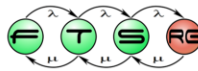


# Fürtözés és replikáció

Micskei Zoltán  
(részben Medgyesi Zoltán munkája alapján)



Utolsó módosítás: 2013. 04. 29.

# Bevezető

- Cél: hibatűrés
  - számítógép hibák tolerálása
  
- Mikor éri meg:
  - Egy géppel elérhető: ~99%-os rendelkezésre állás (évi max 3,5 nap kiesés)
  - Ha ennél jobbat akarunk
  
- Redundancia beépítése

# Tartalom

- **Fürtök**
  - **Fürtök csoportosítása**
  - Terheléselosztó fürtök
  - Feladatátvételi fürtök
  
- **Replikáció**
  - Elsődleges – másodlagos séma
  - Multimaster

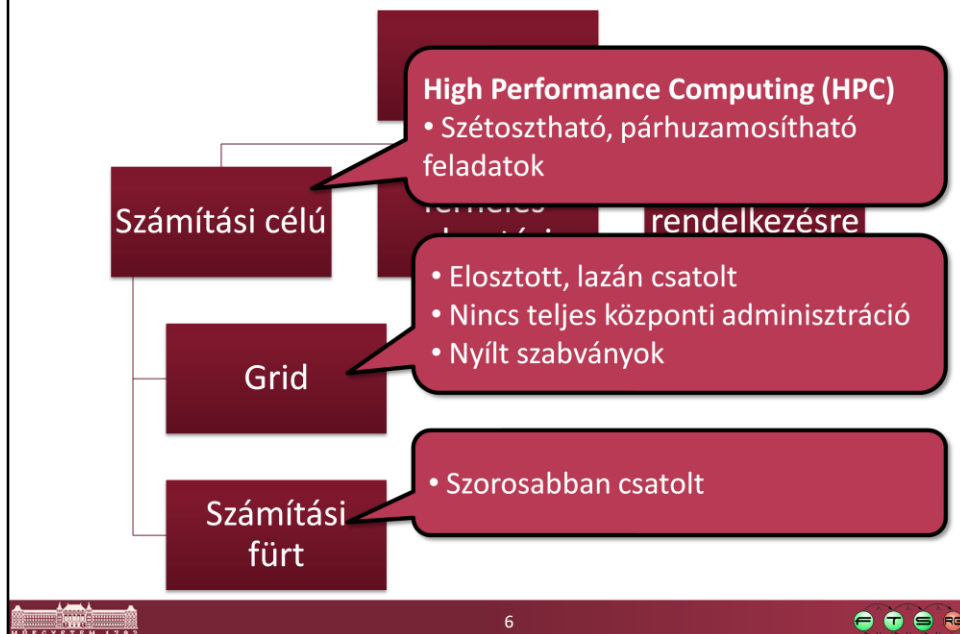
## A számítógépfürt

**Fürt (cluster):** különálló *számítógépek együttese*, amelyek egymással együttműködve és azonos szolgáltatásokat, alkalmazásokat futtatva egyetlen rendszerként, *virtuális kiszolgálóként* jelennek meg az ügyfelek számára.

# Fürtök (egy lehetséges) csoportosítása



## Fürtök (egy lehetséges) csoportosítása

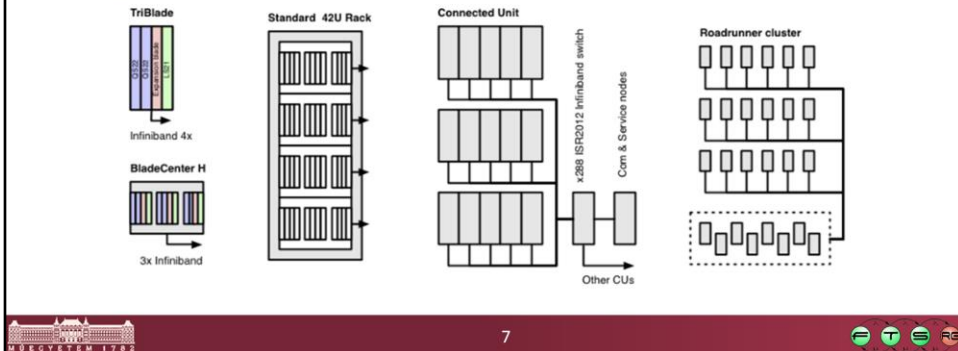


Grid: lásd pl. Ian Foster. „What is the Grid? A Three Point Checklist”, July 20, 2002.  
URL: <http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>

# Számítási fürt példa: IBM Roadrunner



- 12,960 IBM PowerXCell 8i CPU
- 6,480 AMD Opteron dual-core CPU
- Összesen: 130 464 mag



- Forrás: [http://en.wikipedia.org/wiki/IBM\\_Roadrunner](http://en.wikipedia.org/wiki/IBM_Roadrunner)
- A Roadrunner a 2008-as leggyorsabb számítógép, azóta vannak gyorsabbak is.

