



Är detta ett prestanda Case?

Christer Johansson

(v2) RUG 2006-01-25

Det tycks ibland som om SELECT satser med CASE expression tar längre tid än om det man utelämnar dem

Det är en känsla jag haft men är det så?

(Det borde ju vara så, då data måste "editeras" innan resultatet lämnas)

Peters vy över plan_table (sida 1)

```

CREATE VIEW UUUUUUU.VEXPLAIN AS
SELECT
    SMALLINT(QUERYNO)          AS QNO
    , SUBSTR(DIGITS(QBLOCKNO),5,1)
    !!' '!!SUBSTR(DIGITS(PLANNO),5,1) AS PNO
    , CASE
        WHEN MIXOPSEQ>0 THEN ' MX'
        WHEN METHOD=1 THEN
            CASE
                WHEN JOIN_TYPE = '' THEN ' NLJ'
                ELSE ' NLJ-'!!JOIN_TYPE
            END
        WHEN METHOD=2 THEN
            CASE
                WHEN JOIN_TYPE = '' THEN ' MSJ'
                ELSE ' MSJ-'!!JOIN_TYPE
            END
        WHEN METHOD=3 THEN 'SORT'
        WHEN METHOD=4 THEN
            CASE
                WHEN JOIN_TYPE = '' THEN ' HYJ'
                ELSE ' HYJ-'!!JOIN_TYPE
            END
        ELSE PT.QBLOCK_TYPE
    END AS QBLOCK
    ,SUBSTR(PT.TNAME,1,8) AS TABLE
    , CASE WHEN PT.TABNO = 0 THEN ''
        ELSE SUBSTR(DIGITS(TABNO),5,1) END AS TB
    , PT.ACCESTYPE AS AC
    , CASE
        WHEN ACCESTYPE = 'R' THEN ''
        WHEN METHOD=3 THEN ''
        ELSE
            SUBSTR(DIGITS(PT.MATCHCOLS),5,1) CONCAT
            CASE WHEN IX.COLCOUNT IS NULL THEN ''
                ELSE '/' CONCAT SUBSTR(DIGITS(IX.COLCOUNT),5,1)END
        END AS COL
    , SUBSTR(PT.ACCESSNAME,1,8) AS INDEX
    , CASE
        WHEN (ACCESTYPE = 'I' OR ACCESTYPE = 'I1'
            OR ACCESTYPE = 'N' OR ACCESTYPE = 'MX')
            THEN PT.INDEXONLY
        ELSE ''
        END AS IX
    , CASE WHEN METHOD <> 3 THEN ' ' ELSE
        CASE WHEN SORTN_JOIN='N' THEN '.' ELSE SORTN_JOIN END
        !!' '
        !!CASE WHEN SORTC_UNIQ='N' THEN '.' ELSE SORTC_UNIQ END
        !!CASE WHEN SORTC_JOIN='N' THEN '.' ELSE SORTC_JOIN END
        !!CASE WHEN SORTC_ORDERBY='N' THEN '.' ELSE SORTC_ORDERBY END
        !!CASE WHEN SORTC_GROUPBY='N' THEN '.' ELSE SORTC_GROUPBY END

```

Peters vy över plan_table (sida 2)

```

END      AS "JN_UJOG"
, SUBSTR(TSLOCKMODE,2,2)
!!!  '!!IPREFETCH
!!!  '!!COLUMN_FN_EVAL
                                AS "TS PF CF"
, CASE WHEN MIXOPSEQ=0 THEN ' '
      ELSE SUBSTR(DIGITS(MIXOPSEQ),5,1) END AS MX
-
,PT.PROGNAME
,PT.VERSION
,PT.TIMESTAMP
,PT.QUERYNO
,PT.QBLOCKNO
,PT.PLANNO
,PT.MIXOPSEQ
,PT.APPLNAME
FROM      uuuuuuu.PLAN_TABLE      AS PT
LEFT JOIN  SYSIBM.SYSINDEXES IX
ON IX.CREATOR = PT.ACCESSCREATOR
AND IX.NAME   = PT.ACCESSNAME
;

```

QNO	PNO	QBLOCK	TABLE	TB	AC	COL	INDEX	IX	JN_UJOG	TS_P
650	1	1	SELECT	PLAN_TAB	2	R				IS
650	1	2	MSJ-L	SYSINDEX	3	R				IS
650	1	3	SORT						. .Y.	

Det finns 6 case satser på första nivån

- Totalt är det 15 case satser
- 29 when villkor
- En hel del annan editering (substr,digits osv)

Funderingar

Vad händer då man ökar antalet

- enkla when villkor i en case sats?
- case satser?
- case satser i case satser?

