THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

June 22nd, 2022

Krepair: Automatically Repairing .config Files to Cover Patches

Paul Gazzillo, Necip Yildiran
University of Central Florida

UCF
UNIVERSITY OF CENTRAL FLORIDA

#ossummit @paul_gazzillo

THE LINUX FOUNDATION

the linux kernel has a highly active codebase

~30k mailing list messages per month

~6k commits per month, 100s per day

e.g., ~13k commits between v5.12 and v5.13

used in billions of devices

all these code changes need testing

Most active 5.12 bug reporters

kernel test robot	184	16.1%
Syzbot	111	9.7%
Abaci Robot	107	9.4%
Dan Carpenter	44	3.9%
Hulk Robot	41	3.6%
Stephen Rothwell	28	2.5%
Randy Dunlap	19	1.7%
Kent Overstreet	12	1.1%
Guenter Roeck	11	1.0%
TOTE Robot	11	1.0%
Colin Ian King	9	0.8%
Andrii Nakryiko	8	0.7%
Juan Vazquez	7	0.6%
Arnd Bergmann	6	0.5%

https://lwn.net/Articles/853039/

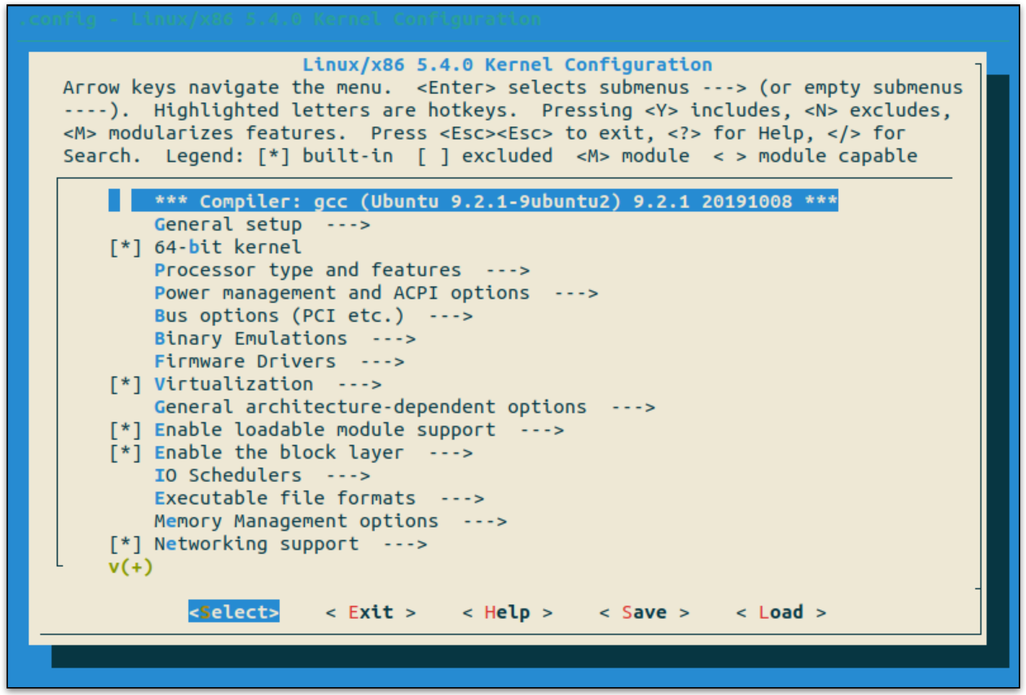
intel 0-day kernel test robot

- suite of tools: compile, boot, performance, etc.
- runs on new commits in linux-next
- continuously runs suite of tools

syzbot

- syzkaller system call fuzz tester
- continuously tests the kernel
- runs on linux-next, other versions

need to pick a .config before building and testing



.config selection with menuconfig

- if .config doesn't build patch, can't test it
- the .config file decides what code to build
  - ~15k configuration options
- even compile-time errors get committed!

example of bug missed due to wrong .config

- 2016: memory leak bug introduced
  - only possible under #ifdef CONFIG_MEDIA_CONTROLLER_DVB
- 2020-05: syzkaller fuzzer capable of finding it
  - but CONFIG_MEDIA_CONTROLLER_DVB is disabled in its mem checking config
- 2020-08: sykaller regenerates configs
  - happens to include CONFIG_MEDIA_CONTROLLER_DVB
- 2020-11: syzkaller finds the bug
- 2020-12: bug is fixed and backported to linux-stable

hard to make a .config to build a specific commit

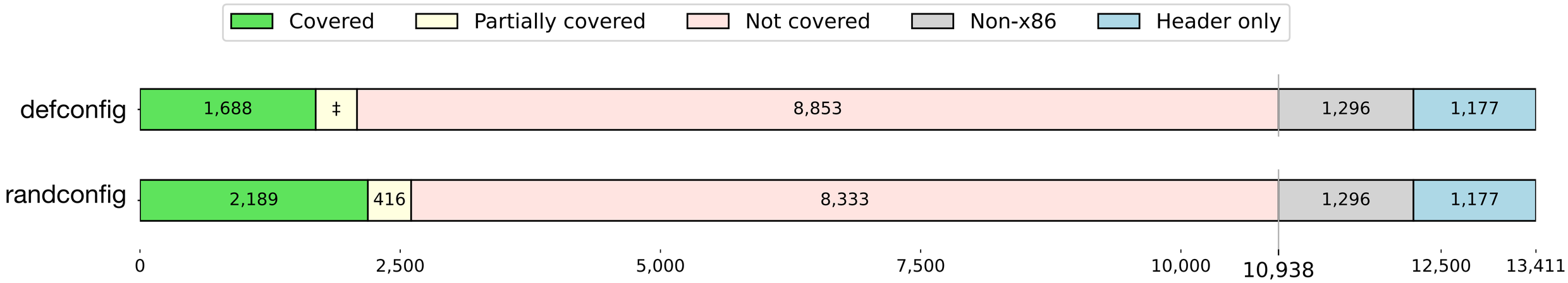
syzbot uses a set of hand-crafted configs

