

TDK -dolgozat

Magda Tamás, Herczog Attila

Pécsi Tudományegyetem
Pollack Mihály Műszaki és Informatikai Kar
Rendszer és Szoftvertechnológiai Tanszék

Magas rendelkezésre állású rendszer megvalósítása

Szerzők:

Magda Tamás, 4. évfolyam

Herczog Attila, 4. évfolyam

Témavezető:

Dr. Várady Géza, PhD
Megbízott tanszékvezető

BEVEZETÉS	4
1 RENDELKEZÉSRE ÁLLÁS.....	5
2 HIBATÚRÓ SZERVEREK	7
3 CLUSTER MEGOLDÁS	8
4 CLUSTER FOGALMAK	11
5 CLUSTER MEGOLDÁSOK A PIACON	12
6 KÉT GÉPBŐL ÁLLÓ CLUSTER MEGVALÓSÍTÁSA.....	14
7 CLUSTER TELEPÍTŐ SZOFTVER BEMUTATÁSA.....	19
8 IRODALOMJEGYZÉK.....	23

Bevezetés

Napjainkban szinte már mindenhol megtalálható a számítógépekkel végzett adatfeldolgozás, kiszolgálás, folyamatirányítás. Ezen rendszerek nagy mennyiségű adatot dolgoznak fel, rengeteg szolgáltatást biztosítanak, például online ügyintézés, online vásárlás, jegyrendelés, foglalás. Belátható, hogy ezen rendszerekben egy rövid kiesés is hatalmas károkat tud okozni az üzemeltetői számára.

Az internet elterjedésével ma már egyre inkább a kisebb cégeknek is szükségük van nagy rendelkezésre állású szolgáltatásokat nyújtó rendszerekre. A cége üzletmenetét biztosító kiszolgálók kiesése esetén a dolgozók nem tudják ellátni a feladataikat, nem tudják kiszálgálni a vásárlókat, mind addig, amíg a hiba elhárítása meg nem történik.

Egy a megbízhatóságot növelő rendszer megvalósításához több lehetőség áll rendelkezésre, kezdve a drága fizetős megoldásoktól a szabad szoftverekkel és egyszerű PC-vel megvalósítható megoldásig. Ez utóbbinak a hátránya az, hogy nekünk kell több szoftvert telepíteni és konfigurálni, amely egy igen összetett több napos munka még profi szakemberek számára is .

Ezen dolgozatunknak a témája egy szabad szoftveres megoldás bemutatása, elkészítése és ezen rendszer telepítési folyamatának egyszerűsítésére irányuló automatizált megoldás megvalósítása

1 Rendelkezésre állás

1.1 Fogalma

Rendelkezésre áll egy alkalmazás vagy erőforrás, amikor a működésének képes eleget tenni, képes feladatokat fogadni és azokat ellátni. Értékét százalékban szokás megadni egy évre vetítve, azaz az évnek hány százalékában kell biztosítani az adott szolgáltatást. Ezen százalékokat szerződésekbe szokták foglalni, mint minimum elvárt rendelkezésre állási értékek. Szerverek esetén ez az idő, amikor képesek kiszolgálni a klienseket.

1. táblázat Rendelkezésre állás százalékosan

Rendelkezésre állás %-ban	Maximális üzem kiesés egy évben
90%	876 óra (36.5 nap)
99%	87 óra (3.6 nap)
99.9%	8.8 óra
99.99%	53 perc
99.999%	5 perc
99.9999%	32 másodperc

1.2 Rendelkezésre állást csökkentő tényezők

Számos tényező befolyásolhatja egy rendszer rendelkezésre állását. Ezek két fő csoportra bonthatóak: tervezett illetve nem tervezett leállások.

Tervezett leállások lehetnek

- Szoftverfrissítés
- Operációs rendszer frissítés
- Hardver karbantartás, bővítés
- Hálózat karbantartás, bővítés

Nem tervezett leállások lehetnek

- Áramszünet
- Szoftver hiba
- Hardver hiba
- Vis Major esetek (tűz, árvíz földrengés)

1.3 Rendelkezésre állás növelésére szolgáló megoldások

A rendelkezésre állás növelése a legtöbb esetben redundancia alkalmazásával történhet, amely segítségével megszüntetésre kerülnek a kritikus hibapontok (angolul: Single Point Of Failure).

Áramszünet kivédésére többfajta megoldás alkalmazható attól függően, hogy mekkora időtartamú áramkimaradást kell áthidalni. Néhány 10 perces áramkiesés ellen szünetmentes tápegységekkel (UPS) lehet védekezni. Ha az adott rendszernek hosszabb villamos energia kiesés esetén is működni kell, ekkor második védelmi vonalként alkalmazhatók dízelgenerátorok. Így az üzemanyag függvényébe, akár több napos, hetes áramszünet sem okoz üzemkiesést.

Az áram szünet mellett a második leggyakoribb hibalehetőség a szerverek valamely hardverének meghibásodása. Ez ellen több szinten is lehet védekezni. Leggyakrabban a merevlemez egységek szoktak meghibásodni, ezért ezekből érdemes többet elhelyezni egy szerverbe, majd ezekre az úgy nevezett RAID technológiát alkalmazni.

További lehetőség a rendelkezésre állás növelése érdekében, hogy a szerverbe több tápegységet tesznek, ezeket esetlegesen két egymástól független áramkörbe kötik be, így az egyik áramkörön bekövetkező áramkimaradás nem kényszeríti leállásra az adott szervert.

Elérhető olyan megoldás is, hogy a szerverben összes hardverből kettő vagy több példány kerül beépítésre. Az ilyen megoldást megvalósító szervereket, hibatűrő szervernek nevezik.

