```
┌──────────────────────── MODULE swscrw ────────────────────────┐
EXTENDS Naturals, Integers, Sequences, FiniteSets, TLC, Bags
CONSTANT NumClients, MaxNumOp, Consistency, K
ASSUME Consistency ∈ {"Eventual", "Consistent_Prefix", "Session", "Bounded_Staleness", "Strong"}
ASSUME MaxNumOp < 10 ∧ NumClients = 1
Cloud ≜ 0
Clients ≜ 1 . . NumClients
--algorithm swscrw{
variables
   chan = [n ∈ 0 . . NumClients ↦ ⟨⟩] ;    FIFO channels

     network functions
    macro send( des, msg ) {
        chan[des] := Append(chan[des], msg) ;
     }

    macro receive( msg ) {
        await Len(chan[self]) > 0 ;
        msg := Head(chan[self]) ;
        chan[self] := Tail(chan[self]) ;
     }

    process ( cosmosdb ∈ {Cloud} )
    variables
        Database = ⟨0⟩ ; msg = ⟨⟩ ;
    { D: while ( TRUE ) {
             receive(msg) ;
             if ( msg.type = "Write" ) {
                Database := Append(Database, msg.dat) ;
    DW:         send(msg.orig, [type ↦ "Ack", dat ↦ Database[Len(Database)], ses ↦ Len(Database)]) ;  }
           else if ( msg.type = "Eventual" )
    DE:        with ( k ∈ 1 . . Len(Database) )
                  send(msg.orig, [type ↦ "Reply", dat ↦ Database[k], ses ↦ k]) ;
           else if ( msg.type = "Consistent_Prefix" )
    DP:        with ( k ∈ 1 . . Len(Database) )
                  send(msg.orig, [type ↦ "Reply", dat ↦ Database[k], ses ↦ k]) ;
           else if ( msg.type = "Session" )
    DS:        with ( k ∈ msg.ses . . Len(Database) )
                  send(msg.orig, [type ↦ "Reply", dat ↦ Database[k], ses ↦ k]) ;
           else if ( msg.type = "Bounded_Staleness" )
    DB:        with ( k ∈ (IF Len(Database) > K THEN Len(Database) − K ELSE 1) . . Len(Database) )
                  send(msg.orig, [type ↦ "Reply", dat ↦ Database[k], ses ↦ k]) ;
           else if ( msg.type = "Strong" )
    DG:        with ( k = Len(Database) )
                  send(msg.orig, [type ↦ "Reply", dat ↦ Database[k], ses ↦ k]) ;
        }
```

```
      }

    process ( client ∈ Clients )
    variables
        m = ⟨⟩ ; op = 0 ; v = 0 ; chistory = ⟨0⟩ ; ses = 1 ;
    {
     CR: while ( op < MaxNumOp ) {
            send(Cloud, [type ↦ Consistency, ses ↦ ses, orig ↦ self]) ;  read
     CRA:   receive(m) ;  Reply
            chistory := Append(chistory, m.dat) ;
            v := m.dat ;
            ses := m.ses ;
             write v + 1
     CW:    send(Cloud, [type ↦ "Write", dat ↦ v + 1, ses ↦ ses, orig ↦ self]) ;
     CWA:   receive(m)   ;  Ack
            ses := m.ses ;
            op := op + 1 ;
        }
      }

 }   \* end algorithm
 BEGIN TRANSLATION
VARIABLES chan, pc, Database, msg, m, op, v, chistory, ses

vars ≜ ⟨chan, pc, Database, msg, m, op, v, chistory, ses⟩

ProcSet ≜ ({Cloud}) ∪ (Clients)

Init ≜  Global variables
        ∧ chan = [n ∈ 0 . . NumClients ↦ ⟨⟩]
         Process cosmosdb
        ∧ Database = [self ∈ {Cloud} ↦ ⟨0⟩]
        ∧ msg = [self ∈ {Cloud} ↦ ⟨⟩]
         Process client
        ∧ m = [self ∈ Clients ↦ ⟨⟩]
        ∧ op = [self ∈ Clients ↦ 0]
        ∧ v = [self ∈ Clients ↦ 0]
        ∧ chistory = [self ∈ Clients ↦ ⟨0⟩]
        ∧ ses = [self ∈ Clients ↦ 1]
        ∧ pc = [self ∈ ProcSet ↦ CASE self ∈ {Cloud} → "D"
                                   □  self ∈ Clients  → "CR"]

D(self) ≜ ∧ pc[self] = "D"
          ∧ Len(chan[self]) > 0
          ∧ msg′ = [msg EXCEPT ![self] = Head(chan[self])]
