

virt. linux auf mikrotik RouterBoard

Chancen und Grenzen der Virtualisierung ;)

Thomas Osterried DL9SAU

27. IPRT Darmstadt, 2011

2.4.2011

Lizenz: CC-BY-SA

Version 2011-04-02-02-final

Motivation

- Energie sparen
- Einfache und übersichtliche technische Infrastruktur
- Weniger Wartungsaufwand
- Bewährte Hardware für Routing im HAMNET für weitere Dienste nutzen
 - insbesondere jener, die nicht oder rudimentär im RouterOS implementiert sind
- Letztes Jahr hier erwähnt. Doch es sind keine Projekte entstanden.
 - Also geht's heute ins Detail
- Weil es Spaß macht ;)

Hardware

- Mikrotik Routerboard RB450G
 - besonders leistungsstark
 - 256 MB RAM
 - 520 MB Flash
 - 5x GBit Ethernet
 - mikro-SDHC Slot für Speicherkarten
 - Architektur: CPU MIPS 24k, 680 MHz.
BIG-Endian
- Andere RB4xx-Boards haben CPU selben Typs, d.h. auch MIPS, auch BE.

Andere OS-Software..

- Linux flashen
 - Pro: Board „befreien“
 - Obwohl RouterOS linux-basiert ist, hat man keine Root-Rechte auf dem System
 - Con: schade drum.
 - Man verliert ein leicht administrierbares System, welches Routingprotokolle beherrscht - nebst vielen anderen Features, die out-of-the-box funktionieren

..Andere OS-Software

- Schon über den Menu-Punkt „MetaROUTER“ gewundert?
 - Auf „+“ gedrückt.
 - Ein weiteres RouterOS startet. Gedacht für Routing-Tests, komplexe Konfigurationen, ..
 - „Import Image“
 - Dateiname und zugew. RAM als Auswahl

Nach Dokumentation gesucht:

<http://wiki.mikrotik.com/wiki/Manual:Metarouter>

Linux booten

- Auf o.g. Webseite gibt es ein „fertiges“ OpenWRT basiertes Linux.
 - download, upload auf routerboard per ftp, Image importieren, fertig
 - Lust auf mehr, z.B:
 - AX.25-Unterstützung einbauen
 - weitere Programme compilieren und installieren
- Selbst übersetzen
 - Der Patch von Mikrotik für den Kernel gelingt fehlerlos auf OpenWRT svn-Entwicklerzweig

OpenWRT..

- Build-Umgebung
 - 4 GB groß
- make menuconfig
 - weitere Auswahl treffen, was noch mit in Kernel und Package mit rein soll
- Compilierung dauert, und ist irgendwann fertig
- In trunk/bin/mr-mips/ liegt nun:
openwrt-mr-mips-rootfs.tar.gz
→ hochladen in das Routerboard, per ftp

..OpenWRT

- Programme aus trunk/bin/ auf einen Webserver kopieren. /etc/opkg.conf anpassen, z.B.

src/gz snapshots <http://www.osterried.it/openwrt/snapshots/trunk/mr-mips/packages>

- Wunsch, weitere Programme zu installieren.
 - Das Lieblingsprogramm gibt es nur für OpenWRT mips**le**, aber nicht für mips**be**. Kleiner aber feiner Unterschied.
 - Keine kompatible OpenWRT mips**BE** Distribution verfügbar
 - Also selber compilieren.

..OpenWRT..

- Jetzt geht das Drama los mit Cross-Compilieren.
- Die Welt ist nicht so rosig wie sie anfangs scheint
 - FPU Probleme, Segmentation Faults, Data Corruption, Kernel-Oopses
 - Schnell stellte sich die Frage:
 - kann ich das auf der IPRT überhaupt empfehlen?

Gast-Linux

- Das ist keine Kritik an OpenWRT (s. geänderter Titel)
- Es ist auch nicht schuld an den o.g. Fehlern
- OpenWRT ist ulibc basiert
 - wie wir nacher sehen, ein System mit Web-Server, conversd-saupp, sshd benötigt
 - 10 MB RAM
 - 10 MB Plattenplatz
- Ein debian-mips läge um ein Vielfaches höher

Der debian-mips Versuch..

