

Váltás PostgreSQL adatbázisrendszerre

Néhány hasznos tanács arra vonatkozóan, hogy miként cserélhetjük le meglévő Microsoft Access adatbázisrendszerünket.

Egyre több vállalat kezd nyílt forrású üzleti rendszerekkel foglalkozni, sokan közülük teljes rendszert igyekeznek kiépíteni, amely a vásárlók számára hozzáférhető webes felülettel az alapként szolgáló adatbázisig terjed. Legtöbb esetben a Linux-PHP-Apache hármast, illetve egy nyílt forrású adatbázist (MySQL-t vagy PostgreSQL-t) foglal magában. A PostgreSQL népszerűsége főleg az utóbbi időben növekedett meg, hiszen mára a program elérte azt a minőséget, amelynél már megbízhatónak és jól használhatónak mondható. Több nagyvállalat biztosít állandóan elérhető támogatást, közöttük a RedHat is. Írásomban azt szeretném megmutatni, hogy mire számíthatunk, ha a Microsoft Accessről nyílt forrású adatbázisrendszerre (itt és most a BSD-típusú felhasználási szerződéssel rendelkező PostgreSQL-re) térünk át.

Michael Calabrese, a Bike Friday nevű kerékpárgyártó cég adatrendszerének felelőse nemrégiben a változás mellett döntött. A Bike Friday egy villámgyorsan növekedő, túra- és hegyi kerékpárok gyártásával foglalkozó cég, székhelyük az Oregon állambeli Eugene-ben található. A vállalatnál PostgreSQL adatbázisban rögzítik az eladásokkal, a gyártással és a vásárlói támogatással kapcsolatos összes adatot. Calabrese mostanában épp a cég e-üzleti rendszerét szándékozik ingyenes programokra lecserélni: Linuxra, Apache-ra és PostgreSQL-re. Egyelőre azonban úgy tervezi, hogy megtartja a munkaállomásokon használt Microsoft Access 97-et annak érdekében, hogy az adatbázis lecserélése miatt szükséges szünet a lehető legrövidebb ideig tartson. Calabrese szerint: „Ha a munkaállomásokon használt kezelőprogramokat nem akarjuk megtartani, egyszerű a dolgunk: csupán le kell futtatni az átalakítóprogramokat és elkezdhetjük megírni az új kezelőrendszert. Ha a kezelőrendszer az Access, amelyet továbbra is szeretnénk használni a PostgreSQL adatbázissal, akkor a fejlődés új irányait úgy jelöltük ki, hogy az induláshoz nincs szükség az egész rendszer átalakítására. A kezelőfelület befagyasztása után

nyugodtan számoljunk egy évet az átalakításra. Ha a változtatásokat fokozatosan vezetjük be, megmarad a választási lehetőség, hogy egy-egy új lehetőséget Accessben vagy PostgreSQL-ben valósítsunk meg”.

A változtatás eszközei

Amennyiben a Microsoft Open DataBase Connectivity (ODBC) meghajtókat betöltjük a PostgreSQL-sablonadatbázisba, máris megtettük az első lépést az Access és a PostgreSQL összeházasításához. Az együttműködéshez az alapvető átalakító eszközök mellett (lásd a *További érdekességek* című részt) néha további ODBC kiszolgálóoldali szolgáltatásokra is szüksége van. Ezeket az *src/interfaces/odbc/odbc.sql* fájlban találhatjuk meg. A PostgreSQL felületfüggetlen 4-es típusú Java adatbázis-kapcsolati felület (JDBC) meghajtót is tartalmaz. Továbbá a C számára készült beágyazott felület (ECPG) is a PostgreSQL részét képezi. A telepítés végeztével Calabrese adatösszesítő eszközöket választott, például a pgAccess – ez Windows és Unix változatban, illetve az exSQL 3.1-es változatában egyaránt hozzáférhető.

Miután a meglévő adatbázisokról a rendelkezésre álló eszközökkel (vagy a *pg_dumpall* segédprogrammal, vagy pedig a *pg_dump* és a *pg_dumpaccount.s* együttes használatával) biztonsági mentést készítettünk és a telepítőt (*Installer*) is lefuttattuk, az adatok átalakításának első lépéseként az Accessben használatos, azonban a PostgreSQL-ben nem megengedett fájlnevek levadászása következett. Az Access meglehetősen szabadelvű, hiszen a fájlnevekben számos olyan karakter használatát lehetővé teszi, amelyeket más adatbázisrendszerek (Oracle, Sybase, PostgreSQL stb.) nem ismernek fel. Így a Bike Friday adatbázisában szereplő nevek jó részét a PostgreSQL által is kezelhető formájúra kellett alakítani, például az *Order Detail* táblázatból *Order_Detail*, a *Shipped?* mezőnevekből pedig *Shipped* vagy *ShippedYN* lett.



