

---

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*  $\triangleq$   $\langle replicas \rangle$

VARIABLE *messages*

*messageVars*  $\triangleq$   $\langle messages \rangle$

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 *viewChangeResponses*

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

VARIABLE *transitions*

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

---

#### Helpers

RECURSIVE  $SeqFromSet(-)$

$SeqFromSet(S) \triangleq$

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

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

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

$Max(s) \triangleq$  CHOOSE  $x \in s : \forall y \in s : x \geq y$

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

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

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

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

---

#### Messaging helpers

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

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

$Replies(reqs, resps) \triangleq messages' = (messages \cup resps) \setminus reqs$

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

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

$AdvanceTime(c) \triangleq$

$\wedge globalTime' = globalTime + 1$

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

$\text{ELSE}$

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

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

$Write(c) \triangleq$

$\wedge AdvanceTime(c)$

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

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

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

$Read(c) \triangleq$

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

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

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

$ChecksumsMatch(c1, c2) \triangleq$

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

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

$IsCommitted(acks) \triangleq$

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

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

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

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

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

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

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

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

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

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


---

Server request/response handling

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

$$\begin{aligned}
& \text{dest} \mapsto c, \\
& \text{type} \mapsto \text{WriteResponse}, \\
& \text{requestID} \mapsto m.\text{requestID}, \\
& \text{viewID} \mapsto \text{viewID}[r], \\
& \text{primary} \mapsto \text{IsPrimary}(r), \\
& \text{index} \mapsto \text{Len}(\log[r]), \\
& \text{checksum} \mapsto \log[r][\text{Len}(\log[r])].\text{checksum}, \\
& \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{viewChangeResponses} \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{viewChangeResponses} \rangle \\
\text{ChangeView}(r) & \triangleq \\
& \text{LET } \text{nextViewID} \triangleq \text{viewID}[r] + 1 \\
& \text{IN} \\
& \wedge \text{Primary}(\text{nextViewID}) = r \\
& \wedge \text{status}' = [\text{status} \text{ EXCEPT } ![r] = \text{ViewChangeStatus}] \\
& \wedge \text{viewID}' = [\text{viewID} \text{ EXCEPT } ![r] = \text{nextViewID}] \\
& \wedge \text{viewChangeResponses}' = [\text{viewChangeResponses} \text{ EXCEPT } ![r] = \{\}] \\
& \wedge \text{Sends}(\{[\text{src} \mapsto r, \\
& \quad \text{dest} \mapsto d, \\
& \quad \text{type} \mapsto \text{ViewChangeRequest}, \\
& \quad \text{viewID} \mapsto \text{nextViewID}] : d \in \text{Replicas}\}) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \log, \text{lastNormalView} \rangle \\
\text{HandleViewChangeRequest}(r, s, m) & \triangleq \\
& \wedge \text{viewID}[r] \neq 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{viewChangeResponses}' = [\text{viewChangeResponses} \text{ EXCEPT } ![r] = \{\}] \\
& \wedge \text{Reply}(m, [\text{src} \mapsto r, \\
& \quad \text{dest} \mapsto s, \\
& \quad \text{type} \mapsto \text{ViewChangeResponse},
\end{aligned}$$

