7.4 Recovery

This section has been set as optional by your instructor.

Failure scenarios

The recovery system supports atomic transactions by ensuring partial transaction results are not saved in a database. The recovery system supports durable transactions by ensuring committed transactions are not lost due to hardware or software failures.

The recovery system must manage three failure scenarios:

1. A ***transaction failure*** results in a rollback. The application program may initiate a rollback due to logical errors. The database system may initiate rollback due to deadlock or insufficient disk space. The operating system may initiate rollback if a hardware or software component fails. Regardless of the cause, the recovery system restores all data changed by the transaction to the original values.
2. A ***system failure*** includes a variety of events resulting in the loss of main memory. Databases initially write to main memory blocks, which are lost when an application, the operating system, or the database system fails. Blocks are subsequently saved on storage media, which normally survive a system failure. If main memory is lost before blocks are written to storage media, data written by committed transactions might be lost. In this event, the recovery system:
    - Recovers data written to main memory, but not storage media, by committed transactions.
    - Rolls back data written to storage media by uncommitted transactions.
3. A ***storage media failure*** occurs when the database is corrupted or the database connection is lost.

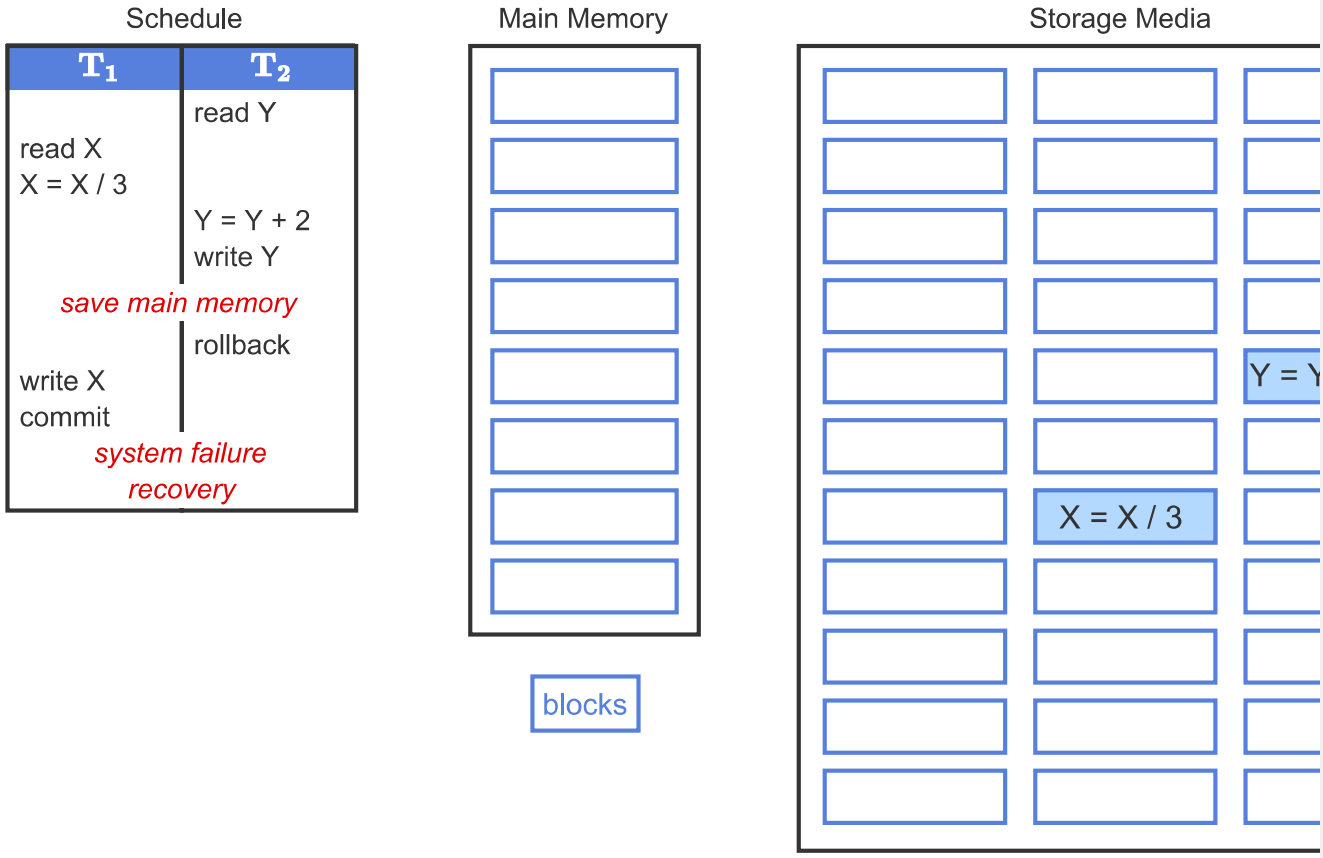
    Many storage systems automatically make redundant copies of data. If one copy of data is corrupted, the storage system automatically switches to a backup copy without intervention by the database or operating system. Nevertheless, storage media do fail occasionally. Alternatively, the connection between the processor and database server might fail. Either way, the database is unavailable to the application.

