Detecting Presence of Metastable Failure States in Distributed Systems

RS3LAB

Yugesh Kothari, Pin-Yen Huang, Sanidhya Kashyap

EPFL

Metastable Failures : *"Have you tried turning it off and on again"* ?

What is a Metastable Failure?

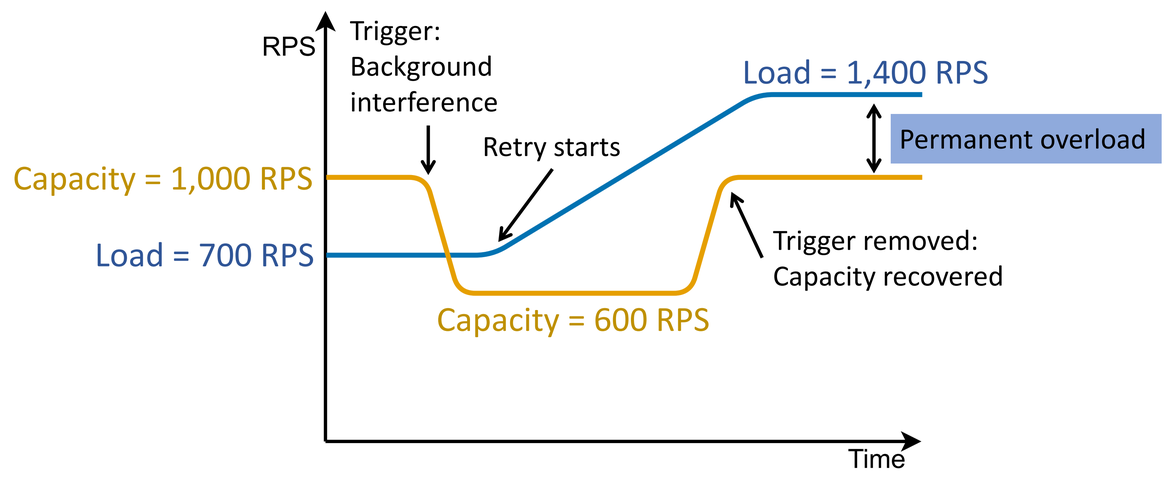
A crash-free, stable *down* state

Characterised by a permanent reduction in goodput of the system

Root cause is often a common-case optimisation for efficiency or reliability

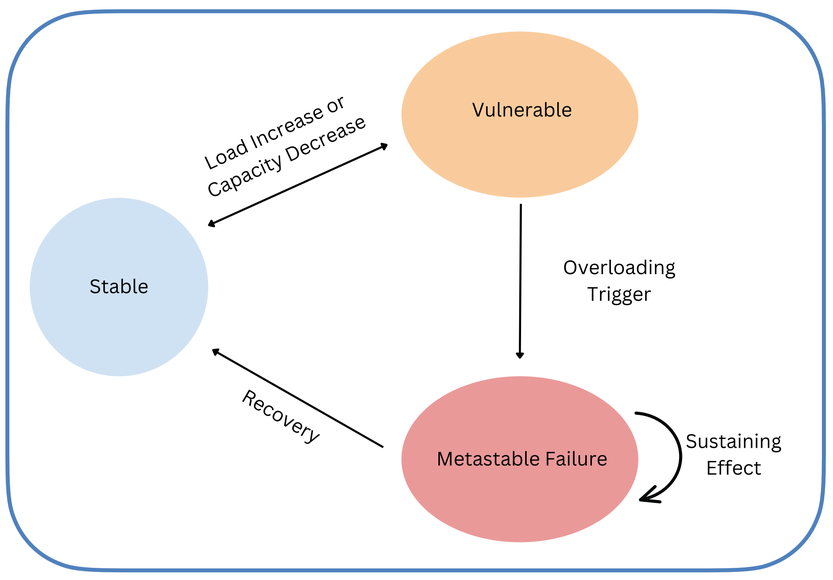
Cause catastrophic outages (4/15 major AWS outages in last decade)

- Example: Retry Storm



Takeaway: Permanent overload even after the trigger is removed

What is Metastability?



