TigerVNC and libjpeg-turbo

The best remote desktop

Red Hat
Adam Tkáč

February 11, 2011

Agenda

1. **TigerVNC & Remote desktop**
   - Use cases
   - Project
   - Architecture
   - TigerVNC 1.1
2. **libjpeg-turbo**
   - Background
   - Speedup 1 - SSE2
   - Speedup 2 - Colorspace extensions
   - Speed comparation

TigerVNC

Use cases

- working on remote computer
  - GUI is more user friendly for many people than CLI
  - some programs don't have CLI interface
- share desktop with multiple users (teaching)
- thin clients

TigerVNC project

- http://tigervnc.org
- forked from TightVNC in 2009 by some TightVNC and TurboVNC developers
- goals of the TigerVNC project
  - maintaining specification of RFB extensions
  - minimal consumption of CPU time
  - TigerVNC server based on the newest X.Org
  - support wide range of platforms (Windows, Linux, Solaris, OS X)

Architecture

- uses RFB (Remote FrameBuffer) protocol
- client – server architecture over TCP/IP
- server sends rectangular differences in it's framebuffer to clients
- client sends key/mouse events to server
- Xvnc (UN*X) and winvnc4 (Windows) servers
- vncviewer client
- libvnc.so module to Xorg server

TigerVNC 1.1

- upcomming major feature release (the first beta is planned to the next week)
- interesting features
  - TLS encryption support
  - X.509 authentication support
  - basic PAM support
  - major speed improvements on x86_64 architecture

libjpeg-turbo

Background

- `http://sourceforge.net/projects/libjpeg-turbo`
- TigerVNC uses JPEG for compression of image rectangles
- JPEG compression/decompression consumes most of CPU time
- fast JPEG routines decrease CPU utilization
- fork of libjpeg implementation, 100% API/ABI compatible
- the fastest open source JPEG implementation

Speedup 1 - SSE2

- JPEG consumes most CPU time by arithmetic operations with vectors
- instructions supported on vast majority of current x86 and x86_64 CPUs
- arithmetic operation on multiple integers/floats per instruction
- separate registers, don't conflict with CPU/FPU registers

Speedup 2 - Colorspace extensions

- original libjpeg supports only RGB pixel format
- VNC server/client can run on different endianesses
- reordering of pixel RGB elements is expensive
- libjpeg-turbo allows native conversion from/to RGB, BGR and other widely used formats

Speed comparation

- 1.8x – 4.5x faster than libjpeg
- comparable with IPP library on x86_64, slightly slower on x86
- visible difference between libjpeg-turbo and libjpeg when working with large images
- the fastest open source JPEG implementation

The end.

Thanks for listening.