További rendelkezésre állás növelő megoldás, hogy nem a szerveren belül alkalmaznak hardver redundanciát, hanem a teljes szerverből helyeznek el többet, majd ezeket speciális megoldásokkal összekötik, így kívülről (felhasználó szemszögéből) egy szervernek látszanak. Ezen megoldás a dolgozatunk fő témája.

2 Hibatűrő szerverek

2.1 Fogalma, tulajdonságai:

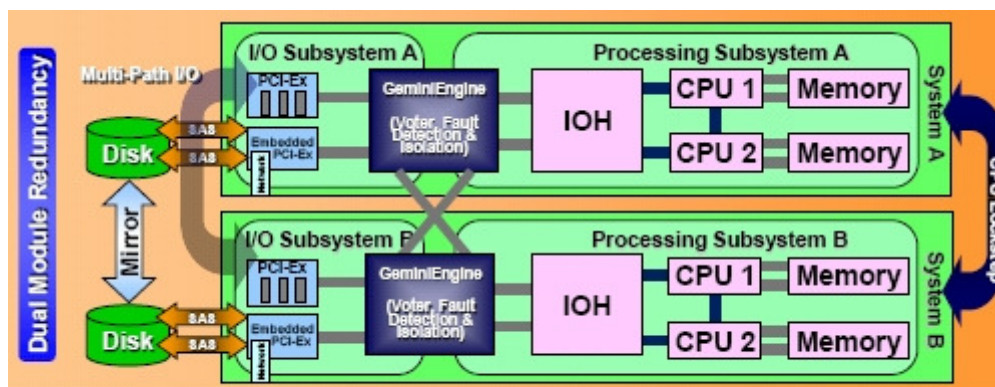
Teljes hardver redundancia. Lényegében két egybe épített szerver (1. ábra), minden hardverből több példány kerül beépítésre. Melyek üzem közben kölcsönösen ellenőrzik egymás működését (2. ábra Redundáns gép felépítése). Ezen funkció teszi lehetővé, hogy az egyik hardver meghibásodása esetén a másik hardver példány leállás nélkül átveszi a hibás összetevő feladatát. Ez a megoldás a szerver árában is jelentősen megjelenik.

Előnye, hogy a rajta futó szoftvereknek nem kell tudni a hardver redundanciáról. A szoftverek módosítás nélkül futathatóak ilyen rendszereken, ebből következően az adminisztrációs költsége sem lesz jelentősen magasabb mint egy hagyományos szervernek..

Hátránya, a magas ára, egy ilyen szerver több milliós tétel, melyet egy kis illetve közepes vállalkozás nem tud kigazdálkodni, viszont ebben a szektorban is egyre inkább szükség van a magas rendelkezésre állású rendszerrel támogatott infrastruktúrára.



1. ábra Redundáns szerver



Lockstep protection of entire CPU, Memory & IOH Intel Subsystems.

2. ábra Redundáns gép felépítése

3 Cluster megoldás

3.1 Cluster fogalma:

Számítógépek összekapcsolása olyan módon, hogy azok egyetlen összefüggő rendszert alkossanak, ezáltal együttesen végzik a megkapott feladatok megoldását.

A Cluster kialakításának célja, hogy több kisebb kapacitású részegységből egyetlen nagyobb kapacitású rendszert hozzanak létre, amely az egyes részegységek meghibásodása ellenére is képes marad szolgáltatásai biztosítására.

3.2 Cluster alkalmazása

Cluster megoldást alkalmazva kedvezőbb áron kiépíthető egy magas rendelkezésre állást biztosító rendszer. Ebben az esetben nem szükséges beruházni egy több milliós hibátűrő kiszolgálóba, hanem több olcsóbb nem redundáns szerver megvásárlásával és speciális szoftverekkel történő konfigurálása után a hibátűrő szervereket megközelítő rendelkezésre állást lehet elérni a rendszer segítségével.

3.3 Előnyök, hátrányok:

Előnye a hibátűrő szerverekkel szemben, a már említett költség megtakarítás a beszerzéskor. További jelentős előnye, hogy itt fizikailag több példány van a szerverből, így ezek elhelyezhetők földrajzilag eltérő helyen is, és a megfelelő összeköttetés biztosítása után ezek képesek egy rendszerként működni.

Hátránya, hogy sokkal több rendszer adminisztrációt igényel, jelentősen összetettebb a telepítése, és esetenként a rendszeren futó szoftvereket is módosítani kell a megfelelő működés érdekében.

Ahol rendkívül magas rendelkezésre állásra van szükség, ott érdemes a cluster technológiát ötvözni a hibátűrő szerverekkel.

3.4 Típusai:

Nagy számítási teljesítményű cluster (HPC):

Sok gép, főként párhuzamos matematika számítások végrehajtására

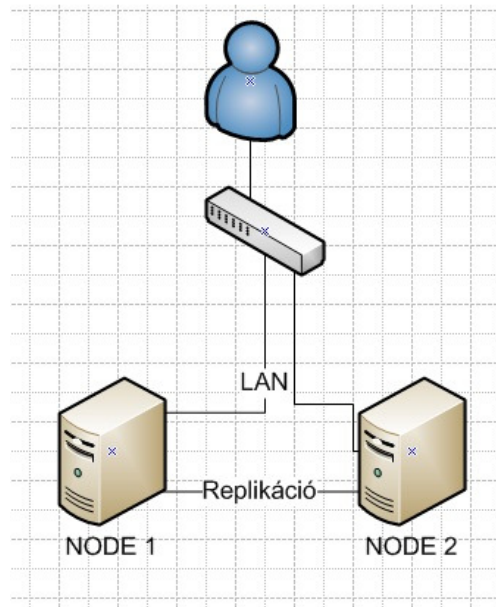
Magas rendelkezésre állású cluster (HA):