In principle, recovery from storage media failure is similar to recovery from system failure. In both cases, the recovery system must restore committed transactions and roll back uncommitted transactions. However, storage media failure may cause massive data loss with

lengthy recovery times. Consequently, most commercial databases implement special techniques for rapid recovery from storage media failure.

PARTICIPATION ACTIVITY 7.4.1: System failure.

1 2 3 4 5 6 7 2x speed



The recovery system must restore 'X = X / 3' and roll back 'Y = Y + 2' in storage media.

Captions

1. The database processes two transactions.
2. The database writes a new Y value to a block in main memory.
3. The database saves main memory to storage media. All saved blocks in main memory are released for reuse.
4. The database rolls back T_2, restoring Y's original value in main memory.
5. The database writes a new X value in main memory and commits T_1.
6. Before main memory is saved to storage media, the operating system fails, and main memory is lost.
7. The recovery system must restore 'X = X / 3' and roll back 'Y = Y + 2' in storage media.

Feedback?

PARTICIPATION ACTIVITY 7.4.2: Failure scenarios.

Select the failure scenario that best describes the example.

1) Application programs run on a client machine. The database runs on a separate server machine. The network between client and server fails, and the database does not respond to any application requests.
   - ○ Transaction failure
   - ○ System failure
   - ◉ Storage media failure

Correct

From the perspective of the operating system and application program, an unresponsive database is the same as a corrupted database. Regardless of cause, an unresponsive database is handled as a storage media failure.

2) The database detects two deadlocked transactions. To break the deadlock, the database rolls back one of the transactions.
   - ◉ Transaction failure
   - ○ System failure
   - ○ Storage media failure

Correct

Deadlock does not normally indicate a system or storage media problem. Deadlock normally indicates a transaction failure and is resolved by rolling back one or more transactions.

3) For unknown reasons, an application program freezes. The data administrator forces termination and restarts the application.
   - ○ Transaction failure
   - ◉ System failure
   - ○ Storage media failure

Correct

An application can freeze for many reasons. Regardless of cause, terminating the application erases the application's main memory and must be handled as a system failure.

Feedback?

Recovery log

A ***recovery log*** is a file containing a sequential record of all database operations. The log and database are stored on different storage media so the log survives database failures. The recovery system uses the log to restore the database after a failure.

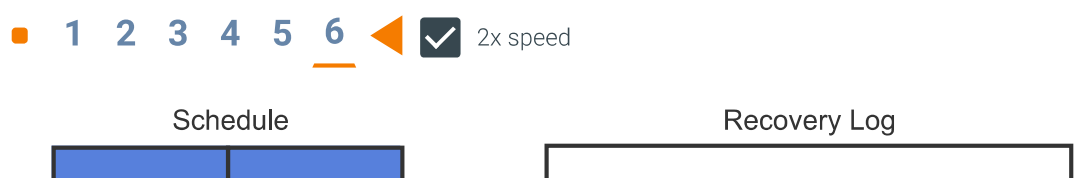
The log contains transaction and data identifiers. Transaction identifiers are assigned by the database system for internal use by the recovery and concurrency systems. Data identifiers correspond to table name, column name, and primary key value, but are compressed or encoded for efficiency.

