DNS mit Bind9

Wolfgang Dautermann
FH Joanneum
wolfgang.dautermann@fh-joanneum.at

14. Mai 2005

Übersicht

DNS - Domain Name System: Geschichte und Konzepte von DNS.

Installation von Bind9: Installation von Bind9

Die eigene Domain: Eine eigene Zone (authorativ) wird konfiguriert. Konfigration als Master oder Slave

Reverse DNS: Wie wird einer IP ein Name zugeordnet

Weitere Ressource Records: Was gibt es sonst noch...

Dynamische Updates: Wie gehts, was ist zu beachten.

Geschichte

- Ursprünglich: `/etc/hosts.txt` (vgl. /etc/hosts), auf einem zentralen Master-Server upgedatet, download von allen Rechnern im Internet. Zunehmend unhandlich.
- 1983 Spezifikation von DNS, erster DNS-Server (Jeeves).
- Anfang der 80er Jahre: Entwicklung von Bind (Berkeley Internet Domain System)
- 1997 Bind Version 8
- Heute: Bind 9.3.1 (ISC)

Konzepte

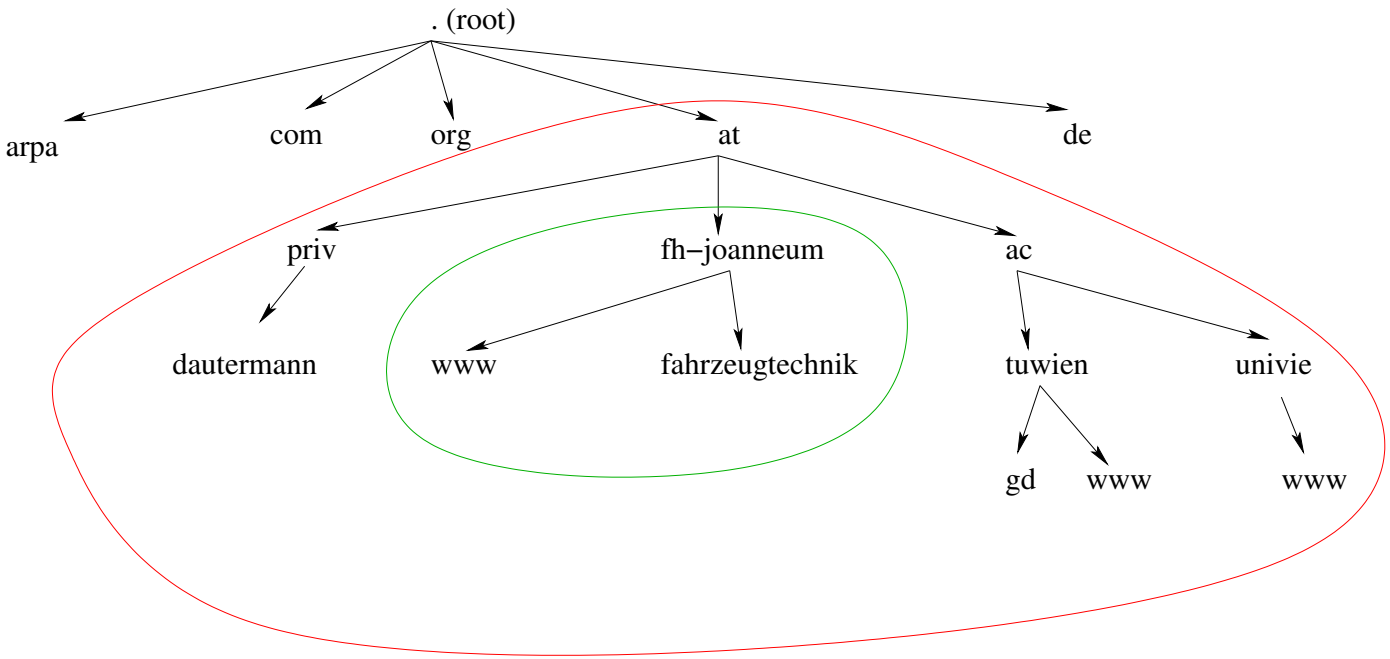

Abbildung 1: hierarchische Domain-Struktur

Konzepte

- "." wird verwaltet von ICANN (Festlegung Top-Level-Domains, Betrieb der 13 Root Nameserver.
- ".at" delegiert an und verwaltet von nic.at (Österr. Registry).
- ".priv.at" (Private Domains, gratis) delegiert an und verwaltet von www.nic.priv.at (www.vibe.at)
- ".fh-joanneum.at"delegiert an und verwaltet von der FH Joanneum.

Ausfallsicherheit

In der Regel mehrere DNS-Server / Zone erforderlich.

- Root-Zone: 13 Server, weltweit verteilt.
- AT-Zone: 9 Server
- Second Level Zonen: mindestens zwei Server (Master/Slave), Konfiguation am Master, Slave übernimmt Konfiguration "automatisch".

