```
──────────────── MODULE VR_without_message ────────────────
EXTENDS Integers, Sequences, FiniteSets

CONSTANTS Replica, Quorum

 Replica Status
CONSTANTS Normal, ViewChange, Recovering

 Client operation
CONSTANT Operation

 types of log blocks
CONSTANTS RequestBlock, ViewBlock

 State on each replica
VARIABLE replicaState

vars ≜ ⟨replicaState⟩

 Special value
CONSTANT None

 Message types for processing client request
CONSTANTS Request

 Sequence with all replicas (for view selection)
CONSTANT ReplicaSequence

Statuses ≜ {Normal, ViewChange, Recovering}

────────────────────────────────────────────────────────────

View ≜ Nat

RequestMessage ≜ [type : {Request}, op : Operation]

LogEntry ≜ [type : {RequestBlock}, opNumber : Nat, m : RequestMessage]
        ∪ [type      : {ViewBlock}, view : View]

TypeOK ≜ ∧ replicaState ∈ [
              Replica → [
                  viewNumber : View,
                  status : Statuses,
                  log : Seq(LogEntry),
                  downloadReplica : Replica ∪ {None},
                  commitNumber : Nat,
                  executedOperations : Seq(LogEntry)
                ]
              ]
```

$$\textsc{assume}\ \mathit{QuorumAssumption} \triangleq \begin{array}{l} \land\ \forall\, Q \in \mathit{Quorum} : Q \subseteq \mathit{Replica} \\ \land\ \forall\, Q1,\ Q2 \in \mathit{Quorum} : Q1 \cap Q2 \neq \{\} \end{array}$$

$$\textsc{assume}\ \mathit{IsFiniteSet}(\mathit{Replica})$$

$$\mathit{Min}(S) \triangleq \textsc{choose}\ x \in S : \forall\, y \in S : x \leq y$$

$$\mathit{Max}(S) \triangleq \textsc{choose}\ x \in S : \forall\, y \in S : y \leq x$$

$$\mathit{lastOpNumber}(l) \triangleq \textsc{if}\ l = \langle\rangle\ \textsc{then}\ 0\ \textsc{else}\ \ l[\mathit{Len}(l)].\mathit{opNumber}$$

$$\begin{array}{l} \mathit{Init} \triangleq\ \land\ \mathit{replicaState} = [r \in \mathit{Replica} \mapsto [\\ \qquad\qquad\qquad \mathit{viewNumber} \mapsto 0, \\ \qquad\qquad\qquad \mathit{status} \mapsto \mathit{Normal}, \\ \qquad\qquad\qquad \mathit{log} \mapsto \langle[\mathit{type} \mapsto \mathit{ViewBlock},\ \mathit{view} \mapsto 0]\rangle, \\ \qquad\qquad\qquad \mathit{downloadReplica} \mapsto \mathit{None}, \\ \qquad\qquad\qquad \mathit{commitNumber} \mapsto 0, \\ \qquad\qquad\qquad \mathit{executedOperations} \mapsto \langle\rangle \\ \qquad\qquad\quad] \\ \qquad\quad] \end{array}$$

$$\mathit{ViewNumber}(r) \triangleq \mathit{replicaState}[r].\mathit{viewNumber}$$

$$\mathit{Status}(r) \triangleq \mathit{replicaState}[r].\mathit{status}$$

$$\mathit{Log}(r) \triangleq \mathit{replicaState}[r].\mathit{log}$$

$$\mathit{LogLen}(r) \triangleq \mathit{Len}(\mathit{Log}(r))$$

$$\mathit{LastNormalView}(r) \triangleq \mathit{Max}(\{0\} \cup \{\mathit{Log}(r)[i].\mathit{view} : i \in \{i \in 1 \mathrel{..} \mathit{LogLen}(r) : \mathit{Log}(r)[i].\mathit{type} = \mathit{ViewBlock}\}\})$$

