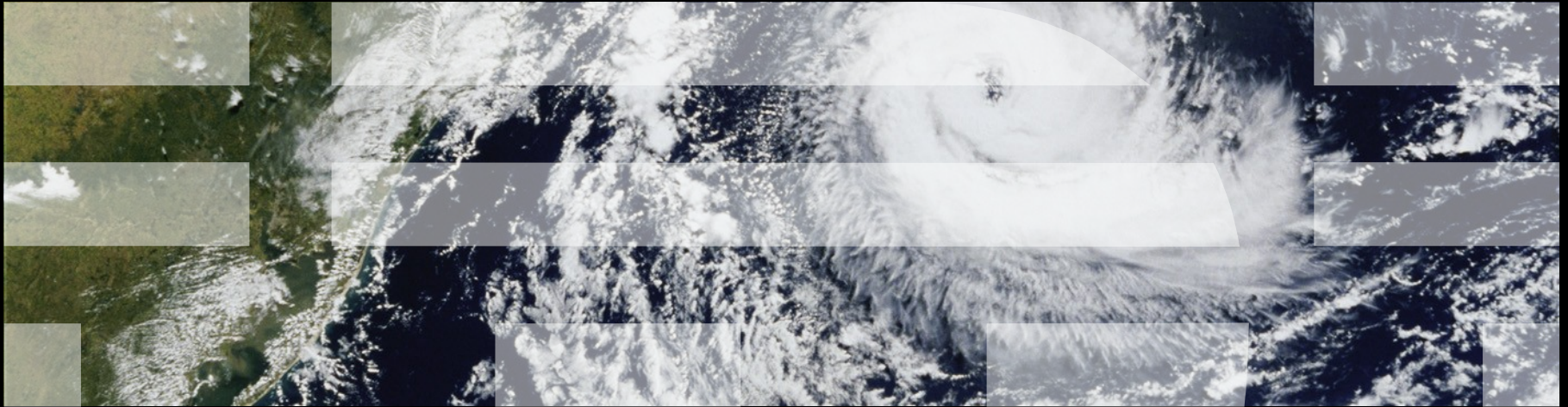


Paul E. McKenney, IBM Distinguished Engineer, Linux Technology Center

October 21, 2012



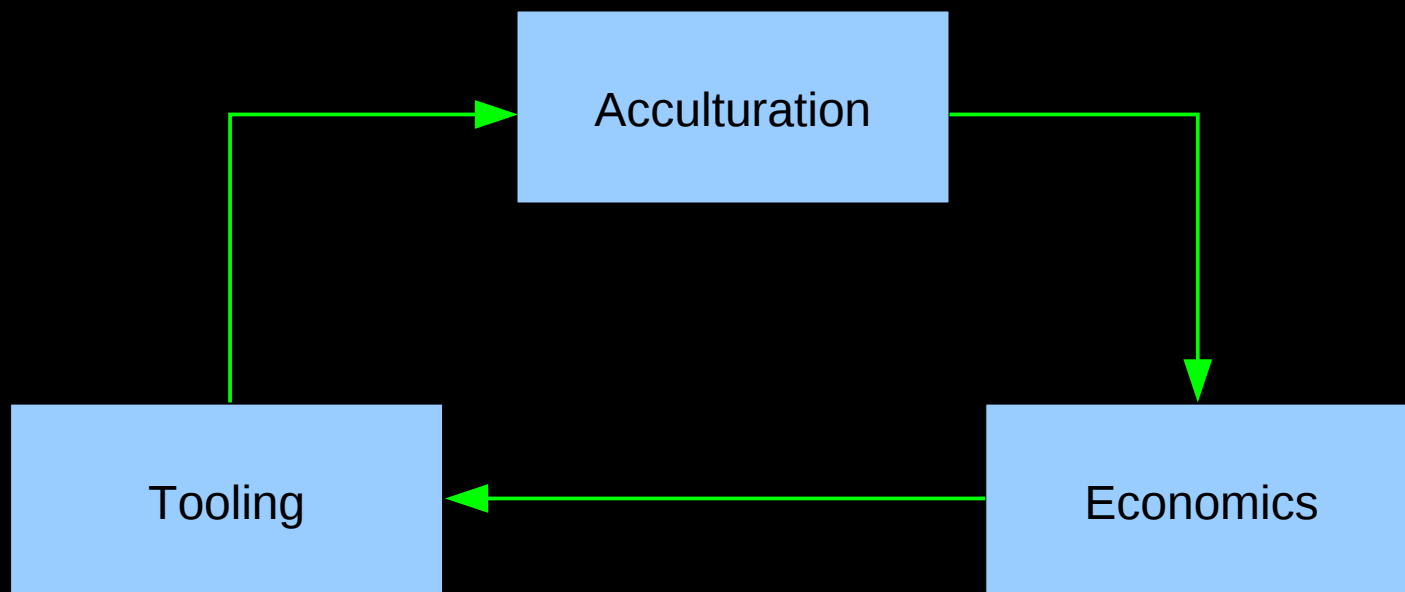
Beyond Expert-Only Parallel Programming?