Salient Features

Triggered by an uncontrolled source of load *(overloading trigger)* when the system is running at peak capacity

A *sustaining effect* keeps the system overloaded even after the trigger is removed

System usually cannot recover without **load-shedding** or **restarts**

Building Distributed Systems that do not exhibit Metastability

Exploring config space for Bugs

A: Throughput *description* for each individual component

B: Summary of system goodput based on interactions and config

C: Simulator loop applies a **symbolic** *overloading trigger* over **symbolic** *config*

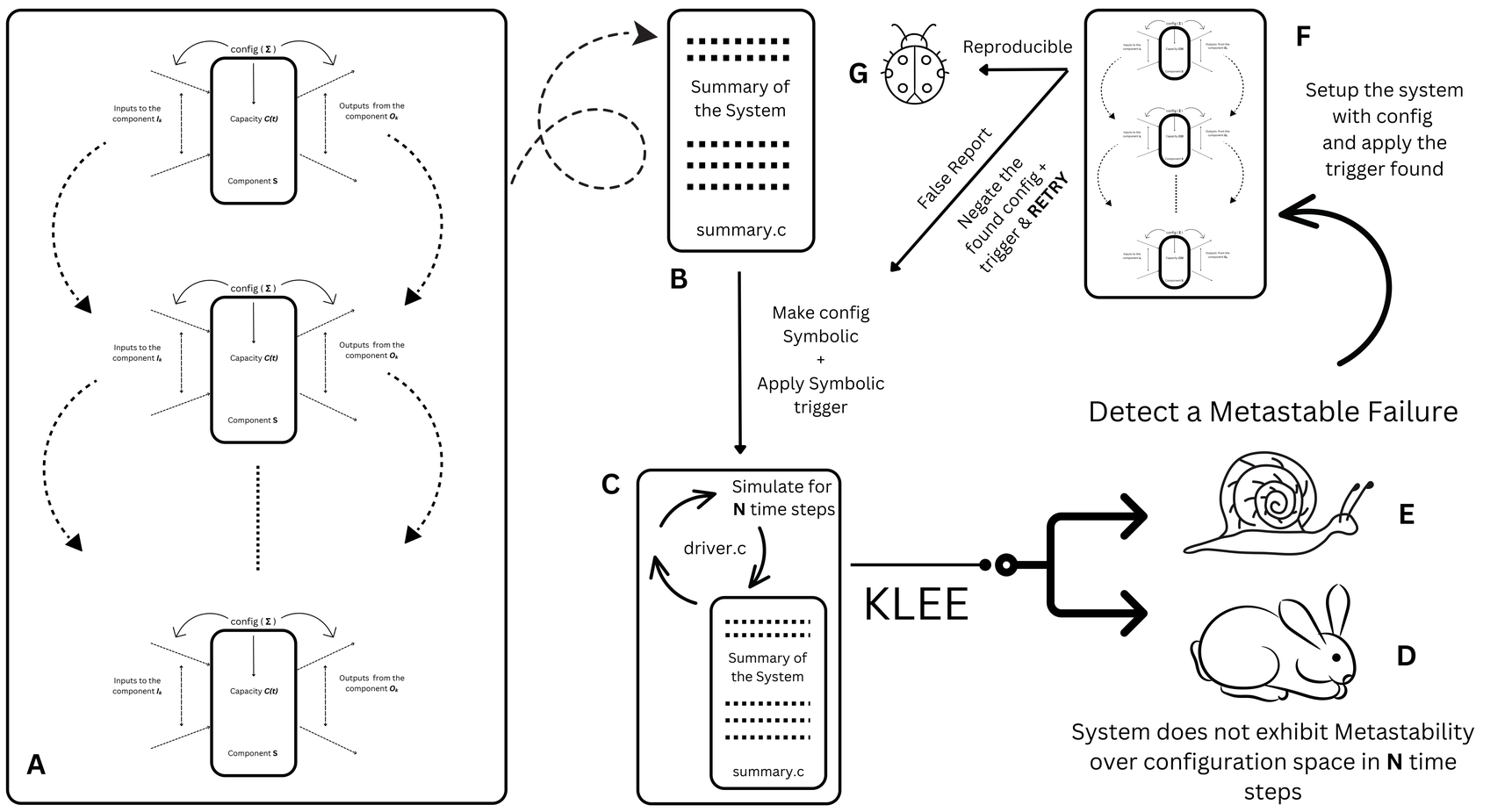
D: No metastability detected

E: Potential trigger found

F: Check if reported trigger is a false alarm

G: Valid trigger found; **system exhibits metastability for that configuration**

F→C: False alarm, retry SymbEx



Want to evolve automated verification techniques to reason about the behaviour of hyperscale software? Talk to us!