



Retiisi

Reaaliaikaiset Internet-palvelut ja SIP

Vikasietoinen SIP-ympäristö

Kirjoittajat:	Mika Mustikkamäki TYT/Wirlab
Kuvaus:	Vikasietoisen SIP-ympäristön kuvaus
Tiedostonimi:	Retiisi-Vikasietoisuus.pdf
Dokumentin tila:	Julkaistu
Versio:	1.2
Päiväys:	8/12/2005

SISÄLLYSLUETTELO

1.Johdanto	3
2.OpenSIPg- ympäristö	3
2.1.SER.....	3
2.2.FreeRADIUS.....	4
2.3.MySQL.....	4
3.Kahdennus ja klusterointi	4
3.1.Palveluiden kahdennus.....	4
3.2.Tietokannan klusterointi.....	6
3.3.Hallintanoodin pystytys.....	6
3.4.Storage-noodien pystytys.....	7
4.Palveluiden valvonta	9
5.Tulokset	12
6.Viitteet	13
7.Lyhenteet	14

1. Johdanto

Kun SIP-puhelut yleistyvät ja palveluiden taakse saadaan asiakasmassaa, nousevat palvelun tavoitettavuus ja laatu tärkeäksi kysymykseksi palveluntarjoajalle. Erityisen tärkeäksi tietoliikenneyhteyksien toimivuuden rinnalle voidaan nostaa SIP-välityspalvelimen sekä muiden palvelussa käytettävien sovellusten luotettavuus ja siten palvelun yleinen toiminnallisuus asiakkaan suunnasta katsottuna. Redundanssi, pienet vasteajat ongelmatilanteissa, sekä vikatilanteista toipuminen ovat oleellisessa osassa palveluita suunniteltaessa ja kehitettäessä. Tässä Retiisi-projektin dokumentissa kuvataan OpenSIPg-ympäristön, sen sovellusten ja resurssien kahdentamista ja varmentamista perustuen avoimen lähdekoodin ratkaisuihin.

Käytetyn OpenSIPg-ympäristön lisäksi tässä esitettävät toimenpiteet pätevät sellaisenaan myös mainittuihin vapaasti verkosta haettaviin sovelluksiin, eivätkä siten ole spesifiseen kaupalliseen tuotteeseen liittyviä. Dokumentti soveltuu niin Retiisi-projektiin osallisten, kuin muidenkin kahdentamisesta ja varmennuksesta kiinnostuneiden tahojen johdanto- ja esimerkkimateriaaliksi.

Dokumentissa on useita viittauksia käyttöjärjestelmäkomentoihin, sekä esimerkkejä hakemisto- ja tiedostolistauksista. Nämä viittaukset ja esimerkit on merkitty Courier-fontilla eri osioiden luettavuuden parantamiseksi.

2. OpenSIPg-ympäristö

OpenSIPg:n ydinosat muodostaa avoimen lähdekoodin sovellukset SER [1] SIP-välityspalvelin, FreeRADIUS [2] RADIUS-autentikointipalvelin, Bind-nimipalvelin [3] ja MySQL-tietokanta [4]. Näiden lisäksi palvelussa käytetään muita avoimen lähdekoodin sovelluksia, kuten MediaProxy:ä [5], joka hoitaa RTP-liikenteen välittämistä SIP-päätelaitteiden välillä, SEMS-sovelluspalvelinta [6] puhelinkonferenssien järjestämiseen, sekä kaupallisia OpenSIPg Manager- ja User-sovelluksia ympäristön hallintaan ja työkalujen tarjoamiseen loppukäyttäjille. Oleellisin osa kahdennuksessa on MySQL-tietokanta, jonka päällä SER, FreeRADIUS ja Bind pääasiassa toimivat. Projektin yhteydessä käytetty OpenSIPg-versio oli 2.0.

Tässä yhteydessä ei käydä läpi OpenSIPg-asennusohjeita, sillä manuaalit ja ohjeistus tulevat kaupallisen ohjelmistoversion mukana.