Az alapvető átalakítóeszközök minden meg nem engedett karaktert eltávolítanak. Ez súlyos gondokat okozhat, hiszen a mit sem sejtő munkaállomási kezelőprogramok (nem értve a helyzetet) minden további nélkül megszákíthatják az adatbázissal való kapcsolatot.

Calabrese azt javasolja, hogy ha a kezelőprogramokat megtartjuk, ne írjuk át az adatbázis neveit, vagy pedig az adatbázist és a kezelőfelületeket is párhuzamosan módosítsuk. Ő maga ezt úgy oldotta meg, hogy az adatbázisban és a kezelőprogramokban saját kezűleg, egyesével írta át a kérdéses karaktereket.

1. lista Meglehetősen pazarló lekérdezés

```
SELECT Orders.SalesRepID
Bikes_ColorsAvailable.Color,
OrderDetails.BuildABikeID, Orders.OrderDate,
BuildABike.ColorID,
Bikes_BasicFrameTypes.FrameName,
Reps.Rep, BuildABike.Frame, Orders.CustID,
Orders.OrderID, OrderDetails.PartID,
Orders.ShipDate,
BuildABike.BikeState
FROM (Reps RIGHT JOIN Orders ON
Reps.RepID = Orders.SalesRepID) INNER JOIN
(((Bikes_BasicFrameTypes INNER JOIN
(OrderDetails INNER JOIN BuildABike ON
OrderDetails.BuildABikeID =
BuildABike.BuildABikeID) ON
Bikes_BasicFrameTypes.FrameTypeID =
BuildABike.FrameTypeID) INNER JOIN
Bikes_ColorsAvailable ON BuildABike.ColorID =
Bikes_ColorsAvailable.ColorID)
INNER JOIN Contacts_CurrentFrame_BABID ON
BuildABike.BuildABikeID =
Contacts_CurrentFrame_BABID.BuildABikeID)
ON Orders.OrderID = OrderDetails.OrderID
WHERE (((OrderDetails.PartID)=6502))
ORDER BY Orders.OrderDate DESC;
```

2. lista Ugyanaz a lekérdezés, de jóval gyorsabban

```
SELECT salesrepid, color, od.buildabikeid, o.orderdate,
bab.colorid, framename, reps.rep, bab.frame,
o.custid, o.orderid, partid, o.shipdate, bab.bikestate
FROM
reps, bikes_basicframetypes b_bft, orders as o,
orderdetails as od,
bikes_colorsavailable as b_co_a,
buildabike as bab
WHERE repid = salesrepid
AND od.buildabikeid = bab.buildabikeid
AND b_bft.frametypeid = bab.frametypeid
AND b_co_a.colorid = bab.colorid
AND o.orderid = od.orderid
AND bab.custid=[Forms]![Customers/Contacts]![ID]
AND entrydate = (
SELECT MAX(entrydate)
FROM buildabike as b2
WHERE bab.frame = b2.frame
);
```

Ez rendben is volt, hiszen a távolabbi tervek között a kezelőprogramok lecserélése is szerepelt. Ami a legfontosabb: a munka ezen szakaszában nagyon fontos, hogy folyamatosan ellenőrizzük, vajon minden működik-e. Ha a meg nem engedett karaktereket kiküszöböltük, az adatok máris készen állnak az átalakításra.

Az adatok átalakítása

Ha azt tervezzük, hogy az Access megtartva kezelőfelületnek, első lépésként csak az adatokat ültetjük át PostgreSQL alá, a pgAdmint rendkívül hasznos eszköznek fogjuk találni. Calabrese az exSQL módosított változatát is felhasználta annak meghatározásához, hogy az Access és a PostgreSQL hogyan kezeli a táblázatok közötti kapcsolatokat. A http://www.geocities.com/musica_6898/postgresaccess_home.html címen elérhető honlapján a nyilvánosság elé tárt változat egy héjprogramot futtat le, mely számos esetben módosítja a mezőtípus-átalakítás menetét (például az Access pénznezé- lését is). A Bike Friday Access kezelőfelülete a PostgreSQL számszerű decimális mezőit szövegmezőként értelmezte. Annak érdekében, hogy az Access ezeket helyesen kezelje, Calabrese a mezőket *Float4*-típusúvá változtatta (a PostgreSQL így nevezi a négybájtos lebegőpontos számokat).

