Are you sick of health checks?!

Ulf Heinrich
SEGUS Inc

Session Code: B05
Wednesday 4th of May, 8:00am | Platform: DB2 for z/OS

Agenda

- DB2 Health Checking basics
  - What requires attention?
  - What happens if you don't take care?
- DB2 Health Checking without additional work
  - How to automate regular procedures?
  - Implementing "notify and recommend" instead of having to identify and analyze manually.
- Performance HealthCheck
  - How to evaluate performance critical components and what to include?
  - Reporting about performance violations as well as general messages regarding the overall health of the system.
  - Check the stress of the Coupling Facility
- Statistics HealthCheck
  - How to check the sanity of your catalog statistics?
  - How to clean up your SYSCOLDIST?
  - Minimal invasive SQL performance optimization made easy!

DB2 Health Checking basics

What's wrong with Health Checks?

- There are various offers for a growing number of conditions, but some checks only lead to anxiety and needless work, while others don't spot weaknesses reliably.

→ Health Checks need to honor:
  - the right balance between pinpointing weaknesses and leaving you sick with worry
  - todays technology agility – forget one time/annual checks

→ Health Checks need to tell you:
  - how critical an issue is
  - how to fix a problem that was detected

DB2 Health Checking basics

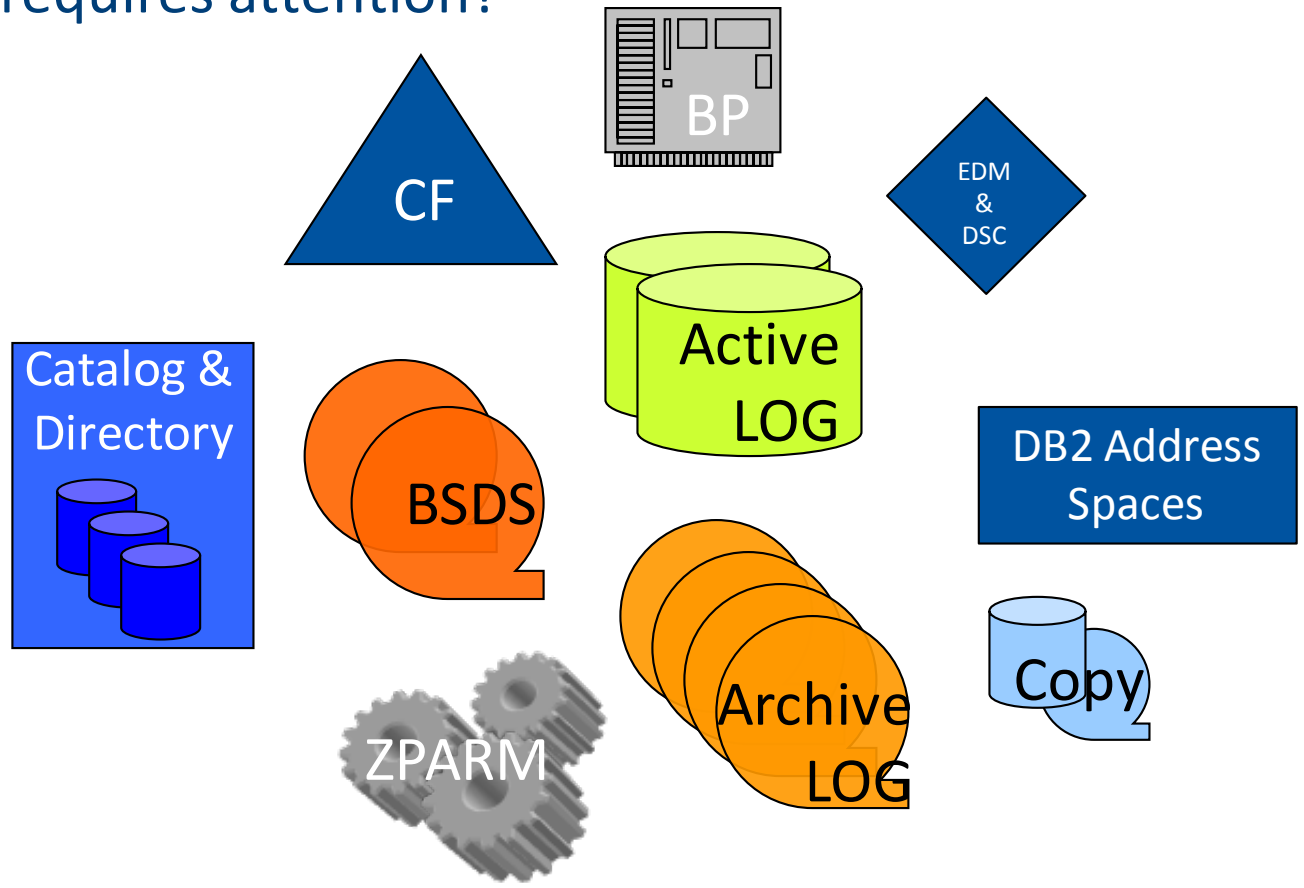
- The good news:
  - DB2 z/OS and the mainframe environment itself is a very robust and reliable platform
  - Todays IT components already handle several situations autonomously
- The bad news:
  - If something goes wrong, it often goes terribly wrong
  - Todays size and complexity of IT environments often eliminate the possibility of quick and easy fixes once a problem really hits you

DB2 Health Checking basics

- What requires attention?

