Tietokantakyselyjen optimointi relaatiotietokannassa

Olli Rissanen

Kandidaatintutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Helsinki, 17. maaliskuuta 2013

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Olli Rissanen

Työn nimi — Arbetets titel — Title

Tietokantakyselyjen optimointi relaatiotietokannassa

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma	17. maaliskuuta 2013	9

Tiivistelmä — Referat — Abstract

Tutkielmassa tutustutaan tietokantakyselyjen optimointiin relaatiotietokantojen hallintajärjestelmien osalta sekä optimoinnin vaikutukseen kyselyjen suorituskyvyssä.

Avainsanat — Nyckelord — Keywords
Information systems Query optimization

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Sisältö

1 Johdanto

Modernit järjestelmät lisäävät jatkuvasti tietokantojen työtaakkaa tiedon määrän kasvaessa. Vaikka tietokanta on vain yksi komponentti koko järjestelmästä, on se usein kriittisin ja vaihtelevin sillä tiedon lukeminen massamuistista on erittäin hidas operaatio verrattuna rekistereiden tai välimuistin käyttöön. Optimoimalla tietokantakyselyjen suoritusta voidaan vaikuttaa suoritettujen operaatioiden määrään sekä muistialueen kokoon ja siten vähentää tietokannan vasteaikaa sekä resurssien käyttöä.[1]

Tietokannan hallintajärjestelmä on kokoelma ohjelmia tiedon tallentamiseen, muokkaamiseen, analysointiin ja keräämiseen tietokannasta. Hallintajärjestelmää käytetään kyselykielellä, joista esimerkiksi SQL on suunniteltu relaatiotietokantojen hallintajärjestelmille. Kyselyn optimointi on toteutettu automaattisena toimenpiteenä tietokannan hallintojärjestelmän sisältämässä kyselyoptimoijassa, ja se on ydintekijä erityisesti relaatiomalliin pohjautuvien hallintajärjestelmien menestyksessä. Kaikista hallintajärjestelmän komponenteista kyselyoptimoijalla on suurin merkitys tietokannan suorituskykyyn. Kyselyoptimoijan tavoitteena on minimoida itse optimointiin käytetty aika ja maksimoida optimoinnista saatu hyöty.[2]

Kyselyoptimoija toimii etsien kyselyä vastaavat kyselysuunnitelmat ja valitsemalla niistä tehokkaimman. Kyselysuunnitelma sisältää sarjan algebrallisia operaatioita tietokannan relaatioille jotka tuottavat tulokseksi halutun vastauksen. Tietokantakyselyä vastaavia kyselysuunnitelmia voi olla useita, sillä kyselyjen algebralliset esitykset voidaan usein esittää monena loogisesti vastaavana esityksenä. Algebrallista operaatiota kohden voi myös löytyä useita toteutuksia, kuten join-operaatiota toteuttavat merge join ja hash join. Toisiaan vastaavat esitykset voivat olla suorituskyvyltään jopa eri asteikolla.

Haasteeksi nousee kyselysuunnitelman luominen ja kyselysuunnitelmien

suorituskyvyn ennustaminen. Optimointi on vaikea hakuongelma, jossa hakualue voi nousta erittäin suureksi kyselyn ollessa monimutkainen.[3] Optimoijan tulee valita pienin mahdollinen hakualue, joka pitää sisällään halvimmat suunnitelmat. Suorituskyvyn ennustamisen ja hakualueen rajauksen lisäksi optimoija tarvitsee tehokkaan algoritmin koko hakualueen läpikäymiseen. On epärealistista odottaa kyselyoptimoijan aina löytävän parhaan kyselysuunnitelman, ja onkin tärkeämpää välttää huonoimpia suunnitelmia ja löytää hyvä suunnitelma. [4]