A kezelőfelület kipróbálása

Száznál is több táblázatával a Bike Friday kezelőfelülete meglehetősen összetett. A felhasználó szemzőgéből nézve a Bike Friday több mint nyolcvan képernyőt használ a megrendelések beírásához, az alkatrésztáblázat megtekintéséhez, a gyártás ütemezéséhez és a raktárkészlet ellenőrzéséhez. Ezért Calabrese-nek biztosnak kellett lennie abban, hogy a rendszert több tucat felhasználó sem fagyaszthatja le. A kipróbálás jó néhány hétig tartott, közben az SQL-lekérdezéseket is szükség szerint módosítani kellett oly módon, hogy az Access-oldalról vagy (ha ez nem volt megoldható) az adatbázis oldalán újraírták őket, egészen addig, amíg elfogadható sebességet kaptak. Az 1. és 2. listában egy jellegzetes és egy gyorsított lekérdezést láthatunk.

A PostgreSQL-lekérdezések hatékonyabbá tétele általában a Create index, vacuum, vacuum analyze, cluster és

explain parancsok használatával történik. Calabrese azonban figyelmeztet, hogy az Access alapértelmezetten a lekérdezéseket nem közvetlenül továbbítja, hanem mindig az általa leghatékonyabbnak ítélt formára alakítja őket. Calabrese a program okoskodását közvetlen lekérdezés használatával kerülte meg, amely közölte az Access-szel, hogy hozzá ne nyúljon a lekérdezéshez, egyszerűen csak továbbítsa a kiszolgáló felé. A Bike Friday PostgreSQL-adatbázisának egyszerűsítése során Calabrese úgy ért el sebességnövekedést, hogy a lekérdezésekben az adatbázisból kisebb, pontosabban körülhatárolt adatokat vett ki. Százezer rendelési adat egyidejű lekérdezése

helyett a lekérdezést úgy alakította ki, hogy az adatbázisnak csak 2000 adatra kellett figyelnie. „Az Access buta – mondja Calabrese –, az összes rekordot kézbe veszi és minden egyes alkalommal az összeset átnézi. Ez rendkívül pazarló módszer. Jelenleg harminc alkalmazottunk van, és ha történetesen minden számítógéppel egyszerre próbálnák meg elérni az adatbázist, ez igen hamar nagyon komoly sebességsökkenést okozna.”

A PostgreSQL ellenőrzése

A változtatás következő lépése a lekérdezések hibaellenőrzése, ahol mindjárt két út közül választhatunk. Az egyik a PostgreSQL ODBC-meghajtójához tartozó ellenőrzőeszközök használata. A meghajtóval készíthetünk egy naplófájlt, és amikor az Access SQL-parancsot küld, a PostgreSQL azt azonnal bevezeti a naplóba, mely a C meghajtó gyökérvénytárban található. Ezzel elcsíphetjük az Access olyan ügyetlen próbálkozásait, amikor mondjuk százezer sort próbál egyszerre behívni. Ebben az esetben például a lekérdezést ezer kisebbre bonthatjuk szét. Ez a napló igazából egy nyomkövetés, mely segítségével gyorsan kiszűrhetjük, ha valami hiba lépett fel, mint ahogy itt is történt.

```
conn=86311032, query=' '
```

```
CONN ERROR: func=SQLDriverConnect,
desc='Error from CC_Connect',
errnum=105, errmsg='The database does not
exist on the server or user authentication
failed.'
```

Azt is megtehetjük, hogyha az Access lekérdezést küld, és a rendszer leáll, a kiszolgálóoldal hibakezelési szintjét (*Debug level*) átállítva a kérelmet csak azért is kiolvassuk. A finomhangolás lényege, hogy minden egyes képernyőn végig kell haladnunk és a kérelmek egyszerűsítésével, összevonásával gyorsítanunk kell őket. Az SQL-t jól ismerők tudják, hogy a rendszer milyen összetett, így elmondhatjuk, hogy fáradságos munka elébe nézünk. Ha azonban a fejlesztésnek ezen a pontján elvégezzük a megfelelő ellenőrzéseket, rengeteg későbbi fejfájástól kímélhetjük meg magunkat.

Mielőtt a rendszer működését visszaállítanánk, a kipróbálás következik. Calabrese folyamatosan figyelte a Bike Friday adatbázisrendszerét, miközben az irodákban már használták a rendszert. „Nemcsak azt kell kipróbálnunk, hogy akadnak-e a kezelőfelületnek hibái, hanem azt is, hogy mekkorára kell a kiszolgálót terveznünk” – mondja Calabrese. Írt egy lekérdező héjprogramot is, amely a három fő gondot okozó részegység (processzor, merevlemez, hálózat) terheltségét kíséri figyelemmel.

