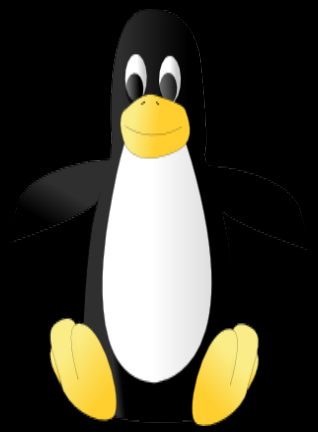
Linux Tiny
Penguin Weight Watchers



Thomas Petazzoni – Free Electrons
thomas@free-electrons.com

Who am I ?

- Since January 2008, works for **Free-Electrons** Embedded Linux and Free software consulting company

- Before, kernel developer for a storage virtualization technology for Linux clusters

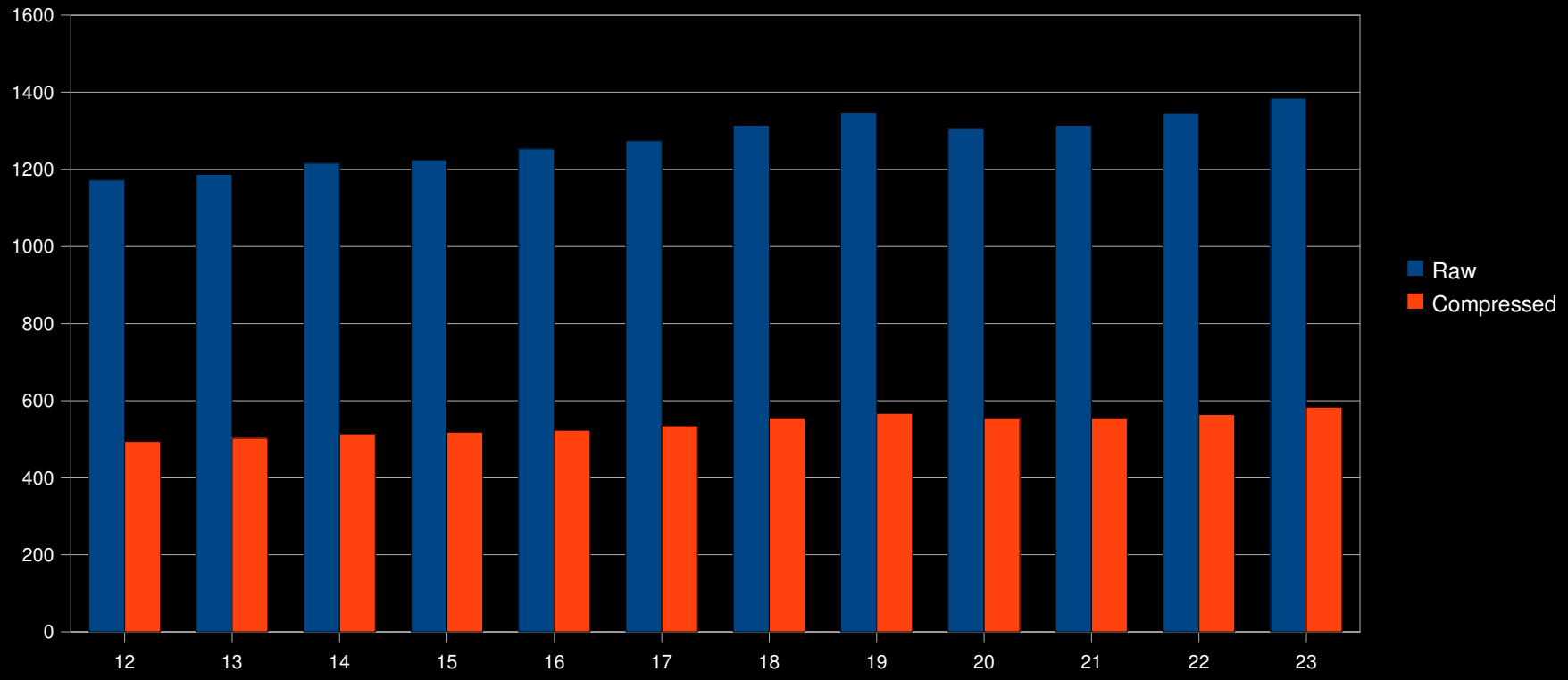
What matters to kernel users ?

