MODULE *Commons*

EXTENDS *Integers*, *Sequences*
VARIABLES *msgs*, *fstates*, *thrds*, *waitForMsgs*, *killed*, *seq*
CONSTANTS *PLACE*, *MXFINISHES*, *PROG_HOME*, *BACKUP*

$ROOT\_FINISH \triangleq$ "distroot"

$REMOTE\_FINISH \triangleq$ "distremote"

$MXTHREADS \triangleq 2$
$MXACTIVITIES \triangleq 20$
$MXMESSAGES \triangleq 200$

$MXFID \triangleq MXFINISHES + 1$

$NotID \triangleq -1$

$NoParent \triangleq 0$

$FIRST\_ID \triangleq 1$

$PIDRange \triangleq NoParent \,..\, MXFID$

$IDRange \triangleq FIRST\_ID \,..\, MXFID$

$NotPlace \triangleq$ CHOOSE $v : v \notin PLACE$

$ThreadID \triangleq 0 \,..\, MXTHREADS - 1$

$NotThreadID \triangleq -5050$

$EMPTY\_BLOCK \triangleq -1$

$BlockID \triangleq 0 \,..\, 4$ $NBLOCKS - 1$

$NotBlockID \triangleq -1000$

$StmtID \triangleq 0 \,..\, 0$ $MXSTMTS - 1$

$I\_START \triangleq -1$

$I\_PRE\_FIN\_ALLOC \triangleq -2$

Record Types

$Sequences \triangleq [aseq : 1 \,..\, MXACTIVITIES,\ mseq : 1 \,..\, MXMESSAGES,\ fseq : IDRange]$

Each thread has a stack, and this is the stack entry

$StackEntry \triangleq [b : BlockID,$
$\quad i : StmtID \cup \{I\_START,\ I\_PRE\_FIN\_ALLOC\},$
$\quad fid : PIDRange]$

the processing unit of program instructions

$$\begin{aligned}
Thread \triangleq\ & [tid : ThreadID,\\
& status : \{\text{“idle”}, \text{“running”}, \text{“blocked”}\},\\
& blockingType : \{\text{“NA”}, \text{“FinishEnd”}, \text{“AsyncTransit”}, \text{“FinishAlloc”}, \text{“AsyncTerm”}\},\\
& stack : Seq(StackEntry)]
\end{aligned}$$

the activities that are pushed to scheduler's ready queue,
and will eventually be fetched by threads

$$\begin{aligned}
Activity \triangleq\ & [aid : Nat,\\
& b : BlockID,\\
& fid : IDRange]
\end{aligned}$$

$$NotActivity \triangleq [aid \mapsto -1,\ b \mapsto NotBlockID,\ fid \mapsto NotID]$$

Input Program: *Block* error used to simulate exceptions

$$\begin{aligned}
Block \triangleq\ & [b : BlockID \cup \{NotBlockID\},\\
& type : \{\text{“NA”}, \text{“async”}, \text{“expr”}, \text{“finish”}, \text{“error”}, \text{“kill”}\},\\
& dst : PLACE \cup \{NotPlace\},\\
& mxstmt : Nat,\\
& stmts : [StmtID \to BlockID \cup \{EMPTY_BLOCK,\ NotBlockID\}],\\
& ran : \text{BOOLEAN}\ ]
\end{aligned}$$

$$PlaceThread \triangleq [here : PLACE,\ tid : ThreadID]$$

$$NotPlaceThread \triangleq [\ here \mapsto NotPlace,\ \ tid \mapsto NotThreadID]$$

$$\begin{aligned}
MasterStatus \triangleq\ & [status : \{\text{“running”}, \text{“seekAdoption”}, \text{“convertDead”}\},\\
& lastKilled : PLACE \cup \{NotPlace\}]
\end{aligned}$$

