Jegyzőkönyvezés a ReiserFS segítségével

Ismerjük meg a Reiser fájlrendszert,
milyen a felépítése és miféle szolgáltatásokat nyújt.

Napjainkban a Linuxszal együtt megjelent néhány új fájlrendszer is, olyan szolgáltatásokkal, melyeket eddig mind az asztali gépek, mind a kiaszolgálók esetében hiányoltunk. Először röviden áttekintjük a ReiserFS néhány fontos szolgáltatását, majd kicsit részletesebben kitérünk a jegyzőkönyvezési réteg működésére.
A ReiserFS a fájlrendszer minden objektumát egy egyszerű B* fában tárolja. A fa a következőket támogatja:

- dinamikus fájlleíró-kiosztás (i-node allocation)
- tömör, indexelt könyvtárak
- átméretezhető elemek
- 60 bites eltolások.

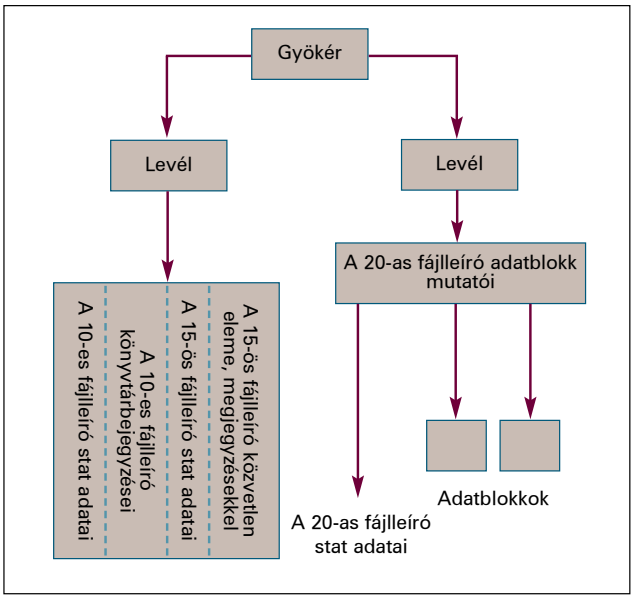
A fában található elemek négy alapvető csoportra oszthatók: a kimutatásadatokra, a könyvtárelemekre, a közvetett és a közvetlen elemekre. Az elemek közt egy kulcs alapján lehet keresni. A kulcs egy azonosítót, a keresett objektum eltolását és az elem típusát tartalmazza.
A ReiserFS könyvtárai tartalmuk változását követve növekednek és csökkennek. A fájl eltolását a könyvtárban a fájlnév egy darabjának használatával tartja nyilván a rendszer. Az így kezelt fájlbejegyzéseket egy fában tárolva nagyméretű könyvtárak hozhatók létre, miközben nem kell a teljesítmény különösebb csökkenésétől tartani, illetve megőrizhető az NFS és a megszokott könyvtárműveletek megfelelő támogatása.
A fájlok esetében a közvetett elemek adatblokkokra mutatnak, a közvetlen elemek pedig becsomagolt fájladatokat tartalmaznak. Így a becsomagolt fájladatok tárolása közvetlenül a fában történik, és a fa csomópontjaiban lévő helyet meg lehet osztani más objektumok elemeivel is. Tehát a nagyméretű fájlokhoz a ReiserFS az ext2 által használtakhoz hasonló blokkmutatókat tárol, de a kisebb fájlok adatait képes egybecsomagolni, ezzel lemezterületet takarít meg.
A fa egyensúlyának megtartásával a fenti elemek mindegyike átméretezhető. Mód nyílik a becsomagolt fájladatokhoz való hozzáfűzésre. Ha a kimutatásadatok közt újabb mezőre van szükségünk, és a lefoglalt terület megnövekszik, be tudja fogadni az újabb adatokat. A lemezformátum sokkal több részletére érdemes kitérni, ezért az érdeklődőknek ajánlom, nézzenek el a ReiserFS honlapjára (lásd a Kapcsolódó címet).