Szolgáltatások maximum nagyon kis kieséssel történő üzemeltetésére

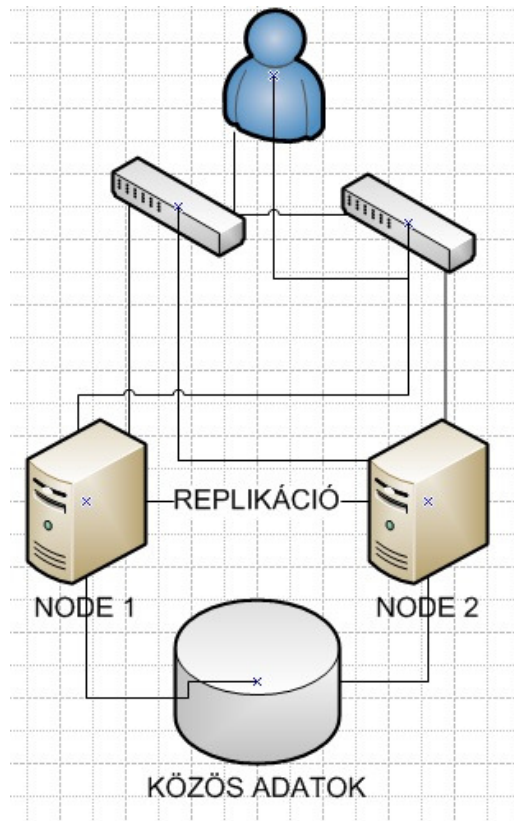
Terhelés elosztó rendszerek (LB)

Több kiszolgáló között, adott algoritmus alapján történő terhelés elosztás. Segít a HA rendszereknek, a kettőt kombinálva hozható létre igazán teherbíró és redundáns rendszer.

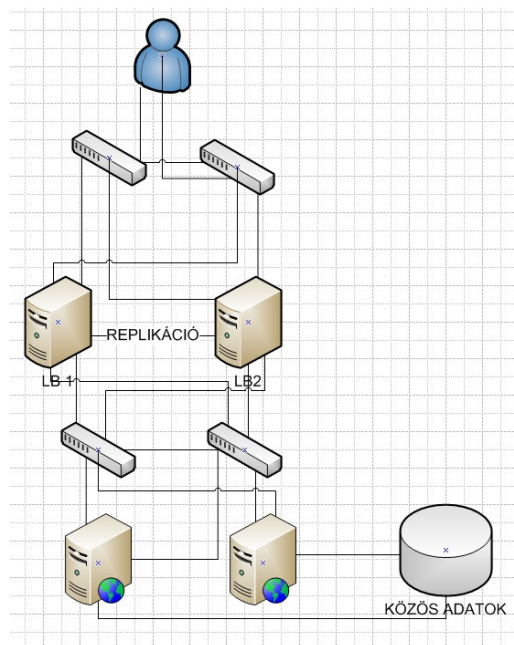
3.1 Cluster topológiák



3. ábra Egyszerű, két gépből álló Cluster



4. ábra Két gépből álló Cluster, közös adat tárolóval



5. ábra Magas rendelkezésre állású rendszer terhelésselosztással

4 Cluster fogalmak

A következőkben a cluster témakör legfontosabb fogalmai kerülnek ismertetésre.

CRM (Cluster Resource Management):

Központilag irányítja a cluster működését, és minden tagjával folyamatosan kommunikál. Dönt, hogy mikor mit kell tenni a folyamatos kiszolgálás fenntartása érdekében.

Split-brain szituáció:

Ha aktív-passzív konfigurációban a cluster tagok közös erőforrást birtokolnak (pl. nem osztott fájlrendszert) és egy hiba bekövetkeztekor a tagok között megszakad a hálózati kapcsolat, akkor a passzív gép úgy dönt, hogy átveszi az aktív szerepét, például felcsatolja a fájlrendszert. Majd egyszer csak visszatér a régi aktív gép. Ekkor két aktív gép lesz, közösen próbálják meg használni a nem osztott erőforrást, ezt nevezzük split-brain szituációnak. Ennek a problémának a megoldását mindenképp meg kell oldani egy cluster megvalósítása esetén, mert komoly problémákat, adatvesztést okozhat.

Quorum:

Határozatképesség jelölése, a minimum követelmény, hogy érdemi döntést lehessen hozni a cluster működésével kapcsolatban. Például: hány cluster tagnak kell elérhetőnek lennie, hogy a cluster működőképes legyen, vagy ki dönt a szolgáltatások „sorsáról”.

Ajánlott fajták:

Simple majority:

Sima többség”, a cluster működéséhez 50% +1 cluster tag szükséges.

Weighted majority:

Súlyozott szavazás. A cluster tagoknak különböző súlya lehet a szavazásban. A működéshez szintén 50% feletti szavazati arány szükséges.

Fencing:

Split-brain szituáció elkerülésére szolgáló megoldás.

Típusai:

Power fencing:

A hibásan működő cluster tagot lekapcsoljuk, és nem engedjük elindulni.
Például egy szünetmentes elveszi a feszültséget az adott gépről.

Fabric fencing:

Eszköz leválasztás. Leválasztás után a gépet nem engedjük ismét hozzáférni az erőforráshoz. Például leállt gép újra indítása után nem engedjük ismét hozzáférni egy nem osztott fájlrendszerhez.

Stonith:

Shot Other Node In The Head, a fencing fizikai megvalósításának módja .

A meghibásodott cluster tag „lelövése”, például szünetmentes vagy távmenedzsment eszköz segítségével.