Finish Types

$$\begin{aligned}
FinishState \triangleq\ & [id : IDRange \cup \{NotID\},\\
& status : \{\text{“unused”}, \text{“waiting”}, \text{“pendingRelease”}, \text{“forgotten”}\},\\
& type : \{\text{“distroot”}, \text{“distremote”}, \text{“NA”}\},\\
& count : Nat,\\
& here : PLACE \cup \{NotPlace\},\\
& parent : PIDRange \cup \{NotID\},\ \text{used only in RESILIENT mode}\\
& root : PIDRange \cup \{NotID\},\ \text{root is the same as } id \text{ for root finishes}\\
& isGlobal : \text{BOOLEAN}\ ,\ \text{used by } P0Finish\\
& eroot : PIDRange \cup \{NotID\}\ \text{root of the enclosing finish (used in } PPoPP14 \text{ dist finish)}\\
&]
\end{aligned}$$

$$\begin{aligned}
MasterFinish \triangleq\ & [\\
& id : IDRange \cup \{NotID\},\\
& numActive : Nat,\\
& live : [PLACE \to Nat],\\
& transit : [PLACE \to [PLACE \to Nat]],
\end{aligned}$$

$$
\begin{array}{l}
\quad liveAdopted : [PLACE \to Nat], \\
\quad transitAdopted : [PLACE \to [PLACE \to Nat]], \\
\quad children : \text{SUBSET } IDRange, \\
\quad backupPlace : PLACE \cup \{NotPlace\}, \\
\quad isReleased : \text{BOOLEAN} \\
]
\end{array}
$$

$$
\begin{array}{l}
BackupFinish \triangleq [\\
\quad id : IDRange \cup \{NotID\}, \\
\quad live : [PLACE \to Nat], \\
\quad transit : [PLACE \to [PLACE \to Nat]], \\
\quad children : \text{SUBSET } IDRange, \\
\quad isAdopted : \text{BOOLEAN }, \\
\quad adoptedRoot : IDRange \cup \{NotID\}, \\
\quad numActive : Nat, \\
\quad isReleased : \text{BOOLEAN} \\
]
\end{array}
$$

Message Types and Utilities

$$NotMessage \triangleq [fid \mapsto NotID,\ src \mapsto NotPlace]$$

$$
\begin{array}{l}
RemoteAsyncMessages \triangleq [mid : Nat, \\
\qquad\qquad src : PLACE, \\
\qquad\qquad dst : PLACE, \\
\qquad\qquad type : \{\text{“async”}\}, \\
\qquad\qquad b : BlockID, \\
\qquad\qquad fid : IDRange]
\end{array}
$$

$$
\begin{array}{l}
ReleaseFinishMessages \triangleq [mid : Nat, \\
\qquad\qquad\qquad\qquad src : PLACE, \\
\qquad\qquad\qquad\qquad dst : PLACE, \\
\qquad\qquad\qquad\qquad fid : IDRange, \\
\qquad\qquad\qquad\qquad type : \{\text{“releaseFinish”}\}]
\end{array}
$$

$$
\begin{array}{l}
MasterTransitMessages \triangleq [mid \mapsto Nat, \\
\qquad\qquad\qquad\qquad\quad src \mapsto PLACE, \\
\qquad\qquad\qquad\qquad\quad dst \mapsto PLACE, \\
\qquad\qquad\qquad\quad target \mapsto PLACE, \\
\qquad\qquad\qquad\qquad fid \mapsto IDRange, \\
\qquad\qquad\qquad\qquad type \mapsto \{\text{“masterTransit”},\ \text{“adopterTransit”}\}]
\end{array}
$$

$$
\begin{array}{l}
MasterLiveMessages \triangleq [mid \mapsto Nat, \\
\qquad\qquad\qquad\qquad\quad src \mapsto PLACE, \\
\qquad\qquad\qquad\quad source \mapsto PLACE, \\
\qquad\qquad\qquad\quad target \mapsto PLACE,
\end{array}
$$

$$
\begin{array}{rl}
dst & \mapsto PLACE, \\
fid & \mapsto IDRange, \\
aid & \mapsto Nat, \\
type & \mapsto \{\text{“masterLive”}, \text{“adopterLive”}\}]
\end{array}
$$

$$
\begin{array}{rcl}
MasterCompletedMessages & \triangleq & [mid \mapsto Nat, \\
& & \quad src \mapsto PLACE, \\
& & \quad dst \mapsto PLACE, \\
& & \quad target \mapsto PLACE, \\
& & \quad fid \mapsto IDRange, \\
& & \quad finishEnd \mapsto \text{BOOLEAN}, \\
& & \quad type \mapsto \{\text{“masterCompleted”}, \text{“adopterCompleted”}\}]
\end{array}
$$

