Von: Hans-Jürgen Barthen DL5DI [mailto:dl5di@gmx.de]

Gesendet: Mittwoch, 26. Mai 2010 20:12

Betreff: ircDDB Installationsinfo allgemein

Unsere Software benötigt auf dem Gateway eine Software, die die Heard-Informationen an eine Datenbank weiterleitet.

Wir haben im ersten Schritt dazu DstarMonitor von AE5PL eingesetzt.

XMonitor sollte diese Aufgabe mit leichten Anpassungen auch übernehmen können, uns stehen nur die Sourcen nicht zur Verfügung um es anzupassen.

Genauso könnten die LastHeard-Informationen (Rufzeichen und Einstiegs-Repeater-Modul mit Zeitstempel) auch aus anderen Applikationen bereitgestellt werden.

Eine Anpassung des Interfaces auf der Datenbankseite des ircDDB-Systems ist jederzeit möglich.

Wir gehen zunächst von dem aus, was wir haben, DSM.

DStarMonitor ist normalerweise unter /opt/dstarmon installiert.

Das ist nur wichtig, weil sich dstarmon und unser Tool den Postgres-Java-Connector teilen und der Pfad in dessen Config auch drinsteht, also ggf. angepasst werden müsste wenn man das anders installiert.

In den Properties für DSM muss ein Treiber hinzugefügt werden und zwar nur der zum Schreiben der MHeardTable.

Wie das im Detail geht steht in der Doku von Pete mit drin (ist auch beigefügt, musst Du aber nicht unbedingt studieren).

Diese Tabelle bekommt dann nur 3 Felder, die Du auch in Michaels Sheet findest:

target_cs, last_mod_time und arearp_cs

XTrust-Benutzer sollten das Default-Reporting an dstarusers.org rausnehmen.

Einige XTrust-Systeme haben DstarMonitor bereits vorinstalliert, dort muss nur in den Properties eine Datenbank hinzugefügt werden.

In dem anhängenden Beispiel ist nur die Kommunikation an unseren Server drin, nicht nach dstaruser.org ! So hab ich das auch bei DBOFOX installiert.

Die Datei sieht dann so aus:

```
stderr=/var/log/dstar/dstarmon.log
GWIntf=eth1
CtrlrIP=172.16.0.1
#
LHDriver=org.postgresql.Driver
LHURI=jdbc:postgresql://141.75.245.225:5432/allstarrouter
LHParameters=user;dm0px;password; 0aNliavqAAg1LD1p
#
```

Anschließend folgen noch einige Einträge für xAPRS, die für uns keine Rolle spielen.

In den Pfad muss dann noch den Postgres-Connector kopieren, er ist im o.a. Archiv unter /opt/ircDDB/install enthalten (postgresql-8.4-701.jdbc3.jar).

Gestartet wird DSM dann mit dem Script monstart.sh, wie es im Archiv enthalten ist.

Nun zum ircDDB-Client, der Software, die sich an den IRC-Server hängt und die dort kommenden Daten in die lokale Datenbank einarbeitet.

Beim ersten Connect lädt die sich auch die Liste vom Server runter, bei Verbindungsunterbrechungen die seit dem letzten Login angefallenen Daten, so dass lokal immer alle Daten aktuell sind.

Nur als Anhalt, was sicherlich das Updaten im Bedarfsfall einfacher macht wenn wir das vereinheitlichen:

1. Lege einen Benutzer „ircddb“ an mit Home-Directory „/opt/ircDDB“
2. Entpacke die Software dahin.

Das sollte dann so aussehen:

```
[root@db0myk-gw ircDDB]# ls -l
insgesamt 136
-rw-r--r-- 1 ircddb root 862 24. Mai 21:48 ircDDB.keystore
-rw-r--r-- 1 ircddb root 338 24. Mai 22:18 ircDDB.policy
-rw-r--r-- 1 ircddb root 318 23. Mai 23:16 ircDDB.properties
-rw-r--r-- 1 ircddb root 10189 23. Mai 09:35 RpPtrApp.class
-rwxr-xr-x 1 ircddb root 890 25. Mai 18:53 run.sh
-rwxr-xr-x 1 ircddb root 51 25. Mai 19:23 start.sh
[root@db0myk-gw ircDDB]#
```

ircDDB ist ein allgemeines Projekt von Michael im Internet (<http://github.com/dl1bff/ircDDB>).

Policy und keystroke **sollen** sicherstellen, dass beim Autoupdate niemand etwas falsches unterschiebt.

Run.sh ist ein endlos laufendes Script, was den autoupdate-Mechanismus beinhaltet.

Der Client kann per Kommando über IRC beendet werden, sicher dann die letzte lauffähige Version, sucht dann nach einem Update, wenn verfügbar installiert er es und startet neu.

Das kannst Du natürlich ändern, bzw. anpassen, falls Du einen eigenen Client schreibst und den von woanders updaten willst.

Im IRC-Kanal kann man die Versionsnummer der Module aller Gateways abrufen (mit „/WHO *“).

Die Specs **werden** noch veröffentlicht.

Start.sh ist die Startdatei.

ircDDB wird vom Instalationsscript automatisch gestartet und beim Neustart des Systems von monstart.sh

Die Config-Datei ircDDB.properties sieht wie folgt aus:

```
irc_server_name=ircddb-server.mdx.de
irc_server_port=9007
irc_channel=#dstar

irc_password=*****
rpPtr_call=db0xyz

debug=0
rpPtr_fix_tables=yes
rpPtr_fix_unsync_gip=no

ext_app=de.mdx.ircDDBApps.RpPtrApp
```

```
jdbc_class=org.postgresql.Driver
jdbc_url=jdbc:postgresql://127.0.0.1:5432/dstar_global
jdbc_username=dstar
jdbc_password=icom
```

Oben die Daten im ersten Block sind für den Zugang zum IRC-Server.