- Ich wollte nicht Stunden mit cross-compile verbringen
 - dafür lernte ich eine Menge Fehler kennen
 - gleich vorweg: **erfolglos**
- debian-mips sollte nach /srv/debian-mips installiert werden. Dann chroot dorthin. Unseren compilierten Kernel wollen wir beibehalten.
- Später wenn alles funktioniert richtig installieren

..Der debian-mips Versuch..

- Vorbereitung

- `mkdir /srv/debian-mips; mkdir /srv/debian-mips`
- `/usr/sbin/debootstrap --foreign --arch mips squeeze \`
`/srv/debian-mips http://ftp.de.debian.org/debian`
- Auf mikrotik kopieren nach `/srv/debian-mips`
- `chroot /srv/debian-mips`

- **FAILED**

- Begründung

- CPU hat keine Floating-Point-Unit (FPU)
- Alle OpenWRT Binaries waren mit der Option „*soft-fpu*“ kompiliert.

..Der debian-mips-Versuch..

- Lösung
 - Kernel neu compilieren
 - CONFIG_MIPS_FPU_EMU=y
 - Nachteil: keiner.
Nun laufen userspace soft-fpu Binaries und auch normale debian-mips binaries gleichermaßen.
- Alles wieder hochladen. Nochmal versuchen.
 - chroot klappt
 - debootstrap/debootstrap –second-stage

..Der debian-mips Versuch..

- dpkg will nicht
 - dpkg nutzt den System call fsync(), der offenbar fehl schlägt
 - Hintergrund:

Der Virtualisierer (RouterOS) reicht file-Operationen des Gastes auf das reale Filesystem in durch.
Der Gast sieht ein Filesystem namens „metafs“ - kam mit dem mikrotik Kernel patch mit.
fsync() ist zwar implementiert. Dem Grund des Scheiterns muß man noch nachgehen.

..Der debian-mips-Versuch..

- Hack
 - `mv /var/lib/dpkg{,.orig}; mkdir /var/lib/dpkg`
 - `mount -t tmpfs tmpfs dpkg`
 - `cp -rpa /var/lib/dpkg{.orig,}`
[..und später bei Erfolg wieder zurückkopieren]
 - Erfolg?
 - Prinzipiell ja. Doch: dpkg bricht wieder ab
 - **Kein Erfolg**

..Der debian-mips Versuch..

- Während des Installationsprozesses verschwand /var/lib (!), und awk war da, konnte aber nicht mehr gelesen werden (I/O error)
 - Nach reboot ist alles wieder da (!)
 - Reproduzierbar
 - **Letztlich kein Erfolg**
- Erklärung
- Große I/O Operationen im Filesystem des Gastes führen zu Data Corruption
 - Fehler entweder im metafs des Gastes oder in der Disk-I/O Weiterleitung des Wirtes.

..Der debian-mips Versuch..

- Fazit
 - Wenn Fehler im Wirt (RouterOS), dann ist der nicht korrigierbar, da Closed Source
 - Showstepper
 - so will man kein BBS laufen lassen
- Das Linux selbst war stabil. für einfache Anwendungen also geeignet.
- Die Instabilität des Linux (auch bei kernel oopses) hat den Wirt nie beeinträchtigt (d.h. kein manueller Neustart nötig gewesen)

..Der debian-mips Versuch..

- `rm -rf /srv/debian-mips` musste mehrfach laufen, denn auch hier zeigen sich die Probleme mit dem Dateisystem
- Versuch: Vielleicht kann er ja in einer Datei arbeiten. Mit `dd image` erzeugt, `mke2fs` auf das image, `loop mount`.

→ Filesystem zeigt Inkonsistenzen!

- I have no name!@OpenWrt:/# `dmesg|tail -3`
 - EXT2-fs error (device loop0): ext2_readdir: bad page in #53111
 - EXT2-fs error (device loop0): ext2_readdir: bad page in #53111
 - EXT2-fs error (device loop0): ext2_readdir: bad page in #53111

..Der debian-mips-Versuch

- Anderer Versuch: network-block-device
 - Daten sollen vom Server kommen
 - soll zeigen daß der vfs-Treiber im Kernel wenigstens funktioniert
 - chroot scheitert mit segfault
→ schade
- Nach vielen Stunden Arbeit (die eigentlich Zeit des Cross-Compiles sparen sollte) wieder zurück zu OpenWRT, damit ich wenigstens etwas präsentieren kann.

back to OpenWRT

- Wir haben viel gelernt
- Bis Fehler eingegrenzt und beseitigt, nur limitierter Nutzen
- Auf dem mitgebrachten System haben wir
 - kernel mit AX.25 Unterstützung
 - conversd-saupp
 - crosscompile war sehr einfach
 - einfachen Web-Server
 - bringt die build-Umgebung mit

Das vorgestellte System..