2.1. SER

SER hoitaa OpenSIPg-ympäristössä SIP-välityspalvelimen ja -rekisteröintipalvelimen tehtävät. Käyttäjät rekisteröityvät päätelaitteillaan välityspalvelimelle, joka tekee käyttäjäautentikoinnin RADIUS-rajapinnan kautta tietokannasta, ja tallettaa sen jälkeen kunkin käyttäjän sijaintitiedon (location) tietokantaan. Puheluissa SER hoitaa SIP-signaloinnin, mutta ei ole ns. stateful proxy, eli se ei ole tietoinen päätelaitteiden istuntojen välisestä dialogista. SIP-välityspalvelin ei myöskään käsittele varsinaista välitettyä RTP-mediaa, vaan media kulkee joko päätelaitteelta päätelaitteelle, tai niiden

välissä olevan MediaProxyn kautta. SER on IETF RFC3261 [7], s.o. Session Initiation Protocol v.2 yhteensopiva.

2.2. FreeRADIUS

RADIUS-palvelua käytetään SIP-asiakkaiden autentikoimiseen (so. tunnistamiseen) välityspalvelimelle ja näihin käyttäjiin liittyvän tiedon, esim. kuntakoodien, puhelutietojen sekä erilaisten attribuuttien tallentamiseen tietokantaan. FreeRADIUS toteuttaa IETF RFC:n 2865 [8], sekä IETF RFC:n 2866 [9] laskutustiedon välittämistä varten.

2.3. MySQL

MySQL-tietokannassa on talletettuna palveluun rekisteröityneiden asiakkaiden käyttäjätunnukset, salasanat, palvelun domain- ja nimipalvelutiedot (mukaanlukien ENUM), sekä Manager-sovellukset käyttäjätunnukset ja salasanat. Näinollen se on kriittisin palvelun osa, jonka toiminta ja saavutettavuus on järjestelmän ydin.

3. Kahdennus ja klusterointi

Kahdennuksella viitataan tässä yhteydessä palvelujen, kuten SIP-välityspalvelimen tai autentikointipalvelimen kahdentamiseen siten, että jos yhden palvelimen toiminta keskeytyy, voidaan ottaa käyttöön varapalvelin, joka on vastaavassa toiminnallisessa tilassa kuin kaatunut palvelin. Kahdennuksella pyritään saavuttamaan häiriötön ja katkoton palvelu, sekä mahdollisimman nopea palautuminen vikatilanteista. Kahdennusta suunniteltaessa on otettava huomioon mm. tilanteet, joissa palvelu voi keskeytyä, sekä mahdolliset "raja-arvot", joiden ylittyessä on syytä olettaa palvelun keskeytyneen ja pyrittävä tämän jälkeen saattamaan varapalvelut toimintaan niin nopeasti kuin mahdollista. Optimaalinen tilanne on silloin, jos primääri- ja sekundääripalvelimien välillä vaihto tapahtuu automatisoidusti ja käyttäjien huomaamatta vikatilanteiden sattuessa.

Klusteroinnilla viitataan tässä yhteydessä tietokantapalvelujen monistamiseen sillä tavoin, että relevantti tietokantasisältö on varmennettu reaaliaikaisesti ja kahdensuuntaisesti. Vikatilanteissa klusteroidut tietokannat otetaan käyttöön toimivilta klusterin koneilta, ja vikaantuneet palvelimet voidaan tällä aikaa huoltaa ja palauttaa sitten takaisin klusteriin. Tietokantataulut kopioituvat automaattisesti muutosten yhteydessä klusterin koneelta toiselle, jolloin kaikki data on redundanttia. Mikäli jokin laite (palvelin) vikaantuu, huoltokatkoksen aikana klusteroituun dataan tulleet muutokset kopioituvat ilman erillistä käskyä automaattisesti palvelimen tiedostojärjestelmään, kun se liitetään verkkoon ja klusteriprosessit ja tietokanta käynnistetään.

