

Trace And Break (TAB): Improving Chaos Engineering by Examining Traces

Our abstract is:

Embracing a microservice architecture typically means embracing failures as a side effect. Complex service interactions, compartmentalized code ownership, and rapid release cycles all but guarantee problems will occur. To avoid folding in the face of such frequent failures, they need to be treated as expected results of normal system operation, and combated with fault tolerant designs.

To this end, the space of “chaos engineering,” or deliberately introducing faults into a system to test its resilience, has become popular. Unfortunately, as microservice systems become larger and larger, introducing failure in “the right place” becomes difficult. Introducing faults into a system is expensive, and with hundreds of services in a system the space of possible faults easily grows too large to explore fully. As such, intelligently selecting faults to inject is critical to a viable chaos engineering framework.

In this paper, we present TAB, a service mesh simulator for exploring heuristics for principled chaos engineering approaches. We present findings for the performance of a collection of novel heuristics we have implemented within this framework on different types of service meshes, and evaluate them. We conclude that simple heuristics are a promising approach for principled chaos engineering but are heavily dependent on the properties of the service mesh.