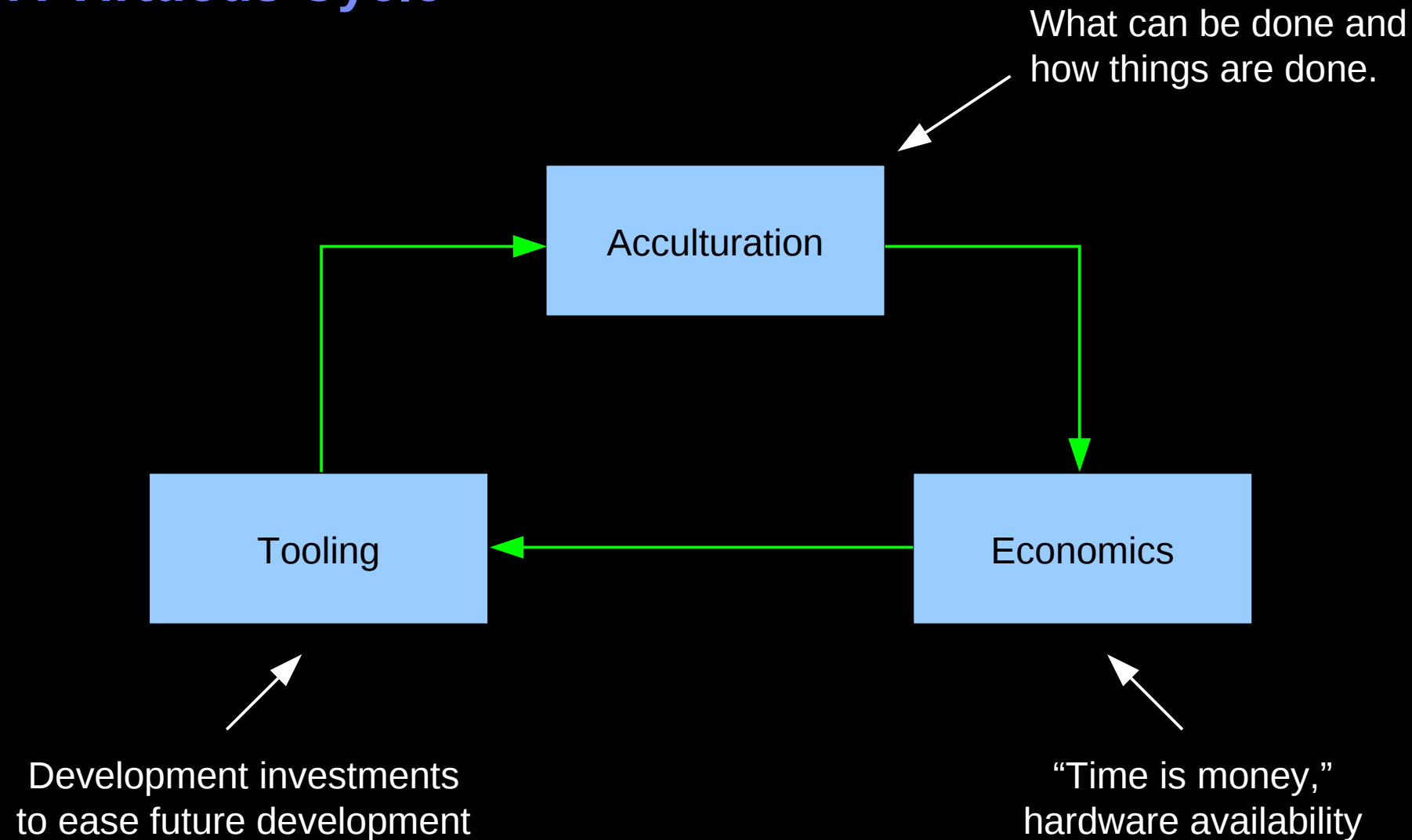
Sea Change In Linux-Kernel Parallel Programming

- In 2006, Linus Torvalds noted that since 2003, the Linux kernel community's grasp of concurrency had improved to the point that patches were often correct at first submission
- Why the improvement?
 - Not programming language: C before, during, and after
 - Not synchronization primitives: Locking before, during, and after
 - Not a change in personnel: Relatively low turnover
 - Not born parallel programmers: Remember Big Kernel Lock!
- So what was it?