Testmiljö

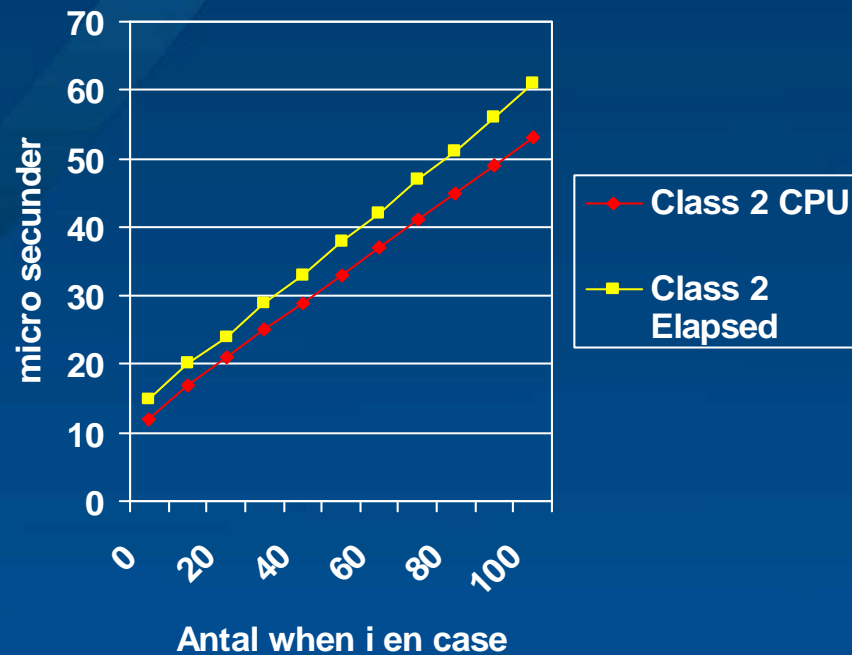
- **En testtabell med 100.000 rader (inga index)**
 - Radlängd 38 bytes
 - 1000 aktiva sidor
- **Alla körningar läser alla rader i tabellen utan sorteringar**
- **En dedikerad bufferpool om 1025 buffrar**
- **Alla (de flesta) SQLerna är körda 100 gånger med de SQLer som de skall jämföras med**
- **Alla körningar är gjorda med statisk SQL**

När vi ökar antalet when villkor i en case sats

```
,case
  when c2 = 'A' THEN 'C'
                               ELSE 'B'
end
,i99999
```

```
,case
  when c2 = 'A' THEN 'C'
  when c2 = '2' THEN 'C'
  when c2 = '3' THEN 'C'
  when c2 = '4' THEN 'C'
  when c2 = '5' THEN 'C'
  when c2 = '6' THEN 'C'
  when c2 = '7' THEN 'C'
  when c2 = '8' THEN 'C'
  when c2 = '9' THEN 'C'
  when c2 = '0' THEN 'A'
                               ELSE 'B'
end
,i99999
```

Snitt Class 2 CPU tid för en fetch



Antalet case expressions i selectsatsen

```

...
...
,s9999
,case
  when s999 > 500 then 999
  else 111
end
,case
  when s99 > 50 then 99
  else 11
end
,case
  when s9 > 5 then 9
  else 1
end
,s2
from casetab

```

Snitt tid för en fetch



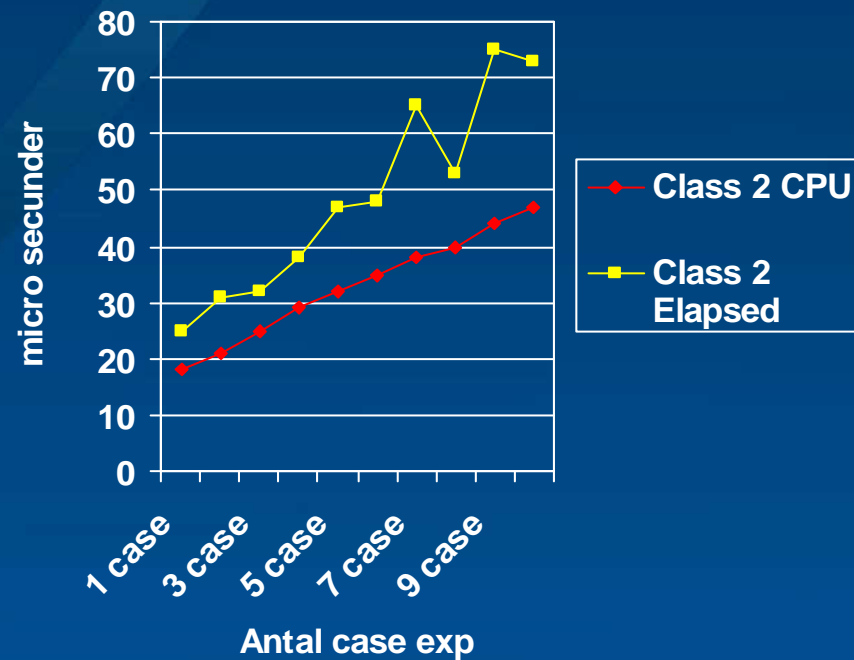
Antalet "case i case" expressions i selectsatsen

```

...
...
end
,case
  when s9 > 5
  then case
    when s2 = 0 then 2
    else 6
  end
else case
  when s2 = 0 then 3
  else 7
end
end
,s2
from casetab

