It Wasn't My Fault!

Understanding OS Fault Propagation Via Delta Execution

Cristiano Giuffrida, Lorenzo Cavallaro, and Andrew S. Tanenbaum
{giuffrida,sullivan,ast}@cs.vu.nl
Vrije Universiteit, Amsterdam, The Netherlands

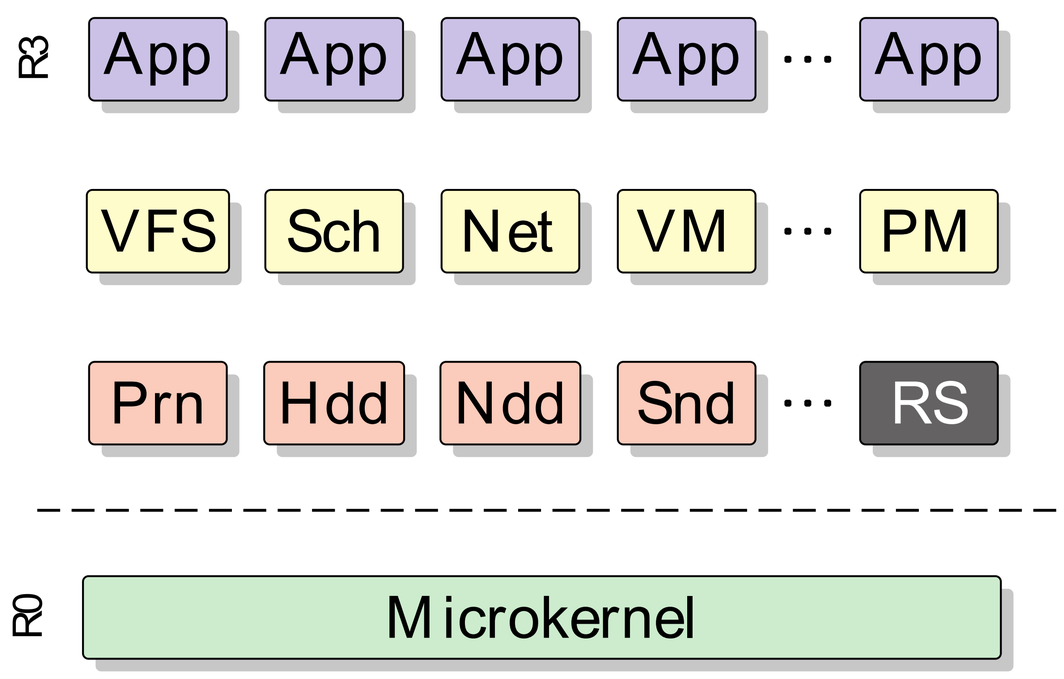
Research Summary

Problem: To recover from operating system crashes, we need to identify and eliminate the damage caused by the faulty execution that led to the crash.

Goal: Observe the way faults propagate throughout the operating system and analyze the behavior of the OS during faulty execution in a fine-grained way.

Our approach: Perform fault injection experiments and isolate the resulting faulty execution in a controlled environment. Compare faulty execution with fault-free execution online to identify all the significant differences.

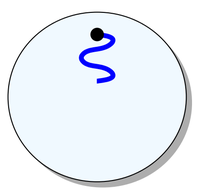
OS Architecture



- **The OS is broken down into several separate processes running in user space**
- **The proposed design results in a multiserver microkernel-based OS architecture**
- **OS InterProcess Communication (IPC) based entirely on message passing**

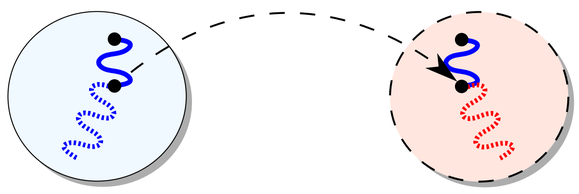
Fault Injection

1. Execute

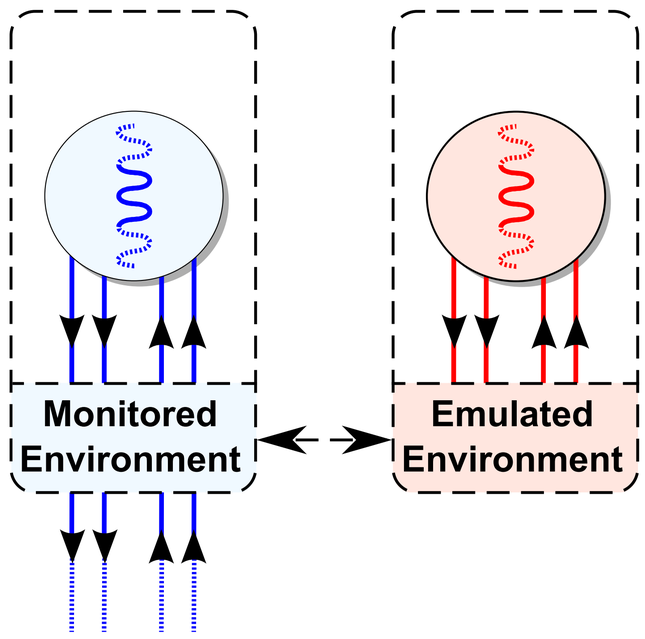


2. Inject Fault

3. Branch Execution



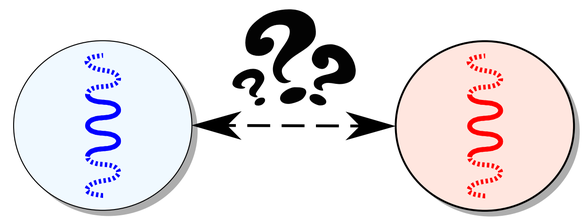
Delta Execution



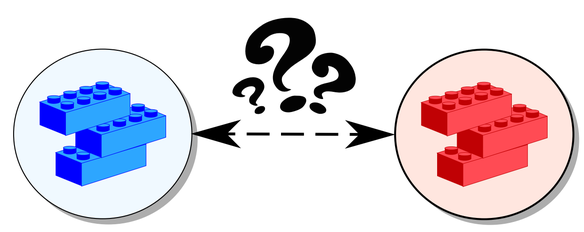
- **Monitor normal execution**
- **Synch at rendezvous points**
- **Replicate IPC traffic**

Online Comparison

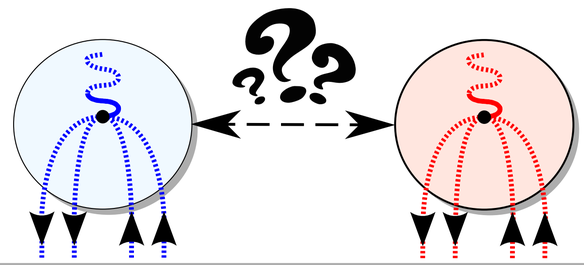
- **Compare Execution**



- **Compare State**



- **Compare IPC Interactions**



Supported by European Research Council under grant ERC Advanced Grant 2008 - R3S3