Server Restart

$$
\begin{array}{l}
\mathit{ServerRestart}(s) \triangleq \\
\quad \text{LET } \mathit{currentState} \triangleq \mathit{serverStates}[s] \text{ IN} \\
\quad \text{LET } \mathit{terminationTime} \triangleq (\mathit{currentState}.\mathit{start} + 1) \text{ IN} \\
\quad \land\ \mathit{currentState}.\mathit{state} \neq \text{“waiting”} \quad \text{Server must be active} \\
\quad \text{This is the only state a server can reach if past termination time} \\
\quad \land\ \mathit{time} \Rightarrow \mathit{terminationTime} \\
\quad \land\ \mathit{serverStates}' = [\mathit{serverStates} \text{ EXCEPT} \\
\qquad\qquad\qquad\qquad ![s].\mathit{state} = \text{“waiting”}, \\
\qquad\qquad\qquad\qquad ![s].\mathit{userId} = \text{“UNSET”}, \\
\qquad\qquad\qquad\qquad ![s].\mathit{metadata} = \text{“UNSET”}, \\
\qquad\qquad\qquad\qquad ![s].\mathit{image} = \text{“UNSET”}, \\
\qquad\qquad\qquad\qquad ![s].\mathit{imageId} = \text{“UNSET”}] \\
\quad \land\ \text{UNCHANGED } \langle \mathit{databaseState}, \mathit{blobStoreState}, \mathit{operations} \rangle \\
\quad \land\ \text{UNCHANGED } \mathit{cleanerVars} \\
\quad \land\ \text{UNCHANGED } \mathit{time}
\end{array}
$$

Server Writes

$$
\begin{array}{l}
\mathit{ServerStartWrite}(s) \triangleq \\
\quad \land\ \mathit{serverStates}[s].\mathit{state} = \text{“waiting”} \\
\quad \land\ \exists\, u \in \mathit{USERIDS},\ m \in \mathit{METADATAS},\ i \in \mathit{IMAGES} : \\
\qquad \land\ \mathit{serverStates}' = [\mathit{serverStates} \text{ EXCEPT} \\
\qquad\qquad\qquad\qquad ![s].\mathit{state} = \text{“started\_write”}, \\
\qquad\qquad\qquad\qquad ![s].\mathit{userId} = u, \\
\qquad\qquad\qquad\qquad ![s].\mathit{metadata} = m, \\
\qquad\qquad\qquad\qquad ![s].\mathit{image} = i, \\
\qquad\qquad\qquad\qquad \text{The time a write request starts} \\
\qquad\qquad\qquad\qquad ![s].\mathit{start} = \mathit{time} \\
\qquad\qquad\qquad\qquad] \\
\qquad \land\ \mathit{operations}' = \mathit{Append}(\mathit{operations}, \\
\qquad\qquad\qquad\qquad\qquad\qquad [\\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{type} \mapsto \text{“WRITE”}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{userId} \mapsto u, \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{metadata} \mapsto m, \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{image} \mapsto i \\
\qquad\qquad\qquad\qquad\qquad\qquad]) \\
\quad \text{Cleaner state needs to be added as unchanged for all server operations} \\
\quad \land\ \text{UNCHANGED } \langle \mathit{databaseState}, \mathit{blobStoreState}, \mathit{cleanerStates} \rangle \\
\quad \land\ \text{UNCHANGED } \mathit{time}
\end{array}
$$

