

ISUG-TECH 2015

Conference

SAP IQ Troubleshooting
Saroj Bagai

Agenda

- ❖ **Welcome**
- ❖ **Speaker Introduction**
- ❖ **SAP IQ Troubleshooting**
- ❖ **Q&A**

Saroj Bagai

- ❖ **Support Architect**
- ❖ **Over 17 years with Sybase/SAP**
- ❖ **17 years working with SAP IQ, SAP ASE and Replication Server**
- ❖ **Areas of expertise in performance and tuning, disaster recovery and critical problem resolution.**

Takeaways from this presentation

- ❖ An understanding of IQ diagnostics log files
- ❖ How to resolve IQ Resource and Operation issues.
- ❖ How to do IQ forced recovery
- ❖ Troubleshooting IQ multiplex issues and RLV issues
- ❖ Troubleshoot IQ Performance issues

IQ Diagnostics Logs

IQ Message Log

- ❖ **IQ_SYSTEM_MSG** is a system dbspace, it points to data file (.iqmsg) path of the IQ database message log.
- ❖ On Simplex and Coordinator Server, default message log file name is <dbname>.iqmsg. Default message log file name for Secondary server is <servername>.iqmsg.
- ❖ Default location for message log is .db(catalog db) directory. Location for the message log can be changed using 'Restore' or 'Alter Dbspace' statement
- ❖ **Syntax:**

Alter dbspace IQ_SYSTEM_MSG rename <new-file-pathname>
- ❖ **Example:**

alter dbspace IQ_SYSTEM_MSG rename '/work/IQ-16_0/demo/newdir/iqdemo.iqmsg'
- ❖ **Stop and Restart IQ Server**

IQ Message Log

❖ Message Log Start Up Switches

- iqmsgnum <num>

- Note: IQMsgNumFiles Server property corresponds to –iqmsgnum startup switch. Takes precedence over the value of –iqmsgnum.

- Specifies the number of archives of old message log maintained by the IQ Server. Allowed values are 0-64. Default value is 0, which means that messages are wrapped in the main message log files. Take effect only if –iqmsgsz or IQMsgMaxSize server property is a non-zero value.

- -iqmsgsz <size>

- Limits the message log size. Allowed values are 0-2047in MB. Default value is 0, which means that there is no limit on message log size. IQMsgMaxSize takes precedence over the value of

- iqmsgsz

IQ Message Log Format

I 12/12 02:08:42. 0000017287

I. 12/12 02:08:42. 0000017287 Exception Thrown from slib/s_packedRec.cxx:1291, Err# 3, tid 9 origtid 9

I. 12/12 02:08:42. 0000017287 O/S Err#: 0, ErrID: 2063 (s_packedRecException); SQLCode: -1006000, SQLState: 'QBA00', Severity: 23

I. 12/12 02:08:42. 0000017287 [20238]: IQ Internal error. Please report this to Sybase IQ support.

I. 12/12 02:08:42. 0000017287 Exception Thrown from slib/s_packedRec.cxx:1291, Err# 3, tid 9 origtid 9

I. 12/12 02:08:42. 0000017287 O/S Err#: 0, ErrID: 2063 (s_packedRecException); SQLCode: -1006000, SQLState: 'QBA00', Severity: 23

I. 12/12 02:08:42. 0000017287 [20238]: IQ Internal error. Please report this to Sybase IQ support.

I. 12/12 02:08:42. 0000017287 Rbck

I. 12/12 02:08:42. 0000017287 PostRbck

I. 12/12 02:08:43. 0000017287 Disconnect: SA connHandle: 197 SA connID: 25 IQ connID: 0000017287
User: user1

IQ Open Database Messages

- I. 12/12 10:31:08. 0000000000 **OpenDatabase Completed** -----> **first Open Database**
- I. 12/12 10:31:08. 0000000000 IQ cmd line srv opts: -iqmc 200 -iqtc 200
- I. 12/12 10:31:08. 0000000000 IQ full cmd line: -n rhvm2_iq1608 -x tcpip{port=9015} -c 128m -gc 20 -gd all -gl all -gm 10 -gp 4096 -iqmc 200 -iqtc 200 iqdemo.db
-ti 4400 -gn 25 -o /work3/rel16_iq_sp08.01/IQ-16_0/logfiles/rhvm2_iq1608.0004.srvlog -hn 5
- I. 12/12 10:31:08. 0000000000 DB: r/w, Main 199MB Buffs=1594, Temp 199MB Buffs=1594, Pgsz=131072/8192blksz/16bpc, BP=1
- I. 12/12 10:31:08. 0000000000 DB: Frmt#: 23F/2T/1P (FF: 03/18/1999)
- I. 12/12 10:31:08. 0000000000 DB: Versn: 16.0.0.807/140617/P/sp08.01/Enterprise Linux64 - x86_64 - 2.6.18-194.el5/64bit/2014-06-17 16:03:33
- I. 12/12 10:31:08. 0000000000 DB: Name: /work3/data1608/iqdemo.db
- I. 12/12 10:31:08. 0000000000 DB: Txn ID Seq: 2102549
- I. 12/12 10:31:08. 0000000000 DB: DBID Blk: 2610
- I. 12/12 10:31:08. 0000000000 DB: IQ Server rhvm2_iq1608, PID 10562, LOGIN sybase

IQ Open Database Messages

I. 12/12 10:31:08. 0000000000 Mem: 428mb/M428

Main Blks: U5944/23%, Buffers: U6/L0

Temporary Blks: U65/2%, Buffers: U4/L0

Main I: L43/P3 O: C3/D3/P0 D:0 C:100.0

Temporary I: L34/P0 O: C4/D4/P0 D:0 C: 0.0

I. 12/12 10:31:08. 0000000000 Collation ISO_BINENG, Case Ignore, Blank Padding On, Comparisons are Conditioned

I. 12/12 10:31:09. 0000000000 RcvyCmpl

I. 12/12 10:31:09. 0000000000 Chk

I. 12/12 10:31:09. 0000000000 ChkDone [NumTxnCP: 0]

I. 12/12 10:31:09. 0000000000 PostChk

I. 12/12 10:31:09. 0000000000 CloseDatabase

IQ Open Database Messages

- I 12/12 10:31:09. 0000000000 Using licenses from: /work3/rel16_iq_sp08/SYSAM-2_0/licenses/kserver.lic:/work3/data1608/*.licl. 12/12 10:31:10. 0000000000 Checked out license for 2 IQ_CORE (2014.0331/31-mar-2014/037A 6F88 CC56 20D8) will expire Tue 01 Apr 2015 12:00:00 AM PDT.
- I. 12/12 10:31:10. 0000000000 WARNING: IQ functionality that requires the IQ_CORE license will be disabled on Tue 01 Apr 2015 12:00:00 AM PDT, unless a suitable IQ_CORE license is obtained before that date.
- I. 12/12 10:31:10. 0000000000 This product is licensed to: Sybase, Inc.-for internal use only.
- I. 12/12 10:31:10. 0000000000 **OpenDatabase Completed** -----**Second open database**
- ...
- I. 12/12 10:31:10. 0000000001 1143425344 **Bypassing RLV recovery because an RLV dbspace could not be found.-->New message in IQ 16**
- ❖ IQ Database always starts from last completed checkpoint.
 - ❖ IQ Database is opened twice.
 - First open replays transaction log and checkpoints.
 - Second open is from clean checkpoint.

Close Database Messages

❖ Normal Shutdown(stop_iq)

.iqmsg

- I. 12/12 11:21:06. 0000001523 PostCmt 0
- I. 12/12 11:21:07. 0000001527 Chk
- I. 12/12 11:21:07. 0000001527 ChkDone [NumTxnCP: 0]
- I. 12/12 11:21:07. 0000001527 PostChk
- I. 12/12 11:21:07. 0000000000 **CloseDatabase**

.Srvlog

- I. 12/12 11:21:07. Database server shutdown due to HUP signal
- I. 12/12 11:21:07. Starting checkpoint of "iqdemo" (iqdemo.db) at Mon Dec 12 2014 11:21
- I. 12/12 11:21:07. Finished checkpoint of "iqdemo" (iqdemo.db) at Mon Dec 12 2014 11:21
- I. 12/12 11:21:07. TCPIP listener on IP address 0.0.0.0:9015 is exiting
- I. 12/12 11:21:07. TCPIP listener on IP address (::):9015 is exiting
- I. 12/12 11:21:08. **Database server stopped** at Mon Dec 12 2014 11:21

Close Database Messages

❖ Normal Shutdown(dbstop)

.iqmsg:

- I. 12/12 11:34:10. 0000000285 Chk
- I. 12/12 11:34:10. 0000000285 ChkDone [NumTxnCP: 0]
- I. 12/12 11:34:10. 0000000285 PostChk
- I. 12/12 11:34:10. 0000000000 **CloseDatabase**

.srvlog

- I. 12/12 11:34:10. <,1,**CONTROL_STOP_ENGINE**
- I. 12/12 11:34:10. Database server shutdown requested by DBSTOP
- I. 12/12 11:34:10. Starting checkpoint of "iqdemo" (iqdemo.db) at Mon Dec 12 2014 11:34
- I. 12/12 11:34:10. TCPIP listener on IP address (::):9015 is exiting
- I. 12/12 11:34:10. TCPIP listener on IP address 0.0.0.0:9015 is exiting
- I. 12/12 11:34:10. Finished checkpoint of "iqdemo" (iqdemo.db) at Mon Dec 12 2014 11:34
- I. 12/12 11:34:11. **Database server stopped at Mon Dec 12 2014 11:34**

Close Database Messages

Abort:

- ❖ There will be no Close Database message in .iqmsg
- ❖ IQ Server process will be terminated with a stack trace (Most of the time)
- ❖ By default core file will be truncated to 32k, unless IQ server was started after setting environment variable IQ_UNLIMIT_CORE to 'YES' and core limit have been set to unlimited on OS as well.
- ❖ In some situations you will also have catalog cache dump with .dmp extension

IQ Server Log

- ❖ All database server messages are logged in .srvlog, such as:
 - Catalog Store Messages
 - Connectivity information
 - Output from different APIs
 - Open Client
 - ODBC
 - JDBC
- ❖ On Windows IQ server log default location is C:\Documents and Settings\All Users\Sybase IQ\logfiles
- ❖ On Unix/Linux, IQ server log default location is \$IQDIR16/logfiles
- ❖ On Unix/Linux default name of server log is <servername>.000n.srvlog
- ❖ On Windows default name of server log is <servername>.00n.srvlog

IQ Server Log

IQ Server Log Switches;

- ❖ **o:** Default location of IQ server log can be changed with **–o** startup switch. It is recommended that file name shouldn't be ended with **.log**, as it can create problems for utilities that perform using IQ database transaction log. To find IQ Server Log name , execute select property ('ConSoleLogFile') ;

```
(DBA)> select property ('ConSoleLogFile')  
property('ConSoleLogFile')
```

/work/rel154_iq_esd2/IQ-15_4/logfiles/hpmmem1542c.0113.srvlog

- ❖ **-on {size [k |m | g]}:** Default 10KB. When IQ database server log reaches the specified size, the IQ server renames the file with extension **.old** and starts the new file with the original name. If **.old** IQ database server log file already exists, it is overwritten, in that case **-os** option should be used. **–os** option cannot be used with **–on**.

