JIT Kernels: An Idea Whose Time Has (Just) Come
Varun Agrawal, Amit Arya, Michael Ferdman, Donald E. Porter

Research Problem
- OS Kernels can be more flexible and perform better
- Poor performance
  - Optimized for wrong machine
    - Stock-compiled for Athlon-64, run on Core-i7
  - Optimized for expected code path and configuration
  - Tons of conditional and dynamically-dead code
  - “Expensive” features not in stock build
- Poor flexibility
  - Hard to patch without rebooting
  - Hard to debug
  - Hard to change configuration and hardware
  - Typically just resort to recompile and reboot

Opportunity
- JIT techniques developed for user software
  - Mature for high-level languages (e.g., Java, .Net)
  - Emerging for C/C++
  - Enable dynamic runtime optimization

Idea: JIT Kernel
- Ship intermediate representation (IR) of kernel
  - IR includes all architectures, devices, and options
  - JIT compile to specific deployment at boot
- Performance improvement
  - Code optimized for actual hardware
  - Dead code eliminated
  - Conditional code (sysctl) optimized
  - Profile-guided optimization can run in idle loop
- Flexibility improvement
  - OS live patching without reboot
  - Deploy one IR everywhere
    - Tailor to HW, kernel config
  - Debugging: Dynamically instrument live code

Challenges
- Native ASM routines must be linked with JITed code
- Cooperative resource management
  - JIT and kernel share memory, CPU time
  - Coordinate recompilation on system changes
  - Idle-time instrumentation and re-optimization
- Tracking data structure definition changes at runtime
- Retaining all #ifdef code in the IR
- Limitations of LLVM

Summary & Status
- Kernel JIT will enable...
  - High performance from tailoring the OS to...
    - Hardware, configuration, and workload
  - Ease of deployment – build once, run everywhere
  - Ease of debugging and security patching
    - Dynamic instrumentation and recompilation
- We JIT+boot FreeBSD on bare metal!
  - Run LLVM on bare metal
  - ASM routines dynamically linked into JIT kernel
- Next Steps
  - Explore optimizations
  - Re-JIT support