Calabrese programja a processzor kihasználtságát aszerint ellenőrizte, hogy a terhelés hány másodpercig maradt 100, 50 és 0 százalékos. A lemez adatátvitelének értékelését úgy végezte, hogy hány olvasási és írási művelet zajlik éppen, illetve mérte az ezek során átvitt kilobájtokat is. A hálózat terhelését a másodpercenkénti csomagszámával és a másodpercenként átvitt bajtok számával írta le. Calabrese azt is javasolja, hogy a lezárt hálózati szakaszban végezzünk árasztásos pingelést (ping -F), így meghatározhatjuk, hogy a kiszolgáló mekkora terhelésnél akad meg. A memóriával egyszerű a helyzet: minél több van belőle, a PostgreSQL annál többet használ föl, és így annál gyorsabb lesz az adatbázis működése.

Természetesen az adatbázis sebességéről a felhasználók véleménye árulkodik leginkább. A lépésenkénti apró várakozási idők a valóságban hatalmas késésekké adódhatnak össze.

Minden vállalatnál, szervezetnél kialakul egy vélemény arról, mi számít lassúnak és mi elfogadhatóan gyorsnak. Így a rendszer főpróbája mindenképpen az lesz, amikor a felhasználók pár óras használat után kimondják a végítéletet: „Csigalassú” vagy „Hm, nem is olyan rossz”.

Végül, miután a rendszer megfelelőnek találtottuk, mi pedig a kezelőfelület minden hibáját kijavítottuk, máris nekiláthatunk egy nyílt forrású alapokra építkező e-üzleti rendszer kiépítésének.



Chris Volpe

(chris@macnet2.com)

New Hampshire-ben él és technológiai leírásokat készít.

mascTovábbi érdekességek

Bruce Momjian: PostgreSQL: Introduction and Concepts (ISBN: 0-201-70331-9, 44,95 dollár, 544 oldal)

➔ <http://www.ca.postgresql.org/docs/awbook.html> címen érhető el.

A gép és az alkatrészek, valamint a PostgreSQL összehangolásáról olvassuk el *Momjian* írását „PostgreSQL Performance Tuning” címmel

➔ <http://www.linuxjournal.com/lj-issues/issue88/4791.html>

F. Scott Barker: Microsoft Access 2000 Power Programming (ISBN: 0-672-31506-8, 49,99 dollár, 1332 oldal, CD-ROM)

PostgreSQL 7.1 kézikönyv

➔ <http://www.ca.postgresql.org/users-lounge/docs/7.1/reference> Postgres GYK

➔ <http://www.ca.postgresql.org/users-lounge/docs/#7.1> Postgres/Access GYK

➔ <http://joelburton.com/resources/pgaccess>

PostgreSQL-leírás és Data Migration Tools

➔ <http://postgresql.crimelabs.net/users-lounge/docs>

Migration Tools: a csomagban a *pgAdmin* (grafikus PostgreSQL-vezérlőfelület), a *phpPgAdmin* (webalapú eszköz a pgAdminhoz hasonló feladatokra) és a PsqLODBC Windows-meghajtó található. Ez utóbbi lehetővé teszi, hogy a PostgreSQL-adatbázist az ODBC-meghajtókon keresztül elérő Windows-alkalmazásokat írjunk.

Még egy érdekesség: az exSQL új, nyilvános változata is elérhető. Az exSQL nagyszerű PostgreSQL-átalakító eszköz. Az új változat, amely az indexeket és az idegen kulcsokat megbízhatóbban kezeli, a

➔ http://www.geocities.com/musica_6898/postgresaccess

_home.html címről tölthető le. A PostgreSQL az idegen kulcsokat a táblázatok összekapcsolására és kapcsolataik kódolására használja. A *Michael Calabrese* által írt exSQL-változat módosítja azokat a szabályokat, melyek meghatározzák, hogy az egyes Access-mezőkből milyen PostgreSQL-mezőtípusok készíthetők. További egyszerűsítések és hibajavítások mellett ez a változat tartalmaz egy parancsfájlt, amely az Access egyik hibáját küszöböli ki: alapértelmezés szerint a program szöveggé alakítja a pénzmezőket. A parancsfájl felülbírálja ezt az alapértelmezést, így futás közben ebből hibák adódhatnak.