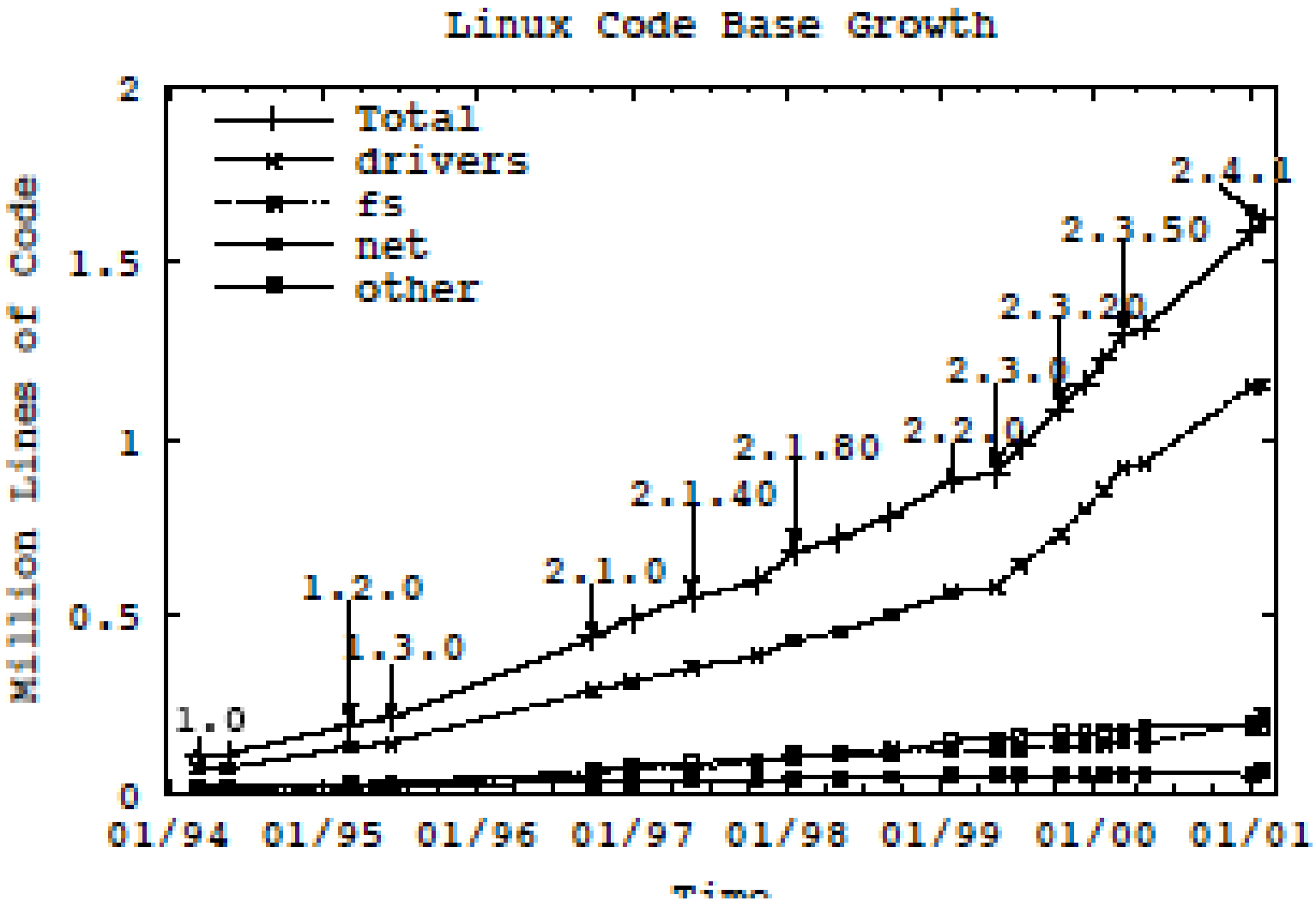
Outline

- An Empirical Study of Operating Systems Errors Andy Chou, Junfeng Yang, Benjamin Chelf, Seth Hallem, and Dawson Engler SOSP 2001
- Improving the Reliability of Commodity Operating Systems Michael M. Swift, Brian N. Bershad, Henry M. Levy, SOSP 2003

