PJSIP: Open Source Compact SIP and Media Stack

Perry Ismangil
and
Benny Prijono
Realtime Multimedia Communications

- Voice over IP (VoIP)
- Audio over IP: remote audio broadcasting
- Unified Communications
- Virtual world conversations
Realtime Multimedia Communications

- Signaling: SIP
- Negotiation: SDP
- Media transmission: RTP/RTCP
- Presence
- NAT traversal
Project Objectives

● Integrated SIP and Media stack package
  ○ Ease of use
● Compliance
  ○ Regression tests against compliant servers and clients
●
  ○ Test by community
●
  ○ Global interoperability events from SIP Forum
Project Objectives

- Very Very Portable
- One of the fastest (performance)
- One of the smallest (footprint & memory)
PJSIP History

Background:

- 2000: SLSIP (C++)
  - pc2call.com, first SIP web based dialer
- 2001-2003: prototyping
PJSIP History

- Feb 2006: PJSIP and PJMEDIA 0.5
  - pjsip.org, dual license

- Nov 2006: Incorporation of Teluu

- Nov 2007: Global SIP Interoperability Testing
PJSIP History

- Oct 2008
  - Release 1.0
    - More than twenty known projects/products using pjsip
    - More than 250,000 lines of code (source: ohloh.net)
PJSIP Community

- Downloads: 800 per month
- More than 500 mailing list members
- From the community
  - Various platform supports
  - Additional language bindings
Platforms

All trademarks belong to respective owners
Framework Architecture

Operating System / Platform Dependent
Technology Choice

- C instead of C++:
  - Size
  - Portability
- Build system:
  - Mixture of autoconf, Microsoft, and Symbian SDK
  - Many more request like Borland, Xcode etc.
## Footprint (linux, i386)

### size -t:

<table>
<thead>
<tr>
<th>Library</th>
<th><code>.text</code></th>
<th><code>.data</code></th>
<th><code>.bss</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>libpjlib</td>
<td>58 KB</td>
<td>0.1 KB</td>
<td>0.7 KB</td>
</tr>
<tr>
<td>libpjlib-util</td>
<td>64 KB</td>
<td>~0 KB</td>
<td>~0 KB</td>
</tr>
<tr>
<td>libpjnath</td>
<td>90 KB</td>
<td>0.9 KB</td>
<td>~0 KB</td>
</tr>
<tr>
<td>libpjmedia</td>
<td>213 KB</td>
<td>0.5 KB</td>
<td>3 KB</td>
</tr>
<tr>
<td>libpjmedia-codec</td>
<td>29 KB</td>
<td>0.3 KB</td>
<td>0.3 KB</td>
</tr>
<tr>
<td>libpjsip</td>
<td>165 KB</td>
<td>1.6 KB</td>
<td>13 KB</td>
</tr>
<tr>
<td>libpjsip-ua</td>
<td>44 KB</td>
<td>0.2 KB</td>
<td>~0 KB</td>
</tr>
<tr>
<td>libpjsip-simple</td>
<td>35 KB</td>
<td>0.5 KB</td>
<td>~0 KB</td>
</tr>
<tr>
<td>libpjsua</td>
<td>96 KB</td>
<td>0.3 KB</td>
<td>0.1 KB</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>794 KB</strong></td>
<td><strong>4.4 KB</strong></td>
<td><strong>17.1 KB</strong></td>
</tr>
</tbody>
</table>

KB
Notes on Footprint

- Done with "size -t" on the static libraries:
  - this measures ALL components in libraries
  - with static linking, only part of the libraries that are used by application are included in the executable
- Lots of features
- Various settings to reduce footprint by up to 129 KB (http://trac.pjsip.org/repos/wiki/FAQ#footprint)
- Heap usage: around 150 KB for application with two active calls
Case Study: IP Phone

- Linux on ARM9
  - familiarity of developers
  - ability to meet feature demands
Case Study: IP Phone

- Third party frameworks criteria
  1. Functionality
  2. Cost
  3. Support
Case Study: IP Phone

- PJSIP deployment
  - Optimization on ARM platform
    - codecs
    - resampling filters
General Porting Challenges

- Compiler differences and quirks
  - e.g. inline modifiers, 64bit integer, warnings

- Endianness
  - network protocols in network byte order, WAV files are little endian

- OS abstraction
  - pick lowest common denominator, e.g. Win32 API SuspendThread()/ResumeThread() are not available in pthread

- Sound device abstraction
  - oss, alsa, jack, wmme, ds, wasapi, wdmks, asio, coreaudio, symbian streaming, aps, nds, iphone...

- Floating-point support
  - fixed-point alternative algorithms
Even More Porting Challenges

- **Power consumption**
  - avoid polling --> Symbian mobiles
- **No heap memory management!**
  - no malloc()?! --> use pjlib pool
- **No OS!**
  - deeply embedded systems, e.g. TI
- **Everything is 16bit!**
  - In TI DSP, sizeof(char) == sizeof(short) == sizeof(int) == 2 (!)
  - SIP is text based protocol!
  - Translator between network and application
- **Broken compiler**
  - Zilog z80 compiler doesn't support nested struct declaration
Roadmap

- Even faster for developers to pickup
- More interoperability testing
- Mobile
Thanks

- Any questions

- Please visit http://www.pjsip.org

- Join our mailing list or leave blog comments
Teluu - Communicate Everywhere

- The company behind pjsip project

- Provides
  - Flexible licensing
  - Professional support
  - Certified and optimised binaries
  - Network of third party consultants

http://www.teluu.com