



Kritikus architektúrák laboratórium

Feladatátvételi fürtök

Mérési segédlet

Készítette: Medgyesi Zoltán, Micskei Zoltán, Honfi Dávid

Utolsó módosítás: 2016. 04. 06.

Verzió: 2.5

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

1 Bevezető

A labor mérései során kritikus rendszerek architektúráját és működését vizsgáljuk különböző technikák segítségével. A rendszer lehetséges felépítési alternatíváit modellezéssel és mérésekkel elemezzük, az így azonosított hibamódok és szűk keresztmetszetek kiküszöbölésére pedig különböző hibatúrést és skálázhatóságot segítő technológiákat próbálunk ki a gyakorlatban.

Jelen mérés során megismerünk egy módszert arra, hogy a rendszerünkben lévő, állapottal rendelkező szolgáltatások rendelkezésre állását hogyan tudjuk növelni.

2 Feladatátvételi fürtök

A feladatátvételi fürtök célja, a terheléelosztó fürtökkel ellentétben, elsődlegesen a rendelkezésre állás növelése. A fürt által nyújtott szolgáltatást általában egy csomópont nyújtja egyszerre, ennek meghibásodása esetén a többiek közül valaki átveszi a szerepét, így csökkentve a szolgáltatás kiesését. Általánosságban elmondható az ilyen fürtökről, hogy legtöbbször állapottal rendelkező szolgáltatásokat futtatnak, a fürt tagjai szorosabban csatolt rendszert alkotnak, a fűrthöz szükséges hardvereszközökkel és szoftverkörnyezettel szemben pedig magasabb elvárásokat és speciális igényeket támasztanak.

2.1 Alapvető fürt modellek

Három alapvető modellbe szokták sorolni a feladatátvételi fürtöket¹.

2.1.1 Megosztott lemezes modell

A *megosztott lemezes* (shared disk) modellben az egyes fűrtagok közös be- és kiviteli alrendszeren keresztül érik el a közös adattárolón lévő adatokat, és akár egyszerre is írhatják-olvashatják azokat. Az egyidejű hozzáférési lehetőségéből fakadóan a szolgáltatások, legyen szó akár adatbázis-kezelésről, egyszerre több fűrtagon is futtathatók. Az egyidejűség természetesen csak alkalmazási szinten biztosított, fizikai szinten – a konzisztencia megőrzése miatt is – gondoskodni kell a műveletek sorosításáról és a kizárólagos hozzáférésről. Ezeket a feladatokat a globális vagy elosztott zárolási szolgáltatás látja el.

A megosztott lemezes modellt alkalmazva elméletileg kiváló rendelkezésre állás érhető el, hiszen valamelyik fűrtag meghibásodása semmilyen formában nem érinti a többi fűrtag működését, a szolgáltatás futtatását egyetlen pillanatra sem kell megszakítani. Ugyanakkor a gyakorlatban több problémával is számolni kell. Egyrészt meghibásodás esetén a hibás fűrtagot meg kell akadályozni a közös adattárolóra vonatkozó zárolás fenntartásában és az adatok helytelen módosításában, másrészt hiba esetén újra kell konfigurálni a fürtöt, újra el kell osztani a beérkező kérések kezelését a tagok között.

A megosztott lemezes megoldás előnye, hogy további fűrtagok hozzáadásával könnyen skálázható, hátránya ugyanakkor, hogy a közös adattároló könnyen szűk keresztmetszetté válhat. Mindig a tényleges alkalmazástól, elsősorban a lemezeire történő írások és az olvasások mennyiségétől, arányától függ, hogy a modell mennyire állja meg a helyét.

Tipikus példája a modellnek az Oracle adatbázis-kezelőjéhez készített Real Application Clusters kiegészítés és az alatta lévő Oracle Clusterware.

2.1.2 Megosztott elem nélküli modell

A *megosztott elem nélküli* (shared nothing) modellben egyszerre minden logikai és fizikai erőforrást kizárólag egy fűrtag birtokolhat és kezelhet. A kizárólagos hozzáférés csak logikai szinten értendő, fizikai szinten mindegyik fűrtagnak csatlakoznia kell például a közös adattárolóhoz, hiszen ennek hiányában az elsődleges fűrtag meghibásodásakor a másodlagos tag nem tudná elérni az adatokat, és

¹ Vigyázat: Ezeket a fogalmakat máshogy is szokták definiálni, például ha az adatbázis szemszögéből nézik, akkor mást hívnak shared nothingnak.

nem tudná futtatni a szolgáltatásokat. Mivel ebben az esetben nincs egyidejű többes hozzáférés, a lemezéléréshez nincs szükség globális zárolásra, ellenben az elérés kizárólagosságát – vagyis azt, hogy egyszerre mindig csak egy fürttag használhassa az adattárolót – garantálni kell.

Egy fürt alapvető architektúrája sokszor ellentmondásosnak tűnhet. A Microsoft feladatátvételi fürtje például a szolgáltatáshoz tartozó adatokat szintén közös elérésű lemezen tárolja, mégis megosztott elem nélküli architektúrájának mondják. A különbség a megosztott lemezes modellhez képest az, hogy ennél a megoldásnál egyszerre csak egy fürttag birtokolhatja és kezelheti a közös erőforrást, a modell megosztott elem nélküli jellege tehát nem sérül.

2.1.3 Replikált lemezes modell

A replikált vagy tükrözött lemezes modellben jellemzően két fürttag szerepel, egy aktív és egy tartalék. Miközben az aktív tag kiszolgálja az ügyfeleket, a rendszer folyamatosan tükrözi az adatok módosításait a tartalék tagra. Ha az aktív tag meghibásodik, a tartalék bármely pillanatban naprakész adatokkal tudja átvenni a feladatokat.

A replikáció többféle szinten és eszközzel is megvalósítható:

- *Binárisan*: A replikáció legalacsonyabb szintű módja a lemezre került adatok bináris továbbítása a tartalék adattárolóra. Ilyen megoldásokat a tárolórendszerekkel foglalkozó cégek, például az EMC kínálatában lehet találni.
- *Fájl- vagy fájlrendszeri szinten*: Ha a szolgáltatás fájlokkal dolgozik, akkor célszerű lehet fájl szinten replikálni az adatokat, ilyen szoftvert is több gyártótól lehet vásárolni.
- *Integrált/alkalmazásszintű replikáció*: Ha a szolgáltatás például adatbázis-kezelést végez, akkor a fájl szintű replikáció célszerűtlen, inkább a változásadatok továbbítására kell törekedni. Példaként az Oracle Streams említhető, amely adatfolyam formájában továbbítja a tartalék adatbázis felé az adatokat, tranzakciókat és egyéb eseményeket.

A replikációs eszköz a fürtszoftverrel is integrálható, illetve a két megoldás együttműködhet egymással.