## Számítási fürt példa: Cray TITAN

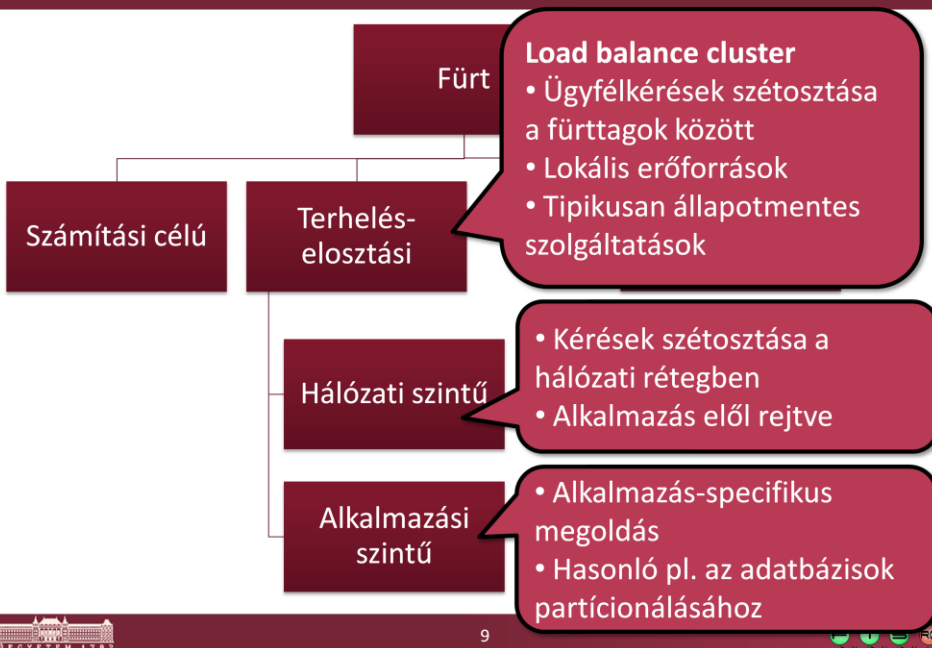


- CPU:
  - 18688 AMD Opteron 6274 16-core CPUs
  - 18688 Nvidia Tesla K20X GPUs
- Memória: 710 TB (598 TB CPU and 112 TB GPU)
- Sebesség: 17.59 petaFLOPS (LINPACK)
- 200 darab rack \* 24 blade

Forrás: [http://en.wikipedia.org/wiki/Titan\\_\(supercomputer\)](http://en.wikipedia.org/wiki/Titan_(supercomputer))



# Fürtök (egy lehetséges) csoportosítása



# Fürtök (egy lehetséges) csoportosítása

## HA cluster

- Szolgáltatás egyik fűrtagon fut, többi tartalék
- Feladatátvétel (failover)

Nagy rendelkezésre állású

- Egy erőforrást egyszerre többen használhatnak
- Alkalmazás szintű zárolás

Megosztott lemezes

- Erőforrás birtoklása kizárólagos

Megosztott elem nélküli

# Tartalom

- **Fürtök**
  - Fürtök csoportosítása
  - **Terheléelosztó fürtök**
  - Feladatátvételi fürtök
  
- **Replikáció**
  - Elsődleges – másodlagos séma
  - Multimaster

## A terheléselosztás dilemmája



### Egyenletes elosztás

- csomópontok terhelésének figyelése
- bonyolultabb elosztó algoritmusok



### Egyszerűség

- kevesebb meghibásodási lehetőség
- kisebb overhead

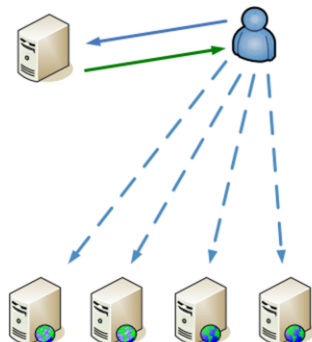


Nem biztos, hogy az a terheléselosztó algoritmus a leghatékonyabb, ami folyamatosan figyeli a csomópontok terhelését, megpróbálja nagyon pontosan megbecsülni az aktuális kérés munkaigényét, és ez alapján nagyon egyenletes elosztást produkálni, mert a sok plusz munka nagy terhelés esetén túl sok időt és erőforrást emészten fel, és emiatt nem lesz hatékony a kiszolgálás.