$$\mathit{OpNumber}(r) \triangleq \mathit{LogLen}(r)$$

$$\mathit{DownloadReplica}(r) \triangleq \mathit{replicaState}[r].\mathit{downloadReplica}$$

$$\mathit{CommitNumber}(r) \triangleq \mathit{replicaState}[r].\mathit{commitNumber}$$

$$\mathit{ExecutedOperations}(r) \triangleq \mathit{replicaState}[r].\mathit{executedOperations}$$

$$\mathit{ExecuteOperation}(\mathit{op}) \triangleq \mathit{op}$$

$$\mathit{ReplicaIndex}(r) \triangleq \textsc{choose}\ i \in 1 \mathrel{..} \mathit{Cardinality}(\mathit{Replica}) : \mathit{ReplicaSequence}[i] = r$$

$$\mathit{PrimaryReplicaInView}(v) \triangleq \mathit{ReplicaSequence}[(v \% \mathit{Len}(\mathit{ReplicaSequence})) + 1]$$

$$\mathit{IsPrimaryInView}(r,\ v) \triangleq \mathit{PrimaryReplicaInView}(v) = r$$

$$\mathit{IsPrimary}(r) \triangleq \mathit{IsPrimaryInView}(r,\ \mathit{replicaState}[r].\mathit{viewNumber})$$

```
$IsDownloadingBeforeView(r) \triangleq$
    $\land\ replicaState[r].downloadReplica \neq None$

$AddClientRequest(r,\ m) \triangleq$
    $\land\ replicaState' = [replicaState\ \text{EXCEPT}\ ![r].log = Append(@,\ [$
                                                  $type \mapsto RequestBlock,$
                                                  $opNumber \mapsto OpNumber(r) + 1,$
                                                  $m \mapsto m$
                                                $])]$

$RecieveClientRequest(p,\ op) \triangleq$
    $\land\ IsPrimary(p)$
    $\land\ Status(p) = Normal$
    $\land\ \neg IsDownloadingBeforeView(p)$
    $\land\ AddClientRequest(p,\ [type \mapsto Request,\ op \mapsto op])$

$FirstIndexOfViewBlock(log,\ v) \triangleq Min(\{Len(log) + 1\} \cup \{i \in 1 \mathrel{..} Len(log) : log[i].type = ViewBlock \land log[i].v$

$MaxLogEntryInView(log,\ v) \triangleq$ LET $first \triangleq FirstIndexOfViewBlock(log,\ v)$
                                IN   IF $\land\ first \leq Len(log)$
                                        $\land\ log[first].view = v$
                                     THEN $FirstIndexOfViewBlock(log,\ v + 1) - 1$
                                     ELSE $0$

$RecievePrepare(r) \triangleq$
    $\land\ \neg IsPrimary(r)$
    $\land\ Status(r) = Normal$
    $\land\ \neg IsDownloadingBeforeView(r)$
    $\land$ LET $primary \triangleq PrimaryReplicaInView(ViewNumber(r))$
      IN      $\land\ ViewNumber(primary) = ViewNumber(r)$ \* Here should be “primary was in $Normal$ status in our view and ha
              $\land\ Status(primary) = Normal$
            $\land\ \lor\ \land\ MaxLogEntryInView(Log(primary),\ ViewNumber(r)) > LogLen(r)$
                    $\land\ Log(primary)[LogLen(r) + 1].type = RequestBlock$
                    $\land\ AddClientRequest(r,\ Log(primary)[LogLen(r) + 1].m)$
                    Recieved Prepare when Primary has already rejected his $log$ entries, for example after recieving $StartView$
               $\lor\ \land\ MaxLogEntryInView(Log(primary),\ ViewNumber(r)) = LogLen(r)$
                    $\land\ ViewNumber(primary) > ViewNumber(r)$
                    $\land\ \exists\ op \in Operation : AddClientRequest(r,\ [type \mapsto Request,\ op \mapsto op])$


$ExecuteRequest(r,\ entry) \triangleq$
    $\land\ replicaState' = [replicaState\ \text{EXCEPT}\ ![r].executedOperations = Append(@,\ entry)]$

$ExecuteClientRequest(r) \triangleq$
    $\land\ Status(r) = Normal$
    $\land\ \neg IsDownloadingBeforeView(r)$
    $\land\ Len(ExecutedOperations(r)) < CommitNumber(r)$
    $\land Len(ExecutedOperations(r)) < Len(Log(r))$
    $\land ExecuteRequest(r,\ Log(r)[Len(ExecutedOperations(r)) + 1])$

$AchievePrepareOkFromQuorum(p) \triangleq$
    $\land IsPrimary(p)$
    $\land Status(p) = Normal$
    $\land \neg IsDownloadingBeforeView(p)$
    $\land Len(ExecutedOperations(p)) = CommitNumber(p)$
    $\land$ LET $newCommit \triangleq CommitNumber(p) + 1$
      IN    $\land \exists\, Q \in Quorum :$
                $\forall\, r \in Q :$
                    $\lor MaxLogEntryInView(Log(r),\ ViewNumber(p)) \geq newCommit$
                    $\lor r = p$
            $\land replicaState' = [replicaState$ EXCEPT $![p].commitNumber = newCommit,$
                                                 $![p].executedOperations = Append(@,\ Log(p)[newCommit])$

$RecieveCommit(r) \triangleq$
    $\land \neg IsPrimary(r)$   Need this?
    $\land Status(r) = Normal$
    $\land \neg IsDownloadingBeforeView(r)$
    $\land \exists\, p \in Replica :$
      $\exists\, newCommit \in CommitNumber(r) + 1 \,..\, Min(\{LogLen(r),\ CommitNumber(p)\}) :$
        $\land CommitNumber(p) > CommitNumber(r)$
        $\land \exists\, Q \in Quorum :$
            $\land p \in Q$
            $\land \forall\, r2 \in Q :$
                $\land LogLen(r2) \geq newCommit$
                $\land Log(r2)[newCommit] = Log(r)[newCommit]$
        $\land replicaState' = [replicaState$ EXCEPT $![r].commitNumber = newCommit]$
```

