**Accelerated Database Recovery (ADR) in SQL Server** is a new feature introduced in SQL Server 2019 to improve the performance of database recovery and minimize the recovery time during crash recovery or the time spent during transaction log rollback and undo operations. ADR aims to reduce the time spent on database recovery after an unexpected shutdown and provide more predictable recovery times.

### **Key Concepts and Benefits of ADR**

1. **Faster Recovery Times**: ADR enables faster crash recovery and shorter transaction log rollback times. SQL Server traditionally spends a lot of time undoing or rolling back transactions during recovery. With ADR, the system improves this by utilizing a new architecture that optimizes these processes.
2. **Instant Undo**: Instead of tracking every change to individual pages (like in traditional recovery), ADR introduces a different approach. SQL Server can immediately "undo" changes based on transaction metadata, thus speeding up recovery.
3. **Minimal Impact on Transaction Processing**: ADR allows transactional workloads to be processed with minimal interruptions, reducing the impact on your system when recovering from an unexpected shutdown or crash.
4. **Automatic Logging and Metadata Management**: ADR keeps a new structure that tracks changes at a more granular level to reduce the rollback time, providing more consistent performance in recovery.

### **How ADR Works**

ADR introduces a new mechanism to handle recovery in SQL Server. The traditional method of recovery included several phases:

1. **Analysis Phase**: The system scans the transaction log and data files to check for uncommitted transactions.
2. **Redo Phase**: The system re-applies committed transactions.
3. **Undo Phase**: The system undoes transactions that were in progress at the time of the crash.

ADR, however, optimizes and redefines this recovery process:

* **Transactional data** is tracked in memory during normal operations. This memory structure makes undo operations fast by simply referencing the metadata of the transaction and not needing to re-apply page-level changes.
* **Metadata-based Recovery** means that SQL Server no longer needs to roll back each individual page that was modified during a transaction. Instead, it simply consults the metadata and manages it more efficiently.

### **Example Use Case of ADR**

Imagine you have an online transaction processing system where large transactions frequently occur. Before the introduction of ADR, if the system crashed due to a power outage or failure, SQL Server would spend a significant amount of time in the recovery process, undoing and rolling back all the uncommitted transactions. This process could take a considerable amount of time, and during this time, the database was unavailable.

With ADR, after a crash, recovery is faster because SQL Server can quickly identify which transactions were committed and which were not and apply the changes more efficiently.

### **ADR's Key Components**

1. **Recovery Metadata**: ADR uses an in-memory structure called *recovery metadata* to track the state of each transaction, providing fast recovery after a crash or restart.
2. **Log Redo Phase with ADR**: During the redo phase, the recovery mechanism re-applies only the necessary transactions without involving a lot of page-level modifications. This is crucial because SQL Server does not need to perform the traditional rollback of transactions from the log files.
3. **Accelerated Undo**: ADR removes the need for undoing transaction changes at the page level. It simply refers to the committed state and handles the changes more efficiently.
4. **Transactional Consistency**: ADR maintains transactional consistency, ensuring that the database is always in a consistent state after a crash. The rollback and undo phases are streamlined to guarantee that no uncommitted transactions are left in the system.

### **Enabling ADR in SQL Server**

ADR is **enabled by default** in SQL Server 2019, and there is no additional configuration required to start using it. However, it can be disabled or managed using specific database options.

To check if ADR is enabled on a database, you can use the following query:

SELECT name, recovery\_model\_desc, is\_auto\_create\_stats\_on, is\_auto\_update\_stats\_on

FROM sys.databases

WHERE name = 'YourDatabaseName';

**To disable ADR for a particular database:**

ALTER DATABASE YourDatabaseName SET ACCELERATED\_DATABASE\_RECOVERY OFF;

**To re-enable ADR for a database:**

ALTER DATABASE YourDatabaseName SET ACCELERATED\_DATABASE\_RECOVERY ON;

### **Performance Considerations**

* **ADR may increase write amplification** on the transaction log. This means that while recovery will be faster, the log writes may be higher during normal operations. However, this is offset by the overall speed of recovery and transaction handling.
* The ADR feature provides a consistent and predictable recovery time, which can improve the overall database availability during recovery scenarios.

### 

### 

### 

### 

### **Practical Example Scenario**

Imagine you are working with an e-commerce website's backend database. It is running SQL Server with ADR enabled, and one day, the server crashes due to a power failure. Here's what happens after the crash:

1. **Post-Crash Recovery**: When SQL Server is restarted, the ADR mechanism kicks in. It immediately identifies the committed and uncommitted transactions using the recovery metadata and processes only the necessary transactions to ensure that the database is in a consistent state.
2. **Log Rollback**: If there were any transactions that were not committed before the crash, ADR efficiently handles the undo phase without rolling back each page individually, speeding up the recovery process.
3. **Completion of Recovery**: The entire recovery process is completed much faster than the traditional recovery, reducing downtime and improving the availability of the database.

### **Summary:**

Accelerated Database Recovery (ADR) in SQL Server significantly improves database recovery time by simplifying and optimizing the recovery process. It is a great addition for systems where recovery time is critical, such as large-scale, high-availability applications. By tracking transaction metadata and streamlining the undo and redo phases of recovery, ADR minimizes the impact of crashes and improves overall database performance.

If you're using SQL Server 2019 or later, ADR is enabled by default, and it's a good idea to monitor its behavior for performance benefits and impact on your system's recovery efficiency.