5 Cluster megoldások a piacon

A kereskedelmi forgalomban jelenleg több nagy gyártó megoldása is megtalálható. Azonban ezek jelentős összegekbe kerülnek, illetve kööttségekkel járnak. Sőt a telepítésük ezeknek sem egyszerű feladat, mondhatni, hogy „pilóta vizsga” szükséges hozzá.

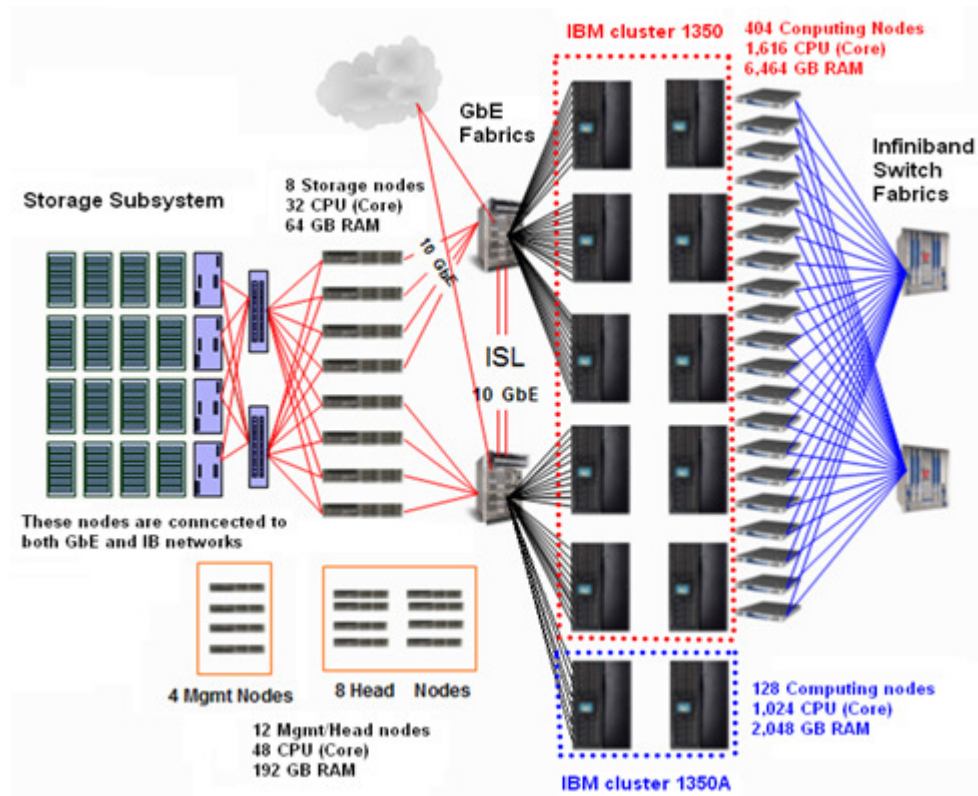
Néhány kereskedelmi termék:

Novell Cluster:

- Az ára 12 000 dollártól kezdődik.
- Novell Enterprise operációs rendszer szükséges hozzá
- Hivatalos weboldal:
<http://www.novell.com/products/openenterpriseserver/ncs.html>

IBM HA Cluster:

- IBM operációs rendszer és szerverek megvásárlása szükséges
- Nehéz telepítés, a telepítési leírás 746 oldal!
(<http://www.redbooks.ibm.com/redbooks/pdfs/sg246769.pdf>)
- Összetett hálózati topológia szükséges



6. ábra IBM Cluster topológia

6 Két gépből álló Cluster megvalósítása

A bemutatott kereskedelmi megoldások alapján, látható, hogy nagy szükség lenne egy költségkímélő, egyszerűen telepíthető megoldásra.

A továbbiakban az általunk elkészített cluster tulajdonságai kerülnek ismertetésre, amely weboldalak kiszolgálására képes.

A megvalósításhoz 2 db szerver számítógépre van szükség, melyeknek 2 db jó minőségű hálózati kártyával kell rendelkezniük. Illetve erősen ajánlott egy professzionális hardveres RAID kártya elhelyezése mind a két gépben, ezzel is növelve az adatbiztonságot illetve a háttértár teljesítményét.

Továbbá szükség van szoftveres összetevőkre is. A cluster tagokon Ubuntu 10.04 LTS szerver operációs rendszer fut. A webes tartalmak kiszolgálást az Apache2 webservert szoftver végzi. Adatbázis szerverként a MySQL-re esett a választás, mert szintén az egyik legelterjedtebb ingyenes megoldás, illetve tud aktív-aktív működést is. A levelezést pedig a postfix nevű smtp szerver biztosítja.

A szerver szoftvereken kívül szükség van még a cluster specifikus programokra is. Egy clusternek sarkalatos pontja az adatreplikáció, erre a célra a DRBD nevű szoftvert használtuk.

Főbb tulajdonságai:

- Blokk szintű replikációt tud, azaz fájlrendszer független.
- Támogatja mind az aktív-passzív mind az aktív-aktív megoldást
- Közel 100%-os replikációt biztosít (TCP szintű Raid-1)

Szintén fontos komponens a már említett CRM, mely egy központi szoftver, amely az egész cluster működését felügyeli és irányítja. Erre megfelelő szoftver a Pacemaker.

6.1 Egy kész rendszer esetén felmerülő problémák

Split-brain szituáció:

A megvalósított kétgépes clusterben előfordulhat a split-brain szituáció, ezt orvosolná a már említett power fencing, viszont egy ilyen eszköz beszerzése költséges. A probléma költség hatékony megoldására megfelelő egy olyan script is, mely „folyamatosan” figyeli a hálózatot és amennyiben szakadást érzékel, azonnal leállítja a kérdéses gépet.

