

Problem

Ensure that two parties collaborating on a DCR graph, can execute the activities in the graph without requiring any degree of trust. We want to ensure fairness in the sense that each collaborator will be able to execute their relevant activities (in so far as they can do so according to the specific DCR graph), and that no collaborator can repudiate executions of activities (neither their own nor other parties'). The resulting global state of the distributed executions must be a result of a valid execution sequence in the workflow.

1. There must be a happened-before relationship between an execution and the executions of its dependent activities 2. All serializations of the happened-before relationships must be valid execution sequences in DCR

Framework

Failure model: Byzantine failures with subsystems exhibiting only crashes (no key guessing)

Workflow: $G = (V, E)$

Activities: $V = \{v_1, \dots, v_n\}$

Activity States: $S = \{s_1, \dots, s_n\}$, where $s_i \subseteq \{Executed, Pending, Included\}$

Relation (E): $R = (\text{condition, response, milestone, include, exclude})$, $e = (v_x, v_y, r_i)$, $E = \{e_1, \dots, e_m\}$

Actors: $P = \{\text{Alice } (A), \text{Bob } (B)\}$.

History, $H_A(t)$, $H_B(t)$: Sequence of activity executions up to, and at time t , as perceived by A and B , respectively.

Execution: (v_i, t, p_i) , where v_i is the activity executed at time t , executed by actor p_i .

Requirements

Consensus :

Termination Eventually each correct process sets its decision value (single activity state)

Agreement Decision vector = activity state 1. Eventual agreement: We can poll for an history, after which agreement on current state is reached. 2. Partial agreement: Bottom is allowed as substitute for any activity state (not if you're responsible for that activity)

Integrity If p_i is correct, then all correct processes decide on v_i , or \perp as the i th component of their vector.

Concurrency : DCR-rules for concurrency (only independent activities can be concurrent) (Tie-breaking)

Correctness : The system must enforce the agreed-upon workflow, so that no party can obtain an advantage by acting out of turn or failing to fulfil an obligation to act.

Non-repudiation : v_i must be provably proposed by p_i , within a negligible probability.

Scenarios

Scenario 1 $A \rightarrow \bullet B$

Alice must never decide \perp for A . Bob must never decide \perp for either A or B .

Scenario 2 $A \rightarrow \bullet B, A \rightarrow \bullet C$

Alice, Bob and Charlie must always decide the same value for A .

Problem: Prevent Alice from telling Bob the state, but not Charlie. (Reduces to FLP.)

Scenario 3 $A \rightarrow \ast B, B \rightarrow \bullet C$

Alice and Bob must agree on the state of A , Bob and Charlie must agree on the state of B , only Charlie must agree on the value of C .

Scenario 4 $A \rightarrow \ast B, B \rightarrow \ast A$

Both Alice and Bob must agree on both A and B .

Problem: Concurrency.

Scenario 5 $A \rightarrow +C, A \rightarrow +D, B \rightarrow \ast C, B \rightarrow \ast D$

Both Alice and Bob must agree on only A and B , respectively. Charlie must agree on everything except D , and Dahlia must agree on everything except C .

Problem: Concurrency (expanded), not serially equivalent if C is included and D is excluded after A and B have executed concurrently.