



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

A Postlock megosztott állapotter használata hagyományos webalkalmazásokban

Bartha Dániel BA8D5T, műszaki informatika szak

Konzulensek:

Majzik István MIT,

Neumark Péter,

Informatikai technológiák szakirány/ Rendszertervezési ágazat
Önálló labor összefoglaló 2010/11. II. félév

Feladat:

A Postlock megosztott állapottereket kezelő alkalmazáshoz, illetve az alkalmazást futtató Erlang környezettel kellett kapcsolatot létesítenem PHP alkalmazásból, hogy a megosztott állapotter elérhető legyen PHP környezetből is.

A felhasznált technológiák:

Postlock: egy Erlang környezetben működő nyílt forrású, megosztott állapottereket kezelő alkalmazás, mely néhány egyszerű adattípus megosztott használatát teszi lehetővé: lista, szótár, adathalmaz. A hozzá kapcsolódó kliensek a kezdeti szinkronizációt követően maguk kezelhetik az állapotterüket: a változtatást követően, a kliens értesíti a Postlock szervert, hogy mi volt a változás, de közben aszinkron módon folytathatja a saját állapotter példányának módosítását, csak azt kell biztosítani, hogy ha a szerver nem fogadja el a változtatást (pl. ütközés más klienssel), akkor vissza tudjon lépni egy korábbi közös állapotba.

Thrift: A Facebook által gondozott, nyílt forrású keretrendszer, melynek segítségével különböző operációs rendszereken futó, különböző programozási nyelveken íródott alkalmazások kommunikációját egyszerűen lehet megvalósítani. A kommunikációban résztvevő változókról, metódusokról formális leírást kell készíteni, melyekből a Thrift fejléceket generál. A generált fejléceket kitöltve lehetővé válik a változók cseréje / távoli eljárások hívása socket alapokon, úgy hogy a kommunikáció felépítésével, kezelésével a felhasználónak már nem kell foglalkoznia.

A megoldás:

A feladat a Thrift fejlécek elkészítése volt, hogy az Erlang kiszolgáló és a PHP kliens közötti adatszerét megkönnyítsük. A Postlock az adatokat egy JSON formátumú stringben küldi és fogadja, így nem volt szükség különleges fejlécek alkalmazására az egyes Postlock adattípusok esetén, elegendő volt egy általános string küldő-fogadó formátumot meghatározni. Ennek annyi hátránya van, hogy a PHP kliensnek mindig elő kell állítani egy speciális formátumú JSON üzenetet, viszont így a Postlock funkcióinak bővülésekor nem kell új Thrift modelleket készíteni.

Az egyik fő probléma a PHP alkalmazások esetében az, hogy hogyan lehet küldeni információt a felhasználónak, ugyanis a PHP RESTful architektúrájú, azaz egy-egy felhasználói futtatást követően „nyugvó” állapotba kerül, a szerveren történő bármilyen változást kizárólag egy következő futtatás során érzékel a program. Emiatt egyrészt valahogy meg kellett oldani, hogy a Postlock szerverhez kapcsolódó PHP klienseknek legyen egy callback felülete, azaz a megosztott állapotterben történő változásokról a kliens egy rajta futó PHP szkript URL-jét meghívva lehet tájékoztatni. Persze ekkor még csak a PHP kliens tud a változásokról, ezt még továbbítani kell a felhasználónak. Ez két módon lehetséges: a felhasználó időnként a háttérben lekérdezi a PHP kliensről, hogy van-e változás az

állapottérben, vagy WebSocket alapokon a felhasználó és a PHP kliens folyamatos kapcsolatban legyen, így rögtön lehet értesíteni a felhasználót a változásokról.

Az fent ismertetett módon így lehetőség van tehát arra, hogy a felhasználó egy PHP kliensen keresztül kapcsolódjon a Postlock szerverhez és az üzeneteit el tudja juttatni, illetve értesüljön a szervertől küldött válaszokról.

Összefoglalás

A feladat célkitűzéseit nem sikerült teljesen megvalósítani, mert az Erlang környezet nem tudja közvetlenül visszahívni a PHP klienst, igaz egy folyamatos lekérdezési modell mellett az Erlang környezet változásai nyomon követhetőek, persze ez éppen ugyanaz a fajta probléma, mint amikor PHP környezetben kellene valahogy tudomást szerezni egy a szerveren bekövetkezett eseményről.

A feladat során megismertem a Postlock alkalmazást és a Thrift technológiát, mindkettőt igen hasznosnak találtam saját területükön. Nehézségeim abból adódtak, hogy az Erlang nyelvet nem ismertem illetve a szerver környezettel voltak gondjaim, de a konzulensemtől minden segítséget megkaptam.