Recuperación de Bases de Datos

M. Andrea Rodríguez-Tastets

Universidad de Concepción,Chile
www.inf.udec.cl\ ∼andrea
andrea@udec.cl

II Semestre - 2012

Concepto de Recuperación

- La recuperación de fallos a transacciones casi siempre equivale a restaurar el valor de los datos al estado consistente más reciente antes de la falla.
- Por lo general, para lograr la restauración, el sistema maneja el diario del sistema o log.

Concepto de Recuperación: Movimientos de disco a memoria (1/2)

- Una o más páginas de disco que contienen los elementos de información que han de actualizarse se colocan en búfers de memoria principal y luego se actualizan en memoria antes de ser escritos otra vez en disco.
- Por lo general, hay un grupo de búfers dentro de la memoria, llamados caché del SGBD.
- Se utiliza un directorio de la caché para seguir la pista de los elementos de la base de datos que están en los búfers.
- Puede ser necesario reemplazar o limpiar algunos de los búfers de la caché para disponer de espacio para un nuevo elemento. Para ello, se usa alguna técnica de reemplazo de páginas: el menos recientemente usado (LRU), o la primera en entrar primera en salir (FIFO).

Concepto de Recuperación: Movimientos de disco a memoria (2/2)

- Cada búfers tiene un bit de ensuciado para indicar si el búfer ha sido o no modificado.
- Cuando se lee por primera vez una página de disco, se asigna la página a un búfer y se coloca el bit ensuciado en 0. Solo se reemplaza la información en el disco, si es que el bit de ensuciado cambia a 1.
- Se usa otro bit de reserva que indica en 1 que una página aún no se puede escribir en disco.
- Para grabar un búfer en disco:
  1. actualización en el lugar: escribe en la misma posición en disco
  2. sombreado: escribe en una posición diferente, lo que hace posible manejar diferentes versiones.

Concepto de Recuperación: estratégica básica

- Si hay daños extensos, entonces se restaura una copia de la base de datos almacenada en disco.
- Si el daño no es extenso (no hay daño físico) pero hay inconsistencia como las descritas en el manejo de transacciones, la estrategia es invertir los cambios que provocaron la inconsistencia, deshaciendo algunas operaciones. Para esto, las entradas mantenidas en el diario del sistema on-line (log) se consultan durante la recuperación.
- Se distinguen las siguientes técnicas de recuperación para fallos no catastróficos:
  1. actualización diferida
  2. actualización inmediata
  3. paginación en sombra

Estratégica básicas

- Actualización diferida: No actualiza hasta que la transacción lleva a su punto de confirmación. Si una transacción falla, no hay necesidad de DESHACER porque no se ha modificado la base de datos. Pero puede ser necesario REHACER.
- Actualización inmediata: Es posible actualizar la BD antes de la confirmación de una transacción. Sin embargo, estas operaciones se graban en el log mediante escritura forzada antes de aplicarse a la base de datos. Puede ser necesario DESHACER y REHACER.
- Paginación en sombra: Se basa en un directorio a las páginas de la BD. El directorio se copia cuando comienza una transacción, pasando a ser un directorio sombra.

Escritura del log

- Cuando se utiliza actualización en el lugar, es necesario utilizar un log para la recuperación. Esto se conoce como escritura anticipada en el disco.
- Los dos tipos de información que maneja el log son:
  1. información necesaria para deshacer: valor antiguo del elemento.
  2. información necesaria para rehacer: nuevo valor del elemento escrito

Robar/forzar (1/2)

- La estrategia no-robar establece que una página de la caché actualizada por una transación no puede escribirse a disco antes de la confirmación de dicha transacción. El bit de reserva indica si la página puede o no escribirse a disco.
- Si el protocolo permite escribir antes de confirmar, se le llama estrategia robar. El robar se usa cuando el gestor de la caché necesita un búfer vacío.
- Si todas las páginas actualizadas por una transacción son escritas inmediatamente a disco cuando se confirma la transacción, se llama estrategia forzar. De lo contrario se llama no-forzar.