$$\begin{aligned}
& \text{viewID} \mapsto m.\text{viewID}, \\
& \text{lastNormal} \mapsto \text{lastNormalView}[r], \\
& \text{log} \mapsto \text{log}[r]) \\
& \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{log}, \text{lastNormalView} \rangle \\
\text{HandleViewChangeResponse}(r, s, m) & \triangleq \\
& \wedge \text{viewID}[r] = m.\text{viewID} \\
& \wedge \text{status}[r] = \text{ViewChangeStatus} \\
& \wedge \text{IsPrimary}(r) \\
& \wedge \text{viewChangeResponses}' = [\text{viewChangeResponses} \text{ EXCEPT } ![r] = \text{viewChangeResponses}[r] \cup \{m\}] \\
& \wedge \text{LET} \\
& \quad \text{isViewPromise}(M) \triangleq \\
& \quad \wedge \{n.\text{src} : n \in M\} \in \text{Quorums} \\
& \quad \wedge \exists n \in M : n.\text{src} = r \\
& \quad \text{viewChanges} \triangleq \{n \in \text{viewChangeResponses}[r] : n.\text{type} = \text{ViewChangeResponse} \wedge n.\text{viewID} = \text{viewID}[r]\} \\
& \quad \text{normalViews} \triangleq \{n.\text{lastNormal} : n \in \text{viewChanges}\} \\
& \quad \text{lastNormal} \triangleq \{\text{CHOOSE } v \in \text{normalViews} : \forall v2 \in \text{normalViews} : v2 < v\} \\
& \quad \text{goodLogs} \triangleq \{n.\text{log} : n \in \{o \in \text{viewChanges} : o.\text{lastNormal} = \text{lastNormal}\}\} \\
& \quad \text{combineLogs}(ls) \triangleq \\
& \quad \text{LET} \\
& \quad \quad \text{logsWith}(i) \triangleq \{l \in ls : \text{Len}(l) \geq i\} \\
& \quad \quad \text{entries}(i) \triangleq \{l[i] : l \in \text{logsWith}(i)\} \\
& \quad \quad \text{quorums}(i) \triangleq \{l \in \text{SUBSET } \text{logsWith}(i) : \text{Cardinality}(l) * 2 > \text{Cardinality}(\text{Replicas})\} \\
& \quad \quad \text{checksums}(l, i) \triangleq \{e.\text{checksum} : e \in l[i]\} \\
& \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 \text{committed}(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 \text{maxIndex} \triangleq \text{Max}(\{\text{Len}(l) : l \in ls\}) \\
& \quad \quad \text{maxCommitted} \triangleq \text{Max}(\{i \in 1 \dots \text{maxIndex} : \text{isCommitted}(i)\}) \\
& \quad \text{IN} \\
& \quad \quad [i \in 1 \dots \text{maxCommitted} \mapsto \text{committed}(i)] \\
& \text{IN} \\
& \quad \text{IF } \text{isViewPromise}(\text{viewChanges}) \text{ THEN} \\
& \quad \quad \wedge \text{Replies}(m, \{\text{src} \mapsto r, \\
& \quad \quad \quad \text{dest} \mapsto d, \\
& \quad \quad \quad \text{viewID} \mapsto \text{viewID}[r], \\
& \quad \quad \quad \text{log} \mapsto \text{combineLogs}(\text{goodLogs})\} : d \in \text{Replicas}) \\
& \quad \text{ELSE} \\
& \quad \quad \wedge \text{Discard}(m) \\
& \quad \wedge \text{UNCHANGED } \langle \text{globalVars}, \text{clientVars}, \text{status}, \text{viewID}, \text{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 \wedge \text{status}[r] = \text{ViewChangeStatus} \\
& \wedge \text{log}' = [\text{log} \text{ EXCEPT } ![r] = m.\text{log}]
\end{aligned}$$

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


---

$$\begin{aligned}
InitMessageVars & \triangleq \\
& \wedge messages = \{\} \\
InitClientVars & \triangleq \\
& \wedge globalTime = 0 \\
& \wedge time = [c \in Clients \mapsto 0] \\
& \wedge requestID = [c \in Clients \mapsto 0] \\
& \wedge responses = [c \in Clients \mapsto [r \in Replicas \mapsto [s \in \{\} \mapsto [index \mapsto 0, checksum \mapsto Nil]]]] \\
& \wedge writes = [c \in Clients \mapsto \{\}] \\
& \wedge reads = [c \in Clients \mapsto \{\}] \\
InitReplicaVars & \triangleq \\
& \wedge replicas = SeqFromSet(Replicas) \\
& \wedge status = [r \in Replicas \mapsto NormalStatus] \\
& \wedge log = [r \in Replicas \mapsto \langle \rangle] \\
& \wedge viewID = [r \in Replicas \mapsto 1] \\
& \wedge lastNormalView = [r \in Replicas \mapsto 1] \\
& \wedge viewChangeResponses = [r \in Replicas \mapsto \{\}] \\
Init & \triangleq \\
& \wedge InitMessageVars \\
& \wedge InitClientVars \\
& \wedge InitReplicaVars \\
& \wedge transitions = 0
\end{aligned}$$


---

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

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

$$\wedge \neg \text{ChecksumsMatch}(r1.\text{checksum}, r2.\text{checksum})$$

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

$$\text{Next} \triangleq$$

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

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



\ \* Modification History  
\ \* Last modified *Mon Sep 21 14:10:18 PDT 2020* by *jordanhalterman*  
\ \* Created *Fri Sep 18 22:45:21 PDT 2020* by *jordanhalterman*