Nagyméretű fájlok támogatása

A ReiserFS jelenleg két fő lemezformátummal rendelkezik.
A 2.4-es kóddal együtt bevezetett új formátum 60 bites fájleltolásokat tesz lehetővé, míg a 2.2-es kódban használt formátum 32 bites eltolásokkal működik. Amikor egy régebbi fájlrendszert fűzünk be az új rendszermaggal, a régi formátumot a rendszer megőrzi, és nem engedi nagyméretű fájlok használatát.
Létezik olyan befűzési lehetőség is, mely az új formátumra alakítja át a fájlrendszert, azonban ennek 2.2-es rendszermag alatti változata egyelőre bétaállapotú. Nem szeretnék elavult adatokat közölni a cikkben, így inkább a ReiserFS honlapjának meglátogatását javaslom, ahol további részletek találhatók a nagyméretű fájlok támogatásáról.

Hogyan működik a jegyzőkönyvezés?

Mielőtt a jegyzőkönyvezés működéséről beszélnénk, először tekintsük át a megoldandó feladatot. Ahhoz, hogy a rendszer esetleges összeomlása után is hibamentes fájlrendszerünk legyen, a frissítésnek atominak kell lennie, azaz vagy teljesen végrehajtódik, vagy semennyire. Például ha blokkokat szeretnénk fűzni egy fájlhoz, frissítenünk kell a fájl blokkmutatóit, blokkokat kell keresni a szabad blokkok listájából, és frissíteni kell a szuperblokkot. Ha a rendszer a változtatások közben összeomlik, lehet olyan fájlmutató, mely továbbra is a szabad blokkok listáján lévő blokkra mutat, vagy a szuperblokk kimutatásadatai nem frissülnek, esetleg a lefoglalt terület elvész (azaz sem a fájlban, sem a szabad blokkok listáján nem szerepel majd).



1. ábra

A ReiserFS jegyzőkönyve egy egyszerű, előreíró jegyzőkönyvezési rendszer (kizárólag metaadatokkal dolgozik). Ennek alapötlete az, hogy mielőtt bármilyen változtatást rögzítenénk a lemezen, azt előbb a jegyzőkönyvbe írjuk. Összeomlás esetén a végrehajtott műveletsorokat újra lejátsszuk, ez lényegében nem jelent mást, mint a kérdéses adatok másolását a jegyzőkönyvből a fő lemezterületre.
Tulajdonképpen nem a változtatások jegyzőkönyvbe rögzítése, az ami megnehezíti a jegyzőkönyvezést. A feladat nehézkes része az, hogy a fájlrendszert ne lassítsuk le egy ráérős teknősbéka sebességére.
A sebesség megtartásának legnyilvánvalóbb módja az, hogy a jegyzőkönyvet nagy méretű, egymást követő adagokban írjuk ki, így csökkentjük a kiírt blokkok számát. A legtöbb fájlművelet kisszámú blokkon végez változtatást, a jegyzőkönyv ezt a tulajdonságot kihasználva több műveletet is képes egyetlen atomi egységben kezelni.
A módosított területeket nem lehet kiírni, míg át nem másoltuk őket a jegyzőkönyvbe, és nem szabadíthatók fel, míg ki nem írtuk őket.
A nagyobb méretű műveletek több magmemóriát foglalnak le ugyan,

