Building products with NetBSD - thin-clients

- Stephen Borrill - Precedence Technologies
- sborrill@NetBSD.org
- sborrill@precedence.co.uk
Contents 1

- Who am I?
- What is a thin-client?
  - Client software
- NetBSD: first encounter
- Relevant experience and jobs
- Product history
  - The End?, A New Hope, Baggage, Brainwave
Contents 2

• Product requirements
• How NetBSD meets requirements
• Problems hit against
• Problems not easily solved
• Future developments
• Changes since September 2007
Who am I?

- Live and work in Cambridge, UK
- Managing Director of Precedence Technologies
- PhD in psychoacoustics (not Computer Science!)
- Citrix Certified Administrator (XenApp/XenServer)
- Citrix Certified Sales Professional
- NetBSD user since 1994
- NetBSD and pkgsrc developer/committer since Jan 2007
What is a thin-client?

- Small physical size
- No moving parts
- Low/medium performance
- Low power consumption
- Fast start time
What is a thin-client?

- No local storage (firmware only)
  - Small amount for local settings perhaps

- Centrally managed from network
  - Clients are commodity items that can be immediately swapped out if they fail

- Contains network client software, but no general applications
  - c.f. a fat-client which has whole OS and applications locally (e.g. MacOS, Windows), plus complex local configuration
Usual client software in thin-clients

- Citrix ICA
- Microsoft RDP
- X11
- Terminal (ssh/telnet/tn3270)
- Web browser
- VNC
NetBSD: first encounter

- Used Masscomp Unix and Irix as part of PhD
- Avid Acorn (ARM) user (at time)
- Wanted a Unix-alike to use and learn on
- Acorn RISCiX was 4.3BSD-based, but expensive, old and not for new machines
- RiscBSD project launched in 1994.
  - RiscBSD became NetBSD/arm32 and then NetBSD/acorn32
Relevant jobs

- **Feb 1996**: started at Acorn Education in tech support (part time – still doing PhD)

- **Apr 1996**: Acorn and Apple UK formed Xemplar Education.
  - Xemplar Education 2\textsuperscript{nd} biggest supplier of IT to UK education. I transferred there full-time
Product history

- **Jan 1996**: Acorn launched Network Computer Reference Design with Oracle
  - NC was 48MHz ARM7500FE, 16MB RAM, 10Mbit Ethernet, custom version of RISC OS, no local storage, boot from NFS, browser in ROM

- **Oct 1996**: Xemplar given 2 pre-release NCs by parent company Acorn. Sales/Marketing took one – I took other. I had a plan...
Product history

- Used NetBSD/arm32 on RiscPC server
  - Apache for web-based UI
  - Wrote web-based administration
  - Wrote webmail package

- Wrote/designed application framework
  - Not true thin-client as, except for web-based software, applications were running on client

- From summer 1997, solution sold to UK schools (NetBSD-based NCServer)
  - Means Apple were selling BSD Unix in 1997; a long time before Mac OS X.
The End?

- **Summer 1998**: Became Network Computer Technical Manager
- **Summer 1998**: Large roll-out throughout UK
- **Jan 1999**: Acorn sell 50% share to Apple, i.e. Xemplar were now 100% Apple
- **Mar 1999**: Most staff made redundant (inc. me)
  - Meant customers throughout UK with paid-up support contracts, but no support staff
- **Apr 1999**: Precedence starts trading. Buys NC stock. Contracted by Apple to provide support. Given all IPR and source code
A New Hope

- **Apr 1999**: Precedence sell CATS to replace aging RiscPCs (running NetBSD/cats)
- Re-position server as being a general purpose Internet/intranet/email/filtering server (NetManager)
- **Nov 1999**: Switch NetManager to NetBSD/i386
- **Apr 2002**: Start complete modular re-write (NG on NetBSD/i386 1.6.1)
- **Today**: NetManager NG 5 selling well (NetBSD/i386 5.0_STABLE)
  - Still do occasional NetManager builds for SGI (mips), PowerMac (macppc), HP Jornada (hpcsh), RiscPC (acorn32) as proof of concept
Baggage

- Still selling 48MHz clients, very proprietary
- Very slow, old RISCOS ICA client
- Evaluated netbooting NetBSD/acorn32 and running Linux/ARM ICA client – no benefit
- Refuse to sell Windows CE/XPe clients
Linux?

