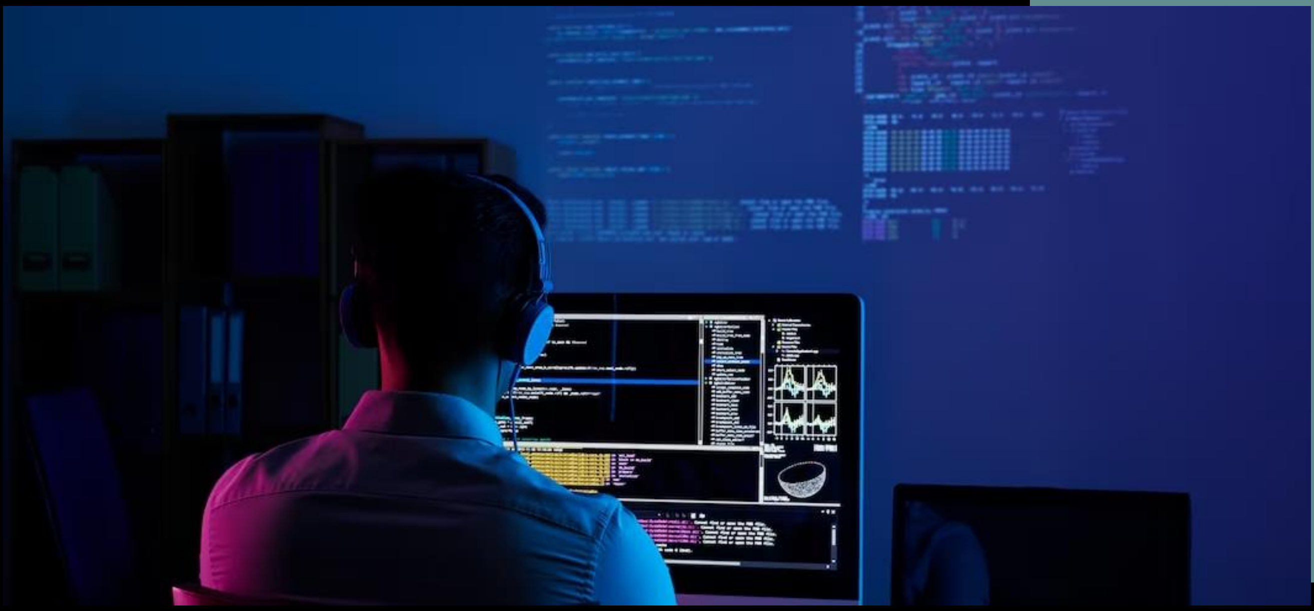
Db2 India

TECHNICAL NEWSLETTER

IBM

APRIL 2023



Technotes

Training Links

Case Studies

Follow us!

Stay up to date on the latest news from the Db2 team

Please Click Here to share your feedback on this Newsletter

Technotes

- How to re-create the resource model and restart the RSCT domain in Db2 pureScale environment
- Db2 PureScale
- Creating an HADR Db2 instance on a Pacemaker-managed Linux cluster with the Db2cm utility
- Recover table data even if the corresponding index EMP page is corrupted
- The resetrsrc command - A brief how-to guide
- IBM AIX: behavior of host name resolution
- HADR and NAT support
- Advanced Log Space Management
- Db2 Backup fails with ADM6006E with Db2 V11 running on Linux Redhat using XFS file system
- How do you extract the Db2diag.log file in a Db2 Warehouse environment?

Training Links

- DB2 PURESCALE IN AWS DEPLOYMENT

IBM® Db2® pureScale® is a Db2 feature that reduces the risk and cost of business growth by providing extreme capacity where you can start with a small cluster, grow as you need online; continuous availability where the automatic recovery orchestration on component failure allows transactions to continue on other healthy members in the event of planned or unplanned member outage; application transparency to avoid the need to make the application cluster-aware to get good scalability, thereby reducing the risk and cost of application changes. This self-managed IBM® Db2 pureScale with TCP/IP interconnect on AWS offering is an extension of the above. They share the same value propositions of the architecture built on the familiar and proven design features from Db2 for z/OS database software to deliver exceptional levels of database availability that satisfy the unique workloads of many applications.