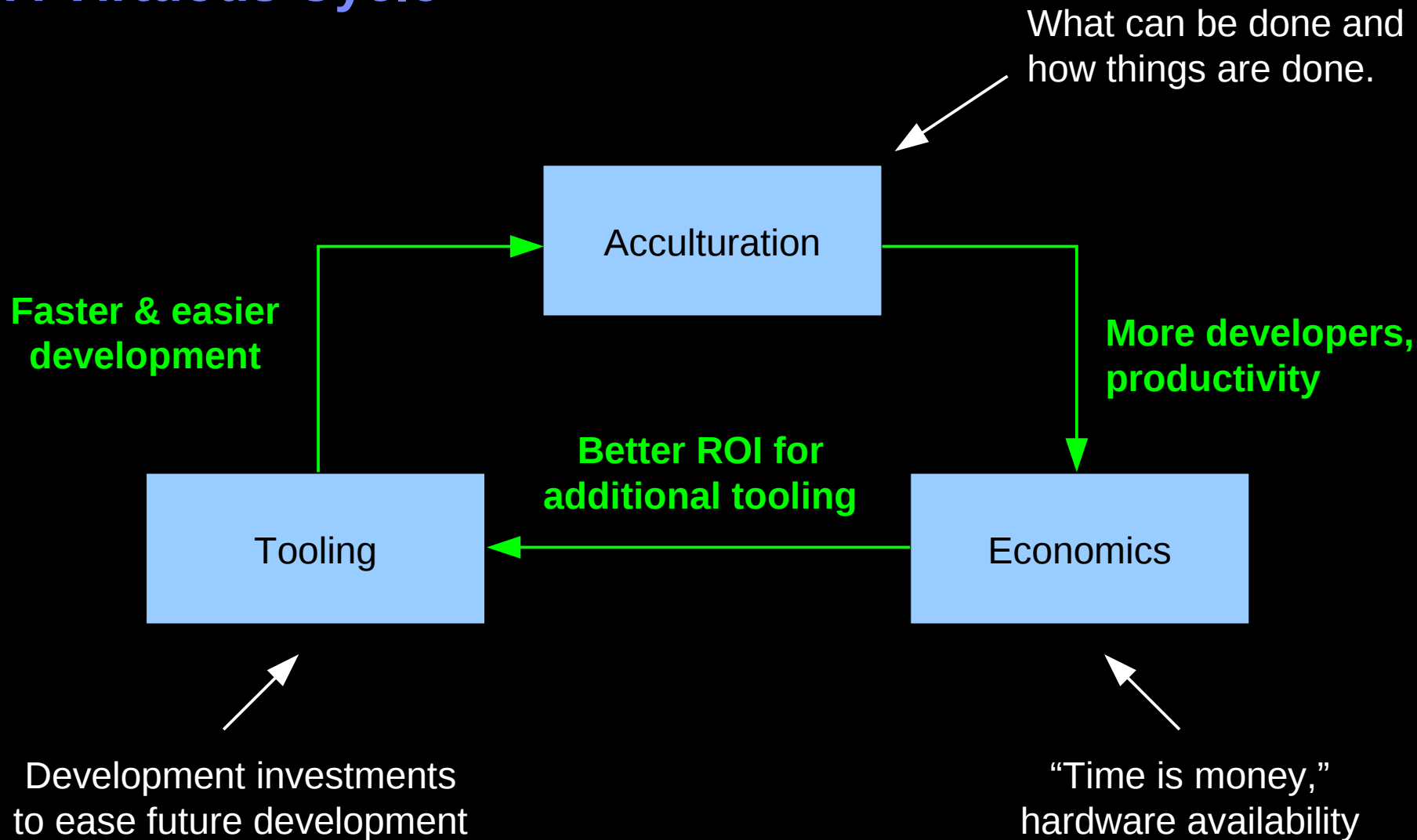
Sea Change In Linux-Kernel Parallel Programming: A Virtuous Cycle