## Hálózati terheléselosztó fürtök fajtái

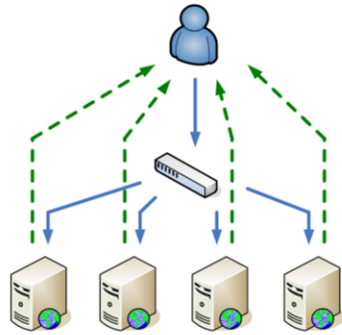
- Round-robin DNS
- Teljesen elosztott
- Központi elemre épülő

## Round-robin DNS



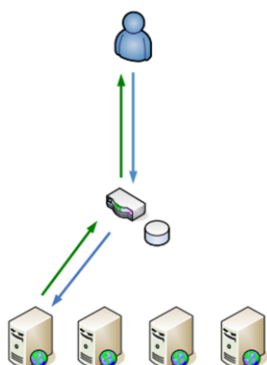
- DNS szerver más-más címet ad vissza kéreésenként
- Pl.: nslookup [www.cnn.com](http://www.cnn.com)
- Előny:
  - egyszerű
  - független fürttagok
- Hátrány:
  - statikus

## Teljesen elosztott NLB fűrt



- Közös IP, MAC cím a fűrtnek
- Kéréseket mindenki megkapja
- Egy csomópont válaszol
- Pl. Microsoft NLB
- Előny:
  - nincs SPOF szétosztó
- Hátrány:
  - Korlátos méret

## Központi elemre épülő



- Központi elosztó (dispatcher)
- Dedikált HW-es megoldások is
- Kifinomult terhelésfigyelés és elosztás
  
- Előny
  - Elosztóban egyéb szolgáltatások (cache, SSL offload...)
  
- Hátrány
  - Elosztó SPOF lehet



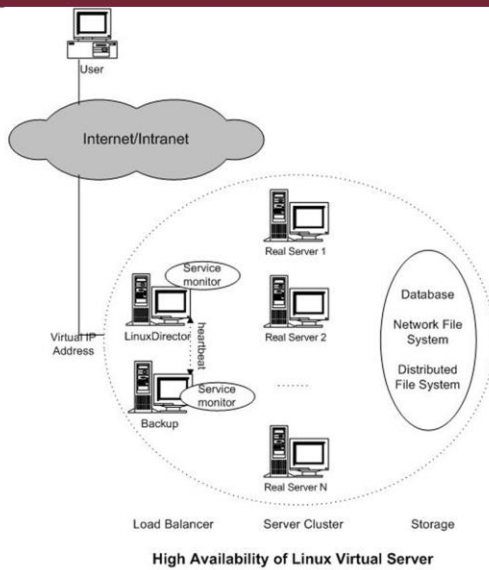
## Probléma: munkamenet megőrzése

- Ügyfél munkamenete tipikusan a webservert memóriájában tárolódik
  - De: ügyfél egymás utáni kéréseit különböző webserverek szolgálják ki
- Terheléselosztó szintű megoldás:
  - Affinitás: adott ügyfél kéréseit mindig ugyanaz a szerver szolgálja ki
- Alkalmazás szintű megoldás:
  - Munkamenet tárolása központi gépen / adatbázisban
  - Munkamenet tárolása a kliensen, elküldése minden kérésben

## Példák: Hálózati terheléselosztók

- RRDNS:
  - majd minden DNS kiszolgáló (bind, MS DNS...)
- Elosztott megoldások:
  - Microsoft Network Load Balancing
- Központi elosztót használó:
  - HW (Cisco, BigIP, Foundry ServerIron, Nortel...)
  - Linux Virtual Server

# Linux Virtual Server

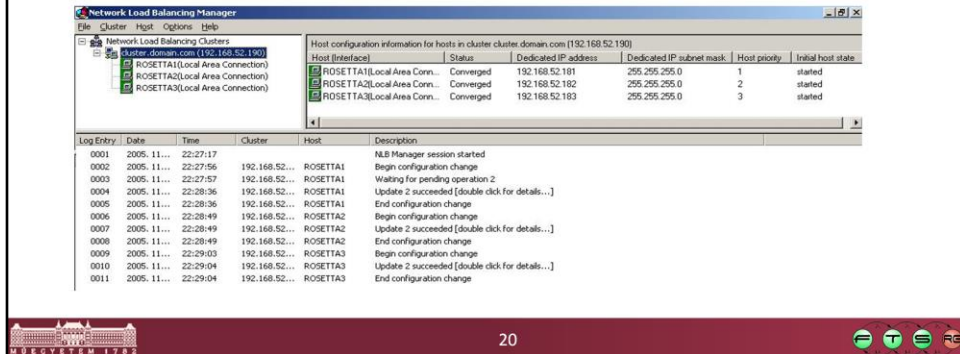


- Elterjedt (pl. sourceforge.net, linux.com...)
- Elosztó: aktív-passzív
- Layer 4 és 7 elosztás