- runtime environment, architecture
- subsystems under test
- specific to tool
- kept update with changes to defaults

intel 0-day kernel test robot uses built-in configs

- all the many defconfigs
- allnoconfig, allyesconfig, allmodconfig
- hundreds of randconfig

unlikely to build a commit with typical .config files



number of commits from v5.13 covered when building for x86
with defconfig or one randconfig

why not just use allyesconfig or allmodconfig?

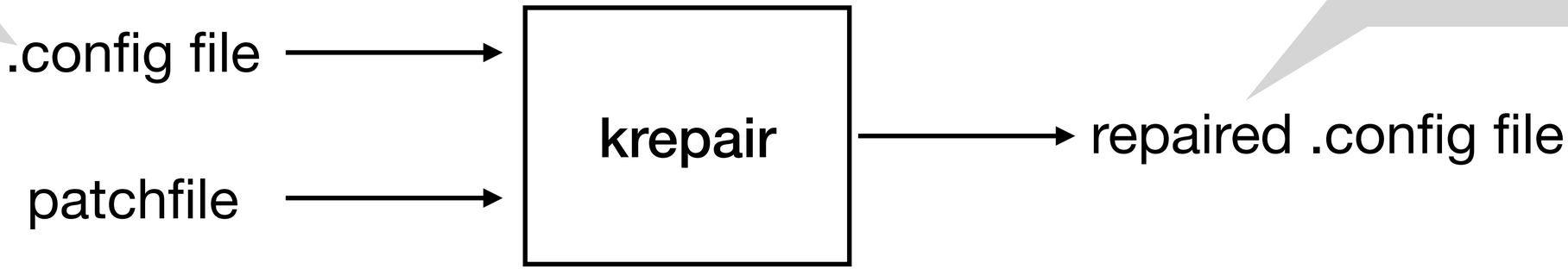
- good for compile-testing, builds most code
  - still doesn’t include all due to mutually-exclusive options, multiple architectures
- not ideal for run-time testing
  - slow to build (3 hours vs. 20 minutes for defconfig)
  - may not be bootable
  - may be too large for testing resource-constrained devices
- can’t do representative performance testing

key problem: lots of ways to select .config files for testing
but no guarantee that committed changes get built

our solution:

doesn't build patchfile

builds patchfile after krepair's fixes



benefits

- fully automated
- agnostic to how testers pick their .config files
- retains most original settings
- keeps build size around the same as original

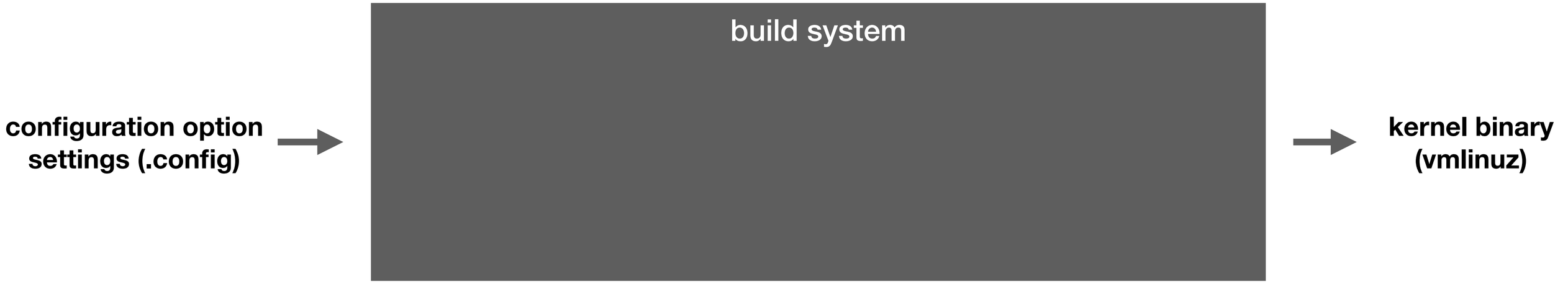
why is it so hard to pick a config file?

THE LINUX FOUNDATION

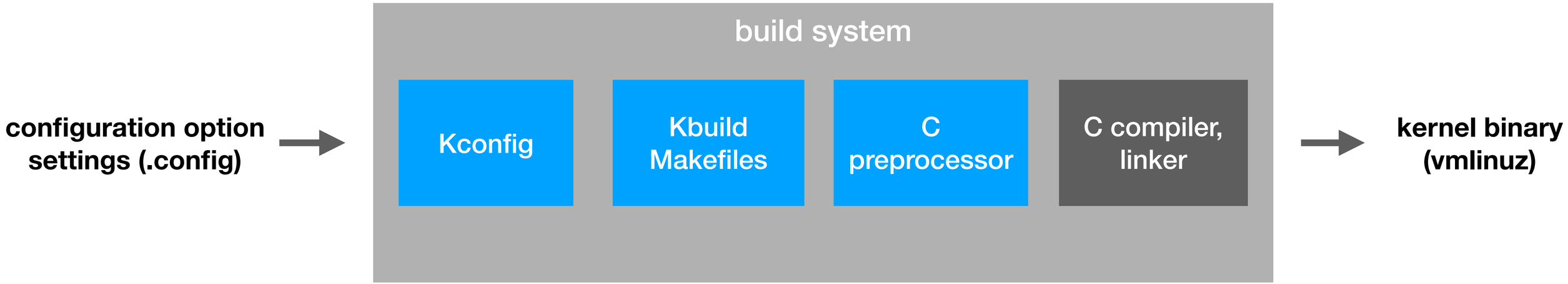
build system controls code inclusion/exclusion