- Gern hätte ich gezeigt
 - dns-Server (bind9)
 - als primärer DNS der Region im HAMNET, konfiguriert mit Zonetransfer zum zugehörigen DNS-HUB
 - Problem: eine Vielzahl von Library-Abhängigkeiten
 - ax25-apps und -tools; libax25
 - Problem: ein zeitliches – Handarbeit. Man kann bei cross-compile nicht einfach gegen die libax25, welche nicht unter /var/lib liegt, linken.

..das vorgestellte System

- avahi
 - aka: mdns, zeroconf, rendezvous, bonjour
 - TIP für's HAMNET
- „Richtiger“ Webserver (apache)

Ausblick

- Cross-Compile von
 - BBS Software, dx-cluster
 - xnet, tnn, wampes 1)
 - APRS Anwendung 1)
 - funkrufmaster 1)
 - Asterisk als VOIP Vermittlungsstelle zwischen FM-Relaisfunkstellen
 - zu 1) der rb450g hat eine serielle Schnittstelle, die an den Gast weitergeleitet werden kann. RPC, TNC f. APRS oder Packet-Einstieg oder RMNC lassen sich hierüber bedienen.
 - Andere Anwendungsbereiche
 - raduis, ldap, kerberos

tnx

Vielleicht sehen wir auf der nächsten IPRT
schon die ersten portierten
Amateurfunkanwendungen
für mipsBE

vy 73,

- Thomas Osterried dl9sau
<dl9sau@darcd.de>

Anlagen

- Kernel Optionen
 - Editiere `target/linux/mr-mips/config-2.6.31`
 - `CONFIG_MIPS_FPU_EMU=y`
 - `CONFIG_BPQETHER=m`
 - `CONFIG_MKISS=m`
 - `CONFIG_6PACK=m`
 - Nacher `build_dir/linux-mr-mips/linux-2.6.31.10/.config` kontrollieren

- System

- root@OpenWrt:~# free

-
-
-
-
-
-

	total	used	free	shared	buffers
Mem:	62652	9132	53520	0	0
Swap:	0	0	0		
Total:	62652	9132	53520		

- root@OpenWrt:~# df -h

-
-
-
-

Filesystem	Size	Used	Available	Use%	Mounted on
/dev/root	455.6M	10.8M	444.8M	2%	/
tmpfs	30.6M	8.0K	30.6M	0%	/tmp
tmpfs	512.0K	0	512.0K	0%	/dev

Anlagen

- Virtualisierte Instanz etwas langsamer
 - Mikrotik Wirt
 - # ping 192.168.88.1
 - PING 192.168.88.1 (192.168.88.1) 56(84) bytes of data.
 - 64 bytes from 192.168.88.1: icmp_req=1 ttl=64 time=0.382 ms
 - 64 bytes from 192.168.88.1: icmp_req=2 ttl=64 time=0.335 ms
 - 64 bytes from 192.168.88.1: icmp_req=3 ttl=64 time=0.338 ms
 - Linux Gast
 - # ping 192.168.88.254
 - PING 192.168.88.254 (192.168.88.254) 56(84) bytes of data.
 - 64 bytes from 192.168.88.254: icmp_req=1 ttl=64 time=1.08 ms
 - 64 bytes from 192.168.88.254: icmp_req=2 ttl=64 time=1.02 ms
 - 64 bytes from 192.168.88.254: icmp_req=3 ttl=64 time=1.27 ms

Anlagen

- **Compilierungs-Tips**

- **crosss-compile:**

- <http://www.frontiernet.net/~beakmyn/CrossCompile.htm>
 - `root@x2go-ubuntu:/home/thomas/openwrt-svn--fuer-mips/software/conversd-saupp/conversd-saupp# .
../../trunk/xcompile.txt`

- **conversd-saupp:**

- Dinge ausklammern die wir lokal nicht unbedingt brauchen:
 - `#READLINE_INCLUDES = -DHAVE_LIBREADLINE
-DHAVE_READLINE_READLINE_H`
 - `#READLINE_LIBS = -lreadline -lncurses`

-

-