# Mitől megy a villamos, avagy a HA-telepekben használt technológiák

Bodnár Csaba, bocs@missioncriticallinux.hu

2003.11.08.

**Kivonat**

Az előadás először áttekinti a Linux-alapú telepek (clusterek) típusait, különös tekintettel a HA-telepekre. Ezután az ezekben használatos különböző technológiákat veszi sorra; vagyis azokat a funkciókat, amelyektől egy rendszer HA-teleppé válik. Végül néhány ismertebb HA-telep szoftvert hasonlítunk össze a fenti funkciók megléte illetve implementációjának minősége szerint.

## Tartalomjegyzék

# 1. A különböző telep-típusok áttekintése

Maga a telep szó (angolul cluster, fürt) nem takar mást, mint hogy valahány számítógépet valamilyen cél érdekében „egybefogunk”, összetartozónak tekintünk. Ha ez a cél a(z)

- magas rendelkezésre állás, akkor magas rendelkezésre állást nyújtó telepről (High Availability, HA, pl. Kimberlite, Red Hat Cluster Manager, ...)
- terhelésmegosztás, akkor terhelésmegosztó telepről (pl. LVS)
- szuperszámítógép teljesítmény elérése, akkor tudományos-technikai célú telepről (High Performance Technical Computing, HPTC, pl. Beowulf)
- hogy egy nagy SMP-gépként látsszon, akkor SSI-telepről (Single System Image) (pl. openMosix, Qlusters, OpenSSI)

beszélünk.

Mivel a legtöbb vállalatnál igazi jelentősége a HA-telepeknek van, a továbbiakban erre koncentrálva áttekintjük, melyek azok a technológiák, amelyek révén egy HA-telep több, mint néhány gép egymás mellé rakva:

# 2. Teljes redundancia (NSPOF, No Single Point Of Failure)

Bár nem technológia, hanem inkább alapelv, érdemes beszélni róla. Mivel HA-telepről van szó, itt a leglényegesebb a magas rendelkezésre állás, vagyis az hogy a rendszer folyamatosan működjön, az esetleges kiesés minimális időtartamra korlátozódjon.

Ennek egyik alapfeltétele, hogy ha valamely hardver eszköz meghibásodik a rendszerben, legyen helyette egy másik, amelyik automatikusan átveszi a szerepét; vagyis ne legyen egyetlen olyan eszköz sem a rendszerben, amelyből csak egy van (Single Point Of Failure).

Ez bizonyos eszközök esetén kis erőfeszítéssel elérhető, más eszközök (pl. hogy egy külső RAID-alrendszerben a külső RAID-vezérlő is redundáns legyen) esetén súlyos százezrekbe vagy akár milliókba kerül. Azt, hogy a tartalék eszközre az átkapcsolás automatikusan megtörténjen, vagy maga a hardver, vagy pedig a HA-telepszoftver intézi el.

# 3. Többségi szavazás (quorum)

Maga a quorum szó latin eredetű (jelentése: akié, akinek a ...), angolszász nyelvterületen a következő jelentéssel bír: Egy szervezet tagjainak olyan gyűlése, amely elég nagy ahhoz, hogy döntőképes legyen.

Telepek esetén is hasonló értelemben használjuk: A telep csak akkor életképes, ha a quorumhoz szükséges számú tag elérhető és működik. Általában a quorumot 50%+1-ben szokták meghatározni, vagyis a telep tagjainak több mint felének működnie kell. E mögött az a logika rejlik, hogy ha valami miatt a kommunikáció megszakad a telep két része között, akkor ne tudjon elindulni mindkettő önálló telepként.

2 tagból álló telepek esetén (ami HA-telep esetén tipikus) a fenti szabályt lehetetlen betartani, hiszen ekkor ha az egyik tag kiesik, akkor leáll a telep; márpedig a HA-telep célja elsődlegesen az, hogy valamely tag esetén a másik átvehesse a szerepét. Így ilyenkor a quorum 50% szokott lenni (vagyis egy tag megléte esetén a telep már életképes).

Gyakran van egy ún. quorum-partíció is egy – az összes tag által elérhető – közös lemezterületen, ezen keresztül jelzik a tagok, hogy el tudják érni a telepet.

# 4. Figyelő csatornák (heartbeat)

Figyelő csatornák segítségével tudja 2 teleptag figyelni egymás elérhetőségét. Az egyik tag rendszeresen jeleket küld a másik felé, ezzel jelezve életképességét. Ha egy ezek a jelek megszűnnek, az azt jelenti, hogy valami probléma van a géppel.

