Üzembiztos kiszolgálótelepek

Az utóbbi időben egyre több állandó rendelkezésre állást ígérő szolgáltatás jelenik meg a piacon. Célszerű utánanézni, hogy igényeinknek melyik megoldás felel meg a legjobban.

A jelenlegi éles versenyhelyzetben különösen gyakran halljuk „az idő pénz” kifejezést. Az üzleti adatok hálózati tárolása és folyamatos elérhetősége képezi a vállalati kiszolgálók lényegét. Legyen szó háttérben működő adatbázisokról, levelek és felhasználók nyilvántartásáról vagy hálózati fájlrendszerekről (NFS), az adattárolási rendszer üzemzavara gyászos következményekkel járhat. A leginkább költségkímélő megoldásnak egy hibatűrő géptelep (más szóval kiszolgálófürt) felállítása tűnik. A fogalom valójában több kiszolgáló összekapcsolását jelenti, melyek bármelyike képes a többi adatbázis- vagy alkalmazás-kiszolgáló feladatait azonnal átvenni. Ha a telep egyik tagja meghibásodna, a többi gép átveszi a leállt kiszolgáló által működtetett szolgáltatásokat. Ez az átvétel szerencsés esetben úgy történik, hogy a felhasználó észre sem veszi.

Egy jellemző megoldás például, hogy több számítógép csatlakozik egy megosztott adattárolóhoz, ami általában SCSI vagy FibreChannel adatsínre kapcsolt lemezeket jelent. A vállalati felhasználásra készült telepeket eredetileg kizárólag a jól ismert cégek, például a Digital, a HP és az IBM gyártottak. A Linux-alapú, olcsó gépeken is működő rendszerek csak nemrégiben váltak elérhetővé.

A világhálón körülnézve bárki találhat többféle, Linuxra épülő teprendszert is, ezek legtöbbje igen vonzónak tűnik – legalábbis papíron. Az ígéretekben általában az szerepel, hogy a rendszer szempillantásnyi idő alatt elvégzi a váltást, legyen szó akárhány elemből álló telepről és tetszőleges számú szolgáltatásról. Könnyen előfordulhat, hogy nem jó megoldásszállítót választunk. Az igazság az, hogy nem minden magas rendelkezésre állást biztosító rendszer növeli adataink megbízhatóságát és elérhetőségét. Éppen ellenkezőleg – egy nem megfelelő választással értékes fájlrendszereinket és adatbázisainkat tulajdonképpen kiszolgáltatjuk a rossz szándékú behatolóknak. Néhány gyártó igyekszik elhallgatni ezt, mások esetében pedig csak hosszas kutatómunka után lelhetjük fel e tényeket a szerződésben.

Jómagam több mint hét éve dolgozom a Unix/Linux-alapú, állandó rendelkezésre állást biztosító rendszerek területén, és jó néhány termék tündöklésének, majd csúfos bukásának voltam tanúja. Felháborítónak tartom, hogy egyes rendszerek hirdetéseiben nagy mellénnyel hivatkoznak olyan szolgáltatásokra, melyeket nyilvánvalóan képtelenek nyújtani. Itt a felhasználó értékes adatai kerülnek veszélybe, ráadásul a hozzá nem értő cégek botrányai rossz fényt vetnek az egész szakmára. Hosszú évek tapasztalatát összegzi az a négy pontból álló lista, mely segítségével felmérhetjük, hogy az adott termék képes-e megfelelni az általunk támasztott követelményeknek. Valójában ezek a szempontok nemcsak a Unix/Linux felületre igazak, hanem bármilyen más operációs rendszer és géptípus esetében is alkalmazhatók. Tehát mielőtt egy fillért is kiadnánk egy magát tökéletesnek hirdető szolgáltatásért, győződjünk meg arról, hogy használata valóban megvédi rendszerünket és adatainkat az alábbi négy helyzetben:

1. Tervezett karbantartás és leállás.
2. Rendszerösszeomlás.
3. Kapcsolattartási zavar.
4 Rendszerlefagyás.

Mind a négy helyzetet nemsokára részletesen tárgyaljuk, és a jellemző hibákról is szót ejtünk, előtte azonban engedtessék meg némi magyarázat arról, hogy mit jelent az adatok épsége. Ennek egyik alapköve, hogy az adatok pontosak és frissek legyenek. Ez eddig elég egyszerűen hangzik, nemde? Egy számítógéptelepen az adatok épsége bír a legnagyobb fontossággal, sorrendben még a folyamatos elérhetőséget is megelőzi. A példák tanulmányozásával bizonyára mindenki érteni fogja, hogy miről beszélek. Az *1. ábra* egy kételemű telepet mutat be, az A és B jelű elemeket egy megosztott SCSI sín köti össze az 1. lemezzel (az egyszerűség kedvéért használtam két elemet, a példa természetesen tetszőleges elemszám esetén is alkalmazható).

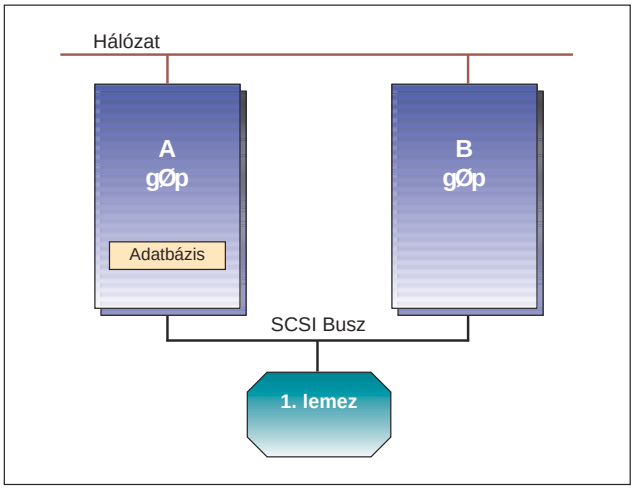
A legtöbb operációs rendszerben a lemezen lévő adatokat fájlrendszeren keresztül érthetjük el. Általában a fájlrendszer befűzi a tárolólemezt, majd létrehozza a felhasználói kapcsolatokat. A nagyobb teljesítmény eléréséért a fájlrendszerek saját adataik friss másolatait a memóriában tárolják. Ebből következik, hogy adataink legfrissebb változata (amit a példában az A elem szolgál ki), az A gép memóriájában tárolt, illetve a lemezen lévő adatokból áll össze.

Nézzük most meg a B nevű gépet. Ha B ugyanezt a fájlrendszert próbálná meg befűzni és használni, a fájlrendszer valódi adatait az A memóriájában tárolt, a B memóriájában lévő, valamint a lemezen elhelyezkedő adatok együttese képezi. Ahhoz, hogy ez a rendszer hibamentesen működjön, olyan fájlrendszerre van szükség, mely képes az egyszerre több helyen tárolt adatok kezelésére. Az olyan rendszereket, ahol több gép is befűzheti ugyanazt a fájlrendszert, telepfájlrendszernek hívjuk. Csupán néhány Unix-változat tartalmaz ilyen jellegű támogatást, linuxos környezetben pedig nem tudok ilyenről (bár valami már készülődik – látogassunk csak el a GFS projekt ➲ http://www.gfs.lcse.umn.edu/ honlapjára).

A telepfájlrendszerek hiányában mi történik akkor, ha több elem egyszerre kísérli meg elérni ugyanazt a fájlrendszert? A következő esetek képzelhetők el:

- *Pontatlan adatok.* Tegyük fel, hogy a Las Vegas-i utunk jól sikerült, és száz dollárt szeretnénk a számlánkra tenni. Mondjuk, hogy a befizetést az A gép kezelte, melynek eredményeképpen a korábbi 25 dolláros egyenlegünk 125 dollárra emelkedett, ezt az A (mint legfrissebb adatot) mindjárt el is helyezi memóriájában lévő gyorstárában. Ekkor, mondjuk hazarepülünk, és kiderül, hogy autónkat csak ötven dollár ellenében vehetjük át a parkolóházból. Ezt a műveletet most B végzi, mely a lemezhez fordulva még mindig 25 dollárt lát, és a „Nincsen elég pénz a számláján” üzenetet küldi. Mindez azért fordulhatott elő, mert a valódi egyenleget (125 dollár) az A gép a saját memóriájában tárolja. A telepek felépítésénél tehát föl kell tennünk a kérdést: milyen kárt okozhat a vállalat számára, ha téves adat adódik továbbításra?
- *Rendszerleállás.* A felhasználói adatok (például a számlaegyenleg) mellett a fájlrendszerek saját értékeiket is a lemezen tárolják, amelyek a felhasználói adatok elrendezésének módját határozzák meg (ezeket legjobban egy tartalomjegyzékhez lehetne hasonlítani). A teljesítmény megtartása érdekében ezek az értékek szintén bekerülnek a memóriában lévő gyorstárba. A fájlrendszerek hamar összezavarodnak, ha ezek az értékek megsérülnek, és egy-egy ilyen hiba általában teljesen

kiborítja őket (mi ezt csak rendszerösszeomlásnak hívjuk). Egy valódi telepfájlrendszer hiányában, ha egy adatot két gép próbál meg elérni egyszerre, az kavarodásokhoz vezet a tartalomjegyzékben, és ebből származnak a „legtakarosabb” rendszerösszeomlások.
Ha egy fájlrendszer valamelyik adata vagy tartalomjegyzéke megsérül, az adatsérülést is okozhat, azt pedig csak a legutolsó biztonsági mentésről történő visszaállítással háríthatjuk el. (Mindenki gyakran és rendszeresen végez biztonsági mentést, ugye? A 64. oldalon olvashatunk egy praktikus megoldást is.) A baj csak az, hogy a műveletek (és az ezekkel



1. ábra Kétgépes telep egy megosztott SCSI lemezzel

járó adatváltozások) sokkal gyakrabban végrehajtódnak, mint a biztonsági mentések, és így az adatsérülés helyrehozása gyakran napokba is beletelik, „pedig a hirdetésben másodperceket emlegettek...”
A fenti elv, mely szerint az elemeknek összehangoltan kell elérniük a fájlrendszereket, az adatbázisokra is érvényes. A legtöbb adatbázis-megoldás nem teszi lehetővé az egyes elemek számára, hogy ugyanazt az erőforrást egyidejűleg érjék el. Figyelemre méltó kivételként az Oracle Parallel Servert (a linuxos változat már készül) és az Informix Extended Parallel Servert említenénk meg.
Az eddig vázoltak lényege, hogy a választott teleprendszernek lehetővé kell tennie, hogy egy fájlrendszert vagy adatbázist egyidejűleg csak egyetlen gép módosíthasson – ez így elég egyszerűen hangzik, de olyan teleprendszert találni, amely ezt az idő száz százalékában képes megvalósítani, már sokkal nehezebb lesz. Most lépjünk tovább és vizsgáljuk meg, hogy mindezek mennyiben érintik a fent említett négy helyzetet.