Summary

- Where are the errors
  - Driver code has three to seven times errors than kernel
- How are bugs distributed
- How long do bugs live
  - In Linux kernel, about 1.8 years
- How do bugs cluster?
  - Clusterings when programmer ignorance of interface or system rules combines with copy-and-paste
- Comparison with OpenBSD and Linux
  - OpenBSD has higher error rates

Size of linux tree

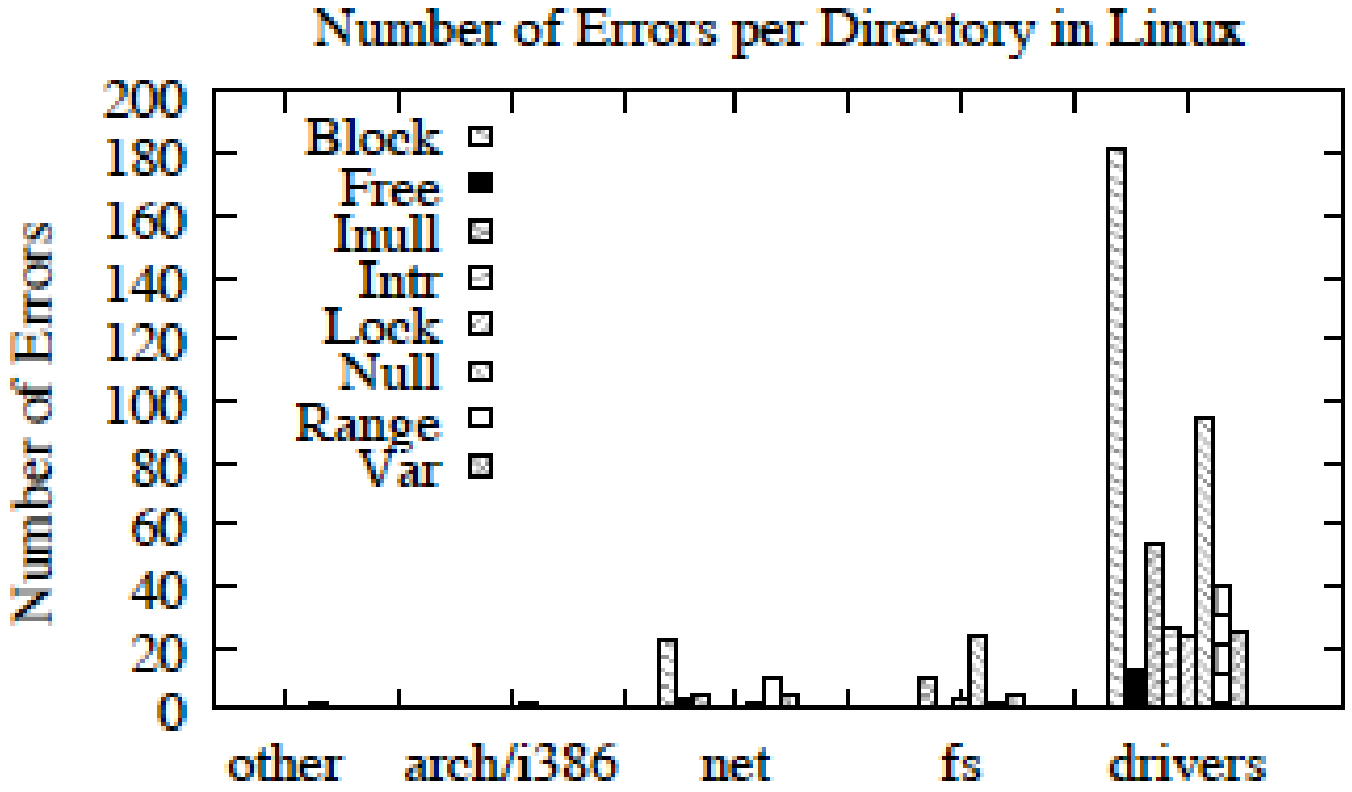


Errors checked

Check	Nbugs	Rule checked
`Block`	206 + 87	To avoid deadlock, do not call blocking functions with interrupts disabled or a spinlock
`Null`	124 + 267	Check potentially NULL pointers returned from routines.
`Var`	33 + 69	Do not allocate large stack variables ($> 1K$) on the fixed-size kernel stack.
`Inull`	69	Do not make inconsistent assumptions about whether a pointer is `NULL`.
`Range`	54	Always check bounds of array indices and loop bounds derived from user data.
`Lock`	26	Release acquired locks; do not double-acquire locks.
`Intr`	27	Restore disabled interrupts.
`Free`	17	Do not use freed memory.
`Float`	10 + 15	Do not use floating point in the kernel.
`Real`	10 + 1	Do not leak memory by updating pointers with potentially `NULL` realloc return values.
`Param`	7	Do not dereference user pointers.
`Size`	3	Allocate enough memory to hold the type for which you are allocating.

