

# Replikointi MySQL Database Server –tuotteella

Åke Tyvi  
Datanomi, IT-insinööriopiskelija  
ake.tyvi@edu.stadia.fi

## Tiivistelmä

*Tässä dokumentissa kuvataan MySQL AB:n MySQL Database Server 5.0.4 Beta –ohjelmiston replikointitavat ja replikoinnin tekninen toiminta alkuperäisjulkaisun Replikointi MySQL Database Server –tuotteella mukaisesti. Tämä dokumentti ei käsittele julkaisun jälkimmäistä osuutta liittyen replikoinnin käyttöönottamiseen.*

## 1. Esittely

Tämä dokumentti on referaatti 16.6.2005 julkaistusta raportista ”Replikointi MySQL Database Server”<sup>1</sup> – tuotteella. Alkuperäinen julkaisu, tämä referaatti ja lyhyt PowerPoint-esitys samasta asiasta on noudettavissa asiakaskäyttöön elektronisessa muodossa internetosoitteesta <http://www.curatores.info/henkilosto.html>.

Alkuperäisen raportin sisältö on jaettu kahteen osaan, joista ensimmäinen käsittelee MySQL-tuotteeseen implementoidun replikointikäytännön teknistä toteutusta, ja jälkimmäinen isäntä-orja -replikaatin toimintaa käytännön demonstraation kautta.

Tässä dokumentissa kuvataan MySQL Database Server –tuotteen replikoinnin tekninen toiminta.

Raportin luettuaan lukija ymmärtää miten replikointi toimii ja miten replikointi kytketään toimintaa.

## 2. Replikointi

Replikointi tarkoittaa lyhyesti kerrottuna tiedon kopiointia paikasta A paikkaan B. Tietojen automaattisessa päivityksessä vain muuttuneet tiedot kopioidaan A:lta B:lle. A:n kopiota B:ssä kutsutaan replikaatiksi.

Asiaan perehtymätöntä lukijaa kehoitetaan tutustumaan replikointiteoriaan, ellei asian teoriaosuus ole lukijalle ennestään tuttu. Alkuperäisen raportin ensilehdillä mainitaan asiaa käsitteleviä teoksia (mm. Andrew S. Tanenbaum, Maarten van Steen, Distributed Systems Principles

and Paradigms:Prentice-Hall, Inc.: ISBN 0-13-088893-1) ja julkaisuja, joiden avulla lukijan on helpompi ymmärtää ja sisäistää käsiteltävän aiheen problematiikka.

## 3. Alustavat selvitykset ja rajaukset

Tutkimussuunnitelmassa on rajattu selaus- eli lukutarkoitusta varten tehty replikaatti transaktioreplikaatista eriyväksi: Mikäli tietokannan kuvaukseen tai sen sisältämään tietoalkioon kohdistuu useita samanaikaisia muutosoperaatioita, on MySQL Database Server -tuotteeseen asennettava erillinen InnoDB-tapahtumanhallintaohjelmisto. Raportin jälkimmäisessä osiossa rakennettava isäntä-orja –selausreplikaatti ei täytä vaatimusta erillisen tapahtumanhallintaosuuden asentamisesta, joten asia sivuutetaan huomautuksella raportin tekstissä.

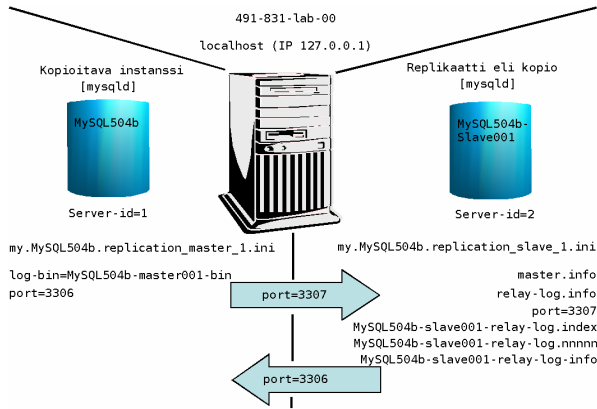
## 4. Tutkimusympäristö ja replikoitu tietokanta

Selvitys on tehty MySQL AB:n MySQL Database Server 5.0.4 Beta –ohjelmistolla, eikä eroavaisuuksia aleneviin ohjelmaversioihin huomioitu.

Selvitystyötä varten asennettiin ohjelma yhdelle laitteistolle, mihin luotiin kaksi erillistä tietokantainstanssia (kuvassa 1 isäntä MySQL504b ja orja MySQL504bSlave001).

Instanssien alustuksessa käytettävät parametrisointi tiedostot on kuvattu raportin liitteessä.

