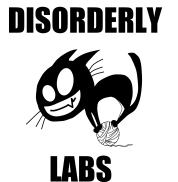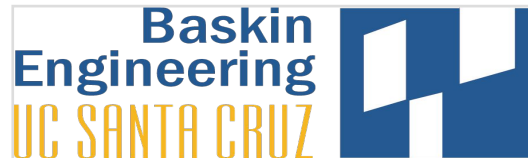# Growing a protocol

Kamala Ramasubramanian

Disorderly Labs
University of California, Santa Cruz

Our case-study

✦ Elastic - distributed data store vendor whose products focus on real time search and analysis

✦ Data replication protocol is <u>based on</u> Primary/Backup

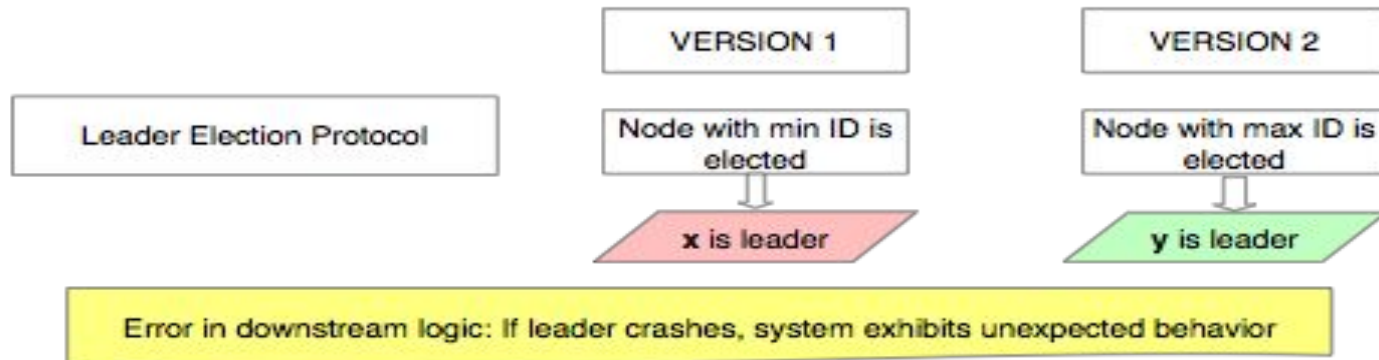✦ Interested in a tool to determine if protocol was behaving correctly in the presence of faults as it evolves

Introduction

✦ Software engineering best practices:
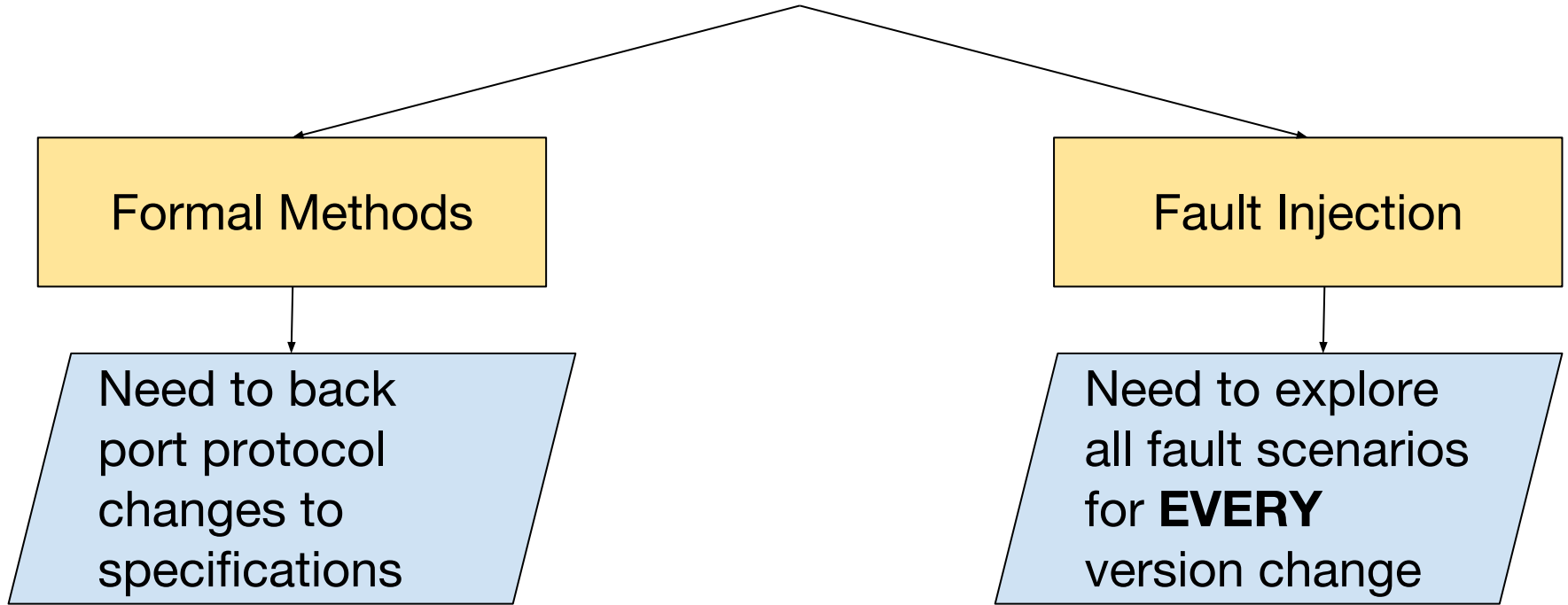
  ✦ Regression testing

