MODULE *ParallelRaftSE*

Copyright (c) 2020 *Xiaosong Gu*

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

CONSTANT *Server*

CONSTANT *Value*

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

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

$Index \triangleq \{0, 1, 2, 3, 4, 5, 6\}$

$Term \triangleq Nat$

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

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

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

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

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

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

$\land\ r2amsgs \subseteq \{\langle t,\ n,\ e,\ i\rangle : t \in Term,\ n \in Index,\ e \in allEntries,\ i \in Server\}$
$\land\ r2bmsgs \subseteq \{\langle t,\ n,\ i,\ j\rangle : t \in Term,\ n \in Index,\ i \in Server,\ j \in Server\}$
$\land\ r3amsgs \subseteq \{\langle t,\ n,\ i\rangle : t \in Term,\ n \in Index,\ i \in Server\}$
$\land\ negMsgs \subseteq \{\langle t,\ i\rangle : t \in Term,\ i \in Server\}$
$\land\ log \in [Server \to [Index \to allEntries]]$
$\land\ leaderLog \in [Term \to [Index \to allEntries]]$
$\land\ vote \in [Server \to [Index \to [Term \to Value \cup \{Nil\}]]]$

$Init \triangleq$
$\land\ r1amsgs = \{\}$
$\land\ r1bmsgs = \{\}$
$\land\ r2amsgs = \{\}$
$\land\ r2bmsgs = \{\}$
$\land\ r3amsgs = \{\}$
$\land\ negMsgs = \{\}$
$\land\ currentTerm = [i \in Server \mapsto 0]$
$\land\ currentState = [i \in Server \mapsto Follower]$
$\land\ vote = [i\ \in Server \mapsto [j \in Index \mapsto [t \in Term \mapsto Nil]]]$
$\land\ leaderLog = [i \in Term \mapsto [j \in Index \mapsto \langle -1,\ Nil,\ \text{FALSE}\rangle]]$
$\land\ log = [i \in Server \mapsto [j \in Index \mapsto \langle -1,\ Nil,\ \text{FALSE}\rangle]]$

$Restart(i) \triangleq$ $\land\ currentState' = [currentState\ \text{EXCEPT}\ ![i] = Follower]$
$\land\ \text{UNCHANGED}\ \langle msgsVars,\ currentTerm,\ negMsgs,\ vote,\ leaderLog,\ log\rangle$

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

$ReceiveHighTerm(i) \triangleq$
$\land\ \exists\, m \in negMsgs :\ \land\ m[i] > currentTerm[i]$
$\land\ m[2] = i$
$\land\ UpdateTerm(i,\ m[1])$
$\land\ \text{UNCHANGED}\ \langle msgsVars,\ negMsgs,\ vote,\ leaderLog,\ log\rangle$

$Timeout(i) \triangleq$
$\land\ currentState[i] \in \{Follower,\ Candidate\}$
$\land\ currentTerm' = [currentTerm\ \text{EXCEPT}\ ![i] = currentTerm[i] + 1]$
$\land\ currentState' = [currentState\ \text{EXCEPT}\ ![i] = Candidate]$
$\land\ (currentTerm[i] + 1) \in Nat$
$\land\ \text{UNCHANGED}\ \langle msgsVars,\ negMsgs,\ vote,\ leaderLog,\ log\rangle$

