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*

Special value

CONSTANT *None*

Sequence with all replicas (for view selection)

CONSTANT *ReplicaSequence*

For state space limitation

CONSTANT *MaxRequests*, *MaxViews*

State on each replica

VARIABLE *replicaState*

$vars \triangleq \langle replicaState \rangle$

$Statuses \triangleq \{Normal,\ ViewChange,\ Recovering\}$

$LogEntry \triangleq [type : \{RequestBlock\},\ opNumber : Nat,\ op : Operation]$
$\quad \cup [type : \{ViewBlock\},\ view : Nat]$

$$
\begin{array}{ll}
TypeOK \triangleq & \land\ replicaState \in [\\
& \quad Replica \to [\\
& \qquad viewNumber : Nat, \\
& \qquad status : Statuses, \\
& \qquad log : Seq(LogEntry), \\
& \qquad downloadReplica : Replica \cup \{None\}, \\
& \qquad commitNumber : Nat \\
& \quad] \\
&]
\end{array}
$$

$$
\begin{array}{ll}
\text{ASSUME } QuorumAssumption \triangleq & \land \forall\ Q \in Quorum : Q \subseteq Replica \\
& \land \forall\ Q1,\ Q2 \in Quorum : Q1 \cap Q2 \neq \{\}
\end{array}
$$

ASSUME $IsFiniteSet(Replica)$

$$Min(S) \triangleq \text{CHOOSE } x \in S : \forall\, y \in S : x \leq y$$

$$Max(S) \triangleq \text{CHOOSE } x \in S : \forall\, y \in S : y \leq x$$

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

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

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

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

$$LogLen(r) \triangleq Len(Log(r))$$

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

$$OpNumber(r) \triangleq LogLen(r)$$

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

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

$$ReplicaIndex(r) \triangleq \text{CHOOSE } i \in 1 \,..\, Cardinality(Replica) : ReplicaSequence[i] = r$$

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

$$IsPrimaryInView(r,\ v) \triangleq PrimaryReplicaInView(v) = r$$

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

$$\begin{aligned}
IsDownloading(r) \triangleq\ & \\
& \wedge replicaState[r].downloadReplica \neq None
\end{aligned}$$

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

$$\begin{aligned}
MaxLogEntryInView(log,\ v) \triangleq\ & \text{LET } first \triangleq FirstIndexOfViewBlock(log,\ v) \\
& \text{IN} \quad \text{IF } \wedge first \leq Len(log) \\
& \qquad\qquad \wedge log[first].view = v \\
& \qquad\ \text{THEN } FirstIndexOfViewBlock(log,\ v + 1) - 1 \\
& \qquad\ \text{ELSE } 0
\end{aligned}$$

$$HasViewBlock(r,\ v) \triangleq \text{LET } ind \triangleq FirstIndexOfViewBlock(Log(r),\ v)$$
$$\text{IN} \quad \land\ ind \leq LogLen(r)$$
$$\land\ Log(r)[ind].view = v$$

$$MaxViewLessOrEq(log,\ v) \triangleq Max(\{0\} \cup \{log[i].view : i \in \{i \in 1\ ..\ Len(log) : log[i].type = ViewBlock \land log[i]$$

$$MaxOpNumBeforeView(log,\ v) \triangleq FirstIndexOfViewBlock(log,\ v) - 1$$

$$RequestBlockCount(log) \triangleq Cardinality(\{i \in \text{DOMAIN } log : log[i].type = RequestBlock\})$$

$$ViewBlockCount(log) \triangleq Cardinality(\{i \in \text{DOMAIN } log : log[i].type = ViewBlock\})$$