Testiympäristöksi raportissa asennettiin MySQL AB:n sivustoilta saatavan **World**-tietokannan, jonka siirtymistä (data ja taulujen rivit) isännän ja orjan välillä käytettiin replikoinnin havainnollistamisessa. Kirjoittaja on liittänyt raporttinsa liitteeksi *World*-tietokantakuvausten siltä varalta, että joku toinen haluaisi toistaa suoritettua kokeen.



Kuva 1 Rakennettu replikointiympäristöstä.

## 5. MySQL-replikoinnilla saavutettavat edut ja sen haitat

MySQL Database Server tukee yksisuuntaista replikointia. Tietoon kohdistuvat muutokset suoritetaan aina isäntäkannassa ja muutokset vieritetään yhden keskitetyn isäntäkannan kautta orjille.

Mitä järkeä on sitten kopioida sama tieto useaan paikkaan? Raportin mukaan tietoa monentamalla

- orja voi toimia tietovaraston varmuuskopiona,
- kysely- ja selaushaut eli SELECT-haku voidaan suorittaa orjalla tasaten näin isäntäkoneen prosessorikuormitusta ja vähennetään turhaa tietoliikennettä ja
- c) voidaan varmuuskopio ottaa orjasta isännän jatkaessa operatiivista toimintaa.

Tapahtumatieto voi tietokoneen virransyötön katketessa 'haihtua ilmaan' ohjelmallisesti toteutetusta puskuri- ja välimuistista, tai kiintolevyn puskurimuistista. Tämän takia tulee puskurimuisti huuhdella levymuistiin ennen tietokantajärjestelmän alas ajamista ja tietokannan käsittävää tietokonelaitteisto varmentaa akustolla eli katkottomalla virransyötöllä.

## 6. Kolme eri tapaa replikoida MySQL Database Server –tuotteella

MySQL tukee kolmea replikointikäytäntöä ilman erillisen HotBackup-ohjelmiston hankkimista:

- Päivitystapahtumia sisältävän 'staattisen' tietokannan replikointi isännältä orjalle,
  - tapahtuma-alttiin isäntäkannan replikointi orjalle ja
  - ketjutettu replikointi isäntä-orja –tyyppisesti.
- Kaikkissa kolmessa vaihtoehdossa voi orja ottaa isännän tehtävät isännän kuoltua.

Kukin replikointikäytäntö perustuu isäntä-/palvelinlaitteistolla ylläpidettävään kirjanpitoon (i. ts. so. binääriloki) tietokantaan kohdistuvista muutoksista. Binääriloki siirretään isännältä orjalle käsiteltäväksi ja binäärilokeista pidetään kirjaa, jonka mukaan lokikierto on jäljitettävissä.

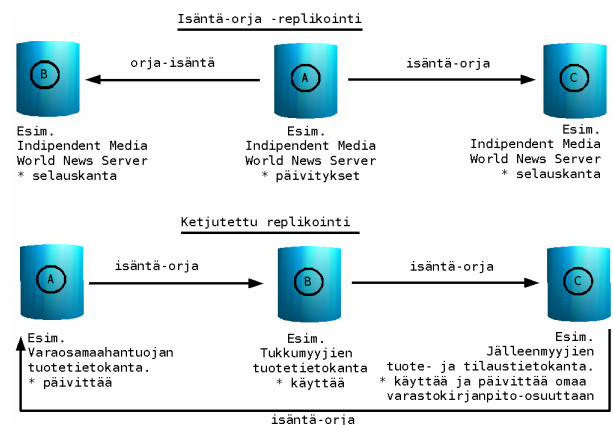
### Päivitystapahtumia sisältävän tietokannan kopiointi

Tässä replikointitavassa isäntäkannan binäärilokit kopioidaan orjalle ja replikaatin käynnistyttyä se päivittyy binäärilokien avulla ajan tasalle. Tietokantaan ei koskaan kohdistu muutostapahtumia.

### Tapahtuma-alttiin kannan kopiointi

Tapahtuma-alttiilla replikoinnilla tarkoitetaan määrätyn isäntäkannan kautta muille suoritettavaa, keskitettyä tapahtumakäsittelyä, jossa tapahtumien suoritusjärjestys pysyy muuttumattomana.

Tietoalkioon kohdistuvien muutosten lisäksi tässä replikointityypissä huomioidaan myös toipuminen ei toivotuista tilanteista, kuten keskeytynyt tapahtumakäsittely.



Kuva 2 Staattisen kannan ja ketjutetun kannan replikointi.

Mallissa yksi tietokannoista toimii isäntänä ja muut orjina. Isäntä pitää yllä numeroituja kiertäviä lokeja, mihin se kirjoittaa orjille lähetettävät tapahtumapäivitykset.