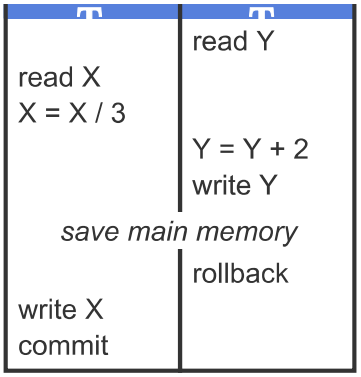
The recovery log contains four types of records:

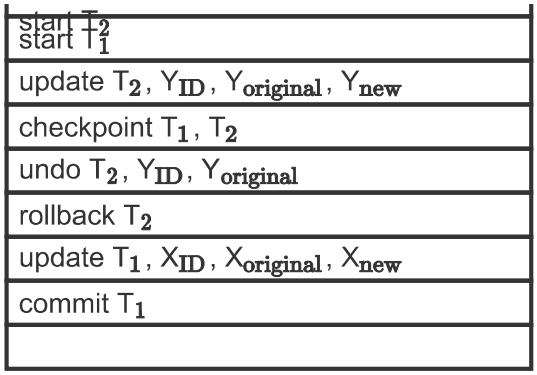
1. An ***update record*** indicates a transaction has changed data. Update records include the transaction identifier, the data identifier, the original data value, and the new data value. Update records may optionally track insert and delete operations.
2. A ***compensation record*** , also known as an ***undo record*** , indicates data has been restored to the original value during a rollback. Compensation records include the transaction identifier, the data identifier, and the restored (original) data value. Compensation records are necessary because the log must capture every database change, including changes executed by a rollback.
3. A ***transaction record*** indicates a transaction boundary. Three types of transaction records exist: start, commit, and rollback. Transaction records include the 'start', 'commit', or 'rollback' indicator and the transaction identifier.
4. A ***checkpoint record*** indicates that all data in main memory has been saved on storage media. When the database executes a checkpoint, transaction processing is suspended while all unsaved data and log records are written to storage media. In the event of a system failure, the recovery system reads only log records following the last checkpoint, rather than the entire file. Checkpoint records include a 'checkpoint' indicator along with the identifiers of all transactions that are active (uncommitted) at the time of the checkpoint.

The above four record types provide a complete history of database operations and enable recovery from all three failure scenarios.

PARTICIPATION ACTIVITY 7.4.3: Recovery log records.







When T_1 commits, a commit record is written in the log.

Captions

1. When the transactions execute, start records are written in the recovery log.
2. When T_2 writes Y, an update record is written in the log.
3. When the system saves all updates from main memory to storage media, a checkpoint record is written in the log.
4. When T_2 rolls back, compensation and rollback records are written in the log.
5. When T_1 writes X, an update record is written in the log
6. When T_1 commits, a commit record is written in the log.

Feedback?

PARTICIPATION ACTIVITY 7.4.4: Recovery log.

1) After all updates have been reversed in a rollback, a(n) _____ record is written in the log.

transaction

Check Show answer

Correct

transaction or rollback

A 'rollback' record marks the completion of a rollback. 'Rollback' is a type of transaction record.

2) When a transaction deletes a table row, a(n) _____ record is written in the log.

update

Check Show answer

Correct

update

Insert, update, and delete operations are all tracked with an 'update' record.

3) A(n) _____ record

indicates all data is saved from main memory to storage media.

checkpoint

Check Show answer

Correct

checkpoint

Saving all changes from main memory to storage media is called a checkpoint. When a checkpoint is initiated by the database or a database administrator, a 'checkpoint' record is written in the log.

4) A(n) _____ record always appears in the log at the beginning of a transaction.

transaction

Check Show answer

Correct

transaction or start

In the log, all transactions begin with a 'start' record. 'Start' is a type of transaction record.