- 100s of thousands of lines of build code
- complex dependencies between options

build system turns .config file into binaries



let’s look at the phases of build process



the build system components that select code

patches add and remove lines of code

```
--- a/drivers/irqchip/irq-gic.c
+++ b/drivers/irqchip/irq-gic.c
@@ -127,35 +124,27
 #ifdef CONFIG_GIC_NON_BANKED
-static void *gic_get_common_base(union gic_base *base)
+static void enable_frankengic(void)
 {
- return base->common_base;
+ static_branch_enable(&frankengic_key);
 }
 #else
-#define gic_set_base_accessor(d, f)
+#define enable_frankengic() do   while(0)
 #endif
@@ -1165,7 +1149,7
- gic_set_base_accessor(gic, gic_get_percpu_base);
+ enable_frankengic();
```

- #ifdef controls line inclusion
- CONFIG_GIC_NON_BANKED needs to be enabled/disabled
- this patch modifies irq-gic.c

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=8594c3b85171b6f68e34e07b533ec2f1bf7fb065

makefiles control file inclusion

```
1 // from drivers/irqchip/Makefile
2 obj-$(CONFIG_ARM_GIC)   += irq-gic.o
```

- irq-gic.o inclusion is controlled by CONFIG_ARM_GIC
- .config must include both makefile and ifdef options

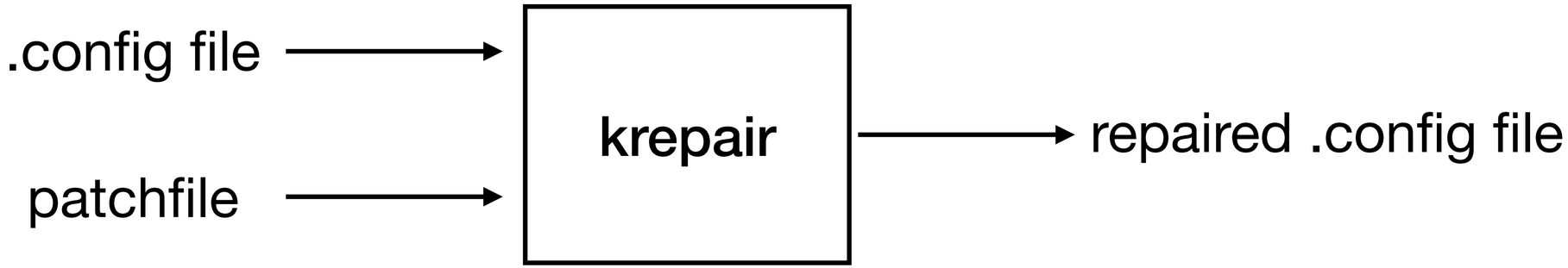
kconfig encodes dependencies between options

```
1  config ARM_GIC
2      bool
3  config ARM_GIC_PM
4      bool
5      depends on PM
6      select ARM_GIC
7  config GIC_NON_BANKED
8      bool
9
10 // from kernel/power/Kconfig
11 config PM
12     bool "Device power management core functionality"
13
14 // from arch/arm/mach-exynos/Kconfig
15 if ARCH_EXYNOS
16 config ARCH_EXYNOS4
17     bool "Samsung Exynos4"
18     default y
19     select GIC_NON_BANKED
20 endif
```

