Universitatea “Politehnica” din București

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

**Studiul bazelor de date distribuite in sistemul de gestiune al bazelor de date MariaDB**

**Proiect de diplomă**

prezentat ca cerință parțială pentru obținerea titlului de

Inginer în domeniul Calculatoare și Tehnologia

Informației

programul de studii de licență Ingineria

Informației (CTI – INF)

Conducător științific Absolvent

S.l. Ing. Valentin PUPEZESCU Bogdan - Petru MATRAGOCIU

Anul

2014

Cuprins

1. Introducere
2. Baze de date

Introducere

* 1. Scurt istoric
  2. Noțiuni generale
  3. Tipuri de date

1. MariaDB
   1. Introducere
   2. Scurt istoric
   3. Concepte
   4. MariaDB vs. MySQL
   5. MariaDB vs. NoSQL
   6. Replicarea
   7. Avantaje si dezavantaje
2. Java
   1. Introducere
   2. Scurt Istoric
   3. Sintaxa Java
   4. OOP
   5. Integrarea cu baza de date
3. Tehnologii user-interface
   1. HTML
      1. Introducere
      2. Structura unei pagini HTML
      3. Tag-uri, elemente si atribute
   2. CSS
      1. Introducere
      2. Selectori in CSS
      3. Proprietăți CSS
   3. JavaScript
   4. jQuery
4. Proiectarea aplicației
   1. Prezentare generala a interfeței
      1. Secțiunea X
      2. Secțiunea Y
   2. Structura bazei de date
   3. Testarea replicării
5. Concluzii
6. Bibliografie
7. Anexe

Replicarea

Replicarea in MariaDB se realizează la fel a in MySQL. Replicarea este fundația pentru a construi aplicații mari de performanta înalta folosind MariaDB. Replicarea permite configurarea unui server sau a mai multor servere, replici a altui server. Acest lucru nu este valabil doar pentru aplicațiile de înalta performanta dar si pentru rezolvarea altor problem cu mar fi, sharing-ul de date cu mașini aflate la distanta, păstrarea unei copii de rezerva sau păstrarea unei copii pentru testare sau învățare (training).

In continuare for fi analizate aspectele toate aspectele replicării. Pentru început va fi prezentat modul de funcționare, apoi configurarea unui server, proiectarea unor configurații mai avansate de replicare, management-ul si optimizarea serverelor replicate.

Problema replicării

Problema replicării se rezolva prin punerea la dispoziție a unui server care sa fie sincronizat cu un altul. Mai multe salve-uri se pot conecta la un singur master, iar un salve se poate comporta la rândul sau ca un master. Master-ii si salve-urile pot fi aranjați in diferite topologii. Se poate replica întregul server, doar anumite baze de date sau se pot alege doar anumite tabele pentru replicare.

MariaDB suporta 2 tipuri de replicare : replicare „statement-based” sau replicare „row-based. „Statement-based” sau replicare logica este disponibila de la MySQL 3.23 si este cea mai folosita in acest moment. Replicare „row-based” este valabila de la MySQL 5.1. Ambele tipuri de replicare salvează schimbările efectuate pe baza de date in așa numitul „binary log” iar acest log este rulat pe slave. Ambele tipuri de replicare sunt asincrone, asta însemnând ca datele copiate pe slave nu au certitudinea ca sunt actualizate la un anumit moment de timp. Nu exista nicio garanție asupra timpului de actualizare a datelor pe slave. Slave-ul poate poate garanta actualizarea datelor după secunde, minute sau chiar ore daca query-urile au dimensiuni mari.

Replicarea MySQL este ca mai compatibila cu versiunile anterioare. Asta înseamnă ca un server cu o versiune mai noua poate fi slave pentru un server cu o versiune mai veche fără probleme. Totuși versiunile mai vechi in principiu nu pot fi slave pentru un server cu o versiune mai noua (nu pot înțelege funcționalități noi apărute si pot fi diferențe intre formatul de fișier folosit pentru replicare). Replicare nu presupune un overhead mare pe master, presupune doar activarea funcției de „binary logging”. Fiecare slave poate adăuga o încărcare suplimentara dar infima datorita operațiilor I/O de rețea.