2.2 A feladatátvételi fürtökkel kapcsolatos alapfogalmak

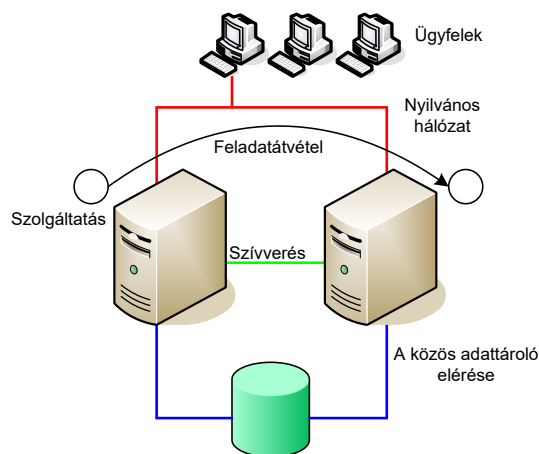
Az alábbi ábrán (1. ábra) egy alapszintű, két tagból álló fürt látható. A nagy rendelkezésre állású szolgáltatás alapesetben a bal oldali gépen fut, ez fogadja az ügyfelek kéréseit. Az adatok tárolása egy *közös adattárolón* történik. A fürttagok *szívverések* (heartbeat), rövid hálózati üzenetek továbbításával jelzik egymásnak a működőképességüket.

Az ilyen fürtöknél általában fontos kikötés, hogy a fürttagok konfigurációja hardver és szoftver szempontjából azonos legyen; erre nemcsak a szolgáltatás zökkenőmentes futtatása, de a megfelelő teljesítmény méretezése miatt is szükség lehet.

Ha a bal oldali számítógép meghibásodik, akkor a fürtszoftver érzékeli ezt, a jobb oldali számítógépen elindítja a szolgáltatást – ezt nevezzük *feladatátvételnek* (failover). Ettől kezdve a tartalék gép használja a közös adattárolót és fogadja az ügyfelek kéréseit.

Ha később a bal oldali gép ismét üzemképessé válik, akkor lehetőség van arra, hogy ismét ez futtassa a szolgáltatást. A feladatátvétellel 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 fürttagot ideiglenesen, például karbantartási célból kivehesse a fürtből. Az ilyen áttételeket *átkapcsolásnak* (switchover), egyes esetekben *felügyeleti feladatátvételnek* (administrative failover), az ellenkező irányú műveletet pedig *visszakapcsolásnak* (switchback) nevezik.

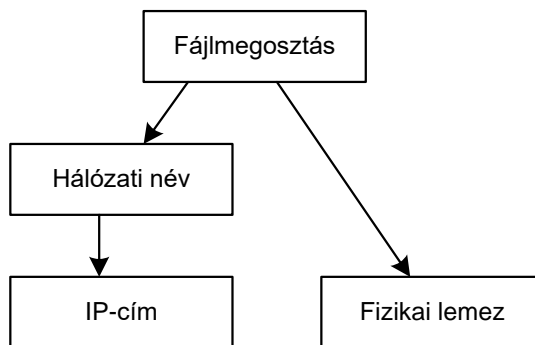


1. ábra: Alapszintű feladatátvételi fürt

A feladatátvétel és az átkapcsolás kapcsán fontos megemlíteni, hogy az előbbi esetében hiba történik, tehát nagyobb a valószínűsége az adatsérülésnek vagy az adatvesztésnek, a szolgáltatás pillanatnyi állapota elveszhet. Az átkapcsolásnál elméletileg minden rendszerelem kifogástalanul működik, a szolgáltatás felé lehet jelezni, hogy zárja le az éppen futó kiszolgálásokat, ez a művelet tehát minimális kockázattal jár.

2.3 Fürtözési alapelemek

Az elmúlt évtizedekben jó néhány feladatátvételi fürtözési megoldás született, ám vannak olyan alapvető elemek és szolgáltatások, amelyek valamilyen formában – esetleg más névvel, más struktúrában – gyakorlatilag minden megvalósításban megtalálhatók, ezek alkotják a fürtök üzemeltetéséhez szükséges alapvető infrastruktúrát. Sok fejlesztő további szolgáltatások, lehetőségek beépítésével próbálta versenyképesebbé tenni a saját megoldását.



2. ábra: Példa függőségi fára

A fürtök minden fizikai és logikai eszközt *erőforrásként* (resource) kezelnek. A fürt számára egyaránt erőforrás a fizikai lemez, a hálózati név, az IP-cím vagy a fürtözött szolgáltatás, például egy adatbáziszserver vagy újabban akár egy virtuális gép is.

Az erőforrások *erőforráscsoportokat* alkotnak. A fürtsoftver feladatátvételtkor és feladat-visszavételkor erőforráscsoportokkal dolgozik, ezzel biztosítva, hogy az egymástól függő erőforrások a megfelelő fürttagra kerüljenek.

Az erőforrások közötti függőségeket a *függőségi fával* ábrázolhatjuk. Egy erőforrás akkor függ a másiktól, ha az szükséges a működéséhez. Maga a függőségi fa általában ugyan nem jelenik meg a fürtsoftverben, de a függőségeket meg kell adnia a fürt üzemeltetőjének vagy a fürtözött szoftver fejlesztőjének. Függőség csak azonos erőforráscsoportban található erőforrások között lehet, ugyanis az erőforráscsoportok egymástól függetlenül indíthatók, állíthatók le és helyezhetők át a fürttagok között.

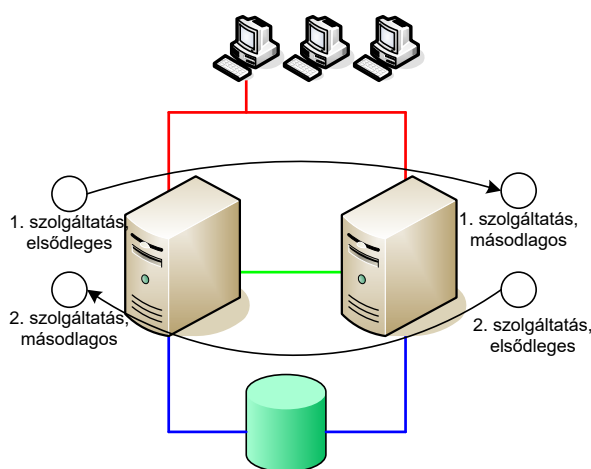
Feladatátvételtkor a forrás fürttagon a fürtsoftver felülről lefelé haladva követi a függőségi fát, mindig azokat az erőforrásokat állítva le, amelyektől már nem függenek más erőforrások. A cél fürttagon

fordított a folyamat, a fűrtsoftver alulról haladva építi fel a fát, tehát minden erőforrást úgy indít el, hogy a működéséhez szükséges más erőforrások már készen állnak.

2.4 Feladatátvételi topológiák

Az első szakaszban említett modellek közül – bár mindegyikre találunk megvalósítást – a megosztott elem nélküli a leginkább elterjedt. A megosztott elem nélküli modell jellegzetessége a feladatátvétel. A fűrtök kiépítésekor az egyes számítógépek között többféle feladatátvételi topológia is kialakítható, ez határozza meg, hogy mely fűrttagok szolgálnak egymás tartalékaiként, illetve az egyes fűrttagok milyen szerepet (aktív kiszolgálói, tartalék) játszanak. Ki kell emelni, hogy a topológiák a fűrtsoftverektől gyakorlatilag függetlenek, kialakításuk – a megfelelő feladatátvételi szabályok megadásával – a rendszergazda feladata.

