Hoikin mahdollinen: Bittikarttaindeksi

Marja Kärmeniemi,
Oracle Finland Oy
Business Intelligence and Warehouse Consulting

Tietovarastoinnin ennakoimattomat kyselytarpeet ja suuret taulut tarkoittavat myös suurta määrää isoja indeksejä. Voi olla liian kallista ylläpitää kaikkia tietotarpeita varten sopivia paksuja b-puuindeksejä, joita Ari Hovi käsittelee artikkelissaan. Tällöin tarvitaan laihoja indeksejä, joita voidaan käyttää ennakoimattomasti. Bittikarttaindeksit ovat näistä hoikimpia.

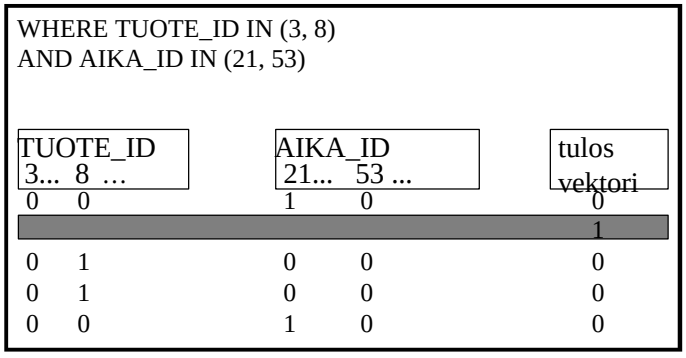
Bittikarttaindeksit muodostuvat bittivektoreista, joita on yksi kutakin erilaista sarakearvoa kohti. Vektorin yksi bitti vastaa taulun riviä siten, että arvon ollessa 1, rivillä on kyseinen arvo. Bittikarttaindeksien soveltuvuus riippuu paljolti indeksoitavan sarakkeen kardinaliteetista eli erilaisten arvojen lukumäärästä. Sarakkeen kardinaliteetti kertoo, kuinka monta erilaista arvoa siinä esiintyy. Suurin hyöty bittikarttaindekseistä saadaan, kun sarakkeen kardinaliteetti on pieni eli siinä esiintyy vähän erilaisia arvoja verrattuna taulun rivimäärään. Staattiset bittivektorit talletaan yleensä tiivistettyinä, joten kun sarakkeen kardinaliteetti on pieni, on bittikarttaindeksin koko merkittävästi pienempi kuin vastaavan b-puuindeksin koko.

Kun SQL-lauseen hakuehdossa on useampi sarake yhdistetty AND tai OR operaatiolla, voidaan niiden bittivektorit yhdistää vastaavilla boolean operaattoreilla yhdeksi vektoriksi ennen kuin tulosjoukko luetaan taulusta (kuva: bittivektorien yhdistäminen). Sarakkeille täytyy joko olla bittikarttaindeksit tai niille voidaan luodaan sellaiset dynaamisesti. Ainoastaan tulos, joka perustuu ehdon täyttävien rivien lukumäärän laskemiseen (COUNT), voidaan saada ilman, että tiedot luetaan taulusta. Tällainen COUNT-kysely onkin erittäin nopea.

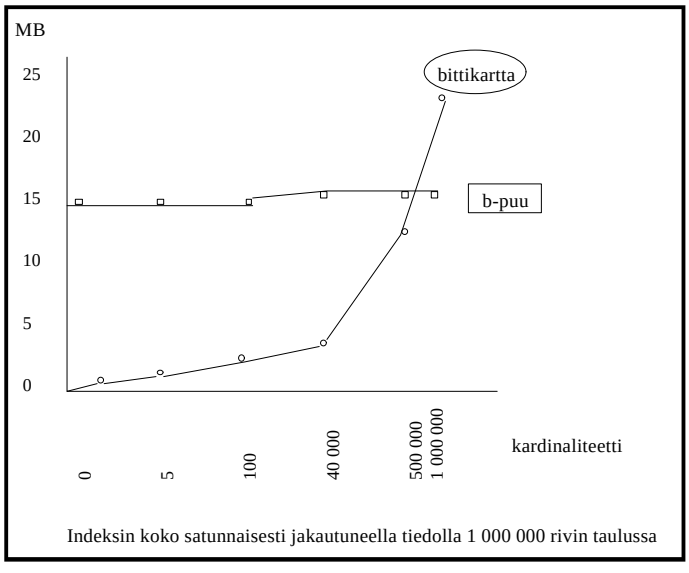
Seuraavissa kappaleissa verrataan bittikarttaindeksejä b-puuindekseihin.