View Changing

```
$\rightarrow E1$
$TimeoutStartViewChanging(r) \triangleq$
    $\land Status(r) = Normal$
    $\land replicaState' = [replicaState$ EXCEPT $![r].downloadReplica = None,$
                                         $![r].viewNumber = @ + 1,$
                                         $![r].status = ViewChange]$

$\rightarrow E1$
$RecieveStartViewChange(r) \triangleq$
    $\land Status(r) = Normal$
    $\land \exists\, r2 \in Replica :$
        $\land ViewNumber(r2) > ViewNumber(r)$
        $\land Status(r2) = ViewChange$
    $\land \exists\, newView \in ViewNumber(r) + 1 \,..\, ViewNumber(r2) :$
        $replicaState' = [replicaState \ \text{EXCEPT} \ ![r].downloadReplica = None,$
                                                    $![r].viewNumber = newView,$
                                                    $![r].status = ViewChange]$
```

TODO: *ADD* → *Er* and → Em states and transitions

Become Primary

Em → *Mc*

```
$AchieveDoViewChangeFromQuorum(p) \triangleq$
    $\land IsPrimary(p)$
    $\land Status(p) = ViewChange$
    $\land \exists\, Q \in Quorum,\ recievedReplicas \in \text{SUBSET}\ Replica :$
        $\land p \in Q$
        $\land Q \subseteq recievedReplicas$
        $\land \forall\, r \in recievedReplicas :$
            Problem with $WithMsg\ Spec$, because other replicas could start new $View$
            $\land ViewNumber(r) = ViewNumber(p)$
            $\land Status(r) = ViewChange$
            Problem with $WithMsg\ Spec$, because there are can be saved messages with old state ($lastNormalView$, $opNumber$
            And here no such state is saved + other replicas could increase their state
               $\Rightarrow maxVV$, $maxN$, $maxReplica$ and new commit can easily differ from $WithMsgs\ Spec$
        $\land \text{LET}\ maxVV \triangleq Max(\{LastNormalView(r) : r \in recievedReplicas\})$
                $maxN \triangleq Max(\{OpNumber(r) : r \in \{r \ \ \in recievedReplicas : LastNormalView(r) = maxVV\}\}$
                $maxReplicaIndex \triangleq Max(\{ReplicaIndex(r) : r \in \{r \in recievedReplicas : LastNormalView(r) =$
                $maxReplica \triangleq \text{CHOOSE}\ r \in recievedReplicas : ReplicaIndex(r) = maxReplicaIndex$
          $\text{IN}\ \ \land replicaState' = [replicaState\ \text{EXCEPT}\ ![p].log = \text{IF}\ maxReplica = p\ \text{THEN}\ Log(p)\ \text{ELSE}\ \ SubSe$
                                                        $![p].downloadReplica = maxReplica,$
                                                        $![p].commitNumber = Max(\{CommitNumber(r) : r \in r$
                                                        $![p].status = Normal]$
```

Mc → *Mc* / *Mc* → *M*

```
$MasterDownloadBeforeView(p) \triangleq$
    $\land IsPrimary(p)$
    $\land Status(p) = Normal$