• Start to sell 233MHz+ Linux-based clients
  – Very poor support
  – Basic software

• Found alternative Linux-based clients
  – Slick UI
  – Pain to buy (have to import)
  – Expensive
  – No obvious future development plan
Brainwave

• Develop NetBSD-based solution to convert old PCs into thin-clients

• Chose name: ThinIT

• Started on 2\textsuperscript{nd} May 2003. First release on 23\textsuperscript{rd} May 2003 (v1.00)
ThinIT v1 outline

- Based on NetBSD 1.6 minimal install (base.tgz)
- Run from HDD with root mounted read-only
- Citrix ICA/Microsoft RDP clients only
- Use Linux emulation for Citrix ICA client
- Easy install (CDROM/floppy)
Brainwave v2

- Linux-based thin-clients expensive and few upgrades. No control over software.

- Why not use ThinIT on OEM hardware?

- Start from scratch to fix deficiencies of v1

- ThinIT v2.00 released Sep 05 for OEM hardware

- ThinIT v2.05 released for general PC: Sep 07

- ThinIT v2.08 in development (NetBSD 5.0-based)
Brainwave v2

- Search out low-cost, high performance clients
  - 1\textsuperscript{st} generation TCX released Sep 05 (1GHz VIA-C3, 64MB flash, 128MB RAM)
  - 1\textsuperscript{st} generation TCM released Jan 06 (1.5GHz laptop, 128MB flash, 256MB RAM)
  - 4\textsuperscript{th} generation TCX released Sep 08 (1GHz VIA-C7, 128MB flash, 512MB RAM)
  - 4\textsuperscript{th} generation TCM being built in Shanghai as I speak (12” widescreen laptop, 1.6GHz VIA-C7M, 1GB RAM, 128MB SSD)
  - Custom version for Asus EeePC
Product requirements (NetBSD)

- Needs to run from flash
- Easy build infrastructure
- Easy to test during development
- Should be able to boot from various sources (CDROM, USB, PXE, Flash, HDD)
- Should have slick consumer-facing user interface
  - No kernel text
  - No command line
  - GUI configuration
- Wide hardware support, but excellent performance on known hardware
- Needs to have small footprint (memory/storage)
Product requirements (General)

- Needs to be difficult to rip off or subvert
- Centrally configured
- Remote management
- Modular and extensible
- Many more session types
- Needs to run latest client software
How NetBSD meets requirements 1

- Run from flash
  - NetBSD installer has ffs image as root filesystem embedded in kernel with mdsetimage
  - Very easy to extend and build custom images
  - Requires small tweaks to boot multi-user as designed for single-user installation process

- Easy build infrastructure
  - build.sh can build whole system
  - Single make can do a lot

- Easy to test during development
  - Xen support good; generate filesystem image for Xen
How NetBSD meets requirements 2

- Boot from various sources
  - Very easy – just one file + bootloaders
  - Tweak pxebooters to hardwire TFTP path
    - bootxx_cd9660 means no more 2.88MB limit
    - boot.cfg allows boot menu for choice of kernels
      - e.g. run live from CD or run installer
  - Can still build floppies (though 3+ needed)
    - bootxx_fat16 allows USB booting w/o formatting
  - NTFS boot (dual boot with Windows w/o partitioning)
How NetBSD meets requirements 3

