A felhasználói viselkedés vizsgálata (3. rész)

A sorozat előző részében megvizsgáltuk az adatgyűjtési módszereket. Most már ideje használni is ezeket, hogy a legyen mit elemeznie a Nagy Testvérnek.

A legjobb megoldás

A legfrissebb adatok alapján az internet legelterjedtebb webkiszolgálója az ***Apache***, a maga közel 67%-os részesedésével. Éppen ezért döntöttem úgy, hogy a modellezést ennek a programnak az eseménynaplói alapján fogom bemutatni. A választást az is megerősíti, hogy a témával kapcsolatban a ***Google*** közel 160 olyan szoftvert talált, amelyek az eseménynapló valamiféle feldolgozásával, statisztikák készítésével és modellezéssel foglalkozik.
Az ***Apache*** előnye, hogy teljesen nyílt forrású, rengeteg platformon és operációs rendszeren fut (***Unix***, ***Linux***, ***BSD***, ***Microsoft Windows***, ***Novell Netware***), széleskörűen támogatott, és rengeteg kiegészítő modul érhető el hozzá. Beállítása egyszerű, egészen szélsőséges terhelési viszonyok között is használható, kiegészítő modul használatával pedig akár elosztottan is futtatható. Az internetes közösség folyamatosan jelentkezik újabb és újabb modulokkal. (Van például adatbázisba naplózó modul is.)

Apache naplózás

Az ***Apache*** naplózó modulja a ***mod_log_config*** (➲ ***http://httpd.apache.org/docs/mod/mod_log_config.html***), amely alapból feltelepül minden fent említett rendszeren. Az eseménynaplóba kerülő bejegyzések részletessége igen változatos lehet, a modellezés leghatékonyabb támogatásához azonban érdemes a legrészletesebben naplózni.
Az ***Apache*** webszerveren a `LogFormat` direktíva segítségével adható meg, hogy milyen adatok és milyen sorrendben kerüljenek az eseménynapló egy bejegyzésébe. A lehetséges direktíva értékek:

- `%a`: az ügyfél IP címe,
- `%A`: a kiszolgáló (helyi) IP cím,
- `%b`: az elküldött adat nagysága (Common Log Format formátum),
- `%c`: a kapcsolat státusza a kérés teljesítése után,
- `%f`: a lekért erőforrás (fájl) neve a helyi fájlrendszeren,
- `%h`: az ügyfél neve,
- `%H`: a kérés protokollja,
- `%l`: a távoli eseménynapló neve (általában üres),
- `%m`: a kérés metódusa,
- `%q`: a lekérdezést tartalmazó karakterlánc (***query string***), kiegészítve egy kérdőjellel ha létezik, egyébként üres, (például: `?action=register&step=3&sess=12345678`)
- `%r`: a kérés első sora, például: "`GET /modul.php?action=register ➥&step=3&sess=12345678 HTTP/1.0`",
- `%s`: a kérés kiszolgálásának státusza (HTTP protokoll státusz kód),
- `%t`: időbélyeg (***Common Log Format***),
- `%T`: a kérés kiszolgálására fordított idő, másodpercben,
- `%u`: távoli felhasználó (azonosításból származik),
- `%U`: a kért URI a ***query string*** nélkül, például: "`/modul.php`",
- `%v`: a kiszolgáló szerver neve.

Amint azt már a sorozat korábbi részeiben említettem a fenti adatok alapján nem lehet egyértelműen elkülöníteni az egyes felhasználókat, főleg ha azok azonos proxy vagy tűzfal mögül látogatják az elemezni kívánt portált.
A modul lehetőséget nyújt az ügyfél felől érkező kérésekben található változók értékének eseménynaplóba írására is. Ennek segítségével tudjuk naplózni az ügyfél böngészőjének típusát és a „előző oldal" vagy küldő oldal (***referer***) értékét is. Ehhez a következő direktívákat kell használnunk:

- `%{Referer}`: a referer értéke,
- `%{User-Agent}`: a böngésző típusa.