$$
\begin{array}{l}
\mathit{ServerWriteBlob}(s) \triangleq \\
\quad \text{LET } \mathit{currentState} \triangleq \mathit{serverStates}[s] \text{ IN}
\end{array}
$$

```
    LET terminationTime ≜ (currentState.start + 1)IN
    ∧ time < terminationTime  Can only start this state if server is live
    ∧ currentState.state = "started_write"
    ∧ ∃ id ∈ UUIDS :
        ∧ blobStoreState[id] = [status ↦ "UNSET", image ↦ "UNSET"]
        ∧ blobStoreState' = [blobStoreState EXCEPT
                                ![id] = [
                                    image ↦ currentState.image,
                                    created ↦ time
                                    ]]
        ∧ serverStates' = [serverStates EXCEPT
                                ![s].state = "wrote_blob",
                                ![s].imageId = id]
    ∧ UNCHANGED ⟨databaseState, operations⟩
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time

ServerWriteMetadataAndReturn(s) ≜
    LET currentState ≜ serverStates[s]IN
    LET terminationTime ≜ (currentState.start + 1)IN
    ∧ time < terminationTime  Can only start this state if server is live
    ∧ currentState.state = "wrote_blob"
    ∧ databaseState' = [databaseState EXCEPT
                            ![currentState.userId] = [
                                metadata ↦ currentState.metadata,
                                imageId ↦ currentState.imageId]]

    ∧ serverStates' = [serverStates EXCEPT
                                ![s].state = "waiting",
                                ![s].userId = "UNSET",
                                ![s].metadata = "UNSET",
                                ![s].image = "UNSET",
                                ![s].imageId = "UNSET"]
    ∧ UNCHANGED ⟨blobStoreState, operations⟩
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time

ServerFailWrite(s) ≜
    LET currentState ≜ serverStates[s]IN
    LET terminationTime ≜ (currentState.start + 1)IN
    ∧ time < terminationTime  Can only start this state if server is live
    ∧ serverStates[s].state ∈ {"started_write", "wrote_blob"}
    ∧ serverStates' = [serverStates EXCEPT
                                ![s].state = "waiting",
                                ![s].userId = "UNSET",
```

```
                                    ![s].metadata = "UNSET",
                                    ![s].image = "UNSET",
                                    ![s].imageId = "UNSET"]
    ∧ UNCHANGED ⟨databaseState, blobStoreState, operations⟩
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time
```

Server Reads

```
ServerStartRead(s) ≜
    LET currentState ≜ serverStates[s] IN
    ∧ serverStates[s].state = "waiting"
    ∧ ∃ u ∈ USERIDS :
            serverStates' = [serverStates EXCEPT
                                    ![s].state = "started_read",
                                    ![s].userId = u,
                                    The time a read request starts
                                    ![s].start = time
                                    ]
    ∧ UNCHANGED ⟨databaseState, blobStoreState⟩
    ∧ UNCHANGED operations
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time

ServerReadMetadata(s) ≜
    LET currentState ≜ serverStates[s] IN
    LET terminationTime ≜ (currentState.start + 1) IN
    ∧ time < terminationTime  Can only start this state if server is live
    ∧ currentState.state = "started_read"
    ∧ databaseState[currentState.userId].metadata ≠ "UNSET"
    ∧ serverStates' =
        [serverStates EXCEPT
            ![s].state = "read_metadata",
            ![s].metadata = databaseState[currentState.userId].metadata,
            ![s].imageId = databaseState[currentState.userId].imageId]
    ∧ UNCHANGED ⟨databaseState, blobStoreState⟩
    ∧ UNCHANGED operations
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time

ServerReadMetadataAndReturnEmpty(s) ≜
    LET currentState ≜ serverStates[s] IN
    LET terminationTime ≜ (currentState.start + 1) IN
```

```
    ∧ time < terminationTime  Can only start this state if server is live
    ∧ currentState.state = "started_read"
    ∧ databaseState[currentState.userId].metadata = "UNSET"
    ∧ serverStates' = [serverStates EXCEPT
                          ![s].state = "waiting",
                          ![s].userId = "UNSET",
                          ![s].metadata = "UNSET",
                          ![s].image = "UNSET",
                          ![s].imageId = "UNSET"]

    ∧ operations' = Append(operations,
                          [
                              type ↦ "READ",
                              userId ↦ currentState.userId,
                              metadata ↦ "UNSET",
                              image ↦ "UNSET"
                          ])
    ∧ UNCHANGED ⟨databaseState, blobStoreState⟩
    ∧ UNCHANGED cleanerVars
    ∧ UNCHANGED time

ServerReadBlobAndReturn(s) ≜
    LET currentState ≜ serverStates[s] IN
    LET terminationTime ≜ (currentState.start + 1) IN
    ∧ time < terminationTime  Can only start this state if server is live
    ∧ currentState.state = "read_metadata"
    ∧ operations' = Append(operations,
                          [
                              type ↦ "READ",
                              userId ↦ currentState.userId,
                              metadata ↦ currentState.metadata,
                              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
```