Kardinaliteetti ratkaisee kokoeron

Bittikarttaindeksit ovat suositeltavia ja parhaimmilaan, kun sarakkeen kardinaliteetti on pieni. Sybase suosittelee Low_Fast indeksiä, kun sarakkeen kardinaliteetti on alle 1000. Oracle suosittelee bittikarttaindeksejä myös tauluihin, joissa kardinaliteetti voi olla huomattavasti suurempikin, jos sarakkeen arvot toistuvat yli 100 kertaa. Bittikartta sopii siis myös 1 000 000 rivisen taulun sarakkeeseen, jossa on 10 000 erilaista arvoa. Kardinaliteetin kasvaessa menetetään bittikarttaindeksien tilansäästöetu b-puuindekseihin verrattuna.



Kuva: Bittivektorien yhdistäminen



Kuva: Indeksien koko

Pieni koko on bittikarttaindeksien tärkeä etu b-puuindekseihin verrattuna, jos tarvitaan paljon indeksejä suuriin tauluihin. Keskimäärin ne vievät vähemmän kuin 25 % vastaavan b-puuindeksin vaatimasta levytilasta. Toisaalta ne voivat olla suurempia kuin vastaavat b-puut, jos kardinaliteetti on lähellä taulun rivien lukumäärää. (Kuva: Indeksien koko)

Varautuminen ennakoimattomiin kyselyihin

Kolme yhden sarakkeen bittikarttaindeksiä voi korvata jopa kuusi erilaista kolmesarakkeista b-puuindeksiä, joissa indeksin alkusarakkeiden pitää osua yhteen hakuehdon kanssa.

Pysyvät bittikarttaindeksit ovat pienempiä myös tilanteessa, jossa ennakoimatonta käyttöä varten tehtäisiin vastaavia yhden sarakkeen b-puuindeksejä, joita sitten käytettäisiin kyselyissä useampia kerrallaan. Useamman b-puuindeksin riviosoittimien (rowid, rid), jotka ovat 5-10 tavua pitkiä, lajittelu vaatii myös tilaa keskusmuistista enemmän kuin bittikarttavektorien käsittely.

Jos kyselyä varten on olemassa täsmälleen osuva b-puuindeksi eli ns. paksu indeksi, on tämä bittikarttaindeksejä nopeampi saantipolku.

Päivitysten aiheuttamat lukot

Vaikka bittikarttaindeksien päivitys onnistuukin, kuten ainakin Oraclen tietokannoissa, niin niitä ei suositella tiheästi päivittyviin tauluihin, sillä indeksiviittauksia on lukittuna enemmän kuin b-puussa johtuen indeksin rakenteesta. Tietovarastossa tämä ei yleensä ole ongelma, sillä taulut päivitetään tavallisesti lataamalla ne ajankohtana, jolloin taulut eivät ole käytössä tai käyttö on minimissään.

Bittikartat myyntituottokyselyssä

Jos esimerkkimme (kuvat: myyntituottokysely, myyntifaktat ja dimensiot) indeksit ovat bittikarttoja, voidaan indeksien kokoa ja niiden lukemiseen kuluvaa aikaa arvioida seuraavasti:

Koko: oletus 25% vastaavasta b-puusta eli 0,25* (10 bytes * 1 000 milj riviä) = 2,5 GB

Laskelmassa oletetaan, että bittikarttaindekseistä voidaan lukea vain kiinnostavat vektorit:

0,5 % tuoterivien määrästä = > 0,005 * 2,5 GB = 12,5 MB (50 tuote_id-vektoria)

5 % päivämääristä = > 0,05 * 2,5 GB = 125 MB (50 aika_id-vektoria)

Indeksien lukeminen kestää 0,5 s/MB * 138 MB = 70 s

Taulurivien lukemiseen kuluu sama aika kuin muissakin esimerkeissä eli 500 s/GB * 3 GB = 1500 s = 25 min (ei huomioitu rinnakkaiskäsittelyä).

Bittikarttaindeksejä tuotteissa

Staattiset Bittikartat

Oracle7 ja Oracle8 tietokannoissa on valittavissa bittikarttaindeksointi. Ainoa rajoitus on, että bittikartat eivät kelpaa viite-eheystarkistuksiin.

Sybase Adaptive Server IQ tietokanta perustuu bittikarttaindeksointiin. Indeksejä on useampaa tyyppiä:

*Low_Fast sarakkeille, joiden kardinaliteetti on alle 1000

*High_Group ja High_Non_Group sarakkeille, joiden kardinaliteetti on yli 1000. High_Group indeksi luodaan automaattisesti, jos sarakkeelle on asetettu PRIMARY KEY tai UNIQUE määre.

Dynaamiset Bittikartat

IBM DB2 UDB muuntaa optimoijan päätöksellä b-puuindeksit dynaamisesti bittikartoiksi. Saantipolkuna on Dynamic Bitmap Index ANDing, jossa B-puuindeksit muunnetaan bittikartoiksi yksi kerrallan liittäen ne aina edelliseen tulosvektoriin. Vektori on tiivistetty hajautin algoritmilla (hash RID).