$$
\begin{array}{rcl}
BackupGetAdopter & \triangleq & [\ mid \mapsto Nat, \\
& & \quad src \mapsto PLACE, \\
& & \quad dst \mapsto PLACE, \\
& & \quad fid \mapsto IDRange, \\
& & \quad actionType \mapsto \{\text{“transit”}, \text{“completed”}, \text{“live”}\}, \\
& & \quad aid \mapsto Nat, \\
& & \quad finishEnd \mapsto \text{BOOLEAN}, \\
& & \quad type \mapsto \text{“backupGetAdopter”}]
\end{array}
$$

$$
\begin{array}{rcl}
GetAdopterDone & \triangleq & [\ mid \mapsto Nat, \\
& & \quad src \mapsto PLACE, \\
& & \quad dst \mapsto PLACE, \\
& & \quad source \mapsto PLACE, \\
& & \quad target \mapsto PLACE, \\
& & \quad fid \mapsto IDRange, \\
& & \quad adoptedRoot \mapsto IDRange, \\
& & \quad actionType \mapsto \{\text{“transit”}, \text{“completed”}, \text{“live”}\}, \\
& & \quad aid \mapsto Nat, \\
& & \quad finishEnd \mapsto \text{BOOLEAN}, \\
& & \quad type \mapsto \text{“backupGetAdopterDone”}]
\end{array}
$$

$$
\begin{array}{rcl}
MasterTransitDone & \triangleq & [\ mid \mapsto Nat, \\
& & \quad src \mapsto PLACE, \\
& & \quad dst \mapsto PLACE, \\
& & \quad target \mapsto PLACE, \\
& & \quad fid \mapsto IDRange, \\
& & \quad type \mapsto \text{“masterTransitDone”}, \\
& & \quad isAdopter \mapsto \text{BOOLEAN}, \\
& & \quad submit \mapsto \text{BOOLEAN}, \\
& & \quad success \mapsto \text{BOOLEAN}, \\
& & \quad backupPlace \mapsto PLACE]
\end{array}
$$

$$
\begin{array}{rcl}
BackupTransit & \triangleq & [\ mid \mapsto Nat,
\end{array}
$$

$$
\begin{aligned}
src &\mapsto PLACE,\\
dst &\mapsto PLACE,\\
target &\mapsto PLACE,\\
fid &\mapsto IDRange,\\
isAdopter &\mapsto \text{BOOLEAN},\\
adoptedFID &\mapsto IDRange \cup \{NotID\},\\
type &\mapsto \text{“backupTransit”}]
\end{aligned}
$$

$$
\begin{aligned}
BackupTransitDone \triangleq [\ & mid \mapsto Nat,\\
& src \mapsto PLACE,\\
& dst \mapsto PLACE,\\
& target \mapsto PLACE,\\
& fid \mapsto IDRange,\\
& type \mapsto \text{“backupTransitDone”},\\
& success \mapsto \text{BOOLEAN},\\
& isAdopter \mapsto \text{BOOLEAN},\\
& adoptedFID \mapsto IDRange \cup \{NotID\}]
\end{aligned}
$$

$$
\begin{aligned}
BackupLive \triangleq [\ & mid \mapsto Nat,\\
& src \mapsto PLACE,\\
& dst \mapsto PLACE,\\
& source \mapsto PLACE,\\
& target \mapsto PLACE,\\
& fid \mapsto IDRange,\\
& aid \mapsto Nat,\\
& type \mapsto \text{“backupLive”},\\
& isAdopter \mapsto \text{BOOLEAN},\\
& adoptedFID \mapsto IDRange \cup \{NotID\}]
\end{aligned}
$$

$$
\begin{aligned}
BackupLiveDone \triangleq [\ & mid \mapsto Nat,\\
& src \mapsto PLACE,\\
& dst \mapsto PLACE,\\
& target \mapsto PLACE,\\
& source \mapsto PLACE,\\
& fid \mapsto IDRange,\\
& aid \mapsto Nat,\\
& type \mapsto \text{“backupLiveDone”},\\
& success \mapsto \text{BOOLEAN},\\
& isAdopter \mapsto \text{BOOLEAN},\\
& adoptedFID \mapsto IDRange \cup \{NotID\}]
\end{aligned}
$$

$$
\begin{aligned}
MasterLiveDone \triangleq [\ & mid \mapsto Nat,\\
& src \mapsto PLACE,\\
& dst \mapsto PLACE,\\
& target \mapsto PLACE,\\
& source \mapsto PLACE,
\end{aligned}
$$

