

# Hiba utáni helyreállítás Java alapú rendszerekben

## Önálló laboratórium feladat összefoglalója (1. félév)

Kárer Gyozo (XFICP2)

Konzulens: Dr. Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék  
Informatikai infrastruktúra tervezése szakirány, 2006/2007. I. félév

Napjaink számítástechnikai alkalmazásaiban igen fontos szerep jut a megbízhatóságnak és szolgáltatások folyamatos rendelkezésre állásának. A tranziens jellegű hibák esetén alkalmazható hibaturés megvalósítása általánosságban 4 lépést igényel: (1) Hibadetektálás, (2) hibabehatárolás és kárfelmérés, (3) helyreállítás és (4) hibakezelés és a szolgáltatás folytatása.

Az önálló laboratórium feladat keretében a hiba utáni helyreállítás technikái közül az *állapotmentésen alapuló visszalépo helyreállítással* ismerkedtem meg Java környezetben. A gyakorlatban több javaslatot és megvalósítási példát találhatunk Java alapú helyreállításra. Beszélhetünk olyan eszközökről, amelyek a Standard Java-t és annak könyvtárait alkalmazzák (pl. Object serialization, Long-term persistence) illetve olyanokról amelyek nem szabványos eszközökre épülnek (pl. Thread migration). Ezek a technikák sok tekintetben különböznek egymástól, hatékonyságuk és képességeik összevetése nem egyszerű feladat. Az önálló laboratórium során mintakísérletekkel és mérésekkel kell megmutatni az egyes megoldások elonyeit és hátrányait.

A technikák megismerése után a feladatom volt a technikák idő és tár overhead alapján történő összehasonlítását lehetővé tevő benchmark alkalmazás elkészítése, majd mintakísérletek végzése a szabványos Java alapú technikák felhasználásával.

A feladat elkészítésének első fázisa a mérendő alkalmazás (benchmark) elkészítése volt. Ennek elkészítése során az egyszerűsége törekedtem, azonban olyan szempontokat figyelembe véve, hogy az alkalmazás skálázható legyen a mentendő objektumok mérete, a mentések gyakorisága és hibamentessége szempontjából. Ezeket szem előtt tartva mátrixon elvégzendő műveletekre esett a választásom. Az alkalmazás osztályainak elkészítése után megvalósítottam az Object Serialization és a Long-Term Persistence technikák által megvalósított állapotmentés és állapot-visszaállítás lépéseit. Mindezek megvalósulása után el kellett készítenem a méréseket elvégző környezet elemeit. Ennek elkészítése során a fő cél a rugalmasság megvalósítása, a könnyű testre szabhatóság volt. Ez a könnyű testre szabhatóság annyit takar, hogy a mérést végző személy meghatározhatja a mérés során alkalmazott lépésszámot, továbbá azt, hogy hány lépésenként történjen a mentés, a checkpoint-ok elhelyezése, hány hibát injektáljunk a rendszerbe, ezek milyen eloszlásúak legyenek. Ezen kívül meghatározható még, hogy a helyreállítás sikeréhez a hiba észlelésétől visszafelé számított hányadik állapotmentés betöltése hozza meg a kívánt eredményt.

A mérések során olyan eredményre jutottam, ami idő overhead szempontjából megfelel a szakirodalomban leírtaknak (Object Serialization esetén az idő overhead lényegesen kisebb), azonban a tár overhead mérése által szolgáltatott eredmények kérdésesnek tekinthetőek, hiszen ezek azt mutatják, hogy a Long-Term Persistence technika tár igénye lényegesen kisebb. Ennek a szakirodalomhoz képesti eltérésnek a valószínű oka programozástechnikai jellegű.

A további félévekben érdekes feladat lehet az elosztott rendszerekben alkalmazható eljárások megismerése, valamint hatékony programozási technikák (aspektus-orientált programozás, modell alapú optimalizálás) alkalmazása.