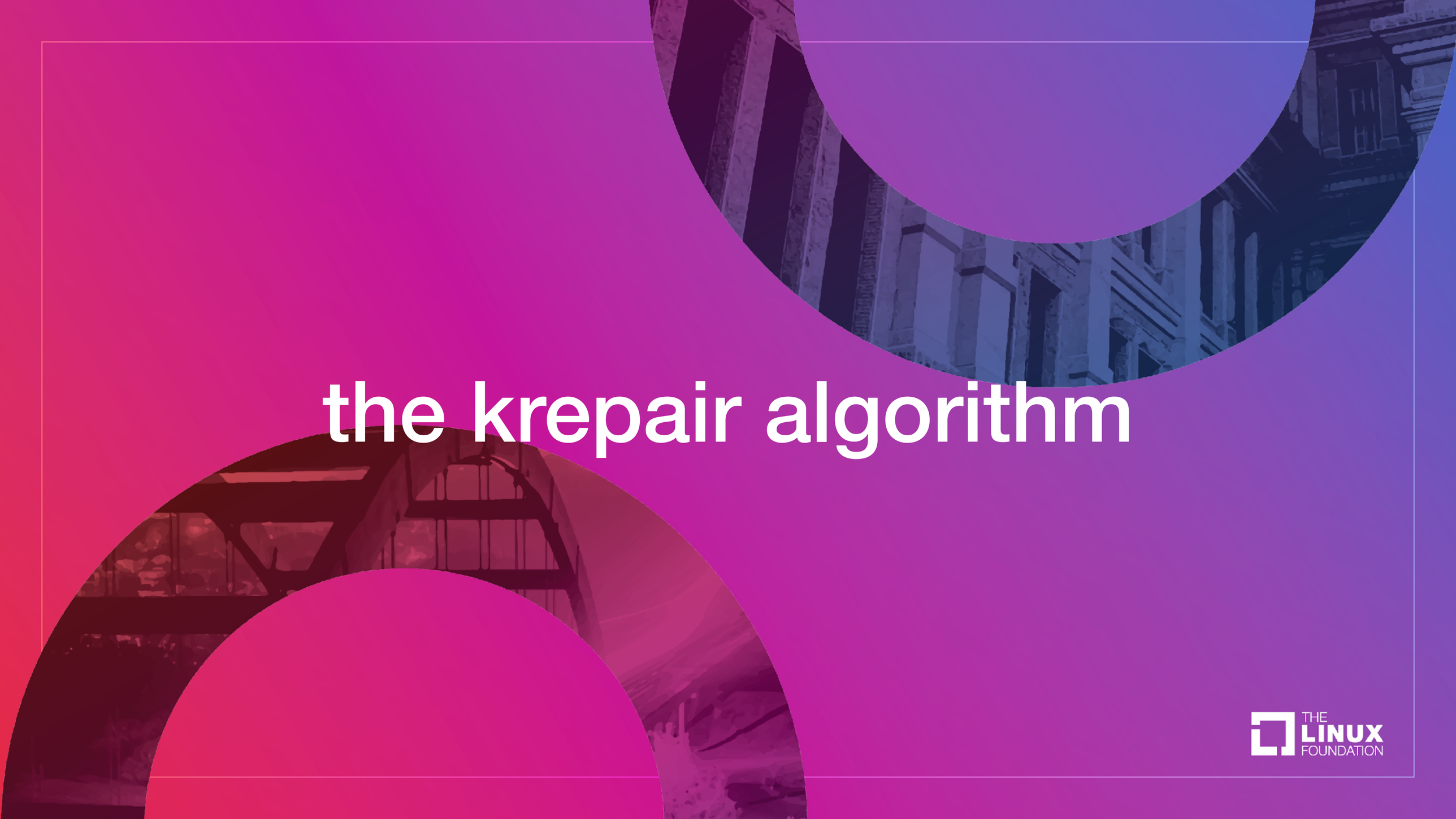
- GIC_NON_BANKED and ARM_GIC are only enabled by other options
- those other options can only be enabled when dependencies are met

hard just to figure out what config options need to be set so that your .config file covers a commit

krepair in three steps



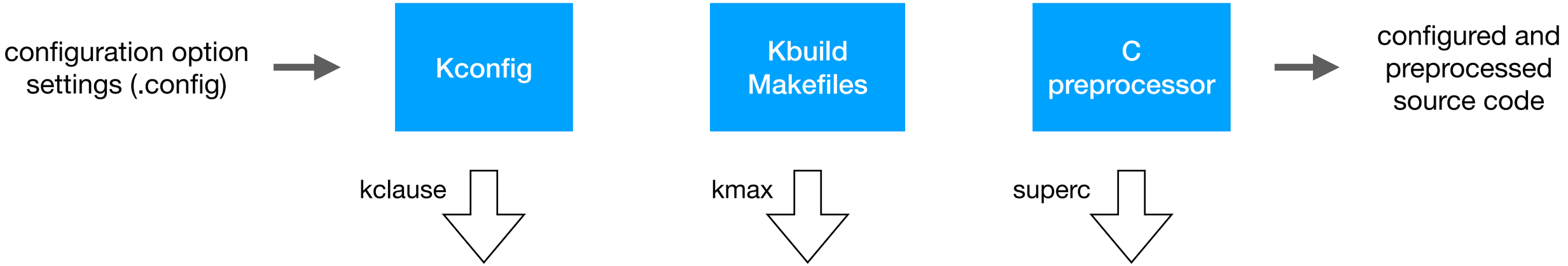
1. figures out requirements on config options that control the lines of code
2. removes options that are preventing patch from being built
3. adds back options that satisfy the requirements to build the lines of code