Két script is készült, az egyik csak a fizikai kapcsolatot ellenőrzi a gépek között, az ip alapú kommunikációt nem. Ennek előnye, hogy nem terheli a hálózatot. A másik script ping alapon működik, így az már a kommunikációt is ellenőrzi.

Postfix mailqueue:

Nem helyezhető át közös tárolóra, mert a syslog folyamat is használja, így az aktív-passzív állapot váltáskor sikertelen a meghajtó lecsatolása. Jelenlegi megoldás egy script, mely folyamatosan szinkronizálja a mailqueue tartalmát a két gép között.

6.2 A megvalósított rendszer tulajdonságai:

A cluster két gépből áll, egy aktív tagból, amely végzi a kiszolgálást, és egy passzív tagból, amely az aktív gép tükör szervere. A két node között folyamatos az adatszinkronizáció a DRBD szoftver segítségével. Az aktív gép leállása esetén a passzív gép 30 másodpercen belül átveszi a szolgáltatásokat. Ez a jelenleg megvalósított web kiszolgálást végző cluster esetén azt jelenti, hogy automatikus felcsatolja a weboldalt és az adatbázist tartalmazó partíciókat, majd beállítja az adott ip címet, melyen elérhetőek a weboldalak és az adatbázis. Ezek után elindítja az apache2 webszervert és a mySQL adatbázis szervert, illetve a levelező szervert. Ezt mind automatikusan teszi emberi beavatkozás nélkül, amint észleli, hogy az eddig aktív gép már nem elérhető. Így garantálva a lehető legkisebb üzemkiesést hiba fellépése esetén.

A megvalósított rendszer adott esetben képes egy részleges terheléselosztásra is. Adott beállítás elvégzése után az addig passzív gép is használható kiszolgálásra, mint adatbázis szerver. A webszerver az egyik gépen futhat, a másikon pedig az adatbázis szerver, megosztva a terhelést a két tag között.

3.2 A telepített rendszer megvalósult migrációja

A cluster építéskor első lépésként meg kell tervezni, hogy milyen szerver szoftverek fognak futni a gépen, milyen elosztással, melyik milyen prioritást kapjon. Milyen szabályok szerint történhet az átköltözés (pl. két szolgáltatás csak együtt futhat).

Ezek után érdemes megtervezni a hálózati topológiát, milyen redundáns útvonalak lesznek, melyik milyen sebességű. A pacemaker szoftver nem követeli meg a redundáns útvonalat, viszont erősen javasolt. A DRBD-nek javasolt 1 Gbit/sec (esetleg 10Gbit/sec) kapcsolatot biztosítani.

Ha ezek megtervezésre kerültek, szintén meg kell tervezni, hogy milyen osztott partíciókat fogunk alkalmazni (pl. az adatbázisok és a weboldalak külön partíción lesznek-e?)

Majd feltelepítésre kerülhet a DRBD.

Érdemes az alábbi felsorolt konfigurációs lehetőségek bekapcsolását megfontolni:

- Fencing eseménykezelő
- Adatintegritás ellenőrzés
- Hálózati adat átvitel ellenőrzés
- További ellenőrző összegek számolásának bekapcsolása

Érdemes még a szinkronizációs sebességet is a hardverekhez igazítani.

```

global {
    usage-count no;
}
common {
    protocol C;
    disk {
        fencing resource-only;
        on-io-error detach;
    }
    handlers {
        fence-peer "/usr/lib/drbd/crm-fence-peer.sh";
        after-resync-target "/usr/lib/drbd/crm-unfence-peer.sh";
        split-brain "/usr/lib/drbd/notify-split-brain.sh root";
        out-of-sync "/usr/lib/drbd/notify-out-of-sync.sh root";
    }
    net {
        data-integrity-alg md5;
        cram-hmac-alg md5;
        shared-secret "corosync_secret";
        after-sb-0pri disconnect;
        after-sb-1pri disconnect;
        after-sb-2pri disconnect;
        rr-conflict disconnect;
    }
    syncer {
        rate 45M;
        verify-alg md5;
        csums-alg md5;
    }
}

```

7. ábra példa egy DRBD főkonfigurációra

A 7. ábrán láthatjuk a DRBD fő konfigurációs fájlt, ahol beállítható az említett adatintegritást védő lehetőségek, illetve az adat replikáció sebességére vonatkozó funkciók. A funkciókról további információ található a <http://www.drbd.org/users-guide-emb/s-configure-resource.html> weboldalon.

Következő lépésként a Pacemaker-t kell telepíteni, ez egyszerűen csomagból megtehető. A következő lépés ennek a beállítása. Az egyik gépen generálni kell egy auth kulcsot, ezzel azonosítják magukat a cluster tagok, hogy ők jogosultak az adott clusterbe belépni. Ezt a kulcsot minden tagra kézzel át kell másolni. Majd a corosync konfigurációs fájlban meg kell adni a már megtervezett redundáns útvonalakat. Ez után a CRM-et kell a saját cli-je segítségével konfigurálni, amit egy xml fájlba fog menteni.

Ezek után a szerver szoftvereket kell telepíteni. Ez itt még a hagyományos egy gépes felállásnál megszokott telepítés.

Ha a telepítés rendben ment, ez után jön az egyik nehéz rész, az adott szerver szoftverek fájljait át kell helyezni a megfelelő drbd partícióra. Erre nem igazán létezik jó megoldást, ezt rengeteg teszteléssel lehet ki kikísérletezni, hogy mit érdemes áthelyezni, mit nem, illetve mi az, amit egyelőre nem is lehet (pl. postfix mailqueue).

