MODULE *storagecleanerimproved*

EXTENDS *Naturals*, *Sequences*, *FiniteSets*

CONSTANTS
USERIDS,
SERVERS,
METADATAS,
IMAGES,
UUIDS,
CLEANERS

VARIABLES
Implementation variables
databaseState,
blobStoreState,
serverStates,
cleanerStates,

We just added a time variable here
time, Natural number representing the number of hours that have passed

Observability variables
operations

$vars \triangleq \langle databaseState, blobStoreState, serverStates, operations, cleanerStates, time \rangle$

$cleanerVars \triangleq \langle cleanerStates \rangle$

Strong Typing

$UserIdVal \triangleq USERIDS \cup \{\text{“UNSET”}\}$
$MetadataVal \triangleq METADATAS \cup \{\text{“UNSET”}\}$
$ImageVal \triangleq IMAGES \cup \{\text{“UNSET”}\}$
$UUIDVal \triangleq UUIDS \cup \{\text{“UNSET”}\}$

$DatabaseRecord \triangleq [$
$\quad metadata : MetadataVal,$
$\quad imageId : UUIDVal$
$]$

A blob store record is modeled to store creation time
$BlobStoreRecord \triangleq [$
$\quad image : ImageVal,$
$\quad created : Nat$
$] \cup \{[$
$\quad status \mapsto \text{“UNSET”},$

$image \mapsto$ "UNSET"
]} It can still be unset

$ServerStateVal \triangleq$
[
$state$: {
"waiting",
"started_write",
"wrote_blob",
"started_read",
"read_metadata"
},
$userId$: $UserIdVal$,
$metadata$: $MetadataVal$,
$imageId$: $UUIDVal$,
$image$: $ImageVal$
]

$CleanerStateVal \triangleq$
[
$state$: {
"waiting",
"got_blob_keys",
"got_unused_keys",
"deleting_keys"
},
$blobKeys$: SUBSET $UUIDS$,
$unusedBlobKeys$: SUBSET $UUIDS$
]

$OperationValue \triangleq$ [$type$: {"READ", "WRITE"},
$userId$: $UserIdVal$,
$metadata$: $MetadataVal$,
$image$: $ImageVal$]

$TypeOk \triangleq$
$\wedge$ $databaseState \in [USERIDS \to DatabaseRecord]$
Blob store is updated to store records. Can be a record or unset
$\wedge$ $blobStoreState \in [UUIDS \to BlobStoreRecord]$
$\wedge$ $serverStates \in [SERVERS \to ServerStateVal]$
$\wedge$ $cleanerStates \in [CLEANERS \to CleanerStateVal]$
$\wedge$ $operations \in Seq(OperationValue)$
$\wedge$ $time \in Nat$ Time is represented as a natural number

$Init \triangleq$
$\land\ databaseState =$
$[u \in USERIDS \mapsto [metadata \mapsto$ "UNSET", $imageId \mapsto$ "UNSET"$]]$
$\land\ blobStoreState =$
$[u \in UUIDS \mapsto\ [status \mapsto$ "UNSET", $image \mapsto$ "UNSET"$]]$
$\land\ serverStates = [s \in SERVERS \mapsto\ [state \mapsto$ "waiting",
$userId \mapsto$ "UNSET",
$metadata \mapsto$ "UNSET",
$imageId \mapsto$ "UNSET",
$image \mapsto$ "UNSET"
$]]$
$\land\ cleanerStates = [c \in CLEANERS \mapsto [$
$state \mapsto$ "waiting",
$blobKeys \mapsto \{\}$,
$unusedBlobKeys \mapsto \{\}$
$]]$
$\land\ operations = \langle\rangle$
$\land\ time = 0$ Time starts at 0

State Machine

$TimePasses \triangleq$
$\land\ time' = time + 1$
$\land$ UNCHANGED $\langle serverStates,\ databaseState,\ blobStoreState,\ operations,$
$cleanerStates\rangle$