the krepair algorithm

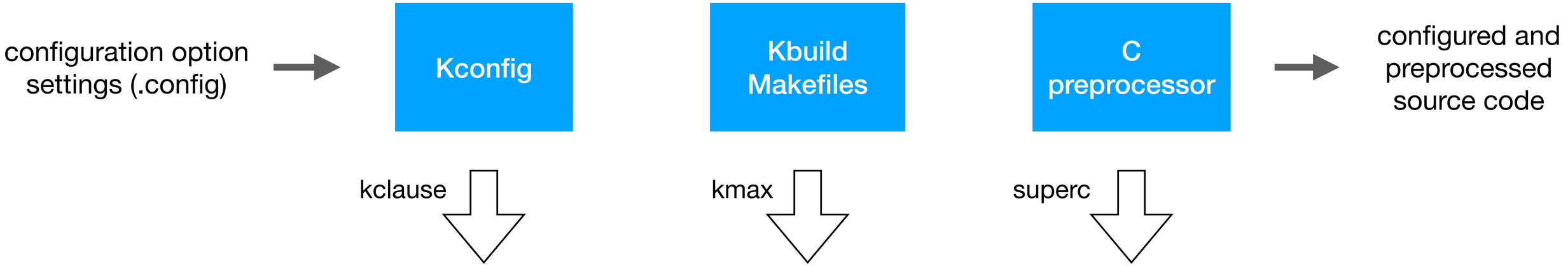
THE LINUX FOUNDATION

we convert build system code into constraints



logical constraints on .config files for each file:line

we convert build system code into constraints

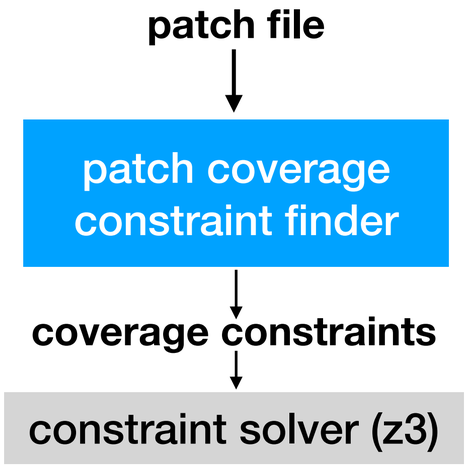


logical constraints on .config files for each file:line

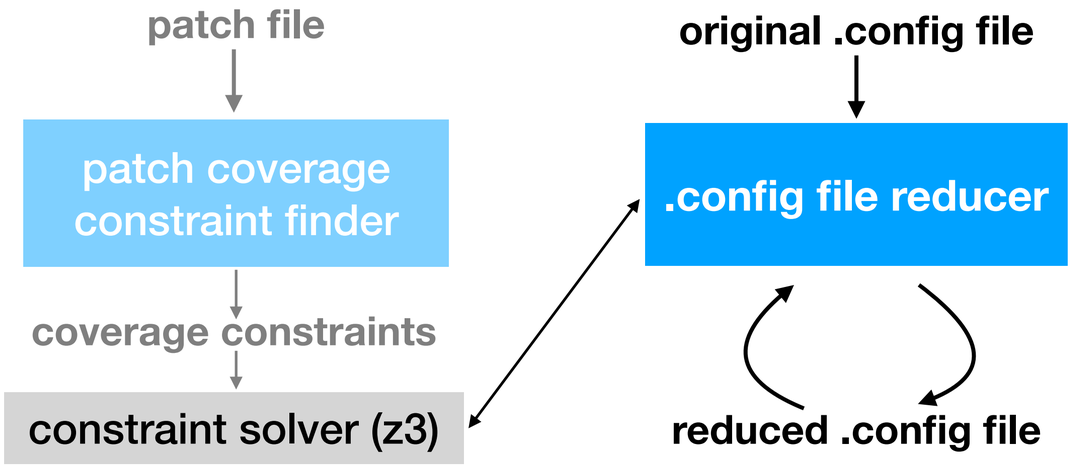
.config file constraints

source code locations

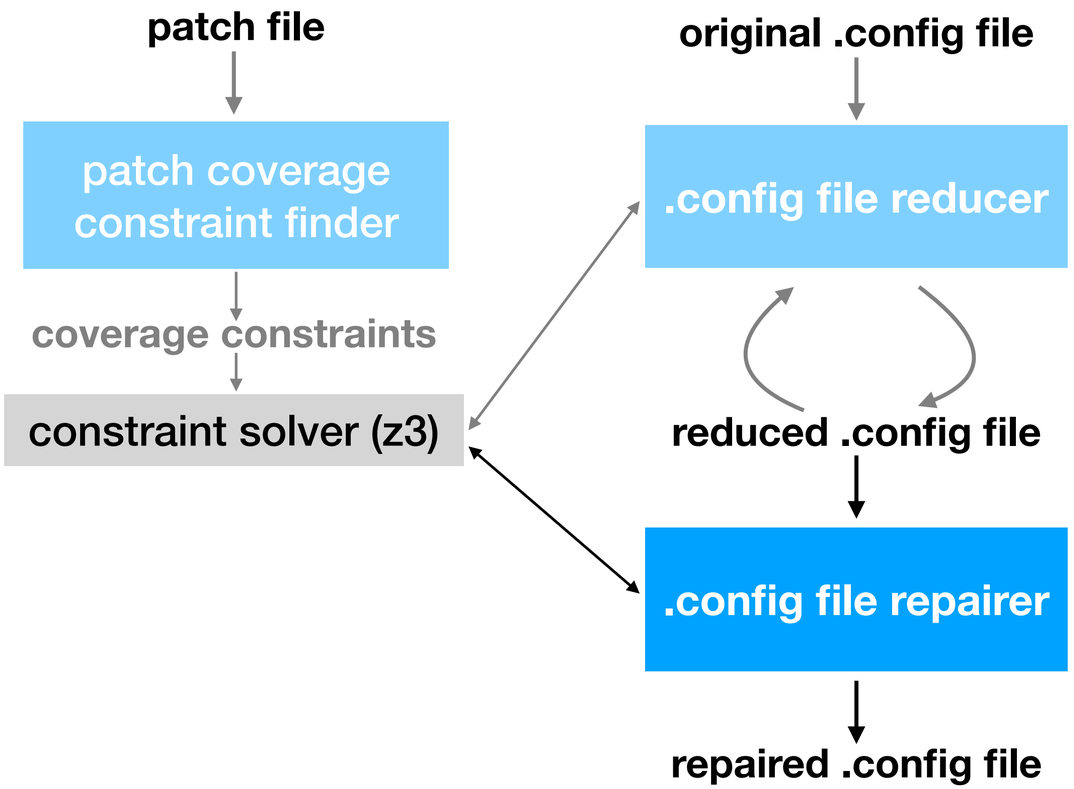
1) figure out .config constraints for covering the patch