```

Snitt tid för en fetch



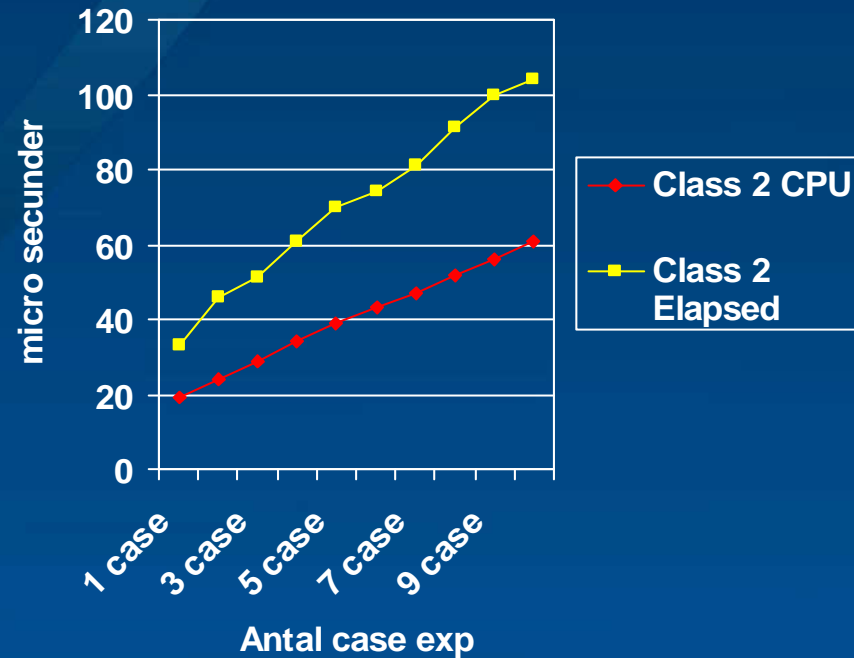
Antalet "case i case i case" expressions i selectsatsen

```

...
...
,case
  when s9 > 5
  then
    case
      when s2 = 0
      then
        case
          when i99999 > 50000
          then 1
          else 6
          end
        else
          case
            when i99999 > 50000
            then 9
            else 4
            end
          end
        end
      end
    end
  end
...

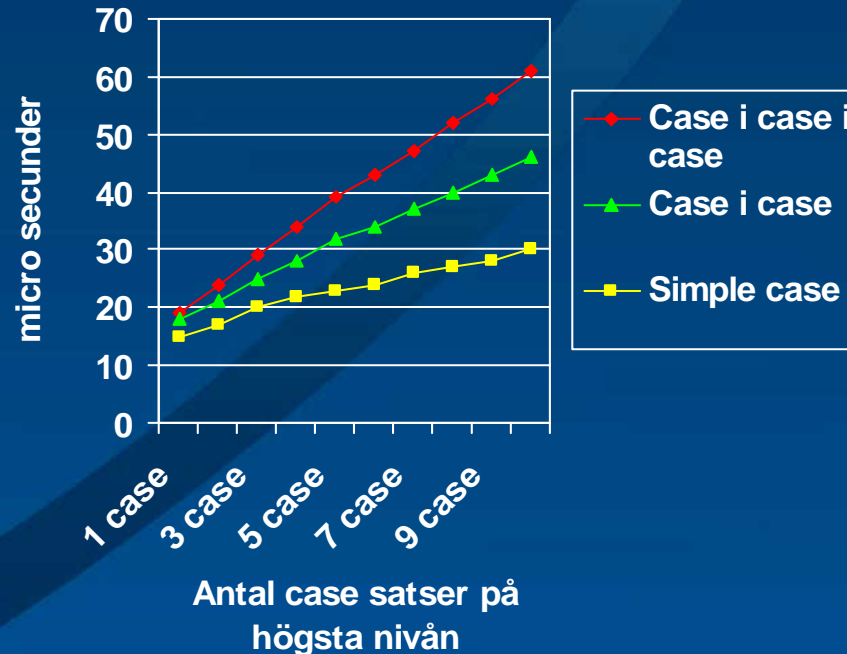
```