# Microsoft NLB

- maximum 32 csomópont
- kieső kiszolgálók detektálása 10 sec alatt
- Speciális szűrő hálózati meghajtó
- Portszabályok, affinitás



Érdeklődőknek bővebb leírás: Medgyesi Zoltán, Micskei Zoltán. Hálózati terheléselosztó fűrtök. Mérési segédlet, [http://mit.bme.hu/~micskeiz/meres/cluster\\_meres/files/O1\\_load\\_balance\\_clusters\\_sagedlet.pdf](http://mit.bme.hu/~micskeiz/meres/cluster_meres/files/O1_load_balance_clusters_sagedlet.pdf)

# Tartalom

- **Fürtök**
  - Fürtök csoportosítása
  - Terheléselosztó fürtök
  - **Feladatátvételi fürtök**
  
- **Replikáció**
  - Elsődleges – másodlagos séma
  - Multimaster

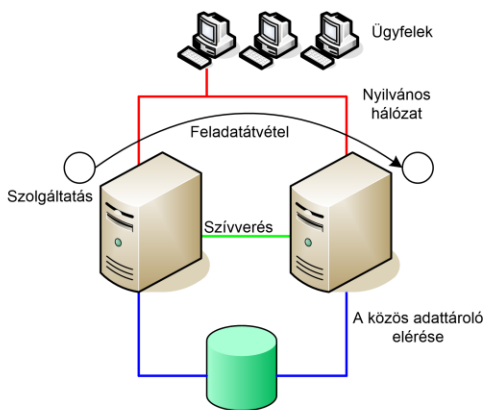
## HA fürtök csoportosítása

- (Klasszikus csoportosítás\*)
- Megosztott lemezes (shared disk)
  - Szolgáltatás több csomóponton fut(hat)
  - Közös erőforrást egyszerre írhatják-olvashatják
  - De: fizikai szinten sorosítás, zárolás használata
  - Pl.: Oracle RAC
- Megosztott elem nélküli (shared nothing)
  - Szolgáltatás egyszerre egy csomóponton fut
  - Egy erőforrást egyszerre egy csomópont birtokol
  - De: fizikai szinten lehet közös elérésű erőforrás

\*M. Stonebraker, The Case for Shared Nothing, 1985, <http://db.cs.berkeley.edu/papers/hpts85-nothing.pdf>



# HA fűrtök - alapfogalmak

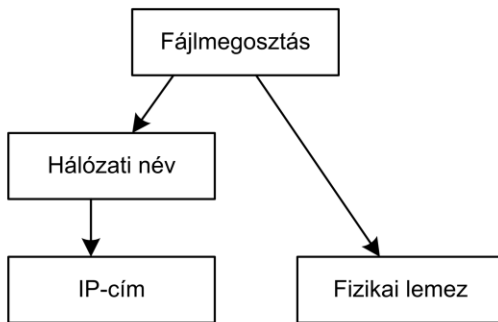


- Csomópont (node)
- Szívverés (heartbeat)
- Feladatátvitel (failover)
- Feladat-visszavétel (failback)
- Átkapcsolás (switchover)

- A csomópontok egymásnak úgynevezett *szívverés* (heartbeat) üzeneteket küldenek, ezek segítségével lehet detektálni, hogy kiesett-e valaki.
- Ha a bal oldali számítógép meghibásodik, akkor a fűrtsoftver érzékeli ezt, a jobb oldali számítógépen elindítja a szolgáltatást, a bal oldali gépen pedig leállítja – ezt nevezzük *feladatátvitelnek* (failover). Ettől kezdve a tartalék gép használja a közös adattárolót és fogadja az ügyfelek kéréseit.
- Ha később a bal oldali gép ismét üzemképesé válik, akkor lehetőség van arra, hogy ismét ez futtassa a szolgáltatást. A feladatátvitellel ellentétes irányú műveletet *feladat-visszavételnek* (failback) nevezzük.
- A legtöbb gyártó megkülönbözteti azt az esetet, amikor a szolgáltatások áttétele hiba miatt történik, és azt, amikor a rendszergazda kezdeményezi a műveletet, például azért, hogy valamelyik fűrttagot ideiglenesen, például karbantartási célból kivehesse a fűrtből. Az ilyen áttételeket *átkapcsolásnak* (switchover), egyes esetekben *felügyeleti feladatátvitelnek* (administrative failover), az ellenkező irányú műveletet pedig *visszakupcsolásnak* (switchback) nevezik.