IQ Server Log

- ❖ **-os {size [k |m | g]}** Default size is 10KB.
 - It specifies the maximum size for the IQ database server log file.
 - Before the IQ server redirects output messages to the IQ server log file, it checks the current file size.
 - If logging message will make the IQ server log exceeds the size specified with **-os** flag, the IQ server will rename IQ Server log file to **yymmddxx.slg**. **-os** switch cannot be used with **-on** switch.
- ❖ **ot <logfile>**: This startup switch truncates the IQ server log file and append output messages to it.
- ❖ To find name of the IQ server log file name:
select property ('ConsoleLogfile')

IQ Server Log

❖ cat redhead_uni16.0009.srvlog

I 12/12 12:47:15. Sybase IQ

I. 12/12 12:47:15. Version 16.0

I. 12/12 12:47:15. (64bit mode)

I. 12/12 12:47:15. Copyright 1992-2013 by SAP AG or an SAP affiliate company. All rights reserved

I. 12/12 12:47:15. Copyright (c) 2013 SAP AG or an SAP affiliate company.

I. 12/12 12:47:15. All rights reserved.

I. 12/12 12:47:15. Use of this software is governed by the Sybase License Agreement.

I. 12/12 12:47:15. Refer to <http://www.sybase.com/softwarelicenses>.

I. 12/12 12:47:15.

I. 12/12 12:47:15. Processors detected: 4 (containing 80 logical processors)

I.12/12 12:47:15. Maximum number of processors the server will use: 4 physicalprocessor(s), 40 core(s)

....

I. 09/13 11:57:59. Starting checkpoint of "unbdb" (unbdb.db) at Fri Dec 12 2014 11:57

I. 09/13 11:57:59. Finished checkpoint of "unbdb" (unbdb.db) at Fri Dec 12 2014

IQ Server Log

- ❖ IQ Server startup parameters and license checkout messages are logged into IQ Server Errorlog
- ❖ Default location on Unix and Linux : \$IQDIR16/logfiles directory
- ❖ Default location on window: C:\Documents and Settings\All Users\Sybase IQ\logfiles directory.
- ❖ On window IQ server Errorlog logfile: iq_server_nt.log
- ❖ On Unix and Linux logfile have 4 digit numbers: <servername>.000n.stderr

IQ Server Errorlog

Starting server redhead_uni16 on redhead at port 6166 (12/12 12:47:08)

Run Directory : /work/data16uni

Server Executable : /work/rel16_iq_sp08.01/IQ-16_0/bin64/iqsrv16

Server Output Log : /work/rel16_iq_sp08.01/IQ-16_0/logfiles/redhead_uni16.0009.srvlog

Server Version : 16.0.0.807/sp08.01 16.0.0/Linux 2.6.18-194.el5

Open Client Version : N/A

User Parameters : '@params.cfg' '-n' 'redhead_uni16' '-x' 'tcpip(port=6166)' '/wprl/data16uni/unidb.db'

Default Parameters : -gp 4096 -ti 4400 -gn 105

...

...

...

IQ Server Errorlog

IQ server starting with:

100 connections (-gm)

170 cmd resources (-iqgovern)

3899 threads (-iqmt)

512 Kb thread stack size (-iqtss)

1996288 Kb thread memory size (-iqmt * -iqtss)

80 IQ number of cpus (-iqnumbercpus)

0 MB maximum size of IQMSG file (-iqmsgsz)

0 copies of IQMSG file archives (-iqmsgnum)

=====

New process id is 71628

SQL Log

- ❖ All SQL Requests are logged into this file.
- ❖ By default SQL Requests are logged into IQ standard server log.
- ❖ IQ DataServer startup options:
 - zr: enables request level logging (NONE, ALL, SQL).
 - zo: Request logging information is logged to a file separate from the default IQ standard server log.
 - zs {size[k | m | g]} : Default 0, which means no max file size limit. When max size is reached, SQL Log is renamed with the extension .old, replacing existing SQL Log with the same name
 - zn : specifies the number of request log files to keep.

SQL Log Format

❖ Example:

=,>,1

=,<,1,PREPARE,create table "DBA"."t1"("a" char null, "b" integer null,)

=,>,1,PREPARE,262163

+1,<,1,EXEC,262163

+439,>,1,EXEC

System Procedures For Analyzing Request Log

- ❖ **sa_get_request_profile:** This system procedure analyzes the request log to determine the execution of similar statements and summarizes the results in the global temporary table `satmp_request_profile`
- ❖ **Example:**
`call sa_get_request_profile('/work3/data1608/sqllog');`
`select * from satmp_request_profile`
- ❖ **sa_get_request_times:** This system procedure analyzes the request log to determine statement execution times and populates the global temporary table `satmp_request_time` with the statements from the log and their execution times
- ❖ **Example:**
`call sa_get_request_times('/work3/data1608/sqllog');`
`select * from satmp_request_time ;`

Bufman Errors

Bufman Errors

- ❖ IQ Server detects Bufman error while reading database pages.
- ❖ Example:
 - I. 12/12 03:25:49. 0000005847 There was an s_buf::Read() error during s_bufman::Find().
 - I. 12/12 03:25:49. 0000005847 Exception Thrown from s_blockmap.cxx:4511, Err# 0, tid 2218 origtid 2218
 - I. 12/12 03:25:49. 0000005847 O/S Err#: 0, ErrID: 2100 (s_corruptdiskblkhdrexception); SQLCode: -1009039, SQLState: 'QSA39', Severity: 14
 - I. 12/12 03:25:49. 0000005847 [20060]: main Bufman: An error was detected on a database page. You may have a damaged index. For additional information, please check your IQ message file or run sp_iqcheckdb. -- (s_blockmap.cxx 4511)
 - I. 12/12 03:25:49. 0000000000 **main Bufman: Incorrect page header read;**
buffer={btype=12,nlb=16,blk=109,pbn=1104824,npb=4,uid=112702,txn=45001107,sp=2,flgs=0x8418
page={btype=12,nlb=16,npb=1,ctr=131,cta=131,pbn=
I. 12/12 03:25:49. 0000000000 1104824,txn=44400239,sp=106,ver=1,pb=102,pus=7}.
s_trycatch.cxx:1951

Bufman Errors

- ❖ IQ server will also print Stack trace in the .iqmsg (IQ Message File).
- ❖ Bufman type errors can be from Main buffer source or Temp Buffer source
- ❖ Both Main and Temp stores have their own buffer manager.
- ❖ s_buf errors have two components
 - Buffer
 - Page
- ❖ Buffer component contains the information, what the Read() was expecting.
- ❖ Page component contains the information, what was actually READ from the disk.

DBCC Errors

DBCC Errors

- ❖ Execute `sp_iqcheckdb` in “check” if possible in “verify” mode after setting database option “**dbcc_log_progress**” to ‘on’.
- ❖ `sp_iqcheckdb` might report errors, which can be corrected or which cannot be corrected.
- ❖ Errors which cannot be corrected:
 - Inconsistent FP indexes (FP lookup table inconsistencies – 1-byte or 2-byte FP is internally inconsistent.
 - VDO errors: Entire table is corrupted, Force-drop the inconsistent table
 - VDO Incorrect First Available Fields
 - VDO Incorrect Next Available Fields
 - VDO Incorrect Used Count Fields
 - VDO Incorrect In-use Bitvec
 - VDO Incorrect In-use Bitmap
 - VDO Incorrect Partial Bitmap
 - VDO Incorrect Deleted Bitmaps

Dbcc Errors

Unrepairable Errors:

- ❖ **Blocks with multiple owner:** - Blocks in use by more than one database object.

Drop inconsistent object

DBCC - 1287 blocks with multiple owners

- ❖ **DBCC Meta-Data Errors** – inconsistent internal page mapping structure, drop and recreate object
 - Blockmap invalid chunk size error count.
 - Blockmap compression bit error count.
 - Blockmap invalid block number error count.