de egyéb egyszerűsítési lehetőségeket is felvetnek. Mivel a ReiserFS mindent egy kiegyensúlyozott fában tárol, a fát gyakran kell módosítani és kiegyensúlyozni. A fa blokkjait lefoglaljuk, módosítjuk, majd egy későbbi kiegyensúlyozás során felszabadítjuk. A nagyobb méretű műveletekkel növeljük annak esélyét, hogy egy-egy blokk felszabadul, mielőtt rögzítenénk a jegyzőkönyvbe vagy a lemezre.
A blokkok esetében gyakori, hogy újra és újra jegyzőkönyvezzük őket. Ha a szuperblokkot az első, a második és a harmadik művelet is érinti, mindegyiknél ki kell egyszer írni a jegyzőkönyvbe. A lemezre azonban nem kell rögzíteni, csak miután a harmadik művelet is véget ért. Az írások száma ezzel összességében kisebb lesz, és ezek túlnyomó része is a soros felépítésű jegyzőkönyvbe történik. Ezzel egyes esetekben előfordulhat, hogy a jegyzőkönyvezés gyorsabb lesz, mint az eredeti fájlrendszer.
Amikor lehetséges, a be- és kiviteli műveletek jegyzőkönyvezését egy külön szál végzi, a kreiserfsd. Ennek révén lehetővé válik a háttérben történő végrehajtás, a felhasználói folyamatok lelassítása nélkül. Emellett viszont a jegyzőkönyv meghatározott méretű, így elképzelhető, hogy a felhasználói folyamatoknak várakozniuk kell, amíg egy-egy új művelethez hely szabadul fel a jegyzőkönyvben. Kulcsfontosságú feladat, hogy a jegyzőkönyvre várakozó folyamatok ne kössenek le olyan erőforrásokat, melyekre a már műveletet végző folyamatoknak is szükségük van.
A fájlrendszerek többségének nem kell tisztában lennie azzal, hogy egy jegyzőkönyvezési réteg is gondoskodik a dolgok biztonságos menetéről, van azonban néhány olyan szabály, amit be kell tartani. Először is nem biztonságos a piszkos pufferek módosítása. SMP-rendszerek esetében egy másik processzor is írhat a pufferbe, míg azt módosítjuk. Ez azt jelenti, hogy a módosítások a művelet teljes végrehajtása előtt a lemezre íródhatnak.
A legtöbb művelet csak korlátozott számú puffert módosít, de a fájlokba történő írások és a csonkítások gyakorlatilag korlátlanok. Ahelyett, hogy a jegyzőkönyvezési rétegben támogatnánk a végtelen hosszú műveletsorokat, inkább összefüggőségi támpontokat iktatunk be a műveletsorokba. Ha a folyó műveletsort be kell fejezni, a fájlrendszer összefüggő állapotának fenntartásához elegendő mennyiségű adatot írunk a jegyzőkönyvbe, majd új műveletsort indítunk el. Ha adat-jegyzőkönyvezést is használunk, az fsyncnek is hasonló ellenőrzéseket kell végeznie.
Szintén a jegyzőkönyvezési réteg által hozott új szabály a kizárólag metaadatokat használó jegyzőkönyvezésnél jelenik meg a blokkok újrafelhasználásával kapcsolatban. Képzeljük el a következő két műveletet:

1. A 200-as blokk lefoglalása, beillesztés a fába.
   A 200-as blokk megváltoztatása és jegyzőkönyvezése.
   A 200-as blokk felszabadítása.
   Az első műveletsor lezárása és végrehajtása.
2. A 200-as blokk lefoglalása adatblokknak.
   A 200-as blokk megváltoztatása, fsync a lemezre.
   A második műveletsor lezárása és végrehajtása.