          ∧ chan′ = [chan EXCEPT ![self] = Tail(chan[self])]
          ∧ IF msg′[self].type = "Write"
```

```
                    THEN $\land Database' = [Database \text{ EXCEPT } ![self] = Append(Database[self], msg'[self].dat)]$
                         $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“DW”}]$
                    ELSE $\land$ IF $msg'[self].type = \text{“Eventual”}$
                              THEN $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“DE”}]$
                              ELSE $\land$ IF $msg'[self].type = \text{“Consistent\_Prefix”}$
                                        THEN $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“DP”}]$
                                        ELSE $\land$ IF $msg'[self].type = \text{“Session”}$
                                                  THEN $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“DS”}]$
                                                  ELSE $\land$ IF $msg'[self].type = \text{“Bounded\_Staleness”}$
                                                            THEN $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“DB”}$
                                                            ELSE $\land$ IF $msg'[self].type = \text{“Strong”}$
                                                                      THEN $\land pc' = [pc \text{ EXCEPT } ![$
                                                                      ELSE $\land pc' = [pc \text{ EXCEPT } ![$
                         $\land$ UNCHANGED $Database$
               $\land$ UNCHANGED $\langle m, op, v, chistory, ses \rangle$

$DW(self) \triangleq \land pc[self] = \text{“DW”}$
                     $\land chan' = [chan \text{ EXCEPT } ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type \mapsto \text{“Ack”},$
                     $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“D”}]$
                     $\land$ UNCHANGED $\langle Database, msg, m, op, v, chistory, ses \rangle$

$DE(self) \triangleq \land pc[self] = \text{“DE”}$
                     $\land \exists k \in 1 \,..\, Len(Database[self]) :$
                         $chan' = [chan \text{ EXCEPT } ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type \mapsto \text{“Reply}$
                     $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“D”}]$
                     $\land$ UNCHANGED $\langle Database, msg, m, op, v, chistory, ses \rangle$

$DP(self) \triangleq \land pc[self] = \text{“DP”}$
                     $\land \exists k \in 1 \,..\, Len(Database[self]) :$
                         $chan' = [chan \text{ EXCEPT } ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type \mapsto \text{“Reply}$
                     $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“D”}]$
                     $\land$ UNCHANGED $\langle Database, msg, m, op, v, chistory, ses \rangle$

$DS(self) \triangleq \land pc[self] = \text{“DS”}$
                     $\land \exists k \in msg[self].ses \,..\, Len(Database[self]) :$
                         $chan' = [chan \text{ EXCEPT } ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type \mapsto \text{“Reply}$
                     $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“D”}]$
                     $\land$ UNCHANGED $\langle Database, msg, m, op, v, chistory, ses \rangle$

$DB(self) \triangleq \land pc[self] = \text{“DB”}$
