MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

GenerateTestCases $\triangleq$ FALSE

Nil $\triangleq$ "<nil>"

Change $\triangleq$ "Change"
Rollback $\triangleq$ "Rollback"

Commit $\triangleq$ "Commit"
Apply $\triangleq$ "Apply"

Pending $\triangleq$ "Pending"
InProgress $\triangleq$ "InProgress"
Complete $\triangleq$ "Complete"
Aborted $\triangleq$ "Aborted"
Failed $\triangleq$ "Failed"
Done $\triangleq$ {*Complete*, *Aborted*, *Failed*}

Node $\triangleq$ {"node1"}

NumTransactions $\triangleq$ 4
NumTerms $\triangleq$ 2
NumConns $\triangleq$ 2
NumStarts $\triangleq$ 2

Path $\triangleq$ {"path1"}
Value $\triangleq$ {"value1", "value2"}

A transaction log. Transactions may either request a set
of changes to a set of targets or rollback a prior change.
VARIABLE *transaction*

A record of per-target proposals
VARIABLE *proposal*

A record of per-target configurations
VARIABLE *configuration*

A record of target masterships
VARIABLE *mastership*

A record of node connections to the target
VARIABLE *conn*

The target state
VARIABLE *target*

A sequence of state changes used for model checking.
VARIABLE *history*

$vars \triangleq \langle transaction, proposal, configuration, mastership, conn, target, history \rangle$

LOCAL $Transaction \triangleq$ INSTANCE *Transaction*

LOCAL $Proposal \triangleq$ INSTANCE *Proposal*

LOCAL $Configuration \triangleq$ INSTANCE *Configuration*

LOCAL $Mastership \triangleq$ INSTANCE *Mastership*

LOCAL $Target \triangleq$ INSTANCE *Target*

$RequestChange(p, v) \triangleq$
$\quad \land Transaction!RequestChange(p, v)$
$\quad \land$ UNCHANGED $\langle mastership, conn, target, history \rangle$

$RequestRollback(i) \triangleq$
$\quad \land Transaction!RequestRollback(i)$
$\quad \land$ UNCHANGED $\langle mastership, conn, target, history \rangle$

$ReconcileTransaction(n, i) \triangleq$
$\quad \land i \in$ DOMAIN $transaction$
$\quad \land Transaction!ReconcileTransaction(n, i)$
$\quad \land$ UNCHANGED $\langle mastership, conn, target, history \rangle$
$\quad \land GenerateTestCases \Rightarrow Transaction!Test!Log([node \mapsto n, index \mapsto i])$

$ReconcileProposal(n, i) \triangleq$
$\quad \land i \in$ DOMAIN $proposal$
$\quad \land Proposal!ReconcileProposal(n, i)$
$\quad \land$ UNCHANGED $\langle transaction \rangle$
$\quad \land GenerateTestCases \Rightarrow Proposal!Test!Log([node \mapsto n, index \mapsto i])$

$ReconcileConfiguration(n) \triangleq$
$\quad \land Configuration!ReconcileConfiguration(n)$

$\land$ UNCHANGED $\langle transaction, proposal, history \rangle$
$\land$ $GenerateTestCases \Rightarrow Configuration!Test!Log([node \mapsto n])$

$ReconcileMastership(n) \triangleq$
$\land$ $Mastership!ReconcileMastership(n)$
$\land$ UNCHANGED $\langle transaction, proposal, configuration, target, history \rangle$
$\land$ $GenerateTestCases \Rightarrow Mastership!Test!Log([node \mapsto n])$

$ConnectNode(n) \triangleq$
$\land$ $Target!Connect(n)$
$\land$ UNCHANGED $\langle transaction, proposal, configuration, mastership, history \rangle$

$DisconnectNode(n) \triangleq$
$\land$ $Target!Disconnect(n)$
$\land$ UNCHANGED $\langle transaction, proposal, configuration, mastership, history \rangle$

$StartTarget \triangleq$
$\land$ $Target!Start$
$\land$ UNCHANGED $\langle transaction, proposal, configuration, mastership, history \rangle$

$StopTarget \triangleq$
$\land$ $Target!Stop$
$\land$ UNCHANGED $\langle transaction, proposal, configuration, mastership, history \rangle$

Formal specification, constraints, and theorems.

