LinuxTesting.org

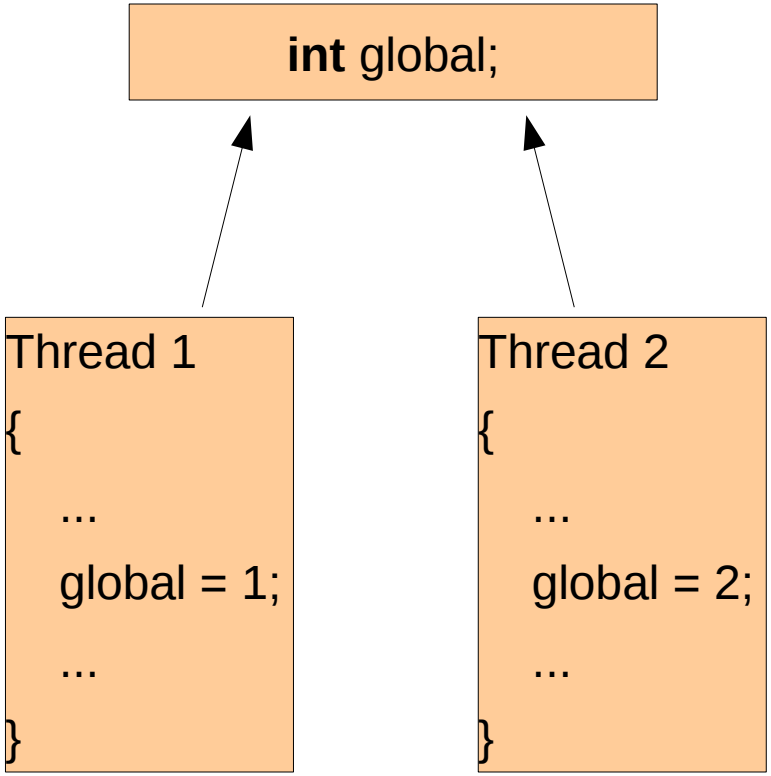
Predicate Abstraction Based Configurable Method for Data Race Detection in Linux Kernel

Pavel Andrianov, Vadim Mutilin, Alexey Khoroshilov

ISP RAS

Institute for System Programming of the Russian Academy of Sciences

Race Condition



A situation, in which simultaneous accesses to the same memory location take place from several threads, one of the accesses is write

Real Data Race

drivers/net/wireless/marvell/libertas/libertas.ko

disconnect:

```
…
kfree_skb(priv->currenttxskb);
priv->currenttxskb = NULL;
priv->tx_pending_len = 0;
...
```

transmit:

```
spin_lock(&priv->driver_lock, flags)
if (priv->currenttxskb == NULL)
    return;
…
priv->currenttxskb->protocol =
eth_type_trans(priv->currenttxskb,
      priv->dev);
netif_rx(priv->currenttxskb);
…
spin_unlock(&priv->driver_lock,
flags)
```

Commit

author	Pavel Andrianov <andrianov@ispras.ru>	2016-06-15 11:34:03 (GMT)
committer	Kalle Valo <kvalo@codeaurora.org>	2016-06-29 15:46:56 (GMT)
commit	f52b041aed77592862c97726b98d78e8dccd72c9 (patch)	
tree	18e3a9e1f0401363f9c70b817965e505f9710d7d	
parent	6edc119ed3b5e860535d49852f8cc8e5be95538d (diff)	