$$
\begin{aligned}
\mathit{fid} &\mapsto \mathit{IDRange},\\
\mathit{aid} &\mapsto \mathit{Nat},\\
\mathit{type} &\mapsto \text{“masterLiveDone”},\\
\mathit{submit} &\mapsto \text{BOOLEAN},\\
\mathit{success} &\mapsto \text{BOOLEAN},\\
\mathit{isAdopter} &\mapsto \text{BOOLEAN},\\
\mathit{backupPlace} &\mapsto \mathit{PLACE}]
\end{aligned}
$$

$$
\begin{aligned}
\mathit{BackupCompleted} \triangleq\ & [\mathit{mid} \mapsto \mathit{Nat},\\
& \mathit{src} \mapsto \mathit{PLACE},\\
& \mathit{dst} \mapsto \mathit{PLACE},\\
& \mathit{target} \mapsto \mathit{PLACE},\\
& \mathit{fid} \mapsto \mathit{IDRange},\\
& \mathit{type} \mapsto \text{“backupCompleted”},\\
& \mathit{isAdopter} \mapsto \text{BOOLEAN},\\
& \mathit{finishEnd} \mapsto \text{BOOLEAN}\ ]
\end{aligned}
$$

$$
\begin{aligned}
\mathit{MasterCompletedDone} \triangleq\ & [\mathit{mid} \mapsto \mathit{Nat},\\
& \mathit{src} \mapsto \mathit{PLACE},\\
& \mathit{dst} \mapsto \mathit{PLACE},\\
& \mathit{target} \mapsto \mathit{PLACE},\\
& \mathit{fid} \mapsto \mathit{IDRange},\\
& \mathit{type} \mapsto \text{“masterCompletedDone”},\\
& \mathit{success} \mapsto \text{BOOLEAN},\\
& \mathit{isAdopter} \mapsto \text{BOOLEAN},\\
& \mathit{backupPlace} \mapsto \mathit{PLACE}]
\end{aligned}
$$

$$
\begin{aligned}
\mathit{BackupCompletedDone} \triangleq\ & [\mathit{mid} \mapsto \mathit{Nat},\\
& \mathit{src} \mapsto \mathit{PLACE},\\
& \mathit{dst} \mapsto \mathit{PLACE},\\
& \mathit{target} \mapsto \mathit{PLACE},\\
& \mathit{fid} \mapsto \mathit{IDRange},\\
& \mathit{type} \mapsto \text{“backupCompletedDone”},\\
& \mathit{isAdopter} \mapsto \text{BOOLEAN},\\
& \mathit{success} \mapsto \text{BOOLEAN}\ ]
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Messages} \triangleq\ & \mathit{RemoteAsyncMessages}\\
& \cup \mathit{MasterTransitMessages}\\
& \cup \mathit{MasterLiveMessages}\\
& \cup \mathit{MasterCompletedMessages}\\
& \cup \mathit{BackupGetAdopter}\\
& \cup \mathit{GetAdopterDone}\\
& \cup \mathit{BackupTransit}\\
& \cup \mathit{MasterTransitDone}\\
& \cup \mathit{BackupTransitDone}\\
& \cup \mathit{BackupLive}
\end{aligned}
$$

$\cup$ *MasterLiveDone*
$\cup$ *BackupLiveDone*
$\cup$ *BackupCompleted*
$\cup$ *MasterCompletedDone*
$\cup$ *BackupCompletedDone*
$\cup$ *ReleaseFinishMessages*

SendMsg(m) $\triangleq$
$msgs' = msgs \cup \{m\}$

RecvMsg(m) $\triangleq$
$msgs' = msgs \setminus \{m\}$

ReplaceMsg(*toRemove*, *toAdd*) $\triangleq$
$msgs' = (msgs \setminus \{toRemove\}) \cup \{toAdd\}$

ReplaceMsgSet(*toRemove*, *toAddSet*) $\triangleq$
$msgs' = (msgs \setminus \{toRemove\}) \cup toAddSet$

Predicates to extract the finish *id* from messages and *fstates*

ExtractFIDFromMSG(*src*, *dst*, *type*) $\triangleq$
LET $mset \triangleq \{m \in msgs : \land m.src = src$
$\land m.dst = dst$
$\land m.type = type$
$\land m.fid \in IDRange$
$\}$
IN IF $mset = \{\}$ THEN *NotID*
ELSE (CHOOSE $x \in mset$: TRUE).*fid*

FindIncomingMSG(*here*, *type*) $\triangleq$
LET $mset \triangleq \{m \in msgs : \land m.dst = here$
$\land m.type = type$
$\land m.dst \notin killed$
$\}$
IN IF $mset = \{\}$ THEN *NotMessage*
ELSE CHOOSE $x \in mset$: TRUE