$Init \triangleq$
$\land$ $transaction = [$
$i \in \{\} \mapsto [$
$type \mapsto Change,$
$index \mapsto 0,$
$values \mapsto [p \in \{\} \mapsto Nil],$
$commit \mapsto Pending,$
$apply \mapsto Pending]]$
$\land$ $proposal = [$
$i \in \{\} \mapsto [$
$change \mapsto [$
$phase \mapsto Nil,$
$state \mapsto Nil,$
$values \mapsto [$
$p \in \{\} \mapsto [$
$index \mapsto 0,$
$value \mapsto Nil]]],$
$rollback \mapsto [$
$phase \mapsto Nil,$
$state \mapsto Nil,$

$$
\begin{array}{l}
\qquad\qquad\qquad \mathit{values} \mapsto [\\
\qquad\qquad\qquad\quad p \in \{\} \mapsto [\\
\qquad\qquad\qquad\qquad \mathit{index} \mapsto 0, \\
\qquad\qquad\qquad\qquad \mathit{value} \mapsto \mathit{Nil}]]]]] \\
\quad \land \mathit{configuration} = [\\
\qquad\quad \mathit{state} \mapsto \mathit{InProgress}, \\
\qquad\quad \mathit{term} \mapsto 0, \\
\qquad\quad \mathit{committed} \mapsto [\\
\qquad\qquad \mathit{index} \mapsto 0, \\
\qquad\qquad \mathit{revision} \mapsto 0, \\
\qquad\qquad \mathit{values} \mapsto [\\
\qquad\qquad\quad p \in \{\} \mapsto [\\
\qquad\qquad\qquad \mathit{index} \mapsto 0, \\
\qquad\qquad\qquad \mathit{value} \mapsto \mathit{Nil}]]], \\
\qquad\quad \mathit{applied} \mapsto [\\
\qquad\qquad \mathit{index} \mapsto 0, \\
\qquad\qquad \mathit{revision} \mapsto 0, \\
\qquad\qquad \mathit{target} \mapsto 0, \\
\qquad\qquad \mathit{values} \mapsto [\\
\qquad\qquad\quad p \in \{\} \mapsto [\\
\qquad\qquad\qquad \mathit{index} \mapsto 0, \\
\qquad\qquad\qquad \mathit{value} \mapsto \mathit{Nil}]]]] \\
\quad \land \mathit{target} = [\\
\qquad\quad \mathit{id} \mapsto 0, \\
\qquad\quad \mathit{running} \mapsto \text{FALSE}, \\
\qquad\quad \mathit{values} \mapsto [\\
\qquad\qquad p \in \{\} \mapsto [\\
\qquad\qquad\quad \mathit{index} \mapsto 0, \\
\qquad\qquad\quad \mathit{value} \mapsto \mathit{Nil}]]] \\
\quad \land \mathit{mastership} = [\\
\qquad\quad \mathit{master} \mapsto \mathit{Nil}, \\
\qquad\quad \mathit{term} \mapsto 0, \\
\qquad\quad \mathit{conn} \mapsto 0] \\
\quad \land \mathit{conn} = [\\
\qquad\quad n \in \mathit{Node} \mapsto [\\
\qquad\qquad \mathit{id} \mapsto 0, \\
\qquad\qquad \mathit{connected} \mapsto \text{FALSE}]] \\
\quad \land \mathit{history} = \langle\rangle
\end{array}
$$

$$
\begin{array}{l}
\mathit{Next} \triangleq \\
\quad \lor \exists\, p \in \mathit{Path},\ v \in \mathit{Value} : \\
\qquad \mathit{RequestChange}(p,\ v) \\
\quad \lor \exists\, i \in \text{DOMAIN } \mathit{transaction} : \\
\qquad \mathit{RequestRollback}(i) \\
\quad \lor \exists\, n \in \mathit{Node} :
\end{array}
$$