Tervezett karbantartás

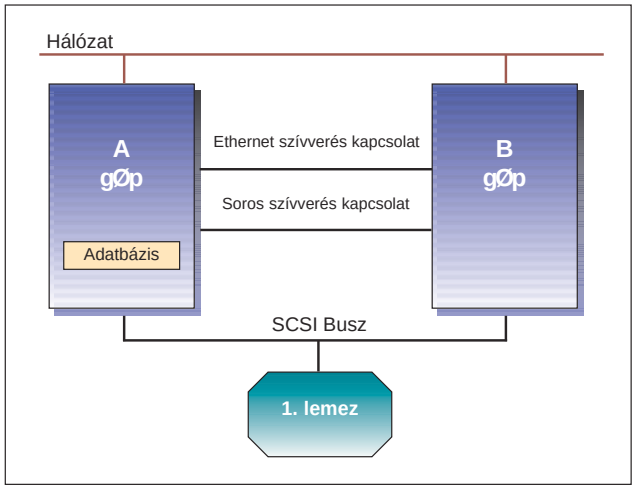
A magas rendelkezésre állású telepek egyik legszebb (és legjobban mellőzött) tulajdonsága, hogy bármelyik gépet eltávolíthatjuk anélkül, hogy ez zavarná a gépen futó szolgáltatásokat (hiszen a többiek átveszik annak feladatait). Így például egyszerűen telepíthetjük valamelyik program legfrissebb változatát, vagy memóriával bővíthetjük a gépet, és eközben a telep egyetlen szolgáltatása sem szünetel, egy tizedmásodpercig sem. Majdnem minden teleprendszer képes így kezelni a tervezett karbantartással járó leállásokat.

Rendszerösszeomlás

Ha valaki azt hiszi, hogy vannak fagyásbiztos operációs rendszerek, az csak szóljon nekem, majd én mutatok neki egy-két érdekes dolgot. Nézzünk szembe a tényekkel: a rendszerösszeomlás az élet része.
A dolgunk „mindössze” az, hogy gyakoriságukat a lehető legkisebbre csökkentsük. Ha a telep egy gépe leáll, a többi erről azonnal tudomást szerez és elosztja a kiesett gépen futó szolgáltatásokat egymás között. Rendszerösszeomlás esetén elméletileg minden teleprendszer képes a meghibásodott gépek feladatait szétosztani. És ez így nagyon jól van – tehát ezek alapján bármelyik rendszert választhatnánk? Na, azért ennyire ne szaladjunk előre. Az alábbi szempontok alapján szépen különválaszthatjuk a valóban értékes megoldásokat a használhatatlanoktól.

Kapcsolattartási zavar

Általában a magas rendelkezésre állású telepek gépei folyamatosan kapnak egymásról adatokat (cluster interconnects). Régebben a nagy telepgyártók saját alkatrészekkel oldották meg e kapcsolatok kezelését. Ez a módszer megbízható, de természetesen nagyon drága és az adott céghez köt bennünket. Olcsóbb megoldást találunk számos más rendszerben: ezeknél a hagyományos hálózaton (Ethernet vagy PPP) keresztül történik az adatcsere. Az ilyen megoldásoknál az egyes gépek rendszeresen váltanak üzenetet, és a kapott válaszok alapján döntenek arról, hogy a kérdéses gép működik-e vagy sem. Az adatforgalom ezen típusát „szívverésnek” (heartbeat) nevezzük.



2. ábra Kétgépes telep Ethernet és soros kapcsolattal

A szívverés-alapú telepek leggyakoribb gondja a kapcsolattartási zavar. Ez akkor következik be, amikor a telep tagjai működnek, de nem képesek kapcsolatba lépni egymással. Nézzük például a *2. ábrán* látható Etherneten és soros kábellel csatlakozó kételemű telepet.
Tegyük fel, hogy üzembe helyeztük a telepet, majd a hétvégére Las Vegasba utaztunk. Ezalatt otthon a szokásos nagytakarítás közben valaki a seprűvel kirántja a hálózati vezetéket. Ilyenkor a két gépnek el kell döntenie, hogy a magas rendelkezésre állás szem előtt tartása mellett mit tegyen. A gépek nem képesek elérni egymást, ezért önállóan kell meghozniuk ezt a döntést. Az alábbiakban felvázolunk néhány elvet. Ezeket számos megvásárolható teleprendszer alkalmazza (sajnos):