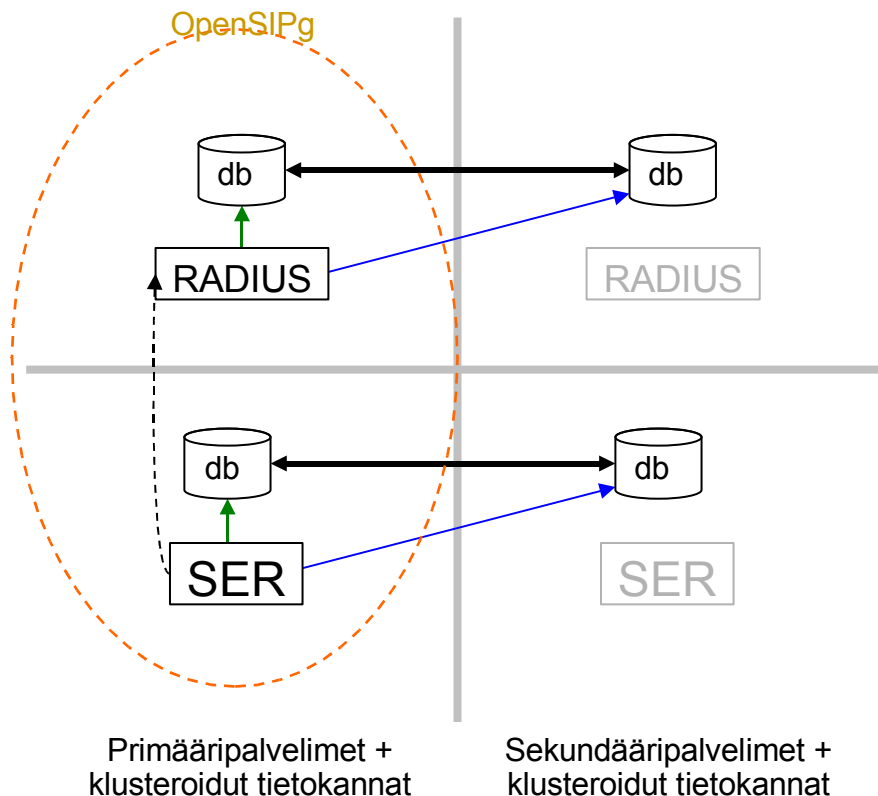
3.1. Palveluiden kahdennus

Ympäristö kahdennettiin SER-, RADIUS-, ja MySQL-palveluiden osalta yhteensä neljälle koneelle siten, että SER ja siihen liittyvä MySQL-tietokanta olivat kahdella koneella ja FreeRADIUS ja sen tietokannat kahdella erillisellä koneella. SER/MySQL-

koneilla oli käynnissä vain relevanteille palveluille (SER, MySQL, Bind, MediaProxy, tcpdump) tarvittavat prosessit ja konfiguraatiot oli muutettu siten, että SER osasi käyttää ulkopuolisella koneella olevia RADIUS-palveluja. RADIUS/MySQL-koneilla vastaavasti SIP-välityspalvelimen prosessit oli ajettu alas ja käynnissä oli vain FreeRADIUS ja MySQL.

Klusterissa mukana olleet koneet nimettiin nimipalveluissa host-nimillä siten, että SER-koneet toimivat osoitteilla xxxx11 ja xxxx12 ja kulloinkin aktiivisena olevaan välityspalvelimeen viitattiin osoitteella xxxxxx (CNAME). RADIUS-koneet olivat vastaavasti xxxx13 ja xxxx14, ja CNAME-tietue aktiiviselle oli xxxxxx. Valvontakoneelle luotiin skriptit, joilla nimipalvelua voitiin päivittää dynaamisesti vikatilanteissa osoittamaan toimiviin palvelinosoitteisiin (ks. tarkemmin kohta 4: Palveluiden valvonta).

Alla kahdennetun ympäristön periaatekuvaus.



