Firebird Configuration, episode 3: InlineSortThreshold and details of sortings

Alexey Kovyazin, President of Firebird Foundation
What operations do require sorting?

- ORDER BY
- GROUP BY
- DISTINCT
- UNION (without UNION ALL)
Sorting example

- SELECT ID, NAME FROM TABLE1 ORDER BY DATE1

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>John</td>
<td>06-JUL-1977</td>
</tr>
<tr>
<td>2</td>
<td>Alex</td>
<td>12-MAR-1980</td>
</tr>
<tr>
<td>51</td>
<td>Tom</td>
<td>07-DEC-1982</td>
</tr>
<tr>
<td>80</td>
<td>Roland</td>
<td>16-MAY-1992</td>
</tr>
<tr>
<td>7</td>
<td>Helen</td>
<td>26-NOV-1996</td>
</tr>
<tr>
<td>19</td>
<td>Nataly</td>
<td>03-APR-1974</td>
</tr>
<tr>
<td>39</td>
<td>Carlos</td>
<td>20-AUG-1987</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Helen</td>
<td>26-NOV-1996</td>
</tr>
<tr>
<td>80</td>
<td>Roland</td>
<td>16-MAY-1992</td>
</tr>
<tr>
<td>39</td>
<td>Carlos</td>
<td>20-AUG-1987</td>
</tr>
<tr>
<td>51</td>
<td>Tom</td>
<td>07-DEC-1982</td>
</tr>
<tr>
<td>2</td>
<td>Alex</td>
<td>12-MAR-1980</td>
</tr>
<tr>
<td>19</td>
<td>Nataly</td>
<td>03-APR-1974</td>
</tr>
<tr>
<td>39</td>
<td>Carlos</td>
<td>20-AUG-1987</td>
</tr>
</tbody>
</table>
How Sorting Works

SELECT ID, NAME FROM TABLE1 ORDER BY DATE1

RESULT OF SORTING

• There are 3 ways how sorting can work:
  - SORT
  - REFETCH/INLINE
  - INDEX
SELECT ID, NAME FROM TABLE1 ORDER BY DATE1

RESULT OF SORTING

ID INTEGER (8 bytes) | NAME VARCHAR(100) | DATE 8 bytes

KEY OF SORTING

116 bytes
SORT

3 Steps
1. Read both result columns and key into the single wide record set, stored in the temp area and, if not enough, on disk
2. Sort recordset by key columns
3. Return results according the fetch command

SELECT ID, NAME
FROM TABLE1
ORDER BY DATE1

Storing ID, NAME, DATE

Ordering ID, NAME, DATE1 by DATE1

Return results (fetching records)
SORT with *

**SELECT ID, NAME FROM TABLE1 ORDER BY DATE1**

- **Record length**
  - ID INTEGER (8 bytes)
  - NAME VARCHAR(100)
  - DATE 8 bytes
  - Total 116 bytes

**KEY OF SORTING**

**SELECT * FROM TABLE1 ORDER BY DATE1**

- * = many fields e.g. 20000 bytes
  - DATE 8 bytes
  - Total 20008 bytes
SORT — small sorting

ID, NAME, DATE1

Record Set = 1Gb

TempCacheLimit = 2Gb
SORT — big sorting

* (ALL), DATE1

Record Set = 3Gb

TempCacheLimit = 2Gb

fb_sort_file = 1Gb
How can we know the size of sorting?

- In isql
  - set explain;

- In trace
  - explain_plan = true
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
<th>Type</th>
<th>Nullable</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDB$FIELD_NAME</td>
<td>(RDB$FIELD_NAME) CHAR(63) CHARACTER SET UTF8</td>
<td>Nullable</td>
<td></td>
</tr>
<tr>
<td>RDB$TYPE</td>
<td>(RDB$GENERIC_TYPE) SMALLINT</td>
<td>Nullable</td>
<td></td>
</tr>
<tr>
<td>RDB$TYPE_NAME</td>
<td>(RDB$TYPE_NAME) CHAR(63) CHARACTER SET UTF8</td>
<td>Nullable</td>
<td></td>
</tr>
<tr>
<td>RDB$DESCRIPTION</td>
<td>(RDB$DESCRIPTION) BLOB segment 80, subtype TEXT</td>
<td>Nullable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHARACTER SET UTF8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDB$SYSTEM_FLAG</td>
<td>(RDB$SYSTEM_FLAG) SMALLINT</td>
<td>Not Null</td>
<td></td>
</tr>
</tbody>
</table>