- **Slick user interface**
  - Kernel boot messages must go
    - `boot -z` is NOT quite silent, but much better in NetBSD 5.0
    - Cheap hack – make `printf` do same as `aprint_normal`
    - Allow verbosity level to be compiled into the kernel so remains quiet with all bootloaders
  - Lock down boot loaders
    - Password protect, hardwire kernel path

- Most of `rc.d` scripts rewritten
  - Differentiate between output to terminal (friendly text) and log to file (debugging)
How NetBSD meets requirements 4

- Slick user interface (continued)
  - Colour text, cursor positioning during text phase
    - Displays Ethernet link and wifi scanning status for example

- vesafb/splashscreen
  - Friendly boot logo (optionally animated)
  - Unfortunately incompatible with ACPI suspend
  - Retired in -current in favour of genfb

- GUI configuration (GTK)

- Lots of pretty screen savers (xlockmore)
User interface at boot time

- vesafb splashscreen (top left)
- non-vesafb version (bottom left)
- boot sequence (top right)
User interface when running
How NetBSD meets requirements 5

- Excellent performance on known hardware
  - Very cut down kernel configs for TCX and TCM clients and optimizations
  
  - Very quick boot times as no probing for unknown devices (fraction of a second after kernel loaded – took 10 seconds on ThinIT 2.05!)
  
  - Tweaked and updated X drivers
How NetBSD meets requirements 6

• Wide hardware support
  – GENERIC-type kernel for standard PC hardware
  – Unified self-optimizing kernel runs on a wide range of hardware
  – Many more drivers in NetBSD 5.0
  – New power-management framework in 5.0
  – Could easily switch to new processor/hardware
    • (though see caveats later)
How NetBSD meets requirements 7

- Small footprint
  - Crunchgen monolithic binary memory efficient
  - Kernel with embedded ramdisk (gzip -9 compressed)
  - Less than 3.5MB (for TCX)
  - Compressed vnd used for modules. Came along just at right time!
  - Severely pruned file list in modules (see later)
  - tmpfs efficient memory file system
Meeting general requirements

- Modular
  - ThinIT kernel is standalone. Knows how to upgrade itself, get settings, speak to various networks and find files on various filesystems
  - Supported by a number of modules which it loads either into RAM (over http/tftp/ftp or from CD) or from local filesystem (ffs, FAT, NTFS)
Meeting general requirements 2

- Modular (continued)
  - Modules are disk images created with makefs and configured as vnds (compressed + encrypted)

- Some required (e.g. libs, X)
- Some useful, but could be removed (e.g. gui)
- Session modules optional (e.g. ica, rdp, vnc)
- Some required by others (e.g. emul is required by ica)
- Potential for custom development
Meeting general requirements 3

- ThinIT Makefile.vnd builds modules
  - Can use pre-built pkgsrc binary packages
    - Removes *.h, *.a, *.la & man pages by default
    - Supports extracting against a fixed list
  - Can include part of src tree (e.g. lib/libc)
  - Can build local sources (e.g. session chooser)
  - Can define extra stages (e.g. uncompressed X fonts as they will be compressed later)
  - Builds ffs image, compresses and encrypts
Meeting general requirements 4

• More session types
  – Adding a new type as easy as adding a new module (based on pkgsrc binary packages)
    • Streamed video and DVD playback (vlc)
    • Web browser (opera)
    • Conferencing (mbone tools: vic, rat, wb)
    • ssh
    • vnc
    • Citrix ICA
    • Microsoft RDP
    • SIP
    • Datalogging

  – This variety of session types is unique in market
Meeting general requirements

- Difficult to rip off
  - Compressed modules encrypted
    - cgd tricky to use because must encrypt AFTER compression
      (ffs on vndz, vndz on ffs, ffs on cgd, cgd on vnd).
    - Extending vnd(4) to support decryption on-the-fly much easier
  
- Check hardware we're running on
  - Check cpu, video, audio, wifi, Ethernet, etc.
  - Refuse to run if doesn't exactly match what is expected
Meeting general requirements 6

