Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Informatikai infrastruktúra témalabor (vimm4325)

Hibatűrő rendszerek megvalósítása — Hálózati terheléselosztó fürtök —

Mérési segédlet
Medgyesi Zoltán, Micskei Zoltán
2008.

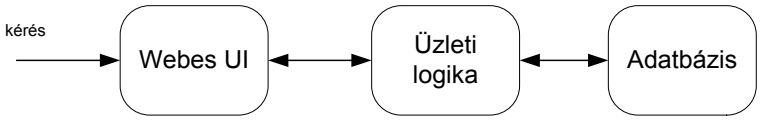
A témalabor mérései során egy tipikus webes alkalmazást kiszolgáló infrastruktúra megbízhatóságát fogjuk elemezni, az azonosított hibapontok kiküszöbölésére pedig különböző hibatűrő technikákat mutatunk be. A mérések felépítése a következő:

1. mérés: infrastruktúra összeállítása, elemzése. Hálózati terheléselosztás alkalmazása.
2. mérés: feladat-átvételi fürtök és replikáció bemutatása.
3. mérés: megbízhatósági modellezés és analízis.

Előismeretek: a mérés során a következő fogalmak ismeretére építünk, ezeket frissítsétek fel! IP cím, alhálózati maszk, privát IP tartományok, MAC cím, ARP protokoll, DNS feloldás menete, unicast/multicast/broadcast kommunikáció.

1 Három rétegű webes alkalmazás felépítése

Webes alkalmazások megvalósítása esetén gyakran használt felépítés a három (vagy finomabb granularitású rendszer esetén az N) rétegű architektúra. Logikai szinten elkülönítjük a főbb funkciókat ellátó rétegeket, úgymint:

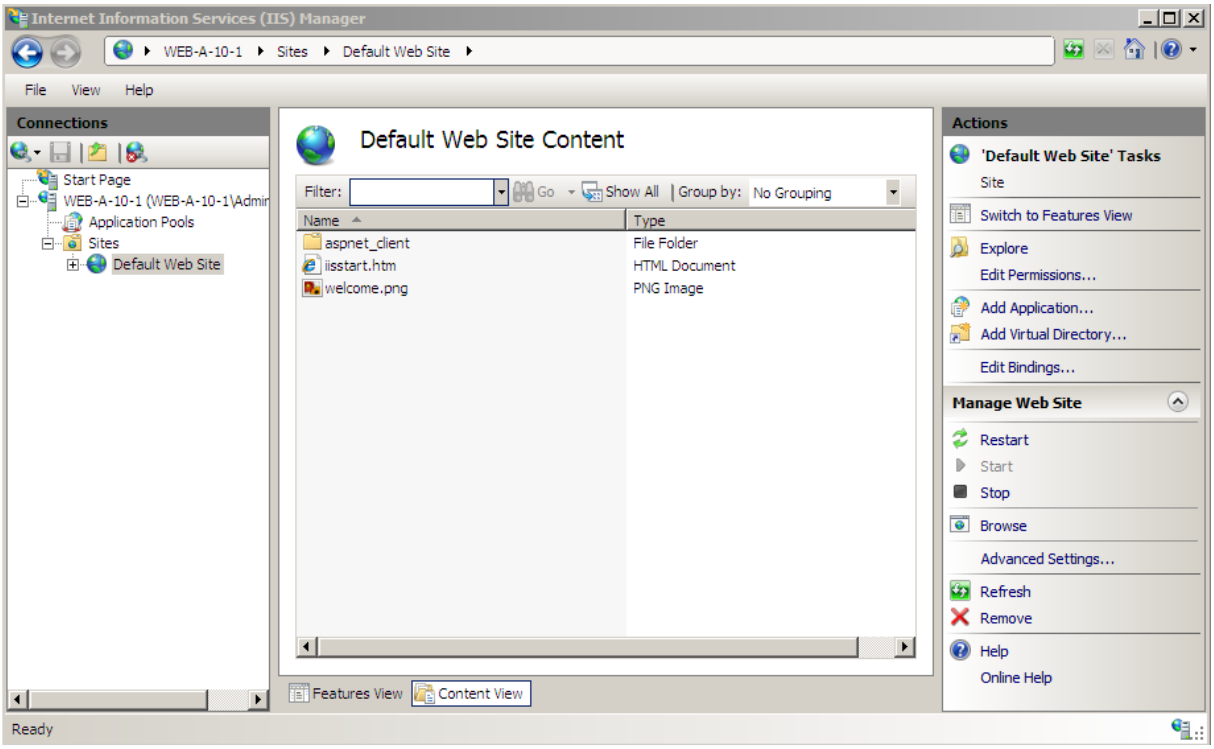


Hasonló rétegezést majd minden manapság használatos platformon létre tudunk hozni, a mostani mérés folyamán a Microsoft eszközein keresztül mutatjuk be a technológia alapjait. Az egyes rétegek a következő alkalmazások segítségével lesznek megvalósítva.

