MODULE *UpdateCluster*

EXTENDS *Integers*, *FiniteSets*

CONSTANT
Reqs, the requests sent by the user
NULL

VARIABLES
queue,
confOK, are we able to get a valid conf ?
lastSubmitted, last cluster submitted
clusterState, last cluster fully deployed
reqState the state of all requests (*reqState*[*req*])

$vars \triangleq \langle confOK, clusterState, reqState, queue, lastSubmitted \rangle$

$NoConcurrentUpdate \triangleq$
$Cardinality(\{r \in \text{DOMAIN } reqState : reqState[r].status = \text{“partial”}\}) < 2$

$TypeInvariants \triangleq$
$\land\ confOK \in \text{BOOLEAN}$ won't change for a specific behavior
$\land\ NoConcurrentUpdate$
$\land\ \forall\, r \in Reqs : reqState[r].status \in \{$
“waiting”, the request (*req*) hasn't been submitted yet
“submitted”, *req* has been submitted
“rejected”, *req* has been rejected (auth problem)
“valid”, auth etc passed
“processing”, the processing of the *req* has started
“partial”, *req* is partially applied (the infra is partially modified)
“partialFailure”, *req* failed in the middle of an application
“success”, *req* has been sucessfully applied
“failure”, the *req* failed before modifying the cluster
“rolledback” the *req* has been *rolledback*
$\}$

Initial State

$Init \triangleq$
$\land\ reqState = [r \in Reqs \mapsto [status \mapsto \text{“waiting”}, rank \mapsto NULL]]$
$\land\ clusterState = [req \mapsto NULL, complete \mapsto \text{TRUE}]$
$\land\ lastSubmitted = 0$
$\land\ confOK \in \text{BOOLEAN}$
$\land\ queue = \{\}$

Actions

$Submit(r) \triangleq$ update request received from the user
LET $currSubmitRank \triangleq lastSubmitted + 1$ IN
$\wedge\ reqState[r].status =$ "waiting"
$\wedge\ lastSubmitted' = currSubmitRank$
$\wedge\ reqState' = [reqState$ EXCEPT $![r].status =$ "submitted", $![r].rank = currSubmitRank]$
$\wedge$ UNCHANGED $\langle confOK, queue, clusterState\rangle$

$Initialcheck(r) \triangleq$ request validation (auth, quotas . . .)
$\wedge\ reqState[r].status =$ "submitted"
$\wedge\ \exists\, ok \in$ BOOLEAN :
IF ok
THEN
$reqState' = [reqState$ EXCEPT $![r].status =$ "valid"$]$
ELSE
$reqState' = [reqState$ EXCEPT $![r].status =$ "rejected"$]$
$\wedge$ UNCHANGED $\langle confOK, queue, clusterState, lastSubmitted\rangle$

$PushToQueue(r) \triangleq$ the request is pushed to queue
$\wedge\ reqState[r].status =$ "valid"
$\wedge\ queue' = queue \cup \{r\}$
$\wedge$ UNCHANGED $\langle confOK, reqState, clusterState, lastSubmitted\rangle$

$ProcessQueue \triangleq$ a request in queue is processed (order of submission not took into account)
$\wedge\ queue \neq \{\}$
$\wedge\ \exists\, r \in queue$:
$\wedge\ queue' = queue \setminus \{r\}$
$\wedge\ reqState' =$ $[reqState$ EXCEPT $![r].status =$ "processing"$]$
$\wedge$ UNCHANGED $\langle confOK, clusterState, lastSubmitted\rangle$

$StartApply(r) \triangleq$ the cluster starts to be modified
$\wedge\ reqState[r].rank = lastSubmitted$
$\wedge\ reqState[r].status =$ "processing"
$\wedge\ clusterState.complete =$ TRUE
$\wedge\ \neg\exists\, x \in$ DOMAIN $reqState : reqState[x].status =$ "partial" don't attempt if an attempt is already on-going
$\wedge$ IF $confOK$
THEN
$\wedge\ reqState' = [reqState$ EXCEPT $![r].status =$ "partial"$]$
$\wedge\ clusterState' = [req \mapsto r,\ complete \mapsto$ FALSE$]$
$\wedge$ UNCHANGED $\langle confOK, lastSubmitted, queue\rangle$
ELSE
$\wedge\ reqState' = [reqState$ EXCEPT $![r].status =$ "failure"$]$
$\wedge$ UNCHANGED $\langle confOK, clusterState, lastSubmitted, queue\rangle$

$CompleteApply(r) \triangleq$ the cluster update finishes

$\land\ reqState[r].status = \text{“partial”}$
$\land\ \exists\, ok \in \text{BOOLEAN} :$
IF ok
THEN
$\land\ reqState' = [reqState \text{ EXCEPT } ![r].status = \text{“success”}]$
$\land\ clusterState' = [clusterState \text{ EXCEPT } !.req = r,\ !.complete = \text{TRUE}]$
$\land\ \text{UNCHANGED } \langle confOK,\ lastSubmitted,\ queue\rangle$
ELSE
$\land\ reqState' = [reqState \text{ EXCEPT } ![r].status = \text{“partialFailure”}]$
$\land\ \text{UNCHANGED } \langle confOK,\ clusterState,\ lastSubmitted,\ queue\rangle$

$Rollback(r) \triangleq$ we assume rollback always works if *confOK* (ie we shouldn't have to rollback if conf not *OK*)

$\land\ reqState[r].status = \text{“partialFailure”}$
$\land$ IF $confOK$
THEN
$\land\ reqState' = [reqState \text{ EXCEPT } ![r].status = \text{“rolledback”}]$
$\land\ clusterState' = [clusterState \text{ EXCEPT } !.complete = \text{TRUE}]$
$\land\ \text{UNCHANGED } \langle confOK,\ lastSubmitted,\ queue\rangle$
ELSE
UNCHANGED $vars$

Requirements

$NoPartialUpdateTermination \triangleq$ we don't want the cluster to end up in a partially update state

$\Diamond\Box(clusterState.complete = \text{TRUE})$

$NoApplicationOfOutdatedReq \triangleq$ we don't want transition updates of successive requests

$\Box(\{r \in \text{DOMAIN } reqState : \text{ENABLED } StartApply(r) \land reqState[r].rank \neq lastSubmitted\} = \{\})$

$EveryReqInQueueIsProcessed \triangleq$ we don't want messages to stay in queue

$\Diamond\Box(queue = \{\})$

Spec

$Fairness \triangleq$

$\forall\, r \in Reqs :$
$\land\ \text{WF}_{vars}(PushToQueue(r))$
$\land\ \text{WF}_{vars}(ProcessQueue)$
$\land\ \text{WF}_{vars}(CompleteApply(r))$
$\land\ \text{WF}_{vars}(Rollback(r))$

$Next \triangleq$

$ProcessQueue \lor$
$\exists\, r \in Reqs :$
$\lor Submit(r) \lor Initialcheck(r)$
$\lor PushToQueue(r)$
$\lor StartApply(r) \lor CompleteApply(r)$
$\lor Rollback(r)$

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

THEOREM $Spec \Rightarrow \Box(TypeInvariants)$
THEOREM $Spec \Rightarrow NoPartialUpdateTermination$
THEOREM $Spec \Rightarrow NoApplicationOfOutdatedReq$
THEOREM $Spec \Rightarrow EveryReqInQueueIsProcessed$