3.2. Tietokannan klusterointi

Tietokannan klusteroinnissa käytettiin kahta ns. storage-noodia, sekä yhtä hallintanoodia per kahdennettava palvelu. Käytetyt MySQL-versiot olivat 4.1.9:stä 4.1.11:een (ympäristö päivitettiin kahteen kertaan projektin aikana). Klusterinoodit koostuivat sekä ndbd-klusteriprosessista että mysqld-tietokantapalvelinprosessista. Klusterin toiminta-ajatus on se, että storage-noodit tallentavat varsinaisen tietokantataulujen datan, sekä kommunikoivat muutokset toisilleen. Hallintanoodin olemassaoloa tarvitaan storage-noodien käynnistyksessä ja vikatilanteissa, joissa storage-noodit voivat kysyä tältä klusterin viimeisimmän tilan. Hallintanoodi tuottaa myös klusterin lokitiedot, joita analysoimalla voidaan selvittää mahdollisia ongelmatilanteita. Itse MySQL-palvelinprosessit käyttävät storage-noodien tarjoamaa klusteria "asiakkaina", eli pyytävät klusterin palveluita käyttöönsä tietokantakyselyjen yhteydessä. Tällä toiminnallisuudella päästään tilanteeseen, jossa klusteria käyttöön otettaessa ei tarvitse päivittää käytössä olevia tietokantasovelluksia, vaan ne näkevät yhteytensä tietokantaan kuten ennen klusterointiakin – MySQL-rajapinnan kautta.

Testiympäristön palvelimien levyt osioitiin siten, että jokaisella koneella oli vain yksi, ns. root-osio. Tuotantoympäristössä on syytä kiinnittää tarkempaa huomiota osiointeihin, ainakin /var-hakemistopuu on syytä ottaa käyttöön omana itsenäisenä levyosionaan. Tällöin laiterikkojen sattua on pienempi todennäköisyys, että kyseinen osio vahingoittuu, sillä MySQL-klusteri käyttää /var/lib/mysql-cluster-hakemistoa oletushakemistonaan. Siellä säilytetään logitiedot, konfiguraatiot, sekä tietokantojen levytiedostot.

3.3. Hallintanoodin pystytys

Hallintanoodin ndb_mgmd-binääri tulee mukana Debian-paketissa mysql-server-4.1. Binääri etsii oletuksena klusterin datahakemistoa ja konfiguraatiotiedostoa kohteesta /var/lib/mysql-cluster, joten se täytyy luoda. Retiisi-klusterin konfiguraatiotiedosto oli:

```
[NDBD DEFAULT]
NoOfReplicas= 2
```

```
[MYSQLD DEFAULT]
[NDB_MGMD DEFAULT]
[TCP DEFAULT]
```

```
[NDB_MGMD]
Id= 1
HostName= 192.168.0.10
ArbitrationRank= 1
```

```
[NDBD]
Id= 3
HostName= 192.168.0.11
DataDir= /var/lib/mysql-cluster
MaxNoOfOrderedIndexes= 256
MaxNoOfUniqueHashIndexes= 256
StopOnError= 0
```

```
[NDBD]
Id= 4
HostName= 192.168.0.12
DataDir= /var/lib/mysql-cluster
MaxNoOfOrderedIndexes= 256
MaxNoOfUniqueHashIndexes= 256
StopOnError= 0
```

```
[MYSQLD]
Id= 7
HostName= 192.168.0.11
```

```
[MYSQLD]
Id= 8
HostName= 192.168.0.12
```

Konfiguraatiossa kerrotaan klusteriin osallituvien storage-noodien määrä, niiden IP-osoitteet, klusterin datahakemistot, sekä kolme tärkeää parametria: `MaxNoOfOrderedIndexes`, `MaxNoOfUniqueHashIndexes` ja `StopOnError`. Kaksi ensimmäistä kasvattavat tietokantataulujen indekseille varattavan muistin määrää ja viimeinen kertoo noodille, ettei sen saa pysähtyä virhetilanteissa. Muistin kasvatus jouduttiin tekemään oletusasetuksiin, sillä OpenSIPg käyttää useampia primääri-indeksiavaimia tauluissa. `StopOnError`-parametri lisättiin, sillä storage-noodien huomattiin kaatuvan aina, mikäli ne menettivät yhteyden (heartbeat) joko toisiinsa tai hallintanoodiin.