5) A(n) _____ record is written in the log whenever an update is reversed during a rollback.

compensation

Check Show answer

Correct

compensation or undo

'Compensation' or 'undo' records indicate that data is restored to its original value during a rollback.

6) A list of active transactions appears in a(n) _____ record.

checkpoint

Check Show answer

Correct

checkpoint

All checkpoint records include a list of active transactions. The list is used during recovery from a system failure.

Feedback?

Recovery from transaction failure

Recovery from a transaction failure is relatively simple. A database component, such as the concurrency system, instructs the recovery system to roll back transaction T. The recovery system reads the recovery log **backwards**, searching for update records for T. For each update record, the recovery system:

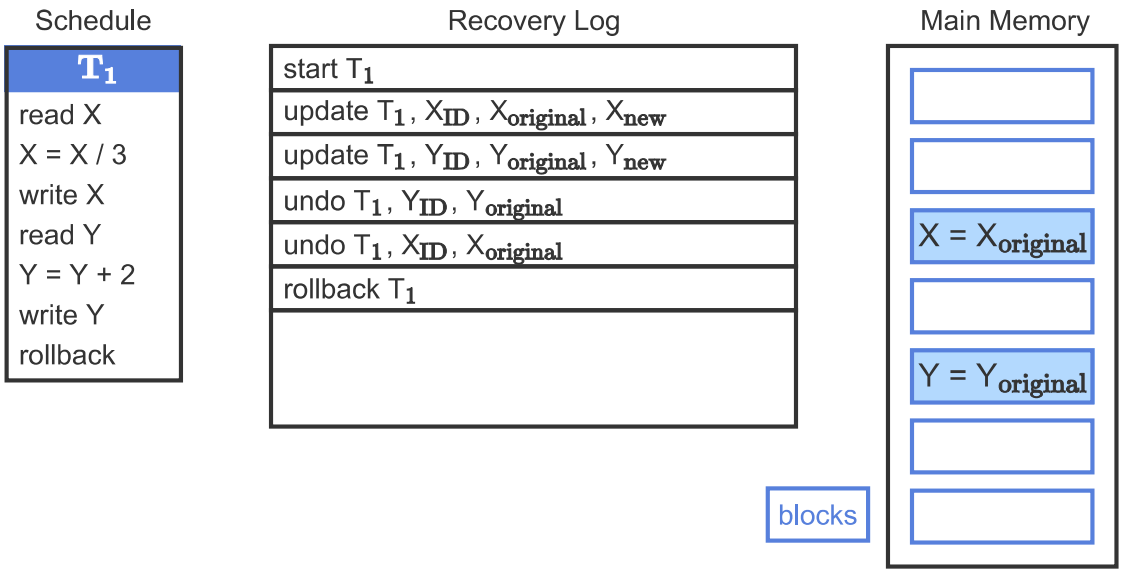
- Restores data D to original value V in the update record.
- Writes a compensation record for T, D, and V to the end of the log.

Eventually, the recovery system reads the 'start T' transaction record and writes a 'rollback T' record to the end of the log. At this point, the rollback is complete.

The recovery system reads the log backwards because data must be restored in reverse order of transaction operations.

PARTICIPATION ACTIVITY 7.4.5: Recovery from transaction failure.

1 2 3 4 5 6 2x speed



Finally, the rollback record is written to the log, and rollback is complete

Captions

1. As the transaction executes, a start record is written to the recovery log.
2. Write statements generate update records in the log and save data in main memory.
3. When the system or programmer initiates rollback, the recovery system reads the log in reverse.
4. Y is restored to original value in main memory, and a compensation record is written to the log.
5. X is restored to original value in main memory, and a compensation record is written to the log.
6. Finally, the rollback record is written to the log, and rollback is complete

Feedback?

PARTICIPATION ACTIVITY 7.4.6: Recovery from transaction failure.