• Difficult to rip off
  – Encrypt/obfuscate embedded filesystem
    • Problem is that kernel and ramdisk are compressed, so same problem as above.
  – Signed modules (future)

  – Licence management
    • Subvert management protocol to count machines sharing licence numbers
    • Allow each client to monitor other clients on the network
Meeting general requirements 7

- Centrally configured and managed
  - Fetches config file with http, ftp or tftp
  - Path configured with DHCP option 151
  - Supports groups and per-machine overrides
  - Plain text format (DOS or UNIX line endings)
    - session.1.type=ica
    - session.1.name=Run Windows
    - session.1.server=icaserver
    - ica.usb=b
  - Shadow screen (x11vnc)
  - Can use ssh for problem solving
  - Documentation on website for using with Windows
    DHCP/IIS
## File sizes in ThinIT v2.08 vs 2.05

<table>
<thead>
<tr>
<th></th>
<th>v2.08</th>
<th>v2.05</th>
<th></th>
<th>v2.08</th>
<th>v2.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdp.enz</td>
<td>110,080</td>
<td>(101,376)</td>
<td>lib.enz</td>
<td>2,545,664</td>
<td>(2,016,256)</td>
</tr>
<tr>
<td>vnc.enz</td>
<td>123,904</td>
<td>(124,928)</td>
<td>xdrv.enz</td>
<td>2,002,432</td>
<td>(2,048,000)</td>
</tr>
<tr>
<td>xvia.enz</td>
<td>162,304</td>
<td>(135,680)</td>
<td>tcx.krn</td>
<td>3,510,604</td>
<td>(2,426,211)</td>
</tr>
<tr>
<td>xi810.enz</td>
<td>223,232</td>
<td>(207,872)</td>
<td>tcm.krn</td>
<td>3,787,473</td>
<td>(2,587,539)</td>
</tr>
<tr>
<td>xsis.enz</td>
<td>281,088</td>
<td>(225,792)</td>
<td>video.enz</td>
<td>4,649,472</td>
<td>(3,930,624)</td>
</tr>
<tr>
<td>ssh.enz</td>
<td>274,944</td>
<td>(261,632)</td>
<td>thinit.krn</td>
<td>5,638,161</td>
<td>(4,196,849)</td>
</tr>
<tr>
<td>confer.enz</td>
<td>420,864</td>
<td>(418,304)</td>
<td>emul.enz</td>
<td>6,132,224</td>
<td>(4,336,640)</td>
</tr>
<tr>
<td>tk.enz</td>
<td>1,228,288</td>
<td>(1,227,776)</td>
<td>x.enz</td>
<td>6,714,368</td>
<td>(5,730,303)</td>
</tr>
<tr>
<td>gui.enz</td>
<td>1,956,352</td>
<td>(1,301,504)</td>
<td>opera.enz</td>
<td>9,082,880</td>
<td>(8,435,712)</td>
</tr>
<tr>
<td>ica.enz</td>
<td>1,985,536</td>
<td>(1,886,720)</td>
<td>TOTAL</td>
<td>38,897,484</td>
<td>(32,333,666)</td>
</tr>
</tbody>
</table>

.krn files are kernels for different machines
Total includes only TCX kernel and TCX video driver (xvia)
2.05 based on 4.0_RC4 (Sep 2007). 2.08 based on 5.0_STABLE (May 2009)
File size changes explained

- TCX kernel in v2.05 = 2,426,211 (4.0_RC4)
- TCX kernel in v2.08 = 3,510,604 (5.0_STABLE)

- Remember, this is kernel + ramdisk (gzipped)

- Raw kernel v2.05 → v2.08 +16% (~170k)
  - New 5.0 options: pseye(4), uvideo(4), PUFFS, etc.
  - Old options enabled: NTFS, ext2fs
File size changes explained