- Webes UI: a webes felületet a Windows Server 2008-ba beépített web szerver, az *Internet Information Services* (IIS) 7.0 szolgáltatja.
- Üzleti logika: az üzleti logikát megvalósító alkalmazás szerver komponensek a *Windows Server 2008* részét képezik, pl. a biztonság és tranzakció kezelést biztosítanak a COM+ szolgáltatások vagy a .NET platform által nyújtott különböző szerver oldali komponensek.
- Adatbázis: az adatbázis szerver egy *Microsoft SQL Server 2005* lesz.

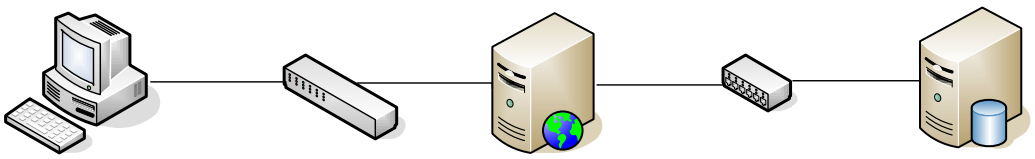
1.1 Web szerver

Az IIS az IIS Manageren keresztül adminisztrálható. A *Sites* rész alatt található, hogy milyen *webhelyeket* szolgál ki, és az ezekhez érkező kéréseket mi alapján különíti el (pl. IP cím, port szám vagy állomás fejléc). Ezeket a webhelyeket egy-egy a gépen lévő fizikai mappához rendeli a web szerver, a mappa tartalmát a *Content View* nézetben mutatja a konzol. A *Features View* résznél a webszerver és a hozzá tartozó különböző kiegészítéseket (CGI, ASP.NET, de ide tartozik a hibás kéréseket diagnosztizáló modul is akár) konfigurálhatjuk. A webhelyekhez automatikusan létrejön egy *webalkalmazás*, de tetszőleges alkönyvtárat is kinevezhetünk külön webalkalmazásnak. A webalkalmazásokat *alkalmazás készletekbe* (application pool) rendezhetjük. Az alkalmazás készletek a webalkalmazások elkülönítésére szolgálnak, külön operációs rendszer folyamatokban futhatnak, és további korlátozásokat adhatunk meg rájuk, pl. mennyi memóriát vagy processzort használhatnak.



1. ábra Az IIS 7.0 felülete

Mivel a mérés során csak egy egyszerű webes alkalmazást használunk, és a szerverek virtuális gépekben futnak majd, így a három logikai réteget két gépre telepítjük.



1.2 Adatbázis szerver

A mérés során az adatbázis szerverrel nem fogunk sokat foglalkozni, most csak egy dolgot emelünk ki. Gondoskodnunk kell arról, hogy a webes alkalmazás elérje az SQL adatbázist. A Microsoft SQL Server két fajta hitelesítési módszert támogat.

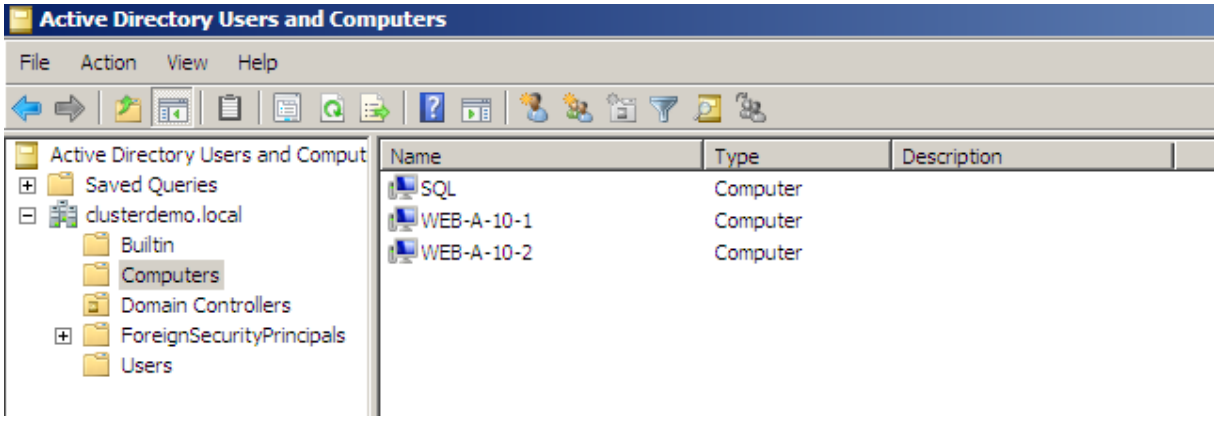
- A felhasználók információit (név, jelszó) az SQL Server tárolja. Ez a kevésbé javasolt megoldás, főleg kompatibilitás miatt maradt benne a termékben. Hátránya, hogy a jelszavakat kevésbé biztonságos módon tárolja.
- Windowsos felhasználók használata. A felhasználókat és adataikat az operációs rendszer tárolja, az SQL Serveren csak jogosultságot osztunk ki nekik. Előnye, hogy így egy helyen kell csak a felhasználókat menedzselni, és bejelentkezéskor csak a jelszóból képzett hash megy át a hálózaton.

A második esetben biztosítani kell, hogy a web szerveren lévő felhasználó, aki a webalkalmazást kiszolgáló folyamatot futtatja, létező felhasználó legyen az SQL Servert futtató gépen is. Ha két egyedül álló (stand-alone) gép lenne a web és SQL szerver, akkor két, egymástól független felhasználói adatbázisuk lenne, mellyel ez a feladat nehézkesen oldható meg[1]. Ezért még egy komponenst használunk, egy közös címtárat.