## HA fürtök - erőforrások

- Minden **erőforrás** (lemez, IP cím, Apache...)
- **Erőforráscsoport**: olyan erőforrások, amiket együtt kell mozgatni



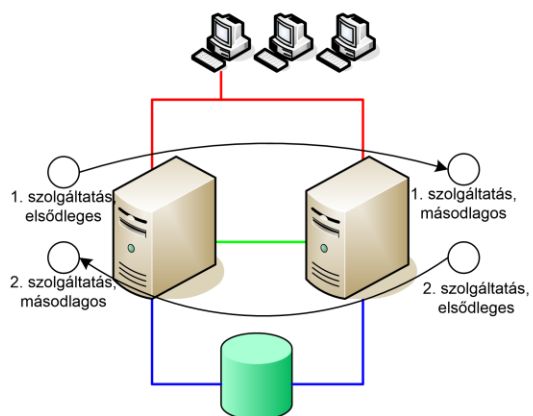
- **Függőségi fa**
- **Erőforrások** leállításának és indításának sorrendje

A *függőségi fa* a fürtben definiált erőforrások közötti függőségeket tartja nyilván.



## Feladatátvételi topológiák (1)

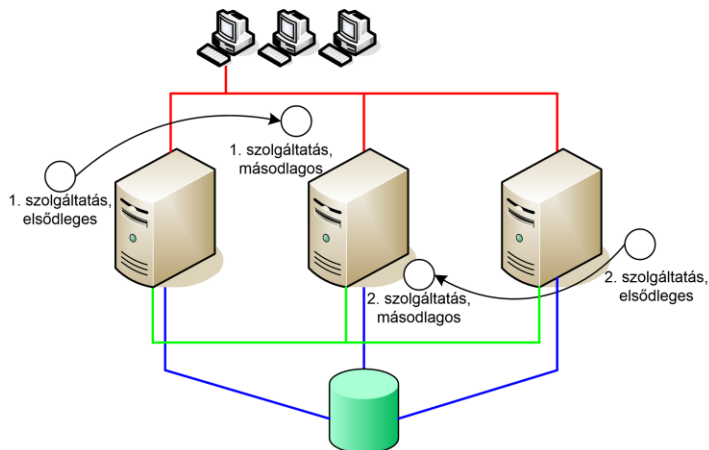
- Feladatátvételi pár (aktív-aktív)



Bár aktív-aktívnek is szokás ezt a sémát hívni, de azt fontos észrevenni, hogy egy szolgáltatás egyszerre csak egy gépen fut a másik gépen ugyanakkor egy másik szolgáltatás aktív.

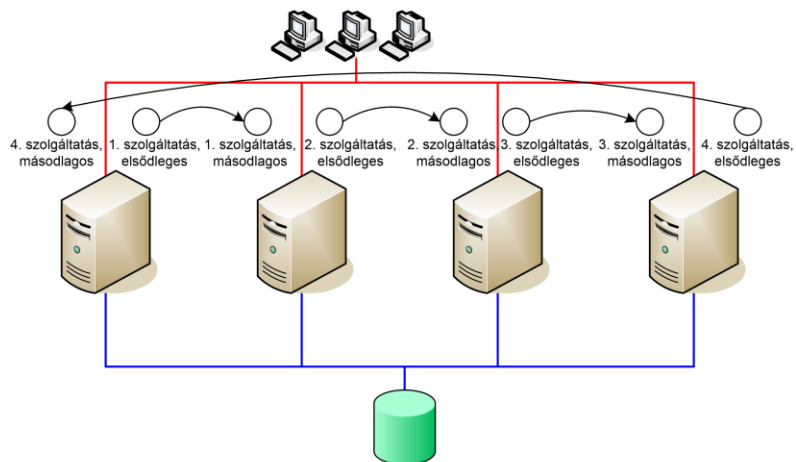
## Feladatátvételi topológiák (2)

- Forró tartalék (N+1)
- Több tartalék (N+M)



# Feladatátvételi topológiák (3)

## ▪ Feladatátvételi gyűrű



## Feladatok, problémák a fűrtökben

- **Tagsági kép fenntartása (group membership):** ki működik a csomópontok közül
- **Csoportkommunikáció (group communication):** üzenetek eljuttatása a többieknek hibák esetén is
- **Tudathasadás (split brain):** fűrt több, független részre szakad
- **Amnézia:** kiesés után újrainduló csomópontot értesíteni a közben történt változásokról
- **Gördülő frissítés (rolling upgrade):** csomópontok frissítése egyesével, többi működik közben