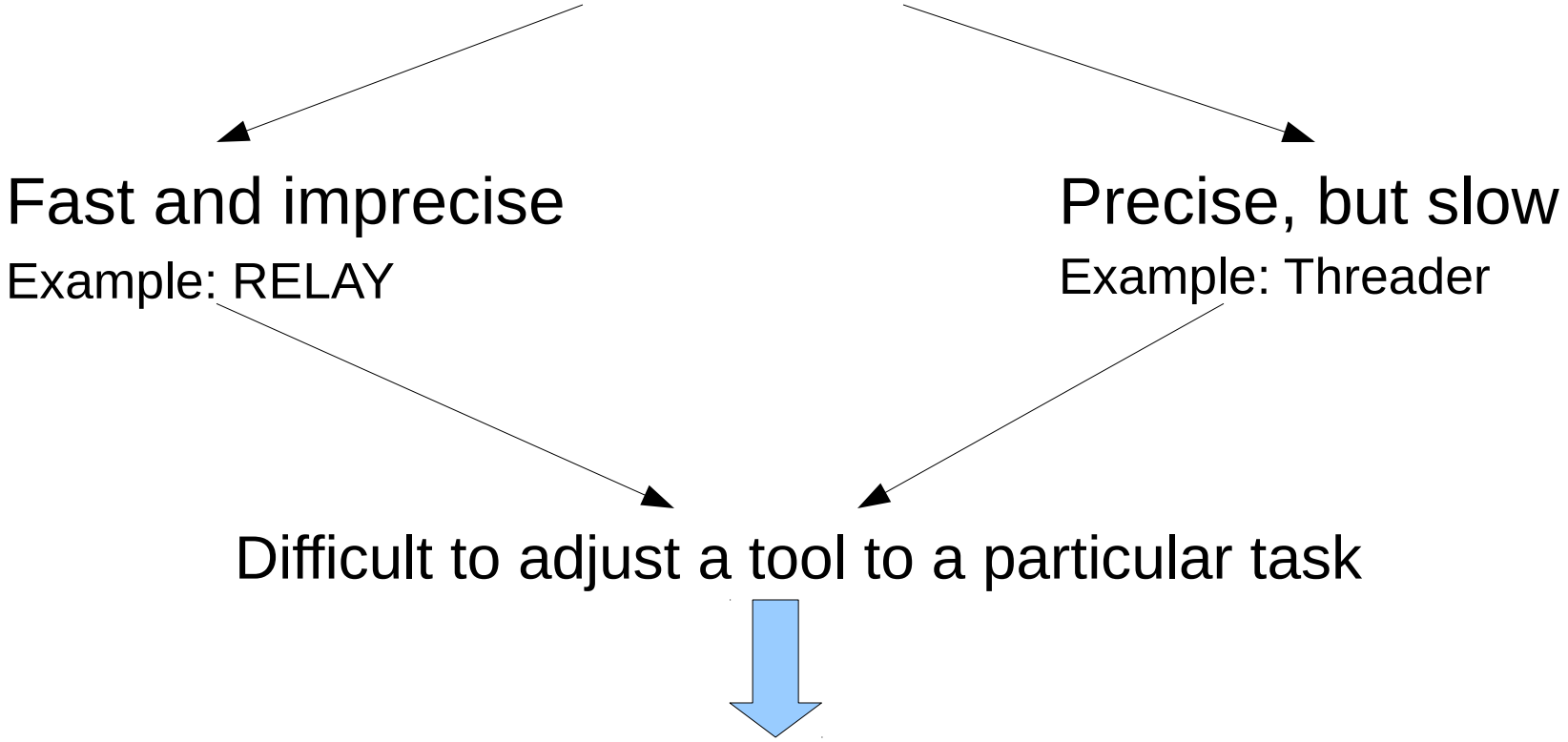
libertas: Add spinlock to avoid race condition

```
lbs_mac_event_disconnected may free priv->currenttxskb
while lbs_hard_start_xmit accesses to it.
The patch adds a spinlock for mutual exclusion.

Tested on OLPC XO-1 (usb8388) and XO-1.5 (sd8686) with v4.7-rc3.

Confirmed that lbs_mac_event_disconnected is being called on the
station when hostapd on access point is given SIGHUP.

Signed-off-by: Pavel <andrianov@ispras.ru>
Tested-by: James Cameron <quozl@laptop.org>
Acked-by: Vaishali Thakkar <vaishali.thakkar@oracle.com>
Signed-off-by: Kalle Valo <kvalo@codeaurora.org>
```

Motivation

- Concurrency bugs make up 20% of all across the file systems (*A Study of Linux File System Evolution*, FAST'13)
- Data race conditions make up 17% of all errors in the Linux kernel (*Analysis of typical faults in Linux operating system drivers,* Proceedings ISP RAN)

Other Tools


Fast and imprecise
Example: RELAY
Precise, but slow
Example: Threader
Difficult to adjust a tool to a particular task
Adjustable analysis?

Lockset Algorithm

Potential data race is a situation, when accesses to the same shared data occur with disjoint sets of locks from two parallel threads, one access is write.

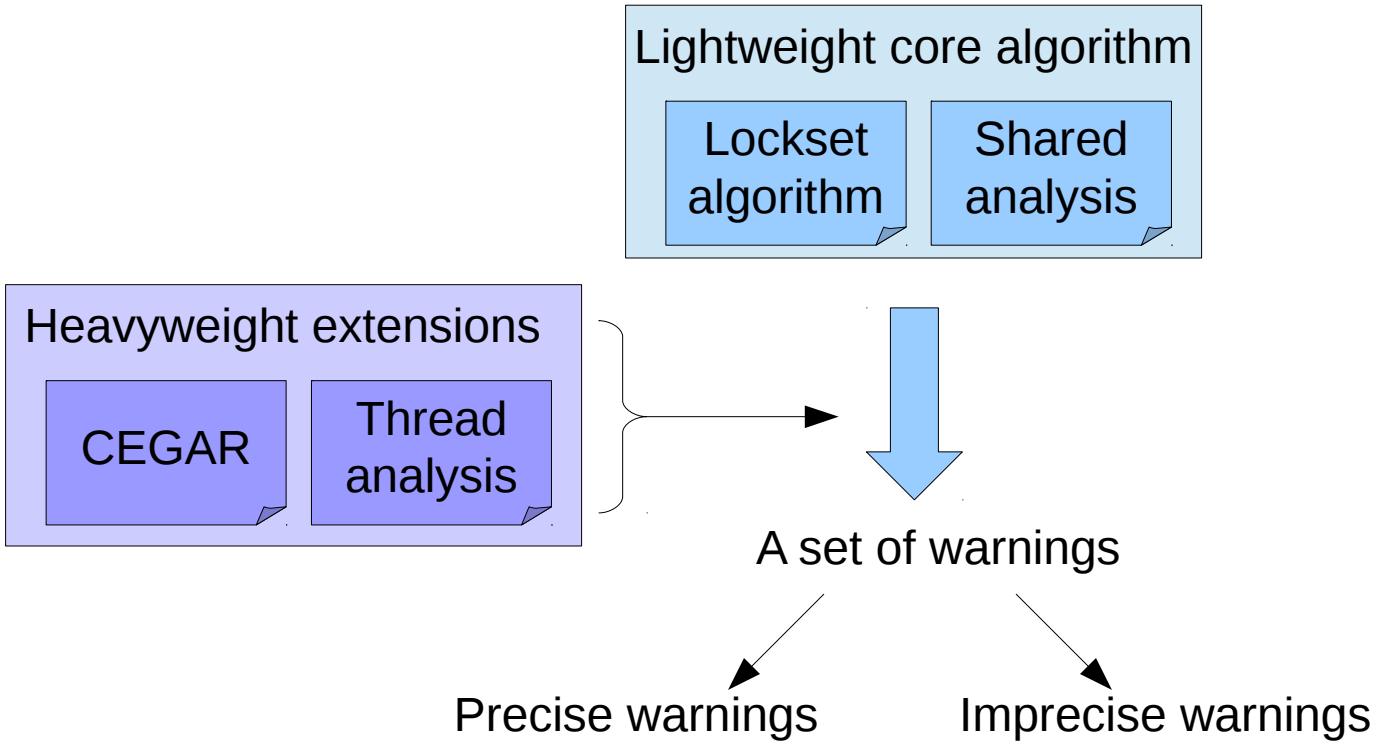
Potential Race Condition

```
...
*a = 1;
...
```

