

Accelerated Database Recovery – A Deep Dive Behind the Magic



SQL Saturday Atlanta 2024

John Morehouse
Principal Consultant



John Morehouse

Principal Consultant


Denny Cherry & Associates

✉ john@dcac.com

 [/in/johnmorehouse](https://www.linkedin.com/in/johnmorehouse)

 [@SQLRUS](https://twitter.com/SQLRUS)

 Sqlrus.com

 He/Him



Community Speaker

Blogger/Tweeter

Conference Organizer

MVP – Data Platform

Friend of Redgate

VMWare vExpert



Certified IT professionals to help achieve IT goals

Clients ranging from small business to Fortune 10 corporations

Help save on costs while improving IT reliability and solving challenges





<https://bit.ly/mypresentationfiles>





Please give a big thank you to
our sponsors & stop by their tables!



Objectives



What is it & why do we care

Components


Rollbacks/Recovery

Demo

Q&A



What is it?



"Accelerated database recovery (ADR) improves database availability, especially in the presence of long running transactions, by redesigning the SQL database engine recovery process."



Why do I
even care?



STILL WAITING FOR ROLLBACK TO FINISH

Accelerated Database Recovery



Faster and consistent database recovery

Almost instantaneous transaction rollback

Aggressive log truncation

Available in Standard Edition

Default in Azure SQL Database/Managed Instance

Azure SQL DB



```
1  
2 DECLARE @version NVARCHAR(124)  
3 SELECT @version = @@VERSION  
4 PRINT @version;  
5 go  
6 ALTER DATABASE Demo1 SET ACCELERATED_DATABASE_RECOVERY = OFF;
```

70 %

Messages

Microsoft SQL Azure (RTM) - 12.0.2000.8
Sep 18 2021 19:01:34
Copyright (C) 2019 Microsoft Corporation

Msg 40517, Level 16, State 1, Line 6
Keyword or statement option 'accelerated_database_recovery' is not supported in this version of SQL Server.

Completion time: 2021-12-10T21:43:18.4331296-05:00



New Components

ADR New Components

sLog

Persisted Version Store

Logical Revert

Aborted Transaction Map

Cleaner



YAY FOR ADR!!!



In-memory object

Contained non-versioned operations

Checkpoints persist contents to disk

Provides aggressive log truncation

FN_DBSLOG(null,null)

Persisted Version Store



Same version store when using RCSI

Retains off-row versions

Availability Group aware

Logical Revert



Facilitates the “undo” at the row-level

Asynchronous in nature

Keeps track of all aborted transactions

Releases all locks immediately after abort

Aborted Transaction Map

Not that ATM →

Tracks all aborted transactions

Implemented as a hash table

Rows are linked to PVS

DMV: Sys.dm_tran_aborted_transactions



Cleaner

Removes versions that are not needed

Asynchronous process

Thread architecture

Cannot disable

Force clean up via
`sp_persistent_version_cleanup`
(synchronous)



ADR Configurations



Select * from sys.configurations where name like 'ADR%';

