MODULE *finer_withoutsync*

EXTENDS *Integers*, *FiniteSets*, *Sequences*, *TLC*

Acceptors

CONSTANTS *Server*

CONSTANTS *Follower*, *Candidate*, *Leader*, *LeaderCandidate*

CONSTANTS *Nil*

all messages sent by servers. A function mapping Message to *Nat*

VARIABLE $r1amsgs$, $r1bmsgs$, $r2amsgs$, $r2bmsgs$, $r3amsgs$, $negMsgs$,
$currentTerm$,
$currentState$,
$vote$,
$leaderLog$

$serverVars \triangleq \langle currentTerm, currentState \rangle$

log entries of every server

VARIABLE *log*

$logVars \triangleq \langle log \rangle$

$vars \triangleq \langle r1amsgs, r1bmsgs, r2amsgs, r2bmsgs, r3amsgs, negMsgs, log,$
$currentTerm, currentState, leaderLog, vote \rangle$

$Value \triangleq Nat$

helpers

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

$Index \triangleq Nat$
$Term \triangleq Nat$

$Min(s) \triangleq \text{CHOOSE } i \in s : \forall j \in s : j \geq i$
$Max(s) \triangleq \text{CHOOSE } i \in s : \forall j \in s : i \geq j$

term *sync* 0

$InitServerVars \triangleq \text{LET } k \triangleq \text{CHOOSE } x \in Server : x \in Server$
IN
$\wedge\ currentTerm = [i \in Server \mapsto 0]$
$\wedge\ currentState = [i \in Server \mapsto Follower]$

$InitLogVars \triangleq \wedge\ log = [i \in Server \mapsto [j \in Index \mapsto \langle -1, Nil, \text{FALSE} \rangle]]$

$Init \triangleq \wedge\ r1amsgs = \{\}$
$\wedge\ r1bmsgs = \{\}$
$\wedge\ r2amsgs = \{\}$
$\wedge\ r2bmsgs = \{\}$

$\land\ r3amsgs = \{\}$
$\land\ negMsgs = \{\}$
$\land\ vote = [i\ \in Server \mapsto [b \in Index \mapsto [t \in Term \mapsto Nil]]]$
$\land\ leaderLog = [i \in Term \mapsto [j \in Index \mapsto \langle -1,\ Nil,\ \text{FALSE}\rangle]]$
$\land\ InitServerVars$
$\land\ InitLogVars$

$allEntries \triangleq \{\langle t,\ v,\ b\rangle : t \in Term \cup \{-1\},\ v \in Value \cup \{Nil\},\ b \in \{\text{TRUE},\ \text{FALSE}\}\}$
$logEntries \triangleq \{\langle i,\ e\rangle : i \in Index,\ e \in allEntries\}$

$TypeInv \triangleq$
$\land\ currentTerm \in [Server \to Nat]$
$\land\ currentState \in [Server \to \{Follower,\ Leader,\ LeaderCandidate,\ Candidate\}]$
$\land\ log \in [Server \to [Index \to\ (Term \cup \{-1\}) \times (Value \cup \{Nil\}) \times \text{BOOLEAN}\ ]]$
$\land\ r1amsgs \subseteq \{\langle t,\ i\rangle : t \in Term,\ i \in Server\}$
$\land\ r1bmsgs \subseteq \{\langle t,\ e,\ i,\ j\rangle\ : t \in Term,\ e \in \text{SUBSET}\ logEntries,\ i \in Server,\ j \in Server\}$
$\land\ r2amsgs \subseteq \{\langle t,\ n,\ e,\ i\rangle : t \in Term,\ n \in Index,\ e \in allEntries,\ i \in Server\}$
$\land\ r2bmsgs \subseteq \{\langle t,\ n,\ i,\ j\rangle\ : t \in Term,\ n \in Index,\ i \in Server,\ j \in Server\}$
$\land\ r3amsgs \subseteq \{\langle t,\ n,\ i\rangle : t \in Term,\ n \in Index,\ i \in Server\}$
$\land\ negMsgs \subseteq \{\langle t,\ i\rangle : t \in Term,\ i \in Server\}$
$\land\ log \in [Server \to [Index \to allEntries]]$
$\land\ leaderLog \in [Term \to [Index \to allEntries]]$
$\land\ vote \in [Server \to [Index \to [Term \to Value \cup \{Nil\}]]]$

$lastIndex(i) \triangleq$ IF $\{b \in Index : log[i][b][1] \neq -1\} = \{\}$
THEN -1
ELSE $Max(\{b \in Index : log[i][b][1] \neq -1\})$

