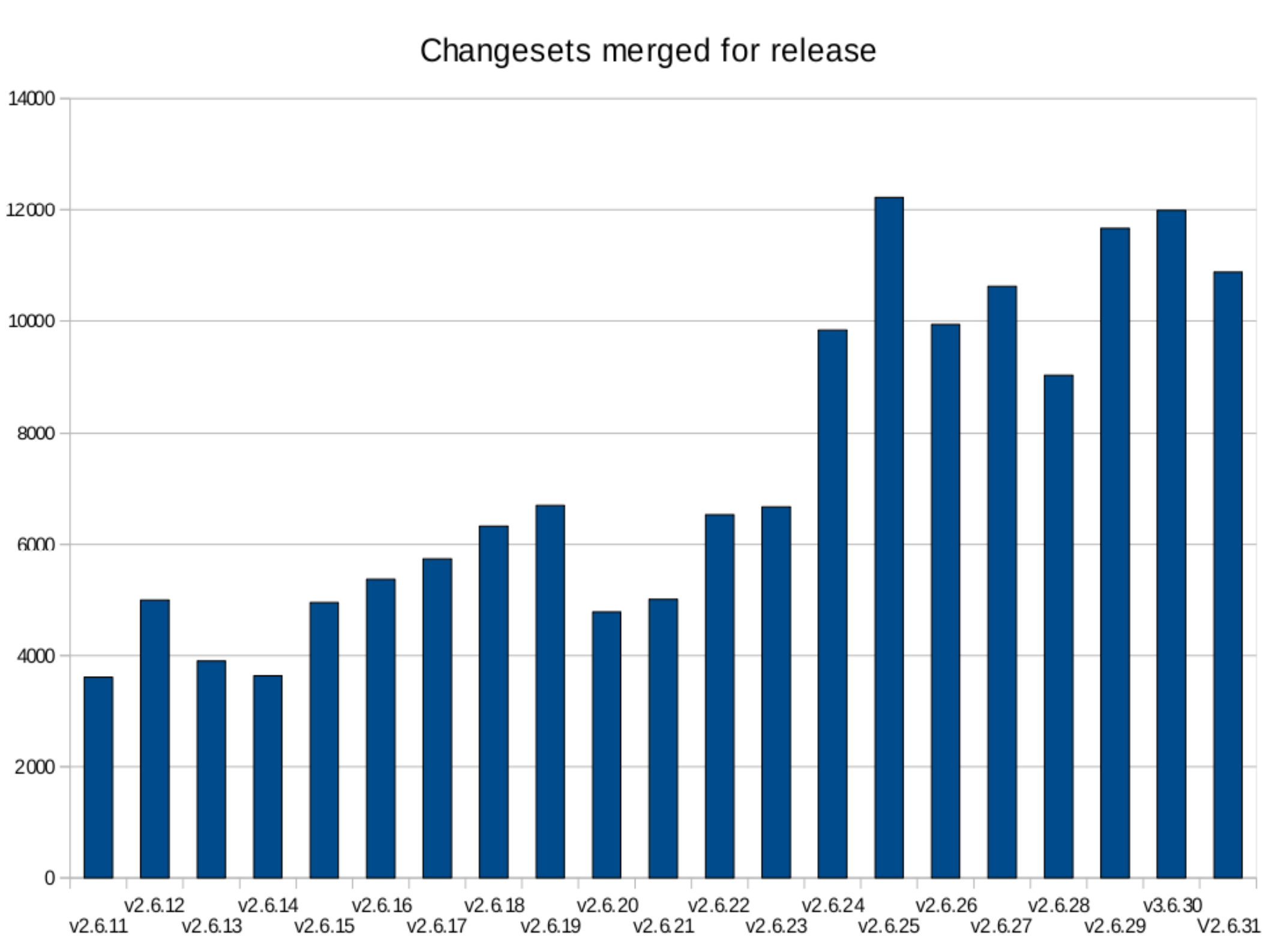
The Kernel Report

RTLWS 11 edition

Jonathan Corbet
LWN.net
corbet@lwn.net

“Famous last words, but the actual patch volume _has_ to drop off one day. We have to finish this thing one day."