  ✦ Root cause analysis

✦ Associate *specific inputs* with *system behaviors*

✦ Does not work while reasoning about fault tolerance properties of distributed systems

An example

Leader Election Protocol

VERSION 1
Node with min ID is elected
x is leader

VERSION 2
Node with max ID is elected
y is leader

Error in downstream logic: If leader crashes, system exhibits unexpected behavior

Naive Solution(s)

```
                            Formal Methods          Fault Injection
```

Formal Methods

Need to back port protocol changes to specifications

Fault Injection

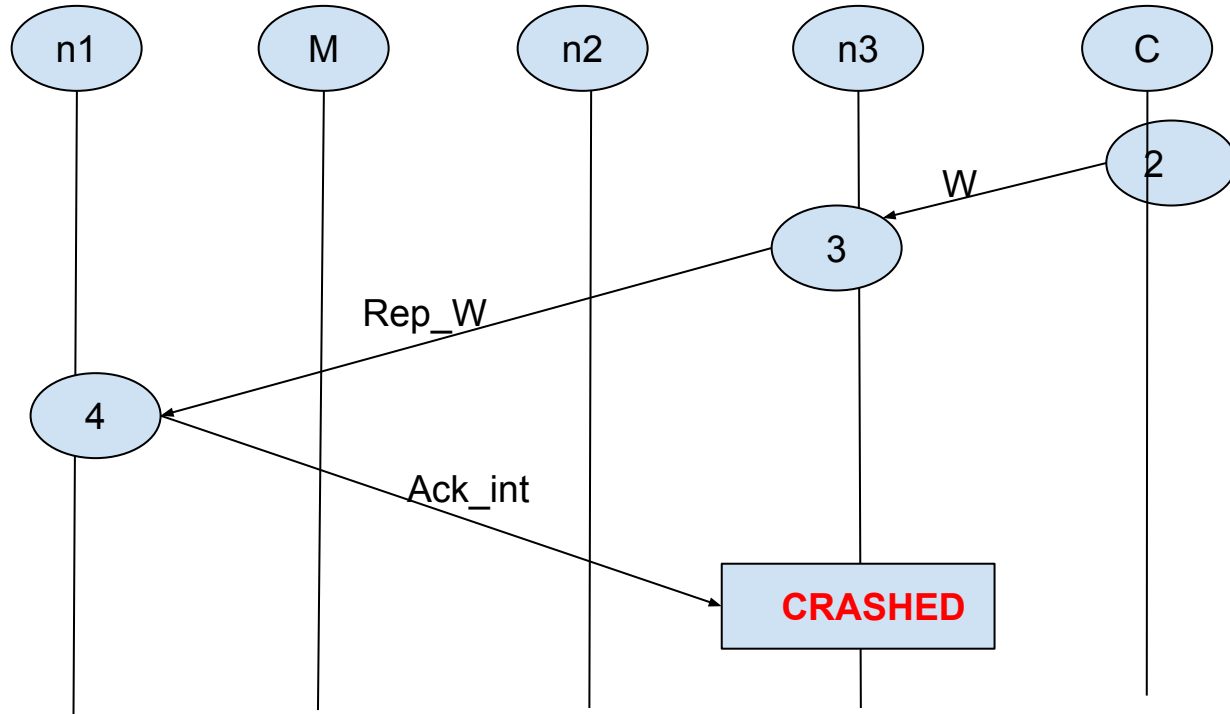Need to explore all fault scenarios for **EVERY** version change

A compromise?