DB2 Health Checking basics

- What requires attention?



DB2 Health Checking basics

- What happens if you don't take care?
  - Some areas immediately get your attention …
    - Availability
    - Performance
  - … other areas get your attention once you rely on them …
    - Recoverability
  - … some may never strike, or they do, all of a sudden
    - Security

→ Thus, do take care !

DB2 Health Checking without additional work

- How to automate regular procedures
  - Define rules following best practices and common rules of thumb that can be verified without manual work

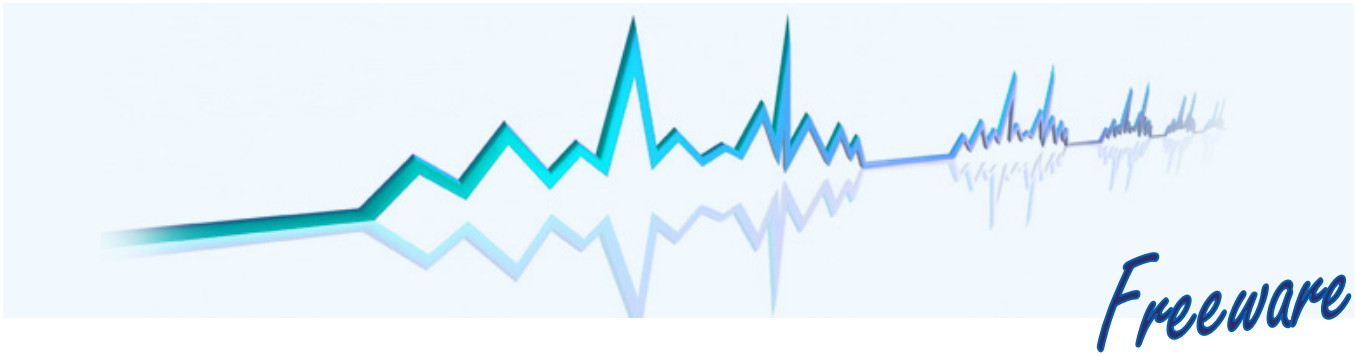
GroupId	Session Id	Severity	Attribute	SubAttribute	SubAttribute2	SubAttribute3	Setting	Operator	Value
DSG8	S814	WARNING	BP0	BPVPSEQ			80	=	100
DSG8	S814	WARNING	BP0	BPDWV1			5	>=	10
DSG8	S814	WARNING	BP0	BPHPSEQ			0	=	100
DSG8	S814	WARNING	BP16K0	BPHPSEQ			0	=	100
DSG8	S814	WARNING	BP16K0	BPVPSEQ			80	=	100
DSG8	S814	WARNING	BP16K0	BPDWV1			5	>=	10
DSG8	S814	WARNING	BP32K	BPVPSEQ			80	=	100
DSG8	S814	WARNING	BP32K	BPHPSEQ			0	=	100
DSG8	S814	WARNING	BP32K	BPDWV1			5	>=	10
DSG8	S814	WARNING	BP8K0	BPVPSEQ			80	=	100
DSG8	S814	WARNING	BP8K0	BPDWV1			5	>=	10
DSG8	S814	WARNING	BP8K0	BPHPSEQ			0	=	100
DSG8	S814	WARNING	GBP0	GBCLASS			5	>	5
DSG8	S814	WARNING	GBP0	GBCHECKP			50004	<=	4
DSG8	S814	WARNING	GBP16K0	GBCHECKP			50004	<=	4
DSG8	S814	WARNING	GBP16K0	GBCLASS			5	>	5
DSG8	S814	WARNING	GBP32K	GBCHECKP			50004	<=	4
DSG8	S814	WARNING	GBP32K	GBCLASS			5	>	5
DSG8	S814	WARNING	GBP8K0	GBCHECKP			50004	<=	4
DSG8	S814	WARNING	GBP8K0	GBCLASS			5	>	5
DSG8	S814	WARNING	GPSCAUSE				28	<=	10
DSG8	S814	WARNING	ZPARM	TWOARCH			YES	=	NO
DSG8	S814	WARNING	ZPARM	RETLWAIT			0	=	2
DSG8	S814	WARNING	ZPARM	LOGAPSTG			0	=	100
DSG8	S814	WARNING	ZPARM	CHKFREQ			50000	<=	6

DB2 Health Checking without additional work

- Implementing “notify and recommend” instead of having to identify and analyze manually
  - Schedule your health checks to run daily, weekly, or at least monthly
  - Assign severity levels to avoid having to review the output
  - Allow to suppress checks, **but fix problems**
  - Include notifications (e.g. via email) to let the system tell you if something requires attention
  - Understand and discuss the links between aspects, like recoverability and performance
  - Don’t be afraid to share your experts knowledge via rules
  - Verify and maintain your rule set *at least* with each new DB2 version

  → Consider the checks as an opportunity, not as risk management – health checks are your friend, there are so many “low hanging” fruits

Performance HealthCheck



Request your copy at http://www.segus.com
→ products → DB2 z/OS Products → Pocket Tools → Performance HealthCheck

Request the freeware @ segus.com

SEGUS

Call us at (800) 327-9650

contact login

home | products | support | consulting | events | corporate | partners

You are here: products | DB2 z/OS Products | Pocket Tools | Performance HealthCheck

Performance HealthCheck (Licensed Freeware)

Performance HealthCheck evaluates performance factors of a complete DB2 subsystem, or a DB2 group and each of its members. It presents an overview of critical factors that might affect the performance of the system, including the Coupling Facility (CF) in a data sharing environment. CF checks cover the level of the microcode, the transfer time, the number of rejects, the false contention percentage, the subchannel busy percentage and finally the all paths busy count.