2. ábra

Rendszerösszeomlás után

Az összeomlást követően a műveletsorokat szabályosan újrajátsszuk. Az egyes műveletsorok újrajátszása közben a 200-as blokk jegyzőkönyvezett változatát a fő lemezre másoljuk, a második műveletsor újrajátszását követően pedig a 200-as blokk egy adatblokk egy fájlban. Csakhogy az fsync által a 200-as blokkba írt adatok már nincsenek a helyükön. A ReiserFS úgy kerüli el ezt a helyzetet, hogy sosem foglal le adatblokkot, amíg nullára nem csökken annak esélye, hogy egy jegyzőkönyv-visszajátszás elavult adatokkal felülírhatja annak tartalmát. Amikor a fájlrendszer betelik, azt jelenti, hogy a műveletsorokat ki kell írnunk a lemezre, és újra felhasználható blokkokat kell találnunk. Ugyanilyen ellenőrzéseket kell végrehajtani, ha egy adatblokkot jegyzőkönyvezzünk, majd később közvetlenül felülírjuk.
Most, hogy nincs szükség az fsck futtatására minden rendszerösszeomlás után, még körültekintőbben kell bánnunk az elveszett állományokkal. Egy leválasztott fájl valójában még nem törlődik, amíg az azt megnyitó folyamatok futása be nem fejeződik. Ha a rendszer összeomlik, mielőtt a törlési művelet befejeződne, a jegyzőkönyv összefüggő fájlrendszert fog létrehozni, viszont valamennyi lemezhelyet továbbra is lefoglal a fájl számára. Mivel a fájl nem szerepel a könyvtárfában, a blokkok visszaállítására nincs többé lehetőség.
A fenti hiba legkönnyebb kiküszöbölése, ha a fájlt egy különleges könyvtárba helyezzük. A ReiserFS könyvtárak nagyon gyorsak, és nem kell különösebben aggódni a zárolások miatt sem, ha a fájlnevek nem ütköznek egymással. Összeomlást követően a könyvtárt kiolvassuk, és befejezzük a fájltörléseket az összes megmaradt objektumhoz. A különleges könyvtárnak valójában fájlnevekre sincs szüksége, csak a fájlok megtalálásához szükséges adatokra. Ez a javítás jelenleg ugyan nem található meg a hivatalos ReiserFS kiadásokban, de a helyzet hamarosan megváltozik.

A felhasználói terület műveletsorai

A felhasználók időről időre szeretnék tudni az alkalmazásprogramozási felület által a felhasználói területre kivitt műveletsorok változatszámát. A ReiserFS jegyzőkönyvezési rétegét befejezett műveletek támogatására tervezték, ezek általában nagyon hamar végrehajtódnak, így nem működne jól egy általános műveletsorkezelő szerepében.
Az azonban nem lenne jó ötlet, hogy atomi írásokat engedélyezzünk a felhasználói terület számára, és ezzel nagyobb ellenőrzést tegyünk lehetővé a műveletek csoportosítására. Ilyen módon ugyanis egy alkalmazás kérhetné egy 64 kB-os fájl létrehozását egy megadott könyvtárban, és mindezt atomi műveletként kezelhetné. Mindeddig elég csekély tervezési munka folyt ezen a területen.

VM beillesztése

Ahogy fogyatkozik a rendszer memóriája, a rendszermagnak ki kell ürítenie a piszkos átmeneti tárak adatait a lemezre, ezáltal memórialapok szabadíthatók fel. Csakhogy a még végre nem hajtott műveletsorok által lekötött memória nem szabadítható fel a végrehajtásig, így a VM gyakorlatilag tehetetlen a fájlrendszer segítsége nélkül. Biztosak szeretnénk lenni abban, hogy a jegyzőkönyvezés a lekötött pufferek miatt nem foglalja el a rendszer memóriájának túlságosan nagy hányadát.
Együtt fogunk dolgozni a VM fejlesztőivel, hogy megfelelően sikerüljön jelezni a fájlrendszernek a memóriahiányt. Az API jelenleg

nincs kőbe vésve, de a felhasználók a jelek szerint hajlanak egy, a laphoz tartozó tárürítő visszahívásos függvény támogatására, illetve egy általános memóriaigény-bejegyző rendszer használatára. Egyelőre nem tudni, hogy mindebből mi valósul meg a 2.4-es rendszermagban, és mi marad a 2.5-ös változatra.