## Megoldások

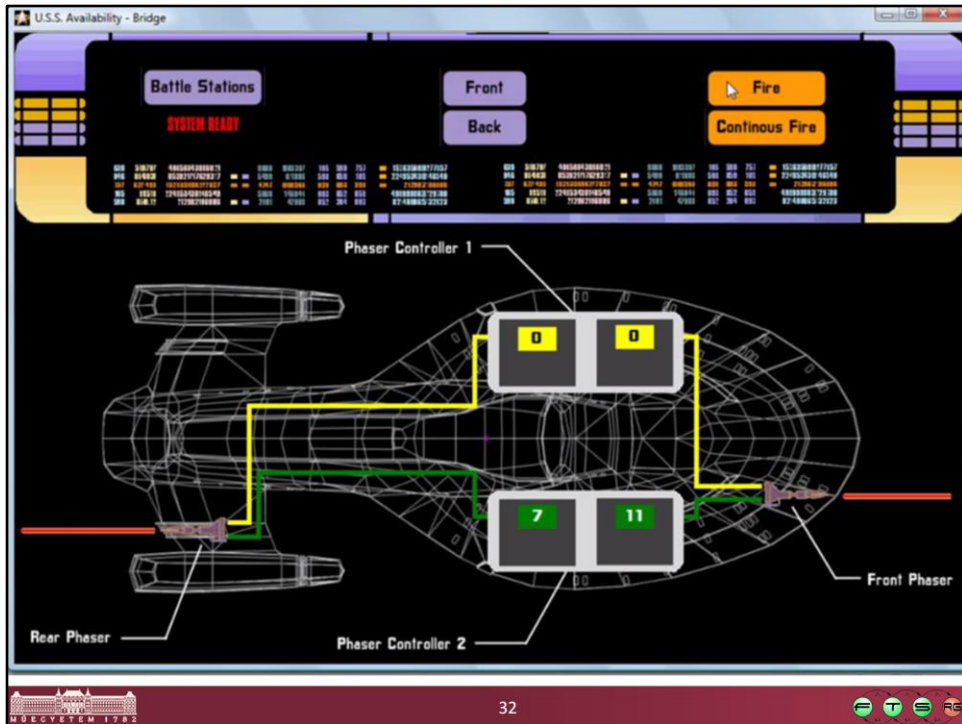
- Sun Solaris Cluster
- IBM High Availability Cluster Multiprocessing
- Oracle Clusterware
- Linux-HA
- SA Forum AIS
- Windows Server Failover Clustering
- VMware vSphere HA
- ...

# Windows Server Failover Clustering

- Maximum 64 csomópont (Windows 8)
- Fürtözhető szolgáltatások: fájl szerver, DHCP, SQL Server, Hyper-V, saját alkalmazás...
- **Quorum** (többség):
  - szavazatok többségének meg kell lenni egy partícióban, hogy az működhessen
  - szavazhat: csomópont, tanú lemez, tanú fájlmegosztás
  - Többféle quorum modell (csomópontok számától függően)



A Windows Server Failover Clusteringet például majd a *Szolgáltatásbiztos rendszertervezés szakirány* laborjában lehet kipróbálni.



Warm-up videó: ([http://www.inf.mit.bme.hu/edu/specialization/presentations/oldalrol\\_elr\\_het\\_o](http://www.inf.mit.bme.hu/edu/specialization/presentations/oldalrol_elr_het_o))

Közvetlen link:

<http://www.inf.mit.bme.hu/sites/default/files/szakirany/demo/msc/uss-availability-demo-hun.avi>

Nagy rendelkezésre állású fűrtök kialakítás az SA Forum AIS szabványának segítségével

# Tartalom

- **Fürtök**
  - Fürtök csoportosítása
  - Terheléelosztó fürtök
  - Feladatátvételi fürtök
  
- **Replikáció**
  - Elsődleges – másodlagos séma
  - Multimaster



## Replikáció

- Adatok tárolása több helyen
- Nem fűrt: kívülről nem egy számítógépként látszik
- Változások szinkronizálása
  - Periodikus / eseményvezérelt átvitel
- Szinkronizáció:
  - Pull / Push
- Melyik adatpéldányt lehet írni:
  - **Primary – secondary** (master – slave): egy írható, többi ennek a másolata, azok csak olvashatóak
  - **Multimaster**: mindegyik példány írható, konzisztencia fenntartása bonyolultabb



A csoportosítás természetesen megint nem fekete-fehér, a fűrtök is használhatnak belül különböző replikációs technikákat.