The foundation of the extensive analysis of Performance HealthCheck is based on IBM's own recommendations and Best Practices.

Key Benefits:

- Evaluates performance critical components of a complete DB2 subsystem or a data sharing group
- Reports provide warnings about performance violations as well as general messages regarding the overall health of the system
- Checks the stress of the Coupling Facility

An extended version of this tool can be purchased to additionally present further details, identify and perform corrective actions to improve DB2's performance.

Freeware Request

Download the Freeware >>

Relevant Products

SQL PerformanceExpert (SPX)

Package Manager

Product Description

Performance HealthCheck

- How to evaluate performance critical components and what to include?

Query	Performance check
1	Mapping tables. Unless you are using a 3rd party tool that manages your mapping tables automatically or you are in DB2 11 NFM, you are faced with creating and maintaining them yourself
2	Page size checker. Large page sizes are good for storing tables with very long rows or tables used regularly for sequential data access. Conversely, storing small, randomly accessed rows in large pages makes for inefficient use of the buffer pools.
3	Segment size checker. If you segment table spaces, DB2 can determine which segments to read, but optimal SEGSIZEs are important. If the prefetch quantity is greater than the SEGSIZE and the segments for the table aren't contiguous, DB2 might read unnecessary pages.
4	Work and catalog buffer pool misuse. Work and sort buffer pools and DB2 catalog buffer pools must always be in their own separate buffer pools that should never be used for user objects.
5	SPT01 size. SPT01 is the directory table space used to store bound packages. If you currently use or plan to use PACKAGE STABILITY, the size of SPT01 will grow, especially when using the extended option which triples the size of SPT01.
6	Different versions of a package. On average, you should only have two or three versions of a package. An excessive number of packages means that package housekeeping is required to FREE unnecessary packages.
7	RELBOUND checker. The RELBOUND column of SYSIBM.SYSPACKAGE or SYSPLAN indicates the DB2 release number of when the package was bound or rebound.
8	Invalid or inoperative packages. A package become invalidated when a privilege that is required by the package is revoked. If proper privileges are granted, DB2 can automatically rebind the package.

Performance HealthCheck

- How to evaluate performance critical components and what to include?

Query	Performance check
9	AVGSIZE of Packages. This check shows you the AVGSIZE of packages in the DB2 subsystem. Too many small packages and/or too many large packages are not generally good for performance.
10	Packages not bound after last RUNSTATs. Current catalog statistics are food for the Optimizer and all performance decisions are based on these statistics. This check identifies the number of packages that were not been bound after the last RUNSTATS.
11	Days since last RUNSTATS. RUNSTATS are necessary to actualize the catalog statistics. It is important that the statistics represent the state of the data.
12	PLAN_TABLE owners. Usually, there must be at least one PLAN_TABLE per user. However, due to the fact that user PLAN_TABLE require a lot of maintenance, there are often many old ones hanging around.
13	Package and plan counts. Total number of packages and plans.
14	Report host languages. This is useful in knowing that packages that were precompiled with out-of-date host languages.
15	Dynamic statement cache stability. The dynamic statement cache is evaluated to determine the volume of activity and stability.
16	EXPLAIN(NO). EXPLAIN(YES) is always recommended when binding and rebinding. When EXPLAIN(NO) is used, there is no PLAN_TABLE data. Without PLAN_TABLE, SQL tuning opportunities are very limited.
17	Counts. Total number of databases, table spaces, tables, and indexes. (Not including Alias's, GTTs or Views)
18	Status H Package statements. Each of these causes an incremental BIND at run time. Now if you use SESSION tables then you must have these otherwise it is worth investigating further.

Performance HealthCheck

- How to evaluate performance critical components and what to include?

Query	Performance check
19	TRACKMOD and IIC counts. If you are a data-sharing shop and have TRACKMOD YES but do *not* take IICs at all then this is very bad for group performance.
20	Buffer pool definitions. Number of work buffer pools and how many thresholds are bad, i.e., VPSEQT not between 90 and 99, DWQT not 50, VDWQT not between 10 and 20.
21	Work datasets. The number defined should always be equal to the number found. The general rule of thumb says at least 6 work data sets should be physically there and found.
22	ZPARM checks. There are a number of performance-critical ZPARMS.

Performance HealthCheck

- Check the stress of the Coupling Facility
  - CF checks should cover the
    - level of the microcode
    - transfer time
    - number of rejects
    - false contention percentage
    - subchannel busy percentage
    - all paths busy count
- Remember that the CF might look great top but when it comes under stress it can „fall apart“ pretty quickly!