Robar/forzar (2/2)

- Un esquema recuperación de actualización diferida sigue un estrategia de no robar.
- Los sistemas tradicionales usan una estrategia robar/no forzar ya que esto permite el manejo de búfers más pequeños que no contengan toda una transacción.
- La idea de no forzar es útil para que una página actualizada de una transacción ya confirmada pueda estar en el búfer y ser usada por otra transacción.
- Para facilitar el trabajo del SGBD, se usan listas de transacciones activas, listas de transacciones confirmadas y listas de transacciones abortadas.

Puntos de control

- Otro tipo de entrada en el log es el llamado punto de control.
- Se escribe en el diario un punto de control en que el sistema escribe en el disco todos los búfers del SGBM que han sido modificados en la base de datos.
- Todas las transacciones confirmadas antes del punto de control no necesitan rehacer sus actualizaciones.
- Establecer un punto de control involucra:
  - suspensión de la ejecución de transacciones temporalmente
  - escritura forzada de todos los búfers de la memoria principal que han sido modificados a disco
  - Reescribir un registro (punto de control) al log y escritura forzada del log en disco
  - Reactivar las transacciones en ejecución.

Rollback: Restauración de transacciones

- Si una transacción T se revierte, cualquier transacción S que haya leído mientras tanto el valor de algún elemento escrito por T también debiera revertirse. Esto en forma de cascada.
- Como la recuperación es cascada es costosa, todas las estrategias de recuperación tienden a que esta recuperación en cascada no sea necesaria.

Rollback: Ejemplo

T_1	T_2	T_3
$R(A)$	$R(B)$	$R(C)$
$R(D)$	$W(B)$	$W(B)$
$W(D)$	$R(D)$	$R(A)$
	$W(D)$	$W(A)$

Operación	old	new
[Inicio T_3]		
$[R_3(C)]$	40	
*$[W_3(B)]$	15	12
[Inicio T_2]		
$[R_2(B)]$	12	
**$[W_2(B)]$	12	18
[Inicio T_1]		
$[R_1(A)]$	30	
$[R_1(D)]$	20	
$[W_1(D)]$	20	25
$[R_2(D)]$	25	
** $[W_2(B)]$	25	26
$[R_3(A)]$	30	
Caida del sistema		

Rollback: Restauración de transacciones

- En la práctica la restauración en cascada de transacciones nunca es necesaria porque en los métodos de recuperación garantizan planes sin cascada o estrictos.
- Por lo tanto, no se necesita grabar cuando se lee un elemento en el log, porque estas operaciones son sólo necesarias para determinar la restauración en cascada.

Técnica de Recuperación: actualización diferida

- Una transacción no puede modificar la base de datos en disco antes de llegar a su punto de confirmación.
- Una transacción no llega a su punto de confirmación antes de grabar todas sus operaciones de actualización en el diario y forzar la escritura del diario en disco.

Actualización diferida: monousuario

- Usar dos listas de transacciones: confirmadas desde el último punto de control activas (solo una en este tipo de sistemas). Aplicar la operación REHACER de todas las operaciones de escritura de las transacciones confirmadas a partir del log. Reiniciar operaciones activas.
- Rehacer una operaciónde escritura OP-ESCRITURA consiste en examinar su entrada de diario y asignar el nuevo valor en la base de datos.

Actualización diferida: multiusario (concurrencia)

- En muchos casos el control de concurrencia está interrelacionado con los procesos de recuperación.
- Asuma un control de concurrencia en dos fases donde todos los desbloqueos se hacen en el commit. La recuperación usa el procedimiento descrito anteriormente.
- Se puede optimizar en el caso que hayan más de una actualización confirmada desde el último punto de control. En tal caso, bastará con REHACER la última actualización.