- Embedded ramdisk v2.05 → v2.08 +65% (+800k)
  - Crunched binary (+600k)
    - WPA (wpa_cli, wpa_supplicant) (150k)
    - Rest mostly due to library changes (e.g. proplib for ifconfig)
  - Other data (+200k)
    - WLAN HAL/firmware (ral, rum, ath, wpi, iwi, bwi)

- General increase in software size
  - Mainly Linux emulation and X libraries

- Need to bear in mind that early TCXs had only 64MB
Remote management with ThinITTool
- Shutdown, reboot, probe, configure, view logs, start sessions, lock screen, play music(!)
- NetBSD + Linux CLI binaries (no library dependencies)

NetBSD% thinittool -a PING
192.168.1.115 ACK:00000001 PING id=002354222142 group=r2 name=room2-1 fw=2.08 type=tcmu2
192.168.1.56 ACK:00000001 PING id=000e0c21fae6 group=r2 name=room2-2 fw=2.08 type=tcx3
192.168.1.110 ACK:00000001 PING id=0000dcc262fa group=r1 name=room1-2 fw=2.07 type=tcm1

NetBSD% thinittool -c 192.168.1.56 SHUTDOWN
NetBSD%
Client Management (Windows)

- Windows CLI + GUI binary
  - compiled on NetBSD with mingw
Client Management (web)

- Web-based frontend on NetManager
Problems hit against

- Few OEM laptop suppliers means pressure to use off-the-shelf laptops
  - Windows tax
  - HDDs redundant if replaced by flash
  - Could invalidate warranty
  - No custom BIOS logos/settings
  - Laptop manufacturers change specifications frequently
  - Usually use brand-new chipsets without immediate OSS support (e.g. Intel 5100 wifi, Atheros wifi, Attansic Ethernet)

- Force upgrade to newer OS
  - So unlike most embedded projects, ThinIT generally needs to run more cutting edge code.
Perils of upgrade to newer OS

- NetBSD 3.1 didn't support WEP with iwi(4) – forced to switch to 4.0_BETA2.
  - Meant lot of work in short time against tight deadlines
  - Compressed vnds broken at switch – bad timing!

- Meant new Linux emulation (SuSE 10 vs 9)
  - More to chop out!

- New Citrix ICA client
  - Required more Linux libraries
Perils of upgrade to newer OS

- Needed new Atheros and agp drivers – forced to switch to 5.0_BETA
  - Need to rework all existing patches against src and pullup drivers from -current
    - vnd encryption
    - Open-source ath(4) HAL
    - Attansic Ethernet
    - Make autoconf messages quieter
    - Allow quiet boot messages to be forced on in the kernel as kernel could be loaded from multiple bootloaders
    - Undo few things prohibited by default in 5.0 (e.g. re-mount filesystem read-only)
    - Specify multiple resolutions for vesafb so that same kernel can be used on normal and widescreen screens neatly
Problems hit against

- Forced to switch to X.org from Xfree86 (widescreen modes, new chipsets)
  - pkgsrc-based for NetBSD 4.0
    - Actually very painless
    - Made module generation much easier as provides a clear list of components and their dependencies
    - Constantly moving target
    - Module size increased (5.7MB vs 3.7MB) – mainly due to including more fonts and including more libraries
  - Now included in main system for NetBSD 5.0
    - Updated less often
    - Supports DRM
    - Easy to cross-build
    - v2.08 still uses pkgsrc version (will switch back later)
Problems not easily solved

- pkgsrc not designed for embedded system
  - Large dependency lists (e.g. gtk2+ is 33+ MB, got this down to 3.5MB ThinIT module)
  - Options framework not widely used enough
  - Packages include everything needed for development (e.g. header files, static libraries)
  - Mitigated by Makefile.vnd described earlier
Problems not easily solved