Performance HealthCheck – sample output

```
DB2 Release 1001 Mode NF detected.
REORG SHRLEVEL CHANGE mapping tables
Table(s) found:                    6
Tablespace(s) too large:           2
Index(s) too small:                6
PGSIZE  found to be inaccurate for       122 tablespace(s).
SEGSIZE found to be inaccurate for       104 tablespace(s).
Work/sort BUFFERPOOL BP32K  is used by        1 index(s).
Work/sort BUFFERPOOL BP1    is used by        1 index(s).
Work/sort BUFFERPOOL BP32K  is used by       64 tablespace(s).
Work/sort BUFFERPOOL BP1    is used by        2 tablespace(s).
Catalog   BUFFERPOOL BP0    is used by       54 index(s).
Catalog   BUFFERPOOL BP32K  is used by        1 index(s).
Catalog   BUFFERPOOL BP0    is used by       44 tablespace(s).
Catalog   BUFFERPOOL BP16K0 is used by       17 tablespace(s).
Catalog   BUFFERPOOL BP32K  is used by       68 tablespace(s).
Catalog   BUFFERPOOL BP8K0  is used by       41 tablespace(s).
SPT01 is currently  1589 MB (  1 GB)
Databases                                 :       88
Tablespaces                               :     3032
Tables                                    :     3510
Indexes                                   :     4181
Collections                               :      257
Packages                                  :     8120
```

Performance HealthCheck – sample output

```
Packages bound with EXPLAIN(NO)                 :     4363
Different versions of one package (min)         :        1
Different versions of one package (max)         :       12
Different versions of one package (avg)         :        1
Packages last bound in DB2 V0710                :       76
Packages last bound in DB2 V0810                :     3704
Packages last bound in DB2 V1010                :     4340
Invalid or inoperative packages                 :     1741
AVGSIZE of packages (min)                       :      964
AVGSIZE of packages (max)                       :   320294
AVGSIZE of packages (avg)                       :    11022
Packages not bound since the last RS (TS):      :     2727
Days since last RUNSTATS TS (min)               :        1
Days since last RUNSTATS TS (max)               :     5257
Days since last RUNSTATS TS (avg)               :      625
PLAN_TABLES in the catalog                      :       12
PLAN_TABLES used by packages / plans            :       10
     459 packages were precompiled with Unknown (Remote/Trigger).
    1689 packages were precompiled with VS COBOL II or IBM COBOL Rel.1.
    2504 packages were precompiled with IBM COBOL Release 2 or higher.
     134 packages were precompiled with C++.
    3119 packages were precompiled with Assembler language.
      10 packages were precompiled with OS/VS COBOL.
     178 packages were precompiled with C.
      27 packages were precompiled with PL/1.
```

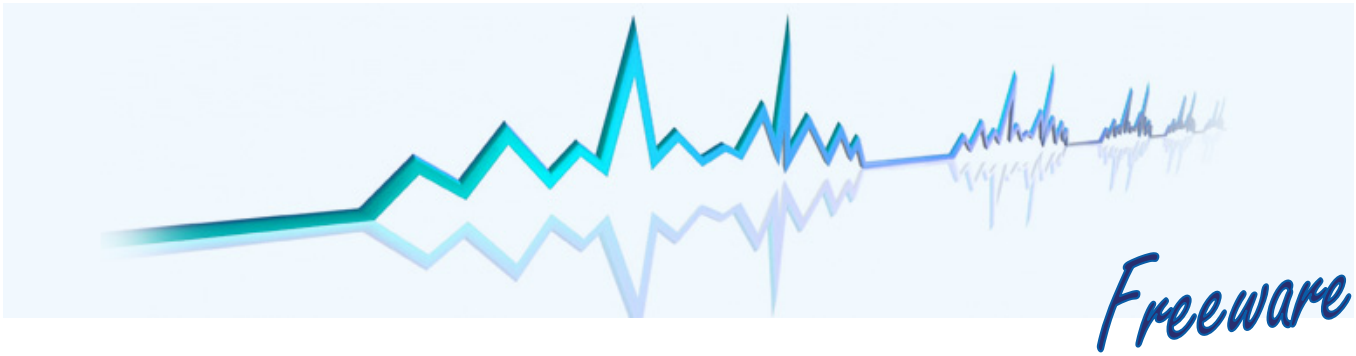
Performance HealthCheck – sample output

```
Dynamic Statement Cache Analysis (1/3)
SQL in the Dynamic Statement Cache       :    15483
First SQL inserted                       : 20140126112503226716
Last SQL inserted                        : 20140127090521769376
Lowest StmtID                            : 000000044
Highest StmtID                           : 000119329
Wait for  30 seconds
Dynamic Statement Cache Analysis (2/3)
SQL in the Dynamic Statement Cache       :    15533
First SQL inserted                       : 20140126112503226716
Last SQL inserted                        : 20140127090552174273
Lowest StmtID                            : 000000044
Highest StmtID                           : 000120075
Wait for 120 seconds
Dynamic Statement Cache Analysis (3/3)
SQL in the Dynamic Statement Cache       :    15906
First SQL inserted                       : 20140126112503226716
Last SQL inserted                        : 20140127090751702757
Lowest StmtID                            : 000000044
Highest StmtID                           : 000124063
Dynamic Statement Cache Stability is good. (No throwouts, growing)
Status H statements in packages          :      212
TRACKMOD YES declared partitions         :    10926
Number of Incremental Image Copies       :    71414
```