Snitt tid för en fetch



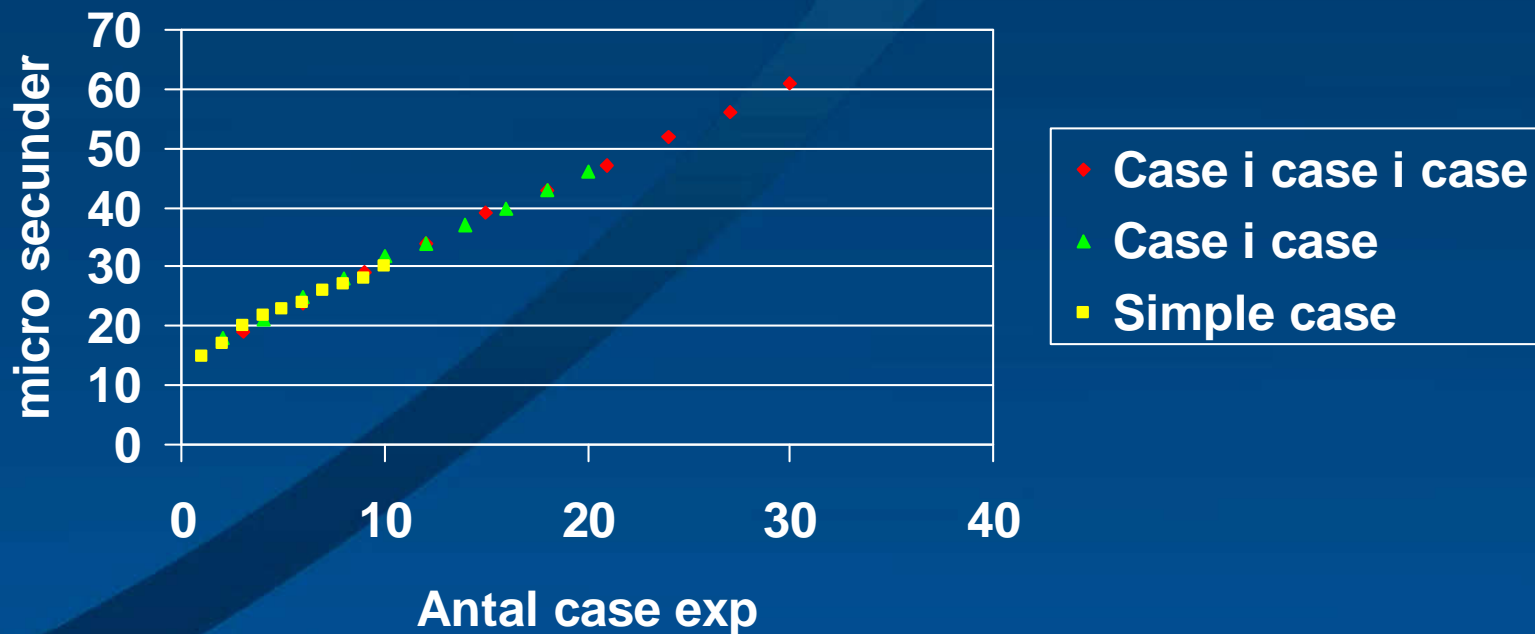
Om man jämför simple case & case i case & case i case i case får man den här fördelningen

Snitt class 2 CPU tid för en fetch



Om man jämför simple case & case i case & case i case i case får man den här fördelningen

Snitt class 2 CPU tid för en fetch



Slutledning

- CPU och elapsed tiden ökar relativt konstant med antalet case statement som man använder
- Att använda ett CASE statement istället för att göra editeringen i ett program ökar kostanden för en fetch med ca 15% - 20%
- Nestade case statements kostar inte mer än onestade
- Antalet och hur långt ner bland when satserna villkoret utförs påverkar svarstiden

Extra material

En liten observation jag gjort i samband med testerna är att kostnaden för första fetch:en är betydligt ”dyrare” än de därefter

Denna körning är utan case satser

SQL_CALL	STMTÄ	INDB2_CPU
OPEN	00110	00:00.000151
FETCH	00111	00:00.000 329
FETCH	00143	00:00.0000 13
CLOSE	00163	00:00.000025

Denna körning är med många case satser

SQL_CALL	STMTÄ	INDB2_CPU
OPEN	00141	00:00.000143
FETCH	00142	00:00. 000386
FETCH	00174	00:00.0000 30
CLOSE	00194	00:00.000024