2.4.1 Feladatátvételi pár



3. ábra: A feladatátvételi pár topológia

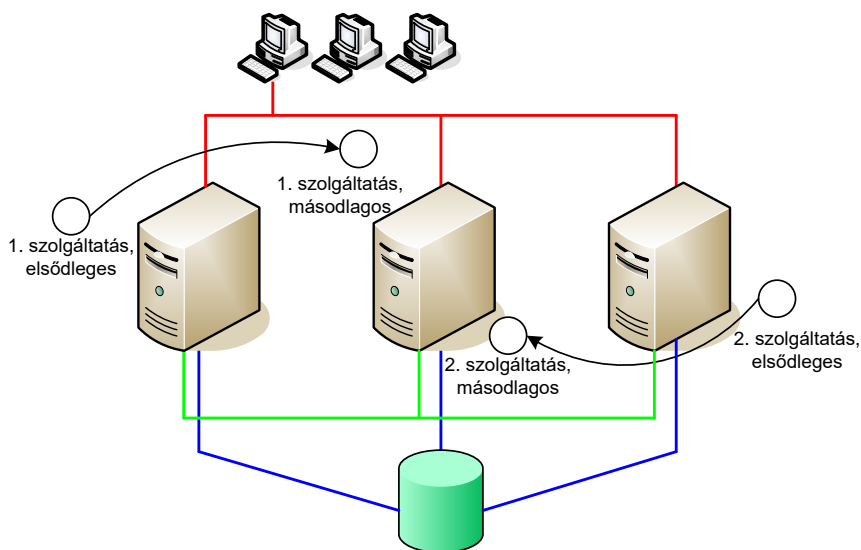
A feladatátvételi pár a legegyszerűbb topológia. Alapesetben egyetlen alkalmazást futtat az elsődleges kiszolgálón, és ennek meghibásodása esetén a másodlagos kiszolgáló veszi át a feladatot, és vele együtt a mindkét tag által elérhető adattároló kezelését. Az alapkiépítés a különféle elemek megkettőzésével, például kettős ügyféloldali hozzáféréssel vagy szívverés-továbbítási kapcsolattal bővíthető.

Ha alapesetben csak az egyik fűrttag futtat szolgáltatást, a másik pedig várakozik, akkor aktív-passzív, ha pedig mindkettő futtat szolgáltatást, akkor aktív-aktív topológiának is szokás nevezni a fenti elrendezést.

A feladatátvételi pár hátránya, hogy csak az egyik fűrttag van kihasználva, a másik csak feladatátvétel esetén jut szerephez. Ezen úgy lehet javítani, hogy az ábrán látható módon mindkét fűrttag elsődleges kiszolgálóként futtat egy-egy alkalmazást, és meghibásodás esetén a másik fűrttagra történik a feladatátvétel – vagyis az előző bekezdésben említett aktív-aktív megoldást használjuk. Ebben az esetben ügyelni kell arra, hogy mindkét fűrttagnak elegendő kapacitással kell rendelkeznie mindkét szolgáltatás egyidejű biztosításához.

2.4.2 Forró tartalék

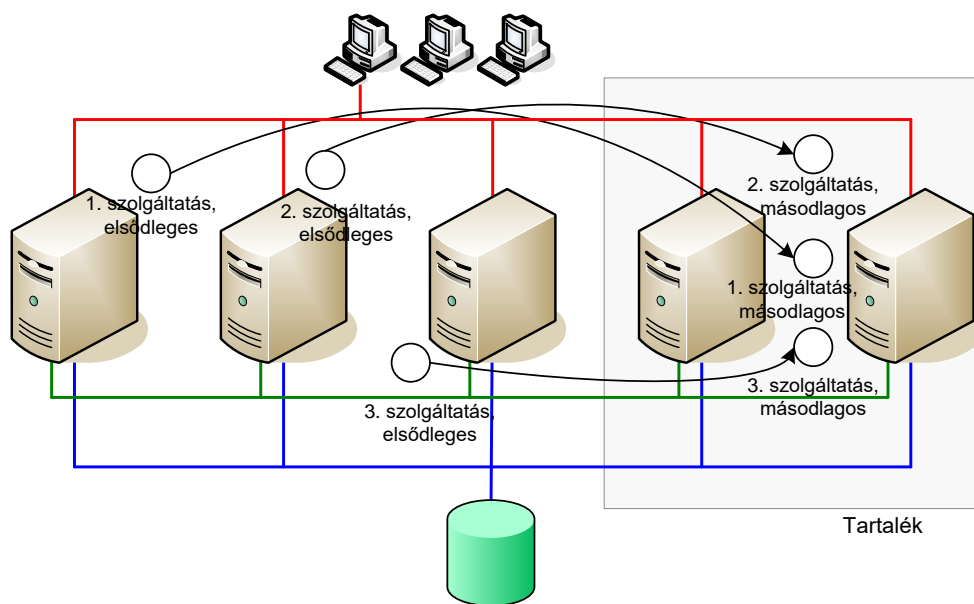
A forró tartalék vagy N+1 (N aktív számítógép, 1 tartalék) topológiában mindegyik szolgáltatás a saját fűrttagján fut, meghibásodás esetén a feladatátvétel ugyanarra a tartalék számítógépre történik. Ez a konfiguráció is viszonylag könnyen összeállítható és kezelhető, de a tartalék számítógépet úgy kell méretezni, hogy akár mindegyik elsődleges fűrttag meghibásodása esetén is képes legyen futtatni a szolgáltatásokat. Szükség esetén ebben az esetben is többszörözhető a szívverések továbbítására és az ügyféloldali hozzáférésre használt hálózat, valamint a közös adattároló elérési kapcsolatai.



4. ábra: A forró tartalék topológia

2.4.3 N+I topológia

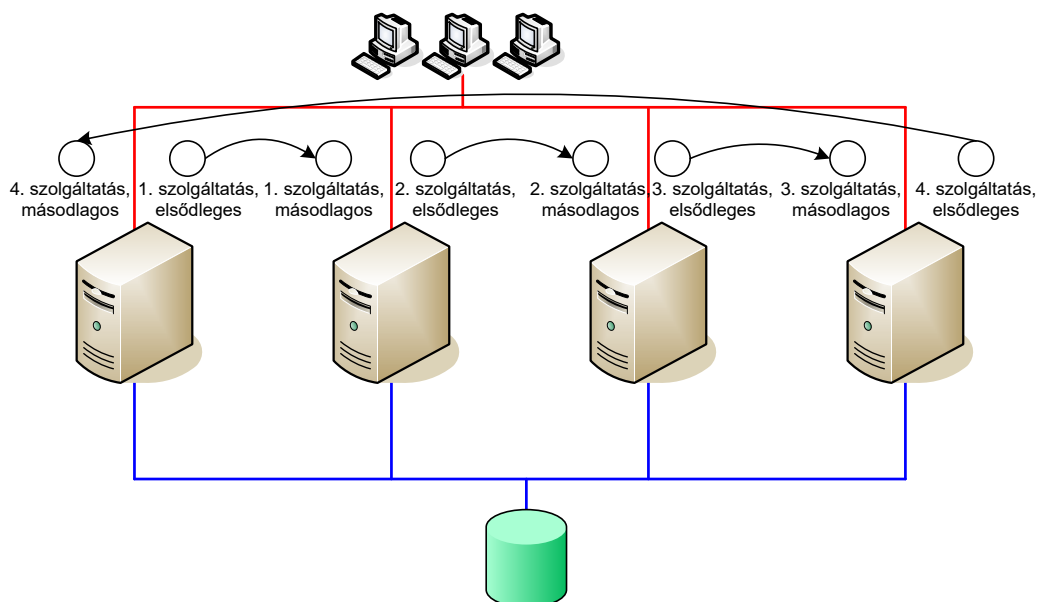
Az N+I topológiában az N számú aktív fűrttagra I számú tartalék számítógép jut, hiba esetén a szolgáltatások – a házirendtől függően – a tartalék gépek bármelyikére kerülhetnek. Ennél a megoldásnál egyrészt könnyű a kapacitástervezés, és a többszörös hibák is jól kezelhetők, másrészt ezeket az előnyöket a kihasználatlanul várakozó fűrttagok viszonylag nagy számával kell megfizetni.