$Restart(i) \triangleq$
$\land$ $currentState' = [currentState$ EXCEPT $![i] = Follower]$
$\land$ UNCHANGED $\langle r1amsgs,\ r1bmsgs,\ r2amsgs,\ r2bmsgs,\ r3amsgs,\ negMsgs,$
$currentTerm,\ log,\ leaderLog,\ vote\rangle$

$UpdateTerm(i,\ b) \triangleq$
$\land\ currentTerm[i] < b$
$\land\ currentTerm' = [currentTerm$ EXCEPT $![i] = b]$
$\land\ currentState' = [currentState$ EXCEPT $![i] = Follower]$

$ReceiveHighTerm(i) \triangleq$
$\land\ \exists\, m \in negMsgs :$
$\land\ m[1] > currentTerm[i]$
$\land\ m[2] = i$
$\land\ UpdateTerm(i,\ m[1])$
$\land$ UNCHANGED $\langle log,\ r1amsgs,\ r2amsgs,\ r1bmsgs,\ r2bmsgs,\ r3amsgs,$
$negMsgs,\ leaderLog,\ vote\rangle$

$Timeout(i) \triangleq$

```
    ∧   currentState[i] ∈ {Follower, Candidate}
    ∧   currentTerm' = [currentTerm EXCEPT ![i] = currentTerm[i] + 1]
    ∧   currentState' = [currentState EXCEPT ![i] = Candidate]
    ∧   currentTerm[i] < 5
    ∧   UNCHANGED ⟨r1amsgs, r1bmsgs, log, r2amsgs, r2bmsgs, r3amsgs,
                   negMsgs, leaderLog, vote⟩

RequestVote(i) ≜
    ∧ currentState[i] = Candidate
    ∧ r1amsgs' = r1amsgs ∪ {⟨currentTerm[i], i⟩}
    ∧ UNCHANGED ⟨serverVars, r1bmsgs, log, r2amsgs, r2bmsgs, r3amsgs,
                 negMsgs, leaderLog, vote⟩

 i: recipient
HandleRequestVoteRequest(i) ≜
    ∧ ∃ m ∈ r1amsgs :
        LET j ≜ m[2]
            grant ≜ m[1] > currentTerm[i]
            entries ≜ {⟨n, log[i][n]⟩ : n ∈ Index}
        IN
          ∨ ∧ grant
            ∧ UpdateTerm(i, m[1])
            ∧ r1bmsgs' = r1bmsgs ∪ {⟨m[1], entries, i, j⟩}
            ∧ UNCHANGED negMsgs
          ∨ ∧ ¬grant
            ∧ negMsgs' = negMsgs ∪ {⟨currentTerm[i], j⟩}
            ∧ UNCHANGED ⟨currentState, currentTerm, r1bmsgs⟩
    ∧ UNCHANGED ⟨log, r1amsgs, r2amsgs, r2bmsgs, r3amsgs, vote, leaderLog⟩


Merge(entries, term) ≜  LET
                          committed ≜ {e ∈ entries : e[3] = TRUE}
                          chosen ≜
                        CASE committed = {} → CHOOSE x ∈ entries : ∀ y ∈ entries : x[1] ≥ y[1]
                        □    committed ≠ {} → CHOOSE x ∈ committed : TRUE
                           safe ≜ chosen[2]
                         IN  ⟨term, safe, chosen[3]⟩

BecomeLeaderCandidate(i) ≜
    ∧ currentState[i] = Candidate
    ∧ ∃ Q ∈ Quorums :
        LET voteGranted ≜ {m ∈ r1bmsgs : m[4] = i ∧ m[3] ∈ Q ∧ m[1] = currentTerm[i]}
            allLog ≜ UNION {m[2] : m ∈ voteGranted}
            valid ≜ {e ∈ allLog : e[2][1] ≠ −1}
            end ≜ IF valid = {} THEN −1 ELSE Max({e[1] : e ∈ valid})
        IN
```

$\land \forall\, q \in Q : \exists\, m \in voteGranted : m[3] = q$
$\land\ leaderLog' = [leaderLog \text{ EXCEPT } ![currentTerm[i]] = [n \in Index \mapsto \text{IF } n \in 0\,..\,end \text{ THEN}$
$Merge(\{l[2] : l \in \{t \in allLog : t[1] = n\}\},\ currentTerm[i]) \text{ ELSE } \langle -1,\ Nil,\ \text{FALSE}\rangle]]$
$\land\ currentState' = [currentState \text{ EXCEPT } ![i] = LeaderCandidate]$
$\land \text{ UNCHANGED } \langle currentTerm,\ r1amsgs,\ r2amsgs,\ r1bmsgs,\ r2bmsgs,\ r3amsgs,\ negMsgs,\ log,\ vote\rangle$