- *A lehető legrosszabb eset feltételezése.* Az A gép tudja, hogy ő szolgálja ki az adatbázist, de nem tud B állapotáról, tehát A folytatja az adatbázis elérését. Eközben B nem képes kapcsolatba lépni A-val, így feltételezi, hogy A kiesett. B ekkor szintén megpróbálja kiszolgálni az adatbázist, ami a már említett kettős hozzáféréshez és rendszerösszeomláshoz vezet (bármilyen hihetetlen, de sok rendszer működik így!).
- *A lehető legjobb eset feltételezése.* Egy, az egész telepre kiható áramkimaradás után A és B egyszerre indul el. Egyik elem sem képes meghatározni a másik állapotát, és a biztonság kedvéért egyikük sem kezdi el az adatbázis kiszolgálását. Mindkettő azt feltételezi ugyanis, hogy a másik működik (csupán kapcsolattartási zavar lépett fel), és a kettős elérés megelőzésére inkább kiszáll a játékból. A végeredmény: mindkét gép üresjáratban várakozik, a szolgáltatások pedig nem működnek. Ez sem tűnik kellemes helyzetnek, de az átmeneti leállás még mindig sokkal jobb, mint ha adataink sérülnének egy rendszerösszeomlás miatt.

Más hibák is tűnhetnek kapcsolattartási zavarnak, például:

- egy hálózati kártya tönkremegy,
- a gépeket összekötő útválasztó elromlik,
- egy Ethernet-vezeték meghibásodik.

E kapcsolattartási zavarok elkerüléséhez alkalmazzunk többféle kapcsolatot a gépek között. Beállíthatjuk a rendszert, hogy az elemek több kapcsolaton, vagy Ethernet- és soros kapcsolaton egyszerre beszélgethessenek egymással. Ugyanígy az is megvalósítható, hogy a hálózati kapcsolatok külön jelelosztókon és útválasztókon keresztül haladjanak, vagy alkalmazzunk kizárólag soros kapcsolatokat.
A legtöbb teleprendszer lehetővé teszi, hogy többszörös kapcsolatokat alakítsunk ki, ezzel is csökkentve a kapcsolattartási zavarok lehetőségét. Ha a választott rendszer nem képes erre, akkor jobban tesszük, ha azonnal egy másikra állunk át.

Rendszerlefagyás

Ez a legveszélyesebb hiba, ami egy teleprendszerrel megtörténhet. Mindenki látott már olyat, hogy egy gép valamilyen különös okból egyszerűen felfüggesztette működését, és olyankor csak a Reset gomb vagy a ki-, majd újra bekapcsolás segített. Szerencsére a dolog ritkán fordul elő. Legalább ennyire furcsa az is, amikor a lefagyott gép egyszer csak ismét működni kezd. Talán már olyan esettel is akadt dolgunk, amikor egy gép megállt, majd kis idő elteltével ismét válaszolt a kérelmekre, tehát működött tovább. Ez bármelyik operációs rendszerrel megtörténhet.
A dolog annyiban érinti a teleprendszereket, hogy tisztában kell lennünk azzal, miként viselkedik a telep egy esetleges lefagyáskor, majd ismételt elinduláskor. Nézzünk most egy példát, mellyel talán sikerül rávilágítanunk erre a nagyon fontos tényezőre. Tegyük fel, hogy az A gép nem válaszol. Mivel előre gondoltunk a kapcsolattartási zavarokból származó hibákra, ezért a telep gépei két Ethernet- és egy soros kábellel csatlakoznak egymáshoz. Nos, lefagyás esetén teljesen mindegy, hogy egy vagy ötven kapcsolat van a gépek között – a „szívverést” ezek egyike sem képes továbbítani, hiszen a gond magával a géppel van. B észleli, hogy A a három csatorna egyikén sem válaszol az üzenetekre, és feltételezi, hogy A leállt. Ekkor B megpróbálja átvenni A feladatait: befűzi a fájlrendszereket és kiszolgálja az adatbázist. De ekkor A ismét működni kezd és folytatja az adatbázissal félbehagyott munkát. Ugye innen már ismerős a helyzet: az erőforrást a két gép egyszerre próbálja meg elérni, és ennek eredményeképpen ismét egy csinos kis rendszerösszeomlás következik be.
Ez utóbbi helyzetben bizonyíthatja a telep, hogy valóban képes teljes mértékben ellátni feladatát. Az adatok épségének megvédéséért a szolgáltatás(ok) átvétele előtt minden elemnek meg kell győződnie arról, hogy a leállt elem nem módosíthatja a fájlrendszert vagy az adatbázist. Ezt a módszert I/O korlátozásnak vagy I/O elrekesztésnek (I/O Fencing, I/O Barrier) hívjuk. Néhány gyártó e kellemetlen gondot egyszerűen azzal intézi el, hogy nem törődik vele, mondván, úgysem túl gyakori helyzetről van szó. Szerencsére ez így is van, de ne felejtsük el, hogy az adatok biztonságának megőrzéséért a lehető legvalószínűtlenebb helyzetben is azonnal cselekednie kell a rendszernek.