Tutkielman rakenteesta

2 workname: Taustaluku

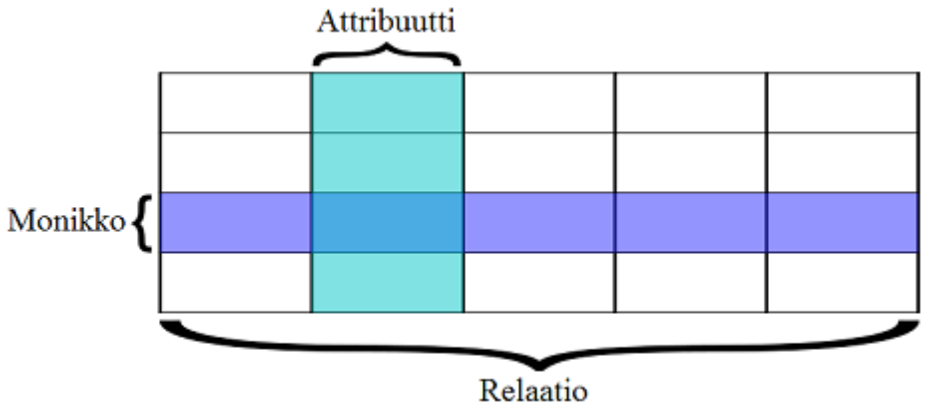
2.1 Relaatiomalli

Relaatiotietokanta on relaatiomalliin perustuva tietokanta. Relaatiomallin keskeinen piirre on kaiken datan esittäminen n-paikkaisen karteesisen tulon osajoukkona, ja se tarjoaa deklaratiivisen menetelmän datan ja kyselyjen määrittämiseen. Relaatiomalli koostuu attribuuteista, monikoista ja relaatioista. Matemaattisessa määritelmässä attribuutti on pari joka sisältää attribuutin nimen ja tyypin sekä jokaiseen attribuuttiin liittyy sen arvojoukko. Monikko on järjestetty joukko attribuuttien arvoja. Relaatio koostuu otsakkeesta ja sisällöstä, jossa otsake on joukko attribuutteja ja keho on joukko monikkoja. Relaation otsake on myös jokaisen monikon otsake. Visuaalisessa esityksissä relaatio on taulukko ja monikko taulukon rivi.

2.2 Kyselyoptimoijan tavoitteet

Tietokantakyselyjen optimoinnilla viitataan tietokantakyselyn suorittamiseen mahdollisimman tehokkaasti. Optimoinnin tavoitteena on joko maksimoida

Kuva 1: Relaatiomalliin perustuva tietokanta



suorituskyky annetuilla resursseilla tai minimodia resurssien käyttö. Mitattavia resursseja ovat suorittimen ja muistin käyttö sekä kommunikointikustannukset. Muistin käyttö jakautuu tallennuskustannukseen sekä ulkomuistiin pääsyn kustannukseen. Tallennuskustannuksella tarkoitetaan ulkomuistin sekä puskurimuistin käyttöä, ja se tulee aiheelliseksi kun muistin käyttö aiheutuu pullonkaulaksi.

Resurssin merkitys riippuu tietokantatyypistä. Hajautetuissa tietokannoissa hitailla yhteysväylillä kommunikointikustannukset hallitsevat kustannuksia. Paikallisesti hajautetuissa tietokannoissa kaikilla resursseilla on sama painoarvo. Keskitetyissä tietokannoissa ulkomuistiin pääsyn kustannus ja prosessorin käyttö ovat oleellisia. Tämän tutkielman aihepiiriin kuuluu vain keskitettyjen tietokantojen optimointi.

2.3 Kyselyoptimoijan toiminta

Tähän SQL-blokki: nested quotet erikseen.

Tietokannan hallintajärjestelmän suorittama SQL-kyselyn prosessointi sisältää useita vaiheita. Aluksi hallintajärjestelmä jäsentää kyselyn syntaktisen ja semanttisen tarkastelun mahdollistamiseksi. Tämän jälkeen kysely muu-

tetaan jäsennetyksi puuksi, joka on kyselyn algebrallinen esitysmuoto. Puun alkiolla on nolla tai enemmän aliisolmua, joiden ulostuloa (output) käytetään ylisolmun (parent node) syötteenä. Esimerkiksi join-operaatiossa solmulla on kaksi alisolmua, joille toteutetaan join-operaatio. Sort-operaatiolla on yksi alisolmu joka järjestetään. Lehtisolmut ovat solmuja jotka suorittavat hakuja (scan) levyltä ja palauttavat saadut tulokset. [5]

todo:validoi ja siisti: Seuraavaksi ysely uudelleenkirjoitetaan (rewrite) valitsemalla hakumetodit (access method), liittämisjärjestyksen (join orders) ja liittämistavat (join methods) tietokannan käyttämien heuristiikkojen pohjalta. Monimutkaisille kyselyille sovelletaan myös muunnossääntöjä. [5]