$RequestVote(i) \triangleq$
$\land\quad currentState[i] = Candidate$
$\land\quad r1amsgs' = r1amsgs \cup \{\langle currentTerm[i],\ i\rangle\}$
$\land\quad \text{UNCHANGED}\ \langle serverVars,\ r1bmsgs,\ r2amsgs,\ r2bmsgs,\ r3amsgs,\ negMsgs,\ log,\ leaderLog,\ vote\rangle$

```
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]}   ⟨m[1], entries
            allLog      ≜ UNION {m[2] : m ∈ voteGranted}                                ⟨n, log[i][n]⟩
            valid       ≜ {e ∈ allLog : e[2][1] ≠ −1}                                   log[i][n]: ⟨t, v
            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 ∈
                                                    ELSE ⟨−1, Nil, FALSE⟩]]
    ∧ currentState' = [currentState EXCEPT ![i] = LeaderCandidate]
    ∧ UNCHANGED ⟨currentTerm, log, msgsVars, vote, negMsgs⟩

RequestSync(i) ≜
    ∧  currentState[i] ∈ {LeaderCandidate, Leader}
    ∧  LET sync ≜ {n ∈ Index : leaderLog[currentTerm[i]][n][1] ≠ −1}
       IN ∃ n ∈ sync : r2amsgs' = r2amsgs ∪ {⟨currentTerm[i], n, leaderLog[currentTerm[i]][n], i⟩}
    ∧  UNCHANGED ⟨serverVars, log, leaderLog, vote, r1amsgs, r1bmsgs, r2bmsgs, r3amsgs, negMsgs⟩

HandleRequestSyncRequest(i) ≜
    ∧ ∃ m ∈ r2amsgs :                                   ⟨currentTerm[i], n, leaderLog[currentTerm[i]][n], i⟩
        LET j ≜ m[4]
```

```
                grant ≜ m[1] ≥ currentTerm[i]
            IN  ∧ ∨ ∧ m[1] > currentTerm[i]
                   ∧ UpdateTerm(i, m[1])
                ∨ ∧ m[1] ≤ currentTerm[i]
                   ∧ UNCHANGED ⟨currentTerm, currentState⟩
                ∧ ∨ ∧ grant
                   ∧ log' = [log EXCEPT ![i][m[2]] = m[3]]
                   ∧ vote' = [vote EXCEPT ![i][m[2]][m[1]] = m[3][2]]
                   ∧ r2bmsgs' = r2bmsgs ∪ {⟨m[1], m[2], i, j⟩}
                   ∧ UNCHANGED negMsgs
                ∨ ∧ ¬grant
                   ∧ negMsgs' = negMsgs ∪ {⟨currentTerm[i], j⟩}
                   ∧ UNCHANGED ⟨log, vote, r2bmsgs⟩
        ∧ UNCHANGED ⟨r1amsgs, r1bmsgs, r2amsgs, r3amsgs, leaderLog⟩

CommitEntry(i) ≜
        ∧   currentState[i] ∈ {Leader, LeaderCandidate}
        ∧   ∃ index ∈ Index, Q ∈ Quorums :
            LET syncSuccess ≜ {m ∈ r2bmsgs : ∧ m[4] = i
                                             ∧ m[3] ∈ Q
                                             ∧ m[1] = currentTerm[i]
                                             ∧ m[2] = index}
            IN  ∧ ∀ q ∈ Q : ∃ m ∈ syncSuccess : m[3] = q
                ∧ leaderLog' = [leaderLog EXCEPT ![currentTerm[i]][index][3] = TRUE]
        ∧   UNCHANGED ⟨serverVars, log, msgsVars, negMsgs, vote⟩

RequestCommit(i) ≜
        ∧ currentState[i] ∈ {Leader, LeaderCandidate}
        ∧ LET committed ≜ {n ∈ Index : leaderLog[currentTerm[i]][n][3] = TRUE}
          IN  ∃ n ∈ committed :
                r3amsgs' = r3amsgs ∪ {⟨currentTerm[i], n, i⟩}
        ∧ UNCHANGED ⟨serverVars, log, r1amsgs, r1bmsgs, r2amsgs, r2bmsgs, negMsgs, leaderLog, vote⟩

HandleRequestCommitRequest(i) ≜
        ∧ ∃ m ∈ r3amsgs :
            LET j ≜ m[3]
                grant ≜ currentTerm[i] ≤ m[1]
            IN  ∧ ∨ ∧ m[1] > currentTerm[i]
                   ∧ UpdateTerm(i, m[1])
                ∨ ∧ m[1] ≤ currentTerm[i]
                   ∧ UNCHANGED ⟨currentTerm, currentState⟩
                ∧ ∨ ∧ grant
                   ∧ log[i][m[2]][1] = m[1]
                   ∧ log' = [log EXCEPT ![i][m[2]][3] = TRUE]
                   ∧ UNCHANGED negMsgs
                ∨ ∧ ¬grant
```

$\land\ negMsgs' = negMsgs \cup \{\langle currentTerm[i],\ j\rangle\}$
$\land$ UNCHANGED log
$\land$ UNCHANGED $\langle msgsVars,\ leaderLog,\ vote\rangle$

$BecomeLeader(i) \triangleq$
$\land\ currentState[i] = LeaderCandidate$
$\land\ currentState' = [currentState$ EXCEPT $![i] = Leader]$
$\land$ UNCHANGED $\langle currentTerm,\ log,\ msgsVars,\ negMsgs,\ leaderLog,\ vote\rangle$

$ClientRequest(i) \triangleq$
LET $ind \triangleq \{b \in Index : leaderLog[currentTerm[i]][b][1] \neq -1\}$
$nextIndex \triangleq$ IF $ind = \{\}$ THEN 0
ELSE $Max(ind) + 1$
IN $\land\ currentState[i] = Leader$
$\land\ nextIndex \in Index$
$\land \exists\, v \in Value : leaderLog' = [leaderLog$ EXCEPT $![currentTerm[i]][nextIndex] = \langle currentTerm[i],$
$\land$ UNCHANGED $\langle serverVars,\ log,\ vote,\ msgsVars,\ negMsgs\rangle$

$Next \triangleq \lor \exists\, i \in Server : Restart(i)$
$\lor \exists\, i \in Server : Timeout(i)$
$\lor \exists\, i \in Server : ReceiveHighTerm(i)$
$\lor \exists\, i \in Server : RequestVote(i)$
$\lor \exists\, i \in Server : HandleRequestVoteRequest(i)$
$\lor \exists\, i \in Server : BecomeLeaderCandidate(i)$
$\lor \exists\, i \in Server : BecomeLeader(i)$
$\lor \exists\, i \in Server : CommitEntry(i)$
$\lor \exists\, i \in Server : ClientRequest(i)$
$\lor \exists\, i, j \in Server : RequestCommit(i)$
$\lor \exists\, i \in Server : HandleRequestCommitRequest(i)$
$\lor \exists\, i, j \in Server : RequestSync(i)$
$\lor \exists\, i \in Server : HandleRequestSyncRequest(i)$

\ * Modification History
\ * Last modified *Tue* May 11 19:47:37 *CST* 2021 by Dell
\ * Created *Mon* May 10 22:09:59 *CST* 2021 by Dell