2) remove options preventing the patch from building



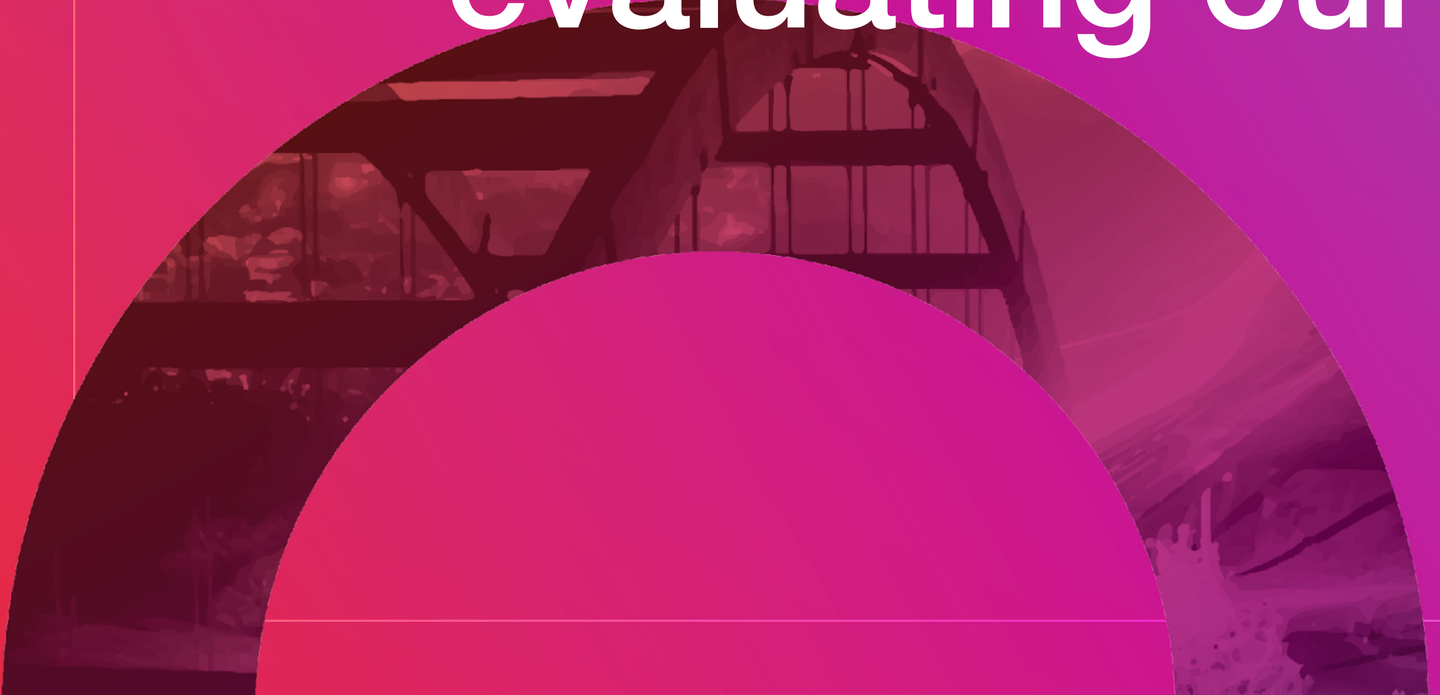
3) add back settings that satisfy coverage constraints



demo

THE LINUX FOUNDATION

evaluating our approach



THE LINUX FOUNDATION

experimental setup

- all patches from v5.12 -> v5.13
  - x86-only configs
- 13,411 patches
  - 1,296 are non-x86
  - 1,177 are header only (nothing to build, future work)
  - 10,938 patches to test on

experiment

- for each patch
  - generate and repair allno, defconfig, one randconfig
- check patch coverage before and after krepair
- compare to coverage of allyesconfig
  - patch coverage
  - number of differences after repair vs. using allyesconfig