- Desktop and enterprise users
  - Perfomance
  - Features
  - (mostly)
- Embedded users
  - Size
  - (especially on the high volumes CE market)

Why size matters ?

- Wish of the kernel community to get the embedded vendors into kernel development
  - They tend to prefer old versions of the kernel
  - Do not work on mainline inclusion, for product life cycle reasons
- Need to address their needs even with the current versions of the kernel
  - So that they don't stick with old releases

Kernel size increase in 2.6



Test case: i386 architecture, allnoconfig + IDE + ext2 + ELF

Between 2.4 and 2.6

- Renesas SH4
  - Compressed: from 654 KB to 864 KB, **+32.1%**
  - In RAM: from 1425 KB to 1679 KB, **+17.8%**
- MIPS NEC VR5500A
  - Compressed: from 807 KB to 897 KB, **+11.2%**
  - In RAM: from 1637 KB to 1819 KB, **+11.1%**
- Fairly old test, probably worst with more recent versions of the kernel

Linux Tiny

« Collect patches that reduce kernel disk and memory footprint as well as tools for working on small systems »

Matt Mackall, December 2003

Short history

- Started in December 2003 by Matt Mackall
- Matt's work sponsored by CELF in 2005/2006
  - Led to mainline inclusion of 17 patches
- Project mostly abandonned in 2006
- In 2007, CELF wish to revive the project
  - Michael Opdenacker, Free Electrons's founder, volunteered to become the new maintainer

Goals

- Improve the **mainstream** kernel
  - Hunt for bloat
  - Provide tools to find bloat
  - Reduce kernel memory consumption
- Remove features not needed in a production or dedicated system
  - No need for core dumps, debugging, console
  - Fine tune for a given system

Goals

- Today
  - 4 MB of RAM
  - 1.5 MB of storage, kernel included with a basic userspace
- Would like to support
  - 2 MB of RAM
  - compressed kernel as small as 300 KB on basic configurations

Current status

- Lots of work already merged by Matt Mackall
- ~50 patches need to be updated and mainlined
  - size reduction patches
  - code cleanup
  - memory size measurement tools
  - http://elinux.org/Linux_Tiny_Patch_Details
- Need to find more bloat and configure it out when possible

How to reduce your kernel size

Start from scratch

- `make allnoconfig`
  - Selects only the minimum features
- Then, add only the features you really need
- A smaller kernel is also a kernel faster to compile !

CONFIG_EMBEDDED

```
[ ] Configure standard kernel features (for small systems)  --->
```

```
[*] Configure standard kernel features (for small systems)  --->
[*] Enable eventpoll support
[*] Enable signalfd() system call
[*] Enable eventfd() system call
[*] Use full shmem filesystem
[*] Enable VM event counters for /proc/vmstat
[*] Enable SLUB debugging support
    Choose SLAB allocator (SLUB (Unqueued Allocator))  --->
```

