MODULE *ParallelRaftSE*

EXTENDS *Integers*, *FiniteSets*, *Sequences*, *TLC*
CONSTANTS *Server*, *Follower*, *Candidate*, *Leader*, *LeaderCandidate*, *Nil*, *Value*

$Quorums \triangleq \{i \in \text{SUBSET } (Server) : Cardinality(i) * 2 > Cardinality(Server)\}$
$Index \triangleq \{0, 1, 2, 3, 4, 5, 6\}$
$Term \triangleq Nat$

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

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

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

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

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

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

$$
\begin{array}{l}
\textit{InitServerVars} \triangleq \text{LET } k \triangleq \text{CHOOSE } x \in \textit{Server} : x \in \textit{Server} \\
\qquad\qquad\qquad\qquad \text{IN} \\
\qquad\qquad\qquad\qquad \land\ \textit{currentTerm} = [i \in \textit{Server} \mapsto 0] \\
\qquad\qquad\qquad\qquad \land\ \textit{currentState} = [i \in \textit{Server} \mapsto \textit{Follower}]
\end{array}
$$

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

$$
\begin{array}{ll}
\textit{Init} \triangleq & \land\ r1\textit{amsgs} = \{\} \\
& \land\ r1\textit{bmsgs} = \{\} \\
& \land\ r2\textit{amsgs} = \{\} \\
& \land\ r2\textit{bmsgs} = \{\} \\
& \land\ r3\textit{amsgs} = \{\} \\
& \land\ \textit{negMsgs} = \{\} \\
& \land\ \textit{vote} = [i\ \in \textit{Server} \mapsto [b \in \textit{Index} \mapsto [t \in \textit{Term} \mapsto \textit{Nil}]]] \\
& \land\ \textit{leaderLog} = [i \in \textit{Term} \mapsto [j \in \textit{Index} \mapsto \langle -1,\ \textit{Nil},\ \text{FALSE}\rangle]] \\
& \land\ \textit{InitServerVars} \\
& \land\ \textit{InitLogVars}
\end{array}
$$

$$
\begin{array}{l}
\textit{Restart}(i) \triangleq \\
\quad \land\ \ \textit{currentState}' = [\textit{currentState} \text{ EXCEPT } ![i] = \textit{Follower}] \\
\quad \land\ \text{UNCHANGED } \langle r1\textit{amsgs},\ r1\textit{bmsgs},\ r2\textit{amsgs},\ r2\textit{bmsgs},\ r3\textit{amsgs},\ \textit{negMsgs}, \\
\qquad\qquad\qquad\qquad \textit{currentTerm},\ \textit{log},\ \textit{leaderLog},\ \textit{vote}\rangle \\
\textit{UpdateTerm}(i,\ b) \triangleq \\
\quad \land\ \textit{currentTerm}[i] < b \\
\quad \land\ \textit{currentTerm}' = [\textit{currentTerm} \text{ EXCEPT } ![i] = b] \\
\quad \land\ \textit{currentState}' = [\textit{currentState} \text{ EXCEPT } ![i] = \textit{Follower}]
\end{array}
$$

$$
\begin{array}{l}
\textit{ReceiveHighTerm}(i) \triangleq \\
\quad \land \exists\, m \in \textit{negMsgs} : \\
\qquad\quad \land\ m[1] > \textit{currentTerm}[i] \\
\qquad\quad \land\ m[2] = i \\
\qquad\quad \land\ \textit{UpdateTerm}(i,\ m[1]) \\
\quad \land \text{UNCHANGED } \langle \textit{log},\ r1\textit{amsgs},\ r2\textit{amsgs},\ r1\textit{bmsgs},\ r2\textit{bmsgs},\ r3\textit{amsgs}, \\
\qquad\qquad\qquad\qquad \textit{negMsgs},\ \textit{leaderLog},\ \textit{vote}\rangle
\end{array}
$$

$$
\begin{array}{l}
\textit{Timeout}(i) \triangleq \\
\quad \land\ \ \textit{currentState}[i] \in \{\textit{Follower},\ \textit{Candidate}\} \\
\quad \land\ \ \textit{currentTerm}' = [\textit{currentTerm} \text{ EXCEPT } ![i] = \textit{currentTerm}[i] + 1] \\
\quad \land\ \ \textit{currentState}' = [\textit{currentState} \text{ EXCEPT } ![i] = \textit{Candidate}] \\
\quad \land\ \ \textit{currentTerm}[i] + 1 \in \textit{Nat} \\
\quad \land\ \ \text{UNCHANGED } \langle r1\textit{amsgs},\ r1\textit{bmsgs},\ \textit{log},\ r2\textit{amsgs},\ r2\textit{bmsgs},\ r3\textit{amsgs}, \\
\qquad\qquad\qquad\qquad \textit{negMsgs},\ \textit{leaderLog},\ \textit{vote}\rangle
\end{array}
$$