Schützt vor Nichterreichbarkeit durch Ausfall des DNS-Servers.

Ausfallsicherheit - Beispiele

```
$ host -t ns at.
at name server ns2.univie.ac.at.
at name server ns1.univie.ac.at.
at name server ns9.univie.ac.at.
at name server sss-nl.nic.at.
at name server ns-us1.nic.at.
at name server ns-uk.nic.at.
at name server sss-us2.nic.at.
at name server sss-jp.nic.at.
at name server ns-de.nic.at.
$ host -t ns fh-joanneum.at
fh-joanneum.at name server dallas.fh-joanneum.at.
fh-joanneum.at name server denver.fh-joanneum.at.
```

Installation

- `rpm`, `yast`, `apt-get`, `pkg-get`, ... oder:
- Download der aktuellen Version (dzt. 9.3.1) von http://www.isc.org/sw/bind/ (Dabei sind u.a. relevante RFCs, B9vARM (BIND 9 Administrator Reference Manual))
- `./configure ; make ; make install`

Installiert werden: Bind-server (named), DNS-Client-tools (nslookup, dig, host, nsupdate,...), Dokumentation (Manpages), Libraries, Include-Files, Admin- und Diagnosetools (named-checkconf, dnssec-keygen, ...)

Konfiguration

Konfigurationsdateien von Bind9::

- Globale Konfigurationsdatei (/etc/named.conf)
- Zonendateien (1 pro Zone)
- ev. weitere Dateien (*.key), ...

Konfiguration - named.conf

"named.conf" besteht aus:

- Globalen Optionen: Zugriffssberechtigungen, Krypto-Keys und Optionen
- (ev.) Server-Liste: Informationen über Partner-Server
- Zoneliste: ein Eintrag / Zone

```
options {
    directory "/var/named"; # Arbeitsverzeichnis fuer Bind
    forwarders { 62.218.221.2; 62.218.221.1; };
    // 195.113.31.123 = http://atrey.karlin.mff.cuni.cz/~mj/sleuth/
    allow-transfer { 195.113.31.123 ; } ;
```

```
        controls {
              allow { 127.0.0.1; } keys { "rndc-key"; };
        };

    key "rndc-key" {
       algorithm hmac-md5;
       secret "streng-geheim==";
    };
};

# Drei Zonen fuer den Bind
# Zone fuer Localhost
zone "localhost" in { type master; file "localhost.zone"; };
# Reverse Lookup fuer Localhost
zone "0.0.127.in-addr.arpa" in { type master; file "127.0.0.zone"; };
# Zone fuer die Root-Server
zone "." in { type hint; file "root.hint"; };
```

```
// add entries for other zones below here

zone "example.org" {
        type master;
        file "example.org"; // oder example.org.zone, ...
};

zone "example.com" {
        type slave;
        masters { 123.123.123.123 ; } ;
        file "example.com";
};
```

Ressource Records oder: was steht in den Zonefiles

- SOA - Record (Start of Authority)
- NS - Records (Nameserver)
- A - Records (Zuordnung Name $\rightarrow$ IP-Adresse)
- CNAME - Records (Aliases)
- MX - Records (Mail Exchanger)
- PTR - Record (Zuordnung IP-Adresse $\rightarrow$ Name)

Beispielzone - localhost

Ein erstes Beispiel - die Zonendatei für localhost:

```
$TTL    604800          ; Time to live
// SOA Record
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
// NS Record
@       IN      NS      localhost.
// A Record
@       IN      A       127.0.0.1
```

SOA Record

Name der Zone (abgekürzt durch "@"

TTL (optional) gibt an, wie lange dieser Eintrag im Cache gehalten werden darf

IN Klasse; üblicherweise INternet.

SOA Recordtyp: SOA Record

Primary Primary NS für diese Zone

Mailaddr. des Verantwortlichen für diese Zone

Serienr.	wird bei jeder Änderung inkrementiert
Refresh	Sekundenabstand in den die Slaves anfragen, ob sich etwas geändert hat
Retry	Sekundenabstand in denen ein Slave wiederholt, falls sein Master nicht antwortet
Expire	wenn der Master auf einen Zonentransfer-Request nicht reagiert, deaktiviert ein Slave nach dieser Zeitspanne in Sekunden die Zone
TTL	negativ-Caching-TTL

A Record: Zuordnung DNS-Name → IP-Adresse.

TTL gibt an [in Sekunden], wie lange dieser Resource Record in einem Cache gültig sein darf

IN Klasse. (Internet)

A Recordtyp: A-Record

IP IP(v4) Adresse

Beispiel:

```
www.example.com.   3600  IN  A  12.34.56.78
```

A Records: Lastverteilung per DNS.