Ejemplo

T_1	T_2
$R(A)$	$R(B)$
$R(D)$	$W(B)$
$W(D)$	$R(D)$
	$W(D)$

T_3	T_4
$R(A)$	$R(B)$
$W(A)$	$W(B)$
$R(C)$	$R(A)$
$W(C)$	$W(A)$

Operación	new
[Inicio T_1]	
[$W_1(D)$]	20
[Confirmar T_1]	
[Punto de control]	
[Inicio T_4]	
[$W_4(B)$]	15
[$W_4(A)$]	20
[Confirmar T_4]	
[Inicio T_2]	
[$W_2(B)$]	12
[Inicio T_3]	
[$W_3(A)$]	30
[$W_2(D)$]	25
Caida del sistema	

Técnica Recuperación: actualización inmediata

- Una operación de actualización se realiza inmediatamente
- Las actualizaciones se graban en el log antes que a la base de datos
- Deben DESHACERSE operaciones de actualización que se hayan realizado por una transacción que falla.
- Dos categorías de algorimos son:
  - Si todas las actualizaciones se graban en un disco antes de confirmar, entonces nunca necesita rehacer transacciones confirmadas
  - Si se permite confirmar antes de escribir los cambios, entonces tenemos el caso más general de DESHACER/REHACER

Procedimiento DESHACER/REHACER: monousuario

Se utiliza en algoritmo de REHACER que definimos antes y el siguiente procedimiento de DESHACER.

- Usar dos listas: transacciones confirmadas desde el punto de control y transacciones activas
- Deshacer todas las operaciones escribir-elemento de las transacciones activas a partir del log usando el siguiente procedimiento DESHACER.

  Deshacer una operación de escritura consiste en examinar su entrada en el log y asignar el valor del elemento antiguo a la base de datos. La anulación de las operaciones de escribir de una o más transacciones en el log debe hacerse en el sentido inverso a aquel en el que aparecen en el log.
- Rehacer todas las operaciones de escrbir de las transacciones confirmadas a partir del log en el orden en que aparecen en el log.

Procedimiento DESHACER/REHACER: multiusuario

Supongamos nuevamente un control de concurrencia de 2 fases donde los desbloqueos ocurren al final.

- Usar dos listas: transacciones confirmadas y transacciones activas
- Deshacer todas las operaciones escribir-elemento de las transacciones activas (no confirmadas) mediante el procedimiento DESHACER.
- Rehacer las operaciones escribir-elemento de transacciones confirmadas a partir del log.

Paginación en la sombra

- En este esquema sólo requiere un diario en un ambiente multiusuario.
- Se mantiene un directorio actual que apunta a las páginas de las bases de datos.
- Al comenzar un transacción se hace la copia del directorio actual en disco a un directorio sombra. El directorio sombra se guarda mientras la transacción usa el directorio actual.
- Las modificaciones se hacen sobre una copia de la BD y se manejan versiones, siendo el directorio actual el que es modificado.
- Para recuperar, basta con liberar las páginas modificadas y desechar el directorio actual.

Algoritmo ARIES (1/3)

- ARIES: conjunto de algoritmos de recuperación usados actualmente en Sistemas de Base de Datos
- Aries utiliza un esquema de robar/no forzar para a escritura basado en tres conceptos: (1) escritura anticipada, (2) repetición de la historia durante el rehacer, y (3) anotación en el log de las modificaciones durante el rehacer.
- Repetición de la historia: ARIES vuelve a trazar todas las acciones del sistema de la base de datos antes de la caída para reconstruir el estado de la base de datos cuando ocurrió la caída. Las transacciones que no estaban confirmadas en el momento de la caída se deshacen.
- Anotación en el log de las modificaciones durante el rehacer evita que ARIES repita las operaciones de DESHACER realizadas si se produce un fallo durante la recuperación que oblique reiniciar el proceso de recuperación.

Algoritmo ARIES (2/3)

- El procedimiento de ARIES consiste de tres pasos: (1) Análisis de páginas modificadas en el búfer y conjunto de transacciones activas. (2) REHACER. (3) DESHACER.
- ARIES necesita el log, la tabla de transacciones y la tabla de páginas sucias. Además utiliza un registro maestro (master log) que maneja los comienzos de los puntos de control.