NORMAL OPERATION

$$AddClientRequest(r,\ m) \triangleq$$
$$\land\ replicaState' = [replicaState \text{ EXCEPT } ![r].log = Append(@,\ m)]$$

$$RecieveClientRequest(p,\ op) \triangleq$$
$$\land\ RequestBlockCount(Log(p)) < MaxRequests$$
$$\land\ IsPrimary(p)$$
$$\land\ Status(p) = Normal$$
$$\land\ \neg IsDownloading(p)$$
$$\land\ AddClientRequest(p,\ [type \mapsto RequestBlock,$$
$$opNumber \mapsto OpNumber(p) + 1,$$
$$op \mapsto op])$$

$$RecievePrepare(r) \triangleq$$
$$\land\ RequestBlockCount(Log(r)) < MaxRequests$$
$$\land\ \neg IsPrimary(r)$$
$$\land\ Status(r) = Normal$$
$$\land\ \neg IsDownloading(r)$$
Has *Replica* with saved request from primary
$$\land\ \lor\ \land \exists\, r2 \in Replica :$$
$$\land\ MaxLogEntryInView(Log(r2),\ ViewNumber(r)) > OpNumber(r)$$
$$\land\ Log(r2)[OpNumber(r) + 1].type = RequestBlock$$
$$\land\ AddClientRequest(r,\ Log(r2)[OpNumber(r) + 1])$$
There is no saved request from primary
$$\lor\ \land \forall\, r2 \in Replica :$$
$$\land\ MaxLogEntryInView(Log(r2),\ ViewNumber(r)) \leq OpNumber(r)$$
+ primary will not generate new Prepare
$$\land\ \text{LET } p \triangleq PrimaryReplicaInView(ViewNumber(r))$$
$$\text{IN} \quad \lor\ ViewNumber(p) > ViewNumber(r)$$
$$\lor\ Status(p) = Recovering$$
suddenly got old sent request from primary
$$\land\ \exists\, op \in Operation : AddClientRequest(r,\ [type \mapsto RequestBlock,$$
$$opNumber \mapsto OpNumber(r) + 1,$$

$op \mapsto op])$

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

$RecieveCommit(r) \triangleq$
$\land\ \neg IsPrimary(r)$
$\land\ Status(r) = Normal$
$\land\ \neg IsDownloading(r)$
$\land$ LET $p \triangleq PrimaryReplicaInView(ViewNumber(r))$
IN $\land \exists\, newCommit \in CommitNumber(r) + 1\ ..\ Min(\{LogLen(r),\ CommitNumber(p)\}) :$ think about
$\land \exists\, Q \in Quorum :$
$\land\ p \in Q$
$\land \forall\, r2 \in Q :$
$\land\ LogLen(r2) \geq newCommit$
$\land \forall\, i \in CommitNumber(r) + 1\ ..\ newCommit :$
$Log(r2)[i] = Log(r)[i]$
$\land\ replicaState' = [replicaState$ EXCEPT $![r].commitNumber = newCommit]$

VIEW CHANGING

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

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

$TODO$: $ADD \to Er$ and $\to$ Em states and transitions

Become Primary
Em $\to Mc$

$$
\begin{array}{l}
AchieveDoViewChangeFromQuorum(p) \triangleq \\
\quad \land\ IsPrimary(p) \\
\quad \land\ Status(p) = ViewChange \\
\quad \land\ \exists\, Q \in Quorum,\ recievedReplicas \in \text{SUBSET } Replica : \\
\qquad \land\ p \in Q \\
\qquad \land\ Q \subseteq recievedReplicas \\
\qquad \land\ \forall\, r \in recievedReplicas : \\
\qquad\quad \land\ \lor\ \land\ ViewNumber(r) = ViewNumber(p) \\
\qquad\qquad\quad \land\ Status(r) = ViewChange \\
\qquad\qquad\quad r \text{ has already joined to new elections} \\
\qquad\qquad \lor\ \land\ ViewNumber(r) > ViewNumber(p) \\
\qquad\qquad \text{Can't just take } LastNormalView(r) \text{ and } OpNumber(r), \text{ because there are can be saved messages with old state } (la \\
\qquad\qquad \text{And here no such state is saved + other replicas could increase their state} \\
\qquad\qquad\quad \Rightarrow maxVV, maxN, maxReplica \text{ and new commit can easily differ from } WithMsgs\ Spec \\
\qquad \land\ \text{LET } maxVV \triangleq Max(\{MaxViewLessOrEq(Log(r),\ ViewNumber(p) - 1) : r \in recievedReplicas\}) \\
\qquad\qquad maxN \triangleq Max(\{MaxOpNumBeforeView(Log(r),\ ViewNumber(p)) : r \in \{r \in recievedReplicas : \\
\qquad\qquad maxReplicaIndex \triangleq Max(\{ReplicaIndex(r) : r \in \{r \in recievedReplicas : LastNormalView(r) = \\
\qquad\qquad maxReplica \triangleq \\
\qquad\qquad\qquad \text{If we are suit then choose ourselves} \\
\qquad\qquad\qquad \text{IF } \land\ maxVV = MaxViewLessOrEq(Log(p),\ ViewNumber(p) - 1) \\
\qquad\qquad\qquad\quad \land\ maxN = MaxOpNumBeforeView(Log(p),\ ViewNumber(p)) \\
\qquad\qquad\qquad \text{THEN } p \\
\qquad\qquad\qquad \text{ELSE CHOOSE } r \in recievedReplicas : ReplicaIndex(r) = maxReplicaIndex \\
\qquad\quad \text{IN } \land\ replicaState' = [replicaState \text{ EXCEPT } ![p].downloadReplica = \text{IF } maxReplica = p \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{THEN } None \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE } maxReplica, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![p].log = \text{IF } maxReplica = p \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{THEN } Append(@, [type \mapsto ViewBlock, view \mapsto \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{ELSE } @, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ![p].status = Normal]
\end{array}
$$