Server Writes

$ServerStartWrite(s) \triangleq$
$\land\ serverStates[s].state =$ "waiting"
$\land \exists\, u \in USERIDS,\ m \in METADATAS,\ i \in IMAGES :$
$\land\ serverStates' = [serverStates$ EXCEPT
$![s].state =$ "started_write",
$![s].userId = u,$
$![s].metadata = m,$
$![s].image = i]$
$\land\ operations' = Append(operations,$
$[$
$type \mapsto$ "WRITE",
$userId \mapsto u,$
$metadata \mapsto m,$
$image \mapsto i$
$])$
$\land$ UNCHANGED $\langle databaseState,\ blobStoreState,\ cleanerStates\rangle$
$\land$ UNCHANGED $time$

$ServerWriteBlob(s) \triangleq$
LET $currentState \triangleq serverStates[s]$
IN
$\land\ currentState.state = \text{“started\_write”}$
$\land \exists\, id \in UUIDS :$
$\land\ blobStoreState[id] = [status \mapsto \text{“UNSET”},\ image \mapsto \text{“UNSET”}]$
$\land\ blobStoreState' = [blobStoreState \text{ EXCEPT}$
$![id] = [$
$image \mapsto currentState.image,$
$created \mapsto time$
$]]$
$\land\ serverStates' = [serverStates \text{ EXCEPT}$
$![s].state = \text{“wrote\_blob”},$
$![s].imageId = id]$
$\land \text{ UNCHANGED } \langle databaseState,\ operations\rangle$
$\land \text{ UNCHANGED } cleanerVars$
$\land \text{ UNCHANGED } time$

$ServerWriteMetadataAndReturn(s) \triangleq$
LET $currentState \triangleq serverStates[s]$
IN
$\land\ currentState.state = \text{“wrote\_blob”}$
$\land\ databaseState' = [databaseState \text{ EXCEPT}$
$![currentState.userId] = [$
$metadata \mapsto currentState.metadata,$
$imageId \mapsto currentState.imageId]]$

$\land\ serverStates' = [serverStates \text{ EXCEPT}$
$![s].state = \text{“waiting”},$
$![s].userId = \text{“UNSET”},$
$![s].metadata = \text{“UNSET”},$
$![s].image = \text{“UNSET”},$
$![s].imageId = \text{“UNSET”}]$
$\land \text{ UNCHANGED } \langle blobStoreState,\ operations\rangle$
$\land \text{ UNCHANGED } cleanerVars$
$\land \text{ UNCHANGED } time$

$ServerFailWrite(s) \triangleq$
$\land\ serverStates[s].state \in \{\text{“started\_write”},\ \text{“wrote\_blob”}\}$
$\land\ serverStates' = [serverStates \text{ EXCEPT}$
$![s].state = \text{“waiting”},$
$![s].userId = \text{“UNSET”},$
$![s].metadata = \text{“UNSET”},$
$![s].image = \text{“UNSET”},$
$![s].imageId = \text{“UNSET”}]$
$\land \text{ UNCHANGED } \langle databaseState,\ blobStoreState,\ operations\rangle$

$\land$ UNCHANGED $cleanerVars$
$\land$ UNCHANGED $time$

Server Reads

$ServerStartRead(s) \triangleq$
$\land$ $serverStates[s].state =$ "waiting"
$\land \exists\, u \in USERIDS :$
$serverStates' = [serverStates$ EXCEPT
$![s].state =$ "started_read",
$![s].userId = u]$

$\land$ UNCHANGED $\langle databaseState, blobStoreState\rangle$
$\land$ UNCHANGED $operations$
$\land$ UNCHANGED $cleanerVars$
$\land$ UNCHANGED $time$

$ServerReadMetadata(s) \triangleq$
LET $currentState \triangleq serverStates[s]$
IN
$\land$ $currentState.state =$ "started_read"
$\land$ $databaseState[currentState.userId].metadata \neq$ "UNSET"
$\land$ $serverStates' =$
$[serverStates$ EXCEPT
$![s].state =$ "read_metadata",
$![s].metadata = databaseState[currentState.userId].metadata,$
$![s].imageId = databaseState[currentState.userId].imageId]$
$\land$ UNCHANGED $\langle databaseState, blobStoreState\rangle$
$\land$ UNCHANGED $operations$
$\land$ UNCHANGED $cleanerVars$
$\land$ UNCHANGED $time$