ADR cleaner retry timeout

ADR Preallocation Factor*

How to Enable ADR



Requires exclusive access to the database

Can specify specific/different filegroup

```
ALTER DATABASE [db_name]
```

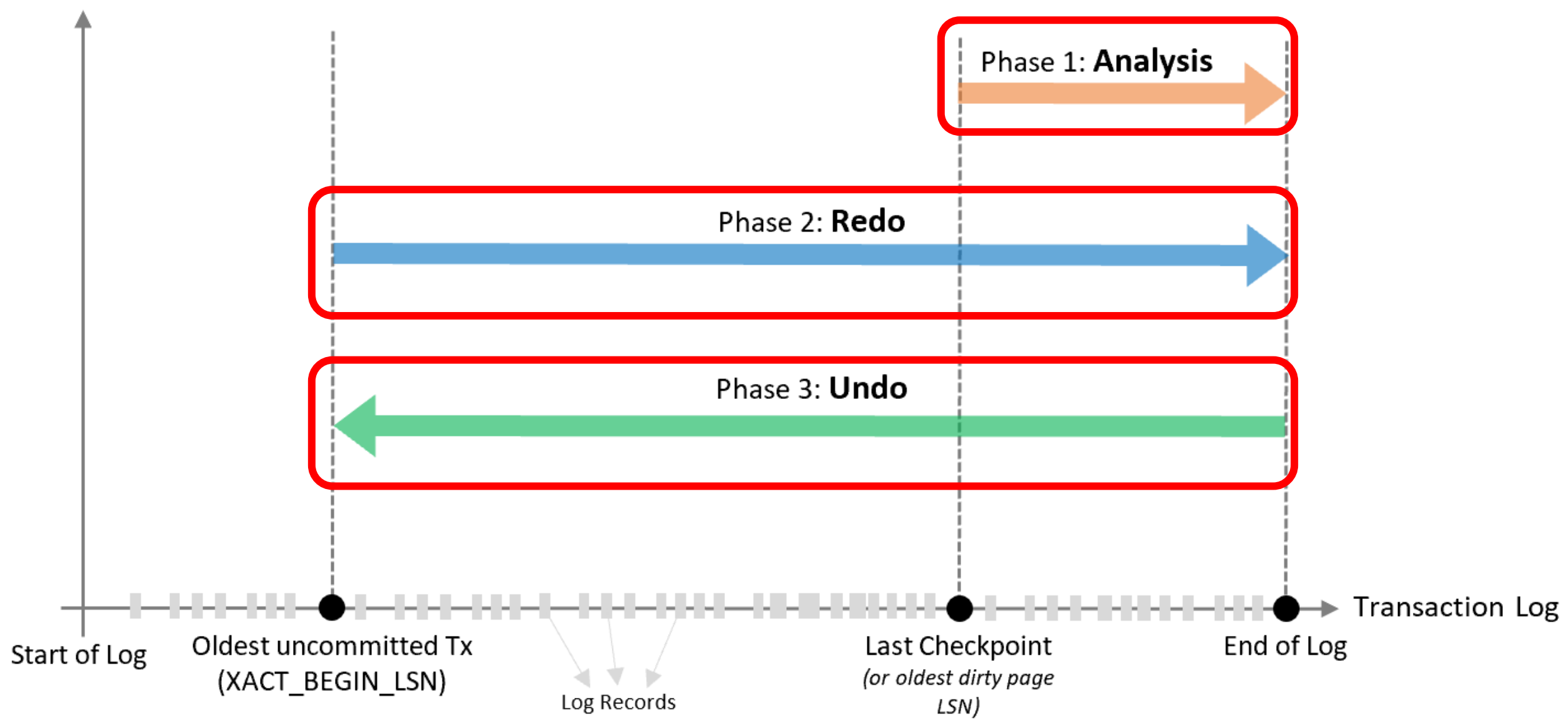
```
SET ACCELERATED_DATABASE_RECOVERY = ON
```

```
(PERSISTENT_VERSION_STORE_FILEGROUP =  
[SomeOtherFG])
```




Rollback

Recovery Phase / Transaction Log (without ADR)