5. ábra: Az N+I topológia

2.4.4 Feladatátvételi gyűrű

A feladatátvételi gyűrűben mindegyik fűrttag az előtte lévőnek a tartaléka, valamint az utolsó fűrttagról az elsőre kerülnek a szolgáltatások hiba esetén. Főként több kisebb alkalmazás fűrtözésekor használható, és viszonylag könnyű kapacitástervezést tesz lehetővé; ugyanakkor több fűrttag meghibásodása esetén egyenetlen terheléeloszlást eredményezhet, és a rendszergazda számára nehéz lehet a topológia áttekintése.



6. ábra: A feladatátvételi gyűrű topológia

2.5 Jellegzetes problémák a fűrtökben

A fűrtök kapcsán ki kell emelni néhány jellegzetes, elméleti problémát.

- **Tudathasadás (split brain):** Tudathasadás akkor történik, ha a fűrt a hálózati kapcsolat megszakadása miatt két részre bomlik, és mindkét rész azt hiszi, hogy a másik rész meghibásodott. Ilyenkor mindkét rész fogadja az ügyfelek kéréseit és megpróbál hozzáférni a közös adattárolón lévő adatokhoz. Mivel a fűrtöt nem ilyen működésre terveztük, a tudathasadás eredménye adatsérülés, adatvesztés lehet. A tudathasadást mindegyik fűrtsoftver meg tudja előzni azzal, hogy a megfelelő eszközökkel minden esetben garantálja, hogy a fűrt valamelyik része többséget, *quorumot* alkosson. A fűrtnek mindig csak a többséget alkotó része viheti tovább a szolgáltatásokat, a másik résznek le kell állnia, és nem szabad használnia az adatokat.
- **Amnézia (amnesia):** Az amnézia kialakulásának menete a következő. Adott egy fűrt például két fűrttaggal. Az első gép meghibásodik, a második rendben átveszi tőle a feladatokat. A rendszergazda leállítja az első gépet, megjavítja, majd újra üzembe helyezi, ám ekkor a második gép is meghibásodik. Ha időközben bármilyen konfigurációmódosítás történt, akkor az első gép elavult információkkal rendelkezik a fűrtől, és nem szabad aktiválni rajta a szolgáltatást. Az amnézia kialakulása azzal kerülhető el, hogy a fűrtkonfigurációt a közös adattároló eszközre írjuk, és lehetővé tesszük a fűrthöz csatlakozó vagy a fűrt első tagjaként induló számítógép számára a konfigurációs adatok elérését. Az amnéziát *időbeli particionálódásnak* (partition in time) is nevezik.
- **Szoftverfrissítés:** Időről időre szükségessé válhat a fűrttagokon futó operációs rendszer és alkalmazások frissítése. A művelet két szempontból kritikus, egyrészt azért, mert ha a frissítéssel megváltozik az alkalmazások viselkedése, akkor az újabb és a régebbi változat együttélése problémákat okozhat, másrészt azért, mert a frissítés véglegesítése sokszor újraindítást tesz szükségessé, ami szolgáltatáskieséshez vezet – csak hogy a fűrt célja éppen ennek az elkerülése. A fűrttagok frissítését *gördülő frissítéssel* (rolling upgrade) szokás végezni. Ennek során a rendszergazda mindig kiléptet egy-egy fűrttagot, elvégzi a frissítését, majd újra üzembe helyezi. Ahogy a művelettel végighalad az összes fűrttagon, úgy „végiggördül” a frissítés az összes gépen.
- **Egyszeres hibapontok (single point of failure, SPOF):** A fűrtépítés során a cél a rendelkezésre állás növelése, tehát olyan struktúrát kell kialakítani, amelyben nincs egyszeres hibapont, vagyis olyan elem, amelynek a meghibásodása a teljes fűrt leállításához és a szolgáltatások működésének megszakadásához vezetne. Ennek érdekében nem elég több számítógépet beléptetni a fűrtbe, de az összes hálózati kapcsolatot meg kell kettőzni, ideértve az ügyfelek hozzáférést biztosító internetkapcsolatot is, az adattárolás céljára pedig redundáns alrendszert kell választani. Ha az egyszeres hibapontokat tágabb értelemben kezeljük, akkor a környezeti hibaforrásokat is figyelembe kell vennünk, gondoskodva a redundáns áramellátásról, a redundáns légkondicionálásról stb.

3 A Windows Server Failover Clustering

A *Windows Server Failover Clustering*² (WSFC) a Windows Server 2016-be beépített nagy rendelkezésre állást biztosító fürtözési technológia. A legújabb verzióban már 64 csomópont képes együttműködni egy fürtben.

A fürt kezdetekben shared nothing architektúrájú megoldás volt, a közös erőforrásokat egyszerre csak egy csomópont birtokolhatta. Ez a 2008 R2 változatban bevezetett Cluster Shared Volumes (CSV) funkcióval megváltozott, ez lehetővé teszi, hogy bizonyos helyzetekben több csomópont is írjon és olvasson egy megosztott lemezről.

3.1 A Windows Server Failover Clustering működése

A WSFC is az első fejezetben ismertetett általános fogalmakkal dolgozik: erőforrásokat, közöttük lévő függőségeket és erőforráscsoportokat definiálhatunk. Ezekon kívül a következő fogalmakat használja még.

Fürtözött szerepek (clustered roles)³: olyan erőforráscsoport, ami IP-címet, hálózati nevet, közös lemezt és valamilyen alkalmazás erőforrást tartalmaz, és így tulajdonképpen egy virtuális szervert alkot. A kliensek ehhez a virtuális szerverhez csatlakoznak, és nincs is tudomásuk róla, hogy az általuk elért szolgáltatást egy fürtözött rendszeren futtatjuk. A fürtözött szerepekhez rendelhetünk prioritásokat, a fürt szoftver ez alapján dönt az elindítási sorrendjükről és az esetleges leállításukról erőforráshiány esetén.

Ügyfél-hozzáférési pont (client acces point): IP-cím és hálózati név együttese, ami meghatározza, hogy egy adott szerepet vagy magát a fürt menedzsment felületét hogyan lehet elérni.

Fürtadatbázis (cluster database): Itt tárolódnak a fürt adatai, az aktuális értékek mellett a fürtöt érintő változások is mind naplózva vannak (pl. csomópont kiesése, csatlakozása, új erőforrás hozzáadása). Ezt a változásnaplózást hívják *Quorum Logging*nak, és a szerepe az, hogy a kieső csomópont újraindulás után ebből értesülhet, hogy mi történt közben. A fürtadatbázisból minden csomópont tárol egy példányt helyileg, a fürt szoftver feladata biztosítani azt, hogy minden jól működő csomópontnál a legfrissebb verzió legyen belőle.