Algoritmo ARIES (3/3)

- Análisis identifica las páginas sucias (actualizadas) en el búfer y el conjunto de transacciones activas en el momento de caída. También se determina el punto apropiado del diario donde debería inicarse las operación REHACER.
- La fase de REHACER usa cierta información del diario para determinar dónde comenzar el REHACER. Además en esta fase el diario indica si la operación ya ha sido realizada y si se necesita rehacer. Durante una recuperación sólo se rehace lo que se necesita.
- La fase DESHACER escanea hacia atrás y deshace las operaciones de las transacciones activas

Diario en ARIES

- El diario maneja un número secuencial de diario (NSD) que corresponde a cada acción de una transacción. Se escribe un NSD para cada acción de: actualizar, confirmar,abortar, deshacer, y finalizar.
- Los campos comunes del diario incluyen: NSD anterior de la transacción, ID de la transacción, y el tipo de registro del diario.
- Para los registros de actualización, también se incluye: ID de la página, la longitud del elemento, desplazamiento desde el comienzo de la página, imagen antes del elemento (valor anterior), imagen posterior (valor posterior).

Algoritmo ARIES: Log del sistema

NSD	ULTIMO_NSD	ID_TRAN	TIPO	ID_PAG	OTRA INF.
1	0	T_1	Actualizar	C	....
2	0	T_2	Actualizar	B	....
3	1	T_1	Confirmar		
4	Punto dde control inicial				
5	Punto de control final				
6	0	T_3	Actualizar	A	...
7	2	T_2	Actualizar	C	....
8	7	T_2	Confirmar		

Tabla de transacciones y de páginas

- La tabla de transacciones tiene una entrada por cada transacción activa, con el id de la transacción, el estado de la transacción y el NSD del registro del diario más reciente.
- La tabla de página contiene el ID de la página y la primera entrada del diario que haya hecho modificación a la página.

Algoritmo ARIES: Tabla de transacciones y páginas

Tabla de Transacciones al momento del punto de control

ID_TRAN	ULTIMO_NSD	ESTADO
T_1	3	Confirmar
T_2	2	en proceso

Tabla de Páginas al momento del punto de control

ID_PAGINA	NSD
C	1
B	2

Algoritmo ARIES: Tabla de transacciones y páginas

Tabla de Transacciones después del análisis

ID_TRAN	ULTIMO_NSD	ESTADO
T_1	3	Confirmar
T_2	7	Confirmar
T_3	6	en proceso

Tabla de Páginas después del análisis

ID_PAGINA	NSD
C	1
B	2
A	6

ARIES: REHACER

- ARIES comienza rehaciendo desde el punto del diario donde sabe que los cambios previos a las páginas sucias han sido ya aplicados a la base de datos en disco.
- ARIES determina esto encontrando el NSD más pequeño, M, de todas las páginas sucias de la tabla de página sucias

ARIES: DESHACER

- La tabla de transacciones identifica las transacciones activas que deben deshacerse.
- la fase comienza de atrás hacia adelante deshaciendo las transacciones activas.

Algoritmo ARIES: Ejercicio

NSD	LOG
00	comienza punto de control
10	finaliza punto de control
20	T_1 escribe P1
30	T_2 escribe P2
40	T_3 escribe P3
50	T_2 commit
60	T_3 escribe P2
70	T_2 finaliza
80	T_1 escribe P5
90	T_3 aborta
	Cae el sistema

Algoritmo ARIES: Ejercicio (preguntas)

1. ¿ Cuál es el NSD almacenado en el registro maestro?
2. ¿ Qué se hace durante la fase de análisis?
3. ¿ Qué se hace durante la fase de REHACER?
4. ¿ Qué se hace durante la fase de DESHACER?

Algoritmo ARIES: Ejercicio (respuestas)

1. 00, ya que es donde comienza el punto de control
2. Durante el análisis sucede:

NSD	ACCION
20	Agregar $(T_1, 20)$ a TT (tabla de transacciones) y $(P1, 20)$ a TP(Tabla de páginas sucias)
30	Agregar $(T_2, 30)$ a TT y $(P2, 30)$ a TP
40	Agregar $(T_3, 40)$ a TT y $(P3, 40)$ a TP
50	Cambia estado de T_2 a commit
60	Cambia $(T_3, 40)$ a $(T_3, 60)$ en TT
70	Remueve T_2 de TT
80	Cambia $(T_1, 20)$ a $(T_1, 70)$ en TT y agrega $(P5, 80)$ a TP
90	No hace nada

Algoritmo ARIES: Ejercicio (respuestas cont.)