***** Blockmap Invalid Block Number Error Count', '53', '*****'**

- ❖ **inconsistent disk space block header** – inconsistent storage for the object. Drop and recreate the object.
- ❖ **dbcc decompress errors**

main Bufman: Decompression error during Read;

buffer={btype=13,nlb=16,blk=325829,pbn=1242378010,npb=3,uid=14935,txn=790126,sp=0,flgs=0x8018

page={btype=13,nlb=16,npb=3,ctr=136,cta=134,pbn=1242378010,txn=248695,sp=19,ver=1,pb=87,pus=63685}. s_buf.cxx:1333

Index Errors

- ❖ **FP Lookup Table inconsistency** : 1-byte or 2-byte FPs gets internally inconsistent . This type of error is unrepairable

I. 12/14 06:54:53. 0000000967 Index Statistics:

I. 12/14 06:54:53. 0000000967 ** **Inconsistent Index: DBA.r56.ASIQ_IDX_T5383_C1_FP** *****

I. 12/14 06:54:53. 0000000967 ** **Inconsistent Index: DBA.r56.c1hg** *****

I. 12/14 06:54:53. 0000000967 Verified Index: DBA.r56.ASIQ_IDX_T5383_C2_FP

I. 12/14 06:54:53. 0000000967 ** Non-Completed Index : DBA.r56.ASIQ_IDX_T5383_C1_FP *****

I. 12/14 06:54:53. 0000000967 ** Non-Completed Index : DBA.r56.c1hg *****

I. 12/14 06:54:53. 0000000967 FP Indexes Checked: 2

I. 12/14 06:54:53. 0000000967 ** FP Lookup Table Inconsistencies: 1 *****

I. 12/14 06:54:51. 0000000967 **DBCC: ERROR - Byte FP lookup table inconsistency, lookupCount[0]=38772 <> barrayCount=59016.**

Fixing Index Errors

- ❖ In some cases , you will be able to select from inconsistent FP by restricting where clause and then extracting data from that column or by adding new column with alter table command and then updating new column with old corrupt column using selective selects. Drop bad_col , rename new column to old column name
- ❖ If it is not possible, then you will need to drop the column, in some cases using force_drop. For Force_drop, it is advisable to contact SAP Product Support for assistance
- ❖ Most of the time, corrupted non-FP indexes can be fixed using sp_iqrebuildindex or drop and create non-FP index.
- ❖ Depending on type of inconsistency, 'drop index', 'alter table drop column', 'drop table' or 'force_drop' can be used to fix the inconsistency

Force_Drop

Steps for force_drop should be used with caution:

- ❖ When force dropping objects, you must ensure that only the DBA is connected to the database. Restart the server immediately after a force drop.
- ❖ The following procedure uses the -gd and -gm switches to restrict database access. The -gd switch only limits users who can start or stop databases on a running server. For a more restrictive method, start the server in forced recovery mode.
- ❖ Restart the server.

```
start_iq -n bad_db_server -x 'tcpip{port=7934}' -gm 1 -gd dba bad_db.db
```

- ❖ You must not allow other users to connect when force dropping objects.
- ❖ SAP recommends using two server startup switches to restrict access:
 - Use -gd DBA so that only users with DBA authority can start and stop databases. (Note that the client must already have a connection to the server to start or stop the database, so this switch does not prevent connections.)
 - Use -gm 1 to allow a single connection plus one DBA connection above the limit so that a DBA can connect and drop others in an emergency.

Force_Drop

- ❖ Set the temporary option **FORCE_DROP** to ON.
- ❖ Drop all inconsistent objects.
- ❖ Use the commands **DROP INDEX**, **ALTER TABLE DROP COLUMN**, or **DROP TABLE** as needed. Do not enter any other DDL or DML commands until after restarting the server.
- ❖ Restart the server.
- ❖ To recover the leaked space and update the allocation map to the correct state, start the server.
`start_iq -n bad_db_server -x 'tcpip{port=7934}' -gm 1 -gd dba bad_db.db`
- ❖ Run **sp_iqcheckdb**(This step resets the database allocation map to the calculated allocation map)
`sp_iqcheckdb 'dropleaks database';`
- ❖ If **dbcc** returns no errors, then issue
checkpoint (twice)
- ❖ Shutdown and Restart IQ Server normally



IQ RLV Troubleshooting

IQ RLV Troubleshooting

- ❖ **RLV store cannot be merged – table <tab> has too many fragments.**

- I. 12/14 21:36:13. 0000016854 Exception Thrown from rplib/rv_TabStoren.cxx:71, Err# 308, tid 3 origtid 3

- I. 12/14 21:36:13. 0000016854 O/S Err#: 0, ErrID: 4098 (db_sqlexception); SQLCode: -1001104, SQLState: 'QRA04', Severity: 14

- I. 12/14 21:36:13. 0000016854 [22040]: Unable to perform merge, table ReportData has too many fragments.
-- (rplib/rv_TabStoren.cxx 71)

- ❖ **This error means there is long running transaction which is keeping too many old fragments of the table in the memory. Table cannot be merged until long running transaction completes. One will need to check with sp_iqtransaction and then drop that connection**

IQ RLV Troubleshooting

- ❖ **RLV Store ran out of space.**

Exception Thrown from slib/s_trycatch.cxx:1420, Err# 0, tid 2 origtid 2

O/S Err#: 0, ErrID: 2096 (s_nodbpaceexception); SQLCode: -1009170, SQLState: 'QSB66', Severity: 14

[20223]: You have run out of space in IQ_RLV DBSpace.

-- (slib/s_trycatch.cxx 1420)

Rbck

100864487168 Current active row count for table RLVTab 2640601

PostRbck

Allocation failed. Dbspace IQ_RLV is OUT OF SPACE

- ❖ **Fix will be to add space to RLV Store**

IQ RLV Troubleshooting

- ❖ Cannot make RLV dbspace Read-only

The RLV dbspace IQ_RLV cannot be made read only because it contains RLV enabled tables

- ❖ RLV dbspace cannot be made READONLY, as long as there are RLV-enabled tables. RLV-enabled tables should be either dropped or disabled, before making RLV dbspace READ-ONLY
- ❖ To Disable RLV :
Alter table <tablename> disable RLV store
- ❖ Alter dbspace to READONLY
alter dbspace IQ_RLV READONLY

IQ RLV Troubleshooting

- ❖ Merge needed before table-level modification
SQL Error -1001105: Table RLVTAB requires an RLV store merge before table-level modification.
- ❖ This error means that table level transaction is modifying an RLV table with BLOCKING set to 'off'
- ❖ To fix this issue, set BLOCKING to 'ON' or do manual merge before performing table-level modification
`sp_iqmergerlvstore 'NON-BLOCKING', 'RLVTab', 'DBA';`

IQ RLV Troubleshooting

❖ RLV Store Out of Memory

Exception Thrown from rplib/rv_TabStoren.cxx:178, Err# 5, tid 3227 origtid 3227

O/S Err#: 0, ErrID: 15008 (rv_PersistenceException); SQLCode: -1013129, SQLState: 'QRLV1', Severity: 14

[22159]: RLV Store has run out of memory

Exception Thrown from rplib/rv_TabStoren.cxx:178, Err# 5, tid 3 origtid 3227

O/S Err#: 0, ErrID: 15008 (rv_PersistenceException); SQLCode: -1013129, SQLState: 'QRLV1', Severity: 14

IQ RLV Troubleshooting

❖ Explanation

To much data is being modified/ added by RLV transactions at once

❖ To Fix

Reduce the size of update/inserts and increase frequency of commits to keep amount of data modified/transaction.

❖ Explanation

Automated merge is not able to keep up with RLV workload

❖ To Fix

- Review RLV merger history table(SYSIQRLVMERGEHISTORY)**
- Adjust the automated merger period(RV_AUTO_MERGE_EVAL_INTERVAL)**
- Adjust automated thresholds, RV_MERGE_TABLE_NUMROWS, RV_MERGE_TABLE_MEMPERCENT, and RV_MERGE_NODE_MEMSIZE RLV Options.**

IQ RLV Troubleshooting

❖ Explanation

Maximum RLV memory configure too low.

❖ To Fix

increase maximum RLV memory using server startup switch `-iqlvmem`
or

increase maximum RLV memory using `sa_server_option`

`sa_server_option 'rlv_memory_mb', <max MB>`

issue manual merge to free of RLV memory

`sp_iqmergerlvstore (merge_type, table_name, [table_owner])`

IQ RLV Troubleshooting

Forced Recovery: If RLV failed to recover due to any kind of corruption such as OS exceptions/checksum errors on reading page from disk or mismatched sequence number on head/tail of page.

Recovery Steps:

- ❖ Add -gm 1 (allow single user connection plus one DBA connection) and -gd DBA (users with SYSTEM OPERATOR system privilege can start and stop database) in param.cfg
- ❖ Add -iqrvrec_bypass 1 switch to params.cfg to by pass all RLV recovery. This option should only be used in emergency to drop corrupted RLV table or for deleting RLV portion of the table. Currently this option disables further logging, but doesn't prevent general RLV operations. Non-DBA users should n't allowed access to database, when recovering RLV with this switch
- ❖ Run sp_iqcheckdb 'allocation/check/verify database' to check consistency of the database.
- ❖ Delete RLV portion of the table(you will lose all the data in the RLV portion of the table), This might leave inconsistent database, but will allow a subsequent recovery

```
delete * from table where rowid() > 2^48
```
- ❖ Stop and restart SAP Sybase IQ server normally i.e without -gm 1 -gd DBA and -iqrvrec_bypass 1 switches.

Stack Trace

How To Analyze Stack Trace

- ❖ A stack trace is collection of active stack frames at a certain point in time during the execution of program.
- ❖ A Stack is a list of pending calls that the thread is executing with Last in First Out
- ❖ Stack contains most important piece of information helpful in debugging.
- ❖ Each line [frame] gives information about the program counter(pc), the symbol of the routine and hexadecimal data associated with the routine.
- ❖ A stack and it's primitives, push and pop, are built into microprocessors
 - stack has dedicated register which contains the stack pointer, the address of the top of the stack
 - op codes and assembler mnemonics for push and pop.
- ❖ When a function is called, it is arguments and return address are PUSHed onto stack
- ❖ When function finishes, return address and arguments are POPped back
- ❖ If the items pushed onto the stack exceeds the memory allocated for the stack, a stack overflows occurs.