Performance HealthCheck – sample output

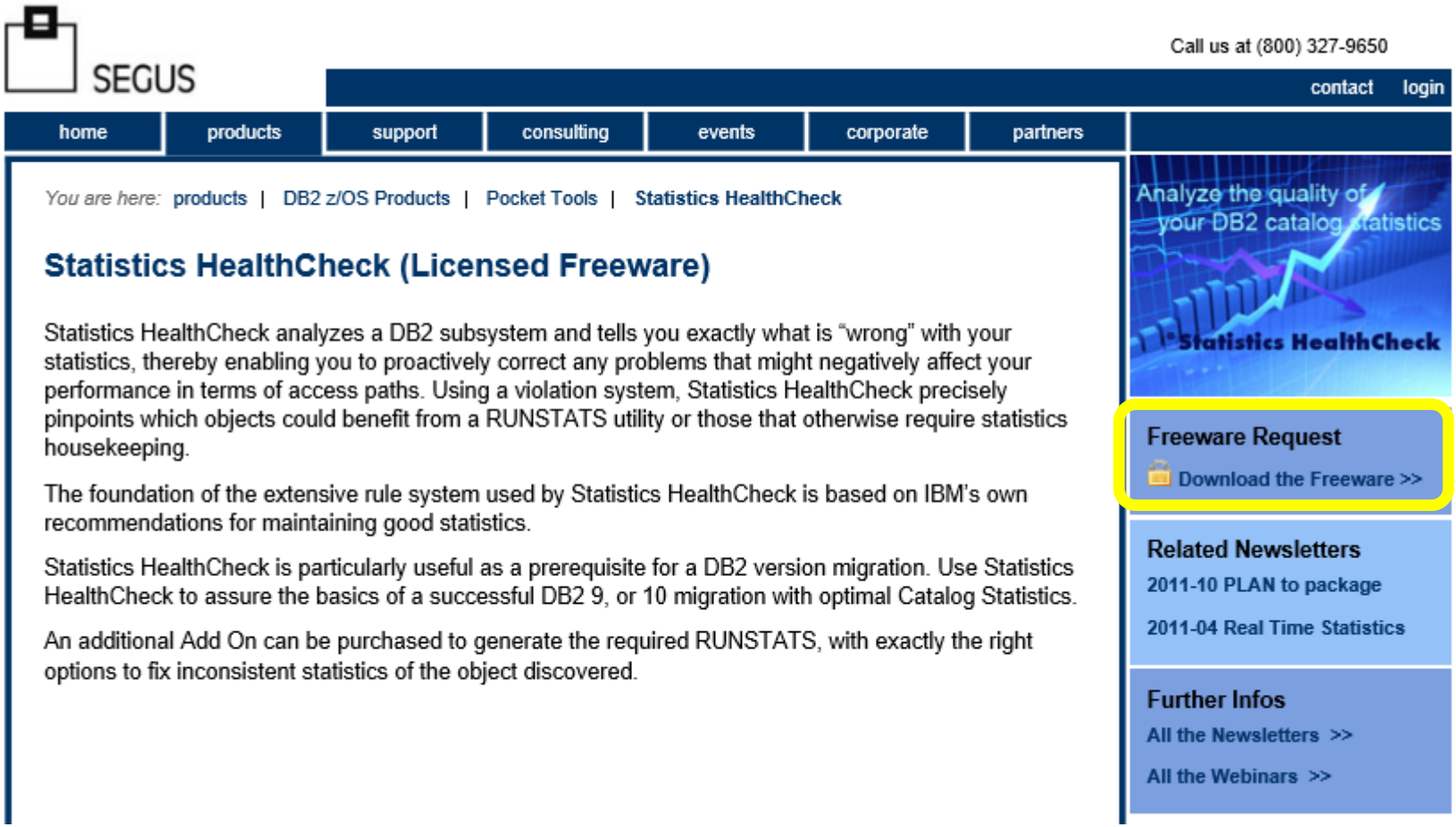
```
A total of        2 WORK BUFFERPOOL(s) have been found.
From these        4 thresholds are incorrectly defined.
WORK Database and tablespace overview.
Defined           9
Found             9
Bad               0
ZPARM MAXKEEPD is too low.
ZPARM MXDTCACH is too low.
Checked 4 ZPARMS and 2 were bad.
```

Statistics HealthCheck for DB2 z/OS



Request your copy at http://www.segus.com
→ products → DB2 z/OS Products → Pocket Tools → Statistics HealthCheck

Request the freeware @ segus.com



Statistics HealthCheck

- How to check the sanity of your catalog statistics?
  - Catalog tables important for performance
    - SYSIBM.SYSCOLDIST
    - SYSIBM.SYSCOLSTATS *
    - SYSIBM.SYSCOLUMNS
    - SYSIBM.SYSINDEXES
    - SYSIBM.SYSINDEXPART
    - SYSIBM.SYSKEYTARGETS 9 and above (same as SYSCOLUMNS)
    - SYSIBM.SYSKEYTGTDIST 9 and above (same as SYSCOLDIST)
    - SYSIBM.SYSROUTINES
    - SYSIBM.SYSTABLES
    - SYSIBM.SYSTABLESPACE
    - SYSIBM.SYSTABSTATS

* degree of parallelism only and, after APAR PK62804, also „sometimes“ used to bound filter factor estimates.