Refer to the schedule below :

T_1 reads X
T_2 reads Y
T_2 writes Z
T_1 writes Y
T_2 writes X
deadlock

Deadlock occurs because T_1 is waiting for an exclusive lock on Y, and T_2 is waiting for an exclusive lock on X. When deadlock occurs, the database rolls back T_2 and completes T_1.

Order the log records to match the schedule.

start T_1	Log record 1 T_1 starts before T_2. Whenever a transaction starts, a start record is written.	**Correct**
start T_2	Log record 2 T_2 starts after T_1 and another start record is written.	**Correct**
update T_2, Z . . .	Log record 3 Reads do not generate log records. The third log record corresponds to T_2 writes Z.	**Correct**
undo T_2, Z . . .	Log record 4 Due to deadlock, T_1 cannot write Y and T_2 cannot write X. The database detects deadlock and rolls back T_2, the last transaction to start. The rollback restores the original Z value and generates an undo record.	**Correct**
rollback T_2	Log record 5 After the original Z value is restored, the rollback of T_2 is complete, marked by a rollback record.	**Correct**
	Log record 6	**Correct**

update T_1, Y . . .	After the T_2 rollback is complete, T_1 is granted an exclusive lock on Y and writes Y. The write generates an update record.	
commit T_1	Log record 7 T_1 runs to completion, marked by a commit record.	**Correct**

Reset

Feedback?

Recovery from system failure

Recovery from a system failure has two phases. The ***redo phase*** restores all transactions that were committed or rolled back since the last checkpoint. The ***undo phase*** rolls back transactions that were neither committed nor rolled back.

In the **redo phase**, the recovery system reads the recovery log **forward** from the latest checkpoint record. While reading the log, the recovery system maintains a list of active transactions. The list is initialized with active transactions in the checkpoint record. As each log record is read:

- Transactions in a 'start' transaction record are added to the list.
- Transactions in a 'commit' or 'rollback' transaction record are removed from the list.
- Data in an update record is set to the new value.
- Data in a compensation record is restored to the original value.

The above redo process performs unnecessary work on rolled-back transactions. The process changes data in update records and later restores the same data in compensation records. In principle, the recovery system could avoid unnecessary work by ignoring all rolled-back transactions. Processing all log records is simpler than ignoring rolled-back transactions, however, and is thus commonly implemented.

At the end of the redo phase, all transactions remaining on the list have no 'commit' or 'rollback' records. These transactions are rolled back in the undo phase.

During the **undo phase**, the recovery system reads the recovery log **backwards** from the last record. While reading the log, the recovery system maintains a list of incomplete transactions. The list is initialized with active transactions remaining from the redo phase. Each transaction on the list is rolled back:

- When an update record is read, data is restored to the original value, and a compensation record is written at the end of the log.

