



DB2 UDB: Using Predictive Query Governor



Skip Marchesani

**Custom Systems Corp
Sparta, NJ 07871**

973-579-1340

smarches@warwick.net



Copyright Custom Systems Corp 2006

DB2 UDB: Using Predictive Query Governor

Disclaimer:

This presentation may contain examples of code and names of companies or persons. The code is given for presentation purposes and has not been tested by IBM and/or Custom Systems Corp. Therefore IBM and/or Custom Systems Corp does not guarantee the reliability, serviceability, or function of the code and the code is provided "AS IS". IBM AND/OR CUSTOM SYSTEMS CORP EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND MERCHANTABILITY. Any names appearing in this presentation are designed to be fictitious and IBM and Custom Systems Corp makes no representations as to the accuracy of the names or data presented in accordance therewith.

Reproduction:

This presentation is the property of Skip Marchesani and Custom Systems Corporation. Permission is granted to make a limited number of copies of this material for non-commercial purposes, providing this page is included with all copies. Express written permission is required for making copies for other purposes.

iSeries, AS/400, OS/400, DB2 UDB, DB2/400 are registered trademarks of the IBM Corporation

Copyright Custom Systems Corp 2006

Overview

- Background
- What's a Query Governor?
- DB2 UDB Predictive Query Governor
- Setting Query Processing Time limit
- New System Value and CL Command
- Inquiry Message CPA4259
- System Reply List
- Tuning a Query using Predictive Query Governor
- Implementation Considerations
- Summary
- V5R2 SQL Information Sources



Copyright Custom Systems Corp 2006

Background

- Ever since the introduction of Query on the S/38 in 1979, users have been asking for a way to throttle a run-a-way query - aka 'The Query that ate New York'

And optionally.....throttle the throat of the end user who created the run-a-way query

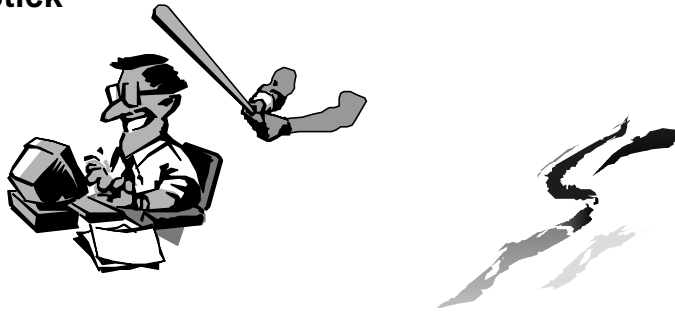


Copyright Custom Systems Corp 2006

Background...

The Solution

- **For the run-a-way query**
 - Use a Query Governor
- **For the end user that creates the run-a-way monster**
 - Use a Big Stick



Copyright Custom Systems Corp 2006

What is a Query Governor?

- **A Query Governor is a mechanism that can throttle or stop the processing of a query that will consume system resources in excess of predetermined limits.**
- **Types of governors**
 - Active
 - Predictive



Copyright Custom Systems Corp 2006

Active Query Governor

- **An active query governor stops the processing of a query after a predetermined resource limit has been reached**
- **Typical resource limits**
 - CPU seconds used
 - Number of records read
 - Number of report lines to print



Copyright Custom Systems Corp 2006

Active Query Governor...

The Active Query Governor Dilemma

- **If a query being processed is canceled when the resource limit is reached, the system resources consumed by the query in reaching the limit cannot be recovered**
- **This means that CPU cycles and disk I/Os are wasted on a query that does not produce useable results!**



Copyright Custom Systems Corp 2006

Predictive Query Governor

- **A predictive query governor estimates the amount of system resource that a query will consume and does not let the query begin processing if that estimate exceeds a predetermined resource limit**
- **Typical resource limit**
 - **Number of CPU seconds to process query**



Copyright Custom Systems Corp 2006

Predictive Query Governor...

Resolves the Active Query Governor Dilemma

- **Takes action BEFORE query begins processing - not DURING query processing**
- **Because the query is not allowed to begin processing, there is no waste of system resources**

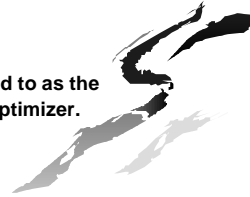


Copyright Custom Systems Corp 2006

Predictive Query Governor...

