Detecting Metastability in Distributed Systems

TBD
EPFL
yugesh.kothari@epfl.ch

Abstract

Distributed systems present new and difficult to study failure modes.

Symbolically executing distrubuted systems is hard.

In this paper we discuss an idea on how to reason about distributed systems using symbolic execution along with a prelimiary evaluation of how this approach can be employed to detect metastable failures.

1 Introduction

Building reliable distributed systems is hard. resliability means uptime with good throughput.

This[1] work introduced mtsb as a class of failure mode for distribtued systems.

Sruvery of distributed system sfailures shows that detection and resiltion is hard.

There is also the general problem of

2 Background

A. Metastable Failures

Metastable failures are a class of failures

B. Symbolic Execution

Symbolic Execution is an automated verification technique

3 Methodology

3.1 Motivation

One of the key challenges in dealing with Metastable failures is that they are usually difficult to diagnose[ref]. This is because, unlike other traditional failures, they manifest not as semantic errors or crashes, but as consistent low good throughput (or goodput). Bronson *et al.* [ref] in their survery show that once such failures occur, it usually takes hours for engineers to bring the system back to its original throughput state.

The next challenge that the class of Metastable failures present is recovery. Aside from the significant downtime compared to other XXX: classical faliures YK: Use another phrase?, recovery from a Metastable failure state is also expensive, since it usually ends up requiring system reboots. System reboots as a mode of recovery come with two major drawbacks - one, it usually means downtime of a service for a noticable amount of time, and two depending on why the metastable fialure occured or what the reboot mechanism is, it could lead to a cascading failure i.e., the system quickly falls back into a metastable failure state because the root cause was not properly identified.

Since there is no obvious reason for this to happen, resolution of such failure states requires hours of diagnosis by site reliability engineers and usually ends in system-reboots.

Since diagnosis of the root case of a Metastable failure is hard, and recovery of the system from such a state is expensive, it becomes interesting to think about ways to detect both

The semantics dont matter, more than in as much they affect the output.

As a simulation model, if we could get it done then it would work.

3.2 Idea

A key characteristic of Metastable failures is that they manifest as a consequence of system configuration, or at the boundary interaction of components in a Distributed System. Put differently, it is important to realise that their occurance is not necessarily inferred from the semantics of processing one input by a component on its fast path, rather from the semantics of interaction of components based on externally observable state and configuration.

We use the above observation to come up with the notion of analysing the externally observable behaviour of a distributed system over a configuration space. The essence of this notion is the following - given a distributed system ,if we are able to distill out the semantics of how a given component will behave with respect to its own configuration, and the externally observable state and behaviour of other components it interacts with, we can then generate a *summary* of this component that is free from its own internal semantics of processing requests. The hope is that such summaries will be simple enough that they can be stitched together to come up with an abstract view of the distributed system that only exposes enough information to reason about the property we care about - which in our case is metastability. This notion is general enough to also support reasoning about other use cases.

Simulate the interaction of a distributed system.

Make parts of the smiulation symbolic i.e., variable.

3.3 Implementation

There are multiple potential XXX: techniquesYK: different word that can be employed to materialize the above idea XXX: add section number A setup that simualates abstract time stpes through a distributed execution.

4 Evaluation

In this section we present prelimiary results that evaluate the effectiveness of Symbolic Execution for detecting Metastability.

5 Conclusion

In this section we present prelimiary results that evaluate the effectiveness of Symbolic Execution for detecting Metastability.

References

[1] BRONSON, N., AGHAYEV, A., CHARAPKO, A., AND ZHU, T. Metastable failures in distributed systems. In *Proceedings of the Workshop on Hot Topics in Operating Systems* (New York, NY, USA, 2021), HotOS '21, Association for Computing Machinery, p. 221–227.