1.3 Címtár

A *címtár* (directory) egy olyan adatbázis, ami felhasználókat, gépeket és egyéb erőforrásokat tárol, és lehetővé teszi ezek keresését és közös helyről történő adminisztrálását. A leggyakoribb címtár formátum alapja az X.500-as szabvány, melynek elérése a Lightweight Directory Access Protocolon (LDAP) keresztül történik. A Microsoft implementációjának neve *Active Directory (AD)*. Az Active Directory főbb jellemzői:

- Az AD-ben tárolható fontosabb *objektumok*: felhasználók, csoportok, számítógépek, nyomtatók, házirendek, de akár felhasználó által definiált típusok is.
- Minden objektumtípushoz egy előre megadott *tulajdonság* halmaz tartozik (pl. a felhasználónak neve, email címe, telefonszáma, stb. van). Hogy egy adott típusú objektumhoz milyen tulajdonságok tartoznak azt a címtárhoz tartozó *séma* definiálja.
- Az Active Directory *hierarchikus* felépítésű. Az adminisztráció egysége a *tartomány* (domain). A tartományt a DNS nevével adjuk meg (melyet általában célszerű elkülöníteni a publikus internetes DNS névtől, pl. clusterdemo.hu és clusterdemo.local). Egy tartomány képez egy biztonsági egységet, pl. a felhasználók a tartományi felhasználójukkal be tudnak jelentkezni bármelyik, a tartományba beléptetett gépre. A tartományon belül a hierarchiát *szervezeti egységeknek* (organizational unit) nevezett tárolók létrehozásával adhatjuk meg.



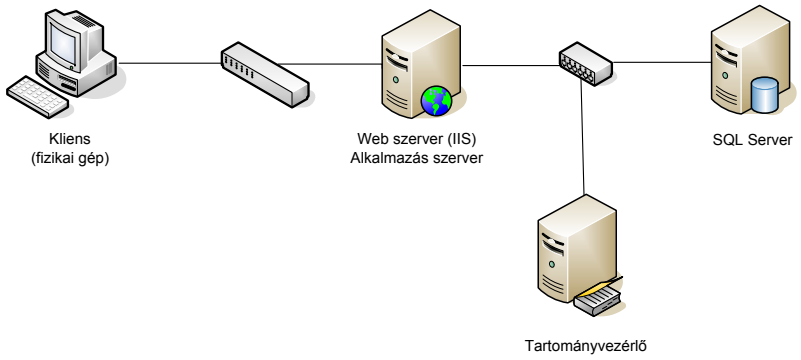
2. ábra Az Active Directory kezelőfelülete

[1] Ha mindkét gépen készítünk egy felhasználót, akinek ugyanaz a neve és jelszava, akkor át tud hitelesíteni. Viszont a két felhasználó jelszava egymástól függetlenül változhat, ami nem szerencsés.

- A tartományokon belül altartományok létrehozásával egy fa (tree) struktúrát alakíthatunk ki. Az így kapott fák *erdőbe* (forest) szerveződnek.
- Az egy tartományba tartozó objektumok közösen menedzselhetőek. Megadhatjuk, hogy milyen szoftverek települjenek fel rá, vagy *csoportházirendek* (group policy) definiálásával beállíthatjuk a gépek és felhasználók tucatnyi jellemzőit (pl. átállíthatják-e a képernyő méretét, mi legyen a Start menüjükben vagy kinek van joga belépni arra a gépre).
- A címtár adatait az úgynevezett *tartományvezérlő* (domain controller) gépek tárolják.

Az Active Directory alap szolgáltatásait és a fontosabb kifejezéseket részletesebben a [1] leírás ismerteti.

Így a jelenlegi rendszerünk a következőképp néz ki.



2 Hibatűrő infrastruktúra kialakítása

A jelenlegi rendszerünk nem tolerálja egyik gép vagy komponens meghibásodását sem, bármilyen meghibásodás esetén a szolgáltatás nem lesz elérhető, nem működik a webes alkalmazásuk. A lehetséges hiba okok feltárásával, ezek hatásainak vizsgálatával részletesebben majd a megbízhatósági modellezés mérésen foglalkozunk, az első két mérés pár hibatűrő technika megismerésére szolgál.

A legtöbb hibatűrő technológia valamilyen redundancia beépítésére alapul. Az egyik gyakori technológia a számítógépek megtöbbszörözése és *fürtök* (cluster) kialakítása. A fürt definíciója nagyjából egységesen szerepel a szakirodalomban: különálló, hétköznapi jellemzőkkel rendelkező számítógépek – fürttagok – együttese, amelyek egymással együttműködve és azonos szolgáltatásokat, alkalmazásokat futtatva egyetlen rendszerként, virtuális kiszolgálóként jelennek meg az ügyfelek számára. A fürt tagjaival olyan hibatűrő, teljesítménynövelő megoldások – terheléselosztás, feladatátvétel – valósíthatók meg, amelyek egyetlen számítógép használatakor nem érhetők el.