3.4. Storage-noodien pystytys

Storage-noodeissa kerrotaan `/etc/mysql/my.cnf` -tiedostossa parametri, joilla `mysqld` pääsee kiinni hallintanoodiin.

```
[mysql_cluster]
ndb-connectstring= 192.168.0.42
```

Lisäksi kerrotaan erillisellä parametrilla klusterin käyttöönotto.

```
[mysqld]
ndbcluster
```

Klusteroitavien tietokantataulujen moottorina täytyy olla joko `ndbcluster` tai sen alias `ndb`. Olemassa olevien tietokantataulujen lisääminen klusteriin tapahtuu `alter table` -komennolla MySQL:n promptista – uudet taulut voidaan luoda oletusarvoisesti `ndbcluster`-moottorille.

```
ALTER TABLE admin ENGINE=NDB;
```

Ensimmäisellä käynnistyskerralla storage-noodien `ndbd`-prosessille täytyy antaa `-initial` -parametri, jolloin tietostojärjestelmään luodaan tarvittavat hakemisto- ja tietostorakenteet tietokantoja ja lokitiedostoja varten. Mikäli `ndbd` käynnistyy oikein, prosessilistauksella (esim. `ps ax |grep ndbd`) pitäisi näkyä kaksi prosessia ko. nimellä. Sitten voidaan käynnistää varsinaiset klusteripalveluja käyttävät MySQL-prosessit. Tämän jälkeen hallintanoodista käsin voidaan tarkastella storage-noodien

klusteriprosessien, sekä mysqld-asiakkaiden tilaa. Onnistuneen hallinta- ja storage-noodien pystytyksen jälkeen hallintaclient (ndb_mgm) näyttää seuraavaa:

```
-- NDB Cluster -- Management Client --
ndb_mgm> show

Connected to Management Server at: localhost:1186

Cluster Configuration
-----
[ndbd(NDB)]      2 node(s)
id=3   (Version: 4.1.11, Nodegroup: 0)
id=4   (Version: 4.1.11, Nodegroup: 0, Master)

[ndb_mgmd(MGM)] 1 node(s)
id=1   @192.168.0.42 (Version: 4.1.11)

[mysqld(API)]   2 node(s)
id=7   (Version: 4.1.11)
id=8   (Version: 4.1.11)

ndb_mgm>
```


4. Palveluiden valvonta

Palveluiden tilan valvontaan käytettiin Retiisi-projektin aikana Nagios-sovellusta [10]. Se on avoimen lähdekoodin sovellus, joka tarjoaa laajennettavan plugin-struktuurin erilaisten verkon palvelimien tarkkailuun. Myöhemmin OpenSIPg-ympäristöön on tullut oma, erillinen valvonta- ja ylläpitotoiminto, jolla voidaan hallita palveluiden uudelleenkäynnistystä, sekä vikasietoisten välityspalvelimien (aktiivinen/passiivinen) toimintaa.

Nagiosin käyttöliittymänä on www-sivu, jonka kautta voidaan palveluiden tilan seuraamisen lisäksi luoda raportteja halutuista palveluista tietyltä aikaväliltä, sekä ajoittaa ja suorittaa uudelleentarkistuksia ja -ajoituksia. Käyttöliittymän kautta voidaan myös selailla palveluiden tilaa erilaisten ryhmä- ja kohdenäkymien perusteella, mutta ne konfiguroidaan silti tekstitiedostojen kautta. Ohessa Host Detail- ja xxxx11-koneen palvelunäkymän kuvaruudut. Host Detail -näkyvästä selviää, ovatko valvottavat koneet verkossa tavoitettavissa, ja yksittäisen koneen näkyvästä voidaan tarkkailla kaikkia kyseiseltä koneelta valvottavia prosesseja.

