



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

## **Automatikus skálázási megoldások felhő-alapú alkalmazásokhoz**



**Fikner Patrícia III. évf, (BSc) mérnökinf. szakos hallgató**  
**Konzulens: Szatmári Zoltán tudományos segédmunkatárs, MIT és**  
**Kocsis Imre tudományos segédmunkatárs, MIT**  
**Informatikai technológiák szakirány, rendszertervezés ágazat**  
**Önálló laboratórium összefoglaló**  
**2014/15. II. félév**

Napjainkban egyre jobban terjedő felhő-rendszerek használatakor nagy figyelmet kell fordítani az erőforrásigényeknek pontos becslésére, hiszen a legtöbb ilyen szolgáltató óra alapú díjazást alkalmaz használat közben. Ahhoz, hogy a költségeket minimalizálni tudjuk, és fel tudjunk készülni a változó terhelésekre, szükség van valamilyen automatikus erőforrás-menedzsmentre. Ennek a problémának a megoldását kutattam a félév során.

Az önálló laboratórium keretében megismerkedtem az auto-scaling technikák előnyeivel és jellegzetességeivel. Több megoldást is találtam a témában, végül egy tudományos cikket alapul véve a Kalman-szűrő alkalmazásánál állapodtam meg. Ehhez további kutatásokat kellett végezni a működése és felhasználási területeit illetően. A félév jelentős részében főleg irodalomkutatást végeztem, tanulmányoztam az Amazon és Openstack által használt technikákat, valamint megtanultam az Amazon által kínált auto-scaling szolgáltatást használni. Elkezdtem implementálni egy olyan megoldást, ami majd Kalman-szűrő felhasználásával fog adott rendszert skálázni.

A saját auto-scaling megvalósításhoz szükséges volt, hogy megismerkedjek a ha-proxyval load-balancing alkalmazásához, munin-nal a gépek monitorozásához, tomcat-tel a java mintaalkalmazáshoz, postgresql-lel az alkalmazáshoz tartozó adatok tárolásához, és a TPC-W benchmarking alkalmazással, amivel a tesztelést lehet a későbbiekben végezni.

Az implementációval addig sikerült eljutni, hogy van egy működő mintaalkalmazás, ami áll egy alkalmazáserverből és egy adatbázisserverből. Egy csomópont ezt az alkalmazáservert terheli, és egy load-balancer elosztja a terhelést több alkalmazáserver esetén. Ezenkívül egy vezérlő csomópont az alkalmazást monitorozza (python scriptből telenetén keresztül). Mind az öt virtuális gép egy-egy Amazon EC2 példány.