Columns used for access path decisions

SYSCOLDIST / SYSKEYTGTDIST
CARDF
COLGROUPCOLNO / KEYGROUPKEYNO
COLVALUE / KEYVALUE
FREQUENCYF
HIGHVALUE
LOWVALUE
NUMCOLUMNS / NUMKEYS
QUANTILENO
STATSTIME

SYSCOLUMNS / SYSKEYTARGETS
COLCARDF / CARDF
HIGH2KEY
LOW2KEY
n/a / STATS_FORMAT

SYSCOLSTATS
COLCARD
HIGHKEY
LOWKEY

SYSINDEXES
CLUSTERING*
CLUSTERRATIO
CLUSTERRATIOF
DATAREPEATFACTORF
FIRSTKEYCARDF
FULLKEYCARDF
NLEAF
NLEVELS

SYSINDEXPART
LIMITKEY*

* Columns are not updated by RUNSTATS
_ Columns are not updatable

SYSROUTINES
CARDINALITY*
INITIAL_INSTS*
INITIAL_IOS*
INSTS_PER_INVOC*
IOS_PER_INVOC*

SYSTABLES
CARDF
EDPROC*
NPAGES
NPAGESF
PCTROWCOMP

SYSTABLESPACE
NACTIVE
NACTIVEF

SYSTABSTATS
CARD
CARDF
NPAGES

Statistics HealthCheck

- How to check the sanity of your catalog statistics?

Table Name	Columns Used	Notes
SYSCOLUMNS	COLCARDF, HIGH2KEY, LOW2KEY	Number of distinct values for a column Second highest column value Second lowest column value
SYSCOLSTATS	COLCARD, HIGHKEY, LOWKEY	Only for degree of parallelism and used "sometimes" for Filter factor improvement. Introduced with APAR PK62804 for V8 and 9 only.
SYSINDEXES	CLUSTERING[2], CLUSTERRATIO, CLUSTERRATIOF, DATAREPEATFACTORF, FIRSTKEYCARDF, FULLKEYCARDF, NLEAF, NLEVELS	Is the data in the same order as the index? The old clusterratio but still used The actual ordering ratio. If CLUSTERRATIOF is <= 0 then use the CLUSTERRATIO and if that is <= 0 then if the index is CLUSTER YES use 0.95 else use 0.00 The number of times data pages are repeatedly scanned after the index key is ordered. Can only be between NPAGESF and CARDF. No. of distinct values of the first key column, No. of distinct values of the full key How many leaf pages are used. This is part of the I/O cost for index traversal How many levels the index has. Again this is used in the I/O cost calculation.
SYSINDEXPART	LIMITKEY[2]	The highest key value in a given partition
SYSKEYTARGETS	CARDF, HIGH2KEY, LOW2KEY STATS_FORMAT	Number of distinct values of key target Second highest key value Second lowest key value Blank – No stats or padded, N – Varchar are not padded

Statistics HealthCheck

- How to check the sanity of your catalog statistics?

Table Name	Columns Used	Notes
SYSKEYTGTDIST	CARDF,	For TYPE = C number of distinct values in key group, for TYPE= F -1, for TYPE = H the number of distinct values in this QUANTILENO
	KEYGROUPKEYNO	Column numbers of columns in the key
	KEYVALUE,	Frequently occurring value
	FREQUENCYF,	The frequency of KEYVALUE. For TYPE = H the percentage of rows between LOWVALUE and HIGHVALUE for QUANTILENO
	HIGHVALUE,	TYPE = H Highest boundary
	LOWVALUE,	TYPE = H Lowest boundary
	NUMKEYS,	Number of columns in key
	QUANTILENO,	Quantile number of histogram
	STATSTIME	Only used in the case of duplicates
SYSROUTINES	CARDINALITY,	How many rows will be returned
	INITIAL_INSTS,	How many instructions on first call
	INITIAL_IOS,	How many I/Os on first call
	INSTS_PER_INVOC,	How many instructions for all normal calls
	IOS_PER_INVOC	How many I/Os for all normal calls
SYSTABLES	CARDF,	Number of rows in the table
	EDPROC[2],	Name of the edit procedure if it exists
	NPAGES,	How many pages in the table
	NPAGESF,	If NPAGESF <= 0 then NPAGES is used
	PCTROWCOMP	How good the compression is
SYSTABLESPACE	NACTIVE,	How many pages for the complete table space. This is used for I/O cost.
	NACTIVEF	If NACTIVEF is <= 0 then NACTIVE is used
SYSTABSTATS	**CARDF,**	Number of rows in this partition
	NPAGES	Number of pages in this partition, again this is used for I/O cost

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - SYSCOLDIST in a nut shell:
    - SYSCOLDIST is used for:
      - Frequencies
      - Cardinalities
      - Histograms in 9 and above
    - Frequencies are good for - COL op literal
    - Histograms are possibly good for - COL op literal
      - But better for range predicates!
    - Cardinalities are good for - everything!
    - Once data is inserted in SYSCOLDIST it stays there until it is replaced! This can mean that very old data is lurking in this table and should be deleted.

