OS Support for File System Model Checking

Wei Su, Yifei Liu, Erez Zadok, et al.
File systems and Storage Lab, Department of Computer Science, Stony Brook University.

Stony Brook University
Computer Science

1. Background and Motivation

- **File systems are too complicated to be bug-free**
  - ➢ Corruption, Data loss, System crashes
- **Existing work on file system verification**
  - ➢ Cannot check corner cases (Regression suites)
  - ➢ Create file system from scratch (FSCQ SOSP'15)
  - ➢ Only specific type of bugs (eXplode OSDI'06)
  - ➢ Require effort to build a model (JUXTA SOSP'15)
- **Model checking framework MCFS**
  - ➢ Through coverage
  - ➢ Avoids requiring an abstract model
  - ➢ Keeps original behavior of file systems and OS
  - ➢ Applies to most file systems (kernel or user space)
  - ➢ Runs with high performance

3. Key Challenges and Our Attempts

- **Unbounded states to explore**
  - ➢ Compute abstract states to avoid duplicate states, see Figure 1
- **Cannot access in-memory states of file systems**
  - ➢ Only track the persistent states from backing storage
- **Cannot restore in-kernel states (cache incoherency)**
  - ➢ Unmount and remount file system b/w each syscall (hide bug!)
- **How to track full file system states?**
  - ➢ VeriFS: RAM-based FUSE file system, see Figure 2
    - ■ Provides checkpoint and restore APIs via `ioctl` (`ioctl_CHECKPOINT`, `ioctl_RESTORE`)

2. MCFS Framework Design

1. **Randomized Test Engines**
   - Issue system call sequences to each tested file system
2. **Optimized State-Space Exploration**
   - Lets MCFS execute all permutations
3. **Integrity Checks**
   - Verify all tested file systems have identical states
     - ➢ Any discrepancy <=> A possible bug ☹
4. **Abstraction Functions**
   - Convert concrete states into abstract ones
5. **Logging**
   - Reports precise sequences of operation for debugging and reproducibility

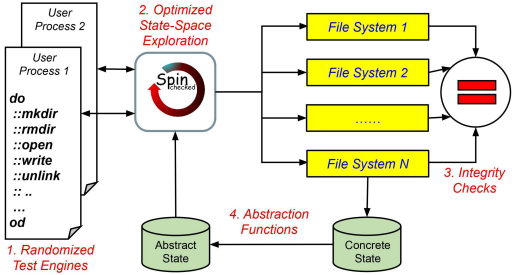

Figure 1: MCFS Model checking framework

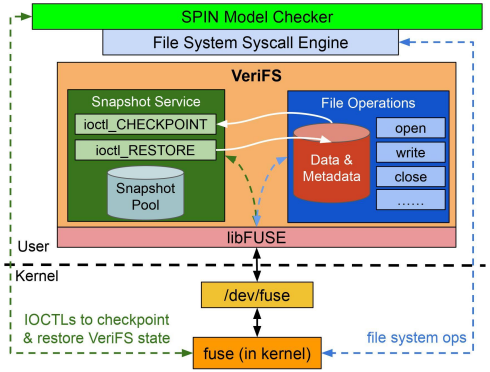

Figure 2: VeriFS architecture

4. Evaluation and Conclusions

- **Ability to find bugs**
  - ➢ Found two bugs for VeriFS
    - ■ Incorrect `truncate`
    - ■ Cache incoherency between OS kernel and VeriFS
  - ➢ Expect to discover bugs in other file systems
- **Conclusions**
  - ➢ Need OS-level Support to address our challenges
  - ➢ Can be applied to other system software
- **Future work**
  - ➢ Checkpoint/restore API for Linux VFS and Ganesha NFS
    - ■ Model-check more file systems
  - ➢ Address current MCFS limitations (e.g., false positives)
  - ➢ Swarm-verification runs model checking in parallel

fsl.cs.stonybrook.edu/~yifei
yifeliu@cs.stonybrook.edu