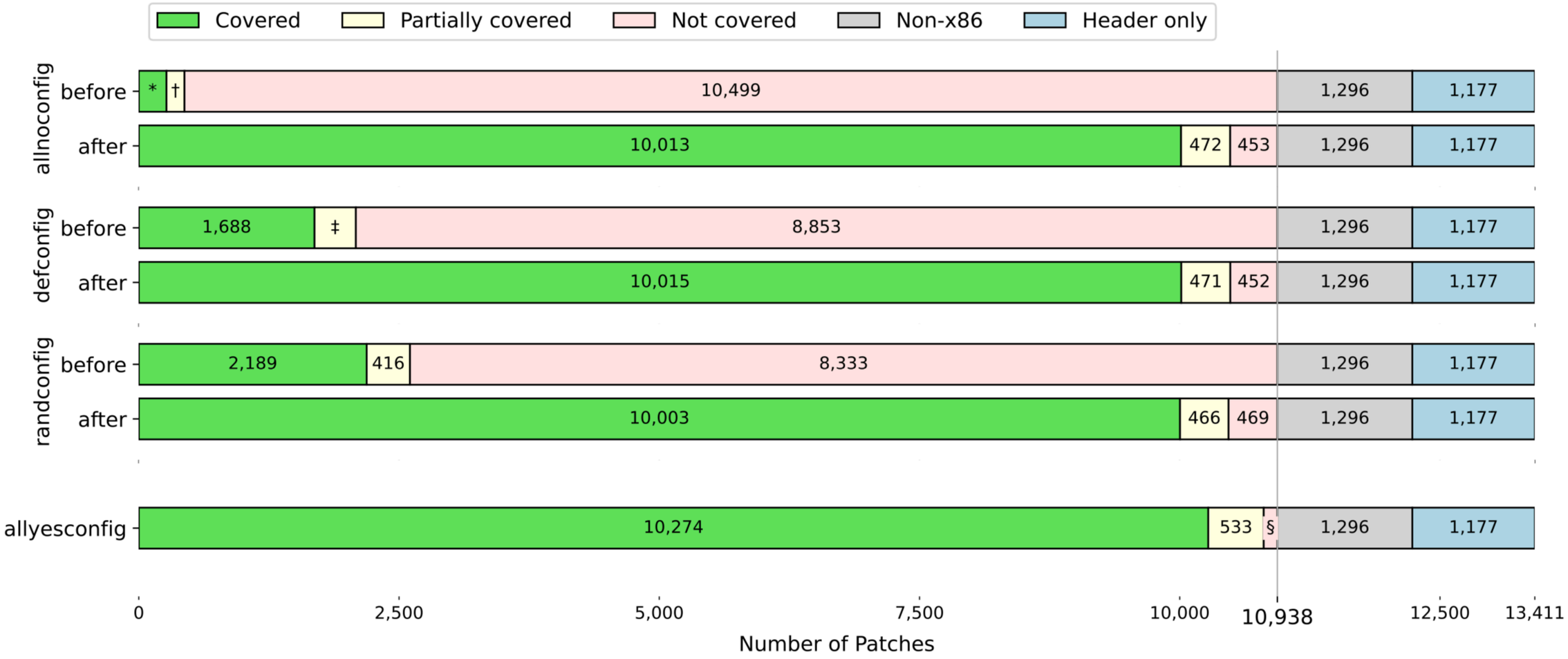


Figure 4: Patch coverage before and after repair for several common configuration files. *266 †173 ‡397 §131.

Comparisons	Min	Median	99th	Max
allnoconfig				
repaired	<0.1%	0.7%	2.5%	5.8%
allyesconfig	77.5%	78.6%	79.1%	79.2%
defconfig				
repaired	0.4%	0.5%	3.0%	4.4%
allyesconfig	70.6%	71.7%	72.5%	72.5%
randconfig				
repaired	0.1%	6.7%	26.8%	35.7%
allyesconfig	46.0%	71.4%	80.6%	84.9%

Table 1: Percentile distribution of repair size and allyesconfig size differences for each configuration file across all patches. Size differences are the percentage of configuration options that have changed out of all 15,446 possible options.

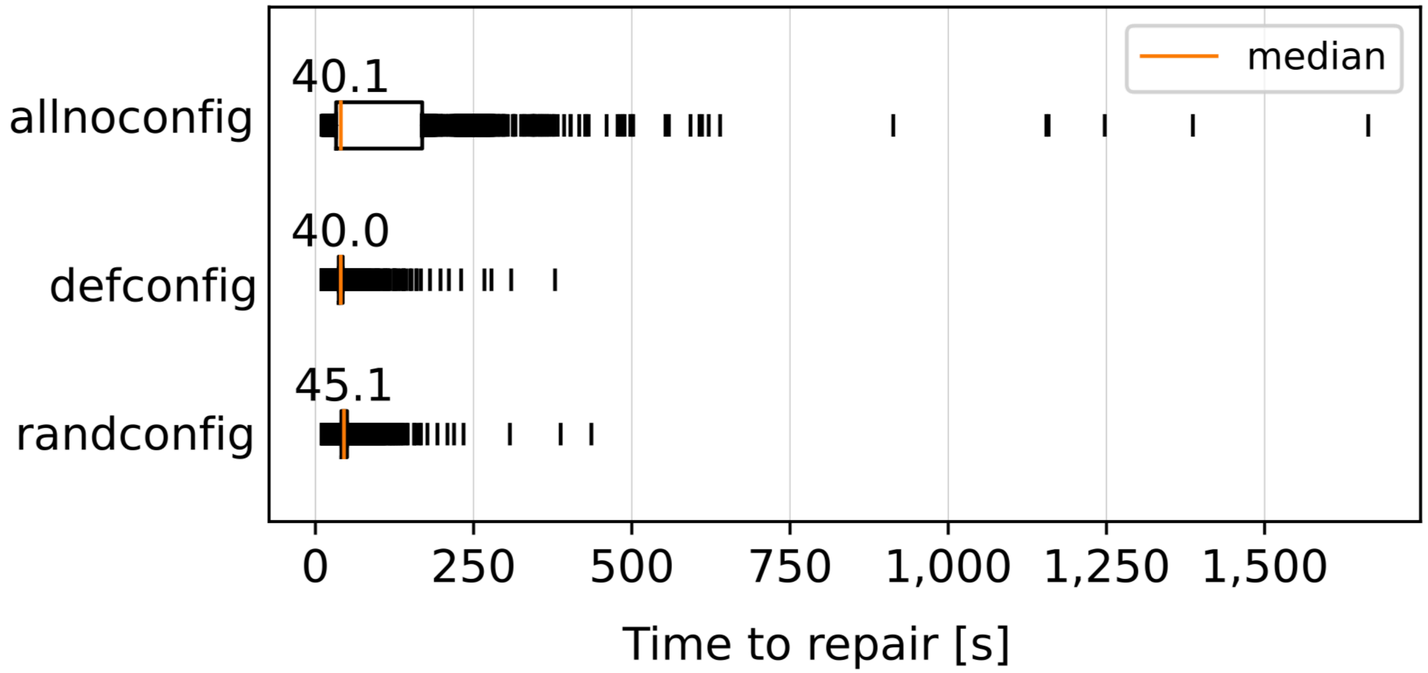


Figure 5: Distribution of krepair running times. The red line is the median (second quartile), the box is the interquartile range (too small to be visible for `defconfig` and `randconfig`), and the vertical lines are the values below and above the first and third quartiles respectively.