Ha ezt a lépést sikerül átlépni utána be kell állítani a pacemakert. Ez sem könnyebb feladat, mint az előző. Itt meg kell adni a szolgáltatásokra vonatkozó szabályokat. Milyen szolgáltatások lesznek, azok milyen sorrendben indulhatnak el, melyik függ melyiktől, hogyan lehet őket áthelyezni, milyen időközönként ellenőrizze és még jó pár lehetőség.

```
<primitive class="ocf" id="Mysql" provider="heartbeat" type="mysql">
  <instance_attributes id="Mysql-instance_attributes">
    <nvpair id="Mysql-instance_attributes-binary" name="binary" value="/usr/sbin/mysqld"/>
    <nvpair id="Mysql-instance_attributes-config" name="config" value="/etc/mysql/my.cnf"/>
    <nvpair id="Mysql-instance_attributes-datadir" name="datadir" value="/var/lib/mysql"/>
    <nvpair id="Mysql-instance_attributes-socket" name="socket" value="/var/run/mysqld/mysqld.sock"/>
    <nvpair id="Mysql-instance_attributes-pid" name="pid" value="/var/run/mysqld/mysqld.pid"/>
  </instance_attributes>
  <operations>
    <op id="Mysql-monitor-60s" interval="60s" name="monitor"/>
  </operations>
  <meta_attributes id="Mysql-meta_attributes">
    <nvpair id="Mysql-meta_attributes-target-role" name="target-role" value="Started"/>
    <nvpair id="Mysql-meta_attributes-migration-threshold" name="migration-threshold" value="3"/>
  </meta_attributes>
</primitive>
```

8. ábra Részlet a CRM konfigurációból, Mysql server beállításai

Ez után lehet csinálni egy átállási tesztet, az egyik gépet leállítjuk, majd a másikon elindul-e minden szolgáltatás.

Majd meg kell oldani még néhány problémát, pl a split-brain szituáció kizárását, vagy pl. az időzített feladatok futását, esetlegesen az autentikációs adatok szinkronba tartását. A split-brain szituáció elkerülése megoldható a fent említett módon, script segítségével, illetve a postfix probléma is scriptel orvosolható.

```

Online: [ node2 node1 ]

Master/Slave Set: DrbdMasterWeb
  Masters: [ node1 ]
  Slaves: [ node2 ]
Master/Slave Set: DrbdMasterMysql
  Masters: [ node1 ]
  Slaves: [ node2 ]
Master/Slave Set: DrbdMasterMail
  Masters: [ node1 ]
  Slaves: [ node2 ]
Resource Group: MysqlCluster
  MysqlFS      (ocf::heartbeat:Filesystem):    Started node1
  MysqlIP      (ocf::heartbeat:IPaddr2):        Started node1
  Mysql        (ocf::heartbeat:mysql):          Started node1
Resource Group: WebCluster
  WebFS        (ocf::heartbeat:Filesystem):    Started node1
  WebIP        (ocf::heartbeat:IPaddr2):        Started node1
  Apache2      (ocf::heartbeat:apache):         Started node1
  MailFS       (ocf::heartbeat:Filesystem):    Started node1
  Postfix      (ocf::heartbeat:postfix):        Started node1
  Courier-auth  (lsb:courier-authdaemon):        Started node1
  Courier-imap  (lsb:courier-imap):              Started node1
  Courier-pop   (lsb:courier-pop):               Started node1
  Proftpd      (lsb:proftpd):                   Started node1
Clone Set: clone-ping
  Started: [ node1 node2 ]

```

9. ábra Sikereses telepítés után a működő rendszer a CRM-ből nézve

6.3 További fejlesztések

A jelenlegi rendszer 0-ról való telepítése néhány napot vesz igénybe (ez természetesen már 2-3 cluster beállítási tapasztalat után értendő), illetve utána is még szükséges pár hét tesztelés, hogy minek a konfigurációja maradt még el. A projekt célja ennek az időnek a rövidítése, egy egyszerűbb telepítési mechanizmus létrehozása.

Irodalomkutatásunk közben hasonló, szabad szoftvereken alapuló kész, „dobozos” megoldással nem találkoztunk, ezért meggyőződésünk, hogy a munkánk hiánypótló.

7 Cluster telepítő szoftver bemutatása

7.1 Program tulajdonságai

A programot Java nyelven készítettük, hogy platform független legyen. Grafikus felülettel rendelkezik, hogy a windowos rendszerekhez szokott szakembereknek is a megszokott grafikus felülettel találkozzanak, és ne keljen nekik a linuxos shell használatát mélyebb szinten megtanulniuk. A program SSH és SCP protokollok segítségével csatlakozik a szerverekhez és adja ki a megfelelő parancsokat illetve másolja fel a működéshez szükséges konfigurációs állományokat.

A fejlesztéshez felhasználtunk két API-t, az SSH kapcsolat létrehozásához a ganymed-SSH 2 –t, az SCP műveletekhez pedig a zehon api-t.