3. Comienza el REHACER desde el NSD 20, mínimo de los NSD en la TP:

NSD	ACCION
20	Rehacer el cambio en P1
30	Chequea si P2 tiene una NSD de escritura a disco mayor que 10 o no. Si es una transacción confirmada, probablemente no necesitemos rehacerla
40	Rehacer el cambio en P3
50	No hace nada
60	Rehacer cambio en P2
70	No hace nada
80	Rehacer cambio en P5
90	No hace nada

Algoritmo ARIES: Ejercicio (respuestas cont.)

4. El DESHACER consiste de $(80, 60)$, ambas son NSD de transacciones activas

NSD	ACCION
80	Deshaga cambios en P5, agrega a registro de deshacer: Deshace T_1 NSD 80, setée undoNextLSN=20, Agrege 20 a la lista de deshacer : $(80, 20)$.
60	Deshaga cambios en P2, agrega a registro de deshacer: Deshace T_3 NSD 60, setée undoNextLSN=40, Agrege 40 a la lista de deshacer : $(40, 20)$.
40	Deshaga cambios en P3, agrega a registro de deshacer: Deshace T_3 NSD 40, T_3 termina.
20	Deshaga cambios en P1, agrega a registro de deshacer: Deshace T_1 NSD 20, T_1 termina.

Ejercicio

Considere los siguientes protocolos de control: 2 fases básico, 2 fases estricto y marcación de tiempo. Para la siguiente secuencias de acciones indique cómo cada uno de los mecanismos de control podría manejar la secuencia:

1. T1:R(X), T2:W(X) T2:W(Y), T3:W(Y), T1:W(Y) T1: Commit,T2:Commit, T3:Commit

Ejercicio: 2 fases básico

T1	T2	T3
S(X)		
R(X)		
		X(Y)
		W(Y)
		D(Y)
		Commit
X(Y)		
D(X)		
	X(X)	
	W(X)	
W(Y)		
D(Y)		
Commit		
	X(Y)	
	W(Y)	
	D(X)	
	D(Y)	
	Commit	

Ejercicio: 2 fases estricto

T1	T2	T3
S(X)		
R(X)		
		X(Y)
		W(Y)
		D(Y)
		Commit
X(Y)		
W(Y)		
D(X)		
D(Y)		
Commit		
	X(X)	
	W(X)	
	X(Y)	
	W(Y)	
	D(X)	
	D(X)	
	Commit	

Ejercicio: marca de tiempo

Asumamos inicialmente
MT(T1) = 1, MT(T2) = 2, MT(T3) = 3
y las siguientes marcas de tiempo inciales para los elementos X e Y:

Elemento	MT-lectura	MT-escritura
X	0	0
Y	0	0

La secuencia de valores de marca de tiempo para los elementos varia segun las instrucciones de la siguientes manera:

- T1:R(X)

Elemento	MT-lectura	MT-escritura
X	1	0
Y	0	0

- T2:W(X)

Elemento	MT-lectura	MT-escritura
X	1	2
Y	0	0

- T2:W(Y)

Elemento	MT-lectura	MT-escritura
X	1	2
Y	0	2

- T3:W(Y)

Elemento	MT-lectura	MT-escritura
X	1	2
Y	0	3

Ejercicio: marca de tiempo (cont.)

- T1:W(Y)

  No puede y debe reinicializar T1. Debido a que T1 no ha escrito nada antes, no afecta las otras transacciones y solo ella debe ser reinicializada con un MT(T1) = 4.