Es dürfen mehrere A-Records zu einem Namen existieren, diese werden in wechselnder Reihenfolge zurückgeliefert.

```
www.example.com.    3600  IN  A  12.34.56.78
www.example.com.    3600  IN  A  12.34.56.79
www.example.com.    3600  IN  A  12.34.56.80
```

Dadurch ist eine einfache Lastverteilung auf mehrere Server möglich.

NS Record

TTL gibt an, wie lange dieser RR in einem Cache gültig sein darf

IN Internet

NS

Server Name des für diese Domäne autoritativen Nameservers

Beispiel:

```
example.com.  1800  IN  NS  ns1.example.com.
example.com.  1800  IN  NS  ns1.example.com.
```

Es *müssen* Namen angegeben werden - keine IPs.

NS Record - Zonendelegation.

Zonendelegation:

```
subdomain.example.com.   1800  IN  NS  ns1.subdomain.example.com.
subdomain.example.com.   1800  IN  NS  ns2.subdomain.example.com.
```

Damit ist für die Auflösung von

`irgendwas.subdomain.example.com.`

nicht mehr der Nameserver `ns1.example.com.` sondern (z.B.) `ns1.subdomain.example.com.` zuständig.

Glue-Records

Problem: Die Katze beisst sich in den Schwanz: Zuständig für die Auflösung (IP!) von `ns1.subdomain.example.com.` sind die DNS-Server von `subdomain.example.com.` (und um dessen IP zu erfahren fragen wir am besten den DNS-Server von `subdomain.example.com`...

Lösung: Glue-Records. Der A-Record für `ns1.subdomain.example.com.` ist zusätzlich(!) in der übergeordneten Zone (example.com.) eingetragen.

CNAME, TXT

Ein CNAME ist ein Alias. Beispiel::

```
www     1800  IN CNAME server
```

TXT - ein frei definierbarer Text:

```
owner   1800  IN TXT "Hello World"
```

Wird z.B. verwendet für SPF:

```
      IN TXT "v=spf1 ip4:12.34.56.78 -all"
```

Email - MX Records

```
example.com     1800  IN MX 10 mailserver.example.com.
example.com     1800  IN MX 20 mailserver.backupdomain.com.
```

Ein oder mehrere für die Domain zuständige Mailserver. Zusätzlich eine Priorität - die niedrigere wird zuerst probiert. Falls kein MX vorhanden ist, versucht der Mailserver den A Resource Record (die IP-Adresse) der Domain festzustellen. Falls der Mailserver die IP-Adresse ermitteln kann, versucht er eine SMTP-Verbindung zu dieser IP aufzubauen.

Reverse DNS

Ich kenne eine IP Adresse. Wie heisst der Server, der sich dahinter verbirgt? (Die Antwort ist in der Regel nicht eindeutig (virtuelle Hosts, Lastverteilung, ...)).

Dazu gibt es die **in-addr.arpa**-Domäne.

Subdomains sind für das Reverse DNS der IP-Adressen zuständig. Die Zone `10.in-addr.arpa` enthält die IP-Adressen von 10.x.y.z, die Zone 168.192.in-addr.arpa enthält die IP-Adressen von 192.168.x.y, ... In diesen Zonen werden mittels PTR-Records

```
1.0.186.192.in-addr.arpa.  1800   IN  PTR   server1.example.com.
```

Der korrespondierende Eintrag der Domäne example.com sieht dann folgendermaßen aus:

```
server1.example.com.        1800   IN  A      192.168.0.1
```

Delegiert werden kann natürlich auch:

```
0.186.192.in-addr.arpa.  1800   IN  NS   server2.example.com.
```

delegiert das Subnetz `192.168.0.XXX` an den DNS-Server `server2.example.com`.

Nachteil: Das funktioniert nur vernünftig an 8-Bit-Grenzen.

Dynamische Updates

Werden im Zonefile deklariert. Beispiele:

```
zone "update1.example.com" {
        type master;
        file "update1.example.com";
        allow-update { 12.34.56.78 ; } ;
};
zone "update2.example.com" {
        type master;
        file "update2.example.com";
        allow-update { key key.example.com ; } ;
};
```

Dynamische Updates

Updates können dann mit dem Befehl `nsupdate` durchgeführt werden. Beispiel:

```
# nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 1.2.3.4
>
```

.

Problem: Replay-Attacke.

Dynamische Updates

Updates können dann mit dem Befehl `nsupdate` durchgeführt werden. Beispiel:

```
# nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 1.2.3.4
>
```

.

Problem: Replay-Attacke.

Master/Slave-Kommunikation

Slave holt sich die aktualisierte Konfiguration entweder in Zeitabständen (Refresh) (wenn die Serial-Number erhöht wurde!) oder automatisch via Notify: Der Master benachrichtigt alle Slaves einer Zone, sobald sich in der Zone etwas geändert hat. Entweder vollständiger Zonetranfer oder inkrementeller Zonetransfer.

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)
- DNS wird ständig weiterentwickelt.