Upgrading Distributed Systems is not rsync

Sameer Ajmani
Barbara Liskov
Liuba Shrira

MIT Lab for Computer Science
Large, Long-Lived Distributed Systems Need Automatic Upgrades

- e.g., CDNs, peer-to-peer nets, sensor nets
- Software must change over time
- No direct operator access
- Upgrades must propagate automatically
- Upgrades must avoid disrupting service
A System Upgrade

- Each node runs at a particular version
- New version information propagates to nodes
- A node upgrades by shutting down, installing new code, transforming its persistent state, and restarting
- These systems are robust, so they tolerate node restarts
Upgrades Can Disrupt Service

• A node cannot service requests while upgrading
• May need to delay a node upgrade
  – To avoid causing too many simultaneous failures
  – To test an upgrade on a few nodes
• Implies *mixed mode* operation
Mixed Mode Operation

Version 2 → Version 1
Simulation Enables Interoperation

- Nodes label outgoing calls with their version
- Nodes dispatch incoming calls to *simulation objects*
  - Installing a simulation object is much faster than upgrading
- Both upgraded and non-upgraded nodes use simulation
- Imperfect simulation may degrade service
Summary

• Upgrades require scheduling and simulation
• Other issues (covered in the paper)
  – Transforms for persistent state
  – Correctness of transforms and simulation
  – An upgrade infrastructure that supports upgrade propagation, scheduling, code installation, transforms, and simulation