$Mc \to Mc$ / $Mc \to M$

$$
\begin{array}{l}
MasterDownloadBeforeView(p) \triangleq \\
\quad \land\ IsPrimary(p) \\
\quad \land\ Status(p) = Normal \\
\quad \land\ IsDownloading(p) \\
\quad \land\ \text{LET } finishPos \triangleq FirstIndexOfViewBlock(Log(DownloadReplica(p)),\ ViewNumber(p) + 1) - 1 \\
\qquad\quad entriesToDownload \triangleq \{i \in CommitNumber(p) + 1\ ..\ finishPos : \\
\qquad\qquad\qquad\qquad\qquad \text{New entry for } r \\
\qquad\qquad\qquad\qquad\qquad \lor\ LogLen(p) < i \\
\qquad\qquad\qquad\qquad\qquad \text{Diff in logs}
\end{array}
$$

$$\lor\ Log(p)[i] \neq Log(DownloadReplica(p))[i]\}$$

IN $\land\ entriesToDownload \neq \{\}$

$\land$ LET $ind \triangleq Min(entriesToDownload)$

$finished \triangleq ind = finishPos$

IN $\land\ replicaState' = [replicaState$ EXCEPT $![p].log =$

IF $finished$

THEN $Append(Append(SubSeq(@, 1, ind - 1),$

ELSE $Append(SubSeq(@, 1, ind - 1), Log(Do$

$![p].downloadReplica =$

IF $finished$

THEN $None$

ELSE $@]$

$Er \to Rc$

$RecieveStartView(r) \triangleq$

$\land \exists\, p \in Replica :$

$\land\ p \neq r$

$\land \exists\, view \in ViewNumber(r)\ ..\ ViewNumber(p) :$

$\land\ IsPrimaryInView(p,\ view)$

$\land\ \lor\ view > ViewNumber(r)$

$\lor\ \land\ ViewNumber(r) = view$

$\land\ Status(r) = ViewChange$

$\land\ HasViewBlock(p,\ view)$ p became $Normal$ Master in view

$\land\ replicaState' = [replicaState$ EXCEPT $![r].downloadReplica = p,$

$![r].viewNumber = view,$

$![r].status = Normal]$

$Rc \to Rc\ /\ Rc \to R$

$ReplicaDownloadBeforeView(r) \triangleq$

$\land \neg IsPrimary(r)$

$\land\ Status(r) = Normal$

$\land\ IsDownloading(r)$

$\land$ LET $entriesToDownload \triangleq \{i \in CommitNumber(r) + 1\ ..\ FirstIndexOfViewBlock(Log(DownloadReplic$

New entry for r

$\lor\ LogLen(r) < i$

Diff in logs

$\lor\ Log(r)[i] \neq Log(DownloadReplica(r))[i]\}$

IN $\land\ entriesToDownload \neq \{\}$

$\land$ LET $ind \triangleq Min(entriesToDownload)$

IN $\land\ replicaState' = [replicaState$ EXCEPT $![r].log = Append(SubSeq(@, 1, ind - 1), Log(Down$