$ServerReadMetadataAndReturnEmpty(s) \triangleq$
LET $currentState \triangleq serverStates[s]$
IN
$\land$ $currentState.state =$ "started_read"
$\land$ $databaseState[currentState.userId].metadata =$ "UNSET"
$\land$ $serverStates' = [serverStates$ EXCEPT
$![s].state =$ "waiting",
$![s].userId =$ "UNSET",
$![s].metadata =$ "UNSET",
$![s].image =$ "UNSET",
$![s].imageId =$ "UNSET"]

$\land$ $operations' = Append(operations,$
Returns an empty record

```
                                    [
                                        type ↦ "READ",
                                        userId ↦ currentState.userId,
                                        metadata ↦ "UNSET",
                                        image ↦ "UNSET"
                                    ])
    ∧ UNCHANGED ⟨databaseState, blobStoreState⟩
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time

ServerReadBlobAndReturn(s) ≜
    LET currentState ≜ serverStates[s]
    IN
    ∧ currentState.state = "read_metadata"
    ∧ operations' = Append(operations,
                                    [
                                        type ↦ "READ",
                                        userId ↦ currentState.userId,
                                        metadata ↦ currentState.metadata,
                                         Looks up image by imageId
                                        image ↦ blobStoreState[currentState.imageId].image
                                    ])
    ∧ serverStates' = [serverStates EXCEPT
                                ![s].state = "waiting",
                                ![s].userId = "UNSET",
                                ![s].metadata = "UNSET",
                                ![s].image = "UNSET",
                                ![s].imageId = "UNSET"]
    ∧ UNCHANGED ⟨databaseState, blobStoreState⟩
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time
```

Cleaner States

This is the main change in the logic.

```
CleanerStartGetBlobKeys(c) ≜
    LET current ≜ cleanerStates[c]IN
    ∧ current.state = "waiting"
    ∧ cleanerStates' = [
        cleanerStates EXCEPT
            ![c].state = "got_blob_keys",
             All keys in blockstore
            ![c].blobKeys = {
                k ∈ UUIDS :
                    LET earliestDeletionTime ≜ blobStoreState[k].created + 2IN
```

```
                    That are not unset
                    ∧ blobStoreState[k] ≠ [
                        status ↦ "UNSET",
                        image ↦ "UNSET"]
                    It must have been created 2 or more hours ago
                    ∧ earliestDeletionTime ≤ time
            }
        ]
    ∧ UNCHANGED ⟨serverStates, databaseState, blobStoreState, operations⟩
    ∧ UNCHANGED time

CleanerGetUnusedKeys(c) ≜
    LET current ≜ cleanerStates[c] IN
    ∧ current.state = "got_blob_keys"
    ∧ cleanerStates' = [
        cleanerStates EXCEPT
            ![c].state = "got_unused_keys",
            ![c].unusedBlobKeys =
                {k ∈ current.blobKeys :
                    ∀ u ∈ USERIDS :
                        databaseState[u].imageId ≠ k}
        ]
    ∧ UNCHANGED ⟨serverStates, databaseState, blobStoreState, operations⟩
    ∧ UNCHANGED time

CleanerDeletingKeys(c) ≜
    LET current ≜ cleanerStates[c] IN
    ∧ current.state ∈ {"got_unused_keys", "deleting_keys"}
    ∧ Cardinality(current.unusedBlobKeys) ≠ 0
    ∧ ∃ k ∈ current.unusedBlobKeys :
        ∧ blobStoreState' =
            [blobStoreState EXCEPT
                ![k] = [status ↦ "UNSET", image ↦ "UNSET"]]
        ∧ cleanerStates' = [
            cleanerStates EXCEPT
                ![c].unusedBlobKeys = current.unusedBlobKeys \ {k}
            ]
    ∧ UNCHANGED ⟨serverStates, databaseState, operations⟩
    ∧ UNCHANGED time

CleanerFinished(c) ≜
    LET current ≜ cleanerStates[c] IN
    ∧ current.state = "deleting_keys"
    ∧ Cardinality(current.unusedBlobKeys) = 0
    ∧ cleanerStates' = [
        cleanerStates EXCEPT
```

```
            ![c].state = "waiting",
            ![c].blobKeys = {},
            ![c].unusedBlobKeys = {}
        ]
    ∧ UNCHANGED ⟨serverStates, databaseState, blobStoreState, operations⟩
    ∧ UNCHANGED time

CleanerFail(c) ≜
    LET current ≜ cleanerStates[c] IN
    ∧ current.state ∈ {"got_blob_keys", "got_unused_keys", "deleting_keys"}
    ∧ cleanerStates' = [
        cleanerStates EXCEPT
            ![c].state = "waiting",
            ![c].blobKeys = {},
            ![c].unusedBlobKeys = {}
        ]
    ∧ UNCHANGED ⟨serverStates, databaseState, blobStoreState, operations⟩
    ∧ UNCHANGED time
```