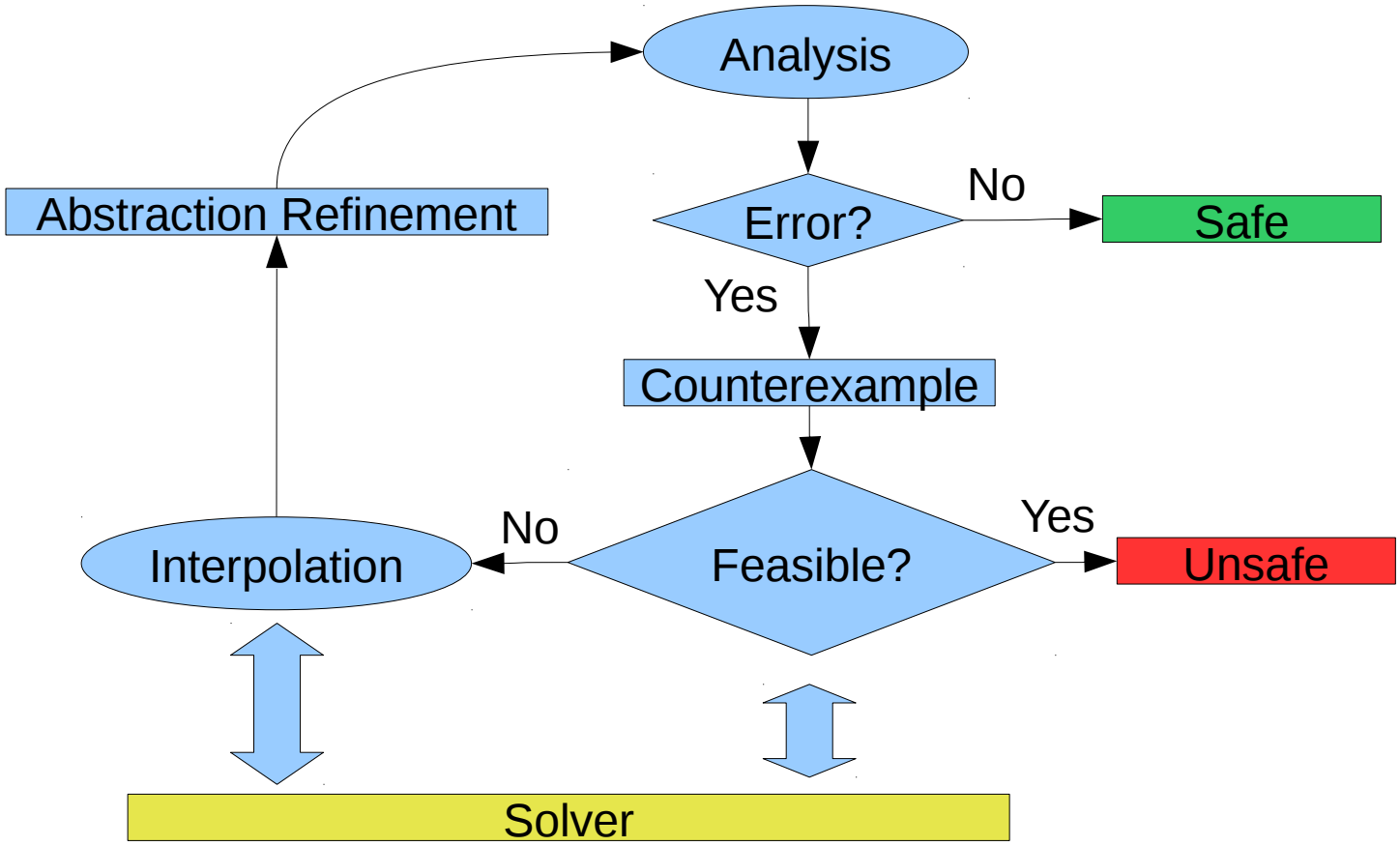
```
...
mutex_lock();
*a = 1;
mutex_unlock();
...
```

- A disjoint set of synchronization primitives
- The same shared data
- Accesses from different threads, which can be executed simultaneously
- Real (reachable) paths