- **Predictive Query Governor is based on the number of CPU seconds required to process a query**
 - **Database optimizer estimates number of CPU seconds required to process a query**
 - **Database administrator(?) establishes & sets number of CPU seconds for query processing time limit**
 - **System level**
 - **Job level**

Note: there is only one optimizer in DB2 UDB. It is often referred to as the query optimizer, database optimizer, SQL optimizer, as well as optimizer. All terms refer to the single optimizer used in DB2 UDB.



Copyright Custom Systems Corp 2006

How It Works

- **Predictive Query Governor compares optimizer estimate with user defined query processing time limit**
- **If estimate EXCEEDS time limit**
 1. **DO NOT start query**
 2. **Send inquiry message CPA4259 to user**
 - **Message reply options**
 - I - ignore message and begin query**
 - C - cancel query**
- **If estimate is equal to or LESS THAN time limit**
 - **Let query run uninterrupted**



Copyright Custom Systems Corp 2006

Setting Query Processing Time Limit

- New command:

- Changes query processing time limit at job level

CHGQRYA - Change Query Attributes

- New System Value:

- Changes query processing time limit at system level

QQRYSIMLMT - Query Processing Time Limit



Copyright Custom Systems Corp 2006

CHGQRYA - New Command

- Change Query Attributes

CHGQRYA QRYTIMLMT(20)

```

Change Query Attributes (CHGQRYA)

Type choices, press Enter.

Job name . . . . . * _____ Name, *
User . . . . . _____ Name
Number . . . . . _____ 000000-999999
Query processing time limit . . 20 _____ 0-2147352578 seconds, *SAME...
Parallel processing degree:
Processing option . . . . . *SYSVAL_   *SAME, *NONE, *IO...
Number of tasks . . . . . _____ 2-99
Asynchronous job usage . . . . *DIST_   *SAME, *DIST, *LOCAL, *ANY...
Apply CHGQRYA to remote . . . . *YES_    *SAME, *YES, *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Note: 2,147,352,578 seconds = 596,487 hours = 24,854 days = 68 years

Copyright Custom Systems Corp 2006

QQRYTIMLMT - New System Value

- Query processing time limit

CHGSYSVAL QQRYTIMLMT VALUE(20)

```
Display System Value

System value . . . . . : QQRYTIMLMT
Description . . . . . : Query processing time limit

Query time limit
  allowed . . . . . : 20_____ *NOMAX, 0-2147352578

Press Enter to continue.

F3=Exit  F12=Cancel
```

Copyright Custom Systems Corp 2006

Inquiry Message CPA4259

- CPA4259 is issued when estimated query processing time exceeds query processing time limit
- Message text:
Estimated query processing time 30 exceeds limit 20
- Reply options:
I - ignore message and begin query processing
C - cancel query
- Default reply is C - cancel query



Copyright Custom Systems Corp 2006

Inquiry Message CPA4259...

- Reply can be controlled with INQMSGRPY parameter on the following commands:
 - CRTJOB
 - CHGJOB
 - CHGJOB
 - SBMJOB
 - BCHJOB



Copyright Custom Systems Corp 2006

Values for INQMSGRPY Parameter

- *RQD (default value)
 - End user sees message and must reply to the inquiry
- *DFT
 - End user does not see message and the default reply is used
 - For CPA4259 default reply is C and query is canceled
- *SYSRPLY
 - System reply list is used to determine message reply



Copyright Custom Systems Corp 2006

System Reply List

- **Used for inquiry messages only**

- **Reply list contains**
 - **Message IDs listed in reply list sequence number**
 - **Optional compare data based on character string found positionally in message data**
 - **Message reply value**
 - ***DFT (default value)**
 - ***RQD**
 - **I, C, R, etc.**
 - **Dump attribute**



Copyright Custom Systems Corp 2006

System Reply List...

- **Allows one or more message replies to be customized for a specific inquiry message.**
- **Message reply value for a given reply list entry is used when the reply list compare data has equal match with the designated character string in the message data**
- **Starting with lowest sequence number, first match found is used**
- **For CPA4259 reply list message reply can be based on**
 - **Job or process name**
 - **User ID**



Copyright Custom Systems Corp 2006

Message Data for CPA4259

- Contains fully qualified job name
- See 2nd panel of Help/2nd level text for CPA4259