Tanú lemez (witness disk): olyan megosztott lemez, ami a fürtadatbázist tartalmazza. A lemezt olyan csatolón és protokollon keresztül kell elérni, ami támogatja, hogy valaki kizárólagos módon lefoglalja az erőforrást, és más ne férhessen olyankor hozzá.

Tanú fájl megosztás (witness file share): olyan, a fürtön kívüli fájlserveren lévő megosztás, ami a fürt több részre szakadása esetén segíthet eldönteni, hogy melyik fele működjön tovább. Nem tárolja a teljes fürtadatbázist, csak olyan információkat, amiből eldönthető, hogy mikor történt a fürt állapotában a legutóbbi változás, melyik csomópontok érhetőek el, stb.

Erőforráscsoport-tulajdonosok: az erőforráscsoportokhoz megadhatjuk, hogy melyik fürttagokon szeretnénk őket futtatni (*preferred owner*), és mely tagok azok, akik egyáltalán birtokolhatják az egyes erőforráscsoportokat (*possible owner*). Ezeknek az értékeknek a megfelelő beállításával alakíthatjuk ki az előző fejezetben ismertetett feladatátvételi topológiákat.

Cluster Shared Volume (CSV): a WSFC fürtözött fájlrendszere, azaz olyan fájlrendszer, ami fel van arra készítve, hogy egyszerre több csomópont is használja. Az NTFS fájlrendszerre épül rá, azt egészíti ki elosztott metaadat- és zárkezeléssel. Alapvetően Hyper-V virtuális gépek tárolására lett kifejlesztve, de a 2012-es verziótól kezdve más szerepek is használhatják⁴.

Karbantartási mód (maintenance mode): ha egy csomópontot karbantartási módba kapcsolunk, akkor a fürt szoftver áthelyezi a rajta lévő szerepeket más csomópontokra, és megakadályozza, hogy egy esetleges későbbi feladatátvétel esetén visszakerüljön rá valami.

² A Windows Server 2003-ban található változat neve Microsoft Cluster Service (MSCS) volt. Azóta azonban nemcsak átnevezés történt, hanem jelentős architektúrabeli változásokat is végrehajtottak.

³ A korábbi verzióban fürtözött alkalmazásoknak vagy szolgáltatásoknak hívták.

⁴ A CSV részletes leírás túlmutat a jelen segédleten, további hasznos információ a működéséről pl. itt olvasható [9][10].

Cluster-Aware Updating (CUA): olyan komponens, ami a csomópontokra az operációs rendszer frissítések telepítését automatizálja a korábban bemutatott gördülő frissítések elv implementálásával.

3.2 Quorumfajták

A *quorum* azt a szavazati többséget jelenti, ami mellett még működhet a fürt. *Szavazatot* (vote) kaphatnak a csomópontok, a tanú lemez és a tanú fájl megosztás. Az aktuálisan beállított quorum fajtájától függ, hogy ezekből hánynak kell működnie, hogy a WSFC ne állítsa le a fürt működését. (Jelentős terminológiai változás, hogy a 2008 előtti változatokban a quorum a quorumlemez jelentette, és csak annak volt szavazata.)

A következő lehetőségeink vannak:

- *Csomópont többség* (Node Majority): a csomópontoknak van szavazata, nem szükséges közös elérésű tanúlemez hozzá. Akkor javasolt, ha páratlan számú csomópont van (lásd 7. ábra).

Node Majority Quorum Configuration, Three Nodes

Two nodes out of three in communication:
the cluster runs



Individual nodes not in communication:
the cluster stops



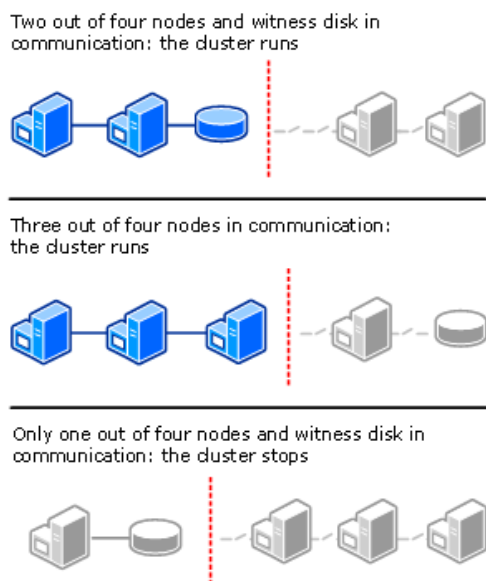
7. ábra Csomópont többségű quorum (Forrás: Technet)

- *Csomópont és lemez többség* (Node and Disk Majority): a csomópontoknak és a tanú lemeznek is van szavazata, páros számú csomópont esetén javasolt. A tolerált hibákat az alábbi ábra szemlélteti (8. ábra).
- *Csomópont és fájlmeosztás többség* (Node and File Share Majority): hasonló, mint az előző, csak itt nem közös elérésű lemez a tanú, hanem egy külső fájlmeosztás. Földrajzilag elosztott fürtök, úgynevezett *geoclusterek* esetén szokás alkalmazni, amikor a csomópontok egymástól távol eső helyeken vannak elhelyezve, így védekezve például az egész szervert érintő hibák ellen (pl. tűzeset vagy természeti katasztrófa).
- *Csak lemez* (No Majority: Disk Only): csak a tanú lemeznek van szavazata. Nem javasolt már a használata, csak kompatibilitás miatt maradt benne, ez megegyezik a régi üzemmóddal.

A Technet kapcsolódó cikke [5] további tanácsokat ad több telephelyes fürtök quorum beállításához.

A Windows Server 2016-ban további finomításokat vezettek be a szavazatok kezelésére (pl. lehetséges dinamikusan változó szavazatokat beállítani), érdeklődők erről itt [11] találnak leírást.

Node and Disk Majority Quorum Configuration, Four Nodes (Plus Disk)



8. ábra Csomópont és lemez többség (Forrás: Technet). Az alsó ábra jobb oldalán a három gép között megszakadt a hálózati kapcsolat, ezért nincs quorum

3.3 Megosztott tárolók

Kritikus kérdés, hogy a közös adatokat milyen módon érik el az egyes fűrttagok. A követelmény a megosztott lemezzel szemben az, hogy (1) mindegyik csomópont elérhesse, (2) legyen lehetőség arra, hogy egyvalaki lefoglalja a lemezt (reserve), és (3) a többiek kikényszeríthessék, hogy megerősítse a foglalását a tulajdonos, és ha nem jelentkezik időben, akkor elvehessék tőle az eszközt.

A SCSI protokoll alkalmas a fenti feladatokra, a WSFC a következő SCSI protokollon alapuló csatolófelületeket támogatja.

