

Web szolgáltatás alapú hibatűrő keretrendszer

Önálló laboratórium feladat összefoglalója (1. félév)

Sáfrány Zsolt (G71SIJ)

Konzulens: Kovács Máté

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

Az első félév – a labor szempontjából - javarészt kutatással, olvasással, a használni kívánt technika megismerésével, valamint a következő félévre kivetített tervek megalapozásával zajlott. A részletesen megismert technológiák:

Irodalomkutatás: XML, SOAP, WSDL, WSDD, weben előforduló Quality of Service igények, ezek biztosításának nehézségei, valamint eddigi próbálkozások, szabványok a QoS igények teljesítésére. A feladathoz az Apache Axis web szolgáltatás szerverét használom, így ennek megismerése is megtörtént, különböző szolgáltatások installálása WSDD segítségével, ami a szolgáltatás leírója, megszabja, hogy pl.: mely függvények legyenek publikálva, és hogyan legyen példányosítva az osztály, kérésenként vagy csak első híváskor. Megismertem hogyan kell WSDL-ből szerver szeletont és kliens csontot generálni.

Tervezés: Különböző szolgáltatásokat találtam ki, amelyeket majd a keretrendszer fog nyújtani teljesítmény és rendelkezésre állás növelése céljából. Ezek fajtái röviden:

„Átlátszó k+1 állapot nélküli HA”: k plusz egy szerverünk van, egy szerver publikus csak, a maradék k nem látható kívülről. A publikus web szerveren csak a keretrendszer van installálva, a publikálni kívánt szolgáltatás implementálva nincs. A kéréseket a központi szerver továbbítja a belső szerverekhez a saját listája alapján – amit visszajelzések valamint timeout-ok alapján tart karban – majd a visszakapott választ továbbítja a kérést indító klienshez.

„Áttetsző k+1 állapot nélküli HA”: Annyiban más az előzőtől, hogy a klienst redirecteli a központi webszerver az egyik belső szerverhez. Azaz nem a központi ws fog válaszolni a kliensnek. Ez gyorsabb, viszont timeoutot nem tud kezelni a központi szerver. Ez kivédhető egy gyors – redirect előtti - „élsz-e még” kérdéssel.

„Több-variánsú k+1 HA”: A tranzienst és permanens programhibák kivédésére. Különböző programokat készítünk, akkor kevés rá az esély, hogy ugyanarra az inputra mindegyik hibás eredményt adjon. Kihhasználva a web szolgáltatások platform függetlenségét, a megírt nyelvvel is lehetne variálni, egyiket Javában míg a másikat .NET-ben implementálva, erősen különböző variánsokat hozhatunk létre. A válaszokból szavazással, vagy elfogadhatósági teszttel lehetne dönteni.

„Wsla k+1 HA”: WSLA leírókat is figyelembe vevő HA szolgáltatás. Belső szerverek rendelkezésre állásának statisztikáját nem kell számolni, mert az adatok adottak.

„Átlátszó k+1 állapot nélküli LB”: A központi szerver felváltva hívogatja a belső web szervereket. Jegyzi a válaszidőket, illetve a kérések számát, így válaszidőt vagy áteresztőképességet tudunk javítani, az alapján, hogy melyik belső szerveret preferáljuk a kérések továbbítása szempontjából.

„Áttetsző k+1 állapot nélküli LB”: Hasonló az előzőhöz, azzal a különbséggel, hogy a kliens redirectálódik az egyik belső szerverhez, nem a központi szerver fog válaszolni.

„Wsla k+1 állapot nélküli LB”: Nincs szükség számításra, belső ws-ek teljesítmény mérésére. Mindezt a hozzácsatlakoztatott szolgáltatások WSLA leírói alapján származtatnánk.

Implementálás: Az implementálási részben eljutottam oda, hogy tetszőleges web szolgáltatást létre tudjak hozni. Írtam egy java osztályt, ami egy egyszerű feladatot lát el (mint pl.: adott szöveget k darabszor egymás után konkatenál), majd a WSDD leíróban megszabva, hogy mely függvényeit akarom publikálni, installáltam web szervizként az osztályt. A WSDD-ből tudok kliens csonkot generálni, amivel Java nyelvben a szolgáltatást objektum orientáltan tudom meghívni. A web szolgáltatásomat más - .NET – környezetből is sikerült meghívnom, ellenőrizve ezzel a platform- és nyelvfüggetlenséget.