

# Making WebKit Faster

## A Look at the Tools

Holger Freyther @ FOSS.IN 2009

# A Simple Task Description

- Make the Qt port of WebKit faster

# How To Get Started?

- Optimize by myth?
- Or pick a more scientific approach?

# A Look At Optimizing By Myth

- Claim something to be the cause
- E.g. Linux Input framework adds to much overhead
- Implement char device in driver
- Get the placebo effect

# Picked The More Scientific Approach

- Defining what faster means
- Make it possible for others to verify
- **I picked this even if it is more work**

# Defining Performance

- How long does it take to load
- How long does it take to ...
- Performance here is the time something takes
- **BUT** also how much memory it consumes

# The Theoretic Approach

- Pick a specific area to optimize
- Create a reduction
- Optimize the reduction
- Look at the general improvement

# Looking At The Tools - Overview

- Mirroring utility
- [libmemusage.so/memusagestat](http://libmemusage.so/memusagestat)
- memprof
- oprofile
- Qt utilities

# Mirroring Utility

- Use real webcontent for testing
- Uses a QWebPage with custom network manager
- All requests are stored to a sqlite db
- Allows everyone to test the same thing

# Libmemusage.So/Memusagestat

- A library that overwrites malloc/free/mmap
- It is part of glibc and working since 2.11
- LD\_PRELOAD=/lib/libmemusage.so
- MEMUSAGE\_OUTPUT=memusage.data
- Visualize with memusagestat

# Memprof And Speedprof

- Comes with library and GUI
- For every malloc/free a backtrace is generated
- Visualizes backtrace, shows a nice graph

# OProfile

- A kernel modules and userspace app
- Interrupts execution, creates a backspace
- Later oprofile can be used to look at it
- Provides many useful bits

# Qt Helpers

- QBENCHMARK macro for benchmarks in test cases
- Generate a paint trace from application execution

# Doing It In Practice

- Use mirrored content
- Record paint trace during the loading test
- Replay on target and watch with OProfile
- Optimize and repeat the previous step

# Leassons Learned

- Creating a benchmark is hard
- Sometimes thinking helps and is better than bruteforce
- Do something less often is a good strategy

**That's all, folks!**

<http://www.secretlabs.de/talks/2009/foss.in>