



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

## Szuperskálázható modellezés

**Szárnas Gábor (U944EQ), I. évf., MSc mérnök inf. szakos hallgató**  
**Konzulens: Izsó Benedek, Ráth István, Varró Dániel, MIT**  
**Szolgáltatásbiztonságra tervezés szakirány**  
**Önálló laboratórium 2. összefoglaló**  
**2012/13. I. félév**

Az önálló laboratórium munkám egy hosszabb projekt része, amelynek célja egy elosztott modelltároló rendszer kialakítása. A modellvezérelt tervezés egyre szélesebb körű alkalmazásának következménye, hogy a modellek komplexitása egyre nő, mind a kézzel készített, mind a generált modellek esetén.

A növekedésnek jelenleg komoly határt szab, hogy a modelltároló rendszerek csak egyetlen számítógépen képesek működni, ezért csak korlátozott méretű modellek tárolására alkalmasak. A problémára egy olyan horizontálisan skálázható rendszer jelentené a megoldást, amely újabb számítógépekkel bővítve egyre nagyobb méretű modellek tárolására lenne képes. Az elosztott tárolás mellett megfogalmazódik a modell párhuzamos feldolgozásának és biztonságos tárolásának igénye is.

A rendszer alapját egy adatbázis-kezelő rendszer képzi, amely képes az elosztott működésre. Az elmúlt években a jól bevált relációs adatbázis-kezelők mellett tucatjával jelentek meg ún. NoSQL elosztott adatbázis rendszerek. Ezek specializált architektúrájukkal a Web 2.0-s alkalmazások igényeinek egy-egy aspektusát próbálják minél hatékonyabban lefedni, a relációs adatbázisokhoz képest gyakran egyszerűbb adatmodellt, gyengébb konzisztenciát, de magasabb rendelkezésre állást, jobb skálázhatóságot nyújtanak. A NoSQL adatbázis rendszerek gyakran egy-egy használati esetre jellemző terhelésre optimalizáltak. A modellalapú tervezőrendszerek tipikus felhasználásai: a modell eseményvezérelt szerkesztése, frissítése, modelltranszformációk és jólfomáltsági validációk futtatása.

A munkám előzménye az Izsó Benedek, Ráth István és Szatmári Zoltán által készített „TrainBenchmark” teszt sorozat, amely különböző – pl. EMF- és RDF-alapú – tárolórendszerekbe betöltött modelleken futtat jólfomáltsági kényszereket kiértékelő lekérdezéseket [1].

A félév során folytattam a TrainBenchmark mérő sorozaton végzett munkát. Továbbfejlesztettem a Neo4j gráf adatbázis-kezelőn végzett mérést a betöltési folyamat átalakításával. Az átalakításhoz a Tinkerpop keretrendszer Blueprints adatmodelljéhez tartozó GraphMLWriter és GraphMLReader osztályokat használtam, amelyek képesek GraphML formátumban kezelni a gráfot. Az átalakítással a betöltési fázis egyszerűen mérhetővé vált.

A generátort és a mérést megfelelően felparaméterezve lehetővé tettem indexet használó és indexet nem használó mérő sorozatok futtatását. A különböző eszközök kimenetének összehasonlításához a Neo4j mérésbe bevezettem az eredményhalmazok (beszúrás szerinti) rendezésének lehetőségét.

Megismerkedtem a 4store triplestore adatbázis-kezelővel. Elkészítettem a telepítését és indítását végző szkripteket. A mérések során a 4store-t egy 4store HTTP szerveren és Java API-n keresztül értem el. A mérést először egyetlen 4store szerverre készítettem el, majd egy 4 gépből álló elosztott 4store fűrtre. Az elosztott méréshez elkészítettem azokat a bash szkripteket, amelyek képesek távolról 4store szerverpéldányokat indítani.

A fűrtön futtatott méréssel jelentős sebességnövekedést sikerült elérnem a lekérdezések esetén.

A témát a Diplomatervezés 1–2. tárgyak keretében fogom folytatni további mérésekkel, az EMF-IncQuery projektben [2] is alkalmazott Rete algoritmus elosztott implementációjával és egy elosztott modelltároló rendszer fejlesztésével.

[1] <https://trac.inf.mit.bme.hu/Ontology/wiki/TrainBenchmark/>

[2] <http://viatra.inf.mit.bme.hu/incquery/>