Paper Presentation and Discussion
on

Automated Statistics Collection in DB2 UDB

(Authors: A Aboulonga, P Haas, M Kandil, S Lightstone, G Lohma, V Markl, I Popivanov, V Raman)

Ranjith Vasireddy

http://www.ee.duke.edu/~rv5/

Mar 2, 2006
Optimiser Impact on System Availability

- Example with 4 tables. ‘Claim’ <1 minute
  - say 45 seconds in worst case is downtime if executed during normal workload
- Once every 7 days
- 2160 seconds = 36 minutes of ‘downtime’
  => Not a highly-available system
- **Simple** system with 4 tables and ‘controlled-workload’ may not be able to achieve five 9s (**just** because of re-optimization, without considering failures).
Motivation

Optimizer

Best Plan

Plan Execution

Statistics

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>tables</td>
<td>number of rows in a table</td>
</tr>
<tr>
<td>columns</td>
<td>number of distinct values for that column</td>
</tr>
<tr>
<td>indexes</td>
<td>number of distinct index keys, clustering of the table with respect to the index, physical properties of the index</td>
</tr>
<tr>
<td>coldist</td>
<td>quantiles and frequent values of a column</td>
</tr>
<tr>
<td>colgroups</td>
<td>distinct number of values for a group of columns</td>
</tr>
</tbody>
</table>
Motivation

- DB stat.s are not incrementally updated
  - Maintenance is too expensive
  - => statistics are likely to be out of date
- If stat.s are refreshed frequently
  - If proper config. Parameters are not set properly (\# of frequent values, \# of quantiles to maintain etc)
- Previous systems

- Utility method
  - DB2: RUNSTATS on a per table basis (RunStats profiles in SYSSTAT.PROFILE)
  - ORACLE: ANALYZE
  - INFORMIX: UPDATE STATISTICS
  - SYBASE: UPDATE STATISTICS
Motivation

- Query Feedback
- UDI activity

- Without **ANY** DBA Intervention
  - ASC decides
    - Which statistics to gather
    - What level of detail to gather
    - When to gather
Feedback Loop

Query

Optimizer

Best Plan

Plan Execution

Statistics

RUNSTATS

Scheduler

Communicates to scheduler its findings about tables with outdated stats for prioritization

Statistical Profile

Query Feedback Analyzer (QFA)

Plan Monitor (PM)

Runtime Monitor (RM)

Q. Feedback Warehouse

RUNSTATS

Statistics

Plan Monitor (PM)

Runtime Monitor (RM)

Q. Feedback Warehouse

Query Feedback Analyzer (QFA)
QF-driven summary

- Monitors query results
- Modifies RUNSTATS profile
- Recommends execution whenever
  - config params are improper
  - stat.s out of date
Another Feedback Loop

Monitor

- Activity Monitor (AM)
- UDI Counter

Analyze

- Activity Analyzer (AA)
- Significant change in stats?

Plan

- Prioritize tables; avoid starvation

Exploit

- Data
- Query

- DML Processor
- Database
- Statistics
- RUNSTATS

- Optimizer
- Best Plan
- Plan Execution

- Scheduler

- Query
- Data

- Optimizer
- Best Plan
- Plan Execution

- Scheduler
UDI-driven summary

- Monitors UDI
- Recommends execution of RUNSTATS
Automated Statistics Collection

Data

Query

Optimizer

Best Plan

Plan Execution

Plan Monitor (PM)

Runtime Monitor (RM)

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Plan Monitor (PM)

Activity Analyzer (AA)

Scheduler

Database

Statistics

Activity Monitor (AM)

UDI Counter

DML Processor

Estimated Cardinalities

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query

Optimizer

Best Plan

Plan Execution

Q. Feedback Warehouse

RUNSTATS

Statistical Profile

Query Feedback Analyzer (QFA)

Statistical Profile

Best Plan

Plan Execution

Query
UDI and QF - driven

- Scheduler combines and triggers RUNSTATS
- Maintenance window
  - RUNSTATS allocated a large resources
  - Throttled background process - impact < 7%
    (non-maintenance window)
  - Frequency and length controlled by DBA
  - End of maintenance window
Neither one is sufficient

- Neither one is sufficient
  - UDI driven approaches are proactive
    - can handle ‘unforeseen queries’
    - May not concentrate on maintaining statistics critical to workload
  - QF-driven are reactive
    - Future data-querying pattern follows past pattern
    - Require learning time
    - focus on critical stats
UDI Driven Process

I/P: list of tables

Checks for reasonable amount of UDI and Load

Checks if ‘analyzable’ columns stats have significantly changed

O/P: prioritized list of tables
Change Analyzer
UDI Driven Process

- DAC verifies that table-related structures are cached in memory
- At least $\tau\%$ are modified ($\tau = 10$)
QFA Driven Process

I/P: list of tables

O/P: prioritized list of tables
Operation of the QFA

- TCA similar to UDI. Is there any difference and use?

- SPA:

  \[
  \text{newcount}_{\text{raa}} = \left( C - \sum_{i=1}^{K} f_i \right) / (d - K)
  \]

- Correlation Analyzer : Pair-wise correlation
QFW Tables
Scheduling Statistics Collection

- Invokes QFA and AA
- DBA controls
  - QFA or AA or both
  - Maximum allowable space for QFW
- Scheduler also invokes RUNSTATS as a throttled background process to collect stats of high priority tables
- CA is invoked to check rate of change
Scheduling Statistics Collection

// G, P, D, Q, C are lists of tables
// T is a table
G := tables to be checked by AA during the initial maintenance iteration

P, D, Q, C := {}
while(true)
{
    // Call the AA on the Tables in G
    D := AA(G);
    // Call the Query Feedback Analyzer
    Q := QFA();
    // prioritize D and Q based on the ranking criteria
    // and merge with list of critical tables C
    P := prioritizeMerge(D, Q, C);
    while (still time in maintenance window)
    {
        T := Pop(P); // T is table in P with highest priority
        execute RUNSTATS on T
        and estimate the data change rate;
    }
    // Construct list for next maintenance interval
    (G, C) := constructDueTables()
    sleep until the next maintenance window;
}
**Prioritizing Tables** (1000s of tables and Terabytes of data)

- **Useful** – more than 0% and less than 50% experiencing change
- **Needed** – recommended by QFA
- **Pressing** – 50% or more rows
- **Urgent** – both **Needed** and **Useful** or **Pressing**
- **Critical** – has been starved either
  - UDI counter is +ve, but an excessive # of iterations have passed since last refresh
  - **RUNSTATS** has never been executed
- **Tables are prioritized within each class**
Recap: Neither one is sufficient

- Neither one is sufficient
  - UDI driven approaches are proactive
    - can handle ‘unforeseen queries’
    - May not concentrate on maintaining statistics critical to workload
  - QF-driven are reactive
    - Require learning time
    - focus on critical stats
Is there an advantage of having both AA and QFA?

=> If yes, what is the ‘% gain’ over AA only and QFA only methods?

=> Or, is it just more resource consuming without ‘considerable’ advantage, if any?
After additional insertions, response time decreased

⇒ Is query 5 and 6 independent of insertions?
After inserting and doing AA only, there is an advantage. Why?

what kind of queries would increase response time after AA or QFA. And why?
Related discussion about combined AA and QFA result

- QFA only/ AA only/ both?
  - May be tables/queries can be classified wrt QFA or AA or both?
Thank You