Sage Advice Part 3: Predictive Index Impact Analysis -- Know Before you CREATE

Scott Richard Hayes, DBI Software, @dbisoftware

Session Code: D11
Wednesday, 16 November, 11:00-12:00
Platform: DB2 for Linux, UNIX, and Windows
Submitted to IDUG...
Abstract & Key Bullet Points

• Whenever a DBA has an index they want to create to solve a performance issue, there is often someone raising an objection “But will this new index cause the database/application any harm?” For those that desire to successfully create indexes with confidence, this session will present a new method for predictively measuring the impacts of any new indexes so that informed decisions can be fearlessly made. Example commands and SQL will be provided.

• This session continues the Sage Advice series from Parts 1 (Weight Analysis) and 2 (Advanced Index Benefit Analysis)

• Bullet Objectives in slide notes...
Our Agenda today...

- Quick Review of Sage Advice, Part 1, Weight Analysis
- Quick Review of Sage Advice, Part 2, Advanced Index Benefit Analysis
- Sage Advice, Part 3, Predicting Index Impact Analysis
Part 1: Quick Review

SAGE ADVICE PART 1:
WORKLOAD WEIGHT ANALYSIS
How much does it weigh?
TOTAL weight and RELATIVE weight...
We have a WEIGHT “Opportunity for Improvement”
Table Performance Analysis
Table Rows Read per Transaction (TBRRTX) & WEIGHT

- Not every TX accesses every table, so we expect Rows Read/#TX to be a small average, normally < 10, and often 3 or less
  - TBRRTX tells you where you have Data Page scans occurring
    - > 10, likely opportunity for improvement
    - > 100, definitely opportunity for improvement
    - > 1,000, crisis! **DO NOT UPGRADE HARDWARE**

- In addition to the cost per TX, find the % of DB Rows Read (**Relative Weight**) by expressing Table Rows Read \( \times 100 / \text{Sum of all Rows Read} \).
### Examples

#### Table Relative Weights and Read I/O Costs

<table>
<thead>
<tr>
<th>Schema</th>
<th>Table</th>
<th>Size (MB)</th>
<th>% Space</th>
<th>Rows Read</th>
<th>% Rows Read</th>
<th>Rows Read/Tx</th>
<th>Rows Read/Sec</th>
<th>Rows Written</th>
<th>% Rows Written</th>
<th>Rows Written/Tx</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSADM</td>
<td>XPX_PROMOTION</td>
<td>59.414</td>
<td>0.000%</td>
<td>314,269,599</td>
<td>9.250%</td>
<td>3.86</td>
<td>28,315.13</td>
<td>0</td>
<td>0.000%</td>
<td>0.000%</td>
</tr>
<tr>
<td>UCSADM</td>
<td>PX_PROMOTION</td>
<td>13,581.410</td>
<td>0.190%</td>
<td>252,701,117</td>
<td>7.430%</td>
<td>3.12</td>
<td>22,767.92</td>
<td>0</td>
<td>0.000%</td>
<td>0.000%</td>
</tr>
<tr>
<td>UCSADM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLESCHEMA</th>
<th>TABNAME</th>
<th>ROWSREAD</th>
<th>PCT_DB_TB_ROWSREAD</th>
<th>TBRTX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WEBSITE_DATA_TB</td>
<td>1325709286</td>
<td>99.23</td>
<td>322086.804</td>
</tr>
<tr>
<td></td>
<td>WEBSITE_DATA_TB</td>
<td>10141825</td>
<td>0.75</td>
<td>2464.000</td>
</tr>
<tr>
<td></td>
<td>SYSPLAN</td>
<td>15695</td>
<td>0.00</td>
<td>3.813</td>
</tr>
<tr>
<td></td>
<td>SYSRoutines</td>
<td>146</td>
<td>0.00</td>
<td>0.135</td>
</tr>
<tr>
<td></td>
<td>SYSEVENTMONITORS</td>
<td>133</td>
<td>0.00</td>
<td>0.032</td>
</tr>
<tr>
<td></td>
<td>SYSHISTOGRAMTEMPLATEBINS</td>
<td>40</td>
<td>0.00</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>SYSTABLES</td>
<td>35</td>
<td>0.00</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>SYSHISTOGRAMTEMPLATEUSE</td>
<td>28</td>
<td>0.00</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>SYSBUFFERPOOLS</td>
<td>18</td>
<td>0.00</td>
<td>0.004</td>
</tr>
<tr>
<td></td>
<td>SYSDAUTH</td>
<td>10</td>
<td>0.00</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>SYSTABLESPACES</td>
<td>8</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSRoleAUTH</td>
<td>8</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSEVENTTABLES</td>
<td>8</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSSOLES</td>
<td>7</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSSERVICES</td>
<td>6</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSSERVICECLASSES</td>
<td>6</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSSWORKLOADS</td>
<td>4</td>
<td>0.00</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SYSMODEGROUPS</td>
<td>2</td>
<td>0.00</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>SYSCONTENTS</td>
<td>2</td>
<td>0.00</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>XCEFFM</td>
<td>46,277.750</td>
<td>0.660%</td>
<td>43,625.040</td>
</tr>
</tbody>
</table>

