

# Modellmanipulációk in-memory adatbáziskezelő rendszerekben

## Önálló laboratórium feladat összefoglalója (2. félév)

Tóth Gergő (V8GHJO)

Konzulensek: Bergmann Gábor, Horváth Ákos

BME Méréstechnika és Információs Rendszerek Tanszék

Informatikai infrastruktúra tervezése szakirány, 2008/2009. I. félév

A nagyméretű modellek hatékony feldolgozása napjaink égető problémája. Ma az operatív tár sebessége egy nagyságrenddel nagyobb a diszkek sebességénél, ezért a teljesítmény érdekében a modelleket célszerű a memóriában tárolni és manipulálni. Az ún. in-memory DBMS-ek az adatbázist teljes egészében a memóriában tárolják, így sok feladatra érdekes alternatívák lehetnek. A kutatásom célja az volt, hogy megvizsgáljam, milyen feladatokra lehet versenyképes egy in-memory adatbáziskezelő rendszer, illetve összehasonlítsam a piacon jelenlevő fizetős és open source termékeket.

Gráf transzformációk relációs adatbázisokban címmel Varró Gergely egy publikációt jelentetett meg. Az írás lényege, hogy tetszőleges problémára felrajzolható egy metamodell, melynek részei mechanikus módon átírhatóak adatbázis sémára. A mintakeresés adatbázis VIEW-k segítségével történik, ezután az eredményhalmazt átadjuk valamilyen procedurális logikának (DBMS tárolt eljárása, vagy alkalmazásba épített DML parancsok), amely INSERT/UPDATE/DELETE műveleteket hajt végre.

A laboratórium során 5 terméket hasonlítottam össze. Kereskedelmi szoftverként az Oracle TimesTen-jét és az IBM solidDB-jét, open source megoldásként pedig a H2 Database-t és a MySQL Community Server-t. Ez utóbbi alapvetően diszkes, de létezik MEMORY típusú táblája, amit szintén alávettem a tesztnek. A feature listák alapján a fizetős szoftverek előnye, hogy nagy rendelkezésre állást tudnak biztosítani állapotmentés és replikáció segítségével.

A Sierpinski háromszög egy önazonos alakzat. Részletessége a végtelenségig növelhető/generálható, és ez a generálás egy benchmark eszköz lehet. Egy Java alkalmazást implementáltam, ami JDBC-n keresztül mind az 5 adatbázison ugyanazokat a Sierpinski transzformációs műveleteket hajtja végre. A TimesTen jött ki a leggyorsabbnak. Ezt nagy meglepetésemre a MySQL követte, a memory típusú táblával. Csupán harmadik lett az open source H2. Hosszas finomhangolás után is csak a diszkes MySQL-hez hasonló teljesítményt tudott produkálni a solidDB. Igaz, annyira sok tuning lehetősége van, hogy elképzelhető, hogy ezen lehetett volna még javítani.

A Petri-háló formalizmust konkurens, eseményvezérelt rendszerek hatékony leírására fejlesztették ki. Az ehhez kapcsolódó benchmarkban a mintakeresés lényegében a tüzelhető tranzíciók keresése, a modellmanipuláció pedig maga a tüzelés. Különböző eszközök léteznek ilyen Petri-hálókat leíró PNML fájlok generálására, ezért a tesztalkalmazásom PNML-t fogad inputként. Ebben a benchmarkban a diszkes MySQL és a TimesTen teljesítménye közötti különbség nagyjából megfelelt a Sierpinski-nél tapasztaltaknak. Meglepő módon a memory táblás MySQL 8%-al lassabb volt a diszkes MySQL-nél. A H2 és a solidDB az eltérő SQL értelmezés miatt ezúttal kimaradt a mérésből – sajnos nagyon jellemző az SQL nyelv dialektusainak különbözősége (DDL, tárolt eljárás, AUTO\_INCREMENT, adminisztrációs utasítások).

Összefoglalásul elmondható, hogy feladattól függően jóval gyorsabb lehet egy in-memory DBMS a hagyományosnál. Legfőbb tanulság, hogy az adatbázis-alapú alkalmazások esetében bármilyen apró tervezési/konfigurációs melléfogás drasztikus teljesítményromláshoz vezethet.