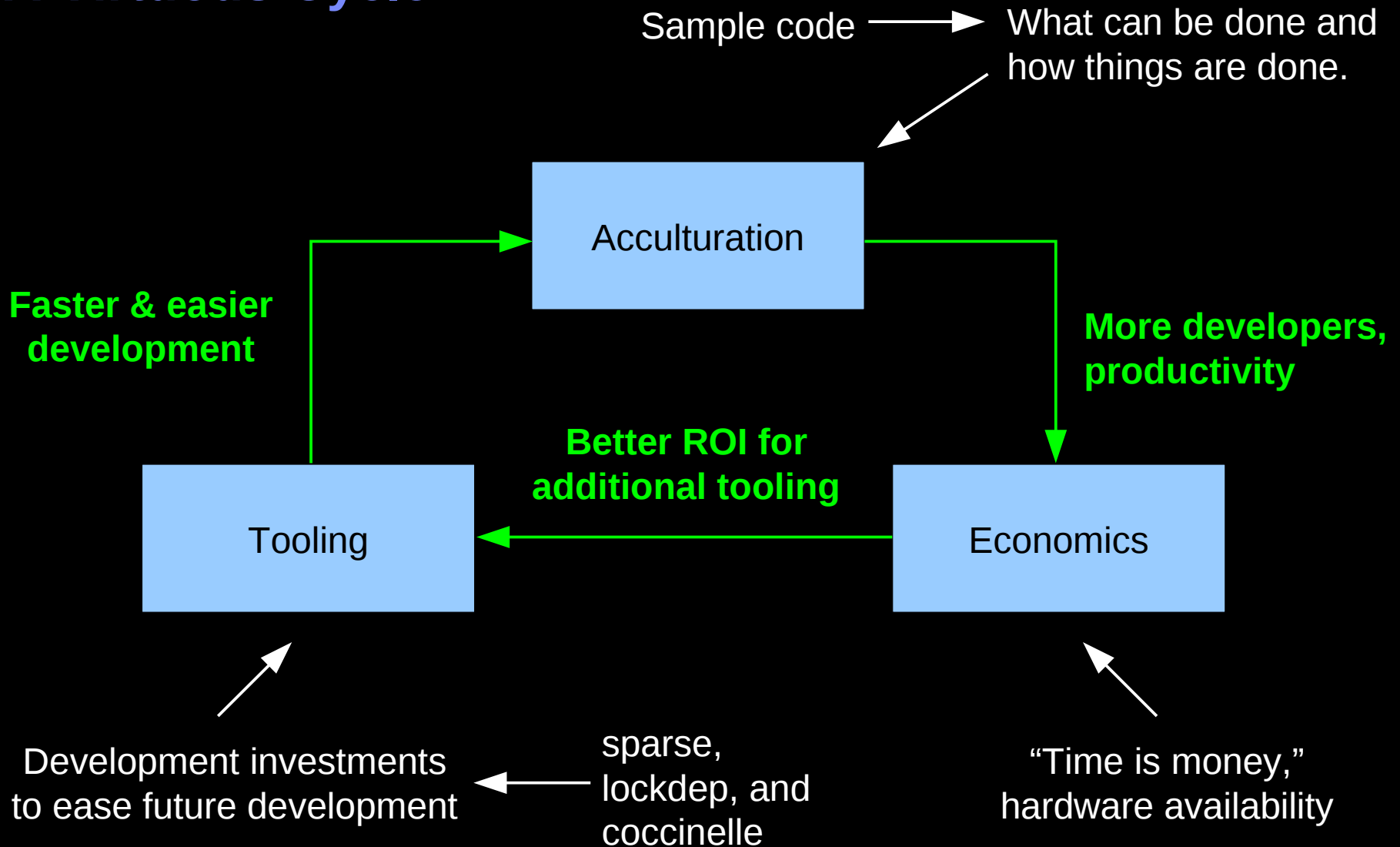
Sea Change In Linux-Kernel Parallel Programming: A Virtuous Cycle



Sea Change In Linux-Kernel Parallel Programming: A Virtuous Cycle



Sea Change In Linux-Kernel Parallel Programming: A Virtuous Cycle



How to Motivate Tooling?

- **sparse**: Motivated by user/kernel pointer errors
 - And by ca-2004 excitement over static analysis of software
 - Extended to concurrency (locking and RCU): approximate analysis
- **lockdep**: Motivated by ca. 2004-5 real-time work
 - Preemptibility greatly increases the probability of concurrency bugs
 - Real-time developers got tired of fixing others' deadlocks
 - The lockdep tool further increases the probability of detecting deadlock
 - Forcing developers to learn to avoid deadlocks
- **coccinelle**: Academic project! (“sed” that understands C)
 - Tested project on Linux kernel, submitted bug reports and fixes
 - Fixes can be automatically generated
 - Patches from coccinelle have been in the top-20 developers
 - (See contributions from Julia Lawall)

What is Left to Work On?

- SMP systems with many hundreds (or thousands) of CPUs
- Special-purpose hardware accelerators (GPGPUs, FPGAs)
- Parallel systems with real-time/energy-efficiency constraints
- Parallelize difficult-to-parallelize applications
- Unit volumes of multicore embedded systems: >100 million
 - Extreme reliability required: With the proper requirements set out!
- Codifying current expert-only techniques for general use
- Rigorous theoretical models of current expert-only techniques
- [Your idea here]

Discussion