```
Additional Message Information

Message ID . . . . . : CPA4259
Date sent . . . . . : 10/08/96      Time sent . . . . . : 19:02:37

Message . . . . . : Estimated query processing time 3 exceeds limit 0 (C I)

Cause . . . . . : The database query time limit attribute has been specified
                  by the CHGQRYA CL command. The query time limit specified is 0 seconds. A
                  database query was about to be started which was estimated to require a
                  longer time to run than that allowed by the query time limit attribute. The
                  estimated time is 3 seconds.
Recovery . . . . . : Type a C to cancel processing of this query, or type an I
                  to continue processing the query.
Possible choices for replying to message . . . . . :
C -- Cancel the running of the query.
I -- Ignore the query time limit and run the query.

Reply . . . . . : C
Press Enter to continue.

F1=Help  F3=Exit  F6=Print  F9=Display message details
F10=Display messages in job log  F12=Cancel  F21=Select assistance level

More...
```

Copyright Custom Systems Corp 2006

Message Data for CPA4259...

- 2nd panel of Help/2nd level text for CPA4259

```
Additional Message Information

Message ID . . . . . : CPA4259
Date sent . . . . . : 10/08/96      Time sent . . . . . : 19:02:37

Technical description . . . . . : The following information explains
the type of access plan used by the query. This information is similar to
that which can be printed using the PRTSQLINF CL command.
├───▶
      DSP04          SKIP          089619
Access plan last saved on 10/08/96 at 19:02:36.
Access plan was saved with DB2 Symmetric Multiprocessing for OS/400
installed on the system.
Query optimizer timed out for file 1.
Access path DESCSTU used for file 1.

Reply . . . . . : C
Press Enter to continue.

F1=Help  F3=Exit  F6=Print  F9=Display message details
F10=Display messages in job log  F12=Cancel  F21=Select assistance level

More...
```

Copyright Custom Systems Corp 2006

Message Data for CPA4259...

- Message data contents and start positions

```

Display Data
Position to line . . . . . _____ Data width . . . . . : 128
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....
Message Data
  Ē   ā           DSP04           SKIP
089619
***** End of data           ↑           ↑
                               Pos 27       Pos 51
                               Bottom
F3=Exit  F12=Cancel  F19=Left  F20=Right  F21=Split
    
```

- Pos 27 - Job or process name
- Pos 51 - User ID

Copyright Custom Systems Corp 2006

Setting Up Reply List Entries

- Default reply list entries as shipped with OS/400

```

Work with System Reply List Entries
System: SPL335
Type options, press Enter.
  2=Change  4=Delete

Sequence  Message
Opt  Number  ID  Reply  Compare Value  Compare Start  Dump
-    -      -   -      -             -
-    10     CPA0700  D      *NONE          *YES
-    20     RPG0000  D      *NONE          *YES
-    30     CBE0000  D      *NONE          *YES
-    40     PLI0000  D      *NONE          *YES

Parameters or command
====> _____
F3=Exit  F5=Refresh  F6=Add  F11=Display entire reply  F12=Cancel
F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 1996.
Bottom
    
```

Copyright Custom Systems Corp 2006

Setting Up Reply List Entries...

Who gets what message reply?

- **SKIP** should always be able to respond to CPA4259
- **AirHead** should always have his queries canceled
- **All other users on workstations that have names DSP01 thru DSP99** should get the default reply for CPA4259



Copyright Custom Systems Corp 2006

Setting Up Reply List Entries...

- **Add new entries for CPA4259**

```
ADDRPYLE SEQNBR( 50 ) MSGID( CPA4259 )  
  CMPDTA( SKIP 51 ) RPY( *RQD )
```

```
ADDRPYLE SEQNBR( 60 ) MSGID( CPA4259 )  
  CMPDTA( AirHead 51 ) RPY( C )
```

```
ADDRPYLE SEQNBR( 70 ) MSGID( CPA4259 )  
  CMPDTA( DSP 27 ) RPY( *DFT )
```



Copyright Custom Systems Corp 2006

Setting Up Reply List Entries...

- Updated reply list entries