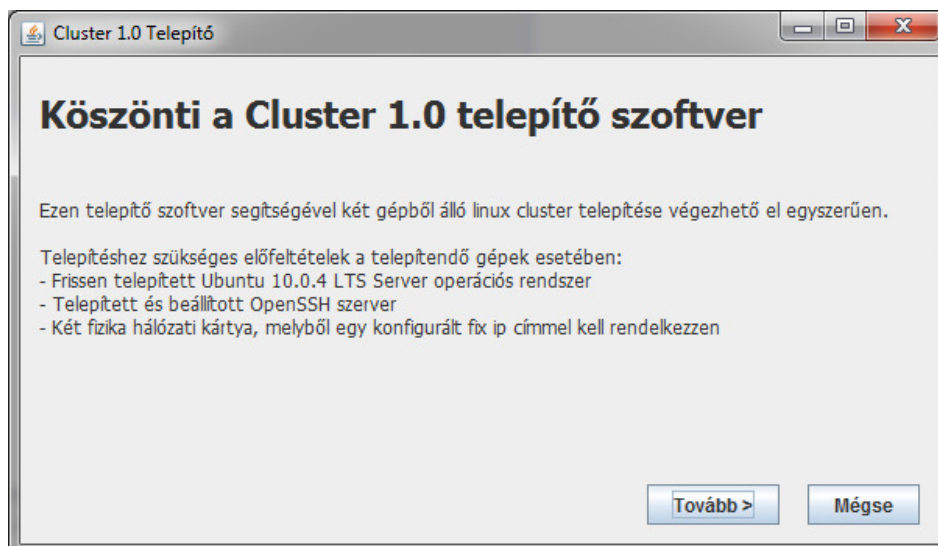
A szoftver 4 ip cím megadása után, automatikusan feltelepít minden szükséges csomagot, elvégzi a beállításokat, illetve a konfigurációs állományok létrehozását.

7.2 Telepítéshez szükséges előfeltételek

A clusterben résztvevő gépeknek a következő kritériumokat teljesíteniük kell, hogy a szoftver problémamentesen el tudja végezni a telepítést.

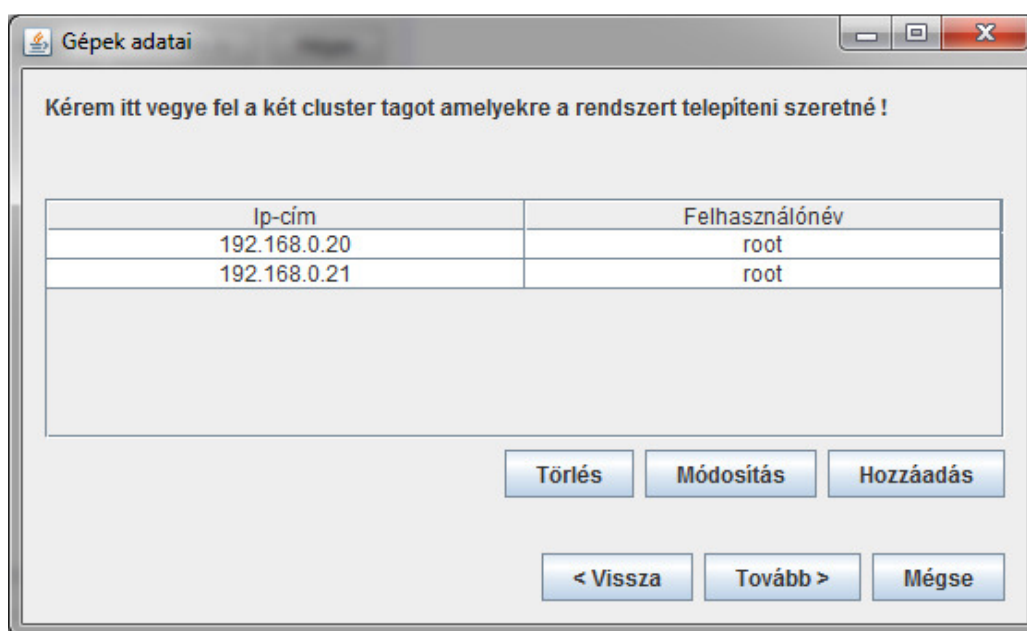
- A telepítendő szerverek Ubuntu 10.04 LTS Server operációs rendszerrel kell rendelkezzenek.
- Telepítve kell rajtuk lennie az OpenSSH szervernek.
- Két fizikai hálózati kártyával kell rendelkezniük, amelyből az egyiknek már rendelkeznie kell fix ip címmel.

Az utibbi két feltétel az Ubuntu megfelelő telepítése esetén könnyen teljesíthető.



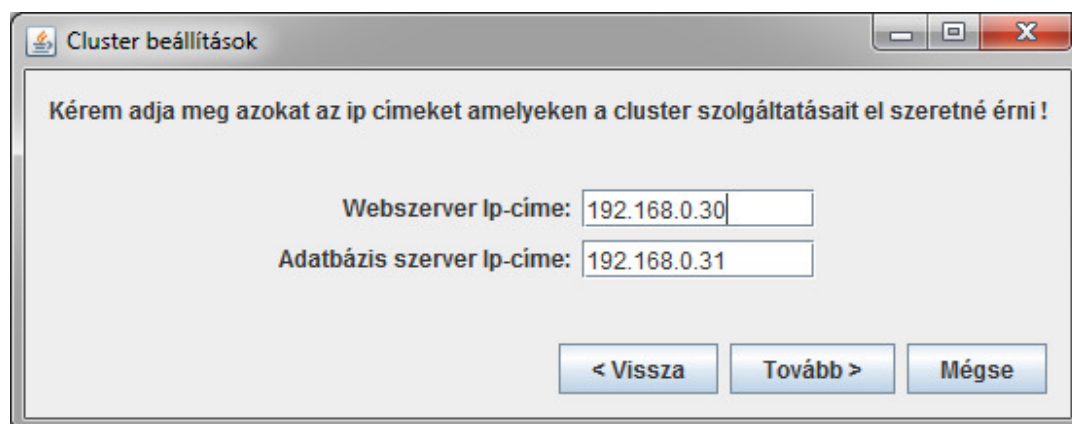
10. ábra A telepítő üdvözlő képernyője is felhívja a figyelmet az előfeltételekre

A következő lépésben fel kell vennünk a két gépet, amelyből létre szeretnénk hozni a clustert. Meg kell adnunk az ip címüket, root jogosultsággal rendelkező felhasználót és a hozzá tartozó jelszót.