Method overview



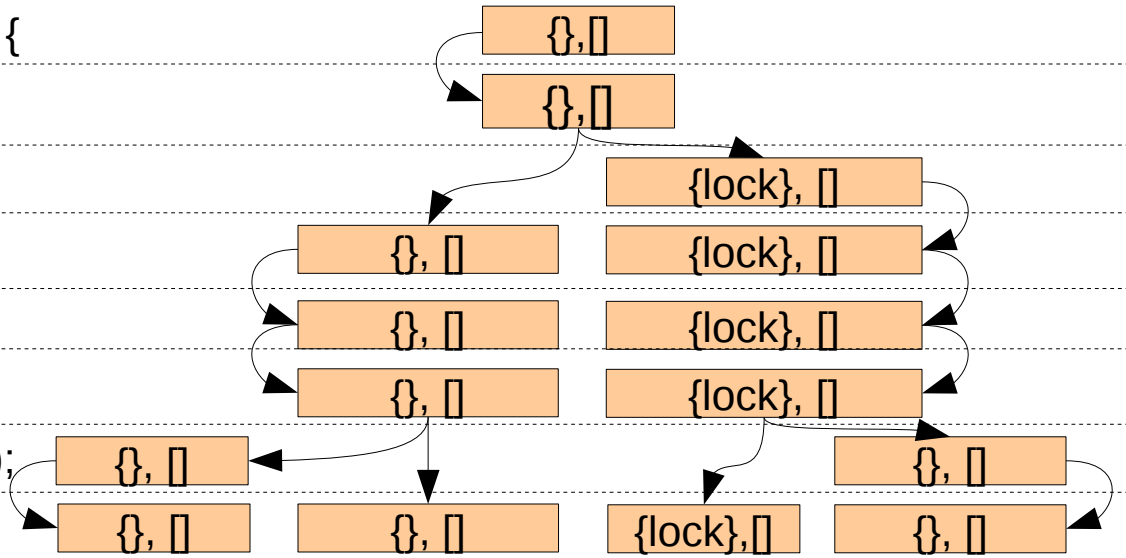
Counter Example Guided Abstraction Refinement



Reachability analysis

based on predicate abstraction

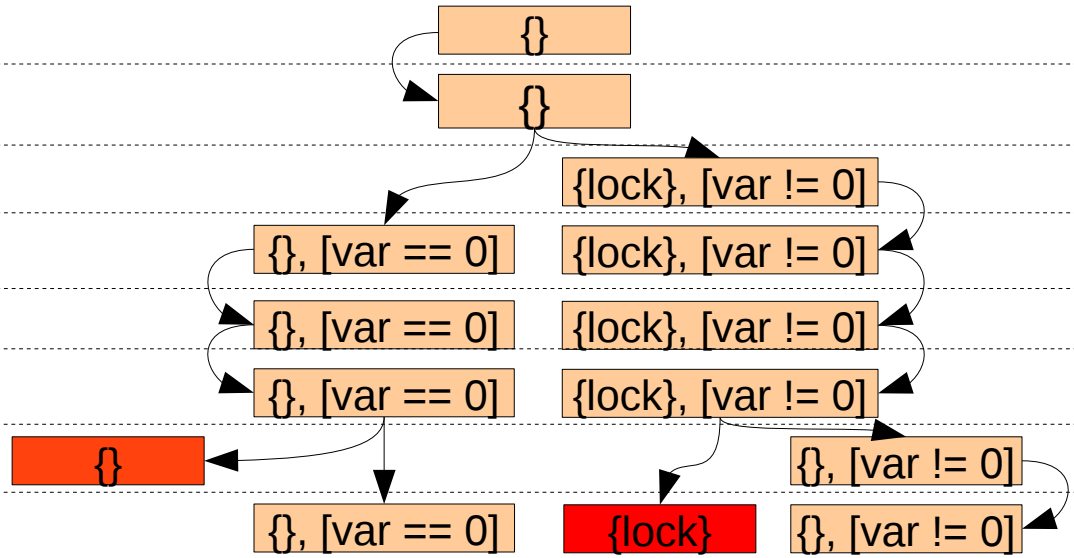
```
int global;
int func(int var) {
        if (var) {
            lock();
        }
        global++;
        if (var) {
            unlock();
        }
}
```