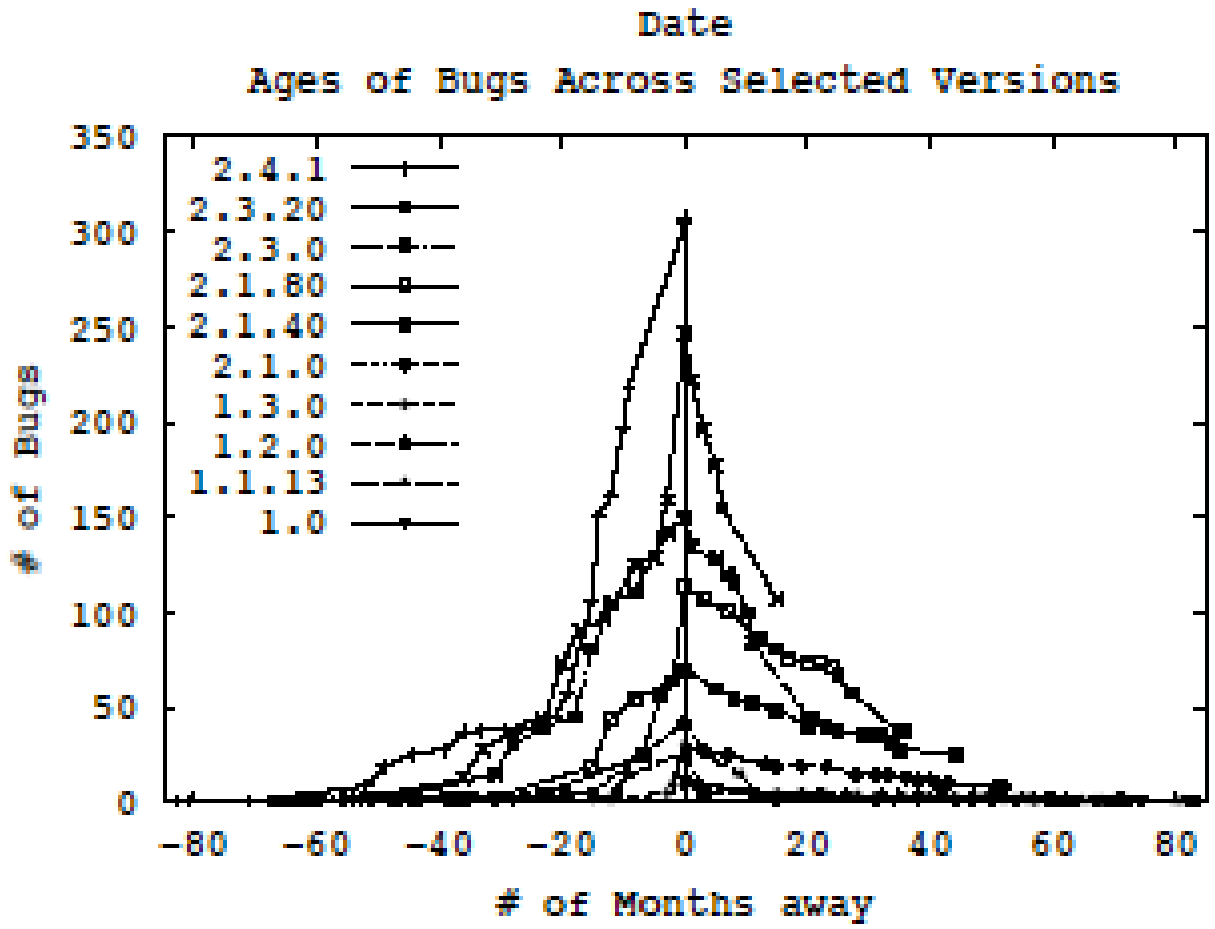
Where are the errors?

