

Dynamic Software Updating for the Linux Kernel

Iulian Neamtiu

Michael Hicks

University of Maryland

Why On-the-fly Kernel Updates?

- Software updates - necessary evil
 - Inconvenient, expensive
- OS update via stop/restart disruptive
 - Loss of OS/application state
- *Dynamic Software Updating* (DSU)
 - User-space programs: easy, safe
 - Challenge: kernel

DSU for User-space Applications

- Ginseng: update C programs while they run
[PLDI'06]
 - Indirect type accesses, function calls
 - Dynamic patch: new/changed code&data, transformers
 - Update: load dynamic patch
 - Safety analyses (type safety)
- Results
 - Off-the-self apps: VsFTPd, OpenSSH, Zebra
 - 3 years of releases: 2002 - 2005
 - Easy to use: minor changes to apps/patches
 - Good performance: 0..30% overhead
 - Minimal disruption: < 5 *ms*

DSU for the Kernel

- Safety challenges
 - Low-level, highly concurrent code
 - Transactions
- Layout & performance constraints
 - Selective indirection

Better Safety with Transactions

- Ginseng enforces *representation consistency*
 - Type safety: old code/new data or vice versa
- Transactions provide *version consistency*
 - Programmer-specified blocks: code/data from same version
 - Delineate logical events (e.g., ADT, top+bottom half)
 - No commit, rollback, log
 - Enforcement: static analyses + light dynamic checks
- When is it safe to update ?
 - Code outside transactions, or
 - Transaction doesn't conflict with update

Selective Indirection

- Performance/representation constraints
- Types
 - Fixed representation/no change expected
 - E.g., page table entry, IP address
 - Non-indirected types updated manually
- Functions
 - Indirection/patch size trade-off
- Static analysis-driven

Conclusions

- Updating the kernel dynamically...
 - Compile kernel specially (selective indirection)
 - Automatic patch generation
 - Safety analyses (version consistency)
- ...leads to better OS maintenance
 - Wide range of updates applied on the fly
 - Security patches, bug fixes, new features
 - Updates easy to construct, safe to apply

<http://www.cs.umd.edu/projects/dsu>