Kirjautuessaan isäntään orja kertoo isännälle edellisen onnistuneen päivityksen jälkeisen sijainnin tapahtumapäivitysten suhteen. Tämän jälkeen orja päivittää tietonsa ajan tasalle ja jää odottamaan isännän ilmoitusta uusien päivitysten saapumisesta.

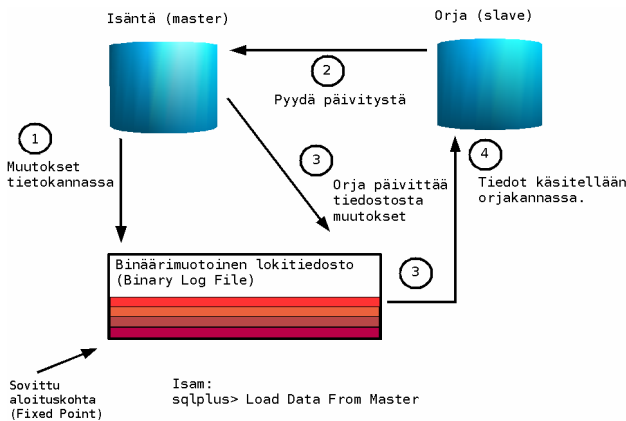
### Ketjutettu replikointi

Palvelinten tietokannat on kytketty ketjutetussa replikoinnissa isäntä-orja –tyylisesti kehään, missä orja on isäntäkannan kopion.

Orjien replikaatteihin ei saa kohdistua transaktioita, vaan ne hoidetaan isäntäkannan kautta.

## 7. MySQL Database Server –tuotteessa olevan replikointikäytännön tekninen kuvaus

1. Isäntätietokanta kirjaa binäärilokiin kaikki tietokantaan kohdistuvat muutokset. Binääriloki sisältää tietystä ajanhetkestä eteenpäin tietokantaan kohdistuneet muutokset (Fixed Point). Orjietokanta päivittää omat tietonsa ajantasalle isännän binäärilokin perusteella.
2. Orjakannan tapahtuma-aloituspiste synkronoidaan isäntäkannan kanssa, jonka jälkeen orjakannan käynnistysparametrit määritetään oikein. Käynnistettäessä orjakanta päivittyy sen sisältämä tieto isäntäkannan binäärilokien pohjalta. Orja kytkeytyy isäntään ja pyytää aika-ajoin uusia päivityksiä sisältäviä binäärilokeja käsiteltäväkseen. Yhteyden katketessa orja pollaa aika ajoin isäntää kytkeytyäkseen siihen uudelleen ja pyytääkseen käsittelemättä olevat binäärilokit prosessoitavaksi.
3. Binääriloki siirretään orjan käsiteltäväksi (jälkimmäinen nuoli kuviossa – ensimmäinen käytäntöä selventävä). Binäärilokitiedostot on kirjattu juoksevin numeroin ja ne ovat kiertolokeja.
4. Viimeisessä vaiheessa orja päivittää pyytämänsä ja saamansa binäärilokin tietojen perusteella oman kantansa ajan tasalle isäntäänsä nähden. Kukin orja kirjaa mitä binäärikopioversiota ja tapahtumaa versiossa se käsittelee.



Kuva 3 Replikointi MySQL Database Server -tuotteessa.

## 8. Replikoinnin implementointi tuotteessa

MySQL Database Server –tuotteessa orja luo käyttöönsä prosessin, jossa toimivat I/O- ja SQL-säie. Isäntätietokanta luo käyttöönsä prosessin, jonka alla toimii Binlog Dump –säie.

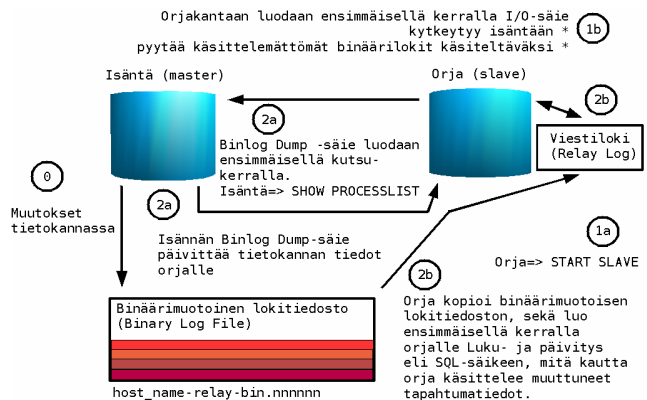
Annettaessa orjakannalle komento START SLAVE luo se itselleen prosessin ja sen alle I/O-säikeen, minkä tehtävänä on kytkeytyä ja ylläpitää yhteyttä isäntätietokantaan, sekä pyytää isännältä kopio käsittelyyn tulevista binäärilokeista (kuva; 1a ja 1b).