Myös Oracle8:n optimoija voi päätyä konversioon (bitmap conversion), jos taululla on valmiiksi vähintään yksi bittikarttaindeksi, jonka ei välttämättä tarvitse kuulua valittuun saantipolkuun. Optimoija ei valitse tätä saantipolkua, jos saantipolussa käytettäisiin vain yhtä b-puuindeksiä.

Oracle Star Transformation

Oracle Star Transformation saantipolun avulla vältetään karteesinen tulo, jos faktataulun sarakkeille on käytettävissä bittikartat, kun dimensiotauluja on useita. Karteesisesta tulosta saattaa tällöin tulla liian suuri. Tähtimuunnos perustuu dimensiotaulujen lukemiseen alikyselyinä. Näitä tulosjoukkoja käytetään ehtoissa, joilla faktataulun tulosrivit etsitään bittikarttaindeksien avulla.

Esimerkissämme tähtimuunnos lisää alkuperäiseen kyselyyn

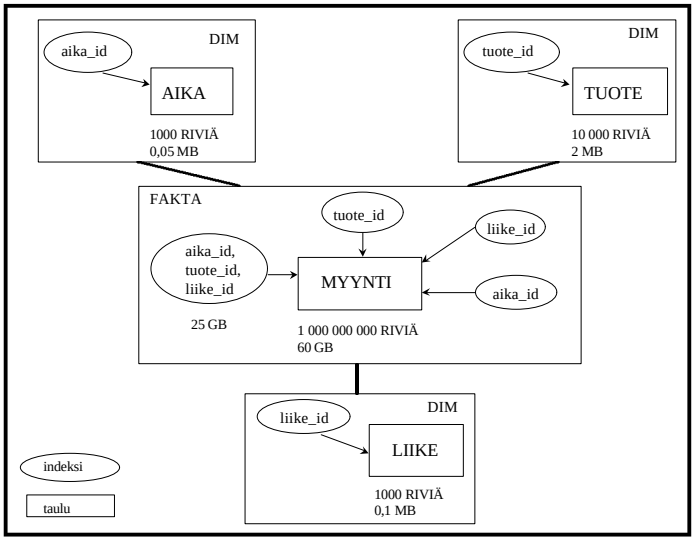
Haluamme tuottaa isosta MYYNTI faktataulusta tietyn tuoteryhmän liikekohtaisen tuottoraportin, jossa vertaamme kahden eri kuukauden myyntiä.

```
SELECT   LIIKE_NIMI,
         KUUKAUSI
         SUM(MYYNTI_MK),
         SUM(KULUT_MK),
         SUM(MYYNTI_MK - KULUT_MK) AS TUOTTO
FROM     AIKA, TUOTE, MYYNTI, LIIKE
WHERE    AIKA.AIKA_ID = MYYNTI.AIKA_ID
         AND TUOTE.TUOTE_ID = MYYNTI.TUOTE_ID
         AND LIIKE.LIIKE_ID = MYYNTI.LIIKE_ID
         AND TUOTERYHMÄ = :TUOTERYHMÄ
         AND KUUKAUSI IN (:X, :Y)
GROUP BY LIIKE_NIMI, KUUKAUSI
```

Oletuksia:
kiinnostavaan tuoteryhmään kuuluu 0,5 % tuotteista
myyntitaulun rivit ovat päivätasoisia ja kaksi kuukautta edustaa 5 % päivistä

Dimensiotaulun AIKA läpiluku kestää alle 0,5 s => 0 s
Dimensiotaulun TUOTE läpiluku kestää alle 0,5 s => 0 s

Kuva: Myyntituottokysely



Kuva: Myyntifaktat ja dimensiot

kaksi viimeistä ehtoriviä:

```
SELECT …
FROM aika, tuote, myynti
WHERE aika.aika_id = myynti.aika_id
AND tuote.tuote_id = myynti.tuote_id
AND tuote.tuoteryhmä = :tuoteryhmä
AND aika.kuukausi IN (:x, :y)
AND myynti.tuote_id IN (SELECT tuote_id FROM tuote WHERE
tuoteryhmä = :tuoteryhmä)
AND myynti. Aika_id IN (SELECT aika_id FROM aika WHERE
kuukausi IN (:x, :y))
```

Ensimmäisen alikyselyn palauttamien tuotteiden bittivektorit haetaan myyntitaulun tuotenumeroon perustuvasta bittikarttaindeksistä ja yhdistetään yhdeksi vektoriksi. Samoin tehdään päivämäärille. Sitten tämä kaksi vektoria yhdistetään ja tuloksena saadun vektorin avulla ratkaistaan oikeat riviosoitteet, joiden avulla kyselyn tulosjoukko luetaan myyntitaulusta. Jos tulosjoukossa on myös dimensiotaulujen sarakkeita, tehdään lopuksi vielä liitokset niihin.

Marja Kärmeniemi,
Oracle Finland Oy
Business Intelligence and Warehouse Consulting