Does RCU Really Work?

And if so, how do we know?
More Than 1.5 Billion Linux Instances Running!!!
More Than 1.5 Billion Linux Instances Running!!!
Woo-Hoo!!! Linux Has Won!!!
More Than 1.5 Billion Linux Instances Running!!!
Woo-Hoo!!! Linux Has Won!!!

But How The #$@%! Do I Validate RCU For This???
How The #$@%! Do I Validate RCU For This???

- A race condition that occurs once in a million years happens several times per day across the installed base
  - I am very proud of rcutorture, but it simply cannot detect million-year races when running on a reasonable test setup
  - Even given expected rcutorture improvements
Maybe Formal Verification?
Maybe Formal Verification? Such as CBMC?

- **C Bounded Model Checker (CBMC)** applies long-standing hardware verification techniques to software.

- **Easy to use:** Given recent Debian-derived distributions:
  
  ```bash
  sudo apt-get install cbmc
  cbmc filename.c
  ```

- **If no combination of inputs can trigger an assertion or cause an array-out-of-bounds error, it prints:**
  
  ```
  VERIFICATION SUCCESSFUL
  ```

- **And since 2015, CBMC handles concurrency!!!**
How Does CBMC Work?

C Code → Logic Expression → SAT Solver → Trace Generation → Verification Result

CBMC
A Few Questions/Objections You Might Have...

- But C is Turing-complete and logic expressions are not!!!
  - Yes, hence “bounded”. You can specify loop/recursion unrolling limits

- But SAT is NP-complete!!!
  - True, but there are now amazing heuristics for SAT
  - 1990: World-class solver handles 100 variables (three 32-bit variables)
  - 2015: x86 laptop does 2M variables. In ten seconds.

- How CBMC possibly handle concurrency???
  - Convert C program to SSA, wire reads to writes using memory model

- If this is really useful, why don't you apply it to RCU???
  - I checked CBMC verification of SRCU into -rcu on December 31, 2016
  - Implementation courtesy of Lance Roy

- Has CBMC really found any SRCU bugs???
  - Yes, though only injected bugs used to test the verification
Other Questions or Comments?

- But C is Turing-complete and logic expressions are not!!!
  - Yes, hence “bounded”. You can specify loop/recursion unrolling limits

- But SAT is NP-complete!!!
  - True, but there are now **amazing** heuristics for SAT
  - 1990: World-class solver handles 100 variables (three 32-bit variables)
  - 2015: x86 laptop does 2M variables. In ten seconds.

- How CBMC possibly handle concurrency???
  - Convert C program to SSA, wire reads to writes using memory model

- If this is really useful, why don't you apply it to RCU???
  - I checked CBMC verification of SRCU into -rcu on December 31, 2016
  - Implementation courtesy of Lance Roy

- Has CBMC really found any SRCU bugs???
  - Yes, though only injected bugs used to test the verification
Legal Statement

- This work represents the view of the author and does not necessarily represent the view of IBM.
- IBM and IBM (logo) are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.