A következő hibákra tudunk felkészülni a mérés során megismerendő technológiákkal:

- Web szerver: További web szervereket állítunk be, és az érkező kéréseket hálózati terheléselosztás segítségével szétosztjuk közöttük. A Microsoftos implementáció neve Network Load Balancing (NLB)[2].
- Adatbázis szerver: feladatátvételi fürt (failover cluster) létrehozása. A csomópontok közül egyszerre egy aktív, hiba esetén egy másik átveszi a szolgáltatást (failover mechanizmus). A fürt szolgáltatásokat a Windows Server Failover Clustering biztosítja.
- Címtár: Az Active Directory-ban beépített hibatűrő mechanizmus van. Egy tartományhoz több tartományvezérlő is tartozhat, ezek egyenrangúak, folyamatosan replikálnak egymás között.

[2] Másnéven: NLBS (Network Load Balancing Service) vagy WLBS (Windows Load Balancing Service).

- Hálózati eszközök: A fentiek után még mindig maradnak egyszeres hibapontok a rendszerben (SPOF), például a hálózati eszközök. Ezek megduplázásával és alternatív hálózati utak létrehozásával viszont a jelenlegi mérésen nem fogunk foglalkozni.

Az első mérés során részletesebben a hálózati terheléselosztással foglalkozunk.

2.1 Hálózati terheléselosztás

A hálózati terheléselosztás alapvető célja inkább a teljesítmény növelés, de természetesen a redundancia miatt hibatűrést is biztosít. Az egymástól független gépeket egy fürtbe szervezzük, ami kívülről egy virtuális címen érhető el. Az erre a címre érkező kérések szétosztása a csomópontok között az egyes megvalósításokban egész eltérő lehet. A legegyszerűbb megoldásokban egy statikus lista alapján megy ez (pl. round-robin DNS), míg a legkifinomultabban a csomópontok folyamatosan kommunikálnak egymással, hogy felmérjék a másik aktuális terheltségét és figyelik, hogy az adott klienst ki szolgálta ki az előző kérésénél.

2.1.1 A hálózati terheléselosztás céljai

A hálózati terheléselosztás alapvető célja a beérkező kérések elosztása a külvilág felé egyetlen kiszolgálónak látszó kiszolgálók csoportja (fürtje) között. Fontos jellemzője, hogy a kérések elosztását önálló, a kérések kielégítésére önmagukban is képes, független kiszolgálók között végzi, illetve az elosztás művelete a hálózati rétegben történik, szemben például az alkalmazások, az alkalmazáskomponensek vagy az objektumok közötti elosztással.

- Állandó jellegű adatokkal dolgozó kiszolgálók esetében alkalmazható (FTP, web), változó adatok kezelésére (adatbázis, levelezés) a közös tárolóeszközt használó feladatátvételi fürtök az alkalmasabbak.
- A fürtözés a kiszolgálói alkalmazás megváltoztatása nélkül történik, a kiszolgáló alkalmazás nem tud arról, hogy őt fürtözik

A hálózati terheléselosztásnál a technológia kiválasztásának alapkérdés, hogy vajon alkalmazzunk-e egy kevésbé egyenletes terheléselosztást eredményező, de egyszerű módszert, vagy egy minél kifinomultabban szabályozott, minél egyenletesebb elosztást biztosító, de magas többletterheléssel járót.

2.1.2 Microsoft Network Load Balancing

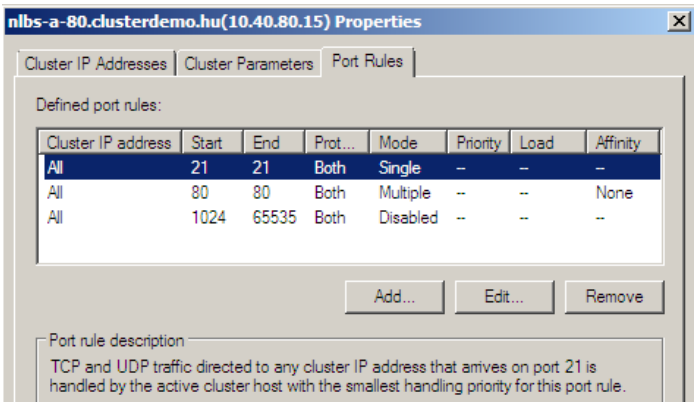
A Microsoft hálózati terheléselosztás implementációja decentralizált, egymással kommunikáló fürttagokból áll. Az NLB legfontosabb tulajdonságai a következők.

- Kieső kiszolgálógépek automatikus felismerése a kiszolgálók között szívverések (heartbeat) üzenetek segítségével.
- Legfeljebb 32 kiszolgáló lehet egy fürtben, a csomópontok egymás között kommunikálnak.
- Kiesés esetén a kérések a túlélő kiszolgálók közötti újraelosztását 10 másodpercen belül elvégzi.
- Nincs központi elosztó elem (ami SPOF lehet), emiatt olcsóbb is.
- Minden kiszolgáló rendelkezik egy dedikált IP-címmel, ez a saját, egyedi címe, és a fürtnek van egy közös virtuális IP-címe, ehhez történik a terheléselosztás.
- A kiszolgáló alkalmazások futását nem figyeli, nem indítja el, és nem állítja le őket, ezt külső figyelő alkalmazásokkal kell elintézni, a távoli állapotlekérdezési/felügyeleti felület biztosított.
- Az egyes kiszolgálók terhelésének dinamikus változását nem veszi figyelembe.