Seuraava vaihe on kyselyn optimointi. Kyselyoptimoijan sisääntulo(input) sisältää kyselyn, tietokannan mallin sisältäen taulujen ja indeksien määritelmät sekä tietokannan tilastoja. Kyselystä haetaan predikaatit, indeksit ja liitokset (joins) kyselysuunnitelmaa varten. Tämän jälkeen luodaan kyselyä vastaavat kyselysuunnitelmat ja arvioidaan niistä paras. Seuraavassa vaiheessa kysely suoritetaan käyttämällä optimoijan tuottamaa kyselysuunnitelmaa.

todo: SQL ja relaatiomalli

todo: relaatiotietokanta vs no-sql

todo: optimoi menee

3 Kyselysuunnitelman suorituskyvyn arviointi

Kyselyoptimoijan tulee arvioida kustannus jokaiselle kyselysuunnitelmalle. Kustannusarviointi koostuu kahdesta vaiheesta: ensiksi arvioidaan jäsennetyn puun jokaisen alkion operaation suorittamiseen kuluva aika. (pipelining, temppirelaatiot.) Tämän jälkeen arvioidaan jokaisen alkion tulosjoukon koko, sekä lisäksi tarkastetaan onko tulosjoukko järjestetty. Solmun tulosjoukko on ylisolmun syöte, joten sen koko ja järjestys vaikuttavat suoraan ylisolmun

arviointiin.

3.1 Tulosjoukon koon arviointi

Operaation kustannus riippuu syötteen koosta. Tutkitaan seuraavaa tapausta:

SELECT attribuuttit
FROM relaatiot
WHERE ehto 1 $\wedge$ ehto 2 $\wedge$ ... $\wedge$ ehto n

Kyselyn palauttamien monikkojen maksimimäärä on relaatioiden karteesinen tulo. Jokainen WHERE-ehto harventaa monikkojen määrää. WHERE-ehdon vaikutusta tulosjoukon kokoon voidaan mallintaa lisäämällä jokaiseen ehtoon vähennyskerroin, joka on oletettu suhde lähtöjoukosta tulosjoukkoon vain kyseisen ehdon osalta. Tulosjoukon koko voidaan siten arvioida kertomalla maksimijoukko vähennyskertoimien tulolla. [4]

WHERE-lauseen ehtojen kertoimia voidaan laskea hyödyntämällä tietohakemistoon tallennettua tilastotietoa.

$$column = value \tag{1}$$

-tyyppiselle ehdolle vähennyskerroin voidaan arvioida kaavalla

$$\frac{1}{NKeys(I)} \tag{2}$$

jos sarakkeessa on indeksi I kyseiselle relaatiolle. Ilman indeksiä kyselyoptimoija käyttää kiinteää arvoa vähennyskertoimen arvioimiseen, kuten 1/10.

3.2 Tietohakemistoon tallennettu tilastotieto

Kustannusten arviointi vaatii tilastotietoa. [4]

todo: täsmennä ja siisti (sis. tietohakemistoon tallennetun tilastotiedon käytön) Tilastotieto on.. Tilastotiedoilla lasketaan kyselysuunnitelmien kustannusarvio ja valitaan pienimmän kustannusarvion omaava suunnitelma. Tietoa ei tallenneta jatkuvasti, sillä se aiheuttaisi esimerkiksi rinnakkaisuusongelmia. Tietokannan valvoja triggeriöi päivityksen esim. komennolla optimize db. Tilastotiedon harvan päivityksen takia valituksi ei aina välttämättä tule tehokkain suunnitelma.

4 Algebralliset muunnoslait (logical optimization)

5 Fyysinen optimointi (physical optimization)

6 Kyselyoptimoijan käyttäminen (hints)

7 case study aiempaan liittyen?

8 yhteenveto

Lähteet

[1] Jyoti Mor, Indu Kashyap, and RK Rathy. Analysis of query optimization techniques in databases. *International Journal*, 47, 2012.

[2] Matthias Jarke and Jurgen Koch. Query optimization in database systems. *ACM Computing surveys (CsUR)*, 16(2):111–152, 1984.

[3] Surajit Chaudhuri. An overview of query optimization in relational systems. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-*

SIGART symposium on Principles of database systems, pages 34–43. ACM, 1998.

[4] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill international editions: Computer science series. McGraw-Hill Education, 2003.

[5] S.M. Mahajan and V.P. Jadhav. An analysis of execution plans in query optimization. In *Communication, Information Computing Technology (ICCICT), 2012 International Conference on*, pages 1–5, Oct.