20 record(s) selected.
Now that you know the TABLES with the heaviest WEIGHTS, what is the heavy SQL driving I/O to the heavy tables?

- STMT_TEXT like %TABLE_NAME% has some limitations
- grep –i “TABLE_NAME” has similar limitations

What are the HEAVIEST SQL – By table? Across the DB?

- CPU %
- Rows Read %
- Logical Reads %
- Physical Reads %
- Rows Written %
- Execution Time %
- Sort Time %
SQL HEAVY WEIGHTS
by CPU Time (microseconds)

```sql
SELECT
    CAST( ( (A.TOTAL_USR_CPU_TIME * 1000000) + A.TOTAL_USR_CPU_TIME_MS
    + (A.TOTAL_SYS_CPU_TIME * 1000000) + A.TOTAL_SYS_CPU_TIME_MS )
    / A.NUM_EXECUTIONS ) AS DECIMAL (15,0) AS AVG_CPU_TIME_MS,
    CAST (A.NUM_EXECUTIONS AS INTEGER) AS NUM_EXECS,
    CAST( ((A.TOTAL_USR_CPU_TIME * 1000000) + A.TOTAL_USR_CPU_TIME_MS
    + (A.TOTAL_SYS_CPU_TIME * 1000000) + A.TOTAL_SYS_CPU_TIME_MS) * 100.0)
    / (Select (SUM(B.TOTAL_USR_CPU_TIME) * 1000000)
    + (SUM(B.TOTAL_SYS_CPU_TIME) * 1000000)
    + SUM(B.TOTAL_USR_CPU_TIME_MS)
    + SUM(B.TOTAL_SYS_CPU_TIME_MS)
    + 1.0
    FROM SYSIBMADM.SNAPDYN_SQL B
    WHERE A.DBPARTITIONNUM = B.DBPARTITIONNUM ) AS DECIMAL(5,2) AS PCT_CPU_TIME,
    SUBSTR(A_STMT_TEXT,1,100) AS CPU_SUCKING_SQL
FROM SYSIBMADM.SNAPDYN_SQL A
WHERE A.NUM_EXECUTIONS > 0
ORDER BY A.DBPARTITIONNUM ASC, 3 DESC, 1 DESC FETCH FIRST 25 ROWS ONLY;
```
SQL HEAVY WEIGHTS
by CPU Time (microseconds) - Examples

25 record(s) selected.
Take a picture of your luggage before you fly - easy description when lost - documents bag condition

• #WISDOM
Part 2: Quick Review

SAGE ADVICE PART 2: ADVANCED INDEX BENEFIT ANALYSIS
So, you found a heavy weight SQL statement, and you passed it to the Design Advisor (db2advis), and the Design Advisor suggests that you create 3, 5, 11, or 13 indexes for a solution!

**HOW MANY?**

**REALLY?**

**OPTIMIZING INDEX SOLUTIONS**
Optimizing Index Solutions
Solving a “Heavy” Query

```sql
SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd, v.verb_desc
    FROM DBIPOC.SUCCESSFUL_HITS_VW A,
            DBIPOC.VERB_DESCRIPTIONS V
WHERE a.domainname = 'webnj1.bbh.com'
    AND a.targetfile = '/blog/rss/Scott_Hayes_rss2.xml'
    AND a.bytesxferd < (SELECT AVG(b.bytesxferd) FROM DBIPOC.SUCCESSFUL_HITS_VW B)
    AND a.hittimestamp < '2011-12-31-21.35.43.304000'
    AND a.actionverb = v.actionverb
fetch first 100 rows only;
```
Optimizing Index Solutions
The IBM Design Advisor (db2advis) gives 5 Indexes!