$RequestSync(i) \triangleq$
$\land\ currentState[i] \in \{LeaderCandidate,\ Leader\}$
$\land \text{ LET } sync \triangleq \{n \in Index : leaderLog[currentTerm[i]][n][1] \neq -1\} \text{IN}$
$\exists\, n \in sync : r2amsgs' = r2amsgs \cup \{\langle currentTerm[i],\ n,\ leaderLog[currentTerm[i]][n],\ i\rangle\}$
$\land \text{ UNCHANGED } \langle serverVars,\ log,\ r1amsgs,\ r1bmsgs,\ r2bmsgs,\ r3amsgs,\ negMsgs,\ leaderLog,\ vote\rangle$

$HandleRequestSyncRequest(i) \triangleq$
$\land \exists\, m \in r2amsgs :$
$\text{LET } j \triangleq m[4]$
$grant \triangleq m[1] \geq currentTerm[i]$
IN
$\land\ \lor \land m[1] > currentTerm[i]$
$\land\ UpdateTerm(i,\ m[1])$
$\lor \land m[1] \leq currentTerm[i]$
$\land \text{ UNCHANGED } \langle currentTerm,\ currentState\rangle$
$\land\ \lor \land\ grant$
$\land\ log' = [log \text{ EXCEPT } ![i][m[2]] = m[3]]$
$\land\ vote' = [vote \text{ EXCEPT } ![i][m[2]][m[1]] = m[3][2]]$
$\land\ r2bmsgs' = r2bmsgs \cup \{\langle m[1],\ m[2],\ i,\ j\rangle\}$
$\land \text{ UNCHANGED } negMsgs$
$\lor \land \neg grant$
$\land\ negMsgs' = negMsgs \cup \{\langle currentTerm[i],\ j\rangle\}$
$\land \text{ UNCHANGED } \langle vote,\ r2bmsgs,\ log\rangle$
$\land \text{ UNCHANGED } \langle r1amsgs,\ r1bmsgs,\ r2amsgs,\ r3amsgs,\ leaderLog\rangle$

$CommitEntry(i) \triangleq$
$\land \exists\, index \in Index,\ Q \in Quorums :$
$\text{LET } syncSuccess \triangleq \{m \in r2bmsgs :\ m[4] = i \land m[3] \in Q \land$
$m[1] = currentTerm[i] \land m[2] = index\}$
IN
$\land\ currentState[i] \in \{Leader,\ LeaderCandidate\}$
$\land \forall\, q \in Q : \exists\, m \in syncSuccess : m[3] = q$
$\land\ leaderLog' = [leaderLog \text{ EXCEPT } ![currentTerm[i]][index][3] = \text{TRUE}]$
$\land \text{ UNCHANGED } \langle serverVars,\ log,\ r1amsgs,\ r1bmsgs,\ r2amsgs,\ r2bmsgs,\ r3amsgs,\ negMsgs,\ vote\rangle$

$RequestCommit(i) \triangleq$
$\land\ currentState[i] \in \{Leader,\ LeaderCandidate\}$
$\land \text{ LET } committed \triangleq \{n \in Index : leaderLog[currentTerm[i]][n][3] = \text{TRUE}\} \text{IN}$
$\exists\, n \in committed : r3amsgs' = r3amsgs \cup \{\langle currentTerm[i],\ n,\ i\rangle\}$
$\land \text{ UNCHANGED } \langle serverVars,\ log,\ r1amsgs,\ r1bmsgs,\ r2amsgs,\ r2bmsgs,\ negMsgs,\ leaderLog,\ vote\rangle$

$$
\begin{array}{l}
HandleRequestCommitRequest(i) \triangleq \\
\quad \land \exists\, m \in r3amsgs : \\
\qquad \text{LET } grant \triangleq currentTerm[i] \leq\ m[1] \\
\qquad\quad\ \ j \triangleq m[3] \\
\qquad \text{IN} \\
\qquad\ \land \lor \land m[1] > currentTerm[i] \\
\qquad\qquad\ \ \land UpdateTerm(i,\, m[1]) \\
\qquad\qquad \lor \land m[1] \leq currentTerm[i] \\
\qquad\qquad\ \ \land \text{UNCHANGED } \langle currentTerm,\, currentState\rangle \\
\qquad\ \land \lor \land grant \\
\qquad\qquad\ \ \land log[i][m[2]][1] = m[1] \\
\qquad\qquad\ \ \land log' = [log \text{ EXCEPT } ![i][m[2]][3] = \text{TRUE}] \\
\qquad\qquad\ \ \land \text{UNCHANGED } negMsgs \\
\qquad\qquad \lor \land \neg grant \\
\qquad\qquad\ \ \land negMsgs' = negMsgs \cup \{\langle currentTerm[i],\, j\rangle\} \\
\qquad\qquad\ \ \land \text{UNCHANGED } log \\
\quad \land \text{UNCHANGED } \langle serverVars,\, r1amsgs,\, r1bmsgs,\, r2amsgs,\, r2bmsgs,\, r3amsgs,\, leaderLog,\, vote\rangle
\end{array}
$$