A figyelő csatornák különböző médiákat használhatnak erre a célra. Leggyakoribb a soros vonali és az ethernet hálózati figyelés, de általában bármilyen TCP/IP-t támogató médián mehet az efféle „életjel”. Akár az is elképzelhető, hogy közösen elérhető lemezterületen jelezzék elérhetőségüket egymásnak a teleptagok. Érdemes legalább 2, különböző típusú egyidejű figyelést alkalmazni, így ha valamelyik meghibásodik (pl. kicsúszik a soros kábel), a másikon keresztül még mindig működik a kommunikáció.

Milyen paraméterezési lehetőségek vannak egy figyelő csatorna esetén? Általában állítható a jelek küldésének „gyakorisága”, továbbá a fogadó tag „érzékenysége”, vagyis hogy hány egymás után következő jel elmaradása után kell megkezdenie a beavatkozást.

# 5. Node „lelövésének lehetősége” (STONITH, Shut The Other Node In The Head)

A STONITH szintén gyakran használt módszer HA-telep környezetben, különösen akkor, ha osztott lemezes alrendszer elérést is alkalmazunk. Mit is jelent ez? Segítségével egy teleptag le tud kapcsolni valamely másik teleptagot, ha úgy látja, hogy az már nem elérhető.

Miért hasznos ez számunkra? Tegyük fel, hogy van egy kételemű HA-telepünk, amely megosztva használ egy „hagyományos” fájlrendszert (nem telepfájlrendszert!) osztott SCSI-buszon keresztül. Ebben az esetben fontos, hogy egyszerre csak egy tag érhesse el a fájlrendszerünket, hiszen ha egyszerre ketten írnának, az adatsérülést okozna.

Előfordulhat, hogy (pl. lefagyás miatt) egyik tag úgy látja, hogy a másik már nem elérhető, átveszi annak szerepét, és – többek között – felcsatolja a közös fájlrendszert. Fontos, hogy a lefagyott tag ne tudjon újra „feléledni” és írni a – már más által átvett – fájlrendszerre. A STONITH-technológia éppen ezt garantálja számunkra, azáltal, hogy az átkapcsolás első lépéseként lekapcsolja a lefagyott tagot.

Hogyan kerülhet sor a STONITH fizikai megvalósítására, vagyis a géplekapcsolásra (a STONITH-implementációkban ezek többségéhez van támogatás):

- Speciális soros vonali kapcsolókkal (a gép ezen keresztül kapja az áramot) (pl. WTI RPS-10, APC)
- Speciális ethernet hálózati kapcsolókkal (a fentihez hasonló, csak a vezérlés nem soros vonalon, hanem etherneten keresztül történik; ezekkel egyszerre több esz-

köz is vezérelhető, ez viszont egy HA-telepben SPOF-fé válhat) (pl. WTI NPS-230, APC, NetBay)

- Nagyobb IBM-kiszolgálók esetén a szervízprocesszorhoz tartozó Service Management Porton keresztül
- Intel alaplapok esetén az EMP-port programozásával
- A reset kapcsoló kivezetésével

# 6. Rendszer működőképesség figyelés (szoftveres ill. hardveres watchdog)

Watchdognak nevezzük azt a mechanizmust, amelynek révén egy gép figyeli a saját magán futó alkalmazás által küldött életjeleket; ha a jelek megszűnnek, a gép újraindítja magát. Ezzel szintén kiküszöbölhető a fentebb, a STONITHnál már említett közös fájlrendszer eléréssel kapcsolatos probléma. Mivel mindkétféle megoldás (STONITH, watchdog) kb. ugyanarra a problémára nyújt valamiféle, egymástól különböző megoldást, az ember vagy egyiket, vagy másikat használja, ízlése és lehetőségei szerint.

Maga a Linux-kernel is biztosít egy watchdog szolgáltatást (insmod softdog, mknod /dev/watchdog c 10 130), amelyet szoftveres watchdognak nevezhetünk. Ezek túlmenően vannak ún. watchdog kártyák, amelyek szintén ugyanilyen elven működnek. Általában elmondható, hogy a hardveres kártyák megbízhatóbbak a szoftveres megoldásnál, hiszen a szoftveres megoldás életképességéhez a kernelnek még valamilyen szinten működnie kell.