What is the benefit, or relative value, of Each Index?

185,651?
*sigh*
We can do better!
Optimizing Index Solutions
Relative Benefit Value Analysis

- **Two Methods to Consider**
  - **Index Addition** – Add indexes one at a time to assess individual value
  - **Index Subtraction** – Subtract Indexes one at a time from the solution set to assess the value lost

- **Design Advisor can be overly aggressive on Index Only Access**
  - Sometimes additional columns are added to existing indexes to achieve IX Only access – we anticipate these will have less value
  - Give consideration to predicates involved when making final decisions on which indexes to implement
Optimizing Index Solutions

Index Addition 1

- Start with a clean Explain & Advise Environment
  - Delete from Explain_Instance
  - Delete from Advise_Index

- Explain the statement
  - `db2batch -d dbipocdb -f 3Table_Heavy_Query.sql -o e explain`
Optimizing Index Solutions
Index Addition 2

- Find the original/”Before” Explain Cost
Optimizing Index Solutions
Index Addition 3B – alternate method

- Populate the ADVISE_INDEX table - CLP
  - `db2 "select current explain mode from sysibm.sysdummy1"`
    - “NO”
  - `db2 "set current explain mode recommend indexes"
  - `db2 -stvf 3Table_Heavy_Query.sql`
    - Does not execute the query!
    - Populates the ADVISE_INDEX table
  - `db2 "set current explain mode NO"
    - So you can run queries again!
Optimizing Index Solutions

ADVISE_INDEX Table 1

- **USE_INDEX** Column – the “magic”
  - ‘Y’ Index Recommended or Evaluated
  - ‘N’ Index not to be Recommended or Evaluated
  - ‘R’ An existing clustering RID index was recommended by Design Advisor to be unclustered – this is the case when a new clustering RID index is recommended for the table
  - ‘I’ Ignore an existing non-unique index for Evaluation. The EXISTS column should be ‘Y’ in this case or the index will not be ignored

- Several other interesting and helpful columns too
  - See sample query and results, next slide
### Optimizing Index Solutions

**ADVISE_INDEX Table 2**

```sql
Execute SQL: db2in105@LPAR21:60018/DBIPOCDB
```

<table>
<thead>
<tr>
<th>PROPOSED_INDEX</th>
<th>ON_TABLE</th>
<th>EXISTS</th>
<th>USE_INDEX</th>
<th>INDEX_COLS</th>
<th>NLEVELS</th>
<th>NLEAF</th>
<th>UNIQUERULE</th>
<th>FIRSTKEYCARD</th>
<th>FULLKEYCARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDX150309253459460</td>
<td>DBIPOC HTML_STATUS_CODES</td>
<td>N</td>
<td>Y</td>
<td>+STATUS_DESC+STATUS_CODE</td>
<td>2</td>
<td>3</td>
<td>D</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>IDX150309253459460</td>
<td>DBIPOC HTML_STATUS_CODES</td>
<td>N</td>
<td>Y</td>
<td>+STATUS_DESC+STATUS_CODE</td>
<td>2</td>
<td>3</td>
<td>D</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>IDX1503092345530</td>
<td>DBIPOC WEBSITE_DATA_TB</td>
<td>N</td>
<td>Y</td>
<td>+WEBSTATUS+BYTESFERD</td>
<td>3</td>
<td>139</td>
<td>D</td>
<td>10</td>
<td>189450</td>
</tr>
<tr>
<td>IDX1503092346050</td>
<td>DBIPOC VERB_DESCRIPTIONS</td>
<td>N</td>
<td>Y</td>
<td>+ACTION_VERB_VERB_DESC</td>
<td>2</td>
<td>3</td>
<td>D</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>IDX1503092346070</td>
<td>DBIPOC WEBSITE_DATA_TB</td>
<td>N</td>
<td>Y</td>
<td>+DOMAINNAME+TARGETFILE+BYTESFERD+HITTIMESTAMP+PROTOCOL+ACTION_VERB+WEBSTATUS</td>
<td>3</td>
<td>896</td>
<td>D</td>
<td>134</td>
<td>134</td>
</tr>
</tbody>
</table>
Optimizing Index Solutions
So, what are those proposed indexes worth?