2.1.2.1 Az NLB-vel kapcsolatos legfontosabb fogalmak

- *Portszabályok*: definiálják, hogy egy adott hálózati port tartományt hogyan kezeljen a fürt. Lehetővé teszik, hogy bizonyos porttartományokhoz kiszolgálókat rendeljünk, illetve bizonyos portok forgalmának fogadását letiltsuk. A szűrések típusa:
  - Egyhosztos szabály: az adott portok forgalma mindig egy fürttaghoz kerül.
  - Többhosztos szabály: a bejövő forgalom elosztása az összes kiszolgáló között történik, a kiszolgálókhoz százalékos értékek adhatók, így a nagyobb kapacitású gépekhez több kérést lehet eljuttatni.
  - Letiltva: az adott porttartomány kiszolgálását le is tilthatjuk, ilyenkor az ezekre a portokra érkező kéréseket eldobja az NLB illesztőprogramja.
- *Ügyfél affinitás*: megadja, hogy hogyan kezeljük az egy ügyféltől származó kéréseket. Többhosztos szabálynál használható, háromféle van:
  - nincs affinitás: azonos IP-cím különböző portjainak forgalma különböző kiszolgálókra kerül; ez a legjobb válaszidejű, legjobban elosztott megoldás (sőt, valószínűleg ilyenkor még portokra sincs semmilyen affinitás, pl. egy weblap képei különböző kiszolgálókról is lekérhetők).
  - Egy ügyfeles affinitás: adott IP-cím minden forgalmát a fürt valamely kiszolgálójára juttatja.
  - C osztályú affinitás: egy C címosztály minden IP-címének minden forgalmát adott kiszolgálóra juttatja.
- *Hosztprioritás*: a legkisebb számú hoszt a legnagyobb prioritású, minden olyan ügyfélforgalmat, amelyet a fürt IP-címére küldtek, és amelyre nem definiáltunk portszabályt, ez kezel.

Az alábbi ábrán látható egy példa port szabályokra. A 21-es portot csak egy csomópont szolgálja ki, a 80-as portra be van állítva a terheléselosztás, míg az 1024 feletti portok le vannak tiltva.



3. ábra Port szabályok

2.1.2.2 Az NLB technológiája

Az NLB-nél alkalmazott hálózati technológia működése a következőképp foglalható össze.

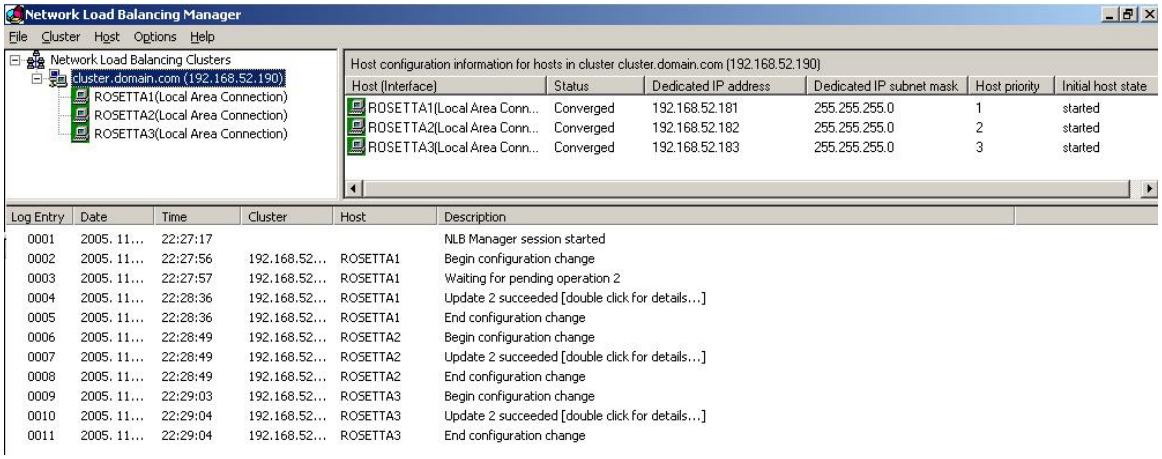
- A kliensek a fürt IP-címére küldik a kéréseket, amit megkap minden csomópont.
- Az NLB illesztőprogramja beékelődik a hálózati kártya illesztőprogramja és a protokollkészlet közé, és ott szűrő feladatot lát el. Egy meghatározott algoritmus alapján eldönti, hogy ezt a fürtnek küldött kérést ennek a csomópontnak kell-e kiszolgálnia, ha igen, akkor továbbítja a felsőbb rétegek felé.

- Az NLB módosítja a fürthöz rendelt hálózati kártya MAC-címét. Ennek az új MAC-címnek a típusa alapján különböztetünk meg *unicast* és *multicast* működési módot.
- Az ügyfélforgalom elosztásakor a következő algoritmust használja. Az elosztás forrás IP-cím és port, illetve egyéb állapotadatok és véletlenszerűsítés alapján történik. Alapvetően sok kisebb, széles felhasználói közösségtől érkező kérés esetén hatékony, pl. webkiszolgálónál, de alacsony többletterhelése miatt más esetekben is hatékony.