Specification / *Next*

```
Next ≜
    Time can pass now
    ∨ TimePasses
    ∨ ∃ s ∈ SERVERS :
        ∨ ServerStartWrite(s)
        ∨ ServerWriteBlob(s)
        ∨ ServerWriteMetadataAndReturn(s)
        ∨ ServerFailWrite(s)
        ∨ ServerStartRead(s)
        ∨ ServerReadMetadata(s)
        ∨ ServerReadMetadataAndReturnEmpty(s)
        ∨ ServerReadBlobAndReturn(s)
    ∨ ∃ c ∈ CLEANERS :
        ∨ CleanerStartGetBlobKeys(c)
        ∨ CleanerGetUnusedKeys(c)
        ∨ CleanerDeletingKeys(c)
        ∨ CleanerFinished(c)
        ∨ CleanerFail(c)


Spec ≜ Init ∧ □[Next]_vars
```

Invariants

Note that the success criteria hasn't changed this whole time

$$
\begin{array}{l}
ConsistentReads \triangleq \\
\quad \lor operations = \langle\rangle \\
\quad \lor \forall\, i \in 1 \,..\, Len(operations) : \\
\qquad \text{LET } readOp \triangleq operations[i] \text{ IN} \\
\qquad \lor \quad \land readOp.type = \text{“READ”} \\
\qquad\qquad \land \quad \lor \exists\, j \in 1 \,..\, i : \\
\qquad\qquad\qquad\qquad \text{LET } writeOp \triangleq operations[j] \text{ IN} \\
\qquad\qquad\qquad\qquad \land writeOp.type = \text{“WRITE”} \\
\qquad\qquad\qquad\qquad \land readOp.userId = writeOp.userId \\
\qquad\qquad\qquad\qquad \land readOp.metadata = writeOp.metadata \\
\qquad\qquad\qquad\qquad \land readOp.image = writeOp.image \\
\qquad\qquad\qquad \lor \\
\qquad\qquad\qquad\qquad \land readOp.metadata = \text{“UNSET”} \\
\qquad\qquad\qquad\qquad \land readOp.image = \text{“UNSET”} \\
\qquad \lor readOp.type = \text{“WRITE”}
\end{array}
$$

$$
\begin{array}{l}
NoOrphanFiles \triangleq \\
\quad \neg\exists\, k \in UUIDS : \\
\qquad \land blobStoreState[k] \neq [status \mapsto \text{“UNSET”}, image \mapsto \text{“UNSET”}] \\
\qquad \land \forall\, u \in USERIDS : \\
\qquad\quad databaseState[u].imageId \neq k
\end{array}
$$

Properties

$$EventuallyNoOrphanFiles \triangleq \Diamond NoOrphanFiles$$

$$AlwaysEventuallyNoOrphanFiles \triangleq \Box EventuallyNoOrphanFiles$$

$$
\begin{array}{l}
StopAfter3Operations \triangleq \\
\quad \land Len(operations) \leq 3 \\
\quad \land time \leq 2
\end{array}
$$

$$
\begin{array}{l}
StopAfter5Operations \triangleq \\
\quad Len(operations) \leq 5
\end{array}
$$