- set current explain mode **EVALUATE INDEXES**
  - USE_INDEX = ‘Y’ for all Proposed Indexes
- `$ db2 -tvf 3Table_Heavy_Query.sql`
- set current explain mode **NO**
- `$ db2 –tvf Query_In_Slide_Notes.sql`
- **376 timerons**
- **Down from 81,524**
  - 99.54% Reduced
Optimizing Index Solutions
Index Addition

• What is the value of each index individually, in isolation?
• Set USE_INDEX to ‘N’ for all Indexes
  • update advise_index set use_index='N';
• For each proposed index:
  • Set USE_INDEX to ‘Y’
    • Update ADVISE_INDEX set USE_INDEX = ‘Y’ where NAME = ‘IXNAME(N)’
  • set current explain mode EVALUATE INDEXES
  • db2 -tvf 3Table_Heavy_Query.sql
  • Retrieve the TOTAL_COST from EXPLAIN_OPERATOR table
    • db2 -tvf Query_In_Slide_Notes.sql
  • Compute Savings Percentage
  • Repeat!
Optimizing Index Solutions
Index Addition – 1st Index

```
$ db2 "set current explain mode NO"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='N'"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092345460'"
DB20000I The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604
```

<table>
<thead>
<tr>
<th>ORIGINAL_COST</th>
<th>ADD_IX1_TOTAL_COST</th>
<th>TIMERON_SAVINGS</th>
<th>VALUE_PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>81524.1953</td>
<td>81524.1406</td>
<td>0.0547</td>
<td>0.0000670</td>
</tr>
</tbody>
</table>
Optimizing Index Solutions
Index Addition – 2\textsuperscript{nd} Index

$ \text{db2 } \text{"set current explain mode NO"}$
DB20000I The SQL command completed successfully.
$ \text{db2 } \text{"update advise_index set use_index='N' where name = 'IDX1503092345460'"}$
DB20000I The SQL command completed successfully.
$ \text{db2 } \text{"update advise_index set use_index='Y' where name = 'IDX1503092345530'"}$
DB20000I The SQL command completed successfully.
$ \text{db2 } \text{"set current explain mode EVALUATE INDEXES"}$
DB20000I The SQL command completed successfully.
$ \text{db2 } -\text{tf 3Table_Heavy_Query.sql}$
SQL0217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604

<table>
<thead>
<tr>
<th>ORIGINAL_COST</th>
<th>ADD_IX2_TOTAL_COST</th>
<th>TIMERON_SAVINGS</th>
<th>VALUE_PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>81524.1953</td>
<td>1844.3532</td>
<td>79679.8421</td>
<td>97.7376615</td>
</tr>
</tbody>
</table>
Optimizing Index Solutions
Index Addition – 3\textsuperscript{rd} Index

```
$ db2 "set current explain mode NO"
DB20000I  The SQL command completed successfully.
$ db2 "update advise_index set use_index='N' where name = 'IDX1503092345530'"
DB20000I  The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092346050'"
DB20000I  The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I  The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W  The statement was not executed as only Explain information requests are being processed.  SQLSTATE=01604
```

<table>
<thead>
<tr>
<th>ORIGINAL_COST</th>
<th>ADD_IX3_TOTAL_COST</th>
<th>TIMERON_SAVINGS</th>
<th>VALUE_PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>81524.1953</td>
<td>81524.1953</td>
<td>0.0000</td>
<td>0E-7</td>
</tr>
</tbody>
</table>
### Optimizing Index Solutions

#### Index Addition – 4th Index

```sql
$ db2 "set current explain mode NO"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='N' where name = 'IDX1503092346050'"
DB20000I The SQL command completed successfully.
$ db2 "update advise_index set use_index='Y' where name = 'IDX1503092346070'"
DB20000I The SQL command completed successfully.
$ db2 "set current explain mode EVALUATE INDEXES"
DB20000I The SQL command completed successfully.
$ db2 -tf 3Table_Heavy_Query.sql
SQL0217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604
```

<table>
<thead>
<tr>
<th>ORIGINAL_COST</th>
<th>ADD_INDEX4_TOTAL_COST</th>
<th>TIMERON_SAVINGS</th>
<th>VALUE_PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>81524.1953</td>
<td>6536.0610</td>
<td>74988.1343</td>
<td>91.9826733</td>
</tr>
</tbody>
</table>
## Optimizing Index Solutions
### Index Addition - Summary