The screenshot shows the Nagios web interface in Mozilla Firefox. The browser title is "Nagios - Mozilla Firefox". The address bar shows the Nagios URL. The interface includes a navigation menu on the left with sections for "General" (Home, Documentation) and "Monitoring" (Tactical Overview, Service Detail, Host Detail, Status Overview, Status Summary, Status Grid, Status Map, 3-D Status Map, Service Problems, Host Problems, Network Outages, Comments, Downtime, Process Info, Performance Info, Scheduling Queue). The main content area displays the "Current Network Status" (Last Updated: Fri Jun 10 09:32:58 EEST 2005, Updated every 90 seconds, Nagios® - www.nagios.org, Logged in as nagios) and "Host Status Totals" (Up: 6, Down: 0, Unreachable: 0, Pending: 0). Below this is a table titled "Host Status Details For Host Groups" with columns for Host, Status, Last Check, Duration, and St. The table lists several hosts, all with a status of "UP".

Host	Status	Last Check	Duration	St
radius1	UP	05-04-2005 08:36:56	97d 23h 29m 5s	(H
gary	UP	03-04-2005 09:11:50	97d 23h 29m 4s	PI
xxxx11	UP	05-04-2005 08:45:06	81d 0h 5m 24s	PI
xxxx12	UP	05-04-2005 08:41:27	59d 22h 1m 4s	(H
xxxx13	UP	05-04-2005 08:43:07	59d 22h 1m 4s	PI
xxxx14	UP	05-04-2005 08:44:26	79d 17h 41m 4s	(H

Kuva: Host Detail

Current Network Status
 Last Updated: Fri Jun 10 09:41:46 EEST 2005
 Updated every 90 seconds
 Nagios® - www.nagios.org
 Logged in as nagios

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

Service Status Details For Host 'xxxx11'

Host	Service	Status	Last Check	Duration	Attempt
xxxx11	01-PING	OK	04-12-2005 09:15:54	59d 22h 9m 41s	1/3
	02-/FS	OK	06-10-2005 09:41:15	81d 0h 12m 2s	1/3
	03-HTTPS	OK	04-12-2005 09:11:46	81d 0h 12m 2s	1/3
	04-MySQL	OK	04-12-2005 09:15:45	59d 23h 5m 42s	1/3
	05-SER process	OK	04-12-2005 09:15:37	59d 23h 5m 42s	1/3
	SIP REGISTER	CRITICAL	05-04-2005 08:44:56	37d 0h 59m 40s	3/3

Kuva: Service Status Details

Nagiosta valvottiin seuraavilla plugineilla: `check_ping`, jolla voidaan lähettää kohteelle ping-viestejä ja tarkistaa niihin saatavia vasteita, `check_disk`, jolla voidaan suorittaa kohteessa (erillisen nrpe -pluginin avulla) määritellyn levyosion tilatarkistus, `check_http`, jolla tarkistetaan HTTP-palvelimen käynnissä oleminen ja vasteaika, `check_mysql`, jolla tehdään kohteelle MySQL-statistiikkakysely, sekä `check_ser_procs`, jolla tehdään SER-prosessien tarkistus nrpe-pluginilla. Alla esimerkki `check_http`-pluginin suoritusmäärittämisestä.

```
# Service definition
define service{
  use      generic-service      ; Name of service template to use

  host_name      xxxx11
  service_description  03-HTTPS
  is_volatile    0
  check_period   24x7
  max_check_attempts 3
  normal_check_interval 100
  retry_check_interval 1
  contact_groups linux-admins
  notification_interval 240
  notification_period 24x7
  notification_options c,r
  check_command  check_https
}
```

