Rugalmas géptelep? Akkor openMosix!

Korábbi számainkban már olvashattak linuxos telepekről. Most egy másfajta megvalósítással ismerkedhetnek meg.

Az openMosix olyan rendszermag-kiterjesztés, ami telepek (cluster) építését teszi lehetővé. Hogy miért is jó? Miért pont ezt válasszuk? Remélem, ez az írás egyértelmű választ ad ezekre a kérdésekre.
A Beowulf szemléletével szemben az openMosix-telepben nincsen központi gép, mindegyik csomópont egyenrangú. A csomópontok (node) egy-, illetve többprocesszoros gépek is lehetnek, amelyeket helyi hálózaton (LAN) keresztül kell összekapcsolnunk. Egy egyszerű beállítófájlban, amelynek felépítését a későbbiekben részletezem, meg kell adni, hogy mely gépekkel osztja meg az erőforrásait, illetve mely csomópontok erőforrásait használja. Ha helyi hálózaton keresztül észleli a másik gép jelenlétét, erőforrásaikat rögtön megosztják. Ez az elgondolás nagyon rugalmassá teszi, mivel nincs rá szükség, hogy egy gép „közösködjön" a hálózatba kapcsolt összes géppel. Természetesen lehetőség nyílik arra is, hogy bármelyik gép erőforrásainak megosztását visszavonjuk. Ekkor egyszerűen visszahívja a távoli gépen futó folyamatokat és a saját processzorán futtatja őket tovább.

```
# /etc/openmosix.map

1     10.0.1.1      1
2     10.0.1.254    1
2     10.0.2.254    ALIAS
3     10.0.2.1      1
```

Tapasztalatok

Ahogy a hivatalos weblapon is olvashatjuk: „MOSIX!= openMosix". Nem véletlenül, a két elnevezés gyakran keveredik. Megesik, hogy a letöltött openMosix folttal (patch) kezelt rendszermag beállítása közben csak Mosixot emleget vagy a menü neve *openMosix*, de ezen a szinten csak Mosix elnevezésű beállításokat találunk és fordítva. Kipróbáltam a MOSIX-1.5.7-es változatát is, és gyakran előfordult, hogy a másik gépen a folyamatok ragadtak, amin csak egy újraindítás segített. Mindenesetre a Mosix- és openMosix-magokkal szerelt gépeket nem lehet vegyesen használni. Én a Debian GNU/Linux Woody változatára telepítettem (`kernel 2.4.17-openmosix`) minden csomóponton, és tökéletesen tette a dolgát. Háromgépes telepet hoztam létre (két 166 MMX, egy 533Celeron) 100 Mbit/s UTP-kábeles ethernethálózaton, ebből is látni, hogy nem kell erőmű a kipróbálásához. Amin Linux-rendszermag futhat, azt biztosan fürtbe tudjuk kötni. Kipróbáltam többprocesszoros géppel is (Dual 533 Celeron), ezzel sem volt gond. Öröm volt nézni, ahogy a `mosmon` parancsot (illetve `mon-openmosix`) futtatva a gépek terheltsége rövid időn belül közel azonos szintet ért el. Egyik 166 MMX gépen egy videokódoló program indítása után rögtön átkerült a másik gépre, és mindössze 100–200 KB/s nagyságú forgalmat hozott létre – százszázalékos processzorterhelés mellett. Ha nem futtatunk erőforrásigényes alkalmazásokat, akkor is 3–7 KB/s nagyságú forgalmat tapasztalhatunk. Most nézzük, hogyan érhetjük el mindezt!