<table>
<thead>
<tr>
<th>Index Name</th>
<th>Timeron Savings</th>
<th>Value %</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDX1503092345460</td>
<td>0.0547</td>
<td>0.0000670</td>
</tr>
<tr>
<td>IDX1503092345530</td>
<td>79679.8421</td>
<td>97.7376615</td>
</tr>
<tr>
<td>IDX1503092346050</td>
<td>0.0000</td>
<td>0.0000000</td>
</tr>
<tr>
<td>IDX1503092346070</td>
<td>74988.1343</td>
<td>91.9826733</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>1896204018 %</strong></td>
</tr>
</tbody>
</table>

- **And the award for LEAST valuable index goes to…**
- **And the award for MOST valuable index goes to…**
- **2nd Place MVI**
Optimizing Index Solutions
Compare Explain Plans

![Compare Explain Plans - Execute SQL](image)
Optimizing Index Solutions
Does a High Value Index have IX Access Only “Baggage”?  

Let’s play Predicate BINGO!
$ db2 –tvf Query_In_Notes.sql

VERB_DESC & PROTOCOL are supporting IX Access Only
SAGE ADVICE PART 3:
PREDICTING INDEX IMPACT ANALYSIS
DBA Performance Analysis Challenges

1. What needs to be fixed or improved?
   - *Make sure you are fighting the right fires* - via Weight Analysis

2. What are the optimal design solutions?
   - *Make sure you are fighting the right fires with the right type of fire extinguishers and equipment* – via Advanced Index Benefit Analysis

3. Will proposed design solutions cause any inadvertent harm? Will benefits exceed expectations? Can “multiple birds be killed with just one stone?”
   - *Make sure you are fighting the right fires without causing inadvertent damaging explosions* – via Predictive Index Impact Analysis
Review
The “Heavy_Query” – 90% of CPU & I/O

```sql
SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd, v.verb_desc
  FROM DBIPOC.SUCCESSFUL_HITS_VW A,
       DBIPOC.VERB_DESCRIPTIONS V
  where a.domainname = 'webnj1.bbhh.com'
  and a.targetfile = '/blog/rss/Scott_Hayes_rss2.xml'
  and a.bytesxferd < (select avg(b.bytesxferd) from DBIPOC.SUCCESSFUL_HITS_VW B)
  and a.hittimestamp < '2011-12-31-21.35.43.304000'
  and a.actionverb = v.actionverb
fetch first 100 rows only;
```
Review

Explain Heavy SQL & Get Costs: 187,411 Timerons

```sql
select dec(total_cost,20,4) as Query_Timeron_Cost,
dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as CPU_cost,
dec(Comm_cost,20,4) as comm_cost,
from Explain_Operator
where operator_type = 'RETURN'
```

**Table:**

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>187411.2500</td>
<td>105847.0000</td>
<td>387620.00</td>
</tr>
</tbody>
</table>
**Review**

**Get Recommended Indexes - 1**

```
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai dug\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timeron_Cost_from_Expl an_O perator.sql

select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm
on Explain_O perator, (select max(explain_time) as maxtime from Explain_O perator) as b where explain_time = b.maxtime and operator_type = '

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>187411.2500</td>
<td>105847.000</td>
<td>3875200960.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

1 record(s) selected.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai dug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode recommend indexes"

set current explain mode recommend indexes
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai dug\NA2016\NA16-Speaker\SQL\db2 -tvf heavy_query.sql

SQL0217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai dug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode no"

set current explain mode no
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai dug\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timeron_Cost_from_Expl an_O perator.sql

select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm
on Explain_O perator, (select max(explain_time) as maxtime from Explain_O perator) as b where explain_time = b.maxtime and operator_type = '

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1760.0567</td>
<td>136.4210</td>
<td>40684996.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

1 record(s) selected.
```
Advanced Index Benefit Analysis (AIBA) identifies that some indexes are more beneficial than others.

Let’s assume we want to create the last 4 indexes after AIBA.

We thus know the tables that we’ll be creating indexes on (impacted tables).