A direktíva-értékek tetszőlegesen kombinálhatóak.
Íme néhány példa:

- Általános: "`%h %l %u %t \"%r\" %>s %b`"

Példa:

```
217.20.135.85 - - [08/Apr/2004:18:06:47 +0200]
➥"GET / HTTP/1.0" 200 16897
```

- Összetett: "`%h %l %u %t \"%r\" %>s %b ➥\"%{Referer}i\" \"%{User-agent}i\"`",

Példa:

```
62.165.209.144 - - [08/Apr/2004:18:30:55 +0200]
➥"GET / HTTP/1.1" 200 16937
```

```
"http://www.oktaton.hu/index.php?inc=portal"
➥"Mozilla/4.0 (compatible; MSIE 6.0; Windows
➥NT 5.1)"
```

- Egyedi: "%h %l %u %t \"%r\" %>s %b
  ➥\"%{Referer}i\" \"%{User-Agent}i\"
  ➥\"%{Cookie}n\" %T %v \"%U\" %c %f",

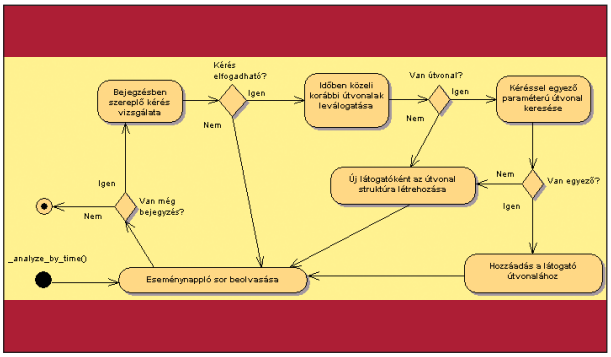
Példa:

```
42.165.229.144 - - [08/Apr/2004:18:30:55 +0200]
➥"GET / HTTP/1.1" 200 16937 "http://www.oktaton.hu/
➥index.php?inc=portal" "Mozilla/4.0 (compatible;
➥MSIE 6.0; Windows NT 5.1)"
➥"42.165.229.144.38811081441855459" 0
➥www.jegyzet.com "/" -
➥/srv/www/www.jegyzet.com/index.php
```

Látható, hogy a felhasználói viselkedés modellezésének pontossága jelentősen növelhető a ***referer*** és a böngésző típusára vonatkozó adatok felhasználásával. Az egyes weblapok (különböző tartalom szerinti) elválasztását a %q és %U direktívák értékének felhasználása könnyíti meg, ugyanis azonos %U értékek esetén a %q értékek különbözősége más-más weboldalt jelent. Ugyanakkor elmondható, hogy az általánosan használt eseménynapló bejegyzés formátumok a modellezés szempontjából felesleges adatokat is tartalmaznak (például: elküldött adat mérete, távoli eseménynapló neve), azonban az általánosan szokásos biztonsági naplózás miatt ezeket általában nem szokás eltávolítani.
A legjelentősebb probléma a fenti eseménynapló szerkezetekkel, hogy nem tartalmaznak a felhasználókat egyértelműen megkülönböztető bejegyzéseket: a %u paraméter értéke általában üres, mert a felhasználói azonosítást a legtöbb helyen nem a webszerver végzi, hanem a portál alkalmazása. Gyakori, hogy egyes erőforrásokat a webszerver által biztosított hitelesítési mechanizmussal (is) védenek, jellemző azonban, hogy az egyes felhasználók ugyanazt a felhasználónév és jelszó párost használják, azaz az így keletkező adat nem használható a felhasználók megkülönböztetésére.

Egyedi felhasználói azonosító

Szerencsére van megoldás, ugyanis az Apache webszerverhez létezik egy modul, amely segítségével minden egyes felhasználóhoz egyedi azonosító rendelhető.
Ez a ***mod_usertrack*** modul (➲ ***http://httpd.apache.org/docs/mod/mod_usertrack.html***).
A modul segítségével minden felhasználóhoz egy munkamenet azonosító rendelhető, függetlenül a kiszolgált, futtatott portál szoftverétől.
A modul működése igen egyszerű: a webszerver minden kérés esetén megkapja a ***HTTP*** fejlécekben a korábban az adott weboldalra juttatott sütiket. A modul megvizsgálja, hogy létezik-e a munkamenet azonosító süti, és ha úgy találja, hogy az létezik, és még ráadásul érvényes is, akkor egyszerűen visszajuttatja a klienshez. A visszaküldést a webkiszolgáló automatikusan végzi a HTTP fejlécben, a süti kezelés módszereinek megfelelően.
Amennyiben nem talál munkamenet azonosító sütit, akkor a modul automatikusan generál egyet. Minden munkamenet azonosítónak egyedinek kell lennie, ami egy nagy (sok ezer lekérés percenként) forgalmú portál esetén nem olyan triviális feladat, mert a szoftveres véletlenszám generátorok hajlamosak ismételni (szekvenciába esni). A modul az egyedi munkamenet azonosítót a következőképpen konstruálja meg:



1. ábra A kérések idő alapú elkülönítésének algoritmusa

- prefix karakterlánc (tetszés szerint beállítható),
- az ügyfél IP címe, vagy a helyi gép (***host***) neve,
- a kérést kiszolgáló folyamat (***process***) azonosítója,
- a kérés időbélyege másodperc pontossággal,
- a kérés időbélyege mikromásodperc pontossággal.