$\exists\, i \in \text{DOMAIN } transaction :$
$ReconcileTransaction(n,\, i)$
$\lor \exists\, n \in Node :$
$\exists\, i \in \text{DOMAIN } proposal :$
$ReconcileProposal(n,\, i)$
$\lor \exists\, n \in Node :$
$ReconcileConfiguration(n)$
$\lor \exists\, n \in Node :$
$ReconcileMastership(n)$
$\lor \exists\, n \in Node :$
$\lor ConnectNode(n)$
$\lor DisconnectNode(n)$
$\lor StartTarget$
$\lor StopTarget$

$Spec \triangleq$
$\land Init$
$\land \Box[Next]_{vars}$
$\land \forall\, p \in Path,\, v \in Value :$
$\text{WF}_{\langle transaction,\, proposal,\, configuration,\, mastership,\, target\rangle}(Transaction!RequestChange(p,\, v))$
$\land \forall\, i \in 1\,..\,NumTransactions : i \in \text{DOMAIN } transaction \Rightarrow$
$\text{WF}_{\langle transaction,\, proposal,\, configuration,\, mastership,\, target\rangle}(Transaction!RequestRollback(i))$
$\land \forall\, n \in Node,\, i \in 1\,..\,NumTransactions :$
$\text{WF}_{\langle transaction,\, proposal,\, configuration,\, mastership,\, target\rangle}(Transaction!ReconcileTransaction(n,\, i))$
$\land \forall\, n \in Node,\, i \in 1\,..\,NumTransactions :$
$\text{WF}_{\langle proposal,\, configuration,\, mastership,\, conn,\, target,\, history\rangle}(Proposal!ReconcileProposal(n,\, i))$
$\land \forall\, n \in Node :$
$\text{WF}_{\langle configuration,\, mastership,\, conn,\, target\rangle}(Configuration!ReconcileConfiguration(n))$
$\land \forall\, n \in Node :$
$\text{WF}_{\langle mastership,\, conn\rangle}(Mastership!ReconcileMastership(n))$
$\land \forall\, n \in Node :$
$\text{WF}_{\langle conn,\, target\rangle}(Target!Connect(n) \lor Target!Disconnect(n))$
$\land \text{WF}_{\langle conn,\, target\rangle}(Target!Start \lor Target!Stop)$

$LimitTransactions \triangleq Len(transaction) \leq NumTransactions$

$LimitTerms \triangleq$
$\lor mastership.term < NumTerms$
$\lor \land mastership.term = NumTerms$
$\land mastership.master \neq Nil$

$LimitConns \triangleq$
$\forall\, n \in \text{DOMAIN } conn :$
$\lor conn[n].id < NumConns$

$\lor \land conn[n].id = NumConns$
$\quad \land conn[n].connected$

$LimitStarts \triangleq$
$\lor target.id < 2$
$\lor \land target.id = 2$
$\quad \land target.running$

$TypeOK \triangleq$
$\land Transaction!TypeOK$
$\land Proposal!TypeOK$
$\land Configuration!TypeOK$
$\land Mastership!TypeOK$

LOCAL $IsOrderedChange(p, i) \triangleq$
$\land \quad history[i].type = Change$
$\land \quad history[i].phase = p$
$\land \quad \neg\exists j \in \text{DOMAIN } history :$
$\qquad \land j < i$
$\qquad \land history[j].type = Change$
$\qquad \land history[j].phase = p$
$\qquad \land history[j].index \geq history[i].index$

LOCAL $IsOrderedRollback(p, i) \triangleq$
$\land \quad history[i].type = Rollback$
$\land \quad history[i].phase = p$
$\land \quad \neg\exists j \in \text{DOMAIN } history :$
$\qquad \land j < i$
$\qquad \land history[j].type = Change$
$\qquad \land history[j].phase = p$
$\qquad \land history[j].index > history[i].index$
$\qquad \land \neg\exists k \in \text{DOMAIN } history :$
$\qquad\quad \land k > j$
$\qquad\quad \land k < i$
$\qquad\quad \land history[k].type = Rollback$
$\qquad\quad \land history[k].phase = p$
$\qquad\quad \land history[k].index = history[j].index$