A ReiserFS és az LVM

Az LVM nagy csokor szolgáltatással bővíti a Linuxot, ezek egyike a csak olvasható pillanatfelvételek készítése egy meghajtóról. Ennek elkészítése roppant gyorsan zajlik. Egy *másolás íráskor* eljárás pedig gondoskodik a pillanatfelvétel frissítéséről, ahogy az eredeti meghajtót módosítjuk. Ezzel a módszerrel gyakorlatilag bármely fájlrendszeren a legtöbb programhoz állandó elérésű, összefüggő biztonsági mentések készíthetők.

A jegyzőkönyvezett fájlrendszer ezt megnehezíti egy kicsit. Amikor a syncet meghívjuk ReiserFS-en, csak kiírjuk a metaadatok változását a jegyzőkönyvbe, azzal a tudattal, hogy egy esetleges rendszerösszeomlást követően a visszajátszás mindent megfelelő állapotba állít vissza. Egy csak olvasható LVM pillanatfelvétel esetében a jegyzőkönyv visszajátszása nem lehetséges. Ehelyett új, általános hívásokat alkalmazhatunk, ezek mindent kiürítenek a lemezre, és szüneteltetik a fájlrendszer újabb módosításait. Amíg a szünet tart, az LVM frissíti a pillanatfelvételt, így az a jegyzőkönyv visszajátszása nélkül is összefüggő lesz. Amikor az LVM végzett, megszüntetjük a fájlrendszer zárolását, és ezután megszokott módon folytatódnak az írási műveletek.

Minden fájlrendszerrel dolgozó műveletnek tudni kell várnia a jegyzőkönyvre, ennek megvalósítása a ReiserFS esetében könnyen ment. Az LVM 0.9 és a ReiserFS 3.6.18 képesek erre a szolgáltatásra, de egyelőre nem tudni, az általános hívások mikor kerülnek be a rendszermagba. Ettől függetlenül a hiányzó részleteket pótló javítások elérhetők lesznek mind a ReiserFS, mind az LVM honlapján.

Az LVM egy másik szolgáltatása az egyik meghajtó tartalmának áthelyezése másikra. Ha olyan területet találunk a lemezen, mely az átlagosnál nagyobb forgalmat bonyolít le, a kérdéses blokkokat áthelyezhetjük egy másik, gyorsabb meghajtóra. Tulajdonképpen a teljes jegyzőkönyvezési területet át lehetne helyezni egy gyorsabb meghajtóra, ezzel csökkentve a fejek terhelését és a fejléptetések számát. Így ténylegesen javítható a sok jegyzőkönyvezést igénylő alkalmazások teljesítménye.

Programbeli RAID

A 2.2-es rendszermagokban a RAID programja lekötött puffereket is kiírhat a lemezre, ezzel megszegheti az írások sorrendjére vonatkozó, a dolgok összefüggő állapotban való tartásához szükséges szabályokat. Csak a meghajtók csíkokra osztása és egybefogása lenne teljesen biztonságos, a tükrözés csak addig megbízható, amíg nem használjuk az üzem alatti helyreállító (on-line rebuild) programkódot. A 2.4-es rendszermagokban minden programból megvalósított RAID-szint megfelelően együttműködik a jegyzőkönyvezett fájlrendszerekkel.

ReiserFS és NFS

A ReiserFS gondokkal küszködik az NFS-támogatást illetően, mivel 64 bitnyi adatot igényel egy objektum azonosításához a fában, az NFS viszont a fájlleírókat azok 32 bites azonosítója alapján próbálja meg azonosítani. A jó hír az, hogy az NFS fájlkezelőnek elegendő helye van a ReiserFS által igényelt többletadat tárolásához. Néhány rendszermagfejlesztő készített olyan API-felületeket, amelyek a fájlrendszer számára lehetővé teszik az ellenőrzést a fájlkezelők egy része felett. A cikk megjelenésének időpontjában valószínűleg már vannak nyilvános javítások, amelyek megfelelő NFS-támogatást adnak a ReiserFS-hez.