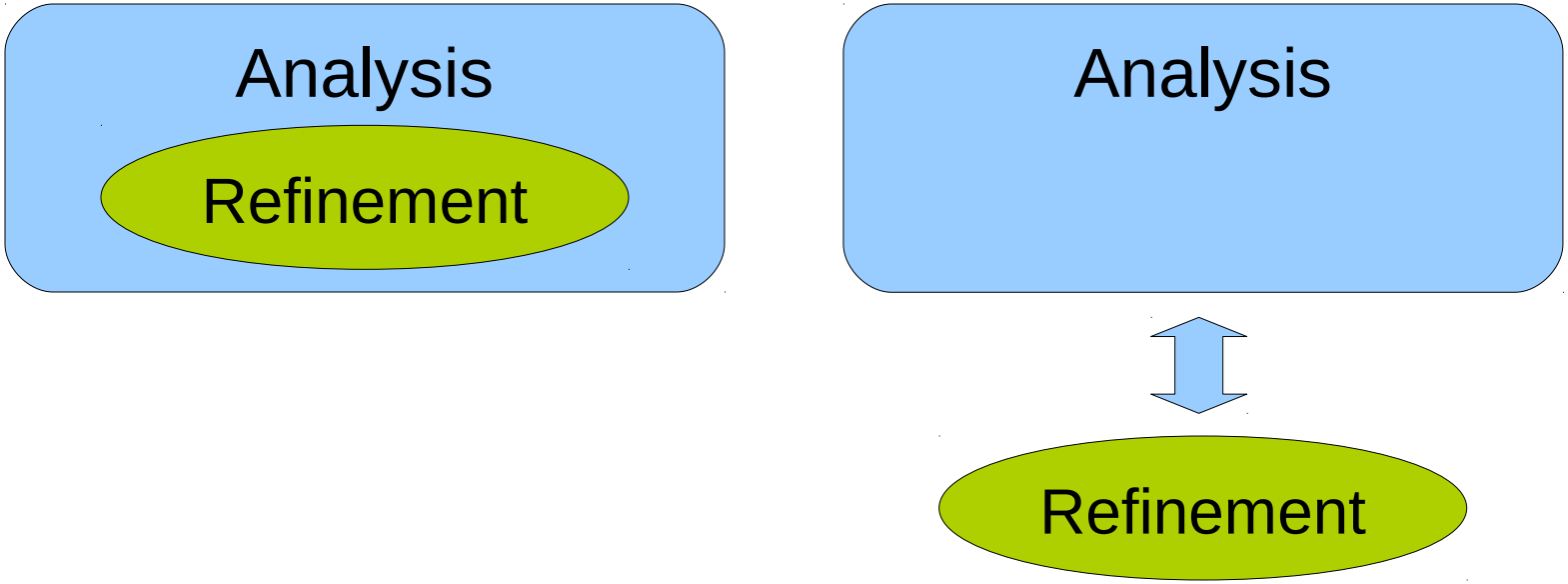
Reachability analysis

based on predicate abstraction

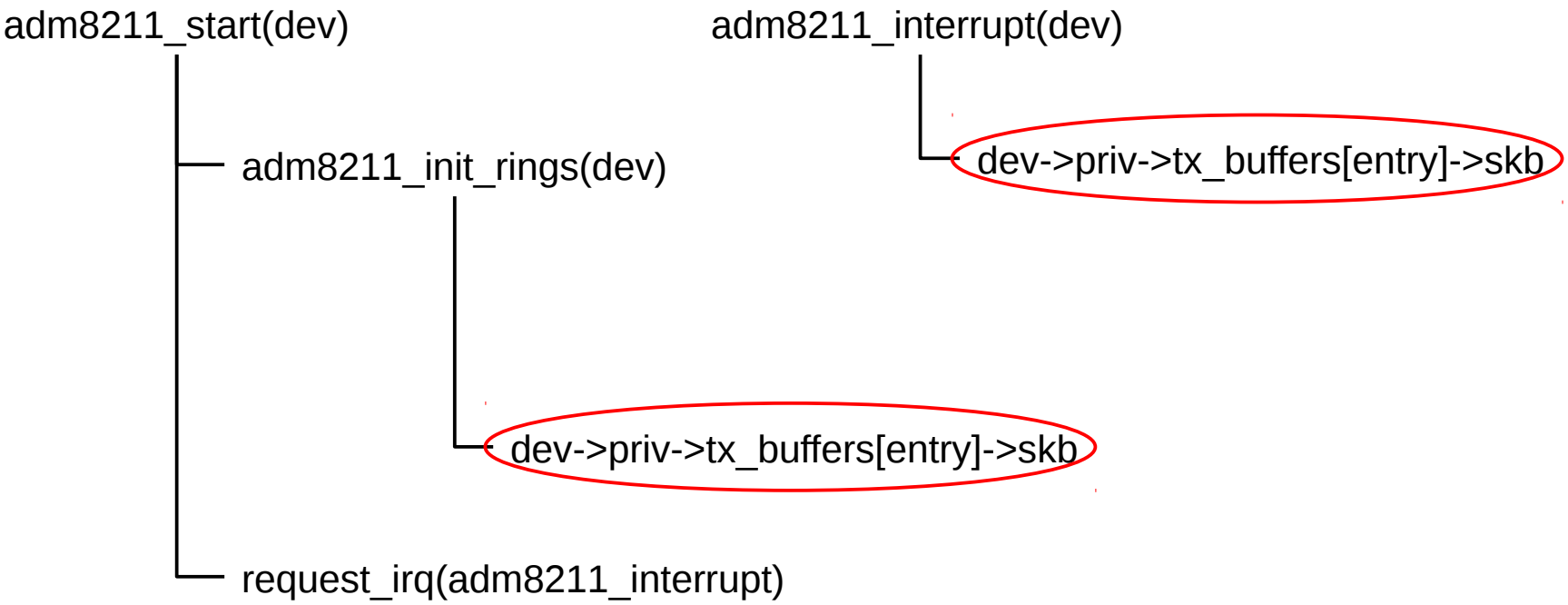
```
int global;
int func(int var) {
        if (var) {
            lock();
        }
        global++;
        if (var) {
            unlock();
        }
}
```