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Understanding SYSCOLDIST content
    - Column TYPE can contain C, F, H or N (for non-padded frequency values)
    - Regardless of the TYPE value, columns TBOWNER, TBNAME, NAME, COLGROUPCOLNO, NUMCOLUMNS, and STATSTIME are used for the same purpose.
    - TBOWNER, TBNAME, and NAME (first column name only) columns are also the non-unique index.
    - COLGROUPCOLNO for a single column object is an empty string, for a multi-column object contains a string of two byte SMALLINT, which contains the column numbers from the original table.
    - NUMCOLUMNS is the number of columns in this group.
    - STATSTIME is the time when RUNSTATS inserted this entry, which is also used when there is a complete duplicate so that DB2 uses the last inserted value.

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Understanding SYSCOLDIST content
    - Frequency data:
    - COLVALUE contains the actual data from the 1 – n columns in the group simply concatenated together.
      - This data might be readable or it might not.
      - Caution must be used when inserting or changing this data as numeric and date, time forms must be the internal DB2 format and not the external format
      - For example, the high bit must be flipped for numbers and watch out for NULLable columns!
    - FREQUENCYF contains a floating point value between 0.0 and 1.0 , which is the frequency that this value occurs for this group in the table.

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Understanding SYSCOLDIST content
    - For cardinality data:
      - CARDF contains a floating point value
      - Which is how many different values for this group occur in the data.

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Understanding SYSCOLDIST content
    - For histogram data:
      - COLVALUE contains the actual data from the 1 – n columns in the quantile simply concatenated together.
        - This data might be readable or it might not.
        - Caution must be used when inserting or changing this data as numeric and date, time forms must be the internal DB2 format and not the external format
        - For example,the high bit must be flipped for numbers and watch out for NULLable columns!
      - FREQUENCYF contains a floating point value between 0.0 and 1.0 which is the frequency that this value occurs for this quantile.
      - HIGHVALUE and LOWVALUE contain the upper and lower boundaries of this QUANTILENO.
      - QUANTILENO is the ordinary number of this quantile (1 - 100)

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Understanding SYSCOLDIST content
    - Example: Importance of cardinality and frequency. If no frequency data existed in the SYSCOLDIST DB2 would assume that all values are equally distributed.
    - Default filter factor would be 1/5 (1/COLCARDF) or 20%

Category	Frequency
Infant	5%
Child	15%
Adolescent	25%
Adult	40%
Senior	15%

This would lead the optimizer to under estimate by 50% for ADULT and to over estimate by 400% for INFANT.

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Why SYSCOLDIST may need to be cleaned up
    - A colleague is on holiday and the telephone rings with end-users or management saying that the system performance is bad. You grab your RUNSTATS JCL, submit the job and all is well…

… or is it?

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - Here is the query that you have found.

```
SELECT A.ICTYPE , A.TIMESTAMP
FROM SYSIBM.SYSCOPY A
WHERE A.ICTYPE IN ('D','I','F','S','W','Y','R','X','Z')
  AND A.ICBACKUP NOT IN ('RP','RB')
  AND A.DBNAME  = ?
  AND A.TSNAME  = ?
  AND (A.DSNUM  = ? OR (? > 0 AND A.DSNUM = 0))
  AND A.TIMESTAMP = (SELECT MAX(B.TIMESTAMP)
                     FROM SYSIBM.SYSCOPY B
                     WHERE B.ICTYPE IN
    ('D','I','F','S','W','Y','R','X','Z')
                       AND B.ICBACKUP NOT IN ('LB','RP','RB')
                       AND B.DBNAME  = ?
                       AND B.TSNAME  = ?
                       AND (B.DSNUM  = ? OR (? > 0 AND B.DSNUM =
   0))    )
WITH UR;
```

  - What RUNSTATS is needed?

Statistics HealthCheck

- How to clean up your SYSCOLDIST?
  - This is the actual RUNSTATS you should run…

```
RUNSTATS TABLESPACE DSNDB06.SYSCOPY
         TABLE(SYSIBM.SYSCOPY)
         COLGROUP(ICBACKUP) FREQVAL COUNT 10
         COLGROUP(ICTYPE) FREQVAL COUNT 1
         COLGROUP(DSNUM) FREQVAL COUNT 1
         SORTDEVT SYSDA
SHRLEVEL CHANGE REPORT YES
```

Statistics HealthCheck

- Minimal invasive SQL performance optimization made easy
  - Quickly, and effectively, analyzes and judges the quality of your entire DB2 catalog
  - Knows the heightened sensitivity within DB2 with regard to bad statistics
  - All current DB2 versions are supported
    - DB2 9
    - DB2 10
    - DB2 11
  - Reporting based on an easy to use batch analysis

Statistics HealthCheck

- Minimal invasive SQL performance optimization made easy
  - The DB2 Optimizer expects different statistics to „fit“ to each other
  - Relationships exist among certain columns of certain tables, below are some examples:
    - Columns within SYSCOLUMNS
    - Columns in the tables SYSCOLUMNS and SYSINDEXES
    - Columns in the tables SYSCOLUMNS and SYSFIELDS
    - CARDF must be equal or larger than any other cardinalities, such as COLCARDF, FIRSTKEYCARDF, FULLKEYCARDF, and CARDF in SYSCOLDIST
    - The number of frequencies collected must be less than or equal to COLCARDF for a column, or CARDF for a column group
    - The sum of frequencies collected for a column or column group must be less than or equal to 1