$$
\begin{array}{l}
BecomeLeader(i) \triangleq \\
\quad \land currentState[i] = LeaderCandidate \\
\quad \land currentState' = [currentState \text{ EXCEPT } ![i] = Leader] \\
\quad \land \text{UNCHANGED } \langle currentTerm,\, log,\, r1amsgs,\, r1bmsgs,\, r2amsgs,\, r2bmsgs,\, r3amsgs, \\
\qquad\qquad\qquad negMsgs,\, leaderLog,\, vote\rangle
\end{array}
$$

$$
\begin{array}{l}
ClientRequest(i) \triangleq \\
\quad \text{LET } ind \triangleq \{b \in Index : leaderLog[currentTerm[i]][b][1] \neq\ -1\} \\
\qquad\ nextIndex \triangleq \text{ IF } ind = \{\} \\
\qquad\qquad\qquad\qquad \text{THEN } 0 \\
\qquad\qquad\qquad\qquad \text{ELSE } Max(ind) + 1 \\
\quad \text{IN} \\
\quad \land currentState[i] = Leader \\
\quad \land \exists\, v \in Value : leaderLog' = [leaderLog \text{ EXCEPT } ![currentTerm[i]][nextIndex] = \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \langle currentTerm[i],\, v,\, \text{FALSE}\rangle] \\
\quad \land \text{UNCHANGED } \langle serverVars,\, log,\, r1amsgs,\, r1bmsgs,\, r2amsgs,\, r2bmsgs,\, r3amsgs,\, negMsgs,\, vote\rangle
\end{array}
$$

$$
\begin{array}{ll}
Next \triangleq & \lor \exists\, i \in Server : Restart(i) \\
& \lor \exists\, i \in Server : Timeout(i) \\
& \lor \exists\, i \in Server : ReceiveHighTerm(i) \\
& \lor \exists\, i \in Server : RequestVote(i) \\
& \lor \exists\, i \in Server : HandleRequestVoteRequest(i) \\
& \lor \exists\, i \in Server : BecomeLeaderCandidate(i) \\
& \lor \exists\, i \in Server : BecomeLeader(i) \\
& \lor \exists\, i \in Server : CommitEntry(i) \\
& \lor \exists\, i \in Server : ClientRequest(i) \\
& \lor \exists\, i,\, j \in Server : RequestCommit(i) \\
& \lor \exists\, i \in Server : HandleRequestCommitRequest(i)
\end{array}
$$

$\lor \exists\, i, j \in Server : RequestSync(i)$
$\lor \exists\, i \in Server : HandleRequestSyncRequest(i)$

$Inv \triangleq \land TypeInv$

$Acceptors \triangleq Server$
$Ballots \triangleq Term$
$Instances \triangleq Index$
$ballot \triangleq currentTerm$
$leaderVote \triangleq [i \in Ballots \mapsto [j \in Index \mapsto \langle leaderLog[i][j][1],\ leaderLog[i][j][2]\rangle]]$
$1amsgs \triangleq \{\langle m[1]\rangle : m \in r1amsgs\}$
$1bmsgs \triangleq \{\langle m[1],\ \{\langle e[1],\ \langle e[2][1],\ e[2][2]\rangle\rangle : e \in m[2]\},\ m[3]\rangle : m \in r1bmsgs\}$
$2amsgs \triangleq \{\langle m[1],\ m[2],\ \langle m[3][1],\ m[3][2]\rangle\rangle : m \in r2amsgs\}$

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

$A \triangleq$ INSTANCE $multipaxos$

THEOREM $Spec \Rightarrow A!Spec$

\ * Modification History
\ * Last modified *Tue Apr* 21 22:26:20 *CST* 2020 by *assstriker*
\ * Created Sun *Mar* 08 02:26:17 *CST* 2020 by *assstriker*