m.src, m) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = ViewChangeResponse \\
&\quad \wedge HandleViewChangeResponse(m.dest, m.src, m) \\
&\quad \wedge Transition \\
&\vee \exists m \in messages : \\
&\quad \wedge m.type = StartViewRequest \\
&\quad \wedge HandleStartViewRequest(m.dest, m.src, m) \\
&\quad \wedge Transition
\end{aligned}$$

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


---

\ \* Modification History  
\ \* Last modified Mon Sep 21 17:36:58 PDT 2020 by jordanhalterman  
\ \* Created Fri Sep 18 22:45:21 PDT 2020 by jordanhalterman