```

Work with System Reply List Entries
System: SPL335

Type options, press Enter.
2=Change 4=Delete

Sequence Message
Opt Number ID Reply Compare Value Compare Start Dump
- 10 CPA0700 D *NONE *YES
- 20 RPG0000 D *NONE *YES
- 30 CBE0000 D *NONE *YES
- 40 PLI0000 D *NONE *YES
- 50 CPA4259 *RQD SKIP 51 *NO
- 60 CPA4259 C AirHead 51 *NO
- 70 CPA4259 *DFT DSP 27 *NO

Parameters or command
====>
F3=Exit F5=Refresh F6=Add F11=Display entire reply F12=Cancel
F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 1996.

```

Copyright Custom Systems Corp 2006

Using the Reply List

- Change INQMSGRPY parameter to *SYSRPLY for jobs to be controlled by reply list
- Methods to change INQMSGRPY parameter
 - Commands
 - CRTJOB
 - CHGJOB
 - SBMJOB
 - BCHJOB
 - At end user sign-on via initial program in user profile
 - User written code
 - Stored procedures for client server applications



Copyright Custom Systems Corp 2006

Using the Reply List...

- For SKIP and AirHead
 - CHGJOB for respective job description
- For all other users
 - Initial program in user profile
 - CHGJOB if number of job descriptions is small



Copyright Custom Systems Corp 2006

Tuning a Query Using Predictive Query Governor

- Predictive Query Governor can be used to tune a query without actually doing the query processing
 - Conserves CPU cycles and disk I/Os
- Good technique to use on queries that will be used on a frequent or repetitive basis
- How much resource will a (first time) query use?
- Will work with ANY iSeries or AS/400 query product



Copyright Custom Systems Corp 2006

To Tune a Query

1. Set query processing time limit to zero
CHGQRYA QRYTIMLMT(0)
2. Start query to be tuned
3. When CPA4259 issued reply with 'C' to cancel
4. Review optimizer messages in job log for index recommendations (HELP to see 2nd level text)
5. Create recommended index/access path(s)
6. Re-run query to verify improvement



Copyright Custom Systems Corp 2006

Tuning an SQL Query

- SQL ordered SELECT statement
- Select all records in file ENAMES

```
SELECT * FROM ENAMES ORDER BY STUDNT DESC
```

```

                                Command Entry                                SPL335
                                                                Request level:  1
All previous commands and messages:
> chgqrya qrytimlmt(0)
> strsql
CONNECT to relational database SPL335 completed.
Current connection is to relational database SPL335.
Estimated query processing time 21 exceeds limit 0 (C I)
? C
➔ All access paths were considered for file ENAMES.
Access path built for file ENAMES.
Estimated query processing time 21 exceeds limit 0.
Connection to relational database SPL335 ended.
SQL cursors closed.
                                                                Bottom
Type command, press Enter.
====>
F3=Exit   F4=Prompt   F9=Retrieve   F10=Exclude detailed messages
F11=Display full   F12=Cancel   F13=Information Assistant   F24=More keys
    
```

Copyright Custom Systems Corp 2006

'Access Path Built....' 2nd Level Text (Press F1 or HELP)

- **CPI4321 - Access path built for file ENAMES**

```
Additional Message Information
Message ID . . . . . : CPI4321
Date sent . . . . . : 10/03/96      Time sent . . . . . : 16:25:39
Message . . . . . : Access path built for file ENAMES.

Cause . . . . . : A temporary access path was built to access records from
member ENAMES of file ENAMES in library ECROCHLIB for reason code 1. This
process took 0 minutes and .0 seconds. The access path built contains 0
entries. The reason codes and their meanings follow:

    1 - Perform specified ordering/grouping criteria.
    2 - Perform specified join criteria.
    3 - Perform specified record selection to minimize I/O wait time.

The access path was built using the following key fields. The key fields
and their corresponding sequence (ASCEND or DESCEND) will be shown:
→  STUDNT      DESCEND.