A modul a fenti adatokat összefűzi, és az így előálló karakterlánc lesz az egyedi munkamenet azonosító.
A modul megbízhatósága általában véve a szerver oldali adatgyűjtésnél ismertetett problémák miatt nem 100%-os. A kliens oldalon a sütik elfogadásának letiltásával a felhasználó minden egyes letöltésekor új munkamenet azonosító keletkezik, hiszen a böngésző nem tárolhatja a sütiket, azaz nincs hol tárolni az azonosítót. Az újabb böngészőkben lehetőség van arra, hogy a munkamenet azonosító sütik ne kerüljenek letiltásra. Ezt a modul megfelelő konfigurációjával segíthetjük elő.
A modul a következő beállításokkal vezérelhető (a beállításokat az ***Apache*** kiszolgáló beállítási fájljában kell elhelyezni):

- `CookieDomain`: a süti érvényessége, egyes böngésző beállítások esetén a böngésző csak az éppen használt szervertől fogad el sütit, ezért explicit módon meg kell adni, hogy melyik tartományhoz (***domain***) tartozik a süti.
- `CookieExpires`: ezzel az értékkel tulajdonképpen azt mondjuk meg, hogy mennyi ideig lehet a felhasználó egy látogatáson belül inaktív (nem kér le újabb oldalt a szervertől). Ha letelik a megadott idő, akkor a következő oldallekérés esetén a felhasználót új látogatóként kezeljük. Könnyen elképzelhető, hogy közben másvalaki ült le a géphez.
- `CookieName`: opcionális, a süti elnevezése.
- `CookiePrefix`: az azonosító elejére kerülő tetszőleges karakterlánc, opcionális.
- `CookieStyle`: a létező süti szabványoknak megfelelő típusú süti küldés beállítása (nagyon régi böngészők használata esetén lehet szükséges a módosítása).

- `CookieTracking`: mivel a modul betöltése esetén nem kerül automatikusan használatra, ezért explicit módon kell bekapcsolni.

Egy tipikus konfigurációs példa:

```
CookieDomain ".jegyzet.com"
CookieExpires "30 minutes"
CookieName "www-jegyzet-com-user-tracking"
CookieStyle "Cookie"
CookieTracking "on"
```

A modul beállítási lehetőségei közül a legfontosabb a `CookieExpires`, mert ennek értéke akár jelentősen is befolyásolhatja a felhasználói modellezést. A paraméter értékét minden esetben az adott portál felhasználási viszonyaihoz kell megállapítani figyelembe véve a látogatók áramlását és szokásait.
A modul egyetlen feladata tehát az eseménynapló bejegyzések kiegészítése egy munkamenet azonosítóval, amely azonosító a feldolgozás során a felhasználók szétválogatását, megkülönböztetését szolgálja.

Felhasználók elkülönítése

Ha valamilyen oknál fogva nem tudjuk használni a ***mod_usertrack*** modult, akkor is van lehetőség a felhasználók elkülönítésére.
Ebben az esetben a látogatók elkülönítése és útvonalaik összegyűjtése elsődlegesen az IP cím, távoli eseménynapló, távoli felhasználó adatok alapján történik, de az ismertetett problémák miatt ez önmagában nem elegendő az egyértelmű megkülönböztetéshez. További segítséget nyújt a ***referer*** és a böngésző adatok jelenléte. Azonban a legegyszerűbb szerkezetű eseménynaplók nem tartalmaznak az IP címen és a kérés időpontján kívül több olyan adatot, amely segítené a látogatók elkülönítését, vagy egy nagyvállalati hálózatból érkező látogatók esetén jellemzően a böngésző adatok is azonosak.
Az idő alapú elkülönítés során (*1. ábra*) az éppen vizsgált kérést a megadott időintervallumon belül eső, kéréssel rendelkező látogatóhoz próbáljuk meg csatolni, oly módon, hogy a fent említett adatok közül a lehető legtöbb egyezzen. Teljes egyezés esetén a vizsgált kérést az adott látogatóhoz kapcsoljuk, ha nem találtunk teljes egyezést, akkor mint új látogató kezeljük, és egy új útvonal tömb bejegyzést nyitunk neki.

Folytatás

A sorozat következő részében az elkülönített felhasználóink útvonalait fogjuk részletesen kielemezni és elkészítjük az átlagos viselkedést bemutató modelleket.

Beszédes Balázs (beszedes@ei.hu)
24 éves, az e-Média Informatikánál mérnök-informatikus. Hobbija a kerékpározás és a kirándulás.