# 7. Erőforráscsoportok (resource group, service)

A HA-telep infrastruktúra általában biztosít számunkra egy olyan keretet, amelyben alkalmazásunkat, a hozzá tartozó adatokat és egyéb erőforrásainkat elhelyezhetjük. Ezt egyik fajta terminológia szerint erőforráscsoportnak, másik szerint egyszerűen szolgáltatásnak nevezzük. Mik tartoznak tipikusan bele egy ilyen erőforráscsoportba:

- Alkalmazás indító/leállító scriptek
- Maga az alkalmazás, a hozzá tartozó adatokkal
- IP-címek
- Lemezes eszközök, amelyeken az alkalmazás és az adatok elhelyezkednek
- Csatolási pontok

# 8. Alkalmazásfigyelés (Application monitoring)

Gyakori elvárás egy HA-telepszoftverrel szemben, hogy figyelje a rajta futó alkalmazások állapotát, és probléma esetén indítsa újra az alkalmazást. Ezt a legtöbb HA-szoftver támogatja, vagy tervbe van véve a támogatás elkészítése.

Általában az elterjedtebb alkalmazásokhoz (Oracle, mysql, apache, ...) adnak példa figyelő scripteket, amelyek minimális módosítást igényelnek; egyéb alkalmazások esetén pedig nekünk kell megírni a scriptet. Miből is állhat egy ilyen script? Pl. bejelentkezik egy adatbázisba, lekérdez 1-2 rendszertáblát és ha mindent rendben talál, akkor a megfelelő visszatérési értékkel lép ki.

# 9. Telep-állományrendszerek (cluster filesystem)

Telep-állományrendszereknek nevezzük azokat az állományrendszereket, amelyeket egyszerre több teleptag egyidejűleg is használhat. Míg egy hagyományos fájlrendszert egyszerre csak egy gép csatolhat fel (mount), egy osztott fájlrendszerrel ezt többen is megtehetik.

Az utóbbi években jelentős fejlődésen ment keresztül ez a fajta technológia is, bár még általában nincs beintegrálva a HA-szoftverekbe – viszont különösebb integrációra nincs is szükség, hiszen a fájlrendszer, amit eddig a HA-szoftvernek kellett külön csatolnia erőforráscsoport indítása estén, az most mindig minden teleptagon rendelkezésre áll, tehát nem kell külön foglalkozni vele.

A legismertebb GPL-licenszes implementáció az OpenGFS.

Az egyik legfontosabb probléma maga a tény, hogy egyidejűleg több teleptag érheti el ugyanazt a csatolt fájlrendszert, vagyis a – tagok közötti – zárolás (locking) megoldása. Az OpenGFS-ben az egyes tagok memexp moduljai működnek együtt a zárolás folyamatos nyilvántartása érdekében. A zárolásokat egy közös helyen tárolják, ami lehet:

- Egy közös lemezterület (elkülönítve a fájlrendszertől és a naplózástól (journal)), amelyet DMPE-eszköznek ( Device Memory Export Protocol) neveznek
- Egy memexpd hálózati démon, amely vagy memóriában, vagy a helyi lemezen tárolja a zárolásokat

Az OpenGFS felhasználók többsége az utóbbit használja, bár szerintem az előbbi HA-szempontból jobb megoldásnak tűnik, hiszen nincs egyetlen olyan kitüntetett gép, aminek kiesése esetén a zárolási információk elvesznek.

# 10. Összehasonlító adatok

Az alábbi táblázatban néhány ismertebb, GPL-licenszelésű HA-szoftver tulajdonságait hasonlítjuk össze, a fenti szempontok szerint:

# 11. Összefoglalás

A fenti technológiák alkalmazásával komplett HA-megoldások építhetők. A HA-telepszoftverek fejlődése során – egyre újabb és újabb technológiák beépítése révén – a végső cél a egy SSI-típusú telepszoftverré válás lehet, amely egyetlen nagy rendszernek láttatja a telepet és egyaránt rendelkezik a HA és a terhelésmegosztó képességekkel.

| Jellemzők | Heartbeat | Kimberlite | Red Hat Cluster Manager | Failsafe |
|---|---|---|---|---|
| Quorum van? | nem | igen | igen | nem |
| Quorum-eszköz van? | nem | igen | igen | nem |
| Figyelő csatornák | igen | igen | igen | igen |
| STONITH támogatás | igen | igen | igen | igen |
| watchdog támogatás | igen | nem | igen | nem |
| Erőforrás csoportok | igen | igen | igen | igen |
| Alkalmazás figyelés | nincs | nincs | van | van |

1. táblázat. HA-szoftverek összehasonlítása funkcionalitás szerint

# 12. Hivatkozások


- http://oss.missioncriticallinux.com/projects/kimberlite/
- http://www.linux-ha.org/
- http://www.redhat.com/
- http://oss.sgi.com/projects/failsafe/
- http://opengfs.sourceforge.net/