

Understanding, Detecting and Localizing Partial Failures in Large System Software¹

October 15, 2020

¹Chang Lou, Peng Huang, and Scott Smith. “Understanding, Detecting and Localizing Partial Failures in Large System Software”. In: *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*. 2020, pp. 559–574.

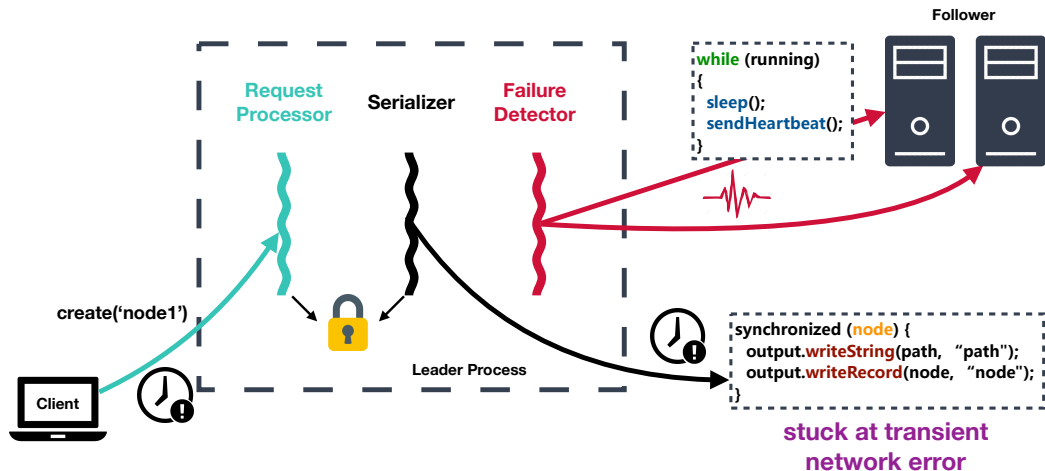
Overview

Problem definition

Case Study

What is a Partial Failure?

An Example

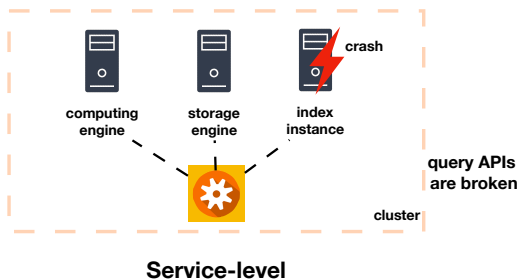
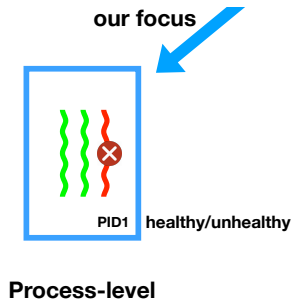


What is a Partial Failure?

Definition

A partial failure is, in a process π to be when a fault **does not** crash π but causes safety or liveness violation or severe slowness for some functionality $R_f \subsetneq R$

Scope: In this paper, we will specify the partial failure at the **process** granularity instead of **service**.



Study methodology

100 partial failure cases from five large, widely-used software systems

- ▶ Crawl all bug tickets tagged with critical priorities in the official bug trackers
- ▶ Filter tickets from testing and randomly sample the remaining failures tickets.

Interestingly, **54%** of them occur in the most recent **three** years' software releases
(*average lifespan of all systems is 9 years*)

Software	Language	Cases	Versions	Date Range
ZooKeeper	Java	20	17 (3.2.1–3.5.3)	12/01/2009–08/28/2018
Cassandra	Java	20	19 (0.7.4–3.0.13)	04/22/2011–08/31/2017
HDFS	Java	20	14 (0.20.1–3.1.0)	10/29/2009–08/06/2018
Apache	C	20	16 (2.0.40–2.4.29)	08/02/2002–03/20/2018
Mesos	C++	20	11 (0.11.0–1.7.0)	04/08/2013–12/28/2018