- ✦ LDFI - Lineage Driven Fault Injection
  - ✦ Uses techniques from databases, logic programming and fault injection techniques

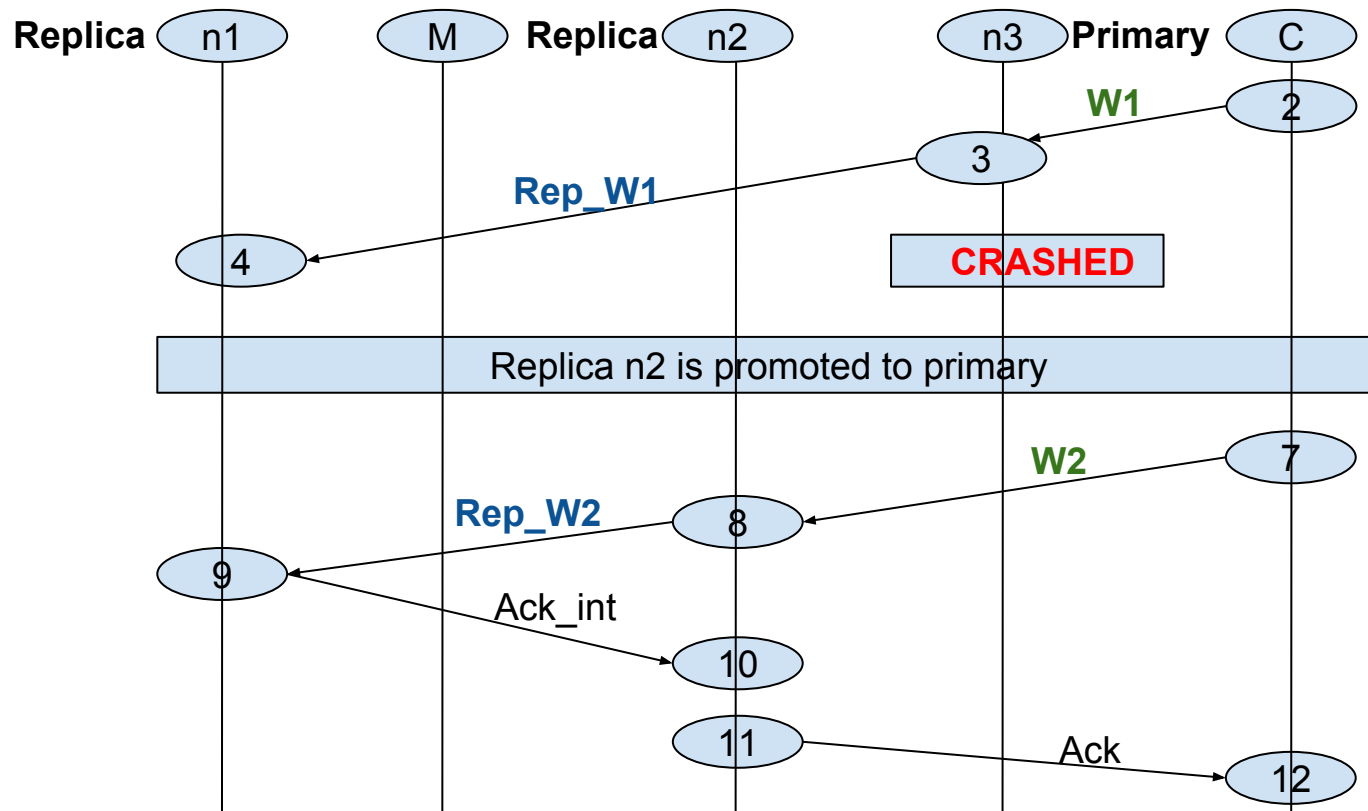- ✦ Reasons about *how* a good outcome occurred to determine *why* a bad outcome *might* occur