Szinkronizáció:

-Push: akinél volt a változás, az „nyomja” át a többieknek

- Pull: a replikáció kliensei „húzzák” le a változásokat

# Tartalom

- **Fürtök**
  - Fürtök csoportosítása
  - Terheléselosztó fürtök
  - Feladatátvételi fürtök
  
- **Replikáció**
  - **Elsődleges – másodlagos séma**
  - Multimaster

## DEMO Primary – secondary séma: DNS

- BIND9
- Zóna fájl csak az elsődleges szerveren írható
- Zóna fájl verziózva
- Másodlagos szerverek: zone transfer
  - induláskor, vagy ha az elsődleges értesíti (notify)
  - lehet csak a változásokat (incremental zone transfer)



36



Installing A Bind9 Master/Slave DNS System,  
[http://www.howtoforge.com/debian\\_bind9\\_master\\_slave\\_system](http://www.howtoforge.com/debian_bind9_master_slave_system)

BIND 9 Administrator Reference Manual, <http://www.bind9.net/manuals>

Konfig fájl: /etc/bind/named.conf, /etc/bind/named.conf.local,  
/etc/bind/named.conf.options  
Zóna fájlok: /var/cache/bind könyvtárban

Log üzenetek: /var/log/syslog

Parancsok:

ellenőrzés: named-checkconf, named-checkzone  
adminisztrálás: rndc

----

Zone transfer megnézése:

- masteren:

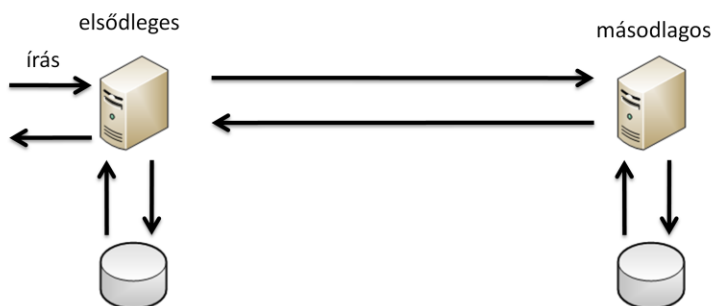
- a zónában módosítani: serial változtatása, új A rekord felvétele
- sudo rndc reload
- tail /var/log/syslog

- slave:

- cat /var/cache/bind/zona\_fajl.db
- nslookup

## Primary – secondary séma: adatbázisok

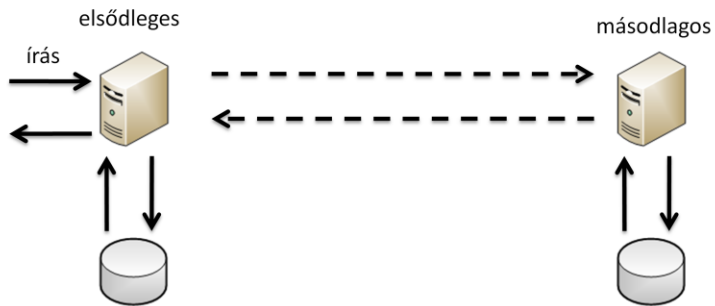
- Szinkron:



- „Zero data loss”, atomi írás
- Teljesítményvesztés az ára

## Primary – secondary séma: adatbázisok

- Aszinkron:



- Helyi írás befejezése után egyből visszatér
- Mi legyen, ha a másodlagos írása közben hiba lesz?

# Tartalom

- Fürtök
  - Fürtök csoportosítása
  - Terheléselosztó fürtök
  - Feladatátvételi fürtök
  
- Replikáció
  - Elsődleges – másodlagos séma
  - **Multimaster**

## Multimaster replikáció: Active Directory

- Multimaster replikáció
  - bármelyik DC-n módosíthatunk
- Flexible Single Operations Master (FSMO)
  - 5 szerep, amiből egyszerre csak egy lehet
  - RID master, Schema master...
- Optimalizációk
  - csak a változott attribútum megy át
  - store and forward elv: változások továbbterjesztése

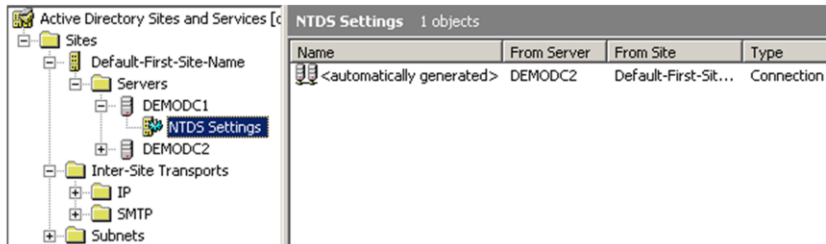


### Active Directory Replication Technologies

<http://technet2.microsoft.com/windowsserver/en/library/53998db6-a972-495e-a4e7-e3ca3f60b5841033.mspx>