- Dependence on Linux emulation for Citrix ICA and Opera
  - Means tied to x86
  - NetBSD 5.0 does not support Linux 2.6 emulation
    - Unable to use latest Citrix ICA client
  - Opera do not want to pursue a native version of their browser at this point in time
    - After more work shrinking module sizes, we might try switching to Firefox (or Google Chrome?)
- Precedence is a Citrix Global Alliance Partner
  - Good reputation within Citrix UK means we might get source ($20k last time)
Exciting NetBSD developments

- PUFFS/refuse
  - ntfs-3g (r/w NTFS) useful for dual booting

- Lots of cool filesystems that could be exported to ICA and RDP sessions
  - gphotofs (PTP cameras that are not mass-storage)
  - ntfs-3g (access local data on dual boot machine)
  - obexfs (mobile phones)
  - psshfs (mount remote data over ssh)

- Could be used as basis for amd replacement
  - Would allow NFS to be removed from ThinIT kernel
  - amd is overkill for mounting USB pen drives on demand
Exciting NetBSD developments

- **RUMP (Runnable Userspace Meta Program)**
  - Allows kernel code to be compiled into userland program
  - Fantastic for debugging (single step with gdb)
  - A good way to access untrusted filesystems
    - e.g. `rump_ffs /dev/sd0e /mnt`

- **Video4Linux2 framework**
  - `uvideo(4)` allows access to modern webcams
  - Much better video conferencing and streaming
Exciting NetBSD developments

- Journaling FFS (WAPBL)
  - Currently use FFS on flash storage mounted read-only; switch to read-write to save settings then switch back
  - ThinIT machine should be able to be switched off any time. Currently small window where filesystem would be unclean (esp. during update process)
  - WAPBL avoids the need for fsck and switching between r/o and r/w

- HAL to detect device insertion automatically
  - Core part of netbsd-desktop project
Future developments for ThinIT

• Reduce reliance on Ethernet
  – 3G integrated or over Bluetooth
  – OpenVPN

• Network 'master' mode
  – One client provides services for others on LAN
  – Need DHCP, NAT, OpenVPN, firmware updates
    (bozohttpd in 5.0 good for this)

• Support more esoteric peripherals
  – Interactive whiteboards used widely in education
Future developments for ThinIT

- Get “Citrix Ready” certification
  - Just need to fix a problem with Linux emulation

- Switch to gtk2

- Rewrite local client management so can be run on Windows for remote configuration

- Get DRM/Xv working for better video playback

- Track NetBSD developments (of course)
Improvements since Sep 2007

- “Slow releases. 4.0 very late. When 5.0?”
  - Put 4.0 down to experience
  - 5.0 releng process much slicker and better managed
  - (seems churlish to complain too loudly, there's only been 3 ThinIT releases in that time too)
Improvements since Sep 2007

- “Too many developers working at the cutting edge without backporting”
  - Because 4.0 release process was so slow, -current was too different
  - Excitement about the huge step forward with 5.0 drives development; more developers are running it
Improvements since Sep 2007

- “Difficult to fund development”
  - TNF employed ad@ full-time for SMP development with great results.
  - New, expanded board and successful fund raising means more focus
  - GSoC is potentially a good model for funding small development
Improvements since Sep 2007

- “BSD licence clearly better than GPL, but advertising clause is difficult to support (list of names larger than the advert!”

  - NetBSD has switched to 2-clause BSD licence

  - Actively contacting other contributors to ask them if they are willing to remove their names
Conclusion

- NetBSD is excellent for embedded work
  - Quick to develop on
  - Clean code
  - Powerful bulk and cross-building tools
- BSD as whole ideal for product development
  - Commercially friendly licence
  - Integrated kernel/userland
- Has familiar problems seen throughout OSS
  - Lack of device drivers for new hardware
- NetBSD 5.0 is a huge leap forward for embedded and traditional desktop/server systems
ThinIT – lose the wait

- http://www.fastclients.co.uk/
- http://www.thinit.info/