Catching Bugs early …
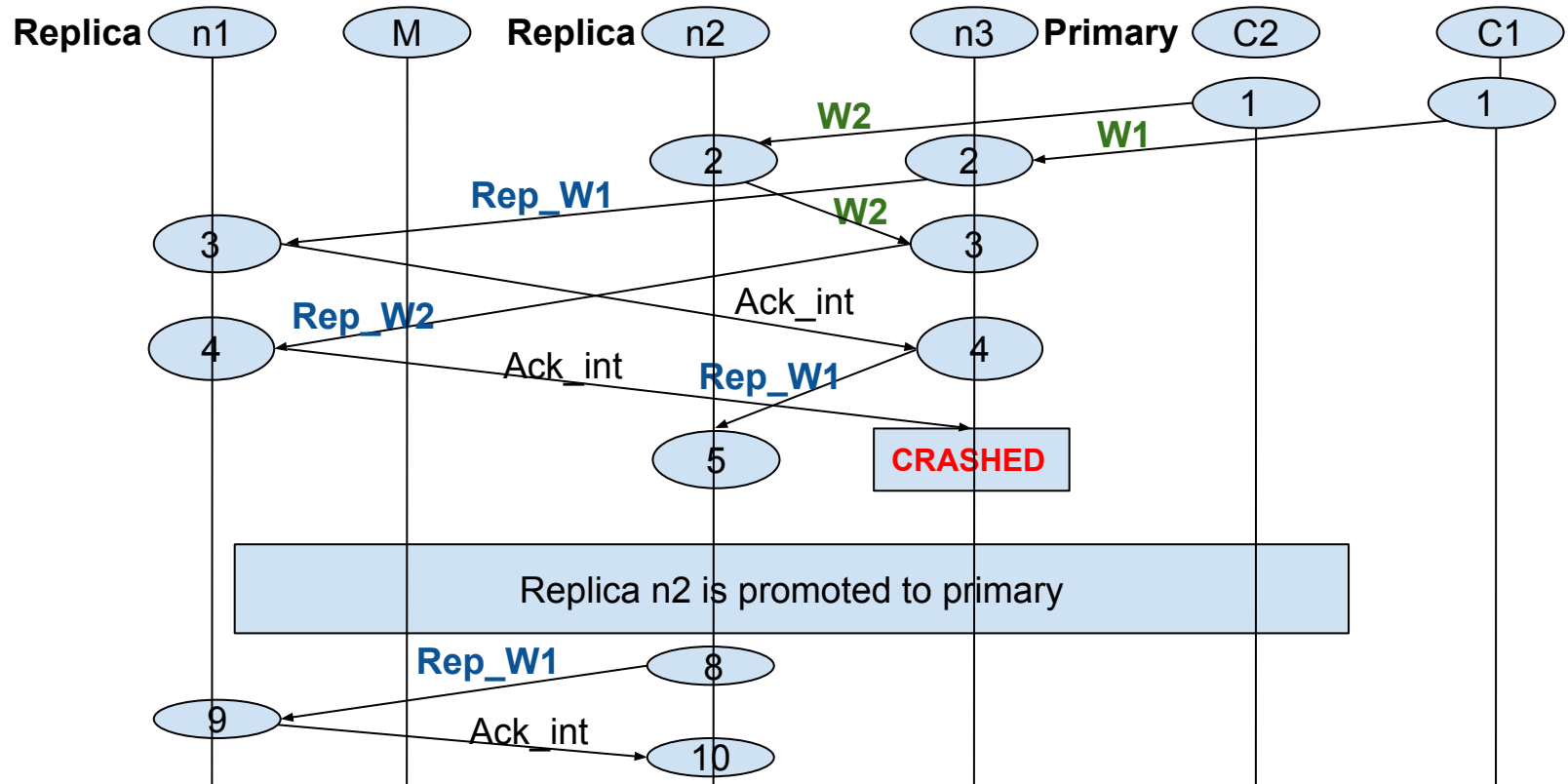
Dormant bugs

Avoiding expensive operations

A tale of two optimizations

- Specially handling certain cases to avoid expensive operations

- Re-sync mechanisms

The "simplicity" of an optimization is not a barometer in understanding if it *could* violate guarantees of the system

Past & Future Work

✦ Concurrency bugs

  ✦ Explore not just schedules, but reorderings as well.

✦ Input Generation

  ✦ Different fault scenarios based on input data selected. Impacts bugs found.

Summary

✦ Debugging distributed systems is hard!

✦ Need tools which straddle verification and testing

✦ Demonstrated that LDFI can be deployed as a tool in this space to find interesting bugs

Takeaways

- For fault tolerant properties, specific inputs do not characterize system behavior

- Interactions representing the same behavior can look wildly different

- Trusting your intuition can lead you wrong

Questions ?

kamala.ramas@ucsc.edu
https://github.com/KamalaRamas/molly