Telepítés

Először is a rendszermagunkat kell megfelelő állapotba hoznunk. A ➲ http://openmosix.sourceforge.net/ címről tölthetjük le a legújabb rendszermagfoltot (kernel-patch) vagy Debian/GNU Linux alatt egyszerűen adjuk ki a következő parancsot:

```
apt-get install
➥kernel-patch-openmosix
```

Ez 2.4.16-os rendszermaghoz készült. Ha nem akarunk ennyire beavatkozni Linuxunk lelkivilágába, egy `user-mode-linux` rendszermagot is letölthetünk – képességeit saját főkönyvtárrendszerrel (rootfs) futtatva próbálhatjuk ki. Miután letöltöttük a rendszermagfoltot, másoljuk a rendszermag könyvtárába, majd a megfelelő könyvtárában rendszergazdaként adjuk ki a következő parancsokat:

```
zcat openMosix-2.4.17-2.gz | patch -p1
&& make menuconfig
&& make dep
&& make clean
&& make bzImage
&& make modules
&& make modules_install
```

A beállítások jelentése

- `openMosix process migration support`: bejelölésével a folyamatokat önműködően elosztja a csomópontok között, hogy a terhelésüket kiegyenlítse.
- `Support clusters with a complex network topology`: bejelölésével a csomópontok összetett hálózaton is összeköthetők.
- `Stricter security on openMOSIX ports`: biztonsági beállítás, hogy az olyan TCP/UDP-kapukat (port), amelyeket a Mosix használ, felhasználó ne érje el; hogy az erőforrásokat ne használhassa olyan csomópont, ami nincs a beállítófájlban felsorolva; hogy más folyamat ne használhasson olyan kapukat, amelyek a Mosixnak vannak fenntartva.
- `Level of process-identity disclosure (0-3)`: beállítja, hogy mennyi adatot tároljon az egyes folyamatokról.
  0 esetén: nincs további adat; 1: PID (/TGID, ha különbözik a PID-től);
  2: PID (/TGID), UID, GID; 3: PID (/TGID), UID, GID, PGRP, SESSION, COMMAND.
- `Create the kernel with a "-openmosix" extension`: a rendszermag egy *-openmosix* kiterjesztést kap.
- `openMOSIX File-System`: bejelölve openMosix-fájlrendszert (MFS) használhatunk. Erre akkor lehet szükségünk, ha egy csomópont fájlrendszerét az összes többi csomóponton el szeretnénk érni.
- `Poll/Select exceptions on pipes`: engedélyezi, hogy értesítsen egy programot, ha a csőből (pipe) olvasnak. Beállítva a `ioctl` (`pipefd`, `TCSBRK`, `arg`) is használható lesz, hogy kivételeket állíthassunk be, illetve törölhessünk

(exception conditions). Ha az (`arg & 1`) igaz, akkor – ha valaki olvasott a csőből – kivétel keletkezik. Ha az (`arg & 2`) igaz, akkor keletkezik kivétel, ha már nem olvasnak a csőből. Az alapbeállítás az, hogy egyik esetben sem keletkezik kivétel. Egy kivétel kezelője visszatérhet a `select` rendszerhívással, ami azt okozhatja, hogy a `poll` rendszerhívás visszatérési értéke belekerül a `POLLPRI` -be. Megkaphatjuk annak alsó becslését, hogy hány bájtot próbáltak olvasni a folyamatok a csőből az `ioctl` (`pipefd`, `TIOCGWINSZ`, `0`) segítségével.
Létezik olyan rendszermagfolt, ami lehetővé teszi a közvetlen fájlrendszerelérést (Direct File-System Access – DSFA). Csak akkor vehetjük igénybe, ha a Mosix-fájlrendszert is bekapcsoltuk.
Ha a rendszermag kész, állítsuk be az indításkezelőt (boot manager), és telepítsük az openMosix-csomagot (Debian: `apt-get install openmosix`).