Yllä olevassa määritellään mm. palvelun tarkistuksen frekvenssi, maksimi uudelleentarkistusyritykset, jos tarkistus ei onnistu, ryhmät joihin otetaan yhteyttä mikäli tietyt raja-arvot palvelun tarkistuksessa ylittyvät, sekä itse tarkistuskomento, jonka tarkemmat määrytykset löytyvät erillisestä konfiguraatitiedostosta. Ohessa check_https-pluginin määrytykset. Kuten esimerkistä näkyy, on Nagioksella helppo tehdä useamman koneen valvontaympäristö, sillä komentomäärytykset hyväksyvät muuttujilla määriteltyjä kohteita (esim. \$HOSTADDRESS\$) , jotka parsitaan varsinaisesta palvelujen tarkkailua määrittävästä konfiguraatitiedostosta. Näin ollen samalla palvelumäärytyksellä voidaan kontrolloida usean kohteen tarkistuksia.

```
# 'check_https' command definition
define command{
command_name      check_https
command_line      $USER1$/check_http -H $HOSTADDRESS$ -S
}
```

Projektin alkaessa Nagioksella oli tarkoitus suorittaa myös korjaavia toimenpiteitä, mikäli valvotuissa palveluissa olisi tapahtunut vikaantumisia, mutta tarvittavien skriptien kehittäminen ja testaaminen todettiin projektin sisällön ja etenkin klusteroinnin aiheuttaman työmäärän puitteissa mahdottomaksi. Periaatteellisella tasolla Nagioksen on kuitenkin mahdollista suorittaa tällaisia korjaavia toimenpiteitä nk. event handler -komennoilla.

Manuaalinen palveluiden kontrollointi suoritettiin kuitenkin valvontakoneille tehtyjen dynaamisten nimipalvelunpäivitysskriptien avulla. Mikäli havaittiin, että jokin klusterin storage-noodeista oli alhaalla, ajettiin kyseistä kohdetta varten luotu nsupdate-skripti. Esimerkiksi jos primääri SER-kone (xxxx11/xxxxnn) kaatui, ajettiin oheinen skripti:

```
server ns1.wirlab.net
;
update delete xxxxn.wirlab.net. A
send
;
update add xxxxn.wirlab.net. 0 IN A 192.168.0.12
send
;
```

Tämä skripti suoritti dynaamisen nimipalvelupäivityksen, jossa xxxxn host-osoite siirrettiin viittamaan sekundääri SER-palvelimeen (xxxx12). Vastaavat päivitysskriptit oli jokaiselle klusterin kahdennetulle koneelle, jolloin minkä tahansa kriittisen palvelun kaatuessa näiltä koneilta, voitiin nimipalvelu siirtää osoittamaan toisaalle. Automatisoimalla nimipalvelupäivitykset Nagioksen avulla palvelimien hallinnasta oltaisiin saatu tehokkaampaa, mutta Nagioksen tiettyjen toiminnallisten rajoitteiden puitteissa tämä ei ollut mahdollista.

5. Tulokset

Kahdentamisen ja klusteroinnin kannalta projekti toteutui menestyksekkäästi. Tarvittavat osakokonaisuudet rakennettiin valmiiksi kuten suunniteltiin ja klusterin toiminnallisuus todettiin odotusten mukaiseksi. SER- ja RADIUS-palvelut saatiin toimimaan kahdennusympäristössä kuten suunniteltiin, ja lisäksi em. palveluiden konfiguraatitiedostoilla pystyttiin luomaan esimerkiksi SER:ille useamman RADIUS-palvelimen autentikointiketju vikatilanteiden varalta. Jatkokehitysmahdollisuudeksi voidaan katsoa konfiguraatioiden dynaamisempi hallinta, esimerkiksi autentikointiketjujen automatisointimahdollisuus järjestelmän asennusvaiheessa. Tällaisenaan kahdentamisen konfigurointi verrattuna muuhun OpenSIPg-ympäristöön on huomattavasti monimutkaisempaa. OpenSIPg on tuotteistettu, ja Retiisi-klusteriympäristö ei.