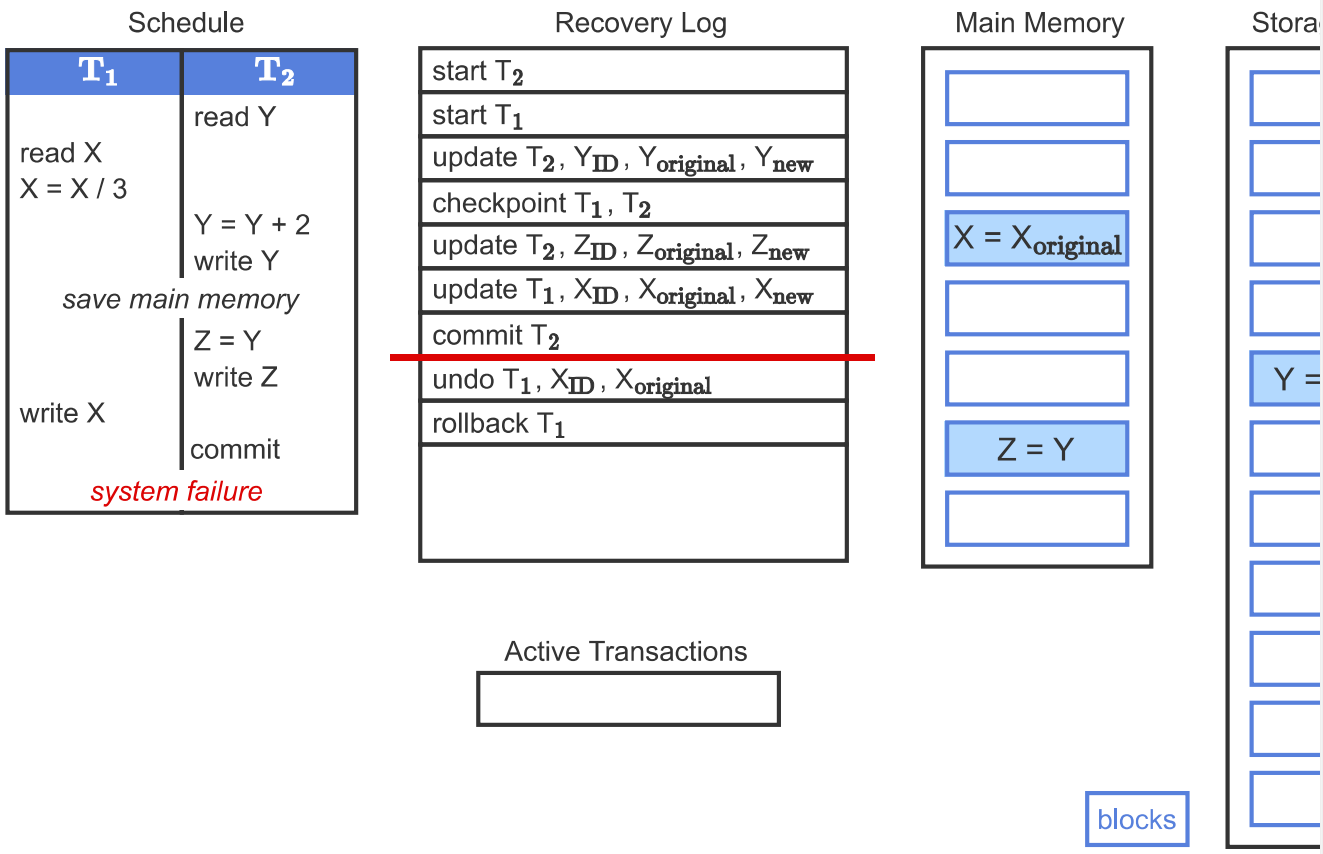
- When the 'start' transaction record is read, the transaction is removed from the list, and a 'rollback' transaction record is written at the end of the log.

The above process is the same as the rollback process for transaction failures, described above.

When the transaction list is empty, the undo phase ends, and recovery is complete.

PARTICIPATION ACTIVITY 7.4.7: Recovery from system failure.

1 2 3 4 5 6 7 2x speed



The system reads the 'start T_1' record in the log, writes a rollback record and deletes T_1 from the Recovery is complete.

Captions

1. The schedule executes normally. The checkpoint record indicates transactions T_1 and T_2 are active.
2. The system fails, and memory is lost.
3. The recovery system initializes the list of active transactions from the last checkpoint.
4. During the redo phase, the system reads the log forward from the checkpoint and restores data from update records.
5. The commit record in the log removes T_2 from the transaction list.

6. During the undo phase, the system reads the log backwards and rolls back changes made by transactions in the list.
7. The system reads the 'start T_1' record in the log, writes a rollback record and deletes T_1 from the list. Recovery is complete.

Feedback?

PARTICIPATION ACTIVITY 7.4.8: Recovery from system failure.

1) During the **redo** phase, what log records remove a transaction from the active transaction list?
   - Commit records only
   - Rollback records only
   - (•) Both commit and rollback records

Correct

Either a commit or rollback record marks the end of a transaction in the log. When a transaction ends, the transaction is removed from the active list.

2) During the **redo** phase, what log records generate a database write?
   - Update records only
   - Compensation (undo) records only
   - (•) Both update and compensation (undo) records

Correct

The redo phase 'replays' all actions of completed transactions. Completed transactions include rollbacks, so both update and compensation records are processed.