Kétféle üzemmódban képes működni az NLB fürt:

- *Unicast*: a csomópontok elrejtik a saját MAC címüket, és a közös, unicast MAC címet rendelik a fürthöz megadott hálózati csatolóhoz. A fürt IP címét és a csomópont dedikált IP címét is erre a MAC címre oldja fel az NLB. A válaszok küldésekor a forrás MAC-cím szintén „hamis", az NLB a fürt MAC-címéből és a hoszt prioritásából származtatja.
- *Multicast*: a csomópontok megtartják a saját MAC címüket, és egy új, közös multicast MAC címet használnak az NLB fürtnek. A csomópont dedikált IP címe a saját fizikai MAC címére oldódik fel.

Minden csomópont periodikusán heartbeat üzeneteket küld szét, ezzel jelezve a többieknek, hogy még működik. A heartbeat forgalom mindig az NLB-hez rendelt hálózati kártyán keresztül bonyolódik. Unicast esetben ez egy broadcast üzenet küldésével történik, multicast esetben pedig a fürt multicast MAC címére küldött üzenettel. Ha valamelyik csomópont kiesését észlelik, akkor egy un. konvergencia folyamatot indítanak el, melynek során eldöntik, hogy ki maradt még benn a fürtben, és ki lesz a default host (a legkisebb hosztprioritással rendelkező csomópont).



4. ábra Network Load Balancing Manager

Az NLB menedzselése a Network Load Balancing Manageren keresztül történik, a mérésen ezt a felületet fogjuk használni.

A beállításokat a csomópontok a registry-ben tárolják, a következő kulcs alatt: HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WLBS

A további hálózati terheléselosztási technológiák ismertetése a mérés weboldalán megtalálható, opcionális, leírásban található.

2.1.2.3 Az NLB működés közben

A következő hálózati forgalom részlet a technológia működését mutatja be. A fürt multicast üzemmódot használ, két csomópontból áll.

Num	Source Address	Dest Address	Summary
1	Vmware_91:ec:0b	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat
2	Vmware_91:ec:0b	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat
3	Vmware_cf:30:96	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat
4	Vmware_91:ec:0b	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat
5	Vmware_cf:30:96	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat
6	Intel_b5:46:fb	Broadcast	ARP: Who has 10.40.210.1? Tell 10.40.30.1
7	Vmware_76:0f:33	Intel_b5:46:fb	ARP: 10.40.210.1 is at 00:0c:29:76:0f:33
8	10.40.30.1	10.40.210.1	DNS: Standard query A nlbs-a-30.clusterdemo.hu
9	10.40.210.1	10.40.30.1	DNS: Standard query response A 10.40.30.15
10	Intel_b5:46:fb	Broadcast	ARP: Who has 10.40.30.15? Tell 10.40.30.1
11	Vmware_cf:30:96	Intel_b5:46:fb	ARP: 10.40.30.15 is at 03:bf:0a:28:1e:0f
12	10.40.30.1	10.40.30.15	TCP: 1097 > http [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=...
13	10.40.30.15	10.40.30.1	TCP: http > 1097 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 ...
14	10.40.30.1	10.40.30.15	TCP: 1097 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
15	Vmware_91:ec:0b	Intel_b5:46:fb	ARP: 10.40.30.15 is at 03:bf:0a:28:1e:0f
16	10.40.30.1	10.40.30.15	HTTP: GET / HTTP/1.1
17	10.40.30.15	10.40.30.1	HTTP: HTTP/1.1 200 OK (text/html)
18	10.40.30.1	10.40.30.15	TCP: 1097 > http [ACK] Seq=391 Ack=370 Win=65167 Len=0
19	10.40.30.1	10.40.30.15	TCP: 1097 > http [FIN, ACK] Seq=391 Ack=370 Win=65167 Le...
20	10.40.30.15	10.40.30.1	TCP: http > 1097 [ACK] Seq=370 Ack=392 Win=63850 Len=0
21	Vmware_91:ec:0b	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat
22	Vmware_cf:30:96	MS-Loadbalancing-Multicast_0a:28:1e:0f	0x886f: MS NLB heartbeat

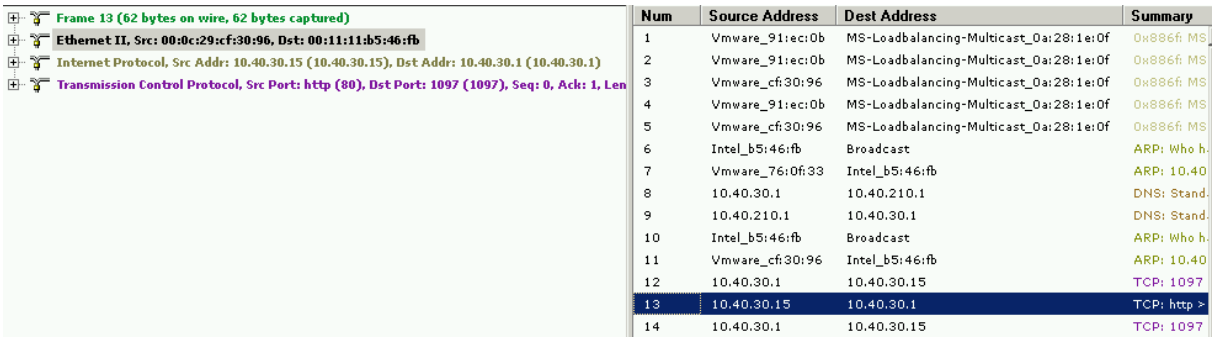
1-5: heartbeat üzenetek a fürt két tagja között
Forrás: valamelyik fürttag MAC címe
Cél: fürt közös multicast MAC címe, ide küldött üzeneteket minden tag megkapja