→ **Statistics HealthCheck verifies and reports more than 80 conditions**

Statistics HealthCheck – sample output

```
Tables read . . . . . . . . . . . . . . . . . . . . : 1286
Tables ignored . . . . . . . . . . . . . . . . . . . : 0
Tables checked . . . . . . . . . . . . . . . . . . . : 1286
Perfect tables . . . . . . . . . . . . . . . . . . . : 1
Indexes checked . . . . . . . . . . . . . . . . . . . : 1731
Columns checked . . . . . . . . . . . . . . . . . . . : 22332
Number of critical rule violations . . . . . . . . . : 35
Number of serious rule violations . . . . . . . . . . : 821
Number of warning rule violations . . . . . . . . . . : 3815
Number of informational rule violations . . . . . : 25701
Total number of rule violations . . . . . . . . . . : 30372
Number of other findings . . . . . . . . . . . . . . : 302

Critical violations
Frequency < 0 . . . . . . . . . . . . . . . . . . . . : 0
Frequency > 100 . . . . . . . . . . . . . . . . . . . : 0
Frequency count > cardf . . . . . . . . . . . . . . . : 0
Frequency count > colcardf . . . . . . . . . . . . . : 2
Frequency sum < 0 . . . . . . . . . . . . . . . . . . : 0
Frequency sum > 100 . . . . . . . . . . . . . . . . . : 0
SYSCOLDIST cardf outside allowable range . . . . . : 0
SYSCOLUMNS - low2key high2key empty . . . . . . . . : 4
SYSCOLUMNS aggregate colcardf < SYSCOLDIST cardf . : 0
SYSCOLUMNS 1st cardf <> SYSINDEXES firstkeycardf . : 0
SYSCOLUMNS 1st cardf <> SYSINDEXES fullkeycardf . . : 1
SYSINDEXES 1 col ix firstkeycardf <> fullkeycardf: 4
SYSTABLES cardf < SYSCOLDIST cardf . . . . . . . . : 0
SYSTABLES cardf < SYSCOLUMNS colcardf . . . . . . . : 18
SYSTABLES cardf < SYSINDEXES firstkeycardf . . . . : 0
SYSTABLES cardf < SYSINDEXES fullkeycardf . . . . . : 1
SYSTABLES no RUNSTATS . . . . . . . . . . . . . . . : 5
```

Statistics HealthCheck – sample output

```
Serious violations
Missing leading column index frequencies . . . . : 0
Missing multi-column index cardinalities . . . . : 820
SYSCOLDIST cardf group <> SYSINDEXES fullkeycardf: 0
SYSCOLDIST cardf sub-set > super-set . . . . . : 0
SYSCOLUMNS - ALTERs . . . . . : 0
SYSINDEXES fullkeycard < SYSINDEXES firstkeycard : 0
SYSTABLES cardf <> unique index fullkeycardf . . : 1

Informational violations
Number of index parts <> number of parts . . . . : 0
SYSCOLDIST colgroup with a default colcardf . .  : 0
SYSCOLDIST colno > number of columns . . . . . : 0
SYSCOLDIST duplicate cardinalities . . . . . : 0
SYSCOLUMNS - low2key > high2key . . . . . : 1
SYSCOLUMNS - SYSINDEXES statstime not equal . .  : 4456
SYSINDEXPART - SYSINDEXES statstime not equal .  : 1735
SYSINDEXPART statistics missing . . . . . : 0
SYSTABLES - SYSCOLUMNS statstime not equal . . . : 19470
SYSTABLES - SYSINDEXES statstime not equal . . . : 19
SYSTABLES - SYSTABLESPACE statstime not equal .  : 0
SYSTABLES - SYSTABSTATS statstime not equal . .  : 0
SYSTABLES cardf < sum of SYSINDEXPART cardf . .  : 0
SYSTABLES cardf > sum of SYSINDEXPART cardf . .  : 0
SYSTABLES cardf < SYSINDEXPART cardf . . . . . : 20
SYSTABLESPACE - SYSTABSTATS no. parts not equal  : 0

Other findings
SYSCOLDIST not used in index cardinality records : 0
SYSCOLDIST not used in index frequency records . : 10
SYSINDEXES clustering = Y and clustered = N . .  : 292
```

```
Warning violations
Missing multi-column index frequencies . . . . . : 3635
SYSCOLDIST large differences found in STATSTIME  : 4
SYSCOLDIST statistics obsolete . . . . . : 42
SYSROUTINES with default statistics . . . . . : 96
SYSTABLES - SYSTABSTATS sum cardf not equal. . . : 0
SYSTABLES - SYSTABSTATS sum npages not equal . . : 0
SYSTABLES default cardf . . . . . : 0
SYSTABLES statistics obsolete . . . . . : 38
SYSTABLESPACE - SYSTABSTATS sum nactive not equal: 0
```

Summary

- Certain areas of DB2 should be carefully & regularly checked
- Rule based systems, along with severity levels, allow a high degree of automation
- Notification capabilities forward violations to the appropriate people in charge, without them having to review excessive reports every day
  - → Health Checks don‘t necessarily create additional work, but spot what requires your attention and let you act proactively
  - → Low hanging fruits are pinpointed and open up great opportunities
  - → The right checks make your life easier and significantly reduce situations where your critical applications hit weaknesses

Ulf Heinrich
SEGUS Inc.
u.heinrich@seg.de

B05
Are you sick of health checks?!

Please fill out your session evaluation before leaving!