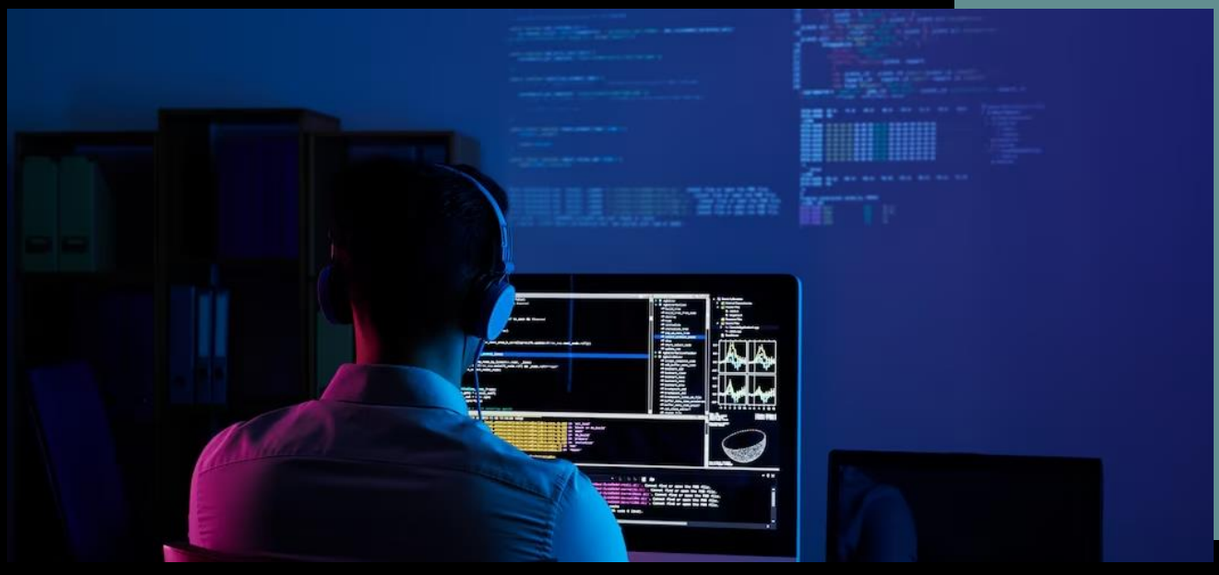
Db2 India

TECHNICAL NEWSLETTER

JUNE
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

Db2 India

TECHNICAL NEWSLETTER

JUNE
2023

Technotes

- Db2 Tablespace Full condition resolution
- Db2 V11.5 Upgrade restrictions
- Db2 resolve Index corruption
- Db2 commands slow in AIX
- Db2 simple steps of native encryption
- Getting Db2 encryption info
- Db2 Event Monitor creation

Please Click Here to share your feedback on this Newsletter

Db2 India

TECHNICAL NEWSLETTER

JUNE 2023

Training Links

- Db2 PURESCALE IMPLEMENTATION IN LINUX

The main purpose of this paper is to describe detailed steps on how to configure a Db2 pureScale environment on Linux VM images. The database administrators should be able to create an environment for their own testing and learning purposes by following this document. This will be a TCPIP-based Db2 pS instance, therefore, you cannot consider it for any performance benchmarking. For real-time performance testing, you must configure Db2 pS with high-speed interconnect.

While setting up Db2 pS instance on VM images for learning and testing purposes, you will not require any high-speed interconnect and SAN storage. We will make use of a TCPIP network along with ISCSI drivers to configure a Db2 pS instance here.

This document covers detailed steps on how to create a Db2 pS instance consisting of two members and two CFs.

UPCOMING TRAINING

We are coming up with a training session on: "Db2 Database Monitoring"

Please join and clear your doubts after the session. Date & Time: 27th June 2023 at 3-4 pm IST

Location: Click Here to Join

Click Here to share feedback on the training session.

Please Click Here to share your feedback on this Newsletter

Db2 India

TECHNICAL NEWSLETTER

IBM

JUNE 2023

Case Study

PERFORMANCE ISSUE DUE TO SEQUENCE GENERATION

This time we will discuss a case study related to performance issue due to Sequence generation

As part of performance issue data collection, we collected db2mon report and db2fodc
In the db2mon report we could see unexpected high values for
PCT_LG_DSK = 92.32
LOG_WRITE_TIME_PER_IO_MS = 2.9636

In an ideal OLTP environment LOG_WRITE_TIME_PER_IO_MS should be less than 1 MS
So, 2.9 MS is a high value

The process which does the log writing is db2loggw
To check why this log write is high we see the process stack of the db2loggw process

We see sequence generation functions before the log write
sqloWaitThreshold
_Z8sqlpgildP9sqeBsuEduP14sqlpMasterDbcbmb
_Z14sqlpWriteToLogP8sqeAgentP11SQLP_TENTRYmmmP14SQLP_LREC_PARTP9SQLP_LSN8Pm
_Z11sqlpWriteLRP8sqeAgentmmmmmP14SQLP_LREC_PARTmP9SQLP_LSN8Pm
_Z23sqlpWriteLRSingularTranP8sqeAgentmmmmmP14SQLP_LREC_PARTPmP9SQLP_LSN8bS3_S3
_Z15sqldSeqGenerateP8sqeAgentP8SQLD_SEQ
_Z20sqlri_SeqGetNextImplPK10sqlz_valueS1_S1_PS_S2_S2_S2_PmP8sqlrr_cb

We will also see Db2 agents stuck in this stack in write to log waiting for sequence latch
SQLO_LT_SQLD_SEQ__seqLatch

sqloWaitThreshold
sqlpgildP9sqeBsuEduP14sqlpMasterDbcbmb
sqlpflogP9sqeBsuEduP14sqlpMasterDbcbPK9SQLP_LSN8
sqlbgbWARMR12SQLB_WARM_CB
_sqlbProcessTPL_CallWARMP8sqeAgentR
sqlbProcessTPLP8sqeAgentP
sqlpWriteToLogP8qeAgentP

Continued...

Db2 India
TECHNICAL NEWSLETTER
JUNE
2023

Case Study

PERFORMANCE ISSUE DUE TO SEQUENCE GENERATION

This time we will discuss a case study related to performance issue due to Sequence generation

Continued…

The holder db2 agent of the sequence latch is in latch conflict waiting to generate the sequence

SQLO_SLATCH_CAS6418getConflictComplexEm
sqldSeqGetP8sqeAgentiiiP8SQLD_SEQPS2_
5sqldSeqGenerateP8sqeAgentP8SQLD_SEQ
sqlri_SeqGetNextP8sqlrr_cb

So, this leads to the performance issue

The agent is generating the sequence and holding SQLO_LT_SQLD_SEQ__seqLatch latch
other agents trying to write to log and waiting for this latch to generate sequence

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

To avoid this issue check for all the sequences in the database

Increasing the cache value for the sequence resolves this 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