Todettiin, että MySQL 4.1-versiosta alkaen mukana tullut klusterointituki on jo kehittynyt niin pitkälle, että se voidaan ottaa käyttöön ei-kriittisessä tarkoituksessa olevissa palvelimissa. MySQL Cluster -ohjelmistoversioiden välillä on vielä joitain ongelmia, esimerkiksi management- ja storage-noodien välinen kommunikointi vaatii tietyn tasoiset MySQL-versiot molemmille noodeille toimiakseen kunnolla. Muutamat ongelmat, joita havaittiin vielä versioiden 4.1.11 kanssa, ovat klusteritestien jälkeen korjattu ohjelmiston kehittäjän toimesta. Kuitenkin on huomattava, ettei MySQL-klusterointi vielä 4.1-versiossa ole sovelias kriittiseen tuotantokäyttöön. Testeissä todetut ongelmat mahdollisine tietokantojen korruptoitumisineen johtavat siihen, että (epätodennäköisestäkin) MySQL-klusterin kaatumisesta johtuva korjausaika saattaa kestää pidempään kuin laite- tai ohjelmistovian korjaaminen perinteisemmässä, ei-klusteroidussa tietokantaympäristössä. Lähtökohtainen suositus klusteroidun MySQL:n käyttöön otossa on tukisopimuksen tekeminen MySQL-yhtiön kanssa.

Klusterissa olevilla storage-noodeilla on syytä ottaa tarkkaan huomioon klusteroitavien tietokantataulujen koko, sillä klusterin data on ainoastaan keskusmuistissa. Näin ollen muistin määrästä huolehtiminen on tärkeää suorituskyvyn ja toiminnan kannalta. Ohjeita muistin määrän laskentaan löytyy MySQL-klusterin FAQ-sivulta [11].

Valittu Debian GNU/Linux-alusta mahdollisti sujuvan versioiden päivityksen, mutta samalla todettiin että tietokannan päivitys vaatii tarkkaavaisuutta ylläpidolta. Mikäli versiopäivityksen yhteydessä tietokantataulujen sisällölle tapahtuu jotain, kaikki muutokset kopioituvat senhetkiselä master-palvelimelta klusterin muille storage-noodeille. Varmuskopioiden ottaminen levyjärjestelmästä, sekä klusteriprosessien alasajo ennen versio- ja ohjelmistopäivityksiä on suotavaa.

Palveluiden valvonnan automatisointia ja palvelukatkoksiin reagoitua ei pystytty toteuttamaan Nagioksella niin pitkälle kuin suunniteltiin (ks. kappale 4). Retiisi-projektin kuluessa OpenSIPg-alustaan on tullut kuitenkin oma, Retiisin toteutuksesta erillinen kahdennus- ja varmennustoiminnallisuus, jonka toimintaa demottiin Retiisi-projektin johtoryhmän kokouksessa 16.12.2005.

6. Viitteet

- [1] SIP Express Router, <http://www.iptel.org/ser/>
- [2] FreeRADIUS, <http://www.freeradius.org/>
- [3] Bind, <http://www.isc.org/index.pl?sw/bind/>
- [4] MySQL, <http://www.mysql.com/>
- [5] MediaProxy, <http://www.ag-projects.com/MediaProxy.html>
- [6] SIP Express Media Server, <http://sems.berlios.de/>
- [7] Session Initiation Protocol, <http://www.ietf.org/rfc/rfc3261.txt?number=3261>
- [8] Remote Access Dial-In User Service,
<http://www.ietf.org/rfc/rfc2865.txt?number=2865>
- [9] RADIUS Accounting, <http://www.ietf.org/rfc/rfc2866.txt?number=2866>
- [10] <http://www.nagios.org/>
- [11] MySQL Reference Manual: MySQL Cluster FAQ,
<http://dev.mysql.com/doc/refman/5.0/en/mysql-cluster-faq.html>

7. Lyhenteet

CNAME	Canonical Name
DB	Database
ENUM	Enumeration
GNU	GNU is Not UNIX
HTTP(S)	HyperText Transport Protocol (Secure)
IETF	Internet Engineering Task Force
NDBD	Network Database Daemon
RADIUS	Remote Access Dial In User Service
RFC	Request For Comments
RTP	Realtime Transport Protocol
SEMS	SIP Express Media Server
SER	SIP Express Router