IQ Stack Trace

- ❖ IQ connection can have stack on more than one thread(kernel process).
- ❖ For active task, IQ uses the stack to store certain task-specific information
 - Currently active functions and instructions
 - Parameters to the functions
 - Local Variables of the functions
- ❖ In many error conditions, the IQ process pops its stack and prints it out – tracing back through the stack

Where to find Stack Trace?

- ❖ Stack trace is found in the .iqmsg and the stktc-YYYYMMDD-hhmmss.iq
- ❖ Stack trace log is only generated for fatal errors.
- ❖ Stack trace for Non-Fatal errors is logged in the .iqmsg only.
- ❖ By default IQ truncates the core file to 32k, which is not needed. In some cases, support engineer might ask for a complete CORE file, in that case, IQ server needs to be started after setting environment variable IQ_UNLIMIT_CORE to 'YES' and core limit on OS should be set to unlimited (Refer to particular Operating System manuals on how to set core file size to unlimited on the Host Machine).

Stack Trace: Diagnostic Tool

- ❖ Stack trace provide useful information related to what a process was doing when Non-Fatal/Fatal condition was encountered
- ❖ Stack traces are often preceded by Command/Query, which caused this error condition.
- ❖ In case of Query, partial html query plan is also logged in .iqmsg.
- ❖ Search in SAP xSearch, by picking up good keywords from the stack, may turn up an answer.

Stack Traces and Bugs

- ❖ **Stack trace indicates an error condition that was encountered. Error condition is not necessarily a bug**

Stack trace – Example

**** Error from IQ connection: SA connHandle: 3212 SA connID: 19 IQ connID: 0000091219 User: dba** Time of error: 2014-11-15 14:07:12 ** IQ Version: SAP IQ/16.0.0/140804/P/sp08.03 ** OS info: IQ built on: Enterprise Linux64 - x86_64 - 2.6.18-194.el5, Executed on: Linux/IPROD1/2.6.32-358.el6.x86_64/#1 SMP Tue Jan 29 11:47:41 EST 2013/x86_64 ** Command status when error occurred: NO COMMAND OR CURSOR ACTIVE ** Parser command text:**

begin declare @i integer; set @i = 0; while(@i <= 500) loop insert into "dba"."TABCOUNT" select "Table_name","Tab1","dba"."FN_TabCount"("TABLE_NAME"),"dba"."FN_TABSIZE"("TABLE_NAME") from "dba"."Migtab_Count" where "Tabseq_Id" = @i; -- BETWEEN @i+1 AND @i+50; commit work; set @i = @i+1 -- @i+50; end loop end

===== Thread Number 140245667288832 (IQ connID: 0000091219) =====

pc: 0x7f8bc97b3672 pcstkwalk(stk_trace*, int, db_log*, hos_fd*)+0x32

pc: 0x7f8bc97b3ce1 ucstkgentrace(int, int)+0x121

pc: 0x7f8bc97b4f1e DumpAllThreads(char const*, unsigned int, int)+0x12e

pc: 0x7f8bc95421c0 hos_ABORT(char const*, unsigned int, hos_exception const&, char*, char*)+0x2a0

pc: 0x7f8bc9796771 Throw(hos_exception const&)+0x3e1

pc: 0x7f8bc970d17a hos_assertexception::ThrowException(char const*, int, hos_assertexception::errors)+0x2a

pc: 0x7f8bcb61557c st_txncb::ReleaseSavepoint(st_txnsp*)+0xfdc

pc: 0x7f8bcb6159c3 st_txncb::ReleaseSavepoint(unsigned int)+0xf3

Stack trace - Example

```
pc: 0x7f8bca2f97a9 st_command::CheckErrorInReleaseSavepoint(unsigned int)+0x73
pc: 0x7f8bca239efa st_command::st_command(st_command::st_cmdtype, a_statement const*, int)+0x43a
pc: 0x7f8bc90dda3b df_dmlcommand::df_dmlcommand(a_statement*, UIConnection*, UICursor*,
      df_dmlcommand::window_create_flag, a_cursor_spec*, st_command::st_cmdtype, a_plist_node*)+0x3b
pc: 0x7f8bc90ddfe6 df_CloseCursor::df_CloseCursor(UICursor*)+0x36
pc: 0x7f8bca28636b st_iqddl::CloseCursor(UICursor*)+0x1b
pc: 0x7f8bc907be68 UIQDML_CloseCursor+0x18
pc: 0x7f8bca28d936 st_SAIQInterfaceInfo::callFunction()+0x26
pc: 0x7f8bca2f857b st_SAIQInterface::RunIQFunc(st_SAIQInterfaceInfo*)+0x189
pc: 0x7f8bca28e2aa st_SAIQInterface::Execute(int, void*, unsigned int (*)+0x20a
pc: 0x7f8bc97c5e20 saint_iqthresholdddl::CloseCursor(ICursor*, a_cursor_spec*)+0xd0
pc: 0x7f8dd75dccb4 df_OmniRowScan::Complete()+0x94
pc: 0x7f8dd75b9619 dfo_Scan::DoComplete()+0x29
pc: 0x7f8dd75a1f4e dfo_Base::Complete()+0x3e
```

Stack trace – First Line

**** Error from IQ connection: SA connHandle: 3212 SA connID: 19 IQ connID: 0000091219 User: dba ** Time of error: 2014-11-15 14:07:12 ** IQ Version: SAP IQ/16.0.0/140804/P/sp08.03 ** OS info: IQ built on: Enterprise Linux64 - x86_64 - 2.6.18-194.el5, Executed on: Linux/IPROD1/2.6.32-358.el6.x86_64/#1 SMP Tue Jan 29 11:47:41 EST 2013/x86_64 ** Command status when error occurred: NO COMMAND OR CURSOR ACTIVE ** Parser command text:**

begin declare @i integer; set @i = 0; while(@i <= 500) loop insert into "dba"."TABCOUNT" select "Table_name","Tab1","dba"."FN_TabCount"("TABLE_NAME"),"dba"."FN_TABSIZE"("TABLE_NAME") from "dba"."Migtab_Count" where "Tabseq_Id" = @i; -- BETWEEN @i+1 AND @i+50; commit work; set @i = @i+1 -- @i+50; end loop end

Note:

- ❖ **SA ConnHandle: 3212**
- ❖ **IQ Connection ID: 0000091219**
- ❖ **UserID: dba**
- ❖ **Time of error: 2014-11-15 14:07:12**
- ❖ **IQ Version: SAP IQ/16.0.0/140804/P/sp08.03**
- ❖ **OS info: IQ built on: Enterprise Linux64 - x86_64 - 2.6.18-194.el5**
- ❖ **IQ Host Machine: Linux/IPROD1/2.6.32-358.el6.x86_64/#1 SMP Tue Jan 29 11:47:41 EST 2013/x86_64 ****

Stack trace – First Line

**** Command status when error occurred: NO COMMAND OR CURSOR ACTIVE ** Parser command text:**
begin declare @i integer; set @i = 0; while(@i <= 500) loop insert into "dba"."TABCOUNT" select
"Table_name","Tab1","dba"."FN_TabCount"("TABLE_NAME"),"dba"."FN_TABSIZE"("TABLE_NAME")
from "dba"."Migtab_Count" where "Tabseq_Id" = @i; -- BETWEEN @i+1 AND @i+50; commit work; set
@i = @i+1 -- @i+50; end loop end

Stack trace – Routine after hos_ABORT

pc: 0x7f8bcb61557c st_txncb::ReleaseSavepoint(st_txnsp*))+0xfdc

- ❖ Note routine which was being executed at the time when it got signal
- ❖ IQ server got abort from st_txncb::ReleaseSavepoint()



Support Portal

Knowledge Base & Incidents



Release, Upgrade & Maintenance Info



Remote Support



Software Downloads



Keys, Systems & Installations



Support Programs & Services



Users & Authorizations



Documentation



Search for SAP Notes, Knowledge Base Articles and more

st_txncb::Release Savepoint



Go

Enter Note/KBA Number

FEEDBACK



Download Software



Request a Key



Edit System



Manage Remote
Connections



Report an Incident



View my Incidents

New to Support?

[Start Tutorial](#)

[Getting Started](#)

[Register for a Portal Demo](#)

[Simplify logging in! Install a](#)



Support Portal

Knowledge Base & Incidents ➤

Release, Upgrade & Maintenance Info ➤

Remote Support ➤

Software Downloads ➤

Keys, Systems & Installations ➤

Support Programs & Services ➤

Users & Authorizations ➤

Documentation ➤

🏠 Search result page

SAP Support Knowledge Search

ONE SEARCH, MULTIPLE REPOSITORIES

SAP Note

Repositories

Support Content

SAP Notes (1)

Narrow Down Your search results

Application Area (1)

Search for

Languages: ☒ English ☐ German ☐ Japanese

 Show advanced search options

SAP Notes

Displaying 1-10 of 1 results

[Hide abstracts](#)

Sort by: ▼

Search in Support Academy

FEEDBACK

SAP Knowledge Base Article



Statistic⁰



Printer-Friendly Version



PDF Version



Add to favorites



Subscribe



Quick link



Can't see edit button?