    $\land IsDownloadingBeforeView(p)$
    $\land ViewNumber(p) = ViewNumber(DownloadReplica(p))$   If replica will increase view, then this Primary could on
    $\land LogLen(p) \leq LogLen(DownloadReplica(p))$
    $\land \lor \land LogLen(p) = LogLen(DownloadReplica(p))$
            $\land replicaState' = [replicaState\ \text{EXCEPT}\ ![p].log = Append(@,\ [type \mapsto ViewBlock,\ view \mapsto ViewNumb$
                                                        $![p].downloadReplica = None]$
        $\lor \land LogLen(p) < LogLen(DownloadReplica(p))$
            $\land replicaState' = [replicaState\ \text{EXCEPT}\ ![p].log = Append(@,\ Log(DownloadReplica(p))[LogLen(p) +$
```

$Append(newLog,\ [type \mapsto ViewBlock,\ view \mapsto viewNumber[p]])$

```
Er → Rc
TODO: setup download up to View metablock
RecieveStartView(r) ≜
    ∧ ∃ p ∈ Replica :
          ∧ IsPrimary(p)
          ∧ Status(p) = Normal
          ∧ ¬IsDownloadingBeforeView(p)
           Problem with WithMsgs Spec if p has already increased view
          ∧ ∨ ViewNumber(p) > ViewNumber(r)
            ∨ ∧ ViewNumber(p) = ViewNumber(r)
              ∧ Status(r) = ViewChange
          ∧ replicaState′ = [replicaState EXCEPT ![r].log = SubSeq(Log(r), 1, Min({LogLen(r), CommitNumbe
                                                 ![r].downloadReplica = p,
                                                 ![r].viewNumber = ViewNumber(p),
                                                 ![r].status = Normal]

Rc → Rc / Rc → R
ReplicaDownloadBeforeView(r) ≜
    ∧ ¬IsPrimary(r)
    ∧ Status(r) = Normal
    ∧ IsDownloadingBeforeView(r)
    ∧ IF LogLen(DownloadReplica(r)) ≤ LogLen(r)
        THEN ∧ replicaState′ = [replicaState EXCEPT ![r].downloadReplica = None]
        ELSE LET newEntry ≜ Log(DownloadReplica(r))[LogLen(r) + 1]
               IN   ∧ replicaState′ = [replicaState EXCEPT ![r].log = Append(@, newEntry),
                                                           ![r].downloadReplica =
                                                               IF newEntry = [type ↦ ViewBlock, view ↦ Vie
                                                                THEN None
                                                                ELSE @]
```

```
Next ≜ ∨ ∃ r ∈ Replica, op ∈ Operation : RecieveClientRequest(r, op)
       ∨ ∃ r ∈ Replica : RecievePrepare(r)
       ∨ ∃ p ∈ Replica : AchievePrepareOkFromQuorum(p)
       ∨ ∃ r ∈ Replica : RecieveCommit(r)
       ∨ ∃ r ∈ Replica : ExecuteClientRequest(r)
    ∨ ∃ r ∈ Replica: TimeoutStartViewChanging(r)
    ∨ ∃ r ∈ Replica: RecieveStartViewChange(r)
    ∨ ∃ r ∈ Replica: AchieveDoViewChangeFromQuorum(r)
    ∨ ∃ r ∈ Replica: RecieveStartView(r)
    ∨ ∃ p ∈ Replica: MasterDownloadBeforeView(p)
    ∨ ∃ r ∈ Replica: ReplicaDownloadBeforeView(r)
```