$![r].downloadReplica =$

Have just downloaded our View meta Block

IF $Log(DownloadReplica(r))[ind] = [typ$

THEN $None$

ELSE $@]$

$$
\begin{array}{l}
\textit{Finishing} \triangleq \\
\quad \land \text{LET } r \triangleq \textit{ReplicaSequence}[1] \\
\qquad \text{All Committed} \\
\quad \text{IN} \quad \land \textit{CommitNumber}(r) = \textit{OpNumber}(r) \\
\qquad \textit{MaxRequests} \text{ commands are stored} \\
\qquad \land \textit{RequestBlockCount}(\textit{Log}(r)) = \textit{MaxRequests} \\
\qquad \text{All replicas equal} \\
\qquad \land \forall\, r1 \in \textit{Replica} : \\
\qquad\qquad \land \textit{Log}(r1) = \textit{Log}(r) \\
\qquad\qquad \land \textit{CommitNumber}(r1) = \textit{CommitNumber}(r) \\
\qquad\qquad \land \textit{ViewNumber}(r1) = \textit{ViewNumber}(r) \\
\qquad\qquad \land \textit{Status}(r1) = \textit{Normal} \\
\qquad\qquad \land \textit{DownloadReplica}(r1) = \textit{None} \\
\quad \land \text{UNCHANGED } \langle \textit{replicaState} \rangle
\end{array}
$$

$$
\begin{array}{l}
\textit{NormalOperationProtocol} \triangleq \\
\quad \lor \exists\, r \in \textit{Replica},\ op \in \textit{Operation} : \textit{RecieveClientRequest}(r,\ op) \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{RecievePrepare}(r) \\
\quad \lor \exists\, p \in \textit{Replica} : \textit{AchievePrepareOkFromQuorum}(p) \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{RecieveCommit}(r)
\end{array}
$$

$$
\begin{array}{l}
\textit{ViewChangeProtocol} \triangleq \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{TimeoutStartViewChanging}(r) \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{RecieveStartViewChange}(r) \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{AchieveDoViewChangeFromQuorum}(r) \\
\quad \lor \exists\, p \in \textit{Replica} : \textit{MasterDownloadBeforeView}(p) \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{RecieveStartView}(r) \\
\quad \lor \exists\, r \in \textit{Replica} : \textit{ReplicaDownloadBeforeView}(r)
\end{array}
$$

$$
\begin{array}{l}
\textit{Next} \triangleq \lor \textit{NormalOperationProtocol} \\
\qquad\quad \lor \textit{ViewChangeProtocol} \\
\qquad\quad \lor \textit{Finishing}
\end{array}
$$

Full *Spec*

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

$$\textit{FullSpec} \triangleq \textit{Spec} \land \mathrm{SF}_{\textit{vars}}(\textit{Next})$$

INVARIANTS

$$CommitedLogsPrefixesAreEqual \triangleq$$
$$\forall\, r1,\, r2 \in Replica :$$
$$\forall\, i \in \text{DOMAIN } Log(r1)$$
$$\cap \text{DOMAIN } Log(r2)$$
$$\cap\, 1 \,..\, Min(\{CommitNumber(r1),$$
$$CommitNumber(r2)\}) :$$
$$Log(r1)[i] = Log(r2)[i]$$

$$KeepMaxRequests \triangleq \forall\, r \in Replica : RequestBlockCount(Log(r)) \leq MaxRequests$$

$$KeepMaxViews \triangleq \forall\, r \in Replica : ViewNumber(r) + 1 \leq MaxViews$$

THEOREM $Spec \Rightarrow TypeOK$

THEOREM $Spec \Rightarrow CommitedLogsPrefixesAreEqual$

THEOREM $Spec \Rightarrow KeepMaxRequests$

THEOREM $Spec \Rightarrow KeepMaxViews$

Properties

$$EventuallyFinished \triangleq \Diamond(\text{ENABLED } Finishing)$$

THEOREM $FullSpec \Rightarrow EventuallyFinished$

---

* Modification History
* Last modified *Fri* May 05 16:06:53 *MSK* 2023 by *tycoon*
* Created *Wed Dec* 28 15:30:37 *MSK* 2022 by *tycoon*