-- Andrew Morton
September, 2005 (2.6.14)


Changesets merged for release
14000
12000
10000
8000
6000
4000
2000
0
v2.6.11
v2.6.12
v2.6.13
v2.6.14
v2.6.15
v2.6.16
v2.6.17
v2.6.18
v2.6.19
v2.6.20
v2.6.21
v2.6.22
v2.6.23
v2.6.24
v2.6.25
v2.6.26
v2.6.27
v2.6.28
v2.6.29
v3.6.30
V2.6.31

2.6.27 -> 2.6.31++

(October 9, 2008 to September 18, 2009)

48,000 changesets merged
2,500 developers
400 employers

The kernel grew by 2.5 million lines

2.6.27 -> 2.6.31++

(October 9, 2008 to September 18, 2009)

48,000 changesets merged
2,500 developers
400 employers

The kernel grew by 2.5 million lines

That come out to:

140 changesets merged per day
7267 lines of code added every day

The employer stats

None	19%	Atheros	2%
Red Hat	12%	academics	2%
Intel	7%	Analog Devices	2%
IBM	6%	AMD	1%
Novell	6%	Nokia	1%
unknown	5%	Wolfson Micro	1%
Oracle	4%	Vyatta	1%
consultants	3%	HP	1%
Fujitsu	2%	Parallels	1%
Renesas Tech	2%	Sun	1%

2.6.27 (October 9, 2008)

Ftrace

UBIFS

Multiqueue networking

gspca video driver set

Block layer integrity checking

2.6.28 (December 24, 2008)

GEM graphics memory manager

ext4 is no longer experimental

-staging tree

Wireless USB

Container freezer

Tracepoints

2.6.29 (March 23, 2009)

Kernel mode setting

Filesystems
- Btrfs
- Squashfs

WIMAX support

4096 CPU support


EXIT

2.6.30 (June 9)

TOMOYO Linux

Integrity measurement

ext4 robustness fixes

R6xx/R7xx graphics support

Adaptive spinning mutexes

Object storage device support

FS-Cache

Nilfs

preadv()/pwritev()

Threaded interrupt handlers

2.6.31 (September 9)

Performance counter support

Char devices in user space

Kmemleak

fsnotify infrastructure

TTM and Radeon KMS support

Storage topology

...about finished?

...about finished?

...so what's left?

2.6.32 (early December)

Devtmpfs

Lots of block scalability work

Performance counter improvements

Scheduler tweaks

Kernel Shared Memory

HWPOISON

Networking

“Based on all the measurements I'm aware of, Linux has the fastest & most complete stack of any OS.”

-- Van Jacobson

But...

Scalability remains a problem

Especially with:

- High network speeds
- Small packets

Packet filtering and firewalling

iptables has served us well since 2.4

Problems:

- Much duplicated code
- Difficult user-space interface
- Inflexible

Nftables

Remove protocol-awareness from the kernel
- ...replace with a dumb virtual machine

Rules are translated in user space

Advantages
- Much smaller code base
- Greater flexibility
- Better performance

Other networking stuff

Network namespace development
...still...

Netfilter improvements

802.15.4 stack (Zigbee and more)
2.6.31

Lots of wireless driver work

Filesystems

ext4

Advantages
- Better performance
- Many limits lifted
- ext3 compatibility

Still stabilizing
- But generally works quite well

Btrfs

A totally new filesystem

Advantages

- Performance
- Full checksumming
- Snapshots
- Internal volume management / RAID

Merged for 2.6.29

- Still very experimental

Solid-state storage

Rotating storage is dying

- ...well, maybe...

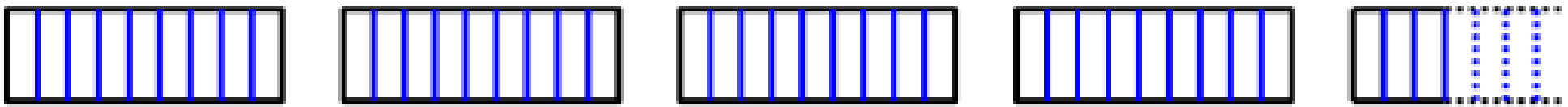
Solid-state devices are cool

- Fast
- Power-efficient
- Shock-resistant

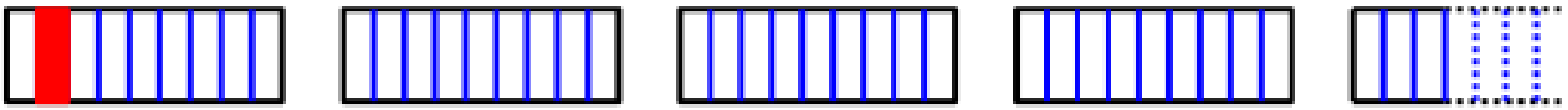
Solid-state storage

Also presents some challenges...