- Drivers have more errors than can be accounted by the code size
  - Drivers are written by many developers who many not understand the kernel
  - Drivers are not debugged as much as kernel proper



Ages of bugs

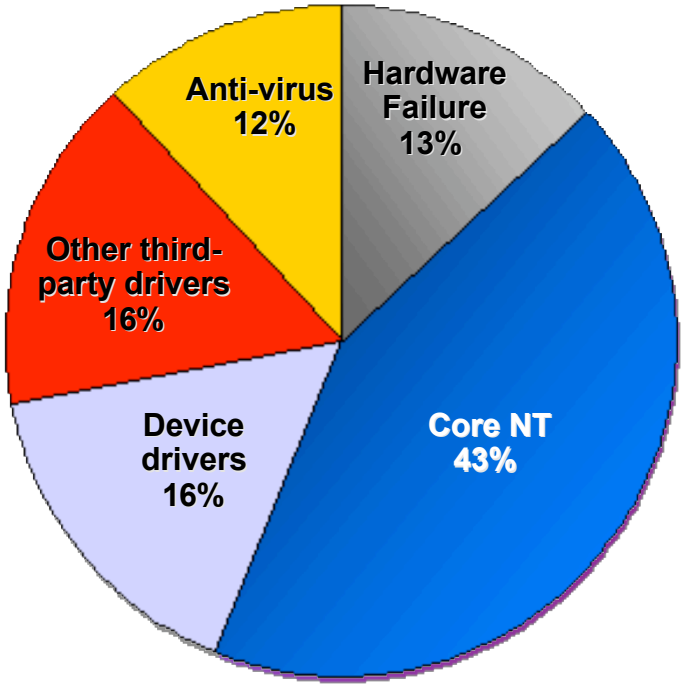
- Most bugs are fixed quickly



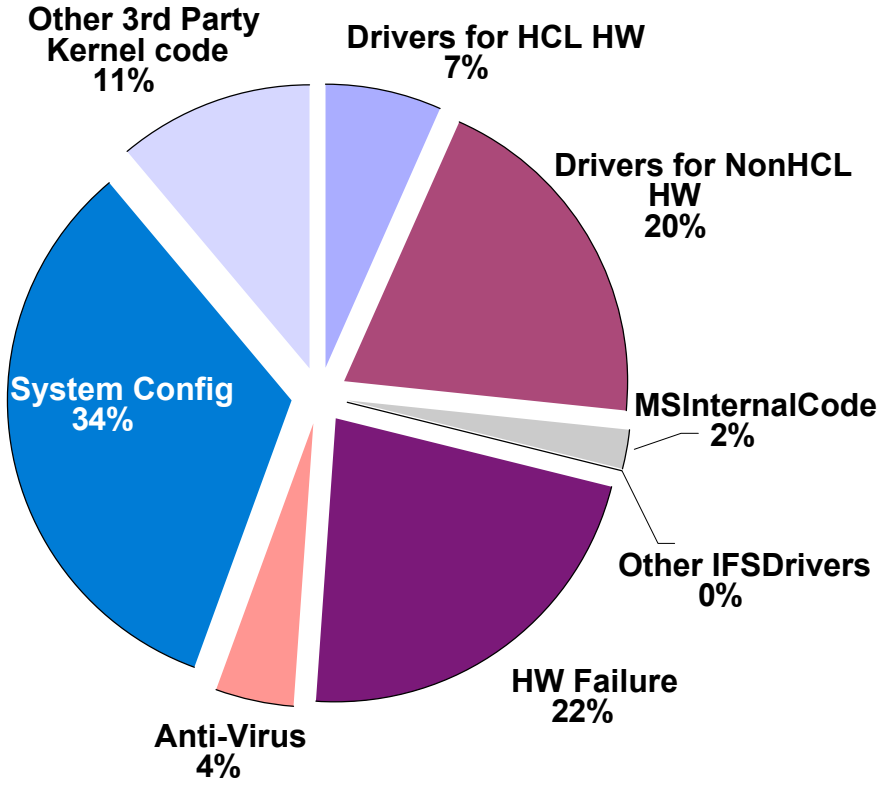
Windows 2000 - Failure Analysis.