Three Distinct Table Names are Impacted.
Predictive Index Impact Analysis (PIIA) – Step 1
Determine SQL that Impacts the Impacted Tables

- For each impacted table, determine the SQL queries that have contributed I/O
  - In Sage Advice Part 1, we looked at SQL queries that would find “heavy” queries contributing I/O to a table or the database overall
  - Recall that:
    - STMT_TEXT like %TABLE_NAME% has some limitations
    - grep -i “TABLE_NAME” has similar limitations
    - Query the package cache with MON_GET or SYSIBMADM views
      - Be mindful to include relevant and significant workload timeframes when finding SQL
  - Consider capturing and concatenating workloads from different time periods
    - Sample query in notes
Predictive Index Impact Analysis (PIIA) – Step 2
Determine the Distinct Impacting SQL

• For efficiency, determine the DISTINCT SQL statements (workload) of SQL across the UNION ALL of impacted tables.
  • For Example:
  • SELECT A.C1, B.C1 FROM TB1 A, TB2 B WHERE A.ID1 = B.ID2
    • This SQL would contribute I/O to BOTH tables TB1 and TB2, but for PIIA it only needs to be analyzed once.

• This step is optional but can save time and processing

• By this point, you have determined dozens, hundreds, or maybe thousands of (distinct) SQL that contribute I/O to the impacted tables. Henceforth, we’ll simply call this the “IMPACTING WORKLOAD”
Predictive Index Impact Analysis (PIIA) – Step 3

EXPLAIN the IMPACTING WORKLOAD

• For each (distinct) SQL within the Impacting Workload:
  • Set USE_INDEX = ‘N’ for ALL Contemplated Indexes
  • EXPLAIN the SQL statement to learn its current/original Timeron Cost (Explain Mode EVALUATE INDEXES).
  • Set USE_INDEX = ‘Y’ for the Indexes that you intend to create per your AIBA (4 out of 5 in our earlier example)
  • EXPLAIN the SQL statement to learn its forecasted/new Timeron Cost
  • Compute **Original** Timeron Cost – **New** Timeron Cost = Timeron **Savings** (or degradation if negative), and determine the Savings Percent. Savings% could be multiplied against workload execution totals to predict new relative weights (heaviness)
  • Tabulate the sums of all Original Timeron Costs and New Timeron Costs to understand overall workload impact
Predictive Index Impact Analysis (PIIA) ILLUSTRATED

• From earlier slides, there were 5 proposed indexes against 3 different tables.
  • Based on AIBA, we’re assuming that 4 of the 5 indexes will be created: IDX1602060629500, IDX1602060630060, IDX1602060630030, and IDX1602060629480
  • For our Impacting Workload, for sake of example, we’ll assume there are 10 distinct statements driving I/O to our 3 different tables. Each of these will be stored individually in a file Snn.SQL where “nn” is the distinct statement number. For convenience, our original heavy query will be contained within file S00.SQL.
Predictive Index Impact Analysis for S00.sql

Original: 187,411  New: 1,760  Savings: 185,651  99.06%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf Update_Advise_Index_Use_IDX_EQ_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ("IDX1602060629500", "IDX1602060630060", "IDX1602060630030", "IDX1602060629400")
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v set current explain mode evaluate indexes
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf S00.sql
SELECT a.hittimestamp, a.actionverb, a.protocol, a.bytesxferd, v.verb_desc FROM DBIPOC.SUCCESSFUL_HITS_UV A, DBIPOC.VERB_DESCRIPTIONS V
innname = 'vebnj1.bbh.com' and a.targetfile = '/blog/rss/Scott_Hayes/rss2.xml' and a.bytesxferd < (select avg(b.bytesxferd) from DBIPOC.SUCCESSFUL_HITS_UV B) and a.hittimestamp < '2011-12-31-21.35.43 304000' and a.actionverb = v.actionverb fetch first 100 rows only
SQL0217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v set current explain mode no
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timeron_Cost_from_Explain_Operator.sql
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm in Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as h where explain_time = h.maxtime and operator_type = "ith UR"

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1760.0701</td>
<td>136.4210</td>
<td>49762252.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S01.sql

Original: 93,690  New: 26  Savings: 93,664  99.97%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf Update_Advice_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I  The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I  The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf S01.sql
SELECT ACTIONVERB, TARGETFILE FROM DBIPOC.SUCCESSFUL_HITS_UW WHERE DOMAINNAME = :1s FETCH FIRST 10 ROWS ONLY
SQL0217W  The statement was not executed as only Explain information requests are being processed.  SQLSTATE=01004