Elemento	MT-lectura	MT-escritura
X	4	2
Y	0	4

En resumen, al final las ejecucion se lleva a cabo de la siguiente forma:

T1	T2	T3
	W(X)	
	W(Y)	
		W(Y)
R(X)		
W(Y)		
commit		
	commit	
		commit

Ejercicio: recuperación

Asuma ahora la siguiente secuencia del diario (log) con el protocolo de concurrencia de 2 fases estricto y analice cómo se aplica el algoritmo de recueración ARIES. Asuma que X e Y son páginas.

LOG	
00	$W_1(X)$
10	$W_3(Y)$
20	T_3 Commit
30	comienza punto de control
40	finaliza punto de control
50	Finaliza T_3
60	$W_1(Y)$
70	T_1 Commit
80	Finaliza T_1
90	$W_2(X)$
100	$W_2(Y)$
El sistema se cae	

Recuperación: Tabla de transacciones y páginas

Tabla de Transacciones al momento del punto de control

ID_TRAN	ULTIMO_NSD	ESTADO
T_1	00	en proceso
T_3	20	confirmada

Tabla de Páginas al momento del punto de control

ID_PAGINA	NSD
X	10 Y
20	

Ejercicio: recuperación (cont.)

Durante el análisis sucede:

NSD	ACCION
50	Remueve T_3 de TT
60	Cambia($T_1, 60$)
70	nada
80	Remueve T_1 de TT
90	agrega $(T_2, 90)$ a TT y $(X, 90)$ a TP
90	Camba $(T_2, 100)$ en TT

Ejercicio: recuperación (cont.)

Comienza el REHACER desde el NSD 10, mínimo de los NSD en la TP:

NSD	ACCION
50	No hace nada
60	Rehacer cambio en Y
70	No hace nada
80	No hace nada
90	Rehacer cambio en X
100	Rehacer cambio en Y

Ejercicio: recuperación (cont.)

El DESHACER consiste de (100), el último NSD de la única transacción activa T_2

NSD	ACCION
100	Deshaga cambios en Y, agrega a registro de deshacer: Deshace T_2 NSD 100, setée undoNextLSN=90, Agrege 90 a la lista de deshacer : (90).
90	Deshaga cambios en X, agrega a registro de deshacer: Deshace T_2 NSD 90, T_2 termina.

Ejercicio: recuperación (cont.)

Considere:

NSD	Acción
00	$R_1(X)$
10	$W_1(Y)$
20	$W_2(Y)$
30	$W_3(X)$
40	Commit T_3
50	Finaliza T_3
60	Abort T_1
70	Comienza punto de control
80	Finaliza punto de control
90	Commit T_2
100	Finaliza T_2
110	$R_1(X)$
120	$W_1(Y)$
130	$R_1(Y)$
Se cae el sistema	

1. Indique las trablas de transacciones y de páginas sucias al finalizar el punto de control
2. Indique qué sucede durante la fase de análisis del proceso de recuperación con las tablas de algoritmo ARIES
3. Indique los pasos que van ocurriendo en la fase de REHACER y DESHACER del algoritmo ARIES

Ejercicio: respuesta

1. Consideramos 2 tabla : TT(TransID,ultimoNSD) la tabla de transacciones y DPT(paginaID,recNSD) la tabla de páginas con conflicto (dirty page table). Para el momento del punto de copntrol las tablas tienen lo siguiente:

ID_TRAN	ULTIMO_NSD	ESTADO
T_1	60	abortada
T_2	20	en proceso

ID_PAGINA	NSD
X	30
Y	10

2. El análisis comienza con en comienzo del punto de control y hace:

   90: Nada
   100 Elimina T_2 de TT
   110: Nada
   120: Cambia (T_1,120)
   130: Nada

   La TT final tiene una entrada : (T1,120). La tabla DPT final tiene dos entradas: (X,30) y (Y,10).

Ejercicio: respuesta

3. La fase de Redo: Comienza en 10, las menor secuencia en al DPT

10	Rehace cambio Y
20	Rehace cambio Y
30	Rehace cambio X
40-110	Nada
120	Rehace cambio Y

4. Fase Undo: Se comienza con el undo en 120 y llega el final del deshace. Al terminar la recuperación sólo se ha perdido los cambios hechos por la transacción T_1.