

TÖL303G

Gagnasafnsfræði

Vikublað 11

Snorri Agnarsson

31. október 2024

Efni vikunnar

Höldum áfram að fjalla um þýðingu og ágæðun fyrirspurna í gagnasafnskerfum. Byrjum að ræða um færslur (*transactions*) og tengt efni svo sem hreyfingaskrár (*log*) og læsingar. Lesið aftur kafla 1.2.4 og lesið 6.6. Kannski byrjum við að ræða þriggja laga högun upplýsingakerfa.

Verkefni

Finnið Gradiance verkefni á venjulegum stað, verkefni vikunnar heita „Verkefni 11“.

Verkefnið sem skila skal í Gradescope er eftirfarandi.

Klárið eftirfarandi Java forrit og keyrið það á nokkra mismunandi vegu. Þennan forritstexta má finna í Canvas í skránum `V11.java-beinagrind` og `V11.java-beinagrind-utf8`.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;

// Notkun: java -cp .;sqlite-jdbc-....jar V11 <args>
//          þar sem <args> er: [autocommit|noautocommit] [index|noindex]
// Eftir:   Búið er að mæla tíma fyrir gagnagrunnsaðgerðir og
//          skrifa niðurstöður
public class V11
{
    public static void main( String[] args )
```

```

throws Exception
{
    Class.forName("org.sqlite.JDBC");
    boolean USE_AUTOCOMMIT = args[0].equals("autocommit");
    boolean USE_INDEX = args[1].equals("index");

    Connection conn = null;
    try
    {
        conn = DriverManager.getConnection("jdbc:sqlite:v11.db");
        conn.setAutoCommit(USE_AUTOCOMMIT);

        /* Hér vantar Java forritstexta sem gerir eftirfarandi:
         * 1. Eyðir töflunni R ef hún er til, þ.e. (í SQLite)
         *      DROP TABLE IF EXISTS R
         * 2. Eyðir vísaskránni RINDEX ef hún er til, þ.e.
         *      DROP INDEX IF EXISTS RINDEX
         * 3. Býr til töflu R sem hefur dálk key af tagi INTEGER
         *      sem skal vera lykill, og dálk value af tagi DOUBLE, þ.e.
         *      CREATE TABLE R( key INTEGER PRIMARY KEY, value DOUBLE )
         * 4. Býr til vísaskrá RINDEX fyrir dálkinn value í R, þ.e.
         *      CREATE INDEX RINDEX ON R(value)
         *      en aðeins ef USE_INDEX er satt.
         * 5. Býr til PreparedStatement pstmt til að setja gildi í R, þ.e.
         *      INSERT INTO R VALUES(?,?)
         */

        long start,end;

        start = System.nanoTime();
        int i;
        for( i=0 ; i!=1000000 ; i++ )
        {
            /*
             * Hér vantar Java forritstexta til að bæta við tvennd (i,y)
             * í R þar sem y skal vera slembitala jafndreifð á bilinu
             * [0,1[. Kallið Math.random() skilar slíkum tölum.
             * Notið pstmt til að framkvæma innsetninguna.
             * Einnig skal fylgjast með tímanum með kalli á
             * System.nanoTime(). Ef meira en ein mínúta er liðin síðan
             * lykkjan hófst skal hætta í lykkjunni.
             *
             * Takið eftir að ef USE_AUTOCOMMIT er satt þá gerist sjálfkrafa
             * COMMIT eftirhverja SQL aðgerð í þessari lykkju.
             */
        }
        if( !USE_AUTOCOMMIT ) conn.commit();

        end = System.nanoTime();
        System.out.println("Tími fyrir "+
                           i+" innsetningar: "+
                           (double) (end-start)/1e9
                           );

        System.out.println("Tími per innsetningu: "+
                           (double) (end-start)/1e9/i
                           );

        start = System.nanoTime();
    }
}

```

```

        ResultSet r =
            stmt.executeQuery
            ("SELECT COUNT(*) FROM R WHERE "+
             "value BETWEEN 0.01 AND 0.10"
            );
        r.next();
        System.out.println("Niðurstaða leitar: "+r.getInt(1));
        System.out.println("Tími fyrir leit: "+
            (double) (System.nanoTime()-start)/1e9
            );
    }
    catch(SQLException e)
    {
        System.err.println(e.getMessage());
    }
    finally
    {
        try
        {
            if(conn != null)
                conn.close();
        }
        catch(SQLException e)
        {
            System.err.println(e);
        }
    }
}
}

```

Skilið eftirfarandi í Gradescope:

- Öllum forritstexta fyrir ykkar endanlega Java forrit.
- Eftirfarandi töflu, útfylltri með viðeigandi tölum til að sýna tímann sem það tekur fyrir hverja innsetningu í töfluna R. Helst skal sýna meðaltíma fyrir hverja innsetningu af 1.000.000 innsettum röðum. Það er hins vegar ekki alltaf raunhæft. Ef í ljós kemur að það er ekki raunhæft þá skuluð þið láta texta fylgja með til að útskýra hvernig tölurnar eru reiknaðar. Notið viðeigandi einingar í töflunni, til dæmis sekúndur (s), nanósekúndur (ns), millisekúndur (ms), míkrósekúndur (μs), mínútur (mín), klukkustundir (klst). **Munið að taflan á að sýna meðaltíma per aðgerð, ekki heildartíma.**

Tími fyrir innsetningu			
Án vísis		Með vísi	
Án AutoCommit	Með AutoCommit	Án AutoCommit	Með AutoCommit

- Einnig eftirfarandi töflu, útfylltri með viðeigandi tölum. Ef ykkur finnst þörf á getið þið breytt forritinu þannig að margar leitir séu framkvæmdar til að hægt sé að reikna meðaltíma. Þá þarf hins vegar að sjá til þess að ekki sé aftur og aftur verið að framkvæma sömu leitina.

Tími fyrir leit	
Án vísis	Með vísí