2111037 - SAP IQ 16 server crashes showing ReleaseSavePoint operation in stack, involves

Overall Rating ★★★★★ 0 Votes

local temp table

Version 1 ▾

Validity: 18.12.2014 - active

Language English

Show Change Log

Content: [Summary](#) | [Header Data](#) | [Product](#)

Symptom

Generated stack should show following functions for a thread causing crash

- SAP IQ Server crash at 'st_txnrb::ReleaseSavepoint'
- 'st_command::CheckErrorInReleaseSavepoint'
- 'st_command::st_command'
- 'df_dmlcommand::df_dmlcommand'
- 'df_CloseCursor::df_CloseCursor'

Environment

- Pre SAP IQ 16 sp08.21

Cause



Resolution

- ❖ **CR 774469: IQ Server crash during a transaction which involves creation of a local temporary table as a part of transaction fixed in IQ 16 sp08.21 and above.**



Hung Server

Hung Server

- ❖ Collect at least 3 pstack samples 3-5 minutes apart
- ❖ If Secondary node is hanging, collect pstack samples for both Coordinator and Secondary Nodes
- ❖ Collect .iqmsg, .srvlog and -zr log
- ❖ Collect iostat (# iostat -t -d -x 5 20) and vmstat samples

Hung Server - Example

- ❖ Deadlocks between threads, during data load, pstack shows lot of versioning blockmap Reallocate , allocation and deallocation Checkpoint waiting for txncb lock

Threads (1): 514 - Waiting on main Freelist lock

fffffd7ffe1bdd08 mutex_lock_impl

fffffd7ffe1bddfb mutex_lock

fffffd7fd39acaf5 unsigned long long s_dbmvbio::Allocate

fffffd7fd35ebc27 unsigned long long s_VersioningBlockmap::Reallocate

fffffd7fd36d6c46 void s_buf::DirtyInternal

fffffd7fd36d5ab0 void s_buf::DirtyBlockmapPage

fffffd7fd35f0f2a s_buf*s_blockmap::VersionSBuf

fffffd7fd35ea3f2 void s_VersioningBlockmap::Deallocate

fffffd7fd36d7010 void s_buf::DirtyInternal

fffffd7fd36d51d5 void s_buf::Dirty

Hung Server - Example

fffffd7fd3930a81 void s_pageListCursor::Dirty
fffffd7fd3b15d52 void v2s_bmcontext::bmlfblkdirty
fffffd7fd3b41d2d int v2s_bmcontext::bmlfselfunion
fffffd7fd3bd5798 void v2s_bmcontext::bmselfsetrng
fffffd7fd3617c40 void s_bm::SetBits
fffffd7fd364d695 void s_bmTSCursor::Flush
fffffd7fd364d2c3 int s_bmTSCursor::TestAndSetBitIfOff
fffffd7fd3804040 int hs_dplInsert::InsertLocal
fffffd7fd3803d8e int hs_dplInsert::Insert
fffffd7fd2addf49 void hdb_hinsert::InsertBulkByIndex
fffffd7fd2acf9d0 int hdb_insertIter::LocalExecWork
fffffd7fd2ad0eaa int hdb_insertIter::ExecWork
fffffd7fd33e70f6 void workAllocator::DoWork

Hung Server - Example

Threads (1): 743 - Holding some txncb locks, waiting on another txncb lock

fffffd7ffe1bdd08 mutex_lock_impl

fffffd7ffe1bddfb mutex_lock

fffffd7fd3afee99 void st_txncb::CheckpointLock

fffffd7fd3aed55d unsigned st_txnMgr::Checkpoint

fffffd7fd2c98401 void db_Checkpoint::Execute

fffffd7fd3a52482 void st_command::DoCmdThroughResourceGate

fffffd7fd3abc583 unsigned st_iqtxn::Checkpoint

fffffd7fd34478f2 UIQTxn_Checkpoint

fffffd7fd3ac01b2 void st_thrInterfaceInfo::callFunction

fffffd7fd3ac4925 void st_threadInterface::IQwaitForFunc

fffffd7fd3ac3262 IQLeaderThread

fffffd7fd3407856 int hos_thread::Main

Hung Server – Example

Threads (1): 607 - has main Freelist lock, waiting on busy freelist buffer.

fffffd7fd308004b void hos_condvar::Wait

fffffd7fd36d48d9 void s_buf::WaitForSignal_DontCareIfBusy

fffffd7fd36ea298 s_buf*s_bufhash::Find

fffffd7fd36fea79 s_buf*s_bufpartition::FindLogicalBuffer

fffffd7fd35cbb0b s_buf*s_blockmap::Find

fffffd7fd392b06b void*s_pageListCursor::PositionPrefetchedPage

fffffd7fd3b18f88 void v2s_bmcontext::bmlfblkget

fffffd7fd3b9ae46 void v2s_bmcontext::bmselfreldif

fffffd7fd362928f void s_bm::TurnOff

fffffd7fd378c12f void s_dbmvbio::AdjustFreeList

fffffd7fd38920e8 void s_iqstore::AdjustFreeList

fffffd7fd3ab6b05 int st_globalFIMgr::RemoveWriterFreelist

Hung Server - Example

fffffd7fd2b9efe4 void db_processMpxAck::Execute

fffffd7fd3a52482 void st_command::DoCmdThroughResourceGate

fffffd7fd3abab72 unsigned st_iqctl::ProcessMpxAck

fffffd7fd3445ed1 UIQCtl_ProcessMpxAck

fffffd7fd3ac01b2 void st_thrInterfaceInfo::callFunction

fffffd7fd3ac4925 void st_threadInterface::IQwaitForFunc

fffffd7fd3ac3262 IQLeaderThread

fffffd7fd3407856 int hos_thread::Main

fffffd7fd32ddc0d int hos_lwtask::Start

fffffd7ffe1c504b _thr_setup



Troubleshooting Multiplex

Troubleshooting Multiplex

Version Buildup:

- ❖ IQ snapshot versioning creates multiple versions of IQ tables/objects on every DML
- ❖ Multiple versions occupy space, causing IQ main store to run low on space and in some cases Out of Space
- ❖ New transactions can only see latest versions
- ❖ Database pages that belong to older versions are garbage collected and put back into free list to reclaim space
- ❖ Only Coordinator can drop versions. It cannot drop any version if it is still in used on any secondary node in the multiplex
- ❖ IQ assumes that the versions are in use till the Secondary node informs Coordinator about the versions that are no longer in use
- ❖ It is best practice to commit current transactions periodically, and allow the Writer to drop old table version to free disk space. Specifying the `auto_commit` option helps minimize space due to minimize version buildup.

Version Buildup

- ❖ If IQ main store is running low on space due to version buildup, check space used by older versions by executing `sp_iqstatus` and `sp_iqfile`

`sp_iqstatus`

Other Versions:

2 = 0Mb

Active Txn Versions:

0 = C:0Mb/D:0Mb

...

...

...

Main Tivlog Size:

Pages: 1, Recs: 8, Replays: 0/0

Version Buildup

❖ sp_iqfile

Note information in Usage and BlkTypes in sp_iqfile output:

(DBA)> sp_iqfile

DBSpaceName	DBFileName	Path	SegmentType	RWMode	Online	Usage
DBFileSize	Reserve	StripeSize	BlkTypes			
FirstBlk	LastBlk	OkToDrop				

IQ_SYSTEM_MAIN	IQ_SYSTEM_MAIN	/data1541/iqdb.iq	MAIN	RW	T	58 9.76G 0B
1K	1H,255968F,32D,481277A,16X,128M,222B,32C					
1	1280000	N				

Version Buildup

Identifier	Block Type	Identifier	Block Type
A	Active Versions	R	RLV Free List Manager
B	Backup Structures	T	Table Use
C	Checkpoint Log	U	Index Use
D	Database Identity	N	Column Use
F	Free List	X	Drop at Checkpoint
G	Global Free list Manager		
H	Header Blocks of the Free List		
I	Index Advice Storage		
M	Multiplex CM		
O	Old Version		

Version Buildup

- ❖ Run `sp_iqtransaction` to make sure there are no long running transactions. This is applicable to Simplex as well
- ❖ Execute `sp_iqmpxinfo` and monitor if all included Secondary Servers are running. Coordinator must reserve versions indefinitely if a Secondary Server is not running to report its version data.
- ❖ Check to make sure event `ev_iqmpxstatus` is enabled and running.
- ❖ Run `sp_iqversionuse()` on the Coordinator to determine who is holding on to older version on which Secondary Node in the Multiplex
- ❖ Use 'Alter multiplex Server' command to exclude Secondary Server that is down for an extended period of time.

`alter multiplex server iqmpxq1 role 'excluded`

- Database Option `MPX_AUTOEXCLUDE_TIMEOUT` specifies timeout in minutes for autoexcluding a Secondary Server on the Coordinator. Default: 60 min and max 10080 min (1 week) and 0 indicates that Secondary Server should be never excluded.

TLV Log

sp_iqmpxdumptlvlog

- ❖ This stored procedure can be used to examine contents of the table version log

- ❖ Syntax:

sp_iqmpxdumptlvlog [main] , [asc | desc]

Truncating The Transaction Log

- ❖ **Shutdown the IQ Server**
- ❖ **Start the server with `-m` startup switch**
- ❖ **Stop and restart server without `-m` startup switch**

Collisions – Handling Dynamic Collisions

Dynamic Collisions:

- ❖ Dynamic collisions occurs, when DDL change is committed on IQ objects(table,procedure,function or view while that Object is in use i.e same object is being queries by Secondary Node
- ❖ The Dynamic collision results from TLV replay of shared IQ object schema updates (alter/drop)
- ❖ Dynamic collision occurs only on Secondary Servers.

Handling Dynamic Collisions:

- ❖ Dynamic collisions are handled automatically as part of TLV replay
- ❖ The user connection using the object on the secondary node that causes the dynamic collision is forcefully disconnected and Message is logged in the .iqmsg

Collisions – Handling Dynamic Collisions

Shared IQ Store update DDL statement

drop table DBA.t1

Disposition: **SQLSTATE:42W21** --dropped 1 connection(s) for table

DBA.t1 Retry successful

- ❖ Dynamic collisions can be avoided by scheduling schema changes during low peak activity

When Synchronization is Necessary

- ❖ Some of the events can cause Secondary Node to automatically shut down. In some cases, Secondary Server will need to be Synchronized before it can be started.
 - The Secondary Server status has been changed from Excluded to INCLUDED.
 - A dbfile has been added to the IQ_SYSTEM_MAIN dbspace
 - A Coordinator failover has occurred and this Secondary Server is the former Coordinator Server.
 - The Coordinator server has been started in single node mode, then restarted in normal mode.