Die Zugangsdaten zum IRC-Server sind die gleichen wie die zur Datenbank in den DSM-Properties (s.o.).

Dann kommen 2 Schalter, die Unterschiede zwischen den Systemen ermöglichen:

```
rptr_fix_tables=yes bzw. no
  bei "yes" werden eventuell fehlende Zeilen in sync_gip und sync_mng
  angelegt. Vorsicht: die Voreinstellung ist "no", wir sollten bei unseren
  Repeatern also "rptr_fix_tables=yes" eintragen. Bei Test-Benutzern
  ausserhalb unseres "Cross-Trust-Netzes" laesst man die Zeile einfach weg
  ;)
```

Das heißt, wer keine sync_gip hat, lässt den Eintrag weg.

Wir haben den bei Icom- und OpenG2 gesetzt, beide nutzen die sync_gip zum Erreichen der anderen Gateways.

```
rptr_fix_unsync_gip=yes bzw. no
  dieser Parameter wird nur dann benoetigt, wenn die "rptr_fix_tables"-Funktion
  aktiviert ist. Beim Wert "yes" wird zusaetzlich zu einer neu angelegten
  Zeile in sync_gip auch eine Zeile in unsync_gip eingetragen.
  Bei openg2 gibt es diese Tabelle nicht, deswegen muss
  z.B. bei DB0FOX "rptr_fix_unsync_gip=no" drinstehen. Bei den anderen
  Relais kann die Zeile entfallen, weil die Voreinstellung "yes" ist.
```

Das sollte selbsterklärend sein.

Ich denke mal, Du wirst die unsync_gip bei DMOPX genau wie bei DB0FOX nicht haben.

Das Abschalten verhindert dann Fehlermeldungen beim Schreiben in nicht vorhandene Tabellen.

Im unteren Block kommen dann noch die Zugangsdaten zur lokalen Datenbank mit den allseits bekannten Login-Daten.

Das müsstest Du dann ggf. anpassen falls Du die geändert haben solltest.

Kein root-Zugriff wie beim gw_schedule-Script, sondern wie es sein sollte über den vorgesehenen User-Account.

In der run.sh findest Du unten den Pfad zu dem Java-Postgres-Connector

```
CP="app.jar:ircDDB.jar:/opt/dstarmon/postgresql-8.4-701.jdbc3.jar"
```

Du siehst am Pfad, wir nehmen den unter dstarmon installierten Connector.

Das musst Du ggf. auch anpassen oder wenn Du was anderes als DStarMon nimmst den Connector ins Verzeichnis ircDDB kopieren.

Zum Schluss dann noch was ganz Wichtiges, die Postgres-Trigger-Funktion.

Diese Funktion hängt sich direkt in Postgres rein, fängt INSERTs und UPDATEs vom Trust-Server ab und ersetzt die darin enthaltenen Routing-Infos ggf. durch aktuellere Daten aus unserer Datenbank.

Installiert wird der Trigger als User postgres mit

```
psql -U dstar dstar_global < trigger.txt
```

Du findest das Textfile im Source-Directory.

Das sollte es dann hoffentlich auch gewesen sein.

Wenn es irgendwelche Probleme gibt [bitte einfach melden!](#)

Wie gestern schon besprochen soll das keinerlei Konkurrenz zu neuen Routing-Konzepten, neuen Trust-Ideen etc.. sein.

Das Ganze soll 2 Dinge tun:

1. Die größten Unzulänglichkeiten in Sachen Roaming und IP-Update der Gateways im US-Trust-System fixen, was Dir sicher egal sein kann.
2. Die Möglichkeit schaffen über alle Netzgrenzen hinweg per Callsign-Routing zu kommunizieren.

è das Ganze ohne in die Trust-Server-Angelegenheiten irgendwie einzugreifen.

Ob man später, wenn man in einem Netz ein neues Routingverfahren einführt, die Schnittstelle bei jedem Gateway belässt, oder am Trust-Server sowas anbietet und dort zur Netzkopplung nutzt, ist vollkommen offen und egal.

Wenn also jemand meint, sein Netz wäre ohne diese Kommunikationsebene schneller, kein Problem, nur bitte die Schnittstelle zum anderen System irgendwo belassen, ggf. zentral.

Es wäre schön, wenn wir das als gemeinsame Kommunikationsschnittstelle behalten könnten damit das DStar-Netz für den Benutzer wieder eins wird.

Damit hätte unsere Arbeitsgruppe ihr Ziel dann erreicht.

Ideen, Verbesserungsvorschläge, Hilfe bei der Doku und Installations-Script-Erstellung, alles willkommen!!

Die Quellcodes werden noch komplettiert und dokumentiert und kommen dann alle ins Web.

Die Form hast Du ja schon bei den Sheets gestern gesehen, garantiert 100% Open Source und CC.

73

Hans-Jürgen, [DL5DI](#)

SysOp @[DA5UDI](#), [DB0LJ](#), [DB0MYK](#)

<http://www.prgm.org/>

<http://afuforum.prgm.org>

Tel Shack: [02652 938377](tel:02652938377)

Skype: hjarthen