$Order \triangleq$
$\land \forall i \in \text{DOMAIN } history :$
$\quad \lor IsOrderedChange(Commit, i)$
$\quad \lor IsOrderedChange(Apply, i)$
$\quad \lor IsOrderedRollback(Commit, i)$
$\quad \lor IsOrderedRollback(Apply, i)$
$\land \forall i \in \text{DOMAIN } proposal :$

$\land\ proposal[i].change.phase = Apply$
$\land\ proposal[i].change.state\ = Failed$
$\land\ proposal[i].rollback.phase = Apply \Rightarrow proposal[i].rollback.state \neq Complete$
$\Rightarrow \forall\, j \in \text{DOMAIN}\ proposal : (j > i \Rightarrow$
$(proposal[j].change.phase = Apply \Rightarrow$
$proposal[j].change.state \in \{Nil,\ Pending,\ Aborted\}))$

$Consistency\ \triangleq$
$\land \forall\, i \in \text{DOMAIN}\ proposal :$
$\lor\ configuration.committed.index < i$
$\lor\ configuration.committed.revision < i$
$\Rightarrow \neg\exists\, p \in \text{DOMAIN}\ configuration.committed.values :$
$configuration.committed.values[p].index = i$
$\land \forall\, i \in \text{DOMAIN}\ proposal :$
$\lor\ configuration.applied.index < i$
$\lor\ configuration.applied.revision < i$
$\Rightarrow\ \land \neg\exists\, p \in \text{DOMAIN}\ configuration.applied.values :$
$configuration.applied.values[p].index = i$
$\land \neg\exists\, p \in \text{DOMAIN}\ target.values :$
$target.values[p].index = i$
$\land\ \land\ target.running$
$\land\ configuration.applied.target = target.id$
$\land\ configuration.state = Complete$
$\Rightarrow \forall\, i \in \text{DOMAIN}\ proposal :$
$\land\ configuration.applied.index \geq i$
$\land\ configuration.applied.revision \geq i$
$\Rightarrow \forall\, p \in \text{DOMAIN}\ proposal[i].change.values :$
$\land \neg\exists\, j \in \text{DOMAIN}\ proposal :$
$\land\ j > i$
$\land\ configuration.applied.index \geq j$
$\land\ configuration.applied.revision \geq j$
$\Rightarrow\ \land\ p \in \text{DOMAIN}\ target.values$
$\land\ target.values[p].value\ = proposal[i].change.values[p].value$
$\land\ target.values[p].index = proposal[i].change.values[p].index$

$Safety\ \triangleq\ \Box(Order \land Consistency)$

THEOREM $Spec \Rightarrow Safety$

$Terminates(i)\ \triangleq$
$\land\ i \in \text{DOMAIN}\ transaction$
$\land\ transaction[i].commit \in Done$
$\land\ transaction[i].apply \in Done$
$\land\ transaction[i].index \in \text{DOMAIN}\ proposal$
$\land\ \lor\ \land\ transaction[i].type = Change$
$\land\ \lor\ \land\ proposal[transaction[i].index].change.phase = Commit$

$$
\begin{array}{l}
\quad \land\ proposal[transaction[i].index].change.state\ \in \{Aborted,\ Failed\} \\
\lor\ \land\ proposal[transaction[i].index].change.phase = Apply \\
\quad \land\ proposal[transaction[i].index].change.state\ \in Done \\
\lor\ \land\ transaction[i].type = Rollback \\
\quad \land\ \lor\ \land\ proposal[transaction[i].index].rollback.phase = Commit \\
\quad \quad \land\ proposal[transaction[i].index].rollback.state\ \in \{Aborted,\ Failed\} \\
\quad \lor\ \land\ proposal[transaction[i].index].rollback.phase = Apply \\
\quad \quad \land\ proposal[transaction[i].index].rollback.state\ \in Done
\end{array}
$$

$Termination \triangleq$
$\quad \forall\, i \in 1\,..\,NumTransactions : \Diamond Terminates(i)$

$Liveness \triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$