FindMSG(*type*) $\triangleq$
LET $mset \triangleq \{m \in msgs : \land m.type = type$
$\land m.dst \notin killed$
$\}$
IN IF $mset = \{\}$ THEN *NotMessage*
ELSE CHOOSE $x \in mset$: TRUE

GetActiveFID(*type*, *here*, *pid*) $\triangleq$
LET $mset \triangleq \{id \in IDRange : \land fstates[id].here = here$

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\, fstates[id].root = pid \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\, fstates[id].type = type \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\, fstates[id].status = \text{“waiting”} \\
\qquad\qquad\quad \} \\
\quad \text{IN} \quad \text{IF } mset = \{\} \text{ THEN } NotID \\
\qquad\quad \text{ELSE } (\text{CHOOSE } x \in mset : \text{TRUE})
\end{array}
$$

$$
\begin{array}{l}
GetFinishHome(fid) \triangleq \\
\quad \text{IF } fid = NoParent \text{ THEN } PROG\_HOME \text{ ELSE } fstates[fid].here
\end{array}
$$

$$
\begin{array}{l}
GetEnclosingRoot(parent,\, me) \triangleq \\
\quad \text{IF } parent = NoParent \text{ THEN } NoParent \text{ ELSE } fstates[parent].root
\end{array}
$$

Predicate to extract thread ids with a specific status

$$
\begin{array}{l}
FindThread(here,\, status) \triangleq \\
\quad \text{LET } tset \triangleq \{t \in ThreadID : thrds[here][t].status = status\} \\
\quad \text{IN} \quad \text{IF } tset = \{\} \text{ THEN } NotThreadID \\
\qquad\quad \text{ELSE CHOOSE } x \in tset : \text{TRUE}
\end{array}
$$

$$
\begin{array}{l}
FindThread2(here,\, statusSet) \triangleq \\
\quad \text{LET } tset \triangleq \{t \in ThreadID : thrds[here][t].status \in statusSet\} \\
\quad \text{IN} \quad \text{IF } tset = \{\} \text{ THEN } NotThreadID \\
\qquad\quad \text{ELSE CHOOSE } x \in tset : \text{TRUE}
\end{array}
$$

Resilient Store Types and Utilities

$$Adopter \triangleq [here : PLACE,\, child : IDRange \cup \{NotID\},\, adopter : IDRange \cup \{NotID\}]$$

$$NotAdopter \triangleq [here \mapsto NotPlace,\, child \mapsto NotID,\, adopter \mapsto NotID]$$

$$ConvTask \triangleq [here : PLACE,\, fid : IDRange \cup \{NotID\},\, pl : PLACE \cup \{NotPlace\}]$$

$$NotConvTask \triangleq [here \mapsto NotPlace,\, fid \mapsto NotID,\, pl \mapsto NotPlace]$$

$$GetBackup(p) \triangleq BACKUP[p]$$

Utilities to increment sequences used to give unique ids to finish (*fseq*) messages (*mseq*), and activities (*aseq*)

$$
\begin{array}{l}
IncrFSEQ \triangleq \\
\quad seq' = [aseq \mapsto seq.aseq,\, fseq \mapsto seq.fseq + 1,\, mseq \mapsto seq.mseq]
\end{array}
$$

$$
\begin{array}{l}
IncrMSEQ(c) \triangleq \\
\quad seq' = [aseq \mapsto seq.aseq,\, fseq \mapsto seq.fseq,\, mseq \mapsto seq.mseq + c]
\end{array}
$$

$$
\begin{array}{l}
IncrASEQ \triangleq \\
\quad seq' = [aseq \mapsto seq.aseq + 1,\, fseq \mapsto seq.fseq,\, mseq \mapsto seq.mseq]
\end{array}
$$

$$IncrAll \triangleq$$
$$seq' = [aseq \mapsto seq.aseq + 1,\ fseq \mapsto seq.fseq + 1,\ mseq \mapsto seq.mseq + 1]$$

* Modification History
* Last modified *Wed Dec* 13 15:52:59 *AEDT* 2017 by *u*5482878
* Last modified Sun *Dec* 10 16:09:57 *AEDT* 2017 by *shamouda*
* Created *Wed Sep* 27 09:26:18 *AEST* 2017 by *u*5482878