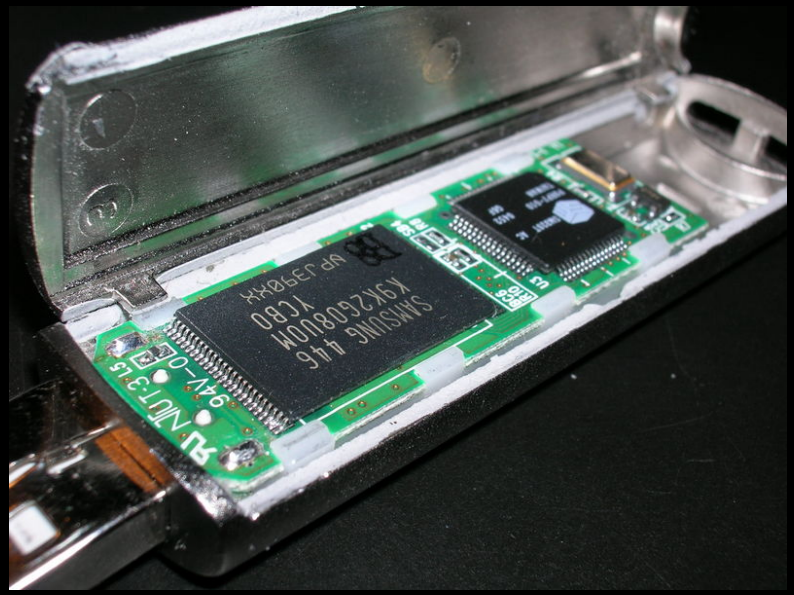
```
-*- Configure standard kernel features (for small systems)
[*]   Enable 16-bit UID system calls
[*]   Sysctl syscall support
[*]   Load all symbols for debugging/ksymoops
[*]     Include all symbols in kallsyms
[ ]     Do an extra kallsyms pass
[*]   Support for hot-pluggable devices
[*]   Enable support for printk
[*]   BUG() support
[*]   Enable ELF core dumps
[*]   Enable full-sized data structures for core
[*]   Enable futex support
```

SLOB allocator

- Alternative to the traditionnal SLAB allocator
- Written by Matt Mackall
- More code-size and memory-consumption efficient
  - But doesn't scale as well as SLAB / SLUB
- Need to see how it compares with the new SLUB allocator

CONFIG_BLOCK

- Allows to completely disable the kernel block layer
- Useful on systems with only Flash storage devices



Kernel Size Tuning Guide

- Compilation of tips and advises on reducing the kernel size
- http://elinux.org/Kernel_Size_Tuning_Guide

Results

- Default 2.6.23
  - raw: 1385 KB, compressed: 583 KB
- Mainstream Linux Tiny patches
  - raw: 1155 KB, compressed: 470 KB
- All Linux Tiny patches
  - raw: 1106 KB, compressed: 454 KB
- Can still do better, compressed size was 320 KB in 2.6.14

Future work

Existing patches

- Keep them updated with recent versions of the kernel
- Find approaches suitable for mainline inclusion
  - Can be a significant amount of work, depending on the cases
- Not let the patches get outdated

Find more features to remove

- Look for unconditionnally compiled code
  - Using `obj-y` in Makefiles
- Examples
  - pcspeaker on i386
  - CPU-specific support on i386
  - pdflush, readahead, swap
    - not necessarly needed on Flash-based systems, systems with read-only filesystems only, swap-less systems, etc.

Find more features to remove

- Other ideas
  - Write a simpler /proc filesystem, with a reduced fileset
    It currently consumes 130 KB
  - Migrate debugging interfaces to *debugfs*
  - do-printk patch by Tim Bird
    printk only on specific files
  - Compile printk() messages only above a given priority, proposed by Rob Landley

Monitor (and prevent ?) size increase

- Measure the size impact of each option of the kernel
  - Anyone knows what happened to Munehiro Ikeda's work presented at ELC 2006 ?
- Measure the size increase between kernel versions
- Provide numbers to the kernel community
  - on -rc releases
  - on the linux-next tree ?

Get involved

- Help us creating a smaller and simpler kernel
  - Opportunities to discover the kernel, learn, experiment
  - Read other's code and discuss mainlinable solutions with core kernel developers
- Web site
  http://elinux.org/Linux_Tiny
- Mailing list
  http://selenic.com/mailman/listinfo/linux-tiny

Quote

« One of my most productive days was throwing away 1,000 lines of code »

Ken Thompson