Liveness

$EventuallyRecieveClientRequest \triangleq \forall r \in Replica\colon \mathrm{WF}\_vars(\exists op \in Operation : RecieveClientRequest(r, op))$

$EventuallyRecievePrepare \triangleq \forall r \in Replica\colon \mathrm{WF}\_vars(\exists m \in msgs : RecievePrepare(r, m))$

$EventuallyRecieveCommit \triangleq \forall r \in Replica\colon \mathrm{WF}\_vars(\exists m \in msgs : RecieveCommit(r, m))$

$EventuallyRecievePrepareOk \triangleq \forall p \in Replica\colon \mathrm{WF}\_vars(\exists m \in msgs : RecievePrepareOk(p, m))$

$$
\begin{aligned}
&LivenessSpec \triangleq\\
&\quad \land\ EventuallyRecieveClientRequest\\
&\quad \land\ EventuallyRecievePrepare\\
&\quad \land\ EventuallyRecieveCommit\\
&\quad \land\ EventuallyRecievePrepareOk
\end{aligned}
$$

Full *Spec*

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

INVARIANTS

$$EveryViewHasAtLeastOnePrimary \triangleq \forall v \in 0 \mathrel{..} 10 : \exists r \in Replica : IsPrimaryInView(r, v)$$

$$
\begin{aligned}
EveryViewHasAtMostOnePrimary \triangleq \forall v \in 0 \mathrel{..} 10 : \forall r1, r2 \in Replica :&\\
(IsPrimaryInView(r1, v)&\\
\land IsPrimaryInView(r2, v)) \Rightarrow r1 = r2&
\end{aligned}
$$

$$PreficiesAreEqual(s1, s2) \triangleq \forall i \in \textsc{domain}\ s1 \cap \textsc{domain}\ s2 : s1[i] = s2[i]$$

$$
\begin{aligned}
ExecutedOperationsPreficesAreEqual \triangleq \forall r1, r2 \in Replica :&\\
&PreficiesAreEqual(\\
&\quad ExecutedOperations(r1),\\
&\quad ExecutedOperations(r2)\\
&)
\end{aligned}
$$

$$
\begin{aligned}
PreficiesOfLenAreEqual(s1, s2, prefLen) \triangleq \forall i \in \textsc{domain}\ s1 \cap \textsc{domain}\ s2&\\
\cap\, 1 \mathrel{..} prefLen :&\\
s1[i] = s2[i]&
\end{aligned}
$$

$$
\begin{aligned}
CommitedLogsPreficesAreEqual \triangleq \forall r1, r2 \in Replica : PreficiesOfLenAreEqual(&\\
&Log(r1),\\
&Log(r2),\\
&Min(\{\\
&\quad CommitNumber(r1),\\
&\quad CommitNumber(r2)\\
&\})\\
)&
\end{aligned}
$$

Properties

$AllClientsWillBeServed \triangleq \forall\, c \in \text{Client}: (pendingRequest[c] \leadsto \neg pendingRequest[c])$

\* Modification History
\* Last modified *Wed Apr* 05 20:16:09 *MSK* 2023 by *tycoon*
\* Created *Wed Dec* 28 15:30:37 *MSK* 2022 by *tycoon*