

Replikáció, mentés és visszaállítás technikai MySQL adatbázisokban

Önálló laboratórium feladat összefoglalója (1. félév)

Székely Levente C3ZQO6

Konzulens: Erdősné Németh Ágnes

BME Méréstechnika és Információs Rendszerek Tanszék

Informatikai infrastruktúra tervezése szakirány, 2005/2006. I. félév

1. Replikáció

A replikáció célja, hogy redundáns egységek beépítésével növeljük a hibatűrést és a gyorsabb elérést. A MySQL támogatja az egyirányú, aszinkron replikációt, amely szerint az egyik szerver a **master**, további egy vagy több szerver a **slave** szerepét tölti be. A master server a saját **Binary log** fájljába tárolja az update-eket, amelyeket elküldi a slave szerver(ek)nek. A logok alapján a slave ugyanazokat az adatmódosításokat végre tudja hajtani a saját adatain. A Binary log minden olyan utasítást tárol, amely adatokat módosított vagy módosíthatott volna. (pl. delete no matched rows) Amikor egy slave csatlakozik a master szerverhez, az utolsó sikeres log olvasás pozíciójáról informálja a mestert. A replikáció kialakítása során figyelni kell arra, hogy minden adatmódosítás a master szerveren történjen először. Amikor egy slave egységet üzembe helyezünk, min. azokat az adatokat kell tartalmaznia, amelyeket a master szerver tartalmazott, amikor engedélyeztük a Binary log fájl írását. Az adatok másolására a master szerverről a slave-re a LOAD DATA FROM MASTER utasítás szolgál (feltétel: minden master tábla MyISAM formátumú legyen). A parancs végrehajtása alatt minden master tábla LOCK-olva van. Az adatok betöltése után a slave szerver(ek) periodikusan kapcsolódnak a masterhez és adatmódosításokat várnak. A master szerver nem tudja sem a hozzá csatlakozó slave egységek számát, sem azt, hogy milyen állapotban vannak az adatok aktualitása szempontjából. Úgy a master, mint a slave esetében egyedi replikációs azonosítóval kell ellátnunk a szervereket. (my.cnf: [mysqld] server-id) Az azonosítónak pozitív egésznek kell lennie a [1,2³²-1] tartományból. Szintén a konfigurációs fájlban történik a master esetén a Binary log engedélyezése is.

2. Mentés, visszaállítás

Biztonsági mentésekre a tárolt adataink fontossága miatt van szükség. Információs rendszereink közvetve vagy közvetlenül különféle meghibásodásoknak vannak kitéve, melyek során az adataink teljesen vagy részlegesen elveszhetnek, megsemmisülhetnek. Az adatok részletessége szempontjából megkülönböztetünk teljes vagy inkrementális mentéseket. *Teljes mentés* esetén az adatbázis teljes tartalmát mentjük. Erre a MySQL standard disztribúciójával szállított eszközök állnak rendelkezésünkre (**mysqlhotcopy**, **mysqldump**). *Inkrementális mentés* esetén csak az utolsó mentés óta történt változásokat mentjük. Erre a replikációnál megismert bináris log fájl nyújt lehetőséget. Az elkészült biztonsági mentéseket célszerű az éles rendszertől eltérő, távoli földrajzi helyen tárolni (**Off-site backup**). Mentéskor biztosítani kell az adatkonzisztenciát (**Offline – Online backup**).

Felhasznált irodalom:

- <http://dev.mysql.com/doc/refman/4.1/en/index.html> - a MySQL online dokumentációja
- *Jeremy Zawodny*: High Performance MySQL