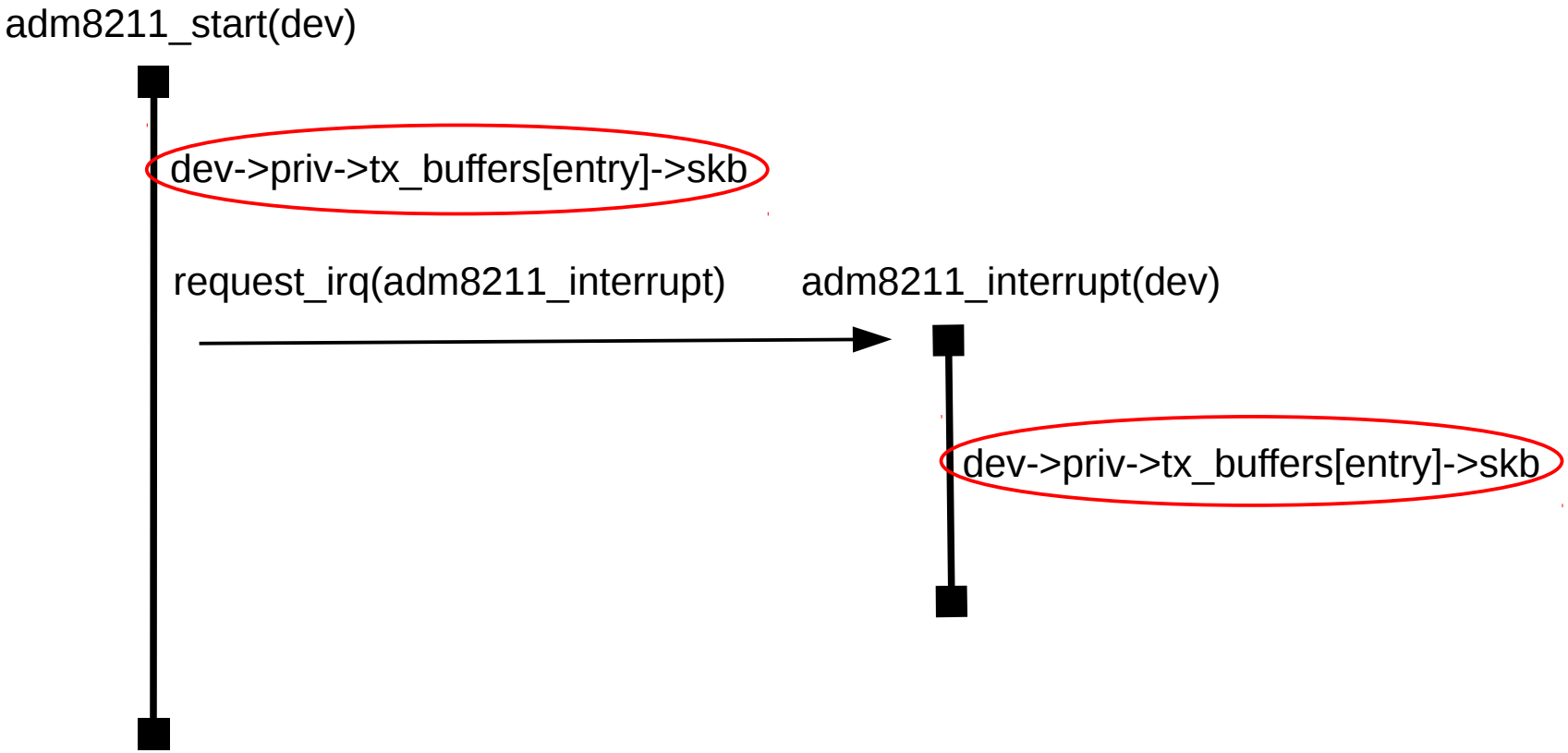
Two Ways of Refinement



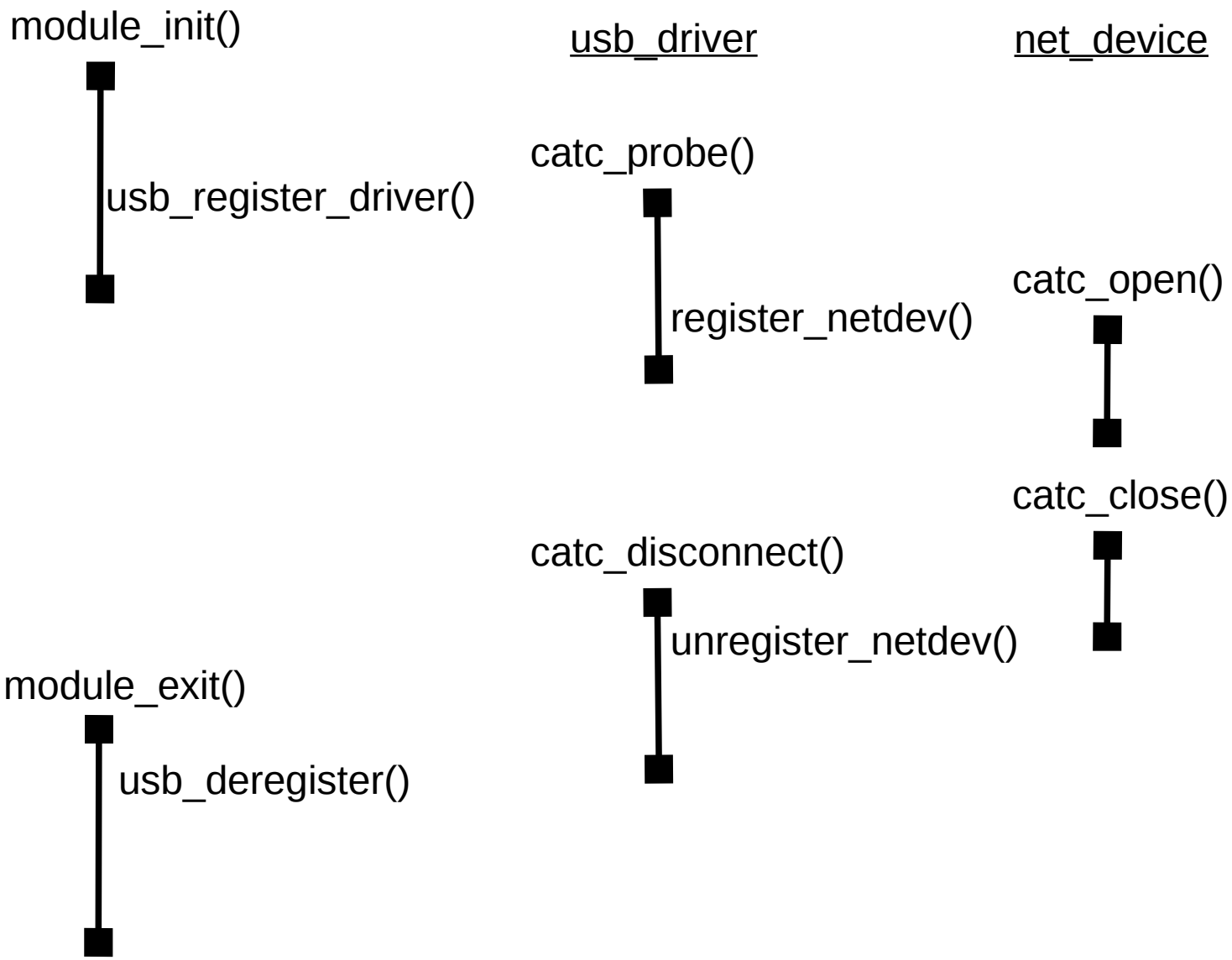
Example of False Alarm



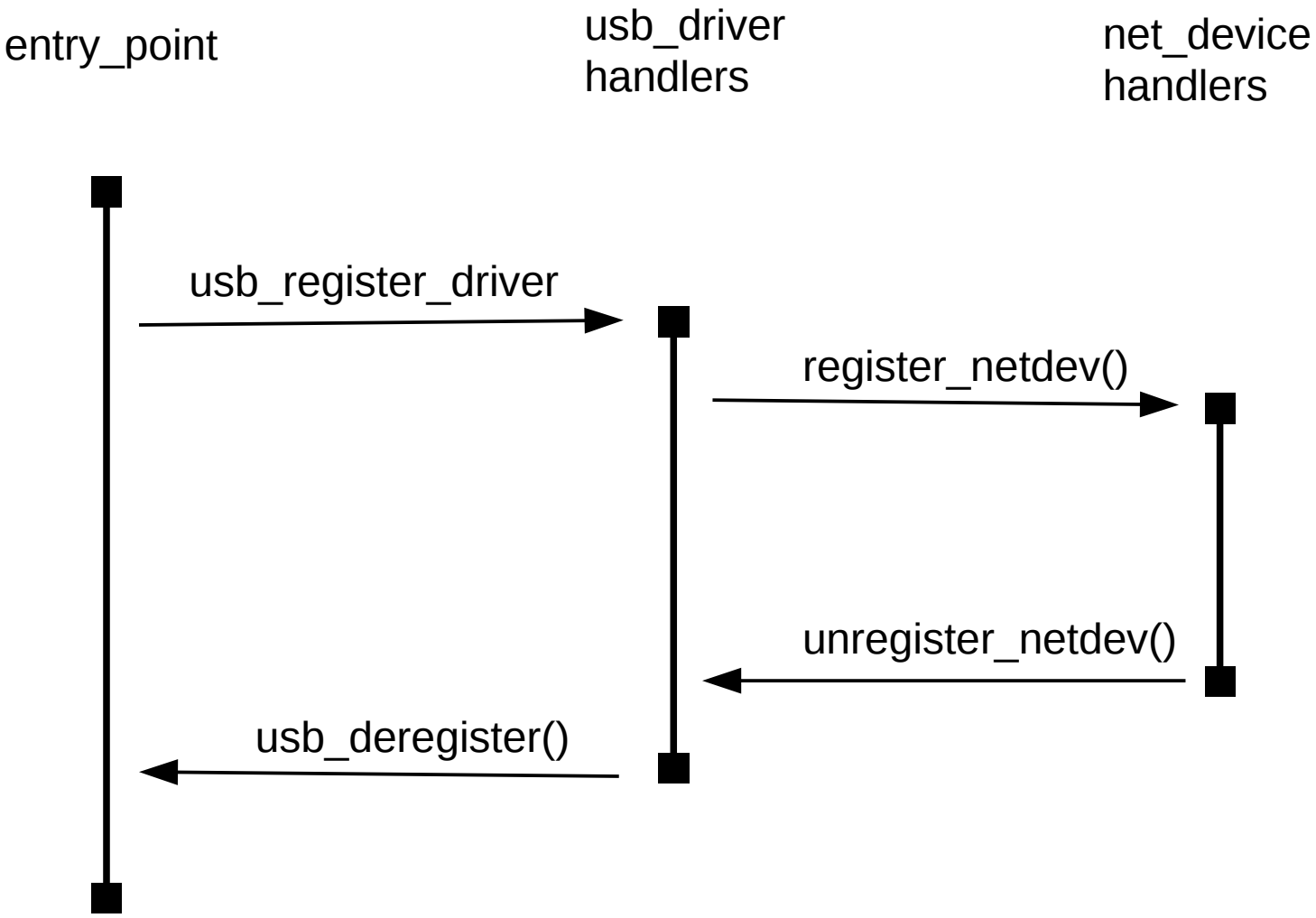
Example of False Alarm



Example of Linux Driver

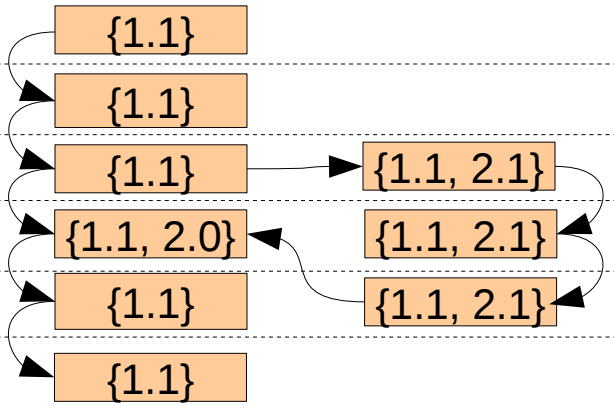


Example of Model



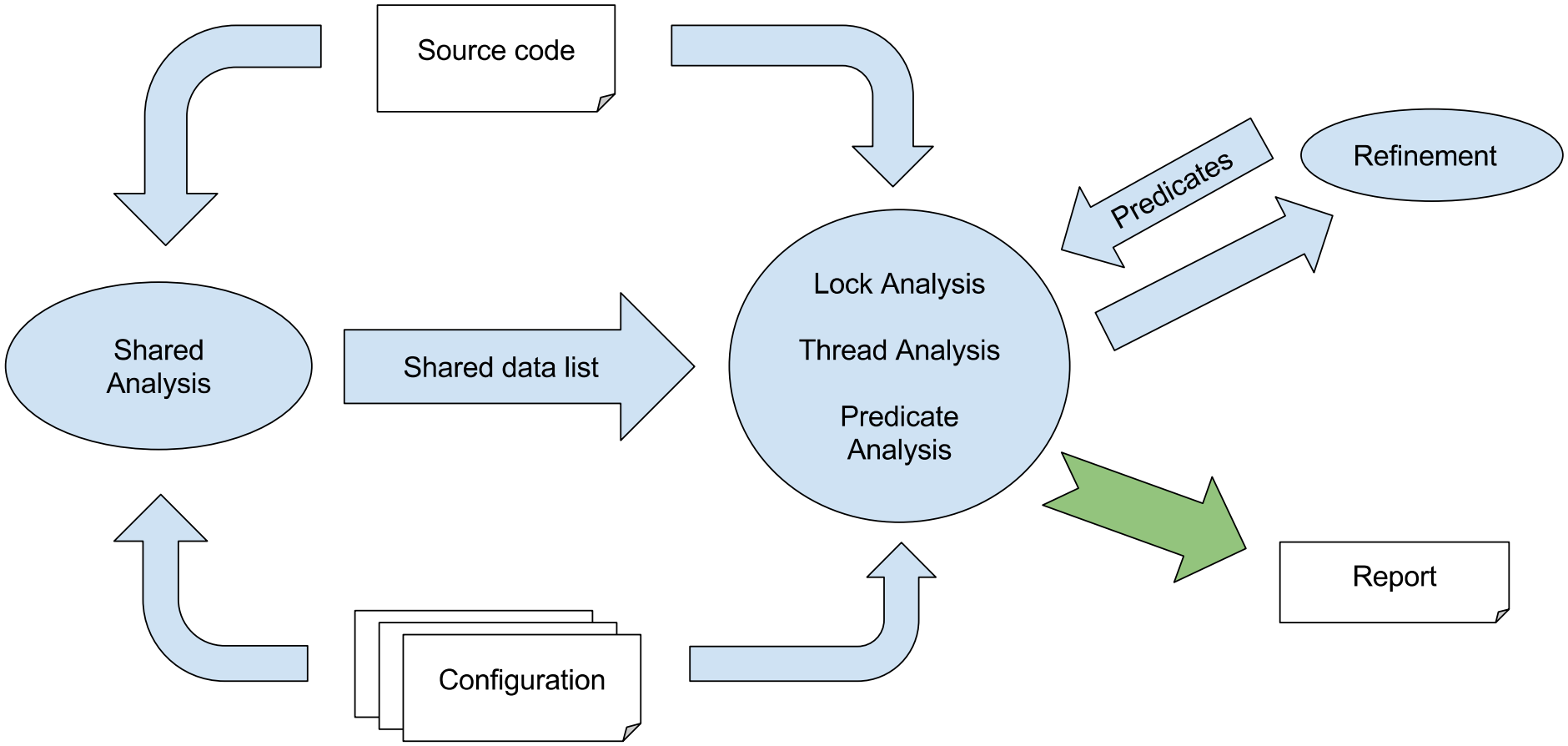
Thread Analysis

```
int global;
Int start() {
    global = 0;
    pthread_create(&thread, .., worker, ..);
    pthread_join(&thread);
    result = global;
}
```




```
int worker() {
        global++;
}
```

Method Overview



Results

113 modules of OS Linux 4.5-rc1 subsystem drivers/net/wireless/

	Unsafes	Unknowns	Safes	Time, h	Memory, Gb
+ Threads, + Refinement	5	61	51	3.2	8.1
- Threads, + Refinement	6	67	44	4.1	4.0
+ Threads, - Refinement	27	57	49	2.3	8.2
- Threads, - Refinement	186	54	43	2.1	3.5

2219 warnings at drivers/

- 2219 warnings = 270 unsafe drivers
  - 55% - imprecision of environment model
  - 10% - simple memory model
  - 10% - operations with lists
  - 10% - other inaccuracies in our analysis
  - 15% - true races
- 290 true warnings = 32 bugs

Conclusion

- Flexible adjustment of the balance between resources and accuracy
- Applicable to industry projects
- Real race conditions are found

Thank you!

Questions?