Tietokantakyselyjen optimointi relaatiotietokannassa

Olli Rissanen

Kandidaatintutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 10. 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	10. maaliskuuta 2013	6

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

Tietokantojen suorituskyky on yhä tärkeämpää tiedon määrän kasvaessa. Optimoimalla tietokantakyselyjen suoritusta voidaan helpottaa käyttäjien tiedonhakua sekä kasvattaa tietokannan suorituskykyä.[1] Erityisesti olio-relaatiokuvausta (ORM, object-relational mapping) toteuttavissa ohjelmointikielissä puhtaan SQL-kyselyjen kirjoittaminen on siirretty käyttäjältä alemmalle tasolle. ORM luo relaatiotietokannan pohjalta käyttäjälle oliotietokannan, jonka suorituskyky on kuitenkin sidottu relaatiotietokantaan. Relaatiotietokannan kyselyjen optimoinnilla pystytään siten saavuttamaan .. [viite ois kiva]

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

Kyselyoptimoija toimii etsien kyselyä vastaavat mahdolliset kyselysuunnitelmat ja valitsemalla niistä tehokkaimman. Kyselysuunnitelma sisältää sarjan algebrallisia operaatioita tietokannan relaatioille jotka tuottavat tulokseksi halutun vastauksen. Tietokantakyselyä vastaavia kyselysuunnitelmia voi olla useita, sillä kyselyiden 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.

Tutkielman rakenteesta

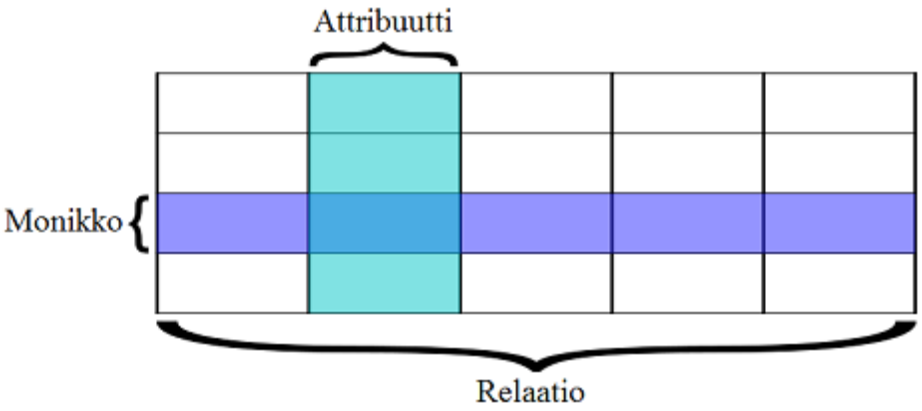
2 workname: Taustaluku

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(body?), 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.

Kuva 1: Relaatiomalliin perustuva tietokanta



2.2 Kyselyoptimoijan toiminta

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

Tietokannan hallintajärjestelmän suorittama SQL-kyselyn prosessointi sisältää useita vaiheita. Aluksi hallintajärjestelmä jäsentää kyselyn syntaktisen ja semanttisen tarkastelun mahdollistamiseksi. Tämän jälkeen kysely muutetaan jäsennetyksi puuksi, joka on kyselyn algebrallinen esitysmuoto.

[4]

todo:validoi ja siisti: Kysely uudelleenkirjoitetaan (rewrite) tämän jälkeen valitsemalla hakumetodit (access method), liittämisjärjestyksen (join orders) ja liittämistavat (join methods) tietokannan käyttämien heuristiikkojen pohjalta. Monimutkaisille kyselyille sovelletaan myös muunnossääntöjä. [4]

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

[2] [3] [5] [6] [7] [8] [9] [10] [4]

3 Tietohakemistoon tallennettu tilastotieto

todo: täsmennä ja siisti

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

5 menetelmä n

6 menetelmien vertailu

7 case study?

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] 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.

[5] Johann Christoph Freytag. *A rule-based view of query optimization*, volume 16. ACM, 1987.

[6] Kiyoshi Ono and Guy M Lohman. Measuring the complexity of join enumeration in query optimization. In *Proceedings of the 16th International Conference on Very Large Data Bases*, pages 314–325. Morgan Kaufmann Publishers Inc., 1990.

[7] Goetz Graefe. Query evaluation techniques for large databases. *ACM Computing Surveys (CSUR)*, 25(2):73–169, 1993.

[8] Francis C Chu, Joseph Y Halpern, and Praveen Seshadri. Least expected cost query optimization: An exercise in utility. *arXiv preprint cs/9909016*, 1999.

[9] A. V. Aho, C. Beeri, and J. D. Ullman. The theory of joins in relational databases. *ACM Trans. Database Syst.*, 4(3):297–314, September 1979.

[10] Richard L. Cole and Goetz Graefe. Optimization of dynamic query evaluation plans. *SIGMOD Rec.*, 23(2):150–160, May 1994.