$$
\begin{array}{l}
\textit{RequestVote}(i) \triangleq \\
\quad \land\ \textit{currentState}[i] = \textit{Candidate}
\end{array}
$$

```
    ∧ r1amsgs' = r1amsgs ∪ {⟨currentTerm[i], i⟩}
    ∧ UNCHANGED ⟨serverVars, r1bmsgs, log, r2amsgs, r2bmsgs, r3amsgs,
                 negMsgs, leaderLog, vote⟩

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, v) ≜
                        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 ≜ IF chosen[2] = Nil THEN v ELSE 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
          ∧ ∀ q ∈ Q : ∃ m ∈ voteGranted : m[3] = q
          ∧ ∃ v ∈ Value : leaderLog' = [leaderLog EXCEPT ![currentTerm[i]] =
                [n ∈ Index ↦ IF n ∈ 0 .. end THEN
                Merge({l[2] : l ∈ {t ∈ allLog : t[1] = n}}, currentTerm[i], v)
                                ELSE ⟨−1, Nil, FALSE⟩]]
    ∧ currentState' = [currentState EXCEPT ![i] = LeaderCandidate]
    ∧ UNCHANGED ⟨currentTerm, r1amsgs, r2amsgs, r1bmsgs, r2bmsgs, r3amsgs, negMsgs, log, vote⟩

RequestSync(i) ≜
```

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

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

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

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

$HandleRequestCommitRequest(i) \triangleq$
$\land\ \exists\, m \in r3amsgs :$
LET $grant \triangleq currentTerm[i] \leq m[1]$
$j \triangleq m[3]$
IN
$\land\ \lor\ \land\ m[1] > currentTerm[i]$

$$
\begin{array}{l}
\quad \land \mathit{UpdateTerm}(i,\ m[1]) \\
\quad \lor \land m[1] \leq \mathit{currentTerm}[i] \\
\quad\quad \land \text{UNCHANGED } \langle \mathit{currentTerm},\ \mathit{currentState} \rangle \\
\land \lor \land \mathit{grant} \\
\quad\quad \land \mathit{log}[i][m[2]][1] = m[1] \\
\quad\quad \land \mathit{log}' = [\mathit{log} \text{ EXCEPT } ![i][m[2]][3] = \text{TRUE}] \\
\quad\quad \land \text{UNCHANGED } \mathit{negMsgs} \\
\quad \lor \land \neg \mathit{grant} \\
\quad\quad \land \mathit{negMsgs}' = \mathit{negMsgs} \cup \{\langle \mathit{currentTerm}[i],\ j \rangle\} \\
\quad\quad \land \text{UNCHANGED } \mathit{log} \\
\land \text{UNCHANGED } \langle \mathit{serverVars},\ r1\mathit{amsgs},\ r1\mathit{bmsgs},\ r2\mathit{amsgs},\ r2\mathit{bmsgs},\ r3\mathit{amsgs},\ \mathit{leaderLog},\ \mathit{vote} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{BecomeLeader}(i) \triangleq \\
\quad \land \mathit{currentState}[i] = \mathit{LeaderCandidate} \\
\quad \land \mathit{currentState}' = [\mathit{currentState} \text{ EXCEPT } ![i] = \mathit{Leader}] \\
\quad \land \text{UNCHANGED } \langle \mathit{currentTerm},\ \mathit{log},\ r1\mathit{amsgs},\ r1\mathit{bmsgs},\ r2\mathit{amsgs},\ r2\mathit{bmsgs},\ r3\mathit{amsgs}, \\
\qquad\qquad \mathit{negMsgs},\ \mathit{leaderLog},\ \mathit{vote} \rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{ClientRequest}(i) \triangleq \\
\quad \text{LET } \mathit{ind} \triangleq \{b \in \mathit{Index} : \mathit{leaderLog}[\mathit{currentTerm}[i]][b][1] \neq -1\} \\
\qquad \mathit{nextIndex} \triangleq \text{IF } \mathit{ind} = \{\} \\
\qquad\qquad\qquad \text{THEN } 0 \\
\qquad\qquad\qquad \text{ELSE } \mathit{Max}(\mathit{ind}) + 1 \\
\quad \text{IN} \\
\quad \land \mathit{currentState}[i] = \mathit{Leader} \\
\quad \land \mathit{nextIndex} \in \mathit{Index} \\
\quad \land \exists\, v \in \mathit{Value} : \mathit{leaderLog}' = [\mathit{leaderLog} \text{ EXCEPT } ![\mathit{currentTerm}[i]][\mathit{nextIndex}] = \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \langle \mathit{currentTerm}[i],\ v,\ \text{FALSE} \rangle] \\
\quad \land \text{UNCHANGED } \langle \mathit{serverVars},\ \mathit{log},\ r1\mathit{amsgs},\ r1\mathit{bmsgs},\ r2\mathit{amsgs},\ r2\mathit{bmsgs},\ r3\mathit{amsgs},\ \mathit{negMsgs},\ \mathit{vote} \rangle
\end{array}
$$

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

$$
\mathit{Inv} \triangleq \land \mathit{TypeInv}
$$

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

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

$A \triangleq \text{INSTANCE } MultiPaxos$

THEOREM $Refinement \triangleq Spec \Rightarrow A!Spec$

\ * Modification History
\ * Last modified *Fri Sep* 11 15:44:23 *CST* 2020 by 15150