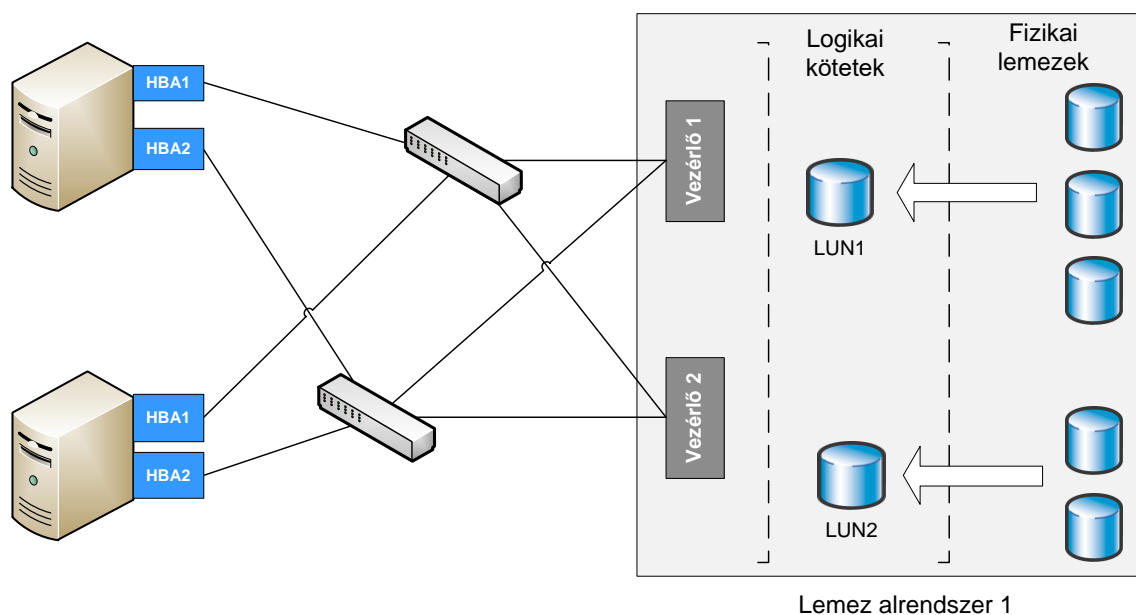
- *Serial Attached SCSI (SAS)* [2]: a párhuzamos SCSI utódja, az eddigi busz topológia helyett pont-pont kapcsolatot használ az eszközök és a vezérlő között⁵.
- *Fibre Channel (FC)* [3]: nagy adatátviteli sebességgel és alacsony késleltetéssel rendelkező protokoll, amit gyakran használnak tárolóeszközök elérésére.
- *iSCSI*: SCSI parancsokat TCP/IP felett átvivő protokoll. Előnye, hogy elvileg nem kellene speciális csatolók és kapcsolók a használatához, normál nagy sebességű Ethernet eszközök használhatóak. Hátránya, hogy a TCP protokollt alapvetően nem ilyen célra találták ki, így teljesítménye elmaradhat a kifejezetten tárolóeszközök számára kifejlesztett protokollokétól.

A lemezeket általában nem közvetlenül kötik rá a fűrt csomópontjaira, hisz így elég nehézkes lenne menedzselni és skálázni a rendszert, hanem lemez alrendszerben helyezik el őket, amelyhez valamilyen speciális *tárolóhálózaton* keresztül (SAN – Storage Area Network) csatlakoznak a csomópontok. A különálló lemez alrendszer előnye, hogy elfedi a kliensek elől a lemezek tényleges fizikai elrendezését, könnyebb menet közben hozzáadni vagy módosítani az elrendezésüket, valamint például a gépektől függetlenül lehet karbantartani, menteni a rajtuk tárolt adatokat. Ennek ellenére viszont a SAN-on keresztül csatlakoztatott lemezeket a számítógépek operációs rendszere és alkalmazásai úgy látják, mintha helyi lemezek lennének, így a megszokott módon dolgozhatnak velük. A SAN-ok először a Fibre Channel protokollt használták, mára már azonban rengeteg egyéb lehetőség is rendelkezésre áll. A Fibre Channel eszközök (a kliensek és a lemezek) gyűrű topológiába vagy kapcsolók (switch) segítségével csillag topológiába szervezhetőek, az iSCSI eszközök a hagyományos Ethernet kapcsolókat használják, így nehéz általánosan beszélni róluk, de a következő ábrán (9. ábra) lévő komponensek nagyrészt közösek.

A fontosabb kapcsolódó fogalmak:

⁵ A *Storage Spaces* funkció segítségével már lehet külső SAS-házakat is megosztott tárhelyként használni.

- *Host Bus Adapter (HBA)*: a kliensben lévő kártya, ami a szükséges csatolófelületet biztosítja.
- *Storage Area Network (SAN)*: olyan dedikált hálózat, amit csak a tárolórendszerekkel kapcsolatos forgalomra használnak. Nemcsak számítógépeket és lemezeket köt össze, hanem például szalagos egységek is szerepelhetnek benne. Mivel kritikus fontosságú, érdemes lehet a SAN csatlókat többszörözni, hogy egy számítógépről több, független úton is elérhető legyen egy lemez (ezt nevezik *multipathing*nek). Így nemcsak a megbízhatóságot, hanem akár teljesítménynövekedést is elérhetünk, ha az operációs rendszer támogatja azt (tehát felismeri, hogy a két külön HBA-n látott lemez az tulajdonképpen egy és ugyanaz).
- *Lemez alrendszer*: egy egyszerű külső lemezházról abban különbözik, hogy általában van benne külön redundáns vezérlő, amivel pl. RAID tömböket lehet konfigurálni, nagyméretű cache-t tartalmaz és távolról menedzselhető. Határ a csillagos ég, 1000-nél is több lemezt tartalmazó rendszereket is lehet már kapni. (Az egyes gyártók a disk array, storage array, storage subsystem kifejezéseket is használják megnevezésként.)



9. ábra Storage Area Network redundáns utakkal



10. ábra EMC Connectrix ED-48000B SAN director és IBM System Storage SAN32B-3 SAN switch



11. ábra IBM System Storage DS6000 és HP StorageWorks 4100 Enterprise Virtual Array

- *Logical Unit Number (LUN)*: eredetileg egy SCSI vezérlőn lévő konkrét lemez azonosítására szolgáló cím, általánosságban már inkább a lemeztömbökből képzett virtuális lemezekre használják SAN környezetben.
- *Zoning/masking*: mivel egy tárolórendszer általában több különböző gép használ, így fontos, hogy a lemezeket el lehessen különíteni, erre szolgál a zónázás. Ilyenkor minden gép csak azokat a lemezeket látja, amelyek az ő zónájában vannak.
- *Network Attached Storage (NAS)*: ez is egy hálózati tároló, azonban a SAN-tól abból különbözik, hogy ezt fájl szinten éri el a kliensek, és nem blokkos eszközként. Így nem is a SCSI protokollt használják az eléréséhez, hanem pl. NFS-t (Unix/Linux) vagy SMB-t (Windows).
- *Direct Attached Storage (DAS)*: a számítógépre közvetlenül rákötött lemez elegánsabb megfogalmazása.