Using EXCLUDED/INCLUDED Server Status

- ❖ Alter multiplex server <servername> , 'EXCLUDED |INCLUDED' command can be used to change Secondary Server status.
- ❖ When Secondary Server Status is changed to EXCLUDED.
 - Automatically shutdown the Secondary Server
 - Removes the Secondary Node from consideration in Version Cleanup
- ❖ Restrictions:
 - The Coordinator Server cannot be EXCLUDED
 - The designated failover Secondary Server cannot be excluded unless it is the last non-excluded Secondary Server in the multiplex.

Using EXCLUDED/INCLUDED Server Status

- ❖ Before altering Secondary Server Status to EXCLUDED , it must be shutdown first.
- ❖ Use Database option MPX_AUTOEXCLUDE_TIMEOUT to set auto timeout for excluding secondary node.
- ❖ If the designated failover node is excluded, the failover will not work till it is included and synchronized.

Secondary Server Emergency Shutdown

❖ Expected:

When first Secondary Server is added to Multiplex
Configuration Change

TLV replay on an offline Dbspace.

When dbfile is added to a user Dbspace, which is not accessible to Secondary servers

Table is created in offlined dbspace

TLV replay will fail on the Secondary Node and Server will be shutdown

❖ Error:

I. 09/18 10:38:33. 0000000014 [21051]: Secondary server cannot maintain catalog sync.--
(dblib/db_iqutility.cxx 4203)

I. 09/18 10:38:33. 0000000014 sp_mpxprocesstlvlog exception, SQLSTATE=QCB11 [Secondary server
cannot maintain catalog sync.

-- (dblib/db_iqutility.cxx 4203)]

Performance Issues

Performance Issues

- ❖ Database profiling IQ system stored procedures
- ❖ Event profiling SA system stored procedures
- ❖ Key performance indicators(IQ Cockpit)
- ❖ Isolating performance issues
- ❖ Common performance problems
- ❖ Diagnostics to collect
- ❖ Common causes for poor query performance

Database Profiling IQ System Stored Procedures

- ❖ **System Stored Procedures that will return database usage statistics.**

Name	Description
sp_iqconnection	<p>This system procedure provides information about connections and versions, including which users using temporary dbspace and which users are holding versions alive, what the connections are doing inside IQ, connection status , database version status and so on.</p> <p>sp_iqconnection [connhandle]</p> <p>sp_iqconnection</p>
sp_iqcontext	<p>This System procedure tracks and displays, by connection, information about statements that are currently executing</p> <p>sp_iqcontext [<connhandle>]</p> <p>sp_iqcontext</p>

Database Profiling System Stored Procedures

Name	Description
<code>sp_iqcheckdb</code>	<p>This system procedure checks validity of the current database. Optionally corrects dbspace or database allocation problems.</p> <p><code>sp_iqcheckdb 'mode target [...] resources resource-percent]</code></p> <p><u>sp_iqcheckdb</u></p>
<code>sp_iqdbstatistics</code>	<p>This system procedure reports results of the most recent <code>sp_iqcheckdb</code>.</p> <p><code>sp_iqdbstatistics</code></p> <p><u>sp_iqdbstatistics</u></p>
<code>sp_iqtablesize</code>	<p>This system procedure provides information about the number of blocks used by each objects in the current database and the dbspace name in which the object is located.</p> <p><code>sp_iqtablesize (table-owner.table-name)</code></p> <p><u>sp_iqtablesize</u></p>

Database Profiling System Stored Procedures

Name	Description
sp_iqdbsize	This system procedure provides the size of the current database. sp_iqdbsize ([main]) sp_iqdbsize
sp_iqdbspaceinfo	This system procedure provides the space usage by each object in the current database. sp_iqdbspaceinfo ['main [table table-name index index-name] [...]] sp_iqdbspaceinfo
sp_iqstatus	This system procedure provides variety of IQ status information about the current database.. sp_iqstatus sp_iqstatus

Event Profiling SA System Procedures

- ❖ Event profiling SA System Procedures return performance statistics for events, functions, stored procedures and triggers.

Name	Description
sa_server_option	This system procedure sets database profiling options. call sa_server_option('procedureprofiling','on') sa_server_option
sa_procedure_profile	This system procedure returns execution times for each line with in procedures, functions, events that have been executed in a database. call sa_procedure_profile [filename, save_to_file]] sa_procedure_profile
sa_procedure_profile_summary	This system procedure summarizes execution times for all events, functions and procedures. call sa_procedure_profile_summary [filename [,save_to_file]] sa_procedure_profile_summary

Key Performance Indicators(SAP IQ Cockpit)

- ❖ **Monitoring options are found under Monitor tab**
- ❖ **Availability and performance of simplex or multiplex environment can be monitored**
- ❖ **Monitoring data in the Monitor tab is displayed by monitoring engine with the exception of statistical Chart is real time, reflective of current session.**
- ❖ **Setting option in the Monitor tab can be used to refreshing interval.**



MONITOR

EXPLORE

ALERT

Overview

All Statistics

Engine

Connections

Transactions

Dbspaces

Backups

Store I/O

Caches

Table Versions

Operations & Requests

Network

Table Pages

Table Locks

Server

Activities

Caches

Version Usage

Details

State: Running

Host: JYOL00547447A

Port: 9714

Server name: vicky_iqdemo0827

Database: iqdemo

Server type: Single Server

Server version: 16.0.0.808

Platform: Linux 2.6.32-431.1.2.el6.x86_64 #1 SMP Sun Nov 24 09:37:37 EST 2013

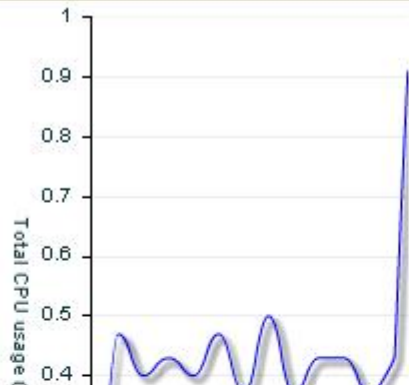
CPU History



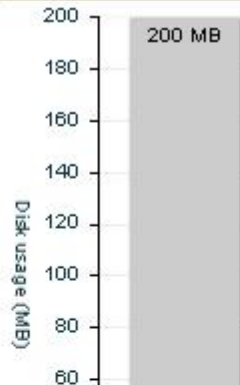
IQ Memory



Disk Usage



Total: 359 MB



Isolating Performance Issues

- ❖ **Is performance problem due to OS/HW**
 - Monitor OS, HW and storage for any bottlenecks or Issues
 - Check for high CPU use, high CPU system time, low CPU user time, high wait time.
 - Check for IO average service times that are above 10ms
- ❖ **Performance problems within SAP Sybase IQ**
 - Look for missing indexes by enabling index advisor
 - Enable query html plan if issue is only with isolated query.

Operating System Diagnostics Tools

Unix

- ❖ **top, topas:** provides runtime look of processor activity in real time. top available on solaris, Linux and HP-UX and topas is available on AIX
- ❖ **ps:** reports process status
- ❖ **vmstat:** provides information about system processes, memory, paging, block Q, traps and CPU activity.
- ❖ **iostat:** provides disk subsystem information
 - Linux: `iostat -t -d -x 5 20`
 - AIX: `iostat -T -D 5 20`
- ❖ **sar:** writes selected OS activity results to standard output

Windows:

- ❖ **Task Manager, Resource Manager:** provides detailed information about windows performance and running applications, CPU usage and other system services.

Common Performance Problems

- ❖ **Disk Swapping and Paging**
- ❖ **Index and Row fragmentation**
- ❖ **Catalog file growth**
- ❖ **Thrashing and Query Execution**

Diagnostics To Collect

- ❖ **Monitor debug output**

- (DBA)> create table iqmondummy (c1 int)

- (DBA)> iq utilities main into iqmondummy start monitor '-debug -interval 20'

- (DBA)> iq utilities private into iqmondummy start monitor '-debug -interval 20'

- (DBA)> iq utilities main into iqmondummy stop monitor

- (DBA)> iq utilities private into iqmondummy stop monitor

- (DBA)> drop table iqmondummy

- ❖ **Output printed to text file with following naming convention:**

- <dbname>.<conn#>-[main|temp]-iqmon,

- Example: unidb.31-main-iqmon

- ❖ **Monitor output file is in .db directory by default**

- ❖ **Monitor_Output_Directory option can be used to control output directory**

IQ Monitor Debug

- ❖ **Buffer manager (main or temp)**
- ❖ **Contention counter**
- ❖ **Dirty page and sweeper thread**
- ❖ **Heap memory manager**
- ❖ **Thread manager**
- ❖ **Free list**
- ❖ **Buffer allocation**
- ❖ **Buffer allocation histogram**
- ❖ **Prefetch information**

Debug Monitor Fields

- ❖ **Hit% (cache_by_type):** Percentage of “Finds” that were already in the cache
 - Should be near 100%
 - Caveat: Find rate may dip at beginning or large query until prefetch is initialized
 - Possible causes of low hit rate
 - Cache too small
 - Algorithm not appropriate for data size (i.e. hash)
 - Poor prefetch (i.e. due to sparse keys in predicate)
- ❖ **GrabbedDirty(Gdirty(cache_by_type)):** Pages that were “grabbed dirty” and synchronously written to disk to free a buffer
 - Indicates that all buffers are either locked or dirty
 - Increase sweeper percent or wash area

Debug Monitor Fields

- ❖ **BusyWaits (Bwaits (cache_by_type) and (contention)):** Find requests that waited while a page was being read from disk or written to disk.
 - Typically near 0
 - Possible causes of high BWait
 - Slow I/O subsystem
 - Too few buffers
 - Caveat
 - Multiple queries reading the exact same pages – first find will read page, subsequent finds will wait for read to complete
- ❖ **LRUNumTimeOuts(Tos(contention)):** Number of times LRU lock timed out.