Press Enter to continue.

F1=Help  F3=Exit  F6=Print  F9=Display message details
F10=Display messages in job log  F12=Cancel  F21=Select assistance level
```

Copyright Custom Systems Corp 2006

Create Index and Re-Run SQL Query

- **Create recommended index on ENAMES**
CREATE INDEX DESCSTU ON ENAMES (STUDNT DESC)
- **Re-run SQL query**
SELECT * FROM ENAMES ORDER BY STUDNT DESC
- **Review messages in job log to verify improvement**



Copyright Custom Systems Corp 2006

Review Messages in Job Log

```

                                Command Entry                                SPL335
                                                                Request level: 1

All previous commands and messages:
> chgqrya qrytimlmt(0)
> strsql
CONNECT to relational database SPL335 completed.
Current connection is to relational database SPL335.
Estimated query processing time 21 exceeds limit 0 (C I)
? C
All access paths were considered for file ENAMES.
Access path built for file ENAMES.
Estimated query processing time 21 exceeds limit 0.
Estimated query processing time 21 exceeds limit 0.
> strsql
CONNECT to relational database SPL335 completed.
→ Current connection is to relational database SPL335.
Estimated query processing time 3 exceeds limit 0 (C I)
? C
→ Query optimizer timed out for file ENAMES.
Access path of file DESCSTU was used by query.
Estimated query processing time 3 exceeds limit 0.
Estimated query processing time 3 exceeds limit 0.

Bottom
Type command, press Enter.
====>

```

```

F3=Exit    F4=Prompt  F9=Retrieve  F10=Exclude detailed messages
F11=Display full  F12=Cancel  F13=Information Assistant  F24=More keys

```

Copyright Custom Systems Corp 2006

SQL Query Performance Improvement

- Without DESCSTU index - 21 seconds of CPU required
- Using DESCSTU index - 3 seconds of CPU required
- 7 to 1 improvement in query CPU time using Predictive Query Governor for tuning
- Remember:
 - Use to tune repetitive or frequently run queries
 - Determine how much resource a (first time) query will use?



Implementation Considerations

- **Optimizer estimate is only an ESTIMATE**
 - Actual mileage may vary...
- **Predictive query governor**
 - **Has minimal impact on system resources**
 - **Takes action before query starts processing**
 - **Can be used with client applications that query database on iSeries or AS/400 server**



Copyright Custom Systems Corp 2006

Implementation Considerations

Query Processing Time Limit

- **Remains in effect for duration of job or interactive session, or until changed by CHGQRYA command**
- **Is dynamic and can be changed under program control based on**
 - **Application**
 - **Time of day, week, month, year**
 - **System resources available**
- **One job can change time limit for another job**



Copyright Custom Systems Corp 2006

Summary

- A predictive query governor estimates the amount of system resource that a query will consume and does not let the query begin processing if that estimate exceeds a predetermined resource limit
- DB2 UDB resource limit is CPU seconds to process query



Copyright Custom Systems Corp 2006

Summary...

- Predictive Query Governor resolves the Active Query Governor Dilemma
 - Takes action BEFORE query begins processing, not DURING query processing
 - Because the query is not allowed to begin processing, there is no waste of system resources
- Predictive Query Governor can be used to tune a query without actually processing the query



Copyright Custom Systems Corp 2006

Summary...

- If Predictive Query Governor is not the answer
 - Remember the big stick!



Copyright Custom Systems Corp 2006

V5R2 SQL Information Sources

- iSeries Information Center Publications - Web or CD
 - SQL Reference
 - SQL Programming Concepts
 - SQL Programming with Host Languages
 - Query Manager Use
 - SQL Messages and Codes
- To access Info Center on the Web
 - <http://publib.boulder.ibm.com/series/v5r2/ic2924/index.htm>
 - In left scroll bar
 - Click on Database - 3rd line from top
 - Click on Manuals
 - Use right scroll bar to scroll down to above SQL publication
- DB2 UDB for iSeries on the Web
 - <http://www.ibm.com/servers/eserver/series/db2/>

Copyright Custom Systems Corp 2006