# Replikációs topológia

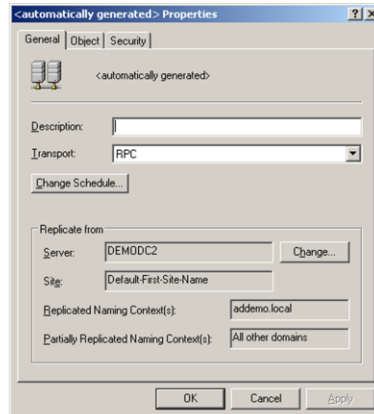
- Telephely: gyors kapcsolattal összekötött DC-k
  - Intra-site: gyakori replikáció, RPC
  - Inter-site: ritkábban, IP/SMTP
- Knowledge Consistency Checker
  - Topológia automatikus létrehozása és frissítése





## DEMO Active Directory replikáció

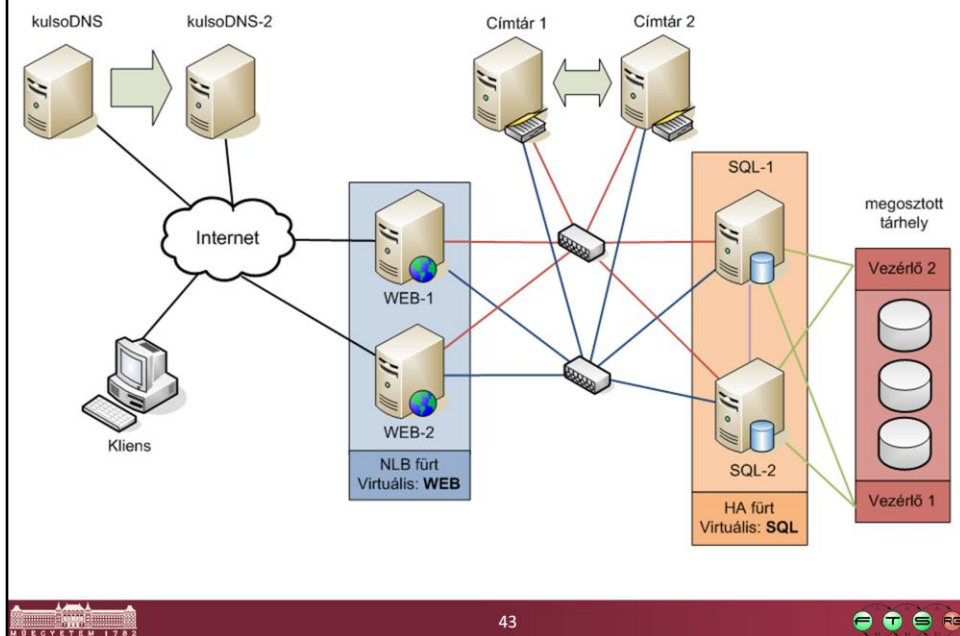
- Változás nyomon követése
- AD Sites and Services
  - Replikáció kikényszerítése
  - Telephelyek beállítása
- Ütközés feloldása
- Eseménynapló



### Ütközés:

- legyen mindkét DC Global Catalog
- szakítsuk meg a kapcsolatot közöttük
- hozzuk létre ugyanolyan cn-ű objektumot
- kapcsolat vissza, replikáljunk
- korábbi objektumot átnevezi a replikációs komponens
- Eseménynapló / Directory Service

# Technikák alkalmazása



## További információ

- Medgyesi Zoltán: [Nagy rendelkezésre állású kiszolgálófürtök vizsgálata](#), Diplomamunka, BME, 2007.
- Szolgáltatásbiztonságra tervezés labor, MSc [segédanyagok](#) (terheléselosztás, feladatátvétel)



- [http://mit.bme.hu/~micskeiz/education/onlab/medgyesi\\_zoltan/medgyesi-zoltan-diploma.pdf](http://mit.bme.hu/~micskeiz/education/onlab/medgyesi_zoltan/medgyesi-zoltan-diploma.pdf)
- <http://www.inf.mit.bme.hu/edu/courses/szbtlab>

# Összefoglalás

- Fürtök, replikációs módszerek
- Többféle technika a számítógép és hálózati utak kiesésének kivédésére
  - Különböző előnyök és hátrányok
  - Különböző bonyolultság és költség
- DE: fürt se véd minden ellen
  - katasztrófa, adminisztrátor hibája, rongálás...
  - Kombinálni kell más módszerekkel