Debug Monitor Fields

- ❖ **IONumWaits(IOWait(Contention)):** Locks on temporary compression buffer pool that waited
- ❖ **FLMutexWaits(FLWait(Contention)):** Number of times a free list lock waited
- ❖ **BmapHTNumWaits(HTWait(Contention)):** Number of waits of buffer hash table lock
- ❖ **MemNTimesWaited(MemWts(Contention)):** Number of heap locks that waited
- ❖ **PFMgrCondVar(Prefetch):** Statistics for manager lock and important for prefetch performance
- ❖ **CPU Sys Seconds(CPU):** Number of system mode CPU Seconds
- ❖ **CPU Total Seconds(CPU):** Total number of system and user seconds
- ❖ **InUse(Flush):** Number of buffers in cache marked as being used. 100% is normal once cache has filled
- ❖ **Pinned(Flush):** Number of locked buffers in cache
- ❖ **ThrNumFree(Free(Threads)):** Number of free threads available for assignment. If low, could indicate thread starvation.
- ❖ **FLIsOutOfSpace(Freelist):** Flag to indicate when the store is fully allocated(full)

Diagnostics To Collect

❖ Query html plan

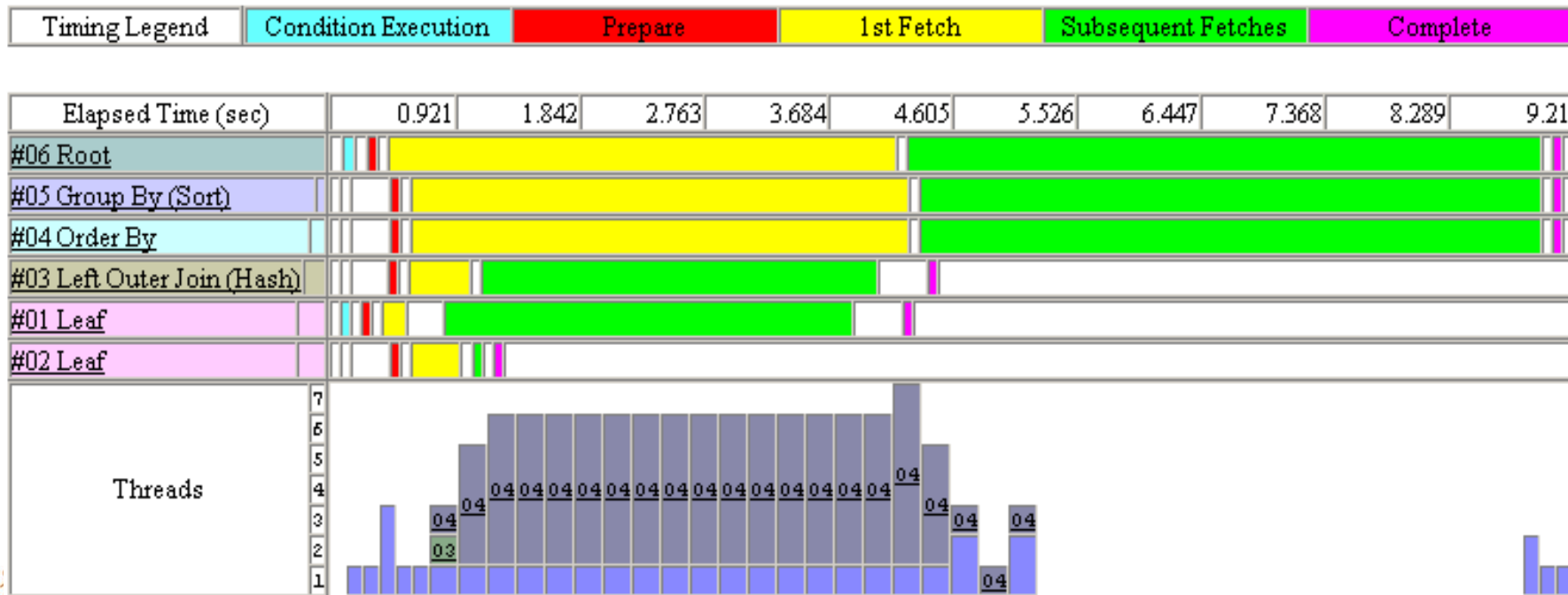
```
set temporary option query_plan = 'ON';  
set temporary option query_detail = 'ON';  
set temporary option query_plan_as_html ='ON' ;  
set temporary option query_plan_after_run='on';  
set temporary option query_timing='on';  
set temporary option DML_OPTIONS10='on';  
set temporary option FORCE_NO_SCROLL_CURSORS ='on' ;  
set temporary option index_advisor='on' ;  
Exec query
```


Common Causes for Poor Query Performance

- ❖ **Inaccurate estimates:**
 - Row Count
 - Distinct Value Counts
- ❖ **Schema Design:**
 - Normalized: specific better performance is seen with selective normalization e.g like conditions
 - De-normalized: selective de-normalization is good for situations with skewed correlations between joined tables
- ❖ **Query expressions less than ideal**

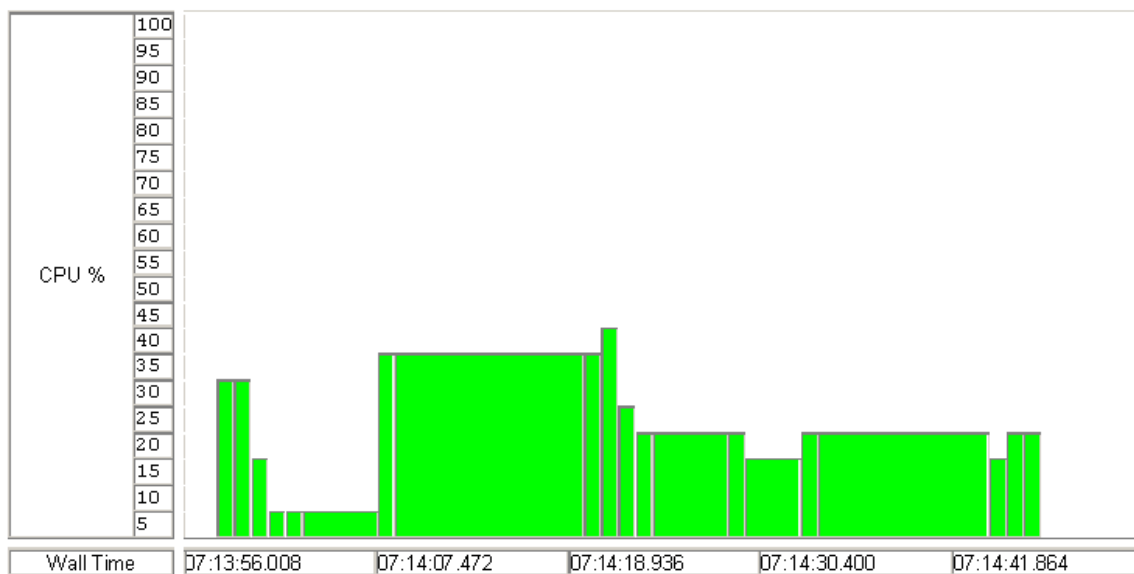
Thread Utilization Diagram

- ❖ Displays threads used over the life of the query
- ❖ Thread usage corresponds to timing Chart



CPU Usage Diagram

CPU utilization diagram: Provides information about the CPU usage throughout the lifecycle of query. Red represents System CPU usage and Green represents User CPU usage.



Query HTML Plan

- ❖ To influence the optimizer's choices for the execution of vertical conditions:
 - Add indexes as suggested by the Index Advisor
 - Supply a user-supplied condition selectivity
 - Set the Index_Preference option
 - Supply a condition hint which allows the user to specify for one condition:
 - Selectivity
 - Index preference
 - Usefulness
 - Execution phase
- ❖ Example 1:

```
( (T.VCHAR1 LIKE '%summ%', 10.50)  
AND (T.VCHAR2 LIKE '%wint%', 'S:2.25')
```

Query HTML Plan

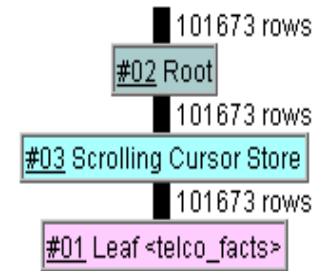
❖ Example 2:

```
SELECT CUSTOMER.*  
FROM CUSTOMER, LOCATION  
WHERE C_LOC_CODE = L_LOC_CODE  
AND L_ZIPCODE = @hv1  
AND (C_LAST_NAME LIKE @hv2, 'E:D, U:1.0')
```

- ❖ A template query where the developer knows an HPDJ algorithm will be used. The condition hint tells the optimizer to delay the execution of the LIKE condition until after the pushed join predicate is evaluated.