6-7: kliens kérdezi a DNS szerver MAC címét
megkapja a választ rá

8-9: kliens kérdezi, a fürt nevéhez tartozó IP címet a DNS szervertől
megérkezik a válasz, 10.40.30.15

10-11: kliens kérdezi a fürt közös IP címéhez tartozó MAC címet
válasz (az a multicast MAC cím, amire pl. eddig a heartbeatek is mentek)

12: http kapcsolathoz szükséges TCP kapcsolat kiépítése



13: Válasz a http kapcsolat kiépítésére. Figyeljük meg, hogy bár a kérés a fürt MAC címére és IP címére ment, a válaszban forrásként a fürt közös IP címe és egyik csomópont MAC címe szerepel! Emiatt tud működni az NLB, mert így a switch-ek

nem tudják megtanulni, hogy ki tartozik a fürt MAC címéhez, és így kénytelenek kiküldeni az összes portjukra a fürtnek küldött csomagokat.[3]

Unicast esetén két fő eltérés van.

- A heartbeat üzenet broadcast üzenet.

66	02:01:0a:28:1e:0f	Broadcast	0x886f: MS NLB heartbeat
67	02:02:0a:28:1e:0f	Broadcast	0x886f: MS NLB heartbeat
88	02:01:0a:28:1e:0f	Broadcast	0x886f: MS NLB heartbeat
89	02:02:0a:28:1e:0f	Broadcast	0x886f: MS NLB heartbeat
90	02:01:0a:28:1e:0f	Broadcast	0x886f: MS NLB heartbeat
91	10.40.30.1	10.40.210.1	DNS: Standard query A nlbs-a-30.clusterdemo.hu
92	10.40.210.1	10.40.30.1	DNS: Standard query response A 10.40.30.15
93	10.40.30.1	10.40.30.15	TCP: 1209 > http [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=..

- A válaszban nem a csomópont saját MAC címe, hanem egy, a fürt MAC címéből a prioritást felhasználva képzett újabb virtuális cím szerepel (amely így minden csomópontra egyedi lesz).

A unicast mód hátránya, hogy mivel a saját IP címet is a közös MAC címre oldja fel, a fürttagok nem tudnak egymással direktben kommunikálni (a heartbeat üzenetek broadcastolva vannak, így azokat ez a probléma nem érinti). Ezen kívül a fürtön kívülről jövő, csak az egyik fürttagnak küldött direkt üzenetet is megkap mindenki (mivel nem a fürt IP címére érkezett, ezért csak a címzett fogja feldolgozni, ám felesleges többletterhelést jelent). Megoldás lehet egy második, másik alhálózatra csatlakozó hálózati csatoló alkalmazása. Multicast módban nincs feltétlen szükség második hálózati csatolóra (bár ha nagy a csak bizonyos fürttagoknak szánt forgalom, akkor érdemes lehet használni). Másfelől viszont, a multicast mód nem mindig használható, például ha a hálózati eszköz nem támogatja, hogy egy unicast IP címre küldött kéréshez egy multicast MAC-cím tartozik.