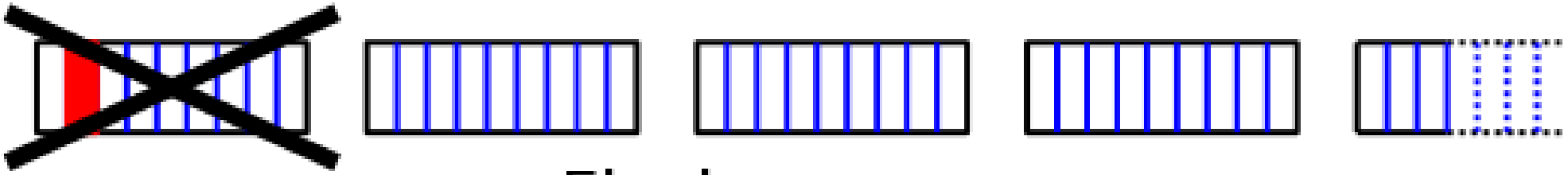
Flash memory



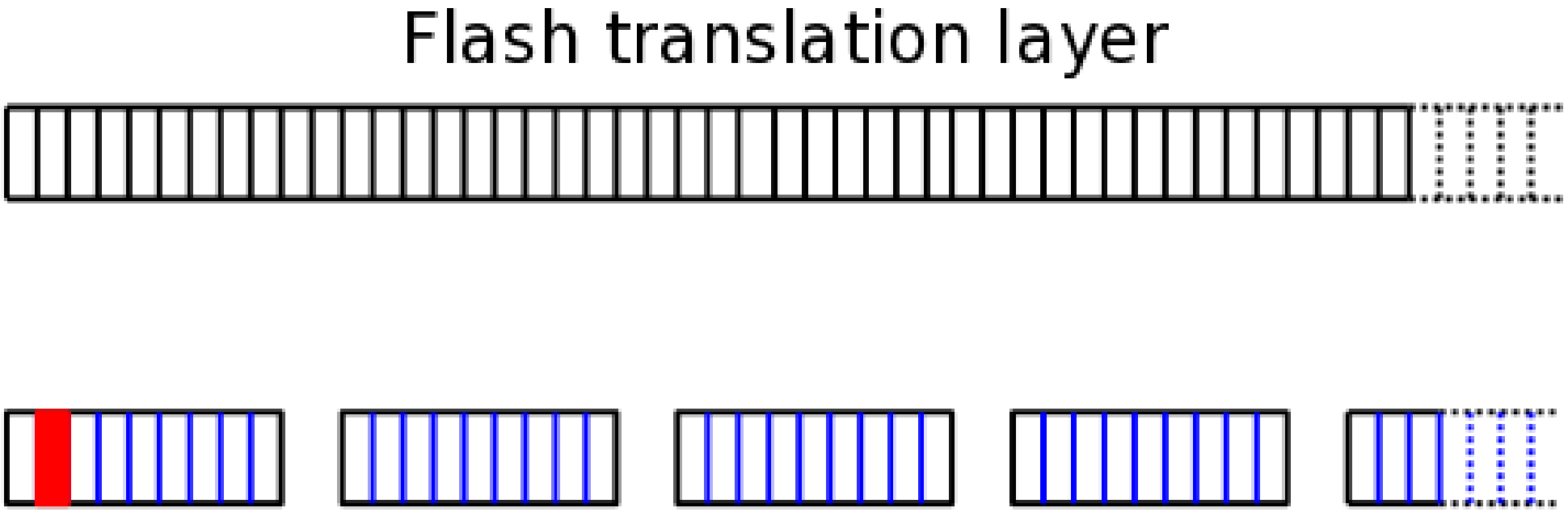
Flash memory

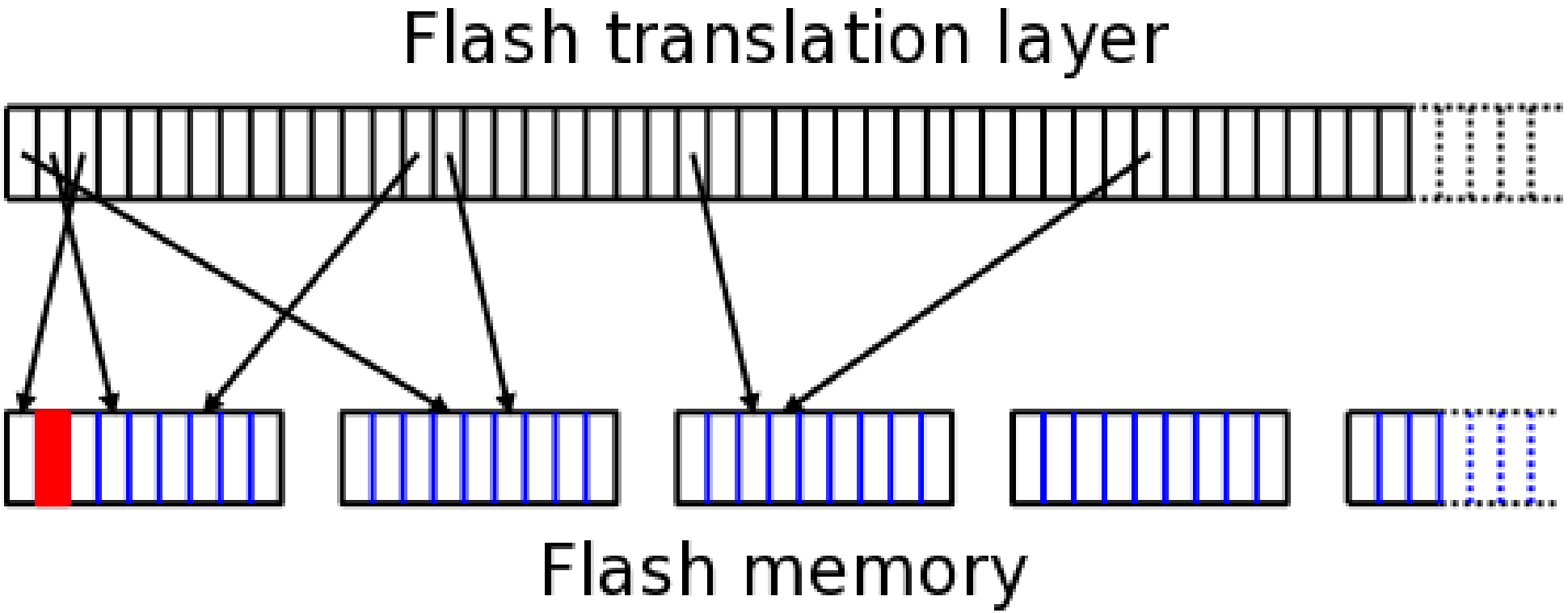


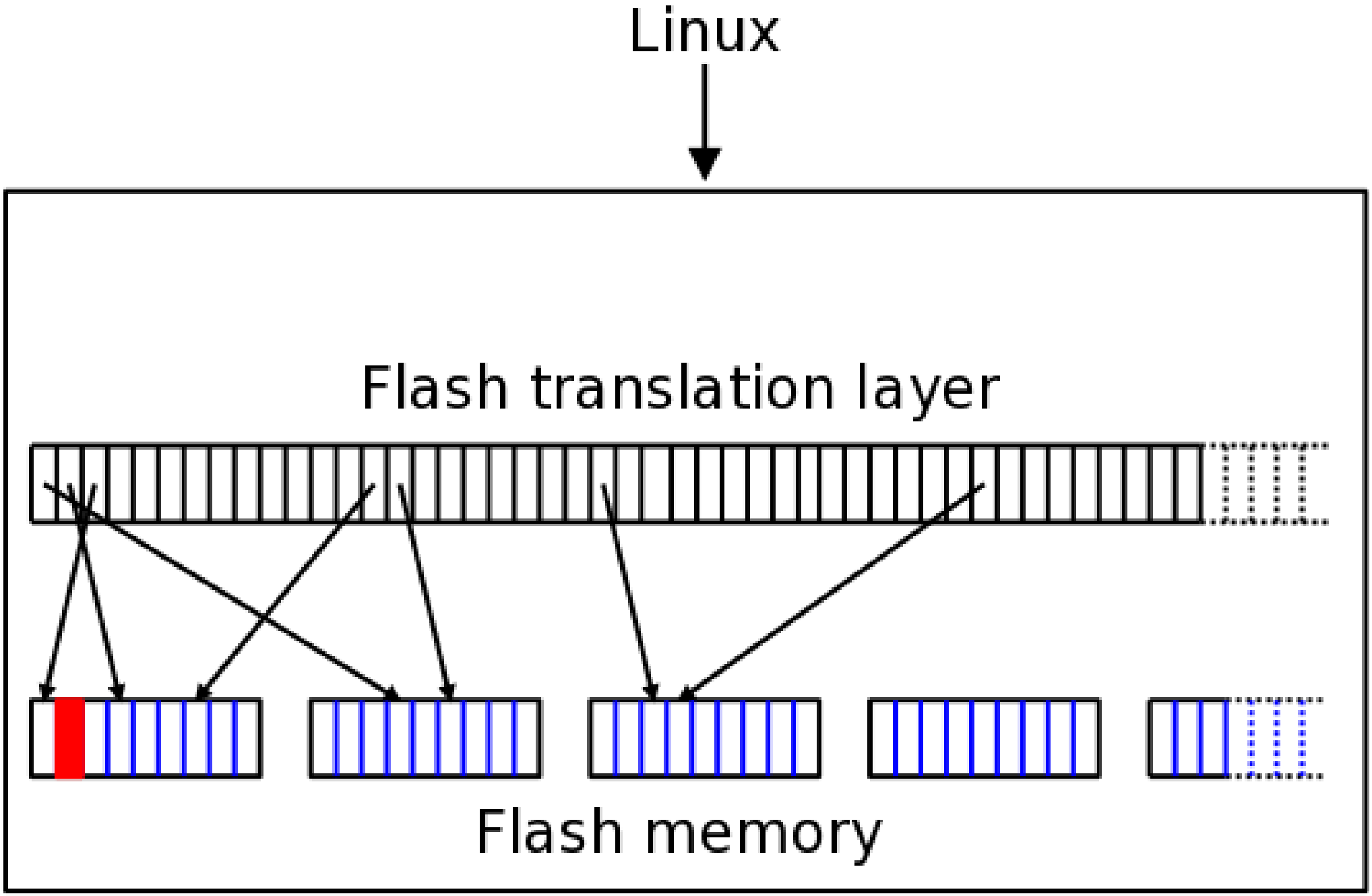
Flash memory



Flash memory


Flash translation layer
Flash memory


Flash translation layer
Flash memory


Linux
Flash translation layer
Flash memory

SSD: What to do?

Figuring out TRIM/DISCARD support

Using topology information

Smarter filesystems

- btrfs, nilfs, ubifs, ...

Solid-state storage

The longer-term problem:

- SSDs will soon be capable of 100,000+ ops/second
- Will the kernel be able to drive them that fast?

Robustness guarantees

ext3 raised the bar for crash robustness

ext4 tried to lower it again

Robustness guarantees

ext3 raised the bar for crash robustness

ext4 tried to lower it again

People complained.

I want a pony!

“No one ever, ever wrote "creat(); write(); close(); rename();" and hoped they would get an empty file if the system crashed during the next 5 minutes.”

-- Valerie Aurora

What kind of guarantees do we owe our application developers?

New APIs?

fbarrier()

acall()

readdirplus()

copyfile() [formerly reflink()]

kevents

A replacement for sockets