Összefoglalás

Aki minden körülmények között az adatok épségét tartja a legfontosabb szempontnak, az a teleprendszer felépítése előtt győződjön meg arról, hogy a választott megoldás mind a négy fent említett tárgykörre képes megfelelő választ adni. Soha ne felejtsük el, a hálózat legfontosabb feladata, hogy az adatok minden pillanatban elérhetők és helyesek legyenek. Ha az adatok épségének megőrzésén takarékoskodunk, az hosszabb leállási időket és elvesztett ügyfeleket jelenthet – ezek bármelyike akár beláthatatlan következményekkel is járhat.

Tim Burke a Mission Critical Linux, Inc. telepmérnöke.
➲ http://www.burke@missioncriticallinux.com/

Szókincstár

Fontosnak tartjuk, hogy kialakuljon végre egy egységes szóhasználat az informatika, és ezen belül a Linux világában. Ennek érdekében igyekszünk minden hónapban összegyűjteni néhány kérdéses szót. Ha bárkinek ötlete, véleménye, kérdése van bármelyik szóval, vagy egyáltalán a témával kapcsolatban, kérjük, írja meg nekünk a szokincstar@linuxvilag.hu címre.

Szótárba

alias	másodnév
gateway	átjáró
hard link	közvetlen (direkt) hivatkozás
hot-swappable	menet/használat közben cserélhető
hub	(hálózati) jelelosztó, elosztó
implementation	megvalósítás
integrity	épség (adaté, hivatkozásé)
interface	illesztőfelület, illesztő, csatolófelület, kezelőfelület, felület
kernel	rendszermag
link	hivatkozás
option	lehetőség
package manager	csomagkezelő
packet filtering	csomagszűrés
platform	felület, géptípus
port	kapu
process	folyamat
router	útválasztó
script, shell script	parancsállomány, héjprogram
service	szolgáltatás
session	munkafolyamat
shell	héj
switch	hálózati kapcsoló
symbolic (soft) link	közvetett (áttételes) hivatkozás
thread	szál

Kérdéses szavak

Az itt következő szavakra találtunk ugyan magyar megfelelőt, de vagy mi magunk sem vagyunk vele teljesen megelégedve, vagy nem tudjuk, hogy minden szempontból, minden szövegkörnyezetben megfelelő-e.

concurrent	együttfutó, egyidejű, versenyhelyzetben lévő
console	konzol
cracker	kalóz, betörő
exploit	akna (a biztonság témakörében)
firewall	tűzfal
hacker	betyár (*A hálózatokhoz és a gépekhez értő jóindulatú szakember, aki képes felderíteni egy rendszer biztonsági réseit.)*
named pipe	nevesített cső/csővezeték
redundancy	többszörös adattárolás, adattöbbszörözés

Ötletet várunk

Ezek a szavak kemény diók. Legtöbbjüket használjuk jövevényszóként, de reméljük, hogy találunk helyett magyar szót. Ha bárkinek van jó ötlete, szeretettel várjuk!

IP multicast, unicast	
record	bejegyzés, sor (néha)
recursive transaction	
socket	illesztőpont, csatlakozó, foglalat, aljzat
compile, link, build	fordít, összeállít, elkészít