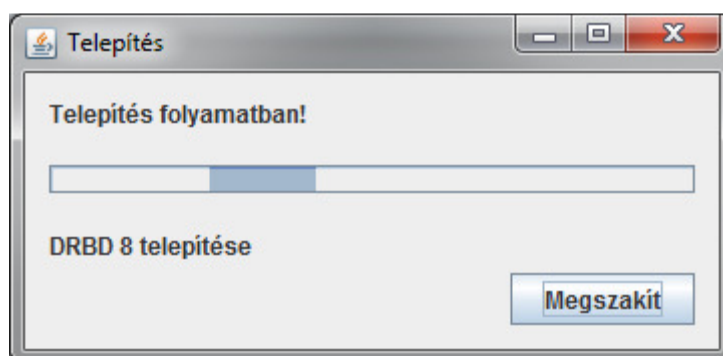


11. ábra A Cluster tagok felvétele

Ezt követően még két ip címet kell megadnunk, amelyeken el szeretnénk majd érni a cluster szolgáltatásait. Ezek után indíthatjuk a telepítést.

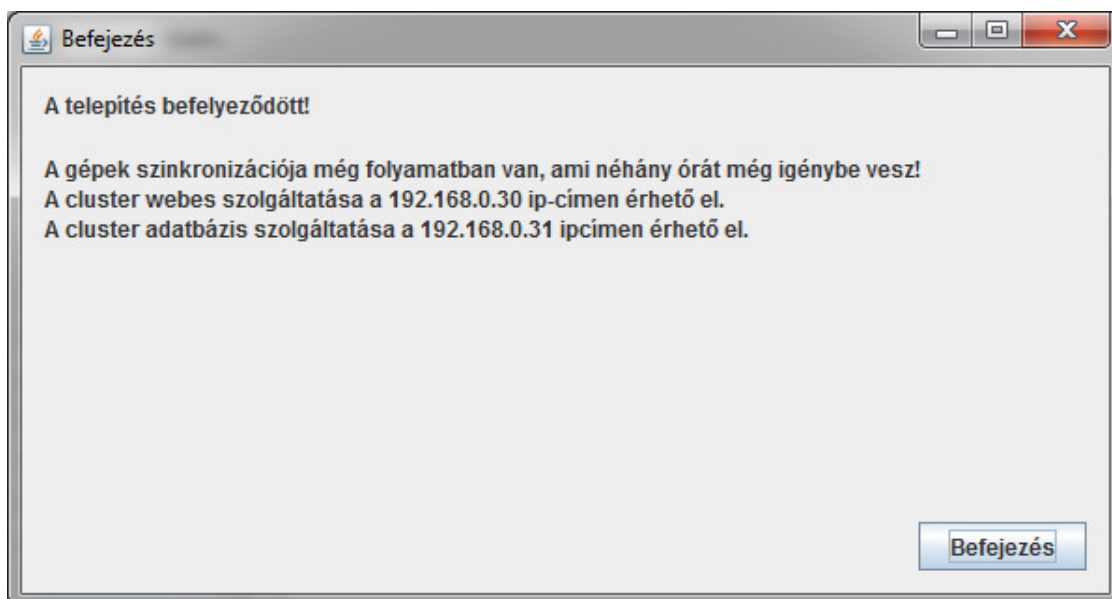


12. ábra További ip címek megadása



13. ábra A telepítés folyamata

Ha elkészült a telepítés, akkor egy befejező képernyő jelenik meg, amely után már használatba is vehetjük a rendszerünket. A gépek szinkronizációja a háttértárak méretétől függően eltarthat még néhány órát, ezt érdemes megvárni és csak utána feltölteni rá a weboldalakokat.



2. ábra Befejező képernyő

7.3 A feltelepített rendszer kritikus pontja(i)

A telepítő szoftver segítségével a clustert, akár egy egyszerű desktop pckre is fel lehet telepíteni. Viszont az egész cluster üzembiztonságát az adat replikáció minősége és sebessége befolyásolja. Ezért az ezt befolyásoló hardverek minőségére nagyobb figyelmet kell szentelni. Mind két gépbe javasolt egy jó minőségű másodlagos gigabites hálózati kártya vásárlása (pl. <http://www.intel.com/content/www/us/en/network-adapters/gigabit-network-adapters/ethernet-server-adapter-i350.html>).

A replikáció sebessége leginkább a lemezek írási sebességétől függ, ezért szintén javasolt gyors lemezek beépítése a gépekbe (10ezer, vagy 15ezer rpm-es diskek).

A háttértárak teljesítményét érdemes megtámogatni, még egy gyorsító tárazásra is képes hardveres RAID vezérlővel(pl.

<http://www.intel.com/content/www/us/en/servers/raid/raid-controller-rs25gb008.html>).

7.4 Összefoglalás

Az elkészült szoftver segítségével, néhány adat megadása után, könnyen hozható létre magas rendelkezésre állást biztosító cluster környezet. A bemutatott fizetős megoldások illetve a saját megoldásunk alapján látható, hogy az általunk fejlesztett program hiánypótló a piacon.

8 Irodalomjegyzék

Ebben a témában információt találni a szoftvereket használó közösségek oldalain lehet. Illetve az online how-to –k állnak rendelkezésre.

- 8.1** DRBD hivatalos oldala: <http://www.drbd.org/>
- 8.2** Pacemaker hivatalos oldala: <http://www.clusterlabs.org/>
- 8.3** Zehon SCP Api oldala: <http://www.zehon.com/>
- 8.4** Ganymed-SSH 2 oldala: <http://www.ganymed.ethz.ch/ssh2/>
- 8.5** OpenAIS hivatalos oldala: <http://www.openais.org>
- 8.6** Linux HA: http://www.linux-ha.org/wiki/Main_Page