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode no"
set current explain mode no
DB20000I  The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timeron_Cost_from_Explain_Operator.sql
select dec(total_cost,20,4) as QUERY_TIMERON_COST, dec(io_cost,20,4) as IO_COST, dec(CPU_cost,20,4) as CPU_COST, dec Komm_cost,20,4) as COMM_COST
from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'R with UR

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.7420</td>
<td>2.0000</td>
<td>242973.3750</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S02.sql

Original: 93,756 New: 39 Savings: 93,717 99.96%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\na16-Speaker\SQL\db2 -tuf Update_Advise_Index_Use IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480')
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\na16-Speaker\SQL\db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\na16-Speaker\SQL\db2 -tuf S02.sql
SELECT ACTION, VERB, TARGETFILE FROM DBIPOC.FAILED_HITS_VU WHERE DOMAINNAME = :1s FETCH FIRST 10 ROWS ONLY
SQL2017W The statement was not executed as only Explain information requests are being processed. SQLSTATE-01604

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\na16-Speaker\SQL\db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\na16-Speaker\SQL\db2 -tuf Query_Timeron_Cost_from_Explain_Operator.sql
select dec(total_cost,20,4) as Query_TIMERON_COST, dec(io_cost,20,4) as IO_COST, dec(CPU_cost,20,4) as CPU_COST, dec(Comm_cost,20,4) as COMM_COST
from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'R'
with UR

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.5943</td>
<td>3.0000</td>
<td>256244.2968</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S03.sql

Original: 93,677  New: 14,053  Savings: 85,624  91.40%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf Update_Advise_Index_Use_IDX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX160206695000', 'IDX160206630030', 'IDX160206639480')
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf S03.sql
SELECT IPADDR, TARGETFILE FROM DBIPOC.WEBSITE_DATA_TB WHERE HIT_TIMESTAMP = :ts ORDER BY TARGETFILE FETCH FIRST 10 ROWS ONLY
SQL0217W The statement was not executed as only Explain information requests are being processed.  SQLSTATE=01604

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timer_on_Cost_from_Expain_Operator.sql
select dec(total_cost,20,4) as Query_Timer_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'H with UR

<table>
<thead>
<tr>
<th>QUERY_TIMER_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>14053.0556</td>
<td>1071.4849</td>
<td>1642515328.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S04.sql

Original: 93,690 New: 4,192 Savings: 89,498 95.53%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tuf Update_Advice_Index_Use_IX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630068', 'IDX1602060630080', 'IDX1602060629480')
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tuf S04.sql
SELECT IPADDR, TARGETFILE FROM DBIPOCSUCCESSFUL_HITS_WHERE HITTIMESTAMP = :ts ORDER BY TARGETFILE FETCH FIRST 10 ROWS ONLY
SQL217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01004

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode no"
set current explain mode no
DB20000I The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naidug\NA2016\NA16-Speaker\SQL\db2 -tuf Query_Timer_Cost_from_Explain_Operator.sql
select dec(total_cost,20,4) as Query_Timer_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as comm from Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_type = 'I'

<table>
<thead>
<tr>
<th>QUERY_TIMER_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>4192.225</td>
<td>1519.2367</td>
<td>69092672.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S05.sql

Original: 27,483  New: 4,268  Savings: 23,215  84.47%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naijug\NA2016\NA16-Speaker\SQL\db2 -tvf Update_Advise_Index_Use_IDX_eq_Y.sql
update advise_index set USE_INDEX = 'Y' where NAME in ('IDX1602060629500', 'IDX1602060630060', 'IDX1602060630030', 'IDX1602060629480001' The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naijug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode evaluate indexes"
set current explain mode evaluate indexes
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naijug\NA2016\NA16-Speaker\SQL\db2 -tvf S05.sql
SELECT IPADDR, BYTESREAD FROM DBIPOG.WEBSITE_DATA_TB WHERE TARGETFILE = :ls FETCH FIRST 10 ROWS ONLY
SQL217W The statement was not executed as only Explain information requests are being processed. SQLSTATE=01604