summary

- krepaired .config files achieve similar coverage to allyesconfig
  - 5x-37x more coverage than the original config files
- but retain most settings from the original .config file
  - 1% to 7% change on average depending on config file, compared to 71%-79% for allyesconfig
- and krepair is fast
  - 4 minutes or less on 99% of repairs

conclusion

THE LINUX FOUNDATION

researchers on the krepair project

Necip Fazıl Yıldıran (UCF)

Jeho Oh (UT Austin)

Julian Braha (UCF)

Julia Lawall (INRIA/Lip6)

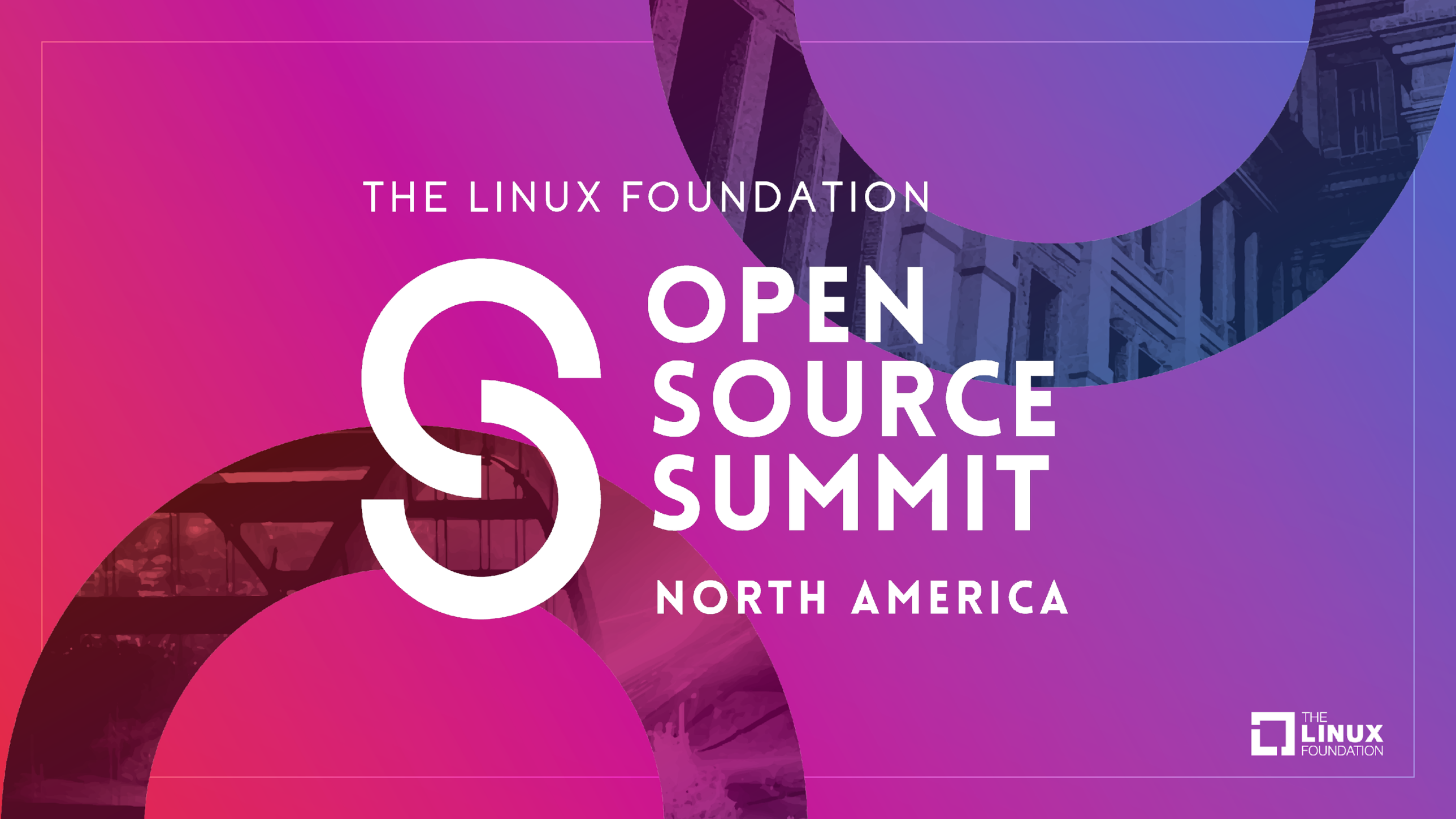
Paul Gazzillo (UCF)

conclusion

- automated testers need to pick configs to boot/run/test
- picking a config is hard; arbitrary configs don’t cover most patches
- krepair automatically edits config files with relatively small changes to ensure patch coverage

supported by NSF CCF-1941816 and CCF-1840934

try it out!
https://github.com/paulgazz/kmax


THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA
THE LINUX FOUNDATION