“Over the years, we've done lots of nice 'extended functionality' stuff. Nobody ever uses them. The only thing that gets used is the standard stuff that everybody else does too.”

-- Linus Torvalds

Virtualization

Mostly done - in the kernel, at least

- Xen Dom0 still out-of-tree

Remaining work: performance, management

KSM

Kernel shared memory

Scan memory for identical pages
- Dump duplicates and share one copy
- Pages marked copy-on-write

Merged for 2.6.32

Compcache

Swap out memory - to memory
- compress it on the way

Can double the amount of apparent memory

Containers

Lots of namespace work done
- Still stabilizing

Yet to do:
- Resource controllers
- Checkpoint/restart

Photo: photohome_uk

Hardware support

Near universal

A few remaining problems
- Graphics adapters
- Some network adapters

The -staging tree
- A home for substandard drivers

Power management

A variation on the hardware support problem

Power management

Photo: Terren in Virginia

Power management

Coming soon: runtime power management

- Better control of devices in a running system

Realtime

“While we never had doubts that it would be possible to turn Linux into a real time OS, it was clear from the very beginning that it would be a long way until the last bits and pieces got merged.”

-- Thomas Gleixner

Status of realtime

Code is mostly stable
- Shipped by numerous vendors

User-visible changes are all in mainline

What's not:
- ~~Threaded interrupt handlers~~
- Sleeping mutexes
- Lots of bits and pieces

Security

TOMOYO Linux
- Pathname-based mandatory access control
- 2.6.30

Integrity measurement
- 2.6.30

Still waiting:
- AppArmor
- fanotify

Open issue: sandboxing

Tracing

Photo: Armel Genon

SystemTap

A powerful dynamic tracing environment

Some problems

- Complex, difficult to use
- Requires lots of ancillary data
- Disconnect with kernel community

Ftrace

Lightweight kernel tracing facility
- Popular with kernel developers

Lots of static tracing options
- Maybe dynamic tracing in the future

Where a lot of the action is

~~Perfcounter~~ Perf Events

Access to performance monitoring registers
- Useful for low-level optimization

Integrated with tracepoints

Lots happening in this area

LTTng

Linux Trace Toolkit

- Well-developed static tracing toolkit
- Extensive user-space tools

Participation

The kernel development community is growing

We still have trouble with:

- Binary-only modules
- Withheld code
- Language barriers
- Cultural differences
- ...

Documentation/development-process

Questions?