3) During the **undo** phase, the recovery system reads the log in reverse and stops at:
   - The most recent checkpoint record
   - (•) The start record for the last transaction

Correct

The recovery system continues reading until the active transaction list is empty. The list is empty when the start record for the last transaction is read.

in the active transaction list

- The first start record after the most recent checkpoint

4) During the **undo** phase, the recovery system writes compensation records for:

- Transactions that do not commit or roll back following the most recent checkpoint
- All transactions listed in the most recent checkpoint
- All transactions that roll back following the most recent checkpoint

Correct

Transactions that do not have a commit or rollback record after the checkpoint remain on the active transaction list at the start of the undo phase.

Feedback?

CHALLENGE ACTIVITY 7.4.1: Recovery.

379958.2369558.qx3zqy7

Jump to level 1

1 2 3 4

The database experiences a system failure.

Schedule

T_1	T_2
	read Y
read X	
X = X * 4	
	Y = Y - 8

After the system fails, what is the first record added to the recovery log?

undo T_2, ZID, Zoriginal

	write Y

save main memory

	Z = Y
write X	
	write Z
Z = X	

system failure

1 2 3 **4**

Check Next **Done**. Click any level to practice more. Completion is preserved.

✔ Expected: undo T_2, Z_{ID}, $Z_{original}$

Undo records are added for all writes that were neither committed nor rolled back prior to system failure. Since `write Z` in T_2 is the last uncommitted write, 'undo T_2, Z_{ID}, $Z_{original}$' becomes the first record added to the log.

Feedback?

Recovery from storage media failure

Availability is the percentage of time a system is working from the perspective of the system user. Many databases require high availability, in excess of 99% of the time. Ex: A database that manages stock exchange trades must have availability approaching 100%.

Availability is a primary concern in recovery from storage media failures.

In principle, recovery from storage media failure might be the same as recovery from system failure, starting at the beginning of the recovery log. The log contains all database operations, so the database may be fully restored from scratch. In practice, however, this approach is not feasible. The log of all database history is exceedingly large, so recovery time is exceedingly long and availability exceedingly low.

Two recovery techniques are commonly used: cold backup and hot backup.

The ***cold backup*** technique periodically creates checkpoints and, while transaction processing is paused, copies the database to backup media. The backup media might be tapes or disk

drives, normally stored in a separate location from the primary database. Typically, checkpoints and backups are taken hourly or daily when database activity is low.

When storage media fails, the recovery system:

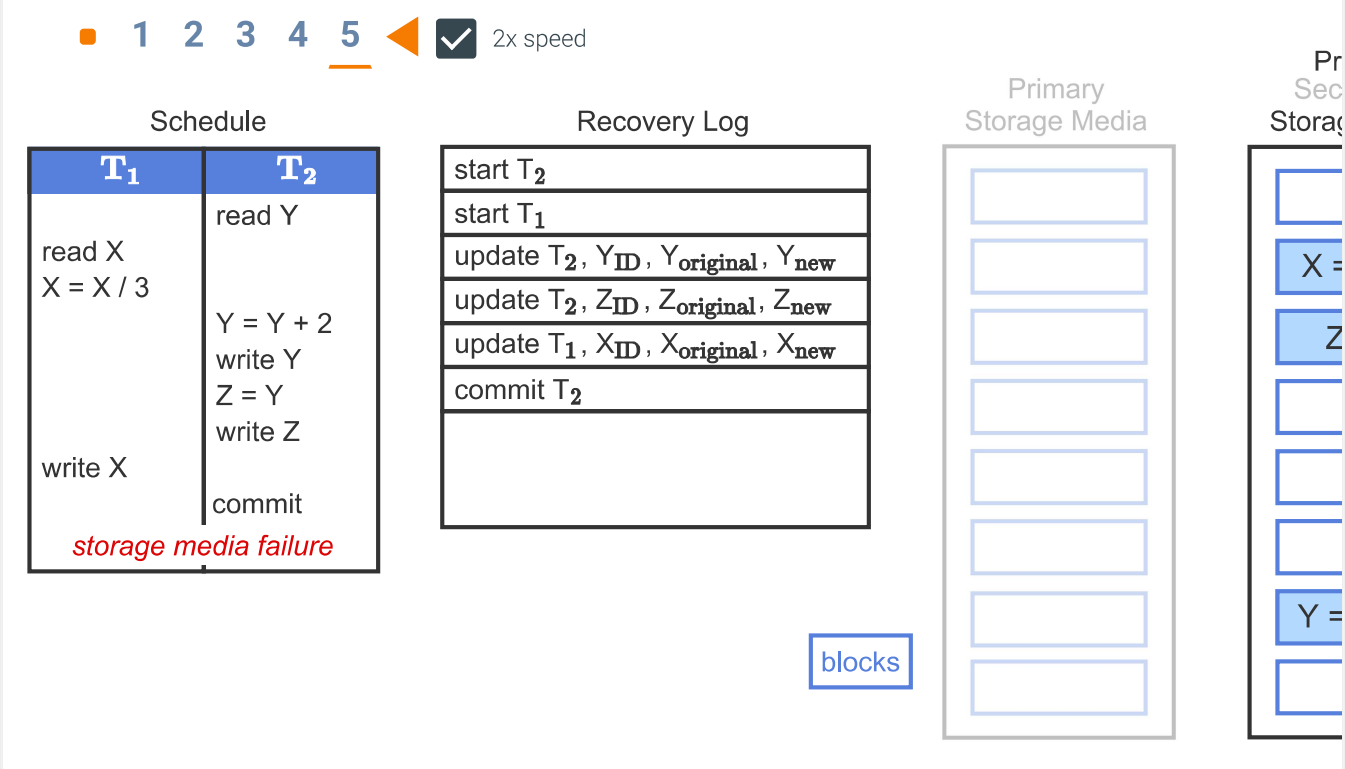
1. Copies the latest backup to the database.
2. Executes the system failure recovery process beginning at the latest checkpoint.

Depending on the backup size and period length, recovery might require minutes or hours. The database is unavailable during recovery, so the cold backup technique is used when low availability is acceptable.

The ***hot backup*** technique maintains a secondary database that is nearly synchronized with the primary database. As the primary database processes transactions, log records are sent to and processed by the secondary database. If the databases share a high-speed connection, the secondary database is only moments behind the primary database.

When storage media for the primary database fails, the secondary database becomes primary. This can be accomplished, for example, by swapping the internet addresses of primary and secondary servers. The new primary database is available in a matter of seconds, as soon as outstanding log records are processed.

PARTICIPATION ACTIVITY 7.4.9: Hot backup technique.



The secondary database is now synchronized and becomes the primary database.

Captions

1. As transactions are executed, log records are sent to a secondary database.
2. The secondary database may be in a remote location. Updates to the secondary database may be slightly delayed.
3. The primary database fails.
4. After a momentary delay, the secondary database receives the remaining log records.
5. The secondary database is now synchronized and becomes the primary database.

Feedback?

PARTICIPATION ACTIVITY 7.4.10: Recovery from storage media failure.

1) Database availability is:

- The percentage of time that a database is working properly.
- The number of seconds per day that a database is responsive to application programs.
- **The percentage of time that a database is responsive to application programs.**

Correct

Database availability reflects overall responsiveness of the database and is measured as a percentage.

2) With a cold backup, recovery from storage media failure reads the log:

- From the beginning.
- **From the latest checkpoint.**
- From the latest commit record.

Correct

In a cold backup, a copy of the database is made at each checkpoint. During recovery, the database restores the copy and then reads the log starting at the checkpoint.

3) With a hot backup, as a transaction executes against the primary database:

- (●) Log records are sent to the secondary database.
- ( ) The secondary database is updated synchronously, when the transaction commits or rolls back.
- ( ) The operating system synchronizes the secondary database.

Correct

Sending log records to the secondary database ensures the secondary database is only moments behind the primary database, without delaying transaction processing.

Feedback?

How was this section? Provide feedback