```
MODULE swscop
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 swscop{
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]) ;
       }
```

```
48      }

50      process ( client ∈ Clients )
51      variables
52          m = ⟨⟩ ; op = 0 ; chistory = ⟨0⟩ ; ses = 1 ;
53      {
54       CW: while ( op < MaxNumOp ) {
55               op := op + 1 ;
56               send(Cloud, [type ↦ "Write", dat ↦ op, ses ↦ ses, orig ↦ self]) ;
57       CWA:    receive(m)   ;  Ack
58               ses := m.ses ;
59                read
60       CR:     send(Cloud, [type ↦ Consistency, ses ↦ ses, orig ↦ self]) ;
61       CRA:    receive(m) ;   Reply
62               chistory := Append(chistory, m.dat) ;
63               ses := m.ses ;
64           }
65      }

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

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

74  ProcSet ≜ ({Cloud}) ∪ (Clients)

76  Init ≜   Global variables
77           ∧ chan = [n ∈ 0 .. NumClients ↦ ⟨⟩]
78           Process cosmosdb
79           ∧ Database = [self ∈ {Cloud} ↦ ⟨0⟩]
80           ∧ msg = [self ∈ {Cloud} ↦ ⟨⟩]
81           Process client
82           ∧ m = [self ∈ Clients ↦ ⟨⟩]
83           ∧ op = [self ∈ Clients ↦ 0]
84           ∧ chistory = [self ∈ Clients ↦ ⟨0⟩]
85           ∧ ses = [self ∈ Clients ↦ 1]
86           ∧ pc = [self ∈ ProcSet ↦ CASE self ∈ {Cloud} → "D"
87                                      □   self ∈ Clients  → "CW"]

89  D(self) ≜  ∧ pc[self] = "D"
90             ∧ Len(chan[self]) > 0
91             ∧ msg′ = [msg EXCEPT ![self] = Head(chan[self])]
92             ∧ chan′ = [chan EXCEPT ![self] = Tail(chan[self])]
93             ∧ IF msg′[self].type = "Write"
94                  THEN ∧ Database′ = [Database EXCEPT ![self] = Append(Database[self], msg′[self].dat)]
95                       ∧ pc′ = [pc EXCEPT ![self] = "DW"]
```

```
                    ELSE  ∧ IF msg′[self].type = "Eventual"
                                THEN  ∧ pc′ = [pc EXCEPT ![self] = "DE"]
                                ELSE  ∧ IF msg′[self].type = "Consistent_Prefix"
                                            THEN  ∧ pc′ = [pc EXCEPT ![self] = "DP"]
                                            ELSE  ∧ IF msg′[self].type = "Session"
                                                        THEN  ∧ pc′ = [pc EXCEPT ![self] = "DS"]
                                                        ELSE  ∧ IF msg′[self].type = "Bounded_Staleness"
                                                                    THEN  ∧ pc′ = [pc EXCEPT ![self] = "DB"
                                                                    ELSE  ∧ IF msg′[self].type = "Strong"
                                                                                THEN  ∧ pc′ = [pc EXCEPT ![s
                                                                                ELSE  ∧ pc′ = [pc EXCEPT ![s
                          ∧ UNCHANGED Database
                ∧ UNCHANGED ⟨m, op, chistory, ses⟩

DW(self) ≜  ∧ pc[self] = "DW"
            ∧ chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ "Ack",
            ∧ pc′ = [pc EXCEPT ![self] = "D"]
            ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

DE(self) ≜  ∧ pc[self] = "DE"
            ∧ ∃ k ∈ 1 .. Len(Database[self]) :
                chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ "Reply
            ∧ pc′ = [pc EXCEPT ![self] = "D"]
            ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

DP(self) ≜  ∧ pc[self] = "DP"
            ∧ ∃ k ∈ 1 .. Len(Database[self]) :
                chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ "Reply
            ∧ pc′ = [pc EXCEPT ![self] = "D"]
            ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

DS(self) ≜  ∧ pc[self] = "DS"
            ∧ ∃ k ∈ msg[self].ses .. Len(Database[self]) :
                chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ "Reply
            ∧ pc′ = [pc EXCEPT ![self] = "D"]
            ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

DB(self) ≜  ∧ pc[self] = "DB"
            ∧ ∃ k ∈ (IF Len(Database[self]) > K THEN Len(Database[self]) − K ELSE 1) .. Len(Database[
                chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ "Reply
            ∧ pc′ = [pc EXCEPT ![self] = "D"]
            ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

DG(self) ≜  ∧ pc[self] = "DG"
            ∧ LET k ≜ Len(Database[self])IN
                chan′ = [chan EXCEPT ![(msg[self].orig)] = Append(chan[(msg[self].orig)], ([type ↦ "Reply
            ∧ pc′ = [pc EXCEPT ![self] = "D"]
```

```
143                 ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

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

148 CW(self) ≜ ∧ pc[self] = "CW"
149              ∧ IF op[self] < MaxNumOp
150                   THEN ∧ op' = [op EXCEPT ![self] = op[self] + 1]
151                        ∧ chan' = [chan EXCEPT ![Cloud] = Append(chan[Cloud], ([type ↦ "Write", dat ↦
152                        ∧ pc' = [pc EXCEPT ![self] = "CWA"]
153                   ELSE ∧ pc' = [pc EXCEPT ![self] = "Done"]
154                        ∧ UNCHANGED ⟨chan, op⟩
155              ∧ UNCHANGED ⟨Database, msg, m, chistory, ses⟩

157 CWA(self) ≜ ∧ pc[self] = "CWA"
158              ∧ Len(chan[self]) > 0
159              ∧ m' = [m EXCEPT ![self] = Head(chan[self])]
160              ∧ chan' = [chan EXCEPT ![self] = Tail(chan[self])]
161              ∧ ses' = [ses EXCEPT ![self] = m'[self].ses]
162              ∧ pc' = [pc EXCEPT ![self] = "CR"]
163              ∧ UNCHANGED ⟨Database, msg, op, chistory⟩

165 CR(self) ≜ ∧ pc[self] = "CR"
166             ∧ chan' = [chan EXCEPT ![Cloud] = Append(chan[Cloud], ([type ↦ Consistency, ses ↦ ses[self
167             ∧ pc' = [pc EXCEPT ![self] = "CRA"]
168             ∧ UNCHANGED ⟨Database, msg, m, op, chistory, ses⟩

170 CRA(self) ≜ ∧ pc[self] = "CRA"
171              ∧ Len(chan[self]) > 0
172              ∧ m' = [m EXCEPT ![self] = Head(chan[self])]
173              ∧ chan' = [chan EXCEPT ![self] = Tail(chan[self])]
174              ∧ chistory' = [chistory EXCEPT ![self] = Append(chistory[self], m'[self].dat)]
175              ∧ ses' = [ses EXCEPT ![self] = m'[self].ses]
176              ∧ pc' = [pc EXCEPT ![self] = "CW"]
177              ∧ UNCHANGED ⟨Database, msg, op⟩

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

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

184 Spec ≜ Init ∧ □[Next]_vars

186 END TRANSLATION

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

Invariants for single $client(ID = 1)$ writing with op ++

$Eventual \triangleq chistory[1][Len(chistory[1])] \in \{Database[Cloud][i] : i \in 1 \,..\, Len(Database[Cloud])\}$

$Consistent\_Prefix \triangleq chistory[1][Len(chistory[1])] \in \{Database[Cloud][i] : i \in 1 \,..\, Len(Database[Cloud])\}$

$Session \triangleq pc[1] = \text{“CW”} \Rightarrow chistory[1][Len(chistory[1])] \in \{Database[Cloud][i] :$
$i \in ses[1] \,..\, Len(Database[Cloud])\}$

$Bounded\_Staleness \triangleq pc[1] = \text{“CW”} \Rightarrow chistory[1][Len(chistory[1])] \in \{Database[Cloud][i] :$
$i \in (\text{IF } Len(Database[Cloud]) > K \text{ THEN } Len(Database[Cloud]) - K \text{ ELSE } 1) \,..\, Len(Database[Cloud])\}$

$Strong \triangleq pc[1] = \text{“CW”} \Rightarrow chistory[1][Len(chistory[1])] = Database[Cloud][Len(Database[Cloud])]$