Alkalmazás-szintű hibatűrés – hibatűrő Apache modul

Önálló laboratóriumi beszámoló összefoglalása

Baffia Zoltán <zoltan@baffia.hu>

2002. december

Az önálló laboratoriumi feladatom annak megoldása, hogy egy adott, webszerverekből álló klaszteren belüli hardver hibákat a hozzá csatlakozó kliensek számára transzparensen lehessen kezelni. Ez gyakorlati szempontból annyit jelent, hogy ha egy felhasználó csatlakozott a klaszterben lévő egyik szerverhez, és ezáltal a kérésének kiszolgálása megkezdődött, akkor egy esetlegesen bekövetkező hardver hiba esetén ne kelljen újra csatlakoznia és kérésének kiszolgálását elölről kezdeni, hanem egy, a klaszterben lévő másik szerver azt folytatni tudja anélkül, hogy ebből a felhasználó bármit is érzékelne.

Az egyszerűség kedvéért a rendszert két szerver számítógép esetére terveztem. Mindkét rendszeren Linux operációs rendszerre telepített Apache web szervernek kell működnie, ami tartalmazza a saját modulomat és annak működéséhez nélkülözhetetlen kommunikációt megvalósító szoftvert, a *Spread*-et.

A szerverekhez egymástól függetlenül érkeznek a felhasználói kérések, és azok függetlenül szolgáltatják rájuk a válaszokat a HTTP protokol alapján. A szerverek külön számítógépen futnak, azonban egy közös belső hálózatban (LAN) helyezkednek el, amin keresztül a kommunikációjuk zavartalan. Mindkét számítógépen megegyező módon vannak telepítve a szoftverek és az általuk biztosított szolgáltatás is azonos, amint ezt a feladatom megfogalmazásában is feltételeztem. A szervereken belül kitüntetett szerepe van a saját moduljaimnak (mindkét szerveren egy-egy), amik a hibatűrést megoldják. A köztük lévő virtuális kommunikációs csatorna a korábban már említett *Spread* program segítségével hozható létre.

Amikor a saját modulom végrehajtására kerül a sor, akkor a többihez hasonlóan megkap egy mutatót, ami az aktuális kéréshez létrejött *request_rec* rerkordra mutat. Ezen keresztül minden információt el lehet érni, ami a hibatűréshez szükséges. Egyszerűségi megfontolások alapozták meg a döntést, miszerint a kéréshez kapcsolódó mindhárom adatstruktúrát (*request_rec*, *server_rec*, *conn_rec*) fel kell használni a hibatűrésnél. A rendszer későbbi finomítása során ez az adatmennyiség tovább csökkenthető.

Az alapötlet, hogy a működés során mindkét szerverben karbantartok egy másolatot a másik szerveren lévő aktuális feladatokról, az ottanihoz hasonló egymáshozfűzéssel. Azaz ha egy új kérés érkezik, akkor a modul a hozzá kapcsolódó struktúrák egy másolatát elküldi a másik szervernek, ami beilleszti az általa karbantartott listába. Ha egy kérés kiszolgálása befejeződött (a kiszolgálást elvégző szervernek a feladataiból kikerültek a hozzá kapcsolódó rekordok), akkor ez a másik szerveren tárolt sorból is törlődik. Ezt a feldolgozás végén, szintén egy üzenettel jelzi az egyik modul a másiknak.

A modulom a kiszolgálás két fázisában tevékenykedik, az elsőben és az utolsóban. Természetes egyszerűséggel adódik, hogy hiba esetén a működő szerveren, annak saját feladatai mellé kell beilleszteni a másik által elkezdett, de a meghibásodás miatt be nem fejezett kéréseket. Ezt is az első fázisban oldottam meg, mintha azok új kérésekként érkeztek volna. Így a működő szerver – a szabad kapacitásának megfelelően – ezeket a kéréseket is ki fogja szolgálni.

A *Spread* nyújt egy olyan szolgáltatást is, ami értesíti az őt igénybe vevő programot, amennyiben annak a csoportjába tartozó működő gépek számában változás történik. Az nem más, mint a tagsági kép folyamatos karbantartása. Ebből az információból tehát könnyedén értesülhet a modulom arról, hogy a másik szerver működésképtelenné vált, tehát meg kell kezdeni az azon lévő kérések hozzáfűzését az aktuális szerverhez érkezett kérések listájához.

A teljes rendszer implementálása jelenleg még folyamatban van, így az eddig már megvalósított részeket szeretném most bemutatni.

Az első feladat a már említett *Spread* programmal való kapcsolat felvétele, melyet nyilvánvalóan a szerver indításakor, mint egy inicializálásként kell megtenni. Ezt követően már az üzentküldés megoldottnak illetve elérhetőnek tekinthető, a szoftver programozói felületén keresztül. A modulom feladatának elvégzése szempontjából két fontos eseménykezelő függvény van. Az egyik a kérés kiszolgálásánál létrejövő gyermek (*child*) folyamat inicializálásakor fut le, a másik pedig ennek a szálnak a lezásárásakor. Az előbbi feladata az adatstruktúrák továbbítása a másik modul felé, az utóbbié pedig egy üzenet küldése, minek hatására a másik szerveren futó modul az adott kérést kitörli az aktuális feladatok másolatát tartalmazó listából. Ezek már megvalósított funkciók, melyeket a tesztelés során helyesen működőnek találtam.

Másfelöl szükséges a *Spread*-en keresztül érkezett üzenetek kezelése, nevezetesen egy új kérésnek a listához kapcsolása, egy régi kérés eltávolítása abból, valamint az így karbantartott lista „aktiválása”, vagyis az aktív feladatokat tartalmazó listával való egyesítése. A jelenlegi állás szerint a lista karbantartása és „aktiválása” manuálisan már működik, azonban az, hogy mindez a *Spread*-en keresztül érkezett üzenetek hatására, mintegy automatikusan működjön, még nincs megvalósítva.