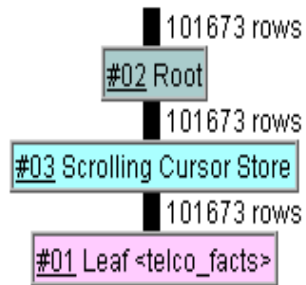
Scrolling Cursor Store Node

- ❖ Scrolling Cursor Store node usually found just below the Root Node
- ❖ Affected by the database option *Force_No_Scroll_Cursors* (default 'ON')
- ❖ The store buffers all rows from a query result set
- ❖ Allows scrolling through results (forward and backwards)
i.e 'Scrollable Cursor'
- ❖ Consumes Temp Cache and Temp Store (for large results)
- ❖ Typically this option is set OFF for better query performance
- ❖ Some query client and ETL tools may require this option be set 'ON'



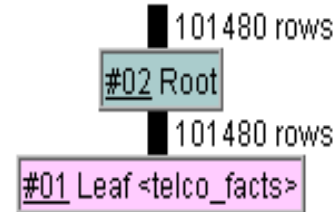
Temp Memory Usage with Scrolling Cursors

Force_No_Scroll_Cursors='OFF'



#02 Root	
Child Node 1	#03
Generated Result Rows	101673
Estimated Result Rows	39000000
User Name	DBA (SA connHandle: 1 SA connID: 4)
Est. Temp Space Used (Mb)	2082.8
Act. Temp Space Used (Mb)	5.5
Cursor Name	SQL_CUR_e023a28
Requested attributes	Scroll Read Only Hold Chained

Force_No_Scroll_Cursors='ON'



#02 Root	
Child Node 1	#01
Generated Result Rows	101480
Estimated Result Rows	39000000
User Name	DBA (SA connHandle: 1 SA connID: 4)
Est. Temp Space Used (Mb)	0.0
Act. Temp Space Used (Mb)	0.0
Cursor Name	SQL_CUR_e023a28
Requested attributes	No Scroll Read Only Hold Chained

Leaf Node – Represents Access To A Table In The IQ Store

- ❖ All table indexes and columns used in the query are identified and accessed in the Leaf node
- ❖ Types of Leaf nodes
 - (Regular) Leaf
 - Aggregation Leaf
 - Grouping Leaf
 - Distinct Leaf
 - Ordered Leaf
 - Proxy Leaf (Proxy Table)
 - SA Leaf (table in the SA database)

Leaf Node Details

Row Counts

- ❖ Total rows in the table
- ❖ Generated rows after conditions(if any)

Conditions (Predicates)

- ❖ Selectivity
- ❖ Usefulness
- ❖ Elapsed time(to execute)
- ❖ Rows remaining (post execution)
- ❖ Index Used
- ❖ Threads Used

#03 Leaf	
Table Name	CUSTOMER
Parent Node	#05
Table Row Count	7500000
Generated Result Rows	1501166
Estimated Result Rows	1501166
Generated Post Invariant Predicate Rows	1501166
Estimated Post Invariant Predicate Rows	1501166
Invariant Predicate Thread Allowance	3
Parallel Source Work Units	9
Effective Primary Key Column 1	CUSTOMER.C_CUSTKEY
Condition 1 (Invariant)	(CUSTOMER.C_MKTSEGMENT = 'AUTOMOBILE')
Condition 1 Selectivity	0.20015547
Condition 1 Usefulness	3.79984453
Condition 1 Elapsed time	0:00:00.070056
Condition 1 Rows remaining after condition	1501166
Condition 1 Execution Method	Probe lookup table into bitmap cursor
Condition 1 Index	FP(1) DBA.CUSTOMER.ASIQ_IDX_T741_C7_FP
Condition 1 Number of Threads Used	8

Selectivity, Index Selection And Usefulness

For each predicate the Optimizer determines

- ❖ **Selectivity:** Portion (percentage) of the table rows that satisfy that predicate.
- ❖ **Index Used** to resolve the predicate.
- ❖ **Usefulness Score**
- ❖ **Estimated Selectivity:** If no metadata exists the optimizer estimates selectivity based on the predicate operator

Predicate Type	Percent of Table(Estimated)	Selectivity
Equality (=)	20%	0.2000000
Open Range(>)	40%	0.4000000
Between	40%	0.4000000
Like (%)	20%	0.2000000
Inter-Column equality(t.a=t.b)	30%	0.3000000
Inter-column comparison	50%	0.5000000

Index-Based Selectivity

- ❖ When a 'useable' index exists it can provide accurate selectivity
 - Exact selectivity is shown as a precise number without trailing zeroes such as: 0.34982376

Note: Some functions can negate the ability to use an index for selectivity

Example: `SUBSTRING(t.a, 5, 5) = 'homes'`

Optimizer might use default selectivity

20% for an equality search (.20000000)

Usefulness

- ❖ **Usefulness value ranges: 0.0 to 10.0 (10.0 = most useful)**
 - Predicate with highest score is executed first
 - Remaining predicates executed in descending order of usefulness score
- ❖ **Factors determining Usefulness scoring**
 - Selectivity – how many rows in the table will be eliminated by this predicate
 - Execution Phase
 - Type of predicate (operator)
 - Index available (including optimized FP index)
 - Speed of predicate execution
 - Resources required to execute the predicate

Metadata For Join Performance

- ❖ Individual columns which are primary keys always declare as a PRIMARY KEY (or create a UNIQUE HG index)
 - Candidate key columns create a UNIQUE HG index if used in joins
- ❖ All other join columns create an HG index
- ❖ Multi-column primary keys where tables will be joined using all the keys create a multi-column PRIMARY KEY (or a multi-column UNIQUE HG index)
 - If all keys are NOT used in a Join then a Primary Key (or multi-column Unique index) is not necessary
 - For example, the Primary Key for a Fact table would almost never be necessary (except to maintain entity integrity)

Checklist

- ❖ **Missing Indexes**
- ❖ **Missing IQ Unique**
- ❖ **Mismatched Datatypes**
- ❖ **Sub-Optimal Datatype choice**
- ❖ **Review Join Conditions**
- ❖ **Check for Over-Aggressive Push-Downs**
- ❖ **Check for Cache Thrashing**
- ❖ **Excessive Data Skew**
- ❖ **Missing Referential Constraint**
- ❖ **Thread Starvation**
- ❖ **Excessive Expression Evaluations**

Missing Indexes

- ❖ **Check for Base Distinct (EST)**
 - Only concentrate on this line if the optimizer needed the distinct count
 - Requires HG or have flat N-bit
 - Wide-range predicates on numbers requires HNG
 - Infra-table column comparisons require CMP index
- ❖ **Selectivity 'magic' numbers**
 - Check for selectivity e.g 0.2000000

Missing IQ Unique

- ❖ Look for 'Projection Bytes' in Leaf node for low cardinality columns
- ❖ Each release of IQ adds more algorithms that can be pushed into the enumeration table
- ❖ Can make large performance difference for large tables
- ❖ Cannot be added by Alter Table

Mismatched Datatypes

- ❖ Some more efficient join algorithms cannot be used with mismatched datatypes (signed/unsigned, char/varchar length etc)
- ❖ May cause less efficient algorithms to be used

Suboptimal Datatype Choice

- ❖ Common cause of sub-optimal performance is not using optimal data types
- ❖ All tables should have keys, and keys should be an integral data type, and preferably unsigned
- ❖ Do not use CHAR or VARCHAR columns as keys for joins, ORDER BYs, or GROUP BYs if at all possible

Review Join Conditions

- ❖ Look for: Join (Nested Loop)
- ❖ Optionally followed by a Filter node
- ❖ May be a missing join condition
- ❖ May be a join condition that precludes using an efficient join algorithm
- ❖ Disjunctions (<condition> OR <condition>)
- ❖ Range predicates (<, >, Between, Like)
- ❖ Complex expressions ($T1.x + T2.y = 5$)

Over-Aggressive Push-Downs

- ❖ Duplicate predicate selectivity
- ❖ NLPD: Especially sensitive to too many pushes
- ❖ Meaning of semi-join in plan: actually IN cost estimates

SemiJoin V: 2 D: 77446.5 T: 0 A: ORs S: 0 O: 0

V – Actual # distinct values for the IN

D – Est. # rows per distinct value

T – Actual time taken

A – Algorithm chosen: bitmap 'OR' or column scan

S – Est. cost using column scan

O – Est. cost using bitmap disjunction

Cache Thrashing

- ❖ **Hash Thrashing**
 - Happens when significant under-estimate
 - Causes excessive reads and writes from the temp cache
 - Solution: fix estimates or use temp table or avoid hash
- ❖ **Correlated Subquery Thrashing**
 - Causes excessive reads and writes from the main cache
 - Solution: rewrite the query as a join instead of a subquery
- ❖ **Confirm using IQ buffer cache monitor report**

Excessive Data Skew

- ❖ **IQ assumes modest data skew**
- ❖ **Sometimes data is heavily skewed**
 - **Mostly affects push-down joins**
 - **No simple way to workaround**

Missing Referential Integrity Constraint

- ❖ multicolumn primary keys must be defined
- ❖ In joins, where it affects estimates, look for:
Join Result Constraint: Many to Many
- ❖ In a GROUP BY it affects performance directly by forcing the engine to sort or hash on extra columns.

Thread Starvation

- ❖ **Configured by startup Switch**
 - Total threads in server : -iqmt
 - Set Max_IQ_Threads_Per_Connection
 - Can generally be set safely up to 100
- ❖ **Examine using IQ cache monitor with -debug**
- ❖ **Look for 'Serial' in Query_Plan_After_Run on UNION, and non-push-down join nodes**

Excessive Expression Evaluations

- ❖ **Look at production lists**

Production 1: $(T.B * T.C)$

Production 2: $(T.A * (T.B * T.C))$

- ❖ Seeing a **CAST** in a production may be a symptom of mismatched data types
- ❖ Finding nodes with very high row counts that are producing complex expressions may imply a less than optimal schema design
 - Consider creating a derived column to hold a pre-computed expression result for each commonly used expression

Useful Links

- ❖ **Product documentation**

<http://help.sap.com/iq>

- ❖ **How to get best results from an SAP Search**

<https://service.sap.com/sap/support/notes/2081285>

- ❖ **Support Portal**

<https://support.sap.com/home.html>

- ❖ **SAP Communities**

<http://scn.sap.com/community/iq>

- ❖ **Social Media Product Support Channels**

<https://www.facebook.com/SapProductSupport>

<https://twitter.com/SAPSupporthelp>

Useful Links

- ❖ **IQ WIKI**
<http://wiki.scn.sap.com/wiki/display/SYBIQ/IQ>
- ❖ **IQ 16 Best Practices**
<http://scn.sap.com/docs/DOC-39896>
- ❖ **IQ Hardware Sizing Guide**
<http://scn.sap.com/docs/DOC-41455>

Questions and Answers

Thank You for Attending

Please complete your session feedback form