Ensimmäisellä orjakannan generoimalla kutsukerralla isäntä luo itselleen prosessin ja sen alle Binlog Dump -säikeen, minkä tehtävänä on vastata orjan I/O-säikeen pyyntöihin uuden binäärilokin toimittamisesta orjietokannan käsiteltäväksi. (kuva; 2a ja 2b)

I/O-säie lukee Binlog Dump -säikeen lähettämän tiedon ja tallentaa vastaanottamansa tiedostot orjan viestiloki – tietohakemistojen alle.

Orja luo itselleen binäärilokien käsittelyä varten SQL-säikeen.

Orja toimii kahden toisistaan toiminnallisesti riippumattoman ja erillisesti säikeen avulla, jolloin toisen tehtävästä suoriutumisenopeus ei vaikuta toisen säikeen toimintaan; SQL-säie pystyy hoitamaan tapahtumapäivitykset orjalle ilman tapahtumakäsittelyn nopeuden vaikutusta binäärilokitiedostojen levyllä kirjoitusnopeuteen. I/O-säie voi kommunikoida isännän Binlog Dump –säikeen kanssa samanaikaisesti SQL-säikeen prosessoidessa edeltävän binäärilokin sisältämiä tapahtumatietoja.



Replikointia varten tarvitaan kolme säiettä: Binlog Bump -säie (2a), I/O- (1b), sekä Luku- ja päivitys -säie (2b).

Kuva 4 Replikoinnin implementointi tuotteessa

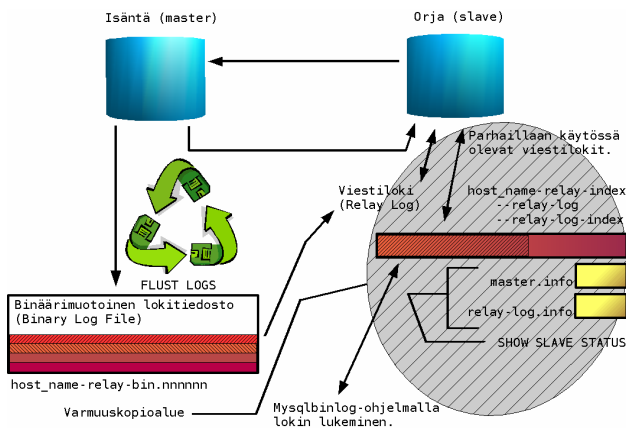
## Replikointiprosessien tarkastelu

Replikointiprosesseja ja säikeiden tilaa tutkitaan antamalla MySQL-komentorivieritorille `SHOW PROCESSLIST` -komento. Näin käyttäjä saa kaiken Binlog Dump-säiettä koskevan tiedon isäntäkannalta ja vastaavasti I/O- ja SQL-säiettä koskevan tiedon orjakannalta. Isäntä- ja orjasäikeillä on omat statuksensa, joiden perusteella käyttäjä tietää missä prosessin vaiheessa replikointi on.

## Binääriloki

Binäärilokiin tallentuvat kaikki tietokantaa tai sen sisältämää tietoa muuttavat lausekkeet. Lokiin talletetaan lisäksi lauseen suoritusta koskevia tilastotietoja (esim. suoritus aika).

Binäärilokia tarvitaan tietokannan palauttamisessa edeltävästä varmuuskopiosta sen nykyiseen tilaansa tai mahdollisimman lähelle viimeksi tietokantaan kohdistettua tapahtumaa, sekä tietokantatapahtumien siirtämiseksi isäntäkannalta orjakannalle.



Kuva 5 Binäärilokien käsittely orjalla.

## 9. Mitä muuta raportti käsittelee?

Loppuosa raporttia käsittelee tietokantakoneen parametrisointia ja replikoinnin käynnistämistä. Raportti kertoo lyhyesti parametritiedostojen sijainnista, parametreista, mysq-d-ohjelmasta, sekä luo lyhyen katsauksen käytävissä oleviin replikointikomentoihin isäntä- ja orjainstanssilla, ja lopuksi käsittelee mallitapauksena World-tietokannan parametrisoinnin ja replikoinnin isännältä orjalle. Lopussa tarkastellaan käynnistysparametreja ja tehdään raportista yhteenveto.

## 10. Loppusanat

Raportin tekijän on saavuttanut tutkimukselle asettamansa tavoitteen selvittää MySQL Database Server -tuotteen replikoinnin tekninen toiminta, sekä osoittanut myöhemmin toistettavissa olevan koneen avulla ymmärtäneensä asian.

<sup>i</sup> Curatores Ltd. / Åke Tyvi, Replikointi MySQL Database Server -tuotteella -alustava selvitys -; Omakustanne, 6/2005; ISBN 952-5494-02-2.