SQL> show table rdb$types;
Example — sorting records 284 bytes

SQL> select rdb$type_name from rdb$types order by rdb$type;

Select Expression

  -> Sort (record length: 284, key length: 8)

  -> Table "RDB$TYPES" Full Scan
Example — sorting record 548 bytes

```
SQL> select * from rdb$types order by rdb$type;
```

Select Expression

-> Sort (record length: 548, key length: 8)

-> Table "RDB$TYPES" Full Scan
Record size calculation

• Header for each record — always aligned to 32 bytes
  - 8 bytes — transaction's number
  - 1 byte — DBKEY validity
  - 8 bytes — DBKEY
  - 8 bytes — key for recordset

• Returned data
  - Integer — 8 bytes
  - Timestamp — 8 bytes
  - Varchar (NNN) = NNN * charset (4 for UTF8)
Some calculations

- Sorting 1000000 records, 548 bytes = ~522Mb
- CREATE TABLE T1(i1 integer, v1 varchar(999);
  - SELECT * FROM T1 ORDER BY i1
    - Each record 8b+999*4 (UTF8)+24 bytes = 4028 bytes
    - 1mln records = ~3,7Gb
Example

SELECT FIRST 10 * FROM T1 ORDER BY F1

• Table size = 1000000 records
• Width of record to sort (ALL + F1) = 4K
• F1 size = 8 bytes

To show 10 first records, Firebird will sort 4Gb of data!
REFETCH

- In firebird.conf of Firebird 4
  - InlineSortThreshold=NN #bytes

- In firebird.conf of HQbird 3
  - SortDataStorageThreshold=NN #bytes

- If optimizer sees that record is bigger than specified limit, it will sort only keys, and re-read records by the key.
Example Refetch

SQL> create table TEST11(I1 integer, vbig varchar(20000));
SQL> show table TEST11;
I1       INTEGER Nullable
VBIG      VARCHAR(20000) Nullable

SQL> select * from TEST11 ORDER BY I1;
Select Expression
    -> Refetch
        -> Sort (record length: 28, key length: 8)
            -> Table "TEST11" Full Scan
How refetch works

1. Read records, build keys for records set
2. Sort keys as usual
3. Re-read records according the sorted keys
4. Return results
ORDER INDEX

• If there is index matching the condition in ORDER BY or GROUP BY, Firebird will try to use index (with some limitation)
  – In this case Firebird will not use temp area to store records
Why ORDER INDEX is not always great

- It can be used to order only one table
- Access through index usually requires 3-4x more fetches operations than NATURAL fetching and therefore slower for sorting large amounts of data!
<table>
<thead>
<tr>
<th></th>
<th>SORT</th>
<th>REFETCH</th>
<th>INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Temp area?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sort record and key?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Good to sort millions of records with small or moderate size?</td>
<td>Yes, will</td>
<td>Depends</td>
<td>No</td>
</tr>
<tr>
<td>Good to sort millions of records with extra large size (&gt;16K)?</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Good to return only a few (&lt;1000) first rows?</td>
<td>No</td>
<td>Depends</td>
<td>Yes</td>
</tr>
<tr>
<td>Order/Group condition can have columns from various tables?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reads tables twice?</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Recommendations for tuning InlineSortThreshold

- Find queries with SORT plan generating large sort files
- Set InlineSortThreshold less than their records length
- Monitor production
Analysing actual temp space usage

• Unfortunately, in the current versions of vanilla Firebird there is no monitoring of used space in TempCacheLimit
  – HQbird has TempSpaceThreshold parameter to report operations which use more space than set
Questions?

• ak@firebirdsql.org

•
Allocating and deallocating temp space

- TempCacheLimit sets the limit, the memory is allocated on demand.
- Temp space is allocated by blocks, specified by parameter TempBlockSize, default =1Mb
- Temp space is deallocated when query finishes the fetch — i.e., when all records of resultset are returned to the client