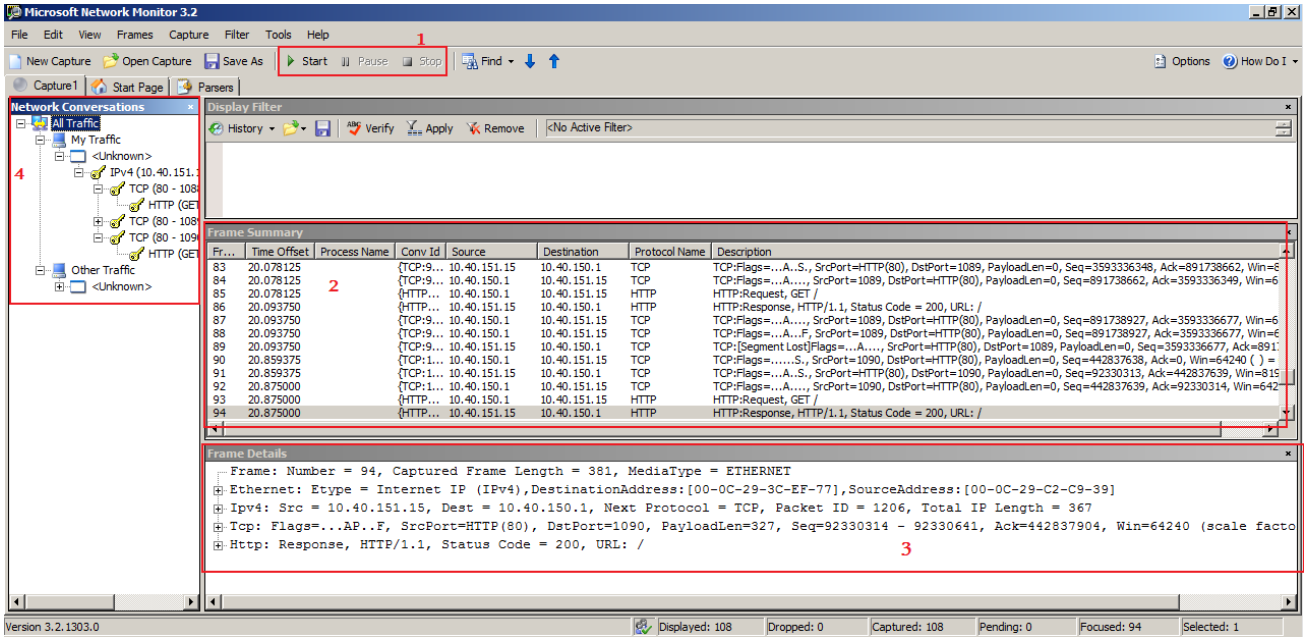
2.2 Hasznos programok

A mérés során hasznosak lehetnek a következő parancsok.

- ipconfig /all: MAC címek, IP címek és egyéb hálózati beállítások áttekintése.
- ipconfig /displaydns: helyi DNS cache megtekintése
- arp –a: helyi ARP cache megtekintése
- ipconfig /flushdns: helyi DNS cache ürítése
- arp –d *: helyi ARP cache ürítése (rendszergazdai jogok kellenek hozzá)
- nlb.exe: a Network Load Balance parancssori felülete
  - nlb query: fürt állapotának lekérdezése
  - nlb display: csomópont paramétereinek lekérdezése
  - nlb start, nlb stop: csomópont elindítása és leállítása
- Microsoft Network Monitor: hálózati forgalom monitorozó eszköz. Az 5. ábra szemlélteti a felület egyes elemeit:
  1. Forgalom rögzítésének elindítása/leállítása.
  2. Elkapott hálózati csomagok listája a típusukkal.

[3] Az Ethernet hálózatokban a 2. rétegbeli címzés MAC-címek alapján történik. Ha egy switch olyan Ethernet-keretet kap, amelyben számára még ismeretlen MAC-cím szerepel, akkor nem tudhatja, hogy melyik portján kell kiküldenie a keretet; tehát az összes portján kiküldi. Amikor a hálózati kommunikáció során a megszólított számítógép választ küld, akkor a válaszában szerepel a saját hálózati csatolójának a MAC-címe. A válasz továbbításakor a switch látja, hogy az adott MAC-címmel melyik portján érkezett keret, és ekkor ezt a címet az adott porthoz rendeli. A később beérkező keretek kezelésekor már nincs szükség arra, hogy ezeket az összes portján kiküldje, a továbbítást csak a hozzárendelt porton keresztül végzi el, ezzel csökkentve a hálózati forgalmat.

3. Csomagban lévő hálózati rétegek tartalma: közegelérési (jelen esetben Ethernet), hálózati (IP), szállítási (TCP) és alkalmazási (HTTP).
4. Hálózati kommunikációk listája, segítségével könnyen kiválaszthatjuk az egy konkrét forgalomhoz tartozó csomagokat.



5. ábra Network Monitor hálózati forgalom figyelő eszköz

További információ

[2] NLB leírása a Techneten (áttekintés, tervezés, üzemeltetés).

[3] Technikai részletek az NLB működéséről

Hivatkozások

[1] Active Directory Collection, Microsoft Technet, URL: http://technet2.microsoft.com/WindowsServer/en/library/6f8a7c80-45fc-4916-80d9-16e6d46241f91033.mspx?mfr=true

[2] Network Load Balancing Clusters, Microsoft Technet, URL: http://technet2.microsoft.com/WindowsServer/en/library/98d46a24-96d8-412c-87d8-28ace62323d21033.mspx?mfr=true

[3] Network Load Balancing Technical Overview (Windows 2000), Microsoft Technet, URL: http://www.microsoft.com/technet/prodtechnol/acs/reskit/acrkappb.mspx

[4] How Network Load Balancing Technology Works, Microsoft Technet, URL: http://technet2.microsoft.com/WindowsServer/en/library/6f8a7c80-45fc-4916-80d9-16e6d46241f91033.mspx?mfr=true

[5] Network Load Balancing cluster node does not successfully converge, Microsoft Knowledge Base, URL: http://support.microsoft.com/kb/812870/en-us

[6] Using the "WLBS QUERY" Command to Determine the State of an WLBS/NLB Cluster, URL: http://support.microsoft.com/kb/242242/en-us

[7] How Cluster Integrity Is Monitored in WLBS, Microsoft Knowledge Base, URL: http://support.microsoft.com/kb/232711/EN-US/

[8] How Does Network Load Balancing Algorithm Works Internally, URL: http://support.microsoft.com/?kbid=556068&SD=tech