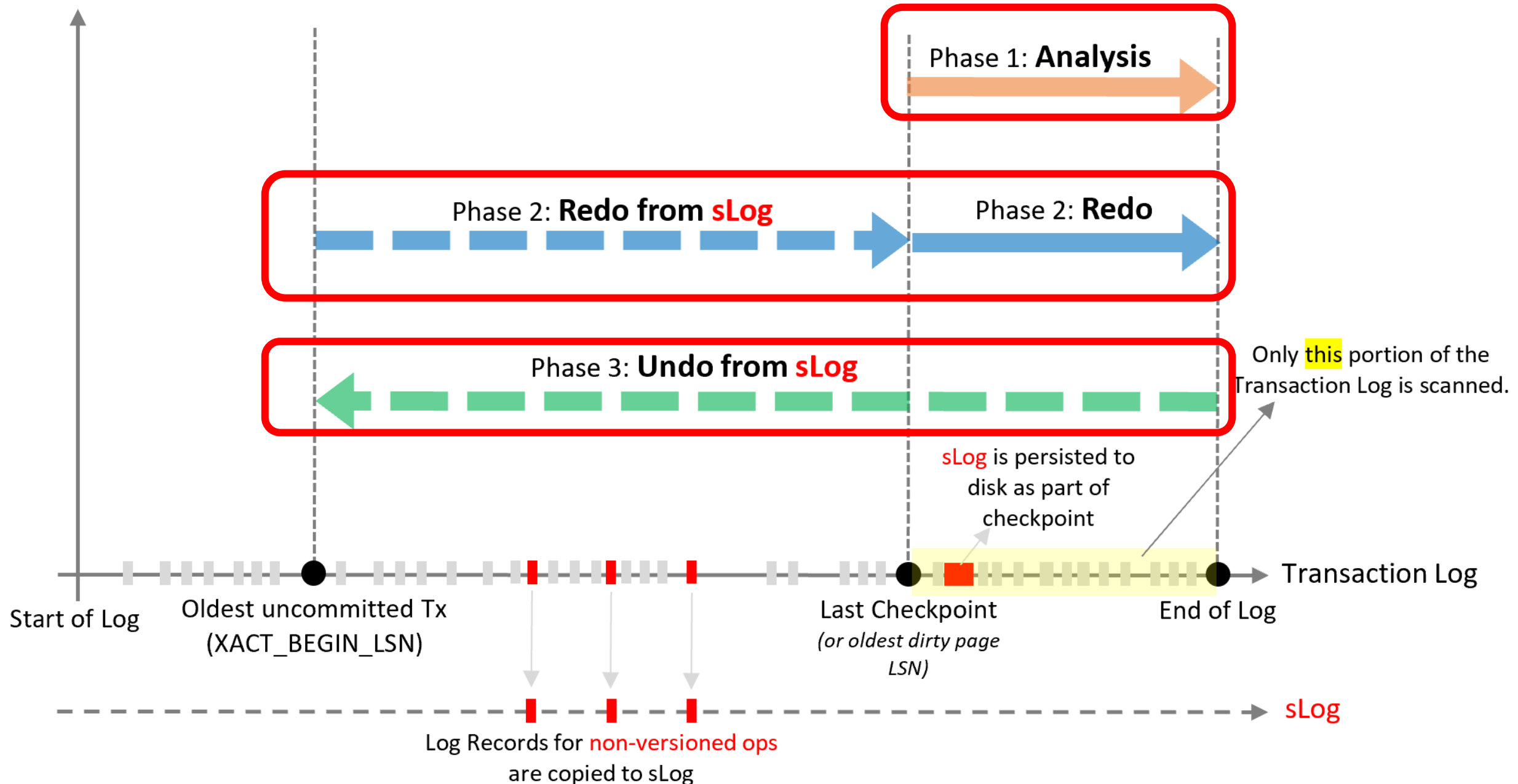


Rollin',
rollin',
rollin'

Rawhide!

**Rollback is
proportional to
longest running
transaction**

Recovery Phase / Transaction Log / sLog (with ADR)





DEMO

When Not to Deploy ADR



Extremely highly transactional

High volume of data modifications

Most likely edge case

Monitor PVS Store for issues

Summary



Faster rollbacks

Faster crash recovery upon startup/failover

Aggressive log truncation

Enable for majority of use cases

Simple to enable

Resources



SQLCallStackResolver – <http://aka.ms/sqlstack>

Article - <https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/how-does-accelerated-database-recovery-work/>

Whitepaper - <http://bit.ly/SQLADRExplained>



Questions?
Answers!

**Got
Questions?**

**Follow Me on
~~Twitter~~ X!**

**Check out
my blog!**

John Morehouse

Denny Cherry & Associates Consulting

DCAC*



john@dcac.com



Sqlrus.com



@SQLRUS



/in/johnmorehouse



Slides & Demos