Beállítás

Debian/GNU Linux alatt a */etc/openmosix.map*, Mosix esetén pedig a */etc/mosix/mosix.map*) fájlban kell megadnunk a hálózati csomópontokat. A beállítást elvégezhetjük kedvenc szövegszerkesztőnkkel vagy az `update-cluster` programmal is. Meg kell adni a csomópont azonosítószámát, IP-címét és a csomópontok számát a tartományon. Lehetőség nyílik külön alhálókon lévő gépek összekapcsolására is. Ekkor az átjáró mindkét (vagy több) IP-címét meg kell adni. Ha például két alhálónk (10.0.1.0/24, 10.0.2.0/24) és egy átjárónk (10.0.1.254, illetve 10.0.2.254 IP-címekkel) van, az *1. listában* szereplő módon kell beállítani.
Ha már az openMosixot támogató rendszermag fut, indítsuk újra az openMosixot: `/etc/init.d/openmosix restart`. Ha még nem az fut, a gépet az új rendszermaggal indítsuk újra. Mindezt az összes csomóponton meg kell tennünk.
A `mosid` fájlrendszer használatba vétele hasonlóan egyszerű. Először hozzuk létre az *mfs* könyvtárat: `mkdir /mfs`. A */etc/fstab* fájlhoz hozzá kell adnunk egy sort a *2. listán* látható módon. Ekkor a közvetlen fájlrendszerelérést (DFSA) bekapcsoltuk, ami az `mfs_mnt` a */mfs/[openMosix_ID]/* könyvtáron keresztül minden csomóponton elérhető lesz. Ha mindent jól csináltunk, már működik is, és elosztja az erőforrásokat.
Futtassunk nagy erőforrásigényű alkalmazásokat, és nézzük, hogyan nő a többi gép terhelése. Mindezt megtehetjük a `mosmon` parancs futtatásával, illetve attól függően, honnan szereztük be, `mon`, `mon -openmosix` néven találjuk meg. Ha elindítunk egy folyamatot, akár rögtön másik gépre költözhet (migrate), ami a telep (cluster) bármelyik gépe lehet. Ekkor a folyamat két részre szakad: egy felhasználói és egy rendszerszintű részre. Csak a felhasználói rész kerülhet át egy távoli csomópontra, míg a rendszerszintű mindig ott marad, ahol a folyamatot indítottuk. Az utóbbi felelős a rendszerhívások feloldásáért. Éppen ezért az olyan folyamatok, amelyekben gyakoriak a rendszerhívások, csak kis részben költöztethetők át másik csomópontra, vagyis kismértékű gyorsulást tapasztalhatunk. Ha tudni akarjuk, hogy valamelyik folyamatunk hol fut, futtassuk ezen a csomóponton a következő parancsot: `cat /proc/<pid>/where`. A parancs egy azonosítót ad vissza, ami annak a csomópontnak a sorszáma, ahol a folyamat fut. Mindezt egyszerűbben is megtudhatjuk, ha feltelepítjük az `mps` csomagot (Debian: `apt-get install mps`). Az `mps` a `ps` parancshoz, az `mtop` pedig a `top` parancshoz hasonlóan működik, ráadásul hasznos adatokat is megjelenít (melyik csomóponton fut a folyamat stb.). További hasznos tájékoztatást találhatunk a */proc/mosix* könyvtárban (mely csomópontokat éri el, mekkora azok terheltsége, milyen folyamatokat használ stb.).

```
# /etc/fstab

# tobbi fs beallitasai
# ...

#   [device_name]                          [mount_point]
mfs      defaults    0 0    mfs_mnt       /mfs
➥mfs    defaults,dfsa=1      0 0
```

Kapcsolódó címek

- http://www.openmosix.org
- http://www.mosix.org
- http://howto.ipng.be/Mosix-HOWTO
- http://openMosix.sourceforge.net
- http://packages.debian.org/unstable/net/openmosix.html
- http://packages.debian.org/unstable/net/kernel-patch-openmosix.html
- http://packages.debian.org/unstable/net/mps.html

Összefoglalás

Egyszerű használata és rugalmassága következtében hatékonyan alkalmazható üzleti célokra, otthoni gépek összekapcsolására, illetve egyetemi vagy középiskolai gépparkok teljesítményének, kihasználtságának növelésére. Napjainkban az egyetemeken egyre inkább elterjednek a linuxos telepek, amelyek olyan teljesítményt nyújtanak a kutatások számára, ami mindeddig nem volt elérhető. Felhasználási területeinek száma korlátlan, akár Beowulf-telepeket is összeköthetünk vele.

Kolcza Péter
(kpeter@sysconfig.hu)