MODULE *DistFinish*

Resilient distributed finish as implemented in *PPoPP*14 See *FinishState.FinishResilientDistributed*

EXTENDS *Integers*, *Sequences*

CONSTANTS *PLACE*, *MXFINISHES*, *PROG_HOME*, *BACKUP*

VARIABLES *fid*, *fstates*, *msgs*, *thrds*, *pstate*, *waitForMsgs*, *killed*, *fbackups*, *seq*

INSTANCE *Commons*

$$Terminated \triangleq \\ \quad \land fstates[fid].status = \text{“forgotten”}$$

$$Running \triangleq \\ \quad \land fstates[fid].status = \text{“waiting”}$$

$$IsRoot \triangleq \\ \quad \land fstates[fid].type = \text{“distroot”}$$

$$LastActivity \triangleq \\ \quad \land fstates[fid].count = 1$$

$$\begin{array}{l} SendMasterAddChild(eroot,\ erootPlace,\ here) \triangleq \\ \quad \land SendMsg([mid \mapsto seq.mseq, \\ \qquad\qquad src \mapsto here, \\ \qquad\qquad dst \mapsto erootPlace, \\ \qquad\qquad eroot \mapsto eroot, \\ \qquad\qquad fid \mapsto fid, \\ \qquad\qquad type \mapsto \text{“addChild”}]) \\ \quad \land IncrMSEQ(1) \\ \quad \land waitForMsgs' = waitForMsgs \cup \{[src \mapsto erootPlace, \\ \qquad\qquad\qquad\qquad\qquad\qquad dst \mapsto here, \\ \qquad\qquad\qquad\qquad\qquad\qquad fid \mapsto fid, \\ \qquad\qquad\qquad\qquad\qquad\qquad eroot \mapsto eroot, \\ \qquad\qquad\qquad\qquad\qquad\quad type \mapsto \text{“addChildDone”}\ ]\} \end{array}$$

$$\begin{array}{l} SendMasterTransit(dst) \triangleq \\ \quad \land dst \neq fstates[fid].here \\ \quad \land \text{LET}\ parentId \triangleq fstates[fid].parent \\ \qquad\quad here \triangleq fstates[fid].here \\ \qquad\quad root \triangleq fstates[fid].root \\ \qquad\quad rootPlace \triangleq GetFinishHome(fstates[fid].root) \\ \qquad \text{IN} \quad \land SendMsg([mid \mapsto seq.mseq, \\ \qquad\qquad\qquad\qquad\quad src \mapsto here, \\ \qquad\qquad\qquad\qquad\quad dst \mapsto rootPlace, \\ \qquad\qquad\qquad\qquad target \mapsto dst, \\ \qquad\qquad\qquad\qquad\quad fid \mapsto root, \\ \qquad\qquad\qquad\qquad type \mapsto \text{“masterTransit”}]) \end{array}$$

$$\land\ waitForMsgs' = waitForMsgs \cup \{[src \mapsto rootPlace,\ dst \mapsto here,\ target \mapsto dst,\ fid \mapsto root,\ type \mapsto \text{“masterTransitDone”}\ ]\}$$
$$\land\ IncrMSEQ(1)$$

$SendMasterTransitToLive(src,\ actId,\ inMsg,\ here,\ root)\ \triangleq$
LET $rootPlace\ \triangleq\ GetFinishHome(root)$
IN $\land\ ReplaceMsg(inMsg,\ [mid \mapsto seq.mseq,\ src \mapsto here,\ source \mapsto src,\ target \mapsto here,\ dst \mapsto rootPlace,\ fid \mapsto root,$ always refer to the root state $aid \mapsto actId,\ type \mapsto \text{“masterLive”}])$
$\land\ waitForMsgs' = waitForMsgs \cup \{[src \mapsto rootPlace,\ dst \mapsto here,\ fid \mapsto root,\ aid \mapsto actId,\ source \mapsto src,\ target \mapsto here,\ type \mapsto \text{“masterLiveDone”}\ ]\}$