- DB2 WAREHOUSE DEPLOYMENT

The IBM Db2 Click to Containerize family encompasses several tools which provide customers with the ability to quickly modernize their Db2 landscape. The Db2 Shift utility is part of the Click to Containerize family and can be used to clone a copy of Db2 into an OpenShift, Kubernetes, Cloud Pak for Data (CP4D), or a standard Db2 instance.

UPCOMING TRAINING

We are coming up with a training session on: “DB corruption and recovery options of db2dart command”

Please join and clear your doubts after the session. Date & Time: 30th March 2023 at 3-4 pm IST

Location: Click Here to Join

Click Here to share feedback on the training session.

Case Study

LOB Insert Update Performance

If you are using LOB data in your database, you need to keep several designing aspects in mind

In one recent case, several slowness is seen when inserting or updating table having LOB data

We see many Insert/Update on the same table going into Lock-wait

If we dump and check the process stack file of the lock holding agent, we see the following function stack

thread_wait

getConflictComplex

getConflict

sqlbfix

sqldx_fixPage

sqldxGetBseg

sqldxGetBseg

sqldx_ld_alloc_bsegs

sqldxBuildDescriptor

This shows latch contention while creating a buddy segment during LOB insertion

Checked and revealed there are several Lob columns in Table.

We can get more information about LOB in the below tech-note.

https://www.ibm.com/support/pages/lob-space-allocation-and-management

Steps to improve LOB Insert/Update performance

Check if The LOB architectural limit is getting reached

https://www.ibm.com/support/pages/db2-fails-architectural-page-limit-reached-errors-while-performing-insert-data-table-blob-field

db2pd -tcbstats will show

LobSize - The number of pages in the large object.

Check if any of the LOB columns can be converted into Inline LOB

Inline LOBs improve the performance of queries that access LOB data, because no additional I/O is required to fetch, insert, or update this data. Moreover, inline LOB data is eligible for row compression.

ADMIN_EST_INLINE_LENGTH function can be used to estimate the inline length

https://www.ibm.com/docs/en/db2/11.5?topic=aracp-admin-est-inline-length-estimate-length-required-inline-data

Check if a separate tablespace for all LOB data can be created

Bufferpool is not used to cache LOB data

Make sure all LOB data is stored in a separate tablespace and enable FSC for that tablespace

This will ensure caching for LOB data and improve performance

Case Study

Performance issue due to Sequence generation

As part of of performance issue debugging we collect process stacks

db2pd -stack all -repeat 5 5

The above command will collect process function stacks for all processes 5 times every 5 secs

In this process we can see the slow agent is in the following function stack

SQLO_SLATCH_CAS6418getConflictComplexEm

*sqldSeqGetP8sqeAgentiiiP8SQLD_SEQPS2_*

5sqldSeqGenerateP8sqeAgentP8SQLD_SEQ

sqlri_SeqGetNextP8sqlrr_cb

And this process is holding the latch SQLO_LT_SQLD_SEQ__seqLatch

We can also see many processes waiting on the above latch

Some agents can also be seen stuck while writing to the log

sqloWaitThreshold

sqlpgildP9sqeBsuEduP14sqlpMasterDbcbmb

sqlpflogP9sqeBsuEduP14sqlpMasterDbcbPK9SQLP_LSN8

sqlbgbWARMR12SQLB_WARM_CB

_sqlbProcessTPL_CallWARMP8sqeAgentR

sqlbProcessTPLP8sqeAgentP

sqlpWriteToLogP8qeAgentP

So, the above information clearly shows that the bottle neck is while creating the sequence cache

Check for all the sequences in the database

Increasing the cache value for the sequence resolves the performance issue

Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence.

One important point to remember is In the event of a system failure, all cached sequence values that have not been used in committed statements are lost