32% of NT 4 faults, 27% of W2k faults

NT4



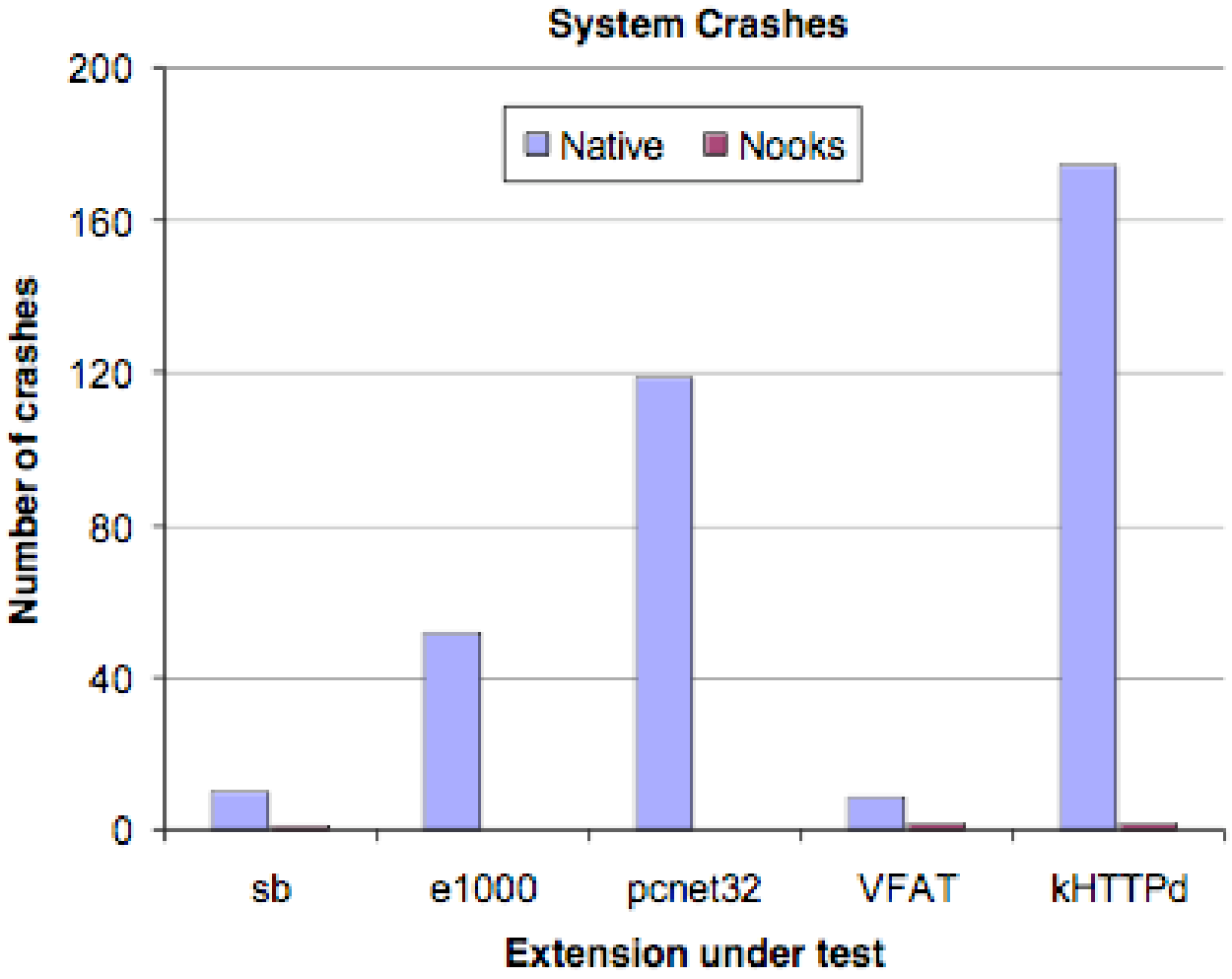
Windows 2000



Source: Brendan Murphy, Sample from PSS Incidents:

Nooks functions

- Isolation:
  - Prevent extensions from damaging the kernel
- Interposition:
  - Integrate existing extensions into the Nooks environment
- Object tracking
  - Track all kernel resources used by extensions
- Recovery
  - Detect and recover from a wide variety of extension faults


System Crashes
200
160
120
80
40
0
Native
Nooks
Number of crashes
sb
e1000
pcnet32
VFAT
kHTTPd
Extension under test