Replicarea este in general folositoare pentru a scala citiri care pot fi direcționate către un slave, dar nu este o modalitate buna de a scala scrieri, doar in cazul in care este făcută așa cum trebuie. Atașarea multor slave-uri la un master presupune scrierea datelor pe fiecare din mașini. Fiecare sistem este limitata la viteza de scriere pe care o are cea mai slaba mașina.

Replicare este ineficienta când există mai mult de câteva slave-uri deoarece se duplica date fără rost. De exemplu un singur master cu 10 slave-uri va stoca 11 copii a acelorași date si va duplica mare parte din date in 11 cache-uri diferite. Acesta este analog cu „11 way RAID to 1” la nivel de server care specifica faptul ca acest timp de arhitectura nu este un mod de a utiliza eficient aparatura hardware. Totuși acest tip de arhitectura poate fi întâlnit destul de des. In continuare vor fi discutate moduri de a evita aceasta problema.

Probleme rezolvate de replicare

Cele mai întâlnite utilizări ale replicării:

* Distribuția datelor (Data distribution)

Replicare in MariaDB si MySQl nu folosește de obicei foarte mult din lungimea de banda a rețelei si poate fi oprita sau pornita la cerere. Prin urmare este recomandată păstrarea unei copii a datelor într-un loc distant din punct de vedere geografic cum ar fi alt centru de date. Slave-ul de la distanta poate lucra chiar cu o conexiune intermitenta (intenționată sau nu). Totuși daca este necesara o replicare fără timp de răspuns mare va fi nevoie de o conexiune stabila.

* Distribuirea încărcării (Load balancing)

Replicare poate ajuta la distribuția query-urilor de citire pe ,ai multe servere, lucru ce funcționează foarte bine pe aplicații cu operații intensive de citire. Se poate efectua „load balancing” doar cu câteva linii de cod. Pe scala mica se poate utiliza abordarea simplista cum ar fi nume de host „hardcoded” sau DNS round-robin (care face ca mai multe IP uri sa pointeze către aceeași gazdă). Exista si abordări mai sofisticate. Soluții standard mai costisitoare sunt produsele pentru distribuție a încărcării pe rețea care funcționează bine împreună cu serverele MySQL. Proiectul server virtual Linux (LVS) se pliază pe aceasta situație.

* Rezerva (Backup)

Replicarea este o tehnica buna pentru a păstra o copie de rezerva a datelor. Însă un slave nu este prin definiție un backup.

* Disponibilitate mare si failover

Replicarea ajuta la evitarea problemei ce presupune prăbușirea aplicației daca exista un singur punct care daca pica poate face aplicația inutilizabila (defectare hardware a mașinii ce tine serverul de baze de date). Un sistem bun de failover este deținerea de slave-uri replicate ce pot lua locul master-ului in caz de defectare.

* Testarea actualizărilor (updates) aduse sistemului de gestiune de baze de date

Este o practica des întâlnită setarea unui slave care are o versiune actualizata de MariaDB/MySQL si folosirea ei pentru a testa buna funcționalitate înainte de a actualiza toate sistemele.

Cum funcționează replicarea

Modul in care sunt replicate datele este prezentat in continuare. Văzută de la un nivel înalt replicarea presupune 3 pași:

1. Masterul înregistrează schimbările in fișierul de „binary log”. (Aceste modificări se numesc evenimente ale „binary log”)
2. Slave-ul copiază fișierul „binary log” de la master in log-ul sau „relay log”.
3. Slave-ul reia evenimentele din „relay log”, aplicând modificările pe datele sale.

Aceasta este vederea de ansamblu. Fiecare pas este destul de complex. In continuare va fi prezentata replicarea mai in detaliu.

Prima parte a procesului este cea de „binary logging” care are log pe master. Înainte ca orice tranzacție sa înceapă sa modifice date pe master, master-ul înregistrează schimbările in „binary log”. Tranzacțiile sunt scrise serial chiar daca ele au fost intercalate in momentul execuției lor. După această scriere in log, masterul indică motorului de stocare (storage engine) să salveze (to commit) tranzacțiile.