                     $\land \exists k \in (\text{IF } Len(Database[self]) > K \text{ THEN } Len(Database[self]) - K \text{ ELSE } 1) \,..\, Len(Database[$
                         $chan' = [chan \text{ EXCEPT } ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type \mapsto \text{“Reply}$
                     $\land pc' = [pc \text{ EXCEPT } ![self] = \text{“D”}]$
                     $\land$ UNCHANGED $\langle Database, msg, m, op, v, chistory, ses \rangle$

$DG(self) \triangleq \land pc[self] = \text{“DG”}$
                     $\land$ LET $k \triangleq Len(Database[self])$IN
                    chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ “Reply
                ∧ pc′ = [pc EXCEPT ![self] = “D”]
                ∧ UNCHANGED ⟨Database, msg, m, op, v, chistory, ses⟩

cosmosdb(self) ≜ D(self) ∨ DW(self) ∨ DE(self) ∨ DP(self) ∨ DS(self)
                     ∨ DB(self) ∨ DG(self)

CR(self) ≜ ∧ pc[self] = “CR”
           ∧ IF op[self] < MaxNumOp
                 THEN ∧ chan′ = [chan EXCEPT ![Cloud] = Append(chan[Cloud], ([type ↦ Consistency, se
                      ∧ pc′ = [pc EXCEPT ![self] = “CRA”]
                 ELSE ∧ pc′ = [pc EXCEPT ![self] = “Done”]
                      ∧ chan′ = chan
           ∧ UNCHANGED ⟨Database, msg, m, op, v, chistory, ses⟩

CRA(self) ≜ ∧ pc[self] = “CRA”
            ∧ Len(chan[self]) > 0
            ∧ m′ = [m EXCEPT ![self] = Head(chan[self])]
            ∧ chan′ = [chan EXCEPT ![self] = Tail(chan[self])]
            ∧ chistory′ = [chistory EXCEPT ![self] = Append(chistory[self], m′[self].dat)]
            ∧ v′ = [v EXCEPT ![self] = m′[self].dat]
            ∧ ses′ = [ses EXCEPT ![self] = m′[self].ses]
            ∧ pc′ = [pc EXCEPT ![self] = “CW”]
            ∧ UNCHANGED ⟨Database, msg, op⟩

CW(self) ≜ ∧ pc[self] = “CW”
           ∧ chan′ = [chan EXCEPT ![Cloud] = Append(chan[Cloud], ([type ↦ “Write”, dat ↦ v[self] + 1,
           ∧ pc′ = [pc EXCEPT ![self] = “CWA”]
           ∧ UNCHANGED ⟨Database, msg, m, op, v, chistory, ses⟩

CWA(self) ≜ ∧ pc[self] = “CWA”
            ∧ Len(chan[self]) > 0
            ∧ m′ = [m EXCEPT ![self] = Head(chan[self])]
            ∧ chan′ = [chan EXCEPT ![self] = Tail(chan[self])]
            ∧ ses′ = [ses EXCEPT ![self] = m′[self].ses]
            ∧ op′ = [op EXCEPT ![self] = op[self] + 1]
            ∧ pc′ = [pc EXCEPT ![self] = “CR”]
            ∧ UNCHANGED ⟨Database, msg, v, chistory⟩

client(self) ≜ CR(self) ∨ CRA(self) ∨ CW(self) ∨ CWA(self)

Next ≜ (∃ self ∈ {Cloud} : cosmosdb(self))
          ∨ (∃ self ∈ Clients : client(self))

Spec ≜ $Init \land \Box[Next]_{vars}$

 END TRANSLATION
```

```
Messages ≜
        [type : {"Eventual", "Consistent_Prefix", "Bounded_Staleness", "Strong"}, dat : {0 .. Nat}, ses : {0 .. Nat
∪       [type : {"Reply", "Ack"}, dat : {0 .. Nat}, ses : {0 .. Nat}]

 Invariants for single client(ID = 1) writing with op ++
Eventual ≜ chistory[1][Len(chistory[1])] ∈ {Database[Cloud][i] : i ∈ 1 .. Len(Database[Cloud])}

Consistent_Prefix ≜ chistory[1][Len(chistory[1])] ∈ {Database[Cloud][i] : i ∈ 1 .. Len(Database[Cloud])}

Session ≜ pc[1] = "CW" ⇒ chistory[1][Len(chistory[1])] ∈ {Database[Cloud][i] :
i ∈ ses[1] .. Len(Database[Cloud])}

Bounded_Staleness ≜ pc[1] = "CW" ⇒ chistory[1][Len(chistory[1])] ∈ {Database[Cloud][i] :
i ∈ (IF Len(Database[Cloud]) > K THEN Len(Database[Cloud]) − K ELSE 1) .. Len(Database[Cloud])}

Strong ≜ pc[1] = "CW" ⇒ chistory[1][Len(chistory[1])] = Database[Cloud][Len(Database[Cloud])]
```