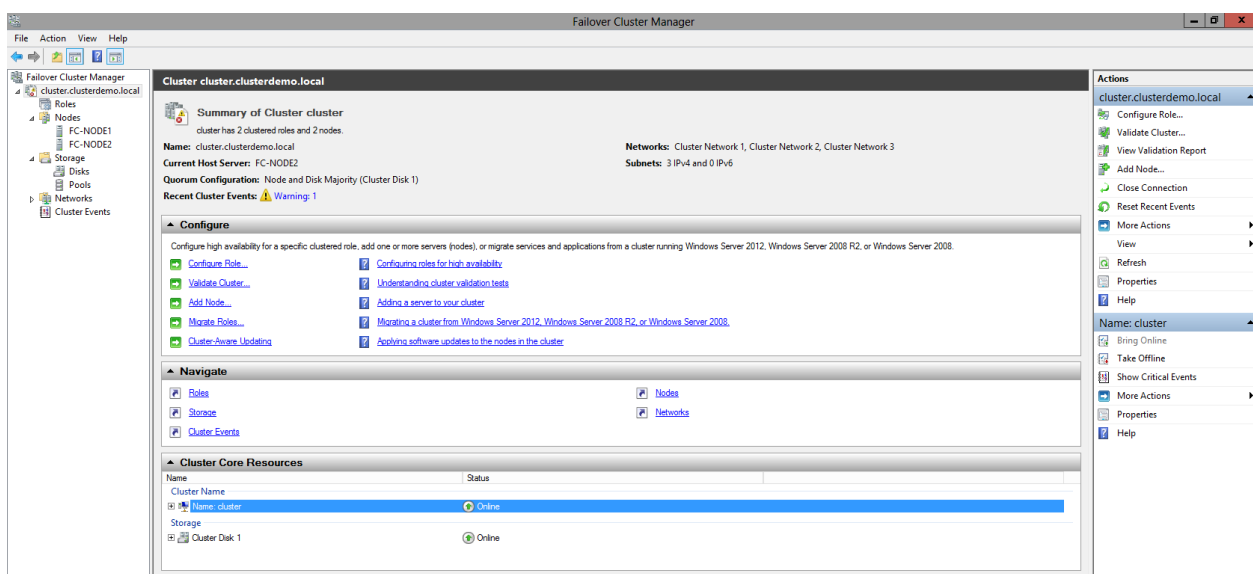
A mérés során iSCSI felületű tárolót szimulálunk szoftveresen, az iSCSI-ről további részletek az *Informatikai technológiák laboratórium 1.* [6] kapcsolódó mérési segédletében találhatóak.

3.4 Fürt létrehozása

Elég sok követelménynek kell megfelelnie a felhasznált hardver- és szoftverkörnyezetnek, hogy alkalmas legyen fürt létrehozására. A csomópontokban azonos (vagy legalábbis nagyon hasonló) hardvernek kell szerepelnie, azonosnak kell lennie a BIOS és firmware verzióknak, ugyanazoknak az operációs rendszer frissítéseknek kell fent lenniük. A gépeknek tartományi tagoknak kell lenniük, DNS-t kell használniuk a névfeloldáshoz. Ajánlott legalább két különböző hálózati interfész használata (ezekhez különböző alhálózatban lévő IP-címeket kell használni), ezekből a fürt szoftver egy *'Microsoft Failover Cluster Virtual Adapter'* nevű virtuális adaptert készít, amin keresztül a fürttagok transzparens módon tudnak kommunikálni az egyes hálózati kártyák meghibásodása esetén is. A követelményeket sokáig lehetne még folytatni, ezeket szerencsére a fürt telepítése előtt ellenőrzi a rendszer.

WSFC fürt létrehozásakor a következő lépéseket kell elvégeznünk.

- A Failover Clustering funkció telepítése a csomópontokra.
- Hálózati kártyák és közös tárolók csatlakoztatása és beállítása.
- Fürtellenőrzés (cluster validation) lefuttatása. Ez ellenőrzi a hardver és szoftver környezetet (megfelelő CPU architektúra, azonos frissítési szint az egyes csomópontokon, tartományi tagság, stb.), a hálózati eszközöket és beállításokat (pl. hálózati kártyák, IP-címek) és a tárolót (pl. támogatja-e a szükséges SCSI parancsokat, elég kicsi-e a késleltetése).
- Fürt létrehozása: csomópontok kiválasztása és a fürt virtuális címeinek megadása.
- Alkalmazás beállítása a fürtön. A kliens hozzáférési pontok és alkalmazás beállításainak megadása után a fürtrendszer a háttérben létrehozza a szükséges erőforrásokat és függőségeket.



12. ábra: Failover Cluster Management felület

A fenti ábrán látható egy példa a fürt menedzsment felületére.

3.5 Megjegyzések

Feladatátvételi fürtöket akkor érdemes alkalmazni, ha egyébként az infrastruktúra többi része (hardver alkatrészek, szerverszoba környezete, rendszergazdák képzettsége, informatikai házirendek) már jó minőségű, és azok javításával már nem nagyon lehetne tovább növelni a rendelkezésre állást, de szeretnénk azt mégis például 99%-ról 99,99%-ra javítani. Az ilyen magas igények miatt a Microsoft csak azokon a konfigurációkon támogatja a WSFC-et, amiket előzetesen bevizsgáltak és megkapta a „Certified for Windows Server 2016” logót. Ennek a részleteit a [4] oldalon találjuk.

A fürtbeállításokat a csomópontok helyileg a %SystemRoot%\Cluster\Clusdb fájlban tárolják (fürtadatbázis). Ez megtalálható a rendszerleíró-adatbázisban is a HKEY_LOCAL_MACHINE\Cluster kulcs alatt.

A fürt teljes kezelést meg lehet oldani PowerShell segítségével is, ehhez a FailoverCluster modul cmdletei segítenek.

A fürttel kapcsolatos részletes információkat az eseménynaplóban és annak trace log részében láthatjuk (tracert parancssori eszköz).

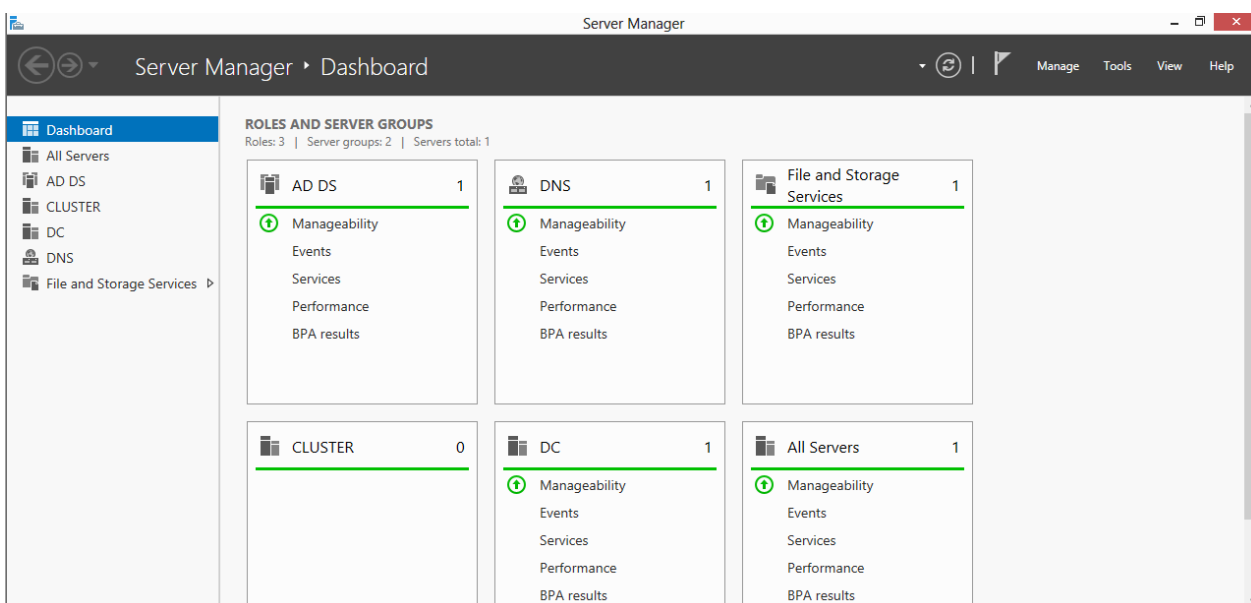
4 Függelék: A Windows Server 2016 menedzsment felülete

