

emulab

Disk-Failure Injection Framework for Fault-Tolerant Systems Research

Yathindra Naik,
Mike Hibler, Eric Eide, Robert Ricci
University of Utah



Introduction

Motivation:

- Storage is one of the common problematic subsystems in a cloud environment.
- Need to make upper layers resilient to storage failures.
- Need a framework to study the impact of disk failures on a large testbed.

PRObE:

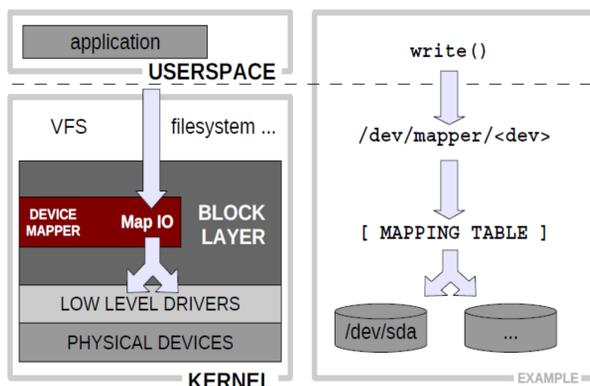
- Aims to build a large testbed for systems research with such a framework.

Goals:

- Lets users of Emulab testbed simulate various disk errors.
- Provide a scriptable and repeatable means to inject failures.
- Compress real disk-failure timelines into shorter timelines for experimentation.
- Replay I/O traces from real systems to model real disk failures.
- Try it out by signing up for a free account at <http://emulab.net/>

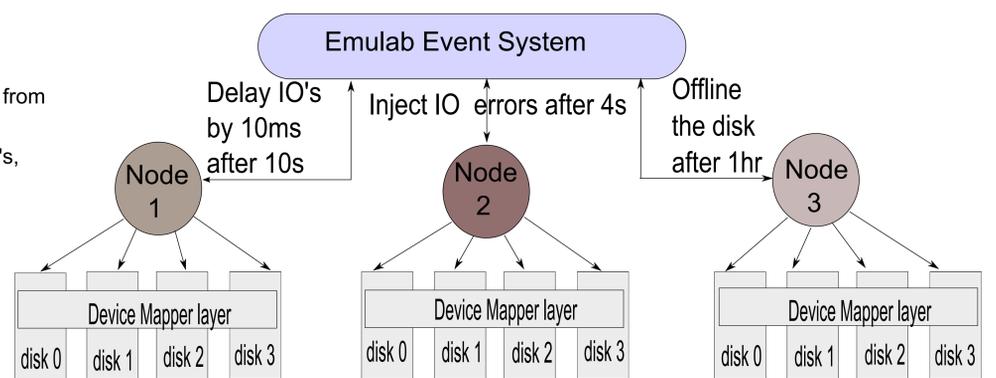
System Architecture

Device mapper – mapped device access



Types of failure targets

- Linear**: 1:1 Mapping of sectors from real disk to virtual disk
- flakey**: Fail a percentage of IO's, corrupt reads/writes
- Delay**: Delay IO's
- Error**: Mark sectors bad



1. Device Mapper on Linux

- Maps virtual disks onto real storage disks
- Provides various disk target types
- Target types can be used to simulate disk failures
- Ability to dynamically change disk target type

2. Event System/NS on Emulab

- Ability to schedule/trigger disk faults at later point in time
- NS syntax to script disk failure experiments

3. Disk-Agent for Emulab

- Interfaces libdevmapper and Event system
- Listens to disk events and invokes various disk failure conditions

Example

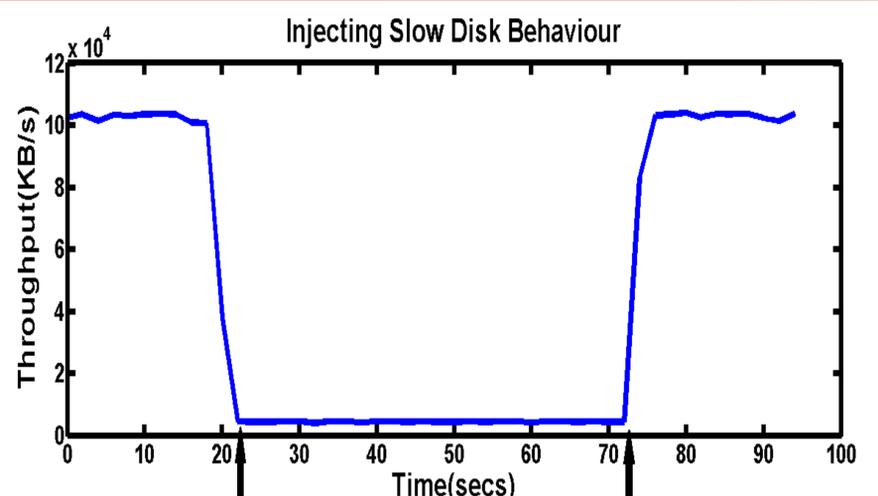
Typical NS TCL script to specify experiments on Emulab

```

set nodeA [$ns new node]
set disk0 [$nodeA disk-agent -type "linear" -mountpoint "/" -mnt"]
$ns at 0 "$disk0 run"
set disk0 [$nodeA disk-agent -type "delay" -mountpoint "/" -mnt" -parameters "100"]
$ns at 22 "$disk0 run"
set disk0 [$nodeA disk-agent -type "linear" -mountpoint "/" -mnt" ]
$ns at 72 "$disk0 run"

```

The above NS script allocates a physical node, specifies a disk which starts out being a good disk and 22 seconds later, we turn it into a slow disk by delaying the I/O's by 100ms. And then, at 72nd second we turn it back into a normal disk.



Injecting slow disk behaviour through Emulab Event system. The read throughput in KB/s observed when IO's are slowed by 100 ms.