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naijug\NA2016\NA16-Speaker\SQL\db2 -v "set current explain mode no"
set current explain mode no
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\naijug\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timeron_Cost_from_Explain_Opera
select dec(total_cost,20,4) as Query_Timeron_Cost, dec(io_cost,20,4) as io_cost, dec(CPU_cost,20,4) as cpu_cost, dec(Comm_cost,20,4) as Comm_cost, Explain_Operator, (select max(explain_time) as maxtime from Explain_Operator) as b where explain_time = b.maxtime and operator_ with UR

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>4268.2304</td>
<td>325.6221</td>
<td>484924704.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S06.sql

Original: 93,677 New: 59 Savings: 93,618 99.94%

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai\NA2016\NA16-Speaker\SQL\db2 -tvf Update_Advise_Index_Use_IDX_eq_N.sql
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai\NA2016\NA16-Speaker\SQL\db2 -v "set current explain node evaluate indexes"
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai\NA2016\NA16-Speaker\SQL\db2 -tvf Update_Advise_Index_Use_IDX_eq_Y.sql
DB200001 The SQL command completed successfully.

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai\NA2016\NA16-Speaker\SQL\db2 -tvf S06.sql
SELECT IPADDR, ACTIONWERB, PROTOCOL FROM DDIPOC.WEBSITE_DATA_TB WHERE DOMAINTIME = :ls
SQL0217W The statement was not executed as only Explain Information requests are being processed. SQLSTATE=01604

C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai\NA2016\NA16-Speaker\SQL\db2 -v "set current explain node no"
C:\Users\Scott\Documents\shayes\dbi\Conferences\idug\nai\NA2016\NA16-Speaker\SQL\db2 -tvf Query_Timeron_Cost_from_Explain_Opaterator.sql

<table>
<thead>
<tr>
<th>QUERY_TIMERON_COST</th>
<th>IO_COST</th>
<th>CPU_COST</th>
<th>COMM_COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>58.7012</td>
<td>4.5661</td>
<td>154283.2500</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Predictive Index Impact Analysis for S07.sql
Original: 40,330  New: 54  Savings: 40,276  99.87%

Predictive Index Impact Analysis for S08.sql
Original: 27,483  New: 4,268  Savings: 23,215  84.47%

Predictive Index Impact Analysis for S09.sql
Original: 93,756  New: 39  Savings: 93,717  99.96%
Predictive Index Impact Analysis
The Grand Finale – Drum Roll Please!
PIIA – Do you kill multiple birds with a few stones? Any adverse consequences? Safe to create indexes?

Impacting Workload

<table>
<thead>
<tr>
<th>Query</th>
<th>OLD</th>
<th>NEW</th>
<th>DIFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>187411</td>
<td>1760</td>
<td>-185651</td>
</tr>
<tr>
<td>1</td>
<td>93690</td>
<td>26</td>
<td>-93664</td>
</tr>
<tr>
<td>2</td>
<td>93756</td>
<td>39</td>
<td>-93717</td>
</tr>
<tr>
<td>3</td>
<td>93677</td>
<td>14053</td>
<td>-85624</td>
</tr>
<tr>
<td>4</td>
<td>93690</td>
<td>4192</td>
<td>-89498</td>
</tr>
<tr>
<td>5</td>
<td>27483</td>
<td>4268</td>
<td>-23215</td>
</tr>
<tr>
<td>6</td>
<td>93677</td>
<td>59</td>
<td>-93618</td>
</tr>
<tr>
<td>7</td>
<td>40330</td>
<td>54</td>
<td>-40276</td>
</tr>
<tr>
<td>8</td>
<td>27483</td>
<td>4268</td>
<td>-23215</td>
</tr>
<tr>
<td>9</td>
<td>93756</td>
<td>39</td>
<td>-93717</td>
</tr>
<tr>
<td>Total</td>
<td>844953</td>
<td>28758</td>
<td>-816195</td>
</tr>
</tbody>
</table>

Off the chart savings!
COMMERCIAL BREAK - Folks! Don’t Do this the Hard Way! It’s Time Consuming and Error Prone!

DBI’s Brother-Panther® Automates this Analysis!
COMMERCIAL BREAK - Folks! Don’t Do this the Hard Way! It’s Time Consuming and Error Prone!

DBI’s Brother-Panther® Automates this Analysis!
Sage Advice Part 3: Predictive Index Impact Analysis -- Know Before you CREATE

Please fill out your session evaluation before leaving!