$SendMasterLiveToCompleted(finishEnd)\ \triangleq$
LET $root\ \triangleq\ fstates[fid].root$
$rootPlace\ \triangleq\ GetFinishHome(fstates[fid].root)$
$here\ \triangleq\ fstates[fid].here$
IN $\land\ SendMsg([mid \mapsto seq.mseq,\ src \mapsto here,\ dst \mapsto rootPlace,\ target \mapsto here,\ fid \mapsto root,$ always refer to the root state $finishEnd \mapsto finishEnd,\ type \mapsto \text{“masterCompleted”}])$
$\land\ waitForMsgs' = waitForMsgs \cup \{[src \mapsto rootPlace,\ dst \mapsto here,\ target \mapsto here,\ fid \mapsto root,\ isAdopter \mapsto \text{FALSE},\ type \mapsto \text{“masterCompletedDone”}\ ]\}$
$\land\ IncrMSEQ(1)$

$Alloc(type,\ here,\ parent,\ root) \triangleq$
LET $encRoot \triangleq GetEnclosingRoot(parent,\ fid)$
$encRootPlace \triangleq$ IF $fid = FIRST\_ID$ THEN $PROG\_HOME$ ELSE $fstates[encRoot].here$
IN $\land fstates[fid].status =$ "unused"
$\land fstates' = [fstates$ EXCEPT $![fid].id = fid,$
$![fid].count = 1,$
$![fid].status =$ "waiting",
$![fid].type = type,$
$![fid].here = here,$
$![fid].parent = parent,$
$![fid].root = root,$
$![fid].eroot = encRoot,$
$![fid].isGlobal =$ IF $type =$ "distremote"
THEN TRUE
ELSE FALSE]

needed for the local path of $Runtime.runAsync$
$NotifyLocalActivitySpawnAndCreation(here,\ act) \triangleq$
$\land fstates[fid].status =$ "waiting"
$\land fstates' = [fstates$ EXCEPT $![fid].count = @ + 1]$

$NotifySubActivitySpawn(dst) \triangleq$
$\land fstates[fid].status =$ "waiting"
$\land fstates' = [fstates$ EXCEPT $![fid].isGlobal =$ TRUE]
$\land SendMasterTransit(dst)$

$AllocRemoteAndNotifyRemoteActivityCreation(src,\ act,\ inMsg,\ type,\ here,\ parent,\ root) \triangleq$
$\land SendMasterTransitToLive(src,\ act.aid,\ inMsg,\ here,\ root)$
$\land here \neq NotPlace$
$\land type =$ "distremote" create and notify
$\land Alloc(type,\ here,\ parent,\ root)$

$NotifyActivityTermination(finishEnd) \triangleq$
$\land\ fstates[fid].status =$ "waiting"
$\land fstates[fid].count > 0$
$\land$ IF $LastActivity \land \neg fstates[fid].isGlobal$
THEN $\land fstates' = [fstates$ EXCEPT $![fid].count = @ - 1,$
$![fid].status =$ "forgotten"]
$\land msgs' = msgs$
$\land seq' = seq$
$\land waitForMsgs' = waitForMsgs$
ELSE IF $LastActivity \land fstates[fid].isGlobal$
THEN $\land SendMasterLiveToCompleted(finishEnd)$
$\land fstates' = [fstates$ EXCEPT $![fid].count = @ - 1,$
$![fid].status =$ IF $fstates[fid].type =$ "distremote"
THEN "forgotten"
ELSE "pendingRelease"]

$$\begin{array}{ll} \text{ELSE} & \land\, fstates' = [fstates \text{ EXCEPT } ![fid].count = @ - 1] \\ & \land\, msgs' = msgs \\ & \land\, seq' = seq \\ & \land\, waitForMsgs' = waitForMsgs \end{array}$$

* Modification History
* Last modified *Mon Dec* 11 16:46:52 *AEDT* 2017 by *u*5482878
* Last modified Sun *Dec* 10 12:28:32 *AEDT* 2017 by *shamouda*
* Created *Tue Nov* 07 17:50:59 *AEDT* 2017 by *u*5482878