A mérés során Windows Server 2016-ot futtató virtuális gépeket fogunk használni, a függelék röviden összefoglalja az új felhasználói és menedzsment felület tudnivalóit.

A Windows Server 2016-t két változatban lehet telepíteni. *Server Core* opció esetén (ami most már az alapértelmezett beállítás) nem települnek a GUI eszközök nagy része, és a szerveret helyileg PowerShell segítségével lehet kezelni. *Server with GUI* opció esetén a korábbi teljes GUI felület települ, ilyenkor az áttevített *Server Manager* eszközt vagy a korábbi MMC alapú felületeket lehet használni.

A javasolt foratókönyv az, hogy Server Core típusú szervereket használjunk, és ezeket távolról egy GUI-s szerverről vagy Windows 8.1 munkaállomásról adminisztráljuk (Windows 8.1-re is elérhető az összes GUI menedzsment eszköz a *Remote Server Administration Tools* csomagban). Ehhez az új Server Manager vagy az új Powershell cmdletek már WinRM (WS-Management) alapú távoli elérést használnak, ami most már a legtöbb esetben alapértelmezetten engedélyezve van.

Server Manager: A Server Manager új felületét úgy alakították ki, hogy azzal könnyű legyen akár nagyszámú távoli kiszolgálót adminisztrálni (13. ábra).



13. ábra: Az új Server Manager felülete

A szerverekből csoportokat (server group) lehet létrehozni. Az egyes szerverek hozzáadása után a Server Manager felderíti, hogy azokon milyen szerepek vannak telepítve, és mindegyikhez megjeleníti a megfelelő külön felületet.

A korábbi MMC eszközöket a Tools menüpontból vagy egy adott szerver jobb gombos menüjéből lehet elérni. Bár egyes funkciók kezelése már átkerült az új Server Manager felületére, pl. kötetek kezelése, sok funkciót még a régi eszközökben lehet elérni, pl. a feladatátvételi fürtök kezelése.

PowerShell: A legtöbb beállítást viszont most már PowerShellen keresztül lehet elérni. A hálózati címek beállításától kezdve a fürtök létrehozásáig van most már szinte mindenhez PowerShell cmdlet. A Technet oldalon elérhető a részletes referencia az összes cmdletről [7].

A mérés során esetleg szükséges legfontosabb cmdletek a következők:

- `Get-Help <cmdlet> -Full`: megjeleníti adott cmdlet leírását példákkal együtt.
- `Get-Command -Module <module>`: listázza egy adott modulhoz tartozó összes cmdletet.
- `Stop-Computer, Restart-Computer`: leállítja és újraindítja a számítógépet.
- `Get-NetAdapter`: megjeleníti a számítógép hálózati csatlakozóinak adatait.
- `Get-NetIPConfiguration`: megjeleníti az IP-beállításokat (IP-címek, DNS kiszolgálók, átjáró...)
- `Enter-PSSession`: belép egy távoli gépre, és azon hajthatunk végre parancsokat interaktívan.
- `Get-WindowsFeature, Install-WindowsFeature`: listázza a telepített szerepeket vagy képességeket.

Ezen kívül szükség lesz majd az iSCSI kliens és kiszolgáló kezelésére szolgáló cmdletekre.

5 További információ, hivatkozások

- [1] Microsoft. „Failover Clustering Overview”, URL: <http://technet.microsoft.com/en-us/library/hh831579.aspx>
- [2] Wikipedia. „Serial Attached SCSI”, URL: http://en.wikipedia.org/wiki/Serial_Attached_SCSI
- [3] Wikipedia. „Fibre Channel”, URL: http://en.wikipedia.org/wiki/Fibre_channel
- [4] Microsoft. „Windows Server Catalog of Tested Products”, URL: <http://www.windowsservercatalog.com>
- [5] Microsoft. „Configure and Manage the Quorum in a Windows Server 2012 Failover Cluster”, URL: <http://technet.microsoft.com/en-us/library/jj612870.aspx>

- [6] Tóth Dániel. „Háttértár rendszerek”, mérési segédlet, Informatikai technológiák laboratórium 1., BME MIT, URL: <http://www.inf.mit.bme.hu/edu/courses/itlab1>
- [7] Microsoft. „Windows PowerShell Support for Windows Server 2012”, URL: <http://technet.microsoft.com/en-us/library/hh801904.aspx>
- [8] Jane Yan. „Introduction of iSCSI Target in Windows Server 2012”, The Storage Team Blog, URL: <http://blogs.technet.com/b/filecab/archive/2012/05/21/introduction-of-iscsi-target-in-windows-server-2012.aspx>
- [9] Elden Christensen. „Developing highly-available, scale-out applications for Windows Server 2012”, URL: <http://channel9.msdn.com/Events/Build/2012/3-051>
- [10] Elden Christensen. „Cluster Shared Volume (CSV) Inside Out”, Clustering and High-Availability blog, 2 Dec 2013, URL: <http://blogs.msdn.com/b/clustering/archive/2013/12/02/10473247.aspx>
- [11] Microsoft. „What's New in Failover Clustering in Windows Server”, URL: <http://technet.microsoft.com/en-us/library/dn265972.aspx>

Információk a korábbi verziókban lévő fürtözési megoldásokról

Ezek egy részében már nem aktuálisak az információk a terminológia és architektúra változás miatt.

- [12] W. Vogels *et al.* „The Design and Architecture of the Microsoft Cluster Service”, in Proc. of FTCS'98, IEEE, 1998. URL: <http://dl.acm.org/citation.cfm?id=796898>
- [13] Server Clusters: Architecture Overview, URL: <http://download.microsoft.com/download/0/a/4/0a4db63c-0488-46e3-8add-28a3c0648855/ServerClustersArchitecture.doc> (jó áttekintő és részletesebb architektúra bemutatás)
- [14] Microsoft Server Clusters Technical Reference (Windows Server 2003), URL: [http://technet.microsoft.com/en-us/library/cc759014\(W.S.10\).aspx](http://technet.microsoft.com/en-us/library/cc759014(W.S.10).aspx)
- [15] Microsoft. „An update is available that adds a file share witness feature and a configurable cluster heartbeats feature to Windows Server 2003 Service Pack 1-based server clusters”, KB 921181, URL: <http://support.microsoft.com/kb/921181> (witness files share és a heartbeat mechanizmus leírása)