Írások gyorstárazása

A teljesítmény növelésére egyes újabb lemezmeghajtók alapértelmezett állapotban visszaírásos gyorstárazást használnak. Ez azt jelenti, hogy a meghajtó a műveletet még azelőtt befejezettnek jelzi, hogy az adat ténylegesen az adathordozóra került volna. A blokk továbbra is a meghajtó gyorstárában van, ahol megváltozhat az írások sorrendje. Ebben az esetben a metaadatok változásai még azelőtt kiíródnak, hogy jegyzőkönyv végrehajtaná a blokkműveleteket, és ez hibához vezethet, ha időközben megszűnik a rendszer tápellátása. Nagyon fontos, hogy azoknál az eszközöknél, amelyek nincsenek ilyen hibaesetekre felkészítve, a visszaírásos gyorstárazást letiltsuk. Egyes RAID-vezérlőkártyáknak saját akkumulátorral felszerelt visszaírásos gyorstáruk van, ezek áramkimaradás esetén sem vesztik el tartalmukat. Használatuk ugyan biztonságos, de rendszeresen ellenőrizni kell az akkumulátorok állapotát. Megdöbbentő teljesítménynövekedést lehet tapasztalni a hasonló gyorstárak használatával, főleg jegyzőkönyv-igényes alkalmazásoknál, például a levelezőkiszolgálóknál.

Levelezőkiszolgálók

A levelezőkiszolgálók a legrosszabbak a jegyzőkönyvezett fájlrendszerek számára, mivel minden egyes fájlművelet befejeztéről meg kell bizonyosodniuk. Az fsyncet használják, esetleg egyéb trükkök egész sorát, hogy elkerüljék az üzenetek elvesztését egy esetleges rendszerösszeomláskor. Emiatt a fájlrendszernek gyakran rendkívül kicsi műveletsorokat is le kell zárnia.

A levelezőkiszolgálóknak valamilyen gyors módszerrel a lemezre kell írnunk az új fájlokat, az adatok jegyzőkönyvezése ebben segítségükre lehet. Az fsync hívás alatt jegyzőkönyvezzük az adatblokkokat és a fájlnak a fába helyezéséhez szükséges metaadatokat, ezzel egy nagyméretű, soros írást hozva létre. Ha az írásra kerülő fájl csak egy átmeneti várakozási sorba kerül, lehet, hogy soha nem írjuk ki a lemezre. Gyors, csak jegyzőkönyvezésre használt meghajtóval együtt alkalmazva az adat-jegyzőkönyvezés jelentős teljesítménynövekedést hozhat. Jelenleg a ReiserFS 2.4-es programkódja nem támogatja az adat-jegyzőkönyvezést. A 2.2-es változatú rendszermagokhoz készült kiadásban van adat-jegyzőkönyvezés, de ha át akarjuk ültetni a 2.4-es rendszermagra, akkor az eltérő gyorstárazó rendszere miatt a kódot át kell alakítani.

A versenytársak

Az új linuxos fájlrendszerek megjelenése rendkívül fontos. A rendszergazdák megválaszthatják azt a terméket, amely a legjobban megfelel az általuk futtatott alkalmazáshoz, a programozók pedig mások eredményeivel hasonlíthatják össze saját döntéseiket. Egészében véve a Linux csak nyerhet azon, hogy a közösség tagjai kiválasztják és használják az egyes fájlrendszerek leghasznosabb szolgáltatásait.

Chris Mason
(mason@suse.com) Mielőtt belefogott a ReiserFS fejlesztési tervezetébe rendszergazdaként tevékenykedett. Jelenleg teljes munkaidőben rochesteri otthonából (New York) dolgozik a SuSE-nak.

Kapcsolódó cím

➲ http://www.reiserfs.org/ további adatokat találhatunk a ReiserFS telepítésével és használatával kapcsolatban, illetve a résztvevő programozókról is. Ezenkívül olvashatunk a lemezformátumról, és a leggyakrabban felmerülő általános kérdésekre is választ lelhetünk.